

Praktikum 9

Pemrograman Android GUI 2 Layar

Dosen : Ir. Nanang Syahroni M.Kom

Pokok Bahasan

- Konsep pemrograman Java OOP pada Android
- Deklarasi pemrograman Android
- Penggunaan Button, Textview, Textfield, dll.

Tujuan Belajar

Dengan praktikum ini mahasiswa diharapkan dapat:

- Mengenal tentang pemrograman java pada Android
- Mengenal tentang komponen pemrograman Android
- Mengetahui fungsi file AndroidManifest.xml, main.xml dan string.xml
- Dapat memahami cara kerja component dan intent dalam sistem Android
- Dapat menggunakan component dan intent secara efektif.

Pendahuluan

1. Layout mengatur tata letak komponen dan tampilan aplikasi Adnroid. Ada beberapa macam layout dalam pemrograman Android seperti pada table berikut ini:

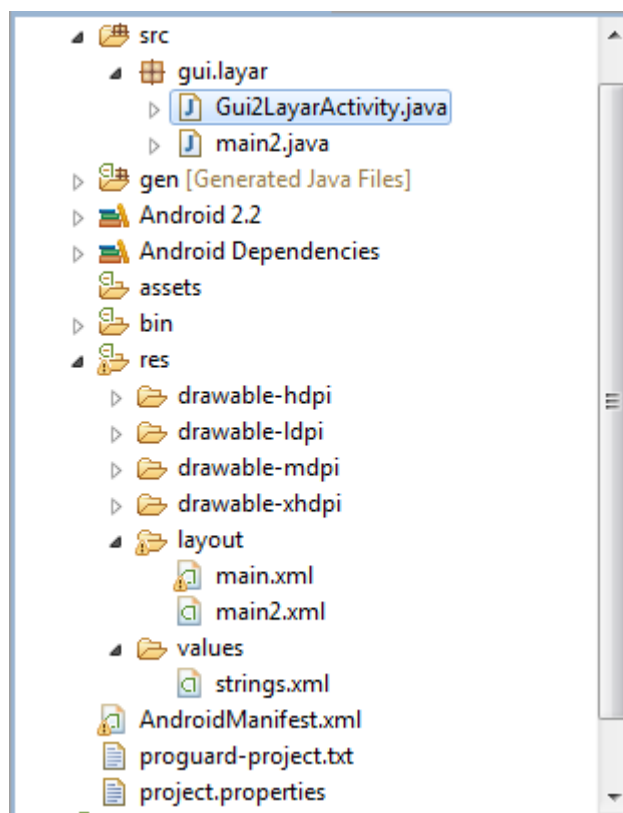
Layout Manager	Deskripsi
Linear Layout	Mengatur posisi layout pada layar secara linear (horizontal ataupun vertikal).
RelativeLayout	Mengatur posisi layout pada layar secara relative.
TableLayout	Mengatur layout agar berbentuk seperti tabel.
FrameLayout	Mengatur layout yang memungkinkan untuk mengubah tata letak komponen.
TabLayout	Mengatur layout agar dapat berbentuk tab.

Penggunaan layout ini biasanya tidak hanya untuk menampilkan secara dinamis suatu view, namun juga dapat diisi dengan banyak item.

2. Beberapa macam component Android yang biasa dipakai dalam pembuatan aplikasi, diantaranya : Text Control, Button, Check Box, Radio Button, List, Grid, Date and Time.
3. Beberapa macam text control meliputi : TextView, EditText, AutoCompleteTextView, dan MultiAutoCompleteTextView.

Praktek : Program GUI dengan 2 Layar

1. Pertama-tama butlah project baru, misalnya dengan nama **Gui2Layar**. Selanjutnya pada *Package Explorer* akan terlihat susunan seperti berikut:



2. Kita lihat file **AndroidManifest.xml** dalam aplikasi **Gui2Layar**, agar dapat memfasilitasi pembuatan 2 layar maka editlah seperti contoh berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="gui.layar"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="8" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".Gui2LayarActivity"
            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=".main2" >
    </activity>
</application>
</manifest>

```

3. Kemudian edit file **Gui2LayarActivity.java** seperti contoh dibawah ini. File inilah yang mengatur logic dari aplikasi, berdasarkan informasi yang ada di *AndroidManifest.xml*, maka file inilah yang akan dijalankan pertama kali ketika user ingin menjalankan aplikasi.

```

package gui.layar;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class Gui2LayarActivity extends Activity {
    EditText nm, almt;
    Button pilih, reset;
    public static String nama;
    public static String alamat;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        nm = (EditText) findViewById(R.id.EditText01);
        almt = (EditText) findViewById(R.id.EditText02);
        pilih = (Button) findViewById(R.id.Button01);
        reset = (Button) findViewById(R.id.Button02);
        pilih.setOnClickListener(new click01());
        reset.setOnClickListener(new click02());
    }
    class click01 implements Button.OnClickListener {
        public void onClick(View v) {
            nama = nm.getText().toString();
            alamat = almt.getText().toString();
            Intent i = new Intent(Gui2LayarActivity.this, main2.class);
            finish();
            startActivity(i);
        }
    }
    class click02 implements Button.OnClickListener {
        public void onClick(View v) {
            nm.setText("");
            almt.setText("");
        }
    }
}

```

4. Pada class **Gui2LayarActivity** diatas yang perlu kita perhatikan disini adalah pada baris *setContentView(R.layout.main)*; Artinya, *file xml* yang akan digunakan menjadi tampilan activity ini adalah pada file dengan nama *main.xml* dalam folder *res/layout/* pada project kita.

5. Kemudian edit file *main.xml* seperti contoh dibawah. File inilah yang mengatur tampilan, sesuai dengan parameter yang diberikan ke metode *setContentView()*.

```
<?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:background="@color/warna"
    android:orientation="vertical" >

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

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

        <TextView
            android:id="@+id/TextView01"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nama" >
        </TextView>

        <EditText
            android:id="@+id/EditText01"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="" >
        </EditText>
    </LinearLayout>

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

        <TextView
            android:id="@+id/TextView02"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Alamat" >
        </TextView>

        <EditText
            android:id="@+id/EditText02"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="" >
        </EditText>
    </LinearLayout>

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

```

        <Button
            android:id="@+id/Button01"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Proses" >
        </Button>

        <Button
            android:id="@+id/Button02"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Reset" >
        </Button>
    </LinearLayout>
</LinearLayout>

```

6. Kemudian edit file *string.xml* tempat mendeklarasikan beberapa string, seperti contoh berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Mencoba GUI 2 Layer</string>
    <string name="app_name">Gui2Layar</string>
    <color name="warna">#114488</color>
    <color name="warna2">#881199</color>
</resources>

```

7. Kemudian buatlah file *main2.java* seperti contoh dibawah ini. File inilah yang mengatur logic dari aplikasi di layar kedua, berdasarkan informasi yang ada di *AndroidManifest.xml*.

```

package gui.layar;

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

public class main2 extends Activity {
    TextView dtnama, dtalamat;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main2);
        dtnama = (TextView) findViewById(R.id.TextView01);
        dtalamat = (TextView) findViewById(R.id.TextView02);
        dtnama.setText("Nama = " + Gui2LayarActivity.nama);
        dtalamat.setText("Alamat = " + Gui2LayarActivity.alamat);
    }
}

```

8. Kemudian buatlah file *main2.xml* seperti contoh dibawah dalam */res/layout*. File inilah yang mengatur tampilan layar kedua sesuai dengan parameter yang diberikan ke metode *setContentView()* pada program *main2.java*.

```

<?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:background="@color/warna2"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="" >
    </TextView>

    <TextView
        android:id="@+id/TextView02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="" >
    </TextView>

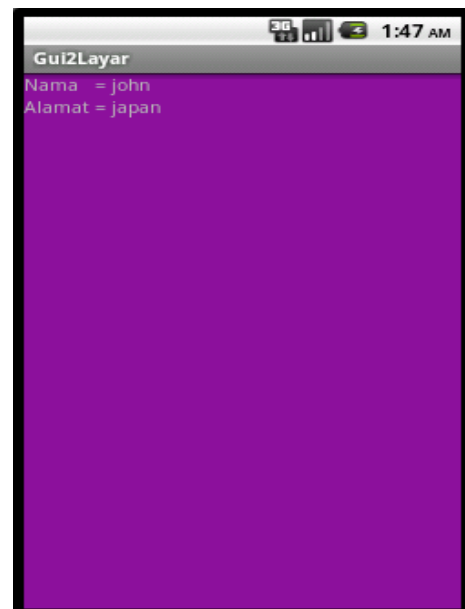
</LinearLayout>

```

9. Jika proses edit telah selesai semua, maka click **Run**, lalu tunggu beberapa saat, maka pada emulator akan terlihat seperti contoh layar ke 1 seperti dibawah.
10. Jika proses mengisi data telah selesai, maka click **Proses**, selanjutnya akan ditampilkan layar ke 2 seperti contoh dibawah.



Layar ke 1



Layar ke 2