



ANDROID

UNIVERSITAS KOMPUTER INDONESIA

BAB V.

USER INTERFACE (UI)

PART 2 - BASIC VIEW

Dosen Pembina :

Bella Hardiyana, S. Kom

Julian Chandra, S. Kom

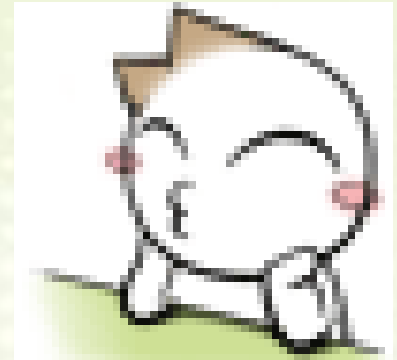


View

View dalam pemrograman Android berfungsi sebagai komponen UI.

View dibagi menjadi tiga kategori, yaitu :

- **Basic view** : view yang sering digunakan, seperti: TextView, EditText, Button, dsb.
- **Picker view** : berisi view TimePicker dan DatePicker.
- **List view** : view yang berfungsi untuk menampilkan daftar (list), meliputi ListView dan SpinnerView.



Basic View

Basic View terdiri dari beberapa view, yaitu :

- **TextView** : berfungsi sebagai label, hanya dapat menampilkan tulisan saja.
- **EditText** : berfungsi sebagai view inputan berupa teks.
- **Button** : berfungsi sebagai view tombol.
- **ImageButton** : mirip dengan Button, hanya saja dapat ditambahkan gambar.
- **CheckBox** : view yang umumnya berbentuk kotak dan dapat diceklis. Hanya ada dua status, yaitu checked dan unchecked.
- **ToggleButton** : view berbentuk tombol dan memiliki lampu indikator.
- **RadioGroup** : view yang berfungsi untuk mengelompokkan RadioButton.
- **RadioButton** : view yang hanya bisa dipilih salah satu dari kelompok RadioGroup.

Contoh BasicView

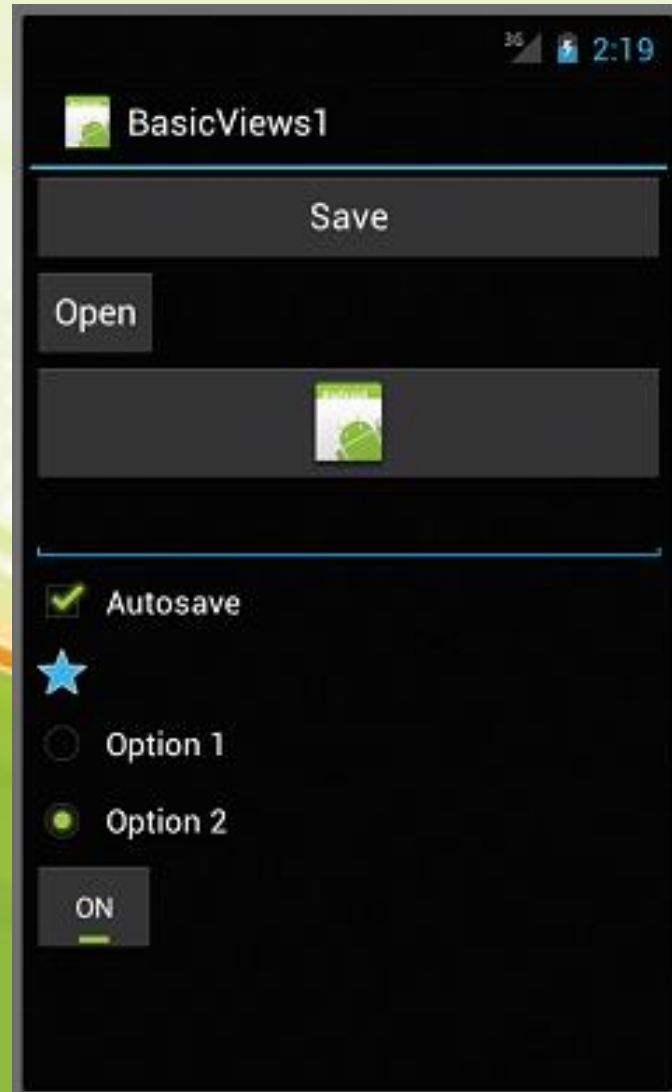
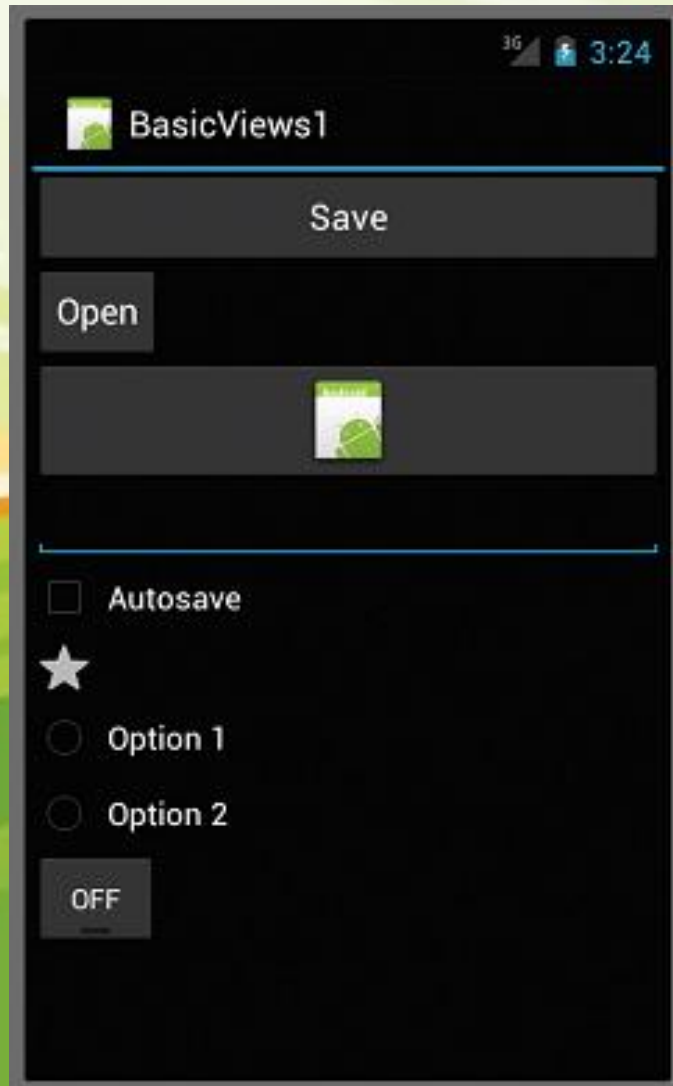
Perhatikan semantik dibawah ini

```
<?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/btnSave"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="save" />
    <Button android:id="@+id/btnOpen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Open" />
    <ImageButton android:id="@+id/btnImg1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher" />
    <EditText android:id="@+id/txtName"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <CheckBox android:id="@+id/chkAutosave"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Autosave" />
```

```
<CheckBox android:id="@+id/star"
    style="?android:attr/starStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<RadioGroup android:id="@+id/rdbGp1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <RadioButton android:id="@+id/rdb1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Option 1" />
    <RadioButton android:id="@+id/rdb2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Option 2" />
</RadioGroup>
<ToggleButton android:id="@+id/toggle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Contoh penggunaan TextView dapat dilihat pada Bab 3. Hello World

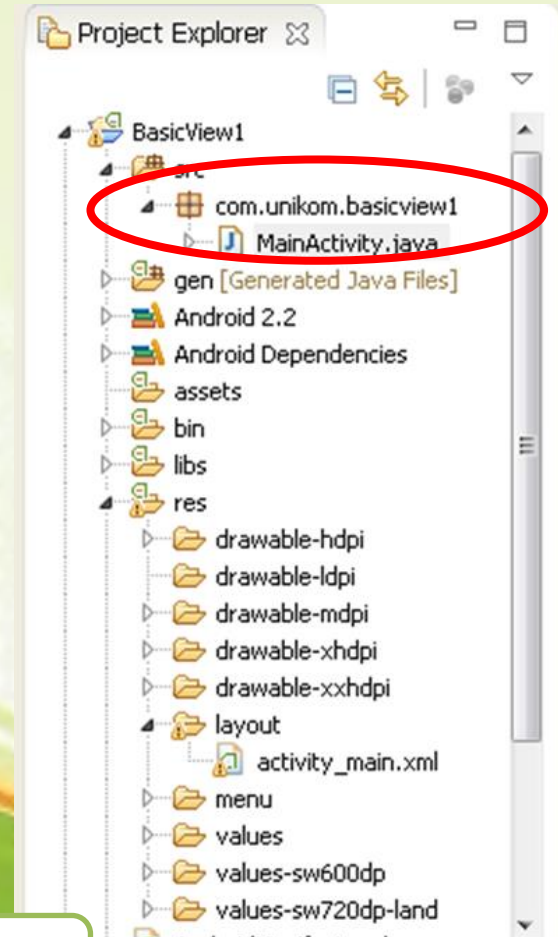
Tampilan Output



Menambah Perintah Untuk BasicView

Menambahkan perintah ketika view ditekan/dipilih.

Untuk menambahkan perintah, tentu saja kita harus mengubah koding java yang ada pada folder **src/nama_paket**. Buka file java tersebut.



Ayo dicoba

Iya bu 😊

Tahapan dalam pengkodean (1)

Berikut adalah tahapan-tahapan yang harus dilakukan jika kita akan memberikan perintah pada view yang ingin kita atur.

1. Cek layout xml dan pastikan **semua view yang ingin diatur memiliki id.**

```
Contoh : <Button android:id="@+id/btnSave"
```

2. Import kelas **View pada **android.view.View****

3. Import semua kelas view tersebut dengan format

```
android.widget.NamaView;  
Contoh : android.widget.Button;
```



Tahapan dalam pengkodean (2)

4. Buat sebuah **referensi** untuk masing-masing view dan **casting** semua objek tersebut **berdasarkan** komponen **viewnya**. Sedangkan objek yang **diwakili** referensi tersebut diambil dari **atribut** yang terdapat pada **kelas R.id** dengan pemanggilan menggunakan metode **findViewById()** . **Format penulisan :**

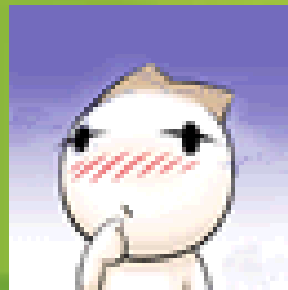
```
KelasView objRef = (KelasView) findViewById(R.id.atribut) ;
```

Contoh :

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

```
CheckBox checkBox = (CheckBox) findViewById(R.id.chkAutosave);
```

```
RadioGroup radioGroup = (RadioGroup) findViewById(R.id.rdbGp1);
```



Tahapan dalam pengkodean (3)

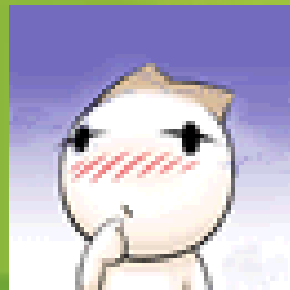
5. Sediakan metode yang didalamnya terdapat ***anonymous inner class*** yang berfungsi untuk **mengimplementasi Listener**.

Catatan Penting :

- Pemanggilan Interface OnClickListener terdapat pada kelas View, sehingga tidak perlu di import.
- Pemanggilan Interface OnCheckedChangeListener harus di import, karena terdapat pada kelas yang berbeda.

Contoh :

```
checkBox.setOnClickListener(new View.OnClickListener() { }  
radioGroup.setOnCheckedChangeListener(new OnCheckedChangeListener() { }  
toggleButton.setOnClickListener(new View.OnClickListener() { }
```

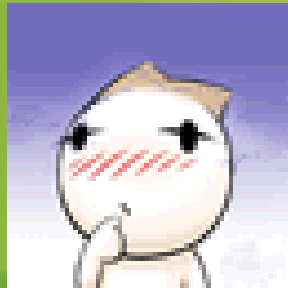


Tahapan dalam pengkodean (4)


6. Sediakan informasi sebagai penanda bahwa koding yang kita buat berjalan dengan baik.

Pada kasus ini saya menggunakan Toast. Toast merupakan sebuah teks yang akan muncul sesaat di layar. Pada dasarnya Toast masih merupakan sebuah view, maka kita wajib mengimportnya terlebih dahulu dengan paket : **android.widget.Toast**

```
private void DisplayToast(String msg) {  
    Toast.makeText (getBaseContext () , msg ,  
    Toast.LENGTH_SHORT) .show () ;  
}
```



Anda Bingung? Ada Pertanyaan?



Kalo gitu
sekarang coba
praktikan.
InsyaAllah bisa

Bu, aku belum
ngerti nih 😞

PERINGATAN

Diperlukan pengetahuan OOP Java bila ingin menguasai teknik berikut ini.

Anda boleh *copy-paste*, tetapi pahami langkah-langkahnya.

MainActivity.java (1)

```
package com.unikom.basicview1;

import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.RadioButton;
import android.widget.Toast;
import android.widget.ToggleButton;

public class MainActivity extends Activity {
    private void DisplayToast(String msg) {
        Toast.makeText(getApplicationContext(), msg,
            Toast.LENGTH_SHORT).show();
    }

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

MainActivity.java (2)

```
//Koding klik untuk view Button
Button btnOpen = (Button) findViewById(R.id.btnOpen);
btnOpen.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        DisplayToast("Button Open ditekan");
    }
});

Button btnSave = (Button) findViewById(R.id.btnSave);
btnSave.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        DisplayToast("Button Save ditekan");
    }
});

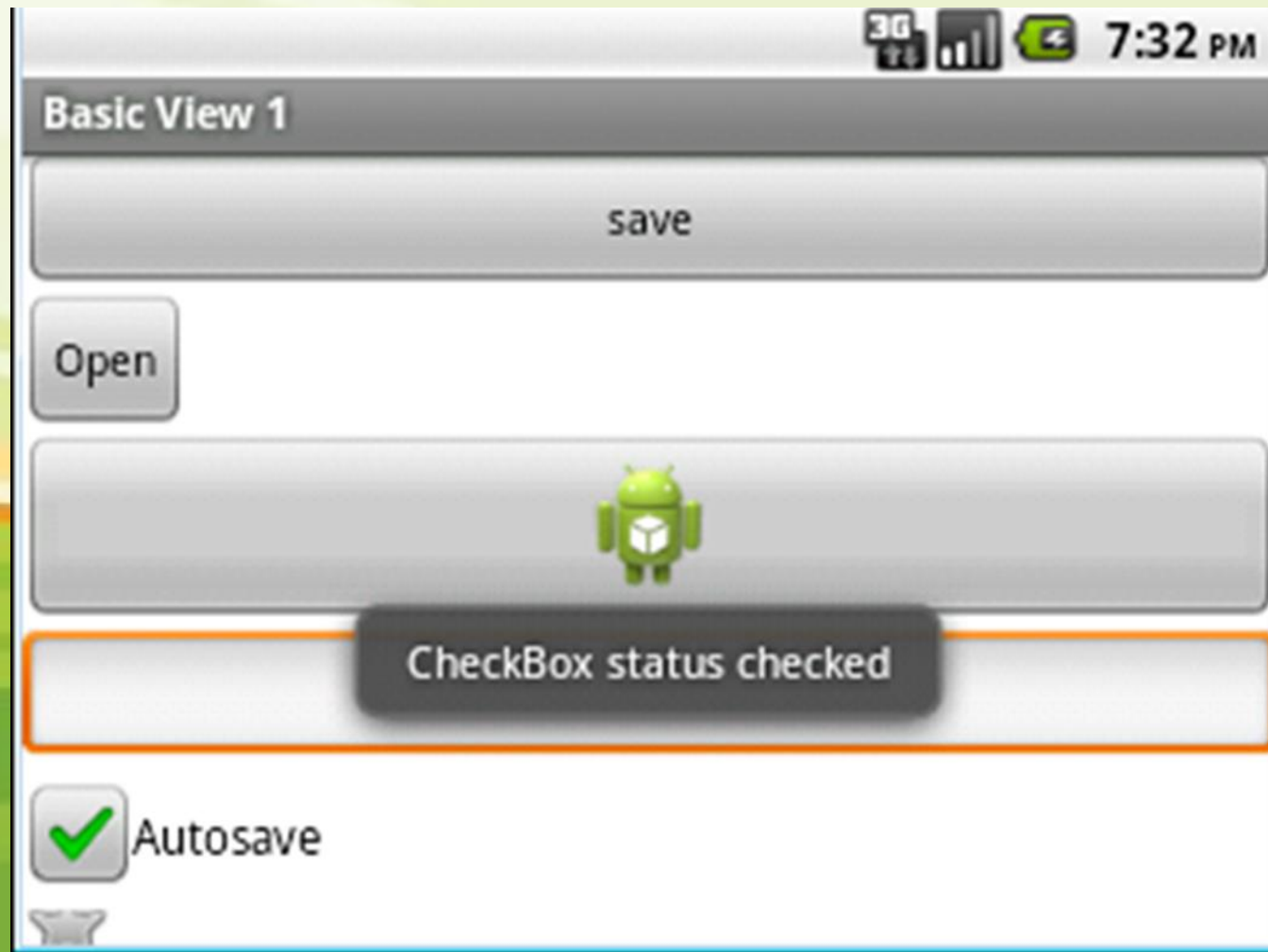
//Koding klik untuk view CheckBox
CheckBox checkBox = (CheckBox) findViewById(R.id.chkAutosave);
checkBox.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if(((CheckBox)v).isChecked())
            DisplayToast("CheckBox status checked");
        else
            DisplayToast("CheckBox status unchecked");
    }
});
```

MainActivity.java (3)

```
//Koding klik untuk view RadioGroup dan RadioButton
RadioGroup radioGroup = (RadioGroup) findViewById(R.id.rdbGp1);
radioGroup.setOnCheckedChangeListener(new OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        RadioButton rb1 = (RadioButton) findViewById(R.id.rdb1);
        if(rb1.isChecked())
            DisplayToast("Option 1 status checked!");
        else
            DisplayToast("Option 2 status checked!");
    }
});
```

```
//Koding klik untuk view ToggleButton
ToggleButton toggleButton = (ToggleButton) findViewById(R.id.toggle1);
toggleButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if(((ToggleButton)v).isChecked())
            DisplayToast("Toggle button status On");
        else
            DisplayToast("Toggle button status Off");
    }
});
}
```

Tampilan Output

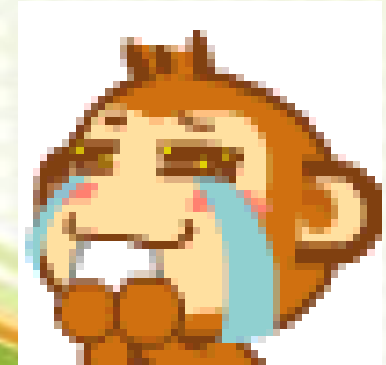


ProgressBar View

ProgressBar View digunakan ketika kita ingin menampilkan status progres dalam bentuk animasi. Misalnya ketika aplikasi sedang membuka suatu resource, maka aplikasi tersebut menampilkan animasi loading dalam bentuk lingkaran yang memutar.

Berikut terdapat beberapa tema/style yang dapat digunakan dalam membuat ProgressBar, yaitu :

- `Widget.ProgressBar.Horizontal`
- `Widget.ProgressBar.Small`
- `Widget.ProgressBar.Large`
- `Widget.ProgressBar.Inverse`
- `Widget.ProgressBar.Small.Inverse`
- `Widget.ProgressBar.Large.Inverse`



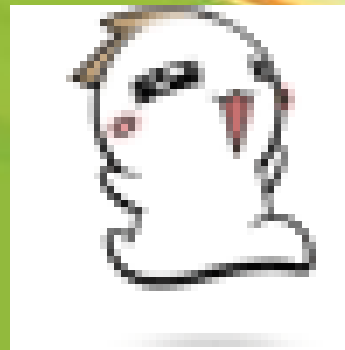
CATATAN

Untuk menjalankan animasi tersebut, dibutuhkan sebuah Thread yang akan bekerja dibelakang aplikasi.

activity_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/txtStatus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Harap Tunggu" />
    <ProgressBar android:id="@+id/progressbar"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        style="@android:style/Widget.ProgressBar.Horizontal" />
</LinearLayout>
```

Numpang
Nongol 😊



MainActivity.java (1)

```
package com.unikom.basicview2;

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

public class MainActivity extends Activity {
    static int progress;
    ProgressBar progressBar;
    int progressStatus= 0;
    Handler handler= new Handler();

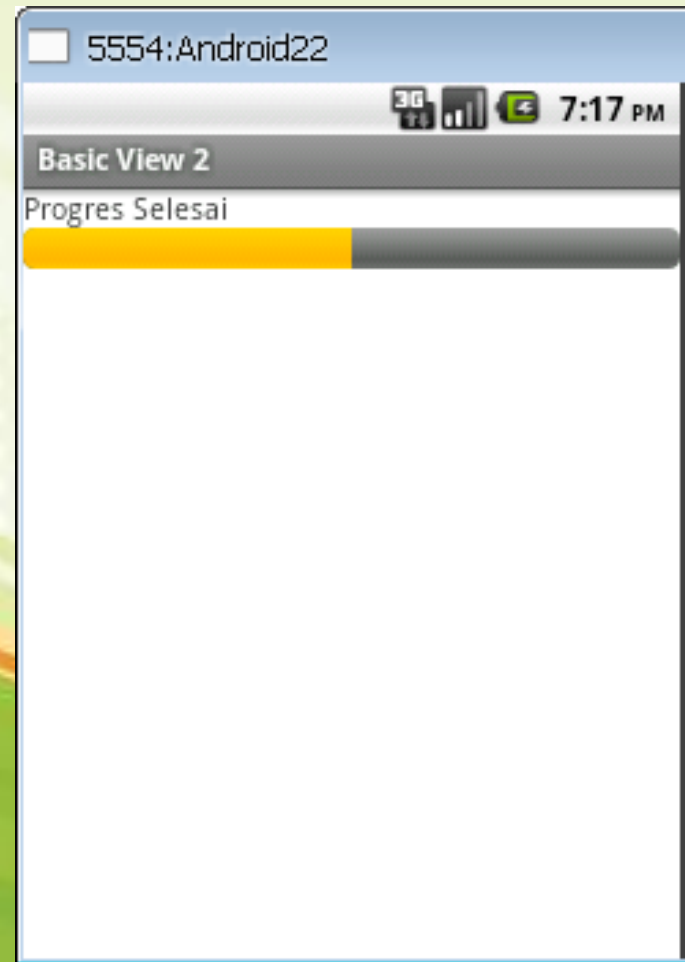
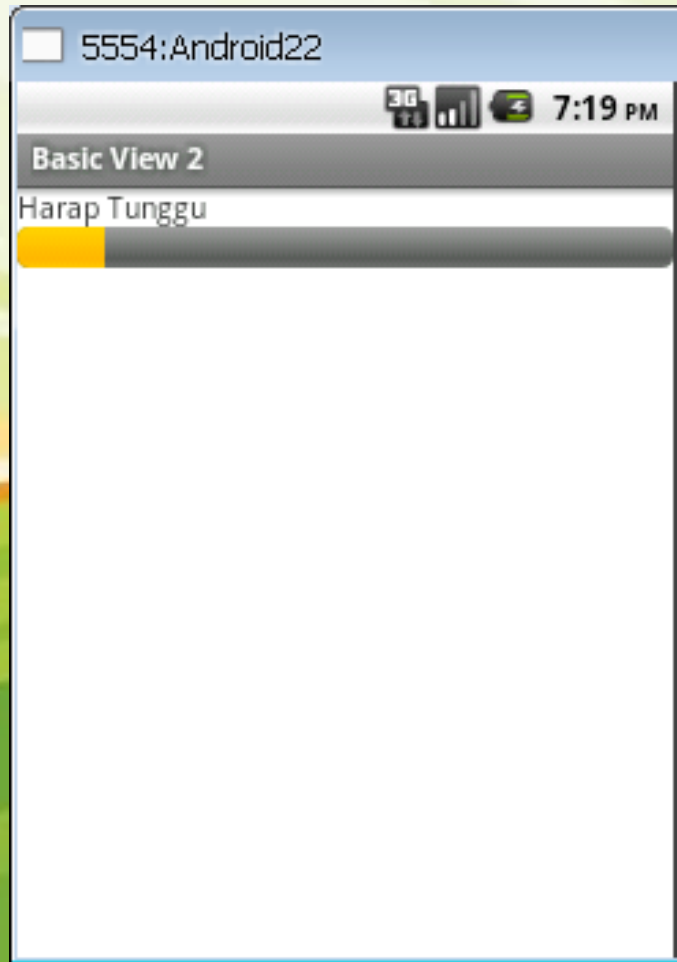
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        progress=0;
        progressBar= (ProgressBar) findViewById(R.id.progressBar);
        progressBar.setMax(200);
        new Thread(new Runnable() {
            public void run() {
                while(progressStatus < 100) {
                    progressStatus= doSomeWork();
                }
            }
        }).start();
    }
}
```

MainActivity.java (2)

```
handler.post(new Runnable() {  
    public void run() {  
        progressBar.setProgress(progressStatus);  
    }  
});  
}  
handler.post(new Runnable() {  
    public void run() {  
        TextView txtStatus = (TextView) findViewById(R.id.txtStatus);  
        txtStatus.setText("Progres Selesai");  
    }  
});  
}  
private int doSomeWork() {  
    try{  
        Thread.sleep(50);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    return ++progress;  
}  
}).start();  
}  
}
```


Tampilan Output



AutoCompleteTextView View



AutoCompleteTextView mirip seperti EditText. Beberapa sumber menyebutkan bahwa AutoCompleteTextView merupakan kelas anak dari EditText. AutoCompleteTextView berguna untuk melengkapi ketikan yang kita input berdasarkan daftar kosa kata yang sudah ditentukan sebelumnya.

CATATAN

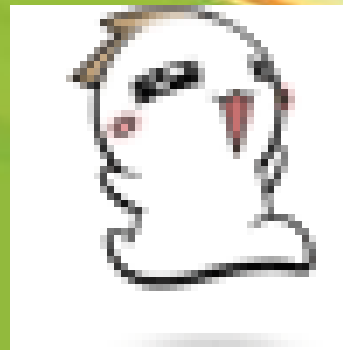
Gunakan AVD versi 4.0 Ice Cream Sandwich untuk melihat hasil yang maksimal. Karena pada versi sebelumnya, teks pada AutoCompleteTextView tidak terlihat. Karena warna teks dan layar berwarna putih.



activity_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:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Nama Orang Indonesia" />
    <AutoCompleteTextView android:id="@+id/txtNama"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

Numpang
Nongol Lagi 😊



MainActivity.java

```
package com.unikom.basicview3;
import android.os.Bundle;
import android.app.Activity;
import android.widget.AdapterView;
import android.widget.AutoCompleteTextView;

public class MainActivity extends Activity {
    String[] namaOrang= {
        "Lina Fadilah", "Lina Purnama", "Julian Chandra",
        "Chandra Pradipta", "Udin Saepudin", "Baharudin Habibie",
        "Fadilah Aditya", "Cecep Gorbacep", "Asep Gumasep"
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Mendaftarkan List Adapter berdasarkan String namaOrang
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_dropdown_item_1line, namaOrang);
        //Mengatur komponen mana yang ingin diolah
        AutoCompleteTextView textView = (AutoCompleteTextView)
            findViewById(R.id.txtNama);
        //3 huruf yang dibutuhkan untuk menjalankan AutoText
        textView.setThreshold(3);
        textView.setAdapter(adapter);
    }
}
```


Terima Kasih

