MODUL PRAKTIKUM

INTRAKSI MANUSIA dan KOMPUTER

UNIVERSITAS TEKNOLOGI MATARAM

2019

Jl. Kampus Universitas Teknologi Mataram, Kekalik, Mataram

MODUL PRAKTIKUM DESAIN WEB

BAB 1

SEKILAS TENTANG HTML

1. PENGENALAN HTML

Sebelum kita bahas lebih detail tentang CSS dan berbagai hal yang berkaitan dengan style tersebut, seperti kata pepatah “tak kenal maka tak saying”, maka ada baiknya jika kita sedikit belajar lagi tentang bahasa dasar dalam pembuatan dokumen web yang dapat dibuat menggunakan CSS yaitu HTML. HTML atau Hypertext Markup Language adalah sebuah protokol yang digunakan untuk membuat format suatu dokumen web yang mampu dibaca dalam browser dari berbagai platform komputer manapun. Sifat dari bahasa HTML ini adalah client script dimana dokumen tersebut dapat dibuka di dalam komputer stand alone yang tidak membutuhkan server untuk dapat menampilkannya di dalam browser. Dokumen HTML merupakan file yang pada umumnya ber-esktensi .htm atau .html dimana bahasa HTML tersebut tersusun atas *tag* atau *syntax* yang berformat <isi tag>.

Adapun aturan penulisan HTML tersebut adalah :

a. Dalam penulisannya, tag HTML diapit dengan dua karakter kurung siku (angle bracket) “<” dan “>”
b. Tag HTML secara normal selalu berpasangan, contoh : `<div>` dan `</div>`
c. Tag HTML pertama dalam suatu pasangan adalah tag awal, dan tag kedua merupakan tag akhir, contoh : `<index>` akan diakhiri dengan `</index>`.
d. Tag HTML tidak case sensitive dimana tag dengan huruf kecil sama dengan tag dengan huruf besar, contoh : <b> sama dengan <B>
e. Jika dalam suatu tag terdapat tag lagi, maka penulisan tag akhir tidak boleh bersilang dan harus berurutan. Contoh : `<b><i> huruf tebal dan miring </i></b>`
f. Penulisan script HTML selalu diawali dengan `<html>` dan diakhiri dengan `</html>`

Secara teknis web browser akan melakukan penerjemahan dokumen untuk ditampilkan dalam bentuk html jika dokumen tersebut terdapat diantara tag `<html>` dan `</html>`. Ada baiknya dalam penulisan HTML menggunakan huruf kecil untuk menghadapi HTML generasi berikutnya (rekomendasi dari W3C untuk HTML 4.0 dan XHTML)

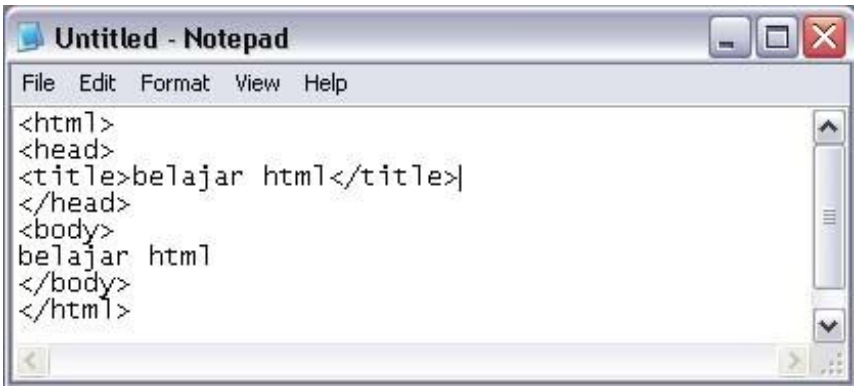
Dalam HTML struktur dibagi menjadi dua bagian yaitu :

a. Bagian Kepala (head)
   Bagian ini memuat informasi mengenai kepala dokumen
b. Bagian Badan (body)
   Bagian ini memuat informasi mengenai badan dokumen

Adapun struktur utama penulisan dalam HTML adalah :

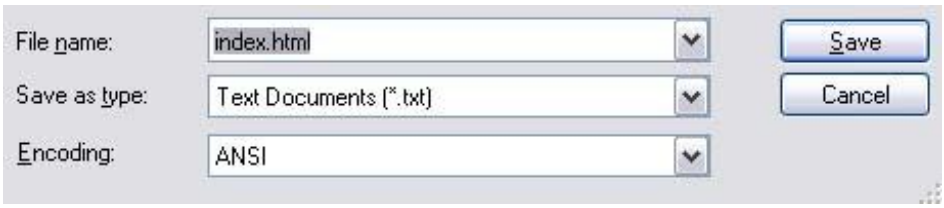
```
<html>
<head>
<!—- Bagian Head --!>
</head>
<body>
<!—- Bagian Body --!>
</body>
</html>
```

Sedangkan pembuatan dokumen web menggunakan HTML pun sangat mudah karena anda dapat membuatnya menggunakan program editor standar Windows seperti Notepad dan menuliskan bahasa dan aturan HTML dan kemudian menyimpannya dalam ekstensi .htm atau .html.
Untuk menggunakan program notepad dapat dipilih lewat menu Start – Program Files – Accessories – Notepad. Kemudian ketikkan kode html anda seperti contoh berikut :



Gambar 1.1. Script html dalam program notepad

Kemudian simpan dengan memilih menu File – Save As, dan simpan dengan ekstensi .htm atau .html



Gambar 1.2. Menyimpan dokumen dengan ekstensi .html

Setelah itu tampilan hasil dokumen html dari script yang telah dibuat tersebut dapat anda lihat menggunakan web browser seperti Internet Explorer, Netscape Navigator, Mozilla Firefox, Opera dan sejenisnya.



Gambar 1.3. Dokumen html dilihat dalam web browser Internet Explorer

2. DOKUMEN HEAD

Dalam bagian head ini kita dapat meletakkan beberapa tag diantara <head> dan </head> yaitu :

a. Judul

Merupakan tag yang mendeklarasikan judul dari suatu dokumen HTML. Struktur tag untuk judul ini adalah :

```
<title>...judul...</title>
```

b. URL

Merupakan tag yang menyatakan URL asal suatu dokumen HTML. Struktur tag untuk url asal adalah :

```
<base href=”...”>
```

c. Link

Merupakan tag untuk menghubungankan antara suatu dokumen dengan dokumen lainnya dalam website yang dibuat. Struktur tag untuk menghubungkan dokumen adalah :

```
<link href=”...”>
```

d. Meta

Merupakan tag yang menyatakan suatu meta informasi dalam suatu dokumen HTML. Struktur tag untuk meta informasi adalah :

```
<meta name=”...” content=”...”>
```

e. Index

Merupakan tag yang menyatakan bahwa dokumen tersebut diperkenankan di indeks oleh server web. Struktur tag untuk isi index adalah :

```
<isindex>
```

f. Style

Merupakan tag yang mendefinisikan style dari dokumen HTML. Struktur tag untuk mendefinisikan style adalah :

```
<style ahref=”...”>
```

Contoh penggunaan dalam HTML :

```
<html>
<head>
<title>Belajar HTML</title>
<link href="global.css" rel="stylesheet"
type="text/css">
</head>
<body>
... dokumen html ...
</body>
</html>
```

3. ATRIBUT BODY

Bagian body dalam HTML adalah bagian dimana dokumen web akan terlihat dalam browser. Dalam bagian ini dapat berisi atribut-atribut, aturan dan isi dari dokumen web yang akan ditampilkan. Penulisan atribut body ini biasanya dituliskan sesudah tag “<body” dan sebelum “>” maupun sesudah tag <body> dan sebelum tag </body>. Dalam bagian body tersebut terdapat beberapa atribut yang bisa digunakan. Beberapa atribut tersebut adalah :

a. Warna Latar

Atribut tag untuk mendefinisikan warna latar untuk isi di dalam dokumen web dengan menuliskan “bgcolor=nilai warna”. Contoh :

```
<body bgcolor=”lime”>
```

b. Warna Teks

Atribut tag untuk mendefinisikan warna teks untuk isi di dalam dokumen web dengan menuliskan “text=nilai warna”. Contoh :

```
<body text=”green”>
```

c. Warna Link

Jika anda memiliki sebuah tombol untuk menghubungkan (link / hyperlink) dalam dokumen web yang anda buat dengan dokumen web lainnya, anda dapat mendefinisikan warna dari tombol link tersebut ketika kita klik dengan menggunakan tag `<link=”nilai warna1” alink=”nilai warna2”>` contoh :

```
<body link=”green” alink=”grey”>
```

Contoh penerapan dalam dokumen html :

```
<html>
<head>
<title>Dokumen Body</title>
</head>
<body bgcolor="lime" text="green" link="white"
alink="grey">
Teks  berwarna lime<br>
<a href=#>tombol1</a>
<a href=#>tombol2</a>
</body>
</html>
```

Dalam browser script diatas akan terlihat dua tombol yaitu tombol1 dan tombol2 dan ketika kita klik tombol tersebut maka akan berubah menjadi warna hijau seperti terlihat pada gambar berikut :



Gambar 1.4. Tampilan script dalam browser

4. FORMAT TEKS

Dalam dokumen HTML, teks mempunyai bentuk dalam pemformatan dalam dokumen tersebut. Berikut ini adalah beberapa tag beserta fungsinya untuk melakukan pengaturan dan format tampilan teks dalam dokumen HTML.

a. Font

Dalam setting format teks, tag font disini memiliki beberapa macam nilai, diantaranya adalah :

- "face" atau jenis font
- "size" atau ukuran font
- "color" atau warna font
- "style" atau style font

Contoh penulisan syntaknya adalah :

```
<font face=″verdana″ size=″4 px″ color=″green″
style="background-color:″#0066CC"> teks dokumen
</font>
```

b. Paragraf

Digunakan untuk membuat teks dokumen yang berbentuk paragraf. Syntaknya adalah :

```
<p>teks dalam paragraf</p>
```

c. Perataan

Digunakan untuk mengatur perataan teks dalam sebuah paragraph dalam dokumen web yang dibuat, syntaknya adalah :

```
<align=...>teks
```

Nilai dari perataan adalah :

- Rata kanan = right
- Rata kiri = left
- Rata tengah = center
- Rata kanan-kiri = justify

d. Preformatted Text

Digunakan untuk mengembalikan format teks ke posisi awal atau posisi default pertama kali sebelum diformat. Syntaks penulisan preformatted teks adalah :

```
<pre>...</pre>
```

e. Line break

Line break adalah pemenggalan atau pemutusan kalimat dengan menggunakan garis dengan perintah <hr>. Adapun property <hr> adalah "align" untuk perataan garis, "size" untuk ukuran dari garis, dan "color" untuk warna garis seperti pada contoh berikut :

```
<hr align=”..." size="..." color="...” width=”…">
```

Dalam pemenggalan kalimat dalam dokumen web selain menggunakan garis horizontal, anda dapat juga menggunakan tag <br /> untuk pemenggalan baris dalam sebuah kalimat. Biasanya tag ini digunakan pada kalimat dalam paragraf.

```
<p>kalimat satu<br />kalimat dua<br />kalimat 3</p>
```

f. Komentar

Anda dapat memberikan komentar dalam dokumen web yang tidak akan ditampilkan dalam browser menggunakan perintah :

```
<!-- Komentar -->
```

g. Format Heading

Heading dalam HTML sama seperti dalam Ms. Word yaitu format ukuran huruf dalam dokumen HTML. Syntak penulisannya adalah

```
<H1> Teks Heading </H1>
```

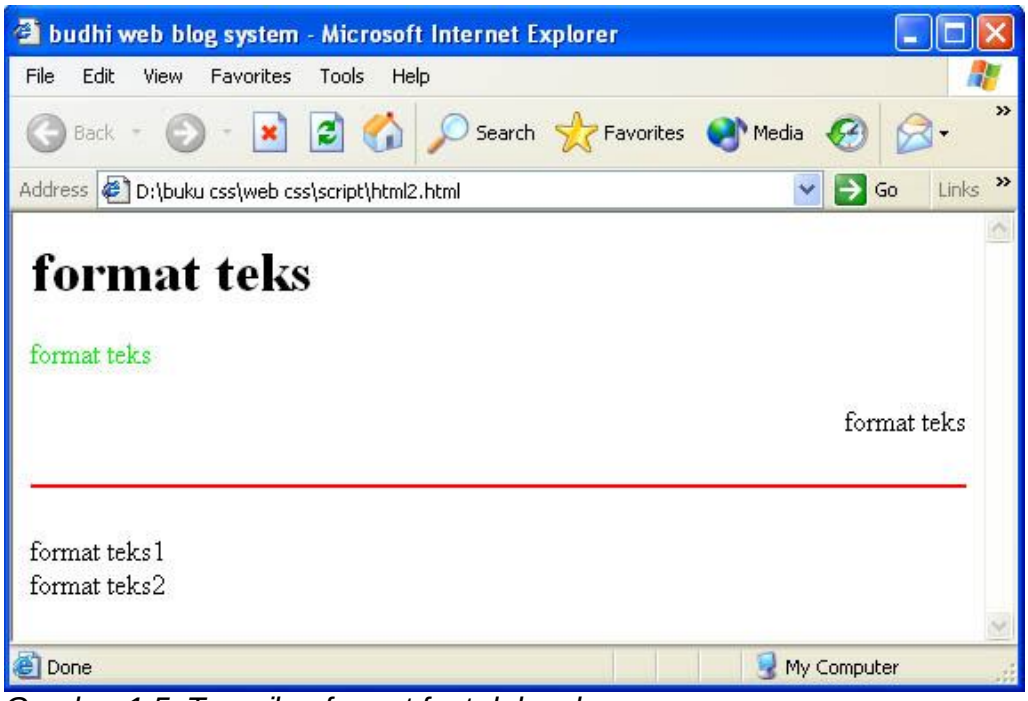
Teks yang dihasilkan ketika menggunakan script ini adalah ukuran teks menurut level heading yaitu dari level heading 1 <H1> sampai dengan level heading 6 atau <H6>.

Contoh penerapan dalam HTML :

```
<html>
<head>
<title>budhi web blog system</title>
</head>
<body>
<H1> format teks </H1>
<font face=”verdana” size=”4 px” color=”green”
style="background-color:”#0066CC"> format
teks</font>
<p align="right">format teks</p>
<hr align="center" size="2" color="red">
```

```
    <p align="left"> format teks1<br>format
    teks2<br></p>
    <!-- format teks --!>
    </body>
    </html>
```

Tampilan dalam browser akan terlihat :



Gambar 1.5. Tampilan format font dalam browser

6. DAFTAR

Penggunaan daftar dalam HTML biasanya digunakan untuk menampilkan informasi-informasi dalam bentuk option.

a. Membuat daftar berurut (Ordered List)
   Syntak :

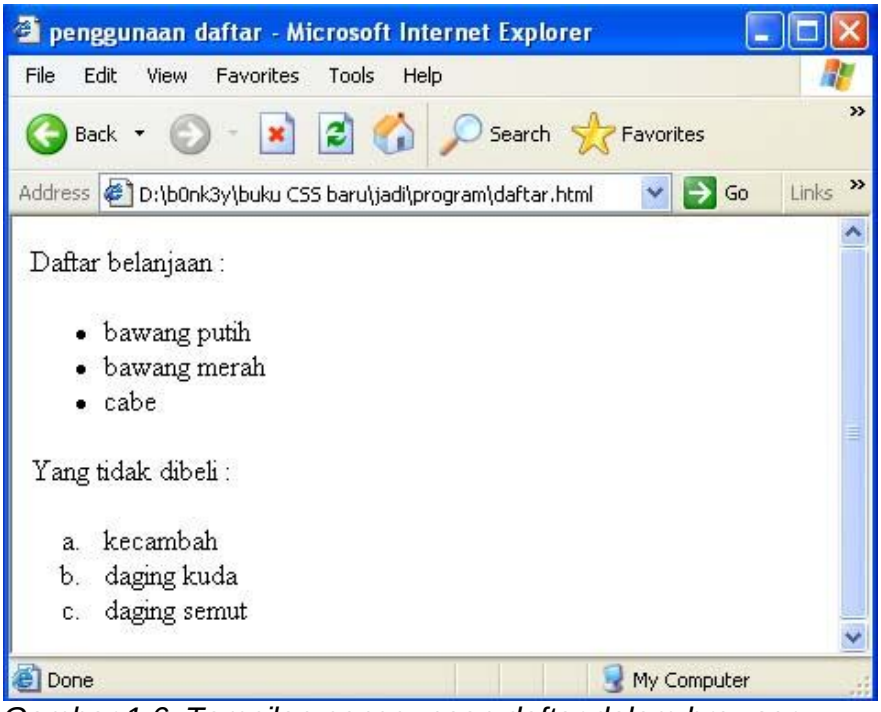
   1. `<OL>...</OL>` : Untuk membuat daftar tak berurutan
   2. `<LI>...</LI>` : Isi daftar pilihan

b. Membuat daftar tak berurut (Unordered List)
   Syntak :

   - `<UL>...</UL>` : Untuk membuat daftar berurut
   - `<LI>...</LI>` : Isi daftar pilihan

Dapat juga anda merubah type daftar menggunakan <ol type=”awalan”> seperti contoh berikut :

```
<html>
<head><title>penggunaan daftar</title>
<body>
Daftar belanjaan :
<ul>
<li>bawang putih
<li>bawang merah
<li>cabe
</ul>
Yang tidak dibeli :
<ol type=a>
<li>kecambah
<li>daging kuda
<li>daging semut
</body>
</html>
```



Gambar 1.6. Tampilan penggunaan daftar dalam browser

7. GAMBAR

Gambar atau image dalam halaman Web memegang fungsi yang sangat penting. Selain fungsinya untuk informasi gambar, image juga akan menambahkan unsur artistik yang akan membuat halaman web nampak lebih hidup sehingga akan lebih menarik dari sisi tampilan web yang dibuat. Sintaks penulisannya adalah :

```
<img src="lokasi-gambar/nama-gambar.tipe" />
```

Sedangkan atribut untuk gambar tersebut berupa :

`src` : letak atau alamat gambar
`alt` : alternatif atau alternate gambar
`width` : lebar gambar
`height` : tinggi gambar

Contoh:

```
<img src=”C:/Documents and Settings/AllUsers/
Documents/My Pictures/Sample Pictures/sunset.jpg”>
```

8. HYPERLINK

Hyperlink atau link dalam HTML berfungsi untuk menghubungkan antara halaman web atau dokumen satu dengan lainnya agar saling terhubung. Syntak dasar penulisannya adalah :

```
<a href=” lokasi/nama.tipe”> Teks yang dijadikan link
</a>
```

Link dalam HTML dibagi menjadi dua yaitu :

a. Link ke halaman lain

Syntak :

```
<a href=”alamatfile/nama.ekstensi”> Hyperlink ke
halaman </a>
```

Contoh :

```
<a href=”c:/web/profil.html”>Profil
Perusahaan</a>
```

b. Link ke servis internet lain

HTTP :

```
<a href=”http://alamathttp/nama.ekstensi”>
Hyperlink ke http </a>
```

FTP :

```
<a href=”ftp://alamatftp/nama.ekstensi”>
Hyperlink ke ftp </a>
```

Mailto :

```
    <a href=”mailto:budi.skom@gmail.com”>Email Saya
</a>
```

9. WARNA

Warna dalam html dapat ditentukan dengan tiga cara, yaitu dengan memasukkan nilai warna dalam nama (common name), hexadecimal, dan nilai RGB.

a. Nilai warna Common name

Anda dapat menentukan warna dengan menggunakan nilai warna common name, yaitu dengan memasukkan warna yang diinginkan dan biasanya menggunakan nilai warna dalam bahasa inggris seperti contoh :

```
    color:blue bgcolor:lime
```

b. Nilai warna Hexadecimal

Dengan nilai warna hexadecimal anda dapat menentukan warna dengan menggunakan nilai warna dalam bentuk hexadecimal yang sama dengan yang digunakan jika kita menggunakan html sebagai contoh :

```
    color:#00ffff bgcolor:#00ccff
```

c. Nilai warna RGB

Dengan menggunakan nilai warna RGB anda dapat menentukan warna dengan menggunakan nilai RGB dengan memasukkan nilai dari warna werah hijau dan biru (Red, Green, Blue) seperti contoh sebagai berikut :

```
    color:rgb(255,255,204)
```

10. T A B L E

Tabel merupakan elemen yang terdiri dari sel dan tersusun atas baris dan kolom. Digunakan untuk membuat bentuk tabel yang berisi kolom dan baris, Syntaknya :

```
  <table>atribut table</table>
```

Atribut table adalah :

`<tr>...</tr>`	: membentuk baris tabel
`<td>...</td>`	: membentuk kolom table

Table juga memiliki atribut untuk pengaturannya yaitu :

`width`	: lebar table
`height`	: tinggi table
`border`	: tebal garis table
`cellspacing`	: spasi antar sel
`cellpadding`	: spasi dalam sel

`align` : perataan teks dalam sel
`bgcolor` : warna background table

Di antaranya tag tersebut dimasukkan perintah pembuatan baris dan kolom. Pembuatan baris, selalu mendahului pembuatan kolom seperti terlihat dalam contoh berikut :

```
<html>
<head><title>membuat table</title></head>
<body>
<table width=300 height=100 border=1 bgcolor=”grey”>
<tr bgcolor=”red”>
 <td width=100>kolom1 baris 1</td>
 <td>kolom2 baris 1</td>
</tr>
<tr height=80>
<td>kolom1 baris 2</td>
<td>kolom2 baris 2</td>
</tr>
</table>
</body>
</html>
```



Gambar 1.7. Tampilan script table dalam browser

11. Form

Form dalam HTML dapat digunakan untuk mengirim data melewati web dan kadang juga digunakan sebagai contact form untuk konversi informasi yang di input-kan oleh user ke dalam e-mail. Dalam penggunaan dalam dokumen web, form harus di kaitkan dengan script program yang akan memproses data yang di inputkan oleh user ke dalam form tersebut.
Tag-tag yang digunakan dalam membuat form dengan HTML berupa :

form, **input**, **textarea**, **select** dan **option**.

Dalam HTML form dalam penulisannya dengan HTML membutuhkan atribut action yang akan menentukan kemana isi dari form yang akan di kirim nantinya. Atribut method akan menentukan tentang bagaimana data dalam form tersebut di kirim yang memiliki nilai get (nilai default) atau post dimana nilai-nilai tersebut adalah nilai yang paling banyak dipakai.
Jadi sebuah elemen form akan terlihat seperti contoh berikut :

```
<form action="processingscript.ekstensi"
method="post"> </form>
```

Tag input adalah inti dari sebuah form, dimana tag input dapat memiliki sepuluh form dan harus ditutup dengan tanda miring "/" seperti contoh :

- `<input type="text" />` adalah textbox standar. Dapat memiliki sebuah atribut value, yang mengatur teks dalam textbox.
- `<input type="password" />` sama dengan textbox, namun menampilan tanda bintang yang menggantikan karakter yang diketikkan.
- `<input type="checkbox" />` adalah bentuk checkbox, yang dapat di tampilkan dan tidak oleh user. Tag ini memiliki sebuah atribut checked, yang digunakan dalam format <input type="checkbox" checked="checked" />.
- `<input type="radio" />` hampir sama dengan checkbox, namun user hanya dapat memilih sebuah radio button dalam kelompok tersebut. Tag ini juga memiliki atribut checked, yang digunakan sama dengan checkbox.
- `<input type="file" />` adalah suatu daerah yang menampilkan file-file dalam komputer seperti ketika anda membuka atau menyimpan dokumen dalam program.
- `<input type="submit" />` biasanya berada pada sebuah tombol dimana ketika dipilih maka akan mengirim **(**submit) form tersebut. Anda dapat mengontrol penampakan text dalam tombol submit (sama seperti dalam tipe button dan reset) dengan atribut value, seperti contoh :
  `<input type="submit" value="ini adalah nilai tombol"/>.`
- `<input type="image" />` adalah sebuah gambar **(**image) dimana ketika dipilih maka akan mengirimkan form. Tag ini juga membutuhkan atribut src, seperti tag img.

- `<input type="button" />` adalah sebuah tombol **(**button**)** yang tidak akan melakukan apapun tanpa penambahan kode ekstra lagi.
- `<input type="reset" />` adalah sebuah tombol dimana ketika dipilih maka akan mengulang kembali **(**reset) field dalam form.
- `<input type="hidden" />` adalah sebuah field yang tidak akan terlihat dan digunakan untuk melewatkan informasi seperti nama alamat email pada form yang akan dikirimkan.

Sebuah textarea biasanya berupa textbox yang luas yang membutuhkan atribut baris (rows) dan kolom (cols) seperti pada contoh berikut :

```
<textarea rows="5" cols="20">teks</textarea>
```

Tag select bekerja dengan tag option dalam pembuatan sebuah drop-down select boxes. Seperti pada contoh berikut :

```
<select>
<option value="opsi pertama">Option 1</option>
<option value="opsi kedua">Option 2</option>
<option value="opsi ketiga">Option 3</option>
</select>
```

Ketika form dikirimkan, nilai dari opsi yang terpilih juga akan dikirimkan. Hampir sama dengan atribut checked dari checkbox dan radio button, sebuah tag option juga dapat memiliki atribut selected yang akan digunakan dalam format :

```
<option value="hewan" selected="selected">kucing</option>
```

Keseluruhan tag yang disebutkan diatas akan terlihat bagus dalam halaman web, namun jika anda mengaitkan form anda dalam sebuah program form maka perintah yang ada di dalamnya akan diabaikan. Hal ini disebabkan karena filed form tersebut memerlukan sebuah nama **(**name)**.** Jadi atribut name harus ditambahkan dalam field seperti pada contoh berikut :

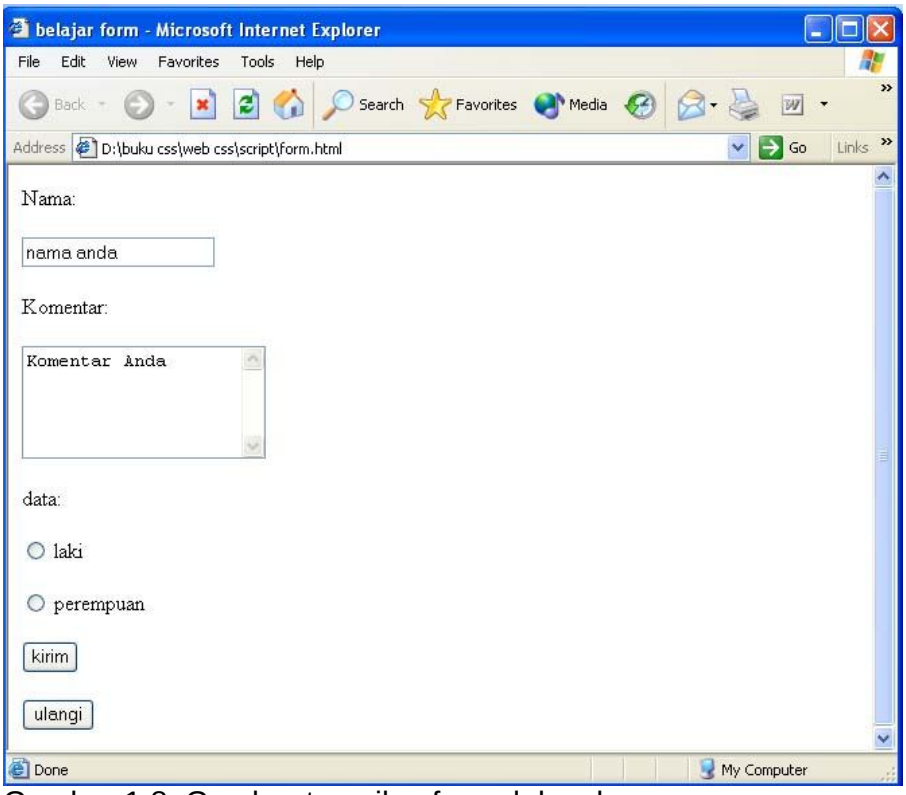
```
<input type="text" name="teks komentar" />
```

Dalam contoh berikut, form akan bekerja jika terdapat file 'koneksi.php' yang ditetapkan dalam atribut action dalam tag form untuk menangani file dalam form yang dikirimkan)

```
<html><head>
<title>belajar form</title>
</head>
<body>
<form action="koneksi.php" method="post">
```

```
<p>Nama:</p>
<p><input type="text" name="nama" value="nama anda"
/></p>
<p>Komentar: </p>
<p><textarea name="komentar" rows="5"
cols="20">Komentar Anda</textarea></p>
<p>data:</p>
<p><input type="radio" name="data" value="Laki-laki"
/> laki</p>
<p><input type="radio" name="data" value="Perempuan"
/> perempuan</p>
<p><input type="submit" value="kirim" /></p>
<p><input type="reset" value="ulangi"/></p>
</form>
</body>
</html>
```

Tampilan dalam browser akan terlihat seperti gambar berikut :



Gambar 1.8. Gambar tampilan form dalam browser

BAB 2

TENTANG CSS

CSS atau Cascading Style Sheets adalah sebuah cara untuk memisahkan isi dengan layout dalam halaman-halaman web yang dibuat. CSS memperkenalkan *template* yang berupa style untuk dibuat dan mengijinkan penulisan yang mudah dari halaman-halaman yang dirancang. Hal ini sangat penting, karena halaman yang menggunakan CSS dapat dibaca secara bolak balik, dan memperbolehkan isinya untuk dilihat oleh pengunjung dari manapun. CSS mampu menciptakan halaman yang tampak sama pada resolusi layar dari pengunjung yang ber-beda tanpa memerlukan suatu tabel. Dengan CSS ini anda akan lebih mudah dalam melakukan setting tampilan keseluruhan web hanya dengan menggantikan atribut-atribut atau perintah dalam style CSS dengan atribut yang kita inginkan, tanpa harus merubah satu per satu atribut tiap elemen yang ada dalam situs yang kita buat. Saat ini CSS merupakan style yang banyak digunakan dikarenakan berbagai kemudahan dan lengkapnya atribut yang dimilikinya, disamping berbagai kemudahan lain yang ditawarkannya. Penggunakan CSS dalam web akan lebih efisien dikarenakan CSS dapat digunakan untuk penggunaan secara berulang pada tag-tag tertentu, sehingga kita tidak usah mengetikkan ulang seluruh perintah pemformatan seperti halnya HTML klasik. CSS akan lebih mudah dipelajari jika anda telah mengetahui struktur pembuatan dokumen web dengan bahasa HTML.

1. CARA KERJA

Cara kerja CSS sangatlah mudah, karena CSS hanya membutuhkan style sebagai penentu dari font, warna, dan format-format lain untuk memformat atribut sebuah halaman web yang kita buat. Tiap style memiliki dua buah elemen dasar yaitu *"selector"* dan *"declaration"*. sebuah "selector" biasanya adalah tag HTML, sementara "declaration" adalah satu atau beberapa perintah/nilai dari css yang menunjukkan type bentuk yang diaplikasikan pada selector. Declaration ini biasanya di tandai dengan tanda kurung kurawal { } , dan perintah atau nilai css yang berbeda dipisahkan satu dengan yang lain dengan menggunakan titik-koma ; seperti terlihat pada contoh berikut *:*

```
<style type="text/css">
.format {font-family:verdana; color:blue;}
</style>
```

Disini terlihat bahwa :

`.format` adalah selector, dan
`{font-family:verdana; color:blue;}` adalah declaration.

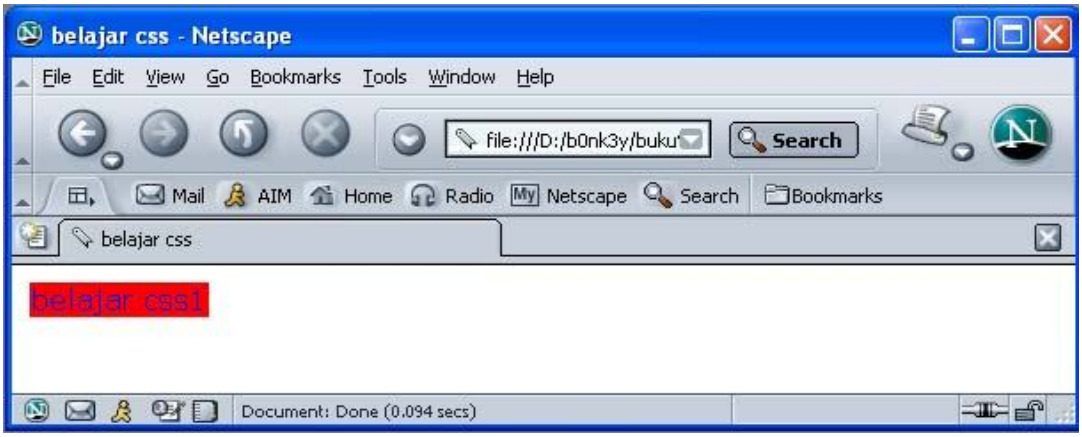
Selector-selector dan style dalam CSS akan dimasukkan dalam sebuah tempat yang sama dalam isi dokumen HTML atau dibuat diluar dokumen HTML yang nantinya akan dipanggil untuk menentukan isi tampilan dokumen HTML. Anda hanya menunjuk pada selector-selector dimana akan mengaktifkan sebuah style yang anda inginkan.
Hal yang paling umum dalam memasukkan kode Style Sheet dengan menggunakan tag <style>, tag <style> ini selalu tampil dalam bagian <head> dan sebelum </head> dari dokumen anda, garis besarnya seperti ini :

```
<html>
<head>
<style type="text/css">
... aturan-aturan css ...
</style>
</head>
<body>
... Dokumen html yang akan diberikan aturan CSS ...
</body>
</html>
```

Untuk menentukan font dan warna-warna tiap kali anda memulai sebuah cell table, anda dapat menentukan sebuah style dan kemudian anda tinggal menunjuk ke style dalam cell table anda. Bandingkan contoh dari sebuah table berikut yang dibuat menggunakan HTML sederhana:

```
<html>
<head><title>belajar css</title>
</head>
<body>
<table>
<tr><td bgcolor="red" align="center"><font
face="verdana" color="blue">belajar css1
</font></td></tr>
</table>
</body>
</html>
```

Tampilan dalam Browser akan terlihat seperti gambar di bawah ini :

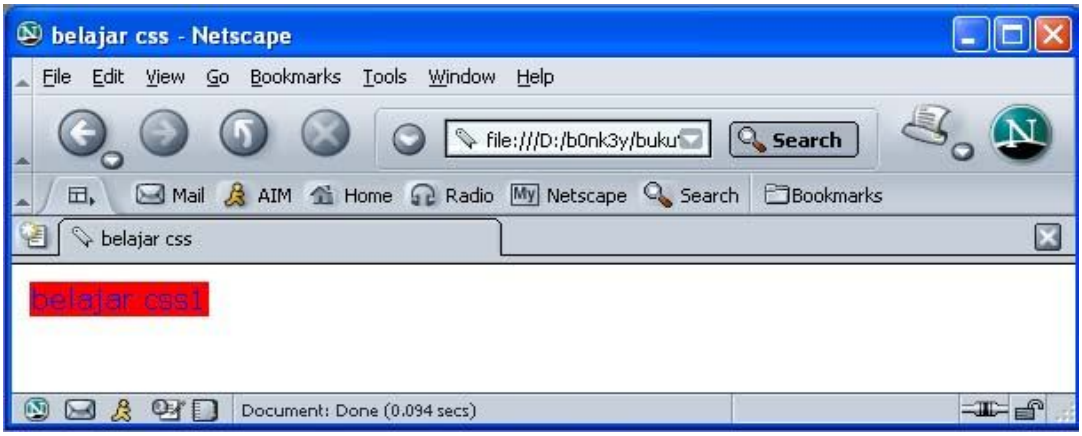


Gambar 2.1. Tampilan script dalam browser Netscape 7

Bandingkan jika dokumen tersebut dibuat dengan menggunakan CSS (dengan menganggap bahwa sebuah selector yang dipanggil *"format"* telah ditetapkan yang akan mengatur format teks dokumen html tersebut).

```
<html>
<head><title>belajar css</title>
<style type="text/css">
.format {background-color:red; color:blue; font-
family:verdana;}
</style>
</head>
<body>
<table>
<tr><td class="format">belajar css1</td></tr>
</table>
</body>
</html>
```

Tampilan dalam Browser akan terlihat seperti gambar berikut :



Gambar 2.1. Tampilan script dalam browser Netscape 7

Dengan menggunakan CSS, maka dokumen yang dibuat menggunakan HTML murni akan dapat lebih disingkat dan akan lebih mengefisiensikan waktu yang dibtuhkan. Bayangkan jika kita membuat sebuah situs yang kompleks dengan banyak atribut untuk tiap halaman dalam situs kita. Maka kita harus menentukan tiap atribut untuk tiap halaman tersebut satu per satu. Bagaimana jika terdapat seratus halaman dalam situs tersebut? Bayangkan berapa banyak waktu yang kita habiskan hanya untuk memformat tiap halaman jika terdapat kesalahan dalam halaman-halaman tersebut? Namun masalah tersebut dapat lebih diminimalisir dengan menggunakan CSS karena kita hanya akan membuat sebuah style yang akan memformat keseluruhan halaman dalam situs yang dibuat dan kita cukup menentukan selector dalam style yang akan digunakan dalam halaman tersebut.

2. KEUNTUNGAN CSS

Sementara CSS memungkinkan anda untuk memisahkan layout dari isi dokumen, CSS juga memungkinkan anda untuk menentukan layout yang lebih ampuh dibandingkan dengan HTML.

Dengan CSS, anda dapat :

a. Menentukan tampilan dari halaman-halaman web dalam sebuah tempat tanpa mengulangnya kembali melalui script html anda. Ini akan lebih efisien dibandingkan menentukan warna dan font tiap kali kita memulai sebuah cell baru dalam sebuah table.
b. Dengan mudah mengubah tampilan dari halaman-halaman web walaupun setelah halaman tersebut dibuat. Setelah style ditentukan dalam sebuah tempat, anda dapat mengubah tampilan keseluruhan situs sekaligus tanpa harus mengganti tag-tag tertentu ketika mengubah salah satu tampilan dari elemen dalam halaman anda.
c. Menentukan ukuran font dan atribut-atribut yang sejenis dengan keakuratan yang sama dengan sebuah “word processor” dan tidak dibatasi hanya untuk tujuh ukuran font berbeda yang telah ditetapkan dalam HTML.
d. Menentukan isi hingga ukuran pixel dari tiap halaman web dengan akurat.
e. Menentukan kembali keseluruhan tag-tag HTML. Sebagai contoh jika anda ingin membuat tebal tag menjadi berwarna merah menggunakan font yang special dapat dilakukan dengan sangat mudah.
f. Menentukan style yang kita inginkan untuk sebuh link, seperti menyingkirkan garis bawah yang biasa terlihat di bawah link.
g. Menentukan layer-layer yang dapat diposisikan di atas layer-layer lainnya (kadang digunakan untuk menu-menu menjadi terlihat lebih menarik).
h. Halaman web anda akan di load lebih cepat, karena tidak diisi dengan tag-tag yang mengatur tampilan web anda. Style yang mengatur tiap halaman akan disimpan dalam sebuah dokumen CSS yang hanya di load sekali ketika pengunjung memasuki situs anda.

Hanya satu kekurangan dari CSS, yaitu :

Hanya dapat bekerja dalam browser versi 4 atau lebih. Namun jika kita lihat dewasa ini bahwa lebih dari 95% dari keseluruhan browser pada saat ini telah banyak yang mendukungnya. Jadi kekhawatiran tentang tidak dapat ditampilkannya situs yang kita buat menggunakan CSS akan makin terkikis.

BAB 3

PENEMPATAN CSS DALAM HTML

Menempatkan style CSS dalam dokumen web yang kita buat dalam bahasa html dapat dilakukan dengan beberapa cara dan tingkatan yang berbeda sesuai dengan keinginan kita. Cara penempatan dalam HTML adalah dengan menggunakan beberapa cara, diantaranya adalah menggunakan Inline Style, Internal Style, External style, dan juga teknik Importing dengan Internal Style.

1. Inline Style

Adalah CSS yang dibuat dalam sebuah tag HTML yang hanya berlaku untuk dokumen yang diapitnya saja. Biasanya teknik ini digunakan pada pemformatan khusus pada sebuah elemen HTML dan tidak digunakan untuk memformat seluruh elemen dalam dokumen web. Contohnya adalah sebagai berikut :

```
<html>
<head>
<title>belajar css</title>
</head>
<body>
<font style="font-family:Arial; font-size:
20px; background-color:yellow;">Penerapan Inline
Style </font></body>
</html>
```

Dalam script di atas dapat diketahui bahwa style tersebut memiliki nama style arial yang memiliki nilai atau value untuk style tersebut. Adapun nilai style adalah jenis font arial dengan ukuran 20 pixel dan memiliki warna latar kuning. Jika dilihat dalam browser, akan terlihat seperti pada gambar berikut :



Gambar 3.1. Penerapan Inline Style dilihat dalam browser

2. Internal Style

Pada teknik ini style diletakkan pada tengah tag antara tag <head> dan </head>. Aturan-aturan dalam style ini diatur sedemikian rupa untuk digunakan pada suatu tempat maupun untuk keseluruhan situs. Contoh dari teknik ini terlhat pada contoh berikut :

```
<html>
<head>
<title>Belajar CSS</title>
<style type="text/css">
.blog {font-family: Verdana, Arial, Helvetica,
sans-serif; font-size: 24px; color: white;
background-color: blue;}
</style>
</head>
<body>
<div class=”blog”>Penerapan Internal Style</div>
</body>
</html>
```

Maksud dari <!-- aturan css --> diatas adalah berguna agar browser yang tidak mendukung CSS tidak menampilkan perintah-perintah CSS, jadi perintah CSS didalam <!-- --> akan dianggap sebagai komentar HTML dan tidak akan ditampilkan dalam browser walaupun browser tersebut tidak mendukung CSS sekalipun. Penerapan dari teknik ini dalam dokumen html akan terlihat seperti dalam gambar berikut :



Gambar 3.2. Penerapan Internal Style dilihat dalam browser

3. Import Style

Dengan teknik ini sebuah Style tidak disimpan pada halaman tersebut, namun untuk menghubungkannya dengan halaman web yang kita buat digunakan perintah import yang terdapat pada style CSS. Seperti terlihat pada contoh berikut :

```
<html>
<head>
<title>Belajar CSS</title>
<style type="text/css">
<!--
@import url
(http://www.nama_situs.com/global.css);
-->
</style>
</head>
<body>
<div class=blog>Penerapan Import Style</div>
</body>
</html>
```

Pada teknik diatas, dokumen HTML akan mengambil atau import style CSS untuk dipergunakan dalam dokumen tersebut. Dimana pada contoh diatas dokumen akan mengimport file style dengan nama global yang berekstensi CSS pada alamat url http://www.nama_situs.com dengan menggunakan perintah :

```
@import url
(http://www.nama_situs.com/global.css);
```

4. External Style

Dengan menggunakan teknik ini kita dapat memanggil style CSS pada file CSS yang kita inginkan dengan menggunakan perintah “Link rel” yang berfungsi untuk menghubungkan kedalam sebuah style CSS eksternal dengan nama yang telah kita tentukan seperti terlihat pada contoh :

```
<link rel="stylesheet" type="text/css"
href="default.css">
```

Dalam contoh diatas kita menghubungkan sebuah dokumen html dengan sebuah style eksternal dengan nama default.css yang telah kita buat sedemikian rupa sehingga style tersebut dihubungkan untuk memformat tampilan dalam dokumen html tersebut.

Dalam mengimport file CSS, kita dapat langsung mengimport beberapa file CSS sekaligus dalam sebuah dokumen agar memperkaya tampilan dokumen kita seperti terlihat pada contoh berikut :

```
<link rel="stylesheet" type="text/css"
href="default1.css">
<link rel="stylesheet" type="text/css"
href="default2.css">
<link rel="stylesheet" type="text/css"
href="default3.css">
```

Atau kita dapat menggunakan perintah :

```
<style>
<!--
@import url(default1.css);
@import url(default2.css);
@import url(default3.css);
-->
</style>
```

Perbedaan dalam teknik terakhir adalah perintah import tersebut akan mengambil style CSS dengan cara import style dari alamat url yang kita tentukan. Cara ini lebih efisien dikarenakan dokumen yang kita beri perintah import tersebut hanya akan mengimport file css yang pada akhirnya, style CSS tersebut hanya perlu di download sekali saja dalam dokumen style sheet eksternal yang terpisah dengan dokumen HTML asli. Ketika surfing ke dalam situs anda maka CSS akan disembunyikan dalam komputer user sehingga akan memperkecil ukuran file sebuah situs yang kita buat. Jadi jika anda menggunakan cara terakhir ini, maka sebelumnya anda harus membuat sebuah dokumen style CSS eksternal tersebut dengan ekstensi .css dimana dokumen style tersebut berisi aturan yang akan mengatur tampilan dokumen web yang dibuat

BAB 4

STRUKTUR CSS

Dalam pembuatan dokumen web menggunakan style CSS di kenal adanya aturan atau struktur penulisan baku agar style CSS yang kita buat tersebut dapat ditampilkan dengan baik dalam dokumen web yang dibuat. Contoh :

```
<html>
<head>
<title>HTML Selector</title>
<style type="text/CSS">
<!--
p {font-family: verdana;}
-->
</style>
</head>
<body>
<p> penggunaan HTML Selector dalam CSS</p>
</body>
</html>
```

Pada contoh diatas dipergunakan tag HTML sebagai selector dan setiap huruf yang berada diantara tag dan akan memiliki jenis font verdana sesuai dengan value atau nilai yang diberikan dalam property style tersebut. Seperti terlihat dalam gambar berikut :



Gambar 4.1. Penerapan HTML selector dilihat dalam browser

1. SELECTOR

Jika dalam HTML kita mengenal adanya tag-tag yang menyusun sebuah dokumen HTML, maka dalam CSS kita juga mengenal dengan sebutan

selector. Selector adalah nama-nama yang diberikan untuk style-style yang berbeda baik itu style internal maupun eksternal. Dalam style tujuan, anda harus menentukan bagaimana tiap selector seharusnya bekerja yang dinamakan properties di dalam tanda { } atau curly bracket yang dapat memiliki nilai berupa font, warna dll, kemudian dalam body di halaman web anda, anda arahkan ke selector-selector ini untuk mengaktifkan style tersebut.

Karena CSS memiliki aturan tersendiri dalam penulisan sintaknya dalam CSS, dan penulisan sintak tersebut terbagi ke dalam tiga bagian yaitu: Selector, Property, dan Value. Sebagai penjelas coba perhatikan contoh berikut :

```
selector {property: value;}
```

Selector dalam CSS adalah elemen yang akan didefinisikan dalam style CSS, sedangkan elemen ini dapat berupa tag HTML maupun Class. Sedangkan Property dalam CSS adalah atribut yang berfungsi untuk mendefinisikan Selector, sedangkan Value adalah nilai atau harga dari sebuah Property.

Sebagai contoh :

```
<html>
<head>
<style type="text/css">
b.besar {color:red; font-size:15px; font-
family:arial; }
</style>
</head>
<body>
<b>ini adalah huruf besar dengan html</b><br>
<b class="besar">ini adalah huruf besar dengan CSS</b>
</body>
</html>
```

Dalam kasus ini B.besar adalah selector. Contoh diatas akan menghasilkan output sebagai berikut :



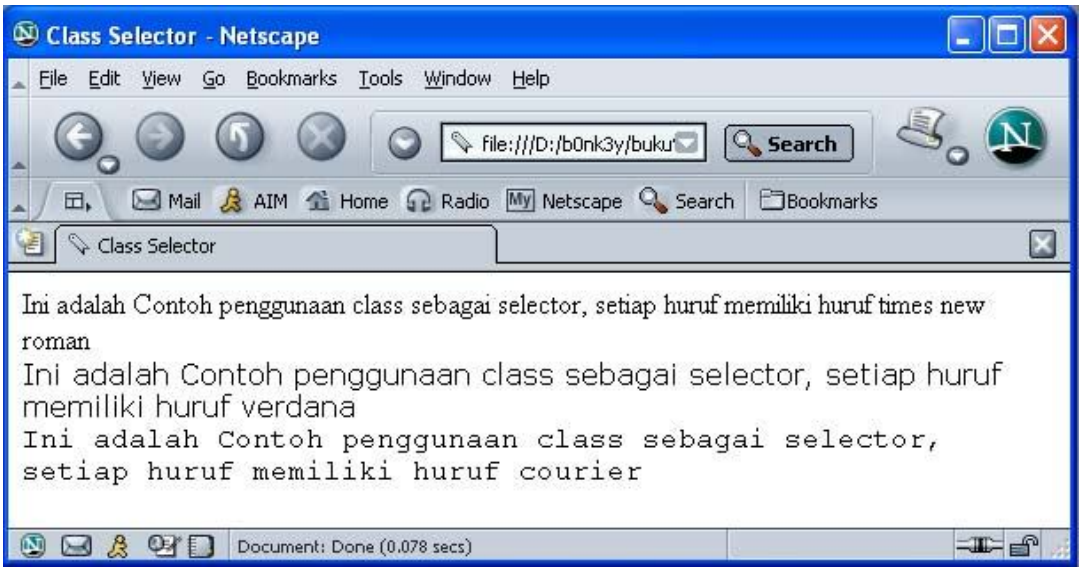
Gambar 4.2. Penerapan selector dilihat dalam browser

2. CLASS SELECTOR

Digunakan untuk menentukan style yang dapat digunakan tanpa menentukan lagi tag HTML sederhana. Penulisan class adalah dengan menggunakan tanda titik atau dot. Selain aturan penulisan diatas, CSS juga mengenal Class Selector dimana dalam elemen yang sama kita dapat menerapkan lebih dari satu style. Adapun contohnya adalah sebagai berikut :

```
<html>
<head>
<title>Class Selector</title>
<style type="text/CSS">
<!--
f.times {font-family: times;}
f.verdana {font-family: verdana;}
.courier {font-family: "courier";}
-->
/style>
</head>
<body>
<f class="times"> Ini adalah Contoh penggunaan class
sebagai selector, setiap huruf memiliki huruf times
new roman</f><br>
<f class="verdana"> Ini adalah Contoh penggunaan
class sebagai selector, setiap huruf memiliki huruf
verdana</f><br>
<f class="courier"> Ini adalah Contoh penggunaan
class sebagai selector, setiap huruf memiliki huruf
courier</f>
</body>
</html>
```

Tampilan script tersebut jika dilihat dalam browser akan terlihat seperti dalam gambar berikut :



Gambar 4.3. Penerapan Class selector dilihat dalam browser

Pada contoh diatas, tag f memiliki 2 class yang berbeda untuk memformat paragraf yaitu .times dan .verdana dikarenakan tag class .times dan .verdana adalah class yang dimiliki oleh tag f maka ia hanya dapat dikenakan pada elemen f saja. Lain halnya dengan class .courier, ia dapat dipergunakan oleh semua elemen tanpa harus menggunakan tag f di awal dan akhir elemen yang akan di format.

Nama untuk tiap class dapat berbeda-beda, Anda dibebaskan menentukan nama class sendiri, namun ada baiknya memilih nama dari tiap class yang mengacu kepada fungsi dari class tersebut. Hal ini ditujukan agar anda tidak bingung mempergunakan class pada saat menggabungkannya ke dalam tag HTML. Contoh :

```
<p class="blog"> Ini teks untuk paragraph pada blog
</p>
<p class="menukiri">Ini teks untuk paragraph pada
menu kiri</p>
```

3. ID SELECTOR

ID selector digunakan untuk menentukan style yang berhubungan dengan objek-objek dengan sebuah ID yang unik. Berbeda dengan Class Selector yang dapat kita pergunakan pada berbagai tag, ID Selector ini memiliki sifat-sifat yang berbeda. ID selector hanya dapat dipergunakan pada satu elemen saja pada setiap halaman web yang kita buat. ID Selector menggunakan tag # (kres) sebelum menggunakan nama Selector. Adapun contohnya adalah sebagai berikut :

```
#helvetica {font-family: Helvetica;}
```

Dalam penerapannya pada tag HTML, Anda dapat menggunakan aribut span ataupun div, seperti terlihat pada contoh berikut :

```
<span id="helvetica">Ini adalah huruf
helvetica</span>
```

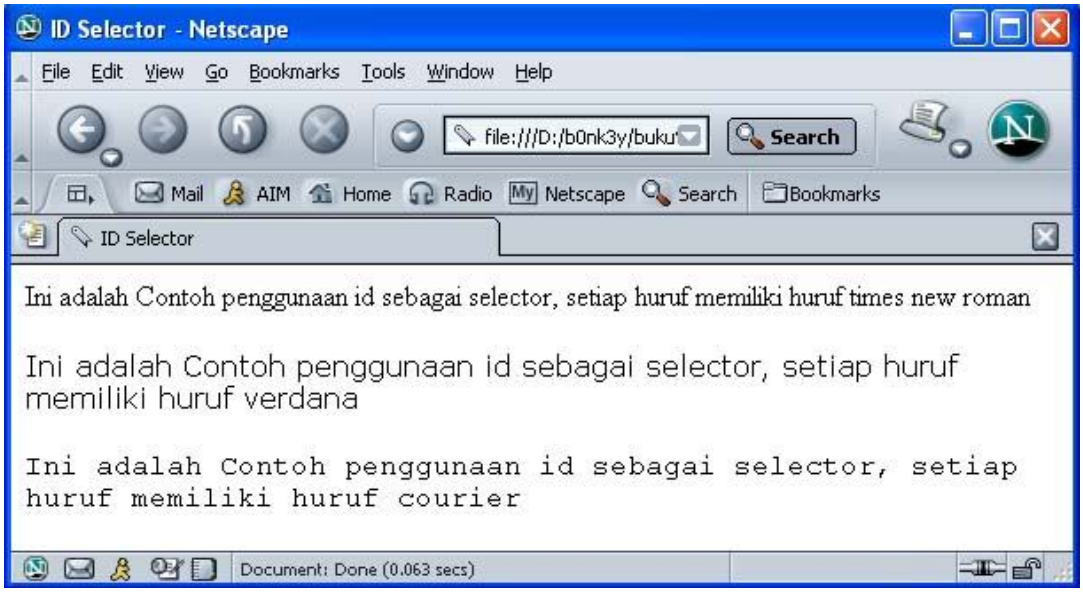
Atau

```
<div id=”helvetica”>Ini adalah huruf helvetica</div>
```

Adapun penulisan dalam dokumen dapat dituliskan seperti pada contoh berikut :

```
<html>
<head>
<title>ID Selector</title>
<style type="text/CSS">
<!--
#times {font-family: times;}
#verdana {font-family: verdana;}
#courier {font-family: "courier";}
-->
</style>
</head>
<body>
<div id="times"> Ini adalah Contoh penggunaan id
sebagai selector, setiap huruf memiliki huruf times
new roman</div><br>
<div id="verdana"> Ini adalah Contoh penggunaan id
sebagai selector, setiap huruf memiliki huruf
verdana</div><br>
<div id="courier"> Ini adalah Contoh penggunaan id
sebagai selector, setiap huruf memiliki huruf
courier</div>
</body>
</html>
```

Tampilan yang terlihat dalam browser akan terlihat seperti pada gambar berikut ini :



Gambar 4.4. Penerapan ID selector dilihat dalam browser

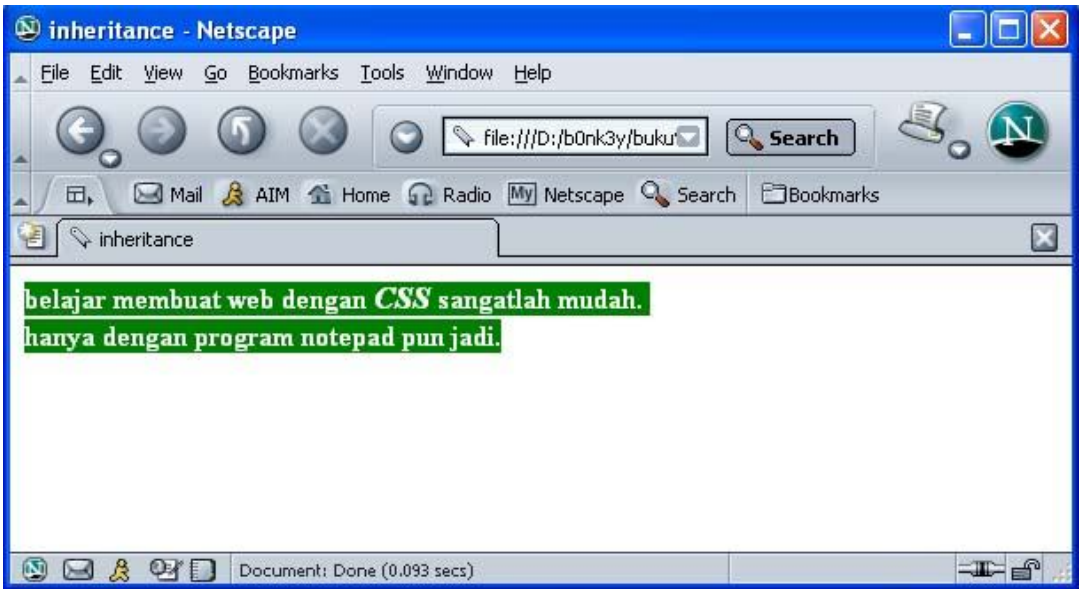
Selain yang dijelaskan diatas, dalam CSS kita juga mengenal adanya Pewarisan (inheritance), Pseudo Classes, Pseudo-Element, dan Selectors-Kontekstual (Contextual Selector).

4. PEWARISAN / INHERITANCE

Yang dimaksud pewarisan atau inheritance disini adalah satu kondisi dimana bagian-bagian HTML dalam dokumen kita yang kekurangan dalam spesifikasi aturan CSS dapat mengikuti atau dibuat menurut aturan CSS yang menutupinya. Setiap aturan yang tidak ada dalam format huruf yang dibuat dengan HTML atau ingin menggantikannya dengan aturan CSS, tetap akan mempengaruhi huruf yang ada didalam kurungan CSS.

```
<html>
<head><title>inheritance</title>
<style type="text/css">
b {background-color : green; color : white;}
</style>
</head>
<body>
<B>
belajar membuat web dengan <font
size="+1"><em>CSS</em></font> sangatlah mudah. <br>
hanya dengan program notepad pun jadi.
</B>
</body>
</html>
```

Jika dilihat dari contoh diatas dapat dilihat bahwa tag <B> dalam aturan CSS akan mendefinisikan warna font dan warna latar dari font sedangkan tag <B> dalam HTML bisa diartikan sebagai bold atau format untuk menebalkan huruf. Sedangkan tag <font> yang berada dalam lingkup CSS dapat menerapkan aturan HTML dasar akan tetapi tetap terpengaruh dengan kondisi warna dan tebalnya huruf dari aturan CSS yang melingkupinya. Cara ini dapat anda gunakan untuk dikombinasikan lebih lanjut dengan tag-tag lain agar sesuai dengan keperluan dalam dokumen anda.



Gambar 4.5. Penerapan Inheritance dilihat dalam browser

5. PSEUDO CLASSES

Pseudo classes dalam CSS dibuat tebal dalam selector-selector untuk menentukan sebuah statemen atau relasi selector, dan penulisannya hanya dipisahkan dengan sebuah tanda titik dua : diantara selector dan pseudo class. Adapun contoh format penulisannya adalah :

```
selector:pseudo class { property: value; }
```

Banyak format CSS yang tidak didukung oleh kebanyakan browser, namun terdapat empat pseudo classes yang dapat digunakan secara aman ketika diaplikasikan kedalam hyperlink atau link, yaitu :

* link : untuk tampilan sebuah link yang belum dikunjungi.
* visited : untuk tampilan sebuah link ke halaman yang telah dkunjungi.
* active : untuk tampilan sebuah link yang sedang aktif (ketika di klik).
* hover : untuk tampilan sebuah link yang hover (ketika cursor melintas diatasnya).

Contoh penggunaan :

```
a.snowman:link {color:blue;}
a.budi:visited {color:purple;}
a.budi:active {color:red;}
a.budi:hover {text-decoration:none; color:blue;
background-color:yellow;}
```

Anda dapat menggunakan hover pseudo class dengan elemen-elemen lain selain link. Sayangnya Internet Explorer masih tidak mendukung metode ini karena banyak tampilan yang dapat anda buat dan terlihat bagus dalam browser lain namun tidak terlihat dengan menggunakan browser ini.

6. PSEUDO ELEMENT

Pseudo element dalam CSS ditempatkan dalam sebuah selector sama seperti pada pseudo classes, penulisannya akan seperti contoh berikut :

```
selector:pseudoelement {property: nilai;}.
```

Contoh setting huruf dan baris pertama :
Pseudo element pada huruf pertama di aplikasikan elemen dari huruf pertama dan baris pertama pada atas baris sebuah elemen. Anda dapat membuat huruf besar dan huruf tebal baris pertama pada paragraf dengan menggunakan perintah berikut :

```
p:first-letter {font-size:5px; float:left;}
p:first-line {font-weight:bold;}
```

BEFORE DAN AFTER

Pseudo element before dan after digunakan dengan property content untuk menempatkan isi dari sebuah elemen tanpa menyentuh kode HTML. Nilai dari property content ini dapat berupa open-quote, close-quote, no-open-quote, no-close-quote, berbagai string yang disertakan dalam tanda kutip atau berbagai image gambar menggunakan url(nama image) seperti contoh berikut :

```
blockquote:before {content: open-quote;}
blockquote:after {content: close-quote;}
li:before {content: "POW: "}
p:before {content: url(images/jam.jpg)}
```

7. SELEKTOR KONTEKSTUAL / CONTEXTUAL-SELECTOR

Selektor Kontekstual dalam CSS hampir sama dengan pernyataan-pernyataan kondisional dimana deklarasi dalam Style-Sheet CSS yang ada akan berpengaruh jika menemui kondisi atau keadaan tertentu. Contoh dari selector kontekstual adalah :

```
<html>
```

```
<head>
<title>selector kontekstual</title>
<style type="text/css">
b em {font-family: verdana; background: blue; color:
white; }
</style>
</head>
<body>
<b>
belajar membuat web dengan <em>CSS</em>sangatlah mudah
<br>
</b>
<em>hanya dengan program notepad pun jadi</em>
</body>
</html>
```

Dalam contoh diatas dapat dilihat bahwa dokumen web tersebut akan menerapkan style CSS tersebut pada dokumen yang berada dalam tag (font verdana, warna latar biru dan font berwarna putih) dimana tag tersebut berada di dalam tag dan (bold/tebal), sedangkan tag yang berada di luar tag dan tag tidak akan terpengaruh dengan setting style tersebut atau hanya memiliki nilai biasa atau huruf miring. Jika dilihat dlam browser maka tampilannya adalah sebagai berikut :



Gambar 4.6. Penerapan Selector Kontekstual dilihat dalam browser

BAB 5

PENGELOMPOKAN DAN NESTING CSS

Dalam dokumen web yang dibuat menggunakan style CSS kita dapat memformat berbagai aturan yang dibuatnya menjadi makin ramping sesuai dengan keinginan kita. Adapun format untuk style CSS tersebut berupa :

1. PENGELOMPOKAN / GROUPING

Dalam CSS kita mengenal adanya pengelompokan atau grouping, pengelompokan disini berfungsi untuk mengelompokan berbagai selector ID atau class yang memiliki nilai property yang sama. Anda dapat memberikan property yang sama tersebut ke dalam sejumlah selector tanpa harus mengulang penulisannya dengan membagi selector-selector tersebut dengan koma.Pegelompokan dalam CSS berfungsi untuk menyingkat berbagai penulisan dari aturan-aturan yang banyak hingga membentuk aturan baru yang jauh lebih singkat, contoh dibawah ini adalah mengelompokkan tag yang memiliki nilai yang sama :

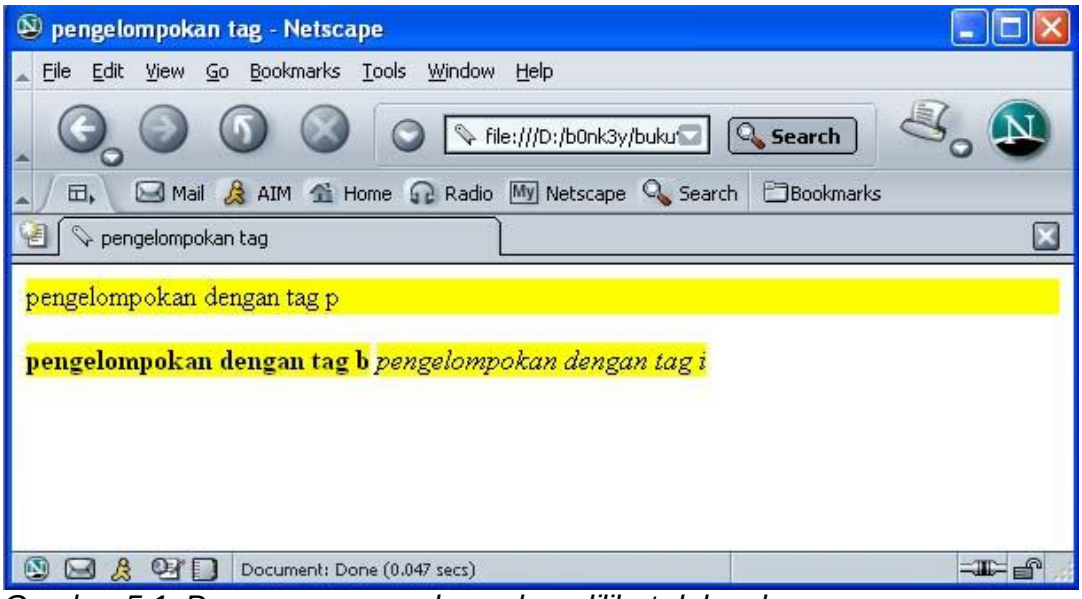
```
<style type="text/CSS">
p {background-color: yellow;}
b {background-color: yellow;}
i {background-color: yellow;}
</style>
```

Maka, dalam pengelompokan menggunakan CSS penulisannya dapat diringkas menjadi :

```
<html>
<head>
<title>pengelompokan tag</title>
<style type="text/CSS">
p, b, i {background-color: yellow;}
</style>
</head>
<body>
<p>pengelompokan dengan tag p</p>
<b>pengelompokan dengan tag b</b>
<i>pengelompokan dengan tag i</i>
</body>
</html>
```

Maksud dari aturan diatas adalah CSS akan memberikan nilai p yang dalam dokumen html seharusnya adalah format untuk membuat paragraf, namun juga memiliki nilai warna background biru yang sama dengan tag b

(bold/tebal) dan i (italic/miring) seperti terlihat dalam browser Netscape berikut ini :



Gambar 5.1. Penerapan pengelompokan dilihat dalam browser

Bila kita menerapkan pegelompokan tersebut dalam selector class, ID dan tag, maka akan terlihat seperti contoh berikut :

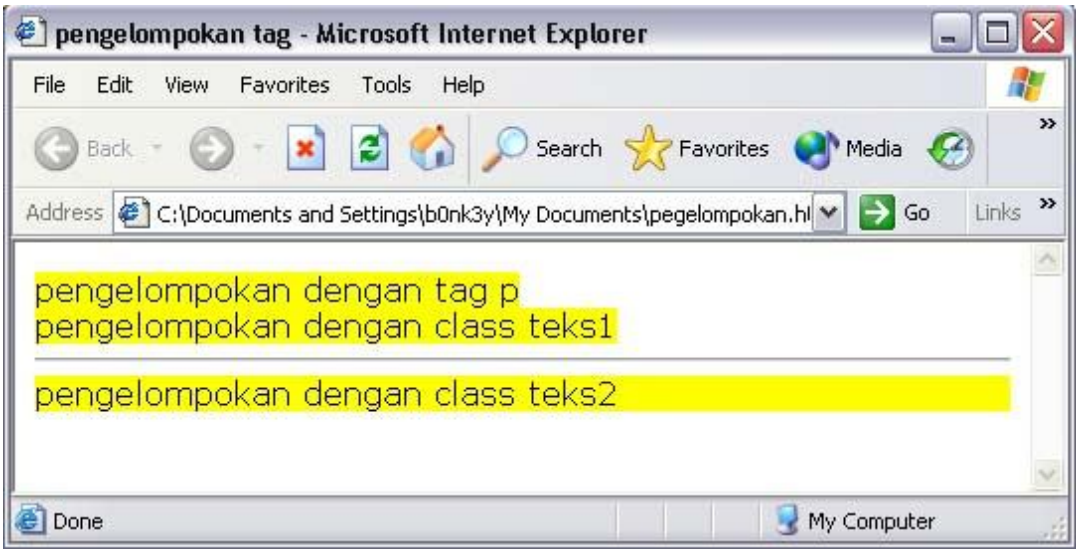
```
<style type=”text/CSS”>
p {font-family: Arial, Verdana;}
.teks1 {font-family: Arial, Verdana;}
#teks2 {font-family: Arial, Verdana;}
</style>
```

Maka, dalam pengelompokan dengan CSS penulisannya akan menjadi :

```
<html>
<head>
<title>pengelompokan tag</title>
<style>
.teks, .title, #teks2 {font-family:verdana;
background-color:yellow;}
</style>
</head>
<body>
<span class="teks">pengelompokan dengan tag
p</span><br>
<span class="title">pengelompokan dengan class
teks1</span><hr>
```

```
<div id="teks2">pengelompokan dengan class teks2
</div>
</body>
</html>
```

Terlihat dalam browser akan seperti :



Gambar 5.2. Tampilan pengelompokan class, ID dan tag dalam browser

2. NESTING

CSS memiliki struktur bahasa yang baik sehingga dapat dimanfaatkan untuk menyingkat style CSS agar lebih efisien dalam dokumen web yang dibuat menggunakan CSS. Anda tidak membutuhkan class atau ID selector yang terlalu banyak untuk membuat sebuah dokumen web dengan bantuan CSS dikarenakan anda dapat menentukan property dari selector untuk ditempatkan dalam selector lain. Sebagai contoh :

```
#atas {background-color: #ccc; padding: 2em}
#atas h1 {color: #ff0;}
#atas p {color: red;font-weight: bold;}
```

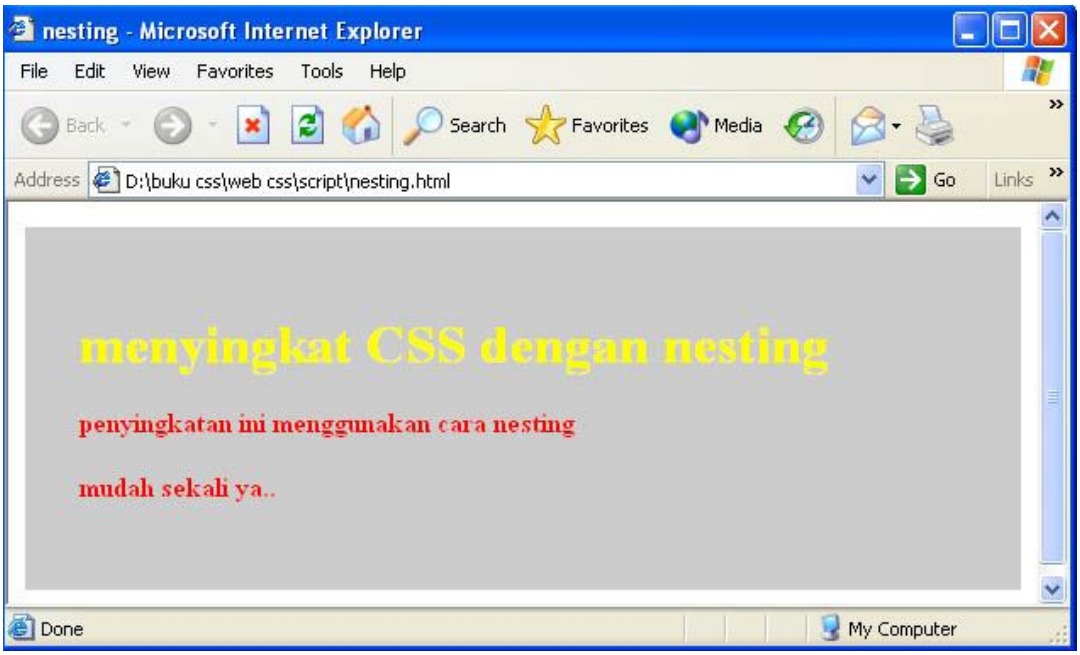
Dengan menghilangkan kebutuhan akan class atau ID yang cukup banyak tersebut, maka anda dapat menyingkatnya dengan menggunakan cara nesting, sehingga dalam penulisannya akan menjadi lebih singkat seperti ini :

```
<div id="atas">
<h1>menyingkat CSS dengan nesting</h1>
<p>penyingkatan ini menggunakan cara nesting</p>
<p>mudah sekali ya..</p>
</div>
```

Dengan cara ini yaitu memisahkan selector dengan spasi maka dapat di artikan bahwa h1 yang berada dalam ID atas dimana memiliki nilai warna latar #ccc dan warna isi #ff0 dalam h1 serta penambahan tag p dalam ID atas yang berarti merah dan tebal.
Cara ini dapat menjadi lebih kompleks lagi dikarenakan cara ini dapat diterapkan lebih dari dua tingkatan seperti di dalam dan di dalamnya ID atau class tertentu yang kita inginkan hingga diakhiri dengan penutup seperti pada contoh diatas. Adapun script keseluruhan jika anda menggunakan nesting dalam html adalah :

```
<html>
<head><title>nesting</title>
<style>
#atas {background-color: #ccc; padding: 2em}
#atas h1 {color: #ff0;}
#atas p {color: red;font-weight: bold;}
</style></head>
<body>
<div id="atas">
<h1>menyingkat CSS dengan nesting</h1>
<p>penyingkatan ini menggunakan cara nesting</p>
<p>mudah sekali ya..</p>
</div>
</body>
</html>
```

Tampilan dalam browser akan terlihat :



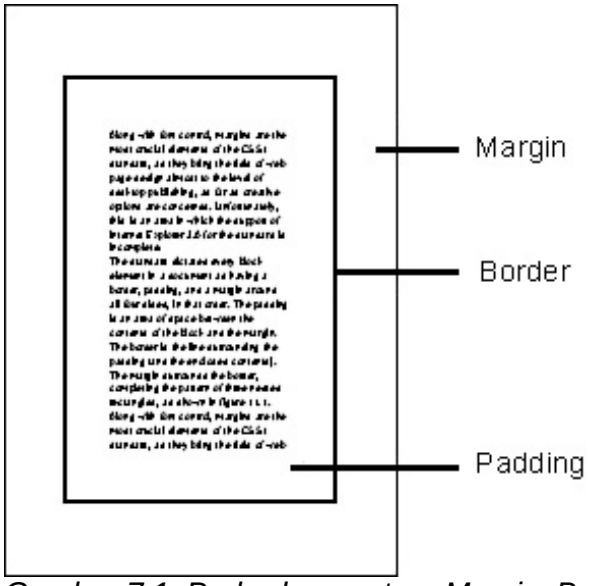
Gambar 5.3. Tampilan dokumen dengan teknik nesting

BAB 7

LAYOUT DALAM CSS

3. LAYOUT

Dalam CSS kita juga mengenal adanya properti – properti yang dipergunakan untuk mensetting layout dokumen web dengan menggunakan style CSS. Salah satunya adalah yang kita kenal dengan menggunakan properti margin dan padding. Margin dan padding dalam web yang dibuat menggunakan teknologi CSS biasanya digunakan untuk memformat tampilan layout situs. Model format dengan style CSS dapat diartikan bahwa tiap-tiap elemen akan dikelilingi oleh tiga area yang berbeda. Area tersebut adalah *padding*, *border*, dan *margin.* Masing-masing entitas tersebut dapat diatur melalui property CSS yang mengijinkan para developer untuk secara singkat mengatur tampilan dan posisi dari setiap elemen HTML yang dibuatnya.



Gambar 7.1. Perbedaan antara Margin, Border, dan Padding

a. Margin

Margin adalah pemberian jarak atau batas pada suatu objek dengan objek yang berada diluar. Adapun nilai margin (garis tepi / daerah pinggir) halaman web dapat kita atur besar dan letaknya melalui properti dari margin seperti “margin-top”, “margin-bottom”, “margin-right”, dan “margin-left”. Contoh penggunaan margin dalam mensetting dokumen web adalah sebagai berikut :

```
<html>
<head><title>mengatur margin</title>
<style type ="text/css">
```

```
    body {margin-top:30; margin-bottom:30; margin-
    right:40; margin-left:40}
    </style>
    </head>
    <body>
    Bagaimana mengatur margin?<br>
    Dengan menggunakan property margin dong..
    </body>
    </html>
```

Jika dilihat dalam browser maka akan terlihat seperti pada gambar :



Gambar 7. 2. Penggunaan atribut margin dilihat dalam browser

Dalam contoh diatas dapat diketahui bahwa dokumen tersebut akan memiliki lebar margin atau garis tepi atas sepanjang 30 pixel, lebar garis tepi bawah sepanjang 30 pixel, garis tepi kanan 40 pixel dan garis tepi kiri sepanjang 40 pixel.
Anda juga dapat memampatkan seluruh property margin dengan menggunakan Shorthand property, yaitu cara penulisan margin dengan menulis hanya valuenya saja seperti :

```
    <style="text/css">
    body {margin: 30px 30px 40px 40px}
    </style>
```

Yang perlu diperhatikan dengan penulisan shorthand property adalah urutan penulisannya yang berurutan yaitu urutan dimulai dari property top, right, bottom, dan left. Atau anda dapat mengeset nilai yang sama untuk lebar keseluruhan margin baik margin atas, margin bawah, margin kanan, dan margin kiri dengan menuliskan :

```
<style=”text/css”>
body {margin: 30px}
</style>
```

b. Border

Dalam CSS anda juga dapat menyesuaikan lebar border dengan menggunakan property “border-top-width”, “border-left-width”, dan “border-rigth-width” dan menentukan nilainya, atau dengan menata suatu border dengan shortcut property “border-width” seperti pada contoh berikut :

```
<html>
<head><title>mengatur border</title>
<style type ="text/css">
body {border-top-width: 30px; border-right-width:
30px; border-bottom-width: 40px; border-left-
width:40px}
</style>
</head>
<body>
Bagaimana mengatur border?<br>
Dengan menggunakan property border dong..
</body>
</html>
```

Jika dilihat dalam browser maka akan terlihat seperti pada gambar :



Gambar 7.3. Penggunaan atribut border dilihat dalam browser

Dalam contoh script diatas berarti halaman web yang dibuat akan menggunakan lebar border atas sebesar 30 pixel, lebar border kanan sebesar 30 pixel, lebar border bawah sebesar 40 pixel, dan lebar border kiri sebesar 40 pixel. Atau anda dapat menentukan lebar keseluruhan border dengan menggunakan script :

```
body {border-width: 30px}
```

Anda dapat pula menentukan lebar border dengan kata bantu "thick", "medium", "thin" dan "none", seperti :

```
body {border-top-width: thick; border-right-
width: medium; border-bottom-width: thin; border-
left-width: none}
```

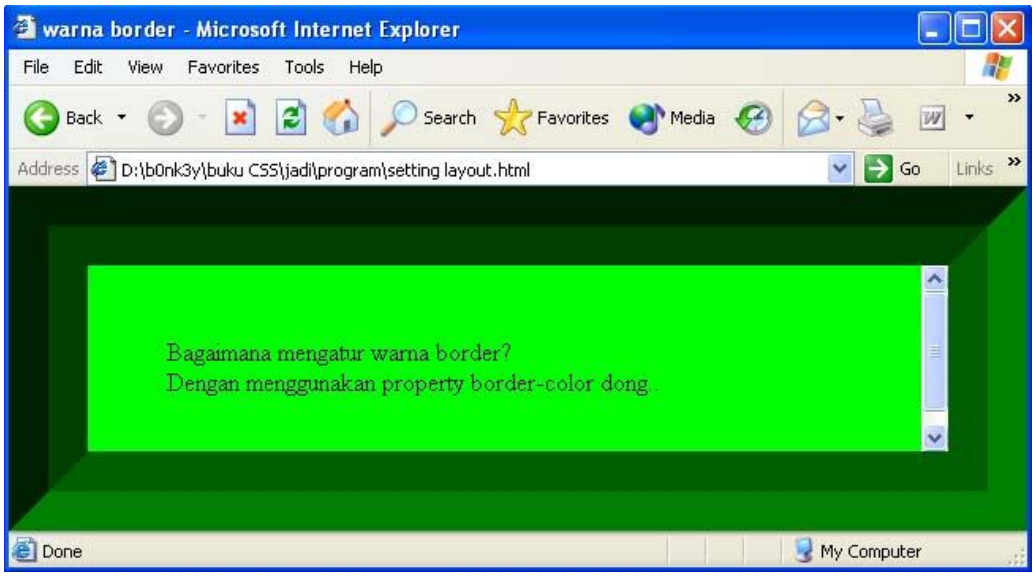
Atau dapat juga dengan menentukan lebar keseluruhan border dengan menggunakan salah satu kata bantu yang ada. Seperti contoh :

```
body {border-width: thick}
```

Dalam CSS kita juga dapat mengatur warna border halaman web yang dibuat dengan menggunakan propertiy “border-color”. Seperti terlihat dalam contoh :

```
<html>
<head><title>warna border </title>
<style type ="text/css">
body {border-color: green; border-width: 50px;
background:lime}
</style>
</head>
<body>
Bagaimana mengatur warna border?<br>
Dengan menggunakan property border-color dong..
</div>
</body>
</html>
```

Dalam contoh diatas, warna border akan menjadi kuning dengan ketebalan border keseluruhan sebesar 10 pixel dan warna background lime. Sayangnya property ini tidak didukung dalam browser Netscape namun jika anda menggunakan browser Internet Explorer, maka akan terlihat seperti dalam gambar berikut :



Gambar 7.4. Penggunaan atribut warna border dilihat dalam Browser Internet Explorer

c. Padding

Padding adalah pemberian jarak atau batas antara suatu objek dengan objek-objek yang berada di dalam objek tersebut. Padding dalam CSS dapat diatur menggunakan property pengaturan padding seperti “padding-top”, “padding-bottom”, “padding-right”, dan “padding-left”. Seperti contoh berikut :

```
<html>
<head><title>mengatur padding</title>
<style type ="text/css">
body { border-color: green; border-width: 50px;
background:lime; background:lime; padding-top:30;
padding-bottom:30; padding-right:40; padding-
left:40}
</style>
</head>
<body>
Bagaimana mengatur padding?<br>
Dengan menggunakan property padding dong..
</body>
</html>
```

Jika dilihat dalam browser maka akan terlihat bahwa halaman tersebut memiliki padding atas sebesar 30 point, padding bawah sebesar 30 point, dan padding kanan sebesar 40 point seperti pada gambar :



Gambar 7.5. Penggunaan atribut padding dilihat dalam Browser

3. POSISI

Dalam pembuatan layout dengan menggunakan CSS anda dapat menentukan posisi sebuah elemen dengan menggunakan fungsi dari position. Adapun property position tersebut berupa absolute, relative, dan static atau fixed. Nilai dari static merupakan nilai default dari seluruh elemen yang terlihat dalam HTML. Relative hampir seperti nilai static namun elemen-elemennya dapat di set dari posisi original dengan menggunakan nilai top, right, bottom, dan left. Posisi absolute akan menarik sebuah elemen keluar dari alur normal dalam html dan dapat diletakkan dimanapun dalam halaman web dengan menggunakan nilai top, right, bottom dan left. Property fixed seperti pada absolute, namun secara teori elemen yang menggunakan property ini akan tetap berada pada posisi tersebut walau halaman tersebut kita scroll ke atas-bawah. Namun untuk property fixed ini walau terlihat bekerja baik dalam browser seperti Mozilla dan Opera, namun tidak bekerja sama sekali pada Internet Explorer.

Penerapannya di dalam dokumen web anda dihitung dari posisi atau koordinat samping kiri atas (absolute) atau mengkalkulasikan dari posisi atau koordinat dimana elemen tersebut dimasukkan (relative).

- **Posisi Absolut**

  Jika anda menentukan posisi ke dalam posisi absolute maka akan dikalkulasikan dari sebelah ujung kiri atas halaman, kecuali layer tersebut didefinisikan didalam layer yang lain, dimana dalam kasus ini akan dikalkulasikan dari sebelah ujung kiri atas dari layer utama.

```
<div style="position:absolute; top:10; left:50;
font-family:Verdana; font-size:35pt; color:green">
```

- **Posisi Relatif**
  Jika anda menentukan posisi menjadi relative, maka akan di posisikan relative kedalam tag yang diusung dalam stylenya. Dan jika anda menambahkan sebuah layer dengan posisi relative di tengah halaman, maka posisi akan dikalkulasikan tepat dari tempat yang sama dalam tengah halaman dimana ia ditambahkan.

```
<div style="position:relative; top:-50; left:5;
color:red; font-size:80px; z-index:4">
```

4. MENENTUKAN POSISI

Ketika property position mengindikasikan sumber luar dari koordinat system, maka property left dan top akan menentukan posisi yang tetap pada layer, yaitu posisi sebelah kiri halaman (left) dan posisi sebelah kanan halaman (right). Anda dapat memasukkan baik nilai positif dan negative untuk property tersebut, jadi sangat mungkin untuk menempatkan isi atau content menjadii lebih tinggi dan lebih ke kiri dalam halaman dibandingkan posisi logical dalam kode atau script yang dibuat dengan menggunakan html dimana layer tersebut didefinisikan. Sehingga pada bawah kode html anda akan dapat memasukkan kode untuk layer yang diposisikan pada atas halaman yang dihasilkan.

Baik property left dan top dapat diubah secara dinamis dengan menggunakan bahasa Javascript. Ini artinya bahwa anda akan dapat menggerakkan objek pada layar bahkan setelah halaman web tersebut selesai di load. Kegunaan lain dari layer yang dibuat dengan CSS adalah berupa menu yang keluar ketika mouse bergerak melewati sebuah link yang telah dberikan script ini.

5. POSISI Z-INDEX

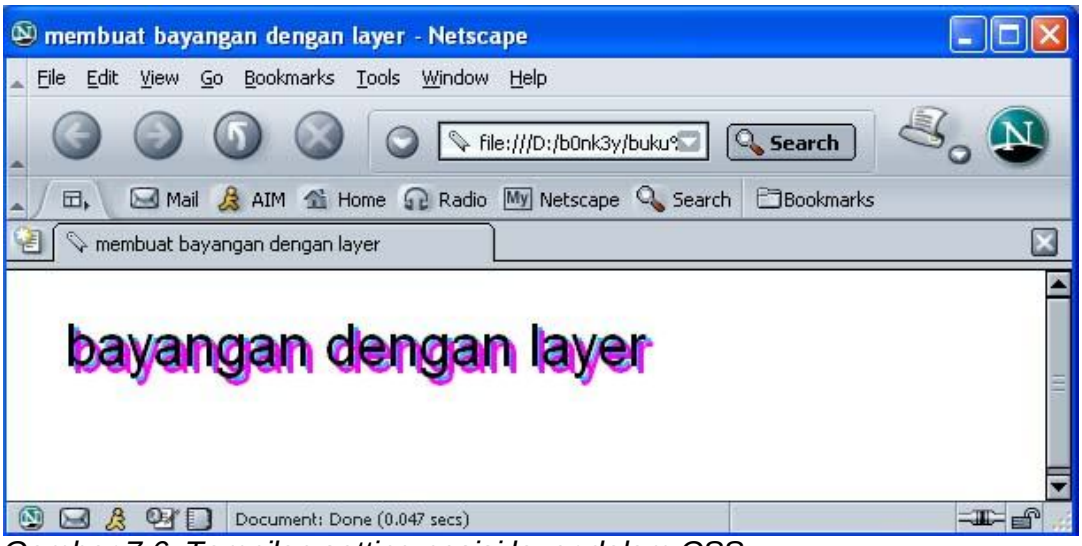
Properti “z-index”, berfungsi untuk meletakkan “stacking order” dari layar-layar yang diletakkan antara satu dengan yang lain. Nilai z – index yang tertinggi akan menekan elemen yang terendah, dan nilai yang terendah akan secara pasti mengikuti aturan di atasnya. Sebagai contoh adalah sebuah permainan kartu dimana jika sebuah as sekop jika berada pada dasar maka dapat dikatakan memiliki z-index:1;. Jika sebuah ratu hati berada pada atas maka dapat dikatakan memiliki z-index:52;. Hebatnya lagi bahwa z-index ini dapat dibuat menjadi lebih baik lagi menggunakan Javascript, namun tentu saja kita juga harus menguasai bahasa ini.

Anda dapat membuat beberapa halaman di atas tiap halaman lainnya seluruhnya dalam sebuah halaman. Ketika user memilih sebuah link, maka akan secara otomatis memindahkan layer dengan info yang kita inginkan diatasnya daripada membuka sebuah halaman baru. Teknik membuat efek seperti ini saat ini banyak mengacu pada DHTML (Dinamic HTML atau gabungan antara Javascript dan CSS) yang lebih mengarah pada tujuan tersebut.

Salah satu penggunaan properti “z-index” adalah untuk menciptakan bayangan (drop shadow) – contoh berikut adalah penerapan z-index untuk membuat bayangan :

```
<html>
<head><title>membuat bayangan dengan layer</title>
<style>
.posisi1 {position:relative; top:15; left:25; z-
index:10; color:#00000; font-family: Arial; font-
size:25pt}
.posisi2 {position:relative; top:-22; left:28; z-
index:6; color:#00ccff; font-family:Arial; font-
size:25pt}
.posisi3 {position:relative; top:-59; left:29; z-
index:6; color:#ff00ff; font-family:Arial; font-
size:25pt}:
</style>
</head>
<body>
<div class="posisi1">bayangan dengan layer</DIV>
<div class="posisi2">bayangan dengan layer</DIV>
<div class="posisi3">bayangan dengan layer</DIV>
</body>
</html>
```

Dalam browser akan terlihat seperti gambar berikut :



Gambar 7.6. Tampilan setting posisi layer dalam CSS

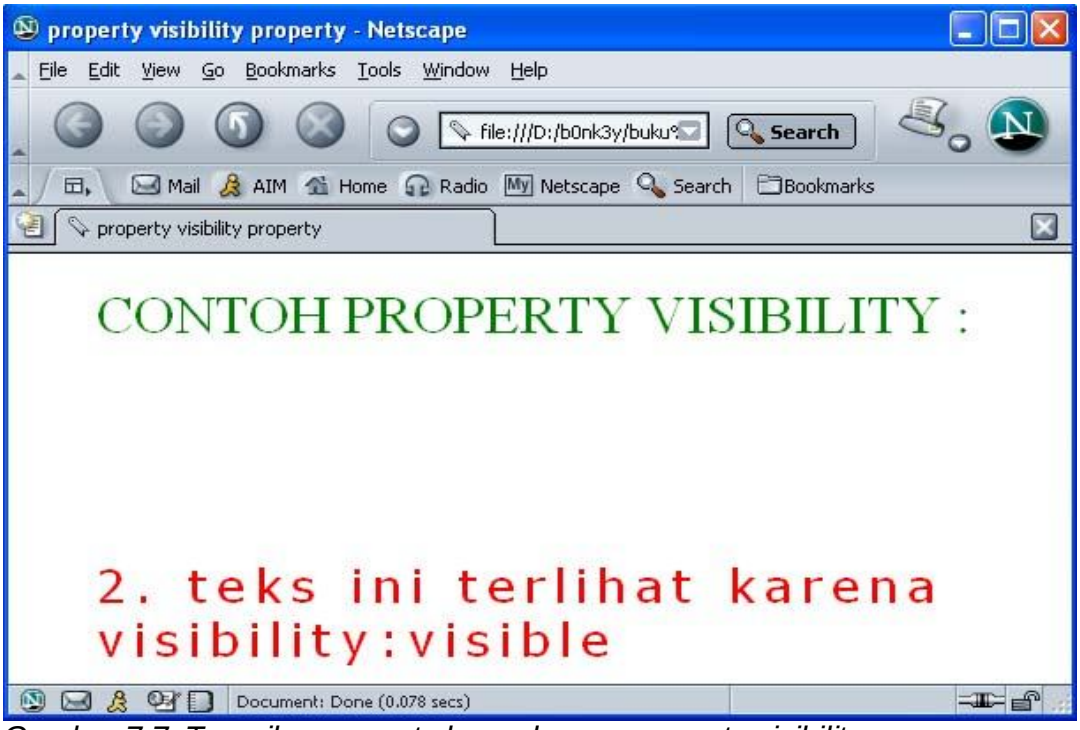
6. VISIBILITY PROPERTY

Property terakhir adalah visibility property yang memungkinkan anda untuk membuat layer yang tidak terlihat untuk menambahkan berbagai bentuk pop up menu dan berbagai efek lain dalam halaman anda. Nilai dari property ini adalah “visible” untuk memeperlihatkan layer dan “hidden” untuk menyembunyikan layer.

Dengan dynamic HTML maka akan memudahkan anda mengubah penampakan dari layer menurut masing-masing event. Kegunaan yang paling umum adalah dalam membuat menu yang dapat keluar (pop up) seperti menu Start dalam Windows dimana submenu-submenu yang akan keluar (pop up) tersebut harus dibuat sebagai layer yang tidak terlihat (invisible layer). Kemudian, ketika sebuah mouse melintas diatas sebuah link maka layer yang kita tentukan akan terlihat. Namun sekali lagi bahwa browser Netscape agaknya kurang mendukung dengan teknologi CSS ini.
Berikut adalah contoh pembuatan dokumen dengan properti “visibility”, yang menerima nilai “visible” dan “hidden” :

```
<html>
<head><title>property visibility property</title>
</head>
<body>
<div style="position: absolute; top:10; left:50; font-
family:Verdana;
font-size:30pt; color:green; font-variant:small-caps">
contoh property visibility :
</div>
<div style="position:absolute; top:180; left:50;
letter-spacing:5px;
font-family:Verdana; font-size:20pt;
color:red;visibility:hidden">
1. teks ini tidak terlihat karena visibility:hidden
</div>
<div style="position:absolute; top:180; left:50;
letter-spacing:5px;
font-family:Verdana; font-size:20pt; color:red;
visibility:visible">
2. teks ini terlihat karena visibility:visible
</div>
</body>
</html>
```

Dalam browser akan terlihat seperti gambar berikut :



Gambar 7.7. Tampilan property layer dengan property visibility

7. FLOAT

Seperti artinya yaitu mengapung, maka elemen yang akan di buat akan dibuat ke kanan atau kiri dari sebuah garis dengan isi yang melingkupinya. Floating digunakan untuk memposisikan elemen yang lebih kecil dalam sebuah halaman. Seperti pada contoh berikut :

```
#navigasi {float: left; width: 10px;}
#navigasi2 {float: right; width: 10px;}
#isi {margin: 0 10px;}
```

Jika anda tidak menginginkan elemen berikutnya mengelilingi objek yang di floating, maka anda dapat menerapkan property clear.

clear: left (akan menghilangkan elemen float kiri)
clear: right (akan menghilangkan elemen float kanan) dan
clear: both (akan menghilangkan elemen float kiri dan kanan)

Sebagai contoh jika anda membuat sebuah footer dalam halaman anda, anda dapat menambahkan sebuah id 'footer' dan menambahkan :

```
#footer {clear: both;}
```

Contoh :

```
<html>
<head><title>belajar float</title>
<style>
```

```
#navigasi {float: left; width: 10px;}
#navigasi2 {float: right; width: 10px;}
#footer {clear: both;}
</style>
</head>
<body>
<div id=navigasi>float kiri</div>
<div id=navigasi2><p>float kanan</div>
<div id=footer>clear foat</div>
</body>
</html>
```

Jika dilihat dalam browser akan terlihat seperti pada gambar berikut ini :



Gambar 7.8. Tampilan float dalam Browser