

## MODUL 3

### Membuat Desain Layouting Android

#### a. Menulis XML

Extensible Markup Language (XML) adalah notasi untuk menulis file yang berisi informasi yang disebut element. Sebagai contoh sebuah file yang menggambarkan tata letak layar yang mungkin mengandung unsur - unsur tombol dan gambar. Awal setiap element di tandai dengan tag yang di kelilingi dengan karakter < dan >.

Didalam element terdapat potong - potongan informasi kecil yang disebut dengan attribute yang ditulis dalam tag awal. Setiap attribute masing - masing terdiri oleh nama dan nilai. Sebagai contoh perhatikan gambar berikut.



Pada gambar di atas terdapat elemen TextView dengan attribute **android:layout\_width** sebagai nama attributenya dan **“match\_parent”** sebagai nilai dari attributenya.

Dalam membuat aplikasi android xml digunakan untuk membuat tampilan. Pada dasarnya kode - kode xml yang kita buat nantinya akan di convert kedalam bit java. Kita juga bisa membuat tampilan untuk aplikasi android menggunakan kode - kode java atau di class java, akan tetapi dengan kode java akan mengakibatkan kode yang sangat kompleks.

Dengan xml akan memudahkan kita untuk membuat tampilan sebuah aplikasi yang kita buat. Didalam menulis kode xml ada beberapa aturan yang akan kita gunakan untuk memudahkan kita untuk membuat layout dengan xml.

## **b. Root Layout, View Group, LinearLayout dan RelativeLayout, FrameLayout**

Untuk membuat sebuah tampilan di aplikasi android kita memerlukan layout. Layout adalah sebuah tampilan dari sebuah aplikasi. Didalam membuat tampilan atau layouting di android ada dua komponen yang perlu diperhatikan diantaranya adalah root layout dan view group.

### **1. Root Layout**

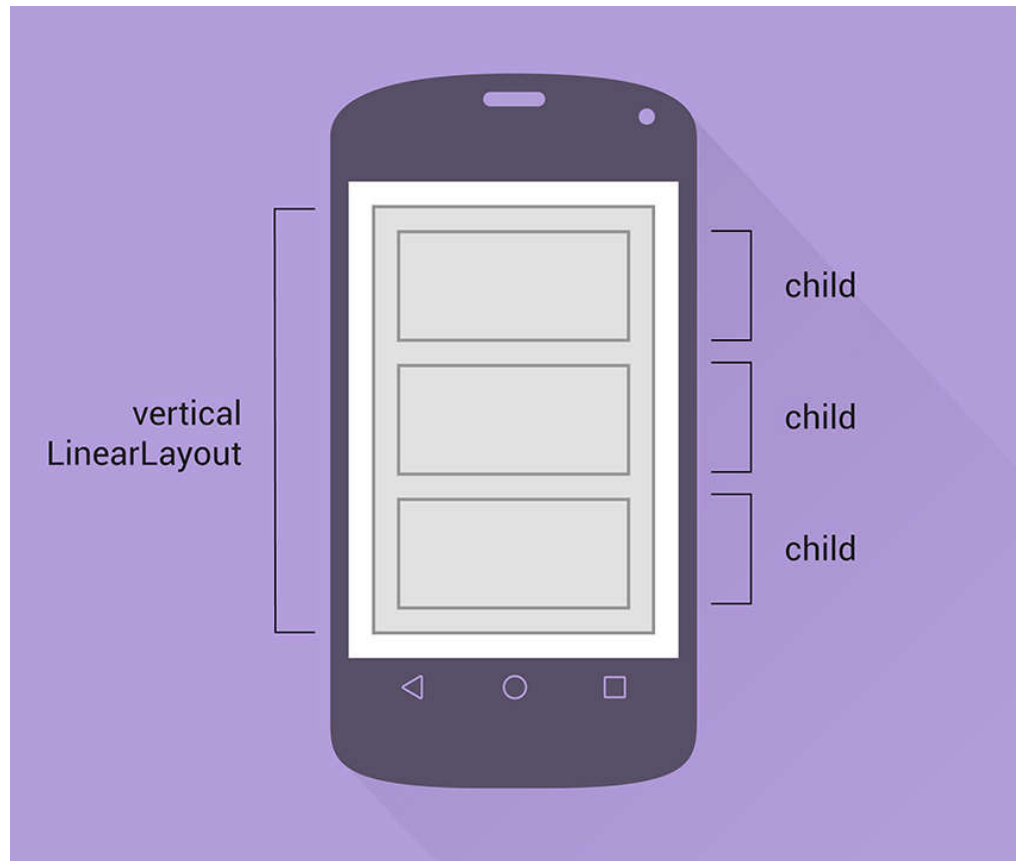
Root layout atau juga bisa disebut parent layout biasanya ditandai dengan kode

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

untuk root view adalah element penting didalam sebuah layout di aplikasi android. Setiap kita membuat sebuah layout maka root view harus ada didalamnya, karna root layout akan di jadikan parent dimana setiap view yang akan kita buat nantinya akan berada didalam parent tersebut.

### **2. View Group**

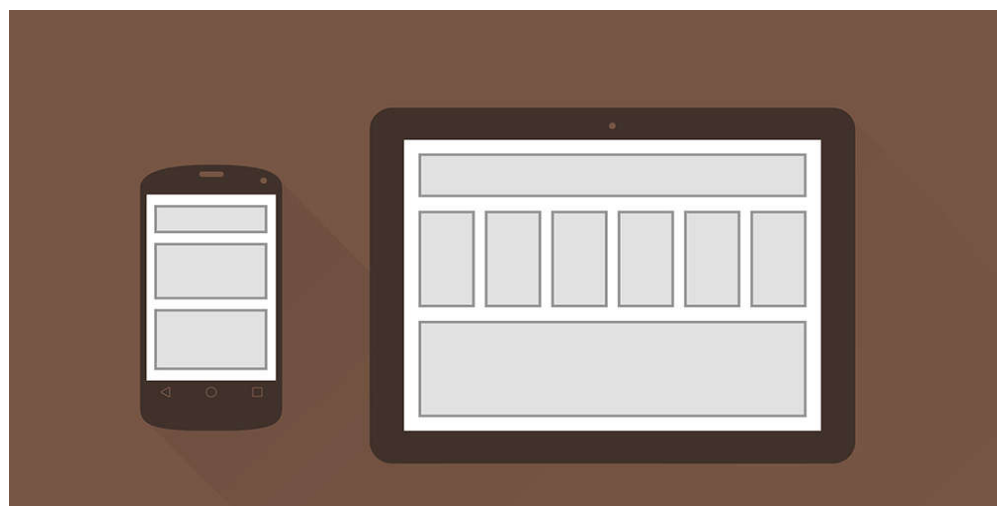
View group adalah sebuah view yang memilki dua elemen yaitu element parent dan child. Berikut ini adalah ilustrasi dari view group.



Pada gambar diatas vertical LinearLayout adalah parent layout dan 3 elemen child adalah child layoutnya.

### 3. LinearLayout

LinearLayout adalah salah satu element xml yang berfungsi untuk membuat layout yang bersifat vertical atau horizontal.



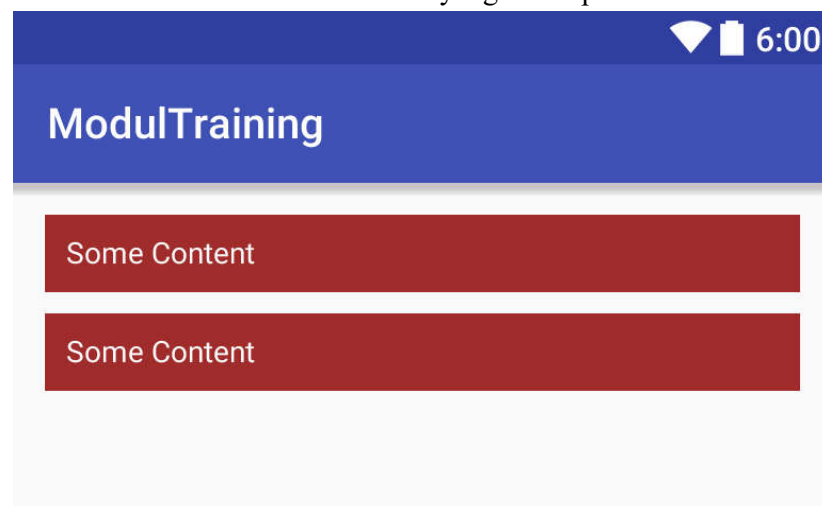
Diatas adalah ilustrasi contoh dari layout vertical dan horizontal dari LinearLayout. Selanjutnya berikut ini adalah contoh code xml.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Some Content"
        android:textColor="#FFFFFF"
        android:layout_margin="5dp"
        android:padding="10dp"
        android:background="#a02c2c"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Some Content"
        android:textColor="#FFFFFF"
        android:layout_margin="5dp"
        android:padding="10dp"
        android:background="#a02c2c"/>
</LinearLayout>
```

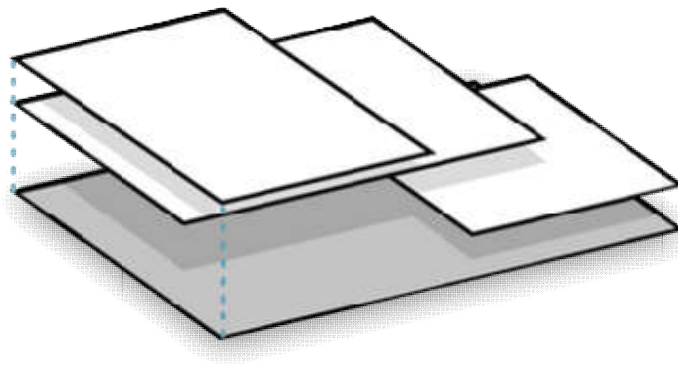
Dibawah ini adalah hasil dari kode yang di tampilkan dari kode di atas.



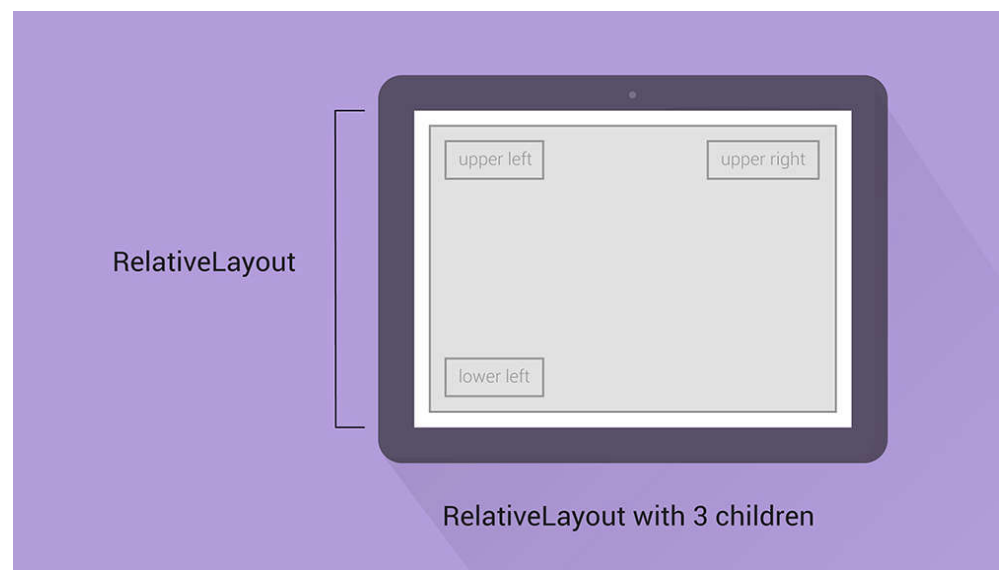
### 1. FrameLayout

Layout ini adalah layout yang paling sederhana. Layout ini akan membuat komponen yang ada didalamnya menjadi menumpuk atau saling menutupi

satu dengan yang lainnya (layering). Komponen yang paling pertama pada layout ini akan berada dibawah komponen-komponen diatasnya. FrameLayout memiliki kemampuan untuk menjadi container untuk fragment-fragment didalam sebuah Activity. Berikut ilustrasi dari penggunaan FrameLayout terhadap child view yang dimiliki didalamnya. Untuk layout ini kita akan bahas pada bab selanjutnya.



## 2. RelativeLayout



RelativeLayout adalah element xml yang berfungsi untuk membuat layout yang relative pada aplikasi android. Maksudnya relative disini adalah sebuah layout yang memungkinkan kita mengatur elemen secara relative jika di LinearLayout terbatas hanya untuk posisi horizontal dan vertical maka di RelativeLayout bisa lebih dari sekedar itu. Didalam RelativeLayout ada beberapa komponen yaitu parent view adalah RelativeLayout nya dan child

view adalah view yang ada didalam relative layout itu. yang ada didalam Berikut ini adalah contoh dari layout relative.

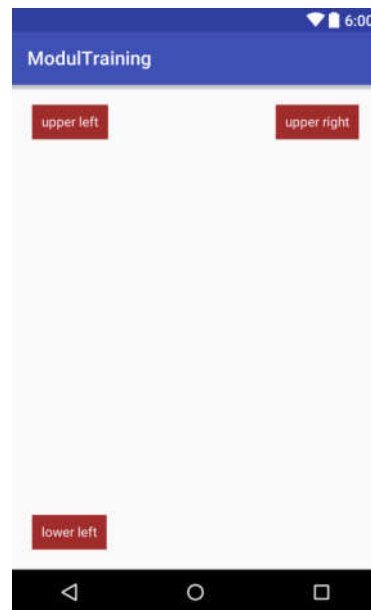
```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:text="upper left"
        android:textColor="#FFFFFF"
        android:layout_margin="5dp"
        android:padding="10dp"
        android:background="#a02c2c" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:text="upper right"
        android:textColor="#FFFFFF"
        android:layout_margin="5dp"
        android:padding="10dp"
        android:background="#a02c2c" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:text="lower left"
        android:textColor="#FFFFFF"
        android:layout_margin="5dp"
        android:padding="10dp"
        android:background="#a02c2c" />
</RelativeLayout>
```

Jika kode diatas dijalankan maka akan jadi seperti berikut ini tampilanya

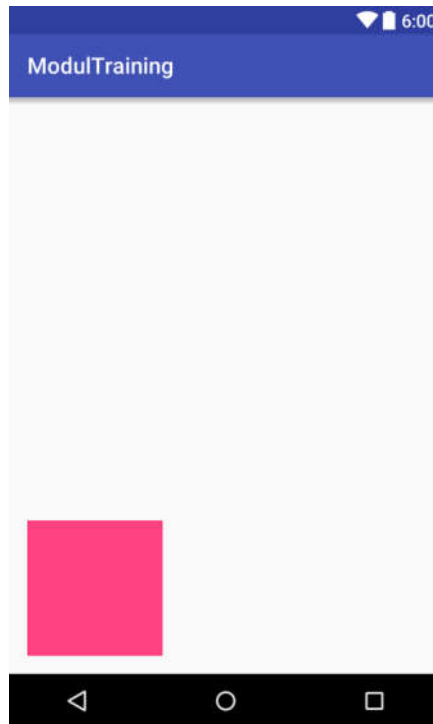


Ada beberapa attribute relative layout yang hanya bisa di pakai dalam child view (*anak dari view relative layout atau view yang berada dalam relative layout*) yang parent nya adalah relative layout.

1. `android:layout_alignParentBottom="true"` berfungsi untuk membuat child view dari relative layout selalu dalam posisi bawah dari view relative layout. Berikut contohnya

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <View
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:background="#FF4081"
        android:layout_alignParentBottom="true"/>
</RelativeLayout>
```



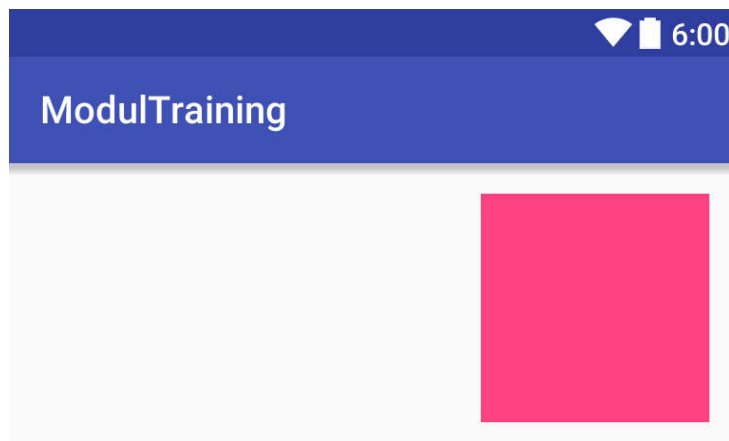
2. `android:layout_alignParentRight="true"` dan `android:layout_alignParentLeft="true"` berfungsi untuk membuat child view berada dalam posisi ke kanan dan ke kiri parent view/RelativeLayout. Berikut ini adalah contoh nya:

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <View
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:background="#FF4081"
        android:layout_alignParentRight="true"/>
</RelativeLayout>
```

dan berikut ini hasilnya





3. `android:layout_below="@id/view1"` yang berfungsi untuk membuat child view berada dibawah view berdasarkan id view yang didaftarkan.

Dan untuk `android:layout_above="@id/view1"` berfungsi kebalikan dari `layout_below`, jika `layout_below` adalah di bawahnya maka untuk `layout_Berikut` ini adalah contoh untuk `layout_below`:

<RelativeLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="16dp">
```

<TextView

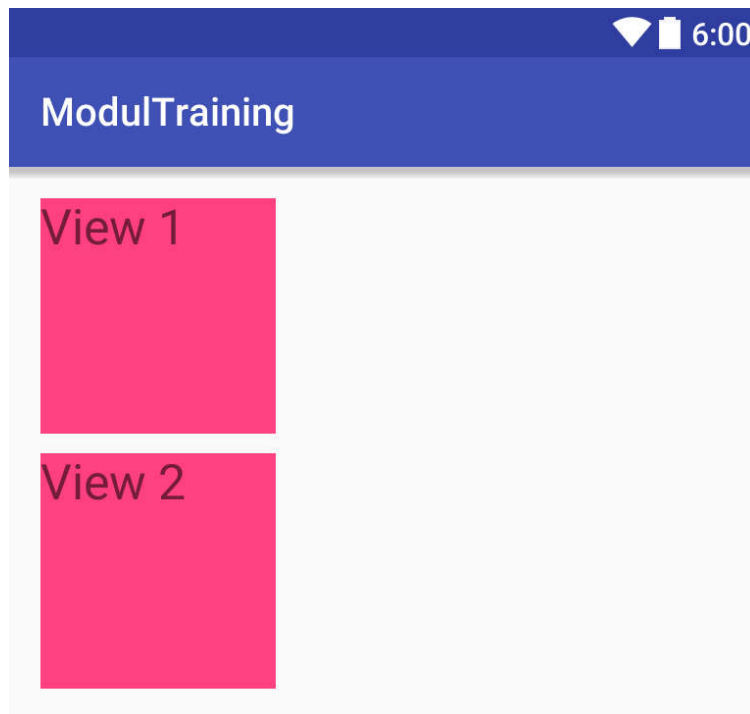
```
android:id="@+id/view1"
android:layout_width="120dp"
android:layout_height="120dp"
android:background="#FF4081"
android:text="View 1"
android:textSize="25dp"
android:layout_marginBottom="10dp"/>
```

<TextView

```
android:id="@+id/view2"
android:layout_width="120dp"
android:layout_height="120dp"
android:background="#FF4081"
android:layout_below="@id/view1"
android:text="View 2"
android:textSize="25dp"/>
```

</RelativeLayout>

Perlu diperhatikan untuk menggunakan `android:layout_below` kita harus membuat view yang diberikan id dengan cara `android:id="@+id/view1"` untuk nama id terserah sesuai dengan keinginan anda. Untuk kode diatas akan menghasilkan view seperti berikut ini.



Dan berikut ini untuk contoh `layout_above`

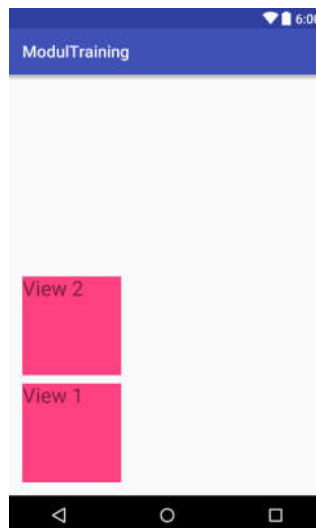
```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/view1"
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:background="#FF4081"
        android:text="View 1"
        android:textSize="25dp"
        android:layout_marginTop="10dp"
        android:layout_alignParentBottom="true"/>
```

```

<TextView
    android:id="@+id/view2"
    android:layout_width="120dp"
    android:layout_height="120dp"
    android:background="#FF4081"
    android:layout_above="@id/view1"
    android:text="View 2"
    android:textSize="25dp"/>
</RelativeLayout>

```



### c. TextView and ImageView

TextView dan ImageView adalah salah satu elemment yang harus kita pelajari untuk membuat tampilan pada aplikasi android dimana keduanya ini pasti akan sangat sering kita gunakan untuk membuat sebuah tampilan untuk aplikasi android.

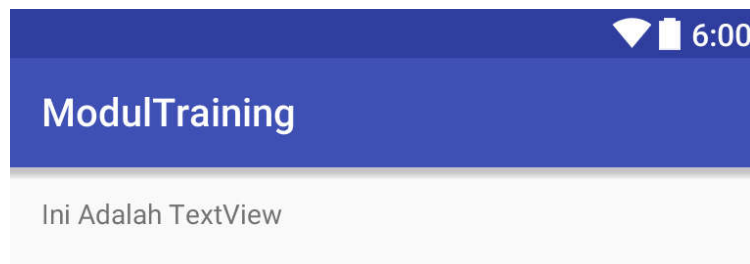
#### 1. TextView

TextView adalah elemment yang berfungsi untuk membuat tampilan yang berisikan text. Berikut ini adalah contoh kodenya

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="Ini Adalah TextView"/>

```



Pada contoh diatas kita dapati ada 3 attribute diantaranya :

1. **android:layout\_width** adalah nama dari attribute yang berfungsi untuk mengatur lebar dari sebuah textview itu sendiri, dan **"match\_parent"** adalah nilai yang menunjukkan bahwa width atau lebar dari TextView itu adalah matching dengan parent maksudnya adalah mengikuti panjang dari layout induknya atau bisa dibilang lebarnya 100%, seandainya layout parentnya adalah 500dp lebarnya makan lebar dari match\_parent juga 500dp.
2. **android:layout\_height** adalah attribute yang berfungsi untuk mengatur tinggi dari TextView. Jika kita atur nilainya adalah **"match\_parent"** maka tinggi dari TextView ini adalah mengikuti tinggi dari layout parent nya, dan jika diatur menjadi **"wrap\_content"** maka tinggi dari textview ini hanya sesuai dengan kebutuhan text yang akan ditampilkan.
3. **android:text** adalah attribute yang digunakan untuk membuat atau memberikan text. Dan **"Ini Adalah TextView"** adalah text yang akan ditampilkan.

## 2. ImageView

ImageView adalah elemment yang berfungsi untuk menampilkan gambar. Berikut ini adalah contohnya

**<ImageView**

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@mipmap/ic_launcher"/>

```

Sama seperti attribute textview diatas akan tetapi disini ada attribute `android:src` dan hanya bisa digunakan untuk element ImageView yang berfungsi untuk menunjukkan alaman sebuah Image/Gambar yang akan ditampilkan.

#### d. Button, RadioButton, Switch dan EditText

Didalam sebuah aplikasi, apapun itu baik aplikasi web, ios dan android semuanya memiliki tiga komponen penting diantaranya input, proses dan output. Pada bagian ini kita akan membahas tentang view input dalam aplikasi android.

##### 1. Button

Button adalah sebuah element xml yang berfungsi sebagai input control yang berupa tombol. Button adalah salah satu hal penting didalam aplikasi android. Karena banyak aktivitas pada aplikasi Android yang membutuhkan Button untuk mengeksekusinya, seperti Login, Logout, Pencarian, atau Menu. Button dapat berisi tulisan, gambar, atau kombinasi keduanya.

Berikut ini adalah contoh membuat button dalam xml :

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Submit"/>
```

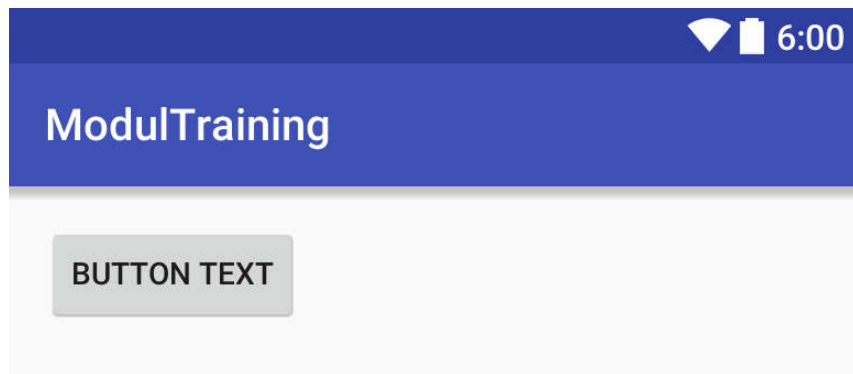
Sama seperti view yang lainnya attribute layout\_width dan layout\_height adalah attribute yang wajib ada. Button juga memiliki beberapa macam diantaranya adalah ButtonText, ImageButton, ButtonText dan Image.

##### a. ButtonText

Button text adalah salah satu button yang paling sering digunakan. Berikut ini adalah contohnya

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button Text"/>
```

simple dan mudah berikut adalah hasilnya

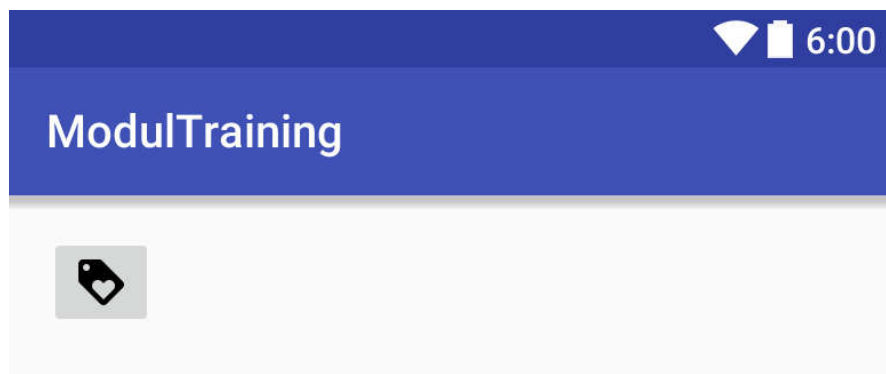


b. ImageButton

Button image adalah solusi untuk membuat tombol yang menggunakan image. Berikut adalah contoh:

<ImageButton

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_loyalty_black_24dp"/>
```

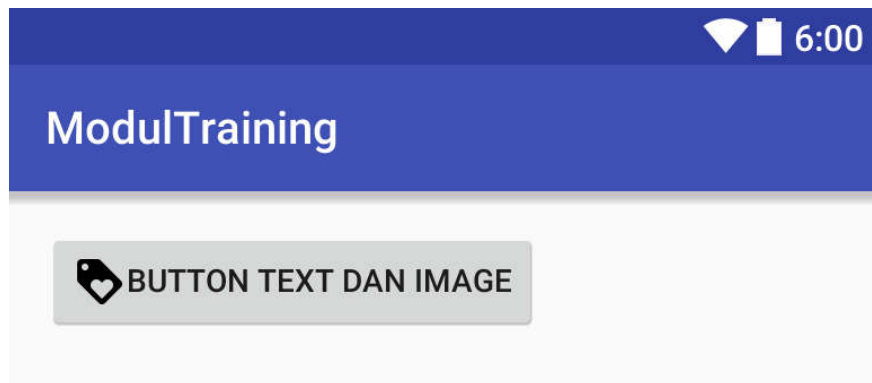


c. ButtonText dan Image

Jika ingin membuat tombol yang mengandung text dan image bisa menggunakan Button Text dan Image berikut contohnya

<Button

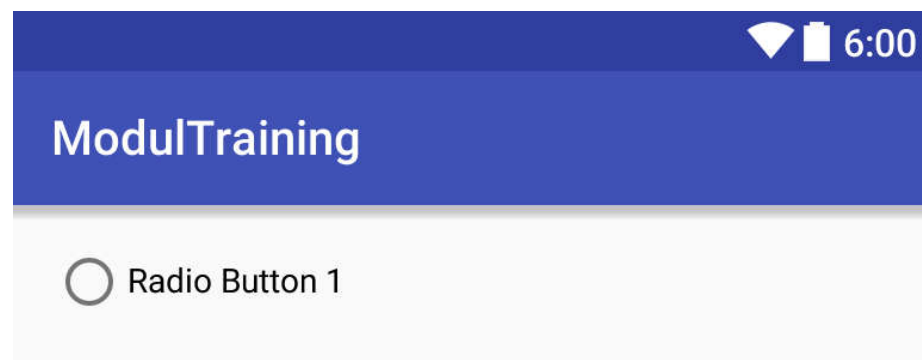
```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button Text dan Image"
    android:drawableLeft="@drawable/ic_loyalty_black_24dp"/>
```



## 2. RadioButton

RadioButton adalah element xml yang juga termasuk input control yang berupa list pilihan. Berikut ini adalah contoh nya:

```
<RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Radio Button"/>
```



Berikut ini jika ingin membuat lebih dari satu radio button

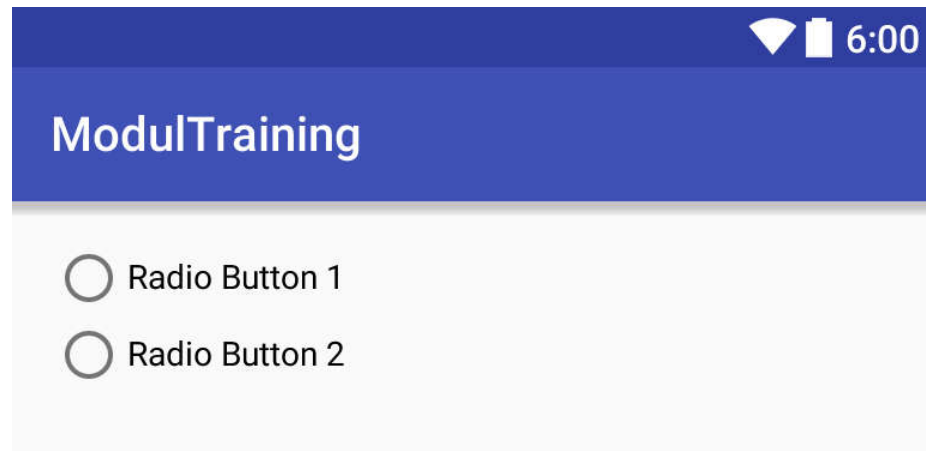
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:orientation="vertical">

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Radio Button 1"/>

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="Radio Button 2"/>
    </LinearLayout>
```

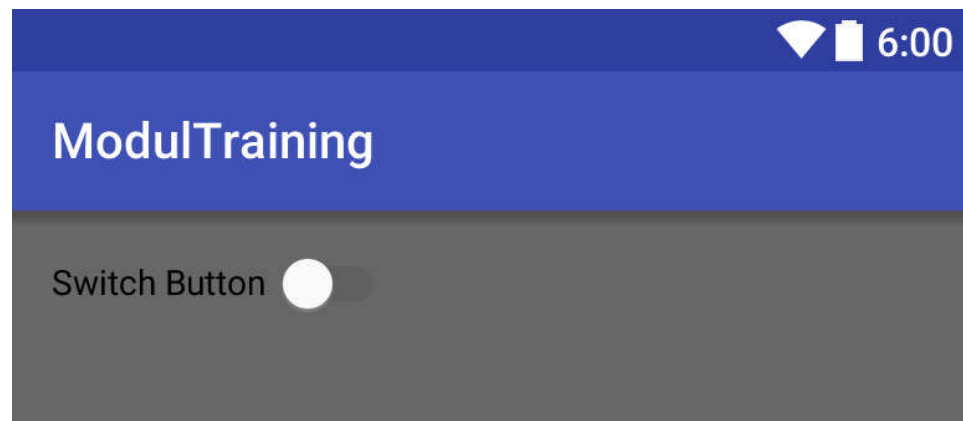
dan berikut ini hasilnya



### 3. Switch Button

Switch button adalah element yang berfungsi untuk membuat pilihan true atau false berikut ini adala contoh penggunaanya

```
<Switch
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Switch Button  " />
```



### 4. EditText

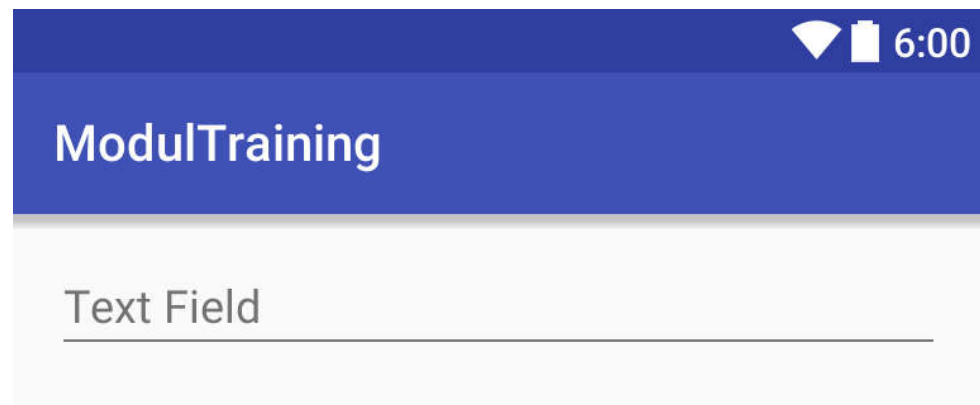
EditText adalah salah element yang berfungsi untuk menampilkan text field atau inputan text. Biasanya digunakan untuk mengisi biodata, email,



password maupun membuat postingan yang berupa text. Dan berikut contohnya

<EditText

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Text Field"/>
```



EditText atau juga disebut Text Field memiliki beberapa type text inputan. Berikut ini tipe-tipe input pada Text Fields :

1. “text” : Karakter keyboard normal
2. “textEmailAddress” : Karakter keyboard normal dengan karakter ‘@’ pada salah satu tombolnya

<EditText

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:inputType="textEmailAddress"/>
```

3. “textUri” : Karakter keyboard normal dengan karakter ‘/’ pada salah satu tombolnya

<EditText

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Text Uri"
    android:inputType="textUri"/>
```

4. “number” : Karakter angka pada Text Fields dan keyboard

<EditText

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```

    android:hint="Number"
    android:inputType="number"/>

```

5. “phone” : Keyboard seperti keypad telpon

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Phone"
    android:inputType="phone"/>

```

6. “textCapSentences” : Karakter keyboard normal dengan huruf besar pada setiap awal kalimat.

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Capital Sentences"
    android:inputType="textCapSentences"/>

```

7. “textCapWords” : Karakter keyboard normal dengan huruf besar pada setiap kata. Cocok digunakan untuk judul atau nama.

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Capital Words"
    android:inputType="textCapWords"/>

```

8. “textAutoCorrect” : Karakter keyboard normal yang mengoreksi kata-kata yang salah (kata dalam Bahasa Inggris).

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Auto Correct"
    android:inputType="textAutoCorrect"/>

```

9. “textPassword” : Karakter normal keyboard dengan karakter yang disembunyikan dalam titik.

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:inputType="textPassword"/>

```

10. “textMultiLine” : Karakter normal keyboard untuk tulisan yang panjang dengan memperbolehkan ganti baris.

```

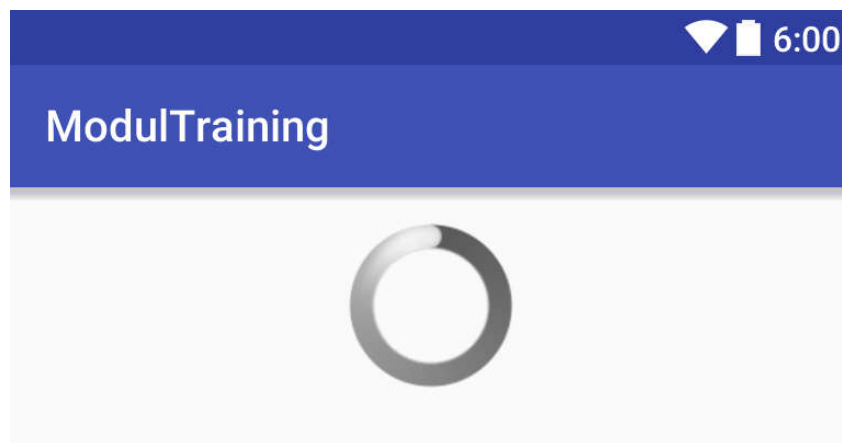
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Multi Line"
    android:inputType="textMultiLine"/>

```

### e. ProgressBar

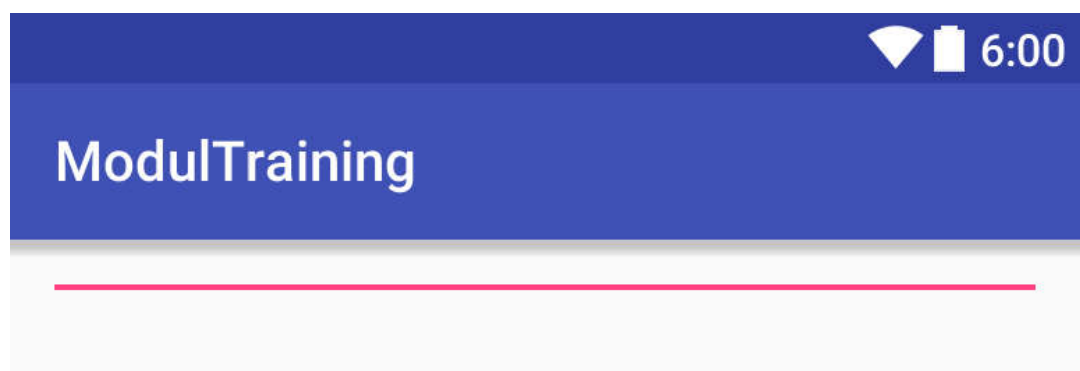
ProgressBar adalah view yang berfungsi untuk menampilkan progres. Biasanya digunakan untuk keperluan loading data misalnya ketika kita sedang mengambil data dari internet maka bisa menggunakan progressbar untuk sebagai penanda jika kita sedang mengambil atau meload data.

```
<ProgressBar  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    style="@android:style/Widget.ProgressBar.Large"/>
```



### f. Views

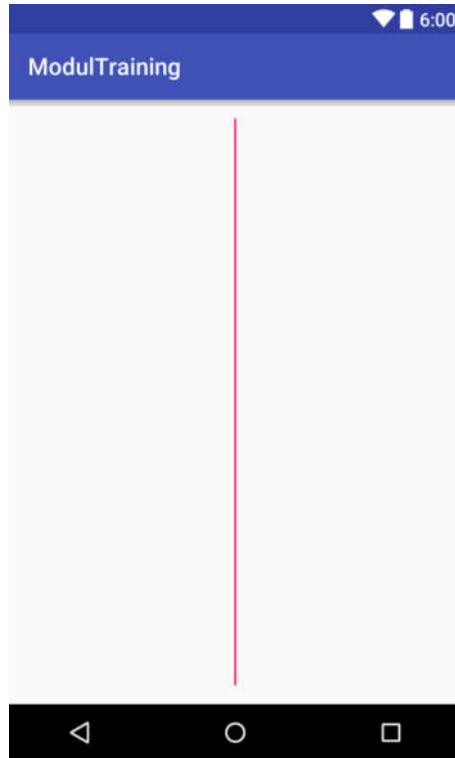
View adalah element xml yang paling basic dalam android biasanya digunakan untuk membuat sebuah view garis seperti berikut ini :



Pada view diatas hanya menampilkan garis dan tidak menampilkan yang lainnya dan untuk attribute dari view ini sangat terbatas. Berikut kodenya

```
<View  
    android:layout_width="match_parent"  
    android:layout_height="2dp"  
    android:background="@color/colorAccent"/>
```

Atau kita juga bisa membuat garis vertikal



Berikut ini kodenya

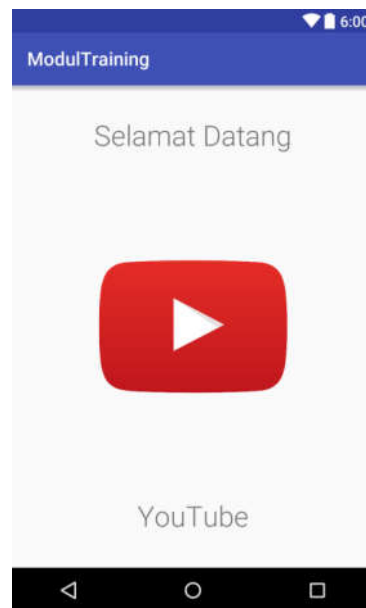
```
<View  
    android:layout_gravity="center"  
    android:layout_width="2dp"  
    android:layout_height="match_parent"  
    android:background="@color/colorAccent"/>
```

Pada intinya view adalah view kosong

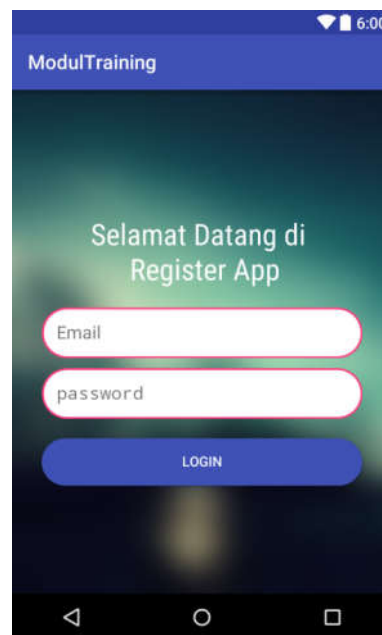
**g. Latihan (Tugas Mandiri)**

Buatlah layout seperti gambar berikut:

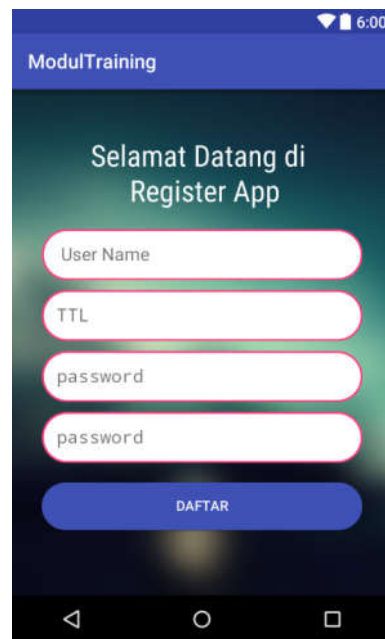
- a. YouTube splash screen



- b. Login app



## c. Register App



## d. CourtCounter App

