Membuat Aplikasi ANDROID

Untuk Pemula



Lhugita Fazry

DAFTAR ISI

DAFTAR ISI I

1. MENGENAL ANDROID STUDIO 1
1.1. INSTALASI JAVA DEVELOPMENT KIT (JDK) 1
1.2. JENDELA KERJA ANDROID STUDIO 3
1.3. ANDROID SDK MANAGER 5
1.4. ANDROID VIRTUAL DEVICE MANAGER 6
1.5. GENYMOTION 12
1.6. ANDROIDMANIFEST.XML 19
1.7. JAVA DAN XML 21

2. SEKILAS TENTANG JAVA 23
2.1. APA ITU JAVA? 23
2.2. NAMA VARIABEL 23
2.3. OBJEK DAN KELAS 24
2.4. TIPE DATA PADA JAVA 24
2.5. PEWARISAN (*INHERITANCE*) 26

3. MEMBUAT PROJEK PERTAMA 27
3.1. MEMBUAT PROJEK BARU 27
3.2. STRUKTUR PROJEK ANDROID STUDIO 33
3.3. MENJALANKAN PROJEK PADA ANDROID VIRTUAL DEVICE 34
3.4. MENJALANKAN PROJEK PADA GENYMOTION 35
3.5. MENJALANKAN PROJEK PADA PERANGKAT ANDROID 36

4. MENGENAL ACTIVITY 37
4.1. APA ITU ACTIVITY? 37
4.2. ACTIVITY LIFECYCLE 42

5. RESOURCE 45
5.1. JENIS RESOURCE 46
5.2. RESOURCE LAYOUT 46
5.3. RESOURCE TEKS 48
5.4. RESOURCE ARRAY TEKS 50
5.5. RESOURCE DRAWABLE 51
5.6. RESOURCE MENU 54

6. VIEW 58
6.1. MENGENAL VIEW DAN VIEWGROUP 58

6.2. TextView 62
6.3. EditText 64
6.4. Button 66
6.5. ImageView 69
6.6. ImageButton 70
6.7. Spinner 73
6.8. CheckBox 75
6.9. AutoCompleteTextView 77
6.10. ToggleButton 79
6.11. RadioButton 81
6.12. RadioGroup 83
6.13. ProgressBar 85

7. LAYOUT 88
7.1. Atribut Layout 88
7.2. Frame Layout 92
7.3. LinearLayout 94
7.4. RelativeLayout 97

8. DIALOG 100
8.1. Toast 100
8.2. Snack Bar 100
8.3. Alert Dialog 101
8.4. Progress Dialog 101
8.5. Activity Sebagai Dialog 102

9. MENU 103
9.1. Menu Sebagai Action 104
9.2. Popup Menu 105
9.3. Context Menu 107

10. SQLITE 110
10.1. SQLiteHelper 114
10.2. Melakukan Query Data 116
10.3. Menambah Data 116
10.4. Mengupdate Data 117

12. LIST VIEW 118
12.1. Apa itu List View? 118
12.2. Default Adapter 118

13. PROJECT: MEMBUAT APLIKASI SEKOLAHKU 120

Ebook ini tersedia gratis bagi siapa saja yang ingin belajar membuat Aplikasi Android.

Anda bisa mendapatkan versi cetak dari Ebook ini, Anda bisa mengunjungi halaman berikut:

http://bit.ly/VersiCetak

Setiap pembelian versi cetak, akan mendapatkan bonus TShirt Android Keren berikut:



Anda bisa belajar membuat Aplikasi Android secara tatap muka dengan mengunjungi halaman berikut:

www.rumahcoding.co.id

Follow akun **Instagram** dan **facebook** kami untuk mendapatkan tips dan trick seputar programming, programming news atau berdiskusi seputar Ebook ini.

@idrumahcoding

1. Mengenal Android Studio

Android Studio merupakan software pengembang yang digunakan untuk membuat sebuah aplikasi android. Pada buku ini, versi yang akan digunakan adalah Android Studio 1.5.1. Android Studio dapat digunakan pada berbagai sistem operasi komputer termasuk Windows, Linux dan Mac. Untuk menginstall Android Studio pada komputer, maka kita harus menginstall Java Development Kit (JDK) terlebih dahulu.

1.1. Instalasi Java Development Kit (JDK)

Pada buku ini, versi JDK yang akan kita gunakan adalah JDK 7. JDK7 tersedia gratis pada website *Oracle*. Anda dapat mengunduh file instalasi JDK7 melalui tautan berikut:

www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Pada halaman unduh JDK7, anda dapat melihat beberapa pilihan file untuk anda download. Pilihlah salah satu file yang sesuai dengan sistem operasi yang anda gunakan. Sebelum itu, jangan lupa untuk menyetujui lisensi dengan menekat button *Accept License Agreement.*

Java SE Development Kit 7u79

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ◉ Decline License Agreement

Product / File Description	File Size	Download
Linux x86	130.4 MB	jdk-7u79-linux-i586.rpm
Linux x86	147.6 MB	jdk-7u79-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u79-linux-x64.rpm
Linux x64	146.4 MB	jdk-7u79-linux-x64.tar.gz
Mac OS X x64	196.89 MB	jdk-7u79-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.79 MB	jdk-7u79-solaris-i586.tar.Z
Solaris x86	96.66 MB	jdk-7u79-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.67 MB	jdk-7u79-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u79-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140 MB	jdk-7u79-solaris-sparc.tar.Z
Solaris SPARC	99.4 MB	jdk-7u79-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24 MB	jdk-7u79-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.4 MB	jdk-7u79-solaris-sparcv9.tar.gz
Windows x86	138.31 MB	jdk-7u79-windows-i586.exe
Windows x64	140.06 MB	jdk-7u79-windows-x64.exe

Setelah proses unduhan selesai, kemudian lanjutkan dengan proses instalasi sesuai dengan sistem operasi pada komputer yang anda miliki. Berikut adalah tutorial proses instalasi JDK7 pada komputer dengan sistem operasi Mac OS.

1.2. Instalasi Android Studio

Android Studio tersedia secara gratis pada website resmi android. Anda dapat mengunjungi tautan berikut ini untuk mengunduh file instalasi Android Studio.

developer.android.com/sdk/index.html

Berikut adalah tampilan dari halaman unduh Android Studio. Untuk mengunduh, klik tombol *Download Android Studio* yang berwarna hijau.

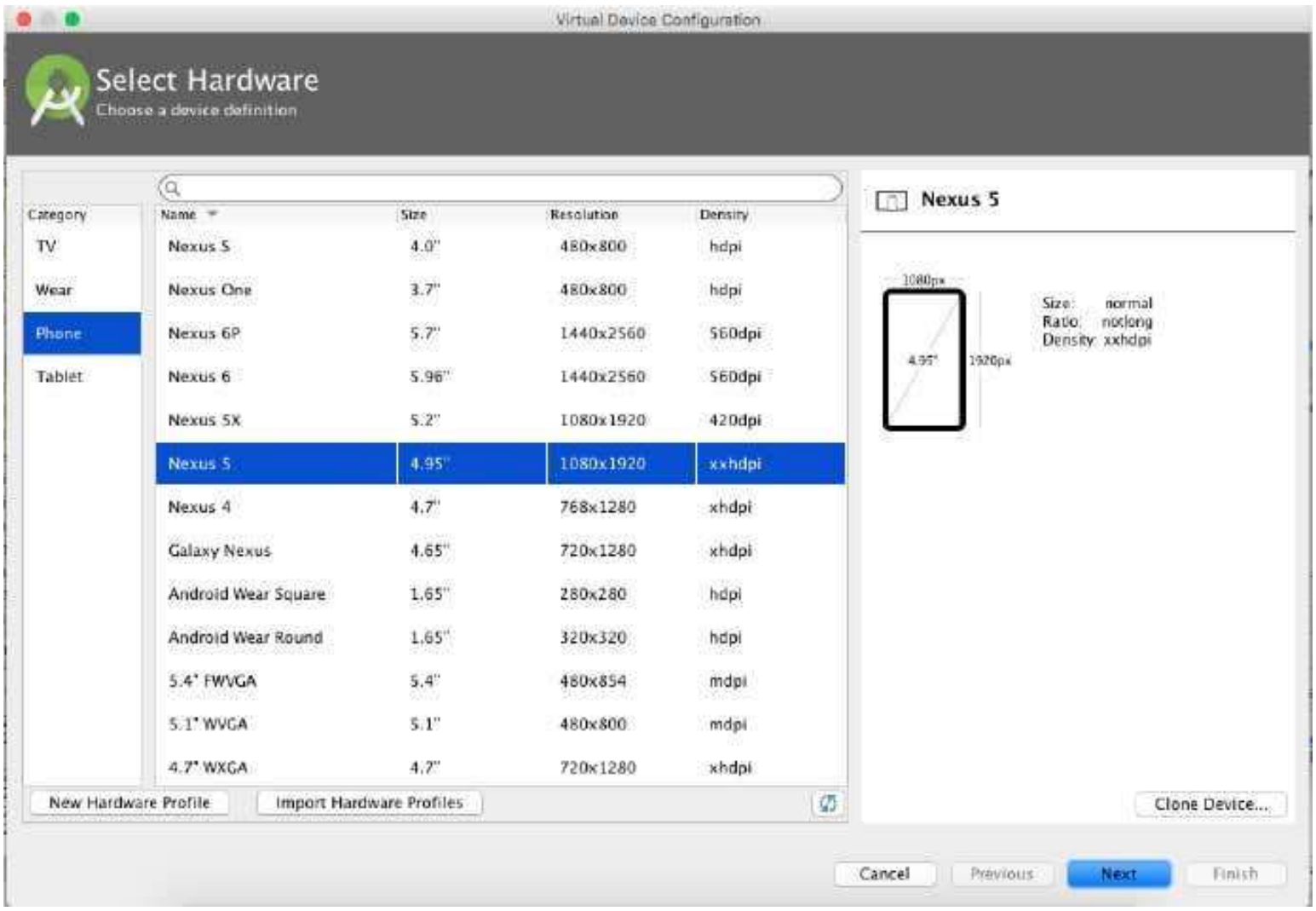


Setelah file instalasi berhasil diunduh dengan baik. Maka lanjutkan dengan proses instalasi dengan melakukan klik ganda pada file instalasi tersebut.

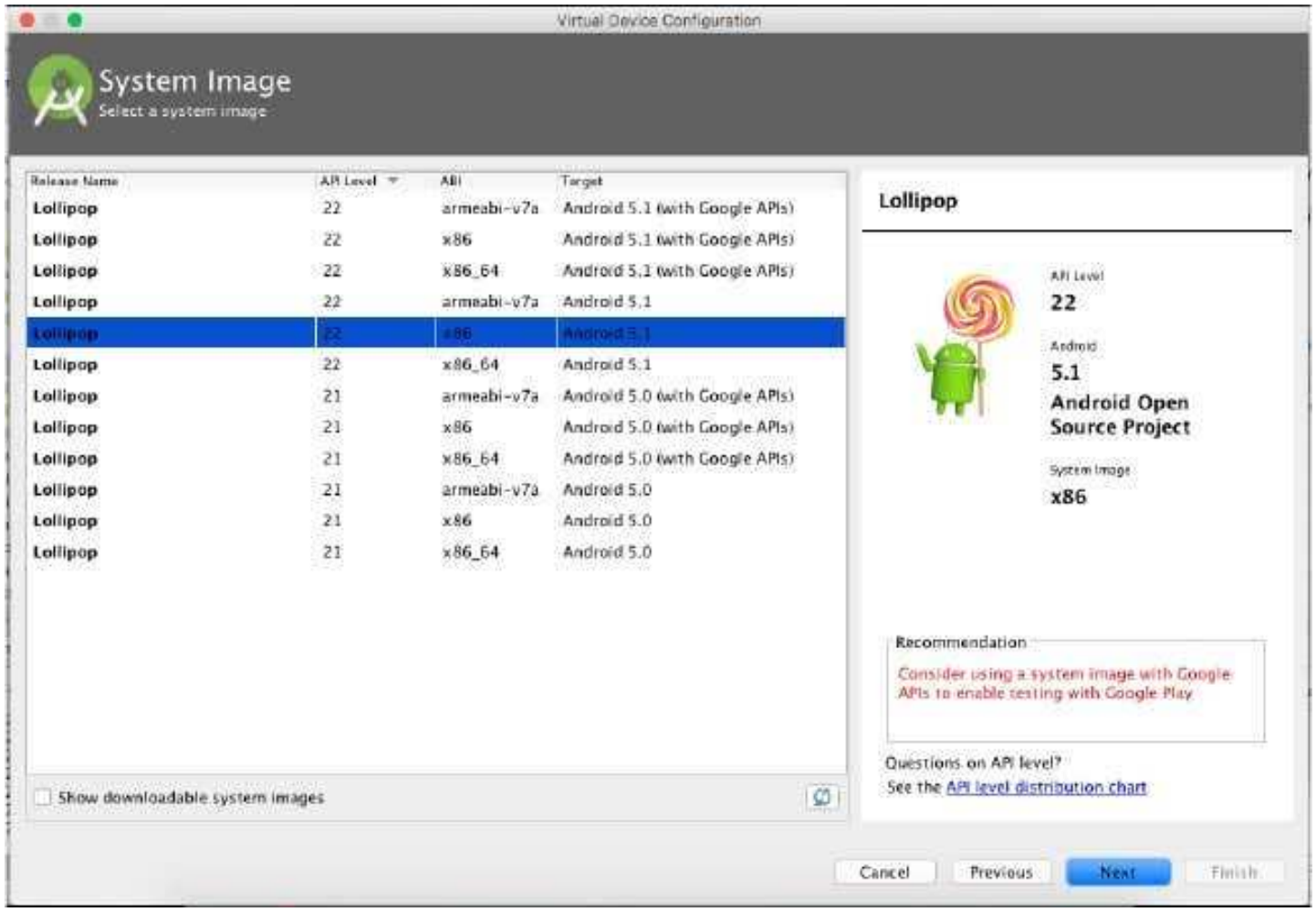
1.2. Jendela Kerja Android Studio

Pada sub bab ini, kita akan membahas mengenai beberapa fungsi dasar dari Android Studio yang akan membantu kita dalam memahami bagaimana cara menggunakan software Android Studio. Tampilan standar dari Android Studio tertera seperti gambar di bawah ini.

@idrumahcoding



3. Pilih kategori dan jenis perangkat virtual kemudian tekan tombol *Next* maka akan muncul jendela pemilihan versi android untuk perangkat virtual yang akan kita gunakan.



4. Pilih versi android yang akan anda pakai kemudian klik tombol *Next* maka akan muncul jendela verifikasi konfigurasi. Anda dapat melakukan konfigurasi tambahan pada jendela tersebut. Kemudian klik tombol *Finish* maka proses pembuatan perangkat virtual telah selesai. Perangkat virtual yang baru saja kita buat sudah bisa kita pergunakan untuk pengetesan aplikasi.

@idrumahcoding



5. Untuk memulai mengetes aplikasi, anda perlu menjalankan perangkat virtual yang baru saja anda buat dengan meng-klik tombol  pada jendela *Android Virtual Device Manager*. Silahkan tunggu beberapa saat, perangkat virtual akan segera aktif.


5554:Nexus_5_API_22
9:38
Google

1.5. GenyMotion

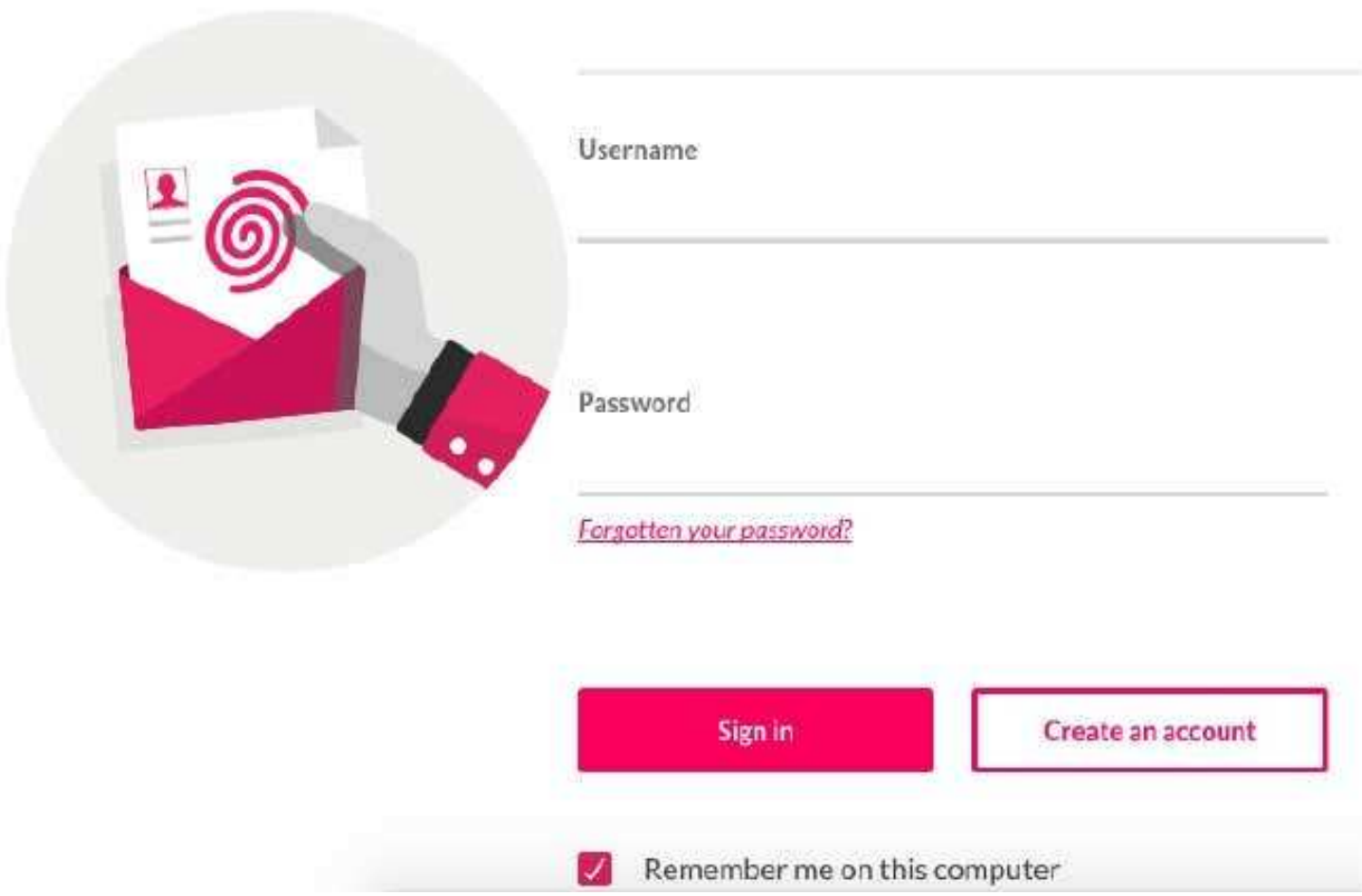
Walaupun kita bisa menggunakan perangkat virtual pada Android Studio dengan mudah, akan tetapi perangkat virtual tersebut sangat lambat dibandingkan dengan ponsel sungguhan. Kelambatan perang virtual tersebut terjadi baik pada saat perangkat dinyalakan(*booting*) maupun ketika dioperasikan. Kurangnya performa perangkat virtual ini membuat proses tes aplikasi menjadi terhambat.

Sebagai alternatif, kita bisa menggunakan GenyMotion sebagai ganti perangkat virtual bawaan Android Studio. GenyMotion manawarkan solusi yang lebih baik dari perangkat virtual yang sudah ada di Android Studio. GenyMotion bisa digunakan secara gratis untuk keperluan pribadi sedangkan untuk keperluan komersial anda bisa menggunakannya dengan harga sekitar $136 per tahun.

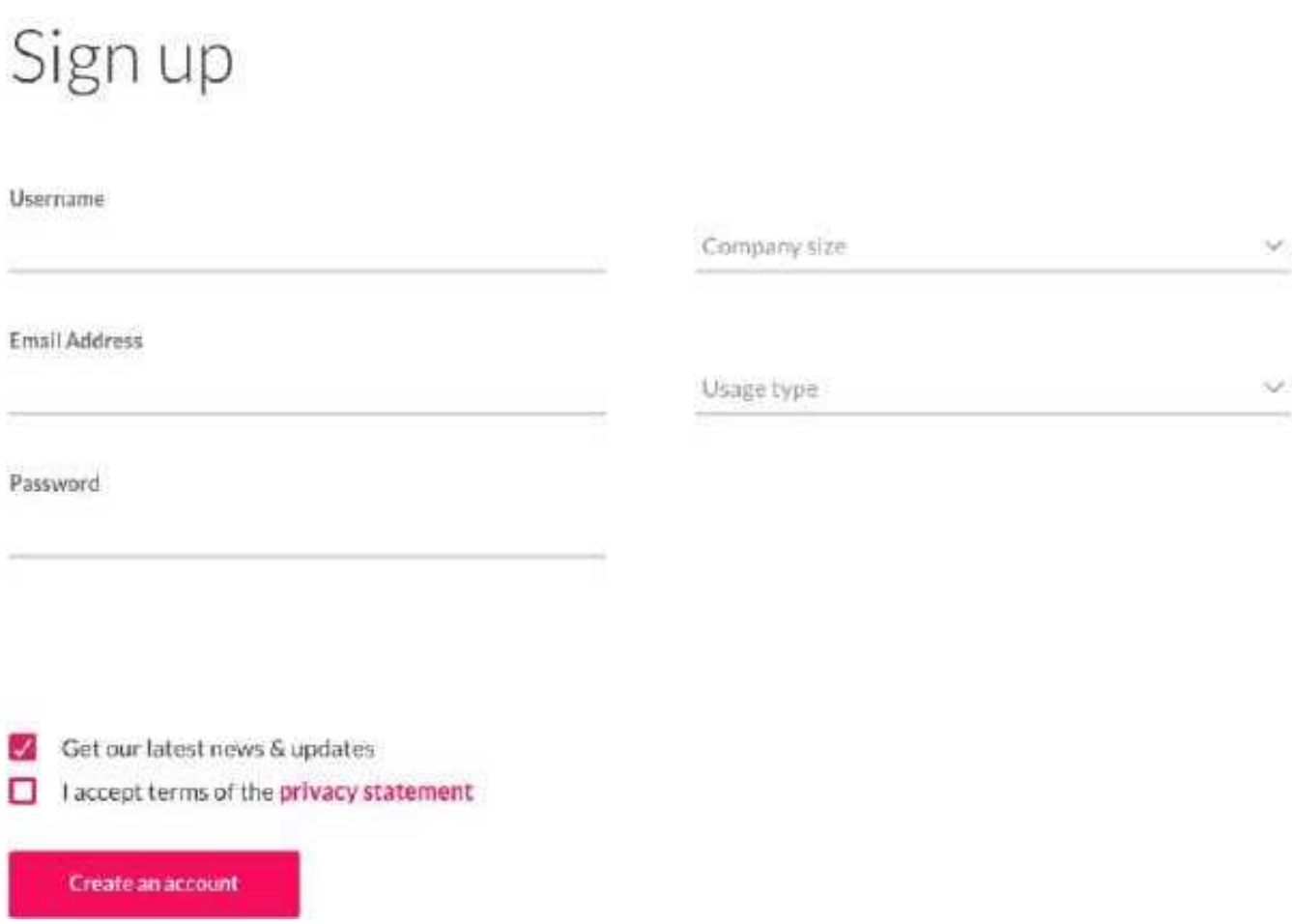
Untuk mendownload GenyMotion, silahkan mengunjungi alamat website berikut:

www.genymotion.com/download

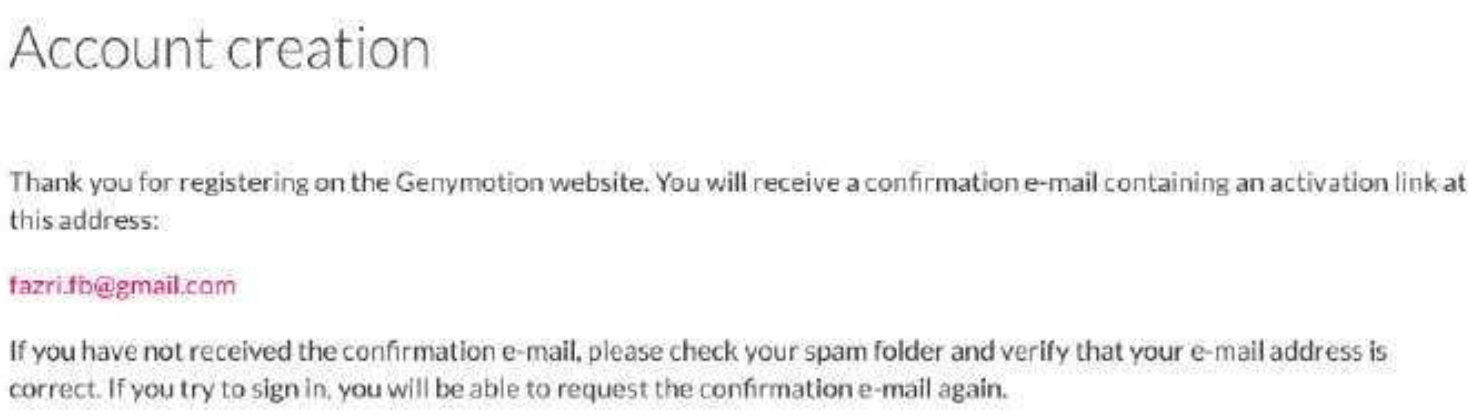
Anda diharuskan untuk membuat akun sebelum dapat mengunduh GenyMotion. Untuk membuat akun baru di GenyMotion, silahkan klik tombol *Create an account*.



Silahkan isi data-data yang diperlukan pada formulir *Sign up*. Isilah *username*, *email address* dan *password* dengan data yang benar. Untuk field *company size* bisa diisi dengan *personal use* sedangkan field *usage type* bisa diisi *development*.

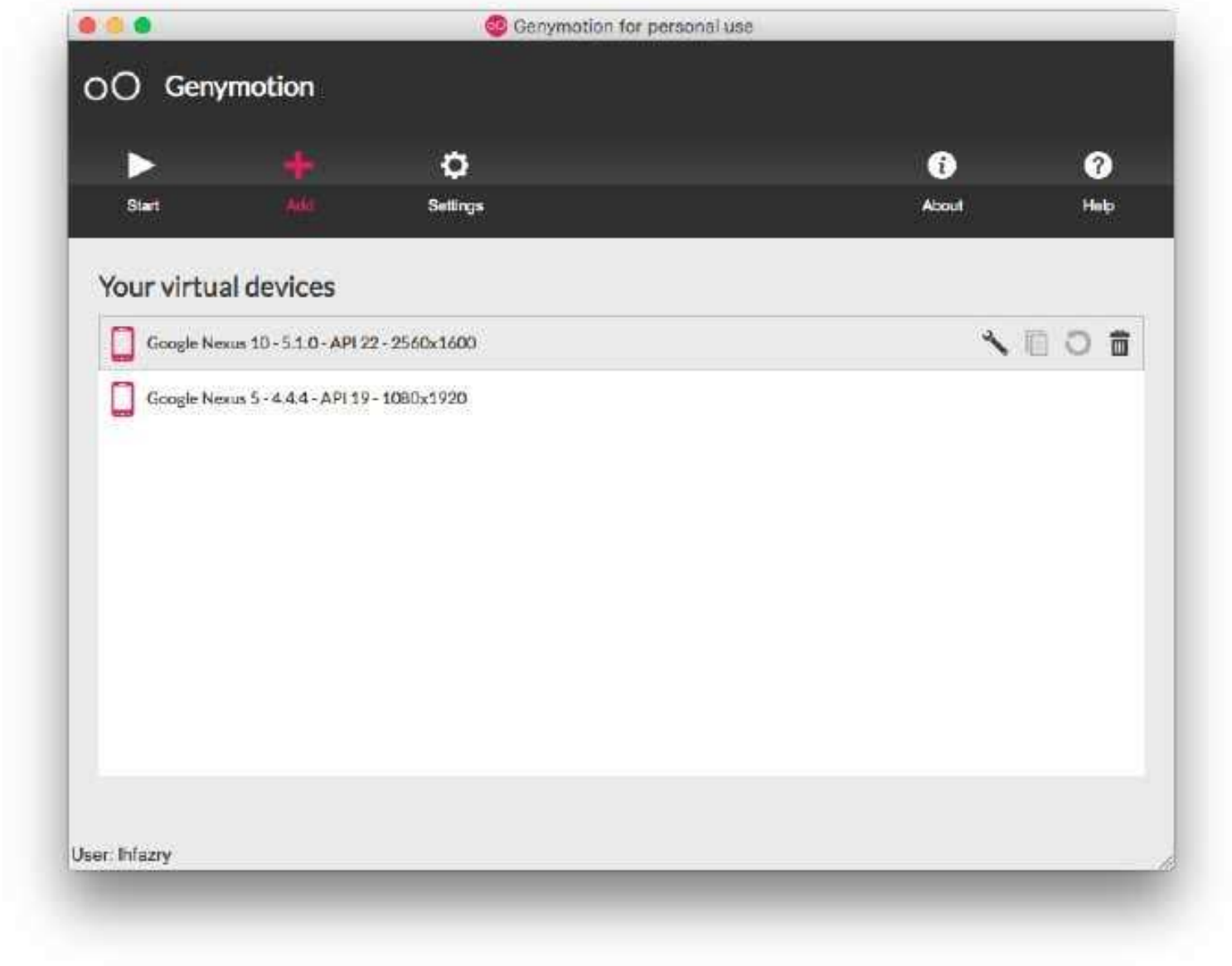


Setelah semua *field* diisi dengan baik, jangan lupa mencentang kotak *I accept terms of the privacy statement* kemudian dilanjutkan dengan meng-klik tombol *Create an account*. Maka akan muncul halaman konfirmasi berikut:

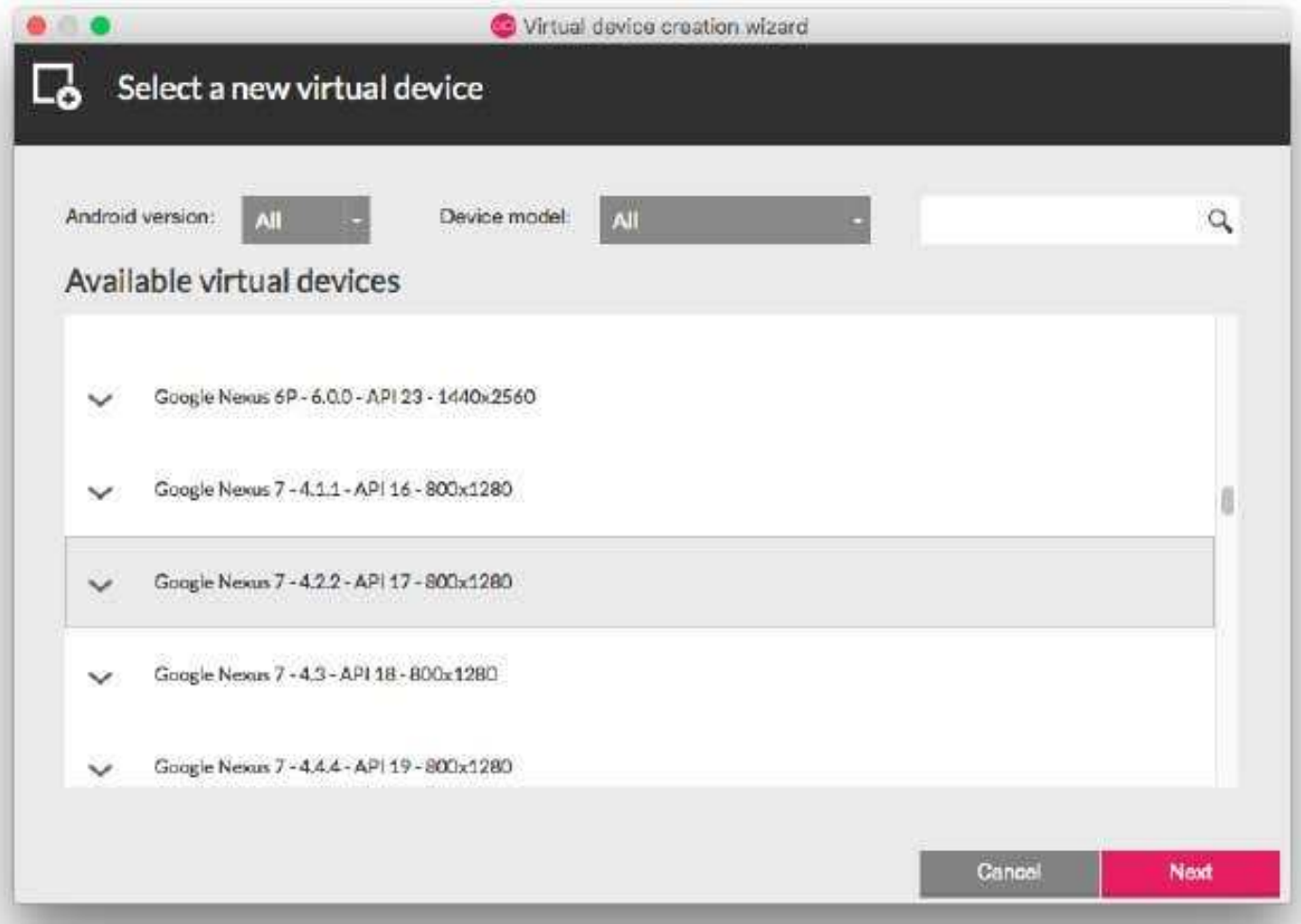


Pada tahap ini, pembuatan akun GenyMotion telah berhasil. GenyMotion telah mengirimkan sebuah email yang berisi tautan untuk mengakitvasi akun yang baru saja anda buat. Silahkan cek email anda, email yang anda terima kira-kira seperti gambar di bawah.

Setelah proses instalasi selesai kemudian jalankan GenyMotion maka akan muncul jendela utama GenyMotion seperti tampak pada gambar di bawah.

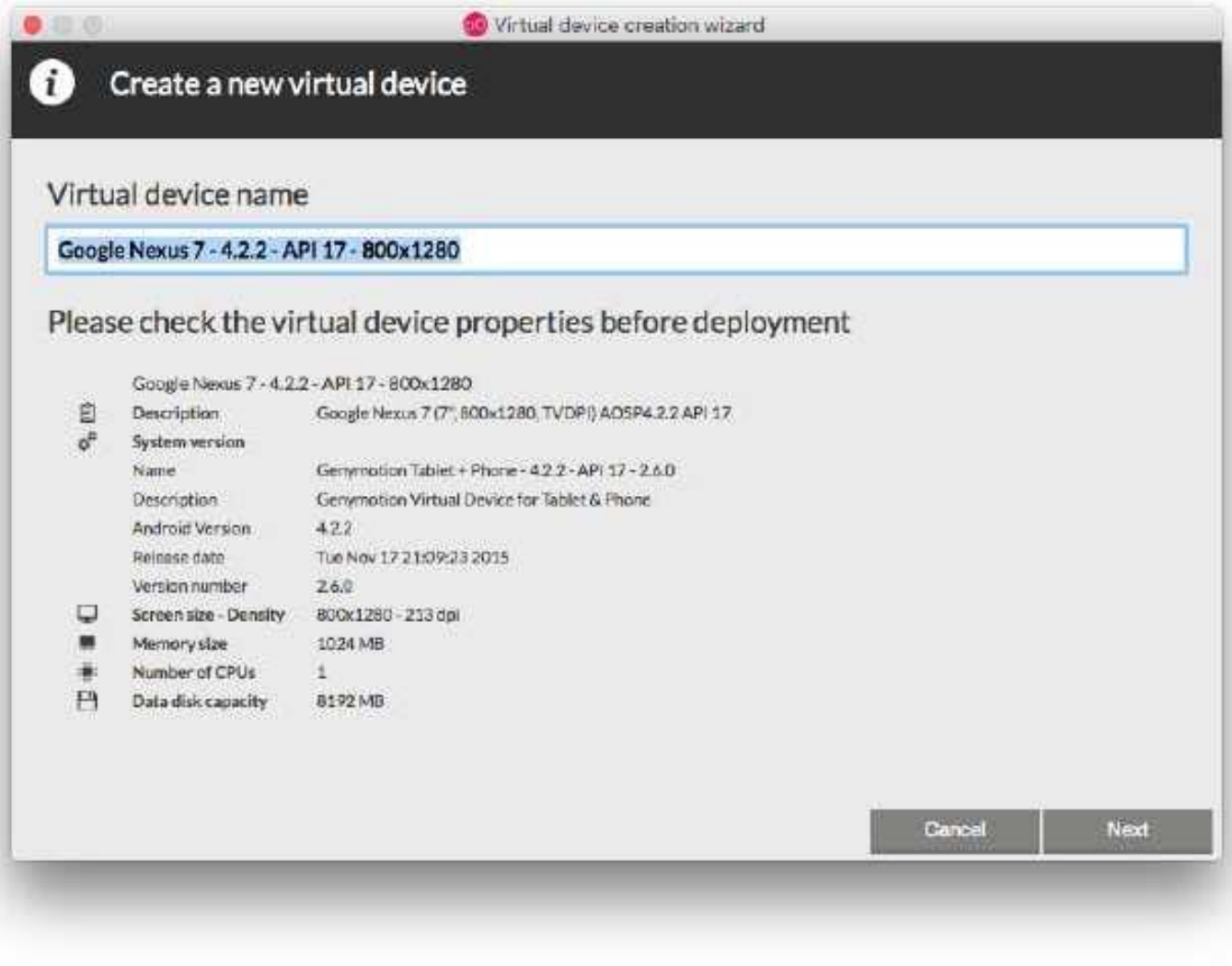


Sama halnya dengan *Android Virtual Device*, di GenyMotion anda juga harus menambahkan perangkat virtual baru sebelum bisa melakukan tes. Silahkan klik tombol *Add* untuk menambah perangkat virtual baru pada GenyMotion. Maka akan muncul jendela *Virtual device creation wizard*.



Silahkan pilih jenis perangkat yang ingin anda buat lalu klik tombol *Next*. Kemudian muncul jendela selanjutnya, lanjutkan saja dengan meng-klik tombol *Next*.

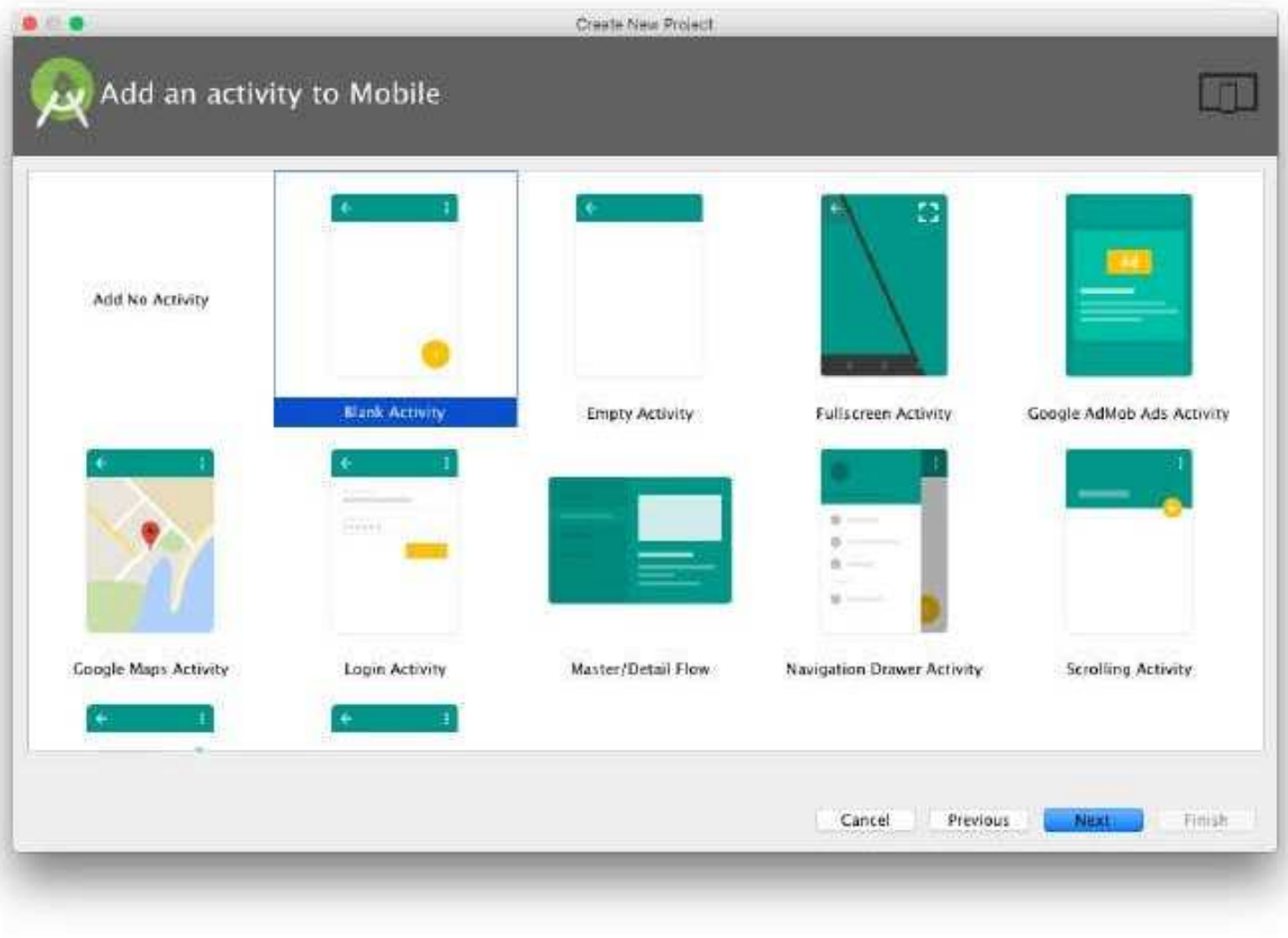
@idrumahcoding



GenyMotion akan mengunduh beberapa file yang dibutuhkan setelah itu proses pembuatan perangkat virtual pada GenyMotion telah selesai.

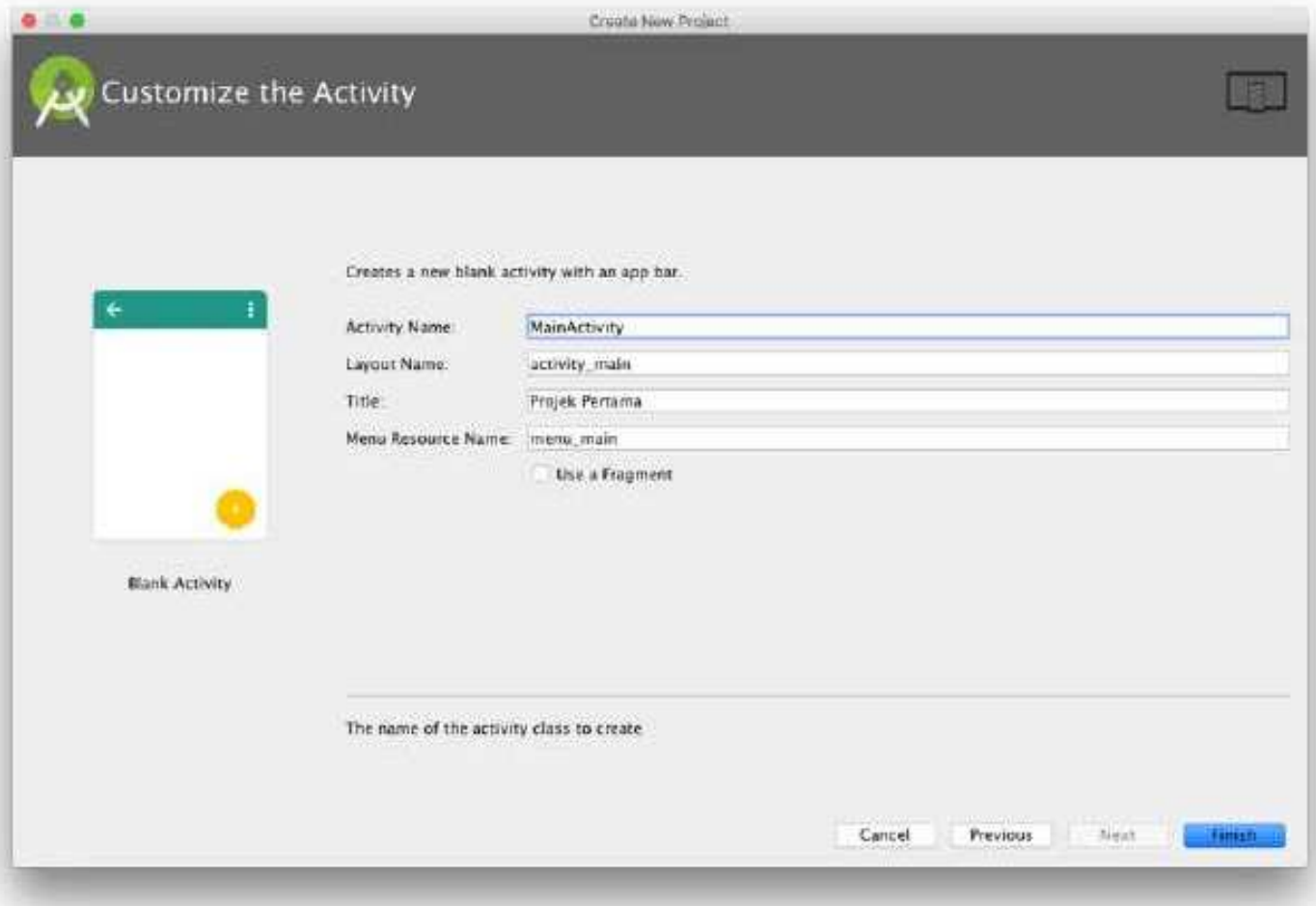
SDK kita bisa mengatur aplikasi hanya dapat berjalan jika ponsel target terinstall android dengan versi minimal sesuai dengan versi yang kita atur pada pilihan tersebut. Misal kita set *API 15 (Ice Cream Sandwich)*, maka aplikasi tersebut tidak akan bisa diinstall pada ponsel dengan android di bawah *Ice Cream Sandwich*. Selain itu, aplikasi tersebut juga tidak akan muncul di *Google Play Store* jika kita mencari melalui ponsel dengan android di bawah *Ice Cream Sandwich*.

4. Kemudian klik tombol *Next* maka akan muncul jendela *Add an activity to Mobile*.



Jendela ini memungkinkan kita untuk menambah *activity* baru ke dalam projek. Untuk penjelasan mengenai *activity* akan dijabarkan pada bab selanjutnya. Untuk sementara pilih *Blank Activity*.

5. Kemudian klik tombol *Next* maka akan muncul jendela *Customize the Activity*.



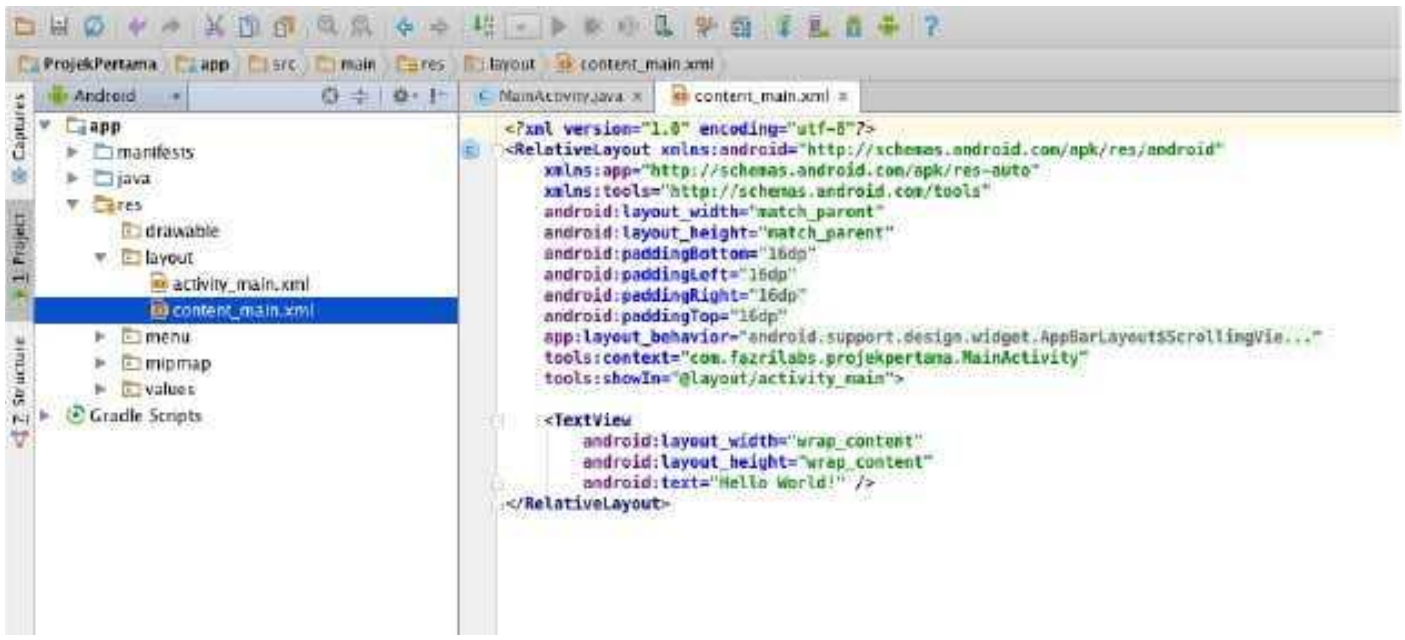
Pada jendela tersebut kita bisa mengatur beberapa konfigurasi *activity* yang akan kita buat. Ada beberapa *field* yang harus kita isi. Isilah dengan data-data berikut:

Field	Nilai	Keterangan
Activity Name	**MainActivity**	Merupakan nama *class* dari *activity* yang akan dibuat
Layout Name	**activity_main**	Merupakan nama *layout* untuk *activity* yang akan dibuat
Title	**Projek Pertama**	Merupakan judul dari *activity*
Menu Resource Name	**menu_main**	Merupakan nama file untuk *resource* dari menu

Untuk sementara pastikan *checkbox Use a Fragment* tidak dicentang.

6. Kemudian klik tombol *Finish*. Silahkan tunggu sebentar, Android Studio akan membuatkan file-file projek yang

terkoneksi dengan internet karena Android Studio akan mendownload beberapa library ke dalam projek melalui gradle. Jika proses telah selesai, maka akan muncul jendela kerja berikut.



3.2. Struktur Projek Android Studio

Ketika kita membuat sebuah projek baru, maka Android Studio akan secara otomatis membuatkan beberapa folder dan file dalam sebuah struktur yang cukup rapih. Hal tersebut dapat mempercepat kita memulai sebuah projek baru. Berikut adalah struktur sebuah projek pada Android Studio.

a. *manifests*: Folder tempat meletakan file AndroidManifest.xml
b. *java*: Folder tempat meletakkan file-file Java
c. *res:drawable*: Folder tempat meletakkan file-file gambar
d. *res:layout*: Folder tempat meletakkan file-file xml untuk layout
e. *res:menu*: Folder tempat meletakkan file-file xml untuk menu
f. *res:mipmap*: Folder tempat meletakkan file-file gambar untuk icon aplikasi
g. *res:values*: Folder tempat meletakkan file-file xml yang lain.

4. Mengenal Activity

4.1. Apa Itu Activity?

Secara sederhana, sebuah *activity* pada aplikasi android mewakili sebuah layar yang tampak dalam aplikasi tersebut. Di dalam layar tersebut bisa terdapat beberapa komponen *user interface* baik berupa *TextView*, *Button*, *EditText*, *Menu* dan komponen-komponen yang lain. Perhatikan layar utama dari sebuah aplikasi berikut.



Layar aplikasi tersebut dibentuk oleh sebuah *activity*. Dalam layar tersebut terdapat 3 buah tombol. Apabila salah satu tombol diklik, maka akan berganti ke layar kedua.



Layar kedua pada aplikasi tersebut merupakan sebuah *activity* yang lain. Jadi dapat kita simpulkan bawah sebuah aplikasi android bisa terdiri dari satu atau lebih *activity*. Keseluruhan activity secara bersama-sama membangun sebuah aplikasi yang sempurna.

Sebuah *activity* merupakan sebuah kelas yang ditulis menggunakan bahasa pemrograman Java. Berikut ini adalah contoh activity dari Projek Pertama yang telah kita buat pada bab sebelumnya. Silahkan buka projek pertama anda dan buka file *MainActivity* pada folder *java/com.fazrilabs.projekpertama*.

```
package com.fazrilabs.projekpertama;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar)
findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

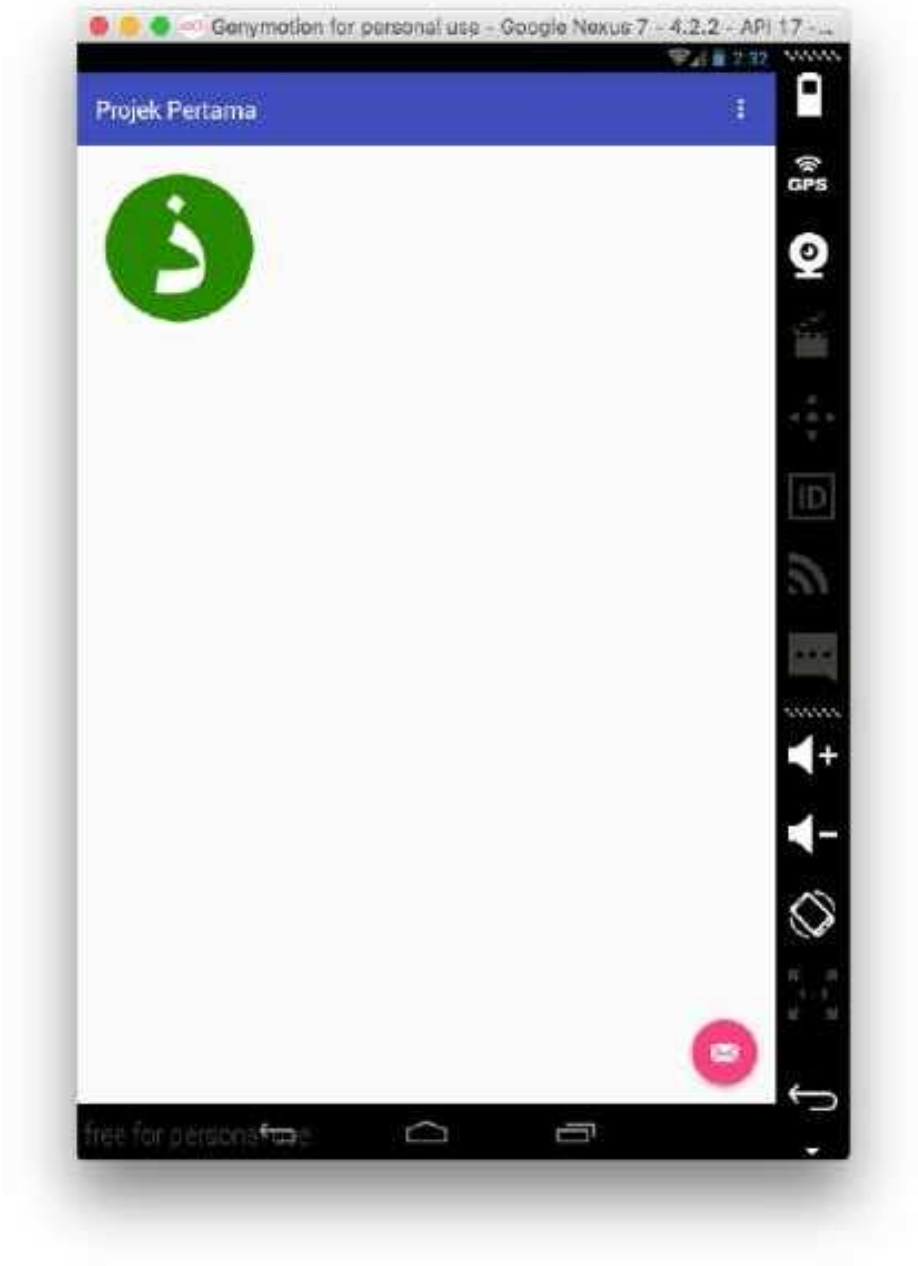
        FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own
action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action
bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/adzkar"/>
</RelativeLayout>
```

Perhatikan bahwa penggunaan resource drawable di atas dilakukan tanpa menuliskan ekstensinya, yaitu **@drawable/adzkar** bukan **@drawable/adzkar.png**. Berikut adalah pratinjau dari layout tersebut.



Walaupun ukuran layar perangkat android berbeda-beda, akan tetapi android akan secara otomatis menyesuaikan ukuran gambar dengan ukuran layar. Android memberikan keleluasan kepada pengembang aplikasi untuk menyediakan gambar berbeda untuk ukuran layar yang berbeda. Jadi untuk sebuah gambar, anda bisa menyediakan beberapa versi ukuran, android akan secara otomatis mengambil gambar yang sesuai dengan ukuran layar. Untuk keperluan tersebut, anda harus membuat folder tambahan di dalam folder *res,* yaitu:

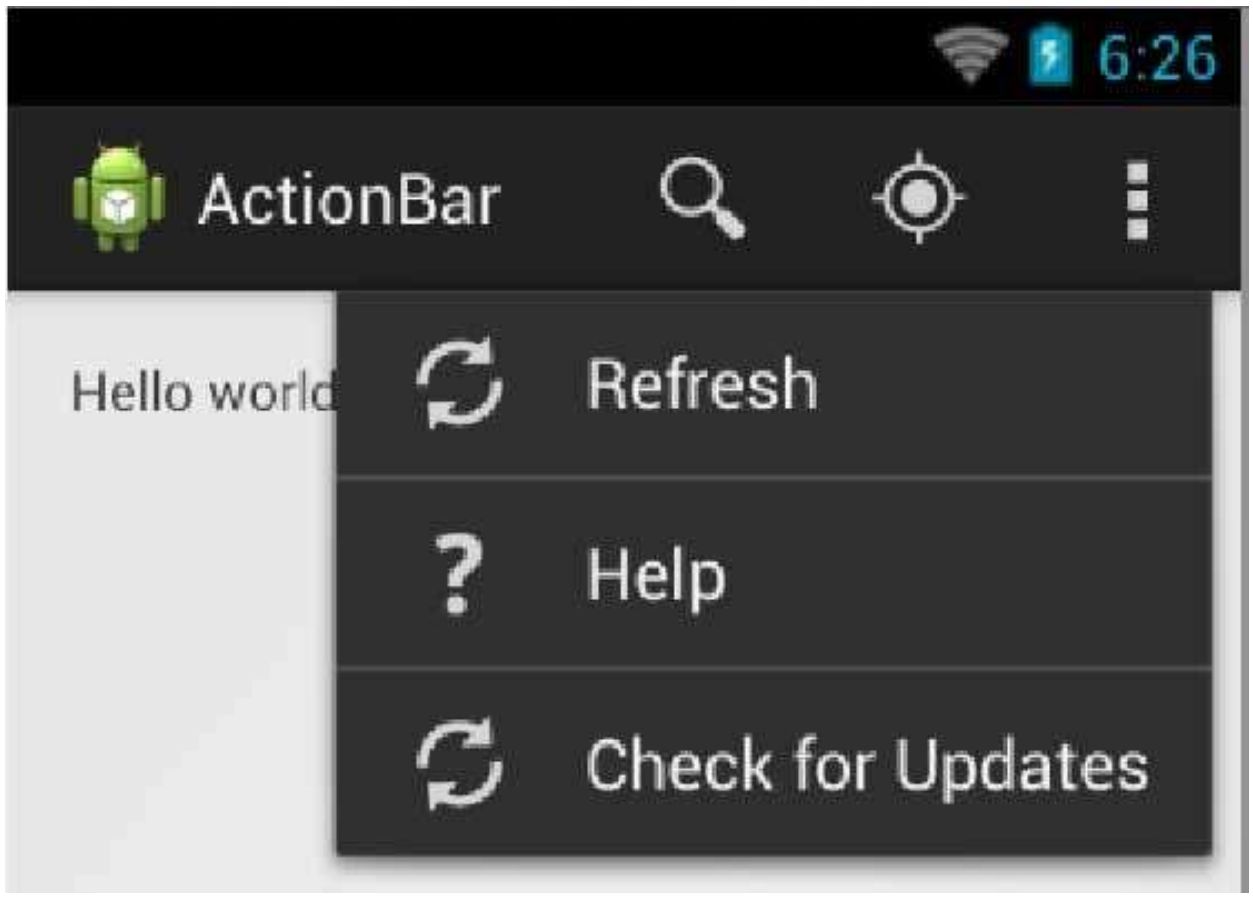
1. *drawable-ldpi* (low)
2. *drawable-mdpi* (medium)
3. *drawable-hdpi* (high)
4. *drawable-xhdpi* (extra-high)

Untuk mempermudah anda memahami konsep di atas, perhatikan contoh variasi ukuran gambar dari *adzkar.png*.

mdi	hdpi	xhdpi	xxhdpi
			
48 x 48	72 x 72	96 x 96	144 x 144

5.6. Resource Menu

Salah satu aspek penilaian pengguna aplikasi terhadap kualitas sebuah aplikasi adalah *user friendly*. Dengan kata lain, pengguna akan lebih menyukai aplikasi yang mudah ketika digunakan. Salah satu faktor terpenting agar aplikasi yang anda buat mudah digunakan adalah dengan membuat menu. Menu merupakan bagian penting dari aplikasi android. Menu dapat mempermudah user dalam melakukan navigasi ke berbagai halaman dalam aplikasi. Gambar di bawah menunjukkan sebuah menu.



Membuat menu dalam aplikasi android sangatlah mudah. Anda cukup mendefinisikan menu tersebut sebagai sebuah file xml yang diletakkan di dalam folder */res/menu*. Berikut adalah contoh sebuah resource menu dengan nama file *menu_main.xml*.

```
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

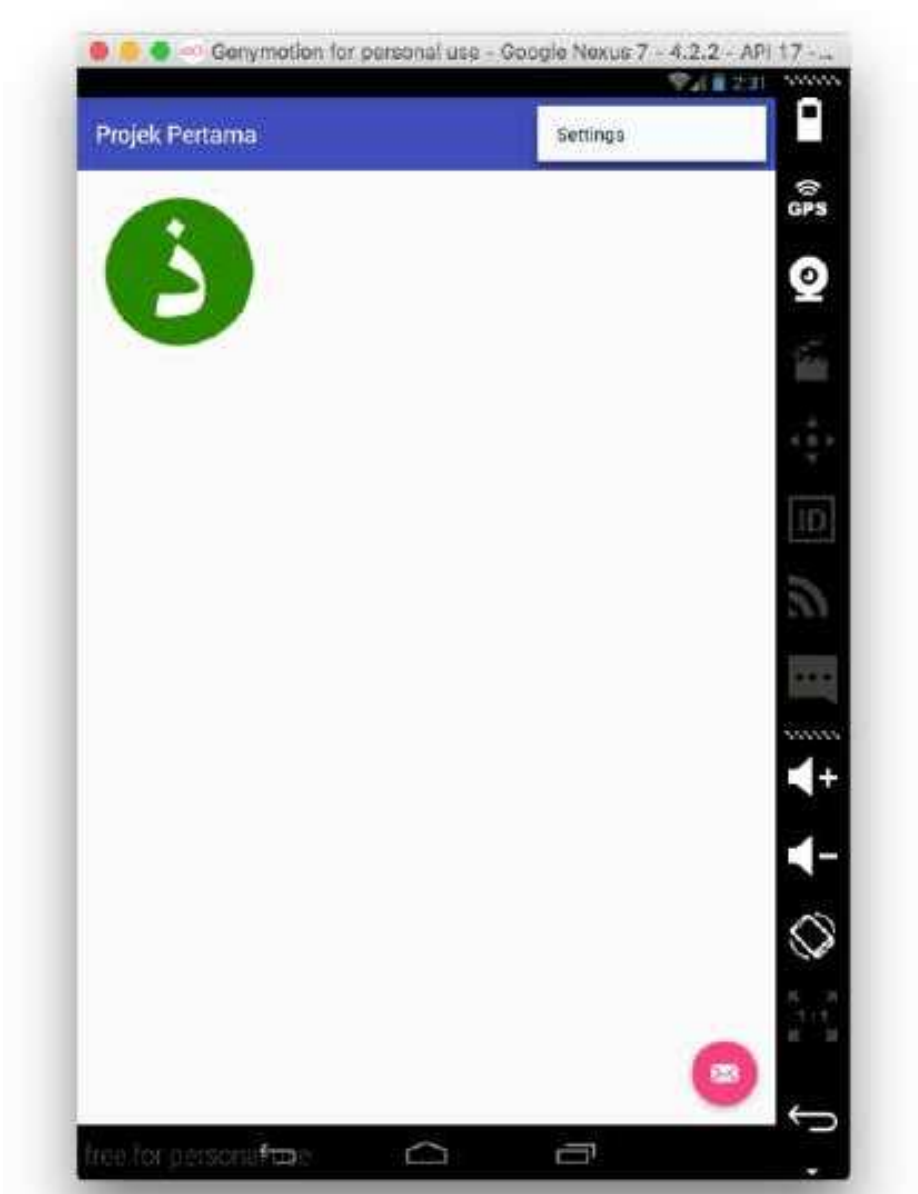
tools:context="com.fazrilabs.projekpertama.MainActivity">
    <item
        android:id="@+id/action_settings"
        android:title="@string/action_settings"
        app:showAsAction="never" />
</menu>
```

Maksud kode xml di atas adalah mendaftarkan sebuah menu dengan id **action_settings** dan judul **@string/action_settings** yang merujuk pada resource teks. Sedangkan potongan kode **app:showAsAction="never"** maksudnya adalah bahwa menu tersebut adalah popup menu. Kemudian anda perlu mendaftarkan

menu tersebut pada sebuah activity agar ditampilkan pada activity yang bersangkutan.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
```

Hasilnya adalah sebagai berikut:



Untuk mengetahui ketika menu tersebut diklik adalah dengan cara meng-*override* metode pada onOptionsItemSelected activity.

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
```

```
        if (id == R.id.action_settings) {
            // lakukan sesuatu disini
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
```

6. View

View merupakan salah satu materi yang penting untuk dipelajari. Setiap aplikasi android pasti akan menggunakan view sebagai salah satu komponennya. Memahami setiap view beserta konfigurasinya sangat membantu seorang pengembang dalam mengembangkan aplikasi android.

6.1. Mengenal View dan ViewGroup

Secara sederhana, view dapat kita artikan sebagai komponen tatap muka atau biasa disebut dengan *user interface*. Masing-masing komponen memiliki fungsi dan tujuan yang berbeda. Akan tetapi semuanya bersinergi satu sama lain untuk membentuk satu tujuan utama. Pengembang aplikasi dapat menyusun berbagai macam view dalam satu layar untuk suatu tujuan yang mereka inginkan.

Sedangkan viewgroup atau disebut juga *parents* merupakan jenis view yang dapat menampung view-view yang lain sebagai anak-anaknya. Contoh viewgroup adalah LinearLayout, RelativeLayout, FrameLayout, dan lain-lain. Masing-masing viewgroup memiliki karakteristik berbeda-beda sesuai dengan maksud dan tujuannya. ViewGroup akan dibahas lebih mendetail pada bab berikutnya ketika membahas tentang layout.

Berikut adalah beberapa atribut view yang perlu anda pahami:

1. *id*

Apabila anda ingin mengakses sebuah view melalui kode java, maka anda harus memberikan atribut **android:id** pada view tersebut. Nilai dari atribut tersebut adalah sembarang teks dengan format **@+id/idview**. Nilai dari **android:id** harus unik dalam 1 file xml. Perhatikan contoh kode xml berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"/>
</LinearLayout>
```

Kode xml di atas mendefinisikan sebuah *button* dengan **android:id** adalah **button1**. Untuk mengakses button tersebut melalui kode java, gunakan format R.id.***idview***. Perhatikan potongan kode java berikut:

```
Button button1 = (Button) findViewById(R.id.button1);
```

Selain berguna untuk mengakses dalam kode java, pemberian id juga berguna ketika anda ingin mengakses view tersebut dalam file xml yang sama. Teknik ini biasa dipakai dalam RelativeLayout. Perhatikan contoh kode xml berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_below="@+id/button1"
        android:text="Button 1"/>
</RelativeLayout>
```

Pada kode xml di atas, didefinisikan dua buah button dalam sebuah RelativeLayout. Id dari kedua button tersebut masing-masing **button1** dan **button2**. Button2 mengacu kepada button1 dengan menggunakan atribut **android:layout_below="@+id/button1"**. Pembahasa detil mengenai RelativeLayout akan dibahas pada bab berikutnya.

2. *layout_width* dan *layout_height*

Setiap view harus memiliki minimal dua atribut berikut, yaitu **android:layout_width** dan **android:layout_height**. Kedua atribut tersebut masing-masing mendefinisikan lebar dan tinggi dari view. Berikut adalah nilai yang bisa anda gunakan untuk mengisi kedua atribut tersebut.

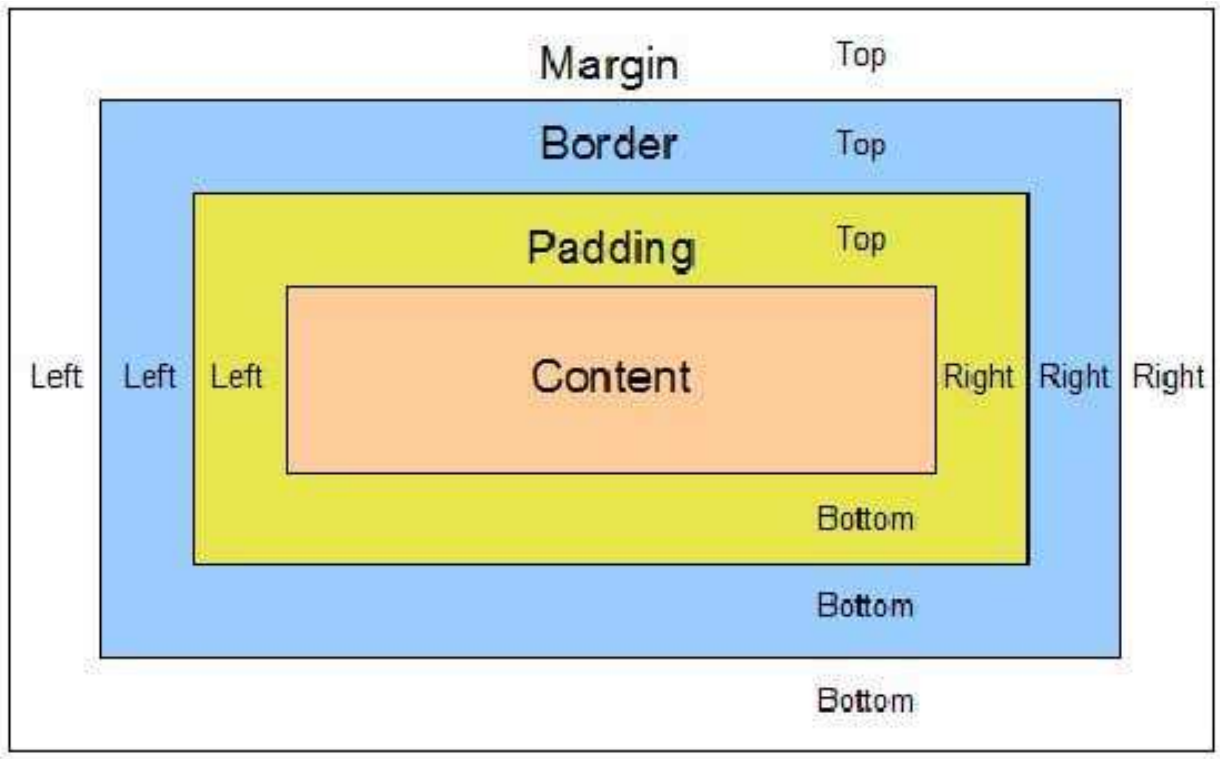
Nilai	Keterangan
wrap_content	Ukuran menyesuaikan dengan ukuran konten view tersebut
match_parent	Ukuran menyesuaikan dengan ukuran kontainer

Selain kedua nilai tersebut, anda juga bisa memberikan sembarang nilai dengan menggunakan satuan **dp** (density-independent pixels). Berbeda dengan piksel (*px*), **dp** merupakan satuan abstrak yang berdasarkan pada kerapatan piksel dari sebuah layar. Berikut adalah contoh potongan kode xml yang menggunakan **dp**.

```
<TextView
    android:layout_width="150dp"
    android:layout_height="20dp"
    android:text="@string/hello_world" />
```

3. *margin* dan *padding*

Margin adalah jarak antara batas (*border*) view dengan *parent* atau view lainnya. Sedangkan *padding* adalah jarak antara batas view dengan konten view tersebut. Gambar di bawah mengilustrasikan bagaimana margin dan padding terletak pada sebuah view.



Untuk memberikan padding dan margin pada sebuah view, anda cukup memberikan nilai padding dan margin tersebut dalam satuan **dp**. Perhatikan penggunaan margin dan padding di bawah ini.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:layout_margin="5dp"
    android:text="@string/hello_world" />
```

Potongan kode xml di atas adalah pendefinisian sebuah view *TextView*. Atribut **android:padding="10dp"** maksudnya adalah memberikan padding sebesar **10dp** pada setiap sisi view. Sedangkan atribut **android:layout_margin="5dp"** adalah untuk memberikan margin sebesar **5dp** pada setiap sisi view.

Android mengizinkan anda untuk mengatur margin dan padding hanya pada sisi tertentu. Tabel berikut merangkum beberapa atribut untuk keperluan tersebut.

Atribut	Keterangan
android:paddingTop	Mengatur padding atas view
android:paddingRight	Mengatur padding kanan view
android:paddingBottom	Mengatur padding bawah view
android:paddingLeft	Mengatur padding kiri view
android:layout_marginTop	Mengatur margin atas view
android:layout_marginRight	Mengatur margin kanan view
android:layout_marginBottom	Mengatur margin bawah view
android:layout_marginLeft	Mengatur margin kiri view

6.2. TextView

TextView merupakan view yang berfungsi untuk menampilkan teks. Teks yang ditampilkan merupakan teks yang tidak dapat diedit oleh user. Berikut adalah beberapa atribut TextView yang penting untuk dipelajari:

Atribut	Keterangan
android:id	Mendefinisikan ID unik untuk TextView
android:gravity	Mengatur penyelarasan (*alignment*) teks pada TextView
android:text	Teks yang akan ditampilkan
android:textAllCaps	Menampilkan teks dalam huruf besar semua
android:textColor	Mengatur warna teks dalam format *#rgb*
android:textSize	Mengatur ukuran teks dalam *dp*
android:textStyle	Mengatur gaya teks: *normal*, *italic*, or *bold*
android:typeface	Mengatur *typeface*: *normal*, *sans*, *serif* or

6.5. ImageView

ImageView merupakan view yang dapat digunakan untuk menampilkan gambar. Anda dapat menampilkan gambar dengan tipe png dan jpg. Sebelum anda dapat menggunakan gambar pada ImageView, anda harus meletakkan gambar tersebut pada folder */res/drawable*. Berikut beberapa atribut ImageView yang penting untuk dipahami:

Atribut	Keterangan
android:id	Mendefinisikan ID unik dari ImageView
android:scaleType	Mengatur jenis skala yang digunakan, antara lain: *center*, *center_crop*, *center_inside*, *fit_center*, *fit_end*, *fit_start*, *fit_xy*, *matrix*.
android:src	Image yang akan ditampilkan

Perhatikan contoh penggunaan ImageView dalam layout berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scaleType="center"
        android:src="@drawable/adzkar"
        android:layout_centerInParent="true"/>
```

File xml di atas akan menghasilkan tampilan layout seperti pada gambar di bawah ini:



6.6. ImageButton

ImageButton merupakan Button yang dapat menampilkan gambar sebagai ganti teks. Dengan menggunakan image, sebuah ImageButton akan tampak lebih menarik oleh user dibandingkan

Button biasa. Berikut adalah beberapa atribut penting dari ImageButton:

Atribut	Keterangan
android:id	Mendefinisikan ID unik dari ImageButton
android:src	Image yang akan ditampilkan

Perhatikan contoh penggunaan ImageButton di bawah ini:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/adzkar"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

Kode xml di atas akan menghasilkan tampilan seperti pada gambar di bawah ini:

@idrumahcoding



Cara mengakses ImageButton melalui kode java hampir mirip dengan Button. Perhatikan contoh potongan kode java di bawah ini:

```
ImageButton imageButton = (ImageButton)
findViewById(R.id.imageButton1);
imageButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // imageButton diklik oleh user
    }
});
```

6.7. Spinner

Spinner merupakan view yang mempunyai drop down menu sehingga user dapat memilih salah satu item dalam drop down menu tersebut. Berikut adalah beberapa atribut Spinner yang penting untuk diketahui:

Atribut	Keterangan
android:id	Mendefinisikan ID unik dari ImageButton
android:entries	Array teks yang akan dipilih oleh user
android:prompt	Hint untuk user

Untuk mendemonstrasikan penggunaan Spinner, langkah pertama anda perlu mendefinisikan array teks pada resource *string.xml*.

```
<string name="country_prompt">Pilih negara</string>
<string-array name="countries">
    <item>Indonesia</item>
    <item>Malaysia</item>
    <item>Jepang</item>
    <item>Singapura</item>
    <item>Thailand</item>
</string-array>
```

Setelah itu anda dapat menggunakan array teks tersebut untuk mengisi atribut **android:entries** pada Spinner.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Spinner
       android:id="@+id/spinner1"
       android:layout_width="200dp"
```

```
    android:layout_height="50dp"
    android:entries="@array/countries"
    android:prompt="@string/country_prompt"
    android:layout_centerInParent="true">

</Spinner>
</RelativeLayout>
```

Berikut adalah tampilan dari Spinner tersebut:



Berikut adalah potongan kode java untuk mengakses Spinner dan mengetahui kapan sebuah item dipilih oleh user:

```
final Spinner spinner = (Spinner)
findViewById(R.id.spinner1);
spinner.setOnItemClickListener(new
```

```
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
        String selectedCountry =
spinner.getSelectedItem().toString();
    }
});
```

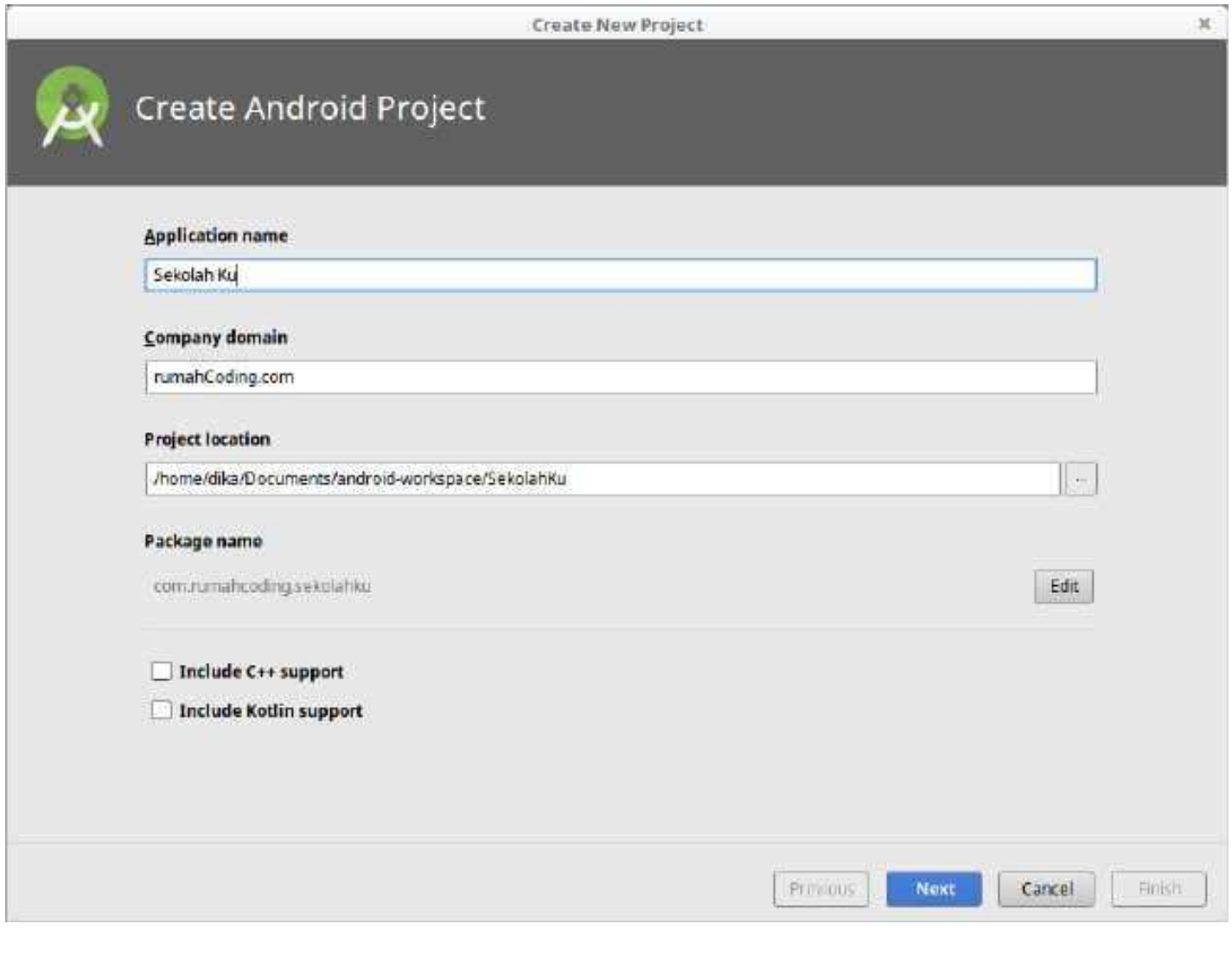
6.8. CheckBox

CheckBox merupakan jenis input yang memiliki kondisi on dan off. CheckBox biasa digunakan dalam sebuah form sebagai field yang memerlukan jawaban ya atau tidak. Berikut adalah beberapa atribut CheckBox yang penting:

Atribut	Keterangan
android:id	Mendefinisikan ID unik dari CheckBox
android:text	Teks yang akan ditampilkan
android:drawableRight	Gambar yang akan ditampilkan di sebelah kanan teks
android:drawableBottom	Gambar yang akan ditampilkan di sebelah bawah teks

Perhatikan contoh penggunaan CheckBox pada layout di bawah ini:

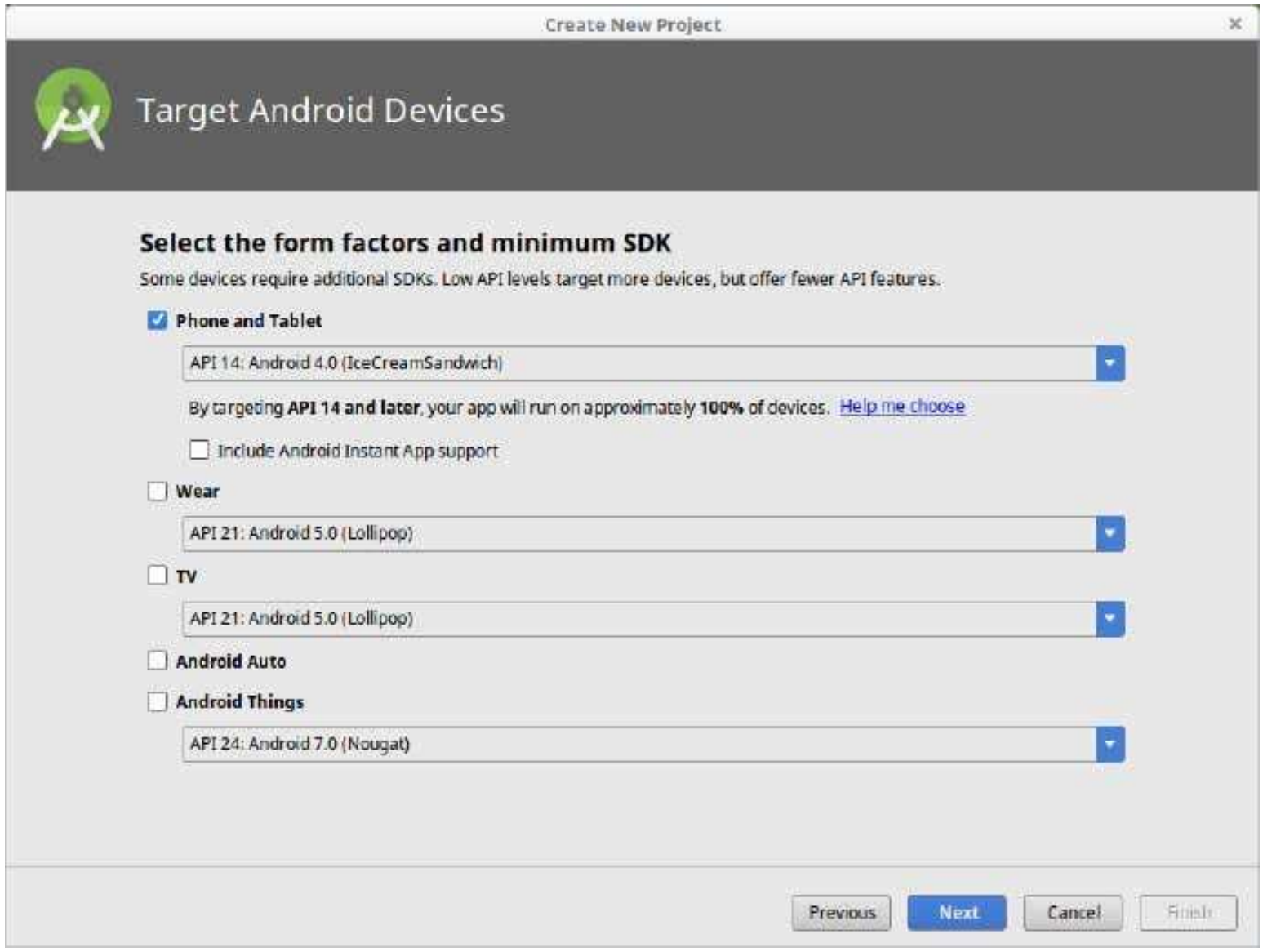
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```



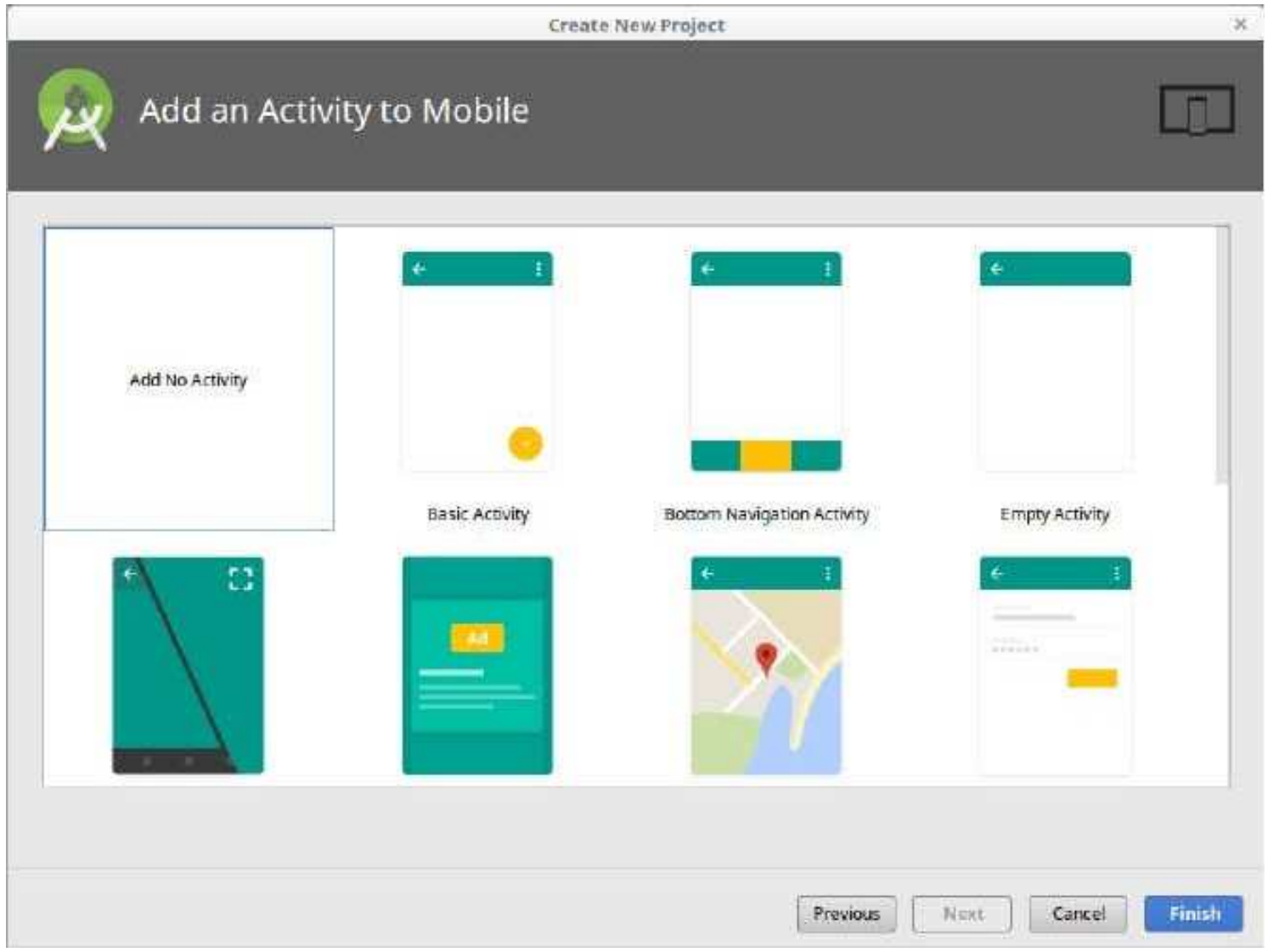
Untuk kolom application name isikan dengan Sekolah Ku dan untuk kolom Company domain diisikan dengan rumahCoding.com. Hal yang perlu diingat disini adalah kolom Project location yang merupakan lokasi dimana project kita tersimpan dan jangan beri tanda ceklist pada checkbox pilihan Include C++ support dan Include Kotlin support karena hanya membutuhkan bahasa pemrograman java dalam membuat aplikasi ini.

Jika sudah klik Next sehingga muncul tampilan seperti dibawah ini :

@idrumahcoding



Pada bagian ini, hanya beri tanda ceklist pada Phone and Tablet kemudian pilih targetnya API-nya versi 15. Klik Next untuk langkah selanjutnya sehingga muncul tampilan berikut:



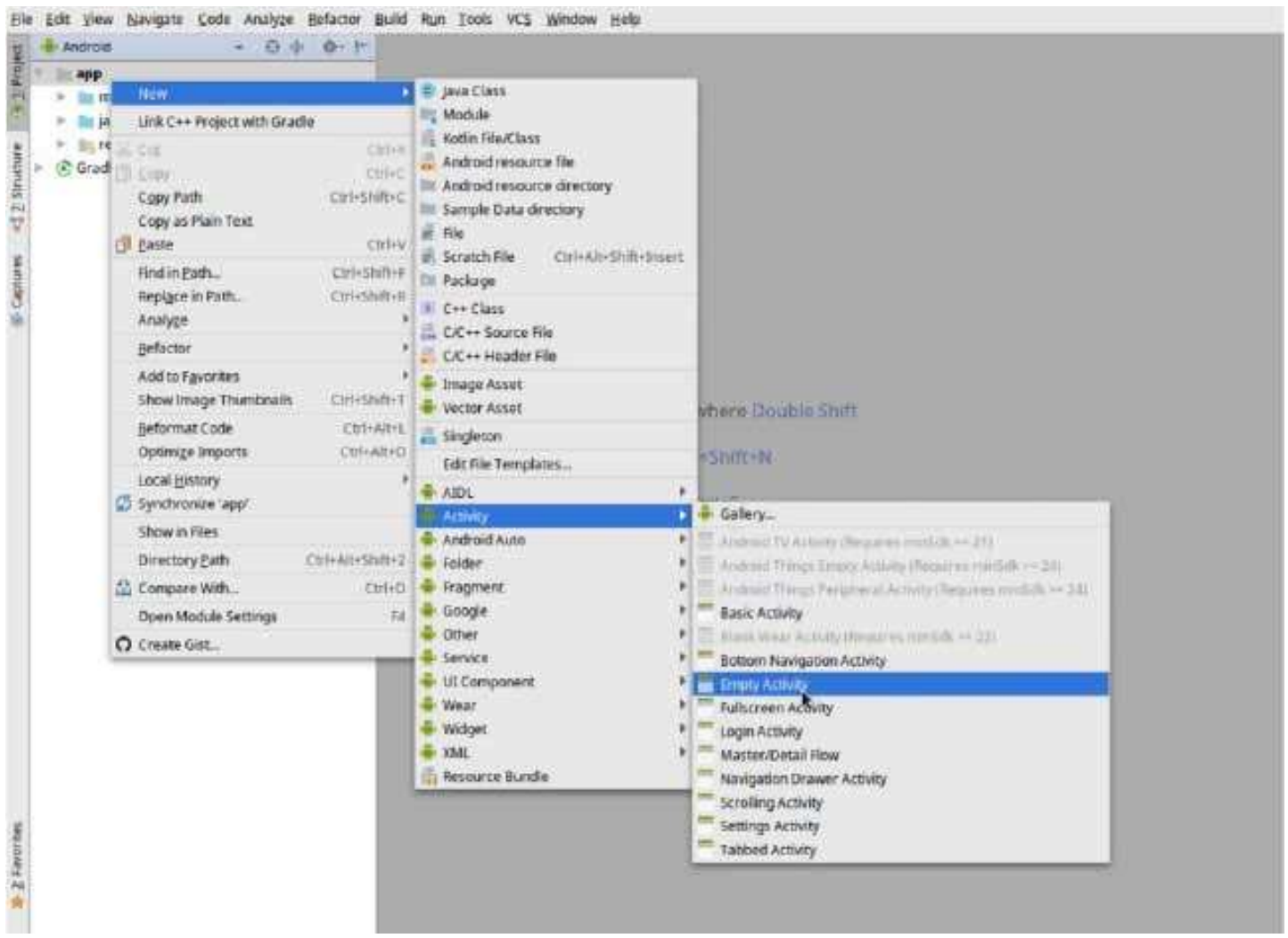
Pada langkah ini pilih Add No Activity karena nanty kita akan membuatnya nanti. Klik finish sehingga muncul tampilan berikut:



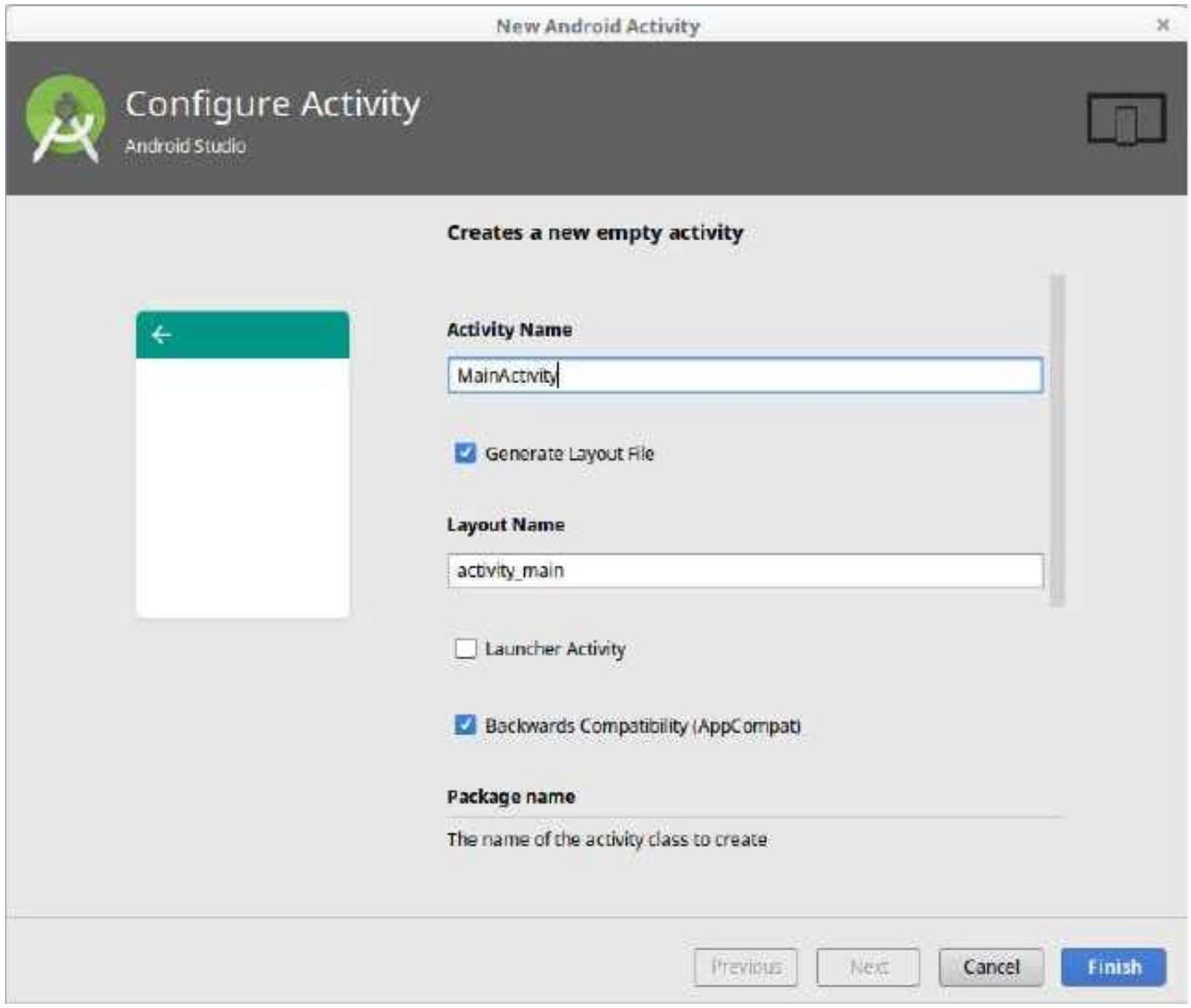
Pada bagian ini anda perlu sedikit bersabar karena project Sekolah Ku sedang mengunduh depedencies(library) yang dibutuhkan dalam membuat aplikasi Sekolah Ku.

2. Membuat FormActivity

Klik kanan pada bagian app seperti yang nampak digambar dibawah ini :



Pilih *New* > *Activity* > *Empty Activity* sehingga muncul dialog seperti gambar dibawah ini :



Ganti isi dari kolom Activity Name menjadi FormActivity. Hal yang perlu diingat adalah beri tanda ceklist pada checkbox Generate Layout File. Pemberian tanda ceklist ini bertujuan untuk membuat layout dari activity yang akan dibuat. Ingat selalu activity adalah gabungan dari file java dan file xml dimana file java merupakan controller yang akan menghandle aksi-aksi yang mungkin dilakukan di dalam activity dan file xml merupakan configurasi tampilan yang akan dilihat pada layar hp android. Langkah-langkah ini akan selalu digunakan setiap kali anda ingin membuat activity baru jadi mengingat langkah ini ini merupakan hal yang penting.

FormActivity sendiri memiliki tujuan untuk menghandel inputan data siswa yang diberikan oleh pengguna aplikasi. Kegiatan yang dihandle oleh activity ini adalah validasi input yang diberikan pengguna dan penyimpanan data siswa yang diinput ke dalam database.

3. Membangun Layout FormActivity

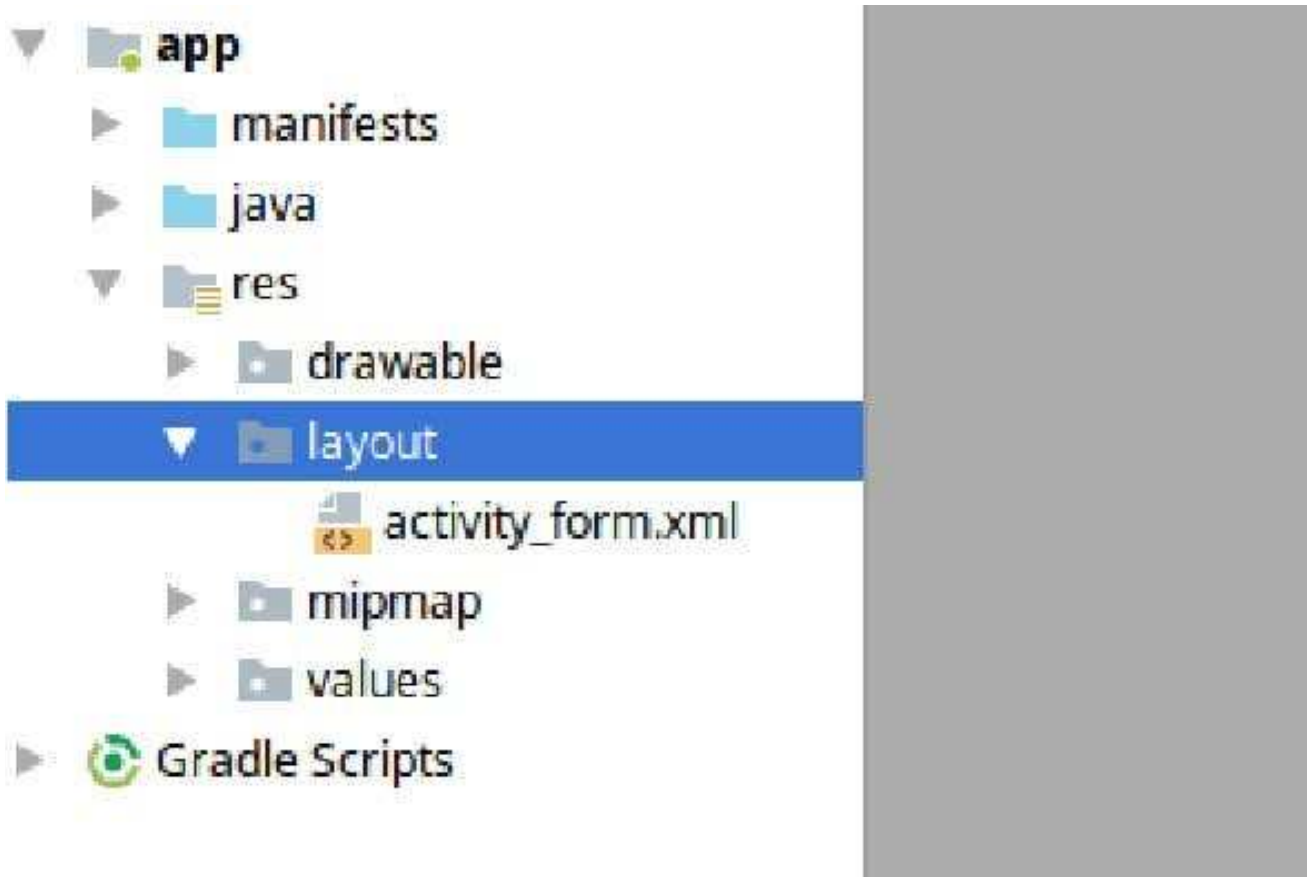
Sebelum mulai membangun layout dari FormActivity, kita akan mengedit terlebih dahulu file configurasi warna yang terdapat dalam res > values > color.xml. Pengeditan ini bertujuan untuk mengubah warna default dari aplikasi. Untuk lebih jelas dimana lokasi file konfigurasinya, lihat gambar dibawah ini :



Buka file color.xml tersebut dan ubah kodenya menjadi seperti dibawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#fa7500</color>
    <color name="colorPrimaryDark">#fa7500</color>
    <color name="colorAccent">#ff8a1e</color>
    <color name="white">#ffffff</color>
</resources>
```

Sekarang buka file configurasi layout yang merupakan file xml yang terletak didalam folder **res > layout**. Ingat kembali materi mengenai Struktur Projek Android Studio yang menjelaskan bahwa setiap file configurasi xml selalu berada pada folder res dan file configurasi xml dari layout selalu terletak dalam **res > layout**. Memahami struktur project adalah hal yang penting. Untuk lebih



Ubah isi dari file **activity_form.xml** tersebut sehingga menjadi seperti ini :

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:orientation="vertical"
    tools:context=".FormActivity">

    <LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="horizontal"
   android:baselineAligned="false">

   <EditText
      android:id="@+id/nama_depan_field"
      android:layout_width="0dp"
                  android:layout_height="wrap_content"
```

```
findViewById(R.id.nama_depan_field);
        namaBelakangField =
findViewById(R.id.nama_belakang_field);
        noHpField = findViewById(R.id.no_hp_field);
        emailField = findViewById(R.id.email_field);
        birtDateField = findViewById(R.id.tgl_lahir_field);
        rbPria = findViewById(R.id.pria_rb);
        rbWanita = findViewById(R.id.wanita_rb);
        spJenjang = findViewById(R.id.jenjang_spinner);
        chkBaca = findViewById(R.id.baca_checkbox);
        chkNulis = findViewById(R.id.chk_nulis);
        chkGambar = findViewById(R.id.chk_gambar);
        alamatField = findViewById(R.id.input_alamat);

        Button btnSave =
findViewById(R.id.btn_save_student);
    }
}
```

Sekarang kita akan buat beberapa method yang akan digunakan untuk menghandle input dari pengguna aplikasi. Caranya adalah dengan menambahkan kode dibawah ini setelah method dengan nama **onCreate.**

```
private void save() {

}

private void showDialogPilihTanggal() {

}

private String getGenderTerpilih() {
    return null;
}

private String getHobiTerpilih() {
    return null;
}
```

```
private boolean validateInput() {
    return true;
}

private boolean validateNama(EditText editText) {
    return true;
}

private boolean validateAlamat() {
    return true;
}

private boolean validateEmail() {
    return true;
}

private boolean validateNoHp() {
    return true;
}

private boolean isEmpty(EditText editText) {
    return true;
}

private String getValueFrom(EditText editText) {
    return null;
}
```

Penjelasan singkat mengenai tujuan membuat method diatas :

- Method **save** nantinya akan digunakan untuk melakukan perintah menyimpan data siswa ke dalam database sqlite.
- Method **showDialogPilihTanggal** nantinya akan digunakan untuk menampilkan dialog pemilihan tanggal.
- Method **getGenderTerpilih** nantinya akan digunakan untuk mengambil pilihan gender. Method ini nantinya akan memberikan nilai balikan dengan tipe data String .

- Method **`getHobiTerpilih`** nantinya akan digunakan untuk mengambil pilihan hobi. Method ini nantinya akan memberikan nilai balikan dengan tipe data String.
- Method **`validateInput`** nantinya akan digunakan untuk mengecek semua input data yang dimasukkan oleh pengguna aplikasi. Jika semua input valid maka method ini akan mengembalikan nilai true.
- Method **`validateNama`** nantinya akan digunakan untuk mengecek input nama yang dimasukkan oleh pengguna aplikasi. Method ini memiliki satu parameter bertipe EditText, hal ini dikarenakan pada layout activity_form.xml terdapat 2 buah view input nama yang masing-masing viewnya bertipe EditText yakni input nama depan dan input nama belakang. Untuk itulah kita perlu untuk mengecek masing input nama tersebut dengan cara validasi yang sama.
- Method **`validateAlamat`** nantinya akan digunakan untuk mengecek input alamat.
- Method **`validateEmail`** nantinya akan digunakan untuk mengecek input email.
- Method **`validateNoHp`** nantinya akan digunakan untuk mengecek input no hp.
- Method **`isEmpty`** nantinya akan mengecek apakah input pada view bertipe data EditText masih kosong atau tidak.
- Method **`getValueFrom`** nantinya akan digunakan untuk mengambil data yang diinput pada view **`EditText`**.

Pemahaman terhadap tujuan pembuatan method tersebut amatlah penting. Karena mengerti alur logika sebuah program jauh lebih penting dibanding memahami arti kode itu sendiri. Membuat kode program tanpa memahami tujuan dari kode akan menyebabkan masalah serius pada program atau bisa dibilang sebagai hal yang percuma.

6. Memberikan Aksi Pada Button Save dan EditText Tanggal Lahir

Massih dalam file FormActivity.java sebelumnya, arahkan cursor dibawah inisialisasi local variabel **btnSave** yang terdapat didalam method **onCreate**. Kemudian ketikkan kode dibawah ini :

```
btnSave.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
      save();
    }

});

emailField.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
      showDialogPilihTanggal();
    }

});
```

Dengan begitu maka code didalam method **onCreate** menjadi seperti dibawah ini :

```
@Override

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_form);

    namaDepanField = findViewById(R.id.nama_depan_field);
    namaBelakangField =
findViewById(R.id.nama_belakang_field);
    noHpField = findViewById(R.id.no_hp_field);
    emailField = findViewById(R.id.email_field);
    birtDateField = findViewById(R.id.tgl_lahir_field);
    rbPria = findViewById(R.id.pria_rb);
    rbWanita = findViewById(R.id.wanita_rb);
    spinjang = findViewById(R.id.jenjang_spinner);
```

```
    chkBaca = findViewById(R.id.baca_checkbox);
    chkNulis = findViewById(R.id.chk_nulis);
    chkGambar = findViewById(R.id.chk_gambar);
    alamatField = findViewById(R.id.input_alamat);

    Button btnSave = findViewById(R.id.btn_save_student);
    btnSave.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
     save();
  }
    });

    emailField.setOnClickListener(new View.OnClickListener()
{
  @Override
  public void onClick(View v) {
     showDialogPilihTanggal();
  }
});

}
```

Sekarang update kode method **getValueFrom** menjadi seperti dibawah ini:

```
private String getValueFrom(EditText editText) {
      return editText.getText().toString().trim();

}
```

Perhatikan kode: **editText.getText().toString().trim()**, kode ini adalah cara untuk mengambil input yang diketikkan oleh pengguna aplikasi pada view yang tipe datanya adalah EditText.

Sekarang update kode method **isEmpty** menjadi seperti dibawah ini :

```
private boolean isEmpty(EditText editText) {
    return getValueFrom(editText).isEmpty();

}
```

Perhatikan kode **getValueFrom(editText).isEmpty()**: , kode ini memiliki arti mengambil nilai bertipe String dari view bertipe EditText karena method **getValueFrom** mengembalikan nilai String. Dan di classs String sendiri sudah tersedia method dengan nama isEmpty yang berfungsi untuk mengecek nilai dari String tersebut. Jika String tersebut masih kosong maka akan mengembalikan nilai false begitu sebaliknya. Nilai balikan dari class String inilah yang menjadi nilai balikan dari method **getValueFrom**.

Selain itu kode diatas juga menjelaskan bagaimana memanggil atau menggunakan method yang sudah kita buat sebelumnya.

Sekarang update kode method **validateNoHp** menjadi seperti dibawah ini:

```
private boolean validateNoHp() {
    if (isEmpty(noHpField)) {
    noHpField.setError("No Hp Masih Kosong");
    noHpField.requestFocus();
    return false;
    }

    String noHp = getValueFrom(noHpField);
    if (noHp.length() > 12) {
    noHpField.setError("No Hp Tidak Valid");
    noHpField.requestFocus();
    return false;
    }

    return true;
}
```

Seperti nama methodnya dan yang sudah dijelaskan sebelumnya, method ini berfungsi untuk mengecek input mengenai no HP yang diinput oleh pengguna aplikasi.

Perhatikan kode: **if**(isEmpty(**noHpField**)), kode if ini berarti jika **EditText** input no hp masih kosong maka eksekusi kode didalam

blok if lalu mengembalikan nilai false yang berarti input no hp masih belum valid.

Perhatikatikan kode: **noHpField.setError("No Hp Masih Kosong")**, kode ini digunakan untuk memberikan notifikasi kepada pengguna aplikasi bahwa no hp masih kosong karena letak kode ini didalam blok **if(isEmpty(noHpField))**.

Sekarang update kode method **validateEmail** menjadi seperti dibawah ini:

```
private boolean validateEmail() {
    if(isEmpty(emailField)) {
            emailField.setError("Email masih kosong");
            return false;
    }

    String email = getValueFrom(emailField);

    if (!email.contains("@")) {
            emailField.setError("Email tidak valid");
            return false;
    }

    return true;
}
```

Perhatikan kode: **if(!email.contains("@"))**, pada kode ini terdapat tanda seru(!) sebelum kata email. Tanda seru dalam java merupakan negasi yang bisa diartikan dengan kata bukan. Sehingga kode ini digunakan untuk mengecek tiap karakter didalam String apakah didalam String tidak terdapat karakter '@'.

Sekarang update kode method **validateAlamat** menjadi seperti dibawah ini:

```
private boolean validateAlamat() {
    if (isEmpty(alamatField)) {
```

```
            alamatField.requestFocus();
            return false;
        }

        return true;
}
```

Perhatikan kode: **alamatField.requestFocus()**, pada method validasi sebelumnya tentu kita sudah banyak melihat kode ini. Kode ini berfungsi untuk membuat view menjadi meliki focus dibandingkan dengan view lainnya. Pada kasus ini view yang meminta focus adalah bertipe EditText sehingga cursor input akan segera berpindah ke view ini.

Sekarang update kode method **validateNama** menjadi seperti dibawah ini:

```
private boolean validateNama(EditText editText) {
    if (isEmpty(editText)) {
      editText.setError("Nama Tidak boleh kosong");
      editText.requestFocus();
      return false;
    }

    String name = getValueFrom(editText);
    String forbiddenChars = "#!$";
    for (char forbiddenChar : forbiddenChars.toCharArray())
{
   if (name.contains(String.valueOf(forbiddenChar))) {
       editText.setError("Nama Tidak Valid");
       return false;
   }
    }

    return true;

}
```

Perhatikan kode:

```
for(char forbiddenChar: forbiddenChars.toCharArray())
```

Kode bisa dibaca dengan 'untuk setiap karakter yang terdapat dalam koleksi karakter terlarang lakukan pernyataan dibawah ini. Maksud dari penyataan disini adalah kode yang terdapat didalam blok perulangan. Dalam kasus ini, kita mengecek input nama apakah memiliki karakter terlarang, jika memang ada maka method ini akan mengembalikan nilai false.

Sekarang update kode method **validateInput** menjadi seperti dibawah ini:

```
private boolean validateInput() {
    return validateNama(namaDepanField)
    && validateNama(namaBelakangField)
    && validateNoHp()
    && validateEmail()
    && validateAlamat();
}
```

Kode diatas merupakan penyederhaan proses validasi sehingga ketika suato operasi memerlukan validasi semua input yang diberikan hanya tinggal memanggil method ini.

Sekarang ubah kode method **getHobiTerpilih** menjadi seperti dibawah ini:

```
private String getHobiTerpilih() {
   List<String> hobies = new ArrayList<>();
   if (chkBaca.isChecked()) hobies.add("Membaca");
   if (chkNulis.isChecked()) hobies.add("Menulis");
   if (chkGambar.isChecked()) hobies.add("Menggambar");

   return TextUtils.join(",", hobies);
}
```

Perhatikan kode: **List<String> hobies = new ArrayList<>()**,

Dalam hal ini kita akan membuat sebuah koleksi data dari sejumlah hobi yang dipilih oleh pengguna aplikasi dan setiap hobi yang dipilih merupakan nilai yang tipe datanya adalah String. Setelah data koleksi tersebut sudah siap, kita akan membuat String baru berdasarkan dari koleksi tersebut dan akan setiap hobinya akan dipisahkan dengan tanda koma(,). Dan untuk melakukannya kita menggunakan kode ini : , yang nantinya akan mengembalikan nilai berupa String.

Perhatikan juga kode:

```
if(chkBaca.isChecked()) hobies.add("Membaca"),
```

Kode ini digunakan untuk mengetahui apakah hobi membaca terpilih. Jika terpilih(diceklist) maka tambahkan hobi membaca ke koleksi hobi.

Sekarang update kode method **getGenderTerpilih** menjadi seperti dibawah ini :

```
private String getGenderTerpilih() {
    if (rbPria.isChecked()) return "Pria";
    else if (rbWanita.isChecked()) return "Wanita";
    else return "";

}
```

Kode diatas hampir sama artinya dengan kode **getGenderTerpilih**, yang membedakan adalah logika didalam method ini. Jika sebelumnya kita membuat koleksi data hobi, disini kita tidak membuat koleksi gender karena sudah jelas logikanya salah. Jadi hanya salah satu gender saja yang bisa dipilih.

7. Membuat Dialog Pilih Tanggal Pada `FormActivity`

Untuk membuat dialog pilih tanggal pada android maka class activity tersebut harus mengimplementasikan interface OnDateSetListener. Interface inilah yang akan digunakan sebagai aksi ketika tanggal terpilih.

Buka class FormActivity kembali dan tambahkan kode `implements DatePickerDialog.OnDateSetListener` setelah nama class, sehingga kode nya menjadi seperti berikut ini :

```
public class FormActivity extends AppCompatActivity
implements       DatePickerDialog.OnDateSetListener {

//kode lainnya..

}
```

Setelah itu tambahkan kode berikut ini dibawah method :

```
@Override
public void onDateSet(DatePicker view, int year,
                      int month, int dayOfMonth) {

       String dateInString= String.valueOf(dayOfMonth) +
           "-" +
           month +
           "-" +
           year;

    birtDateField.setText(dateInString);

}
```

Perhatikan kode: **`birtDateField.setText(dateInString)`**, kode ini digunakan untuk memberikan value atau nilai kepada class property bernama birtDateField dimana tipe data dari properti ini adakah EditText. Sehingga birtDateField yang tadinya kosong kini menjda memiliki nilai/value.

Sekarang update kode method dengan nama **`showDialogDateChooser`** menjadi seperti dibawah ini :

```
private void showDialogDateChooser() {
    Calendar calendar = Calendar.getInstance();

    DatePickerDialog datePickerDialog = new
DatePickerDialog(this, this,
    calendar.get(Calendar.YEAR),
    calendar.get(Calendar.MONTH),
    calendar.get(Calendar.DAY_OF_MONTH));


    datePickerDialog.show();

}
```

Perhatikan kode:

Calendar calendar = Calendar.*getInstance*(),

kode ini berguna untuk mengetahui tanggal hari ini. Selain itu perhatikan juda kode inisialisasi **DatePickerDialog**. Diproses ini kita akan membuat dialog pilih tanggal dan mengatur waktu default yang terpilih adalah tanggal hari ini.

8. Memberikan Aksi Pada Action Button FormActivity

Sebelum memberikan aksi pada action menu FormActivity, anda perlu memanggil layout menu di kode java. Maksud dari memanggil adalah membuat action menu tersebut dapat tampil pada activity FormActivity. Untuk melakukannya maka kita akan menggunakan modul menu inflater. Untuk menggunakan menu inflater ini maka anda perlu untuk meng-*override* method **onCreateOptionsMenu** dan menggunakan file form_activity_menu.xml yang sudah dibuat sebelumnya**.** Taruh kode berikut ini dibawah method **getValueFrom** :

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.form_activity_menu,
menu);
```

```
    close();
}
```

Sekarang update kode method **findAll** menjadi seperti dibawah ini :

```
public List<Siswa> findAll() {
    List<Siswa> siswaList = new ArrayList<>();

    open();
    Cursor cursor = database.rawQuery("SELECT * FROM siswa",
null);
    while (cursor.moveToNext()) {
        Siswa siswa = fetchRow(cursor);
        siswaList.add(siswa);
    }

    cursor.close();
    close();
    return siswaList;
}
```

Sekarang update kode method **findBy** dengan parameter id menjadi seperti dibawah ini :

```
public Siswa findBy(int id) {
    String sql = "SELECT * FROM siswa WHERE id=?";
    Cursor cursor = database.rawQuery(sql, new
String[]{Long.toString(id)});
    Siswa siswa = fetchRow(cursor);
    cursor.close();
    return siswa;
}
```

Sekarang update kode method findBy dengan parameter nama menjadi seperti dibawah ini:

```
public List<Siswa> findBy(String nama) {
    String sql = "SELECT * FROM siswa " +
            "WHERE nama_depan LIKE ? OR " +
```

```
                "nama_belakang LIKE ?";
        open();
        Cursor cursor = database.rawQuery(sql,
                new String[]{"%" + nama + "%", "%" + nama +
"%"});

        List<Siswa> siswaList = new ArrayList<>();
        while (cursor.moveToNext()) {
            Siswa siswa = fetchRow(cursor);
            siswaList.add(siswa);
        }

        cursor.close();
        close();
        return siswaList;
    }
```

12. Mengupdate Kode Method save Pada **FormActivity**

Sekarang buka kembali class **FormActivity** dan sebelum mengupdate kode method save pada **FormActivity**, kita akan membuat method baru dengan nama **initDataSiswa**. Method ini nantinya akan digunakan untuk mengambil semua data input dari pengguna aplikasi dan menampungnya ke dalam object Siswa. Taruh kode berikut ini dibawah method **getValueFrom** :

```
private void initDataSiswa(Siswa siswa) {
    siswa.setNamaDepan(getValueFrom(namaDepanField));
    siswa.setNamaBelakang(getValueFrom(namaBelakangField));
    siswa.setNoHp(getValueFrom(noHpField));
    siswa.setGender(getGenderTerpilih());
    siswa.setHobi(getHobiTerpilih());
    siswa.setAlamat(getValueFrom(alamatField));
    siswa.setEmail(getValueFrom(emailField));
    siswa.setBirthDay(getValueFrom(birtDateField));
}
```

Sekarang buka kembali class `FormActivity` dan ubah method save menjadi seperti dibawah ini :

```
private void save() {
    if (!validateInput()) return;

    Siswa siswa = new Siswa();
    initDataSiswa(siswa);
    SiswaDatasource datasource = new SiswaDatasource(this);
    datasource.save(siswa);
    finish();
}
```

13. Membuat **ListSiswaActivity**

Sekarang kita akan membuat activity baru dengan nama **ListSiswaActivity**. Langkah-langkah untuk membuat activity ini sama dengan cara membuat **FormActivity**, hanya saja nama dari activity-nya yang berbeda. Activity ini nantinya akan menampilkan data siswa yang sudah tersimpan dalam database sebelumnya. Selain itu diactivity ini juga nantinya akan tersedia fitur pencarian siwa berdasarkan dari namanya.

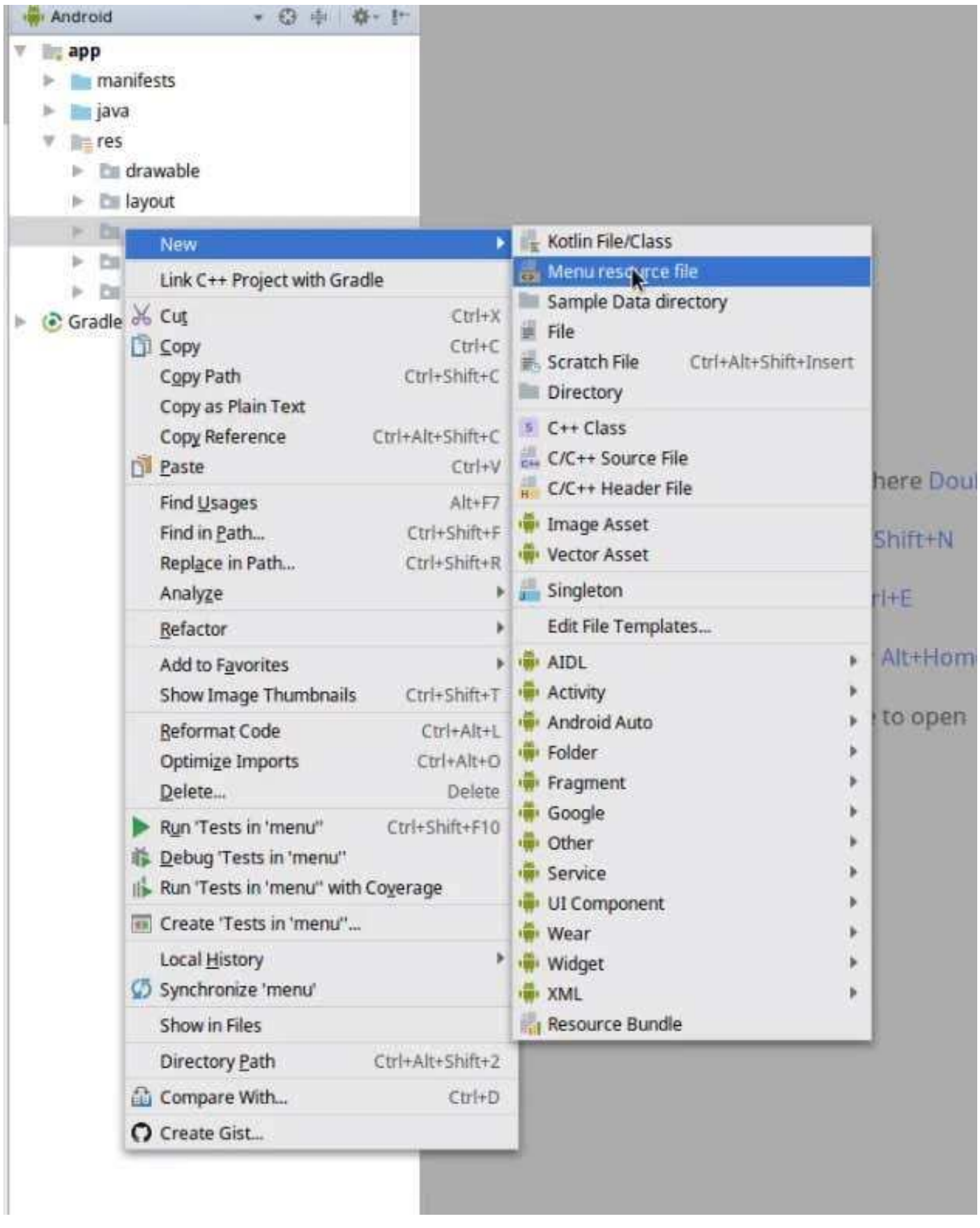
Fitur lain yang nantinya juga ada adalah kita akan membuat context menu, menampilkan detail siswa dan juga action menu. Context menu ini sendiri nantinya akan digunakan untuk melakukan operasi lebih lanjut dalam hal memanipulasi data siswa, baik itu menghapus data siswa terpilih dan mengupdate data siswa terpilih.

Sekarang buka layout file dari activity ini dimana namanya adalah activity_list_siswa yang terletak didalam folder **res > layout**. Setelah itu ubah kodenya menjadi seperti dibawah ini:

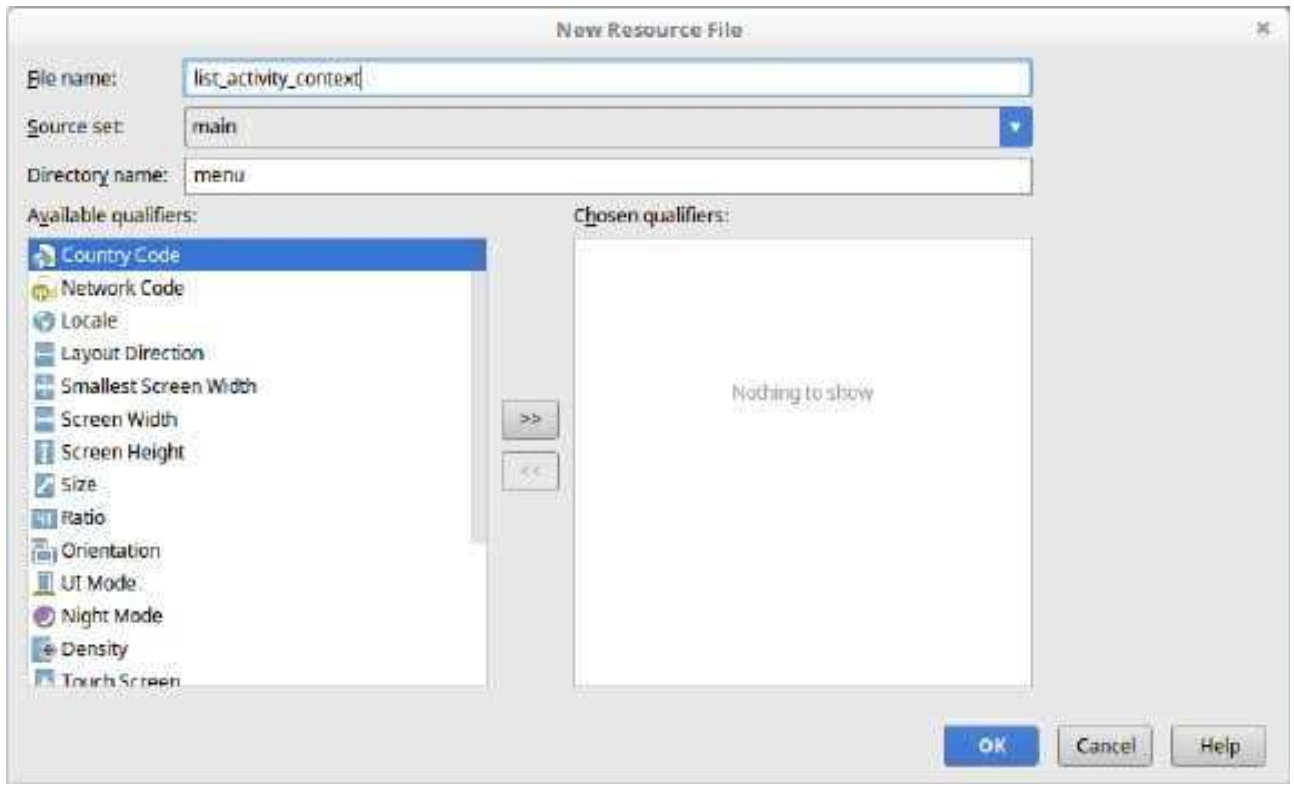
14. Membuat Context Menu Pada **ListSiswaActivity**

Langkah-langkah untuk membuat context menu pada

- Klik kanan pada folder **`res > menu`** dan pilih **`new > Menu`** resource file, seperti yang nampak pada gambar berikut ini :



- Pada tampilan dialog yang muncul, isikan namanya dengan list_activity_context dan klik tombol ok. Seperti yang tampil pada gambar berikut ini :



- Setelah itu buka file menu tersebut dan update kodenya menjadi seperti dibawah ini :

```
<menu
xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/context_update"
        android:title="Update" />

    <item
        android:id="@+id/context_delete"
        android:title="Delete" />
</menu>
```

15. Membuat Action Menu Pada **ListSiswaActivity**

Langkah-langkah untuk membuat action menu pada **ListViewActivity** sama dengan membuat context menu diatas. Hanya saja kali ini pada kolom isian File name, kita isikan namanya menjadi list_siswa_action.

Setelah membuat file tersebut, buka file **list_siswa_action.xml** dan ubah kodenya menjadi seperti berikut ini :

```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/add_student_menu"
        android:title="Tambah Siswa Baru"
        android:icon="@drawable/ic_action_add"
        app:showAsAction="always"/>
</menu>
```

16. Membuat Class **SiswaItemAdapter**

Class ini nantinya akan digunakan sebagai adapter dari item yang akan muncul di activity **ListSiswaActivity**. Buatlah class baru dengan nama **SiswaItemAdapter**, langkah-langkah untuk membuatnya sama dengan langkah-langkah untuk membuat class Siswa. Untuk saat ini kita tidak akan mengubah kode dalam class ini, sehingga kodenya masih seperti dibawah ini :

```
public class SiswaItemAdapter {

}
```

17. Membangun Layout Untuk **SiswaItemAdapter**

Buatlah layout baru dalam folder **res > layout** dan beri namanya siswa_item. Layout ini nantinya akan menjadi tampilan default dari setiap item yang muncul di layout **ListView** Sekarang buka file tersebut dan ubah kodenya menjadi seperti dibawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
android:layout_width="match_parent"
```

```
    android:padding="10dp">

    <ImageView
        android:layout_width="45dp"
        android:layout_height="45dp"
        android:src="@drawable/ic_person"
        android:layout_marginEnd="10dp"
        android:layout_marginRight="10dp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/txt_nama"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Nama"
                android:textSize="20sp"
                android:textStyle="bold"/>

            <TextView
                android:id="@+id/txt_jenjang"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Jenjang"
                android:textAlignment="textEnd"
                android:gravity="end" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
```

- Method **saveDataSiswaBaru**, method ini nantinya akan kita gunakan untuk menyimpan data siswa baru.
- Method **updateDataSiswa**, method ini nantinya akan kita gunakan untuk mengupdate data siswa.

Sekarang update kode method **saveDataSiswaBaru** menjadi seperti dibawah ini :

```
private void saveDataSiswaBaru() {
    Siswa siswa = new Siswa();
    initDataSiswa(siswa);
    SiswaDatasource datasource = new SiswaDatasource(this);
    datasource.save(siswa);
    finish();
}
```

Sebenarnya kode diatas hanya memindahkan kode yang ada di method save ke dalam method. Hal ini dilakukan karena nantinya **FormActivity** akan melakukan 2 aktivitas yaitu menambah data siswa baru dan mengupdate data siswa lama. Selain itu, kedua aktivitas ini juga membutuhkan layout dan proses validasi data yang sama sehingga agar kode menjadi lebih efisien di tambahkan method baru tanpa merubah logika dari kode sebelumnya.

Sekarang update kode method menjadi seperti dibawah ini :

```
private void updateDataSiswa() {
    Siswa siswa = (Siswa)
getIntent().getSerializableExtra("siswa");
    initDataSiswa(siswa);
    SiswaDatasource datasource = new SiswaDatasource(this);
    datasource.update(siswa);
    finish();
}
```

Perhatikan kode: **getSerializableExtra("siswa")**, kode ini berfungsi untuk mengambil data yang dilempar atau dikirim ke dalam activity ini. Dalam kasus ini data yang diterima oleh activity ini merupakan object Siswa dan object ini dikirimkan oleh

ListSiwaActivity. Namun didepan kode ini terdapat kode: **(Siswa) getIntent().getSerializableExtra("siswa")**, dijava hal ini dikenal dengan proses casting. Casting adalah proses membalikkan suatu object menjadi tipe yang sebenarnya. Proses sangat riskan jika anda tidak tahu pasti tipe data asli dari object yang akan dilakukan casting karena jika casting gagal maka akan menyebabkan runtime error dan error yang dimaksud adalah **ClassCastException**.

Sekarang update method menjadi seperti ini :

```
private void save() {
    if (!validateInput()) return;

    String action = getIntent().getStringExtra("action");
    if (action.equals("UPDATE_DATA_SISWA")) {
        updateDataSiswa();
    } else if (action.equals("TAMBAH_DATA_SISWA")) {
        saveDataSiswaBaru();
    }
}
```

Kode diatas yang akan mengatur alur aksi dari menyimpan data ke database. Jika aksinya adalah update maka akan memanggil method **updateDataSiswa** dan jika aksi adalah tambah data siswa bari maka method **saveDataSiswaBaru** yang akan dieksekusi.

Sekarang update kode method **onCreate** dan tambahkan kode berikut dibawah deklarasi **setContentView(R.layout.*activity_form*)**:

```
String action = getIntent().getStringExtra("action");
if (action.equals("UPDATE_DATA_SISWA")) {
    setTitle("Update Data Siswa");
} else if (action.equals("TAMBAH_DATA_SISWA")) {
    setTitle("Tambah Data Siswa");
}
```

Perhatikan kode : , kode ini digunakan untuk mengatur judul activity di app bar atau dengan kata lain mengatur judul activity yang

25. Membuat **DetailActvity**

DetailActivity adalah activity yang akan menampilkan secara detail data siswa yang terpilih pada **ListSiswaActivity**. Langkah-langkah untuk membuat activity ini sama dengan membuat activity lainnya, hanya saja nama activity-nya yang berbeda. Untuk activity ini kita akan menggunakan nama **DetailActivity**.

26. Membangun Layout **DetailActivity**

Sebelum membangun layout dari **DetailActivity**, kita akan membuat layout seperator. Layout ini nantinya akan berguna untuk memisahkan antara satu detail data dengan detail data lainnya. Langkah-langkah untuk membuat layout separator sama dengan langkah membuat layout `siswa_item.xml`, hanya saja namanya yang berbeda karena kali ini kita akan membuat layout dengan nama separator.

Jika sudah berhasil membuat layout tersebut maka buka file-nya dan ubah kode di dalam `separator.xml` menjadi seperti dibawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<View
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_gravity="center"
    android:layout_marginBottom="20dp"
     android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    android:background="@color/colorAccent"/>
```

Sekarang buka file **activity_detail.xml** lalu ubah kodenya menjadi seperti dibawah ini :

@idrumahcoding

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <ImageView
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center"
            android:src="@drawable/ic_person" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <ImageView
                android:layout_width="45dp"
                android:layout_height="45dp"
                android:src="@drawable/ic_label" />

            <LinearLayout
                android:layout_width="match_parent"
                  android:layout_height="wrap_content"
                android:orientation="vertical">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Nama" />

                <TextView
                    android:id="@+id/txt_nama"
```

```
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Nama"
                android:textSize="20sp"
                android:textStyle="bold" />
    </LinearLayout>
</LinearLayout>

 <include layout="@layout/separator"/>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="5dp">

    <ImageView
        android:layout_width="45dp"
        android:layout_height="45dp"
        android:src="@drawable/ic_smartphone" />

    <LinearLayout
          android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="No. Hp" />

        <TextView
            android:id="@+id/txt_no_hp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="No. Hp"
            android:textSize="20sp"
            android:textStyle="bold" />
    </LinearLayout>
</LinearLayout>
```

```
        <include layout="@layout/separator"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="5dp">

            <ImageView
                android:layout_width="45dp"
                android:layout_height="45dp"
                android:src="@drawable/ic_favorite" />

            <LinearLayout
                  android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Gender" />

                <TextView
                    android:id="@+id/txt_gender"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Gender"
                    android:textSize="20sp"
                    android:textStyle="bold" />
            </LinearLayout>
        </LinearLayout>

         <include layout="@layout/separator"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="5dp">

            <ImageVie
```

```
            android:layout_width="45dp"
            android:layout_height="45dp"
            android:src="@drawable/ic_flight_takeoff" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Jenjang" />

            <TextView
                android:id="@+id/txt_jenjang"
                  android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Jenjang"
                android:textSize="20sp"
                android:textStyle="bold" />
        </LinearLayout>
    </LinearLayout>

    <include layout="@layout/separator" />

    <LinearLayout
        android:layout_width="match_parent"
         android:layout_height="wrap_content"
        android:layout_marginBottom="5dp">

        <ImageView
            android:layout_width="45dp"
            android:layout_height="45dp"
            android:src="@drawable/ic_fitness_center" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
```

```
                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Hobi" />

                    <TextView
                        android:id="@+id/txt_hobi"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Hobi"
                        android:textSize="20sp"
                        android:textStyle="bold" />
                </LinearLayout>
            </LinearLayout>

            <include layout="@layout/separator"/>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="10dp">

                <ImageView
                    android:layout_width="45dp"
                    android:layout_height="45dp"
                    android:src="@drawable/ic_location_on" />

                 <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Alamat" />

                    <TextView
                        android:id="@+id/txt_alamat"
```

```
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Alamat"
                    android:textSize="20sp"
                      android:textStyle="bold" />
            </LinearLayout>
        </LinearLayout>

    </LinearLayout>
</ScrollView>
```

Perhatikan kode;

<include layout="@layout/separator"/>,

Kode ini digunakan untuk menggunakan layout yang sudah dibuat sebelumnya ke dalam layout lainnya. Dalam kasus ini kita menggunakan layout **separator.xml** ke dalam layout **activity_detail.xml**.

27. Membuat Controller **DetailActivity**

Actvity ini tidaklah melakukan hal banyak selain menampilkan detail data siswa. Jadi aktivitas yang akan dilakukan activity ini hanya sebatas menerima data siswa dari activity **ListSiswaActivity** dan menampilkan data yang diterima tersebut.

Buka class **DetailActivity** dan update method **onCreate** menjadi seperti dibawah ini:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_detail);

    TextView namaTxt = findViewById(R.id.txt_nama);
    TextView noHpTxt = findViewById(R.id.txt_no_hp);
    TextView genderTxt = findViewById(R.id.txt_gender);
```

```
    TextView hobiTxt = findViewById(R.id.txt_hobi);
    TextView alamatTxt = findViewById(R.id.txt_alamat);

    Siswa siswa = (Siswa)
getIntent().getSerializableExtra("siswa");
    String nama = siswa.getNamaDepan()+"
"+siswa.getNamaBelakang();
    namaTxt.setText(nama);
    noHpTxt.setText(siswa.getNoHp());
    genderTxt.setText(siswa.getGender());
    jenjangTxt.setText(siswa.getJenjang());
    hobiTxt.setText(siswa.getHobi());
    alamatTxt.setText(siswa.getAlamat());
}
```

28. Menambahkan Action Menu Pada **DetailActivity**

Sama dengan penambahan action menu sebelumnya, pertama kita perlu membuat menu resource file yang baru dan beri namanya detail_activity_action_menu. Kemudian buka file yang dibuat tadi dan update kodenya menjadi seperti dibawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/home_menu"
         android:title="Home"
        android:icon="@drawable/ic_home"
        app:showAsAction="always"/>
</menu>
```

29. Memberikan Aksi Pada Action Menu **DetailActivity**

Seperti yang sebelumnya, memberikan aksi pada action menu

java, dalam kasus ini di DetailActivity. Cara untuk memanggilnya adalah dengan meng-*override* method **onCreateOptionsMenu** lalu update kodenya menjadi seperti dibawah ini :

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.detail_activity_action_menu
, menu);
    return super.onCreateOptionsMenu(menu);
}
```

Dan untuk meberikan aksinya maka method yang harus di-*override* adalah method . Berikut adalah kode lengkapnya :

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int itemId = item.getItemId();
    if (itemId == R.id.home_menu) {
        finish();
    }
    return super.onOptionsItemSelected(item);
}
```

30. Mengupdate Kode **ListSiswaActivity**

Buka kembali class **ListSiswaActivity**. Tambahkan method baru dibawah method **addNewSiswa** dan beri nama **viewDetailSiswa** lalu ketikkan kode dibawah ini didalam method tersebut :

```
private void viewDetailSiswa(int position) {
    Siswa siswa = adapter.getItem(position);
    Intent intent = new Intent(this, DetailActivity.class);

   intent.putExtra("siswa", siswa);
    startActivity(intent);
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.rumahcoding.sekolahku">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".SplashScreenActivity"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".LoginActivity" />
        <activity android:name=".FormActivity" />
        <activity android:name=".ListSiswaActivity" />
        <activity android:name=".DetailActivity" />
    </application>

</manifest>
```

Perhatikan kode pada bagian ini :

```
<activity android:name=".SplashScreenActivity"
    android:theme="@style/AppTheme.NoActionBar">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category
android:name="android.intent.category.LAUNCHER" />
```

```
    </intent-filter>
</activity>
```

Kode diatas merupakan konfigurasi untuk activity **SplashScreenActivity**. Konfigurasi activity ini diawali dengan tag **<intent-filter> </intent-filter>** sementara untuk kode konfigurasinya diletakkan diantara tag tersebut. Dalam kasus ini kita mengatur **SplashScreenActivity** menjadi activity utama yang nantinya akan muncul pertama kali saat aplikasi dijalankan.