

PEMROGRAMAN WEB BERORIENTASI OBJEK



CHARLES SITORUS

DAFTAR ISI

HALAMAN SAMPUL....i
DISKLAIMER (*DISCLAIMER*)....ii
KATA PENGANTAR....iii
DAFTAR ISI....iv
GLOSARIUM....ix
PETA KEDUDUKAN BUKU....xi
Peta Konsep : Pemrograman Berorientasi Obyek Kelas XI Semester 1....xii
BAB I PENDAHULUAN....1
A. Deskripsi....1
B. Prasyarat....2
C. Petunjuk Penggunaan....3
D. Tujuan Akhir....3
E. Kompetensi Inti Dan Kompetensi Dasar....4
F. Cek Kemampuan Awal....5
BAB II PEMBELAJARAN....6
A. Deskripsi....6
B. Kegiatan Belajar....6
1. Kegiatan Belajar 1 : Mengenal Pemrograman Berorientasi Obyek 6
a. Tujuan Pembelajaran....6
b. Uraian Materi....6
c. Rangkuman....14
d. Tugas....15
e. Test Formatif....17
f. Lembar Jawaban Test Formatif (LJ)....17
g. Lembar Kerja Siswa....20
2. Kegiatan Belajar 2 : Perangkat Lunak Pemrograman Berorientasi Obyek....21
a. Tujuan Pembelajaran....21
b. Uraian Materi....21
a. Rangkuman....39
b. Tugas....40
c. Test Formatif....42



d. Lembar Jawaban Test Formatif (LJ).....42
e. Lembar Kerja Siswa.....45
3. Kegiatan Belajar 3 : Perangkat Lunak Pemrograman Berorientasi Obyek.....46
BAB II.....46
ATURAN DAN DASAR PEMROGRAMAN BERORIENTASI OBYEK.....46
A. Deskripsi.....46
B. Kegiatan Belajar.....46
1. Kegiatan Belajar 3 :Dasar dan Aturan Pemrograman Berorientasi Obyek.....46
a. Tujuan Pembelajaran.....46
b. Uraian Materi.....47
c. Rangkuman.....56
d. Tugas.....57
e. Test Formatif.....59
f. Lembar Jawaban Test Formatif (LJ).....59
g. Lembar Kerja Siswa.....61
2. Kegiatan Belajar 4 :Dasar dan Aturan Pemrograman Berorientasi Obyek.....62
a. Tujuan Pembelajaran.....62
b. Uraian Materi.....62
c. Rangkuman.....68
d. Tugas.....69
e. Test Formatif.....71
f. Lembar Jawaban Test Formatif (LJ).....71
g. Lembar Kerja Siswa.....74
3. Kegiatan Belajar 5 :Dasar dan Aturan Pemrograman Berorientasi Obyek.....75
a. Tujuan Pembelajaran.....75
b. Uraian Materi.....75
c. Rangkuman.....82
d. Tugas.....82
e. Test Formatif.....85
f. Lembar Jawaban Test Formatif (LJ).....86
g. Lembar Kerja Siswa.....88
4. Kegiatan Belajar 6 :Dasar dan Aturan Pemrograman Berorientasi Obyek.....89
a. Tujuan Pembelajaran.....89

b. Uraian Materi....89
c. Rangkuman....93
d. Tugas....94
e. Test Formatif....96
f. Lembar Jawaban Test Formatif (LJ)....97
g. Lembar Kerja Siswa....98
5. Kegiatan Belajar 7 :Dasar dan Aturan Pemrograman Berorientasi Obyek....99
a. Tujuan Pembelajaran....99
b. Uraian Materi....99
c. Rangkuman....105
d. Tugas....106
e. Test Formatif....108
f. Lembar Jawaban Test Formatif (LJ)....108
g. Lembar Kerja Siswa....110
6. Kegiatan Belajar 7 :Konsep Class dan Obyek....111
a. Tujuan Pembelajaran....111
b. Uraian Materi....111
c. Rangkuman....116
d. Tugas....116
e. Test Formatif....119
f. Lembar Jawaban Test Formatif (LJ)....119
g. Lembar Kerja Siswa....121
7. Kegiatan Belajar 8 :Konsep Class dan Obyek....122
a. Tujuan Pembelajaran....122
b. Uraian Materi....122
c. Rangkuman....128
d. Tugas....129
e. Test Formatif....131
f. Lembar Jawaban Test Formatif (LJ)....132
g. Lembar Kerja Siswa....134
8. Kegiatan Belajar 9 :Konsep Class dan Obyek....134
a. Tujuan Pembelajaran....135
b. Uraian Materi....135



c. Rangkuman....141
d. Tugas....141
e. Test Formatif....144
f. Lembar Jawaban Test Formatif (LJ)....144
g. Lembar Kerja Siswa....145
9. Kegiatan Belajar 10 :Pembungkusan Data....147
a. Tujuan Pembelajaran....147
b. Uraian Materi....147
c. Rangkuman....153
d. Tugas....153
e. Test Formatif....155
f. Lembar Jawaban Test Formatif (LJ)....156
g. Lembar Kerja Siswa....158
10. Kegiatan Belajar 13 :Pewarisan....159
a. Tujuan Pembelajaran....159
b. Uraian Materi....159
c. Rangkuman....166
d. Tugas....167
e. Test Formatif....169
f. Lembar Jawaban Test Formatif (LJ)....169
g. Lembar Kerja Siswa....171
11. Kegiatan Belajar 13 :Pewarisan....172
a. Tujuan Pembelajaran....172
b. Uraian Materi....172
c. Rangkuman....173
d. Tugas....174
e. Test Formatif....176
f. Lembar Jawaban Test Formatif (LJ)....176
g. Lembar Kerja Siswa....178
12. Kegiatan Belajar 13 :Pewarisan....179
a. Tujuan Pembelajaran....179
b. Uraian Materi....179
c. Rangkuman....181

d. Tugas....182
e. Test Formatif....184
f. Lembar Jawaban Test Formatif (LJ)....184
g. Lembar Kerja Siswa....186
13. Kegiatan Belajar 16 :Polimorphisme....187
a. Tujuan Pembelajaran....187
b. Uraian Materi....187
c. Rangkuman....193
d. Tugas....194
e. Test Formatif....196
f. Lembar Jawaban Test Formatif (LJ)....196
g. Lembar Kerja Siswa....198
14. Kegiatan Belajar 18 :Package....199
a. Tujuan Pembelajaran....199
b. Uraian Materi....199
c. Rangkuman....201
d. Tugas....202
e. Test Formatif....204
f. Lembar Jawaban Test Formatif (LJ)....204
g. Lembar Kerja Siswa....206
Daftar Pustaka....207



GLOSARIUM

Abstraksi mengacu pada tindakan yang mewakili fitur penting tanpa termasuk rincian latar belakang atau penjelasan. Kelas menggunakan konsep abstraksi dan didefinisikan sebagai daftar atribut abstrak.

Accessor Methods digunakan untuk membaca nilai variabel pada class,baik berupa instance maupun static. Sebuah accessor method umumnya dimulai dengan penulisan get <namaInstanceVariable>.

Array adalah sebuah variabel/ sebuah lokasi tertentu yang memiliki satu nama sebagai identifier, namun identifier ini dapat menyimpan lebih dari sebuah nilai.

Encapsulation adalah mekanisme yang mengikat bersama-sama kode dan data dalam memanipulasi, dan membuat baik aman dari gangguan luar dan penyalahgunaan. Enkapsulasi merupakan penyimpanan data dan fungsi dalam satu unit (kelas). Sebuah antarmuka yang terdefinisi dengan baik mengontrol akses ke kode tertentu dan data. Data tidak bisa diakses oleh dunia luar dan hanya fungsi-fungsi yang disimpan dalam kelas dapat mengaksesnya.

Field merupakan tipe data yang didefiniskan, sementara method merupakan operasi.

Inheritance (pewarisan) adalah proses dimana satu objek mengakuisisi properti dari obyek lain. Ini mendukung klasifikasi hirarkis. Dengan menggunakan warisan, objek hanya perlu mendefinisikan kualitas-kualitas yang membuatnya unik dalam kelasnya. Hal ini dapat mewarisi atribut umum dari induknya. Sebuah sub - class baru mewarisi semua atribut dari super - class nya.

Java SDK adalah platform dasar java yang diperlukan agar komputer atau laptop dapat digunakan untuk mengeksekusi kode-kode program bahasa java.

Kelas adalah kumpulan objek dari jenis yang sama. Setelah kelas didefinisikan, sejumlah objek dapat dibuat yang termasuk ke kelas tersebut. Kelas adalah cetak biru, atau prototipe, yang mendefinisikan variabel dan metode umum untuk semua objek dari jenis tertentu.

Metode menggambarkan kemampuan objek . Anjing memiliki kemampuan untuk kulit. Jadi kulit ( ) adalah salah satu metode dari kelas Dog.

Mutator methods adalah *method* yang dapat memberi atau mengubah nilai variable dalam *class*,baik itu berupa *instance* maupun *static*.

Netbeans adalah aplikasi editor terpadu (IDE atau Integreted Development Environment) yang akan mempermudah dalam membuat aplikasi karena menyediakan control visual yang penting dalam pemrograman desktop (pemrograman visual).



Objek adalah entitas dasar run-time dalam suatu sistem berorientasi objek. Masalah pemrograman dianalisis dalam hal objek dan sifat komunikasi antara mereka. Ketika program dieksekusi, objek berinteraksi satu sama lain dengan mengirimkan pesan. Objek yang berbeda juga dapat berinteraksi satu sama lain tanpa mengetahui rincian data atau kode mereka.

Operator kondisi ?: adalah operator ternary. Berarti bahwa operator ini membawa tiga argumen yang membentuk suatu ekspresi bersyarat.

Operator precedence didefinisikan sebagai perintah yang dilakukan compiler ketika melakukan evaluasi terhadap operator, untuk mengajukan perintah dengan hasil yang tidak ambigu/ hasil yang jelas.

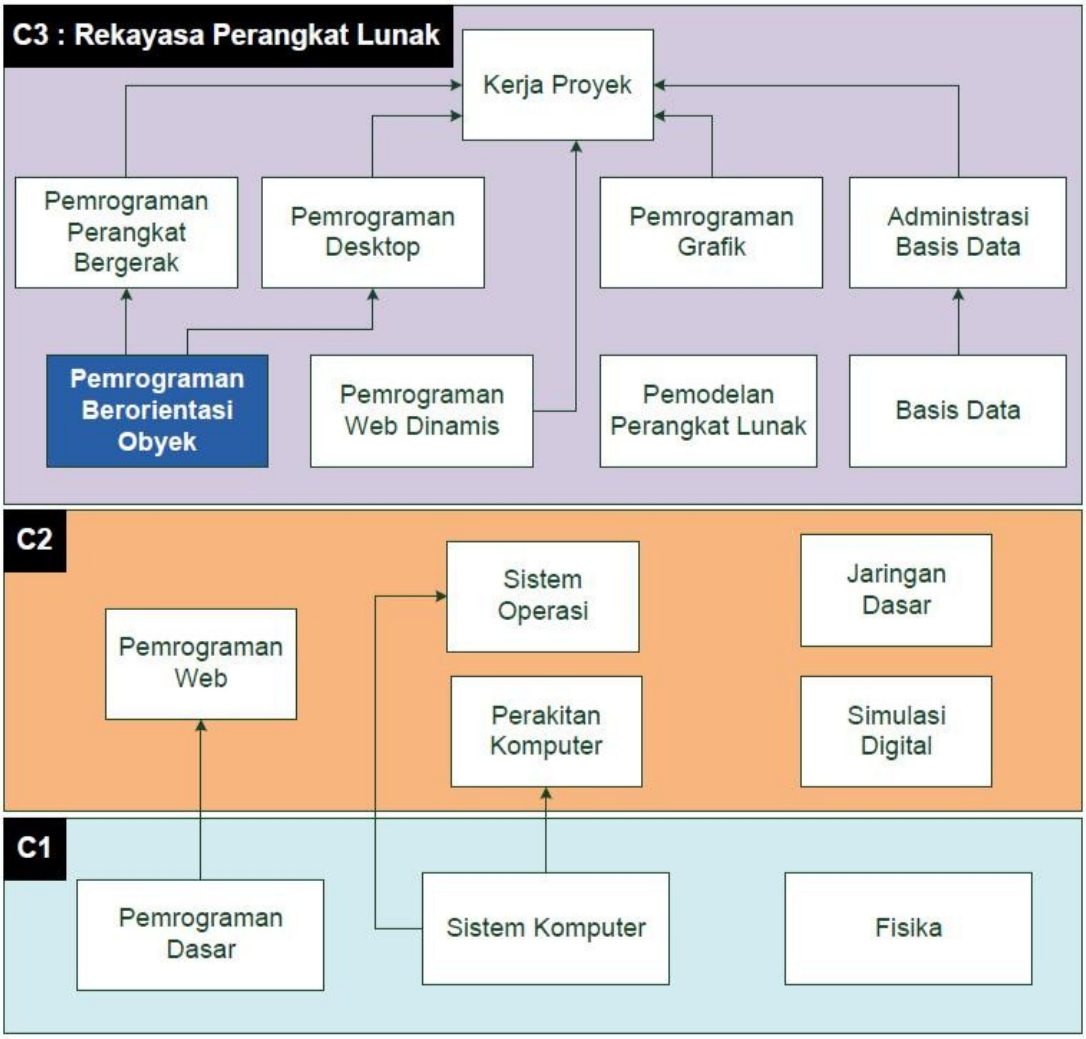
Pesan Passing adalah proses di mana sebuah objek mengirim data ke obyek lain atau meminta objek lain untuk memanggil sebuah metode . Message passing sesuai dengan " metode panggilan " .

Polimorfisme berarti kemampuan untuk mengambil lebih dari satu bentuk . Suatu operasi dapat menunjukkan perilaku yang berbeda dalam kasus yang berbeda . Perilaku tergantung pada jenis data yang digunakan dalam operasi.

Static adalah method yang dapat dipakai tanpa harus menginisialisasi suatu class (maksudnya tanpa menggunakan variabel terlebih dahulu).

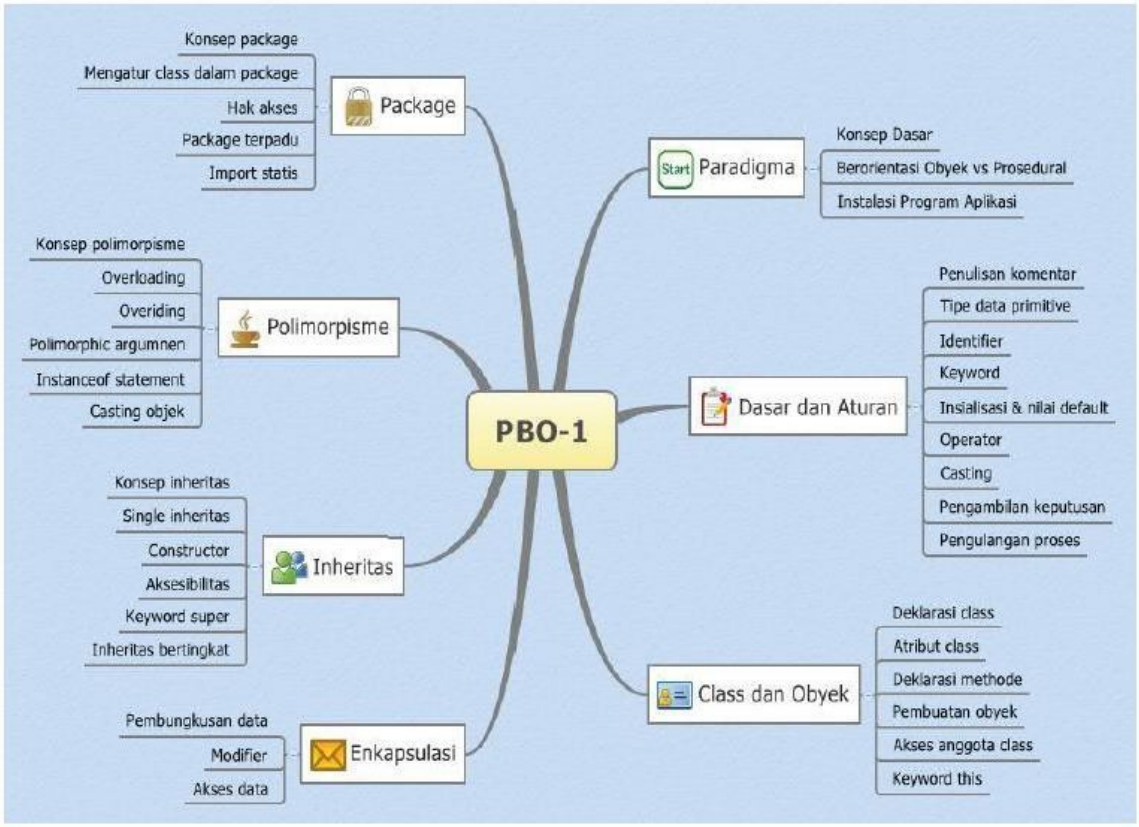
While loop adalah pernyataan atau blok pernyataan yang diulang-ulang sampai mencapai kondisi yang cocok.

PETA KEDUDUKAN BUKU



Keterangan	
C1	Kelompok mata pelajaran Dasar Bidang Keahlian Teknologi Informasi dan Komunikasi
C2	Kelompok mata pelajaran Dasar Program Keahlian Teknik Komputer dan Informatika
C3	Kelompok mata pelajaran Paket Keahlian Rekayasa Perangkat Lunak
	Mata pelajaran Pemrograman Berorientasi Obyek Semester
	1 Mata pelajaran prasyarat

Peta Konsep : Pemrograman Berorientasi Obyek Kelas XI Semester 1



Keterangan	
KD 3.1- 4.1	Paradigma Pemrgraman Berorientasi Obyek
KD 3.2- 4.2	Dasar dan Aturan
KD 3.3- 4.3	Class dan Obyek
KD 3.4- 4.4	Enkapsulasi
KD 3.5- 4.5	Inheritas
KD 3.6- 4.6	Polimopisme
KD 3.7- 4.7	Package

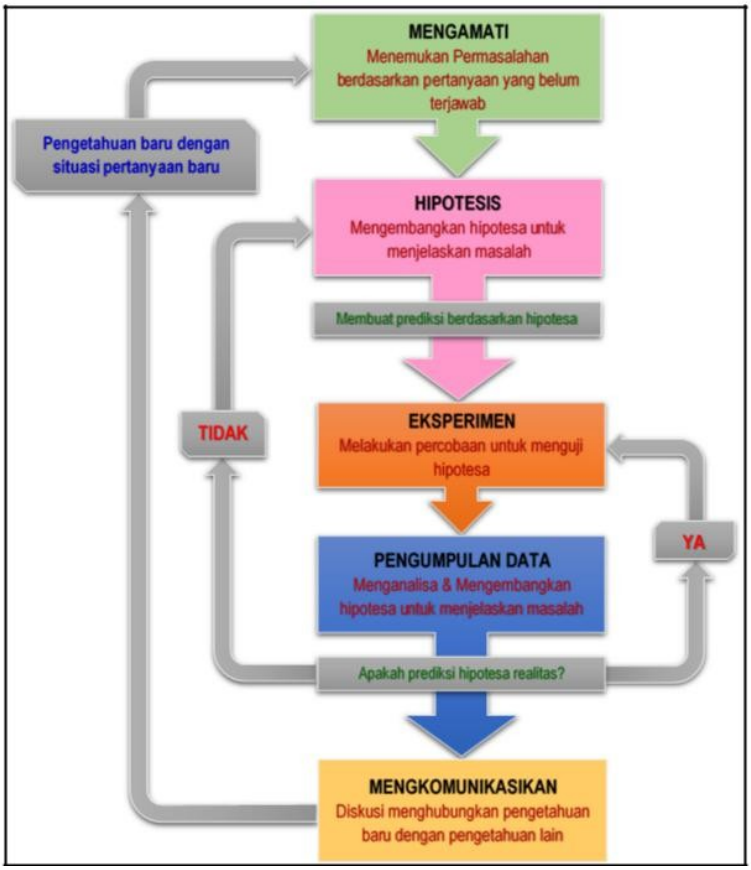
BAB I PENDAHULUAN

A. Deskripsi

Pemrograman berorientasi objek (Inggris: *object-oriented programming* disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Ini adalah jenis pemrograman di mana programmer mendefinisikan tidak hanya tipe data dari sebuah struktur data, tetapi juga jenis operasi (fungsi) yang dapat diterapkan pada struktur data. Dengan cara ini, struktur data menjadi objek yang meliputi data dan fungsi. Selain itu, pemrogram dapat membuat hubungan antara satu benda dan lainnya. Sebagai contoh, objek dapat mewarisi karakteristik dari objek lain.

Salah satu keuntungan utama dari teknik pemrograman berorientasi obyek atas teknik pemrograman prosedural adalah bahwa memungkinkan programmer untuk membuat modul yang tidak perlu diubah ketika sebuah jenis baru objek ditambahkan. Seorang pemrogram hanya dapat membuat objek baru yang mewarisi banyak fitur dari objek yang sudah ada. Hal ini membuat program object-oriented lebih mudah untuk memodifikasi.

Pembelajaran pemrograman berorientasi obyek ini menggunakan metode pendekatan saintifik. Dalam pendekatan ini praktikum atau eksperimen berbasis sains merupakan bidang pendekatan ilmiah dengan tujuan dan aturan khusus, dimana tujuan utamanya adalah untuk memberikan bekal ketrampilan yang kuat dengan disertai landasan teori yang realistis mengenai fenomena yang akan kita amati. Ketika suatu permasalahan yang hendak diamati memunculkan pertanyaan-pertanyaan yang tidak bisa terjawab, maka metode eksperimen ilmiah hendaknya dapat memberikan jawaban melalui proses yang logis. Proses-proses dalam pendekatan scientifik meliputi beberapa tahapan yaitu: mengamati, hipotesis atau menanya, mengasosiasikan atau eksperimen, mengumpulkan atau analisa data dan mengkomunikasikan. Proses belajar pendekatan eksperimen pada hakekatnya merupakan proses berfikir ilmiah untuk membuktikan hipotesis dengan logika berfikir.



Gambar 1. Diagram Proses Metode Saintifik-Eksperimen Ilmiah

B. Prasyarat

Untuk kelancaran pencapaian kompetensi dalam mata pelajaran pemrograman berorientasi obyek ini dibutuhkan beberapa peryaratan baik pengetahuan maupun ketrampilan dasar. Persyaratan tersebut antara lain ialah: Peserta didik telah menguasai mata pelajaran pemrograman dasar. Konsep dan algoritma pemrograman ini dibutuhkan untuk mendukung implementasi pemrograman berorientasi obyek. Disamping itu peserta didik mempunyai kompetensi dalam hal pemanfaatan teknologi informasi, seperti mengoperasikan hardware komputer dan mengoperasikan perangkat lunak aplikasi. Perangkat lunak aplikasi tersebut antar lain ialah pengolah data untuk menganalisis data hasil eksperimen, pengolah kata untuk membuat laporan dan aplikasi presentasi untuk mengkomunikasikan dan mempresentasikan hasil laporan.



C. Petunjuk Penggunaan

Buku pedoman siswa ini disusun berdasarkan kurikulum 2013 yang mempunyai ciri khas penggunaan metode scientifik. Buku ini terdiri dari dua bab yaitu bab satu pendahuluan dan bab dua pembelajaran. Dalam bab pendahuluan beberapa yang harus dipelajari peserta didik adalah diskripsi mata pelajaran yang berisi informasi umum, rasionalisasi dan penggunaan metode scientifik. Selanjutnya pengetahuan tentang persyaratan, tujuan yang diharapkan, kompetensi inti dan dasar yang akan dicapai serta test kemampuan awal.

Bab dua menuntun peserta didik untuk memahami deskripsi umum tentang topik yang akan dipelajari dan rincian kegiatan belajar sesuai dengan kompetensi dan tujuan yang akan dicapai. Setiap kegiatan belajar terdiri dari tujuan dan uraian materi topik pembelajaran, tugas serta test formatif. Uraian pembelajaran berisi tentang diskripsi pemahaman topik materi untuk memenuhi kompetensi pengetahuan. Uraian pembelajaran juga menjelaskan deskripsi unjuk kerja atau langkah-langkah logis untuk memenuhi kompetensi skill.

Tugas yang harus dikerjakan oleh peserta didik dapat berupa tugas praktek, eksperimen atau pendalaman materi pembelajaran. Setiap tugas yang dilakukan melalui beberapa tahapan saintifik yaitu : 1) melakukan pengamatan setiap tahapan unjuk kerja 2) melakukan praktek sesuai dengan unjuk kerja 3) mengumpulkan data yang dihasilkan setiap tahapan 4) menganalisa hasil data menggunakan analisa diskriptif 5) mengasosiasikan beberapa pengetahuan dalam uraian materi pembelajaran untuk membentuk suatu kesimpulan 6) mengkomunikasikan hasil dengan membuat laporan portofolio. Laporan tersebut merupakan tagihan yang akan dijadikan sebagai salah satu referensi penilaaian.

D. Tujuan Akhir

Setelah mempelajari uraian materi dalam bab pembelajaran dan kegiatan belajar diharapkan peserta didik dapatmemiliki kompetensi sikap, pengetahuan dan ketrampilan yang berkaitan dengan materi:

Paradigma pemrograman berorientasi obyek
Dasar dan aturan pemrograman berorientasi obyek
Class dan Obyek
Enkapsulasi data
Pemawarisan
Polimorpisme
Package



E. Kompetensi Inti Dan Kompetensi Dasar

0 **Kompetensi Inti 1 :** Menghayati dan mengamalkan ajaran agama yang dianutnya.

Kompetensi Dasar :

1.1. Memahami nilai-nilai keimanan dengan menyadari hubungan keteraturan dan kompleksitas alam dan jagad raya terhadap kebesaran Tuhan yang menciptakannya

1.2. Mendeskripsikan kebesaran Tuhan yang menciptakan berbagai sumber energi di alam

1.3. Mengamalkan nilai-nilai keimanan sesuai dengan ajaran agama dalam kehidupan sehari-hari.

1 **Kompetensi Inti 2:** Menghayati dan Mengamalkan perilaku jujur, disiplin, tanggung jawab, peduli (gotong royong, kerjasama, toleran, damai), santun, responsif dan proaktif dan menunjukan sikap sebagai bagian dari solusi atas berbagai permasalahan dalam berinteraksi secara efektif dengan lingkungan sosial dan alam serta dalam menempatkan diri sebagai cerminan bangsa dalam menempatkan diri sebagai cerminan bangsa dalam pergaulan dunia.

Kompetensi Dasar:

2.1. Menunjukkan perilaku ilmiah (memiliki rasa ingin tahu; objektif; jujur; teliti; cermat; tekun; hati-hati; bertanggung jawab; terbuka; kritis; kreatif; inovatif dan peduli lingkungan) dalam aktivitas sehari-hari sebagai wujud implementasi sikap dalam melakukan percobaan dan berdiskusi

2.2. Menghargai kerja individu dan kelompok dalam aktivitas sehari-hari sebagai wujud implementasi melaksanakan percobaan dan melaporkan hasil percobaan.

2 **Kompetensi Inti 3:** Memahami, menerapkan dan menganalisis pengetahuan faktual, konseptual dan prosedural berdasarkan rasa ingin tahunya tentang ilmu pengetahuan, teknologi, seni, budaya, dan humaniora dalam wawasan kemanusiaan, kebangsaan, kenegaraan, dan peradaban terkait penyebab fenomena dan kejadian dalam bidang kerja yang spesifik untuk memecahkan masalah.

Kompetensi Dasar:

3.1. Memahami konsep pemrograman berorientasi obyek

3.2. Memahami dasar-dasar dan aturan pemrograman berorientasi obyek



3.3. Memahami konsep class dan obyek
3.4. Memahami konsep enkapsulasi dalam melindungi data dan informasi
3.5. Memahami konsep pewarisan
3.6. Memahami konsep polimorphisme
3.7. Menerapkan penggunaan pakage dalam aplikasi

- **Kompetensi Inti 4:** Mengolah, menalar, dan menyaji dalam ranah konkret dan ranah abstrak terkait dengan pengembangan dari yang dipelajarinya di sekolah secara mandiri, dan mampu melaksanakan tugas spesifik dibawah pengawasan langsung.

  Kompetensi Dasar:

4.1 Menyajikan konsep pemrograman berorientasi obyek
4.2 Menyajikan aturan dan dasar-dasar pemrograman berorientasi obyek
4.3 Menyajikan class dengan memberikan atribut dan metode
4.4 Menyajikan perlindungan data dan informasi melalui mekanisme enkapsulas
4.5 Mengolah hubungan antara class dengan pola pewarisan
4.6 Menyajikan konsep polimorphisme dengan overloading dan overiding
4.7 Menyajikan aplikasi melalui pengelompokan class dalam pakage

- **Cek Kemampuan Awal**



0. Jelaskan perbedaan prmrograman prosedural dengan pemrograman berorientasi obyek !
1. Jelaskan keuntungan pemrograman berorientasi obyek !
2. Jelaskan secara singkat proses kompilasi dan menjalankan program aplikasi berorientasi obyek!
3. Jelaskan secara singkat dan berikan contoh pengertian class, atribut dan metode!
4. Jelaskan pengertian pembungkusan data dengan konsep enkapsulasi !
5. Jelaskan secara singkat konsep pewarisan dalam pemrograman berorientasi obyek !
6. Jelaskan secara singkat konsep polimorpisme dalam pemrograman berorientasi obyek !
7. Jelaskan cara pembuatan package dan sebutkan keuntungannya !

BAB II PEMBELAJARAN

A. Deskripsi

Dalam bab 1 ini akan menjelaskan dan menyajikan konsep pemrograman berorientasi obyek yang terdiri dari 2 kegiatan belajar. Kegiatan belajar 1 akan memahamkan anda tentang paradigma pemrograman berorientasi obyek dan menganalisis perbedaan pemrograman procedural dan pemrograman perorientasi obyek. Kegiatan belajar 2 meliputi penjelasan alur kerja perangkat lunak berorientasi obyek dan melakukan instalasi perangkat lunak. Setiap kegiatan belajar disertai dengan tujuan pembelajaran yang akan dicapai dalam 1 kali tatap muka, uraian materi, tes formatif untuk menguji kompetensi pengetahuan anda, dan tugas atau praktikum(individu dan kelompok) untuk menguji kompetensi keterampilan anda.

B. Kegiatan Belajar

1. Kegiatan Belajar 1 : Mengenal Pemrograman Berorientasi Obyek

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 1 ini siswa diharapkan dapat :

0 Memahami paradigma pemrograman berorientasi obyek

1 Menganalisis perbandingan pemrograman procedural dan pemrograman berorientasi obyek

4 Uraian Materi

23 Paradigma Pemrograman Berorientasi Obyek

Ide dasar pada bahasa berorientasi obyek (POB) adalah mengkombinasikan data dan fungsi untuk mengakses data menjadi sebuah kesatuan unit. Unit ini di kenal dengan nama **obyek.** Obyek sebenarnya mencerminkan pola kerja manusia dalam kehidupan kerja sehari-hari. Sebuah obyek dapat diibaratkan sebagai departemen di dalam sebuah perusahaan bisnis. Contoh departemen.

23 Penjualan

24 Akuntan

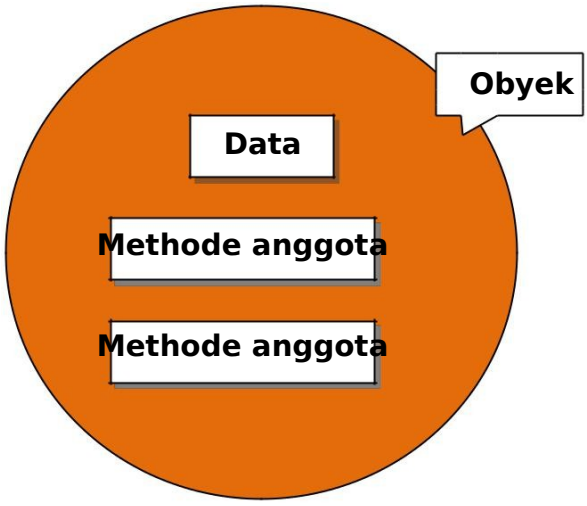
25 Personalia

Pembagian departemen dalam perusahaan merupakan upaya untuk memudahkan pengoperasian perusahaan. Sebagai gambaran, jika anda seorang



menejer penjualan di Kantor Pusat ingin mengetahui data personalia *salesmen*di suatu kantor cabang, apa yang anda lakukan ? Langkah yang anda tempuh pasti tidak dating secara langsung ke ruang personalia dan mencari data pada pada berkas-berkas yang ada pada departemen personalia sesuai yang anda butuhkan. Masalah bagaimana dan siapa yang mencarikan laporan yang diperlukan bukanlah menjadi urusan anda. Analogi dengan hal ini, kalau seseorang bermaksud menggunakan obyek, ia cukup mengirim suatu pesan ke obyek dan obyek itu sendiri yang akan menanganinya.

Bisa dibayangkan, betapa repotnya anda kalau anda sebagai manejer penjualan harus mencari sendiri berkas-berkas yang ada pada departemen personalia.Barangkali anda malah bakal mengobrak-abrik berkas-berkas yang sudah tersusun rapi. Kejadian semacam inilah yang dihindari pada konsep pemrogaman berorientasi obyek.Sebuah gambaran tentang obyek yang berisi data dan fungsi yang memanipulasi data dapat dilihat pada gambar 1.1.



Gambar 1. Data dan fungsi pemanipulasian data pada suatu obyek

➢ **Objek**

Object adalah gabungan antara beberapa data dan fungsi yang masing-masing bekerja bersama-sama dan tidak dapat dipisahkan. Gabungan dari data dan fungsi tersebut akan membentuk suatu object-object yang aktif. Dari kumpulan beberapa object yang sama akan membentuk struktur baru yang disebut class.

Konsep dasar object (*object Oriented)* meliputi tiga hal *:*

23 Is Identical (because Object has own unique ID), yaitu object tersebut mempunyai identitas tersendiri dapat dibedakan dengan yang lain

24 Has Behavior (because Object has Method), yaitu object itu mempunyai prilaku atau sifat-sifat yang khusus

25 Has State (because Object has instance parameter), object mempunyai ukuran yang baku.

Dalam suatu sistem yang komplek seperti dalam project managemen sering kita jumpai objek-objek suatu kelas mempunyai relasi / hubungan dengan object-object dikelas yang lain. Secara umum relasi object dapat dibedakan menjadi tiga dasar hubungan yaitu :

0 Is-a (Generalization, Realization: Inheritance).

1 Has-a (Association).

2 Others (Association , Dependency)

24 Karakteristik Objek

Untuk lebih jelasnya karakteristik objek tersebut dijelaskan sebagai berikut :



Gambar 2.Contoh obyek

23 Identitas berarti data diukur mempunyai nilai tertentu yang membedakan entitas dan inilah yang disebut objek

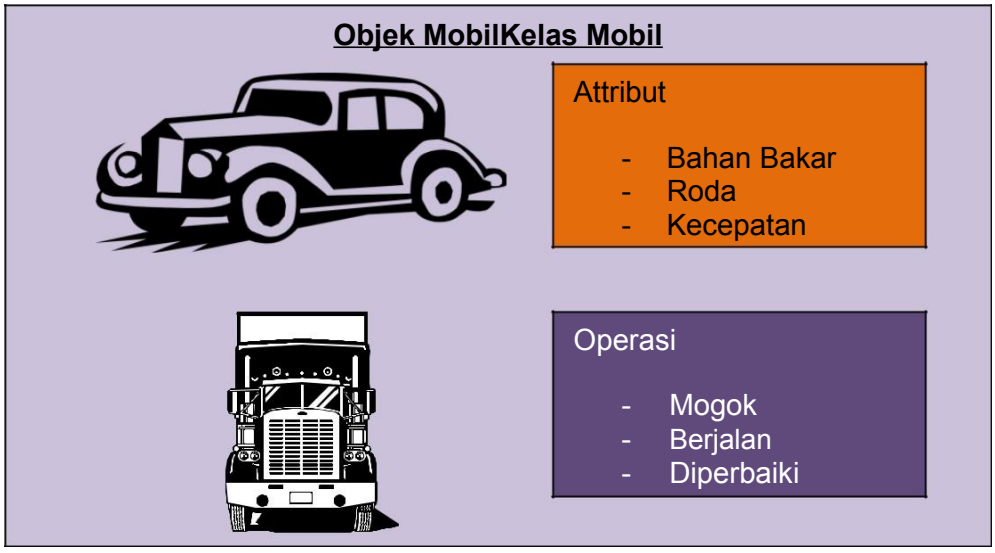
Suatu contoh : Singa merupakan objek dari binatang buas, Manusia merupakan dari makhluk hidup ciptaan Allah SWT, keyboard merupakan objek dari perangkat keras komputer, mobil merupakan objek dari alat transportasi. Setiap objek mempunyai sifat yang melekat pada identitasnya, sehingga dua objek dapat berbeda walaupun bila semua nilai atributnya identik. Lebih jelasnya lihat gambar 3.

Objek didunia nyata sangatlah sederhana, tetapi dalam pemrograman mempunyai penanganan yang unik dan tidak sederhana di dunia nyata. Penanganannya bias saja dinyatakan beberapa cara, seperti alamat, indeks, dari array atau nilai unik dari atribut. Referensi objek seragam dan

independen dari sisi objek, memperbolehkan campuran kumpulan dari objek yang dibuat, seperti file dalam direktori yang berisi file dan subdirektori.

23 Klasifikasi berarti suatu kegiatan mengumpulkan data (atribut) dan prilaku (operasi) yang mempunyai struktur data sama ke dalam satu grup yang disebut kelas. Sesuai dengan contoh di atas maka hewan buas, makhluk hidup, komputer, dan alat transportasi adalah contoh dari kelas.

Kelas menunjukkan abstraksi yang menjelaskan sifat penting pada suatu aplikasi dan mengabaikan yang lain. Setiap kelas menunjukkan suatu kumpulan infinite yang mungkin dari objek. Suatu objek dapat dikatakan sebagai instans dari kelas. Setiap instans dari kelas mempunyai nilai individu untuk setiap nama atribut dan operasi, tetapi memiliki bersama atribut dan operasi dengan instans lain dalam kelas. Gambar di bawah ini menunjukkan dua kelas dengan beberapa instans yang berhubungan dengannya.



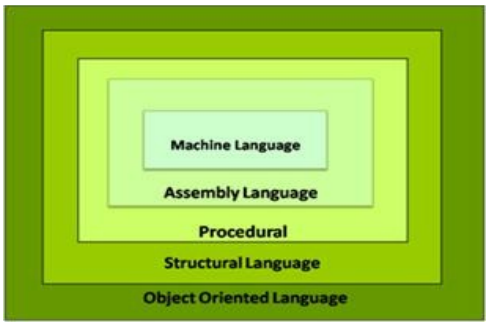
Gambar 3. Contoh obyek dengan properti

Dalam dunia nyata, suatu operasi adalah suatu abstrak dari analogi prilaku terhadap objek-objek yang berbeda. Setiap objek mengetahui bagaimana melakukan operasinya. Dalam bahasa pemrograman berorientasi objek, secara otomatis bahasa akan mamilih metode yang tepat untuk menjalankan operasinya berdasarkan nama dimana dilakuakan operasi terhadapnya. Pengguna dari operasi tidak perlu kuatir berapa banyak metoda yang terdapat dalam implentasi. Kelas baru dapat ditambahkan tanpa mengubah *code* yang

sudah ada, melengkapi metoda adalah melengkapi operasi yang dapat dilakukan terhadap kelas baru.

23 Perbandingan Pemrograman Berorientasi Obyek dengan Pemrograman Terstruktur

Paradigma bahasa pemrograman memberikan model untuk programmer dalam menulis listing program. Paradigma perbedaan dalam bahasa pemrograman sebagai berikut :



- Pemrograman tidak terstruktur atau *Programming Monolithic*
- Pemrograman prosedural
- Pemrograman struktural
- Pemrograman Berorientasi Objek

Gambar 4. Bahasa pemrograman

23 Pemrograman tidak terstruktur atau *Programming Monolithic* :

0 Seluruh permasalahan ini diselesaikan sebagai blok tunggal.

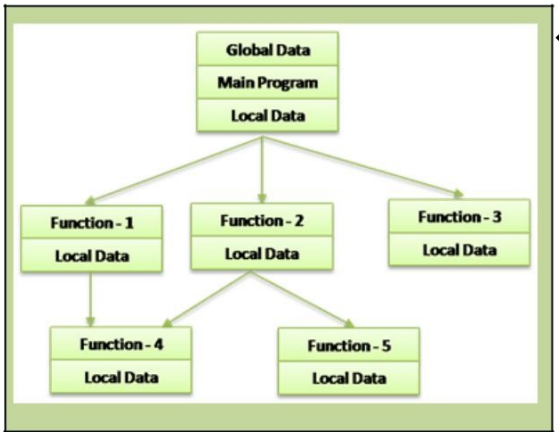
1 Semua data bersifat global dan tidak ada keamanan.

2 Perintah melompat diperbolehkan `jump` dan banyak menggunakan perintah `go to`

3 Cocok untuk permasalah kecil.

4 Sulit untuk melacak kesalahan program

Contoh bahasa pemrograman yang termasuk dalam*Programming Monolithic*adalah Assembly Language, BASIC.



Gambar 5.Diagram bahasa pemrograman

❖ Pemrograman prosedural

- ✓ Masalah yang diberikan dibagi dalam beberapa sub masalah tergantung pada fungsinya.
- ✓ Sub masalah disebut prosedur atau Metode.
- ✓ Prosedur apapun dapat dipanggil pada setiap saat selama pelaksanaan program.
- ✓ Program ini memiliki variabel global dan lokal.

Fitur Pemrograman berorientasi prosedur:

23 Program besar yang terbagi dalam fungsi kecil atau Prosedur.

24 Menggunakan Pendekatan pemrograman Top-Down.

25 Data bergerak bebas dari satu fungsi ke yang lain.

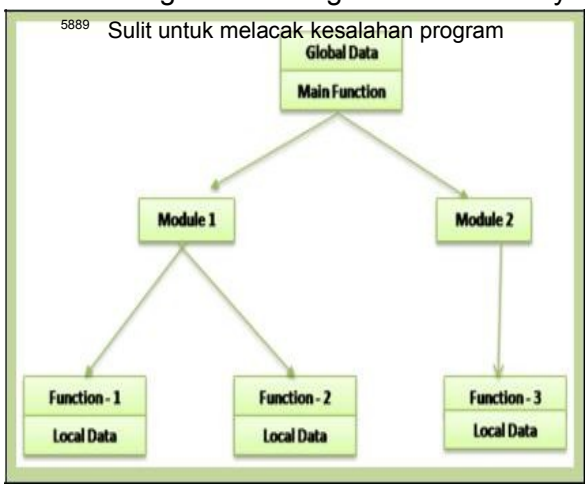
26 Sebagian besar fungsi berbagi data umum.

27 Penekanan diberikan untuk algoritma.

Kekurangan:

5888 Sangat sulit mengidentifikasi data yang digunakan oleh yang berfungsi.

5889 Sulit untuk melacak kesalahan program



5888 Pemrograman Terstruktur

23 Program ini dibagi menjadi modul dan modul tersebut kemudian dibagi menjadi fungsi.

24 Penggunaan Pernyataan goto dihapus atau dikurangi.

25 Setiap modul dapat bekerja independen satu sama lain.

Gambar 6.Diagram bahasa pemrograman

23 Pemrograman berorientasi obyek

- Program ini dibagi menjadi jumlah unit kecil yang disebut Object. Data dan fungsi merupakan properti objek.
- Data dari objek hanya dapat diakses oleh fungsi yang terkait dengan objek tersebut.
- Fungsi satu objek dapat mengakses fungsi objek lain.

Fitur pemrograman berorientasi obyek

- Penekanan diberikan pada data daripada prosedur.
- Masalah dibagi menjadi obyek.
- Struktur data dirancang sedemikian rupa sehingga mereka mengatur objek.
- Data dan fungsi yang diikat bersama-sama.
- Penyembunyian data adalah mungkin.
- Data baru dan fungsi dapat dengan mudah dibuat.
- Obyek dapat berkomunikasi satu sama lain dengan menggunakan fungsi.
- Pendekatan bottom-up yang digunakan dalam membuat program

Perbedaan antara Pemrograman Berorientasi Terstruktur dan Objek

Pemrograman Terstruktur	Pemrograman Berorientasi Obyek
Pendekatan top-down	Pendekatan bottom-up yang diikuti.
Fokus adalah pada algoritma dan kontrol aliran.	Fokus pada model objek.
Program dibagi menjadi beberapa sub modul atau fungsi atau prosedur.	Program ini diselenggarakan dengan memiliki sejumlah kelas dan objek.
Fungsi yang independen satu sama lain.	Setiap kelas berhubungan secara hirarkis.
Tidak ada penerima yang ditunjuk dalam panggilan fungsi.	Ada penerima yang ditunjuk untuk setiap lewat pesan.
Data dan fungsi sebagai dua entitas yang terpisah Views.	Data dan fungsi sebagai satu kesatuan pandangan.
Pemeliharaan mahal.	Pemeliharaan relatif lebih murah.
Reuse Software tidak mungkin.	Membantu dalam penggunaan kembali perangkat lunak.



Fungsi panggilan digunakan. Fungsi abstraksi digunakan. Algoritma diberikan penting. Solution adalah solusi spesifik-domain. Tidak ada enkapsulasi. Data dan fungsi yang terpisah	Message passing digunakan. Data abstaction digunakan. Data diberikan penting. Solution adalah spesifik masalah domain. Enkapsulasi paket kode dan data sama sekali. Data dan fungsi disatukan dalam satu kesatuan.
Hubunganantaraprogrammer dan program ditekankan. Teknik data-driven digunakan.	Hubunganantaraprogrammerdan pengguna ditekankan. Didorong oleh delegasi tanggung jawab.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sedangkan untuk pemrograman terstruktur, menggunakan prosedur/tata cara yang teratur untuk mengoperasikan data struktur. Untuk tata nama, keduanya pun memiliki tatanan yang sama walaupun memiliki pengertian tersendiri:

object oriented menggunakan “method” sedangkan terstruktur menggunakan “function”. Bila di OOP sering didengar mengenai “objects” maka di terstruktur kita mengenalnya dengan ” modules”. Begitu pula halnya dengan “message” pada OO dan “argument” pada terstruktur. “attribute” pada OO juga memiliki tatanan nama yang sepadan dengan “variabel” pada pemrograman terstruktur.

Pemrograman prosedural akan dikatakan lebih baik apabila dalam segala situasi melibatkan kompleksitas moderat atau yang memerlukan signifikan kemudahan maintainability. Manfaat yang dirasakan dalam penggunaan pemrograman prosedural adalah kemampuan kembali menggunakan kode yang sama tanpa menggunakan kode yang berbeda ataupun mengkopinya kembali. Dengan menggunakan “goto”, memudahkan programmer melacak kumpulan data sehingga menghindarkan pemrograman terstruktur menjadi seperti spagethii code.

Pemrograman berorientasikan objek dikatakan lebih baik apabila Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan



mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

c. Rangkuman

Pengembangan berorientasi objek merupakan cara pikir baru tentang perangkat lunak berdasarkan abstraksi yang terdapat dalam dunia nyata. Dalam konteks pengembangan menunjuk pada bagian awal dari siklus hidup pengembangan sistem, yaitu survei, analisis, desain, implementasi, dan pemeliharaan sistem. Hal yang lebih penting dalam pengembangan berorientasi objek adalah konsep mengindentifikasi dan mengorganisasi domain aplikasi dibandingkan dengan fokus penggunaan bahasa pemrograman, berorientasi objek atau tidak.

Object adalah gabungan antara beberapa data dan fungsi yang masing-masing bekerja bersama-sama dan tidak dapat dipisahkan. Gabungan dari data dan fungsi tersebut akan membentuk suatu object-object yang aktif. Dari kumpulan beberapa object yang sama akan membentuk struktur baru yang disebut class.

Pemrograman berorientasi objek (Inggris: object-oriented programming disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Pemrograman terstruktur adalah suatu proses untuk mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program. Selain pengertian diatas Pemrograman Terstruktur adalah suatu aktifitas pemrograman dengan memperhatikan urutan langkah-langkah perintah secara sistematis, logis , dan tersusun berdasarkan algoritma yang sederhana dan mudah dipahami.Prinsip dari pemrograman terstruktur adalah Jika suatu proses telah sampai pada suatu titik / langkah tertentu , maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya / kembali lagi ke baris sebelumnya, kecuali pada langkah – langkah untuk proses berulang (Loop).



0 Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

❖ Mengamati benda dan obyek

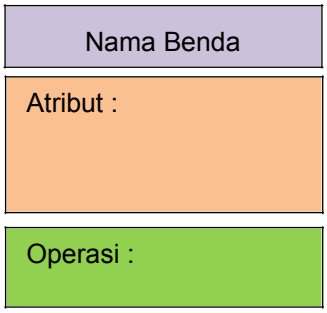
0 Buatlah kelompok dengan anggota 3 – 4 orang

1 Amatilah dan tuliskan 5 benda yang ada disekitar anda

2 Sebutkan ciri-ciri atau atribut masing-masing(minimal 3) dari kelima benda tersebut yang menjadi identitasnya sehingga mudah dikenali.

3 Sebutkan fungsi masing-masing (minimal 2) dari kelima benda tersebut yang merupakan operasi relasi dari ciri-ciri yang sudah teridentifikasi

4 Deskripsikan setiap benda tersebut seperti gambar berikut



6. Buat laporan dan diskusikan dengan teman sekelompok

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal,amati dan diskusikan

❖ Mengamati Bahasa Pemrograman

0 Buatlah kelompok dengan anggota 3 – 4 orang

1 Amatilah dan deskripsikan 2 bahasa pemrograman yang sudah anda kenal

2 Tunjukkanlah perbedaan yang mendasar dari 2 bahasa pemrograman yang sudah anda kenal

3 Sebutkan persamaan fungsi atau kegunaan yang dapat terdapat dalam 2 bahasa pemrograman yang sudah anda ketahui

4 Buatlah tabelperbedaan dan persamaan yang anda dapatkan diskusikan dengan teman sekelompok

No	Bahasa Pemrograman	Sistem Operasi	File Extensi	Jenis Publikasi
1.				
2.				

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatanterhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama



e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



23 Ilustrasikan dalam dunia nyata apa yang disebut :

- 23 Objek
- 24 Atribut
- 25 Methode
- 26 Kelas

24 Sebutkan minimal 5 atribut dan minimal 3 method yangmelekat pada diri anda ?

25 Sebutkan paradigma lain dalam bahasa pemrograman selain paradigma berorientasi objek, bandingkan dan sebutkan masing-masing kekurangan dan kelemahannya ?

26 Mengapa saat ini metodologi berorientasi objek berkembang lebih pesat dibandingkan dengan metode-metode yang lain dalam bahasa pemrograman

5888 ***Lembar Jawaban Test Formatif (LJ).*** **LJ- 01 :**Ilustrasi dalam dunia nyata :



5888 Objek.....
.....
.....
.....
.....

5889 Atribut
….....
.....
.....
.....

5890 Methode.....
.....
.....
.....
.....

5891 Kelas…….....
.....
.....
.....
.....

LJ- 02 : 5 Atribut dan 3 method yang melekat pada diri siswa?



.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 : Paradigma pemrograman berorientasi objek vs pemrograman terstruktur



0 Pemrograman berorientasi obyek

.....
.....
.....
.....
.....
.....
.....
.....

1 Pemrograman terstruktur

.....
.....
.....
.....
.....
.....
.....
.....



.....

LJ- 04: Paradigma pemrograman berorientasi objek lebih banyak digunakan dewasa ini karena :



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

g. Lembar Kerja Siswa.



0 Kegiatan Belajar 2 : Perangkat Lunak Pemrograman Berorientasi Obyek

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 2 ini siswa diharapkan dapat :

0 Memahami alur kerja perangkat lunak pemrograman berorientasi obyek

1 Menyajikanperangkat lunak pemrograman berorientasi obyek

b. Uraian Materi

0 Alur kerja perangkat lunak pemrograman berorientasi obyek

Mengapa memilih Java sebagai perangkat lunak pemrograman berorientasi obyek ?

Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari Sun Micro system yang memiliki kode Green dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada arsitekture tertentu. Mulanya disebut OAK, tetapi karena OAK sendiri merupakan nama dari bahasa pemrograman komputer yang sudah ada. Maka Sun mengubahnya menjadi Java.

Sun kemudian meluncurkan browser dari Java yang disebut Hot Java yang mampu menjalankan applet. Setelah itu teknologi Java diadopsi oleh Netscape yang memungkinkan program Java dijalankan di browser Netscape yang kemudian diikuti Internet Explorer. Karena keunikanya dan kelebihanya, teknologi Java mulai menarik banyak vendor seperti IBM,Symantec, Inprise, dll.Sun merilis versi awal Java secara resmi pada awal tahun 1996 yang kemudian terus berkembang hingga muncul JDK 1.1, kemudian JDK 1.2 yang mulai disebut sebagai versi Java2 karena banyak mengandung peningkatan dan perbaikan. Perubahan utama adalah adanya Swing yang merupakan teknologi GUI (Graphical User Interface) yang mampu menghasilkan window yang portabel. Dan pada tahun 1998 – 1999 lahirlah teknologi J2EE.Diawali dengan servlet dan EJB kemudian diikuti JSP. Java juga menjadi lebih cepat populer di lingkungan server side dikarenakan kelebihanya di lingkungan

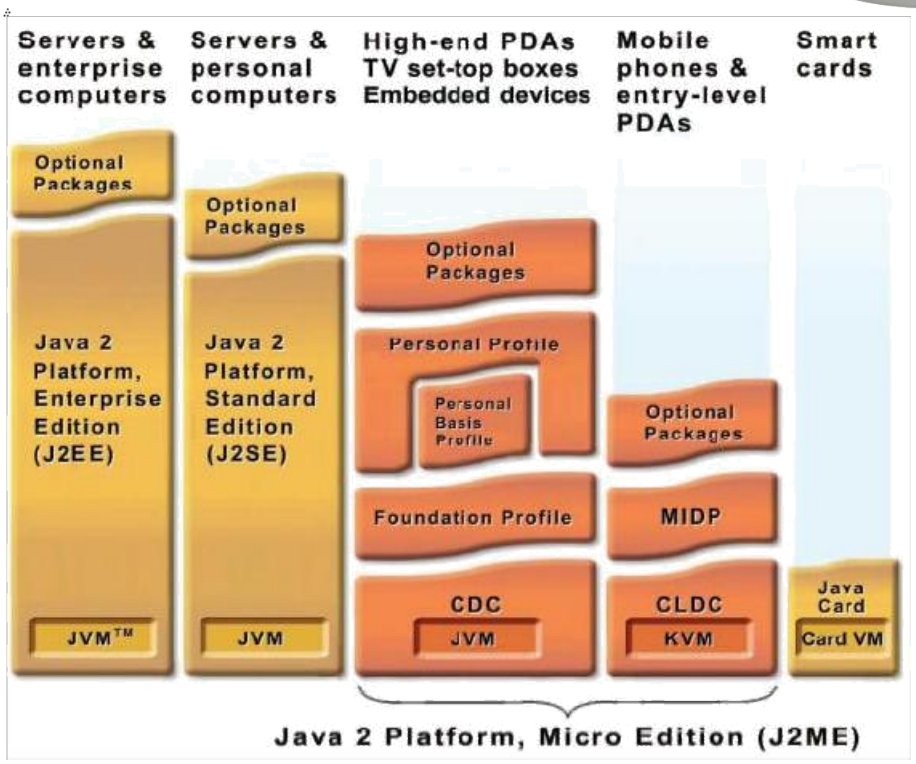
Arsitektur teknologi java

Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++.
Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan.Sebagai bahasa pemrograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua platform. Dan juga dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java. Sedangkan Java bersifat *neutral architecture ,*karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitekture perangkat keras yang disebut sebagai *Java Bytecode*.
Sebagai sebuah platform, Java terdiri atas dua bagian utama, yaitu:

- *Java Virtual Machine* (JVM).
- *Java Application Programming Interface* (JavaAPI).

Sun membagi arsitektur Java membagi tiga bagian, yaitu:

- **Enterprise Java (J2EE)** untuk aplikasi berbasis web, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi. Merupakan superset dari Standar Java
- **Standar Java (J2SE),** ini adalah yang biasa dikenal sebagai bahasa Java.
- **Micro Java (J2ME)** merupakan subset dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk wireless device / mobile device.



Gambar 7. Arsitektur Teknologi Java

1. Java API

Beberapa fitur yang ditawarkan Java API antara lain sebagai berikut :

0 **Applet**

Program Java yang dapat berjalan di atas browser, yang dapat membuat halaman HTML lebih dinamis dan menarik.

1 **Java Networking**

Sekumpulan API (*Application Programming Interface*) yang menyediakan fungsi– fungsi untuk aplikasi – aplikasi jaringan,

sepertipenyediaanaksesuntukTCP,UDP,IPAdrressdanURL. Tetapi Java Networking tidak menyediakan akses untuk ICMP dikarenakan alasan sekuriti dan pada kondidi umum hanya administrator (root) yang bisa memanfaatkan protokol ICMP.

0 **JavaDatabase Connectivity (JDBC)**

JDBC menyediakan sekumpulan API yang dapat digunakan untuk mengakses database seperti Oracle, MySQL, PostgreSQL, Microsoft SQL Server.

Java Security

Java Security menyediakan sekumpulan API untuk mengatur security dari aplikasi Java baik secara *high level* atau *low level*, seperti *public/private key management* dan *certificates*.

JavaSwing

Java Swing menyediakan sekumpulan API untuk membangun aplikasi–aplikasiGUI(GraphicalUserInterface)danmodelGUI yang diinginkan bisa bermacam–macam, bisa model Java, model Motif/CDE atau model yang dependent terhadap platform yang digunakan.

Java RMI

Java RMI menyediakan sekumpulan API untuk membangun aplikasi – aplikasi Java yang mirip dengan model RPC (*Remote Procedure Call*) jadi object - object Java bisa di call secara remote pada jaringan komputer.

Java2D/3D

Java 2D/3D menyediakan sekumpulan API untuk membangun grafik – grafik 2D/3D yang menarik dan juga akses ke printer.

Java Server Pages

Berkembang dari Java Servlet yan digunakan untuk menggantikan aplikasi – aplikasi CGI, JSP (Java Server Pages) yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusi aplikasi Internet.

0 JNI (Java Native Interface)

JNI menyediakan sekumpulan APIyang digunakan untuk mengakses fungsi – fungsi pada library (*.dll atau *.so) yang dibuat dengan bahasa pemrograman yang lain sepertiC,C++,dan Basic.

1 JavaSound

Java Sound menyediakan sekumpulan API untuk manipulasi sound.

2 Java IDL + CORBA

Java IDL (Interface Definition Language) menyediakan dukungan Java untuk implementasi CORBA (Common Object Request Broker) yang merupakan model distributed-

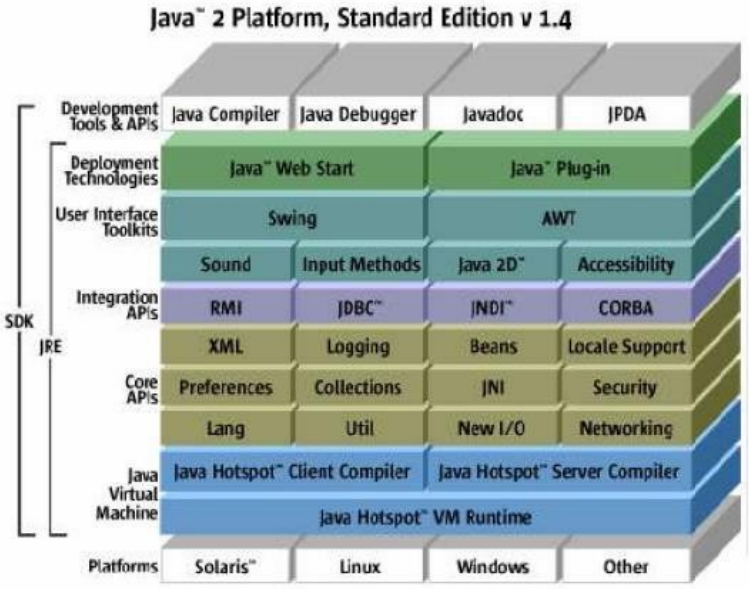
Object untuk solusi aplikasi besar di dunia networking.

0 **JavaCard**

Java Card utamanya digunakan untuk aplikasi – aplikasi pada smart card, yang sederhana wujudnya seperti SIM Card pada handphone.

1 **JTAPI (Java Telephony API)**

Java Telepony API menyediakan sekumpulan API untuk memanfaatkan devices – devices telepony, sehingga akan cocok untuk aplikasi – aplikasi CTI (Computer Telephony Integration) yang dibutuhkan seperti ACD (Automatic Call Distribution), PC- PBX



Gambar 8. J2 SE Standard Edition

Berdasarkan *white paper* resmi dari SUN,Java memiliki karakteristi berikut :

0 Sederhana

Bahasa pemrograman Java menggunakan sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection*.

 Berorientasiobjek(*ObjectOriented*)

Java mengunakan pemrograman berorientasi objek yang membuat

program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antarobjek-objektersebut.

0 Dapat didistribusi dengan mudah

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries* networking yang terintegrasi pada Java.

1 Interpreter

Program Java dijalankan menggunakan interpreter yaitu *JavaVirtual Machine* (JVM).Hal ini menyebabkan *sourcecode* Java yang telah dikompilasi menjadi Java *bytecodes* dapat dijalankan pada plat form yang berbeda-beda.

2 Robust

Java mempunyai reliabilitas yang tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime Exception handling* untuk membantu mengatasi error pada pemrograman.

3 Aman

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak system computer yang menjalankan aplikasi tersebut.

0 ArchitectureNeutral

Program Java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform yang berbeda dengan *Java VirtualMachine*.

1 Portabel

Source code maupun program Java dapat dengan mudah dibawa ke p latform yang berbeda-beda tanpa harus dikompilasi ulang.

2 Performance

Performance pada Java sering dikatakan kurang tinggi.Namun performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers*(JIT).

23 Multithreaded

Java mempunyai kemampuan untuk membuat suatu program yang dapat



melakukan beberapa pekerjaan secara sekaligus dan simultan.

23 Dinamis

Java didesain untuk dapat dijalankan pada lingkungan yangdinamis. Perubahan pada suatu *class* dengan menambahkan properties ataupun method dapat dilakukan tanpa menggangu program yang menggunakan *class* tersebut.

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, *desktop, web* dan lainnya,sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* system operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source.* Sebagian fitur dari Java :

23 Java Virtual Machine (JVM)

JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi hardware dan platform dimana kompilasi kode Java terjadi.Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikanolehJVM.

Aplikasi program Java diciptakan dengan *file* teks berekstensi *.java*.Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupainstruksi kode mesin.Perbedaannya adalah kode mesin harus dijalankan pada sistem komputerdimana kompilasi ditujukan,sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* system komputer dan system operasi.

24 Garbage Collection

Banyak bahasa pemrogaman lain yang mengijinkan eorang programmer mengalokasikan memori pada saat dijalankan. Namun, setelah menggunakan alokasi memori tersebut,harus terdapat cara untuk menempatkan kembali blok memori tersebut supaya program lain dapat menggunakannya. Dalam C, C++ dan bahasa lainnya, adalah programmer yang mutlak bertanggung jawab akan hal ini.Hal ini dapat menyulitkan bilamana programmer tersebut alpa untuk mengembalikan blok memori sehingga menyebabkan situasi yang

dikenal dengan nama *memory leaks.*

Program Java melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek–objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh programmer dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat pada bahasa yang memungkinkan alokasi dinamis.

0 ***Code Security***

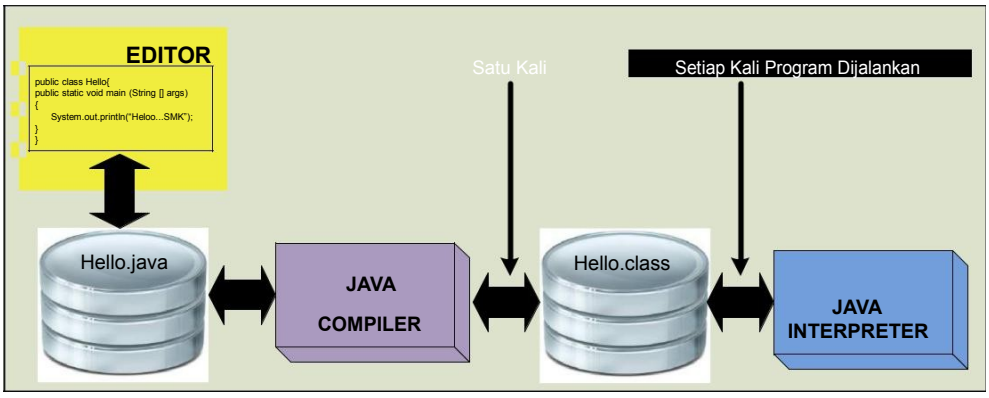
Code Security terimplementasi pada Java melalui penggunaan Java Runtime Environment(JRE).Java menggunakan model pengamanan 3lapis untukmelindungi sistemdari *untrusted Java Code*.

- Pertama,*class-loader* menangani pemuatan kelas Java ke *runtime interpreter.* Proses ini menyediakan pengamanan dengan memisahkank elas– kelas yang berasal dari *local disk* dengan kelas–kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas–kelas yang berasal dari *local disk* yang dimuat terlebih dahulu.
- Kedua,*byte code verifie r*membaca *byte code* sebelum dijalankan dan menjamin *byte code* memenuhi aturan–aturan dasar bahasa Java.
- Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumberdaya seperti sistemfile, *port* jaringan, proses eksternal dan system *windowing*.

Setelah seluruh proses tersebut selesai dijalankan,barulah kode program di eksekusi.

0 Fase-Fase pemrograman Java

Gambardibawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah programJava:



Gambar 9. Fase dari sebuah program Java

Langkah pertama dalam pembuatan sebuah program berbasis Java adalah menuliskan kode program pada *texteditor*. Contoh *text editor* yang dapat digunakan antara lain : notepad,vi,emacs dan lain sebagainya.Kode program yang dibuat kemudian tersimpan dalam sebuah berkas berekstensi *.java.* Setelah membuat dan menyimpan kode program,kompilasi file yang berisi kode program tersebut dengan menggunakan JavaCompiler. Hasildari kompilasi berupa berkas *byte code* dengan ekstensi *.class.* Berkas yang mengandung *byte code* tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan *platform* yang digunakan.

2) Instalasi Program Java SDK dan IDE Netbeans

Java SDK dan NetBeans diperlukan jika anda hendak mulai membuat program dengan bahasa pemrograman Java. ***JavaSDK*** adalah platform dasar Java yang diperlukan agar komputer atau laptop dapat digunakan untuk mengeksekusi kode-kode programbahasa Java, sedangkan ***NetBeans*** adalah aplikasi editor terpadu (IDE atau *Integrated Develepment Environment*) yang akan banyak mempermudah dalam membuat aplikasi karena menyediakan kontrol-kontrol visual yang penting dalam pemrograman desktop (atau lebih dikenal sebagai pemrograman visual).

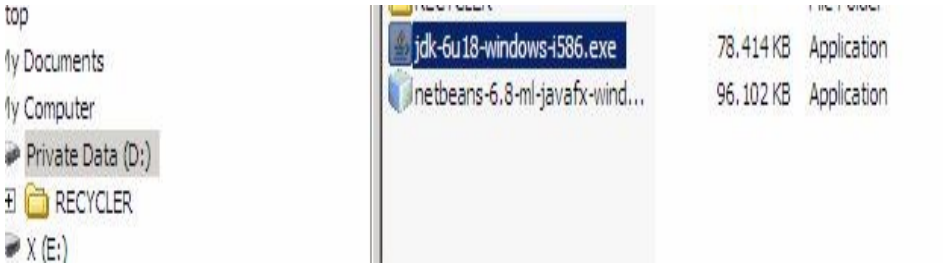
Sebelumnya perlu diketahui bahwa JavaSDK dan NetBeans memerlukan sumberdaya yang cukup tinggi, jadi sama sekali tidak bisa dibandingkan dengan Delphi, C++ Builder, apalagi Visual Basic versi 6 atau sebelumnya. Disarankan mengunakan komputer atau laptop dengan minimal RAM 1 GB dan prosesor *clock-speed* diatas 1GHZ.

Berikut ini langkah-langkah instalasi Java SDK dan NetBeans pada PC

dengan system operasi ***WindowsXPProfessional atau Windows Vista dan Windows 7*** disertai *screenshot.*

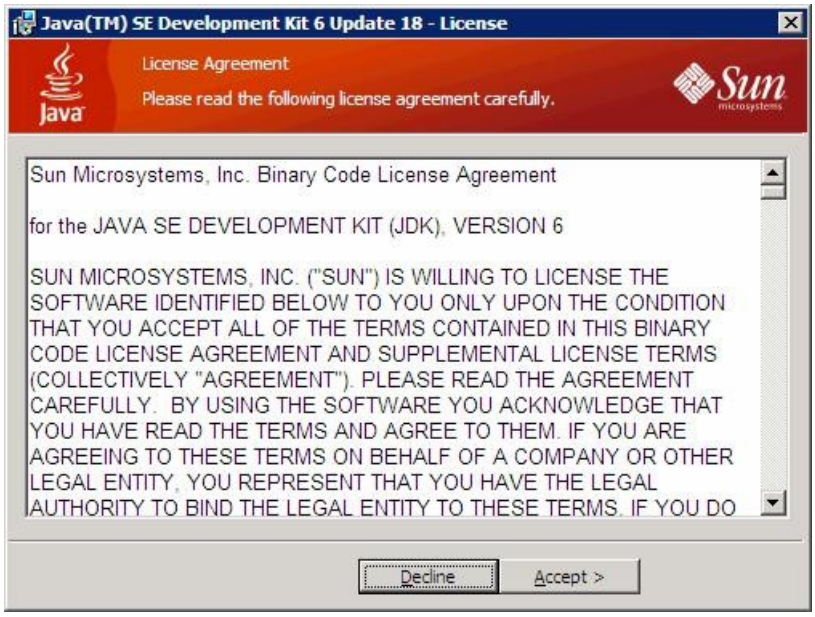
- **Instalasi JavaSDK**

  Java Standart Development Kit (SDK) tersedia untuk di download pada situs Web software Java Sun Microsystem pada :http://java.sun.com. Openfoldertempat file-file instalasi Java SDK.Dalamcontohini file disampan di drive Dpada MyComputer.



Gambar10.Open File Java Development Kit

Klik-dobel file instalasi Java SDK untuk mengeksekusi instalasi. Dalam contoh ini, file yang dieksekusi bernama lengkap ***jdk-6u18-windows-i586.exe***. Dalam beberapa detik, akan muncul kotak dialog berisi lembar persetujuan antara pihak ***Sun Microsystems,Inc.*** Sebagai pembuat software dan pihak anda sebagai pengguna software.



Gambar11.Licence Agreement Java Development Kit

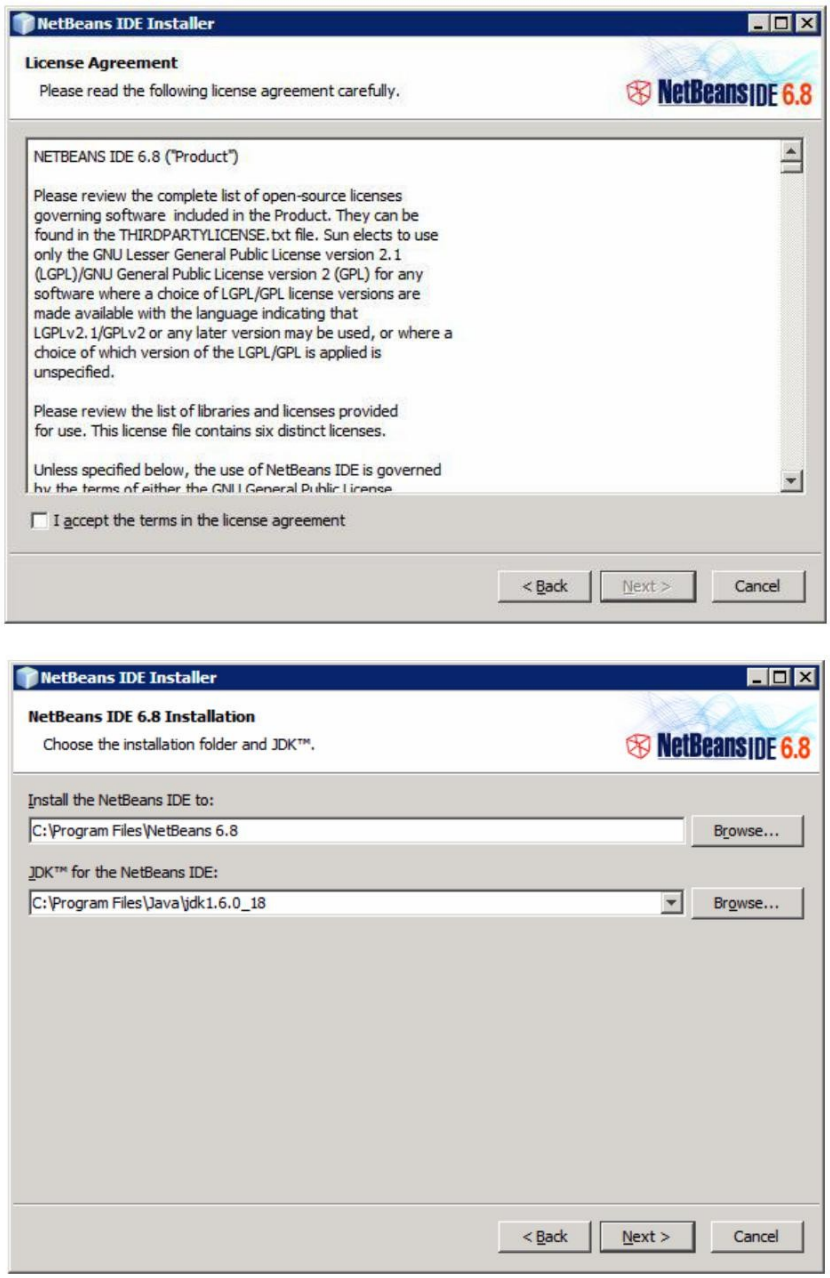
Tahapan-tahapan proses instalasi Java SDK dapat dilakukan dengan mudah dengan mengikuti petunjuk proses instalasi dengan menekan button next sampai pada tahap finish. Saat instalasi selesai, muncul kotak dialog yang memberitakan bahwa instalasi Java SDK lewatkan dengan mengklik tombol *Finish*.



Gambar12.Proses Instalasi Berhasil

- **Instalasi NetBeans**

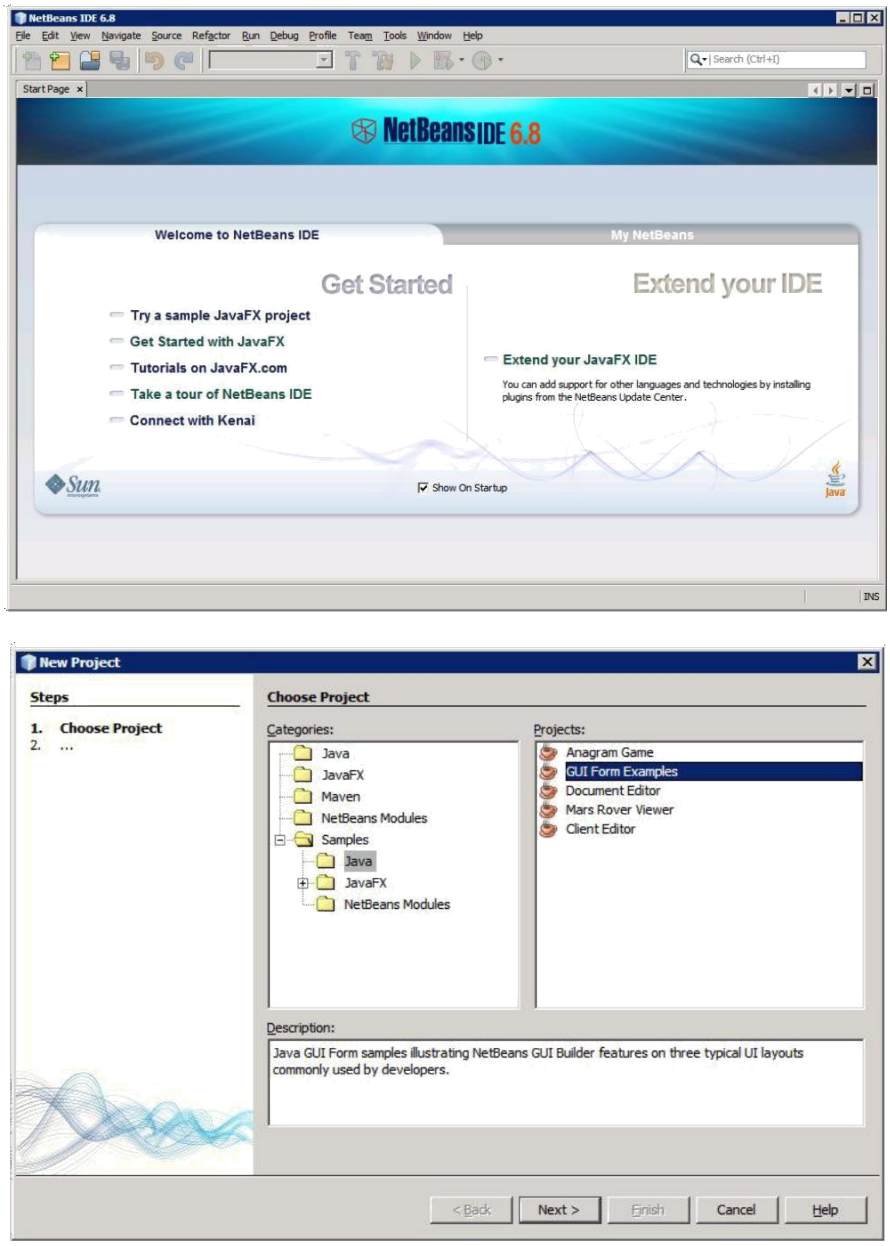
Instalasi Net Beans bisa dimulai dengan mengklik-dobel file instalasi yang ada di komputer atau laptop. Dalam contoh ini, filedi directory D, sehingga file ***netbeans-6.8-ml-javafx- windows.exe.*** Seperti halnya pada instalasi Java SDKatauproduk-produk berlisensi lain,installer NetBeans akan *menyodorkan* halaman persetujuan antara pembuat software dananda sebagai pemakai.



Gambar13.Licence Agreement NetBeans

Untuk memeriksa hasil instalasinya Netbens dengan langsung menggunakannya. Defaultnya, NetBeans bisa dibuka dengan mengklik ***Start*** pada Windows, pilih ***Programs***, pilih kelompok menu ***NetBeans***, lalu klik ***NetBeans IDE***.

Tanda NetBeans sedang dalam proses membuka modul-modul yang diperlukan untuk membuat aplikasi.



Gambar14.Start Up NetBeans

IDE NetBeans mengaharuskan membuat *new Project* terlebih dahulu sebelum menulis *script* program java. Dengan cara klik File new Project , langkah berikutnya memilih aplikasi *Java Aplication*.File dengan ektension .java dibuat untuk memulai menulis program java.

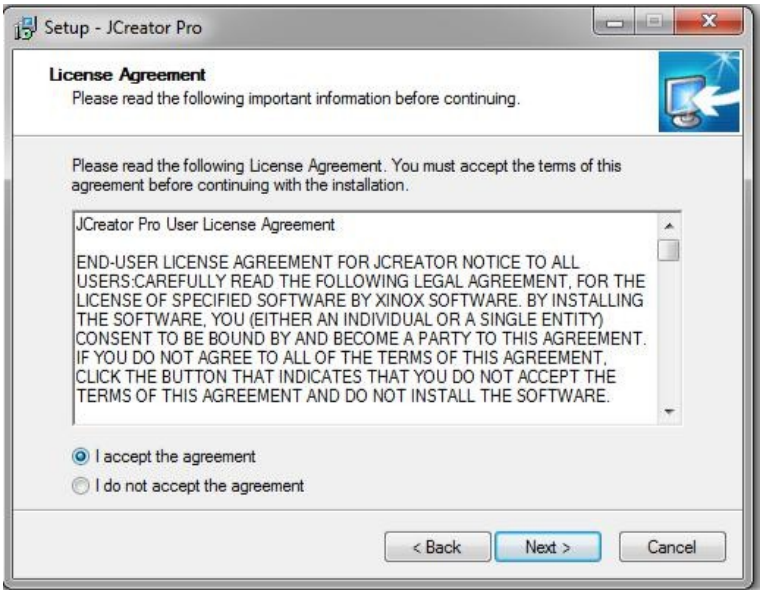
[23] Instalasi JCreator 5 Pro

Instalasi JCreator Pro 5 diawali dengan membuka file setup Jcreator dengan cara mengklik-dobel yang ada di komputer anda. Selanjutnya jika anda sudah membuka file setup akan dimunculkan pernyataan seperti gambar di bawah :



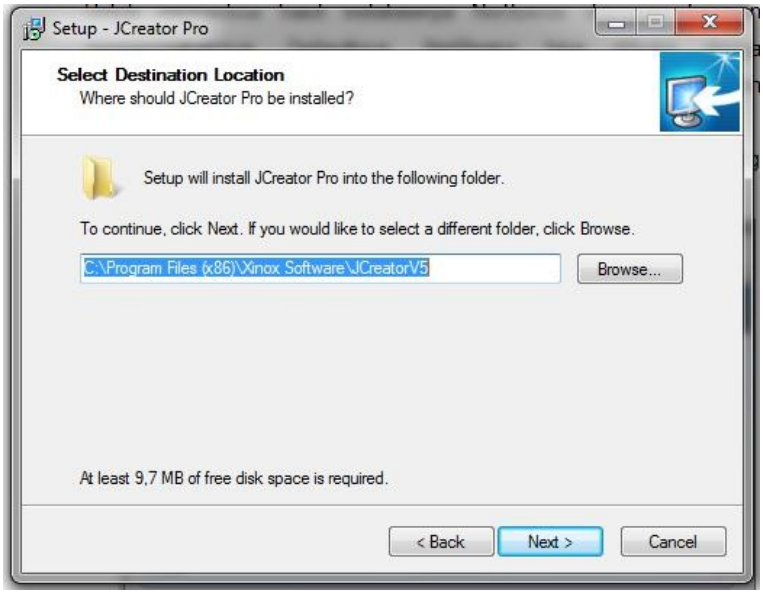
Gambar 15. Start Up JCreator

Klik Next untuk melanjutkan proses instalasi, jika tidak ingin melanjutkan klik cancel.



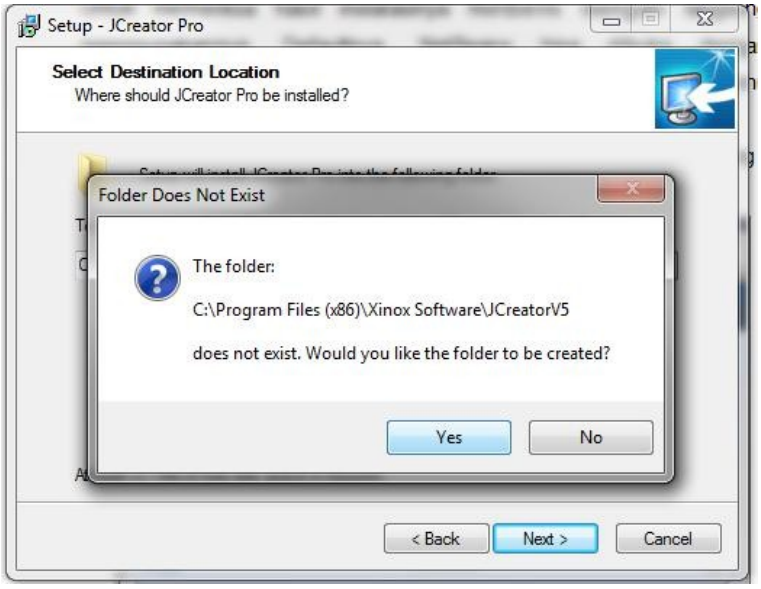
Gambar 16. License Agreement JCreator

Klik lingkaran pada tulisan “I accept the agreement”.



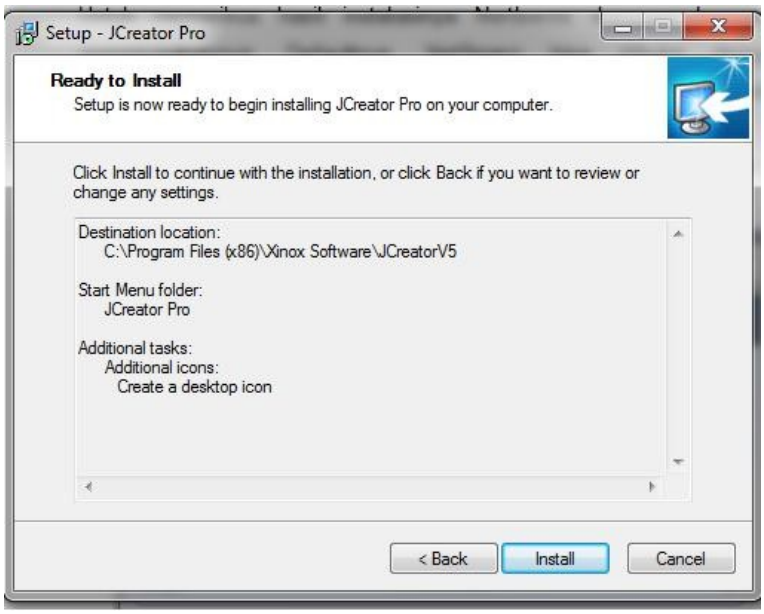
Gambar 17. Destinatio Location JCreator

Klik next untuk melanjutkan instalasi pada drive C:\Program Files, jika anda ingin meletakkan di drive lain, anda bisa meng-klik tombol browse kemudian tentukan tempat yang anda inginkan.



Gambar 18. Create New Folder JCreator

Klik “Yes” untuk membuat sebuah folder baru di C:\Program Files



Gambar 19. Install

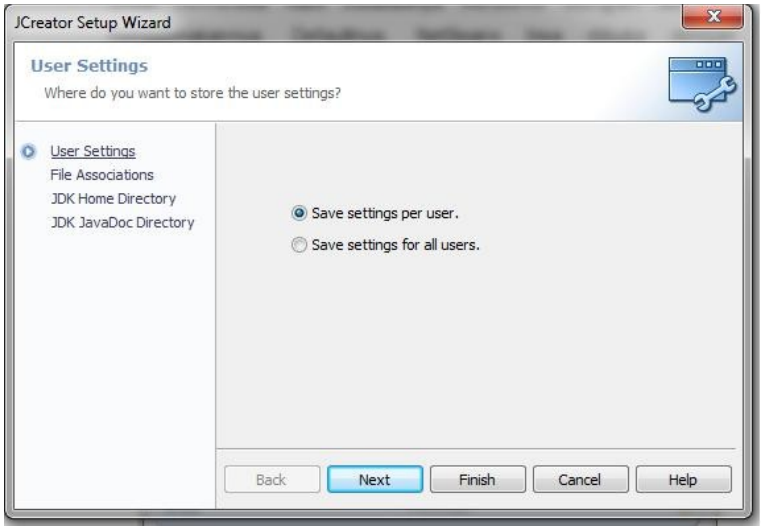
Klik Install dan tunggu proses hinga muncul tampilan seperti di bawah ini.



Gambar 20. Finishing Instalation

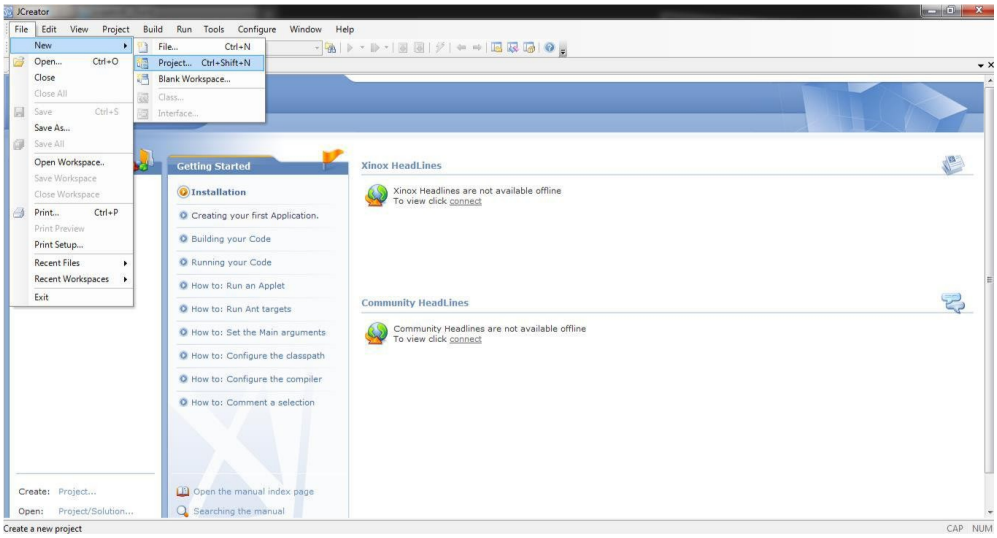
Klik “Finish” untuk menyelesaikan proses instalasi, Beri tanda centang tentang di bagian tulisan “Launch JCreator Pro” untuk membuka langsung program Jcreator dan jangan diberi jika tidak ingin membuka langsung.

23 Membuat Project dengan Jcreator



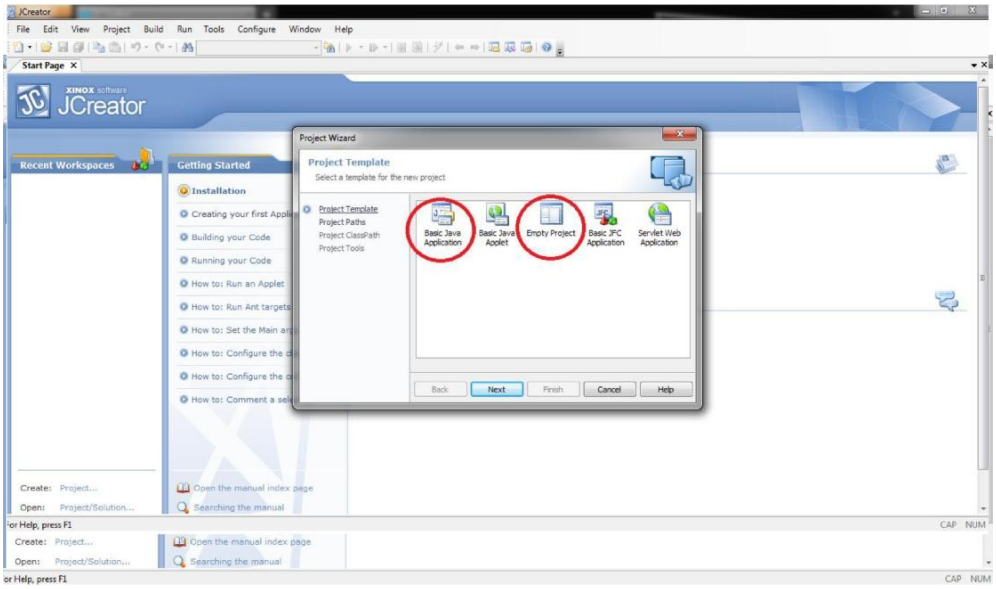
Gambar 21. JCreator Setup Wizard

Ketika pertama kali membuka JCreator maka 37ka nada tampilan seperti ini, klik Finish.



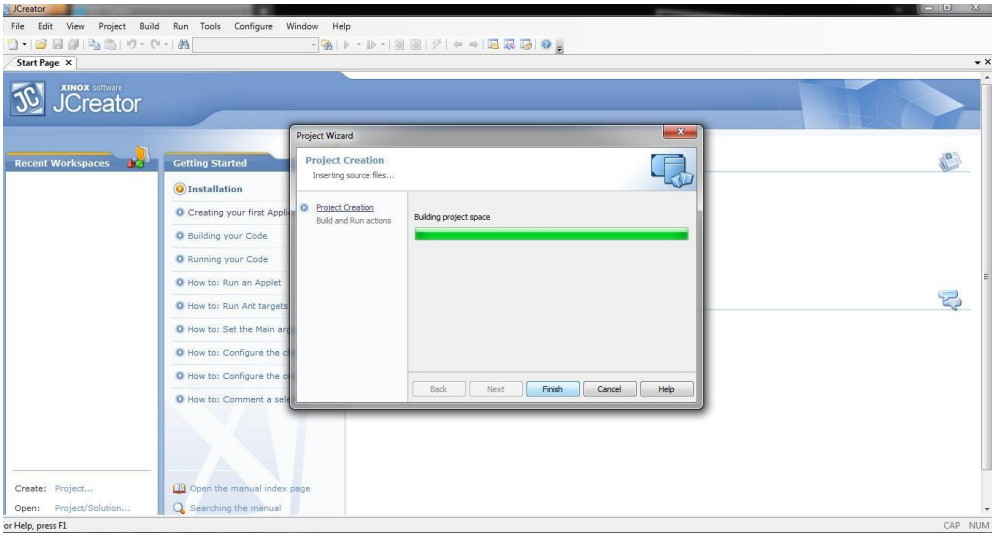
Gambar 22. Main Menu JCreator

Ini adalah tampilan menu utama pada JCreator Pro, klik File di bagian kiri atas, lalu new, dan kemudian klik Project.



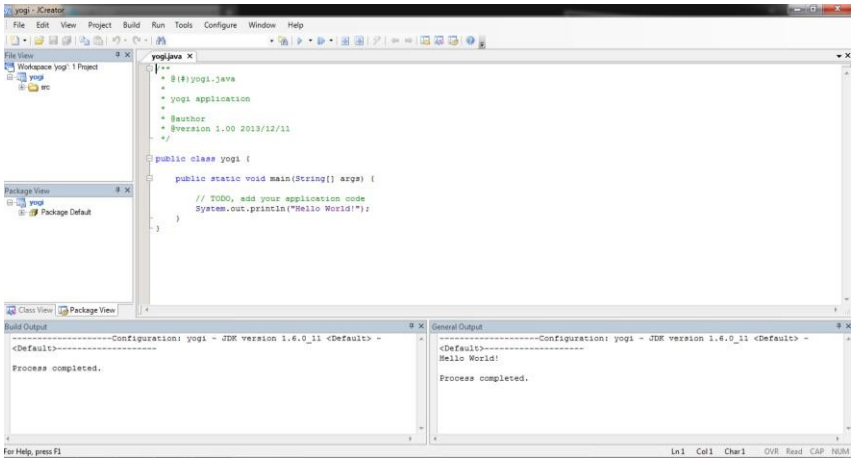
Gambar 23. Project Wizard

Anda akan diminta untuk memilih Object Template, Jika anda pilih Basic Java Application maka akan memberikan project yang instan dengan program Hello World, Jika anda pilih Empty Project maka project belum terisi atau kosong.
Dan pada tampilan ini anda diminta untuk mengisikan sebuah nama project anda pada field name, kemudian klik next.



Gambar 24. Finishing Project

Tunggu proses sampai selesai kemudian klik finish.



Gambar 25. JCreator

Dan inilah tampilan dari Basic Java Application yang berisi program Hello Word.

a. Rangkuman

Arsitektur teknologi Java dibagi menjadi tiga bagian yaitu (1)**Enterprise Java (J2EE)** untuk aplikasi berbasis web, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi. Merupakan superset dari Standar Java (2)**Standar Java (J2SE),** ini adalah yang biasa dikenal sebagai bahasa Java. (3)**Micro Java (J2ME)** merupakan subset dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk wireless device / mobile device. Program Java dijalankan menggunakan interpreter yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi Java *byte codes* dapat dijalankan pada platform yang berbeda-beda.

Langkah pertama dalam pembuatan sebuah program berbasisJava adalah menuliskan kode program pada *text editor*. Contoh *text editor* yang dapat digunakan antara lain : notepad, vi, emacs dan lain sebagainya. Kode program yang dibuat kemudian tersimpan dalam sebuah berkas berekstensi *.java.* Setelah membuat dan menyimpan kode program,kompilasi file yang berisi kode program tersebut dengan menggunakan JavaCompiler. Hasildari kompilasi berupa berkas *byte code* dengan ekstensi *.class.* Berkasyang mengandung *byte code* tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan *platform* yang digunakan.

23 Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

❖ Mengamati benda dan obyek

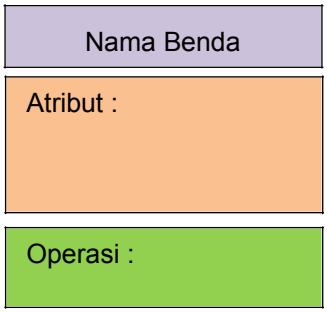
23 Buatlah kelompok dengan anggota 3 – 4 orang

24 Amatilah dan tuliskan 5 benda yang ada disekitar anda

25 Sebutkan ciri-ciri atau atribut masing-masing(minimal 3) dari kelima benda tersebut yang menjadi identitasnya sehingga mudah dikenali.

26 Sebutkan fungsi masing-masing (minimal 2) dari kelima benda tersebut yang merupakan operasi relasi dari ciri-ciri yang sudah teridentifikasi

27 Deskripsikan setiap benda tersebut seperti gambar berikut



6. Buat laporan dan diskusikan dengan teman sekelompok

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama



Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan

❖ Mengamati Bahasa Pemrograman

23 Buatlah kelompok dengan anggota 3 – 4 orang

24 Amatilah dan deskripsikan 2 bahasa pemrograman yang sudah anda kenal

25 Tunjukkanlah perbedaan yang mendasar dari 2 bahasa pemrograman yang sudah anda kenal

26 Sebutkan persamaan fungsi atau kegunaan yang dapat terdapat dalam 2 bahasa pemrograman yang sudah anda ketahui

27 Buatlah tabel perbedaan dan persamaan yang anda dapatkan diskusikan dengan teman sekelompok

No	Bahasa Pemrograman	Sistem Operasi	File Extensi	Jenis Publikasi
1.				
2.				

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

c. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



23 Ilustrasikan dalam dunia nyata apa yang disebut :

23 Objek

24 Atribut

25 Methode

26 Kelas

24 Sebutkan minimal 5 atribut dan minimal 3 method yangmelekat pada diri anda ?

25 Sebutkan paradigma lain dalam bahasa pemrograman selain paradigma berorientasi objek, bandingkan dan sebutkan masing-masing kekurangan dan kelemahannya ?

26 Mengapa saat ini metodologi berorientasi objek berkembang lebih pesat dibandingkan dengan metode-metode yang lain dalam bahasa pemrograman

23 Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Ilustrasi dalam dunia nyata :



23 Objek.....

.....

.....

.....

24 Atribut ….....

.....

.....

.....

25 Methode.....

.....

.....



.....

23552 Kelas…….....

.....

.....

.....

LJ- 02 : 5 Atribut dan 3 method yang melekat pada diri siswa?



.....

.....

.....

.....

.....

.....

.....

.....

.....

LJ- 03 : Paradigma pemrograman berorientasi objek vs pemrograman terstruktur



0 Pemrograman berorientasi obyek

.....

.....

.....

.....

.....

.....

.....

.....

1 Pemrograman terstruktur

.....

.....

.....

.....

.....

.....
.....
.....
.....

LJ- 04: Paradigma pemrograman berorientasi objek lebih banyak digunakan dewasa ini karena :



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....



e. Lembar Kerja Siswa.

0 **Kegiatan Belajar 3 : Perangkat Lunak Pemrograman Berorientasi Obyek**

BAB II
ATURAN DAN DASAR PEMROGRAMAN BERORIENTASI OBYEK

A. Deskripsi

Dalam bab 2 ini akan menjelaskan dan menyajikan konsep pemrograman berorientasi obyek yang terdiri dari 4 kegiatan belajar. Kegiatan belajar 3 akan memahamkan anda tentang paradigma pemrograman berorientasi obyek dan menganalisis perbedaan pemrograman procedural dan pemrograman perorientasi obyek. Kegiatan belajar 4 meliputi penjelasan alur kerja perangkat lunak berorientasi obyek dan melakukan instalasi perangkat lunak. Setiap kegiatan belajar disertai dengan tujuan pembelajaran yang akan dicapai dalam 1 kali tatap muka, uraian materi, tes formatif untuk menguji kompetensi pengetahuan anda, dan tugas atau praktikum(individu dan kelompok) untuk menguji kompetensi keterampilan anda.

B. Kegiatan Belajar

0 Kegiatan Belajar 3 :Dasar dan Aturan Pemrograman Berorientasi Obyek

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 3 ini siswa diharapkan dapat :

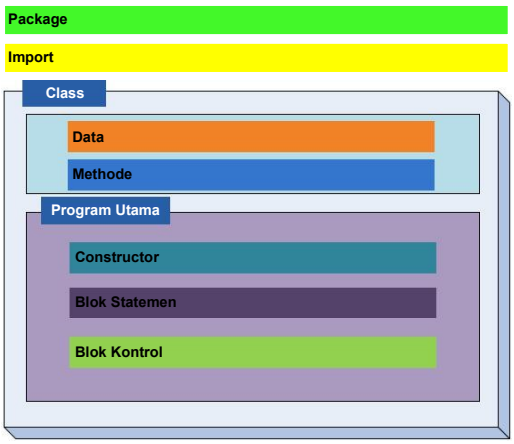
1280 Mengidentifikasi bagian dasar dari program Java.

1281 Memahami perbedaan antara syntax error dan runtime error.

1282 Menganalisis Java literal, keyword,tipe data dasar, dan tipe variabel.

b. Uraian Materi

0 Bagan dasar program java



Gambar 26.Bagian –bagian pemrograman Java

0 Package

Perintah java yang digunakan untuk memberitahukan bahwa suatu class adalah anggota dari package, sedangkan namaPackage dapat berupa susunan direktori tempat dimana file class disimpan atau nama folder.

0 Import

Perintah import digunakan untuk memberitahukan kepada program untuk mengacu pada class-class yang terdapat pada package tersebut dan bukan menjalankan class-class tersebut**.** Dalam program, dapat diimport class-class tertentu saja dan dapat pula mengimport semua class yang terdapat pada package.

1 Class

Merupakan bentuk logis yang menjadi landasan bangun seluruh bahasa pemograman berorientasi object. Class mendefinisikan bentuk dan perilaku object. Class merupakan contoh abstrak dari sebuah object yang telah terbentuk dari proses penyederhanaan. Dengan kata lain class merupakan cikal bakal dari object. Kemudian contoh nyata atau perwujudan dari sebuah object dinamakan instance.

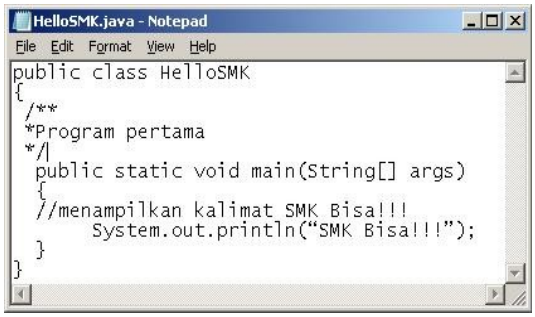
2 Data dan Methode

Data merupakan identitas yang berupa variabel yang menjelaskan properti dari class. Metoda adalah sekumpulan instruksi untuk menjalankan data yang diberi nama dan dapat dipanggil dari manapun di dalam program dengan menuliskan nama metoda tersebut.

3 Program utama

Salah satu metoda yang paling penting di dalam bahasa Java adalah metoda main. Metoda main harus dideklarasikan sendiri oleh programmer di dalam sebuah kelas. Kelas yang mempunyai metoda main disebut dengan kelas main (main class), akan tetapi tidak semua kelas Java harus

mempunyai metoda main. Interpreter Java akan meminta metoda main saat program aplikasi dieksekusi.



Gambar 27.Bagian –bagian pemrograman Java

Barispertamakode:

```
public class HelloSMK
```

menandakan nama class yaitu Hello. Dalam Java , semua kode seharusnya ditempatkan di dalam deklarasi class. Kita melakukannya dengan menggunakan kata kunci class.

Sebagai tambahan, class menggunakan *access specifier* **public**, yang mengindikasikan bahwa class kita mempunyai akses bebas ke class yang lain dari package yang lain pula (package merupakan kumpulan class-class). Baris berikutnya yaitu yang terdiri atas kurung kurawal {menandakan awal blok. Pada kode ini,kita menempatkan kurung kurawal pada baris selanjutnya setelah deklarasi class, bagaimanapun, kita dapat juga meletakkan kurung kurawal ini setelah baris pertama dari kode yang kita tulis. Jadi, kita dapat menulis kode kita sebagai berikut:

```
public class Hello
{
```

atau

```
public class Hello {
```

Tiga baris selanjutnya menandakan adanya komentar Java. Komentar adalah sesuatu yang digunakan untuk mendokumentasikan setiap bagian dari kode yang ditulis. Komentar bukan merupakan bagian dari program itu sendiri, tetapi digunakan untuk tujuan dokumentasi. Komentar itu sendiri dapat ditambahkan pada kode yang Anda tulis sebagai petunjuk yang dapat membantu proses pembelajaran pemrograman yang baik.

```
/**
0 Program Pertama
*/
```

Komentar dinyatakan dengan tanda "/*" dan "*/". Segala sesuatu yang ada diantara tanda tersebut diabaikan oleh compiler Java, dan mereka hanya dianggap sebagai komentar. Baris selanjutnya,

```
public static void main(String[] args) {
```

atau dapat juga ditulis sebagai berikut,



```
public static void main(String[] args)
{
```

Mengindikasikan nama suatu method dalam class **Hello** yang bertindak sebagai **method utama.** Method utama adalah titik awal dari suatu program Java. Semua program kecuali applet yang ditulis dalam bahasa Java dimulai dengan method utama. Yakinkan untuk mengikuti kaidah penulisan tanda yang benar. Baris selanjutnya juga merupakan komentar,

```
//Menampilkan kalimatSMK Bisa!!!
```

Sekarang kita mempelajari 2 cara untuk membuat komentar. Cara pertama adalah dengan menempatkan komenta rdalam /* dan */,dan cara yang lain adalah dengan menuliskan tanda//pada awal komentar. Baris selanjutnya,

```
System.out.println("SMK Bisa!!!");
```

menampilkan teks "HelloWorld!"padalayar.Perintah System.out.println(), menampilkan teks yang diapit oleh tanda *doublepute*("") pada layar. Dua baris terakhir yang terdiri atas dua kurung kurawal digunakan untuk menutup method utama dan masing-masing class secara berurutan.

Program Java yang Anda buat harus selalu diakhiri dengan ekstensi file .java.
Nama File seharusnya sesuai/sama dengan nama class public nya. Sebagai contoh, jika nama class public Anda adalah HelloSMK, Anda harus menyimpan file tersebut dengan nama HelloSMK.java.
3 . Anda harus menulis komentar sebagai penjelasan pada kode yang Anda tulis, yaitu komentar yang berisi keterangan mengenai baris perintah pada class atau apa yang dijalankan oleh method yang Anda tulis tersebut.

0 Perbedaan Syntax Error dan Runtime Error

✓ Syntax Error

Syntax Error biasanya terjadi karena kesalahan penulisan. Mungkin kekurangan sebuah perintah di Java atau lupa untuk menulis tanda titik koma pada akhir pernyataan. Java mencoba untuk megisolasi error tersebut dengan cara menunjukkan baris dari kode dan terlebih dahulu karakter yang salah dalam baris tersebut. Bagaimanapun juga, error belum tentu berada pada titik yang ditunjuk.

Kesalahan umum lainnya adalah dalam kapitalisasi, ejaan, penggunaan dari karakter khusus yang tidak benar, dan penghilangan dari pemberian tanda baca yang sebenarnya. Mari kita mengambil contoh pada program HelloSMK.java, dimana dengan sengaja kita menghilangkan titik koma pada akhir pernyataan dan juga mencoba untuk mengetikkan ejaan yang salah pada sebuah perintah.

Lihatlah pesan error yang ditampilkan setelah peng-compile-an program

dilanjutkan. Pesan error yang pertama memberitahu bahwa di program dijalankan. Pada error yang pertama memberitahu bahwa di program terdapat error pada baris 6. Hal itu menunjuk pada kata setelah static, dimana seharusnya dieja sebagai static. Pada error yang kedua memberitahukan bahwa pada program tersebut kehilangan titik-koma setelah pernyataan.

0 Runtime Error

Sebuah program yang berhasil dikompile belum tentu berhasil di jalankan. Inilah yang dinamakan Run time error, kesalahan ini tidak akan ditampilkan sampai kita menjalankan program tersebut. Hal ini bisa saja terjadi misalnya dikarenakan struktur yang di buat programmer tidak jelas atau mungkin tidak logis.

0 **Java** literal, keyword,tipe data dasar, dan tipe variabel.

✓ Java Keywords

Kata-kunci (keywords) dari sebuah bahasa pemrograman adalah kata-kata yang didefinisikan secara khusus yang hanya dimengerti oleh compiler bahasa pemrograman tersebut, dan tidak dapat digunakan sebagai identitas variabel. Di bawah ini ditampilkan semua kata-kunci (Java keywords) :

Tabel 1. Java Keywords

Abstract	Default	if	private	this
Boolean	Do	implements	protected	throw
Break	Double	import	public	throws
Byte	Else	instanceof	return	transient
Case	Extends	int	short	try
Catch	Final	interface	static	void
Char	Finaly	long	strictfp	volatile
Class	Float	native	super	while
Const	For	new	switch	
Continue	Goto	package	synchronized	assert



Catatan: true, false, dan null bukan termasuk kata kunci akan tetapi mereka termasuk kata-kata khusus, jadi Anda tidak dapat menggunakan mereka sebagai nama variabel pada program Anda.

- **Java Literals**

Literals adalah tanda bahwa tidak terjadi perubahan atau konstan. Macam-macam literals dalam Java adalah : *Integer Literals*, *Floating-Point Literals*, *Boolean Literals*, *Character Literals* dan *String Literals*.

1. Integer Literals

Integer literals dibedakan dalam beberapa format yang berbeda: desimal (berbasis 10), heksadesimal (berbasis 16), and oktal (berbasis 8). Dalam penggunaan tipe data integer pada program, kita harus mengikuti aturan penggunaan beberapa notasi khusus.Untuk angka desimal, kita tidak memerlukan notasi khusus. Kita hanya menulis angka desimal seperti apa adanya. untuk angka heksadesimal, hal itu harus ditandai oleh “0x” atau “0X”. untuk oktal, ditandai oleh“0”. Sebagai contoh, mewakili angka **12**. Penulisan dalam bentuk desimalnya adalah **12**, Sementara dalam heksadesimal, menjadi **0xC**, dan dalam oktal, nilai tersebut sama dengan **014**.

Default tipe data untuk integer literals adalah **int**. Int adalah signed 32-bit value. Pada kasus-kasus tertentu Anda dapat berharap untuk memaksa integer literal untuk menjadi tipe data **long** dengan menambahkan karakter “l” or “L”. tipe data long ditandai oleh ditampilkannya data dalam 64-bit.

2. Floating-Point Literals

Floating point literals mewakili bentuk desimal dengan bagian yang terpisah. Sebagai contoh adalah 3.1415. *Floating point literals* dapat dinyatakan dalam notasi standard atau scientific. Sebagai contoh,

583.45 dinyatakan dalam notasi standard, Sementara 5.8345e2 dinyatakan dalam notasi scientific. Default *Floating point literals* mempunyai tipe data **double** yang dinyatakan dalam 64-bit. Untuk menggunakan ketelitian yang lebih kecil (32-bit) **float**, hanya dengan menambahkan karakter “f” atau “F”.

3. Boolean Literals

Boolean literals hanya memiliki dua nilai, true atau false.

4. Character Literals

Character Literals diwakili oleh karakter single Unicode. Karakter Unicode adalah 16-bit

character set yang menggantikan 8-bit ASCII character set. Unicode memungkinkan

pengunaan simbol dan karakter khusus dari bahasa lain. Untuk menggunakan *character literals*, karakter tersebut di dalam tanda *single pute* (' ') (single quote delimiters). Sebagai contoh huruf a, diwakili sebagai **'a'**. Untuk menggunakan karakter khusus seperti karakter baris baru, *backslash* digunakan diikuti dengan karakter kode. Sebagai contoh, '\n' untuk karakter baris baru atau ganti baris, '\r' untuk menyatakan nilai balik (carriage return), '\b' untuk backspace.

5. String Literals

String literals mewakili beberapa karakter dan dinyatakan dalam tanda double pute(" ")(double quotes). Sebagai contoh *string literal* adalah, **"Hello World".**

E. Tipe Data Primitif

Bahasa pemrograman Java mendefinisikan delapan tipe data primitif. Mereka diantaranya adalah boolean (untuk bentuk logika), char (untuk bentuk tekstual), byte, short, int, long (integral), double and float (floating point).

1. logika - boolean

Tipe data boolean diwakili oleh dua pernyataan : true dan false. Sebagai contoh adalah, boolean result = true; Contoh yang ditunjukkan diatas, mendeklarasikan variabel yang dinamai **result** sebagai tipe data **boolean** dan memberinya nilai **true**.

2. teksual – char

Tipe data character (char), diwakili oleh karakter single Unicode. Tipe data ini harus memiliki ciri berada dalam tanda *single quotes*(' '). Sebagai contoh,

'a' //Huruf a

'\t' //A tab

Untuk menampilkan karakter khusus seperti ' (single quotes) atau " (double quotes), menggunakan karakter escape \. Sebagai contoh,



'\'' //untuk single quotes

'\"' //untuk double quotes

Meskipun String bukan merupakan tipe data primitif (namun merupakan suatu Class),kita akan memperkenalkan mengenai pada bagian ini. String mewakili tipe data yang terdiri atas beberapa karakter. Mereka tidak **termasuk tipe dataprimitif, melainkan suatu class.** Mereka memiliki literal yang terdapat diantara tanda double quotes(“”).

Sebagai contoh,

String message=“Hello world!”

3. Integral –byte, short, int & long

Tipe data integral dalam Java menggunakan tiga bentuk- yaitu desimal,oktal atau heksa desimal. Contohnya,

```
2    //nilai desimal 2
0    //angka 0awal mengindikasikan nilai
oktal 0xBACC //karakter 0x mengindikasikan
nilai heksadesimal
```

Tipe-tipe integral memiliki default tipe data yaitu int. Anda dapat merubahnya ke bentuk long dengan menambahkan huruf l atau L

4. Floating Point –float dandouble

Tipe Floating point memiliki double sebagai default tipe datanya. Floating-point literal termasuk salah satunya decimal point atau salah satu dari pilihan berikut ini,

E or e //(add exponential value) F
or f //(float)
D or d //(double)
Contohnya adalah,

3.14 //nilai floating-point sederhana (a double)
6.02E23 //A nilai floating-point yang besar
2.718F //A nilai float size sederhana
123.4E+306D //A nilai double yang besar dengan nilai redundant D

Pada contoh yang ditunjukkan diatas,23 setelah E pada contoh kedua bernilai positif. Contoh tersebut sama dengan 6.02E+23. Java adalah bahasa pemrograman yang bertipe kuat. Ini maksudnya adalah setiap variabel harus memiliki sebuah tipe yang telah dideklarasikan dan bahasa tersebut memberlakukan pemerikasaan tipe yang kaku. Bahasa Java mempunyai delapan tipe primitif yang ditunjukkan dalam tabel di bawah ini :

Tabel 2. Tipe Data Primitif

Grup	Type Data	Size	Min Value	Max Value
Integral	byte	8 bits	-128	128
	short	16 bits	-32768	32768
	int	32 bits	-2147483648	2147483648
	long	64 bits	-9223372036854775808	9223372036854775808
Real	float	32 bits	± 1.40239846E-45	±3.40282347E+8
	double	64 bits	±4.94065645841246544E-324	±1.79769313486231570E+308
Karakter	char	16 bits	\u0000	\uFFFF
Boolean	boolean	n/a	true atau false	

F.Variabel

Variabel adalah item yang digunakan data untuk menyimpan pernyataan objek. Variabel memiliki tipe data dan nama. Tipe data menandakan tipe nilai yang dapat dibentuk oleh variabel itu sendiri. Nama variabel harus mengikuti aturan untuk identifier.

1. *Deklarasi dan Inisialisasi Variabel*

   Untuk deklarasi variabel adalahsebagai berikut,

   <data tipe><name> [=initial value];

   Catatan: Nilainya berada diantara <> adalah nilai yang disyaratkan, sementara nilai dalam tanda [] bersifat optional. Berikut ini adalah contoh program yang mendeklarasikan dan menginisialisasi beberapa variabel,

Listing Program

```
short x;

int umur;

float gaji;

double data;
```

Inisialisasi variabel dapat dilakukan dengan memberikan nilai pada variabel yang telah dideklarasikan , contoh :



Listing Program

```
int x=21;

int y;

double d = 3.5;

y = (int) d;
```

2. Variabel ReferencedanVariabel Primitif

Sekarang kita akan membedakan dua tipe variabel yang dimiliki oleh program Java. Ada **variabel reference** dan **variabel primitif** .**Variabel primitif** adalah variabel dengan tipe data primitif. Mereka menyimpan data dalam lokasi memori yang sebenarnya dimana variabel tersebut berada.

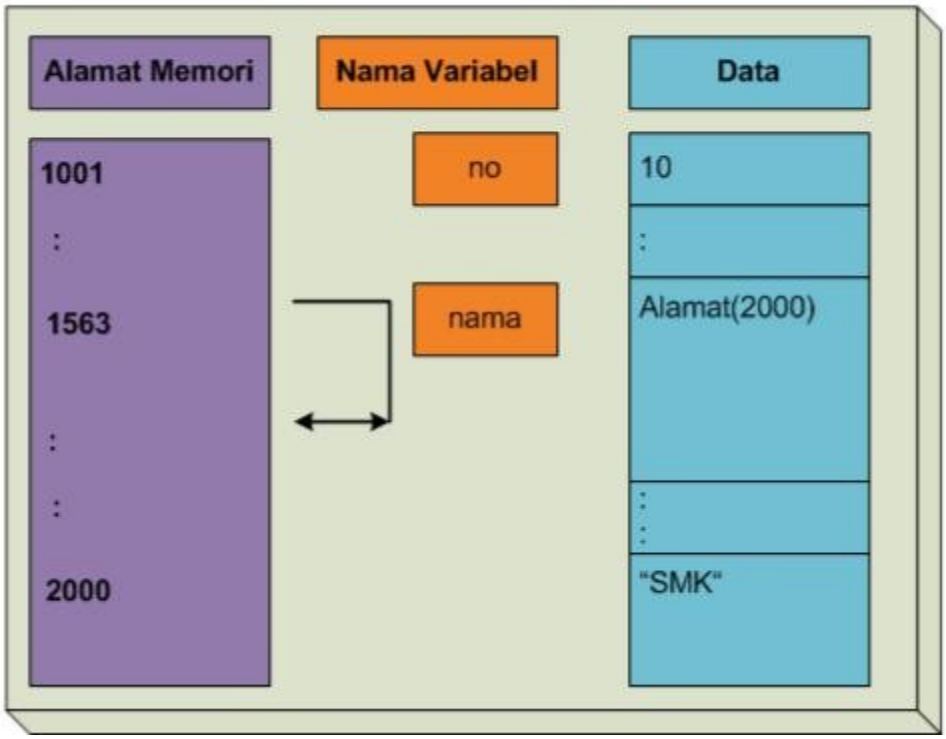
Variabel Reference adalah variabel yang menyimpan alamat dalam lokasi memori. Yang menunjuk ke lokasi memori dimana data sebenarnya berada. Ketika Anda mendeklarasikan variabel padaclass tertentu, Anda sebenarnya mendeklarasikan reference variable dalam bentuk objek dalam classnya tersebut. Sebagai contoh, Apabila kita mempunyai dua variabel dengan tipe data int dan String.

Listing Program

```
int no = 10;

String nama = "SMK ";
```

Dimisalkan ilustrasi yang ditunjukkan dibawah ini adalah memori yang ada pada komputer Anda, dimana Anda memiliki alamat dari setiap sel memorinya, nama variabel dan datanya terbentuk sebagai berikut.



Gambar 28.

c. Rangkuman

Pengembangan berorientasi objek merupakan cara pikir baru tentang perangkat lunak berdasarkan abstraksi yang terdapat dalam dunia nyata. Dalam konteks pengembangan menunjuk pada bagian awal dari siklus hidup pengembangan sistem, yaitu survei, analisis, desain, implementasi, dan pemeliharaan sistem. Hal yang lebih penting dalam pengembangan berorientasi objek adalah konsep mengindentifikasi dan mengorganisasi domain aplikasi dibandingkan dengan fokus penggunaan bahasa pemrograman, berorientasi objek atau tidak.

Object adalah gabungan antara beberapa data dan fungsi yang masing-masing bekerja bersama-sama dan tidak dapat dipisahkan. Gabungan dari data dan fungsi tersebut akan membentuk suatu object-object yang aktif. Dari kumpulan beberapa object yang sama akan membentuk struktur baru yang disebut class.

Pemrograman berorientasi objek (Inggris: object-oriented programming disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.



Pemrograman terstruktur adalah suatu proses untuk mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program. Selain pengertian diatas Pemrograman Terstruktur adalah suatu aktifitas pemrograman dengan memperhatikan urutan langkah-langkah perintah secara sistematis, logis , dan tersusun berdasarkan algoritma yang sederhana dan mudah dipahami.Prinsip dari pemrograman terstruktur adalah Jika suatu proses telah sampai pada suatu titik / langkah tertentu , maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya / kembali lagi ke baris sebelumnya, kecuali pada langkah – langkah untuk proses berulang (Loop).

0 Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

❖ Mengamati benda dan obyek

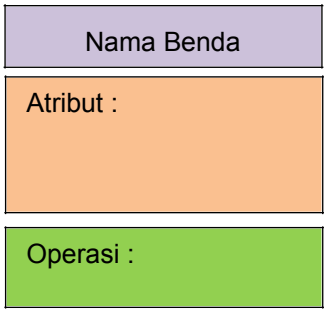
0 Buatlah kelompok dengan anggota 3 – 4 orang

1 Amatilah dan tuliskan 5 benda yang ada disekitar anda

2 Sebutkan ciri-ciri atau atribut masing-masing(minimal 3) dari kelima benda tersebut yang menjadi identitasnya sehingga mudah dikenali.

3 Sebutkan fungsi masing-masing (minimal 2) dari kelima benda tersebut yang merupakan operasi relasi dari ciri-ciri yang sudah teridentifikasi

4 Deskripsikan setiap benda tersebut seperti gambar berikut



6. Buat laporan dan diskusikan dengan teman sekelompok



❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan

❖ Mengamati Bahasa Pemrograman

0 Buatlah kelompok dengan anggota 3 – 4 orang

1 Amatilah dan deskripsikan 2 bahasa pemrograman yang sudah anda kenal

2 Tunjukkanlah perbedaan yang mendasar dari 2 bahasa pemrograman yang sudah anda kenal

3 Sebutkan persamaan fungsi atau kegunaan yang dapat terdapat dalam 2 bahasa pemrograman yang sudah anda ketahui

4 Buatlah tabel perbedaan dan persamaan yang anda dapatkan diskusikan dengan teman sekelompok

No	Bahasa Pemrograman	Sistem Operasi	File Extensi	Jenis Publikasi
1.				

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama



e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



0 Ilustrasikan dalam dunia nyata apa yang disebut :

- 0 Objek
- 1 Atribut
- 2 Methode
- 3 Kelas

1 Sebutkan minimal 5 atribut dan minimal 3 method yangmelekat pada diri anda ?

2 Sebutkan paradigma lain dalam bahasa pemrograman selain paradigma berorientasi objek, bandingkan dan sebutkan masing-masing kekurangan dan kelemahannya ?

3 Mengapa saat ini metodologi berorientasi objek berkembang lebih pesat dibandingkan dengan metode-metode yang lain dalam bahasa pemrograman.

0 ***Lembar Jawaban Test Formatif***

(LJ). **LJ- 01 :**Ilustrasi dalam dunia nyata :



0.0 Objek.....
.....
.....
.....

0.1 Atribut .....
.....
.....
.....

0.2 Methode.....
.....
.....
.....

0.3 Kelas…….....
.....
.....
.....

LJ- 02 : 5 Atribut dan 3 method yang melekat pada diri siswa?



.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 : Paradigma pemrograman berorientasi objek vs pemrograman terstruktur



0 Pemrograman berorientasi obyek

.....
.....
.....
.....
.....
.....
.....
.....

1 Pemrograman terstruktur

.....
.....
.....
.....
.....
.....
.....
.....
.....



LJ- 04: Paradigma pemrograman berorientasi objek lebih banyak digunakan dewasa ini karena :



.....
.....
.....
.....
.....
.....
.....
.....
.....

g. Lembar Kerja Siswa.

0 Kegiatan Belajar 4 :Dasar dan Aturan Pemrograman Berorientasi Obyek

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 4 ini siswa diharapkan dapat :

0 Mengidentifikasi operator dalam program Java.

1 Menyajikan dalam perbedaan antara syntax error dan runtime error.

b. Uraian Materi

1) Operator

Dalam Java, ada beberapa tipe operator. Ada operator aritmatika, operator relasi, operator logika, dan operator kondisi. Operator ini mengikuti bermacam-macam prioritas yang pasti sehingga compilernya akan tahu yang mana operatoruntuk dijalankan lebih dulu dalam kasus beberapa operator yang dipakai bersama-sama dalam satu pernyataan.

1.Operator Aritmatika

Berikut ini adalah dasar operator aritmatik yang dapat digunakan untuk membuat suatu program Java,

Tabel 3. Operator Aritmatika dan Fungsi-Fungsinya

Operator	*Penggunaan*	*Keterangan*
+	op1+ op2	Menambahkan op1 dengan op2
*	op1*op2	Mengalikan op1 dengan op2
/	op1/op2	Membagi op1 dengan op2
%	op1%op2	Menghitung sisa dari pembagian op1 denganop2
-	op1-op2	Mengurangkan op2 dari op1

2.Operator Increment dan Decrement

Dari sisi operator dasar aritmatika, Java juga terdiri atas operator*unary increment* (++) dan operator *unary decrement* (--).Operator increment dan decrement menambah dan mengurangi nilai yang tersimpan dalam bentuk variabel angka terhadap nilai 1. Sebagai contoh,pernyataan,



count = count + 1

count++;

Tabel 4. Operator Increment dan Decrement

Operator	*Penggunaan*	*Keterangan*
++	op++	Menambahkan nilai 1 pada op; mengevaluasi nilai op sebelum diincrementasi/ ditambahkan
++	++op	Menambahkan nilai 1 pada op; mengevaluasi nilai op setelah diincrementasi/ ditambahkan
--	op--	Mengurangkan nilai 1 pada op; mengevaluasi nilai op sebelum didecrementasi/ dikurangkan
--	--op	Mengurangkan nilai 1 pada op; mengevaluasi nilai op setelah didecrementasi/ dikurangkan

Operator increment dan decrement dapat ditempatkan sebelum atau sesudah operand. Ketika digunakan sebelum operand, akan menyebabkan variabel diincrement atau didecrement dengan nilai1, dan kemudian nilai baru digunakan dalam pernyataan dimana dia ditambahkan. Sebagaicontoh,

Listing Program

```
int i =10;
int j = 3;
int k = 0;
k = ++j + i;
```

Ketika operator increment dan decrement ditempatkan setelah operand, nilai variabel yang lama akan digunakan lebih dulu dioperasikan lebih dulu terhadap pernyataan dimana dia ditambahkan. Sebagaicontoh,

Listing Program

```
int i = 10,
    int j = 3;
```

```
int k = 0;
k = j++ + i;
```

3.Operator Relasi

Operator Relasi membandingkan dua nilai dan menentukan keterhubungan diantara nilai- nilai tersebut. Hasil keluarannya berupa **nilai boolean** yaitu true atau false.

Tabel 5. Operator Relasi

Operator	*Penggunaan*	*Keterangan*
>	op1>op2	op1 lebih besar dari op2
>=	op1>= op2	op1 lebih besar dari atau sama dengan op2
<	op1<op2	op1 kurang dari op2
<=	op1<= op2	op1kurang dari atau sama dengan op2
==	op1== op2	op1 sama dengan op2
!=	op1!= op2	op1 tidak sama dengan op2

4.Operator logika

Operator logika memiliki satu atau lebih operand Boolean yang menghasilkan nilai boolean. Terdapat enam operator logika yaitu :

0 (logika AND), & (Boolean logika AND), || (logika OR), | (Boolean logika inclusive OR), ^ (Boolean logika exclusive OR), dan ! (logika NOT).

Pernyataan dasar untuk operasi logika adalah,x1 op x2

Dimana x1,x2 dapat menjadi pernyataan boolean. Variabel atau konstanta, dan op adalah salah satu dari operator &&, &, ||, | atau ^. Tabel kebenaran yang akan ditunjukkan selanjutnya, merupakan kesimpulan dari hasil dari setiap operasi untuk semua kombinasi yang mungkin dari x1 dan x2.



4.1. (logika AND) dan & (Boolean logika AND)

Berikut ini adalah tabel kebenaran untuk && dan &,

Tabel 6. Tabel Kebenaran Logika AND

x1	*x2*	*Hasil*
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

Perbedaan dasar antara operator && dan & adalah bahwa && mensupports **short-circuit evaluations** (atau evaluasi perbagian), sementara operator & tidak. Apa arti dari pernyataan tersebut?

Diberikan suatu pernyataan,
exp1 && exp2

&& akan mengevaluasi pernyataan exp1, dan segera mengembalikan nilai false dan menyatakan bahwa exp1 bernilai false. Jika exp1 bernilai false, operator tidak akan pernah mengevaluasi exp2 karena hasil operasi operator akan menjadi false tanpa memperhatikan nilai dari exp2. Sebaliknya, operator & selalu mengevaluasi ke dua nilai dari exp1danexp2 sebelummengembalikansuatunilaijawaban.

4.2.|| (logikaOR) dan|(booleanlogikainclusive OR)

Berikutiniadalahtabel kebenaranuntuk||dan|,

Tabel 7. Tabel Kebenaran Logika OR

x1	*x2*	*Hasil*
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

Perbedaan dasarantara operator|| dan | adalah bahwa || mendukung short-circuit evaluations (atau proses evaluasi sebagian), sementara | tidak. Apa maksud dari pernyataan tersebut?

Diberikan suatu pernyataan,

exp1 || exp2

|| akan mengevaluasi pernyataan exp1,dan segera mengembalikan nilai true dan menyatakan bahwa exp1 bernilaitrue. Jika exp1 bernilai true, operator tidak akan pernah mengevaluasi exp2 karena hasil dari operasi operatorakan bernilai true tanpa memperhatikan nilai dari exp2. Sebaliknya, operator | selalu mengevaluasi ke dua nilai dari exp1 and exp2 sebelum mengembalikan suatu jawaban suatu nilai.

4.3.^ (Boolean logika Exclusive OR)

Berikut ini adalah tabel kebenaran untuk^,

Tabel 8. Tabel Kebenaran Logika EX-OR

x1	*x2*	*Hasil*
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

Hasil operasi operator exclusive OR adalah TRUE, jika dan hanya jika satu operand bernilai TRUE dan yang lain bernilai False. Catatan jika kedua operand harus selalu dievaluasi untuk menjumlahkanhasildarisuatuexclusiveOR.

4.4!(logika NOT)

Logika NOTdigunakan dalamsatuargumen,dimana argumentersebutdapatmenjadisuatu pernyataan,variabelataukonstanta.Berikutini adalahtabel kebenaranuntukoperatornot!,

Tabel 9. Tabel Kebenaran Logika NOT

x1	*Hasil*
TRUE	FALSE
FALSE	TRUE

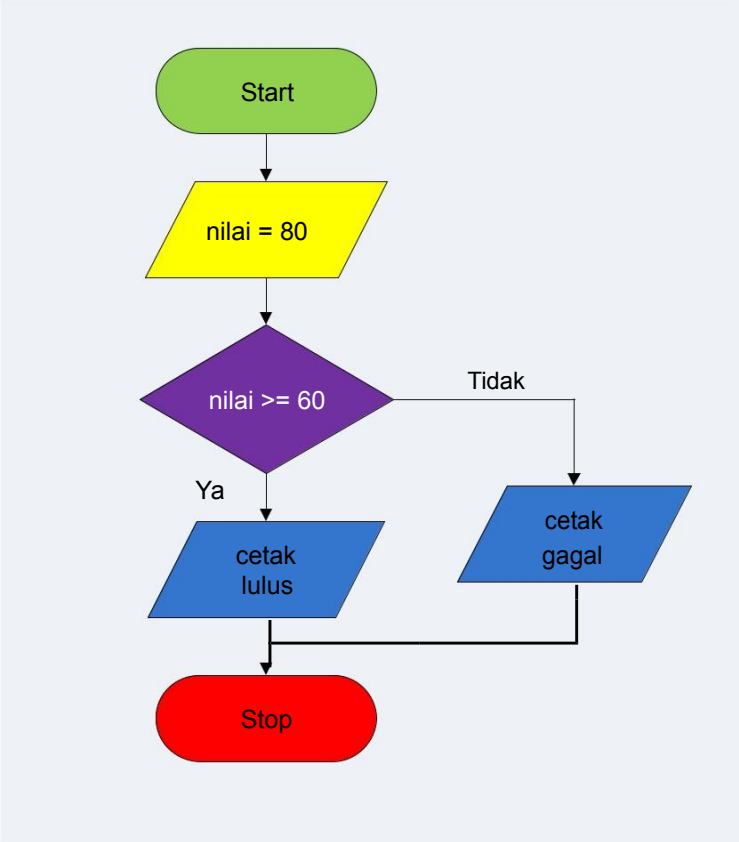
5.Operator Kondisi(?:)

Operatorkondisi **?:** adalah operatorternary.Berarti bahwa operatorini membawa tiga argumen yang membentuk suatu ekspresi bersyarat.Strukturpernyataan yang menggunakan operatorkondisi

adalah,

exp1?exp2:exp3

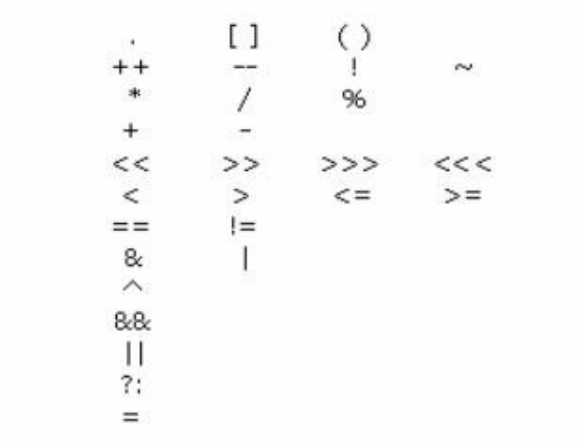
Dimananilaiexp1adalahsuatupernyataanbooleanyangmemilikihasilyang salahsatunya harusberupanilai trueataufalse. Jikaexp1bernilaitrue,exp2merupakanhasiloperasi.Jikabernilaifalse,kemudianexp3 merupakanhasiloperasinya.Berikutiniadalahflowchartyangmenggambarkanbagaimana operator?:bekerja,



Gambar 29. Flowchart Operator Kondisi

6.Operator Precedence

Operatorprecedence didefinisikan sebagai perintah yang dilakukan compilerketika melakukanevaluasiterhadapoperator,untukmengajukanperintahdenganhasilyangtidak ambigu/hasil yangjelas.



Gambar 30. Operator Presedence

Diberikan pernyataanyangmembingungkan,

6%2*5+4/2+88-10

Kita dapat menuliskan kembali pernyataan diatas dan menambahkan beberapa tanda kurung terhadap operator precedence,

((6%2)*5)+(4/2)+88-10;

c. *Rangkuman*

Ada operator aritmatika, operator relasi, operator logika, dan operator kondisi. Operator ini mengikuti bermacam-macam prioritas, operator aritmatika umumnya digunaka n untuk operasi matematika seperti pembagian, perkalian, dan lain -lain. Operator Increment dan Decrement berfungsi untuk menambah dan mengurangi nilai yang tersimpan, operasi relasi digunakan untuk membandingkan dua nilai untuk menentukan keterhubungan diantara nilai-nilai tersebut. Operator logika memiliki lebih dari satu operand boolean yang menghasilkan nilai boolean true dan false. Operator kondisi bersifat ternary, jadi operator ini membawa tiga argumen yang membentuk suatu ekspresi bersyarat. Operator Precedence berfungsi sebagai alat evaluasi terhadap operator untuk mengajukan perintah dengan hasil yang tidak ambigu atau hasil yang jelas.



23 Tugas

Tugas 1

Tulislah suatu program yang bisa mengurutkan tiga *integer.* Ketiga integer dientrikan dari kotak dialog masukan dan disimpan dalam variabel **angka1** , **angka2** , dan **angka3**. Program kemudian mengurutkan angka-angka tersebut sehingga angka1 ≤ angka2 ≤ angka3.

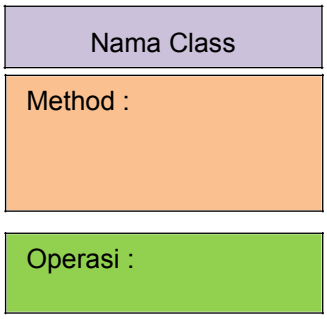
❖ Mengamati Listing Program dan Output Program

5888 Menentukan nama Class

5889 Menentukan variabel yang digunakan

5890 Menentukan nama Method

5891 Gambar Class Diagram



23 Buatlah listing program

24 Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Buatlah program kalkulator sederhana yang bisa menjumlah, mengurangi, mengkalikan dan membagi.

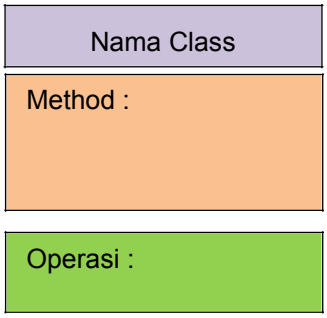
❖ Mengamati Listing Program dan Output Program

□Ã□Ã□Ã□Ã□Ã□Ã□ÈÃÀrÃÃÃÃÃ□Ã□Ã□Ã□Ã□Ã□Ã□ÈÈÃ0 Menentukan nama Class

□Ã□Ã□Ã□Ã□Ã□Ã□ÈÃÀrÃÃÃÃÃ□Ã□Ã□Ã□Ã□Ã□Ã□ÈÈÃ1 Menentukan variabel yang digunakan

□Ã□Ã□Ã□Ã□Ã□Ã□ÈÃÀrÃÃÃÃÃ□Ã□Ã□Ã□Ã□Ã□Ã□ÈÈÃ2 Menentukan nama Method

□Ã□Ã□Ã□Ã□Ã□Ã□ÈÃÀrÃÃÃÃÃ□Ã□Ã□Ã□Ã□Ã□Ã□ÈÈÃ3 Gambar Class Diagram



0 Buatlah listing program

1 Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	



9.

10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



1.Jelaskan secara singkat apa yang disebut :

a.Operasi Aritmatika

b.Operator Increment dan Decrement

c.Operator Relasi

d.Operator Logika

0.0 Sebutkan minimal 5 operator pada java yang anda ketahui

0.1 Sebutkan enam operator logika yang kamu ketahui

0.2 Operator kondisi adalah ternary, apakah yang dimaksud dengan ternary dan beri contohnya.

0 Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Jelaskan secara singkat apa yang disebut:



a)Operator Aritmatika

.....

.....

.....

.....

b) Operator Increment dan Decrement

….....

.....

.....

.....

c)Operator Logika

.....

.....

.....

.....

0 Kelas…….....

.....

.....

.....

LJ- 02 : Sebutkan minimal 5 operator pada java yang anda ketahui !



.....

.....

.....

.....

.....

.....

.....

.....

.....

LJ- 03 : Sebutkan enam operator logika yang kamu ketahui !



.....

.....

.....

.....

.....

.....

.....



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 04: Operator kondisi adalah ternary, apakah yang dimaksud dengan ternary dan beri contohnya. !



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

g. Lembar Kerja Siswa.



0 Kegiatan Belajar 5 :Dasar dan Aturan Pemrograman Berorientasi Obyek

a. *Tujuan Pembelajaran*

Setelah mengikuti kegiatan belajar 5 ini siswa diharapkan dapat :

□À——Ã□Ã□Ã□Ã□Ã□Ã□Ã□Ã□Ã□Ã□ĭ□Ã□Ã□Ã□Ã□Ã□Ã□0 Memahami struktur kontrol pemilihan (if, else,switch)

□À——Ã□Ã□Ã□Ã□Ã□Ã□Ã□Ã□Ã□Ã□ĭ□Ã□Ã□Ã□Ã□Ã□Ã□1

Menggunakanstruktur kontrol pemilihan (if, else,switch) yang digunakan untuk memilih blok kode yang akan dieksekusi

0 *Uraian Materi*

Struktur kontrol pemilihan adalah pernyataan dari Java yang mengijinkan user untuk memilih dan mengeksekusi blok kode spesifik dan mengabaikan blok kode yang lain.

0 *Statementif*

Pernyataan *if* akan menentukan sebuah pernyataan(ataublokkode) yang akan eksekusi jika dan hanya jika persyaratan bernilai benar *(true).* Bentuk dari pernyataan if,

Sintaks Perintah I

```
if(boolean_expression)
statement;
```

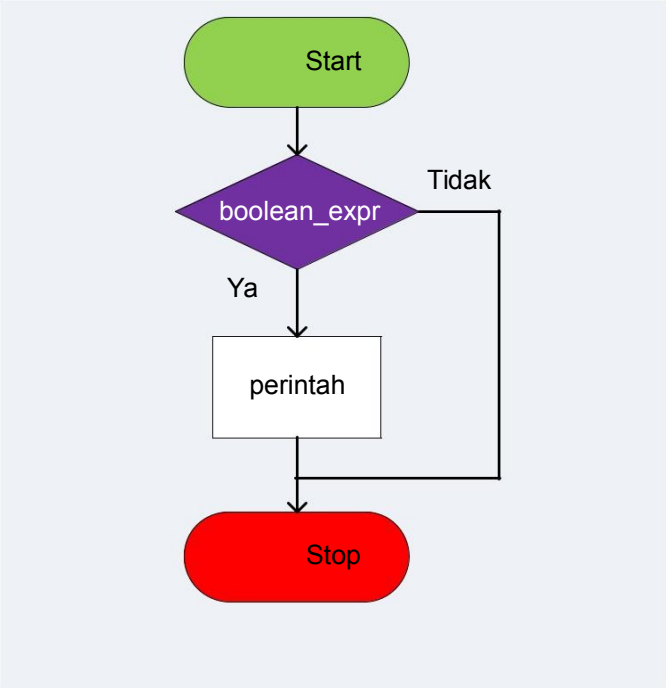
Atau

Sintaks Perintah I

```
if(boolean_expression)
{
      statement1;
    statement2;
}
```

dimana,

*boolean_expression*adalahsebuahpenyataanlogika*(true/false)*atauvari abel bertipe*boolean*.



Gambar 31. Flowchart Statement if

PetunjukPenulisanProgram:

*0 **Boolean_expression** pada pernyataan if harus merupakan nilai boolean).Hal ini berarti persyaratan harus bernilai **true** atau **false**.*

1 Masukkan statement didalam blok if. Contohnya,

```
if(boolean_expression){
//statement1;
//statement2;
}
```

Statementif-else

Pernyataan *if-else* digunakan apabila kita ingin mengeksekusi beberapa pernyataan dengan kondisi *true* dan pernyataan yang lain dengan kondisi *false*.

Bentuk statement if-else,

Sintaks Perintah I

```
if(boolean_expression)
statement;
```

Berikut ini contoh code statement if-else,

Listing Program

```
intgrade=68;
```

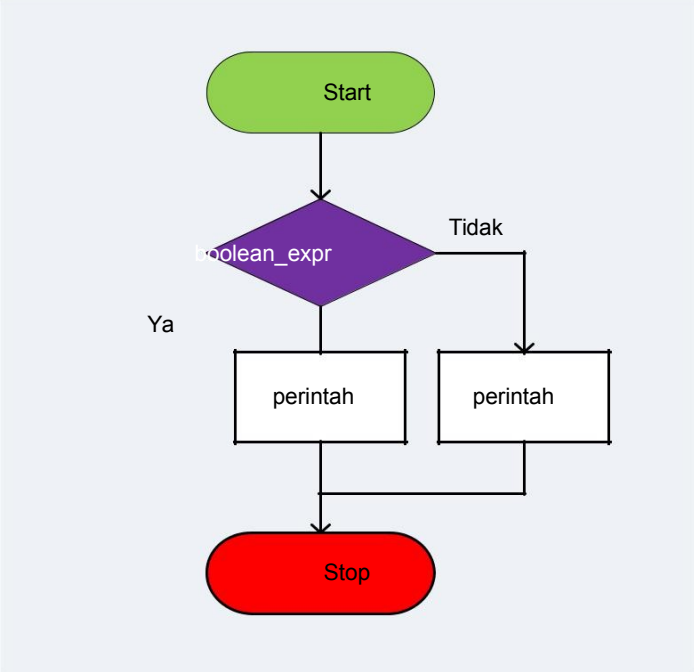


```
if(grade>60)
System.out.println("Congratulations!");
else
System.out.println("Sorryyoufailed");
```

atau

Listing Program

```
intgrade=68;
if(grade>60)
{
  System.out.println("Congratulations!");
  System.out.println("You passed!");
}
else
{
}
System.out.println("Sorryyoufailed");
```



Gambar32.FlowchartStatementIf-Else

PetunjukPenulisanProgram:
0 Untukmenghindarikebingungan, selaluletakkansebuahpernyataanataubeberapa pernyataandidalamblokif-elsedidalamtandakurawal{},

3.Statementif-else-if

Pernyataan pada bagian kondisi *else* dari blok *if-else* dapat menjadi struktur *if-else* yang lain. Kondisi struktur seperti ini mengijinkan kita untuk membuat seleksi persyaratan yang lebih kompleks.Bentuk statement if-else if,

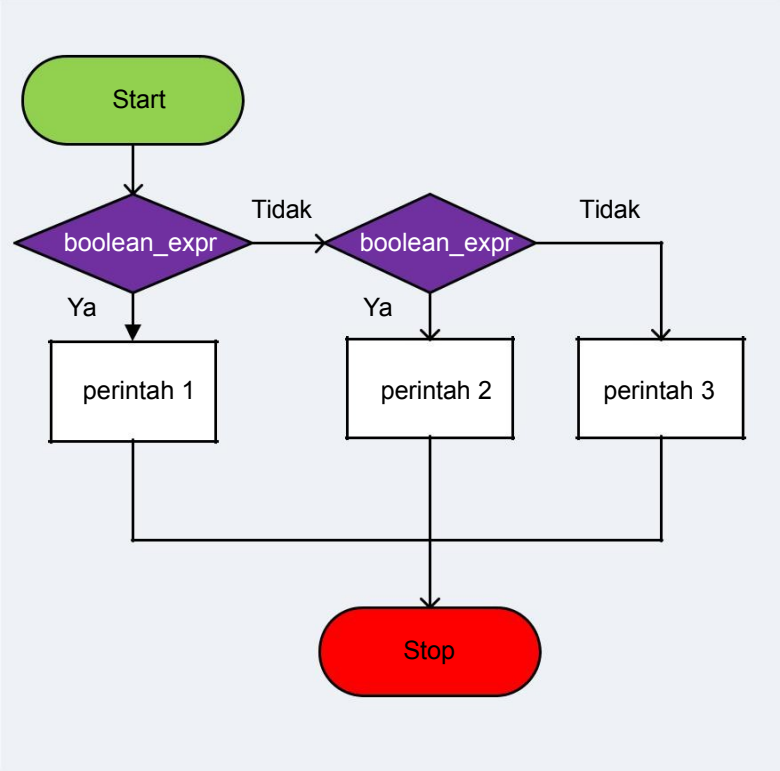
Sintaks perintah If else

```
if(boolean_expression1)
statement1;
elseif(boolean_expression2)
statement2;
else
statement3;
```

Sebagai catatan : anda dapat memiliki banyak blok else-if sesudah



pernyataan *if*.Blok *else* bersifat opsional dan dapat dihilangkan. Pada contoh yang ditampilkan diatas, jika *boolean_expression1* bernilai *true*, maka program akan mengeksekusi *statement1* dan melewati pernyataan yang lain. Jika *boolean_expression2* bernilai *true*, maka program akan mengeksekusi *statement2* dan melewati *statement3*.



Gambar33.FlowchartStatementIf-Else-If

Berikutinicontohcodestatement*if-else-if*

Listing Program

```
intgrade=68;

if(grade>90)

{

System.out.println("Verygood!");

}

elseif(grade>60)

{

System.out.println("Verygood!");
```

```
}
else{}
System.out.println("Sorryyoufailed");
```

5.Statementswitch

Cara lain untuk membuat cabang adalah dengan menggunakan kata kunci ***switch***.*Switch* mengkonstruksikan cabang untuk beberapa kondisi dari nilai.Bentuk statement switch,

Sintaks Perintah Switch

```
switch(switch_expression)
{
casecase_selector1:
statement1;
statement2;
casecase_selector2:
statement1;
statement2;
break;
default:
}
statement1;
statement2;
break;
```

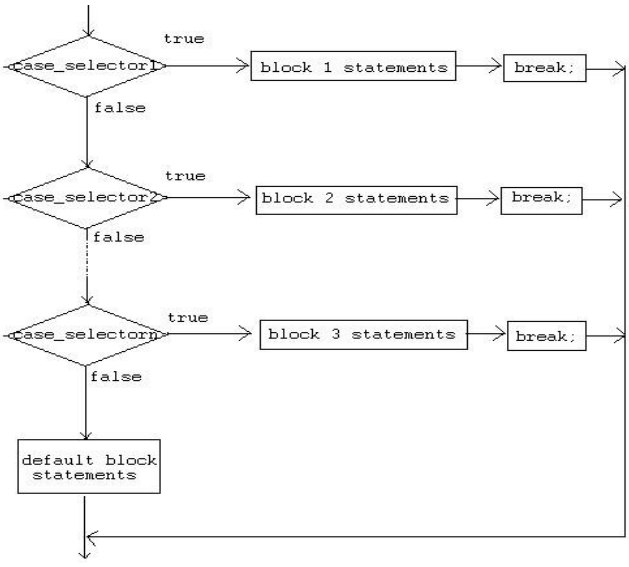
switch_expression adalah ekspresi **integer** atau **karakter** dan *case_selector1*, *case_selector2* dan seterusnya adalah konstanta unik dari nilai *integer* atau karakter. Ketika pernyataan switch ditemukan pada potongan kodeprogram, java pertamakali akan memeriksa *switch_expression*,dan menuju ke *case* yang akan menyamakan nilai



yang dimiliki oleh *switch_expression*.Selanjutnya program akan mengeksekusi pernyataan pada dari kode setelah *case* yang ditemukan sampai menemui pernyataan *break*, selanjutnya akan mengabaikan pernyataan yang lainnya hingga akhir dari struktur dari pernyataan *switch*.Jika tidak ditemui *case* yang cocok, maka program akan mengeksekusi blok *default*. Sebagai catatan, bahwa bagian blok *default* adalah opsional. Sebuah pernyataan *switch* bias jadi tidak memiliki blok kode *default*.

CATATAN:

- Tidak seperti pada pernyataan ***if***, beberapa pernyataan pada struktur pernyataan *switch* akan dieksekusi tanpa memerlukan tanda kurung kurawal (**{}**).
- Ketika sebuah *case* pada pernyataan *switch* menemui ke cocokan, semua pernyataan pada case tersebut akan dieksekusi. Tidak hanya demikian, pernyataan lain yang berada pada case yang sesuai juga akan dieksekusi.
- Untuk menghindari program mengeksekusi pernyataan pada *case* berikutnya, kita menggunakan pernyataan ***break*** sebagai pernyataan akhir pada setiap blok *case*.



Gambar 34.Flowchart Statement Switch

PetunjukPenulisanProgram:

Menentukan penggunaan pernyataan if atau pernyataan switch adalah sebuah keputusan programmer. Programmer dapat menentukan pernyataan yang mana yang akan dipakai berdasarkan kemudahan membaca program dan faktor-faktor yang lain.

Pernyataan if dapat digunakan untuk membuat keputusan berdasarkan rentang nilai
Tertentu atau kondisi tertentu, sedangkan pernyataan switch membuat keputusan hanya berdasarkan nilai unik dari tipe integer atau karakter.

0 Rangkuman

Pernyataan if digunakan untuk membandingkan suatu permasalahan atau objek. If ada bermacam-macam diantaranya, statement if, statement if-else, statement if-else-if, statement switch. Statement if akan mengeksekusi pernyataan hanya jika if bernilai benar atau true. Statement if else digunakan untuk mengeksekusi dua kondisi benar atau salah, true atau false, statement if-else-if bisa digunakan untuk mengeksekusi sebuah kondisi yang lebih dari dua kondisi, dalam statement ini cara penulisan harus berhati-hati. Statement switch hampir sama dengan if-else-if hanya saja swicth mengkonstruksikan cabang untuk beberapa kondisi nilai. Untuk menghentikan program switch menggunakan perintah ***break*** pada case yang sesuaijuga akan dieksekusi.

0 Tugas

Tugas 1

Buatlah suatu program yang berfungsi mengecek suatu nilai. Jika nilai lebih dari 75 cetak “Lulus”, jika nilai kurang dari 75 cetak “Gagal”.



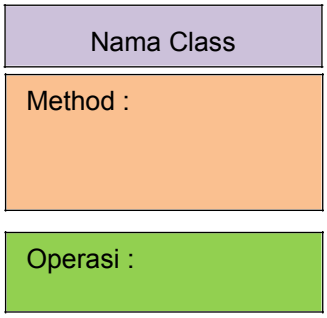
❖ Mengamati Listing Program dan Output Program

0 Menentukan nama Class

1 Menentukan variabel yang digunakan

2 Menentukan nama Method

3 Gambar Class Diagram



0 Buatlah listing program

1 Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Buatlah program pendaftaran siswa baru di sekolah yang memilih jurusan di sekolah tersebut. Terdapat 5 jurusan RPL, Animasi, TKJ, Multimedia, Otomotif. Jika pendaftar memilih salah satu jurusan, cetak nama jurusan yang dipilih.

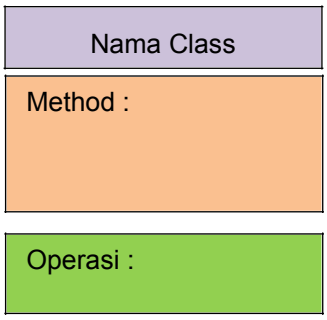
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	



9.

10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Apa yang dimaksud dengan :

Statement if

Statement if-else

Statement if-else-if

Statement switch

Apakah perbedaan menggunakan statement if dan statement switch dalam penulisan program yang sama, dan apakah hasilnya berbeda ?

Apakah kelebihan menggunakan statement switch debandingkan dengan menggunakan statementf if ?

Amati program berikut, dan tulislah apa outputnya !

Listing Program

```
public class coba{
public static void main (String [] args){
int a=3;
switch (a) {
     case 1:
        System.out.println("Nilaia=1");
break;
     case 2:
```

```
            System.out.println("Nilai a=2");
   break;
        case 3:
           System.out.println("Nilai
   a=3");break;
        default:
           System.out.println("Nilai a=4");
   break
             }
       }
}
```

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apa yang dimaksud dengan :



Statement if

.....

.....

.....

.....

Statement if-else

.....

.....

.....

.....

Statement if-else-if

.....

.....

.....

.....

Statement switch

.....

.....



.....

.....

LJ- 02 : Apakah perbedaan menggunakan statement if dan statement switch dalam penulisan program yang sama, dan apakah hasilnya berbeda ?



.....

.....

.....

.....

.....

.....

.....

.....

LJ- 03 : Apakah kelebihan menggunakan statement swicth debandingkan dengan menggunakan statementf if



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LJ- 04: Ouput program :



.....

.....

.....

g. Lembar Kerja Siswa.



Kegiatan Belajar 6 :Dasar dan Aturan Pemrograman Berorientasi Obyek

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 6 ini siswa diharapkan dapat :

Memahami struktur kontrol pengulangan (while, do-while, for)

Menggunakanstruktur kontrol pengulangan (while, do-while, for) untuk menjalankan blok tertentu pada program beberapa kali

Uraian Materi

Struktur kontrol pengulangan adalah berupa pernyataan dari Java yang mengijinkan kita untuk mengeksekusi blok code berulang-ulang sesuai dengan jumlah tertentu yang diinginkan. Ada tiga macamj enis dari struktur kontrol pengulangan yaitu *while, do- while,dan for-loops*.

1. whileloop

Pernyataan *whileloop* adalah pernyataan atau blok pernyataan yang diulang - ulang sampai mencapai kondisi yang cocok. Bentuk pernyataan while,

```
while(boolean_expression){ statement1; statement2;}
```

Pernyataan di dalam *whileloop* akan di eksekusi berulang-ulang selama kondisi *boolean_expression* bernilai benar (*true*). Contoh,pada kode dibawah ini,

Listing Program

```
inti=4;

   while(i>0)

   {

   System.out.print(i);

   i--;

   }
```

Contoh di atas akan mencetak angka 4321 pada layar. Perlu dicatat jika bagia ni --; dihilangkan , akan menghasilkan pengulangan yang terus menerus (**infiniteloop**). Sehingga, ketika menggunakan *whileloop* atau bentuk pengulangan yang lain, pastikan Anda memberikan pernyataan yang membuat pengulangan berhenti pada suatu kondisi.

Berikut ini adalah beberapa contoh *whileloop*,

Listing Program

```
intx=0;

   while(x<10)

   {

   System.out.println(x);

   x++;

   }
```

Listing Program

```
//infiniteloop

   while(true)  System.out.println("hello");
```

Listing Program

```
//noloops

   //statementisnotevenexecuted
   while(false)

   System.out.println("hello");
```

2. do-whileloop

*Do-whileloop*miripdengan*while-loop*. Pernyataandidalam*do-whileloop*akandieksekusibeberapakaliselamakondisibernilaibenar(*true*).Perbedaanantara*while*dan*do-whileloop*adalahdimanapernyataandidalam*do-while loop*akandieksekusisedikitnya**satukali.**



Sintaks do-while loop

```
do{
   statement1;
   statement2;
   ...
   }while(boolean_expression);
```

Pernyataan di dalam *do-while loop* akan dieksekusi pertama kali, dan akan dievaluasi kondisi dari *boolean_expression*. Jika nilai pada *boolean_expression* tersebut bernilai *true*, pernyataan di dalam *do-whileloop* akan dieksekusi lagi.

Berikut ini beberapa contoh do-while loop:

Listing Program

```
   intx=0;
   do
   {
   System.out.println(x);
   x++;
   }while(x<10);
```

Contoh ini akan memberikan output 0123456789 pada layar.

Listing Program

```
   //infiniteloop
   do{
   System.out.println("hello");
   }
   while(true);
```

Contoh di atas akan melakukan pengulangan terus menerus yang menulis kata “hello” pada layar.

Listing Program

```
//oneloop

   //statementisexecutedonce do

   System.out.println("hello");

   while(false);
```

Contoh di atas akan memberikan output hello pada layar.

Panduanpemrograman:
Kesalahanpemrogramanyangbiasaterjadiketikamenggunakando -whileloopadalah lupauntukmenulistitikkoma(;)setelahekspresiwhile.
do{
...}while(boolean_expression)//- salah>tidakadatitikkoma(;)
Sepertipadawhileloop,pastikando-whileloopandaberhentipadasuatukondisi.

3 .forloop

Pernyataan*forloop*memilikikondisihampirmiripsepertistrukturpengulangan sebelumnyayaitumelakukanpengulanganuntukmengeksekusikodeyangsam sebanyakjumlahyangtelahditentukan.

Bentukdariforloop,

Sintaks for loop

```
for(InitializationExpression;  LoopCondition;StepExpression )


{

statement1;

statement2;

...

}
```

Dimana ***InitializationExpression***–inisialisasidarivariabelloop. ***LoopCondition***membandingkanvariabellooppadanilaibatas tertentu.***StepExpression***-melakukanupdatepadavariabelloop.

Berikutiniadalahcontohdariforloop,

Listing Program



```
inti;
for(i=0;i<10;i++)
{
System.out.print(i);
}
```

Padacontohini, pernyataani=0merupakaninisialisasidarivariabel. Selanjutnya, kondisii<10diperiksa. Jikakondisibernilaitrue, pernyataandidalam forloop dieksekusi. Kemudian, ekspresii++dieksekusi, laluakankembalipadabagian pemeriksaanterhadapkondisii<10lagi. Kondisiiniakandilakukanberulang-ulang sampaimencapainilaiyangsalah(false). Contohtadi,adalahcontohyangsamadariwhileloop,

Listing Program

```
inti=0;
while(i<10)
{
System.out.print(i);
i++;
}
```

c. Rangkuman

Java mengijinkan kita untuk mengeksekusi blok code berulang-ulang sesuai dengan jumlah yang diinginkan. Ada pernyataan while loop, yaitu pernyataan yang diulang-ulang sampai mencapai kondisi yang diinginkan jika nilai dari while loop bernilai benar atau true maka akan terus berulang-ulang hingga nilai menjadi false atau salah. Do-while-loop mengeksekusi pernyataan di dalam do-while-loop akan dieksekusi sedikitnya satu kali. For loop hampir mirip seperti struktur pengulangan untuk mengeksekusi kode yang sama sebanyak jumlah yang telah ditentukan.

Tugas

Tugas 1

Buatlah program untuk mencetak angka genap mulai dari 1 – 20.

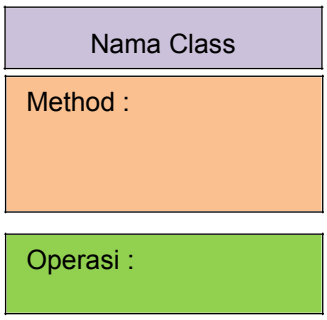
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Buatlah suatu program yang meminta masukan suatu integer dari 1 – 10 dan hasilnya mencetak nilai dengan bentuk segitiga siku-siku.

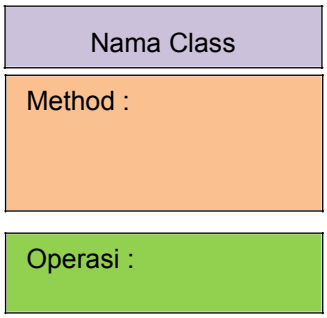
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan



Menentukan nama Method
Gambar Class Diagram



Buatlah listing program
Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. *Test Formatif.*

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Sebutkan 3 statement yang dimiliki oleh struktur kontrol

Jelaskan fungsi dari kata kunci *break* dan *continue* pada perulangan atau looping.

Jika suatu variabel dideklarasikan di dalam suatu kontrol *loop* for, dapatkah variabel tersebut digunakan di luar loop ?

Identifikasi dan betulkan kode berikut ini :

```
public class Test{
    public void main (string[] args) {
        for (int i=0; i < 10; i++);
        tot +=i;

        if (i < j);
            system.out.println(i)
        else
            system.out.println(j);
        while (j < 10);
        {
            j++;
        }:
        do {
            j++;
        }while (j < 10)
    }
}
```

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Sebutkan dan jelaskan secara singkat 3 statement yang dimiliki olehstruktur kontrol :

a) .....
.....
.....
.....

.....
.....
.....
.....
.....

LJ- 02 : Jelaskan fungsi dari kata kunci *break* dan *continue* pada perulangan atau looping.

.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 : Jika suatu variabel dideklarasikan di dalam suatu kontrol *loop* for, dapatkah variabel tersebut digunakan di luar loop ?

.....
.....
.....
.....
.....
.....

.....

.....

.....

.....

LJ- 04: Identifikasi dan betulkan program :



.....

.....

.....

.....

.....

.....

.....

.....

g. Lembar Kerja Siswa.



Kegiatan Belajar 7 :Dasar dan Aturan Pemrograman Berorientasi Obyek

a. *Tujuan Pembelajaran*

Setelah mengikuti kegiatan belajar 7 ini siswa diharapkan dapat :

Mendeklarasikan array single dan multi dimensi

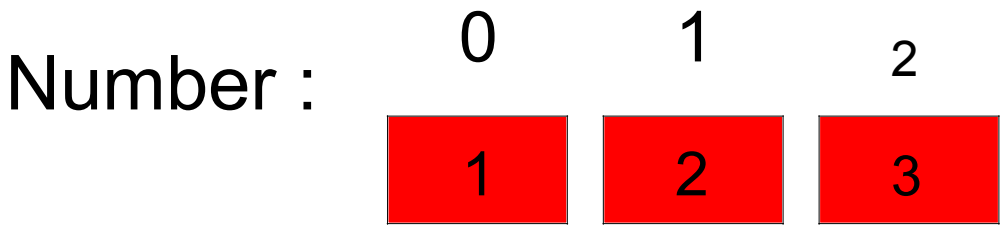
Menerapkan array single dan multi dimensi dalam class

b. *Uraian Materi*

Pada Bab sebelumnya, kita telah mendiskusikanbagaimanacarapendeklarasian berbagaimacamvariabeldenganmenggunakantipedataprimitif.Dalampendeklarasian variabel, kitaseringmenggunakansebuahtipedatabesertanamavariabelatau *identifier*yangunik. Apabilakitainginmenggunakanvariabeltersebut, kitaakan memanggildengannama*identifier*-nya. Sebagaicontoh,kitamemilikitigavariabeldengantipedataintyangmemiliki*identifier* berbedauntuktiapvariabel.

```
intnumber1;
intnumber2;
intnumber3;
number1=1;
number2=2;
number3=3;
```

SepertiyangdapatAndaperhatikanpadacontohdiatas,kodetersebutakansia-sia karenaharusmenginisialisasidanmenggunakansetiapvariabelpadahalsebenarnya variabel-variabeltersebutdigunakanuntuktujuanyangsama. Padabahasa pemrogramanJavamaupundibahasapemrogramanyanglain, terdapatsebuah kemampuanuntukmenggunakansatuvariabelyangdapatmenyimpanbeberapa data danmemanipulasinyadenganlebihefektif. Tipevariabelinilahyangdisebutsebagai **array**.



Gambar33.ContohdariIntegerArray

Sebuaharrayakanmenyimpanbeberapaitemdatayangmemilikitipedata samadidalamsebuahblokmemoriyangberdekatanyangkemudiandibaga imenjadi beberaparuang.Arrayadalahsebuahvariabel/sebuahlokasitertentuyang memiliki satunamasebagai*identifier*,namunidentifierinidapatmenyimpanlebihdar isebuah nilai.

PendeklarasianArray

Arrayharusdideklarasikansepertilayaknyasebuahvariabel. Padasaat mendeklarasikanarray,andaharusmembuatsebuahdaftardaritipedata, yangdiikuti olehsepasangtandakurung[],laludiikutiolehnama*identifier*-nya.Sebagaicontoh,

Sintaks

```
int[]ages;
```

atauAndadapatmenempatkansepasangtandakurung[]sesudahnama*identifier*.Sebagai contoh,

Sintaks

```
intages[];
```

Setelahpendeklarasianarray,kitaharusmembuatarraydanmenentukan berapapanj angnyadengansebuahkonstruktor.ProsesinidiJavadisebutsebagai *instantiation*(istilahdalamJavayangberartimembuat).Untukmeng-



instantiate sebuahobyek, kitamembutuhkansebuahkonstruktor.Kitaakanmembicarakan lagimengenaiinstantiateobyekdanpembuatan konstruktorpadabagianselanjutnya. Sebagaicatatanbahwaukurandariarraytidakdapatdiubahsetelahanda menginisialisasinya.Sebagaicontoh,

Sintaks Deklarasi Array

```
//deklarasi

intages[];

//instantiateobyek

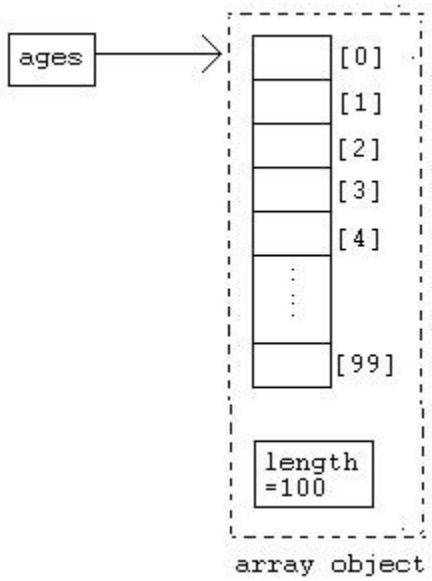
ages=newint[100];
```

ataubisajugaditulisdengan,

Sintaks Deklarasi Array

```
//deklarasidaninstantiate obyek

intages[]=newint[100];
```



Gambar35.InisialisasiArray

Padacontohdiatas, pendeklarasian

tersebutakanmemberitahukankepada compilerJava, bahwaidentifieragesakan digunakansebagainamaarrayyang berisidatabertipeinteger, dandilanjutkan denganmembuatataumeng-*instantiate* sebuaharraybaruyangterdiridari100 elemen. Selainmenggunakansebuahpernyataan *new*untukmeng-instantiatearray, Anda jugadapatmmendeklarasikan, membangun, kemudianmemberikansebuahnilaipada arraysekaligusdalamsebuahpernyataan. Sebagaicontoh,

Listing Program

```
//membuatsebuaharrayyangberisivariabel-variabel
//booleanpadasebuahidentifier.Arrayiniterdiridari4
//elemenyangdiinisilisasikansebagaivalue
//{true,false,true,false}
booleanresults[]={true,false,true,false};
//Membuatsebuaharrayyangterdiridaripenginisialisasian
//4variabeldoublebagivalue{100,90,80,75}
double[]grades={100,90,80,75};
//MembuatsebuaharrayStringdenganidentifierdays.Array
//initerdiridari7elemen.
Stringdays[]={"Mon","Tue","Wed","Thu","Fri","Sat","Sun"};
```

B. Pengaksesansebuah elemen array

Untukmengaksessebuahelemendalamarray,ataumengaksessebagiandari array,Andaharusmenggunakansebuahangkaatauyangdisebutsebagai **indeks**atausubscript.Padasaatmemasukkannilaikedalamarray, sebuahnomorindeksatausubscript anggotaarray,sehinggaprogramdanprogrammerpadaarrayapabiladibutuhkan. Nilaiindeksselalu dalamtipe integer, dimulai dari



angkanoldandilanjutkankeangkaberikutnya sampai akhirarray.Sebagaicatatanbahwaindeksdidalamarraydimulaidari0sampaidengan(ukuranArray-1).

Sebagaicontoh,padaarrayyangkitadeklarasikantadi,kitamempunyai,

Sintaks Elemen Array

```
//memberikannilai10kepadaelemenpertamaarray
ages[0]=10;
//mencetakelemenarrayyangterakhir
System.out.print(ages[99]);
```

Perludiperhatikanbahwasekaliarraydideklarasikandandikonstruksi, nilai yangdisimpandalamsetiapanggotaarrayakandiinisialisasisebagainol. Oleh karenaitu, apabilaAndamenggunakantipedatasepertiString, arraytidakakan diinisalisasimenjadistringkosong"".UntukituAndatetapharusmembuatStringarray secaraeksplisit. Berikutiniadalahcontohkodeuntukmencetakseluruhelemendidalamarray.Dalam contohinidigunakanlahpernyataan*forloop*,sehinggakodekitamenjadilebihpendek.

Listing Program

```
publicclassArraySample
{
publicstaticvoidmain(String[]args)
{
int[]ages=newint[100];
```

```
for(inti=0;i<100;i++)

  P
{

System.out.print(ages[i]);

}

}

}
```

Petunjukpenulisanprogram:

Biasanya, lebihbaikmenginisialisasiataumeng-instantiatearraysetelah Andamendeklarasikannya.Sebagaicontohpendeklarasianya

int[]arr=newint[100];

lebihdisarankandaripada,

```
int[]arr;
arr=newint[100];
```

Elemen-elemendalamn-elemenarraymemilikiindeksdari0sampain-1. Perhatikandisinibahwatidakadaelemenarrayarr[n].Haliniakan menyebabkan array-indexout-of-boundsexception. Andatidakdapatmengubahukurandarisebuaharray

D.ArrayMultidimensi

Arraymultidimensidiimplementasikansebagaiarrayyangterletakdi dalamarray.Arraymultidimensidideklarasikandenganmenambahkanjumlahtandakurungsetelah namaarray.Sebagaicontoh,

Sintaks Array Multidimensi

```
Elemen512x128dariintegerarray
int[][]twoD=newint[512][128];

//karakterarray8x16x24

char[][][]threeD=newchar[8][16][24];

//Stringarray4barisx2kolom

String[][]dogs=
```



```
{
      /
{"terry","brown"},/
U
{"Kristin","white"},
n

{"toby","gray"},t
{"fido","black"}u
k
};
m
```

engaksessebuahelemendidalamarraymultidimensi, samasaja denganmengaksesarraysatudimensi. Misalnyasaja, untukmengakseselemen pertamadaribarispertamadidalamarraydogs,kitaakanmenulis,

Sintaks Array Multidimensi
S

```
System.out.print(dogs[0][0]);
k
```

kodediatasakanmencetakString“terry”dilayar.

c. Rangkuman

Array adalah suatu tipe variabel yang memiliki kemampuan untuk menyimpan beberapa data dan memanipulasinya dengan lebih efektif. Array memiliki suatu nama sebagai identifier, namun identifier ini dapat menyimpan lebih dari sebuah nilai. Tipe variabel ini harus dideklarasi layaknya variabel lain, pada saat mendeklarasikan array anda harus membuat sebuah daftar dari tipe data, yang diikuti oleh sepasang tanda kurung [ ].Sekaliarraydideklarasikandandikonstruksi, nilai yangdisimpandalamsetiapanggotaarrayakandiinisialisasisebagainol. Untuk mengakses array anda harus menggunakan sebuah angka atau yang disebut sebagai indeks atau subscript. Nilai indeks ini selalu bertipe integer dan dimulai dari angka nol dilanjutkan ke angka berikutnya.Sedangkan array multidimensi diimplementasikan seagai array yang terletak di dalam array,

untuk mengakses sebuah elemen sama dengan mengakses array satu dimensi.

Tugas

Tugas 1

Tulislah suatu program yang membaca sepuluh bilangan integer dan kemudian menampilan sepuluh integer tersebut dengan urutan terbalik.

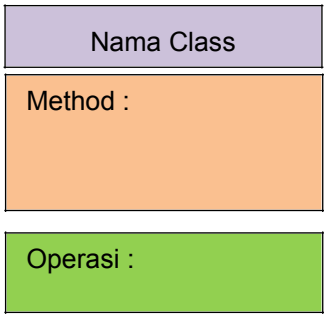
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Tulislah suatu program yang membaca sepuluh bilangan integer dan kemudian menampilan nilai terkecil dan terbesar dari sepuluh integer tersebut.

❖ Mengamati Listing Program dan Output Program

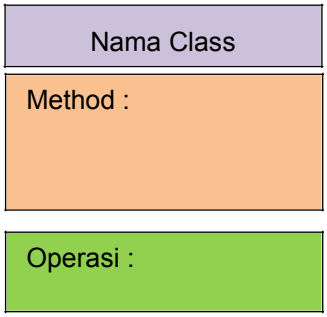
1. Menentukan nama Class



Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

Indikasikan benar atau salah statemen-statemen berikut ini dan berikan alasanya :

Setiap elemen di dalam suatu array bertipe sama.

Ukuran suatu array tetap atau permanen setelah array dideklarasikan

Ukuran suatu array tetap atau permanen setelah array diciptakan

Elemen – elemen di dalam suatu array harus bertipe primitif

Bagaimanakah sintaks mendeklarasikan dan menciptakan sebuah array ?

Dapatkah baris-baris di dalam suatu array dua dimensi memiliki panjang yang berbeda ?

Deklarasikan dan cipatakan suatu array 4 x 5 yang memuat elemen-elemen int.

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Indikasi :

a) .....

.....

.....

.....



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LJ- 02 : Bagaimanakah sintaks mendeklarasikan dan menciptakan sebuah array



.....

.....

.....

.....

.....

.....

.....

.....

.....

LJ- 03 : Dapatkah baris-baris di dalam suatu array dua dimensi memiliki panjang yang berbeda ?



.....

.....

.....

.....

LJ- 04: Deklarasikan dan cipatakan suatu array 4 x 5 yang memuat elemen-elemen int.



.....
.....
.....
.....
.....
.....

g. Lembar Kerja Siswa.



6. Kegiatan Belajar 7 :Konsep Class dan Obyek

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 7 ini siswa diharapkan dapat :

Memahami perbedaan class dan objek

Menyajikan pembuatan class ***b.***

Uraian Materi

1) Perbedaan Class dan Objek

Pada dunia perangkat lunak, sebuah obyek adalah sebuah komponen perangkatlunak yang stukturnya mirip dengan obyek pada dunia nyata. Setiap obyek dibangun dari sekumpulan data (atribut) yang disebut variable untuk menjabarkan karakteristik khusus dari obyek, dan juga terdiri dari sekumpulan method yang menjabarkan tingkah laku dari obyek.Bisa dikatakan bahwa obyek adalah sebuah perangkatlunak yang berisi sekumpulan variable dan method yg berhubungan.Variabel dan method dalam obyek Java secara formal diketahui sebagai variabel instance dan method instance.Hal ini dilakukanuntuk membedakan dari variable class dan method class, dimana akan dibahas kemudian.

Class adalah sturktur dasar dari OOP. Class terdiri dari dua tipe dari anggota dimana disebut dengan field (attribut/properti) dan method.Field merupakan tipe data yang didefinisikan oleh class,sementara method merupakan operasi. Sebuah obyek adalah sebuah instance (keturunan) dari class.

Untuk dapat membedakan antara class dan obyek, mari kita mendiskusikan beberapa contoh berikut ini. Kita memiliki sebuah class mobil dimana dapat digunakan untuk medefinisikan beberapa obyek mobil.Pada tabel dibawah, mobil A dan mobil B adalah obyek dari class mobil.Class memiliki field nomor,plat,warna,manufaktur dan kecepatan yang diisi dengan nilai pada obyek mobil A dan mobil B. Mobil juga dapat berakselerasi, berbelok dan melakukan rem.

Tabel 10. Class Car dan Obyek-Obyeknya

Class mobil		Obyek mobilA	ObyekMobilB
Variabel Intsance	NomorPlat	ABC111	XYZ123
	Warna	Biru	Merah
	Manufaktur	Mitsubishi	Toyota
	Kecepatan	50km/h	100km/h
Method *Instance*	MethodAkselerasi		
	MethodBelok		
	MethodRem		

Ketika diinisialisi, setiap obyek mendapat satu set variabel yang baru.Bagaimanapun, implementasi dari method dibagi diantara objekpada classyangsama.

Class menyediakan keuntungan dari *reusability*. Programmer perangkat lunak dapat menggunakan sebuah kelas beberapa kali untuk membuat banyak objek.

Instansiasi Class

Untuk membuat sebuah objek atau sebuah instance pada sebuah class. Kita menggunakan operator **new**. Sebagai contoh, jika anda ingin membuat instance dari classstring, kita menggunakan kode berikut:

```
String  str2=newString(“Hello world!”);
```

Ini juga sama dengan,

```
Stringstr2= "Hello";
```

Variabel ClassdanVariabelMethod

Selain dari variabel instance,kita juga memungkinkan untuk mendefinisikan variabel dari class, yang nantinya variabel ini dimiliki oleh class. Ini berarti variabel ini dapat memiliki nilai yang sama untuk semua objek pada class yang sama. Mereka juga disebut *staticmember variables*.



Pembuatan Class

Sebelum menulis *class* Anda,pertama pertimbangkan dimana Anda akan menggunakan *class* dan bagaimana *class* tersebut akan digunakan.Pertimbangkan pula nama yang tepat dan tuliskan seluruh informasi atau property yang ingin Anda isi pada *class*. Jangan sampai terlupa untuk menuliskan secara urut *method* yang akan Anda gunakan dalam *class*.

Dalam pendefinisian *class*, dituliskan:

Sintaks Pembuatan Class

```
<
<modifier>class<name>m
{
<attributeDeclaration>*
<constructorDeclaration>*
<methodDeclaration>*
d
}
```

m

ana :

<modifier> adalah sebuah *accessmodifier*, yang dapat dikombinasikan denganti pe*modifier* lain.

Pada bagian ini, kita akan membuat sebuah *class* yang berisi *record* dari siswa. Jika kita telah mengidentifikasikan tujuan dari pembuatan class,makadapat dilakukan pemberian nama yangsesuai.Nama yangtepat pada *class* ini adalah StudentRecord.

Untuk mendefinisikan *class*, kita tuliskan:

Sintaks Pembuatan Class

```
publicclasspStudentRecord
  u
{
//areapenulisankodeselanjutnya
}
```

i

mana,

Public	-	*Class*ini dapat diaksesdari luar*package*
Class	-	*Keyword*yangdigunakanuntukpembuatan *class*dalamJava
StudentRecord	-	*Identifier yang menjelaskan class*

3) DeklarasiAtribut

Dalampendeklarasianatribut,kitatuliskan:

Sintaks Deklarasi Atribut

```
modifier><type><name>[=<default_value>];
```

Langkah selanjutnya adalah mengurutkan atribut yang akan diisikan pada *class.*Untuk setiap informasi, urutkan juga tipe data yang yang tepat untuk digunakan.Contohnya, Anda tidak mungkin menginginkan untuk menggunakan tipe data *integer* untuk nama siswa, atau tipe data *string* pada nilai siswa.Berikut ini adalah contoh informasi yang akan diisikan pada *class* StudentRecord:

```
name        -String
address     -String
age         -Int
mathgrade   -double
englishgrade-
double
sciencegrade          -double averagegrade     -double
```

Instance Variable

Jika kita telah menuliskan seluruh atribut yang akan diisikan pada *class*, selanjut nya kita akan menuliskannya pada kode.Jika kita menginginkan bahwa atribut – atribut tersebut adalah unik untuk setiap *object* (dalam hal ini untuk setiap siswa), maka kita harus mendeklarasikannya sebagai *instance variable*

Sebagai contoh:

Sintaks Deklarasi Atribut

```
publicclassStudentRecord
{
```



```
privateStringname;
privateStringaddress;
privatedintage;
privateidoublemathGrade;
privatedoubleenglishGrade;
m
privatedoublescienceGrade;
privateadoubleaverage;
n
}
a
```

private disini menjelaskan bahwa variabel tersebut hanya dapat diakses oleh *class* itu sendiri.*Object*laintidakdapatmenggunakanvariabeltersebutsecaralangsung.Kitaakan membahastentangkemampuan aksespada pembahasanselanjutnya.

ClassVariableatau StaticVariables

Disamping*instancevariable*,kitajugadapatmendeklarasikan*classvariable* a tauvariabel yang dimiliki *class* sepenuhnya.Nilai pada variabel ini sama pada semua *object* di *class* yang sama.Anggaplahkita menginginkanjumlahdarisiswa yangdimilikidariseluruh*class*, kitadapatmendeklarasikansatu*staticvariable*yangakanmenampungnilaiter sebut.Kita beri namavariabel tersebutdengannama studentCount. Berikutpenulisan*staticvariable:*

Sintaks Class Variable

```
publicclassStudentRecord
{
//areadeklarasi instancevariables
privatestaticintstudentCount;
//areapenulisankodeselanjutnyaK
} i t
  a
```

gunakan *keyword : 'static'*untuk mendeklarasikan bahwa variabel tersebut adalah *static*.



Setelah mengikuti kegiatan belajar 1 ini siswa diharapkan dapat :

1) Memahami paradigma pemrograman berorientasi obyek

Maka keseluruhankodeyangdibuatterlihatsebagaiberikut:

b. Uraian Materi

Paradigma Pemrograman Berorientasi Obyek Sintaks Cla

Variable

Ide dasar pada bahasa berorientasi obyek (POB) adalah mengkombinasikan data dan fungsi untuk mengakses data menjadi sebuah kesatuan unit. Unit ini di kenal dengan nama **obyek**.Obyek sebenarnya mencerminkan pola kerja sebagai departemen di dalam sebuah perusahaan bisnis.Contoh departemen;

- ✓ Penjualan
- ✓ Akuntan

```
publicclassStudentRecord
{
    private    String    address;
    privat      nt       age;
    private    double    mathGrade;
    private    double    englisGrade;
    private    double    scienceGrade;
    private    double    average;
    private    static    intstudentCount;
//areapenulisankodeselanjutnya
}
```

c. *Rangkuman*

Objek adalah sebuah komponen perangkat lunak yang strukturnya mirip dengan objek pada dunia nyata. Dalam bahasa pemrograman bisa dikatan bahwa objek adalah pernagkat lunak yang berisi sekumpulan variabel dan method yang berhubungan. Class adalah struktur dasar dari OOP, class terdiri dari dua tipe dari anggota dimana disebut dengan field dan method. Field merupakan tipe data yang didefiniskan, sementara method merupakan operasi. Untuk membuat class, sebelum menulis nama pertimbangkan dulu nama class dan dimana class tersebut digunakan. Dalam pendeklarasian atribut untuk menggunakan tipe data *integer* unutk nama siswa, atau tipe data *string* pada nilai siswa. Jika anda menginginkan bahwa atribut-atribut tersebut unik, maka dideklarasikan sebagai *instance variable. Class variable* atau *static variable,* variabel ini sama pada semua object di class yang sama. Anda dapat mendeklarasikan satu *static variable* yang akan menampung nilai tersebut.

Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang



berkorespondensi dengan objek-objek dunia nyata.

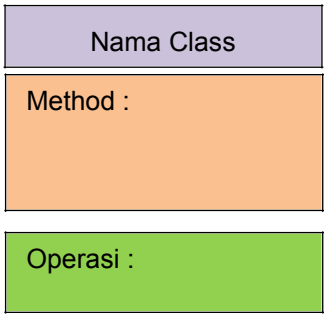
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan

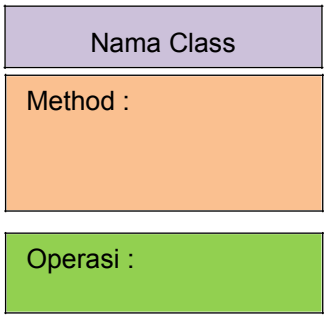
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	



10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. *Test Formatif.*

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Apa yang dimaksud dengan Class ?
Apa yang dimaksud dengan Objek ?
Apa perbedaan instance variabel dan static variabel ?
Bagaimanakah sintaks pada pembuatan class ?

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apa yang dimaksud dengan Class ?



.....
.....
.....
.....
.....
.....
.....

LJ- 02 : Apa yang dimaksud dengan Objek ?



.....
.....
.....
.....

LJ- 03 : Apa perbedaan instance variabel dan static variabel ?



.....

.....

.....

.....

.....

.....

.....

.....

LJ- 04: Bagaimanakah sintaks pada pembuatan class ?



.....

.....

.....

.....

.....

.....

.....

.....



g. Lembar Kerja Siswa.

7. Kegiatan Belajar 8 :Konsep Class dan Obyek

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 8 ini siswa diharapkan dapat :

Memahami method dalam class

Menyajikan penggunaan method dalam class ***b.***

Uraian Materi

Apakah Method itu dan mengapa menggunakan Method?

Sebuah method adalah bagian-bagian kode yang dapat dipanggil oleh program utama atau dari method lainnya untuk menjalankan fungsi yang spesifik.

Berikut adalah karakteristik dari method:

- Dapat mengembalikan satu nilai atau tidak sama sekali
- Dapat diterima beberapa parameter yang dibutuhkan atau tidak ada parameter sama sekali. Parameter bisa juga disebut sebagai argumen dari fungsi
- Setelah method telah selesai dieksekusi, dia akan kembali pada method yang memanggilnya.

Sekarang mengapa kita butuh untuk membuat banyak method? Mengapa kita tidak menuliskan semua kode pada sebuah method? Hal ini karena penyelesaian masalah yang sangat efektif adalah memecah masalah-masalah tersebut menjadi beberapa bagian. Kita juga dapat melakukan hal ini di Java dengan membuat method untuk mengatasi Bagian tertentu dari masalah. Sebuah permasalahan dapat dipecah-pecah menjadi beberapa bagian kecil. Hal ini sangat baik sekali untuk membuat program yang sangat besar.

Memanggil Instance dan memberikan Variabel dari Method Sekarang, untuk mengilustrasikan bagaimana memanggil method, mari kita menggunakan class string sebagai contoh. Anda dapat menggunakan the dokumentasi dari Java API untuk melihat semua method yang tersedia dalam class string. Selanjutnya, kita akan membuat method, kita sendiri. Tapi untuk saat ini, mari terlebih dahulu kita gunakan method yang sudah disediakan oleh Java. Untuk memanggil sebuah instance method, kita dapat menuliskan:



Sintaks Class Variable

```
nameOfObject.nameOfMethod(parameters);
```

mari kita mengambil dua contoh method yang ditemukan dalam classString.

Tabel 11. Deklarasi Methode

Deklarasimethod	Definisi
public charcharAt(intindex)	Mengambil karakter pada indeks tertentu.
public boolean equalsIgnoreCase (String another String)	Membandingkan antar String, tidak case sensitive.

PemberianVariabel DalamMethod

Pada contoh kita sebelumnya, kita sudah pernah mecoba melewati variable pada method. Walaupun kita belum dapat membedakan antara perbedaan tipe variabel yang diberikan (*passing*)ke method dalam Java. Ada dua tipe data variabel *passing* pada method, yang pertama adalah pass-by-value dan yang kedua adalah pass-by-reference.

Pass-by-Value

Ketika pass-by-values terjadi,method membuat sebuah salinan dari nilai variable yang dikirimkan ke method.Walaupun demikian,method

Tidak dapat secara langsung memodifikasinilaivariabelpengirimnyameskipunparametersalinanny asudahdimodifikasi nilainyadi dalammethod.

Pass-by-reference

Ketika sebuah pass-by-reference terjadi,alamat memori dari nilai pada sebuah variabel dilewatkanpadasaatpemanggilanmethod.Haliniberartibahwamethod menyalinalamat memori darivariabelyangdilewatkanpadamethod.Initidaksepertipadapass-by-value, methoddapatmemodifikasivariabelasli dengan menggunakanalamatmemoritersebut, meskipunberbedanamavariabelyangdigunakandalammethoddengan variabelaslinya, keduavariabel ini menunjukkanlokasi dari

datayangsama

Memanggil Method Static

MethodStaticadalahmethodyangdapatdipakaitanpaharusmenginisialisasisuatuclass (maksudnya tanpa menggunakan variabel terlebih dahulu).Method static hanya dimiliki olehclassdantidakdapatdigunakanolehinstance(atauobjek)darisuatuclass.Method static dibedakan dari method yang dapat instance di dalam suatu class oleh kata kunci static.

Untukmemanggil method static,ketik:

Sintaks pemanggilan metho

```
Classname.staticMethodName(params);
```

Pembuatan*Methods*

Sebelumkitamembahas*method*apayangakandipakaipada*class*,marikitaperhatikan penulisan*method*secaraumum.

Dalampendeklarasian*method*,kitatuliskan:

Sintaks pembuatan metho

```
<modifier><returnType><name>(<parameter>*){
<statement>*
}
```

dimana,

<modifier>dapatmenggunakanbeberapa*modifier*yangberbeda

<returnType>dapat berupa seluruhtipedata,termasuk*void*

<name>*identifier*atas*class*

<parameter> ::= <tipe_parameter><nama_parameter>[,]

Accessor Methods

Untuk mengimplementasikan enkapsulasi,kita tidak menginginkan sembarang *object* dapatmengaksesdatakapansaja.Untukitu,kitadeklarasikan



atributdari*class*sebagai *private*.Namun,adakalanyadimanakitamenginginkan*object*lainuntukdapatmengakses data *private*.Dalamhalinikitagunakan*accessormethods.*

Accessor Methods digunakan untuk membaca nilai variabel pada *class*,baik berupa *instance* maupun *static*.Sebuah *accessormethod* umumnya dimulai dengan penulisan ***get<namaInstanceVariable>***.*Method*inijuga mempunyaisebuah*returnvalue*.

Sebagai contoh,kita ingin menggunakan *accessormethod* untuk dapat membaca nama, alamat,nilaibahasaInggris,Matematika,danilmu pastidari siswa.

Marikita perhatikansalahsatucontohimplementasi*accessormethod*.

Listing Program

```
publicclassStudentRecord
{
privateStringname;
publicStringgetName(){
returnname;
}
}
```

dimana,

public -Menjelaskan bahwa*method*tersebut dapat diaksesdari *object*luarclass

String -Tipedata*returnvalue*dari *method*tersebut bertipeString

getName -Nama dari *method*

() -Menjelaskan bahwa*method*tidakmemiliki parameter Apapun

Mutator Methods

Bagaimana jika kita menghendaki *object* lain untuk mengubah data? Yang dapat kita lakukan adalah membuat *method* yang dapat

memberi atau mengubah nilai variable dalam *class*,baik itu berupa *instance* maupun *static*. *Method* semacam ini disebut dengan *mutator methods*. Sebuah *mutator method* umumnya tertulis**set<namaInstanceVariabel>**.

Marikita perhatikan salah satu dari implementasi *mutatormethod*:

Listing Program

```
publicclassStudentRecord
{
privateStringname;
:
:
publicvoidsetName(Stringtemp){
name=temp;
}
}
```

dimana,

public - Menjelaskan bahwa*method*ini dapat dipanggil*object* luarclass

void - *Method*ini tidakmenghasilkan*returnvalue*

setName - Nama dari *method*

(Stringtemp) - Parameteryangakan digunakanpada*method*

Pernyataan berikut:

name=temp;

mengidentifikasi nilai dari temp sama dengan name dan mengubah data pada *instance variable*name.Perlu diingat bahwa *mutatormethods* tidak menghasilkan *return value*.Namun berisi beberapa argumen dari programyangakandigunakanoleh*method*.

MultipleReturn Statements

Andadapatmempunyaibanyak*returnvalues*padasebuah



*method*selamamerekatidak pada blok program yang sama.Anda juga dapat menggunakan konstanta disamping variabel sebagai *returnvalue*.

Sebagai contoh,perhatikan *method* berikut ini:

Listing Program

```
publicStringgetNumberInWords(intnum){
StringdefaultNum="zero";

if(num==1){
return"one";//mengembalikansebuahkonstanta
}
elseif(num==2){
return"two";//mengembalikansebuahkonstanta
}
//mengembalikansebuahvariabel
returndefaultNum;
}
```

Static Methods

Kitamenggunakan*staticmethod*untukmengakses*staticvariable*studentCount.

Listing Program

```
publicclassStudentRecord
{
privatestaticintstudentCount;

publicstaticintgetStudentCount(){
returnstudentCount;
}
}
```

dimana,

public -Menjelaskan bahwa*method*ini dapat diaksesdari *object*luarclass

static -*Method*ini adalah*static*dan pemanggilannya menggunakan **[namaclass].[namaMethod]**.

Sebagaicontoh:

studentRecord.getStudentCount

Int -Tipe*return*dari *method.*Mengindikasikan*method*tersebutharus mempunyai *returnvalue*berupainteger

public -Menjelaskan bahwa*method*ini dapat diaksesdari*object*luarclass

getStudentCount -Nama dari *method*

() -*Method*ini tidakmemiliki parameterapapun

Padadeklarasidiatas,*method*getStudentCount()akanselalumenghasilkan*returnvalue*0 jikakitatidakmengubahapapunpadakodeprogramuntukmengaturnilainya.Kitaakan membahaspengubahannilaidari studentCount pada pembahasan*constructor*.

PetunjukPenulisan Program:
Namamethodharusdimulai denganhurufkecil
Namamethodharusberupa katakerja
Gunakandokumentasisebelummendeklarasikansebuahmethod.Anda dapatMenggunakanJavaDoc.

Rangkuman

Method merupakan bagian-bagian kode yang dapat dipanggil oleh program utama atau dari method lainya. Anda dapat menggunakan the dokumentasi dari Java API untuk melihat semua method yang tersedia dalam class string. Pemberian variabel dalam method terdapat dua tipe data variabel *passing* pada method, yang pertama adalah pass-by-value dan yang kedua adalah pass-by-reference. Pass by value membuat sebuah salinan dari nilai variabel



yang dikirimkan ke method, namun tidak dapat secara langsung memodifikasi nilai variabel pengirimnya. Pass by reference method menyalin alamat memori dari variabl yang dilewatkan pada method, method dapat memodifikasi variabel asli dengan menggunakan alamat memori tersebut. Dalam pembuatan method terdapat accesor method yang berfungsi mengimplementasikan enkapsulasi sehingga kita tidak menginginkan sembarang *object* dapat mengakses data kapan saja. Sedangkan method mutator adalah method yang dapat memberi atau mengubah nilai variabel dalam class, baik itu berupa *instance* maupun *static.* Untuk mengakses *static variable* kita menggunakan *static method.*

Tugas

Tugas 1

Tulislah suatu metode untuk menghitung penjumlahan digit dalam suatu integer.

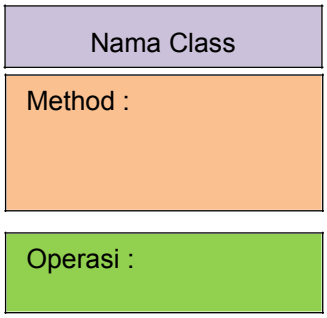
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Tulislah suatu metode untuk menampilkan matrik **n x n.** Setiap elemen 0 atau 1, dibangkitkan secara acak. Tulislah suatu program untuk menguji dan menampilkan suatu matriks 3 x 3.

❖ Mengamati Listing Program dan Output Program

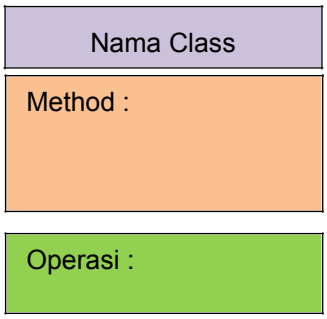
Mengamati bentuk matrik.

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program



No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

Apakah yang membedakan antara accessor method dan mutator method ?

Apa perbedaan konstruktor dan metode ?

Sebutkan dan jelaskan macam pemberian variabel dalam method

Apa yang dimaksud dengan method ? dan berikan contohnya

f. Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apakah yang membedakan antara accessor method dan mutator method ?

.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 02 : Apa perbedaan konstruktor dan metode ?

.....
.....
.....
.....
.....
.....
.....
.....
.....



LJ- 03 : Sebutkan dan jelaskan macam pemberian variabel dalam method



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LJ- 04: Apa yang dimaksud dengan method ? dan berikan contohnya



.....

.....

.....

.....

.....

g. Lembar Kerja Siswa.

8. Kegiatan Belajar 9 :Konsep Class dan Obyek



a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 9 ini siswa diharapkan dapat :

Memahami penggunaan referensi this

Menyajikan penggunaan constructor dalam class

b. Uraian Materi

Reference this

Reference *this* digunakan untuk mengakses instance variable yang dibiaskan oleh parameter.Untuk pemahaman lebih lanjut,mari kita perhatikan contoh pada *method* setAge.Dimisalkankitamempunyai kodedeklarasiberikutpada*method*setAge.

Sintaks reference thi

```
publicvoidsetAge(intage){

age=age;//SALAH!!!

}
```

Nama parameterpada deklarasi ini adalah age,yang memiliki penamaan yang sama dengan *instance variable* age.Parameterage adalah deklarasi terdekat dari *method*, sehingganilaidari parameter tersebutakandigunakan.Maka pada pernyataan

```
age=age;
```

kita telahmenentukannilaidariparameteragekepadaparameter itusendiri.Halinisangattidakkitakehendakipadakodeprogramkita.Untukmengh indarikesalahansemacamini, kitagunakanmetodereferensi***this***.Untukmenggunakantipereferensi ini,kitatuliskan:

```
this.<namaInstanceVariable>
```

Sebagai contoh,kita dapat menulis ulang kode hingga tampak sebagai berikut:

Sintaks reference this

```
publicvoidsetAge(intage){
this.age=age;
}
```

Method ini akan mereferensikan nilai dari parameterage kepada *instance variable* dari*object*StudentRecord.

CATATAN : Anda hanya dapatmenggunakan referensithis terhadap *instance variable* dan BUKAN*static* ataupunclass variabel

Variabel kelas (class variable) adalah variabel yang dideklarasikan di dalam sebuah kelas dan bertidak sebagai data field dari kelas tersebut, sedangan variabel local (local variable) adalah variabel yang dideklarasikan di dalam sebuah metoda. Cakupan variabel lokal dimulai dari posisi variabel tersebut dideklarasikan sampai dengan akhir dari blok metoda yang ditandai dengan closing brace. Cakupan dari variabel kelas meliputi keseluruhan kelas. Pemberian nama yang sama antara variabel kelas dan variabel lokal di sebuah metoda mungkin saja bisa terjadi. Misalnya, metoda set yang digunakan untuk merubah nilai variabel kelas, mungkin saja mendeklarasikan parameter dengan nama sama sebagaimana nama variabel kelas yang nilainya akan dirubah.

Di dalam metoda set, untuk dapat mengacu ke variabel kelas yang nilainya akan dirubah, Anda perlu menggunakan kata kunci this. Apabila variabel kelas tersebut dideklarasikan menggunakan modifier static, maka variabel kelas dapat diakses menggunakan nama-kelas.variabel-static. Di dalam contoh program kelas Warna, terdapat dua variabel kelas yaitu variabel merah dan biru. Kelas Warna juga mendeklarasikan dua buah metoda yaitu metoda setMerah yang memiliki parameter merah dan setBiru yang memiliki parameter biru. Parameter dari kedua metoda tersebut memiliki kesamaan nama dengan dua variabel kelas (data field) dari kelas Warna. Berikut ini adalah kode program kelas Warna yang mungkin dapat memberikan pemahaman lebih jelas tentang penggunaan kata kunci (keyword) this untuk mengacu ke variabel kelas.

Listing Program



```
Deklarasi kelas
class Warna {


Deklarasi variabel kelas (data field)
int merah = 7;
static double biru = 2;


Deklarasi metoda
void setMerah(int merah) {
this.merah = merah;
          }
// Deklarasi metoda static
static void setBiru( double biru)
                { Warna.biru = biru;
          }
}
```

Misalnya c1 adalah variabel acuan yang mengacu ke objek dari kelas Warna. Memanggil metoda dengan pernyataan c1.setMerah(5) sama dengan mengeksekusi c1.merah = 5, dimana kata kunci this diganti dengan c1. Baris nomor 10 adalah pernyataan pemberian yaitu memberikan nilai parameter merah ke data field merah dari objek pemanggil (misalnya c1). Sedangkan di baris nomor 15, pernyataan tersebut mempunyai arti bahwa nilai di parameter biru diberikan ke data field statik biru dari kelas Warna.

2) Construtor

Telah tersirat pada pembahasan sebelumnya, Constructor sangatlah penting pada pembentukan sebuah object. Constructor adalah method dimana

seluruh inisialisasi object ditempatkan.

Berikutiniadalah*property*dariConstructor:

- Constructormemiliki namayang sama dengan*class*
- SebuahConstructormiripdengan*method*pada umumnya,namunhanya informasi– informasi berikut yangdapat ditempatkan pada *header*sebuah constructor,*scope* atauidentifikasi pengaksesan(misal:public),nama darikonstuktordanparameter.
- Constructortidakmemiliki*returnvalue*
- Constructortidak dapat dipanggilsecaralangsung,namun harus dipanggil dengan menggunakanoperator***new***padapembentukansebuah*class*.

Untukmendeklarasikanconstructor,kitatulis,

Sintaks mendeklarasikan consturct

```
<modifier><className>(<parameter>*){

<statement>*

}
```

Default Constructor

Setiapclass memiliki default constructor.Sebuah default constructoradalah constructor yangtidakmemilikiparameterapapun.Jika sebuah*class*tidakmemilikiconstructorapapun, maka sebuahdefaultconstructorakan dibentuksecaraimplisit:

Sebagaicontoh,pada*class*StudentRecord,bentukdefaultconstructorakante rlihatseperti dibawahini :

```
publicStudentRecord()
{
        //areapenulisankode
}
```

MenggunakanConstructor

Untukmenggunakanconstructor,kita gunakankode–kodesebagai berikut:

Listing Program



```
publicstaticvoidmain(String[]args)
        {
//membuat3objek
            StudentRecordannaRecord=newStudentRecord("A
            nna");
            StudentRecordbeahRecord=newStudentRe
            cord("Beah","Philippines");
            StudentRecordcrisRecord=newStudentRe
            cord(80,90,100);

            //areapenulisankodeselanjtunya
        }
```

Sebelumkitalanjutkan,marikitaperhatikankembalideklarasivariabelstaticstudentCount yang telah dibuat sebelumnya.Tujuan deklarasi studentCount adalah untuk menghitung jumlah*object*yangdibentukpada*class*StudentRecord.Jadi,apayangakankitalakukan selanjutnyaadalahmenambahkan nilai dari studentCount setiapkali setiappembentukan *object*pada*class*StudentRecord.Lokasiyangtepatuntukmemodifikasidanmenambahkan nilaistudentCountterletakpadaconstructor-nya,karenaselaludipanggilsetiapkaliobjekterbentuk. Sebagai contoh:

Listing Program

```
publicStudentRecord()
{
            //letakkodeinisialisasi
            studentCount++;//menambahstudent
        }

publicStudentRecord(Stringtemp)
{
            this.name=temp;
```

```
                    studentCount++;//menambahstudent
                }

publicStudentRecord(Stringname,Stringaddress)
{
                    this.name=name;
                    this.address=address;
                    studentCount++;//menambahstudent
                }

publicStudentRecord(doublemGrade,doubleeGrade,doublesGrad
e)
{
                    mathGrade=mGrade;
                    englishGrade=eGrade;
                    scienceGrade=sGrade;
                    studentCount++;//menambahstudent
                }
```

PemanggilanConstructor Denganthis()

Pemanggilan constructordapat dilakukan secara berangkai,dalam arti Anda dapat memanggil constructor di dalam constructorlain.Pemanggilan dapat dilakukan dengan referensi*this()*.Perhatikancontohkodesebagaiberikut:

Listing Program

```
public StudentRecord(){
```



```
this("some string");


}
public StudentRecord(String temp){
this.name = temp;
}


public static void main( String[] args )
{
StudentRecord annaRecord = new StudentRecord();
}
```

Daricontohkodediatas,padasaatbariske13dipanggilakanmemanggilconstructor dasarpada baris pertama.Pada saat baris kedua dijalankan,baris tersebut akan menjalankan constructoryangmemilikiparameterStringpada bariske-6. Beberapahalyangpatut diperhatikan pada penggunaan***this()*** :

Harusdituliskan pada barispertama pada sebuahconstructor

Hanyadapatdigunakanpadasatudefinisiconstructor.Kemudianmetodeinid apat diikuti dengankode–kodeberikutnyayangrelevan

c. Rangkuman

Reference *this* digunakan untuk mengakses instance variable yang dibiaskan oleh parameter. Variabel kelas (class variable) adalah variabel yang dideklarasikan di dalam sebuah kelas dan bertidak sebagai data field dari kelas tersebut, sedangan variabel local (local variable) adalah variabel yang dideklarasikan di dalam sebuah metoda. Constructor sangatlah penting pada pembentukan sebuah object. Constructor adalah method dimana seluruh inisialisasi object ditempatkan.Default constructoradalah constructor yangtidakmemilikiparameterapapun. Pemanggilan constructordapat dilakukan secara berangkai,dalam arti Anda dapat memanggil constructor di dalam constructorlain dengan menggunakan referensi *this()*.

Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

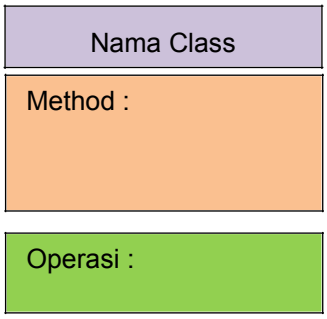
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2



Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan

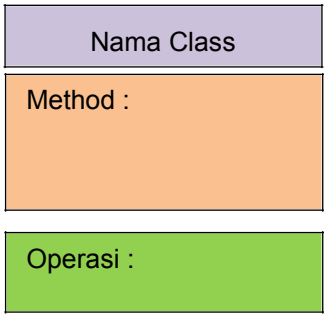
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	

8.

9.

10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

Apakah fungsi dari kata kunci this ?

Sebutkan property dari Constructor !

Apa saja yang perlu diperhatikan dalam penggunaan kata kunci this ?

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apakah fungsi dari kata kunci this ?

.....

.....

.....

.....

LJ- 02 : Sebutkan property dari Constructor !

.....

.....

.....

.....



.....

.....

.....

.....

.....

LJ- 03 : Apa saja yang perlu diperhatikan dalam penggunaan kata kunci this ?



.....

.....

.....

.....

LJ- 04:



.....

.....

.....

.....

.....

.....

.....

.....

.....

g. Lembar Kerja Siswa.



9. Kegiatan Belajar 10 :Pembungkusan Data

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 10 dan 11 ini siswa diharapkan dapat :

Memahami konsep enkapsulasi

Menerapkan konsep enkapsulasi dalam class ***b.***

Uraian Materi

1) Enkapsulasi dan modifier

Enkapsulasimerupakan teknik yang membuat variabel/field class menjadi bersifat privat dan menyediakan akses ke variabel/field melalui public method. Jika field di deklarasikan sebagai private, maka field ini tidak bisa diakses oleh siapapun diluar class, dengan demikian field disembunyikan di dalam class.

Manfaat utama teknik encapsulation adalah kita mampu memodifikasi kode tanpa merusak kode yang telah digunakan pada class lain.Enkapsulasi memiliki manfaat sebagai berikut:

Modularitas

Source code dari sebuah class dapat dikelola secara independen dari source code class yang lain. Perubahan internal pada sebuah class tidak akan berpengaruh bagi class yang menggunakannya.

Information Hidding

Penyembunyian informasi yang tidak perlu diketahui objek lain.

Pada saat membuat, mengatur properties dan class methods, kita ingin untuk mengimplementasikan beberapa macam larangan untuk mengakses data. Sebagai contoh, jika Anda ingin beberapa atribut hanya dapat diubah hanya dengan method tertentu, tentu Anda ingin menyembunyikannya dari object lain pada class. Di JAVA, implementasi tersebut disebut dengan

access modifiers. Terdapat 4 macam access modifiers di JAVA, yaitu : public,

private, protected

dan default. 3 tipe akses pertama tertulis secara ekplisit pada kode untukmengindikasi

kan tipe akses, sedangkan yang keempat yang merupakan tipe default, tidak diperlukan penulisan keyword atas tipe.

No	Modifier	Class	Paket Sama		Paket Berbeda	
			Extend	Instan	Extend	Instan
1	Public	√	√	√	√	√
2	Protected	√	√	√	√	
3	Default	√	√	√		
4	Private	√				

Public

Dapat dilihat pada table diatas bahwa keyword Public dapat diakses didalam class itu sendiri, dapat diakses dengan menggunakan metode extend dan instan pada paket yang sama, serta dapat diakses dengan metode extend maupun instan dalam paket yang berbeda. Artinya hak akses public dapat diakses oleh sembarang object manapun dan dimanapun posisinya serta dengan apapun caranya.

Data maupun method yan bersifat public dapat diakses oleh semua bagian didalam program. Dengan kata lain, data – data maupun method-method yang dideklarasikan dangan tingkat askses public akan dikenali atau dapat diakses oleh semuakelas yang ada didalam, baik yang merupakan kelas turunan maupun kelas yang tidak memiliki hubungan sama sekali. Untuk mendeklarasikan suatu data atau method dengan tingkat akses public, gunakan kata kunci public.

Berikut contoh program sederhana :

Listing Program

```
class atas
{
public int a;
protected int b;
private int c;
}
class bawah{
public static void main(String[]args){
atas objek = new atas(); objek.a=2;

objek.b=3;
System.out.println("nilai a: "+objek.a);
System.out.println("nilai b: "+objek.b);
}
```



```
}
```

program diatas akan menghasilkan tampilan berikut:

nilai a: 2

nilai b: 3

program diatas terdiri dari dua kelas yaitu kelas sekunder yang berisi variabel a, b dan c dengan tingkat akses yang berbeda, dan kelas primer yang berisi objek untuk melakukan instance pada kelas turunan, objek pada kelas primer hanya dapat mengisi nilai pada variabel a dan b karena kedua variabel tersebut memiliki tingkat akses public dan protected, karena variabel c memiliki tingkat akses private maka objek pada kelas primer tidak bisa mengisi variabel tersebut.

Protected

Suatu data maupun method yang dideklarasikan dengan tingkat akses protected dapat diakses oleh kelas yang memilkinya dan juga ileh kelas – kelas yang masih memiliki oleh hubungan turunan. Sebagai contoh, apabila data x dalam kelas A dideklarasikan sebagai protected, maka kelas B (yang merupakan turunan dari kelas A) diizinkan untuk mengakses data x. Namun apabila terdapat kelas lain , misalnya C (yang bukan merupakan turunan dari kelas A maupun B), tetap tidak dapat mengakses data – data yang dideklarasikan dengan tingkat akses protected. Untuk mendeklarasikan suatu data atau method dengan tingkat akses protected, gunakan kata kunci ptotected.

Listing Program

```
public class motor
{
    protected String jenismotor;
    protected String address;
    public motor()

        program turunan:
        program honda.java
        public class honda extends motor
        {
            protected String jenishonda;
            protected String kecepatanhonda;
```

```
public honda()
{
```

dari contoh program protected yang dapat mengakses hanya klas motor dan klas turunanya,yaitu Honda

Private

Dengan mendeklarasikan data dan method menggunakan tingkat akses private, maka data dan method tersebut hanya dapat diakses oleh kelas yang memiliinya saja. Ini berarti data dan method tersebut tidak boleh diakses atau digunakan oleh kelas kelas lain yang terdapat didalam program. Untuk mendeklarasikan suatau data atau method dengan tingkat akses private, gunakan kata kunci private.

Listing Program

```
public class Siswa
{
//akses dasar terhadap variabel
private String nama;
//akses dasar terhadap metode
private String getNama(){
return name;
}
}
```

Pada contoh diatas, variabel nama dan method getNama() hanya dapat diakses oleh method internal class tersebut.

Default

Untuk hak akses default ini, sebenarnya hanya ditujukan untuk class yang ada dalam satu paket, atau istilahnya hak akses yang berlaku untuk satu folder saja(tidak berlaku untuk class yang tidak satu folder/package)

Listing Program

```
public class Siswa{
//akses dasar terhadap variabel
String nama;

//akses dasar terhadap method
String getName(){
return nama;
}
}
```



Pada contoh diatas, variabel nama dan method getNama() hanya dapat diakses oleh method internal class tersebut.

2) Penerapan enkapsulasi dalam class

Kita dapat menyembunyikan information dari suatu class sehingga anggota-anggota class tersebut tidak dapat diakses dari luar. Adapun caranya adalah cukup dengan memberikan akses control private ketika mendeklarasikan suatu atribut atau method. Contoh:

```
private int nrp;
```

Encapsulation (Enkapsulasi) adalah suatu cara untuk menyembunyikan implementasi detail dari suatu class. Enkapsulasi mempunyai dua hal mendasar, yaitu:

information hiding

menyediakan suatu perantara (method) untuk pengaksesan data

Contoh:

Listing Program

```
public class Siswa {

private int nrp;



public void setNrp(int n) {

nrp=n;

}

}
```

Contructor (konstruktor) adalah suatu method yang pertama kali dijalankan pada saat pembuatan suatu obyek. Konstruktor mempunyai ciri yaitu: mempunyai nama yang sama dengan nama class tidak mempunyai return type (seperti void, int, double, dan lain-lain) Contoh:

Listing Program

```
public class Siswa {

private int nrp;

private String nama;

public Siswa(int n, String m) {

nrp=n;

nama=m;

}

}
```

Suatu class dapat mempunyai lebih dari 1 konstruktor dengan syarat daftar parameternya tidak boleh ada yang sama. Contoh:

Listing Program

```
public class Siswa {

private int nrp;

private String nama;

public Siswa(String m) {

nrp=0;

nama="";

}

public Siswa(int n, String m) {

nrp=n;

nama=m;

}

}
```



c. Rangkuman

Enkapsulasimerupakan teknik yang membuat variabel/field class menjadi bersifat privat dan menyediakan akses ke variabel/field melalui public methodManfaat utama teknik encapsulation adalah kita mampu memodifikasi kode tanpa merusak kode yang telah digunakan pada class lain. Di JAVA, implementasi tersebut disebut dengan access modifiers. Terdapat 4 macam access modifiers di JAVA, yaitu : public, private, protected dan default.Public dapat diakses didalam class itu sendiri, dapat diakses dengan menggunakan metode extend dan instan pada paket yang sama, serta dapat diakses dengan metode extend maupun instan dalam paket yang berbeda.Protected dapat diakses oleh kelas yang memilkinya dan juga oleh kelas – kelas yang masih memiliki oleh hubungan turunan. Sedangkan privat maka data dan method tersebut hanya dapat diakses oleh kelas yang memiliinya saja. Default sebenarnya hanya ditujukan untuk class yang ada dalam satu paket, atau istilahnya hak akses yang berlaku untuk satu folder saja(tidak berlaku untuk class yang tidak satu folder/package)

Tugas

Tugas 1

Buatlah program untuk menghitung gaji bersih dari seorang pegawai, pajak ppn sebesar 10% dari gaji kotor.

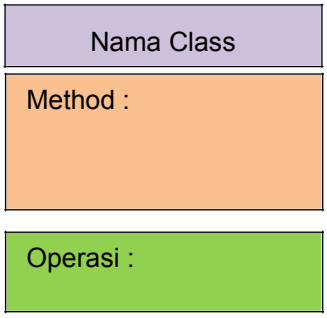
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



5. Buatlah listing program

6. Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan

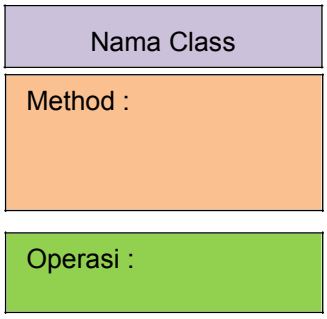
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program



No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Apakah keuntungan enkapsulasi bidang data ?

Apa yang anda pahami terkait keyword private, protected, public?

Apa yang terjadi jika anda membuat sebuah property atau method menjadi private, protected, public ?

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apakah keuntungan enkapsulasi bidang data ?



.....
.....
.....
.....
.....
.....
.....

LJ- 02 : Apa yang anda pahami terkait keyword private, protected, public?



.....
.....
.....
.....

LJ- 03 : Apa yang terjadi jika anda membuat sebuah property atau method menjadi private, protected, public ?



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....



.....

.....

.....

.....

.....

.....

LJ- 04:



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

g. Lembar Kerja Siswa.

10. Kegiatan Belajar 13 :Pewarisan

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 13 ini siswa diharapkan dapat :

Memahami konsep pewarisan

Menciptakan superclass dan subclass ***b.***

Uraian Materi

Konsep Inheritas

Konsep inheritance ini mengadopsi dunia riil dimana suatu entitas/obyek dapat mempunyai entitas/obyek turunan. Dengan konsep inheritance, sebuah class dapat mempunyai class turunan.Suatu class yang mempunyai class turunan dinamakan parent class atau base class. Sedangkan class turunan itu sendiri seringkali disebut subclass atau child class. Suatu subclass dapat mewarisi apa-apa yang dipunyai oleh parent class. Karena suatu subclass dapat mewarisi apa apa yang dipunyai oleh parent class-nya, maka member dari suatu subclass adalah terdiri dari apa-apa yang ia punyai dan juga apa-apa yang ia warisi dari class parent-nya. Kesimpulannya, boleh dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extend) parent class-nya.



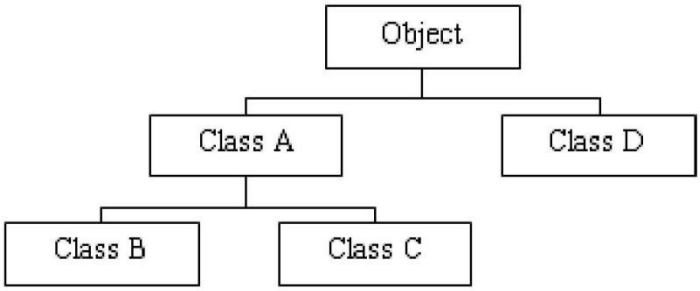
Gambar 36. Mamalia

Dari hirarki diatas dapat dilihat bahwa, semakin kebawah, class akan semakin bersifat spesifik. Class mamalia memiliki seluruh sifat yang dimiliki oleh binatang, demikian halnya juga macan , kucing, Paus dan Monyet memiliki seluruh sifat yang diturunkan dari class mamalia. Dengan konsep ini, karakteristik yang dimiliki oleh class binatang cukup didefinisikan didefinisikan dalam class binatang saja.

Class mamalia tidak perlu mendefinisikan ulang apa yang telah dimiliki oleh class binatang, karena sebagai class turunannya, ia akan mendapatkan karakteristik dari class binatang secara otomatis. Demikian juga dengan class macan, kucing, Paus dan monyet, hanya perlu mendefinisikan

karakteristik yang spesifik dimiliki oleh class-nya masing-masing.Dengan memanfaatkan konsep pewarisan ini dalam pemrograman, maka hanya perlu mendefinisikan karakteristik yang lebih umum akan didapatkan dari class darimana ia diturunkan.

Dalam Java,semuaclass,termasuk classyangmembangunJavaAPI,adalahsubclasses darisuperclassObject.Contohhirarkiclassdiperlihatkandi bawahini.Beberapa class di atas class utama dalam hirarki class dikenal sebagai superclass. Sementara beberapa class di bawah class pokok dalam hirarki class dikenal sebagai subclassdariclasstersebut.



Class hierarchy in Java.

Gambar 37. Hierarki Class di Java

Pewarisan adalah keuntungan besardalam pemrograman berbasis object karena suatu sifatataumethoddidefinisikandalam*superclass*,sifatinisecaraotomatisdiwariskandari semua *subclasses*.Jadi,Anda dapat menuliskan kode method hanya sekali dan mereka dapat digunakan oleh semua subclass.Subclass hanya perlu mengimplementasikan perbedaannyasendiri daninduknya.

Konsepinheritanceinimengadopsiduniariildimanasuatuentitas/obyekdapatmempunyaientitas/obyekturunan.Dengankonsepinheritance,sebuah classdapatmempunyaiclassturunan.Suatuclassyangmempunyaiclass turunandinamakanparentclassataubaseclass.Sedangkanclassturunanitu sendiriseringkalidisebutsubclassatauchildclass.Suatusubclassdapat mewarisiapa-apayangdipunyaiolehparentclass-nya,sehinggamemberdari suatusubclassadalahterdiridariapa-apayangiapunyaidanjugaapa-apa yangiawarisidariclassparent-nya.Kesimpulannya,bolehdikatakanbahwa suatusubclassadalahtidaklainhanyamemperluas(extend)parentclass-



nya.

Kapan menerapkan inheritas?

Kita baru perlu menerapkan inheritance pada saat kita jumpai ada suatu class yang dapat diperluas dari class lain.

Misal terdapat class Pegawaipublic class Pegawai {public String nama;public double gaji;}Misal terdapat class Managerpublic class Manajer {public String nama;public double gaji;public String departemen;}

Dari 2 buah class diatas, kita lihat class Manajer mempunyai data member yang identik sama dengan class Pegawai, hanya saja ada tambahan data member departemen.Sebenarnya yang terjadi disana adalah class Manajer merupakan perluasan dari class Pegawai dengan tambahan data member departemen.

Disini perlu memakai konsep inheritance, sehingga class Manajer dapat kita tuliskan seperti berikut :

public class Manajer extends Pegawai {public String departemen;}

Keuntungan inheritas

Subclass menyediakan state/behaviour yang spesifik yang membedakannya dengan superclass, hal ini akan memungkinkan programmer Java untuk menggunakan ulang source code dari superclass yang telah ada.

Programmer Java dapat mendefinisikan superclass khusus yang bersifat generik, yang disebut abstract class, untuk mendefinisikan class dengan behaviour dan state secara umum.

Deklarasi inheritas

DidalamJavauntukmendeklarasikansuatuclasssebagaisubclassdi lakukan
dengancaramenambahkankatakunciextendssetelahdeklarasinam
aclass, kemudiandiikutidengannamaparentclass-
nya.Katakunciextendstersebut
memberitahukompilerJavabahwakitainginmelakukanperluasancl
ass.

Berikutadalahcontohdeklarasiinheritance:

```
public class B extends A{
.....
}
```

ContohdiatasmemberitahukankompilerJavabahwakitainginmeng-extend classAkeclassB.Dengankatalain,classBadalahsubclass(classturunan) dariclassA,sedangkanclassAadalahparentclassdariclassB. Javahanyamemperkenankanadanyasingleinheritance.Konsepsingle inheritancehanyamemperbolehkansuatusublassmempunyaisatuparent class.Dengankonsepsingleinheritanceini,masalahpewarisanakan dapat diamatidenganmudah. Dalamkonsepdasarinheritancedikatakanbahwasuatusubclassadalahtidaklainhanyamemperluas(extend)parentclass-nya.Contoh:

Listing Program

```
publicclassPegawai{
publicStringnama;
publicdoublegaji;
        }
publicclassManajerextendsPegawai{

publicStringdepartemen;
        }
```

PadasaatclassManajermenurunkanataumemperluas(extend)classPegawai,makaiamewarisidatamemberyangdipunyaiolehclassPegawai. Dengan demikian,classManajermempunyaidatamemberyangdiwarisioleh Pegawai(nama,gaji),ditambahdengandatamemberyangiapunyai (departemen).



Kontrol Pengaksesan

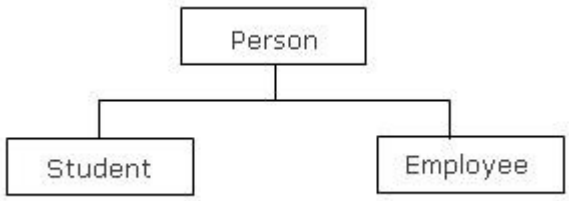
Pengaksesanmemberyangadadiparentclassdarisubclass-nyatidakjauh berbedadenganpengaksesanmembersubclassitusendiri.Contoh: Suatuparentclassdapattidakmewariskansebagianmember-nyakepada subclass-nya.Sejauhmanasuatumemberdapatdiwariskankeclasslain, ataupunsuatumemberdapatdiaksesdariclasslain,sangatberhubungan denganaccesscontrol(kontrolpengaksesan).Didalamjava,kontrol pengaksesandapatdigambarkandalamtabelberikutini:

Tabel 12. Kontrol Pengaksesan Class

Modifier	class yangsama	package	subclass	Class
private	√			
default	√	√		
protected	√	√	√	
public	√	√	√	√

2) MendefinisikanSuperclassdan Subclass

Untuk memperoleh suatu class,kita menggunakan kata kunci **extend**.Untuk mengilustrasikanini,kitaakanmembuatcontoh classinduk.Dimisalkankitamempunyai class indukyangdinamakan Person.



Listing Program

```
public class Person

{
```

```
protected Stringname;
protected Stringaddress;
public Person(){
System.out.println("Inside
Person:Constructor"); name = "";
address = "";
}
public Person(String name,String address ){
this.name = name;
this.address = address;
}
public String getName(){
return name;
}
public String getAddress(){
return address;
}
public void setName( String name )
{ this.name = name;
}
public void setAddress( String add )
{ this.address = add;
}
}
```

Perhatikanbahwaatribut*name*dan*address*dideklarasikansebagai**protected** .Alasannya kita melakukaniniyaitu,kitainginkanatribut-atributiniuntukbisadiaksesolehsubclasses dari superclassess.Jika kitamendeklarasikannya sebagai private,subclasses tidak dapat menggunakannya.Catatan bahwa semua properti dari superclass yang dideklarasikan sebagai**public, protecteddan default**dapat



diaksesolehsubclasses-nya.

Sekarang,kitainginmembuat class lain bernamaStudent.KarenaStudent jugasebagai Person,kitaputuskanhanyameng-*extend*classPerson,sehinggakitadapatmewariskan semuapropertidanmethoddarisetiapclassPersonyangada.Untukmelakukanini,kita tulis,

Listing Program

```
public class Student extends Person
{
public Student(){
System.out.println("Inside Student:Constructor");
//beberapa kode di sini
}
}
```

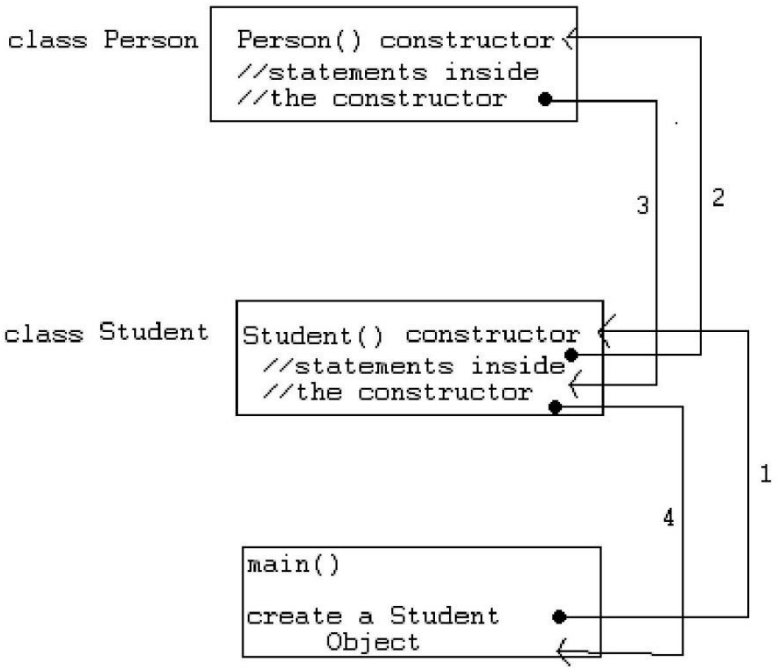
Ketika object Student di-*instantiate*,default constructor dari superclass secara mutlak memintauntukmelakukaninisialisasiyangseharusnya.Setelahitu,pernyataandidalam subclassdieksekusi. Untukmengilustrasikannya,perhatikankodeberikut,

Listing Program

```
public static
voidmain(String[]args )
{
Student anna = new Student();
}
```

Dalam kode ini,kita membuat sebuah object dari class Student.Keluaran dari program adalah,

```
Inside Person:Constructor
Inside Student:Constructor
```



Gambar 38. Alur Program

c. Rangkuman

Konsep inheritance ini mengadopsi dunia riil dimana suatu entitas/obyek dapat mempunyai entitas/obyek turunan. Dengan konsep inheritance, sebuah class dapat mempunyai class turunan. DidalamJavauntukmendeklarasikansuatuclasssebagaisubclassdilakukan dengancaramenambahkankatakunciextendssetelahdeklarasinamaclass, kemudiandiikutidengannamaparentclass-nya.Dalamkonsepdasarinheritancedikatakanbahwasuatusubclassadalahtidaklainhanyamemperluas(extend)parentclass-nya.Pengaksesanmemberyangadadiparentclassdarisubclass-nyatidakjauh berbedadenganpengaksesanmembersubclassitusendiri.semua properti dari superclass yang dideklarasikan sebagaipublic, protecteddan defaultdapat diaksesolehsubclasses-nya.

Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

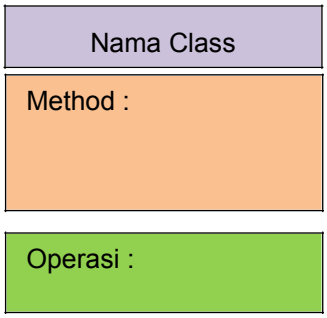
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan

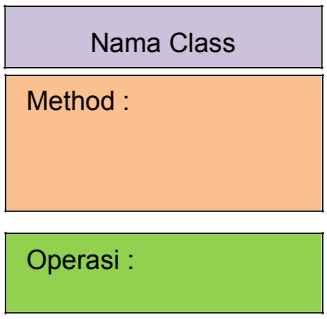
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	



8.

9.

10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



4. Apa arti keyword extends dan implements, kapan menggunakannya ?

Apa yang dimaksud dengan inheritance ?

Apa perbedaan superclass dan subclass

Apa yang anda ketahui tentang interface di java ?

f. Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apa arti keyword extends dan implements, kapan menggunakannya



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LJ- 02 : Apa yang dimaksud dengan inheritance ?



.....
.....
.....
.....
.....

LJ- 03 : Apa perbedaan superclass dan subclass



a. Superclass :

d. .....
.....
.....
.....
.....

Subclasss:

.....
.....
.....
.....
.....
.....

LJ- 04: Apa yang anda ketahui tentang interface di java ?



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....



g. Lembar Kerja Siswa.

11. Kegiatan Belajar 13 :Pewarisan

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 14 siswa diharapkan dapat :

Memahami penggunaan kata kunci super

Menerapkan penggunaan kata kunci cuper dalam inheritas

Memahami konsep overloading dan overriding

Menyajikan overloading dan overriding dalam class ***b.***

Uraian Materi

1) Kata kunci super

Subclassjuga dapatmemanggilconstructorsecara eksplisitdarisuperclass terdekat.Hal ini dilakukan dengan pemanggil construktor**super**.Pemanggil constructorsuperdalam constructordari subclass akan menghasilkan eksekusi dari superclass constructoryang bersangkutan,berdasardariargumensebelumnya.Sebagaicontoh,padacon tohclasssebelumnya.PersondanStudent,kita tunjukkancontoh daripemanggilconstructorsuper.DiberikankodeberikutuntukStudent,

Sintaks kunci super

```
public Student(){

super( "SomeName", "SomeAddress" );

System.out.println("Inside Student:Constructor");

}
```

Kode ini memanggil constructorkedua dari superclass terdekat (yaitu Person) dan mengeksekusinya.Contohkodelain ditunjukkansebagai berikut,

Kode ini memanggil constructorkedua dari superclass terdekat (yaitu Person) dan mengeksekusinya.Contohkodelain ditunjukkansebagai berikut,



Listing Program

```
public Student(){
super();
System.out.println("Inside Student:Constructor");
}
```

Kode ini memanggil default constructordari superclass terdekat (yaitu Person) dan mengeksekusinya.Ada beberapahal yangharusdiingatketikamenggunakan pemanggil constuktorsuper:

a. Pemanggil super() HARUSDIJADIKAN PERNYATAAN PERTAMADALAMconstructor.
b. Pemanggil super()hanyadapat digunakan dalamdefinisi constructor.
c. Termasuk constructorthis() dan pemanggil super() TIDAK BOLEH TERJADIDALAM constructorYANGSAMA.

Pemakaian lain dari superadalah untuk menunjuk anggota dari superclass(seperti reference**this**).Sebagaicontoh,

Listing Program

```
public Student()
{
super.name = "somename";
super.address = "some address";
}
```

c. Rangkuman

Subclassjuga dapatmemanggilconstructorsecara eksplisitdarisuperclass terdekat.Hal ini dilakukan dengan pemanggil construktorsuper.Pemanggil constructorsuperdalam constructordari subclass akan menghasilkan eksekusi dari superclass constructoryang bersangkutan. Pemanggil super()hanyadapat digunakan dalamdefinisi constructor.Termasuk constructorthis() dan pemanggil super() tidak boleh terjadi dalamconstructor yang sama.

Pemakaian lain dari superadalah untuk menunjuk anggota dari superclass(seperti reference**this**).

Tugas

Tugas 1

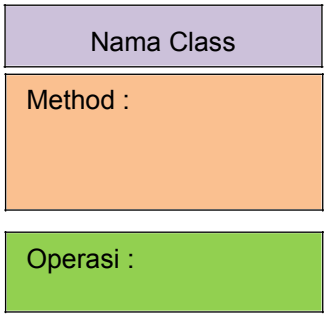
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama

Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan



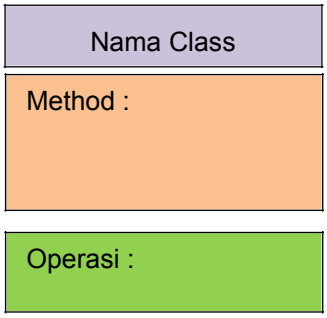
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	

10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Apakah fungsi dari kata kunci super ?

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apakah fungsi dari kata kunci super ?



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 02 :





.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 :



.....
.....
.....
.....
.....
.....
.....

LJ- 04:



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

g. Lembar Kerja Siswa.



Kegiatan Belajar 13 :Pewarisan *a.*

Tujuan Pembelajaran

Memahami konsep overloading dan overriding

Memahami metode final

Menyajikan overloading dan overriding dalam class ***b.***

Uraian Materi

Metode Overloading

Overloadingadalahsuatukeadaandimanabeberapamethodsekaligusdapat mempunyainamayangsama,akantetapimempunyaifungsionalitasyang berbeda.

Overloadinginidapatterjadipadaclassyangsamaataupadasuatuparentclassdan subclass-nya.Overloadingmempunyaiciri-cirisebagaiberikut:
Namamethodharussama
Daftarparameterharusberbeda
Returntypebolehsama,jugabolehberbeda

Contohpenggunaanoverloadingdilihatdibawahini:

```
Gambar(intt 1)      →   1parametertitik,untukmenggambartitik

Gambar(intt 1, int t2)   →   2parametertitik,untukmenggambargaris

Gambar(intt 1, int t2, int t3)   →   3parametertitik,untukmenggambarsegitiga

Gambar(intt 1, int t2, int t3, int t4)   →   3parametertitik,untukmenggambarsegiempat
```

Overriding Method

Overridingadalahsuatukeadaandimanamethodpadasubclassmenolak methodpadaparentclass-nya.Overridingmempunyaiciri-cirisebagai berikut:

Namamethodharussama
Daftarparameterharussama
Returntypeharussama

Untuk beberapa pertimbangan,terkadang class asal perlu mempunyai implementasi berbedadari methodyangkhusus dari *superclass* tersebut.Olehkarenaitulah,method overridingdigunakan.*Subclass*dapatmengesampingkanmethodyangdidefinisikandalam *superclass*denganmenyediakanimplementasi barudari methodtersebut.Misalnyakitamempunyaiimplementasi berikutuntukmethodgetName dalamsuperclass Person,

Listing Program

```
public class Person
{
:
public String getName()
{
System.out.println("Parent: getName");
return name;
}
  }
```

Untukoverride,methodgetNamedalamsubclassStudent,kita tulis,

Listing Program

```
public class Student extends Person
{
public String getName()
{
System.out.println("Student: getName");
return name;
}
}
```

Jadi,ketika kita meminta method getName dari object class Student,method overridde akandipanggil,keluarannya akanmenjadi,



```
Student: getName
```

3) .Method final danclassfinal

Dalam Java,juga memungkinkan untuk mendeklarasikan class-class yang tidak lama menjadi subclass.Class ini dinamakan **class final**.Untuk mendeklarasikan class untuk menjadi final kita hanya menambahkan kata kunci **final**dalam deklarasi class.Sebagai contoh,jika kita ingin class Person untuk dideklarasikan final, kita tulis,

```
public final class Person
{
        //area kode
}
```

Beberapa class dalam Java API dideklarasikan secara final untuk memastikan sifatnya tidak dapat di-*override*. Contoh-contoh dari class ini adalah Integer , Double , dan String. Ini memungkinkan dalam Java membuat method yang tidak dapat di-*override.* Method ini dapat kita panggil **method final**. Untuk mendeklarasikan method untuk menjadi final,kita tambahkan kata kunci final kedalam deklarasi method. Contohnya, jika kita ingin method getName dalam class Person untuk dideklarasikan final, kita tulis,

Sintaks getName

```
public final String getName(){

return name;

}
```

Method static juga secara otomatis final.Ini artinya Anda tidak dapat membuatnya override.

c. Rangkuman

Overloading adalah suatu keadaan dimana beberapa method sekaligus dapat mempunyai nama yang sama, akan tetapi mempunyai fungsionalitas yang berbeda. Overriding adalah suatu keadaan dimana method pada subclass menolak method pada parent class-nya. *Subclass* dapat mengesampingkan method yang didefinisikan dalam *superclass* dengan menyediakan implementasi baru dari method tersebut. Dalam Java, juga memungkinkan untuk mendeklarasikan class-class yang tidak lama menjadi

subclass.Class ini dinamakan **class final**.Untuk mendeklarasikan class untuk menjadi final kita hanya menambahkan kata kunci **final**dalam deklarasi class. Beberapa class dalam Java API dideklarasikan secara final untuk memastikan sifatnya tidak dapatdi-*override*.

Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

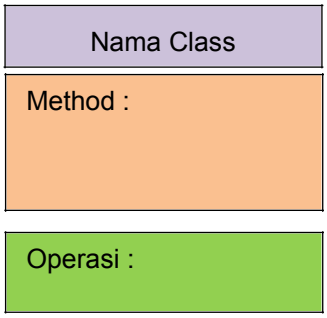
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama



Tugas 2

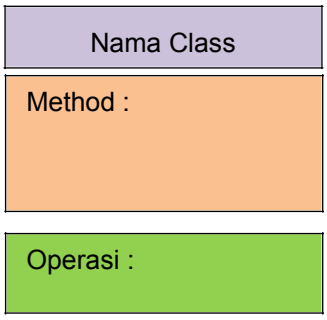
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	

7.

8.

9.

10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Apa arti dari override ?

Apa yang dimaksud dengan method overload ?

Apa perbedaan Method overriding dan method overload ?

Bagaimana anda secara eksplisit memanggil suatu konstruktor superclass dari subclass ?

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apa arti dari override ?



.....

.....

.....

.....

.....

.....

.....



.....
.....
.....
.....
.....

LJ- 02 : Apa yang dimaksud dengan method overload ?



.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 : Apa perbedaan Method overriding dan method overload ?



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 04: Bagaimana anda secara eksplisit memanggil suatu konstruktor superclass dari subclass ?



.....
.....
.....
.....
.....
.....
.....
.....
.....

g. Lembar Kerja Siswa.



13. Kegiatan Belajar 16 :Polimorphisme

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 16 siswa diharapkan dapat :

Memahami konsep polimorfisme

Menyajikan overloading dan overriding dalam class ***b.***

Uraian Materi

1) Metode Overloading

Polymorphism merupakan salah satu konsep penting dalam object oriented programming (OOP) khusunya di bahasa Java setelah abstraction dan inheritance. Polymorphism berarti banyak bentuk. Ada beberapa definisi berbeda tentang polymorphism yang berkaitan dengan pemrograman berorientasi obyek. Sedangkan apa yang dimaksud dengan polymorphism sendiri, sebenarnya sulit untuk didefinisikan. Sejalan dengan contoh yang diberikan, Anda diharapkan dapat mengerti dan memahami konsep polymorphism itu sendiri.

Polymorphism sering dikaitkan dengan penggunaan lebih dari satu metoda dengan nama sama. Penggunaan metoda dengan nama sama dapat diterapkan dengan method overloading dan method overriding. Peran polymorphism sebenarnya tidak terbatas hanya pada hal tersebut. Ada keterkaitan antara polymorphism dan inheritance (turunan).

Dalam konsep turunan, saat obyek dari subclass dikonstruksi, obyek dari superclass juga ikut dikonstruksi. Jadi setiap instance dari subclass adalah juga instance dari superclass. Apabila Anda mendeklarasikan metoda dengan parameter dari tipe superclass, Anda diperbolehkan untuk memberi argumen berupa obyek subclass yang merupakan turunan dari superclass tersebut.

Berikut ini adalah contoh program yang dapat memberikan gambaran berkaitan dengan konsep polymorphism. Perlu dipahami dan dimengerti bahwa kelas Object merupakan akar dari semua kelas Java dan menduduki puncak tertinggi dalam hirarkhi. Program akan mendefinisikan kelas yang berkaitan dengan bidang datar secara sederhana termasuk beberapa kelas turunannya (kelas PersegiPanjang dan Balok) dan membatasi hanya pada penerapan method overriding.

Listing Program

```
Nama file : Polimorphism.java

Contoh penerapan konsep polimorphism


public class Polimorphism {


   public static void main(String[ ] args) {


      cetakObjek(new Balok());

      cetakObjek(new PersegiPanjang());

      cetakObjek(new BangunDatar());

      cetakObjek(new Object());

   }


   public static void cetakObjek(Object objek)

      { System.out.println(objek);

   }


} // Akhir kelas Polimorphism


class Balok extends PersegiPanjang {

   public String toString() {

      return "Mempunyai sisi panjang, lebar dan
tinggi";

   }

}


class PersegiPanjang extends BangunDatar {
```



```
    public String toString() {
        return "Mempunyai sisi panjang dan lebar";
    }
}

class BangunDatar extends Object {
    public String toString() {
        return "Mempunyai berbagai bentuk";
    }
}
```

Baris nomor 14 -16 adalah deklarasi metoda cetakObjek yang mempunyai satu parameter dengan tipe kelas Object. Kelas Object merupakan akar dari semua kelas di Java. Langsung maupun tidak langsung, semua kelas di Java merupakan turunan dari kelas Object. Anda dapat memanggil atau menggunakan metoda cetakObjek dengan argumen berupa obyek yang dibuat dari kelas turunan superclass Object.

Ketika metoda cetakObjek dipanggil (baris nomor 8 – 11), argumen obyek akan diminta. obyek sebagai argumen metoda dapat berupa obyek yang merupakan kelas turunan dari kelas Object yaitu kelas BangunDatar, kelas PersegiPanjang maupun kelas Balok. Masing-masing kelas turunan mendeklarasikan ulang metoda toString yang mempunyai implementasi berbeda. Java Virtual Machine (JVM) akan menentukan secara dinamis implementasi metode toString yang akan digunakan saat program dijalankan. Kemampuan menentukan secara dinamis ini disebut dengan dynamic binding.

Dari gambaran program di atas, apabila parameter sebuah metoda adalah tipe superclass, maka argumen metoda yang diberikan dapat berupa tipe dari subclassnya. Kemampuan seperti inilah yang dimaksud dengan polymorphism. Dari gambaran tersebut, dapat didefinisikan kembali bahwa

polymorphism adalah kemampuan untuk menghasilkan sesuatu yang berbeda dengan cara yang sama. Pemberian obyek dari subclass ke obyek dari superclass dapat dilakukan tanpa perlu melakukan konversi.

Virtual Methode Invocation

Virtual Method Invocation (VMI) bias terjadi jika terjadipolimorfismedan overriding. Pada saat obyek yang sudah di buat tersebut memanggil overridden method pada parent class, compiler Java akan melakukan invocation (pemanggilan) terhadap overriding method pada subclass , dimana yang seharusnya di panggil adalah overridden method. Berikut contoh terjadinya VMI:

Listing Program

```
classParent{
intx=5;
publicvoidInfo(){
System.out.println("IniclassParent");
}
}
classChildextendsParent{
intx=10;
publicvoidInfo(){
System.out.println("IniclassChild");
}
}
publicclassTes{
publicstaticvoidmain(Stringargs[]){
Parenttes=newChild();
System.out.println("Nilaix="+tes.x);
tes.Info();
}
}
```



Hasil dari running program diatas adalah sebagai berikut:

```
Nilaix=5

IniclassCh
ild
```

Polymorphic arguments adalah tipe suatu parameter yang menerima suatu Nilai yang bertipe subclass-nya. Berikut contoh dari polymorphics arguments:

Listing Program

```
classPegawai{

}

classManajerextendsPegawai{

                              …

}

publicclassTes{

publicstaticvoidProses(Pegawaipeg){

                                      …

}


publicstaticvoidmain(Stringargs[]){

Manajerman=newManajer(); Proses(man);

}

   }
```

Pernyataan instance of sangat berguna untuk mengetahui tipe asal dari suatu

Polymorphic arguments. Untuk lebih jelasnya, misalnya dari contoh program sebelumnya, kita sedikit membuat modifikasi pada class Tes dan ditambah sebuah class baru Kurir, seperti yang tampak dibawah ini:

Listing Program

```
classKurirextendsPegawai
{
}
publicclassTes{

publicstaticvoidProses(Pegawaipeg){

if(peginstanceofManajer){

…lakukantugas-tugasmanajer…

}elseif(peginstanceofKurir){

…lakukantugas-tugaskurir…

}else{

…lakukantugas-tugaslainnya…

}

}
publicstaticvoidmain(Stringargs[]){

Manajerman=newManajer();
Kurirkur=newKurir();


Proses(man);
Proses(kur);
}
}
```



Seringkali pemakaian instance of diikuti dengan casting object dari tipe parameter ke tipe asal. Misalkan saja program kita sebelumnya. Pada saat kita sudah melakukan instance of dari tipe manajer, kita dapat melakukan casting object ke tipe asalnya, yaitu manajer. Caranya adalah seperti berikut:

Listing Program

```
if(peginstanceofManajer)

{

 Manajerman=(Manajer)peg;

…lakukantugas-tugasmanajer…


}
```

c. Rangkuman

Polymorphism merupakan salah satu konsep penting dalam object oriented programming (OOP) khusunya di bahasa Java setelah abstraction dan inheritance. Polymorphism berarti banyak bentuk.Polymorphism sering dikaitkan dengan penggunaan lebih dari satu metoda dengan nama sama. Penggunaan metoda dengan nama sama dapat diterapkan dengan method overloading dan method overriding. Pada saat obyek yang sudah dibuat tersebut memanggil overridden method pada parent class, compiler Java akan melakukan invocation (pemanggilan) terhadap overriding method pada subclass dimana yang seharusnya dipanggil adalah overridden method. Polymorphic arguments adalah tipe suatu parameter yang menerima suatu nilai yang bertipe subclass-nya. Pernyataan instance of sangat berguna untuk mengetahui tipe asal dari suatu polymorphic arguments } Seringkali pemakaian instance of diikuti dengan casting object dari tipe parameter ke tipe asal.

Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

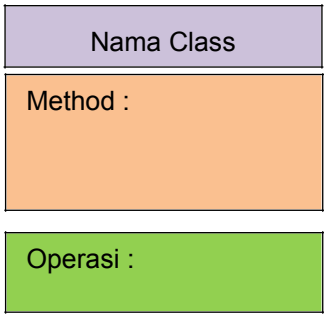
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama



Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan

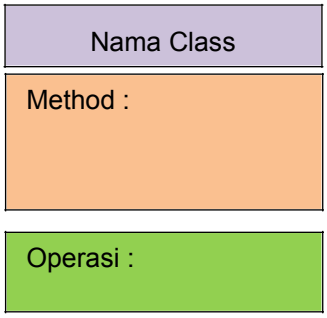
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	

6.

7.

8.

9.

10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Apa itu polymorphisme ?

Apa yang dimaksud dengan polymorphic argument ?

Apa yang dimaksud dengan VirtualMethodInvocation(VMI) ?

Jelaskan perbedaan antara *overloading method* dan *overridding method*

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Apa itu polymorphisme:



.....

.....

.....

.....



.....
.....
.....
.....
.....
.....
.....
.....

LJ- 02 : Apa yang dimaksud dengan polymorphic argument ?



.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 : Apa yang dimaksud dengan VirtualMethodInvocation(VMI) ?



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 04: Jelaskan perbedaan antara *overloading method* dan *overridding method*



.....

.....

.....

.....

.....

.....

.....

.....

.....

g. Lembar Kerja Siswa.



14. Kegiatan Belajar 18 :Package

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 18 siswa diharapkan dapat :

Memahami konsep polimorfisme

Menyajikan overloading dan overriding dalam class ***b.***

Uraian Materi

1) Package

Package adalah sebuah sarana untuk mengelompokkan atau mengorganisasikan kelas dan interface yang sama atau sekelompok menjadi satu unit tunggal dalam library. Package mempengaruhi mekanisme hak akses ke kelas didalamnya. hal terpenting yang diperhatikan pada saat mendeklarasikanpackage, bahwa class tersebut harus disimpan pada suatu directory yang sama dengan nama package nya. Alasan menggunakan package pada java ialah untuk menghindari tabrakan nama kelas yang akan dibuat dengan nama kelas yang sudah ada. Selain itu, salah satu yang menjadi keuntungan menggunakan package adalah untuk mudahnya developer dalam hal mencari dan me-manage akses yang diberikan. mengerti akan konsep dari package akan membantu mengelola dan menggunakan file yang disimpan didalam JAR (Java Archive)

Package juga mempengaruhi mekanisme hak akses ke kelas-kelas di dalamnya.

Pengaruh Package terhadap Method main()

Kelas yang mengandung method main() memilki syarat tidak berada dalam suatu package, dan hirarki posisi foldernya di atas package yang diimport.

Membuat Package

Ada tiga langkah untuk membuat package :
Mendeklarasikan dan memberi nama package.

Membuat struktur dan nama direktori yang sesuai dengan struktur dan nama package.

Mengkompilasi kelas-kelas sesuai dengan packagenya masing-masing.

Mendeklarasikan dan Memberi Nama Package

Deklarasi package harus diletakkan pada bagian paling awal (sebelum deklarasi import) dari source code setiap kelas yang dibungkus package tersebut.

Bentuk umum deklarasi package :

package namaPackage;

Deklarasi tersebut akan memberitahukan kompilator, ke library manakah suatu kelas dikompilasi dan dirujuk.

Syarat nama package :

Diawali huruf kecil

Menggambarkan kelas-kelas yang dibungkusnya

Harus unik (berbeda dengan nama package standard)

Merepresentasikan path dari package tersebut.

Harus sama dengan nama direktorinya.

Contoh package standard :

java.lang (berisi kelas-kelas fundamental yang sering digunakan).

java.awt dan *javax.swing* (berisi kelas-kelas untuk membangun aplikasi GUI)

java.io (berisi kelas-kelas untuk proses input output)

Membuat Struktur Direktori

Pada langkah ini, buatlah direktori menggunakan file manager (di windows menggunakan explorer) sesuai struktur package dari langkah sebelumnya. Kemudian tempatkan kelas-kelas tersebut ke direktori yang bersesuaian (mirip seperti menyimpan file-file ke dalam folder).

Package dapat bersarang di package lain, sehingga dapat dibuat hirarki package.

Bentuk umum pernyataan package multilevel :

package namaPackage1[.namaPackage2[.namaPackage3]];



Contoh hirarki package di JDK :

package java.awt.image;

Compile dan Run Kelas dari suatu Package

Selanjutnya masing-masing kelas tersebut dalam package tersebut dikompilasi menjadi byte code (*.class). Artinya package tersebut siap digunakan.

Menggunakan Package

Ada dua cara menggunakan suatu package yaitu :

Kelas yang menggunakan berada dalam direktori (package) yang sama dengan kelas-kelas yang digunakan. Maka tidak diperlukan import.

Kelas yang menggunakan berada dalam direktori (package) yang berbeda dengan kelas-kelas yang digunakan. Maka pada awal source code di kelas pengguna harus mencantumkan :

import namaPackage.NamaKelas; atau

import namaPackage.;*

Contoh :

import

java.text.DecimalFormat;

import javax.swing.;*

Setting Classpath

Path hirarki package, didaftarkan sebagai salah satu nilai variabel lingkungan yang bernama Classpath.Classpath diset dengan aturan : berawal dari drive (C:\ atau D:\) sampai dengan satu tingkat sebelum kita mendeklarasikan package.

Rangkuman

Package adalah sebuah sarana untuk mengelompokkan atau mengorganisasikan kelas dan interface yang sama atau sekelompok menjadi satu unit tunggal dalam library. Alasan menggunakan package pada java ialah untuk menghindari tabrakan nama kelas yang akan dibuat dengan nama kelas yang sudah ada.masing-masing kelas tersebut dalam package tersebut dikompilasi menjadi byte code (*.class).Path hirarki package, didaftarkan sebagai salah satu nilai variabel lingkungan yang bernama Classpath.Classpath diset dengan aturan.

Tugas

Tugas 1

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan proses pengamatan benda dan obyek disekitar anda dan coba dihubungkan dengan pendekatan berorientasi objek yang akan memandang sistem yang dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

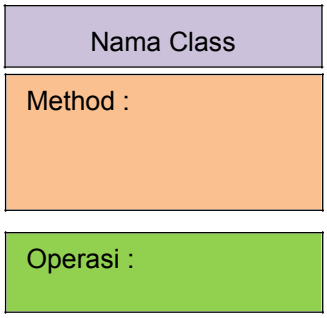
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

❖ Bandingkan dan Simpulkan

Bandingkan hasil pendiskripsian suatu benda yang meliputi (nama benda, atribut, operasi) dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil perbandingan tersebut hal penting apa yang harus dirumuskan secara bersama



Tugas 2

Penugasan dalam kegiatan belajar untuk adalah melatih keterampilan anda dengan melakukan pengamatan beberapa bahasa pemrograman yang sudah anda kenal, amati dan diskusikan

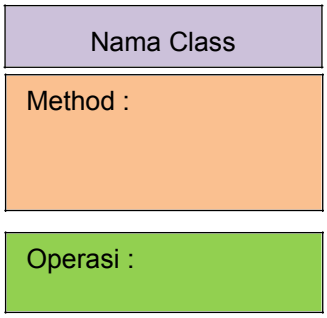
❖ Mengamati Listing Program dan Output Program

Menentukan nama Class

Menentukan variabel yang digunakan

Menentukan nama Method

Gambar Class Diagram



Buatlah listing program

Compile dan debug program

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	

8.

9.

10.

❖ Bandingkan dan Simpulkan

Bandingkan hasil tabel pengamamatan terhadap 2 bahasa pemrograman yang sudah anda kenal dari hasil kerja kelompok anda dengan kelompok lain.

Berdasarkan hasil pengamatam tersebut hal penting apa yang harus dirumuskan secara bersama

e. Test Formatif.

Dalam test ini setiap andaharus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



1. Bagaimana cara mendeklarasikan dan memberi nama pada package ?
   Apa yang dimaksud dengan package ?
   Sebutkan syarat-syarat memberi nama pada package ?
   Apakah package berpengaruh pada class lainya ? Jika iya berikan alasanya ?

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Bagaimana cara mendeklarasikan dan memberi nama pada package ?



.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 02 : Apa yang dimaksud dengan package ?





.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 : Sebutkan syarat-syarat memberi nama pada package ?



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 04: Apakah package berpengaruh pada class lainya ? Jika iya berikan alasanya ?



.....
.....
.....
.....
.....
.....
.....
.....
.....

g. Lembar Kerja Siswa.



Daftar Pustaka

Christian Musnter, Java 2 JDK 5 – Grundlagen , Herdt – Verlag for bildungsmedien Gagh, Bodenheilm, 2006

C. Thomas wu, An Introduction to Object - Oriented Programming with Java, McGraw Hill 2001

Deitel, Java : How to program, Prentice Hall, New jersey, 2002

Joyce Avestro , Introduction Programming 1, Java Education Development Initiatif, 2003

Joyce Avestro , Introduction Programming 2, Java Education Development Initiatif, 2003

Patric Noughton , The Java Handbook, McGrawHill, Inc, 2006

R.H.Sianipar , Teori dan Implementasi Java, Informatika Bandung, 2013

HALAMAN SAMPUL

PEMOGRAMAN BERORIENTASI OBJEK

Semester 2

Untuk SMK / MAK Kelas XI

Penulis : **Eko Subiyantoro**
Editor Materi : **Joko Pitono**
Editor Bahasa :
Ilustrasi Sampul :
Desain & Ilustrasi Buku : **PPPPTK BOE Malang**
Hak Cipta © 2013, Kementerian Pendidikan & Kebudayaan

MILIK NEGARA
TIDAK DIPERDAGANGKAN

Semua hak cipta dilindungi undang-undang.

Dilarang memperbanyak (mereproduksi), mendistribusikan, atau memindahkan sebagian atau seluruh isi buku teks dalam bentuk apapun atau dengan cara apapun, termasuk fotokopi, rekaman, atau melalui metode (media) elektronik atau mekanis lainnya, tanpa izin tertulis dari penerbit, kecuali dalam kasus lain, seperti diwujudkan dalam kutipan singkat atau tinjauan penulisan ilmiah dan penggunaan non-komersial tertentu lainnya diizinkan oleh perundangan hak cipta. Penggunaan untuk komersial harus mendapat izin tertulis dari Penerbit.

Hak publikasi dan penerbitan dari seluruh isi buku teks dipegang oleh Kementerian Pendidikan & Kebudayaan.

Untuk permohonan izin dapat ditujukan kepada Direktorat Pembinaan Sekolah Menengah Kejuruan, melalui alamat berikut ini:

Pusat Pengembangan Pemberdayaan Pendidik dan Tenaga Kependidikan Bidang Otomotif dan Elektronika:

Jl. Teluk Mandar, Arjosari Tromol Pos 5, Malang 65102, Telp. (0341) 491239, (0341) 495849, Fax. (0341) 491342, Surel: vedcmalang@vedcmalang.or.id, Laman: www.vedcmalang.com

DISKLAIMER (*DISCLAIMER*)

Penerbit tidak menjamin kebenaran dan keakuratan isi/informasi yang tertulis di dalam buku tek ini.Kebenaran dan keakuratan isi/informasi merupakan tanggung jawab dan wewenang dari penulis.

Penerbit tidak bertanggung jawab dan tidak melayani terhadap semua komentar apapun yang ada didalam buku teks ini.Setiap komentar yang tercantum untuk tujuan perbaikan isi adalah tanggung jawab dari masing-masing penulis.

Setiap kutipan yang ada di dalam buku teks akan dicantumkan sumbernya dan penerbit tidak bertanggung jawab terhadap isi dari kutipan tersebut. Kebenaran keakuratan isi kutipan tetap menjadi tanggung jawab dan hak diberikan pada penulis dan pemilik asli.Penulis bertanggung jawab penuh terhadap setiap perawatan (perbaikan) dalam menyusun informasi dan bahan dalam buku teks ini.

Penerbit tidak bertanggung jawab atas kerugian, kerusakan atau ketidaknyamanan yang disebabkan sebagai akibat dari ketidakjelasan, ketidaktepatan atau kesalahan didalam menyusun makna kalimat didalam buku teks ini.

Kewenangan Penerbit hanya sebatas memindahkan atau menerbitkan mempublikasi, mencetak, memegang dan memproses data sesuai dengan undang-undang yang berkaitan dengan perlindungan data.

Katalog Dalam Terbitan (KDT)
Rekayasa Perangkat Lunak Edisi Pertama 2013
Kementerian Pendidikan & Kebudayaan
Direktorat Jenderal Peningkatan Mutu Pendidik & Tenaga Kependidikan,
th. 2013: Jakarta

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan yang Maha Esa atas tersusunnya buku teks ini, dengan harapan dapat digunakan sebagai buku teks untuk siswa Sekolah Menengah Kejuruan (SMK) Bidang Studi Keahlian Rekayasa Perangkat Lunak.

Penerapan kurikulum 2013 mengacu pada paradigma belajar kurikulum abad 21 menyebabkan terjadinya perubahan, yakni dari pengajaran (*teaching*) menjadi BELAJAR (*learning*), dari pembelajaran yang berpusat kepada guru (*teachers-centered*) menjadi pembelajaran yang berpusat kepada peserta didik (*student-centered*), dari pembelajaran pasif (*pasive learning*) ke cara belajar peserta didik aktif (*active learning-CBSA*) atau *Student Active Learning-SAL*.

Buku teks "Pemrograman Berorientasi Obyek" ini disusun berdasarkan tuntutan paradigma pengajaran dan pembelajaran kurikulum 2013 diselaraskan berdasarkan pendekatan model pembelajaran yang sesuai dengan kebutuhan belajar kurikulum abad 21, yaitu pendekatan model pembelajaran berbasis peningkatan keterampilan proses sains.

Penyajian buku teks untuk Mata Pelajaran "Pemrograman Berorientasi Obyek " ini disusun dengan tujuan agar supaya peserta didik dapat melakukan proses pencarian pengetahuan berkenaan dengan materi pelajaran melalui berbagai aktivitas proses sains sebagaimana dilakukan oleh para ilmuwan dalam melakukan eksperimen ilmiah (penerapan scientifik), dengan demikian peserta didik diarahkan untuk menemukan sendiri berbagai fakta, membangun konsep, dan nilai-nilai baru secara mandiri.

Kementerian Pendidikan dan Kebudayaan, Direktorat Pembinaan Sekolah Menengah Kejuruan, dan Direktorat Jenderal Peningkatan Mutu Pendidik dan Tenaga Kependidikan menyampaikan terima kasih, sekaligus saran kritik demi kesempurnaan buku teks ini dan penghargaan kepada semua pihak yang telah berperan serta dalam membantu terselesaikannya buku teks siswa untuk Mata Pelajaran basis data kelas XI / Semester 1 Sekolah Menengah Kejuruan (SMK).

Jakarta, 12 Desember 2013

Menteri Pendidikan dan Kebudayaan

Prof. Dr. Mohammad Nuh, DEA

DAFTAR ISI

HALAMAN SAMPUL....i
LAMAN FRANCISDISKLAIMER (*DISCLAIMER*)....ii
KATA PENGANTAR....iii
DAFTAR ISI....iv
GLOSARIUM....vii
PETA KEDUDUKAN BUKU....ix
Peta Konsep : Pemrograman Berorientasi Obyek Kelas XI Semester 2....x
BAB I PENDAHULUAN....1
A. Deskripsi....1
B. Prasyarat....2
C. Petunjuk Penggunaan....3
D. Tujuan Akhir....3
E. Kompetensi Inti Dan Kompetensi Dasar....4
F. Cek Kemampuan Awal....5
BAB II KEGIATAN BELAJAR....6
1. Kegiatan Belajar 1 : Interface....6
A. Tujuan Pembelajaran....6
B. Uraian Materi....6
C. Rangkuman....15
D. Tugas....16
E. Tes Formatif....18
F. Lembar Jawaban Test Formatif (LJ)....19
G. Lembar Kerja Siswa....21

2. Kegiatan 2 :Class Built-in 22
A. Tujuan Pembelajaran 22
B. Uraian Materi 22
C. Rangkuman 32
D. Tugas 33
E. Tes Formatif 35
F. Lembar Jawaban Test Formatif (LJ) 36
G. Lembar Kerja Siswa 37
3. Kegiatan 3 : Exception Handling 38
A. Tujuan Pembelajaran 38
B. Uraian Materi 38
C. Rangkuman 52
D. Tugas 53
E. Tes Formatif 54
F. Lembar Jawaban Test Formatif (LJ) 56
G. Lembqar Kerja Siswa 58
4. Kegiatan 4 : String 59
A. Tujuan Pembelajaran 59
B. Uraian Materi 59
C. Rangkuman 72
D. Tugas 73
E. Tes Formatif 74
F. Lembar Jawaban Test Formatif (LJ) 75
G. Lembar Kerja Siswa 76

5. Kegiatan 5 : Array **77**
A. Tujuan Pembelajaran 77
B. Uraian Materi 77
C. Rangkuman 90
D. Tugas 91
E. Tes Formatif 93
F. Lembar Jawaban Test Formatif (LJ) 94
G. Lembar Kerja Siswa 96
6. Kegiatan 6 : Sistem File **97**
A. Tujuan Pembelajaran 97
B. Uraian Materi 97
C. Rangkuman 103
D. Tugas 105
E. Tes Formatif 106
F. Lembar Jawaban Test Formatif (LJ) 107
G. Lembar Kerja Siswa 109
DAFTAR PUSTAKA 110

GLOSARIUM

Abstract class adalah class yang mempunyai sedikitnya satu abstract method. Abstract class hanya bisa digunakan sebagai super class, dan dapat diturunkan dari class abstract lainnya.

Abstract method adalah method yang belum mempunyai implementasi.

Arrayadalah suatu kumpulan data pada suatu variabel. Array digunakan untuk membuat variabel bisa menampung beberapa data dengan tipe data yang sama alias satu tipe data.

Class StringBuffer adalah pasangan class String yang menyediakan banyak fungsi string yang umum.

Class Wrapper adalah representasi objek sederhana dari variabel- variable non-objek yang sederhana. Ada 10 tipe data Wrapper, yaitu Boolean, Byte, Character, Double, Float, Integer, Long, Number, Short, dan Void.

Collection merupakan istilah umum yang dipakai untuk setiap objek yang berfungsi untuk mengelompokkan beberapa objek tertentu menggunakan suatu teknik tertentu pula.

Exception adalah sebuah event yang menjalankan alur proses normal pada program.

Finally merupakan keyword pada class exception handling yang menunjukan bahwa blockprogram tersebut akan selalu dieksekusi meskipun adanya kesalahan yang muncul atau pun tidak ada.

Interface merupakan sekumpulan dari method-method yang dibuat, namun belum ada operasi di dalam tubuh method tersebut.

Konstruktor merupakan method khusus yang dipakai oleh Java untuk membuat sebuah object didalam kelas dan tiap kelas boleh memiliki lebih dari satu konstruktor.

List merupakan pengelompokan berdasarkan urutan seperti layaknya array, karena itu ia memiliki posisi awal dan juga posisi akhir.

Modifier digunakan untuk menentukan sifat dari suatu kelas dan menentukan preveledge (hak akses) dari kelas lain.

PrintWriter adalah class turunan dari Writer yang memiliki metode tambahan untuk menulis tipe data Java dalam karakter yang bisa dibaca manusial.

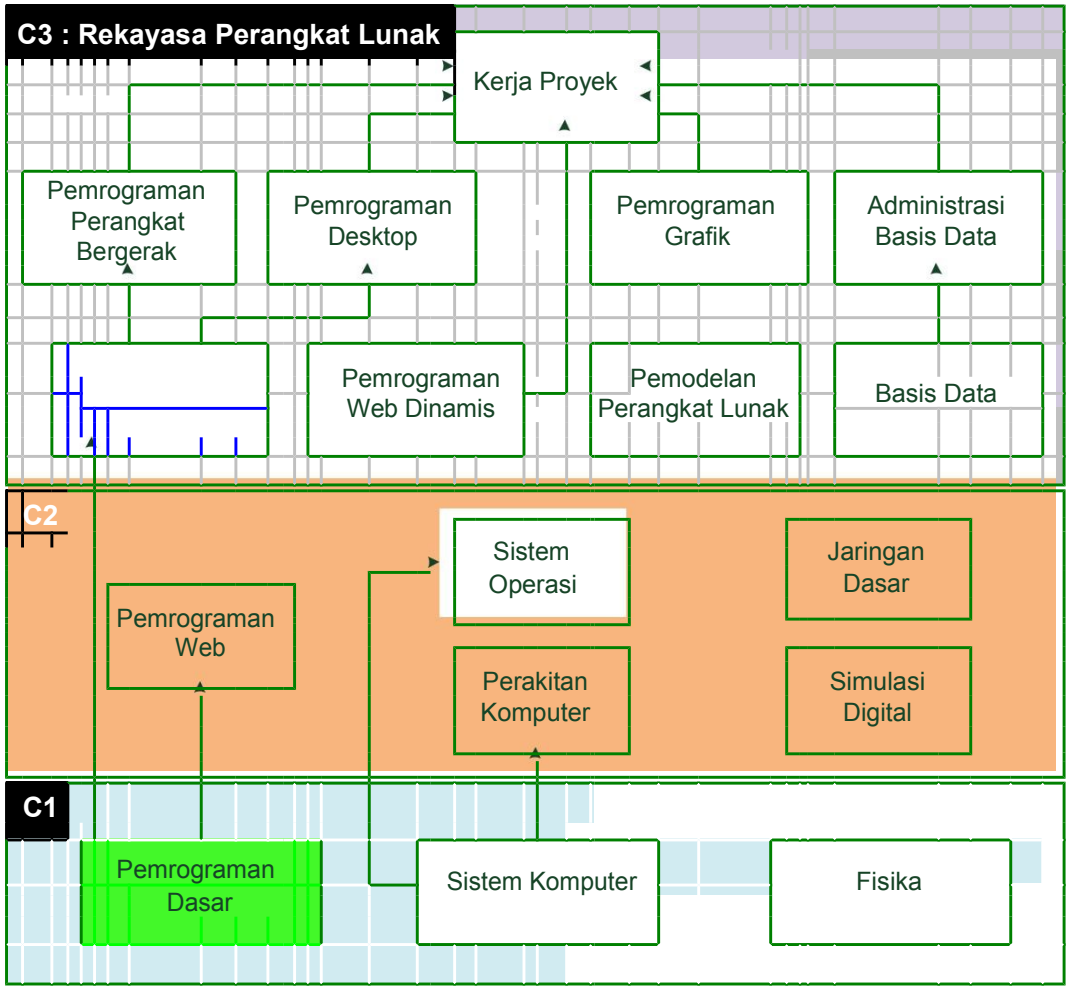
Queue merupakan model pengelompokan berdasarkan metode antrian suatu prioritas tertentu(contoh FIFO-First In First Out).

Set merupakan pengelompokan mengikuti model himpunan dimana setiap anggota-nya harus unik.

Throw digunakan untuk melemparkan suatu bug yang dibuat secara manual.

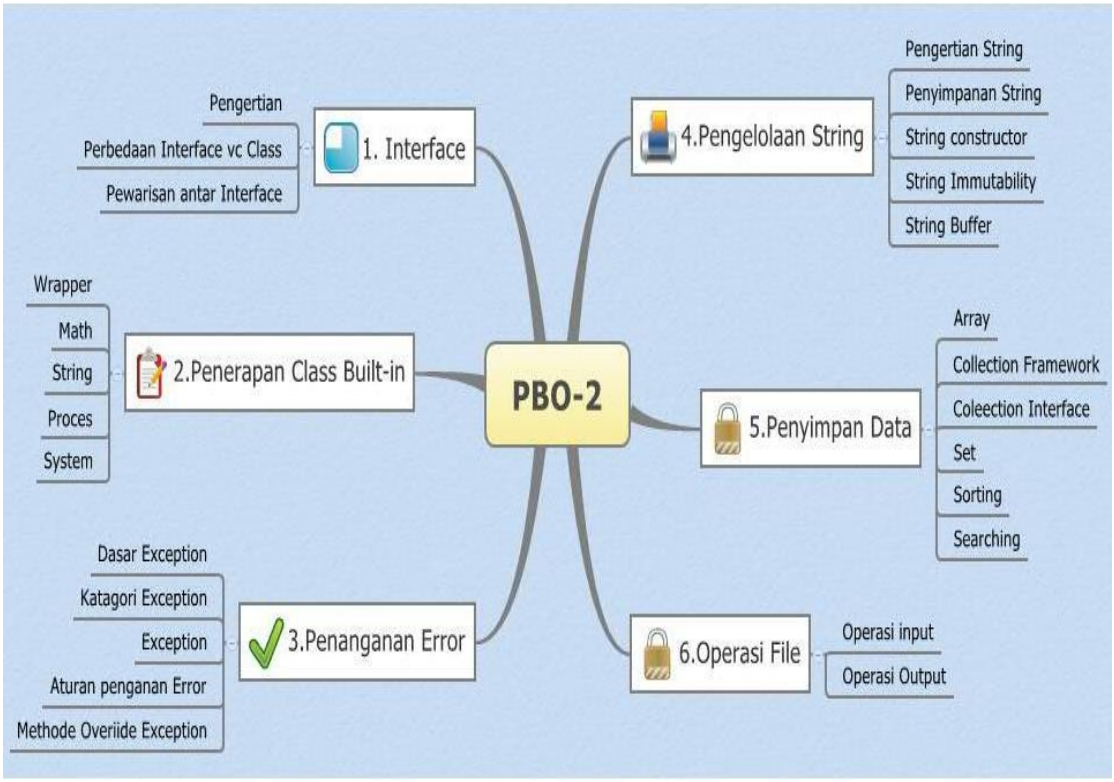
Trydigunakan dalam suatu blockprogram. Keyword ini digunakan untuk mencoba menjalankan blockprogram, kemudian mengenai dimana munculnya kesalahan yang ingin diproses.

PETA KEDUDUKAN BUKU



Keterangan	
C1	Kelompok mata pelajaran Dasar Bidang Keahlian Teknologi Informasi dan Komunikasi
C2	Kelompok mata pelajaran Dasar Program Keahlian Teknik Komputer dan Informatika
C3	Kelompok mata pelajaran Paket Keahlian Rekayasa Perangkat Lunak
	Mata pelajaran Pemrograman Berorientasi Obyek Semester 2
	Mata pelajaran prasyarat

Peta Konsep : Pemrograman Berorientasi Obyek Kelas XI Semester 2



Keterangan	
KD 3.8- 4.8	Interface
KD 3.9- 4.9	Penerapan Class Built-in
KD 3.10- 4.10	Penanganan Error
KD 3.11- 4.11	Pengelolaan String
KD 3.12- 4.12	Penyimpan Data
KD 3.13- 4.13	Operasi File

BAB I PENDAHULUAN

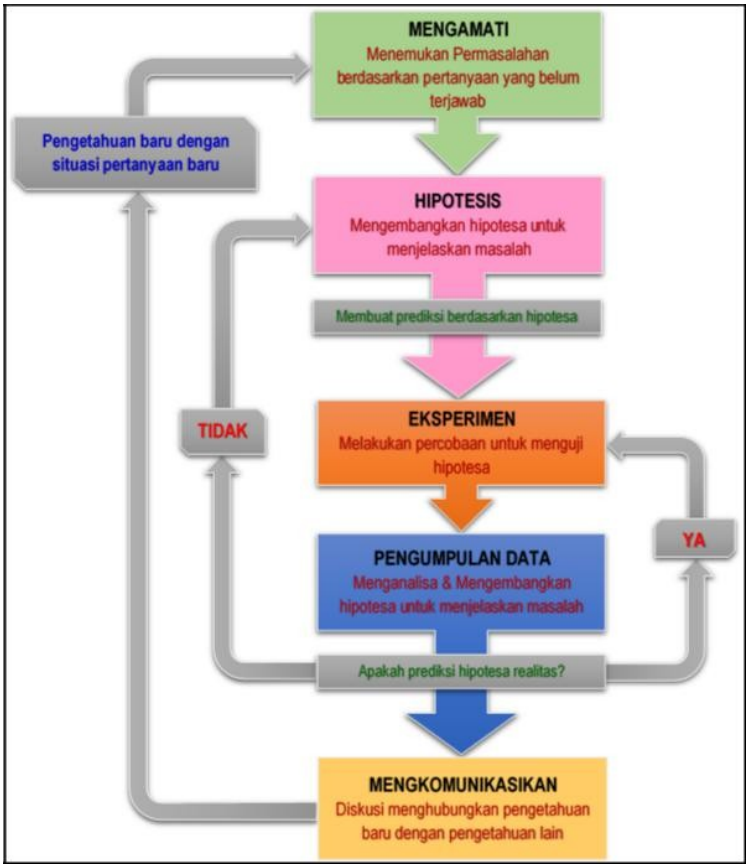
A. Deskripsi.

Pemrograman berorientasi objek (Inggris: *object-oriented programming* disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Ini adalah jenis pemrograman di mana programmer mendefinisikan tidak hanya tipe data dari sebuah struktur data, tetapi juga jenis operasi (fungsi) yang dapat diterapkan pada struktur data. Dengan cara ini, struktur data menjadi objek yang meliputi data dan fungsi. Selain itu, pemrogram dapat membuat hubungan antara satu benda dan lainnya. Sebagai contoh, objek dapat mewarisi karakteristik dari objek lain.

Salah satu keuntungan utama dari teknik pemrograman berorientasi obyek atas teknik pemrograman prosedural adalah bahwa memungkinkan programmer untuk membuat modul yang tidak perlu diubah ketika sebuah jenis baru objek ditambahkan.Seorang pemrogram hanya dapat membuat objek baru yang mewarisi banyak fitur dari objek yang sudah ada.Hal ini membuat program object-oriented lebih mudah untuk memodifikasi.

Pembelajaran pemrograman berorientasi obyek ini menggunakan metode pendekatan saintifik.Dalam pendekatan ini praktikum atau *e*ksperimen berbasis sains merupakan bidang pendekatan ilmiah dengan tujuan dan aturan khusus, dimana tujuan utamanya adalah untuk memberikan bekal ketrampilan yang kuat dengan disertai landasan teori yang realistis mengenai fenomena yang akan kita amati. Ketika suatu permasalahan yang hendak diamati memunculkan pertanyaan-pertanyaan yang tidak bisa terjawab, maka metode eksperimen ilmiah hendaknya dapat memberikan jawaban melalui proses yang logis. Proses-proses dalam pendekatan scientifik meliputi beberapa tahapan yaitu: mengamati, hipotesis atau menanya, mengasosiasikan atau eksperimen,

mengumpulkan atau analisa data dan mengkomunikasikan. Proses belajar pendekatan eksperimen pada hakekatnya merupakan proses berfikir ilmiah untuk membuktikan hipotesis dengan logika berfikir.



Gambar 1. Diagram Proses Metode Saintifik-Eksperimen Ilmiah

B. Prasyarat.

Untuk kelancaran pencapaian kompetensi dalam mata pelajaran pemrograman berorientasi obyek ini dibutuhkan beberapa peryaratan baik pengetahuan maupun ketrampilan dasar. Persyaratan tersebut antara lain ialah: Peserta didik telah menguasai mata pelajaran pemrograman dasar. Konsep dan algoritma pemrograman ini dibutuhkan untuk mendukung implementasi pemrograman berorientasi obyek.Disamping itu peserta didik mempunyai kompetensi dalam hal pemanfaatan teknologi informasi, seperti mengoperasikan hardware komputer dan mengoperasikan perangkat lunak aplikasi. Perangkat lunak aplikasi tersebut antar lain ialah pengolah data untuk menganalisis data

hasil eksperimen, pengolah kata untuk membuat laporan dan aplikasi presentasi untuk mengkomunikasikan dan mempresentasikan hasil laporan.

C. Petunjuk Penggunaan.

Buku pedoman siswa ini disusun berdasarkan kurikulum 2013 yang mempunyai ciri khas penggunaan metode scientifik. Buku ini terdiri dari dua bab yaitu bab satu pendahuluan dan bab dua pembelajaran. Dalam bab pendahuluan beberapa yang harus dipelajari peserta didik adalah diskripsi mata pelajaran yang berisi informasi umum, rasionalisasi dan penggunaan metode scientifik. Selanjutnya pengetahuan tentang persyaratan, tujuan yang diharapkan, kompetensi inti dan dasar yang akan dicapai serta test kemampuan awal.

Bab dua menuntun peserta didik untuk memahami deskripsi umum tentang topik yang akan dipelajari dan rincian kegiatan belajar sesuai dengan kompetensi dan tujuan yang akan dicapai. Setiap kegiatan belajar terdiri dari tujuan dan uraian materi topik pembelajaran, tugas serta test formatif. Uraian pembelajaran berisi tentang diskripsi pemahaman topik materi untuk memenuhi kompetensi pengetahuan.Uraian pembelajaran juga menjelaskan deskripsi unjuk kerja atau langkah-langkah logis untuk memenuhi kompetensi skill.

Tugas yang harus dikerjakan oleh peserta didik dapat berupa tugas praktek, eksperimen atau pendalaman materi pembelajaran. Setiap tugas yang dilakukan melalui beberapa tahapan saintifik yaitu : 1) melakukan pengamatan setiap tahapan unjuk kerja 2) melakukan praktek sesuai dengan unjuk kerja 3) mengumpulkan data yang dihasilkan setiap tahapan 4) menganalisa hasil data menggunakan analisa diskriptif 5) mengasosiasikan beberapa pengetahuan dalam uraian materi pembelajaran untuk membentuk suatu kesimpulan 6) mengkomunikasikan hasil dengan membuat laporan portofolio. Laporan tersebut merupakan tagihan yang akan dijadikan sebagai salah satu referensi penilaaian.

D. Tujuan Akhir.

Setelah mempelajari uraian materi dalam bab pembelajaran dan kegiatan belajar diharapkan peserta didik dapatmemiliki kompetensi sikap, pengetahuan dan ketrampilan yang berkaitan dengan materi:

Interface

Penerapan class built-in

Penanganan Error

Pengelolaan String

Penyimpanan Data

E. Kompetensi Inti Dan Kompetensi Dasar

Kompetensi Inti 1 : Menghayati dan mengamalkan ajaran agama yang dianutnya.

Kompetensi Dasar :

1.1. Memahami nilai-nilai keimanan dengan menyadari hubungan keteraturan dan kompleksitas alam dan jagad raya terhadap kebesaran Tuhan yang menciptakannya

1.2. Mendeskripsikan kebesaran Tuhan yang menciptakan berbagai sumber energi di alam

1.3. Mengamalkan nilai-nilai keimanan sesuai dengan ajaran agama dalam kehidupan sehari-hari.

Kompetensi Inti 2: Menghayati dan Mengamalkan perilaku jujur, disiplin, tanggung jawab, peduli (gotong royong, kerjasama, toleran, damai), santun, responsif dan proaktif dan menunjukan sikap sebagai bagian dari solusi atas berbagai permasalahan dalam berinteraksi secara efektif dengan lingkungan sosial dan alam serta dalam menempatkan diri sebagai cerminan bangsa dalam menempatkan diri sebagai cerminan bangsa dalam pergaulan dunia.

Kompetensi Dasar:

2.1. Menunjukkan perilaku ilmiah (memiliki rasa ingin tahu; objektif; jujur; teliti; cermat; tekun; hati-hati; bertanggung jawab; terbuka; kritis; kreatif; inovatif dan peduli lingkungan) dalam aktivitas sehari-hari sebagai wujud implementasi sikap dalam melakukan percobaan dan berdiskusi

2.2. Menghargai kerja individu dan kelompok dalam aktivitas sehari-hari sebagai wujud implementasi melaksanakan percobaan dan melaporkan hasil percobaan.

Kompetensi Inti 3: Memahami, menerapkan dan menganalisis pengetahuan faktual, konseptual dan prosedural berdasarkan rasa ingin tahunya tentang ilmu pengetahuan, teknologi, seni, budaya, dan humaniora dalam wawasan kemanusiaan, kebangsaan, kenegaraan, dan peradaban terkait penyebab fenomena dan kejadian dalam bidang kerja yang spesifik untuk memecahkan masalah.

Kompetensi Dasar:

3.8. Memahami pembuatan interface

3.9. Menganalisis pemanfaatan class built-in

3.10. Memahami mekanisme penanganan kesalahan

3.11. Memahami string dan berbagai propertinya

3.12. Memahami data collection sebagai media penyimpanan data.

3.13. Menerapkan operasi file dan Input Output(IO)

Kompetensi Inti 4: Mengolah, menalar, dan menyaji dalam ranah konkret dan ranah abstrak terkait dengan pengembangan dari yang dipelajarinya di sekolah secara mandiri, dan mampu melaksanakan tugas spesifik dibawah pengawasan langsung.

Kompetensi Dasar:

4.8Menyajikan hasil pembuatan aplikasi dengan interface

4.9Menyajikan beberapa class built-in dan penerapannya dalam memecahkan masalah

4.10 Mengolah penanganan error dalam mendeteksi kesalahan program

4.11 Mengolah data String dan berbagai propertinya

4.12 Menyajikan data colection sebagai penyimpan data

4.13 Menyajikan operasi file dan operasi Input Output

Cek Kemampuan Awal



Jelaskan perbedaan perbedaan interface dengan class !

Jelaskan fungsi dan manfaat class-class built-in!

Jelaskan secara singkat cara penanganan error!

Jelaskan cara pengelolaan data String !

Jelaskan secara singkat konsep konsep data collection sebagai penyimpan data !

Jelaskan secara singkat operasi file dalam class !

BAB II KEGIATAN BELAJAR

1. Kegiatan Belajar 1 : Interface

Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 1 ini siswa diharapkan dapat :

Memahami pembuatan interface

Menyajikan hasil pembuatan aplikasi dengan interface

Uraian Materi

Pengantar Interface

Di kelas kita sudah belajar apa interface tersebut, untuk mengulang materi di kelas, semoga postingan saya tentang interface kali ini bisa memperjelas tentang konsep interface yang kita pelajari di kelas praktikum dan penjelasan tugas yang sudah dikumpulkan minggu lalu.

Kenapa kita butuh interface? Sebagai pengantar kita harus mengetahui apa yang disebut interface dan kegunaannya dalam pemrograman java khususnya pemrograman berorientasi objek, karena kita akan bermain banyak dengan objek tersebut.

Interface adalah jenis khusus dari blok yang hanya berisi method signatureatau constant. Interface mendefinisikan sebuahsignature dari sebuah kumpulan method tanpa tubuh. Interface mendefinisikan sebuah cara standar dan umum dalam menetapkan sifat-sifat dari class-class. Mereka menyediakan class-classtanpa memperhatikan lokasinya dalam hirarki classuntuk mengimplementasikan sifat-sifat yang umum. Dengan catatan bahwa interface juga menunjukkan polimorfisme, dikarenakan program dapat memanggil method interface dan versi yang tepat dari method yang akan dieksekusi tergantung dari tipe object yang melewati pemanggil method interface.

Untuk lebih mudah memahami, interface merupakan sekumpulan dari method-method yang dibuat, namun belum ada operasi di dalam tubuh method tersebut.Interface bisa diturunkan atau diwariskan kepada class yang ingin

memakai method yang ada dalam masing-masing interface tersebut dengan keyword **extends** [interface yang didefinisikan]. Sebuah class dapat mengimplementasikan 1 interface yang sudah dibuat dengan keyword **implement.**

Ciri-ciri Interface

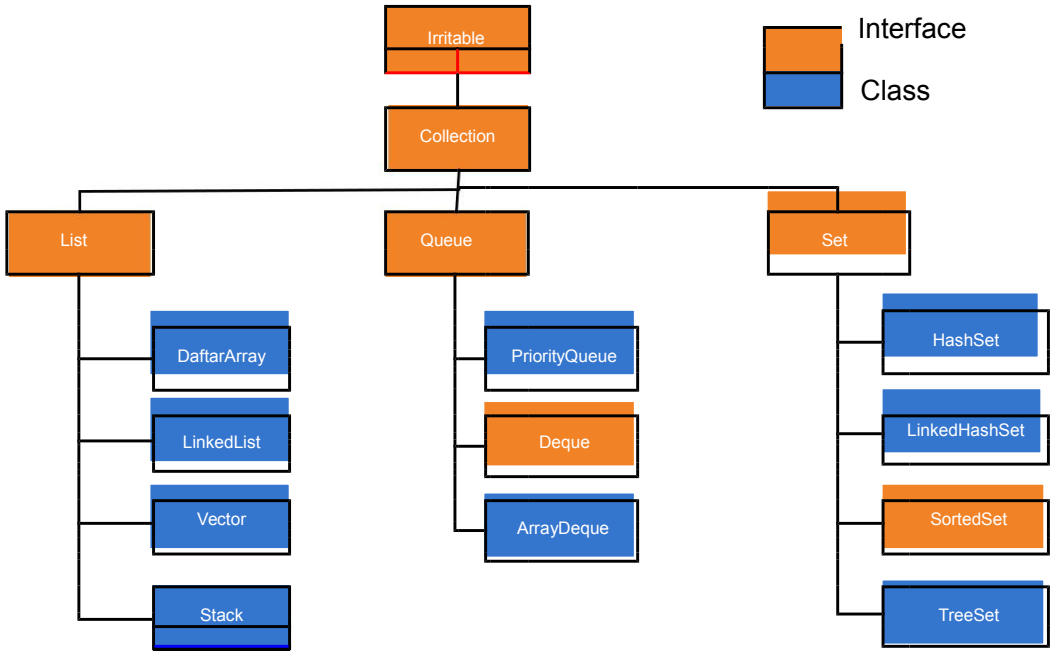
Ciri-ciri dari interface adalah sebagai berikut :

Method interface tidak punya tubuh, sebuah interface hanya dapat mendefinisikan konstanta dan interface tidak langsung mewariskan hubungan dengan class lainnya, mereka didefinisikan secara independent.

Tidak bisa membuat instance atau objek baru dari sebuah interface.

Ciri umum lain adalah baik interface maupun class dapat mendefinisikan method. Bagaimanapun, sebuah interface tidak memiliki kode implementasi sedangkan class memiliki salah satunya.

Perbedaan Interface dan Class



Pendeklarasian Interface

Contoh pendeklarasian interface adalah sebagai berikut :

Listing Program

```
1 interface Operasi
{
public void Penjumlahan();
public void Pengurangan ();
}
```

Implementasi Interface

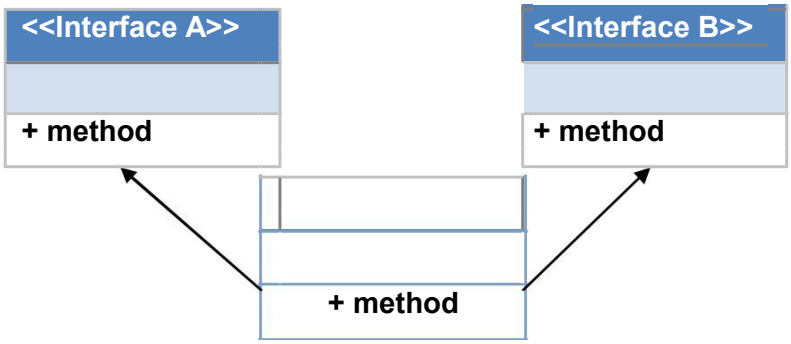
Cara menggunakan interface pada kelas lain, harus menggunakan keyword **implements.** Deklarasi implements interface sebagai berikut :

Listing Program

```
1 public class Calculator implements Operasi{
2 private double bil1, bil2;
3 …//override semua method pada interface
4 public static void main (String []args){
5 Calculator cal = new Calculator(3,4);
6 System.out.print ("Hasil penjumlahan "+cal.getbil1()
+" dan "cal.getbil2()"adalah ");
cal.Penjumlahan();
}
}
```

Multiple Interface

Java tidak memperkenankan adanya multiple inheritance, tetapi java memperbolehkan multiple interface.Dibawah ini adalah ilustrasi multiple interface.



2) Abstract Class

Abstract class adalah class yang mempunyai sedikitnya satu abstract method.Abstract class hanya bisa digunakan sebagai super class, dan dapat diturunkan dari class abstract lainnya.Untuk mendeklarasikan sebuah abstract class digunakan keyword abstract, **[abstract] class [class_name].**

Sebuah abstract class pada dasarnya tidak jauh berbeda dengan class lainnya, yakni juga berisi method yang menggambarkan karakteristik dari kelas abstract tersebut.Perbedaannya yaitu sebuah abstract class bisa berisi method tanpa diimplementasikan, artinya sebuah method tanpa body.Method seperti ini disebut method abstract.

Implementasi Abstract Class

Abstract class tidak bisa dibuat objectnya atau tidak dapat di instasiasi. Object hanya bisa dibuat dari non-abstract class (concrete class). Konsekuensinya suatu abstract class harus diturunkan dimana pada subclass tersebut berisi implementasi dari abstract method yang ada di superclass. Sintaks dalam membuat abstract class adalah sebagai berikut :

Listing Program

```
1 public abstract class Hewan
{
…//definisi class
}
```

Sintaks dalam membuat method abstract class adalah sebagai berikut :

Listing Program

```
1 public abstract class Hewan{
2 void Bernafas(){
System.out.println("Bernafas");
}
}
```

Abstract Method

Abstract method adalah method yang belum mempunyai implementasi.Kita dapat menyatakan suatu method abstract dengan membutuhkan keyword abstract pada deklarasi method tersebut.

Secara umum sintaks pendeklarasian abstract method adalah sebagai berikut :

Listing Program

```
1 abstract class Seniman{
2 public abstract void berkesenian();
3 public void tidur(){
System.out.println("Zzz…");
}
}
```

Listing Program

```
1 class Penyanyi extends Seniman{
2 public void berkesenian(){
System.out.println("Tralala-trilili…");
}
}
```

Listing Program

```
1 public class Explain{
2 public static void main(String args []){
3 Penyanyi Joshua = new Penyanyi();
Joshua.berkesenian();
}
}
```

Perbedaan Abstract Class dan Interface

Abstract Class	Interface
1. Bisa berisi abstract dan non-abstract method.	1. Hanya boleh berisi abstract method.
2. Kita harus menuliskan sendiri modifiernya.	2. Kita tidak perlu menulis public abstract di depan nama method. Karena secara implisit, modifier untuk method diinterfaceadalah *public* dan*abstract*.
3. Dapatmendeklarasikanconstantdan instance variable.	3. Hanyabisamendeklarasikanconstant. Secara implisit variable yang dideklarasikan di interface bersifatpublic, static dan final.
4. Method boleh bersifat static.	4. Method tidak boleh bersifat static.
5. Method boleh bersifat final.	5. Method tidak boleh bersifat final.
6. Suatu abstract class hanya bisa meng-*extend* satu abstract class lainnya.	6. Suatu interface bisa meng-*extend* satu atau lebih interface lainnya.
7. Suatu abstract class hanya bisa meng-*extend* satu abstract class dan meng-implement beberapa interface.	7. Suatu interface hanya bisa meng-*extend* interface lainnya. Dan tidak bisa meng-*implement* class atau interface lainnya.

Pewarisan antar Interface

Dalam OOP sering kali kita mendengar istilah pewarisan (Inheritance) , yaitu sebuah sub-class akan mewarisi behavior(method) ataupun attribut yang ada di dalam super-class nya .

Penggunaan inheritance dapat dilakukan secara overriding ataupun secara overloading method. Overloading berarti mendefinisikan beberapa metode yang memiliki nama sama tetapi dengan sidik yang berbeda. Sedangkan overriding berarti menyediakan suatu implementasi baru untuk suatu metode didalam subkelas.Ini menunjukan bahwa secara konsep super-class hanya menyediakan method-method yang belum terdefinisi secara explicit (jelas), sehingga sub-class dapat memanfaatkan method-method super-class tersebut sesuai kebutuhan di setiap sub-class .

Secara umum fungsi pewarisan dikatakan sebagai metode *reuseability* :

Behavior(method) dideklarasikan dalam superclass, behavior tersebut otomatis diwariskan ke seluruh subclass.

Kita dapat menggunakan kelas yang kita buat sebelumnya (superclass) dengan membuat kelas-kelas baru (subclass) berdasar super class tersebut, dengan karakteristik yang lebih khusus dari behaviour umum yang dimiliki super class.

Kita dapat membuat super class yang hanya mendefinisikan behaviour namun tidak memberi implementasi dari metode-metode yang ada (framework class) , superclass seperti ini disebut kelas abstrak (dengan modifier kelas dan method abstract) dan sub-classnya disebut kelas kongkret , sehingga sub-class dapat mengimplementasi method dari superclass sesuai dengan kebutuhan di sub-classnya tanpa mempengaruhi super-classnya.

Kita dapat mendefinisi method hanya sekali dan method tersebut dapat digunakan oleh seluruh subclass.

Sebuah subclass hanya perlu mengimplementasikan perbedaan antara dirinya sendiri dan parent-nya (super-classnya).

Contoh pewarisan

Listing Program

```
// yang dijadikan super-class
public abstract class Burung{ 3//
method abstract
public abstract void suara();
5// method non-abstract yang akan dioverride
public void bisaTerbang(){
System.out.println("bisaTerbang donk!!");
}
}
```

Listing Program

```
// interface Pernafasan
interface Pernafasan {
// method yang akan di implementasikan
void bernafasLewat();
}
```

Listing Program

```
q//sub-class pertama
2public class bebek extends Burung implements Pernafasan{
3 //meng-override method bisaTerbang() dari kelas Burung
@Override
public void bisaTerbang() {
System.out.println("ups , cuma berjalan !");
}
// meng-implements method suara() dari kelas burung
public void suara() {
System.out.println("kowek-kowek");
}
12//mengimplements method bernafasLewatdari
interfacePernafasan
13 public void bernafasLewat() {
System.out.println("lewat paru-paru");
}
}
```

Listing Program

```
//sub-class kedua
public class perkutut extends burung implements
Pernafasan{
// meng-implements method suara() dari kelas burung
public void suara() {
System.out.println("kuruk-kuruk");
}
7//mengimplements method bernafasLewat dari interface
Pernafasan
8public void bernafasLewat()
{ 9System.out.println("bernafas Lewat paru-paru
juga");
}
}
```

Pada pemrograman Java, sebuah inheritansi yang ditangani dengan metode abstraksi ini disebut single inheritance.Apabila kita ingin membuat pewarisan banyak (multiple inheritance) kita dapat menggunakan metode antar-muka (interface) .

Dalam kode diatas dapat ditemukan sebuah interface ‘Pernafasan’ . Perlu diingat bahwa interface bukan sebuah kelas untuk membuat interface kita menggunakan modifier kelas Interface.

C. Rangkuman

Interface merupakan kumpulan dari method-method yang belum terdapat operasi di dalam tubuh method tersebut.Interface bisa diturunkan atau diwariskan kepada class yang ingin memakai method yang ada dalam masing-masing interface tersebut dengan keyword **extends** [interface yang didefinisikan]. Sebuah class dapat mengimplementasikan 1 interface yang sudah dibuat dengan keyword **implement.**Interface dapat mendefinisikan konstanta. Interface juga tidak dapat membuat instance atau objek baru dari sebuah interface.Baik interface maupun class dapat mendefinisikan method.

Abstract class adalah class yang mempunyai sedikitnya satu abstract method.Abstract class hanya digunakan sebagai super class dan dapat diturunkan dari class abstract lainnya.Pendeklarasian abstract class dengan menggunakan keyword **abstract class.**Abstract class berisi method tanpa diimplementasikan yang disebut dengan method abstract.

Dalam interface terdapat istilah pewarisan (inheritance) yaitu pewarisan method dan atribut dari super-class kepada sub-classnya. Penggunaan inheritance dapat dilakukan secara overriding dan overloading method.

D. Tugas

Tugas 1

Buatlah program berikut :

'Class Bentuk' akan digunakan sebagai abstract class untuk class turunannya yaitu 'Class Lingkaran', 'Class Segitiga', dan 'Class SegiEmpat'

❖ Mengamati Listing Program dan Output Program

Tentukan nama class yang akan digunakan.

Tentukan variabel yang akan digunakan.

Tentukan method yang akan digunakan.

Buatlah class diagram yang menggambarkan bagian-bagian dari class yang telah ditentukan. Contoh class diagram :

Nama class
Method
Operasi

Buatlah listing programnya.

Lakukan kompilasi dan debug pada program.

No	Output Program
q1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil program yang telah kamu buat dengan hasil program teman sebangku atau kelompok lain

Dari hasil perbandingan tersebut, hal penting apa yang harus dirumuskan secara bersama

E. Tes Formatif

Dalam test ini setiap anda harus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Sebutkan definisi dari beberapa istilah berikut :

Interface

Abstract class

Inheritance

Method abstract

Jelaskan perbedaan overloading method dan overriding method!

Identifikasilah kesalahan yang terdapat pada program berikut !

Listing Program

```
1 ArrayList list new ArrayList();
2 list.add ("Mataram");
3 list.add ("Yogyakarta");
4 list.add (new java.util.Date());
5 String kota list.get(0);
6 list.get (3, "Siantar");
7 System.out.println(list.get(3));
```

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Definisi dari istilah :



a)

Interface.....

.....

.....

.....

.....

b) Abstract Class

….....

.....

.....

.....

Inheritance

.....

.....

.....

.....

Method Abstract

……….....

.....

.....

.....

LJ- 02 : Perbedaan overloading method dan overriding method :



.....

.....

.....

.....

.....

.....

.....

LJ- 03 : Kesalahan pada program :



.....

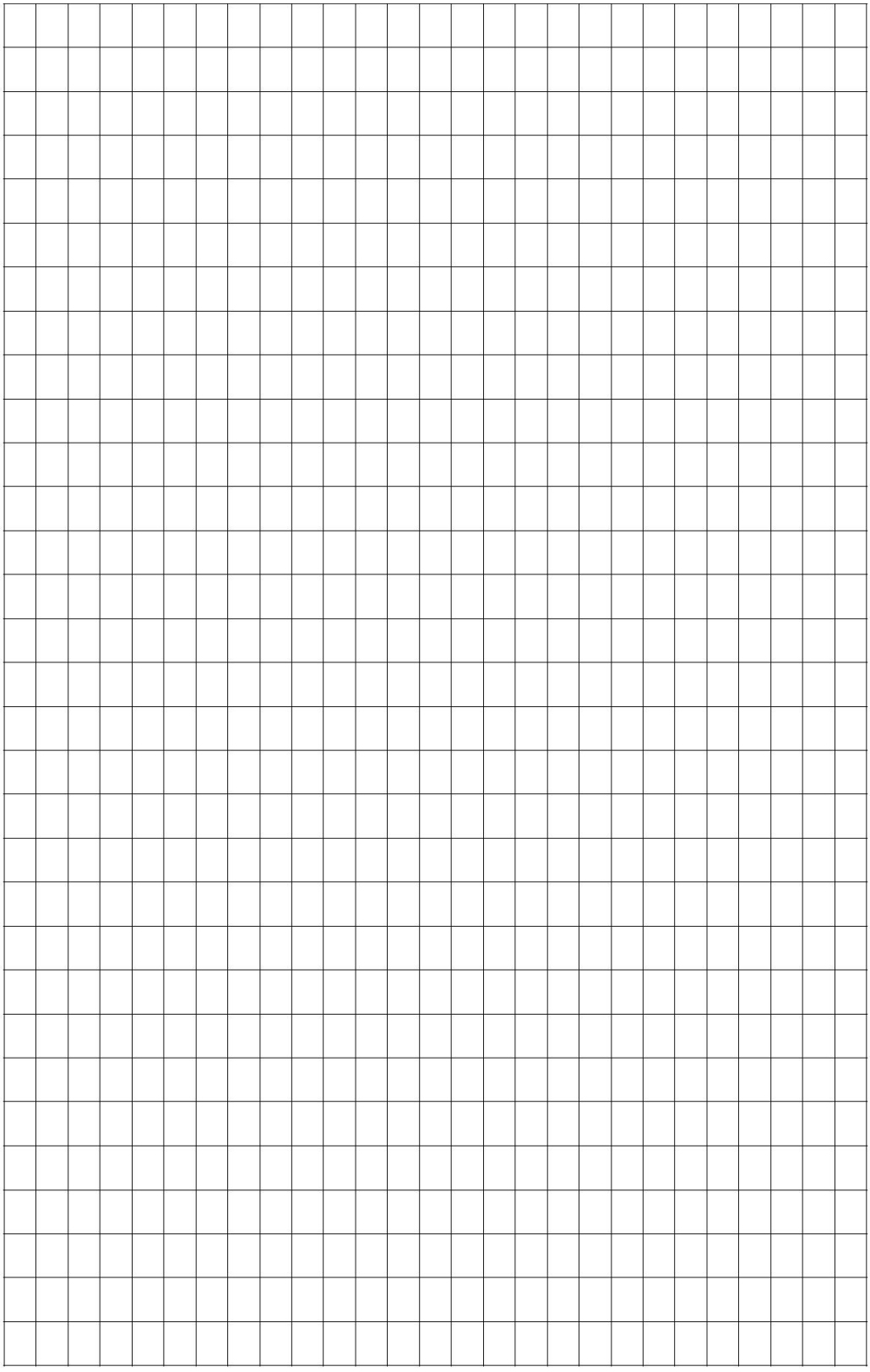
.....

.....

.....

.....

G. Lembar Kerja Siswa



2. Kegiatan 2 :Class Built-in

A. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 2 ini siswa diharapkan dapat :

Menganalisis pemanfaatan class built in.

Menyajikan beberapa class – class built in dan penerapanya dalam memecahkan masalah.

Uraian Materi

Class Math

Javajugamenyediakankonstantadanmethoduntukmenunjukkanperbedaanoperasi matematikasepertifungsitrigonometridanlogaritma.Selamamethod-methodini semuastatic, kita dapatmenggunakannyatanpamemerlukansebuahobjekMath.Untukmelengkapidaft arkonstantadanmethod-methodini, lihatlahacuanpada dokumentasiJavaAPI.Dibawahinibeberapamethod-methodumumyangsering digunakan.

Public Static double abs (double a)

Menghasilkan nilai mutlak.Sebuah method yang dioverload.Dapatjuga menggunakannilaifloatatauintegerataujugalongintegersebagaiparameter, dengankondisitipekembaliannyajugamenggunakanfloatatauintegerataulong.

Public Static double random()

Method ini digunakan untuk membangkitkan suatu nilai double acak dengan rentang lebih besar atau sama dengan nol (0) dan lebih rendah dari 1 **(0 <=Math.random() < 1.0)**. Method ini digunakan untuk menuliskan suatu ekspresi sederhana untuk membangkitkan angka-angka acak dengan sembarang rentang.

Public Static double max (double a, double b)

Menghasilkannilaimaksimum, diantaraduanilaidouble, adan b. Sebuahmethodyangdi-overload.Dapatjugamenggunakannilaifloatatauintegerataujugalong

integersebagaiparameter,dengankondisitipekembalinyajugamenggunakanfloatata uintegerataulonginteger,secaraberturut-turut.

Public Static double min (double a, double b)

Menghasilkannilaiminimumdiantaraduanilaidouble,a dan b.Sebuahmethodyangdi-overload.Dapatjugamenggunakannilaifloatatauintegerataujugalonginteger sebagaiparameter,dengankondisitipekembaliannyajugamenggunakanfloatatauint egerataulonginteger,secaraberturut-turut.

Public Static double ceil (double a)

Menghasilkanbilanganbulatterkecilyanglebihbesaratausamadengana.Mengembal ikan terkecil (paling dekat dengan infinity negatif) nilai ganda yang lebih besar dari atau sama dengan argumen dan sama dengan bilangan bulat matematika. Kasuskhusus : Jika nilai argumen sudah sama dengan bilangan bulat matematika, maka hasilnya adalah sama dengan argumen. Jika argumen adalah NaN atau tak terhingga atau positif nol atau negatif nol, maka hasilnya adalah sama dengan argumen. Jika nilai argumen kurang dari nol tetapi lebih besar dari -1.0, maka hasilnya adalah nol negatif. Perhatikan bahwa nilai Math.ceil (x) adalah persis nilai-Math. Floor(-x).

Parameter: a-nilai.

Pengembalian: yang terkecil (paling dekat dengan infinity negatif) nilai floating-point yang lebih besar dari atau sama dengan argumen dan sama dengan bilangan bulat matematika.

Public Static double floor (double a)

Menghasilkanbilanganbulatterbesaryanglebihkecilatausamadengana.Mengembal ikan terbesar (paling dekat dengan infinity positif) nilai ganda yang kurang dari atau sama dengan argumen dan sama dengan bilangan bulat matematika. Kasus khusus : Jika nilai argumen sudah sama dengan bilangan bulat matematika, maka hasilnya adalah sama dengan argumen. Jika argumen adalah NaN atau tak terhingga atau positif nol atau negatif nol, maka hasilnya adalah sama dengan argumen.

Parameter : a- nilai.

Pengembalian: yang terbesar (paling dekat dengan infinity positif) nilai floating-point yang kurang dari atau sama dengan argumen dan sama dengan bilangan bulat matematika.

Public Static double exp (double a) MenghasilkanangkaEuler,epangkata. Pengembalian Euler nomor e pangkat dari nilai ganda.

Kasus khusus:Jika argumen adalah NaN, hasilnya adalah NaN.Jika argumen adalah infinity positif, maka hasilnya adalah tak terhingga positif. Jika argumen adalah infinity negatif, maka hasilnya adalah nol positif. Hasilnya dihitung harus berada dalam 1 ulp hasil yang tepat. Hasil harus semi-monoton.

Parameter: a - eksponen untuk meningkatkan e untuk.

Pengembalian : ea nilai, dimana e adalah basis dari logaritma natural.

Public Static double log (double a)

Menghasilkanlogaritmanaturaldaria.Mengembalikan logaritma natural (base e) dari nilai ganda.

Kasuskhusus : Jika argumen adalah NaN atau kurang dari nol, maka hasilnya adalah NaN. Jika argumen adalah infinity positif, maka hasilnya adalah tak terhingga positif. Jika argumen positif nol atau negatif nol, maka hasilnya adalah tak terhingga negatif. Hasilnya dihitung harus berada dalam 1 ulp hasil yang tepat. Hasil harus semi-monoton.

Parameter : a – nilai.

Pengembalian: nilai ln a, logaritma natural dari.

Public Static double pow (double a, double b) Menghasilkanapangkatb. Mengembalikan nilai argumen pertama pangkat dari argumen kedua .

Kasus khusus : Jika argumen kedua adalah positif atau negatif nol, maka hasilnya adalah 1,0. Jika argumen kedua adalah 1,0, maka hasilnya adalah sama sebagai argumen pertama. Jika argumen kedua adalah NaN, maka hasilnya adalah NaN.Jika argumen pertama adalah NaN dan argumen kedua

adalah nol, maka hasilnya adalah NaN.(Dalam keterangan di atas, nilai floating - point dianggap integer jika dan hanya jika itu adalah terbatas dan titik tetap dari metode ceil atausama titik tetap lantai metode. Nilai A adalah titik tetap dari metode satu - argumen jika dan hanya jika hasil dari penerapan metode untuknilai sama dengan nilai). Hasilnya dihitung harus berada dalam 1 ulp hasil yang tepat .Hasil harus semi- monoton.

Parameter: a - dasar . b - eksponen .

Pengembalian :nilai ab .

Public Static long round (double a)

Menghasilkanpembulatankeataskelongterdekat.Sebuahmethodyangdi-overload.Dapatjugamenggunakanfloatpadaargumentdanakanmenghasilkanpembulatankeataskeintterdekat.

Public Static double sqrt (double a)

Menghasilkanakarkuadrata.Mengembalikanbulatbenarakar kuadratpositifdarinilai ganda.Kasus khusus:Jika argumentadalahNaNataukurang darinol, maka hasilnya adalahNaN.Jika argument adalahinfinitypositif, maka hasilnyaadalah tak terhinggapositif.Jika argumentpositifnolatau negativenol, maka hasilnya adalahsama denganargumen.Jika tidak, hasilnya adalahnilaigandayang paling dekat denganakar kuadratmatematikayang benardari nilaiargumen. Parameter :a-nilai.

Pengembalian:akar kuadratpositifdari nilai argumen. Jika argumentadalahNaNataukurang darinol, hasilnya adalahNaN.

Public Static double sin (double a) Menghasilkansinussudutadalamradian dan mengembalikan sinus trigonometri dari sudut.Kasus khusus:Jika argumen adalah NaN atau tak terhingga, maka hasilnya adalah NaN. Jika argumen adalah nol, maka hasilnya adalah nol dengan tanda yang sama sebagai argumen. Hasilnya dihitung harus berada dalam 1 penghitungan hasil yang tepat. Hasil harus semi-monoton.

Parameter : a - sudut, dalam radian.

Pengembalian : sinus dari argumen.

qPublic Static double toDegrees (double angrad)

Menghasilkannilaiderajatyangkira-kirasetaradengannilairadianyangdiberikan.Mengubah sudut diukur dalam radian ke sudut kurang lebih setara diukur dalam derajat. Konversi dari radian ke derajat umumnya eksak, pengguna tidak harus mengharapkan cos (toRadians (90,0)) untuk persis sama 0,0.

Parameter: angrad - sudut, dalam radian.

Pengembalian: pengukuran sudut angrad dalam derajat.

Public Static double toRadians (double angdeg)

Menghasilkannilairadianyangkira-kirasetaradengannilaiderajatyangdiberikan. Dan mengubahsudutdiukur dalam derajatke sudutkurang lebih setaradiukur dalamradian. Konversi dariderajat keradianumumnyaeksak. Parameter:angdeg-sudut, dalam derajat. Pengembalian:pengukuranangdegsudutdalam radian.

Class String

ClassStringdisediakanolehJavaSDKdenganmenggunakankombinasicharacterliterals.Tidaksepertibahasapemrogramanlainnya,sepertiCatauC++,stringsdapat digunakanmenggunakanarraydaricharacterataudisederhanakandenganmenggunakanclassString.Sebagaicatatan,bahwasebuahobjekStringberbedadari sebuaharraydaricharacter.

Method-Method String

Public charAt (intidex)

Mengirimkarakterdiindeksyangditentukanolehparameterindex.Mengembalikannilaiarang padaindeks tertentu.Sebuah indeksberkisar dari 0 sampaipanjang1.NilaiCharpertamaurutanberada padaindeks0, berikutnya pada indeks1, danseterusnya,seperti untukpengindeksan array.Jika nilaiarangyang ditentukan olehindeksadalahpengganti, nilai penggantidikembalikan.Ditentukanoleh:charAtdalam antarmukaCharSequence.

Parameter:Indeks-indeksdari nilaiarang.

Pengembalian :nilaiarang padaindeks tertentudaristring ini.NilaiCharpertamaadalahpada

indeks0.Melempar:IndexOutOfBoundsException–jikaargumentindeksnegatif atautidak kurang daripanjang stringini.

Public intCompare To (String another String)

MembandingkanduaStringdanmengirimbilanganintyangmenentukanapakahobjek stringpemanggilkurangdariatausamadengananotherString.Bernilainegatifjika objekyangdilewatkan(passedstring)lebihbesar,0jikakeduastringsama, dan bernilaipositifjikaobjekstringpemanggil(callingstring)lebihbesar.Jika tidak ada indeks posisi di mana mereka berbeda, maka string yang lebih pendek leksikografi mendahului lagi tali. Dalam hal ini, compareTo mengembalikan perbedaan panjang dari string - yaitu, nilai :

this.length ( ) - anotherString.length ( )

Ditentukan oleh : compareTo dalam antarmuka Sebanding <String>.

Parameter : anotherString - String yang akan dibandingkan .

Pengembalian : nilai 0 jika argumen string sama dengan string ini, nilai kurang dari 0 jika string ini adalah leksikografis kurang dari argumen string, dan nilai lebih besar dari 0 jika string ini adalah leksikografi lebih besar dari argumen string.

Public intCompare To Ignore Case (String str)

SerupadengancompareTotetapicaseinsensitivity.Membandingkan dua string leksikografi, mengabaikan perbedaan kasus. Metode ini mengembalikan sebuah integer yang tanda adalah bahwa memanggil compareTo dengan versi normal dari string di mana perbedaan kasus telah dieliminasi dengan memanggil Character.toLowerCase (Character.toUpperCase (karakter)) pada masing-masing karakter. Perhatikan bahwa metode ini tidak mengambil lokal ke account, dan akan menghasilkan pemesanan memuaskan untuk lokal tertentu. Paket java.text menyediakan collators untuk memungkinkan pemesanan lokal-sensitif.

Parameter : str - String yang akan dibandingkan.

Pengembalian : bilangan bulat negatif, nol, atau bilangan bulat positif sebagai String yang ditentukan lebih besar dari, sama dengan, atau kurang dari String ini, mengabaikan pertimbangan kasus.

qPublic Boolean Equals (Object an Object)

Menghasilkannilaitruejikaparametertunggalnyatersusundarikarakteryangsamaden gan objek tempat Anda memanggil equals. Sedangkan jika parameter yangditentukan bukan sebuah objek String atau jika tidak cocok dengan urutan simbol padastring, method akan dikembalikan dengan nilai false.

Public Boolean Equals Ignore Case (String another String)

Serupa dengan equals tetapi case insensitivity.Membandingkan String ini untuk String lain, mengabaikan pertimbangan kasus. Dua string dianggap sama mengabaikan kasus jika mereka adalah sama panjang dan sesuai karakter dalam dua string yang sama kasus mengabaikan. Dua karakter c1 dan c2 dianggap mengabaikan kasus yang sama jika setidaknya salah satu dari berikut ini benar : Dua karakter yang sama (dibandingkan dengan operator ==) Menerapkan metode Character.toUpperCase (char) untuk masing-masing karakter menghasilkan hasil yang sama menerapkan metode Character.toLowerCase (char) untuk masing-masing karakter menghasilkan hasil yang sama.

Parameter : anotherString - String untuk membandingkan String ini terhadap Pengembalian : true jika argumen tidak null dan merupakan String mengabaikan kasus setara; false jika tidak.

Public Void getChars (int srcBegin, int srcEnd,char[ ] dst, int dst Begin)

Mendapatkan characters dari string yang dimulai pada index srcBegin hingga indexsrcEnd dan menyalin karakter-karakter tersebut pada array dst dimulai pada indexdstBegin.

Public intlength()

Menghasilkan panjang String. Public String replace(char oldChar, char newChar). Mengganti karakter, semua yang kemunculan oldChar diganti newChar.

Public String Substring(int beginIndex, int endIndex)

Mengirim substring dimulai dari indeks beginIndex yang ditentukandan berakhirdengan indeks endIndex yang ditentukan. Parameter : beginIndex - indeks awal, inklusif.

Pengembalian : substring yang ditentukan.

Melempar : IndexOutOfBoundsException - jika beginIndex negatif atau lebih besar dari panjang objek String ini.

Public char[] toCharArray()

Mengembalikan array karakteryang sama dengan string ini.

Public StringTrim()

Menghilangkan whitespacedi awal dan akhir objek String.Mengembalikan salinan dari string , dengan terkemuka dan trailing spasi dihilangkan. Jika objek String ini merupakan urutan karakter kosong, atau yang pertama dan terakhir karakter dari urutan karakter diwakili oleh objek String ini keduanya memiliki kode lebih besar dari ' \ u0020 ' (karakter spasi), maka referensi ke objek String ini dikembalikan.Jika tidak, jika tidak ada karakter dengan kode yang lebih besar dari ' \ u0020 ' dalam string, maka objek String baru yang mewakili string kosong dibuat dan dikembalikan. Jika tidak, biarkan k menjadi indeks dari karakter pertama dalam string yang lebih besar dari kode ' \ u0020 ', dan biarkan m menjadi indeks dari karakter terakhir dalam string yang lebih besar dari kode ' \ u0020 '. Sebuah objek String baru dibuat , mewakili substring dari string ini yang diawali dengan karakter pada indeks k dan berakhir dengan karakter pada indeks m- yaitu, hasil dari this.Substring (k , m +1).Metode ini dapat digunakan untuk memangkas spasi (seperti dijelaskan di atas) dari awal dan akhir string. Pengembalian :Salinan string ini dengan leading dan trailing spasi dihapus, atau string ini jika tidak memiliki terkemuka atau tertinggal spasi.

Public Static StringValueOf(-)

Dapat menggunakan tipe data sederhana seperti boolean, integer ataucharacter, ataujuga menggunakan sebuah objek sebagai parameter. Mengirim objek String yangmerepresentasikan tipe tertentu yang dilewatkan sebagai parameter.

Class StringBuilder/StringBuffer

KetikaobjekStringdiciptakan,objekStringtidakbisalagidimodifikasi.ObjekStringBuf ferserupadenganobjekString, kecualikenyataanbahwaobjekStringBuffer bersifatdapatberubahataudapatdimodifikasi,sedangkanpadaobjectStringbersifat konstan. Dapat dikatakan bahwa class StringBuffer ini lebih fleksibel disbanding class String. Panjangdanisidapatdiubahhinggabeberapapemanggilanmethod. Class StringBuilder sama dengan class StringBuffer kecuali bahwa metode metode untuk memodifikasi buffer didalam StringBuffer telah disinkronisasi. Class StringBuilder memiliki 3 konstruktor dan lebih dari 30 metode. StringBuffer dapat digunakan jika diakses oleh beberapa perkerjaan secara bersamaan.Sedangkan StringBuilder digunakan jika diakses oleh satu pekerjaan saja.

Class Wrapper

Sesungguhnya, tipedataprimitivesepertiint, charandlongbukanlahsebuahobjek.Sehingga, variabel-variabeltipedatainitidakdapatmengaksesmethod-methoddari classObject.Hanyaobjek-objeknyata,yangdideklarasikanmenjadireferensitipedata, dapatmengaksesmethod-methoddariclassObject.Adasuatukeadaan,bagaimanapun,ketikaAndamembutuh kansebuahrepresentasiobjekuntukvariabel-variabeltipe primitivedalamrangkamenggunakanmethod-methodJavabuilt-in.Sebagaicontoh, Andabolehmenambahkanvariabletipeprimitif padaobjekCollection.Disinilahclasswrappermasuk.Classwrapperadalahrepresen tasiobjeksederhanadarivariabel- variablenon-objekyangsederhana. Ada 10 tipe data Wrapper, yaitu Boolean, Byte, Character, Double, Float, Integer, Long, Number, Short, dan Void.

Perlu diperhatikan bahwa tipe data wrapper dan tipe data dasar (boolean, byte, char, double, float, int, long, short, void) tidak saling menggantikan. Tipe data dasar dilewatkan ke method dengan *pass by value*, jadi jika membutuhkan *pass by reference* harus memanfaatkan kelas tipe data wrapper. Kelas ini menyediakan versi objek dari tipe data dasar, maka dimungkinkan menambah

method-method untuk masing-masing tipe.

5) Class Process

The ProcessBuilder.start ( ) dan metode Runtime.exec membuat proses asli dan mengembalikan sebuah instance dari subclass Proses yang dapat digunakan untuk mengontrol proses dan mendapatkan informasi tentang hal itu . Proses kelas menyediakan metode untuk melakukan input dari proses, melakukan output ke proses, menunggu proses untuk menyelesaikan , memeriksa status keluar dari proses, dan menghancurkan (membunuh) proses.Metode yang menciptakan proses mungkin tidak bekerja dengan baik untuk proses khusus pada platform asli tertentu, seperti proses asli windowing, proses daemon, proses Win16/DOS pada Microsoft Windows, atau script shell .Secara default, yang subproses dibuat tidak memiliki terminal atau console sendiri. Yang saya standar / O (yaitu stdin, stdout, stderr) operasi akan diarahkan ke proses induk, di mana mereka dapat diakses melalui sungai diperoleh dengan menggunakan metode getOutputStream ( ), getInputStream ( ), dan getErrorStream ( ). Proses induk menggunakan sungai-sungai ini untuk memberi makan masukan ke dan mendapatkan output dari subproses tersebut. Karena beberapa platform asli hanya menyediakan ukuran buffer terbatas untuk standar input dan output stream, kegagalan untuk segera menulis input stream atau membaca output stream dari subproses dapat menyebabkan subproses untuk memblokir, atau bahkan kebuntuan.Dimana diinginkan, subproses I / O juga dapat diarahkan menggunakan metodedari kelas ProcessBuilder.Subproses tidak tewas ketika tidak ada lagi referensi ke objek Proses, melainkan subproses terus melaksanakan asynchronously.Tidak ada persyaratan bahwa proses diwakili oleh objek Proses mengeksekusi asynchronous atau bersamaan sehubungan dengan proses Java yang memiliki objek Proses .

6) Method Process

Public abstract InputStream getInputStream()

Mengembalikaninput streamterhubung ke outputnormalsubprosestersebut.Arusmemperoleh datapipa darioutput standardari prosesdiwakili olehobjekProsesini.Jikaoutputstandarsubprosestelahdiarahkanmenggunakan ProcessBuilder.redirectOutputmaka metode iniakanmengembalikanaliranmasukannull.Jika tidak, jikastandard errorsubprosestelahdiarahkanmenggunakanProcessBuilder.redirectErrorStream makainput streamdikembalikan olehmetode iniakan menerimaoutput standardigabungdanstandard errordarisubprosestersebut.CatatanPelaksanaan:Iniadalah ide yang baikuntukinput streamkembali kebuffered.Pengembalian:input streamterhubung ke outputnormalsubprosesyang

Public abstract InputStream getErrorStream()

Mengembalikaninput streamterhubung ke outputkesalahansubprosestersebut.Arusmemperoleh datadisalurkandarikesalahan outputdari prosesdiwakili olehobjekProsesini.Jika kesalahanstandarsubprosestelahdiarahkanmenggunakanProcessBuilder.redirectErroratauProcessBuilder.redirectErrorStreammaka metode iniakanmengembalikanaliranmasukannull.CatatanPelaksanaan:Iniadalah ide yang baikuntukinput streamkembali kebuffered.Pengembalian:input streamterhubung ke outputkesalahanyangsubprocess.

Class System

ClassSystemmenyediakanberapafielddanmethodbermanfaat, sepertistandard input,standardoutputdansebuahmethodyangbergunauntukmempercepatpenyalinanbagiansebuaharray.Dibawahinibeberapamethodmenarikdariclass System.Sebagaicatatan,bahwasemuamethod-methodclassadalahstatic.

Rangkuman

Ada beberapa class built-in didalam pemrograman java.Yang pertama yaitu class math.Class math dapat berisi method untuk menunjukkan perbedaan operasi matematika seperti fungsi trigonometri dan logaritma.Selama method-

method ini semua static, kita dapat menggunakannya tanpa memerlukan sebuah objek Math.

Yang kedua adalah Class String. Dalam Java, string dapat digunakan menggunakan array dari character atau disederhanakan dengan menggunakan class String.Yang ketiga Class StringBuffer, class ini serupa dengan objek string, kecuali kenyataan bahwa objek String Buffer bersifat dapat berubah atau dapat dimodifikasi, sedangkan pada object String bersifat konstan.
Yang keempat yaitu Class Wrapper, merupakan representasi objek sederhana dari variabel- variable non-objek yang sederhana.

Dan yang kelima yaitu Class Proccess, class ini menyediakan metode untuk melakukan input dari proses, melakukan output ke proses, menunggu proses untuk menyelesaikan, memeriksa status keluar dari proses, dan menghancurkan (membunuh) proses.

D. Tugas

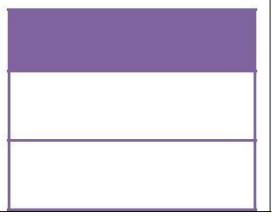
Tugas 1

Buatlah method untuk mengurutkan String menggunakan header sebagai berikut :

```
Public static String urut (String s)
```

❖ Mengamati Listing Program dan Output Program

Tentukan nama class yang akan digunakan.
Tentukan variabel yang akan digunakan.
Tentukan method yang akan digunakan.
Buatlah class diagram yang menggambarkan bagian-bagian dari class yang telah ditentukan. Contoh class diagram :



5. Buatlah listing programnya.
6. Lakukan kompilasi dan debug pada program.

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil program yang telah kamu buat dengan hasil program teman sebangku atau kelompok lain

Dari hasil perbandingan tersebut, hal penting apa yang harus dirumuskan secara bersama

E. Tes Formatif

Dalam test ini setiap anda harus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Sebutkan perbedaan dari StringBuffer dan StringBuilder !

Kesalahan apakah yang terdapat pada kode berikut ini ?

Listing Program

```
1 public void Test{
2 String teks;
3 public void Test (String s){
4 this.text = s;
5 }
6 public static void main (String[] args){
7 Test test = new Test (“ABC”);
8 System.out.println(test);
9 }
10 }
```

Pada method process, apa yang dimaksud dengan ‘Public Abstract Input Stream get Input Stream()’ !

F. Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Perbedaan StringBuffer dan StringBuilder :



.....
.....
.....
.....
.....
.....
.....
.....

LJ- 02 : Kesalahan dalam kode :



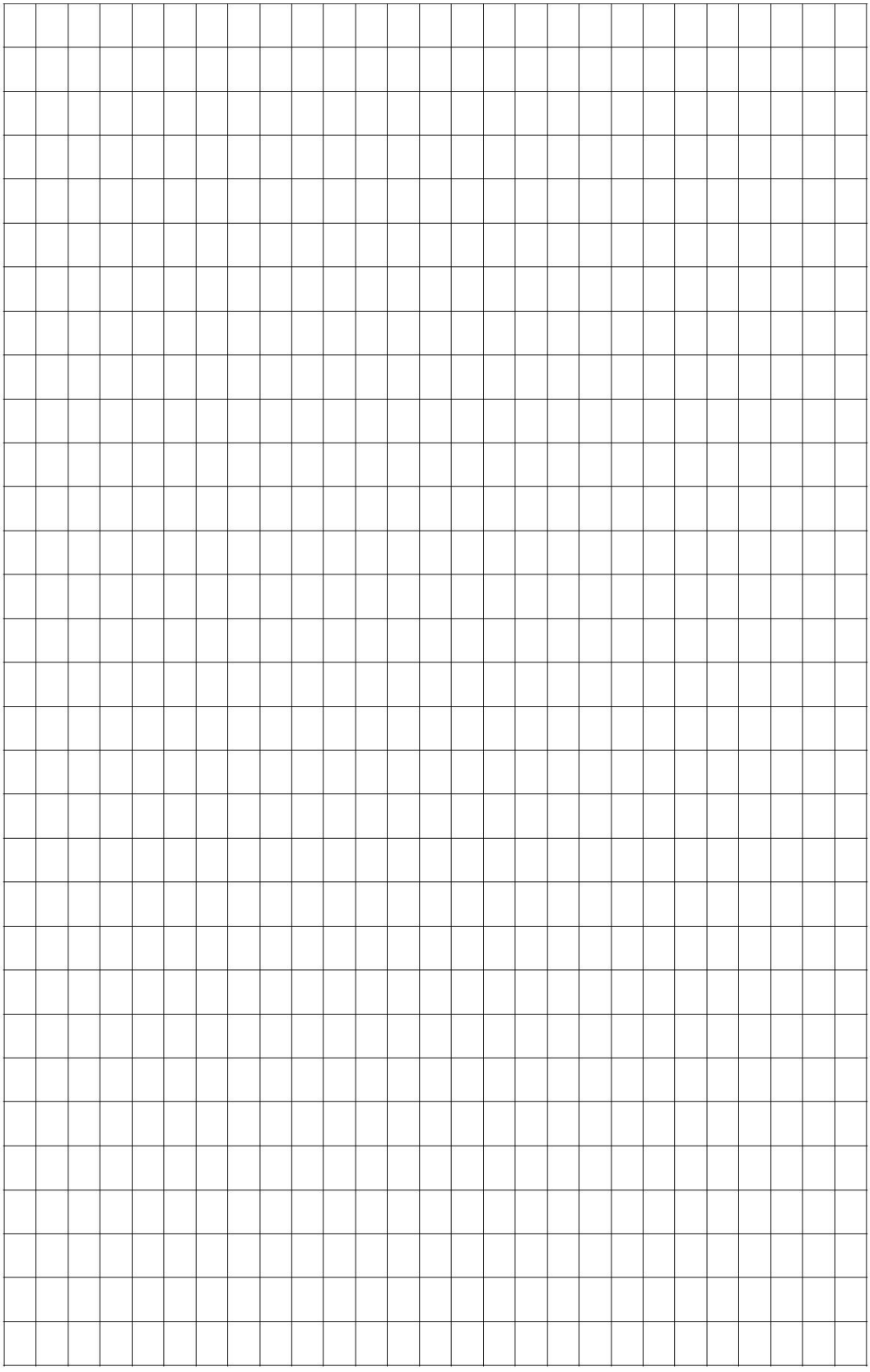
.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 : Pengertian dari Public Abstract Input Stream get Input Stream() :



.....
.....
.....
.....
.....
.....

G. Lembar Kerja Siswa



3. Kegiatan 3 : Exception Handling

A. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 3 ini siswa diharapkan dapat :

Menganalisismekanisme penangganan kesalahan.

Menyajikanbermacam-macam cara untuk mencari tipe kesalahan Uraian Materi.

B. Uraian Materi

Dasar Exception

Exception adalah sebuah event yang menjalankan alur proses normal pada program. Event ini biasanya berupa kesalahan(error) dari beberapa bentuk. Ini disebabkan program kita berakhir tidak normal. Dalam bahasa java, ketika terjadi kesalahan, otomatis akan dilemparkan sebuah objek yang disebut exception, yang kemudian dapat diproses lebih lanjut oleh method yang siap menangani kesalahan tersebut. Method tersebut dapat dipilih untuk menangani exception berdasarkan tipe tertentu.Exception dapat muncul tidak beraturan dalam suatu method, atau dapat juga dibuat secara menual dan nantinya melaporkan sejumlah keadaan kesalahan ke method yang memanggil.

2) Tipe-Tipe Exception

Beberapa exception yang telah digunakan dalam begian-bagian terdahulu adalah ArithmeticException, FileNotFoundException, dan InputMisMatchException. Masih banyak kelas exception lain yang digunakan dalam java, antara lain NullPointerException,ClassNotFoundException,IOException, RunTimeException, IndexOutOBoundsException, IllegalArgumentException, dan masih banyak lagi kelas exception yang digunakan dalam java.

Kelas **Throwable** merupakan akar dari semua kelas exception.Semua kelas exception java mewarisi secara langsung atau tidak langsung dari Throwable. Kita bisa menciptakan kelas exception sendiri dengan cara mewarisi exception atau subclass exception.

Kelas-kelas exception dapat diklasifikasikan menjadi 3 tipe utama : error system, exception, dan exception runtime.

Error system dilempar oleh JVM dan direpresentasikan oleh kelas **Error**. Kelas error mendeskripsikan error internal. Error semacam ini jarang terjadi. Jika terjadi, kita dapat memberitahukan kepada user dan menghentikan program secara normal.contoh sub-kelas Error adalah sebagai berikut :

Kelas	Alasan yang mungkin terjadinya Exception
LinkageError	Kelas punya ketergantungan pada kelas lain, tapi kelas lain telah diubah tanpa menjaga kompatibilitasnya setelah kompilasi dilakukan terhadap kelas pertama.
VirtualMachineError	JVM telah kehabisan sumber daya yang dibutuhkan untuk melanjutkan operasi.

- Exception direpresentasikan dalam kelas **Exception**, yang mendeskripsikan error-error yang diakibatkan oleh program kita dan oleh lingkungan luar. Error ini ditangkap dan ditangani oleh program kita. Beberapa contoh subkelas dari Exception adalah sebagai berikut :

Kelas	Alasan yang mungkin terjadinya Exception
ClassNotFoundException	Percobaan menggunakan suatu kelas yang tidak ada.exception ini terjadi jika kelas yang dijalankan tidak menggunakan perintah java.
IOException	Berkaitan dengan operasi masukan/ keluaran yang tidak valid, membaca melampaui akhir suatu file, dan membuka file yang tidak ada. Subkelas dari IOException antara lain :**InterruptedIOException, EOFException, FileNotFoundException.**

Exception runtime direpresentasikan oleh kelas **RunTimeException**, yang mendeskripsikan kesalahan pemrograman, seperti casting yang salah, pengaksesan array diluar batas, dan kesalahan numerik. Exception runtime dilempar oleh JVM. Beberapa sub-kelas RunTimeException adalah sebagai berikut :

Kelas	Alasan yang mungkin terjadinya Exception
ArithmeticException	Kesalahan pada operasi aritmatika
IndexOutOfBoundsException	Beberapa jenis indeks diluar batas
NegativeArraySizeException	Array diciptakan dengan ukuran negatif
NullPointerException	Penggunaan acuan null yang tidak valid
ArrayStoreException	Penyimpanan array dengan tipe data yang tidak sesuai
ClassCastException	Cast yang tidak valid
IllegalArrayArgumentException	Argumen yang tidak benar
SecurityException	Aturan sekuriti yang dilanggar
IllegalMonitorStateException	Operasi monitor illegal
IllegalStateException	Lingkungan yang tidak benar
UnsupportedOperationException	Operasi yang tidak didukung

Penanganan Exceptions

Untuk menangani exception dalam java, kita gunakan blok try-catch-throw-throws-finally. Apa yang kita lakukan dalam program kita adalah kita menempatkan pernyataan yang mungkin menghasilkan exception dalam blok ini. Bentuk umum dari blok try-catch-finally adalah :

Listing Program

```
try{
//tulis pernyataan yang dapat mengakibatkan exception
//dalam blok ini
}
catch( <exceptionType1><varName1> ){
//tulis aksi apa dari program Anda yang dijalankan jika
ada
//exception tipe tertentu terjadi
}
```

Exception dilemparkan selama eksekusi dari blok *try* dapat ditangkap dan ditangani dalam blok*catch*. Kode dalam blok *finally* selalu di-eksekusi.

Berikut ini adalah aspek kunci tentang sintak dari konstruksi try-catch-finally:

Notasi blok bersifat perintah.

Setiap blok *try*, terdapat satu atau lebih blok *catch*, tetapi hanya satu blok *finally.*

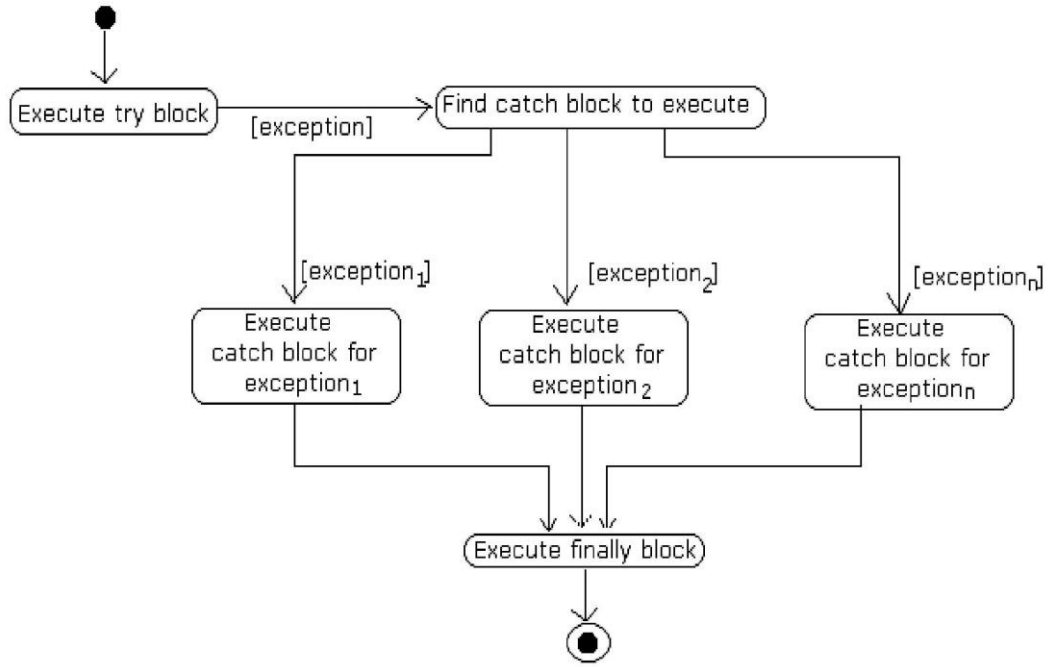
Blok *catch* dan blok *finally* harus selalu muncul dalam konjungsi dengan blok *try*, dan diatasurutan.

Blok *try* harus diikuti oleh **paling sedikit** satu blok *catch* ATAU satu blok *finally,* ataukeduanya.

Setiap blok *catch* mendefinisikan sebuah penanganan exception. Header dari blok *catch*harus membawa satu argumen, dimana exception pada blok tersebut akan ditangani.

Exception harus menjadi class pelempar atau satu dari subclassesnya.

q



Ada lima keywords yang digunakan oleh Java untuk menangani exception ini, yaitu :***Try*, *catch*, *finally*, *throw*, *throws.***

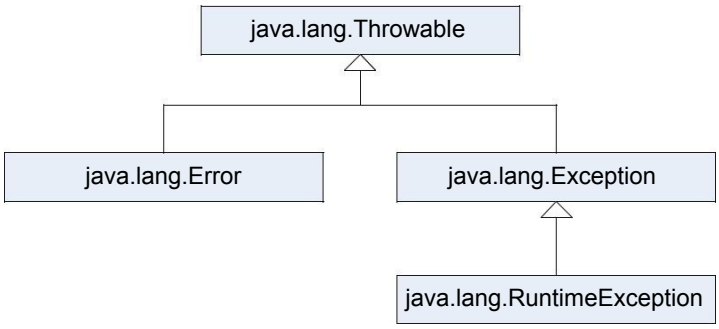
Semua class exception terdapat dalam package **java.lang.**Superclass tertinggi adalah class **Throwable**, tetapi kita hampir tidak pernah menggunakan class ini secara langsung.

Class **Error** ➔ tipe exception yang seharusnya tidak ditangani dengan menggunakan blok try catch karena berhubungan dengan Java run-time

system/evironment. Jadi exception yang terjadi kemungkinannya sangat kristis yang sebaiknya tidak ditangani oleh program kita sendiri.

Class **Exception** ➔ tipe exception yang sebaiknya ditangani oleh program kita secara langsung.

Dalam penggunaannya, kita akan banyak menangani exception yg merupakan turunan dari class **Exception** ini. Salah satu turunannya yang perlu diperhatikan adalah class **RuntimeException**,karena Java memperlakukan class ini & turunannya secara berbeda.



Menampilkan Pesan Exception

Beberapa method standard yang dapat digunakan untuk menampilkan pesan exception merupakan anggota dari kelas java.lang.Throwable

No	Method Pesan Exception	Deskripsi
1	getMessage()	Mengembalikan nilai string yang berisi pesan rinci tentang objek Throwable yang mengalami exception
2	toString()	Mengembalikan nilai string yang berisi pesan singkat tentang objek yang mengalami exception
3	getLocalizedMessage()	Menampilkan pesan exception lokal (yang terjadi pada subkelas saja)
4	printStackTrace()	Method ini bersifat void, dan hanya mencetak informasi tentang objek Throwable

2) Exception Handling

Pada dasarnya, *Exception* merupakan subkelas dari kelas *java*.*lang*.*Throwable*. "*Bukalah dokumentasi java untuk lebih menyakinkan anda*". Karena *Exception* adalah sebuah kelas maka hakikatnya ketika program berjalan dan muncul sebuah bug atau kesalahan maka *bug* tersebut dapat dianggap sebuah *object*. Sehingga ketika *object* ini di tampilkan di layar maka java akan secara otomatis memanggil method *toString* yang terdapat dalam *object* bertipe *Exception* ini. Java memberikan akses kepada *developer* untuk mengambil *object* bug yang terjadi ini dengan mekanisme yang dikenal *ExceptionHandling*. *Exceptionhandling* merupakan fasilitas di java yang memberikan flexibilitas kepada *developer* untuk menangkap *bug* atau kesalahan yang terjadi ketika program berjalan. Contoh *ExceptionHandling* akan dibahas pada bagian berikutnya.

Perbedaan antara *ClassError* dan *ClassException* di java

Seperti yang telah dijelaskan diatas bahwa kelas *Exception* merupakan kelas turunan dari kelas *Throwable* di *packageJava*.*Lang*. Selain *Exception*, java.*lang*.*Throwable* juga memiliki *subclass* yaitu *classError*. Tentu, kita bertanya-tanya, sebetulnya apa sih perbedaan antara *classError* dengan *classException*.

Penjelasan dari ClassError

"An Error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch. Most such errors are abnormal conditions" (JDK 5.0 *Documentation*)

Penjelasan dari *classException*

"The class *Exception* and its subclasses are a form of *Throwable* that indicates conditions that a reasonable application might want to catch. " (JDK 5.0 *Documentation*)Seperti dari penjelasan yang diberikan oleh *JDKDocumentation*, maka dapat kita lihat bahwa *error* dan *exception* pada dasarnya berbeda. *Error* merupakan masalah yang muncul tapi tidak ada alasan yang kuat untuk menangkapnya. Sedangkan *Exception* merupakan kesalahan kecil yang muncul dan ingin diperlakukan sesuai keinginan developer.

3) Keyword penting pada Exception Handling

Ada 5 *keyword* penting dalam java dalam hal *exceptionhandling* :

Try

Keywordtry biasanya digunakan dalam suatu *blockprogram*. Keyword ini digunakan untuk mencoba menjalankan *blockprogram*, kemudian mengenai dimana munculnya kesalahan yang ingin diproses.Keyword ini juga harus dipasangkan dengan keywordcatchatau keywordfinally yang akan dibahas pada point kedua dan ketiga.

Contoh penggunaan :

Listing Program

```
public class A
{
public static void main(String[] args) {
try
{
int a = 1 / 0; // berpotensi untuk menimbulkan kesalahan
yaitu
// pembagian dengan bilangan 0
System.out.println("perintah selanjutnya");
}
catch (Exception kesalahan)
{
System.err.println(kesalahan);
}
}
}
```

Output :

Output

```
java.lang.ArithmeticException: / by zero
```

Perhatikan contoh diatas, ada beberapa hal penting yang perlu dilihat. Pertama, *blockprogram* yag diyakini menimbulkan kesalahan maka ada di dalam *blocktry* and *catch*.Kedua, kesalahan yang muncul akan dianggap sebagai *object* dan ditangkap *catch* kemudian di *assign* ke *variable* kesalahan dengan tipe *Exception*. Ketiga, perintah setelah munculnya kesalahan pada *blocktry* tidak akan dieksekusi.

Catch

Jika anda sudah melihat contoh *try* maka secara tidak langsung anda sudah memahami kegunaan dari keyword ini. Dalam java, keyword *catch* harus dipasangkan dengan *try*. Kegunaan keyword ini adalah menangkap kesalahan atau *bug* yang terjadi dalam *blocktry.* Setelah menangkap kesalahan yang terjadi maka *developer* dapat melakukan hal apapun pada *blockcatch* sesuai keinginan *developer*.

Contoh Program :

Listing Program

```
catch(Exception kesalahan)
{
System.out.println("mohon maaf, terdapat kesalahan pada
program");
//lakukan hal lainnya disini
}
```

Keywordcatchjuga dapat diletakan berulang-ulangsesuai dengan kebutuhan.

Contoh :

Listing Program

```
public class A
{
public static void main(String[] args) {
```

```
try
{
int a = 1/0; //berpotensi untuk menimbulkan kesalahan
yaitu pembagian dengan bilangan 0
System.out.println("perintah selanjutnya");
}
catch(NullPointerException e)
{
}
catch(ArrayIndexOutOfBoundsException e)
{
}
catch(Exception e)
{
}
}
}
```

Finally

Keywordfinally merupakan keyword yang menunjukan bahwa *blockprogram* tersebut akan selalu dieksekusi meskipun adanya kesalahan yang muncul atau pun tidak ada. Setiap try membutuhkan sekurang-kurangnya satu bagian catch atau finally yang cocok. Jika tidak mendapatkan bagian catch yang cocok, maka bagian finally akan dieksekusi sebelum akhir program, atau setiap kali suatu method akan kembali ke pemanggilnya, melalui exception yang tidak dapat ditangkap, atau melalu pernyataan return, bagian finally akan dieksekusi sebelum kembali ke method lagi.

Contoh implementasinya pada program :

Listing Program

```
public class A
{
public static void main(String[] args) {
try
{
```

```
int a = 1/0; }q
finally
{
System.out.println("terima kasih telah menjalankan
program");
}
}
}
```

Output Program diatas:

Output

```
terima kasih telah menjalankan program
```

Jika saya lakukan modifikasi program diatas menjadi :

Listing Program

```
public class A
{
public static void main(String[] args) {
try
{
int a = 1/0;
}
catch (Exception e)
{
System.out.println("ada kesalahan yang muncul");
}
finally
{
System.out.println("terima kasih telah menjalankan
program");
}
}
}
```

Output Program :

Output

```
ada kesalahan yang munculterima kasih telah menjalankan program
```

Perhatikan kedua contoh diatas, *blockfinally* akan selalu dieksekusi meskipun adanya kesalahan atau tidak pada *blocktry*.Berbeda dengan**keyword catch,** keyword**finally** hanya dapat diletakan 1 kali setelah keyword**try**.

Throw

KeywordThrow digunakan untuk melemparkan suatu *bug* yang dibuat secara manual.

Contoh program :

Listing Program

```
public class A
{
public static void main(String[] args) {
try
{
throw new Exception("kesalahan terjadi");
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output Program :

Output

```
java.lang.Exception: kesalahan terjadi
```

Seperti yang anda lihat pada program diatas, pada *keyword***throw new Exception("kesalahan terjadi");**akan melempar *object* bertipe **exception**yang merupakan *subclass* dari *class***Exception** sehingga akan dianggap sebagai suatu kesalahan yang harus ditangkap oleh keyword**catch.**

Perhatikan contoh berikut ini :

Listing Program

```
public class A
{
public static void main(String[] args) {
try
{
throw new B(); //cobalah ganti baris ini dengan à new
B();
}
catch(Exception e)
{
System.out.println(e);
}
}
}
class B extends Exception
{
B()
{
}
public String toString()
{
return "object dengan tipe kelas B";
}
}
```

Output Program :

Output

```
object dengan tipe kelas B
```

Program diatas telah mendefinisikan suatu kelas B *mengextends* dari kelas *Exception*. Ketika kita melakukan *thrownewB();* maka object dari kelas bertipe B ini akan dianggap kesalahan dan ditangkap oleh *blockcatch*. Sekarang jika anda menghilangkan keywordthrow apa yang terjadi? .

Throws

Keywordthrows digunakan dalam suatu method atau kelas yang mungkin menghasilkan suatu kesalahan sehingga perlu ditangkap errornya.Cara mendefinisikannya dalam method adalah sebagai berikut : *<method modifier> type method-name throws* **exception-list1, exceptio-list2, …** {}. Contoh Program :

Listing Program

```
public class A
{
public static void main(String[] args) {
try
{
f();
}
catch(Exception e)
{
System.out.println(e);
}
}
public static void f() throws NullPointerException,
ArrayIndexOutOfBoundsException
{
//implementasi method
throw new NullPointerException();
```

```
17 //throw new ArrayIndexOutOfBoundsException();
}
}
```

Output Program :

Output

```
java.lang.NullPointerException
```

Contoh program lainnya :

Listing Program

```
public class A
{
public static void main(String[] args) {
try
{
f();
}
catch(Exception e)
{
System.out.println(e);
}
}
public static void f() throws NullPointerException,
ArrayIndexOutOfBoundsException
{
//implementasi method
//throw new NullPointerException();
throw new ArrayIndexOutOfBoundsException();
}
}
```

Output Program :

Output

```
java.lang.ArrayIndexOutOfBoundsException
```

Perhatikan kedua contoh pengguaan keyword*throws* pada *method*. Ketika method tersebut dipanggil dalam *blocktry*. Maka method tersebut akan membuat *object* yang merupakan *subclass* dari *classThrowable* dan methodtersebut akan melemparkan kesalahan yang ada dalam *blockmethod* kedalam *blocktry*. Di dalam *blocktry*, kesalahan tersebut kemudian ditangkap kedalam *blockcatch*.

C. Rangkuman

Exception adalah sebuah peristiwa yang menjalankan alur proses normal pada program. Peristiwa ini biasanya berupa kesalahan(error) dari beberapa bentuk. Ini disebabkan program kita berakhir tidak normal. Untuk menangani exception dalam java, kita gunakan blok try-catch finnaly. Ada lima keywords yang digunakan oleh Java untuk menangani exception ini, yaitu :Try, catch, finally, throw, throws.

Semua class exception terdapat dalam package java.lang. Superclass tertinggi adalah class Throwable, tetapi class ini hampir tidak pernah digunakan secara langsung.

Class Error merupakan tipe exception yang seharusnya tidak ditangani dengan menggunakan blok try catch karena berhubungan dengan Java run-time system/evironment. Sedangkan Class Exception merupakan tipe exception yang sebaiknya ditangani oleh program kita secara langsung.

Manfaat dari penggunaan Exception adalah pemisahan antara pendeteksian error(dilakukan didalam suatu metode yang dipanggil) dengan penanganan error (dilakukan didalam metode pemanggil).

D. Tugas

Tugas 1

Tulislah suatu program yang meminta user untuk memasukkan 2 integer dan menampilkan penjumlahan atas keduanya. Program anda harus meminta pengguna memasukkan 2 integer kembali bila inputan sebelumnya tidak tepat. Tipe exception yang digunakan adalah **NumberFormatException**.

❖ Mengamati Listing Program dan Output Program

Tentukan nama class yang akan digunakan.

Tentukan variabel yang akan digunakan.

Tentukan method yang akan digunakan.

Buatlah class diagram yang menggambarkan bagian-bagian dari class yang telah ditentukan. Contoh class diagram :

Nama class
Method
Operasi

Buatlah listing programnya.

Lakukan kompilasi dan debug pada program.

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	

q	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil program yang telah kamu buat dengan hasil program teman sebangku atau kelompok lain

Dari hasil perbandingan tersebut, hal penting apa yang harus dirumuskan secara bersama

E. Tes Formatif

Dalam test ini setiap anda harus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

Sebutkan definisi dari beberapa istilah berikut :

Try

Catch

Finally

Apa kegunaan dari keyword Throw dan Throws ?

Bagaimana cara melemparkan suatu exception? Apakah boleh melempar beberapa exception sekaligus menggunakan satu statement **throw**?

Apa keluaran (output) dari kode berikut ini ?

Listing Program

```
1 public class Test{
2public static void main (String [] args){
try{
int nilai = 30;
if (nilai < 40)
```

```
throw new Exception ("nilai terlalu kecil");
}
catch (Exception ex){
System.out.println (ex.getMessage());
}
System.out.println ("Lanjut setelah blok
catch");
}
}
```

Apakah keluaran kode bila baris

`Int nilai = 30 ;` Diganti dengan `Int nilai = 50 ;`

Jelaskan perbedaan antara catch dan throw !

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Definisi dari istilah :



a)Try

.....

.....

.....

.....

b) Catch

.....

.....

.....

.....

c)Finally

.....

.....

.....

.....

LJ- 02 : Kegunaan dari keyword Throw dan Throws :



.....

.....

.....

.....

.....

.....

LJ- 03 : Cara melempar suatu Exception :



.....

.....

.....

.....

.....

.....

.....

LJ- 04: Keluaran (output) dari kode :



.....

.....

.....

Bila diganti dengan (`nilai int = 50`) :

.....

.....

.....

LJ- 05: Perbedaan catch dan throw :



.....

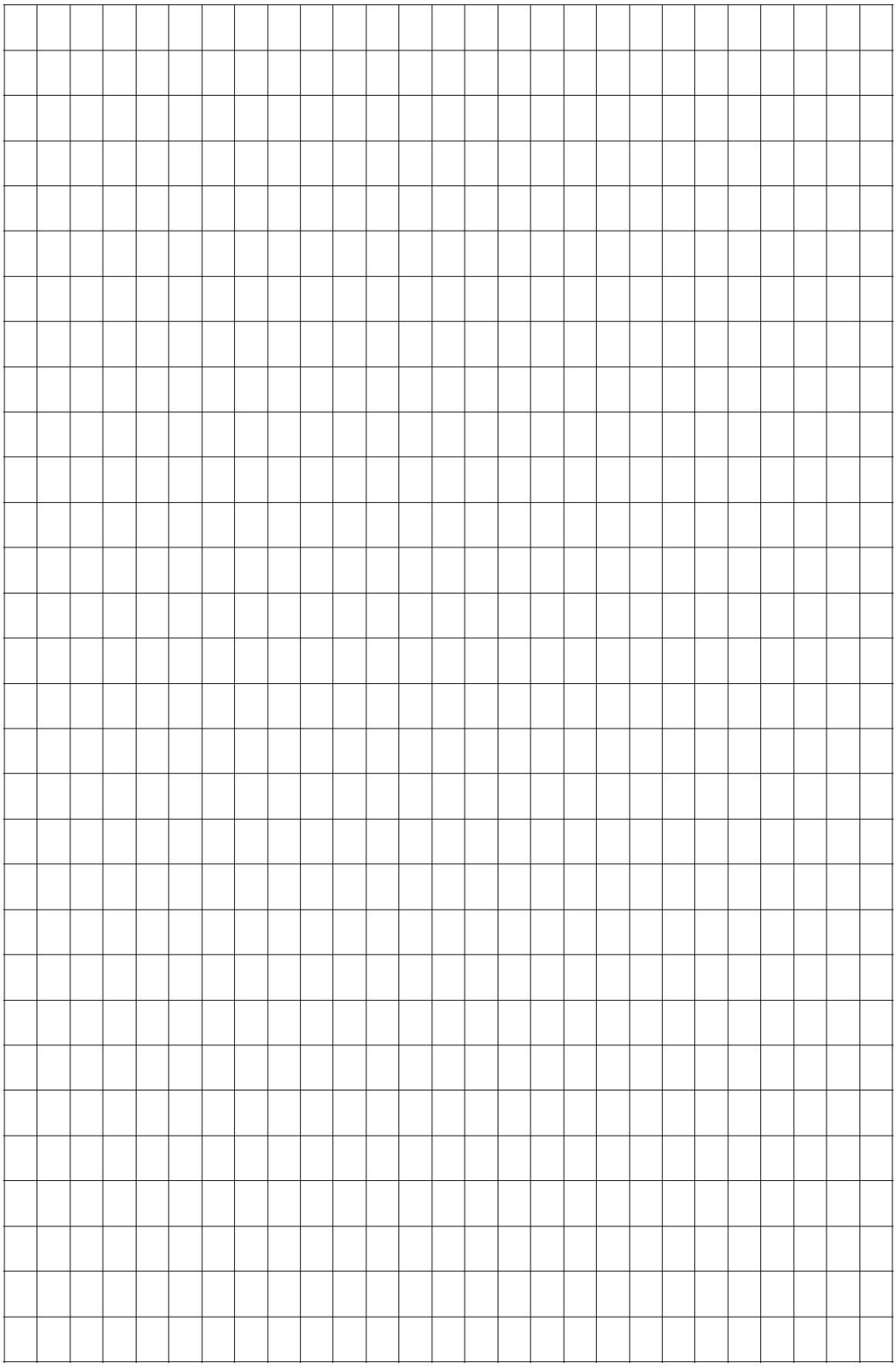
.....

.....

.....

.....

G. Lembqar Kerja Siswa



4. Kegiatan 4 : String

A. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 4 ini siswa diharapkan dapat :

Memahami mekanisme penangganan kesalahan.

Menyajikan bermacam-macam cara untuk mencari tipe kesalahan.

B. Uraian Materi

Pengolahan String

Dalam pemrograman Java string merupakan aspek penting, karena dapatmempelajari mengenai class dan objek melalui penggunaan string. String sebenarnyamerupakan class yang terdapat dalam library Java. Java String merupakan salah satukelas dasar yang disediakan oleh Java untuk memanipulasi karakter.

Membuat Objek String

Java mendefinisikan class String dalam package java.lang.String, sehingga tidak perlu melakukan impor secara eksplisit. Java String digunakan untuk mendefinisikan string yang konstant (tidak bisa berubah

Menggabungkan String

Seringkali dalam pemrograman kita perlu menggabungkan String untuk mendapatkan String baru. Kita dapat menggunakan operator (+) untuk menggabungkan beberapa String.

Contoh penggunaan :

Stringku = “Ini adalah contoh”+”penggabungan String”;

Kita juga dapat menggunakan operator += untuk menggabungkan variabel String,

misalnya seperti contoh berikut :

String kata = ”Ini perkataan”;

Kata += “saya sendiri”;

Menentukan Awal Dan Akhir String

Untuk menentukan awal dan akhir String, kita dapat menggunakan dua fungsi utama, yaitu : startsWith(String s)

Dengan fungsi ini, maka objek String yang bersangkutan akan diperiksa, apakah diawali oleh objek String s, pada parameter fungsi ini endsWith(String s)

Dengan fungsi ini, maka objek string yang bersangkutan akan diperiksa, apakah diakhiri oleh objek string s, pada parameter fungsi ini.

Fungsi diatas akan menghasilkan nilai boolean ***true*** bila benar dan ***false*** bila salah.

Mengurutkan String

Dapat juga melakukan pengurutan string dengan method compareTo().Method ini membandingkan karakter-karakter pada String secara berurutan dari awal String. Misalnya string pertama bernilai “a” dan string kedua bernilai “b”, maka apabila Stringpertama.compareTo(Stringkedua) akan menghasilkan nilai negatif (<0)dan apabila dilakukan sebaliknya akan menghasilkan nilai positif (>0). Nilai 0 akandihasilkan apabila string pertama dan kedua sama.

Mendapatkan Panjang String

Kita dapat memperoleh panjang string dengan menggunakan method length(); seperti contoh berikut ini :

String panjang = “ini panjangnya 17”;

System out.println(panjang.length());

Mencari Posisi Karater Atau SubString Dari String

Ada dua method yang dapat digunakan untuk mencari posisi karakter dari string dan dua method untuk mendapatkan posisi subString dari string.

Method untuk mencari posisi karakter pada String :

*IndexOfchar (karakter)*Memerlukan argumen berupa karakter dan akan mengembalikan nilai posisiindeks dari karakter yang dicari. Posisi yang dikembalikan adalah posisipertama dari karakter yang ditemukan. Bila karakter

tidak ditemukan, makaakan mengembalikan nilai -1.IndexOf(char karakter, int indeks) *s*ama dengan sebelumnya, tetapi memerlukan argumen tambahan, yaitu indeks posisi awal pencarian dalam integer.

Method untuk mencari posisi subString pada String : indexOf(String Str)Penggunaan dan fungsi sama dengan method untuk char. indexOf(String str, ont indeks)Penggunaan dan fungsi sama dengan method untuk char.

2) StringBuffer

StringBufferadalah pasangan class String yang menyediakan banyak fungsi string yang umum. StringBuffer merepresentasikan urutan karakter yang dapat dikembangkan dan ditulis ulang. StringBuffer dapat disisipi karakter dan subString di tengahnya, atau ditambah di belakangnya. StringBuffer memiliki default kapasitas 16 karakter, tapi biasanya ukuran diatur sendiri dengan mendefinisikan kapasitas pada saat pembuatan.

Misalnya adalah sebagai berikut:StringBuffer baru = new StringBuffer(50); Contoh diatas merupakan StringBuffer kosong dengan kapasitas 50 karakter. Ada 3 cara untuk mengefinisikan StringBuffer :

StringBuffer baru = new StringBuffer()

secara tidak langsung variabel baru akan menjadi objek StringBuffer dengan ukuran 16 karakter karena defaultnya adalah 16 karakter

StringBuffer baru1 = new StringBuffer(50)

objek baru1 merupakan StringBuffer dengan panjang karakter 50

StringBuffer baru2 = new StringBuffer(*String*)

objek baru2 merupakan objek StringBuffer dengan panjang karakter ***String* + 16 karakter.**

Berikut merupakan contoh deklarasi StringBuffer :

Listing Program

```
public class SB{
public static void main (String args[]){
String kata = "Java";
StringBuffer baru = new StringBuffer();
```

```
StringBuffer baru1 = new StringBuffer(50);
StringBuffer baru2 = new StringBuffer(kata);
 7 System.out.println("baru    : "+baru.capacity());
 8 System.out.println("baru1   : "+baru1.capacity());
 9 System.out.println("baru1   : "+baru2.capacity());
}
}
```

Berbeda denga type data primitif yang lain, type data String dalam Java diperlakukan sebagai object. Object String dalam Java dapat dibuat dengan dua cara, yaitu:

Penulisan sesuatu di dalam tanda antara petik ganda (literal String). Cara ini digunakan untuk mengakomodasi kebiasaan dari bahasa C/C++

String s = ”Hello World”;

System.out.println(“Hello World”);

Pembuatan object String dengan keyword new.

String s = new String(“Halo”);

Class String mempunyai atribut final, sehingga Class String tidak dapat di-extends / diturunkan.

Dalam Java, terdapat dua jenis memory yaitu:

Stack (tempat local variable dan tumpukan method)

Heap (tempat instance variable dan object), Di dalam heap terdapat bagian memory yang disebut dengan **String constant pool.**

Bila kita membuat object String dengan penulisan sesuatu di antara tanda petik ganda (literal String), maka object String tersebut akan berada di dalam String constant pool. Sedangkan bila kita membuat String dengan keyword new, maka object String tersebut akan berada di dalam heap (tetapi diluar String constant pool). Khusus untuk pembuatan object String dengan keyword **new** (ex: String sample = new String(“Hello World”)), sebenarnya terdiri dari 3 buah proses (yang melibatkan 2 buah object String), yaitu :

Pembuatan object String “Hello World” di dalam String constant pool. Hal ini karena “Hello World” adalah literal String yang otomatis membuat object di String constant pool.

Pembuatan object String “Hello World” di dalam heap (non constant pool).

Penghapusan object String “Hello World” di dalam String constant pool (bila tidak ada yang mereferensi String) ini.

Pada saat java menjumpai literal String (dalam kode program), maka java akan mencari String yang sama dengan literal String tersebut di dalam String constant pool. Bila ternyata di dalam pool ditemukan object String yang sama, maka reference akan menunjuk pada object String di dalam pool tersebut. Bila ternyata java tidak menemukan di dalam String constant pool, maka java akan membuat object String di dalam String constant pool terlebih dahulu.

Object String adalah immutable (tidak dapat diubah) . Pengertian tidak dapat diubah adalah, sekali sebuah object String berisi suatu nilai, maka nilai tersebut tidak dapat diubah (tidak peduli apakah object String tersebut berada pada heap ataupun String constant pool). Dalam praktek pemrograman, kita merasa bahwa object String dapat berubah, hal ini karena yang berubah adalah nilai reference penunjuk object String bukan object String tersebut.

String Class sangat tidak efektif bila kita ingin melakukan banyak modifikasi terhadap suatu String object, hal ini karena sifat dari String Class yang immutable (banyak modfikasi pada suatu kelas String akan dapat menyebabkan banyaknya object String yang terlibat). StringBuffer dan StringBuilder Class mengatasi permasalahan ini (mutable). StringBuffer thread safe sehingga dapat menjamin konsistensi operasi pada String object.

Pada String Class, method-method akan mengembalikan object String baru (hasil modifikasi) tanpa mengubah object String tempat method dipanggil (karena immutable)Pada kelas StringBuffer Class method-method akan memodifikasi object tempat method dipanggil, dan kemudian mengembalikan object tersebut sebagai return value dari method.Method utama pada StringBuffer adalah append dan insert method.Dan dengan Class StringBuffer kita dapat melakukan chaining method.

3) Konstruktor

Konstruktor pada Java merupakan method khusus yang dipakai oleh Java untuk membuat sebuah object didalam kelas dan tiap kelas boleh memiliki lebih dari satu konstruktor.

Karakteristik konstruktor :

Nama Konstruktor = Nama Kelas.

Tidak mengembalikan nilai atau return value termasuk void.

Cara menggunakan konstruktor adalah dengan menggunakan kata kunci **new**. Jika didalam kelas tidak dituliskan konstruktor, Java akan secara default menambahkan konstruktor kosong kedalam kelas tersebut.

Pemanggilan Konstruktor

Membuat konstruktor

Listing Program

```
public class Constr{
Constr ()
{
}
}
```

Konstruktor dipanggil saat membuat sebuah object. Sama seperti membuat object pada class.

Contoh :

Listing Program

```
public class Constructor_1{
float nilaiAkhir;
public Constructor_1(int nilai_akhir){
nilaiAkhir=nilai_akhir;
}
public String grade(){
String nilaigrade;
if(nilaiAkhir>=50)
nilaigrade=”Bagus”;
else
nilaigrade=”Jelek”;
return nilaigrade;
}
public void cetak(){
```

```
15 System.out.println("Grade nilainya = "+grade());
}
public static void main (String [] args){
Constructor_1 hasil = new Constructor_1(67);
hasil.cetak();
}
}
```

Overload Konstruktor

Sebuah class mungkin memiliki lebih dari satu konstruktor dengan parameter yang berbeda satu sama lainnya.

Contoh :

Listing Program

```
public class Constructor_2{
float nilai1, nilai2;
public Constructor_2(int bil1){
nilai1 = bil1;
}
public Constructor_2(int bil1, bil2){
nilai = bil1+bil2;
}
public String grade()
{
String nilaigrade;
if(nilai1>=50)
nilaigrade = "Bagus";
else
nilai grade = "Jelek";
return nilaigrade;
}
public void cetak(){
System.out.println("Grade nilainya = "+grade());
}
public static void main (String [] args){
```

```
22 Constructor_2 hasil = new Constructor_2 (45);
hasil.cetak();
Constructor_2 hasilnya = new Constructor_2 (45,35);
hasilnya.cetak();
}
}
```

Modifier

Modifier digunakan untuk menentukan sifat dari suatu kelas dan menentukan preveledge (hak akses) dari kelas lain. Modifier juga digunakan untuk menentukan relasi (extend atau implements) dengan kelas lainnya. Berikut ini adalah wilayah modifier akses :

Wilayah Akses	Public	Protected	Default	Private
Dikelas yang sama	✓	✓	✓	✓
Beda kelas, di package yang sama	✓	✓	✓	✘
Beda kelas, beda package, dikelas turunan	✓	✓	✘	✘
Beda kelas, beda package, tidak di kelas turunan	✓	✘	✘	✘

Keterangan :

Public : menyatakan bahwa kelas/ method/ attribute dapat diakses oleh kelas lain dimanapun letaknya.

Protected : menyatakan bahwa kelas/ method/ attribute tersebut dapat diakses oleh kelas lain yang berada dalam satu package atau kelas lain tersebut merupakan turunannya.

Private : menyatakan bahwa kelas tersebut tidak dapat diakses sama sekali oleh kelas lain bahkan tidak dapat diakses oleh kelas turunannya. Attribute yang bersifat private hanya dapat diakses oleh method dalam kelas yang sama, kelas lain masih dapat mengakses melalui method tersebut asal method tersebut modifiernya public.

5) Class String

String pada java adalah object dan sifatnya read-only (immutable).Karena sifat immutable ini setiap perubahan terhadap isi string akan dibuat string baru untuk menampung perubahan tersebut.

Contoh :

String s1 = “hello world”;

S1 = “hello java”;

Akan menyebabkan dua kali pengalokasian memory untuk dua objek string tersebut diatas, dengan referensi terakhir s1 ke string “hello java”.

Operator Pada String

Anda dapat menggunakan operator + untuk mengabungan dua string atau string dengan type data primitive lainnya (menyebabkan konversi otomatis dari type lain ke string).

Contoh :Str = “Jumlah data “ + jd;

Method pada class String

Str.length()

Str.compareTo(String Str1)

Str.compareToIgnoreCase(String Str1)

Str.concat(String Str1)

Str.equals(String Str1), membandingkan isi Str sama dengan Str1

Str.equalsIgnoreCase(String Str1)

Str.startsWith(String prefix)

Str.startsWith(String prefix, int offset)

Str.endsWith(String suffix)

Str.indexOf(int ch)

Str.indexOf(int ch, int fromIndex)

Str.indexOf(String str)

Str.indexOf(String str, int fromIndex)

Static String.valueOf(value), mengembalikan string dari suatu nilai Boolean, char, char array, int, long, double, float.

Untuk pencarian dari akhir string gunakan lastIndexOf

Str.trim(), menghapus white space diawal dan akhir string

Str.toLowerCase()

Str.toUpperCase()

Str.charAT()

Contoh : Metode Enkripsi Ceasar secara substitusi. M = (C+3) Mod 26

Class StringBuffer

StringBuffer merupakan string yang sifatnya mutable yang isinya dapat dimanipulasi, sebingga sangat menghemat system resource dan performance yang lebih baik pada operasi yang banyak melakukan manipulasi terhadap isi string.

StringBuffer sb =new StringBuffer() ; //StringBuffer dengan kapasitas 16

StringBuffer sb = new StringBuffer(30); //StringBuffer dengan kapasitas 30StringBuffer sb = new StringBuffer(“HendraSoewarno”);

Method pada class StringBuffer

Sb.ensureCapacity(int kapasitas), set kapasitas StringBuffer ke ukuran baru.

Sb.append(value).

Sb.append(Char[] str, int offset, int len)

Sb.capacity(), mengembalikan ukuran kapasitas StringBuffer

Sb.charAt(int index), mengembalikan karakter pada posisi tertentu dari StringBuffer, index dimulai dari 0 (nol).

Sb.delete(int start, int end), menghapus sejumlah karakter dari StringBuffer

Sb.deleteChatAt(int index)

Sb.getChars(int srcBegin, int srcEnd, Char[] dst, int dstBegin)

Sb.insert(int offset, value)

Sb.length()

Sb.reverse()

Sb.setCharAt(int index, char ch)

Sb.setLength(int newLength)

Sb.substring(int start)

Sb.substring(int start, int end)

Sb.toString()

Contoh :

Listing Program

```
class C3 {
protected String plainText;
C3() {
plainText = "";
}
void setPlainText(String newValue) {
plainText = newValue;
}
String getCipherText() {
StringBuffer result = new StringBuffer("");
char huruf;
for (int i=0; i < plainText.length(); i++) {
if (plainText.charAt(i) != 32) {
huruf = (char)((plainText.charAt(i) - 'A' + 3) % 26 +
'A');
result.append(huruf);
}
}
return result.toString();
}
}
```

Contoh dengan Argument String

Listing Program

```
class testArgs {
public static void main(String[] args) {
C3 c = new C3();
if (args.length < 1)
5 System.out.println("Usage: java testArgs
<plaintext>");
else {
c.setPlainText(args[0]);
```

```
8         System.out.println("Hasil         adalah         :
"+c.getCipherText());
}
}
}
```

Untuk menjalankan program diatas gunakan :

java testArgs <plaintext>

Contoh dengan GUI

Listing Program

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class testString {
public static void main(String[] args) {
final C3 c = new C3();
final JFrame frame = new JFrame("Enkripsi RC3");
JLabel label = new JLabel("Plaintext :");
final JLabel labelh = new JLabel("Hasil :");
final JTextField textField = new JTextField(20);
JButton button = new JButton("Encrypt");
button.setMnemonic('E');
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
c.setPlainText(textField.getText());
labelh.setText("Hasil adalah : "+c.getCipherText());
frame.pack();
}
});
frame.getContentPane().setLayout(new FlowLayout());
frame.getContentPane().add(label);
frame.getContentPane().add(textField);
```

```
frame.getContentPane().add(button);
frame.getContentPane().add(labelh);
frame.pack();
frame.show();
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Class StringBuilder

Kelas StringBuilder menyediakan beberapa method ter-overload untuk menyambung Boolean, char, array char, double, float, int, long, dan String kedalam suatu StringBuilder. Contohnya seperti pada kode dibawah ini menyambungkan String dan char kedalam StringBuilder untuk membentuk suatu String baru.

Listing Program

```
stringBuilder stringBuilder = new StringBuilder();
stringBuilder.append("JAVA");
stringBuilder.append(' ');
stringBuilder.append("itu");
stringBuilder.append(' ');
stringBuilder.append("Tangguh !");
```

Kelas StringBuilder juga dapat menyisipkan boolean, char, array char, double, float, int, long, dan String. Contohnya :

Listing Program

```
1  stringBuilder.insert (5, "dan HTML");
```

Beberapa karakter dapat dihapus dari suatu String dalam StringBuilder menggunakan dua method **delete**, membalikkan string menggunakan method **reserve,** mengganti karakter menggunakan method **replace**, atau menetapkan suatu karakter baru didalam suatu String menggunakan method **setCharAt.**

Contoh :

Listing Program

```
stringBuilder.delete (4,7) mengubah builder menjadi
JAVA Tangguh!.
stringBuilder.deleteCharAt (5) mengubah builder menjadi
JAVA tu Tangguh!.
stringBuilder.reserve () mengubah builder menjadi !
huggaT uti AVAJ
stringBuilder.replace   (0,   3,   "HTML")   mengubah
builder menjadi HTML itu Tangguh !.
5 stringBuilder.setCharAt(0, 'j') menetapkan builder
menjadi jAVA itu Tangguh!.
```

C. Rangkuman

String merupakan class yang terdapat dalam library java. Java string merupakan salah satu kelas dasar yang disediakan oleh java untuk memanipulasi karakter. String bersifat immutable, sehingga perubahan terhadap isi String akan dibuat String baru untuk menampung perubahan tersebut.

StringBuffer merepresentasikan urutan karakter yang dapat dikembangkan dan ditulis ulang. StringBuffer dapat disisipi karakter dan sub String ditengah atau dibelakang. Default StringBuffer memiliki 16 karakter, namun ukuran default ini dapa diatur dengan mendefinisikan kapasitas saat pembuatan.StringBuffer bersifat mutable yang isinya dapat dimanipulasi, sehingga dapat menghemat resource dan performance lebih baik pada operasi yang memanipulasi isi String.
Konstruktor merupakan method khusus yang dipakai oleh java untuk membuat sebuah objek didalam kelas, dan setiap kelas boleh memiliki lebih dari satu kontruktor.

D. Tugas

Tugas 1

Buatlah suatu metode yang mencari jumlah huruf didalam String menggunakan header berikut ini :

```
public static int hitungHuruf (String s)
```

Buat program uji yang meminta pengguna untuk memasukkan suatu string dan yang menampilkan jumlah huruf didalam string.

❖ Mengamati Listing Program dan Output Program

Tentukan nama class yang akan digunakan.

Tentukan variabel yang akan digunakan.

Tentukan method yang akan digunakan.

Buatlah class diagram yang menggambarkan bagian-bagian dari class yang telah ditentukan. Contoh class diagram :

Nama class
Method
Operasi

Buatlah listing programnya.

Lakukan kompilasi dan debug pada program.

No	Output Program
1.	
2.	
3.	
4.	
5.	

6.	
7.	
8.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil program yang telah kamu buat dengan hasil program teman sebangku atau kelompok lain

Dari hasil perbandingan tersebut, hal penting apa yang harus dirumuskan secara bersama

E. Tes Formatif

Dalam test ini setiap anda harus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Apa perbedaan konstruktor dengan method?

Apakah output dari kode berikut ini !

Listing Program

```
String s1 = "JAVA itu tangguh!";
String s2 = s1.replace ("V" , "abc");
System.out.println(s1);
System.out.println(s1);
```

Sebutkan dan jelaskan 3 konstruktor dalam StringBuffer !

Tuliskan 3 statement untuk membalik string s menggunakan method reserve dalam kelas StringBuilder !

F. Lembar Jawaban Test Formatif (LJ). LJ-01 : Perbedaan konstruktor dengan method



.....
.....
.....
.....
.....
.....

LJ- 02 : Output dari kode :



.....
.....
.....
.....
.....
.....

LJ- 03 : 3 konstruktor dalam StringBuffer



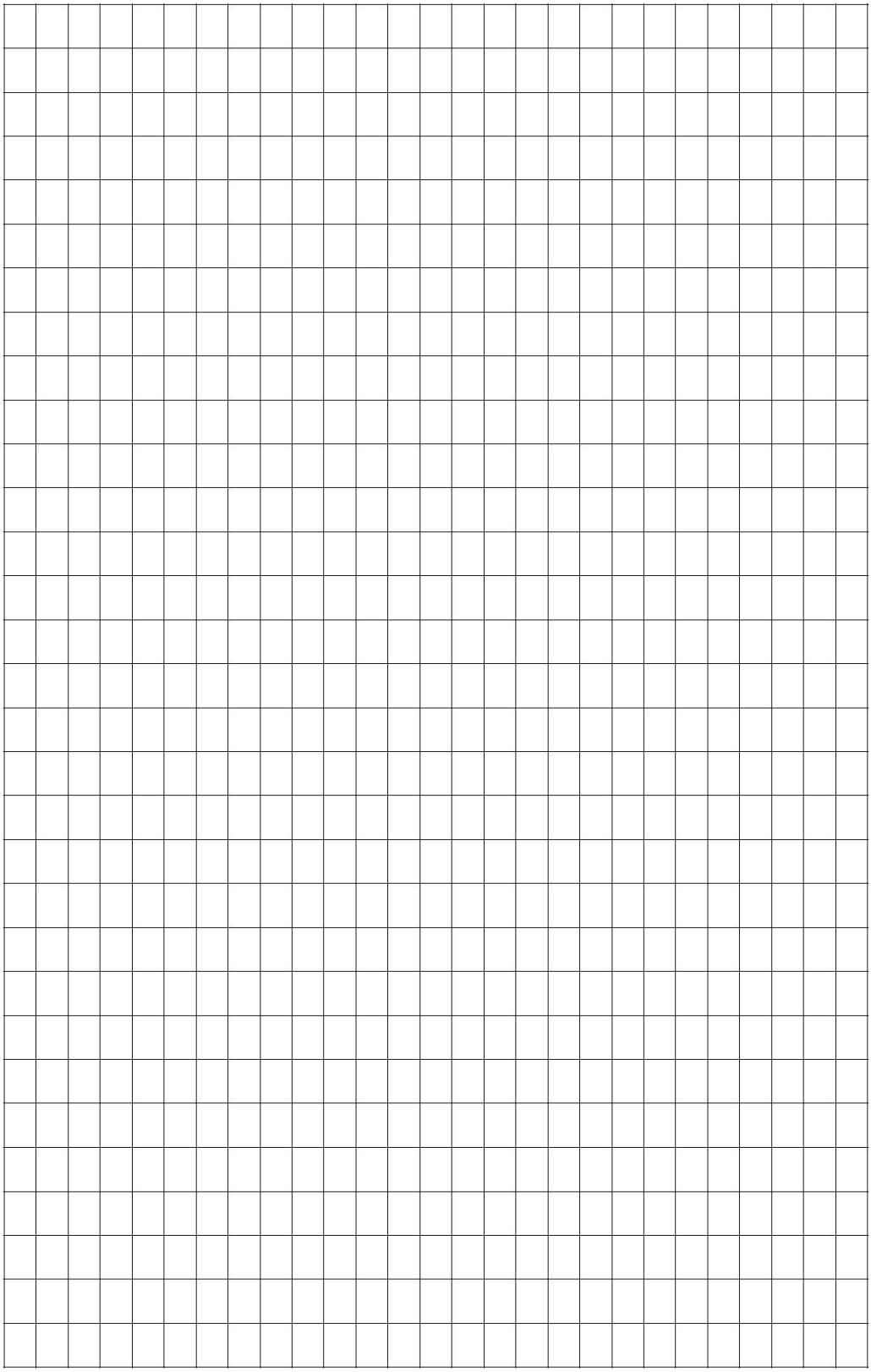
.....
.....
.....
.....
.....
.....
.....

LJ- 04: 3 Statement untuk membalik string s menggunakan method reserve :



.....
.....
.....
.....
.....

G. Lembar Kerja Siswa



Kegiatan 5 : Array

A. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 5 ini siswa diharapkan dapat :

Memahami data collection sebagai media penyimpanan

Menyajikan data collection sebagai penyimpan data

B. Uraian Materi

Deklarasi Array

Array digunakan untuk membuat variabel bisa menampung beberapa data dengan tipe data yang sama alias satu tipe data. Ciri khas varibel yang menggunakan array adalah terdapat simbol **[].** Unyuk mengetahui panjang array yang telah kita buat, kita dapat memakai property *length***,** dengan format sebagai berikut :

`var_array.length` → total elemen array pada dimensi 1.

`var_array[i].length` → total elemen array pada dimensi 2 untuk indeks ke-i pada dimensi 1.

`var_array[i][i].length` → total elemen array pada dimensi 3 untuk indeks ke-I pada dimensi 1 dan indeks ke-I pada dimensi 2dan seterusnya.

Isi dari suatu array dapat kita copy pada array yang lain dengan memanfaatkan method arraycopy() pada class System. Format penulisannya sebagai berikut :

```
System.arraycopy (array1, p1, array2, p2, n);
```

Array memiliki dua tipe, yaituSingle Dimension dan Multi Dimension, sedangkan Proses pendeklarasian variabel yang menggunakan array ada 2 tipe :

Array Single Dimension

Pendeklarasian Variabel Menggunakan Array Tipe Pertama Variabel yang menggunakan array, array ditentukan length/panjangnya terlebih dahulu, sehabis itu baru di inisialisasi.

Contoh

:

Listing Program

```
1 public class Array1{
2 public static void main (String args []) {
3 int nilai [] = new int [3];
4 nilai [0] = 70;
5 nilai [1] = 80;
6 nilai [2] = 65;
7 double ratarata = 0.0;
8 for (int i=0; i<nilai.length; i++)ratarata+=nilai [i];
ratarata/=nilai.length;
System.out.println("Nilai rata-rata = "+ ratarata);
}
}
```

Perlu diingat bahwa index array dimulai dari 0.

Pendeklarasian Variabel Menggunakan Array Tipe Kedua

Variabel yang menggunakan array langsung diinisialisasi.

Contoh :int[] nilai={50,60,70,80};

Lantas bagaimana caranya agar kita mengetahui panjang arraynya, di atas variabel nilai menggunakan array langsung diinisalisasi jika ingin mengetahui panjang arraynya **nilai.length.**

Array MultiDimension

Pada konsep array multidimension, konsepnya sama seperti baris dan kolom pada tabel.

Pendeklarasian Variabel Menggunakan Array Tipe Pertama Sama seperti pada pendeklarasian variabel menggunakan array single dimension, perbedaannya kita harus menentukkan panjang kolomnya, jadi kita menentukan panjang baris dan panjang kolomnya.

Contoh :

Listing Program

```
import java.text.NumberFormat;
public class Array2{
```

```
public static void main (String args []){
NumberFormat nf = NumberFormat.getInstance();
nf.setMaximumFractionDigits(3);
int nilai [][]=new int [2][3];
nilai [0][0]=85;
nilai [0][1]=81;
nilai [0][2]=78;
nilai [1][0]=65;
nilai [1][1]=73;
nilai [1][2]=71;
String MK[]={“RPL”,”PBO”};
double rataRataMK[]= new double[nilai.length];
for (int i=0; i<nilai.length; i++){
for (int j=0; j<nilai[0].length; j++){
rataRataMK[i]+=nilai [i][j];
}
}
rataRataMK[i]/=nilai[0].length;
}
System.out.println(“Nilai Mata Pelajaran\n”);
System.out.println (“MK\ tMinggu1 \ tMinggu2\ tMinggu3\
tRata-Rata”);
for (int i=0; i<nilai.length; i++){
System.out.print(MK[i]+”\t”);
for (int j=0;j<nilai[0].length; j++){
}
System.out.print(nf.format(rataRataMK[i]+”\n”);
}
}
}
```

Seperti biasa indeks baris dimulai dari 0 dan indeks kolom dimulai dari 0.

Pendeklarasian Variabel Menggunakan Array Tipe Kedua

Tipe kedua, proses pendeklarasian langsung dengan inisialisasinya :

Listing Program

```
char [][] abjad={{'A','B'},{'C','D'},{'E','F'}};
```

Untuk mengetahui panjang baris **abjad.length,** sedangkan untuk mengetahui panjang kolom bisa gunakan **abjad.length[0],** Kenapa harus 0 parameter untuk mengetahui panjang kolomnya?, alasannya karena sudah pasti indeks array dimulai dari 0.

Membuat Array

Setelah mendeklarasikan array, maka perlu dibuat array nya terlebih dahulu.Tidak seperti mendeklarasikan variabel tipe data primitif, deklarasi suatu variabel array tidak mengalokasikan memori untuk array tersebut.Kita tidak dapat menugaskan elemen-elemen kepada suatu array kecuali jika array tersebut telah diciptakan. Setelah suatu variabel array diciptakan, kita dapat menciptakan suatu array menggunakan operator **new**dengan sintaks seperti berikut :

```
varRefArray = new tipeElemen[ukuranArray];
```

Statemen ini melakukan dua hal, yaitu menciptakan suatu array menggunakan **new tipeelemen[ukuranArray];** dan menugaskan array yang baru saja diciptakan kepada **varRefArray.**

Ukuran Array dan Nilai Default

Agar memori untuk suatu array dialokasikan, ukuran array harus diberikan untuk menentukan jumlah elemen yang dapat disimpan didalam array itu.Ukuran array tidak dapat diubah setelah array diciptakan. Contoh :

TipeArray namaArray[];
namaArray = new TipeArray [jumlah];

Listing Program

```
1 int nilai [];
2 nilai = new int [5];
```

Dalam program tersebut '[jumlah]' merupakan jumlah data yang dapat ditampung oleh array.

Memproses Array

Ketika memproses elemen-elemen array, seringkali kita menggunakan loop for, karena semua elemen suatu array bertipe sama, dan ukuran array yang telah diketahui, maka sangat cocok untuk menggunakan loop for. Sebagai contoh :

Listing Program

```
1 double [] myList = new double [10];
```

Berikut ini adalah contoh pemrosesan array :

Loop berikut ini menginisialisasi array myList dengan nilai-nilai dari pengguna:

Listing Program

```
1 java.util.Scanner masukkan = new java.util.Scanner
(System.in);
2 System.out.print("masukkan " +myList.length+ "buah
nilai : ");
3 for (int i=0; i<myList.length;i++)
myList[i]=masukkan.nextDouble();
```

Loop berikut ini menginisialisasi array myList dengan nilai-nilai acak antara 0.0 sampai 100.0, tetapi kurang dari 100.0:

Listing Program

```
1 for (int i=0; i<myList.length; i++){
2 myList [i]= Math.random()*100;
```

Untuk menampilkan suatu array, kita harus menampilkan setiap elemen array menggunakan suatu loop sebagai berikut :

Listing Program

```
1 for (int i=0; i<myList.length; i++){
2 System.out.print (myList[i] +" ");
3 }
```

2) Collection

Collection merupakan istilah umum yang dipakai untuk setiap objek yang berfungsi untuk mengelompokkan beberapa objek tertentu menggunakan suatu teknik tertentu pula. Semua class yang berhubungan dengan pengelompokan objek ini dalam java tergabung dalam *Java Collection Framework*, dimana Framework ini diletakan dalam package java.util dan mempunyai dua interface utama, yaitu collection dan map.

Jenis Pengelompokan Collectin ini merupakan pengelompokan satu dimensi. Berdasarkan teknik pengelompokannya terbagi menjadi tiga kelompok yaitu set, list, dan queue. Semua class dalam kelompok collection ini merupakan implementasi dari interface yang sama, maka pada dasarnya cara penggunan class ini adalah sama. Yang membedakannya hanyalah kapan kita harus menggunakannya karena hal ini sangat terkait dengan kebutuhan.

3) Set

Merupakan pengelompokan mengikuti model himpunan dimana setiap anggota-nya harus unik. Urutan maupun letak dari anggotanya tidaklah penting, hanya keberadaan anggotanya saja yang penting. Contoh dalam Program:

Listing Program

```
import java.util.*;
public class setCollection {
public static void main(String[] args) {
Set<String> set1 = new HashSet<String>();
```

```
Colqlections.addAll(set1, "A B C D E ".split(" "));
set1.add("M");
System.out.println("B:  "  +  set1.contains("B"));//ada
H
System.out.println("H: " + set1.contains("H"));//ada N
Set<String> set2 = new HashSet<String>();
Collections.addAll(set2, "E F G H".split(" "));
//Apakah huruf2 di set2 ada di set 1 ?? -> true
12
System.out.println("set2inset1:"+set1.containsAll(set2));
set1.remove("H");
System.out.println("set1: " + set1);
System.out.println("set2 in set1:" +
set1.containsAll(set2));
set1.removeAll(set2);
System.out.println("set2 dihapus dari set1:" + set1);
Collections.addAll(set1, "X Y Z".split(" "));
System.out.println("'X Y Z' ditambahkan ke set1: " +
set1);
}
}
```

List

List merupakan pengelompokan berdasarkan urutan seperti layaknya array, karena itu ia memiliki posisi awal dan juga posisi akhir. Dengan list kita bisa menyimpan suatu objek pada awal atau akhir list, menyisipkan, mengakses serta menghapus isi list pada posisi index tertentu dimana semua proses ini selalu didasarkan pada urutannya. Selain itu isi list tidak harus unik. Beberapa Class java yang mengimplementasi kelas ini antara lain : *Vector, Stack, LinkedList,* dan *ArrayList.*

Contoh dalam Program :

Vector

Listing Program

```
import java.util.Scanner;
import java.util.Vector;
public class VektorJava{
public static void main(String[] args) {
String data = "";
Vector<String> V = new Vector<String>();
Scanner input = new Scanner(System.in);
for(int i=0;i<=3;i++){
System.out.print("Masukan Data "+i+": ");
data = input.next();
V.add(data);
}
System.out.println(V);
String nama = (String) V.get(0);
System.out.println(nama);
}
}
```

Stack

Listing Program

```
import java.io.*;
import java.util.*;
public class StackImplement {
Stack<Integer> stack = new Stack<Integer>();
String str;
int num, n;
public static void main(String[] args) {
StackImplement q = new StackImplement();
}
StackImplement() {
try {
BufferedReader bf = new BufferedReader(
```

```
13 new InputStreamReader(System.in));
14 System.out.print("Banyak Data : ");
15 str = bf.readLine();
16 num = Integer.parseInt(str);
17 for (int i = 1; i <= num; i++) {
18 System.out.print("Masukan Elemen " + i + " : ");
19 str = bf.readLine();
20 n = Integer.parseInt(str);
stack.push(n);
}
} catch (IOException e) {
}
System.out.println("Stack : ");
while (!stack.empty()) {
System.out.print(stack.pop() + "  ");
}
System.out.println();
}
}
```

LinkedList

Listing Program

```
import java.util.LinkedList;
public class LinkListCollection {
public static void main(String[] args) {
LinkedList<String> A = new LinkedList<String>();
String[] nama = {"David","Alfa","Benny"};
//Tambah data data diambil dari array nama;
for(int nList = 0;nList<nama.length;nList++){
A.add(nama[nList]);
}
//Tampil Data
System.out.println("Data Asli : ");
for(int nList = 0;nList<nama.length;nList++){
```

```
13              System.out.println("Indeks      "+nList+":
"+A.get(nList));
}
System.out.println("\nTambah  data  di  Index  ke-3  :
");
A.add(3, "Danni");
for(int nList = 0;nList<=nama.length;nList++){
18              System.out.println("Indeks      "+nList+":
"+A.get(nList));
}
System.out.println("\nDelete data di Index ke-2 : ");
A.remove(2);
for(int nList = 0;nList<nama.length;nList++){
System.out.println("Indeks "+nList+": "+A.get(nList));
}
System.out.println("\nTambah data di Awal list : ");
A.addFirst("Marcelo");
for(int nList = 0;nList<=nama.length;nList++){
System.out.println("Indeks "+nList+": "+A.get(nList));
}
System.out.println("\nTambah  data  di  Akhir  list  :
");
A.addLast("Ferdi");
for(int nList = 0;nList<=nama.length+1;nList++){
System.out.println("Indeks "+nList+": "+A.get(nList));
}
System.out.println("\nTambah data di Akhir list : ");
A.remove(4);
for(int nList = 0;nList<=nama.length;nList++){
38              System.out.println("Indeks      "+nList+":
"+A.get(nList));
```

```
}
}
}
```

ArrayList

Listing Program

```
import java.util.ArrayList;
import java.util.Scanner;
public class ArrayListCollection {
public static void main(String[] args) {
String str = "";
ArrayList<String> senerai = new ArrayList<String>();
Scanner scan = new Scanner(System.in);
while (true) {
try {
System.out.println("\n1. Tambah Data");
System.out.println("2. Lihat Data");
System.out.println("3. Delete Data");
System.out.println("4. Exit");
System.out.print("Masukan pilihan : ");
int pilihan = scan.nextInt();
switch (pilihan) {
case 1:
int jmlDt = 0;
System.out.print("Tambah data : ");
str = scan.next();
senerai.add(str);
jmlDt++;
break;
case 2:
System.out.println("\nIsi senerai sekarang : ");
System.out.println("Jumlah Element : " +
senerai.size());
for (int i = 0; i < senerai.size(); i++) {
```

```
28 System.out.println("Index " + i + ". " +
senerai.get(i));
}
System.out.println("");
break;
case 3:
int indeks;
try {
System.out.print("Masukan Index list yang akan di hapus:
");
indeks = scan.nextInt();
senerai.remove(indeks);
System.out.println(">remove data sukses<");
} catch (IndexOutOfBoundsException a) {
System.out.println("Indeks Array melebihi batas");
}
break;
case 4:
System.out.println("Terima kasih");
System.exit(0);
default:
System.out.println("Inputan Tidak tersedia \n");
}
} catch (Exception e) {
System.err.println("Salah Input");
System.exit(1);
}
}
}
}
```

Implementasi dari ArrayList dan Vector ini mirip seperti array, dimana kita dapat menggakses anggotanya melalui indeksnya. Kelebihannya adalah ia dapat menyesuaikan ukurannya sesuai dengan kebutuhan. Perbedaan dari kedua

kelas ini, vector secara internal telah mendukung sinkronisasi, sedangkan ArrayList secara internal tidak mendukung sinkronisasi.

5) Queue

Queue merupakan model pengelompokan berdasarkan metode antrian suatu prioritas tertentu(contoh FIFO-First In First Out). Beberapa Class java yang mengimplementasi interface *Queue* ini antara lain *PriorityQueue* dan *LinkedList.*Dalam queue terdapat fungsi enqueue yang digunakan untuk mengatur inputan data yang akan masuk antrian, pada fungsi ini juga diperiksa apakah antrian sudah penuh atau belum, jika sudah penuh maka tidak dapat diisi lagi. Fungsi dequeue digunakan untuk mengatur data agarkeluar dari antrian secara tertib, karena queue menggunakan prinsip FIFO. Contoh :

Listing Program

```
import java.io.*;
import java.util.*;
public class QueueJava{
String str;
Int num;
public static void main(String[] args){
QueueJava q = new QueueJava();
}
public QueueJava(){
try{
LinkedList<Integer> list = new LinkedList<Integer>();
BufferedReader bf = new BufferedReader(
new InputStreamReader(System.in));
System.out.print("Banyak Data : ");
str = bf.readLine();
if((num = Integer.parseInt(str)) == 0){
System.out.println("Anda menekan angka nol.");
System.exit(0);
}
else{
```

```
for(int i = 0; i < num; i++){
System.out.print("Masukan Elemen "+i+" : ");
str = bf.readLine();
int n = Integer.parseInt(str);
list.add(n);
}
}
System.out.println("\nElement Pertama   : "
+ list.removeFirst());
System.out.println("Element Terakhir  : "
+ list.removeLast());
32  System.out.println("Element Tengah    : ");
while(!list.isEmpty()){
System.out.print(list.remove() + " ");
}
System.out.println("");
}
catch(Exception e){
System.out.println(e.getMessage()
+ " adalah String.");
System.exit(0);
}
}
}
```

C. Rangkuman

Array merupakan objek yang dapat digunakan untuk menyimpan sejumlah data. Data yang ditampung di array dapat berupa tipe data ataupun kelas(objek). Ciri dari array adalah terdapat simbol [ ].Array ada dua tipe yaitu Array 1 dimention dan Array 2 dimention.

Collection merupakan istilah yang dipakai untuk setiap objek yang berfungsi untuk mengelompokkan beberapa objek tertentu menggunakan teknik tertentu.Semua class yang berhubungan dengan pengelompokkan objek ini tergabung dalam Java Collection Framework.

Set merupakan pengelompokkan mengikuti model himpunan dimana setiap anggotanya harus unik.

List merupakan pengelompokkan berdasarkan urutan seperti array. List dapat menyimpan suatu objek pada awal atau akhir list, menyisipkan, mengakses serta menghapus isi list pada posisi index tertentu dimana semua proses selalu didasarkan pada urutannya. Isi list tidak harus unik.

Queue merupakan model pengelompokkan berdasar metode antrian suatu prioritas tertentu. Contohnya FIFO (first input first output).Dalam queue terdapat fungsi enqueue, berfungsi mengatur inputan pada queue. Dan juga terdapat fungsi dequeue yang mengatur output dari queue menjadi tertib.

D. Tugas

Tugas 1

Buatlah program yang membaca skor siswa, yang mencari skor terbaik, dan menugaskan nilai berdasarkan skema berikut ini :

Nilai A jika skor >= Terbaik – 10;
Nilai B jika skor >= Terbaik – 20;
Nilai C jika skor >= Terbaik – 30;
Nilai D jika skor >= Terbaik – 40;
Nilai E jika skor >= Terbaik – 50;

Program meminta user meng-inputkan jumlah total siswa, semua skor, dan kemudian menyimpulkan nilai berdasar skema diatas.

❖ Mengamati Listing Program dan Output Program

Tentukan nama class yang akan digunakan.

Tentukan variabel yang akan digunakan.

Tentukan method yang akan digunakan.

Buatlah class diagram yang menggambarkan bagian-bagian dari class yang telah ditentukan. Contoh class diagram :

Nama class
Method
Operasi

Buatlah listing programnya.

Lakukan kompilasi dan debug pada program.

No	Output Program
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil program yang telah kamu buat dengan hasil program teman sebangku atau kelompok lain

Dari hasil perbandingan tersebut, hal penting apa yang harus dirumuskan secara bersama

E. Tes Formatif

Dalam test ini setiap anda harus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Sebutkan definisi dari istilah berikut ini :

Queue

Enqueue

Dequeue

Apakah dapat baris-baris didalam suatu array dua dimensi memiliki panjang yang berbeda?

Jelaskan apa yang dimaksud dengan collection ! Disebut dengan apa objek yang ada dalam collection?

Apa pengertian dari linked list? Sebutkan macam-macam dari linked list !

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Definisi dari istilah :



a)Queue.....
.....
.....
.....
.....

b) Enqueue
.....
.....
.....
.....

c)Dequeue
.....
.....
.....
.....

LJ- 02 : Dapatkah baris dalam array dua dimensi memiliki panjang berbeda :



.....
.....
.....
.....
.....

LJ- 03 : Yang dimaksud dengan collection :



.....
.....
.....

.....

.....

.....

LJ- 04: Definisi dari linked list :



.....

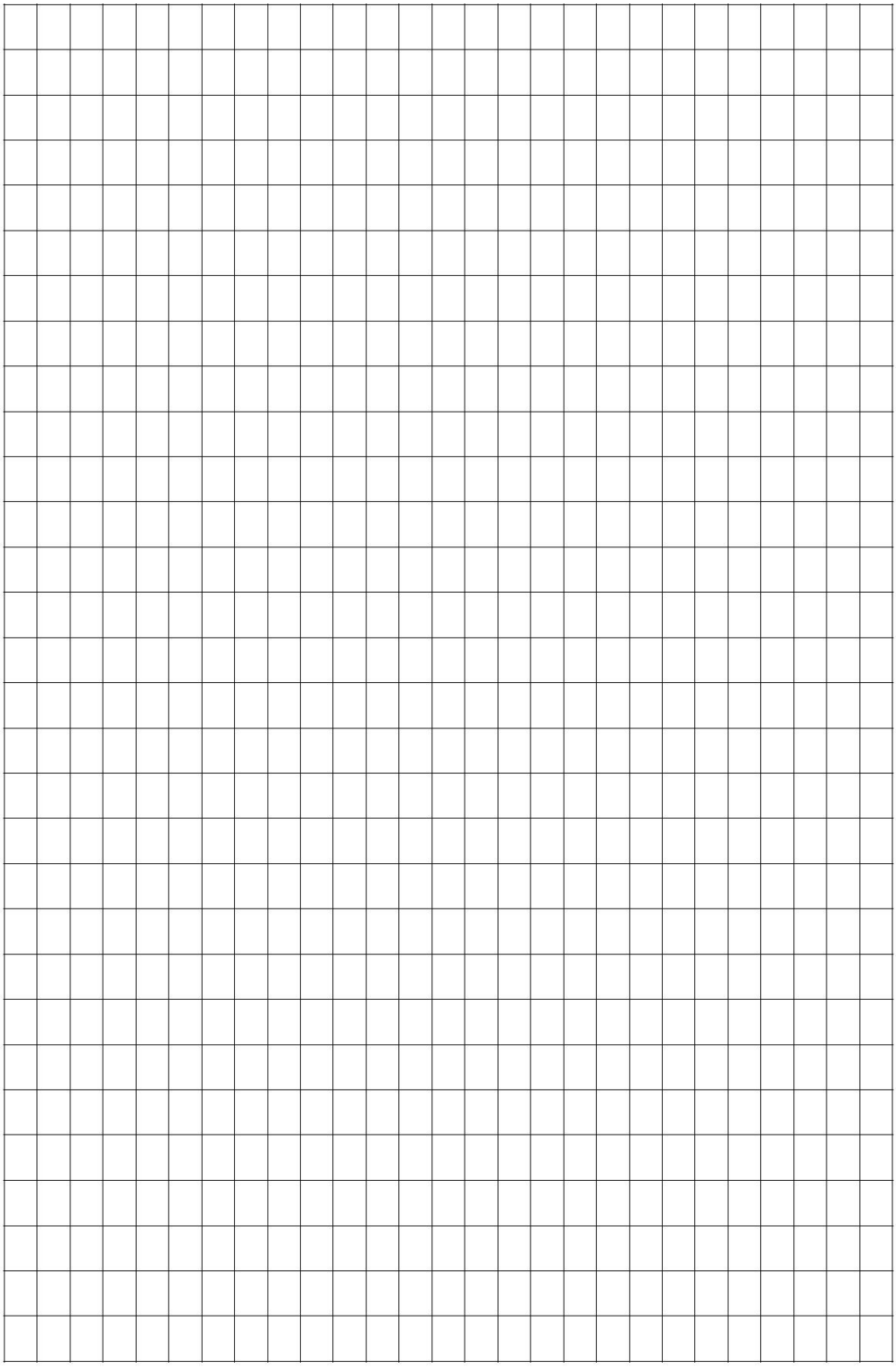
.....

.....

.....

.....

G. Lembar Kerja Siswa



6. Kegiatan 6 : Sistem File

A. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 6 ini siswa diharapkan dapat :

Menerapkan operasi file dan Input Output (IO)

Menyajikan operasi file dan operasi Input

B. Uraian Materi

Sistem File

Class dasar I/O Reader, Writer, InputStream dan OutputStream hanya menyediakan operasi I/O sangat dasar. Misalnya, class InputStream memiliki metode instansi**publicint** read() **throws** IOExceptionuntuk membaca satu byte data dari aliran input. Jika sampai pada akhir dari aliran input , metode read() akan mengembalikan nilai -1. Jika ada kesalahan yang terjadi pada saat pengambilan input, maka pengecualian IOException akan dilemparkan. Karena IOException adalah class pengecualian yang harus ditangani, artinya kita harus menggunakan metode read() di dalam penyataan **try** atau mengeset subrutin untuk **throws** IOException.

Class InputStream juga memiliki metode untuk membaca beberapa byte data dalam satu langkah ke dalam array **byte**. Akan tetapi InputStream tidak memiliki metode untuk membaca jenis data lain, seperti **int** atau **double** dari aliran. Ini bukan masalah karena dalam prakteknya kita tidak akan menggunakan objek bertipe InputStream secara langsung. Yang akan kita gunakan adalah class turunan dari InputStream yang memiliki beberapa metode input yang lebih beragam daripada InputStream itu sendiri.

Begitu juga dengan class OutputStream memiliki metode output primitif untuk menulis satu byte data ke aliran output, yaitu metode**publicvoid** write(**int** b) **throws** IOException, tapi kita hampir pasti akan menggunakan class turunannya yang mampu menangani operasi yang lebih kompleks.Class Reader dan Writer memiliki operasi dasar yang hampir sama, yaitu read dan write, akan tetapi class ini berorientasi karakter (karena digunakan untuk membaca dan menulis data

yang bisa dibaca manusia). Artinya operasi baca tulis akan mengambil dan menulis nilai **char** bukan byte. Dalam prakteknya kita akan menggunakan class turunan dari class-class dasar ini.

Salah satu hal menarik dari paket I/O pada Java adalah kemungkinan untuk menambah kompleksitas suatu aliran dengan membungkus aliran tersebut dalam objek aliran lain. Objek pembungkus ini juga berupa aliran, sehingga kita juga bisa melakukan baca tulis dari objek yang sama dengan tambahan kemampuan dalam objek pembungkusnya.Misalnya, PrintWriter adalah class turunan dari Writer yang memiliki metode tambahan untuk menulis tipe data Java dalam karakter yang bisa dibaca manusial. Jika kita memiliki objek bertipe Writer atau turunannya, dan kita ingin menggunakan metode pada PrintWriter untuk menulis data, maka kita bisa membungkus objek Writer dalam objek PrintWriter.

Contoh jika baskomKarakter bertipe Writer, maka kita bisa membuatPrintWriter printableBaskomKarakter = **new** PrintWriter(baskomKarakter);Ketika kita menulis data ke printableBaskomKarakter dengan menggunakan metode pada PrintWriter yang lebih canggih, maka data tersebut akan ditempatkan di tempat yang sama dengan apabila kita menulis langsung pada baskomKarakter. Artinya kita hanya perlu membuat antar muka yang lebih baik untuk aliran output yang sama. Atau dengan kata lain misalnya kita bisa menggunakan PrintWriter untuk menulis file atau mengirim data pada jaringan.

Untuk lengkapnya, metode pada class PrintWriter memiliki metode sebagai berikut :

Metode untuk menulis data dalam
bentuk yang bisa dibaca manusia

publicvoid print(String s)
publicvoid print(**char** c)
publicvoid print(**int** i) **publicvoid**
print(**long** l) **publicvoid** print(**float**
f) **publicvoid** print(**double** d)
publicvoid print(**boolean** b)

Menulis baris baru ke aliran
publicvoid println()
Metode ini sama dengan di atas akan tetapi keluarannya selalu ditambah dengan baris baru
publicvoid println(String s)
publicvoid println(**char** c)
publicvoid println(**int** i)
publicvoid println(**long** l)
publicvoid println(**float** f)
publicvoid println(**double** d
publicvoid println(**boolean** b)

Catatan bahwa metode-metode di atas tidak pernah melempar pengecualian IOException. Akan tetapi, class PrintWriter memiliki metode**publicboolean** checkError()yang akan mengembalikan true jika ada kesalahan yang terjadi ketika menulis ke dalam aliran. Class PrintWriter menangkap pengecualian IOException secara internal, dan mengeset nilai tertentu di dalam class ini jika kesalahan telah terjadi. Sehingga kita bisa menggunakan metode pada PrintWriter tanpa khawatir harus menangkap pengecualian yang mungkin terjadi. Akan tetapi, jika kita ingin membuat progam yang tangguh tentunya kita harus selalu memanggil checkError() untuk melihat apakah kesalahan telah terjadi ketika kita menggunakan salah satu metode pada PrintWriter.

Ketika kita menggunakan metode PrintWriter untuk menulis data ke aliran, data tersebut diubah menjadi rangkaian karakter yang bisa dibaca oleh manusia. Bagaimana caranya jika kita ingin membuat data dalam bentuk bahasa mesin?
Paket java.Io memiliki class aliran byte, yaitu DataOutputStream yang bisa digunakan untuk menulis suatu data ke dalam aliran dalam format biner. DataOutputStream berhubungan erat dengan OutputStream seperti hubungan antara PrintWriter dan Writer.

Artinya, OutputStream hanya berisi metode dasar untuk menulis byte, sedangkan DataOutputStream memiliki metode writeDouble(**double** x) untuk menulis nilai

double, writeInt(**int** x) untuk menulis nilai int, dan seterusnya. Dan juga kita bisa membungkus objek bertipe OutputStream atau turunannya ke dalam aliran DataOutputStream sehingga kita bisa menggunakan metode yang lebih kompleks.
Misalnya, jika baskomByte adalah variabel bertipe OutputStream, makaDataOutputStream baskomData = **new** DataOutputStream(baskomByte);untuk membungkus baskomByte dalam baskomData.Untuk mengambil data dari aliran, java.io memiliki class DataInputStream. Kita bisa membungkus objek bertipe InputStream atau turunannya ke dalam objek bertipe DataInputStream. Metode di dalam DataInputStream untuk membaca data biner bisa menggunakan readDouble(), readInt() dan seterusnya. Data yang ditulis oleh DataOutputStream dijamin untuk bisa dibaca kembali oleh DataInputStream, meskipun data kita tulis pada satu komputer dan data dibaca pada komputer jenis lain dengan sistem operasi berbeda. Kompatibilitas data biner pada Java adalah salah satu keunggulan Java untuk bisa dijalakan pada beragam platform.

Salah satu fakta yang menyedihkan tentang Java adalah ternyata Java tidak memiliki class untuk membaca data dalam bentuk yang bisa dibaca oleh manusia. Dalam hal ini Java tidak memiliki class kebalikan dari PrintWriter sebagaimana DataOutputStream dan DataInputStream. Akan tetapi kita tetap bisa membuat class ini sendiri dan menggunakannya dengan cara yang persis sama dengan class-class di atas.

Class PrintWriter, DataInputStream, dan DataOutputStream memungkinkan kita untuk melakukan input dan output semua tipe data primitif pada Java. Pertanyaannya bagaimana kita melakukan baca tulis suatu objek?Mungkin secara tradisional kita akan membuat fungsi sendiri untuk memformat objek kita menjadi bentuk tertentu, misalnya urutan tipe primitif dalam bentuk biner atau karakter kemudian disimpan dalam file atau dikirim melalui jaringan. Proses ini disebut serialisasi (serializing) objek.
Pada inputnya, kita harus bisa membaca data yang diserialisasi ini sesuai dengan format yang digunakan pada saat objek ini diserialisasi. Untuk objek kecil, pekerjaan semacam ini mungkin bukan masalah besar. Akan tetapi untuk

ukuran objek yang besar, hal ini tidak mudah.Akan tetapi Java memiliki cara untuk melakukan input dan output isi objek secaraotomatis,yaitu dengan menggunakan ObjectInputStream danObjectOutputStream.Class-class ini adalah class turunan dari InputStream danOutputStream yang bisa digunakan untuk membaca dan menulis objek yang sudah diserialisasi.ObjectInputStream dan ObjectOutputStream adalah class yang bisa dibungkus oleh class InputStream dan OutputStream lain. Artinya kita bisa melakukan input dan output objek pada aliran byte apa saja.Metde untuk objek I/O adalah readObject() yang tersedia pada ObjectInputStream dan writeObject(Object obj) yang tersedia dalam ObjectOutputStream. Keduanya bisa melemparkan IOException. Ingat bahwa readObject() mengembalikan nilai bertipe Object yang artinya harus di-type cast ke tipe sesungguhnya.ObjectInputStream dan ObjectOutputStream hanya bekerja untuk objek yang mengimplementasikan interface yang bernama Serializable. Lebih jauh semua variabel instansi pada objek harus bisa diserialisasi, karena interface Serializable tidak mempunyai metode apa-apa. Interface ini ada hanya sebagai penanda untuk kompiler supaya kompiler tahu bahwa objek ini digunakan untuk baca tulis ke suatu media.

Yang perlu kita lakukan adalah menambahkan "**implementsSerializable**" pada definisi class. Banyak class standar Java yang telah dideklarasikan untuk bisa diserialisasi, termasuk semua komponen class Swing dan AWT. Artinya komponen GUI pun bisa disimpan dan dibaca dari dalam perangkat I/O menggunakan ObjectInputStream dan ObjectOutputStream.

2) Stream, Reader, dan Writer

Tanpa bisa berinteraksi dengan dunia lain, suatu program tidak ada gunanya. Interaksi suatu program dengan dunia lain sering disebut input/output atau I/O. Sejak dulu, salah satu tantangan terbesar untuk mendesain bahasa pemrograman baru adalah mempersiapkan fasilitas untuk melakukan input dan output. Komputer bisa terhubung dengan beragam jenis input dan output dari berbagai perangkat. Jika bahasa pemrograman harus dibuat secara khusus untuk setiap jenis perangkat, maka kompleksitasnya akan tak lagi bisa ditangani. Salah satu kemajuan terbesar dalam sejarah pemrograman adalah adanya konsep (atau abstraksi) untuk memodelkan perangkat I/O. Dalam Java, abstraksi

ini disebut dengan aliran (stream). Bagian ini akan memperkenalkan tentang aliran, akan tetapi tidak menjelaskan dengan komplit. Untuk lebih lengkapnya, silakan lihat dokumen resmi Java.

Ketika berhubungan dengan input/output, kita harus ingat bahwa ada dua kategori data secara umum : data yang dibuat oleh mesin, dan data yang bisa dibaca manusia. Data yang dibuat mesin ditulis dengan model yang sama dengan bagaimana data tersebut disimpan di dalam komputer, yaitu rangkaian nol dan satu. Data yang bisa dibaca manusia adalah data dalam bentuk rangkaian huruf. Ketika kita membaca suatu bilangan 3.13159, kita membacanya sebagai rangkaian huruf yang kita terjemahkan sebagai angka. Angka ini akan ditulis dalam komputer sebagai rangkaian bit yang kita tidak mengerti.

Untuk menghadapi kedua jenis data ini, Java memiliki dua kategori besar untuk aliran : aliran byte untuk data mesin (byte stream), dan aliran karakter (character stream) untuk data yang bisa dibaca manusia. Ada banyak kelas yang diturunkan dari kedua kategori ini.Setiap objek yang mengeluarkandata ke aliran byte masuk sebagai kelas turunan dari kelas abstrak OutputStream. Objek yang membaca data dari aliran byte diturunkan dari kelas abstrak InputStream. Jika kita menulis angka ke suatu OutputStream, kita tidak akan bisa membaca data tersebut karena ditulis dalam bahasa mesin. Akan tetapi data tersebut bisa dibaca kembali oleh InputStream. Proses baca tulis data akan menjadi sangat efisien, karena tidak ada penerjemahan yang harus dilakukan : bit yang digunakan untuk menyimpan data di dalam memori komputer hanya dikopi dari dan ke aliran tersebut.

Untuk membaca dan menulis data karakter yang bisa dimengerti manusia, kelas utamanya adalah Reader dan Writer. Semua kelas aliran karakter merupakan kelas turunan dari salah satu dari kelas abstrak ini. Jika suatu angka akan ditulis dalam aliran Writer, komputer harus bisa menerjemahkannya ke dalam rangkaian karakter yang bisa dibaca maunsia.

Membaca angka dari aliran Reader menjadi variabel numerik juga harus diterjemahkan, dari deretan karakter menjadi rangkaian bit yang dimengerti

komputer. (Meskipun untuk data yang terdiri dari karakter, seperti dari editor teks, masih akan ada beberapa terjemahan yang dilakukan. Karakter disimpan dalam komputer dalam nilai Unicode 16-bit. Bagi orang yang menggunakan alfabet biasa, data karakter biasanya disimpan dalam file dalam kode ASCII, yang hanya menggunakan 8-bit. Kelas Reader dan Writer akan menangani perubahan dari 16-bit ke 8-bit dan sebaliknya, dan juga menangani alfabet lain yang digunakan negara lain.)

Adalah hal yang mudah untuk menentukan apakah kita harus menggunakan aliran byte atau aliran karakter. Jika kita ingin data yang kita baca/tulis untuk bisa dibaca manusia, maka kita gunakan aliran karakter. Jika tidak, gunakan aliran byte. System.in dan System.out sebenarnya adalah aliran byte dan bukan aliran karakter, karenanya bisa menangani input selain alfabet, misalnya tombol enter, tanda panah, escape, dsb.

Kelas aliran standar yang akan dibahas berikutnya didefinisikan dalam paket java.io beserta beberapa kelas bantu lainnya. Kita harus mengimpor kelas-kelas tersebut dari paket ini jika kita ingin menggunakannya dalam program kita. Artinya dengan menggunakan "import java.io.*" di awal kode sumber kita.
Aliran tidak digunakan dalam GUI, karena GUI memiliki aliran I/O tersendiri. Akan tetapi kelas-kelas ini digunakan juga untuk file atau komunikasi dalam jaringan. Atau bisa juga digunakan untuk komunikasi antar thread yang sedang bekerja secara bersamaan. Dan juga ada kelas aliran yang digunakan untuk membaca dan menulis data dari dan ke memori komputer.

C. Rangkuman

Class dasar I/O (input output) Reader, Writer, InputStream dan OutputStream hanya menyediakan operasi I/O sangat dasar. Misalnya, class InputStream memiliki metode instansi publicint read() throws IOException untuk membaca satu byte data dari aliran input. Jika sampai pada akhir dari aliran input , metode read() akan mengembalikan nilai -1. Jika ada kesalahan yang terjadi pada saat pengambilan input, maka pengecualian IOException akan dilemparkan.

Class InputStream memiliki metode untuk membaca byte data dalam satu langkah ke dalam array byte. Tetapi InputStream tidak memiliki metode untuk membaca jenis data lain, seperti int atau double dari aliran.

Class OutputStream memiliki metode output primitif untuk menulis satu byte data ke aliran output, yaitu metode publicvoid write(int b) throws IOException, tapi kita hampir pasti akan menggunakan class turunannya yang mampu menangani operasi yang lebih kompleks.

Class Reader dan Writer memiliki operasi dasar yang samayaitu read dan write, akan tetapi class ini berorientasi karakter. Artinya operasi baca tulis akan mengambil dan menulis nilai char bukan byte.

PrintWriter adalah class turunan dari Writer yang memiliki metode tambahan untuk menulis tipe data Java dalam karakter yang bisa dibaca manusia.Jika kita memiliki objek bertipe Writer atau turunannya, dan kita ingin menggunakan metode pada PrintWriter untuk menulis data, maka kita bisa membungkus objek Writer dalam objek PrintWriter.

Class PrintWriter menangkap pengecualian IOException secara internal, dan mengeset nilai tertentu di dalam class ini jika kesalahan telah terjadi.Ketika kita menggunakan metode PrintWriter untuk menulis data ke aliran, data tersebut diubah menjadi rangkaian karakter yang bisa dibaca oleh manusia.

Untuk membaca dan menulis data karakter yang dapat dimengerti manusia, kelas utamanya adalah Reader dan Writer.Semua kelas aliran karakter merupakan kelas turunan dari salah satu dari kelas abstrak ini. Jika suatu angka akan ditulis dalam aliran Writer, komputer harus bisa menerjemahkannya ke dalam rangkaian karakter yang bisa dibaca maunsia.

D. Tugas

Tugas 1

Buatlah program untuk menampilkan data seperti dibawah ini :

```
Rismon H Sianipar 90
Rebecca V Siahaan 85
```

Buat program menggunakan `java.util.Scanner`, dengan tipe data String dan int. dalam program tersebut, user harus memasukkan datanya, lalu program menampilkan hasil dari inputan user.

❖ Mengamati Listing Program dan Output Program

Tentukan nama class yang akan digunakan.

Tentukan variabel yang akan digunakan.

Tentukan method yang akan digunakan.

Buatlah class diagram yang menggambarkan bagian-bagian dari class yang telah ditentukan. Contoh class diagram :

Nama class
Method
Operasi

Buatlah listing programnya.

Lakukan kompilasi dan debug pada program.

No	Output Program
1.	
2.	
3.	
4.	

5.	
6.	
7.	
8.	
9.	
10.	

❖ Bandingkan dan Simpulkan

Bandingkan hasil program yang telah kamu buat dengan hasil program teman sebangku atau kelompok lain

Dari hasil perbandingan tersebut, hal penting apa yang harus dirumuskan secara bersama

E. Tes Formatif

Dalam test ini setiap anda harus membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Apa yang dimaksud dengan kelas PrintWriter ?

Jelaskan pengertian metode public boolean pada class PrintWriter !

Apa yang kamu ketahu tentang kelas dasar IO Reader, Writer, InputStream, dan OutputStream ?

Jelaskan metode-metode pembacaan jenis data pada Class InputStream !

Apa pengertian dari ObjectInputStream dan ObjectOutputStream ?

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : Pengertian dari kelas PrintWriter :



.....
.....
.....
.....
.....
.....

LJ- 02 : Pengertian metode public Boolean pada class PrintWriter :



.....
.....
.....
.....
.....
.....
.....
.....

LJ- 03 : kelas dasar IO Reader, Writer, InputStream, dan OutputStream :



.....
.....
.....
.....
.....
.....
.....

LJ- 04: Metode-metode pembacaan jenis data pada Class InputStream :

Pengertian ObjectInputStream dan ObjectOutputStream :



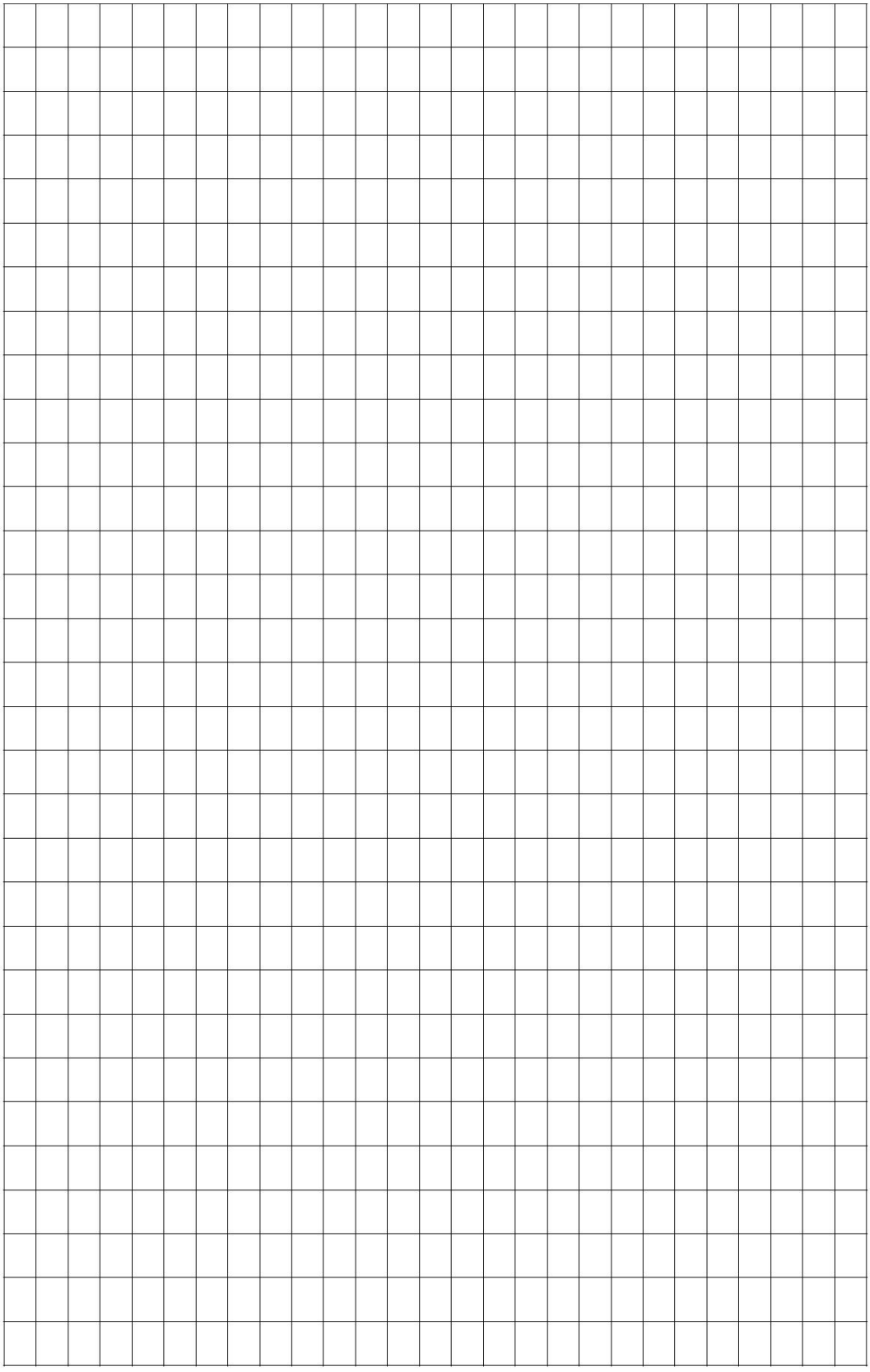
.....
.....
.....
.....
.....

LJ- 05: Pengertian ObjectInputStream dan ObjectOutputStream :



.....
.....
.....
.....
.....

G. Lembar Kerja Siswa



DAFTAR PUSTAKA

Christian Musnter, Java 2 JDK 5 –Grundlagen , Herdt – Verlag forbildungsmedien Gagh, Bodenheilm, 2006

C. Thomas wu, An Introduction to Object- Oriented Programming with Java, McGraw Hill 2001

Deitel, Java : How to program, Prentice Hall, New jersey, 2002

Joyce Avestro , Introduction Programming 1, Java Education Development Initiatif, 2003

Joyce Avestro , Introduction Programming 2, Java Education Development Initiatif, 2003

Patric Noughton , The Java Handbook, McGrawHill, Inc, 2006

R.H.Sianipar , Teori dan Implementasi Java, Informatika Bandung,2013

Kementerian Pendidikan dan Kebudayaan
Republik Indonesia
2013

HALAMAN SAMPUL



Pemrograman Web 1

UNTUK SMK / MAK KELAS X



Penulis	**: Wahyu Purnomo**
Editor Materi	**: Kadek S**
Editor Bahasa	**:**
Ilustrasi Sampul	**:**
Desain & Ilustrasi Buku	**: PPPPTK BOE Malang**

Hak Cipta © 2013, Kementerian Pendidikan & Kebudayaan

MILIK NEGARA
TIDAK DIPERDAGANGKAN

Semua hak cipta dilindungi undang-undang.

Dilarang memperbanyak (mereproduksi), mendistribusikan, atau memindahkan sebagian atau seluruh isi buku teks dalam bentuk apapun atau dengan cara apapun, termasuk fotokopi, rekaman, atau melalui metode (media) elektronik atau mekanis lainnya, tanpa izin tertulis dari penerbit, kecuali dalam kasus lain, seperti diwujudkan dalam kutipan singkat atau tinjauan penulisan ilmiah dan penggunaan non-komersial tertentu lainnya diizinkan oleh perundangan hak cipta. Penggunaan untuk komersial harus mendapat izin tertulis dari Penerbit.

Hak publikasi dan penerbitan dari seluruh isi buku teks dipegang oleh Kementerian Pendidikan & Kebudayaan.

Untuk permohonan izin dapat ditujukan kepada Direktorat Pembinaan Sekolah Menengah Kejuruan, melalui alamat berikut ini:

Pusat Pengembangan & Pemberdayaan Pendidik & Tenaga Kependidikan Bidang Otomotif & Elektronika:

Jl. Teluk Mandar, Arjosari Tromol Pos 5, Malang 65102, Telp. (0341) 491239, (0341) 495849, Fax. (0341) 491342, Surel: vedcmalang@vedcmalang.or.id, Laman: www.vedcmalang.com

DISKLAIMER (DISCLAIMER)

Penerbit tidak menjamin kebenaran dan keakuratan isi/informasi yang tertulis di dalam buku tek ini. Kebenaran dan keakuratan isi/informasi merupakan tanggung jawab dan wewenang dari penulis.

Penerbit tidak bertanggung jawab dan tidak melayani terhadap semua komentar apapun yang ada didalam buku teks ini. Setiap komentar yang tercantum untuk tujuan perbaikan isi adalah tanggung jawab dari masing-masing penulis.

Setiap kutipan yang ada di dalam buku teks akan dicantumkan sumbernya dan penerbit tidak bertanggung jawab terhadap isi dari kutipan tersebut. Kebenaran keakuratan isi kutipan tetap menjadi tanggung jawab dan hak diberikan pada penulis dan pemilik asli. Penulis bertanggung jawab penuh terhadap setiap perawatan (perbaikan) dalam menyusun informasi dan bahan dalam buku teks ini.

Penerbit tidak bertanggung jawab atas kerugian, kerusakan atau ketidaknyamanan yang disebabkan sebagai akibat dari ketidakjelasan, ketidaktepatan atau kesalahan didalam menyusun makna kalimat didalam buku teks ini.

Kewenangan Penerbit hanya sebatas memindahkan atau menerbitkan mempublikasi, mencetak, memegang dan memproses data sesuai dengan undang-undang yang berkaitan dengan perlindungan data.

Katalog Dalam Terbitan (KDT)

Teknik Komunikasi Informasi Edisi Pertama 2013
Kementerian Pendidikan & Kebudayaan
Direktorat Jenderal Peningkatan Mutu Pendidik & Tenaga Kependidikan,
th. 2013: Jakarta



KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan yang Maha Esa atas tersusunnya buku teks ini, dengan harapan dapat digunakan sebagai buku teks untuk siswa Sekolah Menengah Kejuruan (SMK) Bidang Studi Keahlian Teknologi Informasi dan Komunikasi, Program Keahlian Teknik Komputer dan Informatika.

Penerapan kurikulum 2013 mengacu pada paradigma belajar kurikulum abad 21 menyebabkan terjadinya perubahan, yakni dari pengajaran (*teaching*) menjadi BELAJAR (*learning*), dari pembelajaran yang berpusat kepada guru (*teachers-centered*) menjadi pembelajaran yang berpusat kepada peserta didik (*student-centered*), dari pembelajaran pasif (*pasive learning*) ke cara belajar peserta didik aktif (*active learning-CBSA*) atau *Student Active Learning-SAL*.

Buku teks Pemrograman Web ini disusun berdasarkan tuntutan paradigma pengajaran dan pembelajaran kurikulum 2013 diselaraskan berdasarkan pendekatan model pembelajaran yang sesuai dengan kebutuhan belajar kurikulum abad 21, yaitu pendekatan model pembelajaran berbasis peningkatan keterampilan proses sains.

Penyajian buku teks untuk Mata Pelajaran Pemrograman Web ini disusun dengan tujuan agar supaya peserta didik dapat melakukan proses pencarian pengetahuan berkenaan dengan materi pelajaran melalui berbagai aktivitas proses sains sebagaimana dilakukan oleh para ilmuwan dalam melakukan eksperimen ilmiah (penerapan scientifik), dengan demikian peserta didik diarahkan untuk menemukan sendiri berbagai fakta, membangun konsep, dan nilai-nilai baru secara mandiri.

Kementerian Pendidikan dan Kebudayaan, Direktorat Pembinaan Sekolah Menengah Kejuruan, dan Direktorat Jenderal Peningkatan Mutu Pendidik dan Tenaga Kependidikan menyampaikan terima kasih, sekaligus saran kritik demi kesempurnaan buku teks ini dan penghargaan kepada semua pihak yang telah berperan serta dalam membantu terselesaikannya buku teks siswa untuk Mata Pelajaran Pemrograman Web kelas X /Semester 1 Sekolah Menengah Kejuruan (SMK).

Jakarta, 12 Desember 2013
Menteri Pendidikan dan Kebudayaan

Prof. Dr. Mohammad Nuh, DEA

DAFTAR ISI

HALAMAN SAMPUL....i
DISKLAIMER (DISCLAIMER)....ii
KATA PENGANTAR....iii
DAFTAR ISI....iv
GLOSARIUM....ix
PETA KEDUDUKAN BAHAN AJAR....x
A. Kegiatan Belajar....1
1. Kegiatan Belajar 1: Profesi dalam Pengembangan Aplikasi Web 1
a. Tujuan Pembelajaran....1
b. Uraian Materi....1
c. Rangkuman....17
d. Tugas....17
e. Tes Formatif....18
f. Lembar Jawaban Tes Formatif....18
g. Lembar Kerja Siswa....18
2. Kegiatan Belajar 2: Alur dan Perangkat Pengembangan Aplikasi Web....18
a. Tujuan Pembelajaran....18
b. Uraian Materi....18
c. Rangkuman....31
d. Tugas....31
e. Tes Formatif....31
f. Lembar Jawaban Tes Formatif....31
g. Lembar Kerja Siswa....31
3. Kegiatan Belajar 3 : Anatomi dokumen web....32
a. Tujuan Pembelajaran....32
b. Uraian Materi....32
c. Rangkuman....39

d. Tugas 40
e. Tes Formatif 43
f. Lembar Jawaban Tes Formatif 43
g. Lembar Kerja Siswa 43
4. Kegiatan Belajar 4 : Pemformatan Teks dan Paragraf Web 44
a. Tujuan Pembelajaran 44
b. Uraian Materi 44
c. Rangkuman 46
d. Tugas 46
e. Tes Formatif 50
f. Lembar Jawaban Tes Formatif 51
g. Lembar Kerja Siswa 51
5. Kegiatan Belajar 5 : Pembuatan List Minimal 51
a. Tujuan Pembelajaran 51
b. Uraian Materi 52
c. Rangkuman 56
d. Tugas 56
e. Tes Formatif 56
f. Lembar Jawaban Tes Formatif 56
g. Lembar Kerja Siswa 56
6. Kegiatan Belajar 6 : Pembuatan List Kombinasi 57
a. Tujuan Pembelajaran 57
b. Uraian Materi 57
c. Rangkuman 60
d. Tugas 60
e. Tes Formatif 60
f. Lembar Jawaban Tes Formatif 60
g. Lembar Kerja Siswa 60
7. Kegiatan Belajar 7 : Pemformatan Tabel 61
a. Tujuan Pembelajaran 61
b. Uraian Materi 61

c. Rangkuman....63
d. Tugas....63
e. Tes Formatif....67
f. Lembar Jawaban Tes Formatif....67
g. Lembar Kerja Siswa....67
8. Kegiatan Belajar 8 : Tabel dengan Spaning....67
a. Tujuan Pembelajaran....67
b. Uraian Materi....67
c. Rangkuman....69
d. Tugas....69
e. Tes Formatif....72
f. Lembar Jawaban Tes Formatif....72
g. Lembar Kerja Siswa....72
9. Kegiatan Belajar 9 : Pembuatan Tabel dalam Tabel....73
10. Kegiatan Belajar 10 : Desain Halaman Web dengan Konsep Tabel 74
11. Kegiatan Belajar 11 : Format Tampilan Gambar....75
a. Tujuan Pembelajaran....75
b. Uraian Materi....75
c. Rangkuman....77
d. Tugas....78
e. Tes Formatif....86
f. Lembar Jawaban Tes Formatif....86
g. Lembar Kerja Siswa....86
12. Kegiatan Belajar 1 2 : Format Tampilan Audio....87
13. Kegiatan Belajar 13 : Format Tampilan Video dan Animasi....88
14. Kegiatan Belajar 14 : Format Tampilan Gambar dengan Map 89
15. Kegiatan Belajar 15 : Anatomi Hyperlink....90
a. Tujuan Pembelajaran....90
b. Uraian Materi....90
c. Rangkuman....91
d. Tugas....92



e. Tes Formatif 93
f. Lembar Jawaban Tes Formatif 94
g. Lembar Kerja Siswa 94
16. Kegiatan Belajar 16 : Jenis – Jenis Link dalam HTML 95
a. Tujuan Pembelajaran 95
b. Uraian Materi 95
c. RANGKUMAN 97
d. Tugas 97
e. Tes Formatif 101
f. Lembar Jawaban Tes Formatif 101
g. Lembar Kerja Siswa 102
17. Kegiatan Belajar 17 : Link ke Bagian Dokumen HTML 103
a. Tujuan Pembelajaran 103
b. Uraian Materi 103
c. Rangkuman 104
d. Tugas 104
e. Tes Formatif 108
f. Lembar Jawaban Tes Formatif 108
g. Lembar Kerja Siswa 108
18. Kegiatan Belajar 18 : Format Target Link, Email dan Telepon 109
a. Tujuan Pembelajaran 109
b. Uraian Materi 109
c. Rangkuman 110
d. Tugas 110
e. Tes Formatif 111
f. Lembar Jawaban Tes Formatif 111
g. Lembar Kerja Siswa 111
19. Kegiatan Belajar 19 : Anatomi dan cara kerja form 112
a. Tujuan Pembelajaran. 112
b. Uraian materi. 112
c. Rangkuman 118

d. Tugas....119
e. Test Formatif....119
f. Lembar Jawaban Test Formatif (LJ)....119
g. Lembar Kerja Peserta Didik....120
20. Kegiatan Belajar 20: Pemformatan form....122
a. Tujuan Pembelajaran....122
b. Uraian materi....122
c. Rangkuman....126
d. Tugas :....126
e. Test Formatif....126
f. Lembar Jawaban Test Formatif (LJ)....126
g. Lembar Kerja Peserta Didik....127
DAFTAR PUSTAKA....129



GLOSARIUM

CERN adalah Organisation Europeenne pour la Recherche Nucleaire, Pusat Penelitian Fisika Partikel Eropa.

Domain adalah Pola penamaan host di Internet.

HTML adalah Hyper Text Markup Language, sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web.

HTTP adalah Hyper Text Transfer Protocol, tata cara komunikasi untuk mentransfer Web dari server ke komputer lain.

ISP adalah Interrnet Service Provider, perusahaan pemberi jasa akses Internet.

Server adalah komputer yang memberikan layanan di jaringan.

Tag HTML adalah kata kunci (nama tag) yang dikelilingi oleh kurung sudut seperti <html>

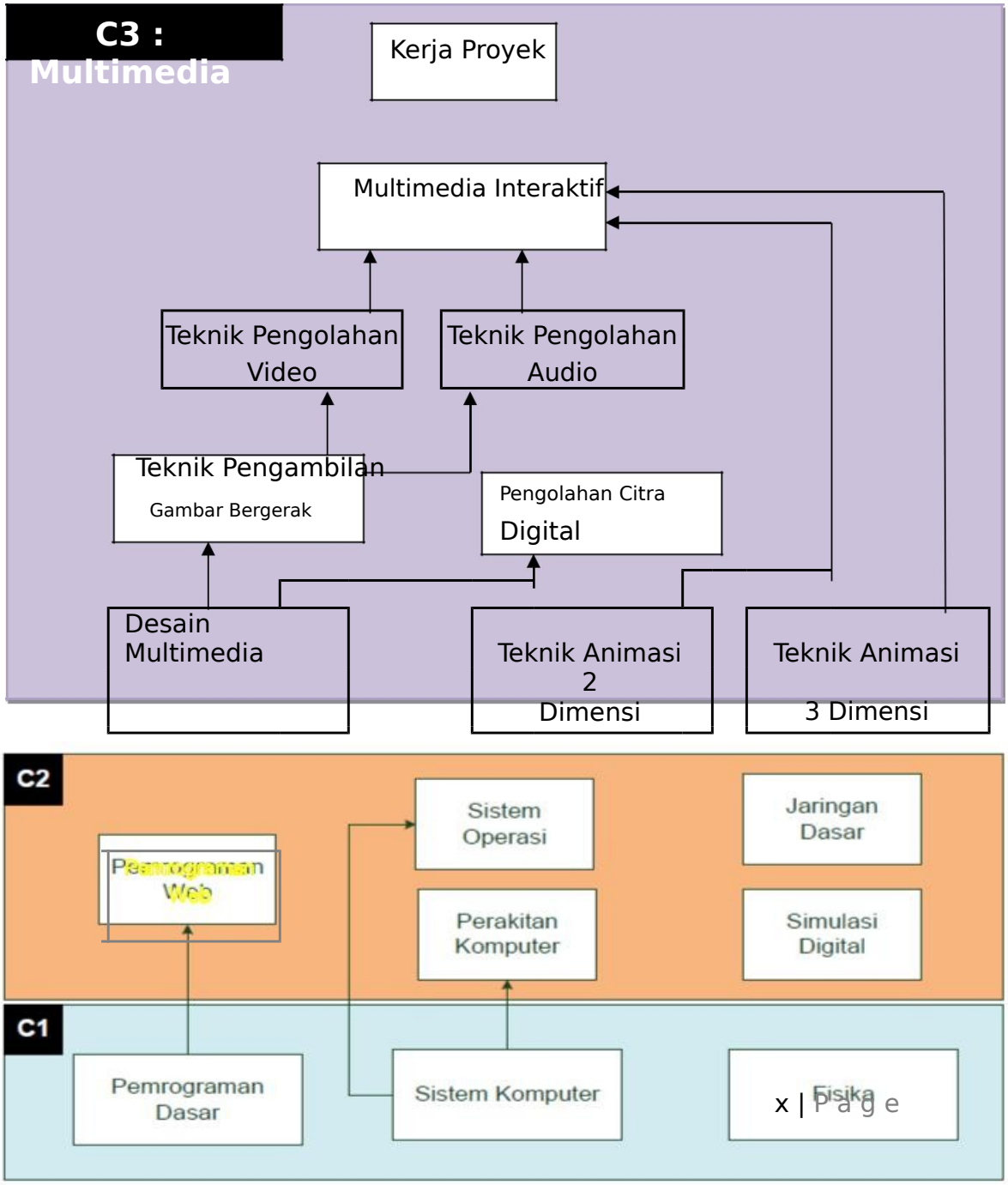
W3C adalah World Wide Web Consortium organisasi yang mengembangkan teknologi web beralamat di www.w3c.org.

Web browser adalah sebuah perangkat lunak aplikasi yang memungkinkan pengguna mengakses Web atau Server di jaringan.

WWW; World Wide Web sering di singkat Web.

PETA KEDUDUKAN BAHAN AJAR

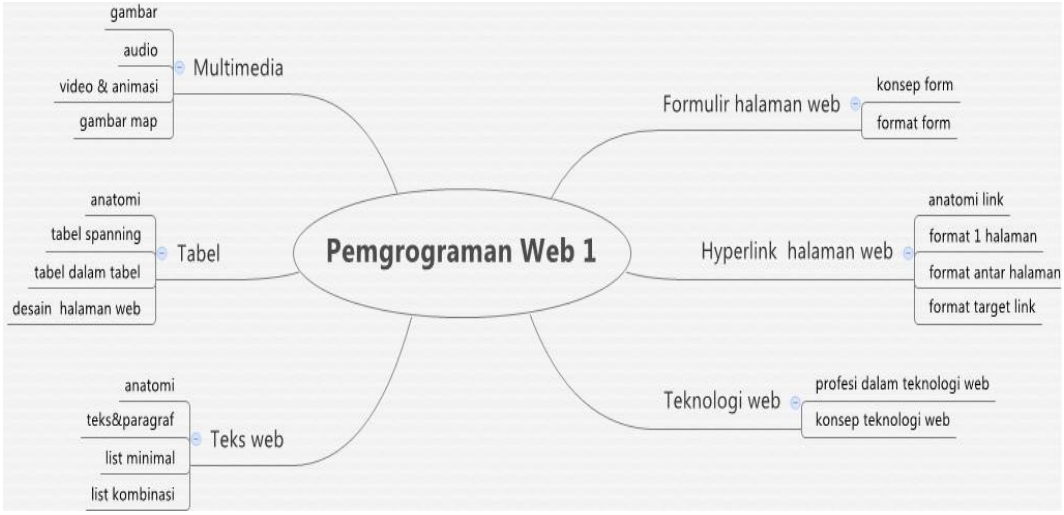
Peta kedudukan bahan ajar merupakan suatu diagram yang menjelaskan struktur mata pelajaran dan keterkaitan antar mata pelajaran dalam satu kelompok bidang studi keahlian. Gambar 1 menjelaskan peta kedudukan bahan ajar untuk program studi keahlian Multimedia. Kelompok C1 merupakan kelompok mata pelajaran wajib dasar bidang studi keahlian. C2 merupakan kelompok mata pelajaran wajib dasar program keahlian dan C3 merupakan kelompok mata pelajaran wajib paket keakhlian.



Gambar 1. Peta Kedudukan Bahan Ajar Kelompok C3

Mata Pemrograman Web

Sementara itu peta konsep mata pelajaran menjelaskan struktur urutan kegiatan belajar dan topik materi pelajaran. Gambar 2 dibawah ini menjelaskan peta konsep mata pelajaran Pemrograman web untuk kelas X semester 1.



Gambar 2. Peta Konsep Mata Pelajaran Pemrograman Web Kelas X Semester 1



A. Kegiatan Belajar

1. Kegiatan Belajar 1: Profesi dalam Pengembangan Aplikasi Web

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 1 ini siswa diharapkan dapat :

Memahami profesi dalam bidang Teknologi Informasi

Memahami profesi dalam pengembangan aplikasi web

Memahami sejarah web

Memahami cara kerja web

b. Uraian Materi

Profesi dalam bidang Teknologi Informasi

Saat ini ada banyak aneka profesi di bidang IT atau Teknologi Informasi. Perkembangan dunia IT telah melahirkan bidang baru yang tidak terlepas dari tujuan utamanya yaitu untuk semakin memudahkan manusia dalam melakukan segala aktifitas. Munculnya bidang IT yang baru juga memunculkan profesi di bidang IT yang semakin menjurus sesuai dengan keahlian masing-masing.

Secara umum, pekerjaan di bidang teknologi informasi setidaknya dapat dikelompokan sesuai bidangnya, misalnya.

a. Kelompok pertama, adalah mereka yang bergelut di dunia perangkat lunak (software), baik mereka yang merancang sistem operasi, database maupun sistem aplikasi.

Pada lingkungan kelompok ini, terdapat pekerjaan-pekerjaan seperti :

*Sistem analis, merupakan orang yang bertugas menganalisa system yang akan diimplementasikan, mulai dari menganalisa system yang ada, kelebihan dan kekurangannya, sampai studi kelayakan dan desain system yang akan dikembangkan.

Programer, merupakan orang yang bertugas mengimplementasikan rancangan system analis, yaitu membuat program (baik aplikasi maupun system operasi) sesuai system yang dianalisa sebelumnya.

Web designer, merupakan orang yang melakukan kegiatan perencanaan, termasuk studi kelayakan, analisis dan desain terhadap suatu proyek pembuatan aplikasi berbasis web.
Web programmer, merupakan orang yang bertugas mengimplementasikan rancangan web designer, yaitu membuat program berbasis web sesuai desain yang telah dirancang sebelumnya.

b. Kelompok kedua, adalah mereka yang bergelut di bidang perangkat keras (*hardware*).

Pada lingkungan kelompok ini, terdapat pekerjaan-pekerjaan seperti :
Technical engineer, sering juga disebut teknisi, yaitu orang yang berkecimpung dalam bidang teknik, baik mengenai pemeliharaan maupun perbaikan perangkat system computer.
Networking engineer, adalah orang yang berkecimpung dalam bidang teknis jaringan computer dari maintenance sampai pada troubleshooting-nya.

c. Kelompok ketiga, adalah mereka yang berkecimpung dalam operasional system informasi.

Pada lingkungan kelompok ini, terdapat pekerjaan-pekerjaan seperti :
*EDP Operator, adalah orang yang bertugas mengoperasikan program-program yang berhubungan dengan electronic data processing dalam lingkungan sebuah perusahaan atau organisasi lainnya.
*System Administrator, merupakan orang yang bertugas melakukan administrasi terhadap system, memiliki kewenangan menggunakan hak akses terhadap system, serta hal-hal lain yang berhubungan dengan pengaturan operasional sebuah system.

.

2) Profesi dalam pengembangan aplikasi web

Ada banyak ragam dalam profesi di dalam bidang Teknologi Informasi, bagaimana dengan profesi yang berada dalam lingkungan pengembangan



aplikasi web? Berikut ini adalah profesi-profesi yang langsung terkait dalam pengembangan aplikasi web, diantaranya :

Web Designer

Seorang desainer Web adalah orang yang bertanggung jawab untuk menentukan tampilan sebuah website. Tugasnya adalah pendisainan tampilan situs (web) mulai dari pengolahan gambar, tata letak, warna, dan semua aspek visual situs. Fokus utama mereka adalah tampilan / layout dari web. Mereka lebih konsen dengan bagaimana halaman terlihat dan apakah berfungsi sempurna ketika sudah diberikan bahasa pemrograman. Didalam pendandanan suatu situs seorang Web Designer harus menguasai :

HTML, DHTML

Pengolah Gambar

Animasi, Movie (Film)

Web Programmer

Web Programmer bertugas dalam melakukan pengcodingan atau pemograman sebuah website agar dinamis. dimana agar sebuah web tersebut dapat telihat mudah bagi seorang web admin.

Jika situs yang akan dibuat mempunya fasilitas interaksi antara pengunjung dan situs misalnya menyangkut dengan transaksi, input output data dan database maka seorang Web Programmer yang akan mengerjakannya dengan membuat aplikasi-aplikasi yang berkerja diatas situs (web). Penguasaan yang biasanya harus dikuasai pada umumnya oleh Web Programmer :

CGI Perl, PHP, MySQL (Unix base)

ASP (NT base)

Java Script dan Applet

Web Administrator

Tugasnya adalah untuk memaintenance suatu server, mengerti akan Sistem Operasi Server, baik itu mulai dari instalasi sampai kepada masalah (troubleshooting), biasanya seorang Web Administrator harus menguasai :
OS Unix (LInux, FreeBSD, dll)
OS NT
Jaringan (LAN, WAN, Intranet)
Keamanan Server

Web Master

Seoarang Web Master adalah seorang yang mengerti akan kesemua hal mulai dari disain, program dan keamanan server namun tidak terlalu turut mencampuri ke masing-masing divisi, cukup dengan mempertanggun jawabkan atas jalannya suatu situs (web). Penguasaan yang harus dimiliki :
HTML, DHTML
CGI Perl, PHP, MySQL, ASP, Java
Penguasaan bermacam OS (Operating System)
Keamanan Server
Jaringan (LAN, WAN, Intranet)

Web Developer

Kegiatan diatas secara keseluruhan dinamakan suatu team yang dinamakan Web Developer.
Web developer memberi bantuan seperti konsultasi web, konsep web yang akan di buat, membangun sebuat website..

Dari semua jenis pekerjaan diatas tidak semua adalah sebuah profesi karena tidak semua orang ahli dalam bidang tersebut. Yang bisa dikatakan sebagai sebuah profesi yaitu jika seseorang sudah ahli di dalam bidang pekerjaan tersebut. Jika di pekerjaan diatas yang bisa dikatakan sebagai sebuah profesi adalah web designer, web programmer, web administrator, web master dan web developer karena dalam bidang tersebut seseorang memang sudah memiliki keahlian di dalamnya.



3) Sejarah web

Internet adalah jaringan komputer yang saling terhubung. Tidak ada perusahaan yang memiliki internet, yang merupakan upaya kerja sama diatur oleh sistem standar dan aturan. Tujuan dari menghubungkan komputer bersama-sama, tentu saja, adalah untuk berbagi informasi. Ada banyak cara informasi dapat dikirimkan antar komputer, termasuk email, transfer file (*File Transfer Protocol*), dan banyak layanan yang lebih khusus yang dibangun dalam layanan Internet. Metode ini standar untuk mentransfer data atau dokumen melalui jaringan dikenal sebagai protokol.

World Wide Web, biasa lebih terkenal disingkat sebagai WWW adalah suatu ruang informasi yang dipakai oleh pengenal global yang disebut URL (Uniform Resource Locator) untuk mengenal pasti sumber daya berguna. WWW sering dianggap sama dengan Internet secara keseluruhan, walaupun sebenarnya ia hanyalah bagian daripada Internet.

merupakan kumpulan *web server* dari seluruh dunia yang mempunyai kegunaan untuk menyediakan data dan informasi untuk dapat digunakan bersama. WWW adalah bagian yang paling menarik dari Internet. Melalui web, para pengguna dapat mengakses informasi-informasi yang tidak hanya berupa teks tetapi bisa juga berupa gambar, suara, video dan animasi.

Kegunaan ini tergolong masih baru dibandingkan surat elektronik, sebenarnya merupakan kumpulan dokumen yang tersimpan di peladen web, dan yang peladennya tersebar di lima benua termasuk Indonesia yang terhubung menjadi satu melalui jaringan Internet. Dokumen-dokumen informasi ini disimpan atau dibuat dengan format HTML (Hypertext Markup Language).

Suatu halaman dokumen informasi dapat terdiri atas teks yang saling terkait dengan teks lainnya atau bahkan dengan dokumen lain. Keterkaitan halaman lewat teks ini disebut pranala. Dokumen infomasi ini tidak hanya terdiri dari teks tetapi dapat juga berupa gambar, mengandung suara bahkan klip video. Kaitan antar-dokumen yang seperti itu biasa disebut hipermedia.

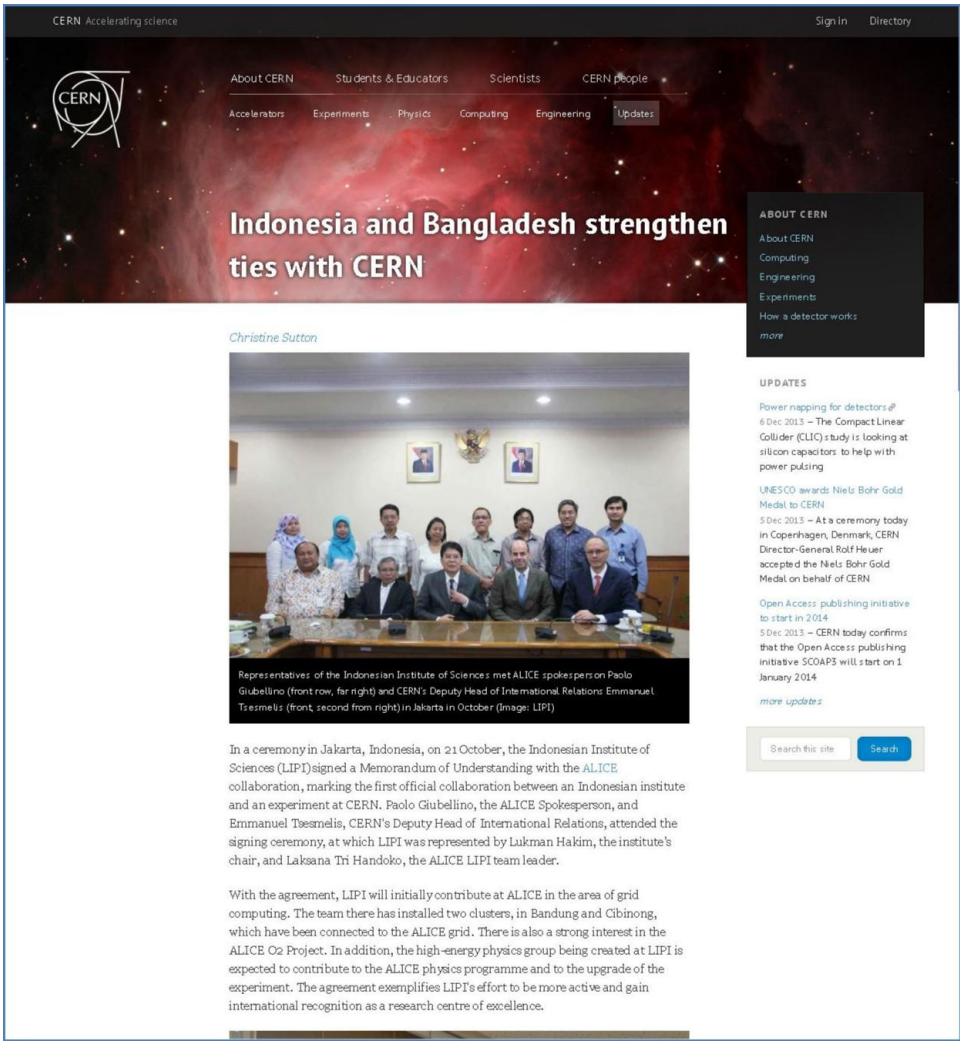
Jadi dapat disimpulkan bahwa WWW adalah sekelompok dokumen multimedia yang saling bertautan dengan menggunakan tautan hiperteks. Dengan mengeklik pranala (*hyperlink*), maka para pengguna bisa berpindah dari satu dokumen ke dokumen lainnya.

adalah suatu program yang ditemukan oleh Tim Berners-Lee pada tahun 1991. Awalnya Berners-Lee hanya ingin menemukan cara untuk menyusun arsip-arsip risetnya. Untuk itu, beliau mengembangkan suatu sistem untuk keperluan pribadi. Sistem itu adalah program peranti lunak yang diberi nama Enquire. Dengan program itu, Berners-Lee berhasil menciptakan jaringan yang menautkan berbagai arsip sehingga memudahkan pencarian informasi yang dibutuhkan. Inilah yang kelak menjadi dasar dari sebuah perkembangan pesat yang dikenal sebagai WWW.



Gambar. Tim Berners-Lee (Sumber : http://www.hdwallpapersinn.com)

dikembangkan pertama kali di Pusat Penelitian Fisika Partikel Eropa (CERN : Organisation Europeenne pour la Recherche Nucleaire), Jenewa, Swiss. Pada tahun 1989 Berners-lee membuat pengajuan untuk proyek pembuatan hiperteks global, kemudian pada bulan Oktober 1990, 'World Wide Web' sudah dapat dijalankan dalam lingkungan CERN. Pada musim panas tahun 1991, WWW secara resmi digunakan secara luas pada jaringan Internet.



Gambar. Website CERN http://home.web.cern.ch

Standart Web

Banyak standar formal dan spesifikasi teknis lainnya dan perangkat lunak mendefinisikan operasi dari aspek yang berbeda dari World Wide Web, Internet, dan pertukaran informasi komputer. Banyak dokumen adalah karya dari World Wide Web Consortium (W3C), dipimpin oleh Berners - Lee, namun ada juga yang diproduksi oleh Internet Engineering Task Force (IETF) dan organisasi lainnya.

Biasanya, ketika standar web dibahas, publikasi berikut ini dilihat sebagai dasar :

Rekomendasi untuk bahasa markup, terutama HTML dan XHTML, dari W3C. Ini menentukan struktur dan interpretasi dokumen hypertext

Rekomendasi untuk stylesheet, terutama CSS, dari W3C.

Standar ECMA Script (biasanya dalam bentuk JavaScript), dari ECMA International

Rekomendasi untuk Document Object Model, dari W3C

.

. HTML 4.0 diperkenalkan oleh World Wide Web Consortium (W3C) pada Desember 1997, yang antara lain menyertakan fitur CSS (Cascading Style Sheets), yang mendukung pembuatan aplikasi yang dinamis. Tahun 1999 muncul HTML 4.1. Setelah kemunculan HTML 4.1, XHTML 1 lahir. Secara prinsip spesifikasi pada XHTML sama seperti pada HTML 4.01. Hal yang membedakan adalah sintaks bahasanya. XHTML menerapkan aturan yang lebih ketat daripada HTML, yakni mengharuskan pembuat dokumen harus mengikuti aturan XML (eXtended Markup Language). Selain itu, dalam penulisan dokumen semua tag dan atribut dalam dokumen harus ditulis dengan menggunakan huruf kecil, sementara pada HTML, tag dan atribut boleh ditulis dengan huruf capital, huruf kecil, ataupun kombinasi. XHTML 1.0 diteruskan dengan XHTML 2.0. Namun, versi yang terbaru ini kurang mendapat tanggapan dari para vendor.

Bagaiman halnya dengan HTML5? HTML5 (angka 5 dan HTML tidak dipisahkan oleh spasi) dikembangkan oleh badan lain yaitu WHATMG (Web Hypertext Application Technology Working Group). Spesifikasi HTML5 belum final saat ini, tetapi diyakini oleh banyak orang akan menjadi standar yang diterima di masa depan. Walaupun belum final, HTML5 telah menebar pesona. Beberapa browser (lihat Tabel 1.1) sudah mendukung HTML5.

Tabel 1.1 Browser yang mendukung HTML5

Browser	Keterangan
Chrome	Mulai versi 1.0
Firefox	Mulai versi 1.5
Internet Explorer	Mulai versi 8



Opera	Mulai versi 9.0
Safari	Mulai versi 1.3

Pada HTML, kekangan dalam menuliskan tag dan atribut dengan huruf kecil tidak ada lagi. mempunyai kebebasan untuk menggunakan huruf kecil, huruf capital, atau kombinasinya. Bahkan, nilai suatu atribut mau diberi tanda petik atau tidak sama saja. Browser tidak lagi mempedulikan hal-hal seperti itu.

Cara kerja web

Sebuah web browser adalah program perangkat lunak yang gunakan untuk mengakses World Wide Web dari Internet. Browser pertama, yang disebut NCSA Mosaic, dikembangkan di Pusat Nasional untuk Aplikasi Super Komputer di awal 1990-an. Yang mudah digunakan, cukup klak-klik sehingga membantu mempopulerkan Web.

Meskipun banyak browser yang berbeda yang tersedia, Microsoft Internet Explorer dan Mozilla Firefox yang jauh yang paling populer. (Banyak pengguna Mac lebih memilih browser Safar) Persaingan untuk mendominasi pasar telah membawa perbaikan terus-menerus untuk perangkat lunak. (keduanya didasarkan pada NCSA Mosaic)

dapat mendownload Internet Explorer dan Firefox secara gratis dari situs web masing-masing perusahaan. Jika sudah memiliki satu browser, dapat menguji yang lain. Juga mencatat bahwa ada sedikit perbedaan antara versi Windows dan Macintosh.

Anatomi Browser

Ketika pertama kali membuka browser web, biasanya dengan mengklik dua kali pada icon di desktop Anda, halaman web yang telah ditetapkan muncul. Halaman ini disebut sebagai halaman “home” atau halaman awal. Dengan Firefox misalnya, mungkin akan dibawa ke halaman rumah Mozilla atau ke halaman yang dipilih oleh penyedia layanan Internet. Tetapi jika ingin, dapat dengan mudah mengubah halaman awal.

The World Wide Web sampai saat ini adalah bagian yang paling populer dari Internet. Setelah menghabiskan waktu di Web akan merasa bahwa tidak ada batas untuk apa yang dapat temukan. Web memungkinkan kaya dan beragam komunikasi dengan memungkinkan untuk mengakses dan berinteraksi dengan teks, grafis, animasi, foto, audio dan video.

Jadi apa saja yang membentuk web ini? Pada tingkat yang paling sederhana, Web secara fisik terdiri dari komputer pribadi atau perangkat mobile, perangkat lunak web browser, sambungan ke penyedia layanan Internet, komputer yang disebut server yang disebut *hosting* dan *router* dan *switch* yang mengarahkan aliran informasi.

Ada banyak pilihan perangkat lunak server, tetapi dua yang paling populer adalah Apache (perangkat lunak open source) dan Layanan Informasi Internet Microsoft (IIS). Apache tersedia secara bebas untuk komputer berbasis Unix dan dapat diinstal pada Mac menjalankan Mac OS X. Ada versi Windows juga. Microsoft IIS adalah bagian dari keluarga Microsoft.

Setiap komputer dan perangkat (modem, router, smartphone, mobil, dll) yang terhubung ke Internet diberi alamat IP numerik yang unik (IP singkatan dari *Internet Protocol*). Sebagai contoh, komputer yang dihosting oreilly.com memiliki alamat IP 208.201.239.100. Semua angka-angka tersebut dapat memusingkan, untungnya ada Domain Name System (DNS) yang dikembangkan untuk memungkinkan kita untuk merujuk ke server dengan nama domainnya, " oreilly.com ", juga. Alamat IP numerik berguna untuk perangkat lunak komputer, sedangkan nama domain adalah lebih mudah diakses oleh manusia. Pencocokan nama domain teks ke alamat IP numerik masing-masing adalah tugas dari server DNS yang terpisah.
Hal ini dimungkinkan untuk mengkonfigurasi server web sehingga lebih dari satu nama domain dipetakan ke satu alamat IP, yang memungkinkan beberapa situs untuk berbagi server tunggal.



Sebuah halaman web adalah dokumen elektronik yang ditulis dalam bahasa komputer yang disebut HTML, singkatan dari Hypertext Markup Language. Setiap halaman web memiliki alamat yang unik, yang disebut URL atau *Uniform Eerources Locator* yang mengidentifikasi di mana web server yang memuat dokumen web.

Sebuah website memiliki satu atau lebih halaman web terkait, tergantung pada bagaimana itu dirancang. Halaman web pada situs yang dihubungkan bersama melalui sistem hyperlink, memungkinkan untuk melompat di antara halaman web dengan mengklik pada link.

Bila menelusuri World Wide Web akan melihat istilah yang sering disebut *"home page"*. Ini adalah halaman sebagai titik awal dari sebuah situs web. Seperti daftar isi dari buku atau majalah, *"home page"* biasanya memberikan gambaran tentang apa yang akan temukan di website. Situs dapat memiliki satu halaman, yang panjang beberapa atau ribuan halaman. Jika tidak ada banyak informasi, *"home page"* mungkin satu-satunya halaman. Tapi biasanya akan menemukan setidaknya beberapa halaman lainnya.

Halaman Web bervariasi dalam desain dan konten, tetapi banyak menggunakan format majalah tradisional. Di bagian atas halaman adalah masthead atau grafis banner, maka daftar item, seperti artikel, sering dengan penjelasan singkat. Item dalam daftar biasanya link ke halaman lain di situs, atau situs lain. Kadang-kadang link ini disorot kata-kata dalam tubuh teks, atau disusun dalam daftar, seperti indeks. Mereka juga dapat menjadi kombinasi dari keduanya. Sebuah halaman web mungkin juga "panas" gambar yang link ke konten lainnya.

Bagaimana bisa membedakan mana teks yang ada tautan/link? Link muncul dalam warna yang berbeda dari teks - biasanya dengan warna biru dan sering digarisbawahi. Bila memindahkan kursor di atas link teks atau melalui link grafis, perubahan icon dari panah ke tangan.

Web adalah media yang dinamis. Untuk mendorong pengunjung untuk kembali ke sebuah situs, banyak penerbit web memperbarui konten sering. Itulah yang membuat Web sangat menarik.

Memahami Alamat Web

World Wide Web adalah koleksi besar file elektronik yang tersimpan di jutaan komputer di seluruh dunia. Hypertext link file-file ini bersama-sama. Uniform Resource Locators atau URL adalah alamat yang digunakan untuk menemukan file.

Informasi yang terkandung dalam URL memberi kemampuan untuk melompat dari satu halaman web yang lain hanya dengan klik. Ketika mengetik URL ke browser atau klik link hypertext, browser mengirim permintaan ke komputer remote, yang disebut server web, untuk men-download satu atau lebih file. Setiap URL unik, mengidentifikasi satu file tertentu.

Apa yang dari khas URL? Berikut adalah beberapa contoh :

http://belajar.kemdikbud.go.id/
Halaman Rumah Belajar Kemdikbud.

http://www.facebook.com/pustekkom
Halaman Facebook untuk Pusat Teknologi Informasi - Kemdikbud.

ftp://rtfm.mit.edu/pub/
Sebuah direktori file di MIT tersedia untuk di-download.

news : rec.gardens.roses
Sebuah newsgroup dengan topik berkebun.



http://blogs.reuters.com/soccer
Sebuah blog tentang sepak bola dari kantor berita Reuters

Bagian pertama dari URL (sebelum dua garis miring) memberitahu jenis sumber daya atau metode akses di alamat itu. Sebagai contoh:

http - dokumen hypertext atau direktori
ftp - file yang tersedia untuk di-download atau direktori file tersebut
news – newsgroup
file - file yang terletak pada drive lokal komputer Anda

Bagian kedua biasanya adalah alamat komputer di mana data atau layanan berada. Bagian tambahan mungkin menentukan nama file, port untuk menghubungkan ke, atau teks untuk mencari dalam database.

Bagian-bagian dari URL

Sebuah URL lengkap umumnya terdiri dari tiga komponen : protokol, nama situs, dan path absolut ke dokumen atau sumber daya, seperti yang ditunjukkan pada Gambar berikut.

http://www.example.com / 2011/samples/first.html

Hal pertama yang dilakukan adalah URL menentukan protokol yang akan digunakan untuk transaksi tertentu. Surat-surat HTTP membiarkan server tahu untuk menggunakan Hypertext Transfer Protocol, atau masuk ke dalam "modus web"
www.example.com
Bagian berikutnya dari URL mengidentifikasi situs dengan nama domainnya. Dalam contoh ini, nama domain adalah example.com.
www

Bagian di awal adalah nama host tertentu di domain tersebut. Nama host " " telah menjadi sebuah kesepakatan. Tetapi, kadang-kadang nama host dapat dihilangkan. Ada bisa lebih dari satu website di domain (kadang-kadangdisebutsubdomain).Misalnya,adajugamungkin development.example.com, clients.example.com, dan sebagainya.

/ 2012/samples/first.html

Ini adalah path absolut melalui direktori pada server ke dokumen HTML yang diminta, first.html. Kata-kata dipisahkan oleh garis miring adalah nama direktori, dimulai dengan direktori root dari host (seperti yang ditunjukkan oleh awal /), URL pada Gambar 2-1 mengatakan ingin menggunakan protokol HTTP untuk terhubung ke server web di Internet disebut www.example.com dan meminta first.html dokumen (terletak di direktori sampel, yang di direktori 2012).

Sebagian dari URL yang akan menggunakan start dengan http, yang merupakan singkatan dari Hypertext Transfer Protocol, metode yang file HTML yang ditransfer melalui Web. Berikut adalah beberapa hal-hal lain untuk mengetahui tentang URL :

Sebuah URL biasanya tidak memiliki spasi.

Sebuah URL selalu menggunakan garis miring ke depan (/ /).

Penulisan URL tidak sensitif huruf besar-kecil. Jadi mengetik "http://www.kemdikbud.go.id" atau "HTTP://WWW.KEMDIKBUD.GO.ID" atau variasi huruf besar dan kecil akan membawa ke halaman yang sama.

Jika mengetik URL tidak benar, browser tidak akan dapat menemukan situs atau sumber daya yang inginkan. Jika mendapatkan pesan error atau mengakses situs yang salah, periksa untuk melihat apakah mengeja alamat dengan benar.



File standar

Jelas, tidak setiap URL yang lihat adalah begitu panjang. Banyak alamat tidak termasuk nama file, tapi hanya menunjuk ke sebuah direktori, seperti ini :

http://www.w3schools.com/
http://w3schools.com/html/
Ketika server menerima permintaan untuk nama direktori bukan file tertentu, terlihat dalam direktori tersebut untuk dokumen default, biasanya bernama index.html.
Jadi, ketika seseorang jenis URL di atas ke browser mereka, apa yang mereka benar-benar akan melihat apakah ini :

http://www.w3schools.com/index.html
http://www.jendesign.com/resume/index.html
Nama file default (juga disebut sebagai file indeks) dapat bervariasi, dan tergantung pada bagaimana server dikonfigurasi. Dalam contoh ini, ia dinamai index.html, tetapi beberapa server menggunakan nama file default.htm. Jika situs menggunakan pemrograman server-side untuk menghasilkan halaman, file indeks mungkin diberi nama index.php atau index.asp. Hanya memeriksa dengan administrator server atau departemen dukungan teknis di layanan hosting untuk memastikan memberikan file default nama yang tepat.
Hal lain yang perlu diperhatikan adalah bahwa dalam contoh pertama, URL asli tidak memiliki garis miring tertinggal untuk menunjukkan itu adalah sebuah direktori. Ketika slash dihilangkan, server hanya menambahkan satu jika menemukan direktori dengan nama itu.

File index ini juga berguna untuk keamanan. Beberapa server (tergantung pada konfigurasi mereka) menampilkan isi direktori jika file default tidak ditemukan. Gambar 2-2 menunjukkan bagaimana dokumen-dokumen di direktori housepics terpapar sebagai hasil dari file default yang hilang.

Salah satu cara untuk mencegah orang dari mengintai di dalam file adalah untuk memastikan ada file index di setiap direktori. Administrator server juga dapat menambahkan perlindungan lain untuk mencegah direktori dari menampilkan di browser.

Di bagian atas Gambar 2-3, melihat halaman web minimal seperti yang muncul dalam browser grafis. Meskipun melihatnya sebagai satu halaman yang koheren, sebenarnya dirakit dari empat berkas terpisah: dokumen HTML (index.html), sebuah style sheet (kitchen.css), dan dua grafis (foods.gif dan spoon.gif). Dokumen HTML akan menampilkan semuanya.

Bagaimana web bekerja

Untuk membungkus pengantar kami bagaimana web bekerja, mari kita melacak aliran khas peristiwa yang terjadi dengan setiap halaman web yang muncul pada layar (Gambar 2-5).

meminta halaman web dengan baik mengetik URL -nya (misalnya, http:// jenskitchensite.com) langsung di browser atau dengan mengklik pada link pada halaman. URL yang berisi semua informasi yang diperlukan untuk menargetkan sebuah dokumen tertentu pada server web tertentu di Internet.

Browser mengirimkan Permintaan HTTP ke server yang disebutkan dalam URL dan meminta untuk file tertentu. Jika URL menentukan sebuah direktori (bukan file), itu adalah sama dengan meminta file default dalam direktori tersebut.

Server mencari file yang diminta dan mengeluarkan respon HTTP.

a. Jika halaman tidak dapat ditemukan, server mengembalikan pesan kesalahan. Pesan tersebut biasanya mengatakan "404 Not Found," meskipun pesan kesalahan lebih ramah dapat diberikan.



b. Jika dokumen ditemukan, server mengambil file yang diminta dan kembali ke browser.

Browser mem-parsing dokumen HTML. Jika halaman berisi gambar (ditandai dengan elemen img HTML) atau sumber daya eksternal lainnya seperti script, kontak peramban server lagi untuk meminta setiap sumber daya yang ditentukan dalam markup.

Browser menyisipkan setiap gambar dalam aliran dokumen di tempat yang ditentukan oleh elemen img. Semua media ditampilkan dalam sebuah halaman web.

c. Rangkuman

Secara umum, pekerjaan di bidang teknologi informasi setidaknya dapat dikelompokan sesuai bidangnya, misalnya.

Kelompok pertama, adalah mereka yang bergelut di dunia perangkat lunak (software), baik mereka yang merancang system operasi, database maupun system aplikasi.

Kelompok kedua, adalah mereka yang bergelut di bidang perangkat keras (hardware).

Kelompok ketiga, adalah mereka yang berkecimpung dalam operasional system informasi.

Profesi-profesi yang langsung terkait dalam pengembangan aplikasi web, diantaranya : Web Designer, Web Programmer, Web Administrator, Web Master, Web Developer.

d. Tugas

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa

2. Kegiatan Belajar 2: Alur dan Perangkat Pengembangan Aplikasi Web

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 2 ini siswa diharapkan dapat :

Memahami Alur Pengembangan Aplikasi Web

Memahami Perangkat Pengembangan Aplikasi Web

b. Uraian Materi

Alur Pengembangan Aplikasi Web

Berdasarkan persepsi dari beberapa pengembang perangkat lunak dan ahli-ahli dalam bidang rekayasa perangkat lunak (software engineering professional), Pengembangan Aplikasi Web tidaklah sama dengan rekayasa perangkat lunak walaupun keduanya melibatkan pemrograman dan pengembangan perangkat lunak.

Walaupun banyak mengadopsi prinsip-prinsip rekayasa perangkat lunak, Pengembangan Aplikasi Web memiliki banyak pendekatan, metoda, alat bantu,



teknik dan panduan yang memenuhi persyaratan pembuatan sistem berbasis web.

Pengembangan sistem berbasis web berbeda dengan pengembangan perangkat lunak konvensional, dimana pengembangan sistem berbasis web lebih banyak menghadapi tantangan. Pengembangan Aplikasi Web adalah gabungan dari print publishing dan pengambangan perangkat lunak, diantara marketing dan perhitungan dan diantara seni dan teknologi.

Alternatif model dari Pengembangan Aplikasi Web adalah sebagai berikut :

Formulasi (formulation)

Kegiatan yang berfungsi untuk merumuskan tujuan dan ukuran dari aplikasi berbasis web serta menentukan batasannya sistem.

Tujuan yang ingin dicapai bisa dibedakan menjadi dua kategori, yaitu :

1. Tujuan yang bersifat informatif

Menyediakan suatu informasi tertentu kepada pengguna, berupa teks, grafik, audio, dan video.

2. Tujuan yang bersifat fungsional

Kemampuan untuk melakukan suatu fungsi yang dibutuhkan pengguna, misal dengan menggunakan aplikasi tersebut seorang guru dapat memperoleh nilai akhir dan statistik nilai guru dari data-data ujian, tugas, kuis yang ia input ke dalam aplikasi.

Perencanaan (planning)

Kegiatan yang digunakan untuk menghitung estimasi biaya proyek pembuatan aplikasi berbasis web ini, estimasi jumlah pengembang, estimasi waktu pengembangan, evaluasi resiko pengembangan proyek, dan mendefinisikan jadwal pengembangan untuk versi selanjutnya (jika diperlukan).

Analisis (analysis)

Kegiatan untuk menentukan persyaratan-persyaratan teknik dan mengidentifikasi informasi yang akan ditampilkan pada aplikasi berbasis web. Analisis yang digunakan pada rekayasa web dilakukan dari empat sisi, yaitu :

1. Analisis isi informasi

Mengidentifikasi isi yang akan ditampilkan pada aplikasi berbasis web ini. Isi informasi dapat berupa teks, grafik, audio, maupun video.

2. Analisis interaksi

Analisis yang menunjukkan hubungan antara web dengan pengguna.

3. Analisis fungsional

Analisis tentang proses bagaimana aplikasi berbasis web ini akan menampilkan informasi kepada pengguna.

4. Analisis konfigurasi

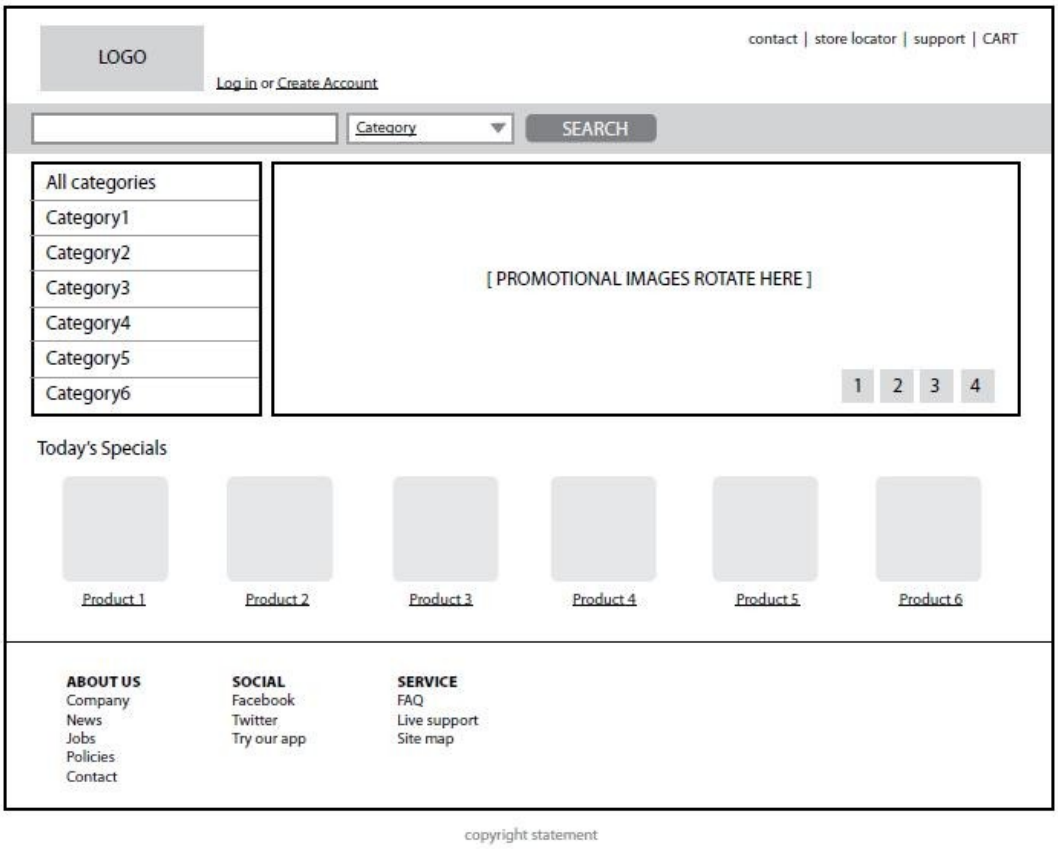
Konfigurasi yang digunakan pada aplikasi berbasis web, internet, intranet, atau extranet. Selain itu, analisis ini juga meliputi relasi database dengan web jika diperlukan.

Rekayasa (engineering)

Terdapat dua pekerjaan yang dilakukan secara paralel, yaitu desain isi informasi dan desain arsitektur web.

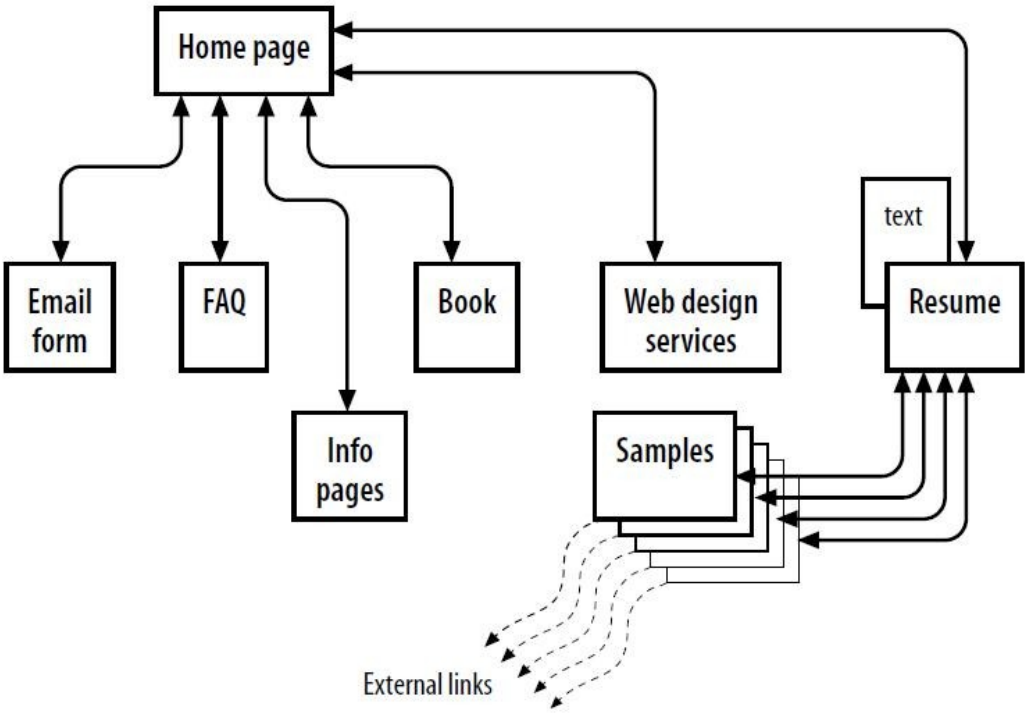
Pada saat tahap desain, ada beberapa hal yang perlu dilakukan :

Pembuatan Diagram Wireframe (Gambar Rangka), tujuan dari diagram gambar rangka adalah untuk menunjukkan bagaimana lay out halaman web dan menunjukkan di mana fungsi dan konten seperti navigasi, kotak pencarian, elemen bentuk dan sebagainya, tanpa desain grafis. Berikut ini contohnya :



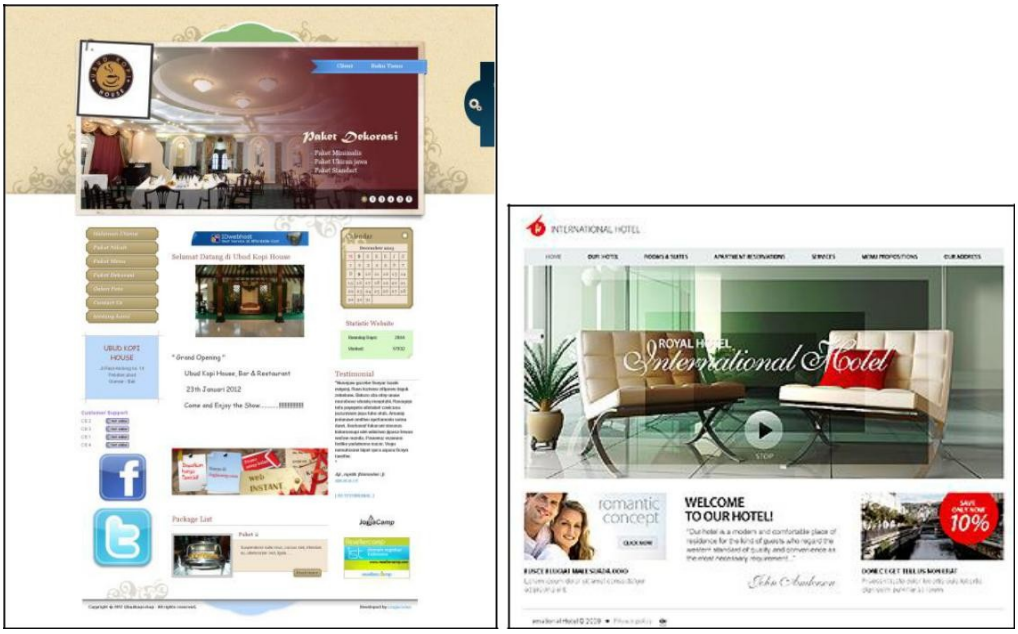
Gambar. Diagram Wireframe sebuah halaman web.

Diagram Situs, Sebuah diagram situs menunjukkan struktur situs secara keseluruhan dan bagaimana halaman individual berhubungan satu sama lain. Gambar …. menunjukkan diagram situs yang sangat sederhana



Gambar. Diagram Situs

Storyboard dan diagram alir pengguna, Storyboard ini bertujuan untuk menunjukkan langkah-langkah yang diperlukan untuk menyelesaikan tugas-tugas, opsi yang mungkin, dan juga memperkenalkan beberapa standar jenis halaman. Gambar 1-3 menunjukkan storyboard sederhana



Gambar. Desain Visual (Sumber: www.idwebhost.com, www.idebagus.com)

Implementasi (*page generation*) & pengujian (*testing*)

Suatu kegiatan pengembangan untuk mewujudkan desain menjadi suatu web site. Teknologi yang digunakan tergantung dengan kebutuhan yang telah dirumuskan pada tahap analisis.

Dalam tahap pengembangan ini, meliputi kegiatan:

***Authoring*,** adalah istilah yang digunakan untuk proses mempersiapkan konten untuk nantinya diunggah di Web, atau lebih khusus, menandai konten dengan tag HTML yang menggambarkan isi dan fungsinya.

Styling, dalam development web, penampilan halaman dalam browser dikendalikan oleh aturan style yang ditulis dalam CSS (Cascading Style Sheets)

***Scripting* dan pemrograman**, ada bahasa pemrograman web terkait diantaranya PHP, Ruby, Python, dan ASP.NET, yang

berjalan pada server dan mengolah data dan informasi sebelum dikirim ke browser pengguna.

Pengujian dilakukan setelah implementasi selesai dilaksanakan. Pengujian meliputi beberapa parameter yang akan menentukan standar aplikasi berbasis web yang telah dibuat. Tahap pengujian adalah suatu proses untuk menguji aplikasi berbasis web yang telah selesai dibuat. Hal ini bertujuan untuk menemukan kesalahan dan kemudian memperbaikinya. Pengembang suatu aplikasi berbasis web mendapat tantangan besar untuk melakukan pengujian karena karakter aplikasi ini yang beroperasi pada jaringan dengan berbagai macam pengguna, berbagai macam sistem operasi, perangkat keras, browser, protokol komunikasi, dll.

Ada beberapa pendekatan yang digunakan untuk melakukan pengujian, yaitu :
1. Pengujian fungsional dan operasional (fungsional and operational testing)

Bertujuan untuk menguji masukan dan keluaran dari aplikasi ini. Hasil keluaran aplikasi bergantung dari teknologi yang digunakan, baik itu
bahasa pemrograman maupun bahasa skrip yang digunakan. Untuk menguji code HTML dan CSS yang digunakan dapat menggunakan alat bantu W3C HTML Validation Service di http://validator.w3.org/ dan W3C CSS Validation Service di http://jigsaw.w3.org/css-validator/

2. Pengujian navigasi (navigation testing)

Hal ini digunakan untuk melihat kesesuaian antara desain navigasi dengan navigasi yang ada di aplikasi. Navigasi berhubungan dengan link-link yang terdapat didalam aplikasi.

Untuk menguji link dapat digunakan alat bantu W3C Link Checker Service di http://validator.w3.org/checklink

3. Pengujian konfigurasi (configuration testing)



Pengujian ini dilakukan pada sistem operasi, browser, sistem perangkat keras dan perangkat lunak pendukung. Pengujian ini dilakukan untuk menentukan batas toleransi kebutuhan aplikasi akan perangkat lunak dan perangkat keras pendukungnya.

Pengujian keamanan dan performansi (security and performance testing)
Pengujian ini dilakukan untuk melihat tingkat keamanan aplikasi dengan cara menguji aspek-aspek yang dapat menimbulkan gangguan keamanan aplikasi maupun server. Keamanan aplikasi sangat bergantung pada teknologi pengembangan website, konfigurasi server yang digunakan dan kelakuan sistem. Pengujian performansi dapat dilakukan bersamaan dengan pengujian keamanan aplikasi, karena keamanan aplikasi berbasis web juga tergantung dari performansi server dan aplikasi tersebut.

Evaluasi oleh konsumen (customer evaluation)

Suatu kegiatan akhir dari siklus proses rekayasa web, akan menentukan apakah web yang telah selesai dibuat tersebut sesuai dengan yang mereka inginkan. Apabila aplikasi berbasis web ini belum sesuai dengan kehendak mereka, maka proses rekayasa web akan terus dilakukan dan dimulai lagi dari tahap formulasi untuk versi berikutnya.

Perangkat pengembangan aplikasi web

Untuk perangkat pengembangan aplikasi web dapat dikelompokan menjadi perangkat keras dan perangkat lunak.

Perangkat Keras

Untuk lingkungan pengembangan web yang nyaman, berikut ini peralatan yang sebaiknya disediakan:

Komputer Utama, dengan operating system Macintosh, Windows, ataupun Linux. Perusahaan pengembangan web profesional cenderung berbasis Macintosh.

Memori tambahan. sebaiknya computer memiliki memori yang besar, karena dalam pengembangan web cenderung kita akan membuka beberapa aplikasi sekaligus.

Sebuah monitor besar. Meskipun bukan keharusan, monitor besar memudahkan untuk desainer visual. Semakin besar monitor, semakin banyak jendela dan panel kontrol dapat terbuka pada waktu yang sama.

Sebuah scanner dan / atau kamera digital. Jika mengantisipasi membuat gambar dan tekstur sendiri, akan memerlukan beberapa alat untuk membuat gambar.

Sebuah komputer kedua. Banyak *web designer* merasa berguna memiliki komputer penguji yang menjalankan platform yang berbeda dari komputer utama yang mereka gunakan untuk pengembangan. Karena browser kerja yang berbeda pada Mac dan pada mesin Windows, itu penting untuk menguji halaman web dalam sebanyak mungkin system operasi berbeda.

Perangkat mobile/smartphone. Saat ini banyak orang mengakses web melalui smartphone/tablet, sehingga kita perlu menguji penampilan dan kinerja situs web yang kita buat pada mobile browser pada smartphone atau perangkat tablet. Namun saat ini banyak web emulator yang dapat digunakan sebagai penguji web kita

Perangkat Lunak

Untuk perangkat lunak pengembangan web dapat dikelompokkan menjadi beberapa, diantaranya :

Web page authoring

Ini adalah kelompok perangkat lunak seperti desktop publishing tetapi format yang dihasilkan adalah halaman web. Aplikasi ini biasanya bersifat “WYSIWYG” (What You See Is What You Get, dibaca “whizzy-wig”) yaitu apa yang kamu lihat,

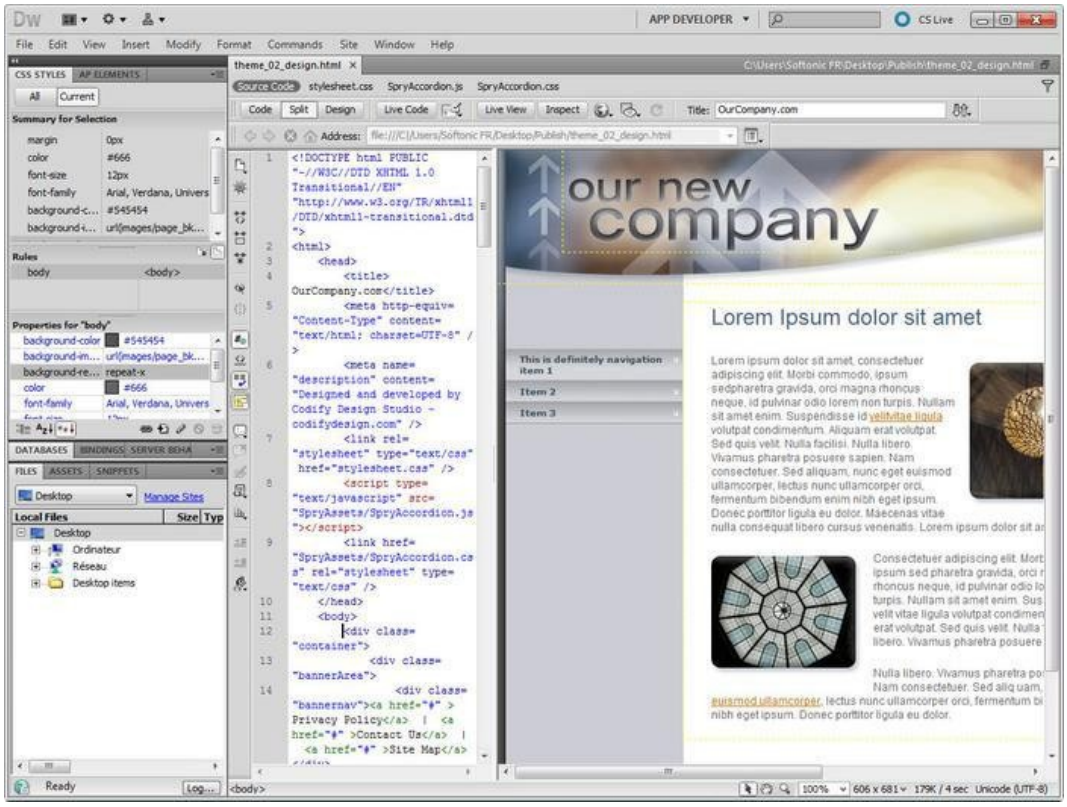
itulah yang kamu dapatkan. Jadi aplikasi ini biasanya ketika membuat tampilannya seperti yang akan dihasilkan.

Beberapa contoh aplikasi yang termasuk dalam kelompok ini :

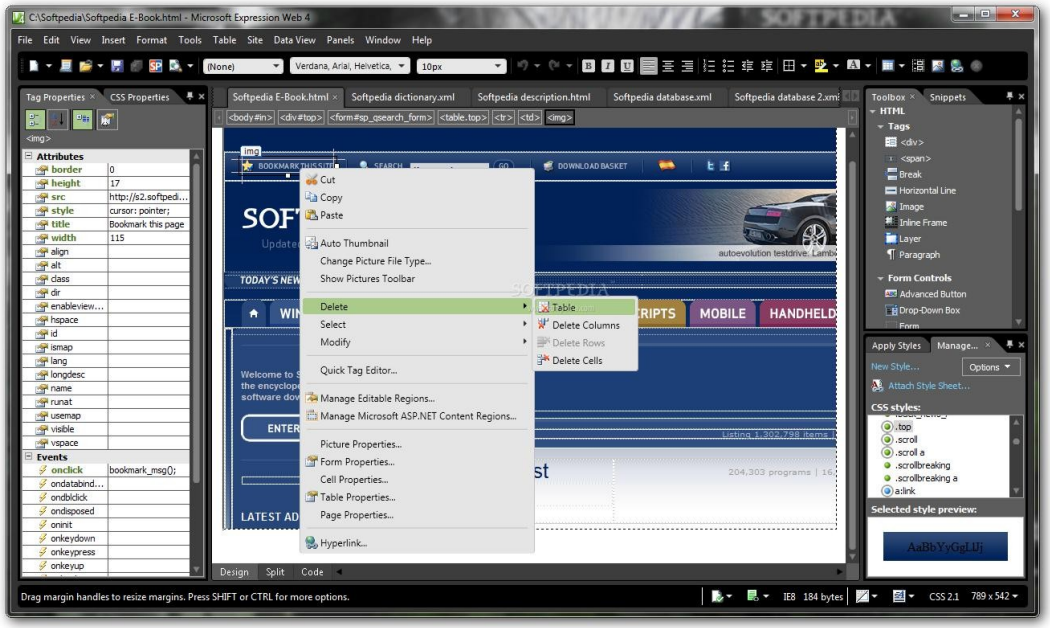
Adobe Dreamweaver, aplikasi ini berbayar dan dibawah lisensi perusahaan Adobe, aplikasi ini banyak digunakan oleh pengembang web. Memiliki banyak fasilitas dan fiture.

Microsoft Expression Web (lisensi dari Windows). Bagian dari Microsoft's suite yang merupakan aplikasi perancangan professional.

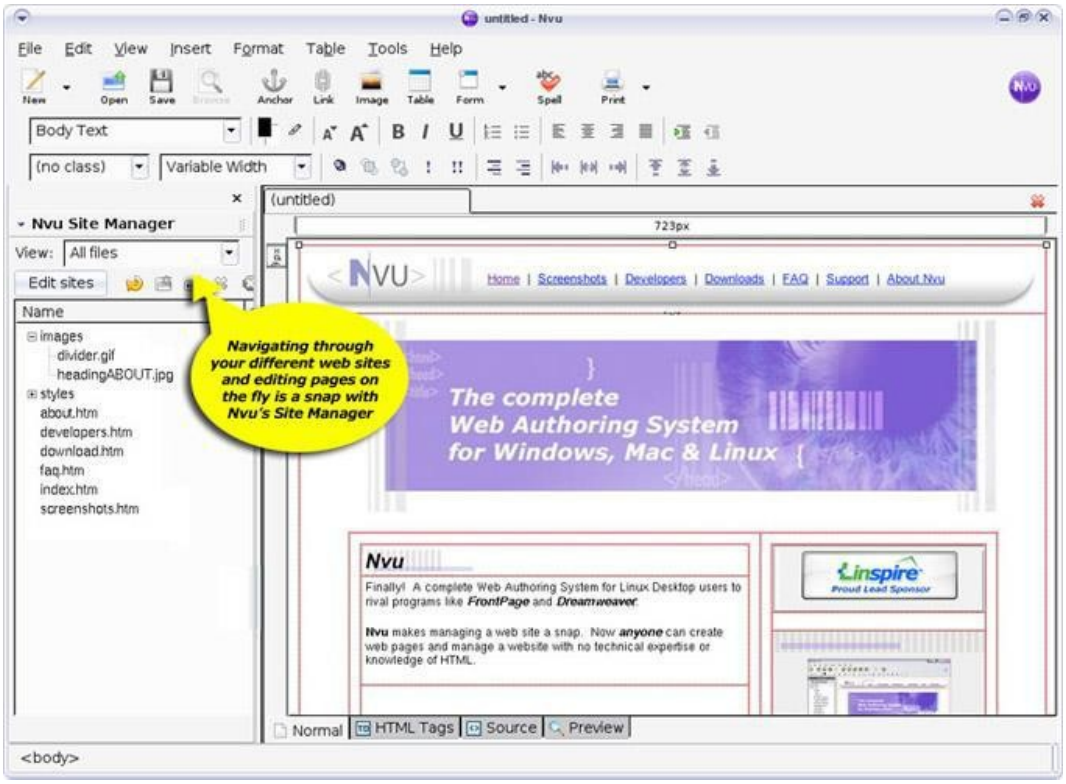
Nvu (Linux, Windows, dan Mac OS X). Tidak ingin membayar untuk Editor WYSIWYG? Nvu (diucapkan N-view, untuk "pandangan baru") adalah sebuah aplikasi open source (sumber terbuka) yang cocok dengan banyak fitur dalam Adobe Dreamweaver, dan dapat diunduh secara gratis di nvu.com.



Gambar. Tampilan aplikasi Adobe Dreamweaver CS6 (sumber : http://imagenes.es.sftcdn.net)



Gambar. Tampilan aplikasi Microsoft Expression Web (sumber : http://i1-win.softpedia-static.com)



Gambar. Tampilan aplikasi Nvu (sumber : http://net2.com/nvu/screenshots.html)

HTML Editor



Ini adalah kelompok perangkat lunak yang kebalikan dari Adobe Dreamweaver Microsoft Expression Web. Aplikasi ini tidak lagi "WYSIWYG" (What You See Is What You Get") tetapi lebih ke kode HTML. Beberapa contoh aplikasi yang termasuk dalam kelompok ini :

TextPad (Windows saja). TextPad adalah plain-text sederhana dan murah untuk editor kode di Windows.

Teks Sublime (Window, Mac, Linux). Editor teks ini terlihat dipreteli namun memiliki banyak fungsi (seperti kode warna dan ikhtisar kode penuh) bagi pengembang web.

Coda oleh Panic (hanya Macintosh). Pengguna Coda seperti alur kerja yang visual, alat-alat manajemen file, dan built-in akses terminal.

Textmate oleh MacroMates (hanya Macintosh). Editor teks canggih ini memiliki alat manajemen proyek dan antarmuka yang terintegrasi dengan sistem operasi Mac. Hal ini semakin populer karena disesuaikan, kaya fitur, dan murah.

BBEdit oleh Bare Bones Software (hanya Macintosh). Untuk pengembang web banyak fitur *shortcut* yang besar telah membuat editor terkemuka berbasis Mac

Aplikasi editing gambar dan menggambar

Untuk menambahkan gambar ke halaman web, akan perlu program editing gambar. Berikut ini beberapa aplikasi yang cukup populer :

Adobe Photoshop. Photoshop dapat dikatakan sebagai standar industri untuk penciptaan gambar baik dalam cetak maupun dunia web.

Adobe Photoshop Elements. Versi ini lebih ringan dari Photoshop dirancang untuk mengedit foto dan manajemen, tetapi beberapa penggemar menemukan bahwa ia memiliki semua alat yang diperlukan untuk menempatkan gambar pada halaman web.

Adobe Illustrator. Karena desainer perlu membuat logo, ikon, dan ilustrasi di berbagai ukuran dan resolusi, untuk itu semua aplikasi ini sangat membantu. Aplikasi ini juga dapat dipadukan dengan Photoshop

Adobe Fireworks. Program grafis web ini menggabungkan editor foto dengan alat untuk membuat ilustrasi berbasis vektor. Ia juga memiliki alat-alat canggih untuk grafis web.

Corel Paint Shop Photo Pro (Windows saja). Editor gambar yang penuh fitur ini populer di pengguna Windows, terutama karena harga yang rendah.

Aplikasi Internet

Karena hasil akhir dari pengembangan halaman web digunakan di Internet, maka diperlukan beberapa aplikasi untuk membuka halaman web, biasa disebut "browser". Berikut ini beberapa aplikasi browser yang cukup populer :

Windows:

- Internet Explorer
- Chrome
- Firefox
- Safari
- Opera

Macintosh OS X:

- Safari
- Chrome
- Firefox
- Opera

Browser perangkat bergerak (mobile)

- Mobile Safari (iOS)
- Android Browser (Android) o
- BlackBerry Browser (RIM)
  - Nokia Series 40 and Nokia Browser for Symbian
  - Opera Mobile and Mini (installed on any device)
  - Internet Explorer Mobile (Windows Phone)
  - Silk (Kindle Fire)

Aplikasi transfer file (FTP/File Transfer Protokol).

Sebuah aplikasi FTP memungkinkan untuk mengunggah dan mengunduh file antara komputer kita dan komputer yang akan menjadi tempat halaman web/web server. Beberapa aplikasi authoring web sebelumnya juga memiliki aplikasi FTP



yang terintegrasi langsung. Ada juga aplikasi FTP khusus, seperti yang tercantum di bawah ini:

Windows

- WS_FTP
- CuteFTP
- AceFTP
- Filezilla

Macintosh

- Transmit
- Cyberduck
- Fetch

c. Rangkuman

Alternatif model dari Pengembangan Aplikasi Web adalah sebagai berikut : Formulasi, Perencanaan (planning), Analisis (analysis), Rekayasa (engineering), Implementasi (*page generation*) & pengujian (*testing*), Evaluasi oleh konsumen (customer evaluation).

Untuk perangkat pengembangan aplikasi web dapat dikelompokan menjadi perangkat keras dan perangkat lunak.

d. Tugas

Bukalah aplikasi browser, bukalah situs penyedia browser yang lain, dan installah. Bandingkan fitur-fitur yang ada.

Bukalah satu situs dengan 2-3 jenis browser berbeda, bagaimana tampilannya?

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa

3. Kegiatan Belajar 3 : Anatomi dokumen web

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 3 ini siswa diharapkan dapat :

Memahami Dasar-dasar HTML

Memahami Properti Dkumen Web

Memahami

b. Uraian Materi

1. Anatomi dokumen Web

Dasar-Dasar HTML

HTTP (Hypertext transfer protocol) merupakan protokol yang digunakan untuk mentransfer data antara web server ke web browser. Protocol ini mentransfer dokumen-dokumen web yang ditulis atau berformat HTML (Hypertext Markup Language). Dikatakan markup language karena HTML berfungsi untuk 'memperindah' file teks biasa untuk ditampilkan pada program web browser hal ini dilakukan dengan menambahkan elemen atu sering disebut sebagai tag-tag pada file teks biasa tersebut.

Saat ini sudah dikenal HTML5 yang menawarkan berbagai fitur menarik yang tidak didukung oleh HTML sebelumnya. Beberapa fitur yang tersedia pada HTML5 antara lain sebagai berikut.

Canvas. Memungkinkan pembuatan gambar dalam kanvas. Jadi, gambar tidak lagi diambil dari gambar utuh, melainkan bisa disusun sendiri, bak menggunakan program Paint.

Header. Berguna untuk menyatakan suatu judul, yang bisa diisi dengan elemen logo dan nama perusahaan.



Footer. Merupakan kebalikan dari header. Sebagai catatan kaki, elemen ini berguna untuk menaruh informasi di bagian bawah halaman web.

- Time. Elemen ini berguna untuk menyajikan informasi tentang waktu.
- Audio. Memungkinkan penyajian player untuk memutar suara.

Video. Memungkinkan player untuk memainkan film.

Secara prinsip, fitur pada HTML dapat dikelompokkan ke dalam :

Struktur halaman ;

Presentasi visual ;

Peranti penyaji gambar ;

Pendukung media ; dan Peningkatan

koneksi dengan JavaScript.

Tag HTML biasanya berupa tag-tag yang berpasangan dan ditandai dengan symbol lebih besar (<) dan (>). Pasangan dari sebuah tag ditandai dengan symbol garis miring (/). Misalnya pasangan dari tag <contoh > adalah</contoh>. Dalam hal ini <contoh> kita sebut sebagai elemen dan biasanya dalam suatu elemen terdapat atribut-atribut untuk mengatur elemen itu. Jadi misalnya elemen <contoh> bila ditulis dengan atributnya adalah sebagai berikut:
<contoh atribut1=”nilai_atribut1”atribut2=”nilai_atribut2=”…>.

Dalam penulisan tag HTML tidaklah case sensitive artinya pengguna huruf kecil ataupun capital tidaklah menjadi masalah.

Struktur Dasar Dokumen HTML

Setiap dokumen HTML memiliki struktur dasar atau susunan file sebagai berikut :

```
<html>
      <head>
           <title>Judul Halaman</title>
      </head>
     <body>
          Berisi tentang text, gambar,
          atau apapun yang
          tampil pada dokumen web.
      </body>
</html>
```

Seperti terlihat, struktur file HTML diawali dengan sebuah tag<html> dan ditutup dengan tag </html>. Di dalam tag ini terdapat dua buah bagian besar, yaitu yang diapit oleh tag<head>…</head> dan tag <body>…</body>.

Bagian yang diapit oleh tag HEAD merupakan header dari halaman HTML dan tidak ditampilkan pada browser. Bagian ini



berisi tag-tag header seperti <tittle>… </tittle> yang berpungsi untuk mengeluarkan judul pada tittle bar window web browser.

Bagian kedua, yang diapit ole tag BODY merupakan bagian yang akan ditampilkan pada halaman web browser nantinya. Pada bagian ini anada akan menuliskan semua jenis informasi berupa teks dengan bermacam format maupun gambar yang ingin sampaikan pada pengguna nantinya.

Pengaturan Properti Dokumen

Properti document diatur melalui atribut-atribut yang terdapat dalam elemen <body>. Sebagai contoh adalah pengaturan warna latar belakng halaman, wana teks, warna link dan lain-lain.

Kode Warna

Dalam pengaturan warna menggunakan kode RGB yang mana ditampilkan dalam nilai heksadesimal. Setiap bagian dua gigit kode menunjukkan banyaknya intensitas dari kombinasi warna merah, hijau dan biru. Sebagai contoh 00 pada dua digit pertama berarti tidak ada warna merah dalam kobinasi warna berikut ini adalah contoh kode warna :

Color	**hexadecimal**	**Color**	**Hexadecimal**
White	#FFFFFF	Black	#000000
Red	#FF0000	Green	#00FF00
Blue	#0000FF	Magenta	#FF00FF

Cyan	#00FFFF	Yellow	#FFFF00
Aquamarine	#70DB93	Chocolate	#5C3317
Violet	#9F5F9F	Brass	#B5A642
Coper	#B87333	Pink	#FF6EC7

Atribut Elemen <body>

BACKGROUND = Lokasi dan nama file (latar belakng dokumen image dokumrn)

BGCOLOR = Warna (warna latar belakng dokumen, default putih)

TEXT = Warna (warna teks dokumen, default hitam)

LINK = Warna (warna link dokumen, default biru)

VLINK = Warna (warna visited link dokumen, default ungu)

ALINK = Warna (warna aktif link dokumen, default merah)

Elemen Heading <h1>… <h6>

Tag heading berfungsi untuk memformat heading (judul dan sub-judul) dari suatu halam web. Heading ini akan memperbesar ukuran huruf unruk setiap jenis heading. Ad 7 buah heading yang dikenal di HTML, yaitu dari <h1> sampai <h6>.

Elemen yang dihilangkan di HTML5



Berikut ini elemen HTML 4.01 yang dihilangkan di HTML5:

<acronym>
<applet>
<basefont>
<big>

<dir>
<font>
<frame>
<frameset>
<noframes>
<strike>
<tt>

Dokumen HTML5

Sebuah dokumen HTML5 seperti berikut ini, cukup ditulis dengan Text Editor ataupun HTML Editor :

```
<! DOCTYPE HTML>
        <html>
          <head>
              <tittle>Tes HTML5</tittle>
           </head>
           <body>
              Selamat belajar HTML5
           </body>
        </html>
```

Berikut ini penjelasan dari dokumen HTML5 di atas :

Sebuah dokumen HTML5 diawali dengan <!DOCTYPE HTML>.

Tanda seperti <html> disebut tag. Sebuah tag seperti itu menyatakan sebuah elemen dalam dokumen html.

Beberapa tag berpasangan. Sebagai contoh, <head> berpasangan dengan </head>.

Namun, tidak semua tag berpasangan. Sebagai contoh, <br> tidak punya pasangan. Khusus untuk tag seperti ini, tidak ada keharusan untuk menyertakan tanda / sebelum >. Pada XHTML, tanda seperti / memang diharuskan untuk ditulis.

Pasangan <html>..</html> menyatakan awal dokumen HTML.

Di dalam <html>..</html> terdapat pasangan <head>..</head> dan <body>..</body>.

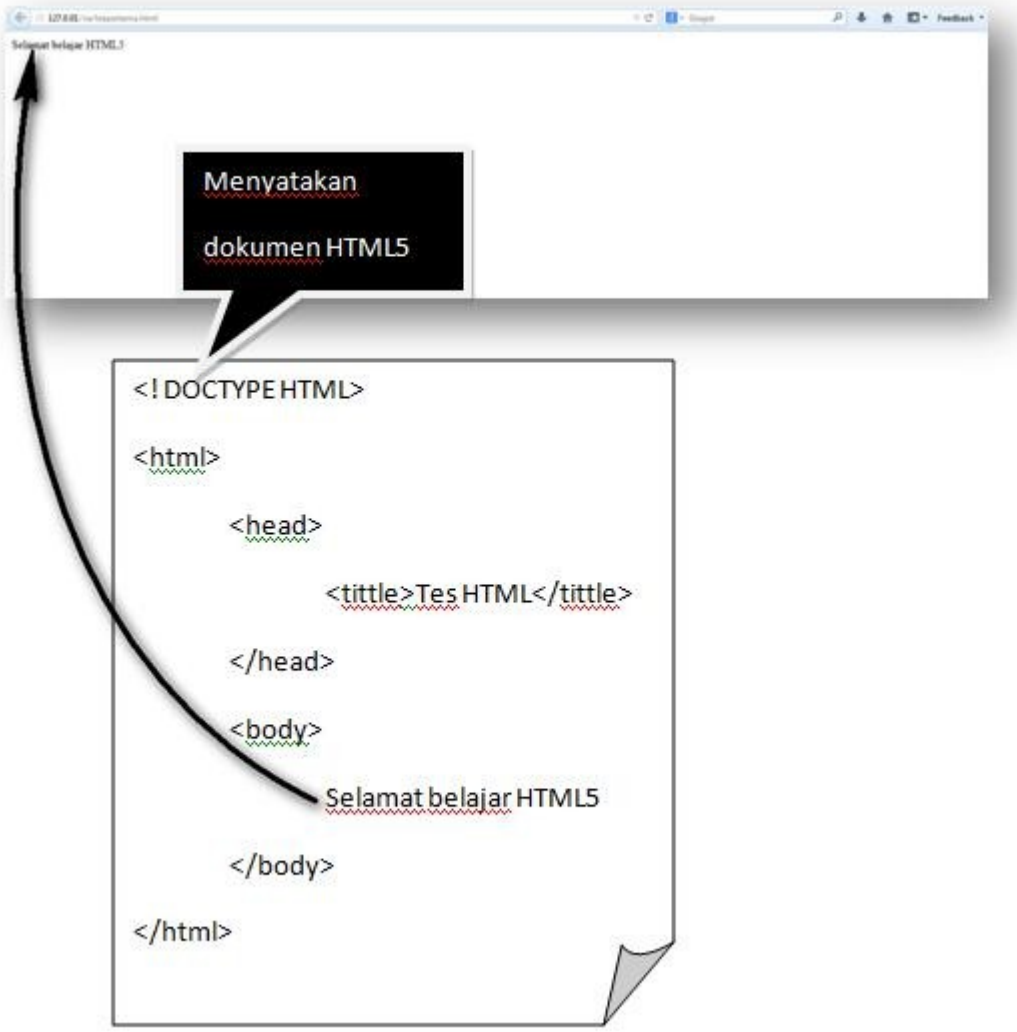
Pasangan <head>..</head> menyatakan bagian judul dokumen HTML. Isinya paling tidak berupa pasangan <tittle>..</tittle>.

Pasangan <body>..</body> menyatakan bagian tubuh dokumen. Pada contoh,

Selamat belajar HTML5

muncul pada tubuh dokumen.

Gambar 3.1 memperlihatkan penjelasan secara visual terhadap dokumen HTML di depan.



Gambar 3.1 Penjelasan dokumen HTML

c. Rangkuman

Dokumen-dokumen web ditulis dalam format HTML (Hypertext Markup Language). Dikatakan markup language karena HTML berfungsi untuk 'memperindah' file teks biasa untuk ditampilkan pada program web browser hal ini dilakukan dengan menambahkan elemen atu sering disebut sebagai tag-tag pada file teks biasa tersebut

Struktur file HTML diawali dengan sebuah tag<html> dan ditutup dengan tag </html>. Di dalam tag ini terdapat dua buah bagian besar, yaitu yang diapit oleh tag<head>…</head> dan tag <body>…</body>

d. Tugas

Untuk membuat dokumen HTML dengan Text Editor paling tidak ada 4 tahap :

Buka Text Editor/HTML Editor

Ketikkan dokumen HTML

Simpan dengan format *.html

Buka hasilnya/dokumen html dengan web browser

Buatlah beberapa dokumen HTML berikut ini. :

HTML Heading

```
<!DOCTYPE html>

<html>

   <head>

            <tittle>Heading</tittle>

   </head>


<body>


<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

<h4>This is heading 4</h4>
```




```
<h5>This is heading 5</h5>

<h6>This is heading 6</h6>


</body>

</html>
```

Bagaimana tampilannya? Apa yang dapat kamu simpulkan tentang elemen Heading?

ii. HTML Paragraph

```
<!DOCTYPE html>

<html>

   <head>

            <tittle>Paragraf</tittle>

   </head>


<body>


<p> Ini adalah paragraf. Ini adalah paragraf.
Ini adalah paragraf. Ini adalah paragraf. Ini
adalah paragraf. Ini adalah paragraf. </p>
<p> Ini paragraph lainnya. Ini paragraph
lainnya. Ini paragraph lainnya. Ini paragraph
lainnya. Ini paragraph lainnya. </p>


</body>
```

```
</html>
```

Bagaimana tampilannya? Apa yang dapat kamu simpulkan tentang elemen Paragraph?

iii. HTML Link

```
<!DOCTYPE html>
<html>
   <head>
            <tittle>Link</tittle>
   </head>

<body>

<a href="http://www.w3schools.com">
Ini adalah link</a>

</body>
</html>
```

Bagaimana tampilannya? Apa yang dapat kamu simpulkan tentang elemen Paragraph?

iv. HTML Image

```
<!DOCTYPE html>
```

```
<html>
   <head>
      <tittle>Gambar/Image</tittle>
            </head>

<body>

<img src="w3schools.jpg" alt="W3Schools.com"
width="104" height="142">

</body>
</html>
```

Bagaimana tampilannya? Apa yang dapat kamu simpulkan tentang elemen Image?

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa

4. Kegiatan Belajar 4 : Pemformatan Teks dan Paragraf Web

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 4 ini siswa diharapkan dapat :

Memahami Pemformatan Teks

Memahami Pemformatan Paragrap Web

Uraian Materi

Pemformatan Teks

Berikut ini adalah contoh yang dihasilkan pemformatan teks pada dokumen web :

Teks ini berformat tebal atau bold

Teks ini berformat strong

Teks ini berformat miring atau italic

Teks ini berformat emphasized

`Teks ini berformat keluaran komputer`

Teks ini berformat dibawah garis atau $_{\text{subscript}}$ dan di atas garis atau $^{\text{superscript}}$

HTML menggunakan tag seperti <b> dan <i> untuk memformat output, seperti teks tebal atau miring.

Tag HTML ini disebut tag format, selengkapnya dalam table berikut :

Tag HTML Pemformatan Teks

Tag	Deskripsi



<b>	Mendefinisikan teks tebal
	Mendefinisikan teks menekankan
<i>	Mendefinisikan teks miring
<small>	Mendefinisikan teks kecil
	Mendefinisikan teks penting
<sub>	Mendefinisikan teks di bawah garis
<sup>	Mendefinisikan teks di atas garis
<ins>	Mendefinisikan teks sisipan
<del>	Mendefinisikan teks dicoret
<mark>	Mendefinisikan teks ditandai

Tag HTML "keluaran computer"

Tag	Deskripsi
<code>	Mendefinisikan teks kode komputer
<kbd>	Mendefinisikan teks keyboard
<samp>	Mendefinisikan teks contoh kode
<var>	Mendefinisikan teks variabel
<pre>	Mendefinisikan teks terformat

HTML Citations, Quotations, and Definition Tags

Tag	Deskripsi
<abbr>	Mendefinisikan sebuah singkatan
<address>	Mendefinisikan alamat atau kontak informasi
<bdo>	Mendefinisikan arah teks

<blockquote>	Mendefinisikan sebuah bagian yang dikutip dari sumber lain
<q>	Mendefinisikan sebuah kutipan pendek
<cite>	Mendefinisikan judul karya
<dfn>	Mendefinisikan sebuah istilah definisi

2. Pemformatan Paragrap

Tag HTML untuk paragrap adalah , dengan tag penutup . Berikut ini contohnya :

```
<p>ini adalah paragrap</p>

<p>Ini paragrap yang lain</p>
```

c. Rangkuman

Pemformatan teks digunakan untuk memformat teks, seperti huruf tebal, miring dsb. .

d. Tugas

Buatlah dokumen-dokumen HTML berikut ini, diskusikan hasilnya dengan teman-temanmu.

```
<html>

<head>

<title>Format Teks</title>

</head>

<body>
```




```
<p>
<b>Tulisan ini ditebalkan (bold)</b>
</p>
<p>
<strong>Tulisan ini ditebalkan
(strong)</strong>
</p>
<p>
<em>Tulisan ini miring (emphasize)
</em>
</p>
<p>
<big>Tulisan ini besar (big)</big>
</p>
<p>
<i>Tulisan ini miring (italic) </i>
</p>
<p>Tulisan ini
<sub>subscript</sub>
</p>
<p>Tulisan ini
<sup>superscript</sup>
</p>
```

```
</body>

</html>
```

```
<HTML>

<HEAD>

<TITLE>Preformat </TITLE>

</HEAD>

<BODY>

<H2>Daftar Harga Keladi:</H2>

<PRE>

Red Flash.........35.000

Red Fire..........60.000

Fannie Munson.....60.000

</PRE>

Harga sewaktu-waktu bisa berubah.

</BODY>

</HTML>
```

```
<html>

<head>
```




```
<title>Tag Preformatted</title>
</head>
<body>
<pre>
 This section provides a brief overview of the menus in
Dreamweaver.
   The File menu and Edit menu contain the standard
   menu items for File and Edit.
menus, such as New, Open, Save, Save All, Cut, Copy, Paste,
Undo,and Redo
</pre>
<pre>
Ini adalah
preformatted text.
Menampilkan       spasi
Dan line break apa adanya.
</pre>
<p>Tag PRE cocok digunakan untuk menampilkan kode bahasa
pemrograman komputer :</p>
<pre>
for i = 1 to 10
     print i
next i
```

```
</pre>

</body>

</html>
```

```
<html>

<head>

<title>quotation</title>

</head>

<body>

tulisan ini BUKAN quotation

<blockquote>

tulisan ini adalah quotation <br>

tulisan ini adalah quotation <br>

tulisan ini adalah quotation <br>

</blockquote>

</body>

</html>
```

e. Tes Formatif



Lembar Jawaban Tes Formatif

Lembar Kerja Siswa

5. Kegiatan Belajar 5 : Pembuatan List Minimal

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 5 ini siswa diharapkan dapat :

Memahami Pembuatan list minimal

Menyajikan hasil pembuatan list minimal

b. Uraian Materi

Pengertian List

List adalah bagian teks di dalam dokumen yang berisi daftar item dari suatu kelompok atau grup tertentu. Sebagai contoh, dalam situs web yang bertema kuliner, list dapat berupa daftar dari makanan dan minuman, beserta harganya. List dapat juga berupa prosedur (urutan langkah-langkah) dari suatu pekerjaan tertentu, yang sifatnya harus dilakukan secara berurutan dari langkah pertama sampai langkah terakhir.

Contoh List :

Daftar Makanan :

Bakso

Soto

Sate

Gule

Daftar Minuman :

Juice Jambu

Juice Alpukat

Juice Tomat

Tipe Daftar dalam Dokumen HTML

Dalam dokumen HTML, tipe daftar dibedakan menjadi tiga, yaitu :

Daftar berurutan (ordered list)



Daftar tidak berututan (unordered list)

Daftar definisi (definition list)

Daftar Berurutan (Ordered List)

Daftar yang berurutan biasanya ditandai dengan penggunaan penomoran tertentu. Penomoran ini bias menggunakan angka (1, 2, 3,…) maupun karakter alphabet terntentu (a, b, c,… atau i, ii, iii,…). Daftar yang berurutan umumnya dipakai untuk item-item yang saling berhubungan satu sama lain, atau untuk menuliskan langkah-langkah atau prosedur dari kegiatan terntentu yang tidak dapat dilakukan secara acak.

Contoh daftar yang berurutan adalah pada cara membuat dokumen HTML, misalnya :

Langkah-langkah membuat dokumen HTML :

Jalankan aplikasi Text Editor

Isikan kode HTML ke dalam Text Editor

Simpan file dengan ekstension .htm atau .html

Jalankan file HTML menggunakan aplikasi Web browser untuk menampilkan hasilnya

Untuk membuat suatu daftar yang berurutan dalam dokumen HTML, digunakan tag <ol> (ordered list), yang berpasangan dengan tag . Tag <ol> digunakan untuk memulai suatu daftar berurutan tertentu, sedangkan berfungsi untuk menandakan bahwa daftar tersebut sudah berakhir.

Masing-masing item di dalam daftar harus dibuat menggunakan tag <li> (list item, yang kemudian perlu ditutup dengan tag . Contoh penulisan adalah sebagai berikut :

```
<ol>
<li>Urutan Pertama</li>

<li>Urutan Kedua</li>
```

```
</ol>
```

Adapun atribut untuk tag <ol> adalah “type” yang menunjukan jenis penomorannnya. Secara standart, tipenya adalah menggunakan penomoran 1, 2, 3, 4 dst.

Selain tipe default, tipe lain yang dapat digunakan untuk keperluan penomoran dalam suatu list adalah :

Tipe	Keterangan
A	Membuat *list* dengan penomoran berupa karakter A, B, C, dst
a	Membuat *list* dengan penomoran berupa karakter A, B, C, dst
I	Membuat *list* dengan penomoran berupa karakter A, B, C, dst
i	Membuat *list* dengan penomoran berupa karakter A, B, C, dst

4) Daftar Tidak Berurutan (Unordered List)

Daftar yang tidak berurutan adalah daftar yang item-itemnya dapat diubah posisinya secara acak. Daftar seperti ni tidak menggunakan penomoran seperti di atas, melainkan hanya dengan menggunakan tanda atau symbol tertentu, misalnya gambar kotak atau bulat (bullet). Contoh :

Daftar buah :

- Mangga
- Jambu
- Semangka
- Rambutan
- Jeruk

Pada contoh di atas, kita dapat mengubah/mengacak secara bebas urutan dari masing-masing item yang ada. Hal ini disebabkan karena setiap item yang ada



tidak memiliki keterkaitan satu sama lain dengan item lainnya, berbeda dengan suatu urutan langkah tertentu yang harus ditulis secara berurutan.

Untuk membuat suatu daftar yang tidak berurutan dalam dokumen HTML, digunakan tag <ul> (unordered list), yang berpasangan dengan tag . Tag <ul> digunakan untuk memulai suatu daftar berurutan tertentu, sedangkan berfungsi untuk menandakan bahwa daftar tersebut sudah berakhir.
Masing-masing item di dalam daftar harus dibuat menggunakan tag <li> (list item, yang kemudian perlu ditutup dengan tag . Contoh penulisan adalah sebagai berikut :

```
<ul>
<li>Item satu</li>

<li>Item dua</li>

</ul>
```

Adapun atribut untuk tag <ul> adalah “type” yang menunjukan jenis tanda untuk setiap item berupa tanda bulatan untuk defaultnya..
Selain tipe default, tipe lain yang dapat digunakan untuk keperluan unordered list adalah :

Tipe	Keterangan	Contoh Penulisan
disk	Tanda bulatan hitam	<ul type=”disk”>
circle	Tanda bulatan putih	<ul type=”circle”>
square	Tanda kotak	<ul type=”square”>

5) Daftar Definisi (Definition List)

Daftar Definisi adalah suatu daftar yang berisi kumpulan definisi dari suatu istilah tertentu, misalnya daftar definisi istilah HTML. Contoh :

HTML

Hyper Text Mark-up Language

WWW

World Wide Web

Browser

Aplikasi untuk membuka dokumen html/situs web.

Pada contoh di atas, kita dapat mengubah/mengacak secara bebas urutan dari

Rangkuman

Tugas

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa



6. Kegiatan Belajar 6 : Pembuatan List Kombinasi

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 5 ini siswa diharapkan dapat :

Memahami Pembuatan list kombinasi

b. Uraian Materi

Pembuatan list Kombinasi

Perhatikan daftar berikut ini :

Daftar Menu

Makanan

Bakso

Mie Ayam

Soto

Minuman

Teh Panas

Jeruk Panas

Kopi

Dapat dilihat bahwa daftar di atas adalah gabungan daftar berurutan dan daftar tidak berurutan.

Untuk membuat daftar diatas pada dokumen HTML maka cukup menggabungkan <ol> dangan <ul>.

Perhatikan penggabungan <ol> dengan <ul> berikut ini :

```
<ol>
```

```
    <li></li>
    <li>
      <ul>
         <li></li>
         <li></li>
         <li></li>
      </ul>
   </li>
</ol>
```

Ketika kita menggabungkan tag tidak berurutan <ul> tipe bullet dengan tag tidak berurutan <ul> pada daftar berikutnya, maka daftar yang kedua oleh browser akan ditampilkan secara otomatis menjadi daftar bullet.
Tidak demikian dengan daftar berurutan <ol>, maka agar tipenya sama dengan daftar berurutan sebelumnya, kita harus memberikan atribut tipe yang sama dengan daftar berurutan sebelumnya. Contoh :

```
    <li> </li>
         <ol type=i>
            <li > </li>
            <li> </li>
         </ol>
    <li>
         <ol type=i>
           <li > </li>
```




```
            <li> </li>
          </ol>
    </li>
    <li>
    </li>
</ul>
```

Memulai dengan nomor tertentu

Untuk membuat daftar bernomor dengan urutan yang tidak dimulai dengan 1 (nomor/huruf awal), maka kita harus mendefinisikan dengan atribut “start” pada elemen tag <ol>.

Misalnya contoh berikut :

```
<ol start="17">
<li>Blok huruf dengan text tool</li>
<li>Klik tab karakter</li>
<li>Pilih jenis huruf pada menu pop-up .</li>
</ol>
```

Maka urutan daftar yang ditampilkan akan dimulai dengan 17, selanjutnya 18 dan 19.

Demikian juga apabila :

```
<ol type=”A” start="17">
<li>Blok huruf dengan text tool</li>
<li>Klik tab karakter</li>
<li>Pilih jenis huruf pada menu pop-up
.</li> </ol>
```

Maka urutan daftar yang ditampilkan akan dimulai dengan Q, selanjutnya R, dan S.

Penggunaan daftar bernomor yang tidak dimulai dengan nomor awal, biasanya diperlukan apabila daftar yang kita buat sudah diselingi beberapa paragraph, sehingga ketika dibuat daftar bernomor, secara default akan dimulai dari awal, padahal harusnya melanjutkan nomor sebelumnya. Maka digunakanlah atribut “start=” tersebut.

Rangkuman

Tugas

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa

.



7. Kegiatan Belajar 7 : Pemformatan Tabel

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 7 ini siswa diharapkan dapat :

Memahami Pembuatan Tabel

b. Uraian Materi

Pembuatan Tabel

Tabel merupakan cara untuk menampilkan informasi dalam bentuk sel yang terdiri atas baris dan kolom. Untuk menampilkan data dalam bentuk table pada HTML, digunakan Tag<Table>.

Elemen table berisi property <tr> untuk menentukan baris(table row) yang didalamnya terdapat properti <td> untuk menampkan data pada setiap sel table(table data).

Struktur elemen table adalah sebagai berikut:

```
<table>
  <tr>
    <td>data baris 1 kolom 1</td>
    <td>data baris 1 kolom 2</td>
  </tr>
</table>
```

Atribut Elemen Tabel

Width	=panjang(lebar table, pixel atau persen)
Height	=panjang(tinggi table, pixel atau persen)
Border	=pixel(tebal garis tepi)
Cellspacing	=pixel(spasi antar sel)
Cellpadding	=pixel(spasi di dalam sel)
Align	=[left|center|right](perataan table)
Bgcolor	=warna(warna latar belakang table)

Atribut Table Row

Align	=[left|center|right](perataan sebaris sel secara horizontal)
Valign	=[top|middle|bottom](perataaan sebaris sel secara vertical)
Bgcolor	=warna(warna latar belakang baris)

Atribut Tabel Data

Align	=[left|center|right](perataan horizontal)
Width	=[top|middle|bottom](perataan vertical)
Height	=pixel(tinggi sel, pixel atau persen)
Bgcolor	=warna(warna latar belakang sel)



Rangkuman

Tugas

01:tabel1.html

Salin script HTML berikut untuk mencoba tag table berikut property kolom dan barisnya, jangan lupa disimpan dengan ekstensi .html.kemudian coba di web browser.

```
<html>
 <head>
  <title>tabel</title>
 </head>
<body>
berikut contoh tabel dengan rowspan dan
colspan
<table width=80% border=2 cellspacing=0
cellpadding=0>
 <tr>
  <td>baris 1 kolom 1</td>
  <td>baris 1 kolom 2</td>
 </tr>
 <tr>
  <td colspan=2>baris 2 kolom 1</td>
```

```
 </tr>
 <tr>
  <td rowspan=2baris 3 kolom 1</td>
  <td>baris 3 kolom 2</td>
 </tr>
</table>
</body>
</html>
```

02: tabel2.html




```
<html>
  <head>
    <tittle>penggunaan
  ALIGN</tittle> </head>
<body>

<table border="1">
<caption>Daftar
wiraniaga</caption> <tr>
   <th colspan ="2" rowspan = "2">WIRANIAGA</th>
   <th colspan ="3">KOTA</th>
</tr>
<tr>
   <th>SEMARANG</th>
   <th>Kudus</th><th>Solo</th>
<tr>
<tr>
   <th rowspan ="2">Jenis
   Kelamin</th> <th>Pria</th>
   <th align= "right">30</td>
```

```
    <td align= "right">20</td>
    <td align= "right">30</td>
</tr>
<tr>
    <th>wanita</th>
    <td align= "right">20</td>
    <td align= "right">8</td>
    <td align= "right">18</td>
</tr>

</table>
</body>
                </html>
```

03: tabel3.html

Buatlah script HTML sehingga menghasilkan tampilan tabel seperti berikut ini (aturlah sedemikian rupa pada atribut-atribut tabel untuk mendapatkan dua buah tampilan yang berbeda, sebagaimana dibawah ini).

Nama	**Usia**



Ali	25
Fahmianto	27

Nama	Usia
Ali	25
Fahmianto	27

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa

8. Kegiatan Belajar 8 : Tabel dengan Spaning

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 8 ini siswa diharapkan dapat :

Memahami Tabel dengan Spaning

b. Uraian Materi

Pembuatan Tabel

Ada kalanya kita membuat table dengan menggabungkan baris, ataupun kolom.

Berikut ini contoh table yang menggabungkan baris :

Benua	Negara
ASIA	Arab Saudi
	India
	Indonesia
	Singapura

Berikut ini contoh table yang menggabungkan kolom :

Benua	EROPA			
Negara	Belanda	Italia	Inggris	Jerman

Untuk membuat table seperti di atas, maka Tabel Atribut Data yang digunakan adalah sebagai berikut

Rowspan =angka(baris yang di span oleh sel)

Colspan =angka(kolom yang di span oleh sel)

Jadi untuk menggabungkan 4 baris seperti table di atas adalah :

<td rowspan=4> …….</td>

Sedangkan untuk menggabungkan 4 kolom seperti table di atas adalah :

<th colspan=4> …….</th>



c. *Rangkuman*

Buatlah dokumen HTML berikut ini :

d. *Tugas*

Buatlah dokumen HTML berikut ini :

```
<html
<head>
<title>Tabel Dengan Rowspan</title>
</head>

<body>
<table width="500" border="1">
 <tr>
  <th scope="col">Nama Perumahan </th>
  <th scope="col">Tipe / Luas tanah (m<sup>2</sup>) </th>
 </tr>
 <tr>
  <td rowspan="4">Griya Indah Permai
  </td> <td>21 / 60 </td>
 </tr>
 <tr>
  <td>36 / 90 </td>
 </tr>
 <tr>
```

```
    <td>45 / 120 </td>
  </tr>
  <tr>
    <td>54 / 120 </td>
  </tr>
</table>
</body>
</html>
```

Buatlah dokumen HTML berikut ini :

```
<html >
<head>
<title>Tabel dengan Colspan</title>
</head>

<body>
<table width="500" border="1">
  <tr>
    <th scope="col">Perumahan</th>
    <th colspan="4" scope="col">Tipe / Luas tanah (m<sup>2</sup>)
</th>
  </tr>
  <tr>
    <td>Griya Indah Permai </td>
```




```
<td>21/60</td>
<td>36/90</td>
<td>45/120</td>
<td>54/120</td>
</tr>
</table>
</body>
</html>
```

3. Buatlah dokumen HTML dari tampilan table berikut ini :

Benua	Negara
ASIA	Arab Saudi
	India
	Indonesia
	Singapura
EROPA	Belanda
	Italia
	Inggris
	Jerman

4. Buatlah dokumen HTML dari tampilan tabel berikut ini :

Benua	EROPA		ASIA	
Negara	Belanda	Italia	Indonesia	India

5. Buatlah dokumen HTML dari tampilan table berikut ini :

apel		mangga
pisang	jeruk	nanas
leci		

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa



9. Kegiatan Belajar 9 : Pembuatan Tabel dalam Tabel

10. Kegiatan Belajar 10 : Desain Halaman Web dengan Konsep Tabel



11. Kegiatan Belajar 11 : Format Tampilan Gambar

a. *Tujuan Pembelajaran*

Setelah mengikuti kegiatan belajar 8 ini siswa diharapkan dapat :

Memahami Format tampilan gambar

b. *Uraian Materi*

Memasukkan Gambar ke dalam Halaman Web

Suatu halaman *web* akan terlihat kaku, terkesan formal, dan sedikit menjemukan bila tidak disertai dengan gambar. Anda bisa lihat saat ini, halaman *web* yang ada di i*nternet*, hampir semuanya memasukkan unsur gambar, animasi, bahkan audio dan video untuk menarik dan membuat tercengang para pengunjungnya. Cara menambahkan audio dan video akan kita bahas pada bagian lain dalam buku ini. Untuk saaat ini, kita hanya akan membahas tentang gambar.

Sebagai contoh lain, misalnya anda ingin membuat halaman *web* yang berisi koleksi foto anda, maupun barang-barang yang sesuai dengan hobi anda. Dalam kasus ini, tentu anda harus memahami teknik untuk menampilkan foto tersebut ke dalam halaman web sehingga akan tampil rapi dan menarik.

Cara yang diperlukan untuk memasukkan atau menambahakan gambar ke dalam suatu halaman *web* sangatlah mudah. HTML telah menyediakan tag khusus untuk keperluan ini, yaitu <img>. Tag ini merupakan tag tunggal atau tidak memiliki pasangan. Atribut terpenting dari tag <img> adalah SRC (source atau sumber), yang berisi file gambar yang akan ditampilkan ke dalam halaman *web*.

Bentuk umum penggunaan tag <img> adalah :

Jika *file* yang akan ditampilkan berada di direktori lain, atau bahkan berada di situs *web* lain, makan anda perlu menyebutkan juga lokasi dari *file* tersebut.

Contoh:

<img src= “../image/wrangler-rubicon.jpg”>

Atau

Format gambar yang sering digunakan dalam halaman web adalah GIF (.gif) dan JPEG (.jpg atau . jpeg). GIF adalah singkatan dari *Graphics Interface Format* sedangkan JPEG, adalah singkatan dari *Joint Photographic Expert Group.* Selain kedua format tersebut, saaat ini juga sudah mulai banyak digunakan gambar dengan format PNG (.png), yanag merupakan singkatan dari *Portable Network graphics*.

Menggabung Gambar dan Teks

Jika kita ingin menggabungkan ataumenyisipkan gambar di dalam suatu teks atau paragraf tertentu, terdapat pilihan posisi gambar yang harus diperhatikan:

Secara vertikal:

Teks bisa berada sejajar dengan bagian atas gambar

Teks bisa berada sejajar dengan bagian tengah gambar

Teks bisa berada sejajar dengan bagian bawah gambar

Secara horisontal:

Gambar bisa berada di bagian kiri teks/paragraf

Gambar bisa berada di bagian kanan teks/paragraf

Untuk posisi gambar yangdilihat dari arah vertikal dapat ditentukan dengan mengisi nilai TOP, MIDDLE, atau BOTTOM ke dalam atribut ALIGN yang terdapat pada tag <img>.

Adapun untuk posisi gambar yang dilihat dari arah horisontal ditentukribut an dengan mengisi nilai LEFT dan RIGHT kedalam ALIGN pada tag <img>. Dengan mengisi atribut tersebut dengan nilai LEFT maka gambar akan berada di sebelah kiri teks. Sebaliknya, jika diisi RIGHT akan menyeabkan posisi gambar akan berada di sebelah kanan teks.



Memperkecil dan Memperbesar Ukuran Gambar

gambar yang ditampilkan pada dokumen HTML terkadang tidak memiliki ukuran gambar yang sama dengan ukuran sebenarnya. Untuk menghasilkan gambar dengan kualitas yang mirip aslinya, penentuan ukuran juga menjadi hal yang penting untuk diperhatikan. Maka dari itu perbandinganya harus sesuai dengan ukuran asli dari gambar tersebut. Untuk menentukan ukuran gambar, kita perlu mengisi atribut *HEIGHT* berfungsi untuk menetukan tinggi gambar dan width untuk menetukan lebar gambar. Perhatikan contoh berikut ini.

Kode diatas akan menampilkan gambar dari *file* jeep.jpg yang berada dalam direktori images sesuai dengan ukuran asli (misal 500X375 pixel).

Untuk memperkecil gambar tersebut, kita tinggal memperkecil nilai-nilai tersebut, misalnya

Memberi keterangan pada gambar

Agar lebih terkesan informatif, anda dapat memberikan keterangan yang relevan dan seperlunya (tidak berlebihan) terhadap gambar yang anda tampilkan pada halaman web. Keterangan ini akan muncul dalam bentuk tooltip (sesaat, hanya beberapa detik) pada saat penunjung mengarahkan kursor (penjujuk mouse) ke atas gambar.

Untuk membuat keterangan semacam ini, kita dapat menggunakan atribut TITLE maupun ALT pada tag <img>.

Rangkuman

…….

d. Tugas

Pratikum 4-1: Memasukkan Gambar ke dalam Halaman Web

Jalankan aplikasi *Text Editor*

Buatlah direktori baru dengan nama **images** di dalam direktori kerja anda

Copy file gambar yang akan di tampilkan ke dalam **D: \ MyHTML \ images**

Catatan : *Dalam contoh ini penulis menggunkana **file rubicon.gif, rubicon.jpeg, dan rubicon.png.***

Isikan kode berikut:

```
<html>
<head>
        <title>Gambar</title>
</head>
<body>
<p>Demo Menambahkan Gambar GIF, JPG, dan PNG</h2>

<p>Menampilkan gambar dalam format GIF:</p>
<img src=”images/rubicon.gif”?>

<p>Menampilkan gambar dalam format JPG:</p>
<img src=”images/rubicon.jpg”/>

<p>Menampilkan gamabr dalam format
PNG:</p> <img src=”images/rubicon.png”/>
```




```
</body>

</html>
```

Simpan *file* dengan nama **gambar.html** dan tempatkan ke dalam direktori kerja
Jalankan *file* tersebut melalu *web browser*

Pada contoh di ats, kita menampilkan tiga gamabar dengan format yanag berbeda. Gambar pertama (paling atas) menggunakan format GIF, Kedua JPG, dan ketiga PNG. Seperti yang anda lihat, format-format tersebut dapat ditampilkan dengan baik di dalam halaman web. Nilai yang kita isikan untuk atribut SRC di atas mengandung nama direktori **images.**

```
<img src=”images/rubicon.gif”/>

<img src=”images/rubicon.jpg”/>

<img src=”images/rubicon.png”/>
```

Ini artinya *file* tersebut disimpan di ddirektori images, dimana direktori images itu sendiri berada dalam direktori kerja (satu direktori dengan *file* **gambar.html**).

Pratikum 4-2: Menggabung Gambar dan Teks (bagian 1)
Jalankan aplikasi *Text Editor*

Copy file yang akan ditampilakan ke dalam **D:\MyHTML\images Catatan***:*
Dalam contoh ini penulis menggunakan file yang sama seperti pada pratikum 4-1.
Isikan kode berikut :

```
<html>

<head>

        <title>Gambar</title>

</head>

<body>

<h2>Demo Menggabung Gambar dan teks (bag.1)</h2>
```

```
<p>
<img src=”images/rubicon.gif”?> align=”top”/>
Jeep Wranger 1 (menggunakan align=TOP)
</p>

<p>
<img src=”images/rubicon.jpg”/> align=”middle”/>
Jeep Wranger 2 (menggunakan align=MIDDLE)
</p>

<p> <img src=”images/rubicon.png”/> align=”bottom”/>
Jeep Wranger 3 (menggunakan align=BOTTOM)
</p>

</body>
</html>
```

Simpan *file* dengan nama **gambar_plusteks.html** dan tempatkan ke dalam direktori kerja
Jalankan *file* tersebut melalui *web browser.*

Dari hasil di atas tanpak jelas bahwa pada gambar bagian pertama teks berada sejajar dengan bagian atas gambar. Untuk melakukan hal ini, kita perlu mengisi atribut ALIGN dengan nilai TOP.

```
<img src=”images/rubicon.gif”align=”top”/>
```



Pada gamabr kedua, teks berada di tengah-tengah gambar karena kita mengisi nilai MIDDLE pada atribut ALIGN.

Terakhir, pada gambar ketiga, teks berada sejajar dengan bagian bawah gamabar karena kali ini atibut ALIGN BOTTOM

Pratikum 4-3: Menggabung Gambar dan Teks (Bagian 2)

Jalankan aplikasi *Teks Editor*

Copy file gambar yang akan ditampilkan ke dalam **D:\MyHTML\images**

Catatan: *Dalam contoh ini penulis menggunakan file yang sama seperti pada Pratikum 4-1.*

Isikan kode berikut :

```
<html>

<head>

        <title>Gambar</title>

</head>

<body>

<h2>Demo Menggabung Gambar dan Teks (Bag. 2) </h2>


<p><u>Gambar ada di sebelah kiri teks:</u></p>

<p>

<img src=”images/rubicon.gif” align”left”/>
```

Pada paragraf ini gambar akan ditempatkan di sebelah kiri teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai LEFT. Pada paragraf ini gambar akan ditempatkan di sebelah kiri teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai LEFT. Pada paragraf ini gambar akan ditempatkan di sebelah kiri teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai LEFT. Pada paragraf ini gambar akan ditempatkan di sebelah kiri teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai LEFT. Pada paragraf ini gambar akan ditempatkan di sebelah kiri teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai LEFT.

p><u>Gambar ada di sebelah kanan teks:</u>

Pada paragraf ini gambar akan ditempatkan di sebelah kanan teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai RIGHT. Pada paragraf ini gambar akan ditempatkan di sebelah kanan teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai RIGHT. Pada paragraf ini gambar akan ditempatkan di sebelah kanan teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai RIGHT. Pada paragraf ini gambar akan ditempatkan di sebelah kanan teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai RIGHT. Pada paragraf ini gambar akan ditempatkan di sebelah kanan teks. Hal ini disebabkan atrbut ALIGN diisi dengan nilai RIGHT.

</body>

</html>

Simpan file dengan nama **gambar_plusteks2.html** dan tempatkan ke dalam direktori kerja.

Jalankan file tersebut melalui web browser

Praktikum 4-4: Memperkecil gambar

Jalankan aplikasi *text editor*

Copy file gambar yang akan ditampilkan ke dalam D: \MyHTML\ images



Catatan: Dalam contoh ini penulis menggunakan file jeep.jpg.
Isikan kode berikut:

```
<html>
<head>
        <title>gambar</title>
<head>
<body>
<h2>demo memperkecil gambar</h2>

<p>
<u>gambar ukuran asli (500X375 pixel :</u><br />
<img src=images/jeep.jpg" />
</p>

<p>
<u> gambar yang diperkecil (320X230 pixel) :</u><br />
<img src="images/jeep.jpg" width ="320" height="230" />
<p>
<body>
<html>
```

Simpan file dengan nama gambar_perkecil.html dan tempatkan kedalam direktori kerja
Jalankan file tersebut melalui web browser

Sebaliknya, untuk memperbesar gambar tersebut, kita perlu memperbesar nilai-nilai dari atribut di atas, misalnya menjadi 600X420 pixel. Kodenya akan tampak seperti berikut:

Praktikum 4-5: Memperbesar Gambar

Jalankan aplikasi *text editor*

Copy file gambar yang akan ditampilkan ke dalam D: \MyHTML\ images

Catatan: Dalam contoh ini penulis menggunakan file jeep.jpg.
Isikan kode berikut:

```
<html>
<head>
        <title>gambar</title>
<head>
<body>
<h2>demo memperkecil gambar</h2>

<p>
<u>gambar ukuran asli (500X375 pixel :</u><br />
<img src=images/jeep.jpg” />
</p>

<u> gambar yang diperkecil (600X440 pixel) :</u><br />
<img src=”images/jeep.jpg” width =”600” height=”440” />
<p>
<body>
<html>
```



Simpan file dengan nama gambar_perkecil.html dan tempatkan kedalam direktori kerja
Jalankan file tersebut melalui web browser

Sebaliknya, untuk memperbesar gambar tersebut, kita perlu memperbesar nilai-nilai dari atribut di atas, misalnya menjadi 600X420 pixel. Kodenya akan tampak seperti berikut:

Hati-hati dalam memperbesar gambar, jangan sampai gamabar terlihat pecah. Cari nilai-nilai optimal untuk mengubah nilai dari tersebut WIDTH dan HEIGHT. Selamat Mencoba!.

Praktikum 4-6: memberi keterangan pada gambar

Jalankan aplikasi *text editor*

Copy file gambar yang akan ditampilkan ke dalam D: \MyHTML\ images
Catatan : Dalam contoh ini penulis menggunakan file offroad.jpg.
Isikan kode berikut:

```
<html>

<head>

        <title>gambar</title>

<head>

<body>

<h2>demo pemberi keterangan pada gambar</h2>

<p>arahkan penunjuk mouse anda ke atas gambar dibawah ini untuk melihat keterangan yang dimaksud!</p>

<p>

<img src=images?offroad.jpg”

        Title=”offroad – rubicon sedang melintasi sungai” />

</p>
```

```
</body>

</html>
```

Simpan file dengan nama gambar_perkecil.html dan tempatkan kedalam direktori kerja
Jalankan file tersebut melalui web browser

Coba Anda perhatikan teks toolip yang muncul pada hasil di atas.teks tersebut adalah teks yang diisikan ke dalam atribut title pada tag <img>. Untuk keperluan yang sama, anda juga dapat menggunakan atribut ALT.

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa



12. Kegiatan Belajar 1 2 : Format Tampilan Audio

13. Kegiatan Belajar 13 : Format Tampilan Video dan Animasi



14. Kegiatan Belajar 14 : Format Tampilan Gambar dengan Map

15. Kegiatan Belajar 15 : Anatomi Hyperlink

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 10 ini siswa diharapkan dapat :

Memahami Anatomi hyperflink

b. Uraian Materi

Pembuatan Tabel

Salah satu kelebihan dokumen HTML adalah memiliki kemampuan untuk saling terkait dengan dokumen lain. Link (sebutan singkat dari hyperlink text) adalah suatu metode yang digunakan dalam HTML untuk membuat hubungan antar halaman yang terdapat dalam satu situs web. Semua halaman web yang ada di internet hampir bisa dipastikan memiliki link. Secara umum, fungsi link adalah untuk memudahkan pengunjung web dalam menulusuri/menjelajahi seluruh isi atau informasi yang tersimpan dalam situs web bersangkutan. Dalam keadaan default (normal), suatu link akan ditandari dengan teks berwarna biru yang memiliki garis bawah. Meskipun demikian, kita dapat mengubah warna dan gaya link sesuai dengan kebutuhan dan keinginan kita melalui pembuatan kode CSS.jika kita mengarahkan kursor ke suatu link tertentu, maka penunjuk mouse akan berubah menjadi gambar tangan dengan satu jari yang sedang menunjuk, seperti yang ditunjukkan oleh gambar di bawah ini.

Contoh link



Konsep atau cara kerja link dalam HTML dapat diilustrasikan melalui gambar berikut :



Pada gambar diatas, masing-masing halaman memiliki keterkaitan satu sama lain. Artinya, masing-masing halaman tersebut dapat dipanggil atau ditampilkan dari halaman manapun. Hal ini bisa terjadi jika setiap halaman di atas memiliki link.

Mengenal Tag <a> (anchor)

Untuk membuat suatu link, HTML menyediakan satu tag khusus, yaitu tag <a>, yang berpasangan dengan . Setiap teks yang diapit oleh tag ini akan ditampilkan sebagai link. Bentuk umum pembuat link dalam dokumen HTML adalah sebagai berikut :

```
<a  href="DokumenLain">Teks
```

Atribut HREF dalam tag <a> berfungsi untuk menentukan dokumen atau halaman mana yang akan dipanggil saat link tersebut di klik atau dipilih oleh pengunjung web.

Contoh :

```
<a href=”index.html”>Home</a><br />

<a href=”about.html”>Tentang Kami</a><br />

<a href=”contact.html”>Kontak</a>
```

Pada contoh diatas kita membuat tiga buah link dengan teks : Home, Tentang kami, dan kontak. Ketika pengunjung melakukan klik terhadap link pertama (home), web browser akan menampilkan dokumen yang disimpan dalam file index.html. Begitu juga dengan link kedua (Tentang Kami) dan ketiga (Kontak), setiap di klik akan menjadikan web browser menampilkan dokumen yang terdapat pada file about.html dan contact.html.

Rangkuman

……

d. Tugas

Jalankan aplikasi Text Editor

Buatlah dokumen HTML berikut :

```
<html>

<head>

<title>Link</title>

</head>

<body>

<h2>Demo Membuat Link</h2>


<h3><u>Dokumen 1</u></h3>


<p>Klik link di bawah ini untuk melihat isi halaman
web pada dokumen2.html:</p>


<p><a href=”link-dokumen2.html”>Lihat
Dokumen 2</a></p>


</body>
</html>
```

Simpan file dengan nama **link-dokumen1.html** dan tempatkan ke dalam direktori kerja.

Buatlah file baru, lalu buatlah dokumen HTML berikut :

```
<html>

<head>

<title>Link</title>

</head>
```




```
<body>

<h2>Demo Membuat Link</h2>


<h3><u>Dokumen 2</u></h3>


<p>

Ini adalah isi halaman yang terdapat pada dokumen 2.
Untuk kembali ke dokumen 1, silahkan klik link di
bawah ini.

</p>


<p>

<a href=”link-dokumen1.html”>Lihat Dokumen 1</a>

</p>

</body>
</html>
```

Jalankan file link-dokumen1.html melalui web browser

Klik link yang ada pada dokumen1. Apa yang kamu lihat ? halaman web akan berpindah ke dokumen 2 bukan ?

Klik link yang ada pada dokumen 2. Kali ini, kamu akan melihat kembali isi halaman dari dokumen1.

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa



16. Kegiatan Belajar 16 : Jenis – Jenis Link dalam HTML

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 11 ini siswa diharapkan dapat :

Memahami jenis-jenis link dalam HTML

b. Uraian Materi

Jenis-jenis Link dalam HTML

Dalam HTML, link dibedakan menjadi 3 jenis, yaitu :

Link absolut

Link relatif

Link ke bagian dokumen tertentu

HTML membedakan ketiga jenis link diatas berdasarkan lokasi atau alamat dokumen yang akan diakses. Dengan demikian, perbedaannya hanya terletak pada nilai atribut HREF-nya saja.

Link Absolut

Link Absolut adalah link yang akan menunjuk ke halaman dari situs web lain. Penulisan alamatnya pun harus ditulis secara lengkap. Sebagai contoh, jika tema atau isi dari situs web yang kita buat adalah pemrograman HTML, maka kita dapat membuat link ke situs web lain yang memiliki tema yang sama atau relevan. Tujuannya adalah agar pengunjung bisa memperoleh informasi lebih tentang tema tersebut yang ‘mungkin’ tidak/belum kita bahas pada situs web kita.

Contoh :

```
<a href=http://www.w3.org/>W3</a>
```

Pada contoh diatas, tujuan link yang dibuat akan mengarah ke situs web http://www.w3.org/.

Link Relarif

Link Relatif adalah link yang tujuannya mengarah ke dokumen-dokumen lain yang masih berada di dalam satu situs web yang sama (komputer yang sama). Penulisan alamat tujuannya pun tidak perlu ditulis secara lengkap, cukup nama dokumennya saja, dan nama direktorinya (jika ada).

Perhatikan contoh kode berikut :

```
<a href=”kontak.html”>Kontak</a>

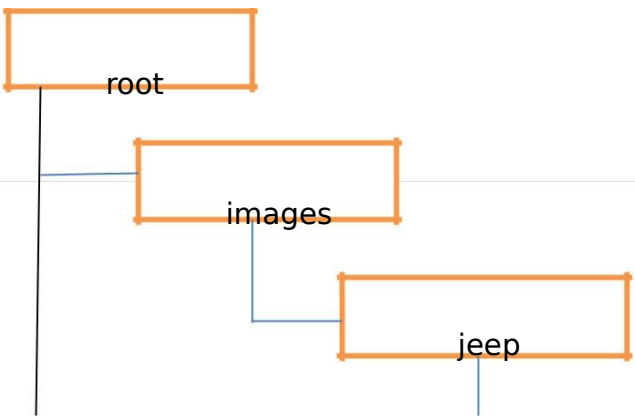
<a href=”images/jeep/rubicon.jpg”>Lihat Foto Rubicon</a>
```

Tidak perlu ditulis lengkap seperti berikut :

```
<a href=http://www.situskita.com/kontak.html>Kontak</a>

<a href=http://www.situskita.com/images/jeep/rubicon.jpg> Lihat Foto Rubicon</a>
```

Kita tidak perlu menulis alamat secara lengkap karena server web akan mencari file kantak.html dan rubicon.jpg di komputeryang sama. jika halaman utama dari situs web yang akan kita buat adalah index. html, maka file kontak. html harus di tempatkan di dalam direktori yang sama dengan file index.html ; sedangkan rubicon. Jpg harus berada di dalam direktori images\jeep. Jika digambarkan, struktur di rektori dan file-nya akan tampak sebagai berikut :





***RANGKUMAN* Menjadikan Gambar sebagai Link**

Sejauh ini kita baru mempelajari link yang berupah teks. Kali ini, kita akan mempelajari bagaimana suatu gambar dijadikan sebagai link ke bagian/dokumen lain. Konsepnya sebenarnya sama saja. Perbedaannya, anda hanya perlu mengganti teks yang dijadikan sebagai link dengan tag <img>

Link yang berupa teks:

```
<a href=”dokumenlain”>teks link</a>
```

Link yang berupa gambar:

```
<a href=”dokumenlain”><img src=”NamaFileGambar” /></a>
```

Contoh :

```
<a href=”detail-rubicon.html”>

<img src=”images/jeep/rubiconl.jpg”   />

</a>
```

Pada contoh di atas kita menjadikan gambar rubicon1.jpg yang tersimpan di dalam direktori images\jep sebagai link untuk menuju ke dokumen detail-rubicon.html.

d. Tugas

Buat direktori baru dengan nama **jeep** di dalam direktori **images**

Copy file gambar yang akan digunakan ke dalam direktori **image\jeep Catatan:** dalam contoh ini penulis menggunakan file **rubicon.jpg**

Jalankan aplikasi text editor

Buatlah dua buah file berikut:

Index.html

```
<html>
<head>
    <title> link< / title>
</head>
<body>
<h2> Demo Membuat link relatif</h2>
<p>Di jual Jeep wrangler rubicon ( DIJAMIN
PUAS ! ) :</p> <p>
Tahun : 2011 <br />
Harga : Rp 975.000.000, - (NEGO) <br />
<p>
<a href= “images/jeep/rubicon.jpg”> Lihat
foto. mobil</a> <br />
Untuk peminat serius, silahkan
<a href= “kontak.html”>hubungi kami</a>
</body>
</html>
```

Kontak.html

```
<html>
<head>
```




```
    <title>link</title>
</head>
<body>
<h2>Demo membuat link relatif</h2>
<p>Untuk peminat serius, silahkan hubunhi
kami di : </p> <strong>Bisa Jeep
Showroom</strong><br /> Jl. Gada sakti no.
13 bandung<br />
999-00333
</p>
<p><a href= “index.html”> kembali ke halaman utama</a></p>
</boby>
</html>
```

Tempatkan ke dalam direktori kerja

Jalankan file **index,html** melalui web brower

Coba buka file index.html dan bagaimana link yang terjadi

Copy file gambar yang akan digunakan ke dalam direktori **images\jeep**

Catatan : dalam contoh ini penulis mengunakan file **rubicon.jpg**

Jalankan aplikasi text editor

Bualah dua file baru dan isikan kode berikut :

Link –gambar .html

```
<html>
<head>
```

```
        <tittle>Link</title>
    </head>
    <body>
    <h2>Demo Membuat Link Berupa gambar</h2>


    <p>Klik gambar dibawah ini untuk
    melihat informasi detail:</p>
    <a href="detail - rubicon.html">
    <img src="images/jeep/rubiconl.jpg" />
    </a>
    </body>
    </html>
```

Detail-rubicon.html

```
<html>
<head>
     <title>link</title>
</head>
<body>
<h2>Demo Membuat Link Berupa Gambar</h2>

<p>
<img src=”images/jeep/rubicon1.jpg”align=”left”/>
Merk: <strong>jeep</strong><br />
```




```
Seri: <strong>Wrangler</strong><br />

Tipe: <strong>Unlimited Rubicon</strong><br />

Tahun: Pembuatan: 2011<br />

Transmisi: Automatic<br />

Silinder: 3.700<br />

Kilometer: 0 km (baru) <br />

Harga: Rp 975.000.00,- <br />

Keterangan tambahan: Mewah dan Gagah (Ready Stock)

<p/>

<p>

<a href=”link-gambar.html”> Kembali ke halaman utama<a/>

<p/>

</body>

</html>
```

Tempatkan kedua file tersebut ke dalam direktori kerja

Jalankan file link-gambar.html melalui web browser

Pada pratikum di atas kita menggunakan gambah sebagai link untuk menuju atau mengarah ke halaman web lain. Untuk mengetahui cara kerjanya, silahkan anda klik gambar di atas dan anda akan menjumpai tampilan seperti berikut:

Klik link Kembali ke halaman utama untuk mengarah ke halaman web semula

Tes Formatif

Lembar Jawaban Tes Formatif

g. Lembar Kerja Siswa



17. Kegiatan Belajar 17 : Link ke Bagian Dokumen HTML

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 17 ini siswa diharapkan dapat :

Memahami jenis-jenis link dalam HTML ***b.***

Uraian Materi

Link ke Bagian Dokumen Tertentu

Link ke bagian dokumen tertentu dapat di bedakan lagi menjadi dua jenis, yaitu :

Link ke bagian tertentu dalam dokumen yang sama

Link ke bagian tertentu dalam dokumen lain

Link ke Bagian Tertentu dalam Dokumen yang sama

Link dapat berperan untuk mengarahkan ke bagian teks atau gambar tertentu yang masih berada dalam satu dokumen yang sama. Link seperti ini biasanya pada halaman web yang memiliki informasi yang panjang atau pada dokumen yang memiliki daftar isi di bagian atasnya. Tujuannya adalah agar para pengunjung tidak perlu melakukan scrolling pada halaman tersebut untuk menuju/kembali tertentu dalam dokumen bersangkutan.

Untuk membuat link semacam ini, terlebih dahulu kita perlu memberikan nama pada bagian teks atau gambar tertentu sebagai bagian yang akan dituju. Pemberian nama tersebut dilakukan dengan cara menuliskan kode berikut :

Teks atau gambar

Selanjutnya untuk membuat link yang akan mengarah ke teks atau gambar yang telah di namai tersebut adalah dengan menuliskan kode seperti berikut :

Teks atau gambar

Tanda # di atas menunjukan bahwa tujuan link adalah nama bagian, bukan berupah file.

Link ke Bagian Tertentu Dalam Dokumen Lain

Selain bagian yang berada dalam satu dokumen, link dapat juga mengarah ke sesuatu bagian tertentu yang terdapat dalam dokumen lain. Untuk membuat link semacam ini, bagian tertentu dalam dokumen lain tersebut harus di tantai dengan pemberian nama seperti pada sub-bab sebelumnya. Perbedaan hanya terletak pada penulisan tujuan link-nya saja. Pada kasus seperti ini, kita perlu menambahkan nama bagian pada tujuan link, seperti berikut:

Teks atau gambar

c. Rangkuman

Buat direktori baru dengan nama

d. Tugas

Buat direktori baru dengan nama **jeep** di dalam direktori **images**

Copy file gambar yang akan digunakan ke dalam direktori **image\jeep**

Membuat link ke bagian tertentu dalam satu Dokumen

Jalanan aplikasi Text Editor

Isikan kode berikut:

```
<html>

<head>

    <title>link>/title>

</head>

<body>

<h2>Demo membuat ke bagian tertentu dalam satu dokumen</h2>

<a name= “atas”></a>


<p> pilih salah satu judul buku berikut untuk melihat

deskripsi : </p> <a href=”#c”> pemrograman

C</a><br />

<a href=”#cpp”> pemrograman C++</a><br      />

<a href=”#bcb”> pemrogaraman Borland

C++Builder</a> </p>
```



pemrograman C

C merupakan bahasa pemrograman yang sudah tidak digunakan lagi kehandalannya dan banyak digunakan untuk membuat program-progarm dalam berbagai bidang, termasuk pembuatan kompilator (compiler) dan sistem operasi. Sampai saat ini, C masih tetap menjadi bahasa populer dan beribawah dalam kanca pemrograman. Sejauh ini, C juga telah menjadi inspirasi bagi kelahiran bahasa-bahasa pemrograman baru, seperti C++, java, dan juga yang lainnya ; sehingga dari sisi sintak kontrol programnya, ketika bahasa ini bisa dikatakan sama. Bahasa pemrograman C sangatlah fleksibel dan portebel, sehingga dapat di tempatkan dan di jalankan di dalam beragam sistem operasi. Pada umum nya, C banyak digunakan untuk melakukan interfacing antar perangkat keras (hardwere) agar dapat berkomunikasi satu sama lainnya.

 kembali ke atas</ a>

pemrograman C++

C++ merupakan bahasa pemrograman yang sangat populer dan telah banyak digunakan untuk mengembangkan perangkat lunak di berbagai bidang; seperti : telekomunikasi, embedded system, bisnis, dan hiburan. Salah satu kehandalan dari C++ adalah bahasa ini mendukung sepenuhnya konsep pemrograman berorientasi objek, atau yang lebih di kenal dengan OPP (object oriented programming). (cross platform), C++ juga telah manjadi bagian hidup dari sebagian para pecinta linux.

kembali ke atas

pemrograman borland C++builder

</p>

```
<p>

C++builder adalah sebuah perangkat lunak berbasis C++ yang digunakan untuk melakukan pembuatan aplikasi (baik visual maupun non-visual) yang berjalan di atas platform microsoft windows. Selain itu, C++builder juga banyak di gunakan sebagai alat untuk pengembangan file-file DLL (Dynamic Link Library ). Dengan adanya kemudahan dan dukungan visual yang di tawarkan di dalamnya, C++builder banyak di gunakan oleh para programmer profesional untuk pengembangan aplikasi yang beraneka ragam jenisnya, seperti : openGL, multimedia, games, database, embedded system, dan banyak lagi yang lainnya.<br />

<a href=”#atas”>kembali ke atas</a>

</p >

</body>

</html>
```

Simpan file dengan nama link-bagiandokumen1.html dan tempatkan ke dalam direktori kerja

Jalan file tersebut melalui web browser

Pada praktikum di atas kita membuat 4 bagian yang di jadikan sebagai tujuan link, yaitu :

Atas. Bagian ini adalah bagian paling atas dari dokumen dan berguna untuk kembali ke bagian menu (Jika penunjung sedang mengaktifkan bagian bawah dokumen)

C. Bagian ini adalah bagian yang menerangkan tentang buku pemrograman C

Cpp. Bagian ini adalah bagian yang menerangkan tentang buku pemrograman C++.

Bcb. Bagian ini adalah bagian yang menerangkan tentang buku pemrograman borland C++Builder.



Untuk menguji cara kerja dari link di atas, coba anda klik link kedua (pemrograman C++), maka web browser akan menampilkan bagian yang di pilih, seperti ditujukan oleh gambar berikut:

Klik link kembali ke atas jika anda ingin kembali ke tampilan semula (tampilan awal).

Membuat Link ke Bagian Tertentu dalam Dokumen Lain

Jalankan aplikasi Text Editor

Buatlah dua file dan isikan kode berikut:

Dokumen-1.html

```
<html>
<head>
    <title>Link</title>
</head>
<body>
<h2>Demo Membuat ke Bagian Tertentu dalam
Dokumen lain</h2> <p>pilih salah satu judul buku
berikut untuk melihat diskripsi :</p> <a
href=”dokumen-2.html#c”pemrograman
C</a><br />
<a href=”dokumen-2.html#cpp”>pemrograman
C++</a><br /> <a href=”dokumen-
2.html#bcb”>
Pemrograman borlond C++Builder</a>
</p>

</body>
</html>
```

```
<html>

<head>

    <title>link</title>

</head>

<body>

<h2>Demo Membuat ke Bagian Tertentu dalam Dokumen
Lain</h2>


<p><a name=”c”><strong>pemrograman C</strong
></a></p>
```

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa



18. Kegiatan Belajar 18 : Format Target Link, Email dan Telepon

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 18 ini siswa diharapkan dapat :

Memahami Format target link, email dan telepon

b. Uraian Materi

Menentukan Target Link

Target atau tujuan yang dihubungkan melalui suatu link dapat ditentukan sendiri oleh kita. Target yang di maksud di sini adalah dimana dokumen akan ditempatkan atau ditampilkan. Dalam keadaan default (ketika atribut TARGET tidak di sertakan), dokumen baru yang ditampilkan pada saat link dipilih akan menutup atau menimpa dokumen lama (dokumen yang mengandung link bersangkutan). Untuk melakukan hal, kita hanya perlu menambahkan atribut TARGET di dalam tag <a>, yang memiliki bentuk umum penulisan seperti berikut:

<a href=”NamaDokumen” target=”NamaTarget”>Teks/Gambar<a/>

NamaDokumen adalah namadokumen yang akan di tampilkan jika link bersangkutan dipilih/diklik dan NamaTarget adalah target atau tempat tujuan dari dokumen yang akan ditampilkan. NamaTarget bisa bernilai _BLANK maupun nama frame (jika layar web kita bagi dengan menggunakan frame)

Link Email

Target atau tujuan yang dihubungkan melalui suatu link dapat ditentukan sendiri oleh kita. Target yang di maksud di sini adalah dimana dokumen akan ditempatkan atau ditampilkan. Dalam keadaan default (ketika atribut TARGET tidak di sertakan), dokumen baru yang ditampilkan pada saat link dipilih akan menutup atau menimpa dokumen lama (dokumen yang mengandung link bersangkutan). Untuk melakukan hal, kita hanya perlu menambahkan atribut TARGET di dalam tag <a>, yang memiliki bentuk umum penulisan seperti berikut:

<a href=”NamaDokumen” target=”NamaTarget”>Teks/Gambar<a/>

Link Telepon

Target atau tujuan yang dihubungkan melalui suatu link dapat ditentukan sendiri oleh kita. Target yang di maksud di sini adalah dimana dokumen akan ditempatkan atau ditampilkan. Dalam keadaan default (ketika atribut TARGET tidak di sertakan), dokumen baru yang ditampilkan pada saat link dipilih akan menutup atau menimpa dokumen lama (dokumen yang mengandung link bersangkutan). Untuk melakukan hal, kita hanya perlu menambahkan atribut TARGET di dalam tag <a>, yang memiliki bentuk umum penulisan seperti berikut:

<a href=”NamaDokumen” target=”NamaTarget”>Teks/Gambar<a/>

Rangkuman

Tugas

Isikan kode berikut:

```
<html>
<head>
        <title>link</title>
</head>
<body>
<h2>Demo Menampilkan Dokumen pada Window Baru</h2>
<p>Klik salah satu link berikut:</p>
<a href=http://www.jeep.com/target=”_blank”>
Jeep.com</a><br />
<a href=http://www.autobildindomesia.com/”
Target=”_blank”>AutoBild Indonesia</a>
</body>
</html>
```

Simpan file dengan nama link-target.html dan tempatkan ke dalam direktori kerja



Jalankan file tersebut melalui web browser

Tes Formatif

Lembar Jawaban Tes Formatif

Lembar Kerja Siswa

19. Kegiatan Belajar 19 : Anatomi dan cara kerja form

a. Tujuan Pembelajaran.

Setelah mengikuti kegiatan belajar 19 diharapkan siswa dapat:

Memahami elemen-elemen HTML untuk membentuk form

Menerapkan elemen-elemen HTML untuk membentuk form

Mengetahui penggunaan elemen-elemen HTML di dalam form

Uraian materi.

Definisi dan fungsi form

Form adalah salah satu bentuk halaman web yang digunakan untuk menerima masukan dari pengguna, untuk selanjutnya masukan dari pengguna tersebut diolah menggunakan bahasa pemrograman web, baik secara server side scripting(misalkan PHP, JSP) ataupun client-side scripting (javascript). Form dapat digunakan untuk berbagai keperluan seperti keperluan login, transaksi penjualan, mengumpulkan informasi atau meminta umpan balik dari pengguna, menawarkan barang/jasa secara on-line dan sebagainya. Form sangat sering kita jumpai dalam satu aplikasi sistem inormasi berbasis web.

Anatomi dari form

Standard atau format umum penulisan dari form adalah sebagai berikut :

```
<Form metode="post/get" action="….">
……
……
……
</form>
```

Berikut ini adalah beberapa variasi yang dapat diterapkan pada atribut**<form> :**

```
<FORM ACTION=action base> form tags </FORM>
<FORM METHOD=method> form tags </FORM>
<FORM ENCTYPE=media type> form tags </FORM>
<FORM ACTION=action base TARGET="target window name"> form
tags
```



</FORM> <FORM SCRIPT=URL> form tags </FORM>

Catatan : target window name diisikan berupa :

_blank

_self

_parent

_top

Atribut – atribut dari elemen HTML `<form>` adalah sebagai berikut :

Atribut	Fungsi
Method	Metode pengiriman data ke tujuan, yaitu : Get : Mengirim data pada server dengan cara melatakkan data pada bagian akhir URL yang ditunjuk Post : Mengirim datanya secara terpisah
3) Action	Aksi yang digunakan jika user menekan tombol submit

Di antara elemen `<form>` dan `</form>` digunakan elemen – elemen HTML `<input>` yang berfungsi untuk mendefinisikan input yang akan dimasukkan oleh pengguna. Format umum dari penulisan elemen HTML `<input>` adalah sebagai berikut :

```
<input
      name="name"
      size="number"
      type="text"|"checkbox"|"radio"|"submit"|"reset"
      value="value"
      checked >
```

Jenis – jenis dari elemen `<input>` ditentukan dari atribut TYPE pada elemen HTML `<input>`. Berikut ini adalah datfar pilihan dan penjelasan dari atribut TYPE :

TYPE	Keterangan
text	Masukan berupa teks
password	Masukan berupa password, terlihat karakter tertentu
checkbox	Masukan berupa pilihan, pilihan bisa lebih dari satu
radio	Masukan berupa pilihan, pilihan hanya satu
submit	Tombol untuk menerima seluruh masukan
reset	Tombol untuk membatalkan seluruh masukan
file	Untuk menerima masukan berupa file
image	Mirip submit, untuk masukan berupa gambar yang dapat di klik
button	Membuat tombol

Setiap jenis masukan (input) dari halaman web minimal harus mempunyai atribut jenis (type), nama (name) dan nilai (value) default. Disamping atribut type, masing-masing elemen input memiliki atribut-atribut yang lain dan beberapa atribut tersebut berlaku untuk beberapa jenis elemen input. Sebagai contoh adalah elemen `<input>` yang atribut TYPE-nya adalah text memiliki atribut antara lain :

NAME, digunakan untuk menamai kotak

VALUE, digunakan untuk menandai atau menampung teks

SIZE, digunakan untuk mengatur ukuran teks pada kotak

MAXLENGTH, digunakan untuk menentukan panjang maksimum teks Masing-masing jenis input memiliki karakteristik yang berbeda, dan pemilihan penggunaannya disesuaikan dengan kesesuaian nilai dari input data yang dikelola.

Elemen- elemen form input



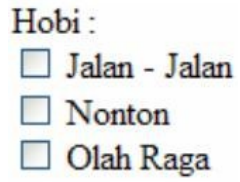
Uraian berikut ini adalah penjelasan tentang elemen-elemen input yang digunakan dalam form beserta dengan format tag-nya :

Form Input Check Box

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=CHECKBOX NAME=name VALUE=value>
<INPUT TYPE=CHECKBOX NAME=name VALUE=value CHECKED>
```

Contoh bentuk tampilannya adalah sebagai berikut :



Form Input File

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=FILE NAME=name ACCEPT=mime type list>
```

Contoh bentuk tampilannya adalah sebagai berikut :



Form Input Hidden

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=HIDDEN NAME=name VALUE=value>
```

Form Input Image

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=IMAGE NAME=name SRC="URL">
<INPUT TYPE=IMAGE NAME=name SRC="URL" ALIGN="alignment">
```

Form Input Password

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=PASSWORD NAME=name>
<INPUT TYPE=PASSWORD NAME=name MAXLENGTH=length>
<INPUT TYPE=PASSWORD NAME=name SIZE=size> <INPUT
TYPE=PASSWORD NAME=name VALUE=value>
```

Contoh bentuk tampilannya adalah sebagai berikut :

Password ••••••••

Form Input Radio Button

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=RADIO NAME=name VALUE=value>
<INPUT TYPE=RADIO NAME=name VALUE=value CHECKED>
```

Contoh bentuk tampilannya adalah sebagai berikut :

Jasa Pengiriman :
- ○ Tiki Over Night Service (ONS)
- ○ Tiki Regular Service
- ○ POS

Form Input Reset

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=RESET>
```

Contoh bentuk tampilannya adalah sebagai berikut :

Reset

Form Input Submit

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=SUBMIT>
<INPUT TYPE=SUBMIT NAME=name>
<INPUT TYPE=SUBMIT VALUE=value>
```

Contoh bentuk tampilannya adalah sebagai berikut :

Submit Query

Form Input Button

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=BUTTON>
<INPUT TYPE=BUTTON NAME=name>
<INPUT TYPE=BUTTON VALUE=value>
```

Contoh bentuk tampilannya adalah sebagai berikut :

Button

10. Form Input Text

Format tag HTML-nya adalah sebagai berikut :

```
<INPUT TYPE=TEXT NAME=name>
<INPUT TYPE=TEXT NAME=name MAXLENGTH=length>
<INPUT TYPE=TEXT NAME=name SIZE=size>
<INPUT TYPE=TEXT NAME=name VALUE=value>
```

Contoh bentuk tampilannya adalah sebagai berikut :



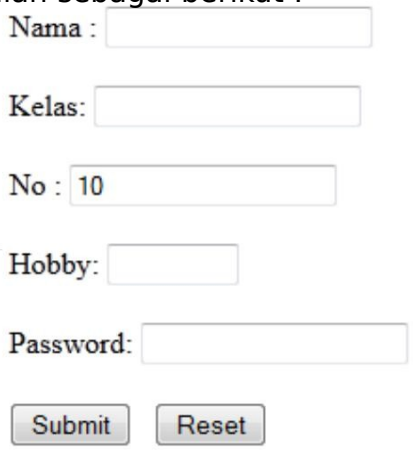
Contoh form sederhana

Berikut ini adalah contoh dari sebuah halaman web yang berisi form yang sederhana :

```
<html>
<head>
   <title> Form 1 </title>
</head>
<body>
<form name=”form1” method=”POST” action=”inputdata.php”>
     Nama : <input type="text" name="nama"> <br><br>
     Kelas: <input type="text" name="kelas" maxlength="2">
<br><br>
     No : <input type="text" name="nomor" value="10">
<br><br>
     Hobby: <input type="text" name="hobby" size="8">
<br><br>
     Password: <input type="password" name="password">
<br><br>
<input type="submit" name="submit"
value="Submit">   <input type="reset"
name="reset" value="Reset"> </form>

</body>
</html>
```

Apabila dibuka di dalam browser, maka tampilan dari susunan kode HTML di atas adalah sebagai berikut :



Apabila tombol “Submit” diklik, maka semua nilai yang sudah diinputkan oleh pengguna akan di olah di dalam file PHP “inputdata.php”.

c. Rangkuman

Form adalah salah satu bentuk halaman web yang digunakan untuk menerima masukan dari pengguna, untuk selanjutnya masukan dari pengguna tersebut diolah menggunakan bahasa pemrograman web, baik secara server side scripting(misalkan PHP, JSP) ataupun client-side scripting (javascript).

Form dapat digunakan untuk berbagai keperluan seperti keperluan login, transaksi penjualan, mengumpulkan informasi atau meminta umpan balik dari pengguna, menawarkan barang/jasa secara on-line dan sebagainya.

Form sering dijumpai dalam satu aplikasi sistem inormasi berbasis web. Setiap jenis masukan (input) dari halaman web minimal harus mempunyai atribut jenis (type), nama (name) dan nilai (value) default.

Disamping atribut type, masing-masing elemen input memiliki atribut-atribut yang lain dan beberapa atribut tersebut berlaku untuk beberapa jenis elemen input. Sebagai contoh adalah elemen `<input>` yang atribut TYPE-nya adalah text memiliki atribut antara lain :

NAME, digunakan untuk menamai kotak

VALUE, digunakan untuk menandai atau menampung teks

SIZE, digunakan untuk mengatur ukuran teks pada kotak

MAXLENGTH, digunakan untuk menentukan panjang maksimum teks

Masing-masing jenis input memiliki karakteristik yang berbeda, dan pemilihan penggunaannya disesuaikan dengan kesesuaian nilai dari input data yang dikelola.



Tugas

Test Formatif.

Dalam test ini setiap peserta didik wajib membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

Sebutkan 5 jenis elemen input pada formulir HTML.

Sebutkan atribut-atribut yang berperan penting dan harus diperhatikan pada elemen elemen form

Sebutkan atribut-atribut yang berperan penting dan harus diperhatikan pada elemen elemen input

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : 5 jenis elemen input HTML

.....

.....

.....

.....

.....

LJ- 02 : Atribut elemen form

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LJ- 02 : Atribut elemen form

.....

.....

.....

.....

.....

g. Lembar Kerja Peserta Didik.



20. Kegiatan Belajar 20: Pemformatan form

a. Tujuan Pembelajaran.

Setelah mengikuti kegiatan belajar 20 diharapkan peserta didik dapat:

Memahami format dari sebuah form

Menerapkan elemen HTML tabel untuk memformat form

b. Uraian materi.

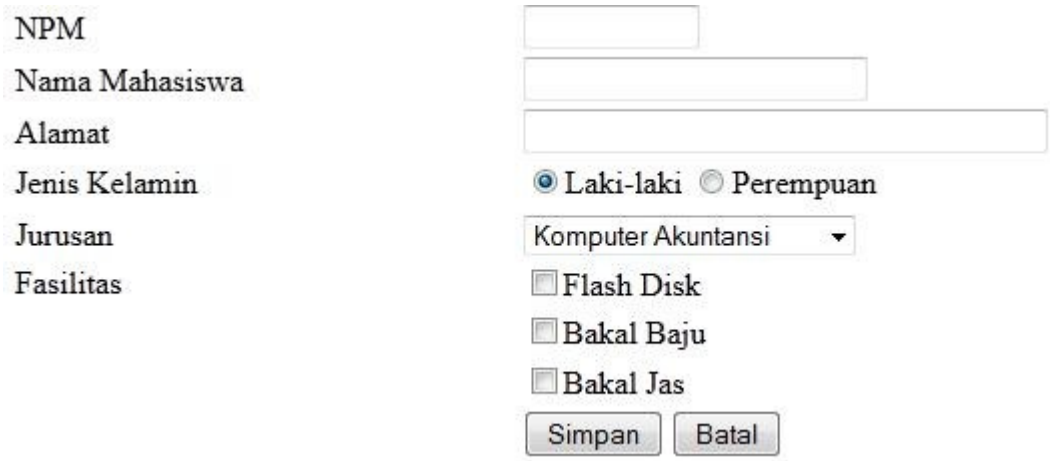
Format form secara horizontal

Untuk membuat tampilan dari sebuah form menjadi lebih tertata dengan rapi, umumnya digunakan elemen HTML <table> untuk memformat form tersebut. Secara umum layout dari form adalah layout horizontal, yakni layout dimana setiap input ditempatkan di samping dari label keterangan masing- masing input tersebut. Berikut ini adalah contoh dari penggunaan elemen <table> untuk memformat tampilan form secara horizontal :

```
<html>
<head><title>CONTOH TAG FORM</title></head>
<body>
<form method="Post">
<table border=0 width="75%">
 <tr>
  <td width="25%">NPM</td>
  <td><INPUT TYPE="TEXT" Name="NPM" MAXLENGTH="10"
SIZE="11"></td>
 </tr>
 <tr>
  <td width="25%">Nama Mahasiswa</td>
  <td><INPUT TYPE="TEXT" Name="NamaMHS" MAXLENGTH="25"
SIZE="25"></td>
 </tr>
 <tr>
  <td width="25%">Alamat</td>
  <td><INPUT TYPE="TEXT" Name="Alamat" MAXLENGTH="50"
SIZE="40"></td>
 </tr>
 <tr>
  <td width="25%">Jenis Kelamin</td> <td><INPUT
  TYPE="RADIO" Name="Jenkel" CHECKED
VALUE="Laki2">Laki-laki
```

```
      <INPUT TYPE="RADIO" Name="Jenkel"
VALUE="Perempuan">Perempuan
  </td>
 </tr>
 <tr>
 <tr>
  <td width="25%">Jurusan</td>
  <td><SELECT Name="Jurusan">
              <OPTION VALUE="KA">Komputer Akuntansi
              <OPTION VALUE="MI">Manajemen Informatika
              <OPTION VALUE="TK">Teknik Komputer
         </SELECT>
  </td>
 </tr>
 <tr>
  <td width="25%">Fasilitas</td>
  <td><INPUT TYPE="CHECKBOX">Flash Disk</td>
 </tr>
 <tr>
  <td></td><td><INPUT TYPE="CHECKBOX">Bakal Baju</td>
 </tr>
 <tr>
  <td></td><td><INPUT TYPE="CHECKBOX">Bakal Jas</td>
 </tr>
  <td></td>
  <td><INPUT TYPE="SUBMIT"
  VALUE="Simpan"> <INPUT TYPE="RESET"
  VALUE="Batal"> </td>
 </tr>
</table>
</form>
```

Dan berikut ini adalah tampilan dari form yang telah di tulis di atas :



Format form secara vertikal

Sejalan dengan perkembangan teknologi mobile, halaman web yang berbasis mobile tentunya sebuah halaman web di dalam perangkat mobile akan memerlukan form di dalam menerima input dari pengguna. Untuk menyesuaikan dengan lebar layar dari perangkat mobile, misalkan handphone ataupun ipad, maka layout dari sebuah halaman web harus disesuaikan, termasuk juga untuk layout dari sebuah form. Umumnya apabila sebuah form akan digunakan pada aplikasi mobile, layout yang diterapkan adalah layout secara vertikal, sesuai dengan karakteristik layar yang tidak terlalu besar. Berikut ini adalah contoh dari sebuah form dengan layout vertikal.

```
<html>

<head>

<title>Contoh Layout Form Vertikal<title>

</head>

<body>

<FORM METHOD="POST"

   ACTION="mailto:teknisi@jardiknas.org"> <H4>FORM</H4>

   <INPUT TYPE="text" NAME="var1" SIZE="30" VALUE="Enter
your name here.">

   <BR><BR>

   <B>Are you a student?</B>

   <INPUT TYPE="checkbox"

   NAME="var2"> <BR><BR>

   <B>How old are you?</B>

   <BR>

   <INPUT TYPE="radio" NAME="var3" VALUE="r1">10 -

   15 <BR>

   <INPUT TYPE="radio" NAME="var3" VALUE="r2">16 - 20
```

```
    <BR>
    <INPUT TYPE="radio" NAME="var3" VALUE="r3">21 - 25
    <BR><BR>
    <INPUT TYPE="submit" NAME="var4" VALUE="Send">
    <INPUT TYPE="reset" NAME="var5" VALUE="Clear">
</FORM>
</body>
</html>
```

Dan hasil dari halaman form yang telah ditulis di atas adalah sebagai berikut :



c. Rangkuman

Untuk membuat tampilan dari sebuah form menjadi lebih tertata dengan rapi, umumnya digunakan elemen HTML <table> untuk memformat form tersebut. Secara umum layout dari form adalah layout horizontal, yakni layout dimana setiap input ditempatkan di samping dari label keterangan masing- masing input tersebut. Sejalan dengan perkembangan teknologi mobile, halaman web yang berbasis mobile tentunya sebuah halaman web di dalam perangkat mobile akan memerlukan form di dalam menerima input dari pengguna. Untuk menyesuaikan dengan lebar layar dari perangkat mobile, misalkan handphone ataupun ipad, maka layout dari sebuah halaman web harus disesuaikan, termasuk juga untuk layout dari sebuah form. Umumnya apabila sebuah form akan digunakan pada aplikasi mobile, layout yang diterapkan adalah layout secara vertikal, sesuai dengan karakteristik layar yang tidak terlalu besar.

Tugas :

Test Formatif.

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.



Jelaskan fungsi dari elemen-elemen table untuk memformat tampilan form

Jelaskan perbedaan dari format horizontal dan format secara vertikal dari sebuah form

Lembar Jawaban Test Formatif (LJ).

LJ- 01 : fungsi dari elemen tabel untuk form

.....

.....

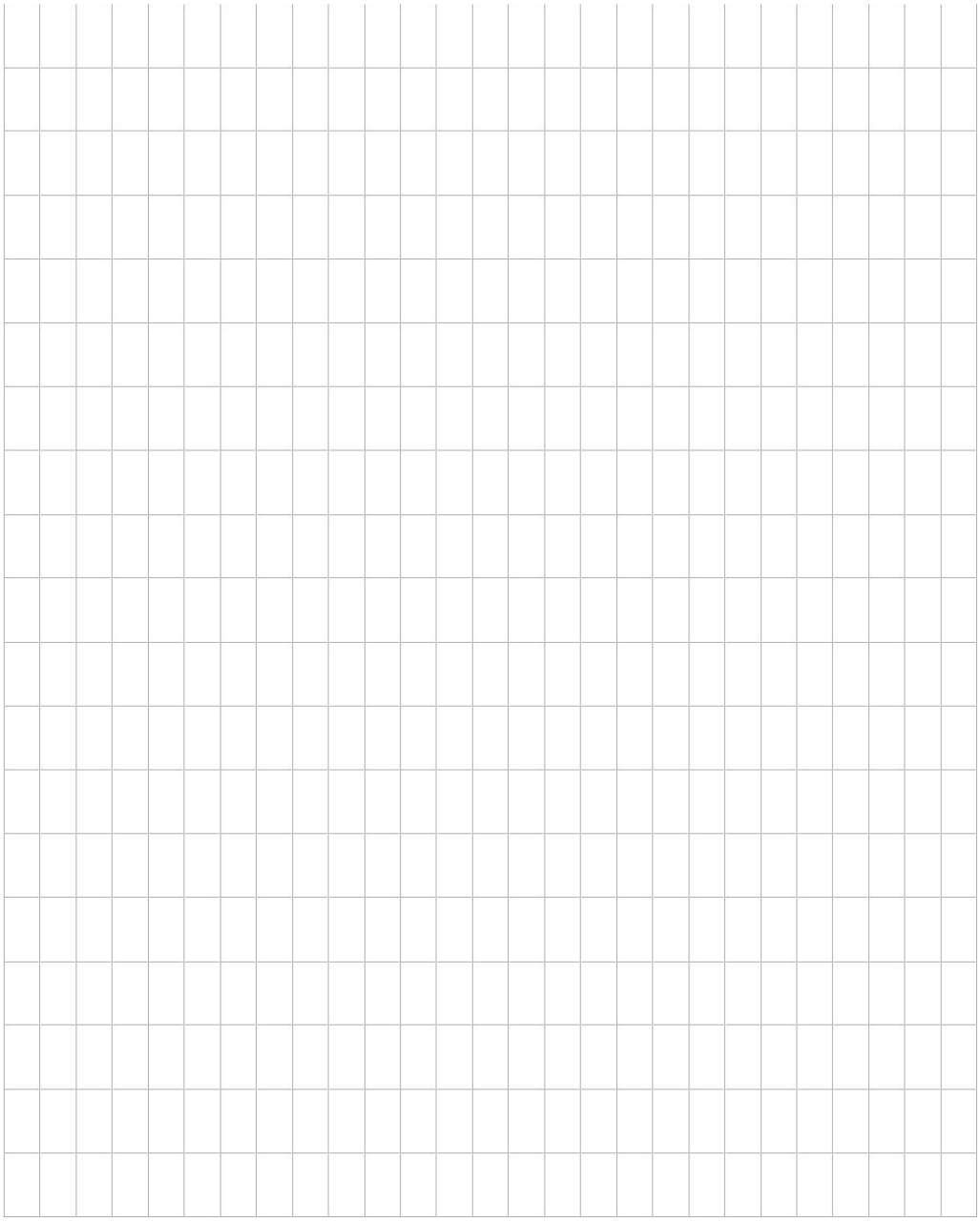
.....

.....
.....
.....
.....
.....
.....
.....

LJ- 01 : perbedaan formt horisontal dan vertikal form

.....
.....
.....
.....
.....
.....
.....
.....
.....

g. Lembar Kerja Peserta Didik.





DAFTAR PUSTAKA

Betha sidik, Ir., Husni I. Pohan, Ir., M.Eng. (Juni 2012) , *Pemrograman Web Dengan HTML Revisi Keempat,* Penerbit Informatika Bandung

Budi Raharjo (September 2011), *Belajar Pemrograman Web*, Penerbit Modula, Bandung, Penerbit O'Reilly Media, Inc.: Kanada

Jennifer Niederst Robbins , *Learning Web Design 4th Edition*

Purnama Pupung Budi (2004), Kiat Praktis menjadi Desainer Web Profesional, PT Elex Media Komputindo, Jakarta

Simarmata Janner (2010), Rekayasa Web, Penerbit Andi, Yogyakarta

Wahana Komputer (2013), Teknik Singkat dan Cepat Menguasai CSS3, Penerbit Andi, Yogyakarta