

Modul

Pemrograman Aplikasi Android



Dyah Fajar Nur Rohmah
2012

DAFTAR ISI

DAFTAR ISI.....	2
CHAPTER 1 INTRODUCING	4
Project Structure.....	4
CHAPTER 2 ANDROID PROJECT	6
CHAPTER 3 VIEWS.....	8
Menambah TextView dan String Value.....	8
Menambah Button.....	9
Memberikan Aksi Pada Button	11
EditText	12
CheckBox.....	14
RadioButton	16
CHAPTER 4 RESOURCES	18
Drawable Resource.....	18
XML Bitmap Drawable	19
Shape Drawable	20
State List Drawable.....	21
String Resource.....	22
Plain String.....	22
Format String.....	23
Color.....	25
CHAPTER 5 LAYOUT.....	30
Linear Layout.....	30
Relative Layout.....	31
Table Layout.....	33
Scroll View.....	34
CHAPTER 6 MULTI SCREEN	38
CHAPTER 7 LOCATION BASED SERVICES	43
Mendaftar Map API Key.....	43

<i>Windows User</i>	43
<i>Linux User</i>	45
Membuat Aplikasi Map.....	46
CHAPTER 8 DATABASE	50
SQLite	50

CHAPTER 1 INTRODUCING

Project Structure

Project Android dibangun berdasarkan direktori yang spesifik seperti saat kita membuat project Java. Untuk membantu memahaminya, kita mulai dengan penjelasan di bawah.

Root Content

Ketika kita membuat sebuah project Android, kita akan memiliki 5 item utama dalam direktori root project kita.

- `AndroidManifest.xml`, adalah file XML yang mendeskripsikan aplikasi yang dibangun dan komponen (activities, services, etc) apa saja yang disediakan oleh aplikasi.
- `build.xml`, adalah script Ant untuk meng-*compile* aplikasi dan meng-*install*nya dalam *device*.
- `bin/`, yang memuat aplikasi setelah di-*compile*
- `src/`, yang memuat *file-file source code* java untuk pembuatan aplikasi
- `res/`, yang memuat "seluruh sumber", seperti *icon*, *GUI Layout*, dan sejenisnya.
- `assets/`, yang memuat *file-file* static lainnya yang ingin dikemas bersama aplikasi untuk di-*deploy* ke dalam *device*.

Activity

Ketika kita membuat sebuah project, kita akan disediakan sebuah *class* dari *activity* utama untuk aplikasi android yang dibuat yang nantinya disimpan di dalam direktori `src/`. Kita bisa memodifikasi file ini dan menambahkan file lainnya ke dalam `src/` sesuai kebutuhan untuk implementasi aplikasi.

Pertama kali project di-*compile*, di luar direktori *activity* "utama", Android akan membuat *file* `R.java`. *File* ini berisi beberapa konstanta dari seluruh "sumber" yang dibuat dan disimpan dalam direktori `res/`. Kita tidak harus memodifikasi *file* `R.java`, karena Android *tools* yang akan menanganinya.

"Resources" Directory

Dalam project Android, kita juga akan menjumpai direktori `res/` yang memuat "sumber-sumber" (*file* tetap seperti gambar yang dikemas bersama dengan aplikasi). Beberapa sub direktori yang akan dijumpai atau dibuat di bawah direktori `res/` terdiri dari:

- `res/drawable/` untuk gambar (PNG, JPEG, etc)
- `res/layout/` untuk spesifikasi UI *layout* yang dibuat dengan XML
- `res/raw/` for general-purpose files (misalnya CSV File of Account Information)
- `res/values/` untuk menyimpan nilai strings, dimensions dan sejenisnya
- `res/xml/` for other general purpose XML Files you wish to ship

The Result

Ketika project di-*compile*, hasil *compiling* akan masuk ke direktori `bin/` di bawah direktori root project. Lebih jelasnya:

- `bin/classes/` memuat *class* java hasil *compiling*

- `bin/classes.dex` memuat the executable created from those compiled Java classes
- `bin/yourapp.apk` adalah file aplikasi Android yang sebenarnya (dimana `yourapp` adalah nama dari aplikasi yang dibuat)

file `.apk` adalah ZIP archive yang berisi *file* `.dex`, hasil *compiling file* sumber (`resources.arsc`), beberapa *file* sumber yang tidak *ter-compile* (seperti yang disimpan di dalam `res/raw/`) dan *file* `AndroidManifest.xml`.

CHAPTER 2 ANDROID PROJECT

Untuk *developing* Android, kita akan menggunakan Eclipse. Seperti biasa, kita akan membuat simple program Hello World..!:

1. Untuk membuat project baru, pilih **File > New > Android Project**. Atau bisa juga dengan mengklik *icon* Android Project Creator  pada eclipse *toolbar*.
2. Isikan *Project Name*, sebagai contoh disini adalah Droid1.
3. Pilih lokasi penyimpanan project. Karena ini adalah project baru, maka pilih *Create New Project in Workspace*. Tandai *checkbox* untuk *Use Default Location*.

Catatan

Jika ingin menyimpan di lokasi lain, hilangkan tanda pada *checkbox*, kemudian, *browse* lokasi yang diinginkan.

4. Pilih *Build Target* untuk aplikasi yang ingin dibuat. Untuk kebanyakan aplikasi, kita akan memilih versi Android yang sesuai dengan *device* yang digunakan yang nantinya dan kebutuhan dari aplikasi yang kita buat.
5. Isikan *Application Name*. Nama ini yang nantinya akan dilihat oleh *user*. Sebagai contoh namanya adalah Droid #1.
6. Isikan *package name* untuk aplikasi yang dibuat. Sebagai contoh com.latihan.droid1.
7. Tandai *CheckBox Create Activity*, kemudian isikan nama activitynya.
8. *Field* Min SDK Version menandai versi SDK yang bisa disupport. Biasanya field ini akan langsung terset sesuai dengan pilihan yang dipilih di *Build Target*. Jika aplikasi yang dibuat ingin dibuat agar support dengan versi sebelumnya, kita bisa mengganti isian dari *field* ini. Ada baiknya jika kita membiarkan *field* ini dengan nilai *default*nya.
9. Kemudian klik tombol **Next**.

Project name: Latihan1

Contents

- ☒ Create new project in workspace
- ☐ Create project from existing source
- ☒ Use default location

Location: D:/Project/ECLIPSE/Latihan1

☐ Create project from existing sample

Samples: ApiDemos

Build Target

Target Name	Vendor	Platform	API ...
<input type="checkbox"/> Android 1.1	Android Open Source Project	1.1	2
<input type="checkbox"/> Android 1.5	Android Open Source Project	1.5	3
<input type="checkbox"/> Google APIs	Google Inc.	1.5	3
<input checked="" type="checkbox"/> Android 1.6	Android Open Source Project	1.6	4
<input type="checkbox"/> Google APIs	Google Inc.	1.6	4
<input type="checkbox"/> Android 2.0	Android Open Source Project	2.0	5
<input type="checkbox"/> Google APIs	Google Inc.	2.0	5
<input type="checkbox"/> Android 2.0.1	Android Open Source Project	2.0.1	6
<input type="checkbox"/> Google APIs	Google Inc.	2.0.1	6
<input type="checkbox"/> Android 2.1	Android Open Source Project	2.1	7
<input type="checkbox"/> Google APIs	Google Inc.	2.1	7

Standard Android platform 1.6

Properties

Application name: Droid #1

Package name: com.latihan

☒ Create Activity: DroidActivity

Min SDK Version: 4

10. Kemudian klik tombol **Finish**.

CHAPTER 3 VIEWS

Menambah TextView dan String Value

Sebelumnya buatlah dulu *value* atau nilai isian Stringnya pada file `strings.xml` yang ada di bawah folder `/res/values` menjadi seperti ini:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, This My First Droid App</string>
    <string name="app_name">Droid #1</string>
</resources>
```

Setelah melakukan pembuatan string tersebut, maka, sintaks pada `R.java` akan berubah menjadi seperti di bawah ini:

```
/* AUTO-GENERATED FILE. DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found. It
 * should not be modified by hand.
 */

package com.latihan;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

Catatan

File `R.java` ini sebaiknya tidak dimodifikasi, karena akan di-*generate* langsung oleh *Android tools*nya.

Pada file layout `main.xml`, tambahkan sintaks program menjadi seperti di bawah ini:


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
</LinearLayout>
```

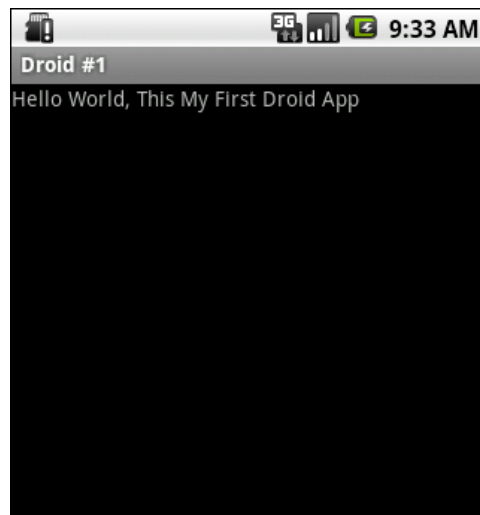
Catatan

Saat kita mendefinisikan *value* atau isi string yang ingin ditampilkan di dalam `strings.xml`, secara otomatis, *Android tools* akan *generate* file `R.java` dengan penambahan sintaks berikut:

```
public static final class string {
    public static final int app_name=0x7f040001;
    public static final int hello=0x7f040000;
}
```

Sehingga, untuk menampilkannya pada `main.xml`, kita hanya perlu memanggil konstanta yang ada pada `R.java` dengan perintah: `android:text="@string/hello"`

Jika sudah, silakan jalankan, maka, tampilan yang seharusnya muncul akan jadi seperti ini:



Menambah Button

Untuk menambah *button*, pada file `main.xml`, ubah sintaks program menjadi seperti di bawah ini:

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

```

    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    android:id="@+id/txtHello"
/>

<Button
    android:text="Exit"
    android:id="@+id/btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Button>

<Button
    android:text="Ganti"
    android:id="@+id/btn2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Button>
</LinearLayout>

```

Untuk nama label tombolnya, kita bisa menuliskan langsung nama labelnya seperti ini:

```
android:text="Exit"
```

Atau kita bisa mendaftarkan string valuenya pada strings.xml dengan menambahkan sintaks ini:

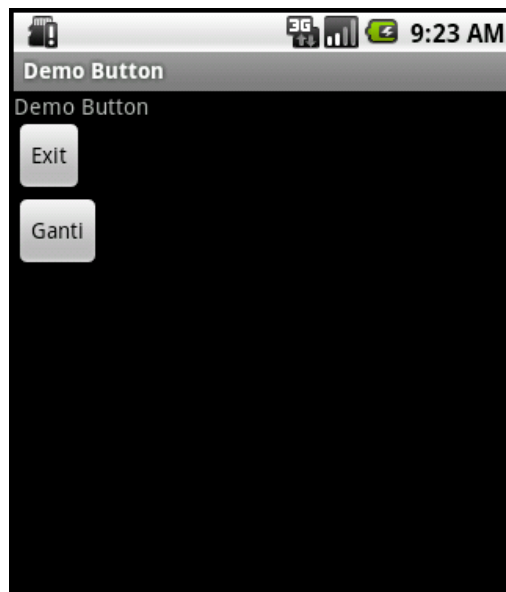
```
<string name="btn1Text">Exit</string>
```

Kemudian pada main.xml-nya, penulisannya menjadi seperti ini:

```
android:text="@string/btn1Text"
```

Catatan

Setelah dijalankan, maka tampilan akan menjadi seperti di bawah ini:



Memberikan Aksi Pada Button

Setelah sebelumnya kita membuat tampilan button, kali ini, kita akan memberikan aksi pada button yang sudah kita buat sebelumnya. Pemberian aksi kepada button ini dilakukan di *class* activity yang tersimpan di bawah folder *src/* dan melibatkan *class interface* *OnClickListener*. *OnClickListener* digunakan untuk memberikan umpan balik yang diminta ketika sebuah *view* di-klik.

Ubahlah *class* activity yang anda punya menjadi seperti di bawah ini:

```
package com.latihan;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class ButtonActivity extends Activity implements OnClickListener{
    Button btnExit;
    Button btnGanti;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btnExit = (Button) findViewById(R.id.btn1);
        btnGanti = (Button) findViewById(R.id.btn2);

        btnExit.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v == btnExit){
            finish();
        }
    }
}
```

Catatan

Ketika kita membuat tampilan aplikasi DemoButton sebelumnya, pada *main.xml* untuk elemen Button, kita menuliskan id untuk button dengan sintaks yang seperti ini:

`android:id="@+id/btn1"` untuk tombol "Exit"

`android:id="@+id/btn2"` untuk tombol "Ganti"

Secara otomatis, Android tools akan membuatkan konstantanya pada *class* *R.java*, sehingga pada *file* tersebut, akan ada penambahan sintaks ini:

```
public static final class id {
    public static final int btn1=0x7f050000;
    public static final int btn2=0x7f050001;
}
```

Sehingga pada instansiasi *Button* di dalam *class* activity, kita memanggil id button yang sudah ada pada class R. java (ex: R.id.*btn1*).

Latihan

Pada project DemoButton, kita akan menggunakan tombol Ganti untuk mengganti string yang berada di atasnya. Maka lakukan modifikasi pada *class* activity yang anda punya sebagai berikut:

1. Buatlah field untuk TextView pada *class* activity yang anda punya dengan menambahkan kode di bawah ini:

```
TextView viewHello;
```

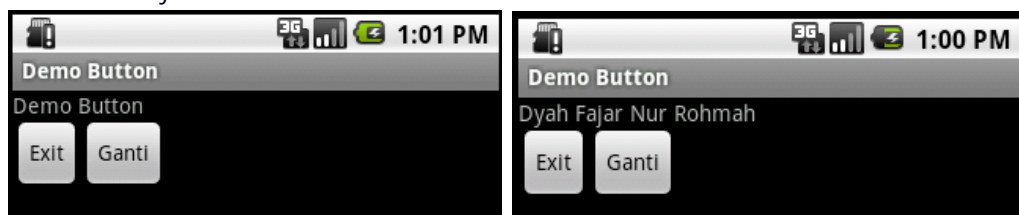
Instansiasikan *field* tersebut seperti di bawah ini:

```
viewHello = (TextView) findViewById(R.id.txtHello);
```

2. Instansiasi btnGanti pada *class* activity yang tersimpan dalam folder *src/*, dan arahkan btnGanti tersebut pada *OnClickListener*.
3. Tambahkan pada method *OnClick(View v)* sintaks program untuk mengganti text "Demo Button" menjadi nama anda seperti di bawah ini:

```
if (v == btnGanti) {
    viewHello.setText("Dyah Fajar Nur Rohmah");
}
```

4. Jalankan project DemoButton, ketika menekan tombol Ganti, text "Demo Button" akan berubah menjadi nama anda.



EditText

Seperti namanya, *widget* ini dikonfigurasi agar bisa dilakukan pengubahan. Contoh program berikut adalah untuk menambah *EditText*.

Berikut adalah sintaks untuk **strings.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Demo EditView</string>
    <string name="title">Type here</string>
    <string name="strBtn1">Tampilkan</string>
    <string name="strBtn2">Exit</string>
</resources>
```

Berikut adalah sintaks program untuk **main.xml**

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

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_horizontal"
    >

    <TextView
        android:text="@string/title"
        android:id="@+id/txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/txtEdit"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
    </EditText>

    <Button
        android:text="@string/strBtn1"
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </Button>

    <Button
        android:text="@string/strBtn2"
        android:id="@+id/btn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </Button>
</LinearLayout>

```

Maka tampilan yang akan muncul adalah sebagai berikut:



Catatan

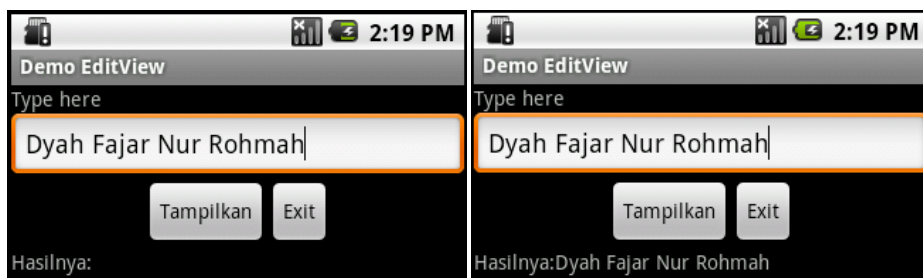
`android:gravity`

Digunakan untuk mengatur bagaimana suatu *content* dalam sebuah object ditempatkan secara X dan Y axis di dalam object itu sendiri. *Value* ini bisa diisi satu atau lebih (dipisahkan dengan "|") dari konstanta-konstanta yang disediakan untuk *gravity*.

Latihan

Pada project untuk menampilkan Edit Text ini, modifikasikan programnya sehingga apa yang ditulis oleh user ditampilkan kembali sebagai text view.

1. Tambahkan *widget* TextView pada layout (`main.xml`) dan letaknya berada di bawah Button.
2. Pada *class* activity, tambahkan sintaks program untuk mendeklarasikan dan menginstansiasi Button, TextView dan EditText.
3. Implementasikan *class interface* OnClickListener().
4. Pada method `OnClick(View v)`, tambahkan sintaks untuk memproses Button, jika Button Tampilkan yang di tekan, maka, apa yang tertulis pada EditText tertampil pada TextView (gunakan method `getText()` dan `setText()`).



CheckBox

Widget ini merupakan widget dengan dua pernyataan, *checked* atau *unchecked*. Untuk lebih jelasnya, berikut adalah contoh program sederhana.

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Please check item you choose</string>
    <string name="app_name">Demo CheckBox</string>
    <string name="check1">Item 1</string>
</resources>
```

main.xml

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

```

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
/>

<CheckBox
    android:id="@+id/cb1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/check1"
/>

</LinearLayout>

```

CbActivity.java

```

package com.latihan;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.CheckBox;
import android.widget.Toast;

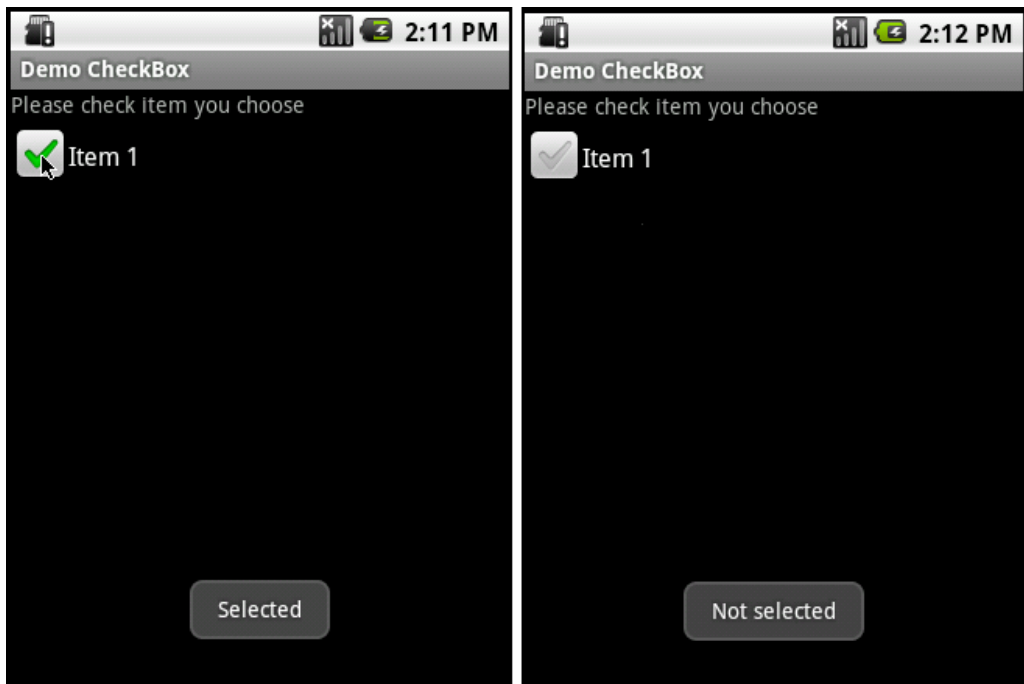
public class CbActivity extends Activity implements OnClickListener{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        CheckBox ck1 = (CheckBox) findViewById(R.id.cb1);
        ck1.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (((CheckBox) v).isChecked()) {
            Toast.makeText(CbActivity.this, "Selected",
                Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(CbActivity.this, "Not selected",
                Toast.LENGTH_SHORT).show();
        }
    }
}

```

Tampilan dari contoh program di atas adalah sebagai berikut: Ketika CheckBox dipilih, akan memunculkan pesan "Selected", ketika CheckBox di-*unchecked*, akan memunculkan pesan "Not selected"



RadioButton

Sama seperti *CheckBox*, widget ini menyediakan dua pernyataan berupa *checked* atau *unchecked*. Tetapi berbeda dengan *CheckBox*, *RadioButton* hanya bisa dipilih salah satu saja dari item yang disediakan.

Biasanya, *RadioButton* digunakan bersamaan dengan *RadioGroup*. ini digunakan untuk membatasi pilihan pada *RadioButton* sehingga hanya bisa dipilih satu saja. Lihat contoh program di bawah ini.

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Demo RadioButton</string>
    <string name="btnText">Exit</string>
    <string name="strTxt">DYAH FAJAR NUR ROHMAH</string>
    <string name="strRd1">Red</string>
    <string name="strRd2">Green</string>
    <string name="strRd3">Blue</string>
</resources>
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <RadioGroup android:id="@+id/RadioGroup01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_horizontal">
    <RadioButton
```



```

        android:text="@string/strRd1"
        android:id="@+id/rdb1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </RadioButton>

    <RadioButton
        android:text="Green"
        android:id="@+id/rdb2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </RadioButton>

    <RadioButton
        android:text="Blue"
        android:id="@+id/rdb3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </RadioButton>

</RadioGroup>

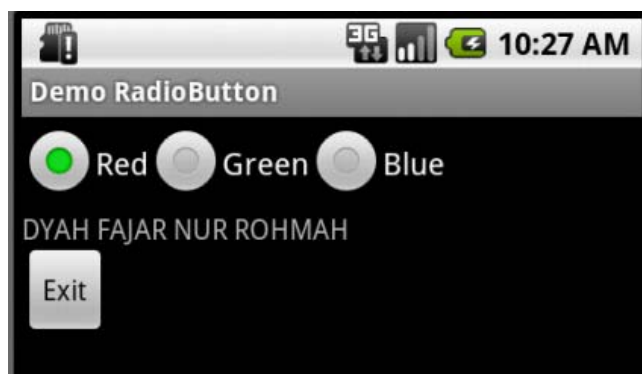
<TextView
    android:text="@string/strTxt"
    android:id="@+id/txtView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#000000">
</TextView>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnProses"
    android:text="@string/btnText">
</Button>

</LinearLayout>

```

Tampilan yang dihasilkan dari contoh program di atas adalah sebagai berikut:



CHAPTER 4 RESOURCES

Drawable Resource

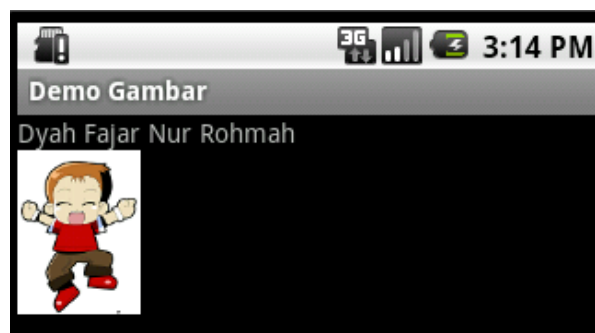
Drawable resource berupa gambar atau pun warna. *Drawable resource* yang berupa gambar harus disimpan dalam direktori `res/drawable`. Di bawah ini adalah format gambar yang bisa dijalankan pada Android:

Format gambar	Ekstensi yang disarankan
Portable Network Graphics (PNG)	.png
Nine-Patch Stretchable Images (PNG)	.9.png
Joint Photographic Experts Group (JPEG/JPG)	.jpg
Graphic Interchange Format (GIF)	.gif

Contoh di bawah ini adalah cara menampilkan gambar (sebelumnya tempelkan gambar pada direktori `res/drawable/`)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:id="@+id/txt"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
    <ImageView
        android:id="@+id/logo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/chibi_hdpi" />
</LinearLayout>
```

Maka tampilan yang nanti akan muncul adalah sebagai berikut:



XML Bitmap Drawable

File resource yang didefinisikan ke dalam *file* XML yang mengarahkan ke suatu *file* bitmap (berupa *file* .jpg, .png, atau .gif). Dengan menggunakan XML ini, kita bisa menentukan sifat-sifat tambahan untuk *file* bitmap.

Contoh Program:

Buat *file* XML dengan tipe bitmap, kemudian lengkapi sintaks menjadi seperti di bawah ini:

```
<?xml version="1.0" encoding="utf-8"?>
<bitmap xmlns:android="http://schemas.android.com/apk/res/android"
    android:src="@drawable/color"
    android:tileMode="repeat">

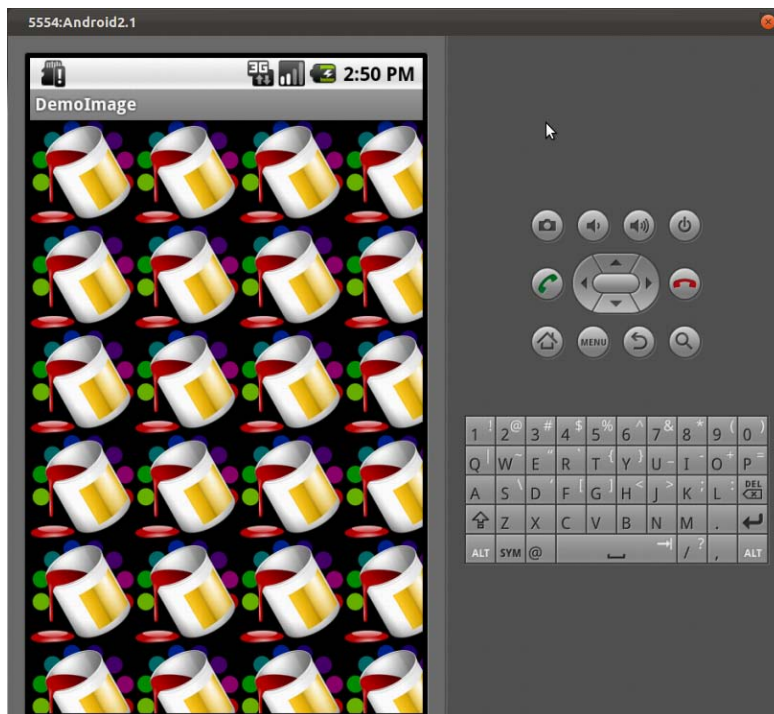
</bitmap>
```

Pada *file* *Layout*, lengkapi sintaks menjadi seperti ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="@drawable/demobitmap" >

</LinearLayout>
```

Maka tampilan yang dihasilkan akan menjadi seperti ini:



Shape Drawable

Shape drawable ini adalah file XML yang menggambarkan bentuk geometri, termasuk warna dan gradiasi. File yang akan dihasilkan nantinya akan masuk ke dalam direktori *drawable*.

Element `<shape>` memiliki beberapa atribut, salah satunya adalah `android:shape`. Atribut ini digunakan untuk menentukan bentuk geometri apa yang akan ditampilkan. Value dari atribut ini adalah *rectangle*, *oval*, *line*, *ring*.







Contoh program:

Buat file XML baru dengan tipe *Drawable*, kemudian pilih bentuk *shape*, beri nama file tersebut dengan *gradientimage.xml*. Isikan program sebagai berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <gradient
        android:startColor="#FFFFFF"
        android:endColor="#0000FF"
        android:angle="45" />
    <padding android:left="7dp"
        android:top="7dp"
        android:right="7dp"
        android:bottom="7dp" />
    <corners android:radius="8dp" />
</shape>
```

Catatan

Ada beberapa elemen di bawah elemen `<shape>` yang dipakai, di antaranya:

-  `<gradient>` : elemen ini digunakan untuk menentukan warna gradasi yang akan dipakai oleh bentuk geometri.
-  `<corners>` : elemen ini hanya bisa digunakan jika bentuk geometrinya adalah *rectangle*. Elemen ini akan menentukan kebulatan sudut suatu bentuk geometri.
-  `<padding>` : elemen ini digunakan untuk memberikan *padding* atau lapisan jarak pada elemen view.
-  `<size>` : elemen ini digunakan untuk menentukan ukuran bentuk geometri.
-  `<solid>` : memberikan warna menyeluruh pada bentuk. Atribut untuk elemen ini hanya `android:color`.
-  `<stroke>` : untuk menentukan ketebalan garis.

Kemudian, pada file *layout/main.xml*, implementasikan file shape tersebut kedalam *TextView* sebagai *background*.

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

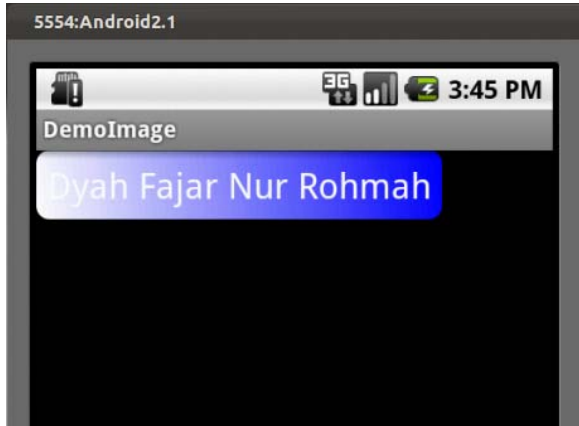
    <TextView android:id="@+id/txt"
        android:layout_width="fill_parent"
```

```

        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:background="@drawable/gradientimage" />
</LinearLayout>

```

Tampilan yang akan dihasilkan:



State List Drawable

Drawable jenis ini, didefinisikan dalam bentuk *file* XML yang menggunakan beberapa gambar untuk satu grafik yang sama, tergantung dari kondisi objek tersebut. sebagai contoh perbedaan gambar tombol saat ditekan dan saat biasa.

Contoh Program:

Buat file XML *drawable* dengan tipe *selector*. Lengkapi program menjadi seperti di bawah ini:

imagebutton.xml

```

<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:state_pressed="true"
        android:drawable="@drawable/button_pressed"></item>
    <item android:drawable="@drawable/button_normal"></item>
</selector>

```

Kemudian pada *file* XML Layout lengkapi program menjadi seperti di bawah ini:

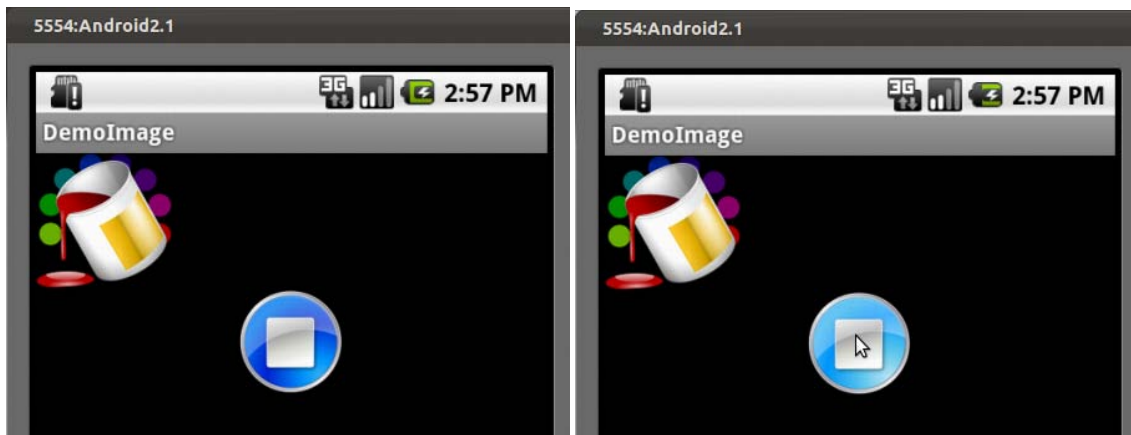
```

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

    <Button android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/imagebutton"></Button>

```

Maka tampilan yang akan dihasilkan adalah sebagai berikut:



String Resource

Resource ini akan menampung sebuah string untuk aplikasi yang membutuhkan sebuah teks untuk ditampilkan. Untuk menambahkan *resource* tipe string ini menggunakan *elemen* `<string>` pada file *resource* yang berada di direktori `/res/values/strings.xml`

Ada beberapa tipe string yang disediakan dengan format yang berbeda, yaitu:

Plain String

Plain string dapat disimpan didalam sebuah *file* XML yang berada di bawah direktori `res/values`. *File* tersebut akan terisi sebuah *root element* yaitu `<resources>` dengan beberapa *child element* yang salah satunya digunakan untuk menyimpan nilai string, yaitu `<string>`.

Contoh di bawah ini adalah penyimpanan teks untuk tombol.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Demo String Resource</string>
    <string name="strBtn1">Tampilkan</string>
    <string name="strBtn2">Exit</string>
</resources>
```

Di bawah ini adalah beberapa contoh format penulisan nilai string dan tampilannya:

String value	Hasil yang ditampilkan
<code>string name="sample_1">Hello, World</string></code>	Hello World
<code>string name="sample_2">"Hello, World"</string></code>	Hello, World
<code>string name="sample_3">This\'ll work</string></code>	This'll work
<code>string name="sample_4">He said, \"No\"</string></code>	He said, "No"

Format String


Resource String juga dapat memuat format string seperti %d, %s, %f, dan menampilkannya dengan menggunakan `String.format(String, object)`.

Perhatikan penggunaan format string dalam contoh di bawah ini:

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">DemoStringColor</string>
    <string name="strText">Jari-jari</string>
    <string name="strBtn">Hitung</string>
    <string name="strHasil">Keliling lingkaran %.2f</string>
</resources>
```

Contoh penulisan
format string



main.xml

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

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/strText" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="text" >
    </EditText>

    <Button
        android:id="@+id/btnHitung"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/strBtn" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:visibility="invisible" />

</LinearLayout>
```

DemoStringColorActivity.java

```

public class DemoStringColorActivity extends Activity {
    Button btnHitung;
    TextView txtHasil;
    EditText editJari;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btnHitung = (Button) findViewById(R.id.btnHitung);
        editJari = (EditText) findViewById(R.id.editText1);
        txtHasil = (TextView) findViewById(R.id.textView2);

        btnHitung.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                onHitung();
            }
        });

        public void onHitung(){
            String jari = editJari.getText().toString();
            int r = Integer.parseInt(jari);
            double hasil = 2 * 3.14 * r;

            Resources res = getResources();
            String keliling = String.format(res.getString(R.string.strHasil),
hasil);

            txtHasil.setText(keliling);
            txtHasil.setVisibility(0);
        }
    }
}

```

Catatan

Nilai konstanta untuk *method* `setVisibility(int status)` ada 3, seperti di bawah ini:

Constant	Value	Description
Visible	0	Menampilkan pada layar
Invisible	1	Tidak ditampilkan,
Gone	2	Disembunyikan seolah tidak pernah ditambahkan di dalam layout

Maka, tampilan yang akan ditampilkan adalah sebagai berikut:



Nilai double/desimal dengan 2 angka dibelakang koma (%.2f)

Color

Resource jenis ini akan menyimpan warna dalam beberapa style value yang bisa digunakan, yaitu:

- 🚦 #RGB
- 🚦 #ARGB
- 🚦 #RRGGBB
- 🚦 #AARRGGBB

Nilai setiap warna akan tersimpan di dalam direktori `res/values/` yang umumnya berada dalam *file* `strings.xml`. *Child element* yang digunakan adalah `<color>`. Perhatikan penggunaan *resource color* di bawah ini:

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Demo RadioButton</string>
    <string name="btnText">Exit</string>
    <string name="strTxt">DYAH FAJAR NUR ROHMAH</string>
    <color name="backBlue">#0000FF</color>
    <color name="backRed">#FF0000</color>
    <color name="backGreen">#00FF00</color>
</resources>
```

main.xml

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

    <RadioGroup android:id="@+id/RadioGroup01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
```

```

        android:orientation="horizontal"
        android:gravity="center_horizontal">

<RadioButton
    android:text="Red"
    android:id="@+id/rdb1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</RadioButton>

<RadioButton
    android:text="Green"
    android:id="@+id/rdb2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</RadioButton>

<RadioButton
    android:text="Blue"
    android:id="@+id/rdb3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</RadioButton>

</RadioGroup>

<TextView
    android:text="@string/strTxt"
    android:id="@+id/txtView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#000000">
</TextView>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/btnProses"
    android:text="@string/btnText">
</Button>
</LinearLayout>

```

RdActivity.java

```

package com.latihan;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.TextView;

public class RdActivity extends Activity implements OnClickListener{
    RadioButton rdb1;
    RadioButton rdb2;
    RadioButton rdb3;
    TextView txt;
    Button btn;

```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    rd1 = (RadioButton) findViewById(R.id.rdb1);
    rd2 = (RadioButton) findViewById(R.id.rdb2);
    rd3 = (RadioButton) findViewById(R.id.rdb3);
    txt = (TextView) findViewById(R.id.txtView);
    btn = (Button) findViewById(R.id.btnProses);

    rd1.setOnClickListener(this);
    rd2.setOnClickListener(this);
    rd3.setOnClickListener(this);
    btn.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    if (v == btn){
        finish();
    } else if (v == rd1){
        txt.setBackgroundResource(R.color.backRed);
    } else if (v == rd2){
        txt.setBackgroundResource(R.color.backGreen);
    } else if (v == rd3){
        txt.setBackgroundResource(R.color.backBlue);
    }
}
}

```

Tampilan yang dihasilkan dari contoh program di atas adalah sebagai berikut:



```
setBackgroundResource(int color)
```

Digunakan untuk mengatur background pada TextView. Biasanya *value* background ini didefinisikan pada file **strings.xml**, dengan nama elemen

Catatan

adalah `drawable` atau `color`. Method ini berelasi dengan *Attribute Name* `android:background`.

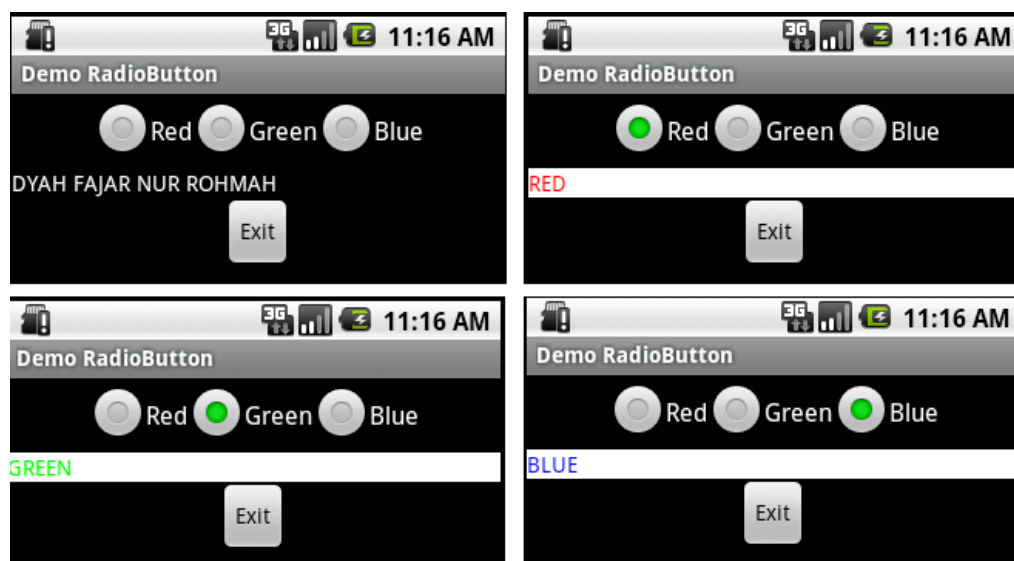
Latihan

Pada project RadioButton di atas, modifikasi program sehingga:

1. Ketika user memilih salah satu pilihan (Red, Green, Blue), selain warna background TextView yang berubah, Text pada TextView pun berubah sesuai apa yang dipilih. Berikut tampilannya.



2. Jika latihan pertama berhasil, coba lakukan modifikasi lagi, sehingga, background TextView berwarna putih, dan text color dari tulisan yang berubah sesuai dengan yang dipilih. (Gunakan method `setTextColor(getResources().getColor(resource_id))` untuk mengubah warna. Pendefinisian warna, bisa ditambahkan dengan elemen `color` pada `strings.xml`, bisa juga menggunakan elemen `drawable` yang sebelumnya sudah dibuat). Tampilan akan menjadi seperti di bawah ini:



Catatan

```
setTextColor(int color), setTextColor(ColorStateList colors)
```

Digunakan untuk mengatur warna dari text pada TextView. Biasanya *value* warna ini didefinisikan pada file **strings.xml**, yang umumnya menggunakan elemen **color**. Method ini berelasi dengan *Attribute Name* **android:textColor**.

CHAPTER 5 LAYOUT

Tata letak tampilan dibagi menjadi beberapa macam. Di antaranya adalah yang sering dipakai yaitu Linear Layout, Relative Layout, List View dan Scroll View.

Linear Layout

Linear layout akan menampilkan elemen-elemen View secara garis lurus, baik vertical ataupun horizontal.

Pada contoh `main.xml` di bawah ini, kita akan membuat view tombol akan saling menyamping dengan menambahkan `LinearLayout` di dalamnya, (sintaks dalam *box* merah adalah sintaks yang perlu ditambahkan):

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

    <TextView
        android:text="@string/title"
        android:id="@+id/txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20dp">
    </TextView>

    <EditText
        android:text=""
        android:hint="Type here"
        android:id="@+id/txtEdit"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
    </EditText>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        >

    <Button
        android:text="@string/strBtn1"
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </Button>
```

```

<Button
    android:text="@string/strBtn2"
    android:id="@+id/btn2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Button>

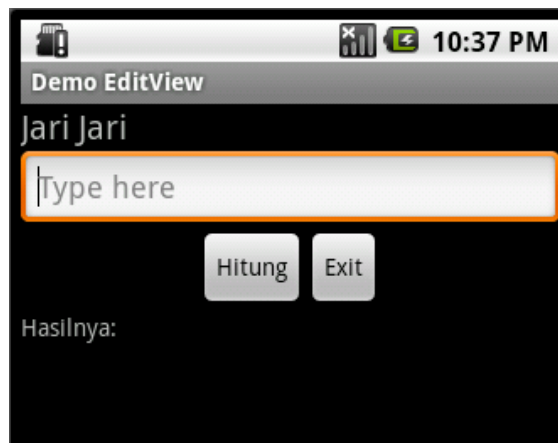
</LinearLayout>

<TextView
    android:text="@string/strHasil"
    android:id="@+id/txtTampil"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    >
</TextView>

</LinearLayout>

```

Maka tampilan yang dihasilkan akan menjadi seperti di bawah ini:



Catatan

Menggunakan terlalu banyak *Nested layout* akan membuat *performance* aplikasi semakin lambat. Semakin banyak *layout* yang dibuat, semakin lama pula aplikasi untuk me-render tampilan.

Cara yang lebih baik untuk membuat tata letak tidak "monoton" adalah dengan menggunakan *RelativeLayout*.

Relative Layout

RelativeLayout memungkinkan pengguna menyusun tata letak secara lebih leluasa. Posisi setiap *widget* bisa diatur relatif pada *widget* yang lainnya (dibawah, atau disamping *widget* sebelumnya). *RelativeLayout* adalah cara terbaik untuk mendesain suatu *interface*, karena dengan ini kita bisa mengurangi *nested ViewGroup* (yang sering terjadi adalah *nested LinearLayout*).

```

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

<RelativeLayout android:id="@+id/RelativeLayout01"

```

```

        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        xmlns:android="http://schemas.android.com/apk/res/android">

        <TextView android:id="@+id/txtNama"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dyah Fajar"
            android:textSize="24dp" />

        <TextView android:id="@+id/txtNumber"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="0898-756-4321"
            android:layout_below="@+id/txtNama" />

        <ImageView android:id="@+id/image"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_launcher"
            android:layout_alignParentRight="true" />

        <Button android:id="@+id/btnEdit"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Ubah"
            android:layout_below="@+id/txtNumber"
            android:layout_marginTop="15dp"
            />

        <Button android:id="@+id/btnCall"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Call"
            android:layout_below="@+id/txtNumber"
            android:layout_toRightOf="@+id/btnEdit"
            android:layout_marginTop="15dp" />

    </RelativeLayout>

```

Maka tampilan yang akan dihasilkan adalah sebagai berikut:

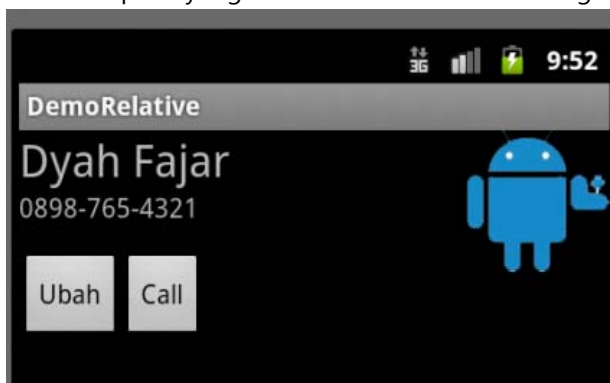


Table Layout

Yang perlu dipahami dalam pembuatan *layout* dengan *TableLayout* adalah bagaimana *widget* atau *View* bekerja dengan baris dan kolom. Dan bagaimana mengatasi sebuah *widget* atau *view* yang memakai lebih dari satu kolom untuk penampilmnya.

Banyaknya baris ditentukan oleh *developer* dengan memasukan *widget* atau *view* sebagai elemen anak dari elemen `<TableRow>`. Dengan demikian, *developer* mengontrol secara langsung banyaknya baris yang diinginkan.

Banyaknya kolom, akan ditentukan oleh Androidnya sendiri, dan para *developer* mengontrol banyaknya kolom secara tidak langsung. Paling sedikit, satu kolom untuk setiap *widget*. Sebagai contoh, jika kita punya 3 baris, baris 1 memiliki tiga *widget*; baris 2 memiliki dua *widget*; dan baris 3 memiliki empat *widget*, maka, paling sedikit akan ada empat kolom disana. Tetapi sebuah *widget* juga bisa memakai lebih dari satu kolom dengan menggunakan atribut `layout_span`.

Berikut adalah contoh program:

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="1" >

    <TableRow>
        <TextView android:text="URL:" />

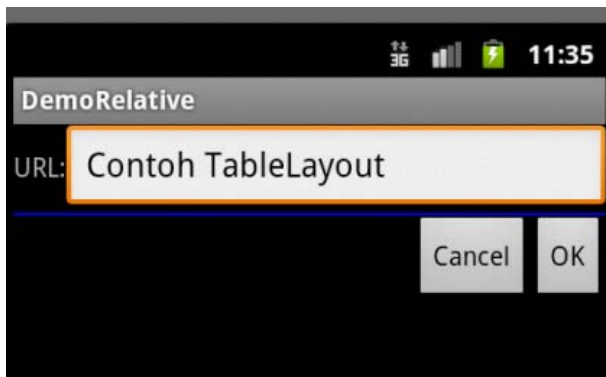
        <EditText
            android:id="@+id/entry"
            android:layout_span="3" />
    </TableRow>

    <View
        android:layout_height="2px"
        android:background="#0000FF" />

    <TableRow>
        <Button
            android:id="@+id/cancel"
            android:layout_column="2"
            android:text="Cancel" />

        <Button
            android:id="@+id/ok"
            android:text="OK" />
    </TableRow>
</TableLayout>
```

Maka tampilan yang akan dihasilkan adalah sebagai berikut:



Scroll View

ScrollView memungkinkan para *developer* membuat tampilan layout lebih panjang dari *space* layar. Dengan menggunakan *ScrollView*, sebagian informasi akan muncul dalam satu waktu, sisanya akan muncul jika pengguna melakukan *scroll* ke atas atau ke bawah.

Berikut ini adalah contoh penggunaan *ScrollView*:

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

    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:stretchColumns="0" >

        <TableRow>
            <View
                android:layout_height="80px"
                android:background="#000000" />

            <TextView
                android:layout_gravity="center_vertical"
                android:paddingLeft="4px"
                android:text="#000000" />
        </TableRow>

        <TableRow>
            <View
                android:layout_height="80px"
                android:background="#cccccc" />

            <TextView
                android:layout_gravity="center_vertical"
                android:paddingLeft="4px"
                android:text="#cccccc" />
        </TableRow>

        <TableRow>
            <View
                android:layout_height="80px"
                android:background="#d44ec7" />
```

```

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#d44ec7" />
    </TableRow>

    <TableRow>
        <View
            android:layout_height="80px"
            android:background="#844eff" />

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#844eff" />
    </TableRow>

    <TableRow>
        <View
            android:layout_height="80px"
            android:background="#045f11" />

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#045f11" />
    </TableRow>

    <TableRow>
        <View
            android:layout_height="80px"
            android:background="#440000" />

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#440000" />
    </TableRow>

    <TableRow>
        <View
            android:layout_height="80px"
            android:background="#884400" />

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#884400" />
    </TableRow>

    <TableRow>
        <View
            android:layout_height="80px"
            android:background="#aa8844" />

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#aa8844" />
    </TableRow>

```

```

<TableRow>
    <View
        android:layout_height="80px"
        android:background="#ffaa88" />

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#ffaa88" />
</TableRow>

<TableRow>
    <View
        android:layout_height="80px"
        android:background="#ffffaa" />

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#ffffaa" />
</TableRow>

<TableRow>
    <View
        android:layout_height="80px"
        android:background="#ffffff" />

        <TextView
            android:layout_gravity="center_vertical"
            android:paddingLeft="4px"
            android:text="#ffffff" />
</TableRow>
</TableLayout>

</ScrollView>

```

Maka tampilan yang akan dihasilkan adalah sebagai berikut:





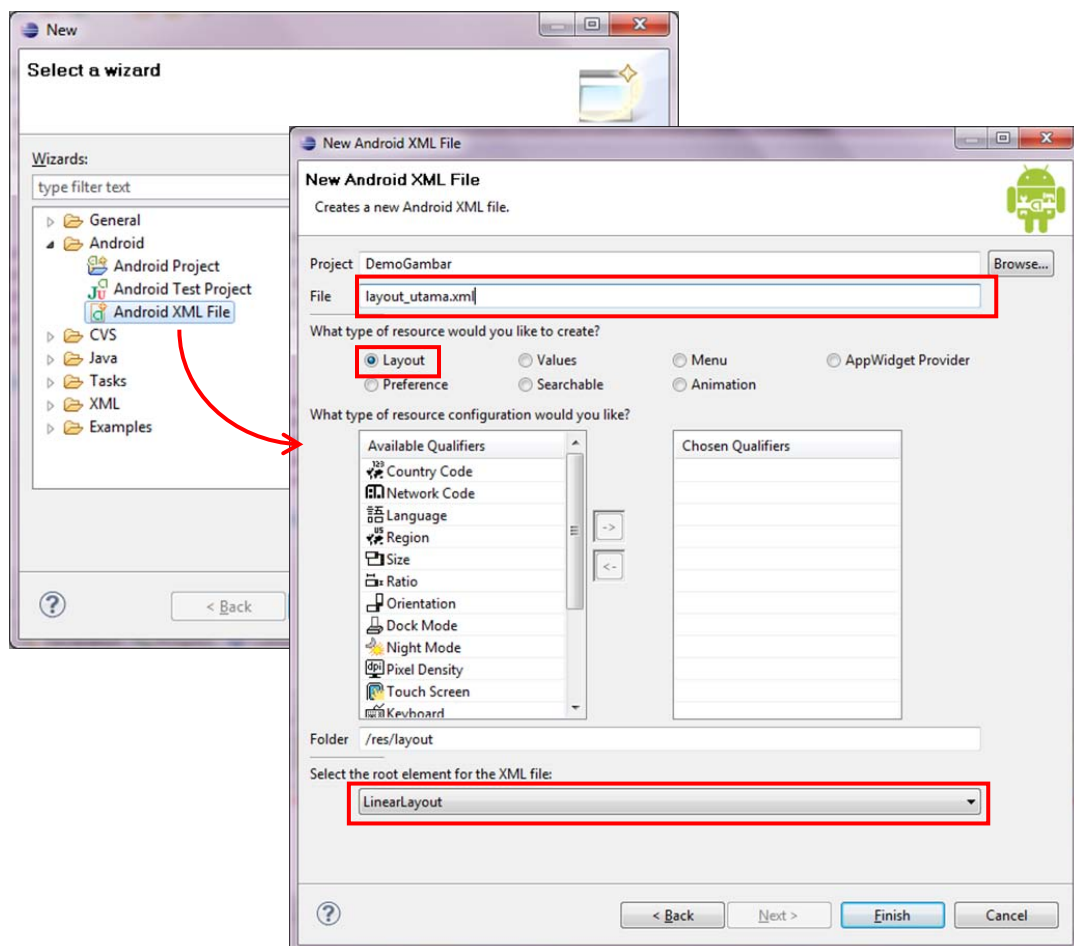
CHAPTER 6 MULTI SCREEN

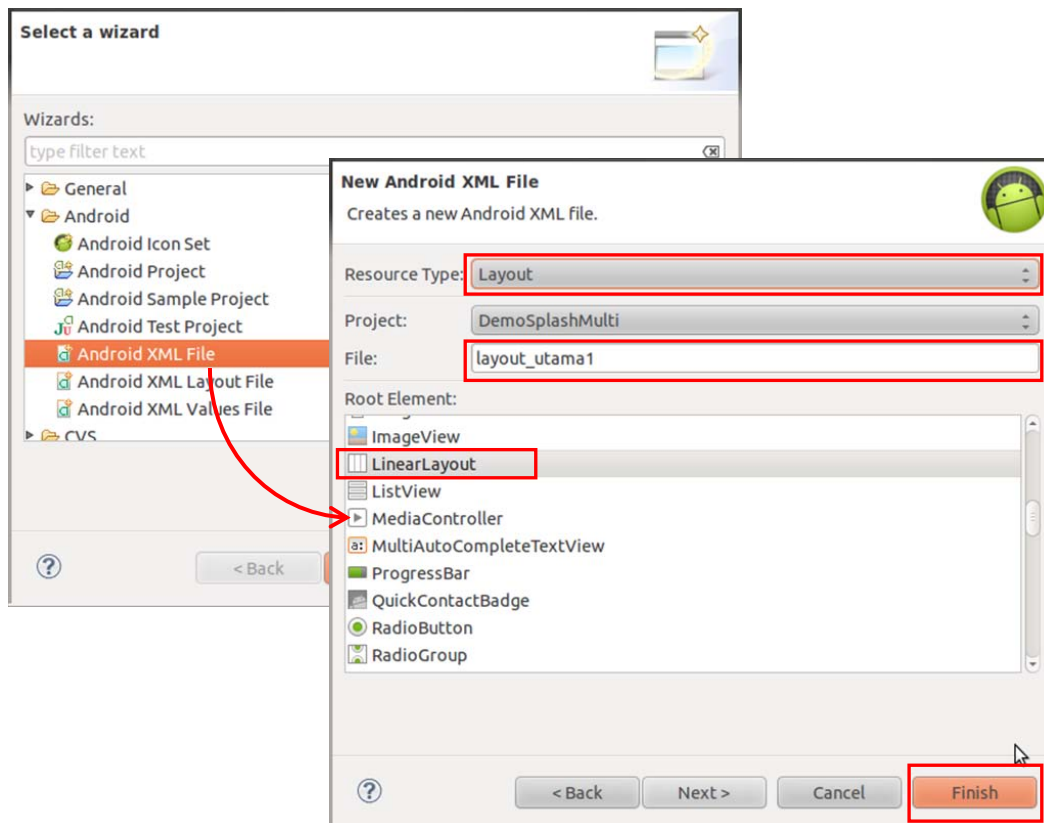
Terkadang, atau sebagian besar aplikasi yang dikembangkan memiliki lebih dari satu *screen*. Tampilan, seperti yang sudah diketahui, dibuat dengan menggunakan XML. Dengan demikian kita akan memiliki lebih dari satu *file* XML berupa *layout*.

Dan yang perlu diperhatikan, setiap *layout* yang kita buat, harus memiliki satu *activity*. Artinya, jika kita akan membuat 2 *layout*, maka di dalam *project* kita akan ada 2 *activity* untuk masing-masing *layout* tersebut.

Bagaimana kita menggabungkannya? Berikut contoh sederhana aplikasi android dengan *multi screen*.

1. Yang pertama dilakukan adalah membuat halaman utama setelah *Splash Screen* dijalankan. Buat *file layout* baru dengan nama ***layout_utama.xml*** dengan tipe *LinearLayout* (Klik kanan pada *project*, pilih **New > Android XML File**).





Lalu isikan *file* tersebut menjadi seperti di bawah ini:

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

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/Judul"
        android:textSize="24dp"
        android:layout_marginBottom="15dp"
        />

    <Button
        android:text="@string/btnKeluar"
        android:id="@+id/Button01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center">
    </Button>

</LinearLayout>
```

Pastikan pada *strings.xml*, id string tersebut sudah didefinisikan

2. Setiap *layout* harus memiliki *Activity*nya sendiri, maka dari itu, buat satu *class* java baru dengan nama **UtamaActivity.java**, dan isikan sintaks programnya menjadi seperti di bawah ini:

```

package com.dyah.latihan;

import android.app.Activity;
import android.os.Bundle;

public class UtamaActivity extends Activity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_utama);
    }
}

```

Arahkan pada layout yang sudah dibuat sebelumnya

3. Tambahkan Activity yang baru kita buat ke dalam AndroidManifest.xml, supaya Activity tersebut dikenali.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.dyah.latihan"
android:versionCode="1"
android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="10" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".SplashScreenActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
            />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name="UtamaActivity"></activity>

    </application>
</manifest>

```

Tambahkan di sini

4. Kemudian buat halaman *splash screen* pada file **main.xml** (Sebagai contoh di bawah ini, *layout splash screen* menggunakan *ImageView* dan *ProgressBar*), dan tambahkan sintaks program untuk Activitynya menjadi seperti di bawah ini:

main.xml


```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/cake"
        android:layout_marginBottom="15dp" />

    <ProgressBar
        android:id="@+id/progressBar1"
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>

```

SplashScreenActivity.java

```

package com.dyah.latihan;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

public class SplashScreenActivity extends Activity {
    /** Called when the activity is first created. */
    private long ms = 0;
    private long splashTime = 3000;
    private boolean splashActive = true;
    private boolean splashPause = false;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Thread th = new Thread(){
            public void run(){
                try {
                    while (splashActive && ms < splashTime) {
                        if (!splashPause){
                            ms = ms +100;
                        }
                        sleep(100);
                    }
                }
            }
        };
    }
}

```

```

        } catch (Exception e) {
            // TODO: handle exception
        } finally {
            Intent i = new Intent(SplashScreenActivity.this,
                UtamaActivity.class);
            startActivity(i);
        }
    };
    th.start();
}
}

```

Activity kedua
yang akan diakses
setelah splash
screen selesai
dijalankan

CHAPTER 7 LOCATION BASED SERVICES

Class `MapView` sering dijadikan sebagai sebuah *external library* yang sangat berguna dalam pembuatan aplikasi yang berbasis lokasi (*Location Based App*), karena *class* tersebut dengan mudah memberikan integrasi antara aplikasi yang dibuat dengan *Google Maps*.

Karena `MapView` memberikan akses langsung pada data *Google Maps*, maka kita perlu registrasi ke *Google Maps* untuk mematuhi syarat-syarat yang berlaku, sebelum kita bisa mengambil data dari *Google Maps*.

Mendaftar Map API Key

Untuk mendaftar Map API Key, langkah pertama yang dilakukan adalah kita harus mendapatkan *MD5 Fingerprint* dari sertifikat atau *key* yang akan digunakan pada aplikasi. Sebelum mengakses halaman registrasi, gunakan *Keytool* untuk *men-generate fingerprint*.

Untuk *men-generate MD5 Fingerprint*, pastikan *path* atau lokasi dari *debug.keystore*. Standarnya, *debug.keystore* akan berlokasi pada:

Windows Vista/ Windows 7 : `C:\Users\<user>\.android\debug.keystore`

Windows XP : `C:\Documents and Settings\<user>\.android\debug.keystore`

Linux : `~/.android/debug.keystore`

Jika kita menggunakan Eclipse/ADT dan ragu dimana lokasi *debug.keystore* berada, kita bisa melihatnya pada **Windows > Preference > Android > Build** untuk melihat *path* lengkapnya.

Jika sudah langkah selanjutnya adalah *men-generate fingerprint* dengan menggunakan *console terminal* atau *command prompt* seperti di bawah ini:

Windows User

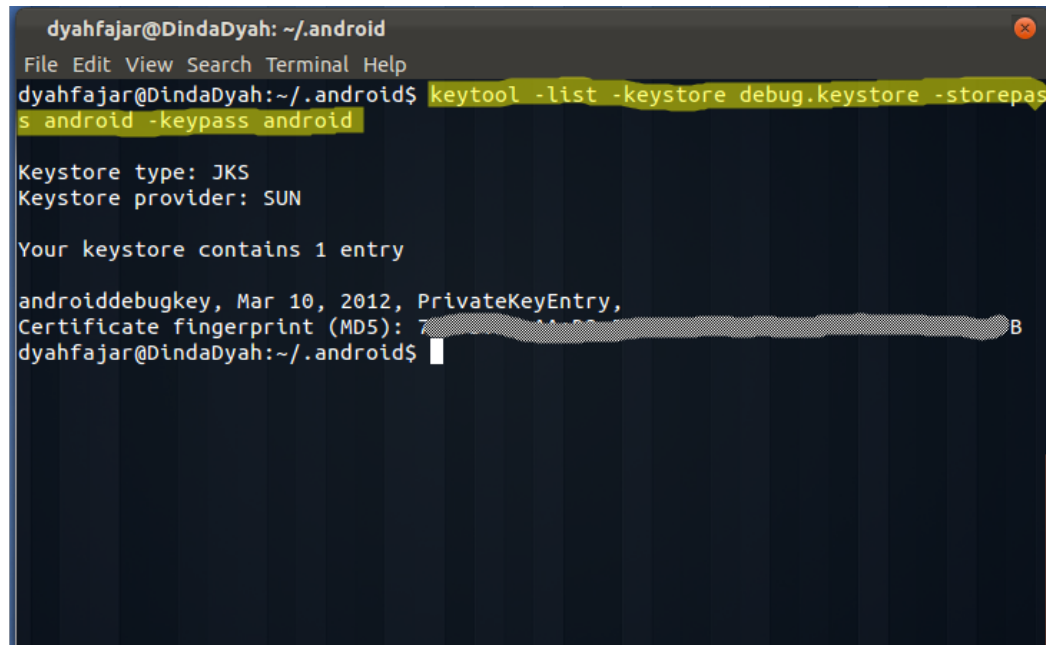
1. Pada *command prompt* ketikkan perintah di bawah ini:

```
Keytool -list -keystore <path_debug_keystore> -storepass android -  
keypass android
```


Linux User

1. Pada *konsole terminal*, masuklah ke dalam folder `.android`, kemudian ketikkan perintah di bawah ini:

```
Keytool -list -keystore <path_debug_keystore> -storepass android -  
keypass android
```



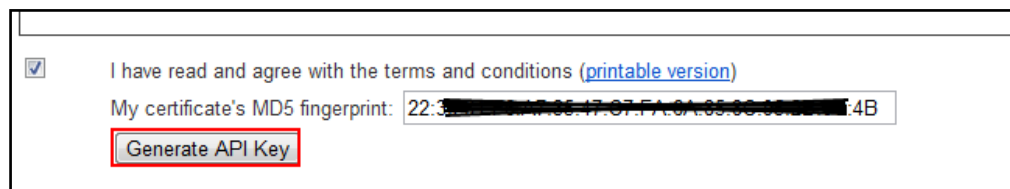
The screenshot shows a terminal window with the prompt `dyahfajar@DindaDyah: ~/.android`. The command `keytool -list -keystore debug.keystore -storepass android -keypass android` has been entered and executed. The output shows the keystore type is JKS, the provider is SUN, and it contains one entry: `androiddebugkey, Mar 10, 2012, PrivateKeyEntry, Certificate fingerprint (MD5): 7...B`. The fingerprint value is partially obscured by a greyed-out area.

2. *Sign up* dengan mengakses halaman:

<https://developers.google.com/maps/documentation/android/maps-api-signup>

pastikan kita sudah memiliki akun google, jika belum, bisa langsung *sign up* pada halaman tersebut.

3. Masukkan *certificate fingerprint* yang sudah kita *generate* ke dalam *field* yang ada pada laman situs, lalu tekan tombol **Generate API Key**.



The screenshot shows a web form with a checked checkbox and the text "I have read and agree with the terms and conditions ([printable version](#))". Below this, there is a label "My certificate's MD5 fingerprint:" followed by a text input field containing the value "22:3...4B". A red box highlights the "Generate API Key" button.

- Thank you for signing up for an Android Maps API key!**
- Your key is:
- ```
Oj_nqt08-ep-g1DnXsl-wak_g-FHfPbaGqHyWVtWA
```
- This key is good for all apps signed with your certificate whose fingerprint is:
- ```
22:7F:0E:2A:C9:D6:-35:F0:0A:-FA:A0:FE:E6:CE:B3:0B:09:4B
```
- Here is an example xml layout to get you started on your way to mapping glory:
- ```
<com.google.android.maps.MapView
 android:layout_width="fill_parent"
 android:layout_height="fill_parent"
 android:apiKey="Oj_nqt08-ep-g1DnXsl-wak_g-FHfPbaGqHyWVtWA"
/>
```
- Check out the [API documentation](#) for more information.

1. Buatlah project android dengan nama DemoMap dengan *Build Targetnya* adalah *Google APIs*.



2. Karena kita akan menggunakan *library* Google Maps yang bukan *library* standar Android, maka kita harus mendeklarasikan *library* tersebut dalam AndroidManifest.xml sebagai *child element* dari elemen `<application>`.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
 package="andro.latihan"
 android:versionCode="1"
 android:versionName="1.0" >

 <uses-sdk android:minSdkVersion="10" />

 <application
 android:icon="@drawable/ic_launcher"
 android:label="@string/app_name" >

 <!-- tambahkan library map ke dalam android -->
 <uses-library android:name="com.google.android.maps" />

 <activity
 android:name=".DemoMapActivity"
 android:label="@string/app_name" >
 <intent-filter>
 <action android:name="android.intent.action.MAIN"
 />

 <category
 android:name="android.intent.category.LAUNCHER" />
 </intent-filter>
 </activity>
 </application>
</manifest>
```

Tambahkan di sini

3. Kita juga butuh untuk mengakses internet terkait dengan *retrieving data google*, maka kita perlu menambahkan *permission* untuk mengakses internet, yang akan diatur didalam AndroidManifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
 package="andro.latihan"
 android:versionCode="1"
 android:versionName="1.0" >

 <uses-sdk android:minSdkVersion="10" />
 <!-- tambahkan permission INTERNET ke dalam android -->
 <uses-permission android:name="android.permission.INTERNET"/>

 <application
 android:icon="@drawable/ic_launcher"
 android:label="@string/app_name" >

 <!-- tambahkan library map ke dalam android -->
 <uses-library android:name="com.google.android.maps" />

 <activity
```

Tambahkan di sini

```

 android:name=".DemoMapActivity"
 android:label="@string/app_name" >
 <intent-filter>
 <action android:name="android.intent.action.MAIN"

/>

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

</manifest>

```

4. Pada main.xml, ubah sintaks program menjadi seperti di bawah ini:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
 android:id="@+id/mainlayout"
 android:orientation="vertical"
 android:layout_width="fill_parent"
 android:layout_height="fill_parent" >

 <com.google.android.maps.MapView
 android:id="@+id/mapview"
 android:layout_width="fill_parent"
 android:layout_height="fill_parent"
 android:clickable="true"
 android:enabled="true"
 android:apiKey="your_map_API_key"

 />

</RelativeLayout>

```

5. Pada activitynya, ubah sintaks program menjadi seperti di bawah ini:

```

package andro.latihan;

import java.util.List;
import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.OverlayItem;

import android.graphics.drawable.Drawable;
import android.os.Bundle;

public class DemoMapActivity extends Activity {
 // Ubah menjadi MapActivity
 @Override
 public void onCreate(Bundle savedInstanceState){
 super.onCreate(savedInstanceState);
 setContentView(R.layout.main);

 // init peta
 MapView mapView = (MapView) findViewById(R.id.mapview);
 mapView.setBuiltInZoomControls(true);
 }
}

```



```

 }

 @Override
 protected boolean isRouteDisplayed() {
 // TODO Auto-generated method stub
 return false;
 }
}

```

Akan bertambah satu method baru

6. Jalankan project, maka hasilnya akan menjadi seperti ini:



## CHAPTER 8 DATABASE

### SQLite

Aplikasi android yang menggunakan SQLite sebagai sarana penyimpanan data, tidak akan lepas dengan salah satu *public class abstract* yaitu `SQLiteOpenHelper`. *Class* tersebut akan membantu dalam mengatur pembuatan database dan manajemen versi database.

Contoh proyek aplikasi android dengan menggunakan SQLite:

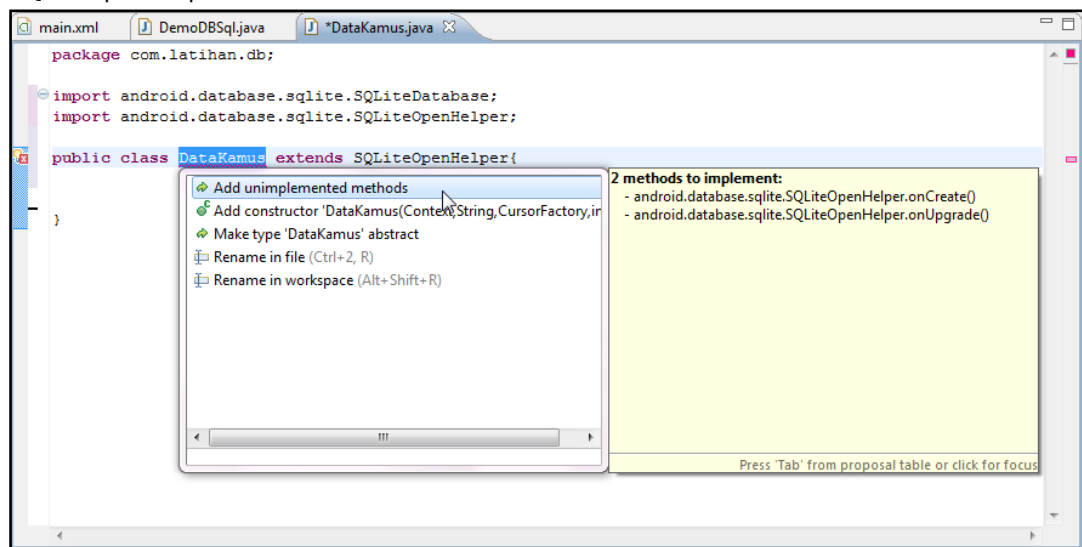
1. Buat proyek dengan nama DemoDBSql.
2. Buatlah class baru yang digunakan untuk pembuatan struktur database dengan nama `DataKamus.java`, lalu tambahkan *extends* ke `SQLiteOpenHelper`:

```
package com.latihan.db;

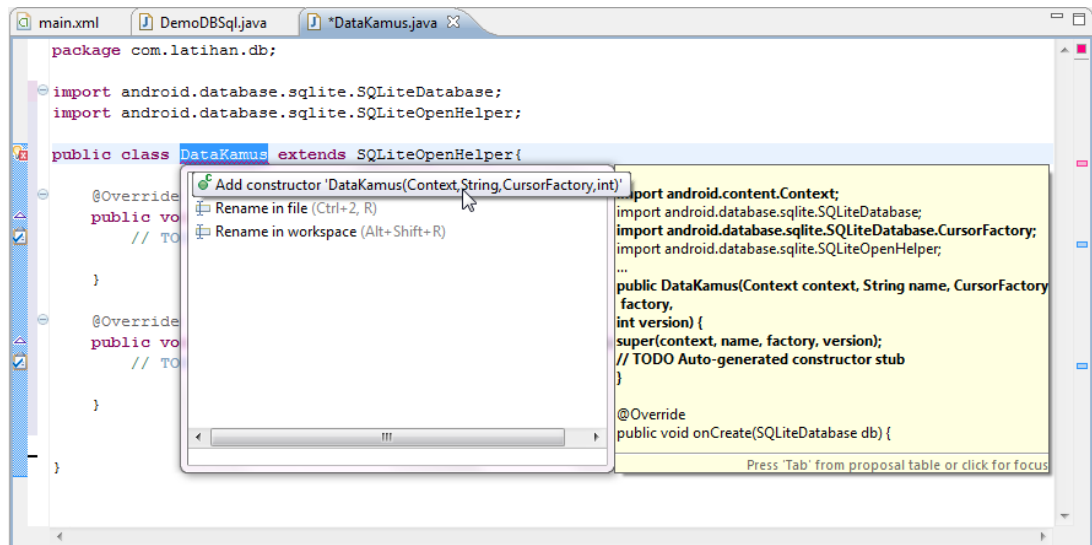
public class DataKamus extends SQLiteOpenHelper {

}
```

Maka, akan muncul *warning* untuk menambahkan *method* yang ada dalam `SQLiteOpenHelper`.



Akan muncul lagi *warning* untuk menambahkan *constructor* yang sesuai dengan `SQLiteOpenHelper`



Kemudian sesuaikan sintaks hingga menjadi seperti ini:

```
package com.latihan.db;

import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DataKamus extends SQLiteOpenHelper{

 private static final String DB_NAME = "dbkamus";
 public static final String INGGRIS = "inggris";
 public static final String INDONESIA = "indonesia";

 public DataKamus(Context context, String name, CursorFactory
factory, int version) {

 super(context, DB_NAME, null, 1);
 // TODO Auto-generated constructor stub
 }

 @Override
 public void onCreate(SQLiteDatabase db) {
 // TODO Auto-generated method stub
 }

 @Override
 public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
 // TODO Auto-generated method stub
 }

 public void createTable(SQLiteDatabase db){
```

Hapus parameter lainnya  
hingga hanya mengisakan  
parameter Context saja

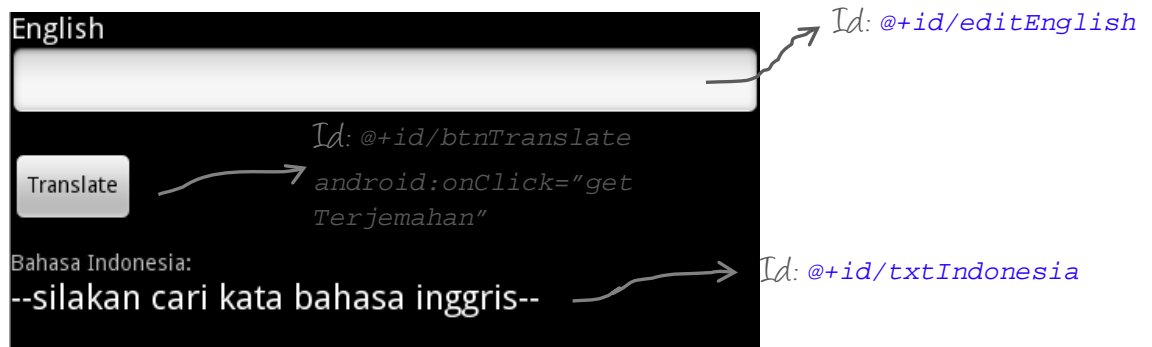
```

 db.execSQL("DROP TABLE IF EXISTS kamus");
 db.execSQL("CREATE TABLE if not exists kamus (id INTEGER PRIMARY KEY AUTOINCREMENT, inggris TEXT, indonesia TEXT);");
 }

 public void generateData (SQLiteDatabase db){
 ContentValues cv = new ContentValues();
 cv.put(INGGRIS, "run");
 cv.put(INDONESIA, "lari");
 db.insert("kamus", INGGRIS, cv);
 cv.put(INGGRIS, "read");
 cv.put(INDONESIA, "baca");
 db.insert("kamus", INGGRIS, cv);
 }
}

```

3. Lengkapi file main.xml hingga membentuk tampilan seperti ini:



4. Lengkapi file Activity menjadi seperti di bawah ini:

```

package com.latihan.db;

import android.app.Activity;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class DemoDBSql extends Activity {
 private SQLiteDatabase db = null;
 private Cursor kamusCursor = null;
 private DataKamus dKamus = null;
 public static final String INDONESIA = "indonesia";
 public static final String INGGRIS = "inggris";
}

```

```

 EditText editInggris;
 TextView txtIndo;
 /** Called when the activity is first created. */
 @Override
 public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);

 dKamus = new DataKamus(this);
 db = dKamus.getWritableDatabase();
 dKamus.createTable(db);
 dKamus.generateData(db);

 setContentView(R.layout.main);
 editInggris = (EditText) findViewById(R.id.editEnglish);
 txtIndo = (TextView) findViewById(R.id.txtIndonesia);
 }

 public void getTerjemahan(View view){
 String result = "";

 String englishWord= editInggris.getText().toString();
 kamusCursor = db.rawQuery("SELECT ID, INGGRIS, INDONESIA FROM
kamus WHERE INGGRIS='"+englishWord+"' order BY INGGRIS", null);

 if (kamusCursor.moveToFirst()){
 result = kamusCursor.getString(2);
 for (; !kamusCursor.isAfterLast(); kamusCursor.moveToNext()){
 result = kamusCursor.getString(2);
 }
 }

 if (result.equals("")){
 result = "tidak ditemukan";
 }
 txtIndo.setText(result);
 }

 @Override
 protected void onDestroy() {
 // TODO Auto-generated method stub
 super.onDestroy();
 kamusCursor.close();
 db.close();
 }
}

```

5. Maka hasil yang nanti ditampilkan adalah seperti berikut:

