

# **MODUL 9**

## **DASAR PEMROGRAMAN & GUI ANDROID**



DISADUR DARI MODUL TRAINING SEAMEO SEAMOLEC

Muhammad Zen Samsono Hadi, ST. MSc.

[zenhadi@eepis-its.edu](mailto:zenhadi@eepis-its.edu)

PRODI TEKNIK TELEKOMUNIKASI  
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

# Tujuan



- Dapat memahami dasar teori pemrograman android
- Dapat memahami mekanisme kerja component dalam sistem operasi android
- Dapat menggunakan *component* secara efektif dalam membuat aplikasi android

# Component Android



- **Activity**  
Menyajikan antar muka visual yang digunakan sebagai interaksi pengguna.
- *Intent* merupakan objek yang memegang isi dari pesan.
- *Activity* menyajikan antar muka. *Activity* berinteraksi dengan pengguna dalam jangka waktu yang lama, dan tetap aktif, bahkan pada saat idle.

# Pengertian File Manifest



Sebelum android dapat memulai *component* aplikasi, maka android harus melihat *component* yang sedang berjalan.

Untuk itu, aplikasi akan mendeklarasikan *component* yang mereka gunakan dalam sebuah file manifest yang digabungkan ke dalam paket file android berekstensi (.apk).

File manifest ini berstruktur XML dan selalu diberi nama AndroidManifest.xml untuk semua aplikasi.

# Contoh File Manifest



```
<?xml version="1.0" encoding="utf-8"?>
<manifest . . . >
  <application . . . >
    <activity
android:name="com.example.project.FreneticActivity"

android:icon="@drawable/small_pic.png"

android:label="@string/freneticLabel"
          . . . >
      </activity>
    . . .
  </application>
</manifest>
```

# Penyaring Intents



Obyek *intent* dapat menampilkan nama target *component* secara eksplisit.

Jika hal ini dilakukan, android akan menemukan *component* (berdasarkan pada deklarasi di dalam file manifest) dan mengaktifkannya.

Namun jika target tidak disebutkan secara eksplisit, maka android harus dapat menentukan *component* yang terbaik untuk merespon intent tersebut.

# Contoh Kode Intents



```
<activity
  android:name="com.example.project.FreneticActivity"
  android:icon="@drawable/small_pic.png"
  android:label="@string/freneticLabel"
  . . . >
  <intent-filter . . . >
<action android:name="android.intent.action.MAIN" />
  <category
android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
  <intent-filter . . . >
<action android:name="com.example.project.BOUNCE" />
  <data android:mimeType="image/jpeg" />
  <category
  android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

# Memulai Task



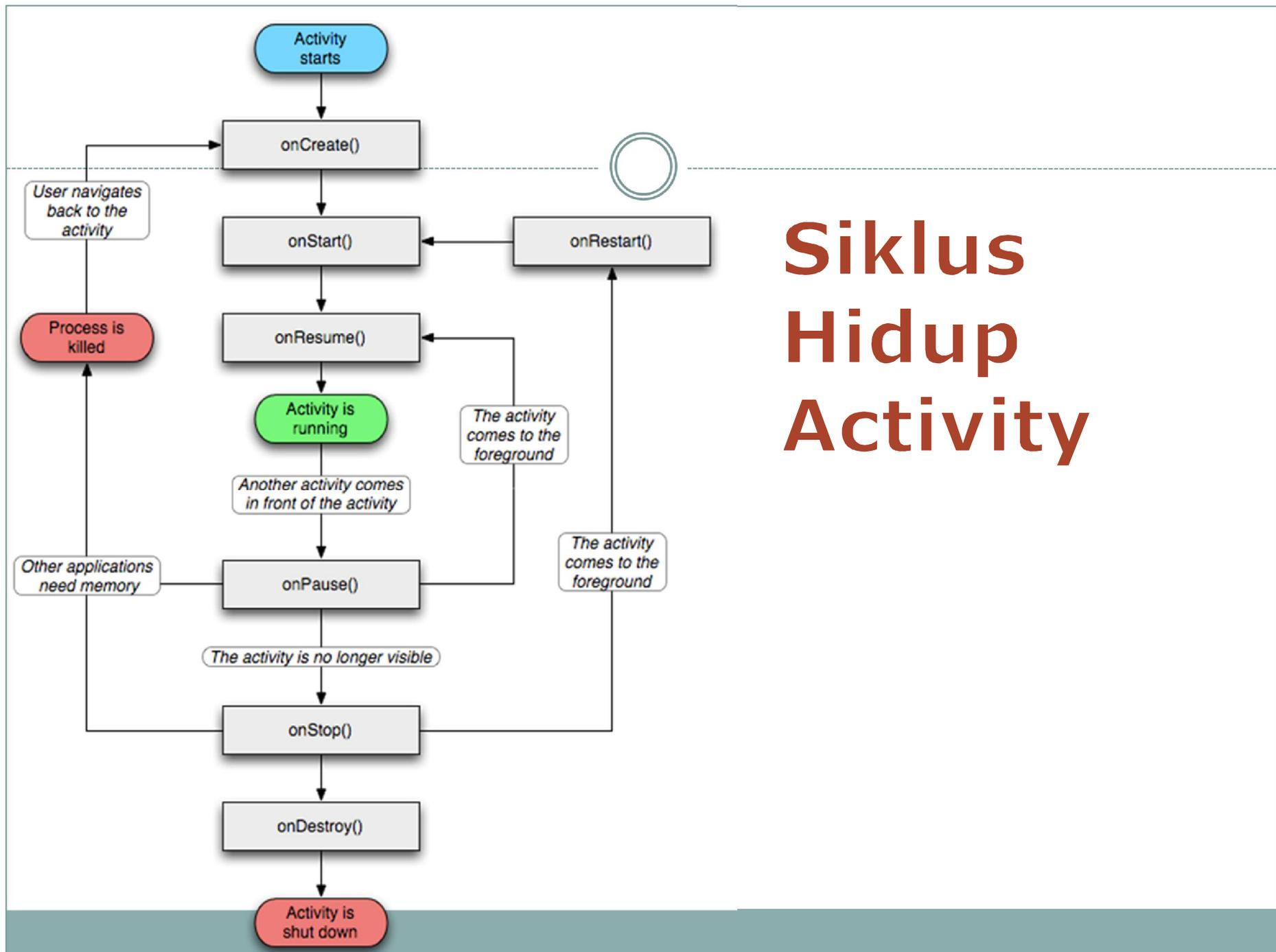
Sebuah activity dapat diatur menjadi entry point untuk sebuah task dengan memberikannya intent filter dengan `android.intent.action.MAIN` yang ditentukan sebagai aksi dan `android.intent.category.LAUNCHER` sebagai kategori.

Filter ini akan menampilkan ikon dan label activity di application launcher, memberikan akses kepada pengguna untuk meluncurkan task dan kembali ke application launcher kapan saja pengguna menginginkannya.

# Siklus Hidup Component



- Active  
Ketika ada di permukaan layar (di atas tumpukan activity pada task yang saat itu sedang berjalan). Activity inilah yang berkonsentrasi pada aksi dari pengguna
- Pause  
Ketika tidak lagi berkonsentrasi pada pengguna namun tetap tampil di layar.
- Stop  
Ketika terhalangi sepenuhnya oleh activity lain.



# Siklus Hidup Activity

# Menyimpan Kondisi Activity



Ketika sistem menghentikan sebuah activity untuk menghemat memori, mungkin saja pengguna ingin kembali ke activity sebelumnya dan mendapatkan activity dalam kondisi yang sama seperti sebelumnya.

Untuk menjaga kondisi agar tampak sama seperti sebelum dihentikan, anda dapat mengimplementasikan method `onSaveInstanceState()` untuk activity tersebut.



# KOMPONEN ANDROID

# Component Android



Ada beberapa macam component Android yang biasa dipakai dalam pembuatan aplikasi, yaitu :

1. Text Control
2. Button
3. Check Box
4. Radio Button
5. List
6. Grid
7. Date and Time

# Text Control



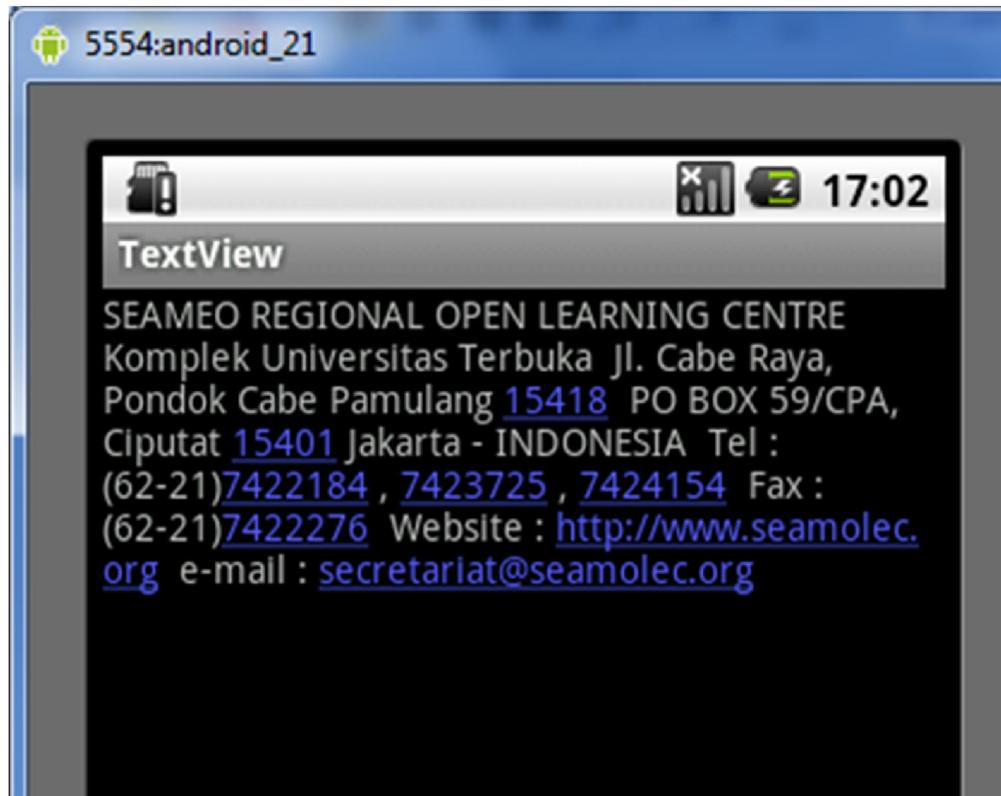
Macam-macam text control meliputi :

1. TextView
2. EditText
3. AutoCompleteTextView
4. MultiAutoCompleteTextView

# TextView



Komponen TextView ini dapat memunculkan tulisan di layar tetapi pengguna tidak dapat mengubah tulisan tersebut.



# EditText



EditText merupakan subclass dari TextView. Dilihat dari namanya, EditText dapat digunakan untuk mengubah teks.



# AutoCompleteTextView



AutoCompleteTextView merupakan TextView dengan fungsi auto-complete. Ketika pengguna menyentuh huruf, maka akan muncul pilihan menu yang dapat dipilih.



# MultiAutoCompleteTextView

MultiAutoCompleteTextView sama seperti AutoCompleteTextView. Perbedaannya adalah ketika kita telah memilih kata, lalu kita menggunakan tanda koma kemudian kita dapat mencari kata lagi.



# Button



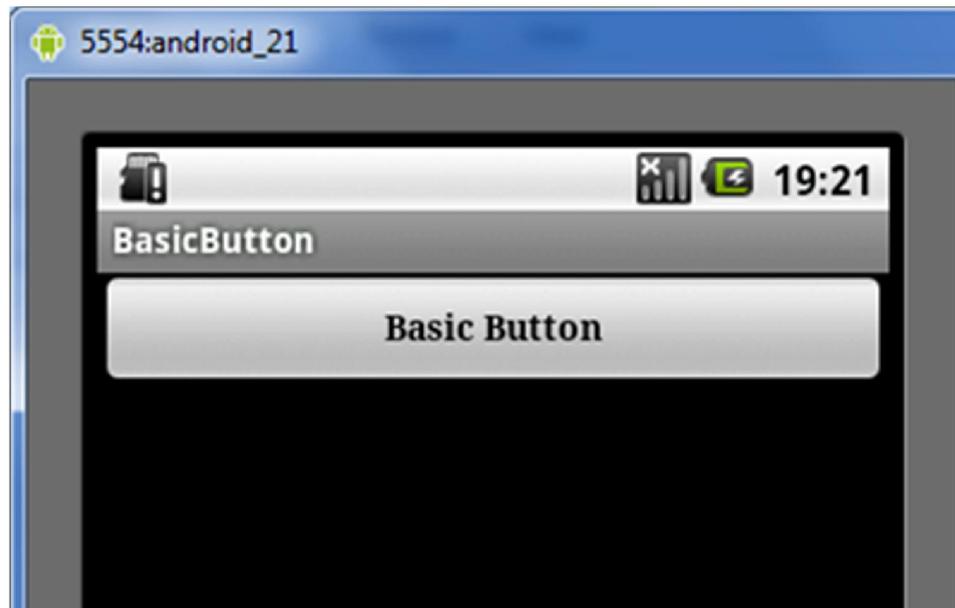
Pada kesempatan ini, akan dibahas 3 jenis *button* yaitu :

1. Basic button
2. Image button
3. Toggle button

# Basic Button



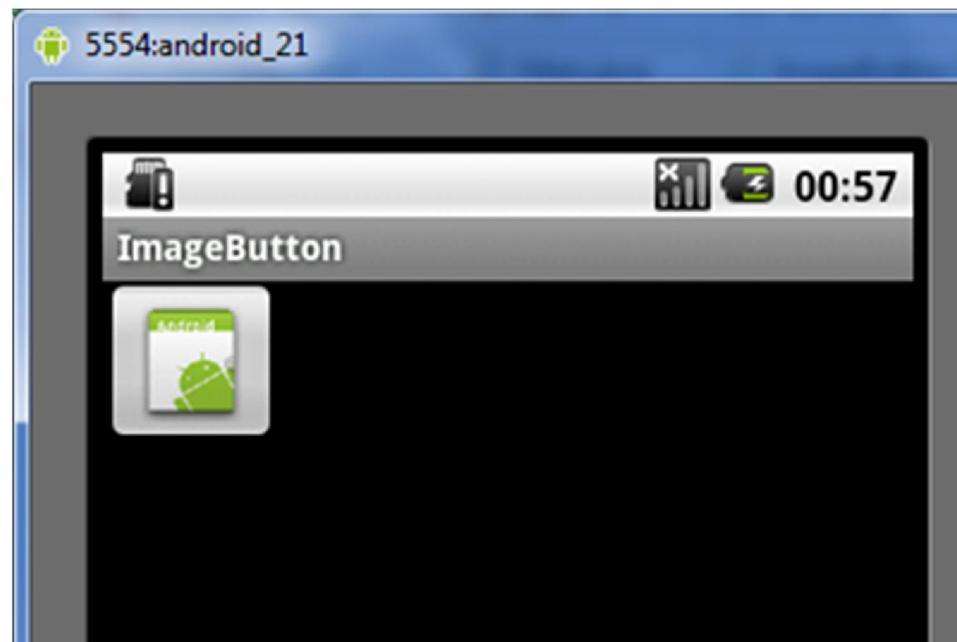
Button yang paling dasar di android adalah `android.widget.Button`. Di class ini hanya dapat menangani event ketika button di tekan.



# Image Button



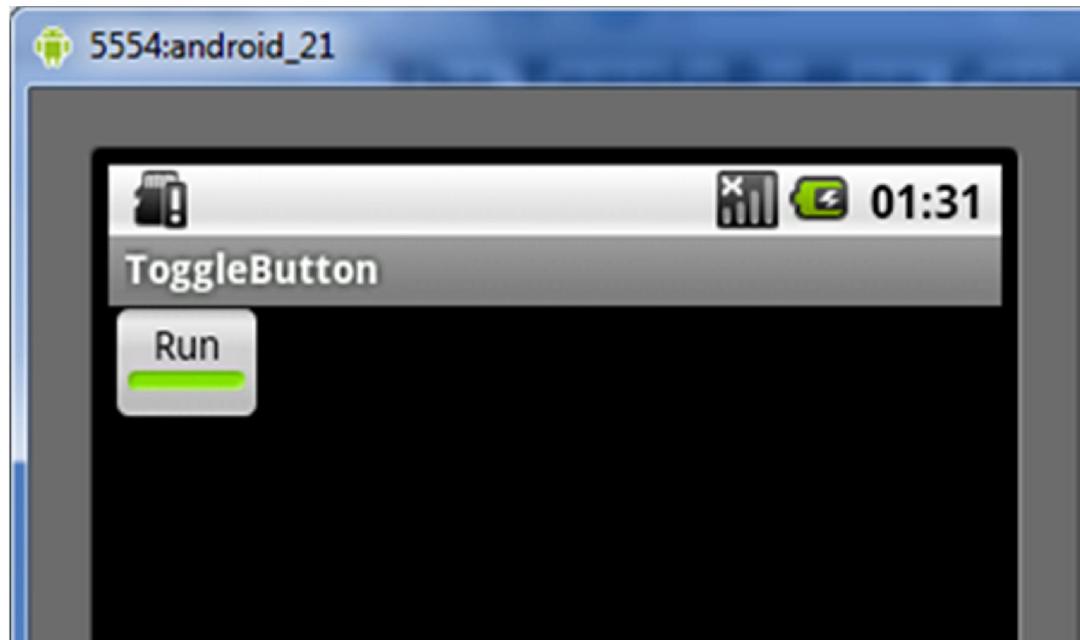
Android pun mendukung image button melalui `android.widget.ImageButton`. Penggunaan image button hampir sama dengan basic button.



# Toogle Button



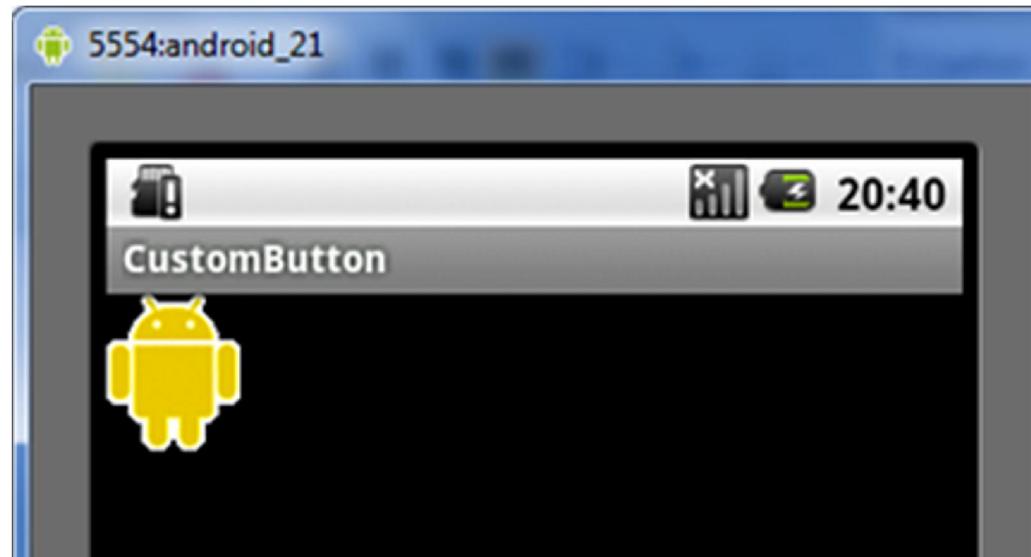
ToggleButton ini seperti check box atau radio button. Button ini mengarah pada On dan Off.



# Custom Button



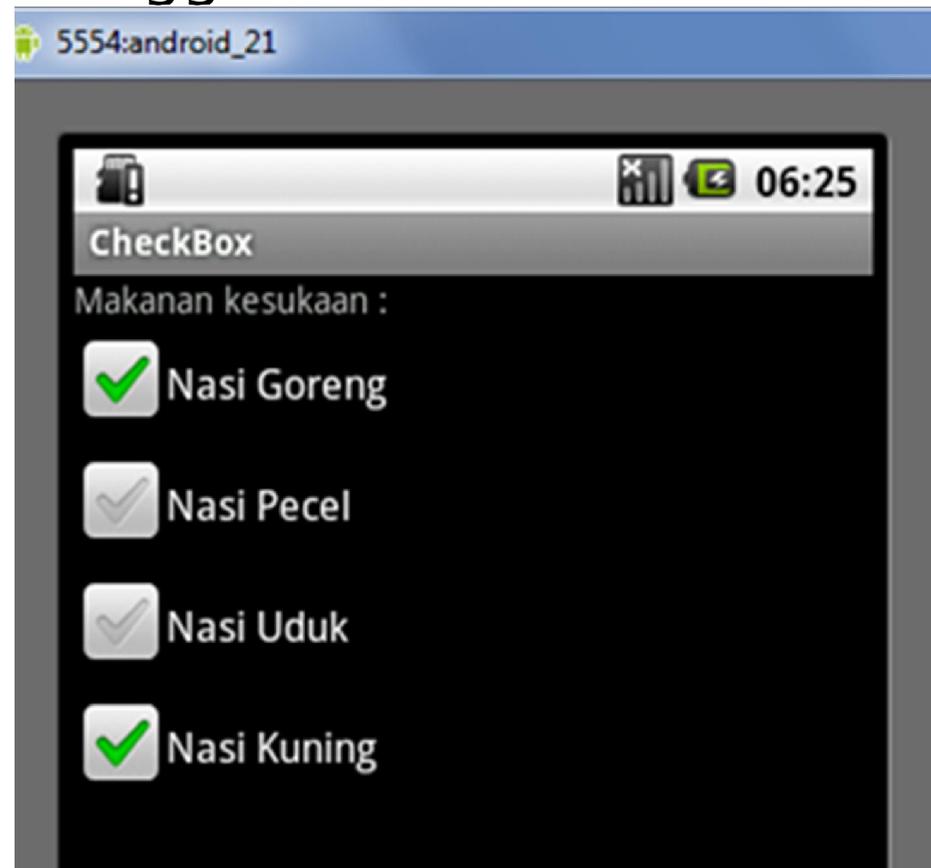
Custom button di sini dibuat menggunakan 3 custom image.



# Check Box



CheckBox memiliki dua keadaan yaitu On dan Off sama seperti ToggleButton.



# Radio Button



RadioButton memberikan pilihan kepada pengguna namun pengguna hanya bisa memilih satu.



# List

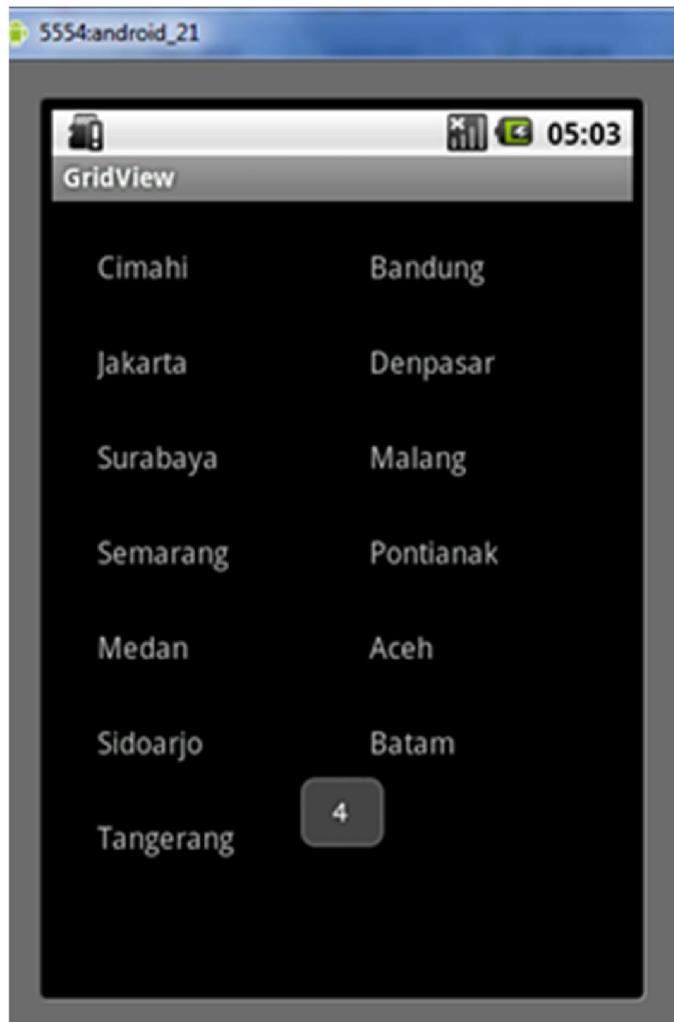


ListView berfungsi untuk menampilkan list dari item – item secara vertikal.

# Grid View



Android memiliki GridView yang dapat menampilkan data dalam bentuk grid. Data yang ditampilkan dapat berupa teks, gambar, dan lain-lain.



# Date and Time

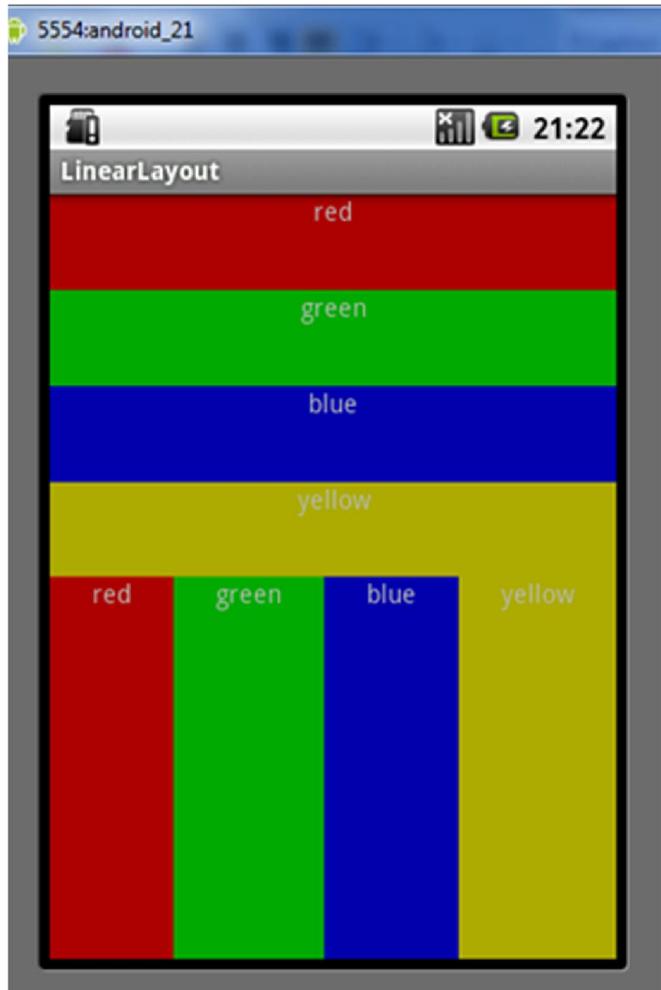
Android juga menyediakan fasilitas DatePicker dan TimePicker yang membuat kita dapat memilih tanggal dan waktu.



# Layout Manager

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

# Linear Layout

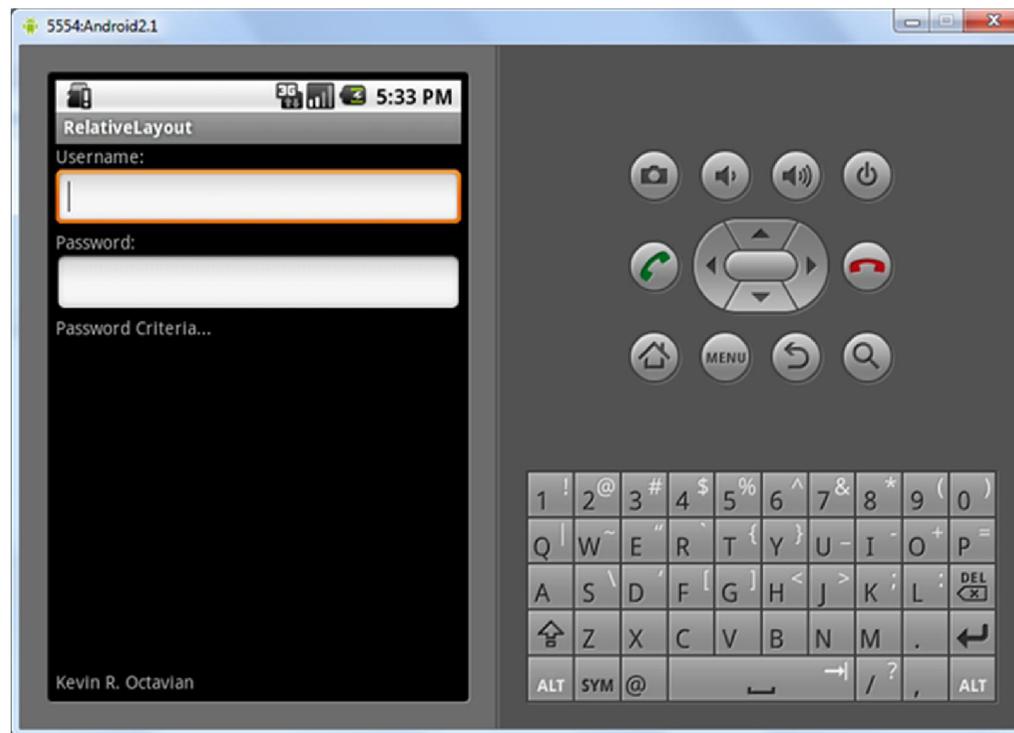


LinearLayout adalah tata letak yang paling sederhana tersedia di Android. LinearLayout mengatur tata letak komponen secara linear horizontal atau vertikal.

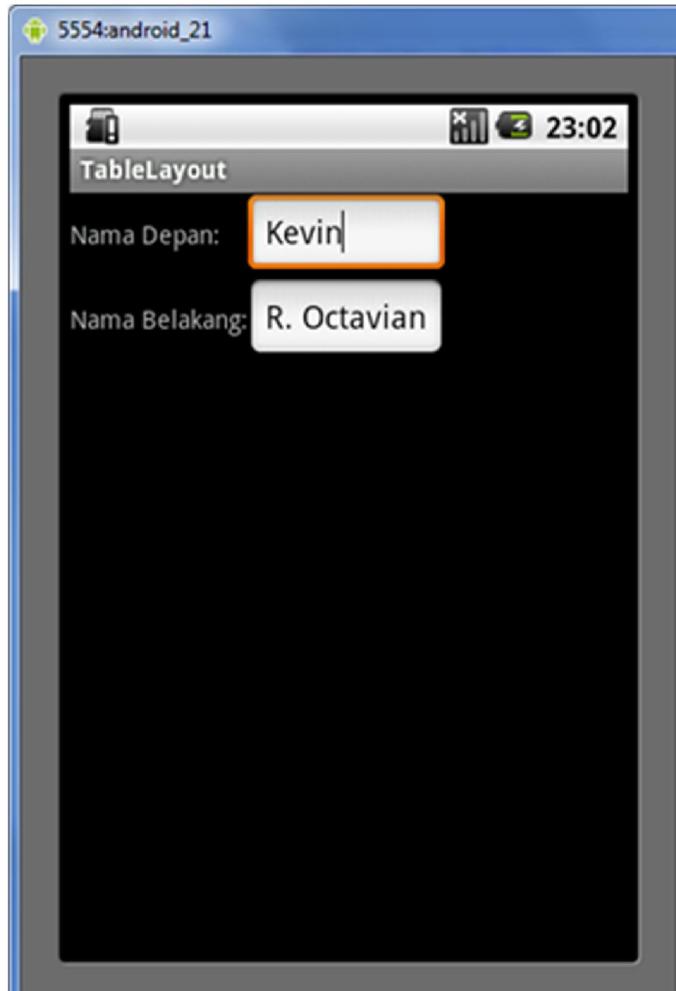
# Relative Layout



RelativeLayout mengatur tata letak komponen dalam hubungan satu sama lain. Hal ini memberikan fleksibilitas lebih dari LinearLayouts.



# Table Layout



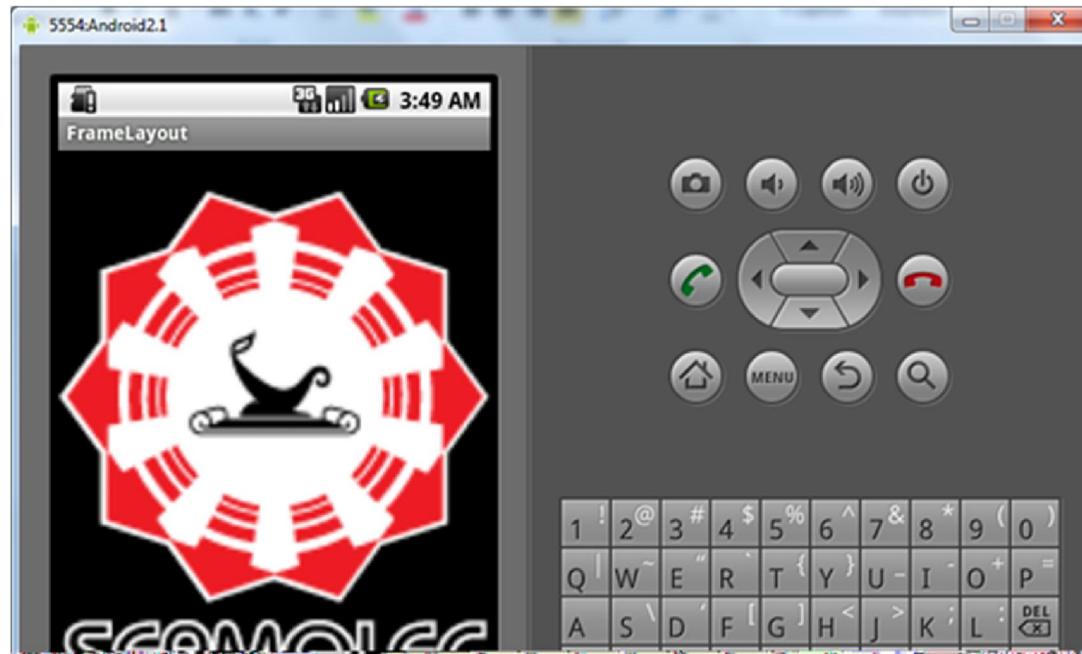
○

TableLayout adalah sebuah ViewGroup yang memperlihatkan elemen View sebagai baris-baris dan kolom-kolom.

# Frame Layout



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



# Tab Layout



Layout ini mengatur tampilan agar dapat berbentuk tab.





# PEMBUATAN GUI ANDROID

# EditText & Button



- Pembuatan Layout di main.xml

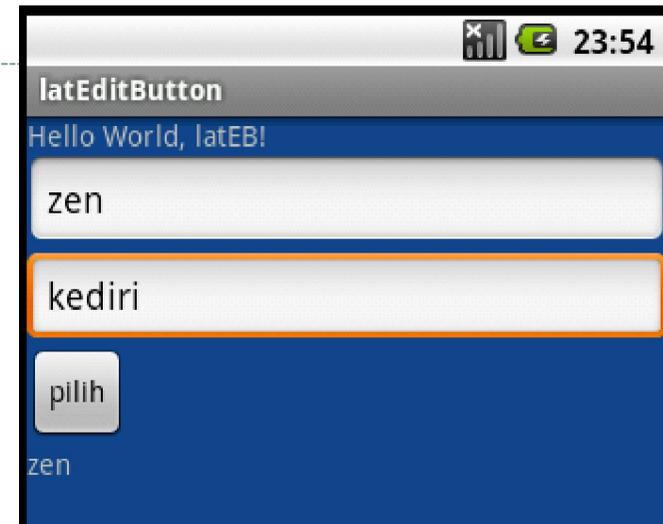


# EditText & Button

- Pembuatan file java



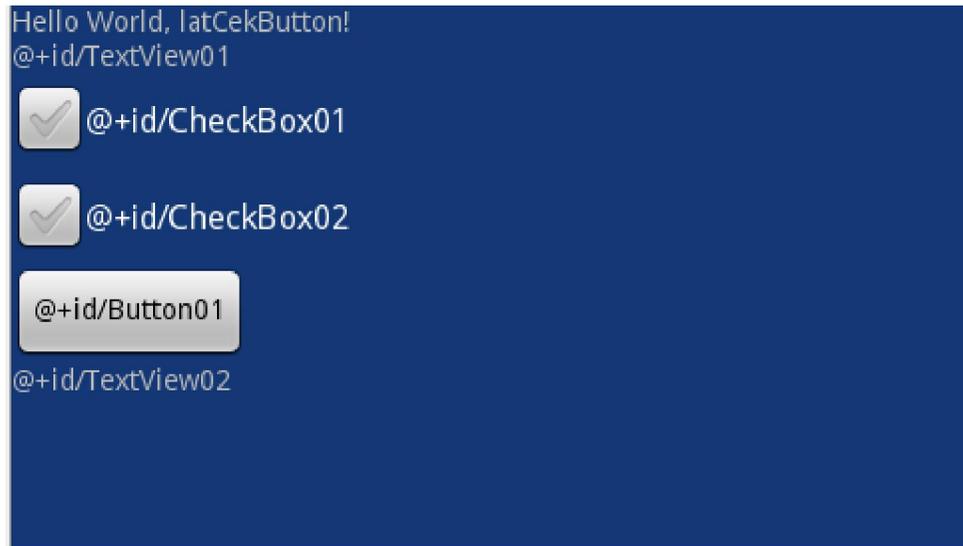
```
1 package pens.edu;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.TextView;
9
10 public class latEB extends Activity {
11     EditText nama, alamat;
12     TextView data;
13     Button pilih;
14     /** Called when the activity is first created. */
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.main);
19         nama = (EditText) findViewById (R.id.EditText01);
20         alamat = (EditText) findViewById (R.id.EditText02);
21         data = (TextView) findViewById (R.id.TextView01);
22         pilih = (Button) findViewById (R.id.Button01);
23         pilih.setOnClickListener(new klik());
24     }
25     class klik implements Button.OnClickListener {
26         public void onClick(View v) {
27             data.setText (nama.getText ());
28         }
29     }
30 }
```



# CheckBox & Button



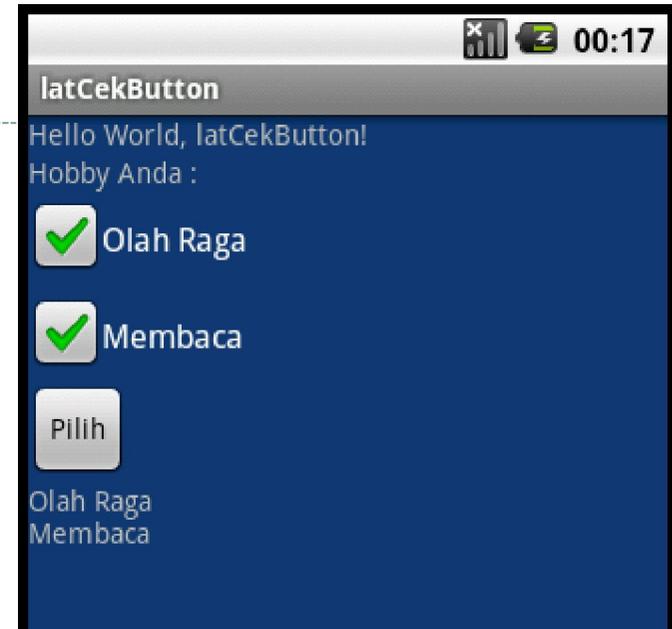
- Pembuatan Layout di main.xml



# CheckBox & Button

- Pembuatan file java

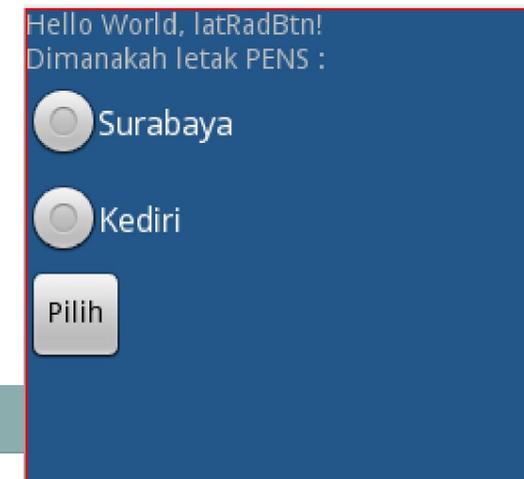
```
1 package pens.edu;
2
3 import android.app.Activity;
4
5
6
7
8
9
10 public class latCekButton extends Activity {
11     TextView data;
12     Button pilih;
13     CheckBox cb1, cb2;
14     /** Called when the activity is first created. */
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.main);
19         cb1 = (CheckBox) findViewById (R.id.CheckBox01);
20         cb2 = (CheckBox) findViewById (R.id.CheckBox02);
21         data = (TextView) findViewById (R.id.TextView02);
22         pilih = (Button) findViewById (R.id.Button01);
23         pilih.setOnClickListener(new klik());
24     }
25     class klik implements Button.OnClickListener {
26         public void onClick (View v){
27             String a = "";
28             if (cb1.isChecked())
29                 a = a + "Olah Raga" + "\n";
30             if (cb2.isChecked())
31                 a = a + "Membaca";
32             data.setText(a);
33         }
34     }
35 }
```



# RadioButton



- Pembuatan Layout di main.xml



# RadioButton



- Pembuatan Layout di main.xml

```
17<RadioGroup android:id="@+id/RadioGroup01"  
18    android:layout_width="wrap_content"  
19    android:layout_height="wrap_content"  
20    android:orientation="vertical">  
21<RadioButton android:text="Surabaya"  
22    android:id="@+id/RadioButton01"  
23    android:layout_width="wrap_content"  
24    android:layout_height="wrap_content"></RadioButton>  
25<RadioButton android:text="Kediri"  
26    android:id="@+id/RadioButton02"  
27    android:layout_width="wrap_content"  
28    android:layout_height="wrap_content"></RadioButton>  
29</RadioGroup>
```

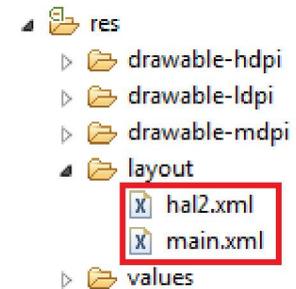
# RadioButton

- Pembuatan file java

```
1 package pens.edu;
2
3 import android.app.Activity;
10
11 public class latRadBtn extends Activity {
12     TextView data;
13     Button pilih;
14     RadioButton rb1, rb2;
15     RadioGroup rg;
16     /** Called when the activity is first created. */
17     @Override
18     public void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.main);
21         rg = (RadioGroup) findViewById (R.id.RadioGroup01);
22         rb1 = (RadioButton) findViewById (R.id.RadioButton01);
23         rb2 = (RadioButton) findViewById (R.id.RadioButton02);
24         data = (TextView) findViewById (R.id.TextView02);
25         pilih = (Button) findViewById (R.id.Button01);
26         pilih.setOnClickListener(new klik());
27     }
28     class klik implements Button.OnClickListener {
29         public void onClick (View v) {
30             if (rb1.isChecked())
31                 data.setText("Surabaya");
32             if (rb2.isChecked())
33                 data.setText("Kediri");
34         }
35     }
36 }
```



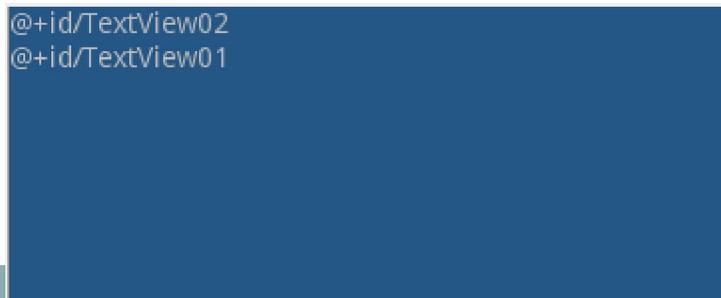
# EditText dengan 2 Tampilan



- Layout di main.xml



- Layout di hal2.xml => tambahkan dari Android XML file

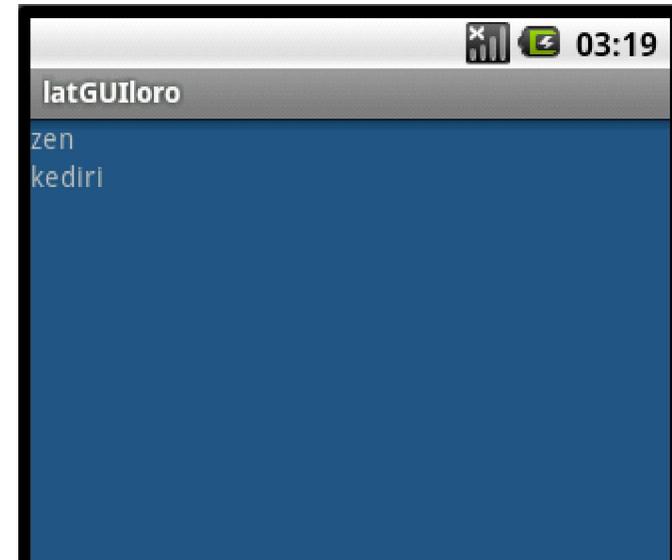
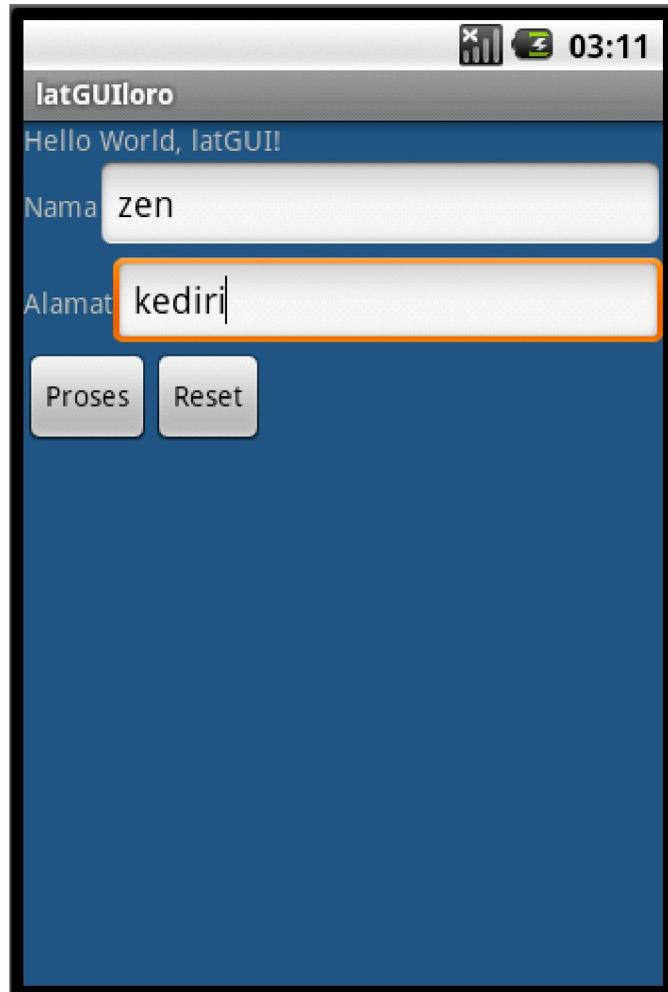


# EditText dengan 2 Tampilan

- main.xml

```
1<?xml version="1.0" encoding="utf-8"?>
2<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3    android:orientation="vertical"
4    android:layout_width="fill_parent"
5    android:layout_height="fill_parent"
6    android:background="@color/varna">
7<TextView
8    android:layout_width="fill_parent"
9    android:layout_height="wrap_content"
10   android:text="@string/hello"
11 />
12<LinearLayout
13   android:orientation="horizontal"
14   android:layout_width="fill_parent"
15   android:layout_height="wrap_content">
16<TextView android:text="Nama"
17   android:id="@+id/TextView01"
18   android:layout_width="wrap_content"
19   android:layout_height="wrap_content"></TextView>
20<EditText android:text=""
21   android:id="@+id/EditText01"
22   android:layout_width="fill_parent"
23   android:layout_height="wrap_content"></EditText>
24</LinearLayout>
25<LinearLayout
26   android:orientation="horizontal"
27   android:layout_width="fill_parent"
28   android:layout_height="wrap_content">
29<TextView android:text="Alamat" android:id="@+id/TextView02" android:layout_width="wrap_content" andr
30<EditText android:text="" android:id="@+id/EditText02" android:layout_width="fill_parent" android:lay
31</LinearLayout>
```

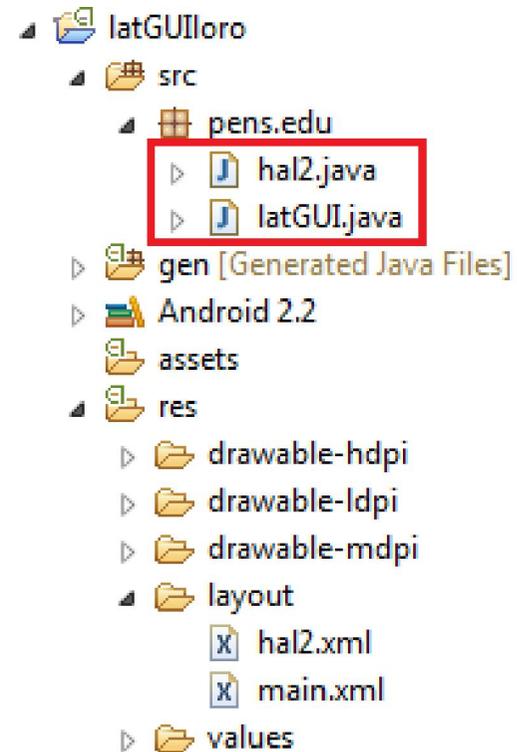
# EditText dengan 2 Tampilan



# EditText dengan 2 Tampilan



- Buat 2 file java  
latGUI.java untuk tampilan ke 1  
hal2.java untuk tampilan ke 2  
(dari klik kanan project ->  
New -> Class)



# EditText dengan 2 Tampilan

- latGUI.java

```
10 public class latGUI extends Activity {
11     EditText nm, almt;
12     Button pilih, reset;
13     public static String nama;
14     public static String alamat;
15     /** Called when the activity is first created. */
16     @Override
17     public void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.main);
20         nm = (EditText) findViewById(R.id.EditText01);
21         almt = (EditText) findViewById(R.id.EditText02);
22         pilih = (Button) findViewById(R.id.Button01);
23         reset = (Button) findViewById(R.id.Button02);
24         pilih.setOnClickListener(new click01());
25         reset.setOnClickListener(new click02());
26     }
27     class click01 implements Button.OnClickListener{
28         public void onClick(View v){
29             nama = nm.getText().toString();
30             alamat = almt.getText().toString();
31             Intent i = new Intent (latGUI.this, hal2.class);
32             finish();
33             startActivity(i);
34         }
35     }
36     class click02 implements Button.OnClickListener{
37         public void onClick(View v){
38             nm.setText("");
39             almt.setText(""); }
40     }
41 }
```

# EditText dengan 2 Tampilan



- hal2.java

```
1 package pens.edu;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.widget.TextView;
6
7 public class hal2 extends Activity {
8     TextView dtnama, dtalamat;
9     /** Called when the activity is first created. */
10    @Override
11    public void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.hal2);
14        dtnama = (TextView) findViewById (R.id.TextView01);
15        dtalamat = (TextView) findViewById (R.id.TextView02);
16        dtnama.setText (latGUI.nama);
17        dtalamat.setText (latGUI.alamat);
18    }
19 }
```

# EditText dengan 2 Tampilan



- Edit di AndroidManifest.xml

```
1<?xml version="1.0" encoding="utf-8"?>
2<manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="pens.edu"
4      android:versionCode="1"
5      android:versionName="1.0">
6  <application android:icon="@drawable/icon" android:label="@string/app_name">
7    <activity android:name=".latGUI"
8      android:label="@string/app_name">
9      <intent-filter>
10         <action android:name="android.intent.action.MAIN" />
11         <category android:name="android.intent.category.LAUNCHER" />
12      </intent-filter>
13    </activity>
14    <activity android:name=".hal2"></activity>
15  </application>
16  <uses-sdk android:minSdkVersion="8" />
17
18</manifest>
```