



ANDROID

UNIVERSITAS KOMPUTER INDONESIA

BAB IV.

USER INTERFACE (UI)

PART 1 - VIEWGROUP

Dosen Pembina :

Julian Chandra W, S.Kom,M.Kom

Bella Hardiyana, S. Kom, M. Kom



USER INTERFACE (UI)

Secara umum arsitektur UI pada aplikasi Android adalah UI yang meliputi Activity dan UI yang terdiri dari komponen. Semua yang berhubungan dengan UI pada aplikasi android terletak pada folder **res/layout/filename.xml**, dimana coding yang digunakan untuk memanggil file tersebut adalah dengan format **R.layout.filename**

PERLU DIINGAT :

Pemanggilan xml tidak dapat dilakukan secara langsung. File R.java yang terletak pada folder **gen/namapaket** berfungsi untuk menyimpan alamat memori dari semua hal yang ada pada folder **res**.



Struktur semantik layout xml (1)

Perhatikan semantik dibawah ini



```
<?xml version="1.0" encoding="utf-8"?>
<ViewGroup xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+[package:]id/resource_name"
    android:layout_height=["dimension" | "fill_parent" | "wrap_content"]
    android:layout_width=["dimension" | "fill_parent" | "wrap_content"]
    [ViewGroup-specific-attr] >
    <View
        android:id="@+[package:]id/resource_name"
        android:layout_height=["dimension" | "fill_parent" | "wrap_content"]
        android:layout_width=["dimension" | "fill_parent" | "wrap_content"]
        [ViewGroup-specific-attr] >
        <requestFocus/>
    </View>
</ViewGroup>
```

Struktur semantik layout xml (2)

1. **<ViewGroup>** : menentukan tata letak komponen view, seperti LinearLayout, RelativeLayout, AbsoluteLayout, dll.
 - **android:id** : resource id, yang berisi variabel unik dari elemen tersebut.
 - **android:layout_height** : mengatur tinggi View/ViewGroup. Terdiri dari :
 - **fill_parent/match_parent** : memenuhi layar.
 - **wrap_content** : sesuai dengan lebar/tinggi teks.
 - **android:layout_width** : mengatur lebar View/ViewGroup.
2. **<View>** : berisikan komponen UI/widget yang dipasang pada layout tersebut, seperti Basic View, Picker View dan List View.
3. **<requestFocus>** : permintaan fokus untuk widget tersebut.



Atribut umum View/ViewGroup

Berikut atribut yang umum digunakan oleh View/ViewGroup :

ATRIBUT	DESKRIPSI
layout_width	Mengatur lebar View/ViewGroup
layout_height	Mengatur tinggi View/ViewGroup
layout_marginTop	Meletakkan View/ViewGroup di posisi atas
layout_marginBottom	Meletakkan View/ViewGroup di posisi bawah
layout_marginLeft	Meletakkan View/ViewGroup di posisi kiri
layout_marginRight	Meletakkan View/ViewGroup di posisi kanan
layout_gravity	Mengatur alignment/posisi View
layout_weight	Mengatur bobot View
layout_x	Mengatur koordinat x View/ViewGroup
layout_y	Mengatur koordinat y View/ViewGroup
id	Memberikan nama id untuk View/ViewGroup
text	Mengatur tulisan yang muncul pada View

ViewGroup

Satu atau lebih view dapat di kelompokkan dalam sebuah ViewGroup. Kelas dasar dari ViewGroup terletak pada paket **`android.view.ViewGroup`**

ViewGroup terdiri dari :

1. `LinearLayout`
2. `AbsoluteLayout`
3. `TableLayout`
4. `RelativeLayout`
5. `FrameLayout`
6. `ScrollView`

Ooohh, jadi gitu ya bu?!!



LinearLayout

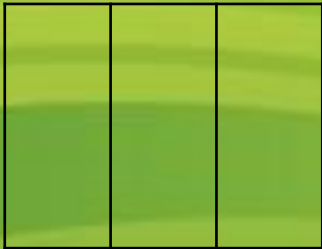
LinearLayout mengurutkan views menjadi single kolom/baris. Terdapat 2 macam penggunaan orientasi pada LinearLayout, yaitu :

1. orientation="vertical"



Catat Ah 😊

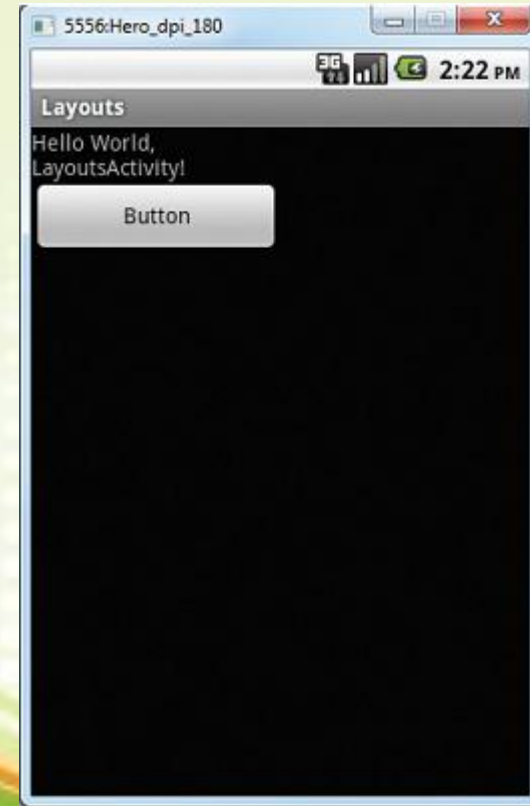
2. orientation="horizontal"



Contoh LinearLayout

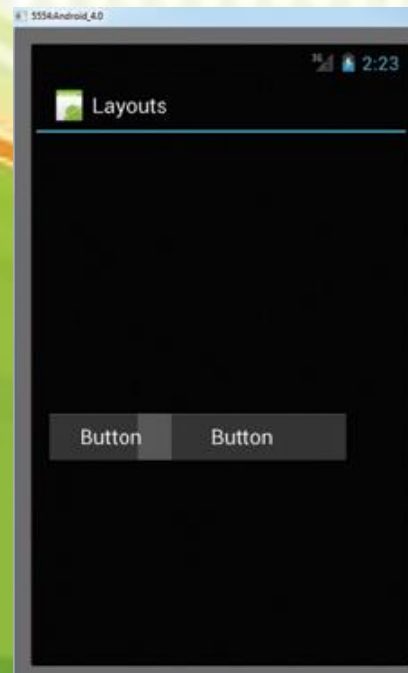
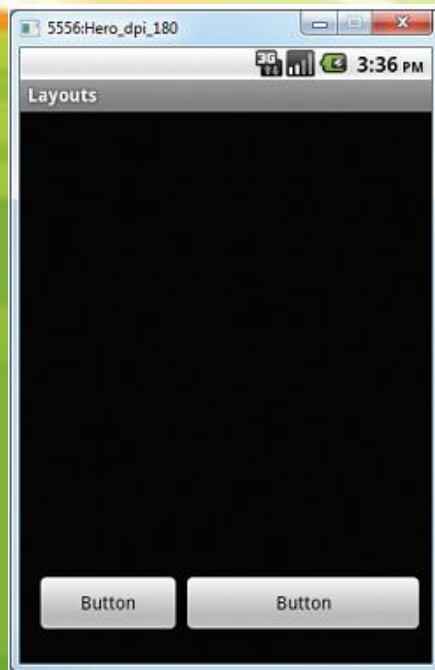
Perhatikan semantik dibawah ini

```
<?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" >
  <TextView
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="@string/hello" />
  <Button
    android:layout_width="160dp"
    android:layout_height="wrap_content"
    android:text="Button" />
</LinearLayout>
```



AbsoluteLayout

Penggunaan AbsoluteLayout terspesifik berdasarkan kordinat x dan y. Kemudahan dari layout ini akan terasa ketika kita menggunakan IDE Eclipse atau DroidDraw. Tetapi, AbsoluteLayout disarankan untuk tidak digunakan. Karena, kelemahan dari layout ini adalah perbedaan bentuk tampilan ketika kita menjalankan aplikasi tersebut pada resolusi handphone yang lebih besar. Perhatikan contoh tampilan dibawah ini.



Contoh AbsoluteLayout

Perhatikan semantik dibawah ini

<AbsoluteLayout

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

<Button

```
    android:layout_width="188dp"  
    android:layout_height="wrap_content"  
    android:text="Button"  
    android:layout_x="126px"  
    android:layout_y="361px" />
```

<Button

```
    android:layout_width="113dp"  
    android:layout_height="wrap_content"  
    android:text="Button"  
    android:layout_x="12px"  
    android:layout_y="361px" />
```

</AbsoluteLayout>

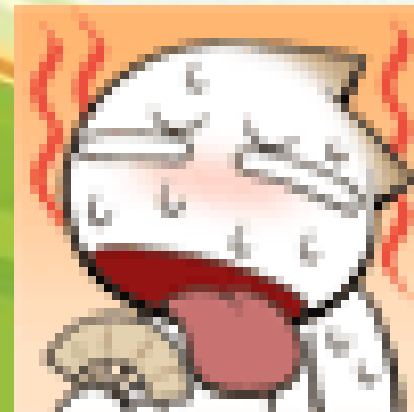


TableLayout

Berdasarkan namanya, penggunaan layout ini berbentuk seperti tabel. Dimana, tampilan layout berbentuk kolom dan baris. Penggunaan layout ini hampir mirip seperti tag **<table>** yang ada pada html. Desain view tergantung penuh berdasarkan **<TableRow>**. Satu buah **<TableRow>** mendefinisikan semua view dari baris tersebut. Besar lebar dari view pada kolom tersebut diambil dari lebar tertinggi dari kolom tersebut.

Catatan Penting : tidak ada konsep merge dalam layout ini.

Materi yang ini kok
pusing ya?!!



Contoh TableLayout

Perhatikan semantik dibawah ini

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

```
<TableRow>
```

```
<TextView
  android:text="User Name:"
  android:width="120dp"
/>
```

```
<EditText
  android:id="@+id/txtUserName"
  android:width="200dp" />
```

```
</TableRow>
```

```
<TableRow>
```

```
<TextView
  android:text="Password:"
/>
```

```
<EditText
  android:id="@+id/txtPassword"
  android:password="true"
/>
```

```
</TableRow>
```

```
<TableRow>
```

```
<TextView />
```

```
<CheckBox
```

```
  android:id="@+id/chkRememberPassword"
```

```
  android:layout_width="fill_parent"
```

```
  android:layout_height="wrap_content"
```

```
  android:text="Remember Password"
```

```
/>
```

```
</TableRow>
```

```
<TableRow>
```

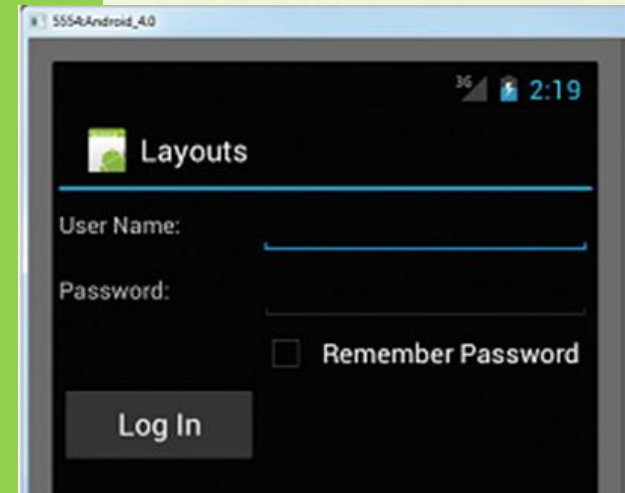
```
<Button
```

```
  android:id="@+id/buttonSignIn"
```

```
  android:text="Log In" />
```

```
</TableRow>
```

```
</TableLayout>
```



RelativeLayout

Penggunaan RelativeLayout bergantung penuh dengan **id** dari masing-masing view yang ada pada layout tersebut. Kita wajib memposisikan view yang ingin kita buat berdasarkan posisi view yang lain.

Berikut atribut penentu posisi yang terdapat pada RelativeLayout :

- layout_alignParentTop
- layout_alignParentLeft
- layout_alignLeft
- layout_alignRight
- layout_below
- layout_centerHorizontal



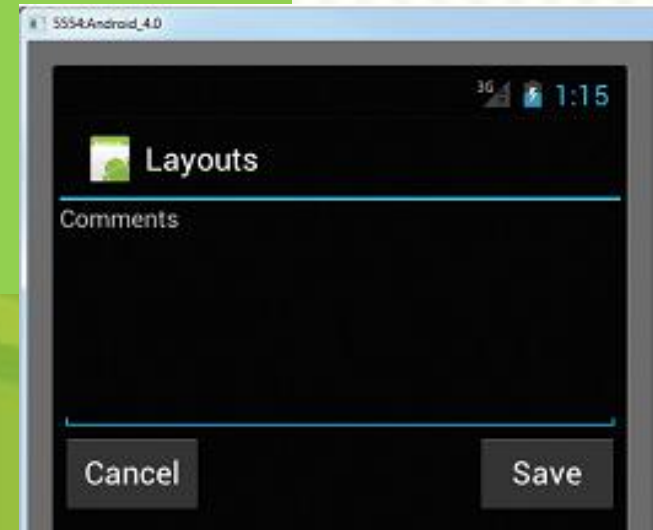
Woi Sadar!
Materinya
masih banyak

Contoh RelativeLayout

Perhatikan semantik dibawah ini

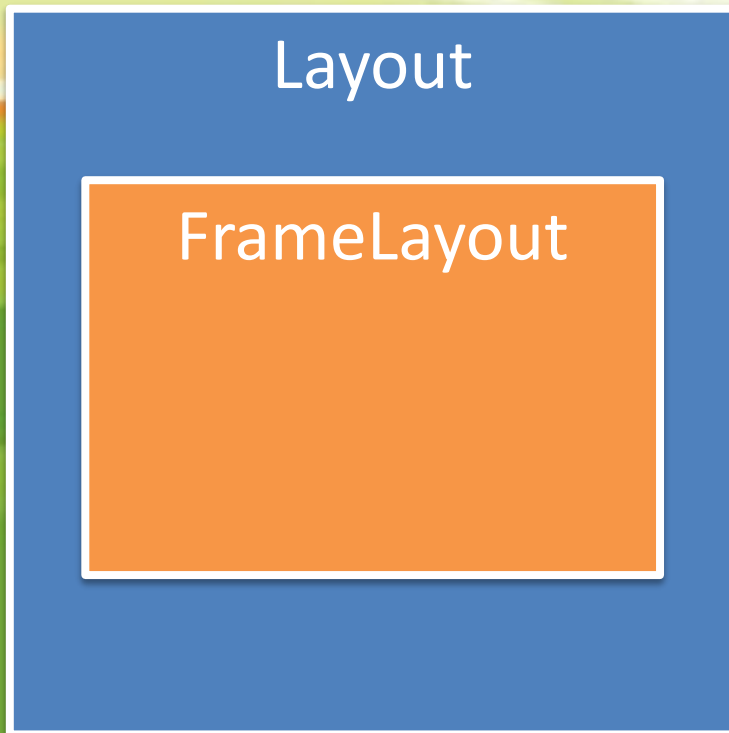
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:id="@+id/RLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android=
"http://schemas.android.com/apk/res/android" >
    <TextView
        android:id="@+id/lblComments"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Comments"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true" />
    <EditText
        android:id="@+id/txtComments"
        android:layout_width="fill_parent"
        android:layout_height="170px"
        android:textSize="18sp"
        android:layout_alignLeft="@+id/lblComments"
        android:layout_below="@+id/lblComments"
        android:layout_centerHorizontal="true" />
```

```
<Button
    android:id="@+id/btnSave"
    android:layout_width="125px"
    android:layout_height="wrap_content"
    android:text="Save"
    android:layout_below="@+id/txtComments"
    android:layout_alignRight="@+id/txtComments" />
<Button
    android:id="@+id/btnCancel"
    android:layout_width="124px"
    android:layout_height="wrap_content"
    android:text="Cancel"
    android:layout_below="@+id/txtComments"
    android:layout_alignLeft="@+id/txtComments" />
</RelativeLayout>
```



FrameLayout

FrameLayout hanya dapat digunakan didalam Layout yang lain. Jadi, dapat disimpulkan bahwa FrameLayout adalah layout didalam layout. Dalam FrameLayout, kita dapat menempatkan satu atau lebih View. Tetapi, semua view yang ada dalam FrameLayout terletak pada posisi kiri atas.



Eh FrameLayout, mau kerjasama ga?

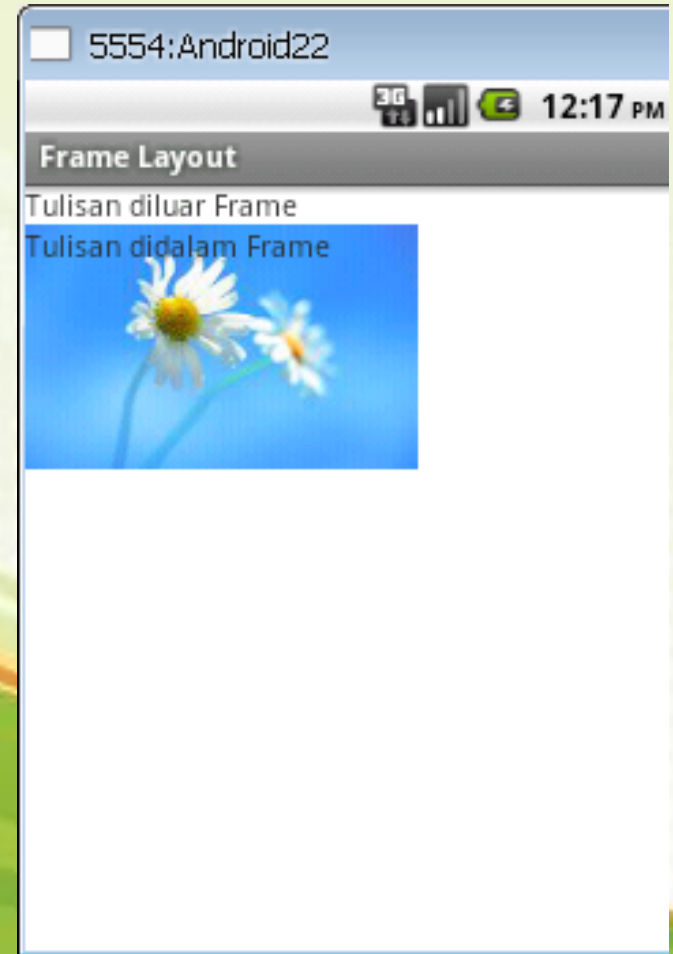
Tentu saja, saya tidak bisa hidup tanpamu 😊



Contoh FrameLayout

Perhatikan semantik dibawah ini

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:id="@+id/RLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <TextView
        android:id="@+id/lblComments"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tulisan diluar Frame"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true" />
    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/lblComments"
        android:layout_below="@+id/lblComments"
        android:layout_centerHorizontal="true" >
        <ImageView
            android:src="@drawable/img"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Tulisan didalam Frame" />
    </FrameLayout>
</RelativeLayout>
```



ScrollView

ScrollView memungkinkan aplikasi dapat melakukan scrolling ke arah vertical. Digunakan jika layout terdapat view yang banyak. Kenapa harus menggunakan ScrollView? Dikarenakan android tidak dapat men-generate otomatis aplikasi ber-scroll. Untuk penggunaannya, tempatkan Layout didalam ScrollView.

ScrollView

Layout

Mirip kayak yang tadi ya!
Tapi terbalik 😊

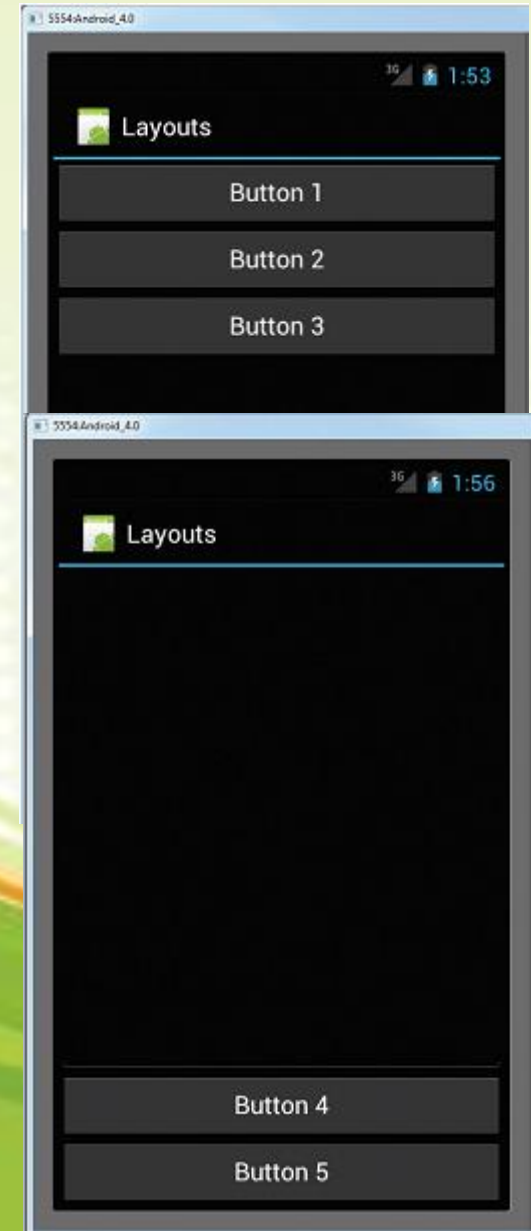
Iya bu



Contoh ScrollView

Perhatikan semantik dibawah ini

```
<ScrollView
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/android" >
  <LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <Button
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Button 1" />
    <Button
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Button 2" />
    <Button
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Button 3" />
    <EditText
      android:layout_width="fill_parent"
      android:layout_height="600dp" />
    <Button
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Button 4" />
    <Button
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Button 5" />
  </LinearLayout>
</ScrollView>
```



Terima Kasih

