

# App Components and User Interface

Presented By Putri Hayati, S.ST, M.Kom





# Kategori

2.0 Komponen Aplikasi Android

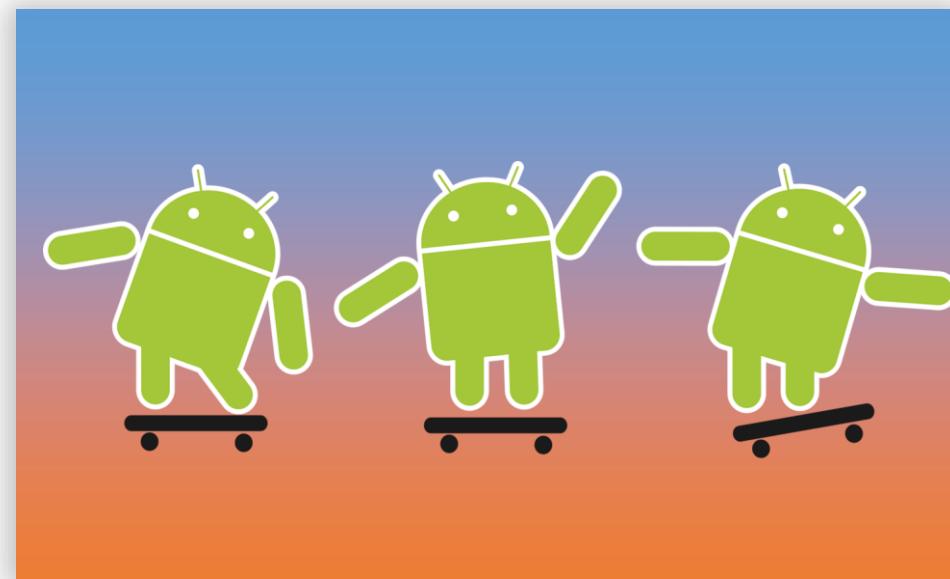
2.1 Antarmuka Pengguna  
(User Interfaces)

Activity

View

Layout

Resource





# Komponen Aplikasi Android



## 2.0 Komponen Aplikasi Android

Mengirim atau menerima sebuah perintah dari aplikasi lain

```
public void sendMessage(View view) {  
    Intent intent = new Intent(this, DisplayMessageActivity.class);
```

Untuk menampilkan antarmuka aplikasi. *Activity* diimplementasikan sebagai *subclass* dari *class Activity*

```
public class MainActivity  
extends Activity { }
```

Untuk komunikasi antar *activity*, diimplementasikan sebagai *subclass* dari *class* dan setiap pesan akan dikirim dan diterima sebagai objek *Intent*.

```
public class MyReceiver  
extends BroadcastReceiver {  
public void  
onReceive(context,intent){} }
```



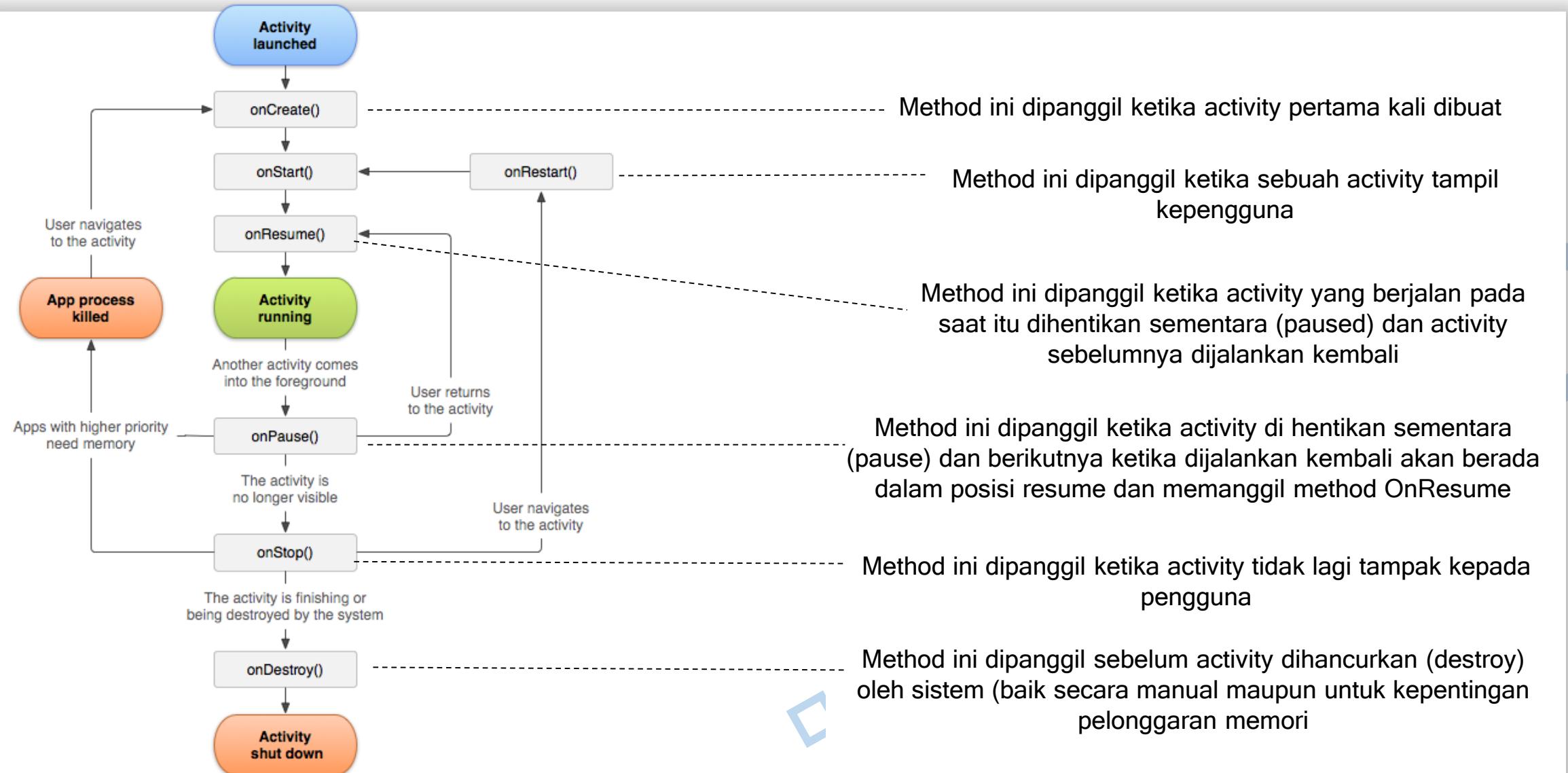
*Service* untuk menjalankan proses aplikasi di belakang layar. *Service* diimplementasikan sebagai *subclass* dari *class Service*

```
public class MyService  
extends Service { }
```

Untuk manajemen data dan *database*. diimplementasikan sebagai *subclass* dari *class Content Provider* dan harus menerapkan satu set standar API yang memungkinkan aplikasi lain untuk melakukan transaksi

```
public class MyReceiver extends  
BroadcastReceiver { public void  
onReceive(context,intent){} }
```

# Android - Activity Lifecycle





# User Interface - Antarmuka Pengguna



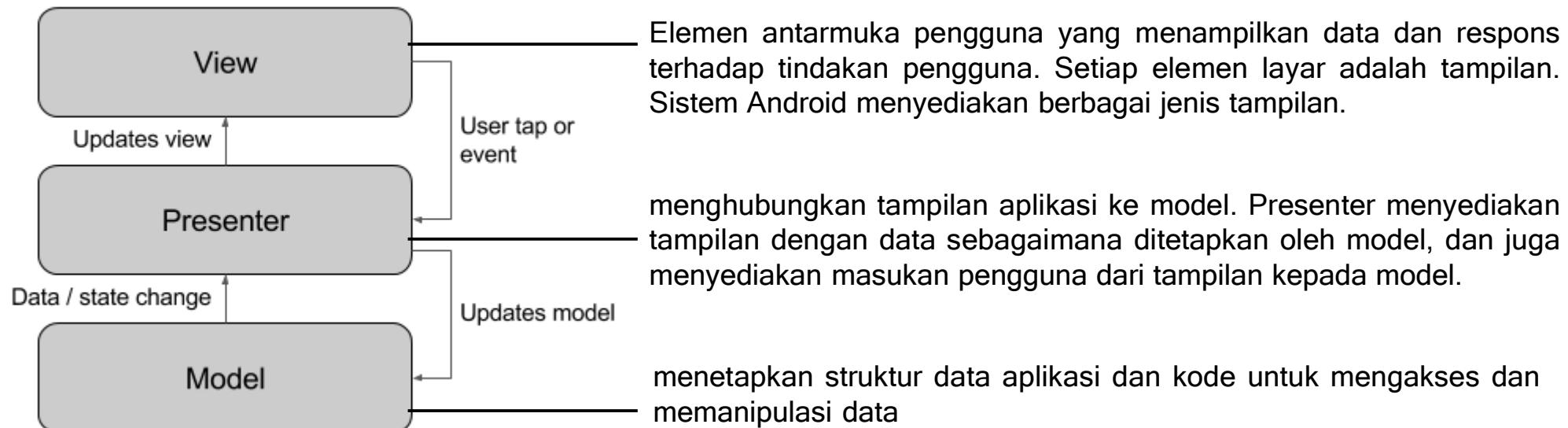
## 2.1 User Interface



- Antarmuka pengguna (UI) untuk aplikasi Android dibuat sebagai hierarki *tata letak* dan *widget*.
- Tata letak adalah objek [ViewGroup](#), container yang mengontrol cara tampilan turunannya diposisikan di layar.
- Widget adalah objek [View](#), komponen UI seperti tombol dan kotak teks.

## Pola model-view-presenter

Aktivitas ke sumber daya layout dari bagian pola arsitektur [model-view-presenter](#)(MVP)





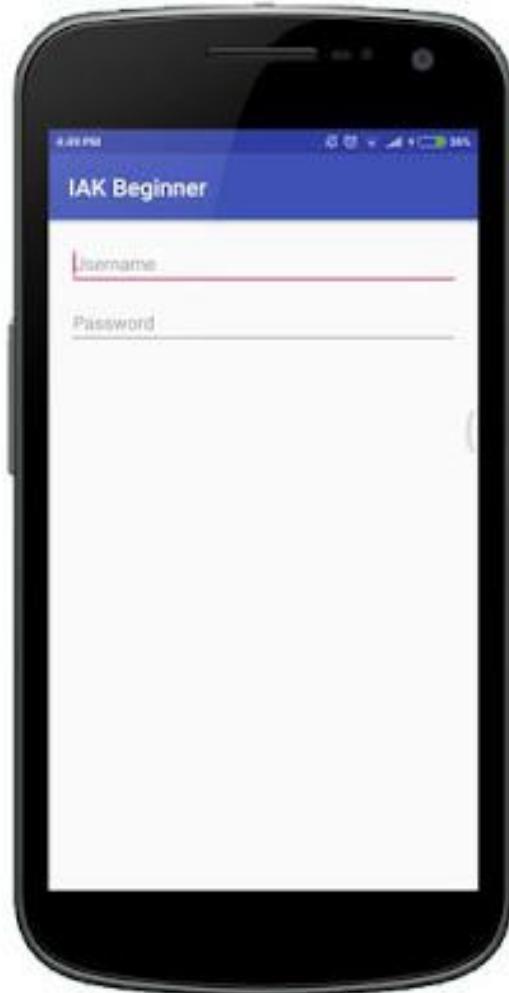
### Tampilan-View

UI terdiri dari hierarki objek yang disebut *tampilan*, setiap elemen layar adalah *tampilan*. Kelas [View](#) menyatakan blok pembangunan dasar untuk semua komponen UI, dan kelas dasar untuk kelas yang menyediakan komponen UI interaktif seperti tombol, kotak centang, dan bidang entri teks.

→ **TextView** - Untuk menampilkan output berupa text pada UI sesuai dengan tampilan gambar disamping atributnya :

- **text** : Text yang ditampilkan
- **textsize** : Ukuran Text
- **textcolor** : Warna Text
- **textALLCAPS** : Menampilkan text semua huruf capital
- **textStyle** : (Normal, Bold, Italic, BoldItalic) untuk text

### Tampilan-View



**EditText**, merupakan elemen UI untuk memasukan dan modifikasi text.  
Atributnya :

```
<EditText  
    android:id="@+id/editText_nama"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="Username"  
    android:inputType="text"  
/>  
<EditText  
    android:id="@+id/editText_password"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="Password"  
    android:inputType="text|textPassword"  
/>
```

### Tampilan-View



**Button**, UI elemen dimana user dapat meng-klik atau melakukan tap untuk menghasilkan sebuah action  
Atributnya :

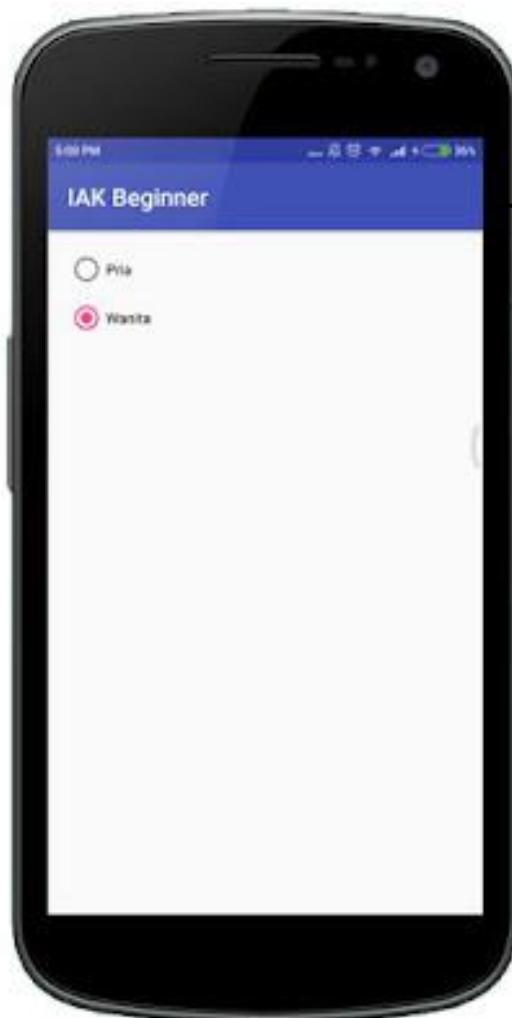
```
<Button  
    android:id="@+id/button_click"  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:layout_centerInParent="true"  
    android:text="CLICK HERE"  
/>
```

### Tampilan-View



→ **ImageView**, elemen yang biasa digunakan untuk menampilkan image file ke dalam aplikasi.  
Atributnya :

```
<android.support.v7.widget.AppCompatImageView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:src="@drawable/sydney"  
    android:scaleType="centerCrop"  
/>
```



### Tampilan-View

RadioGroup dan RadioButton, elemen yang telah disediakan oleh Android Studio yang biasa digunakan untuk menampilkan pilihan yang akan dipilih berupa satu lingkaran kecil dengan titik ditengahnya yang nantinya akan digunakan sebagai opsi input.

Atributnya :

```
<RadioGroup  
    android:id="@+id/rgrp_main1"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
<RadioButton  
    android:id="@+id/rbtn_pria"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="Pria" />  
  
<RadioButton  
    android:id="@+id/rbtn_wanita"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="Wanita" />  
</RadioGroup>
```

### Tampilan-View



**CheckBox**, menampilkan pilihan dimana user bisa memilih lebih dari satu (multiple choice). Atributnya :

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hobi : " />  
  
<CheckBox  
    android:id="@+id/cbgame"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:checked="true"  
    android:text="Main Game" />  
  
<CheckBox  
    android:id="@+id/cb_makan"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:checked="true"  
    android:text="Makan" />  
  
<CheckBox  
    android:id="@+id/komik"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Baca Komik" />
```

### Tampilan-View

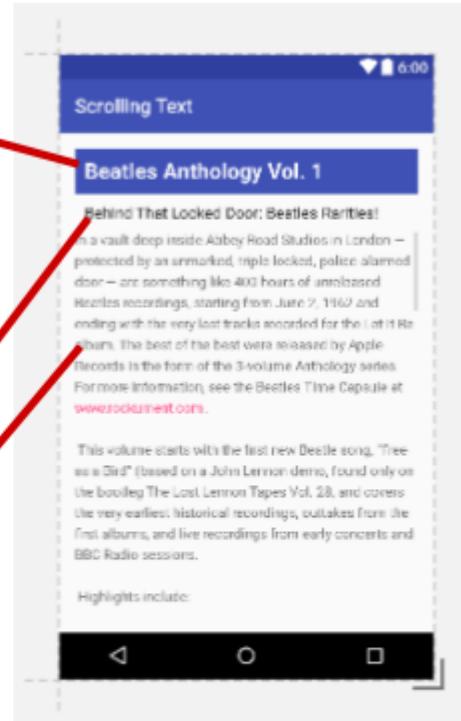
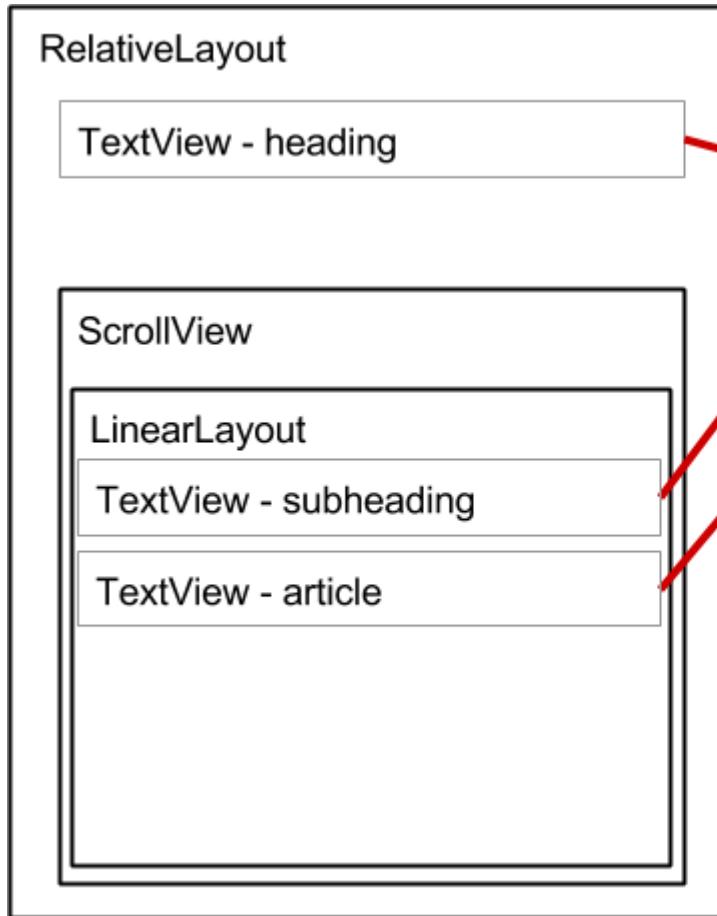


**Spinner**, untuk menampilkan list data dari salah satu data yang ditampilkan. Atributnya :

```
<resources>
    <string-array name="hobi">
        <item>Makan</item>
        <item>Main Game</item>
        <item>Baca Komik</item>
    </string-array>
</resources>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hobi"/>
<Spinner
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:entries="@array/hobi">
</Spinner>
```

### Tampilan-View

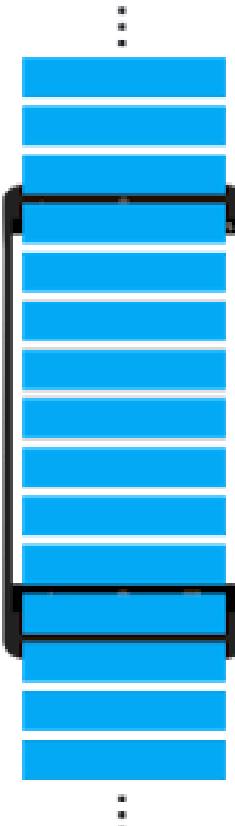


**ScrollView** hanya bisa berisi satu tampilan; akan tetapi, tampilan tersebut bisa berupa grup tampilan yang berisi beberapa tampilan

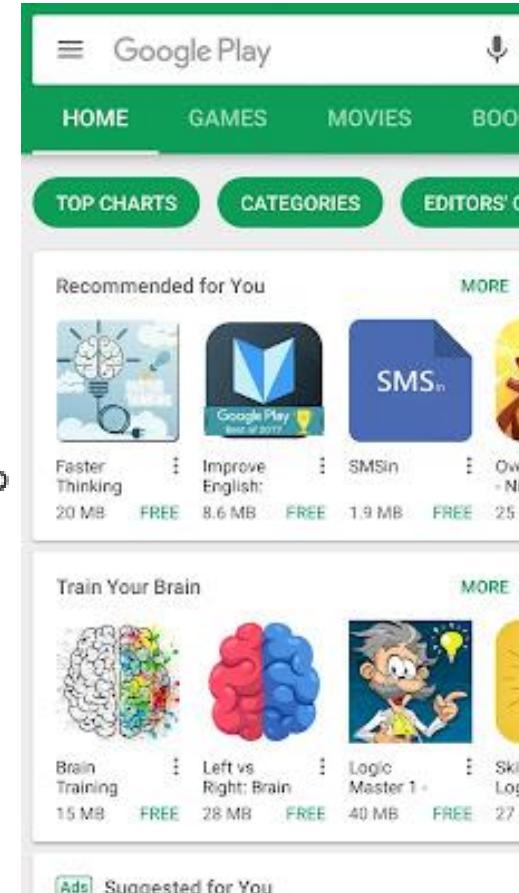
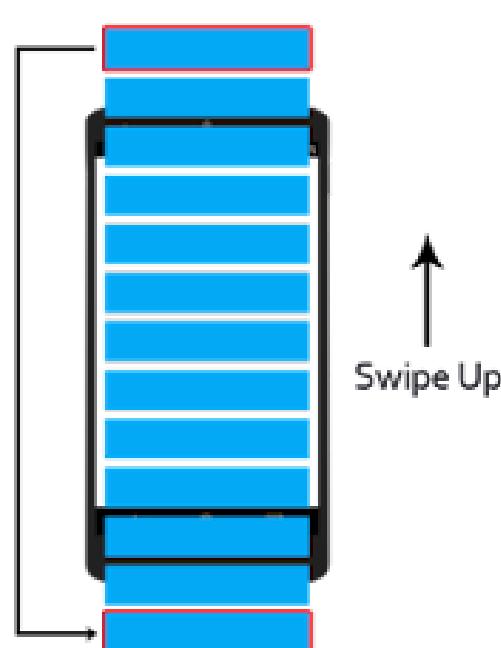
```
<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/article_heading">
```

### Tampilan-View

Listview



RecyclerView

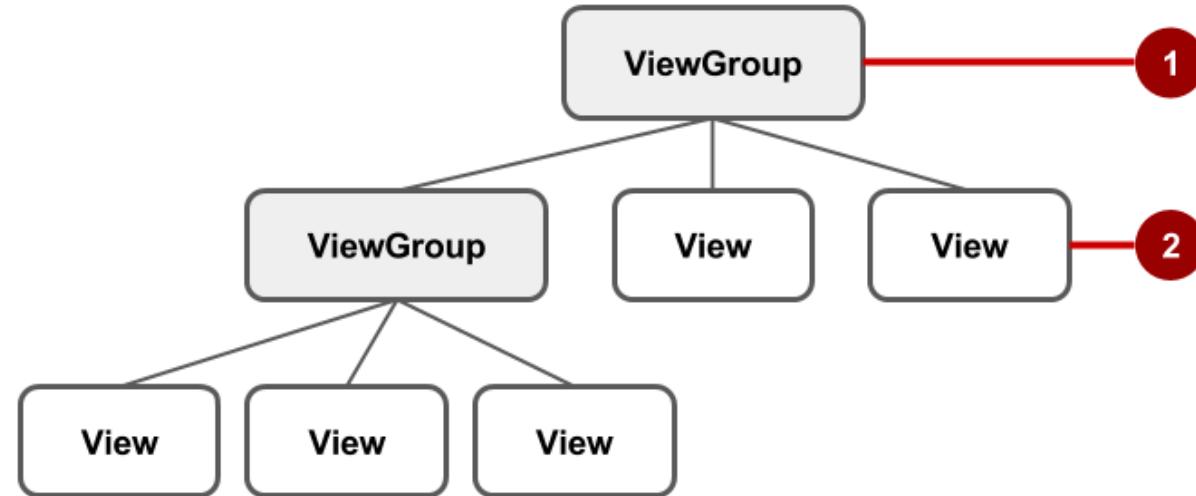


Untuk menggunakan recyclerView,  
Anda harus melakukan beberapa  
langkah, yaitu :

- Menambahkan support library RecyclerView di gradle.
- Menambahkan RecyclerView di layout XML
- Membuat row layout untuk setiap row item.
- Membuat ViewHolder untuk menyediakan referensi ke View pada setiap item data.
- Membuat Recycler View Adapter untuk memberi data ke RecyclerView
- Mem-bind Adapter ke RecyclerView di Activity

## Grup Tampilan Layout

Tampilan untuk layar dikelola dalam hierarki. Di *akar* hierarki ini adalah [ViewGroup](#) yang berisi layout keseluruhan layar. Layar anak grup tampilan bisa menjadi tampilan atau grup tampilan lain seperti yang ditampilkan dalam gambar berikut.:



Dalam gambar di atas:

1. Grup tampilan *akar*.
2. Rangkaian tampilan anak dan grup tampilan pertama yang induknya adalah akar.

### Tampilan - Layout



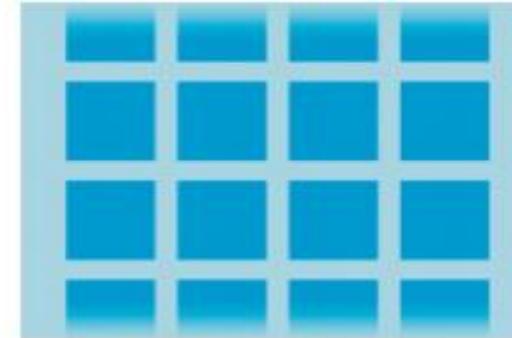
**LinearLayout**

Grup tampilan anak yang diposisikan dan disejajarkan secara horizontal atau secara vertikal.



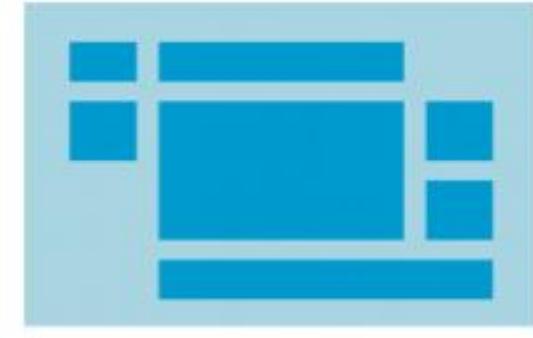
**RelativeLayout**

Grup tampilan anak yang setiap tampilannya diposisikan dan disejajarkan relatif terhadap tampilan dalam grup tampilan. Dengan kata lain, posisi tampilan anak bisa dijelaskan dalam hubungan satu sama lain atau dengan grup tampilan induk.



**GridLayout**

Grup yang menempatkan layar anaknya dalam kotak persegi panjang yang bisa digulir.

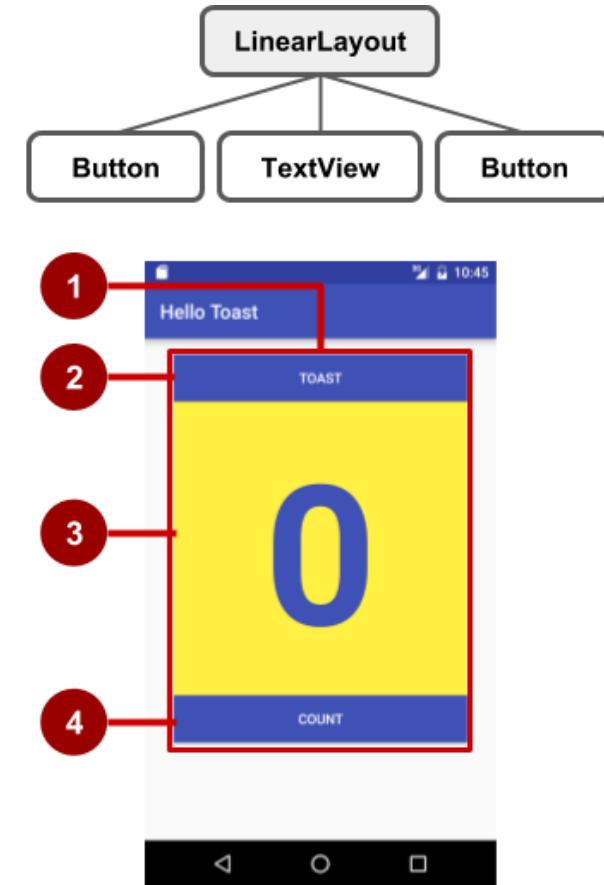
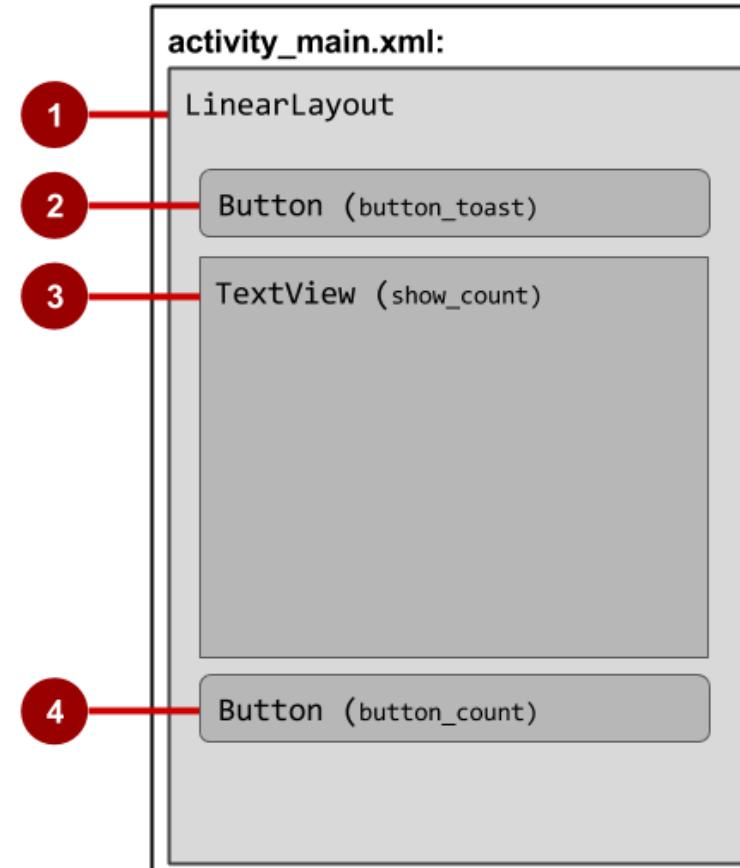


**TableLayout**

Grup tampilan anak yang disusun ke dalam baris dan kolom.

## Tampilan - Layout

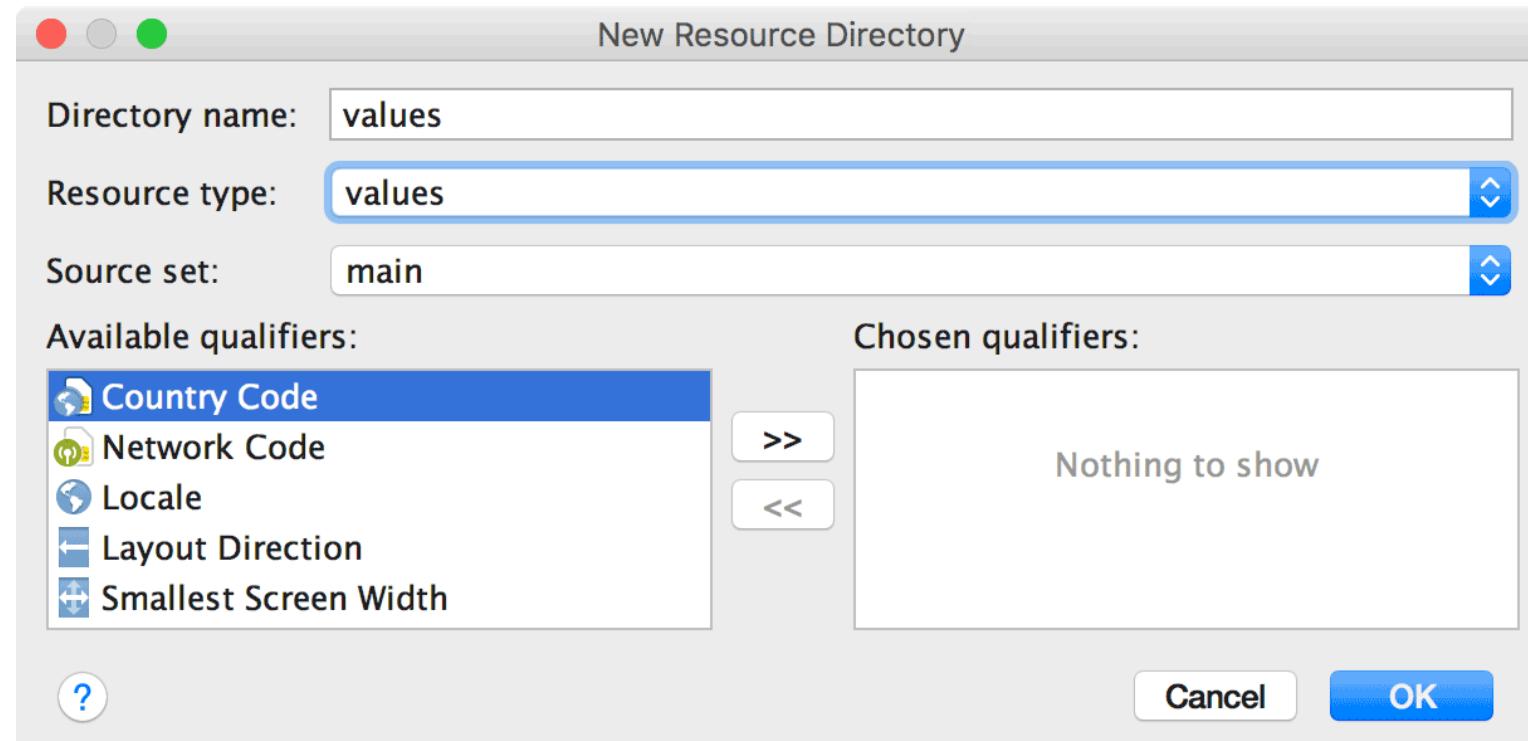
1. Layout akar LinearLayout, yang berisi semua tampilan awal, disetel ke orientasi vertikal.
2. Button (button\_toast) tampilan awal Sebagai tampilan pertama, muncul di bagian atas di layout linear.
3. TextView (show\_count) tampilan awal. Sebagai tampilan kedua, muncul di bawah tampilan anak pertama di layout linear.
4. Button (button\_count) tampilan anak. Sebagai tampilan anak ketiga, muncul di bawah tampilan anak kedua di layout linear.



## Tampilan - Resource

Jika perlu menambahkan direktori resource baru, ikuti langkah-langkah berikut :

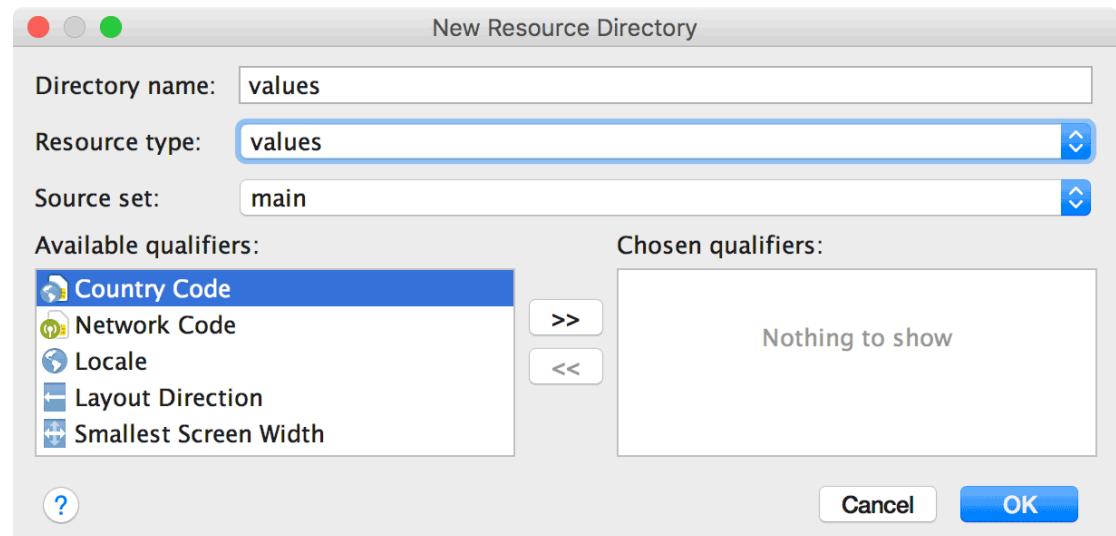
1. Klik modul aplikasi target pada jendela Project, lalu pilih **File > New > Android resource directory**.



## Tampilan - Resource

Lengkapi detail dalam dialog:

- Directory name: Direktori harus dinamai dengan cara yang spesifik untuk jenis resource dan kombinasi pengontrol kualitas konfigurasi. Jadi, Anda tidak boleh mengedit ini kecuali jika Anda ingin menambahkan pengontrol kualitas konfigurasi ke nama direktori secara manual (sebagai gantinya, gunakan Available qualifiers).
- Resource type: Pilih jenis resource yang ingin Anda buat.
- Source set: Pilih set sumber tempat Anda ingin menyertakan tata letak.
- Available qualifiers: Anda dapat menambahkan pengontrol kualitas konfigurasi untuk direktori tata letak dengan mengklik penentu, bukan mengetiknya, dari daftar di sebelah kiri, lalu mengklik Add .
- Setelah Anda menambahkan semua pengontrol kualitas yang diinginkan, klik OK.

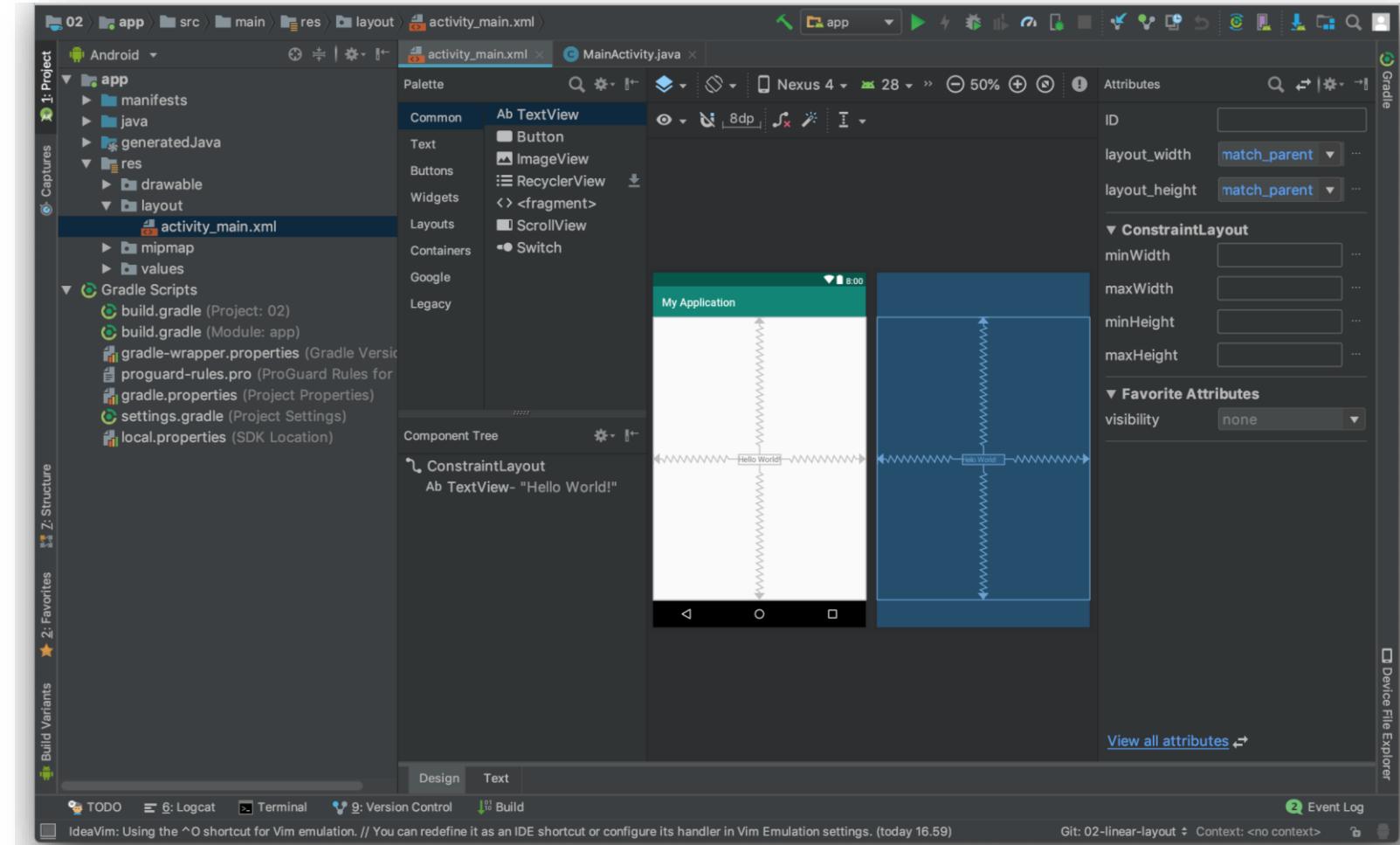




# Praktikum - Membuat Aplikasi LinearLayout

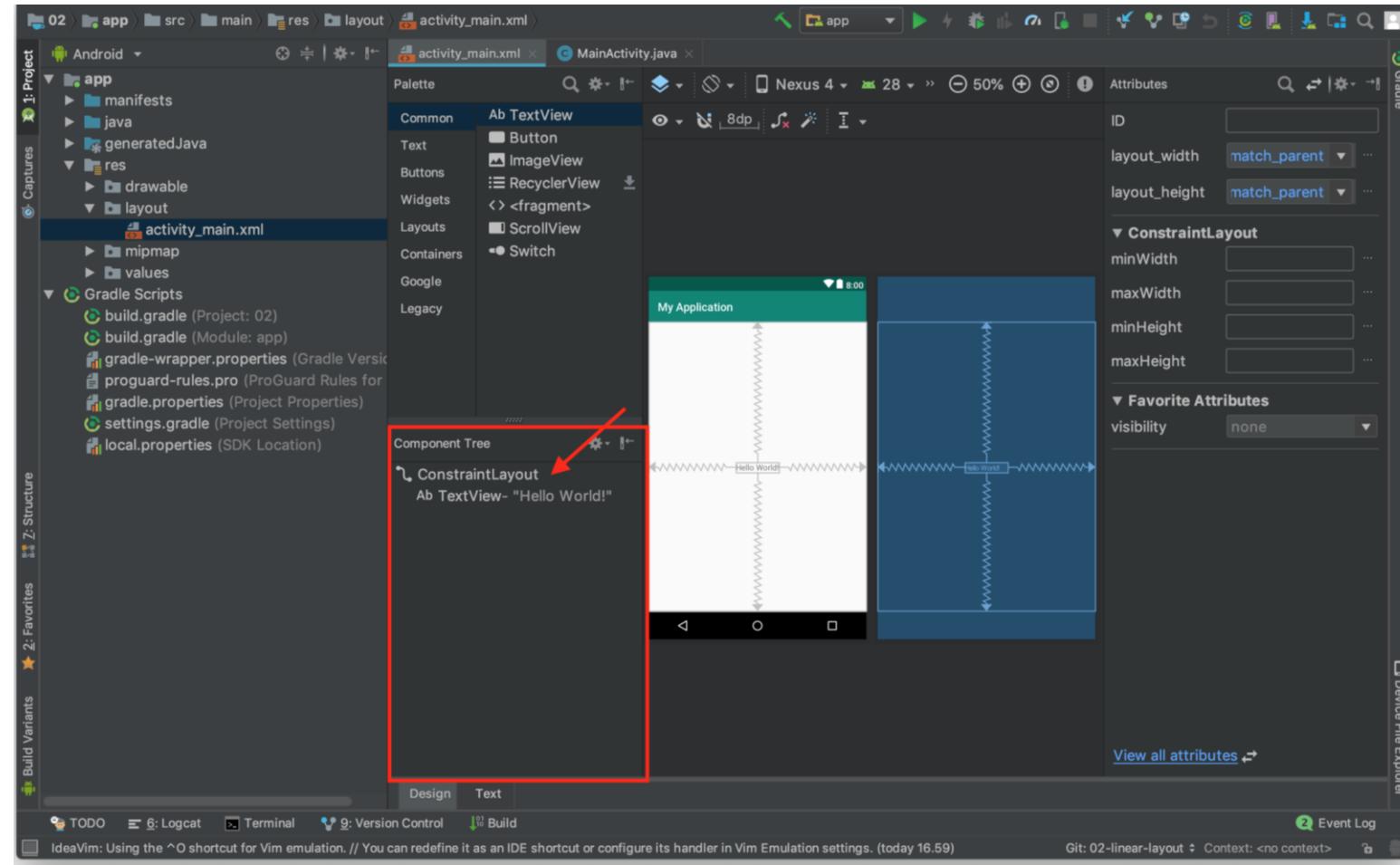


## LinearLayout



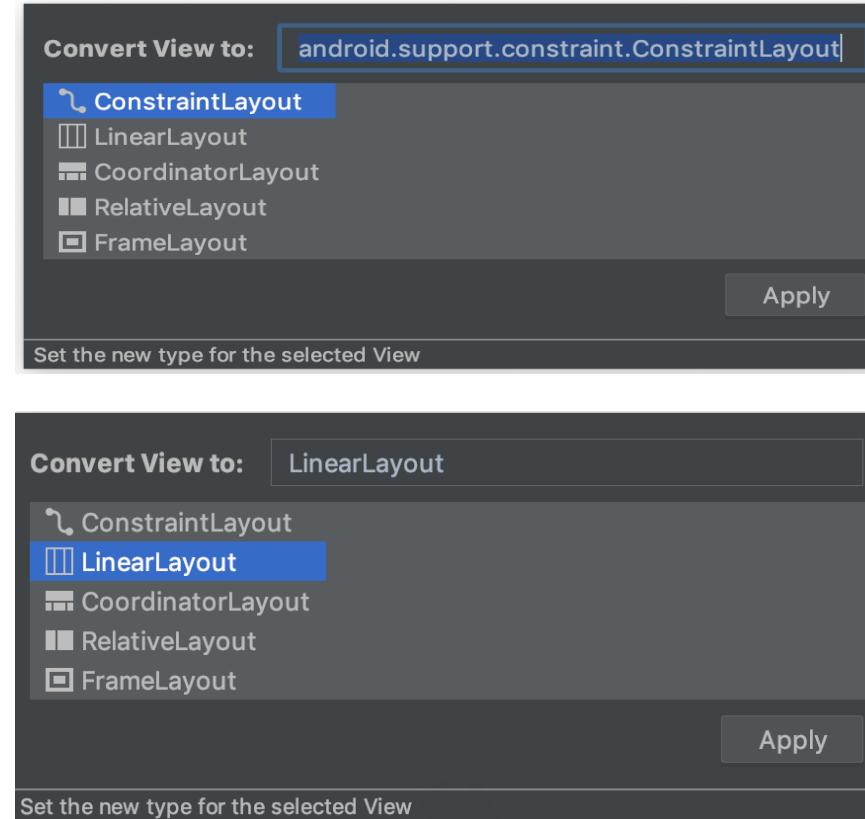
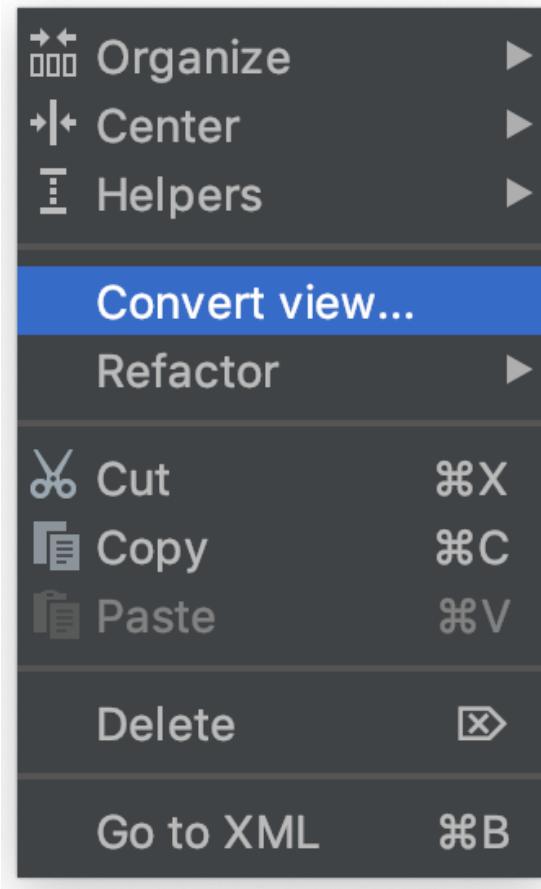
- Buka Android Studio
- Subject : Praktikum\_Nim\_NamaDepan
- Bukaalah project **LinearLayout** yang sudah disediakan.
- Bukaalah file **activity\_main.xml** kemudian buka dengan menggunakan design mode.
- Praktikum, di kumpulkan hari ini, Pukul 23.59 WIB

## LinearLayout



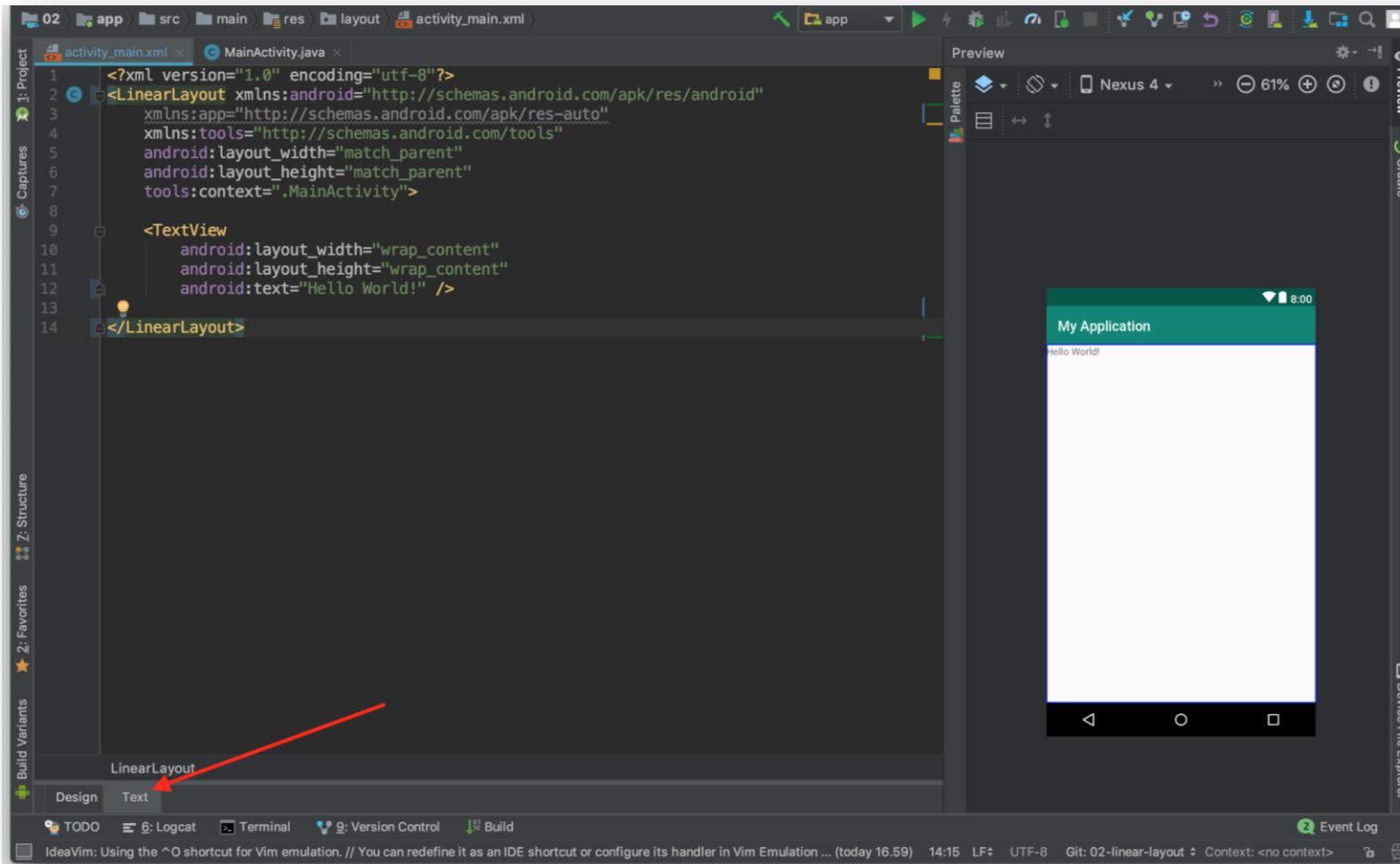
- Pada bagian jendela **Component Tree**, klik kanan pada **ConstraintLayout**.

## LinearLayout



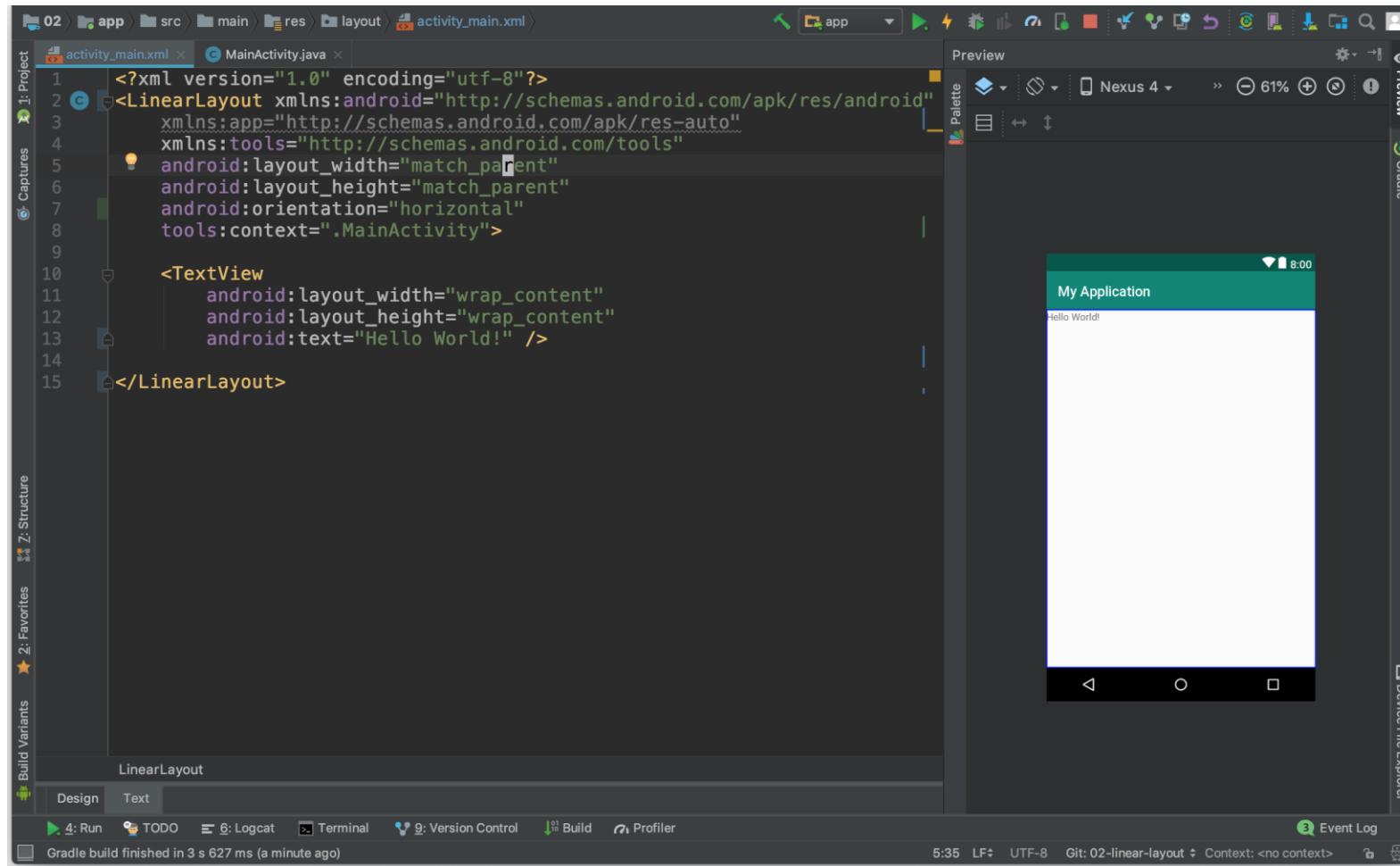
- Kemudian lakukan konversi view dari constraint layout ke linear layout.

## LinearLayout



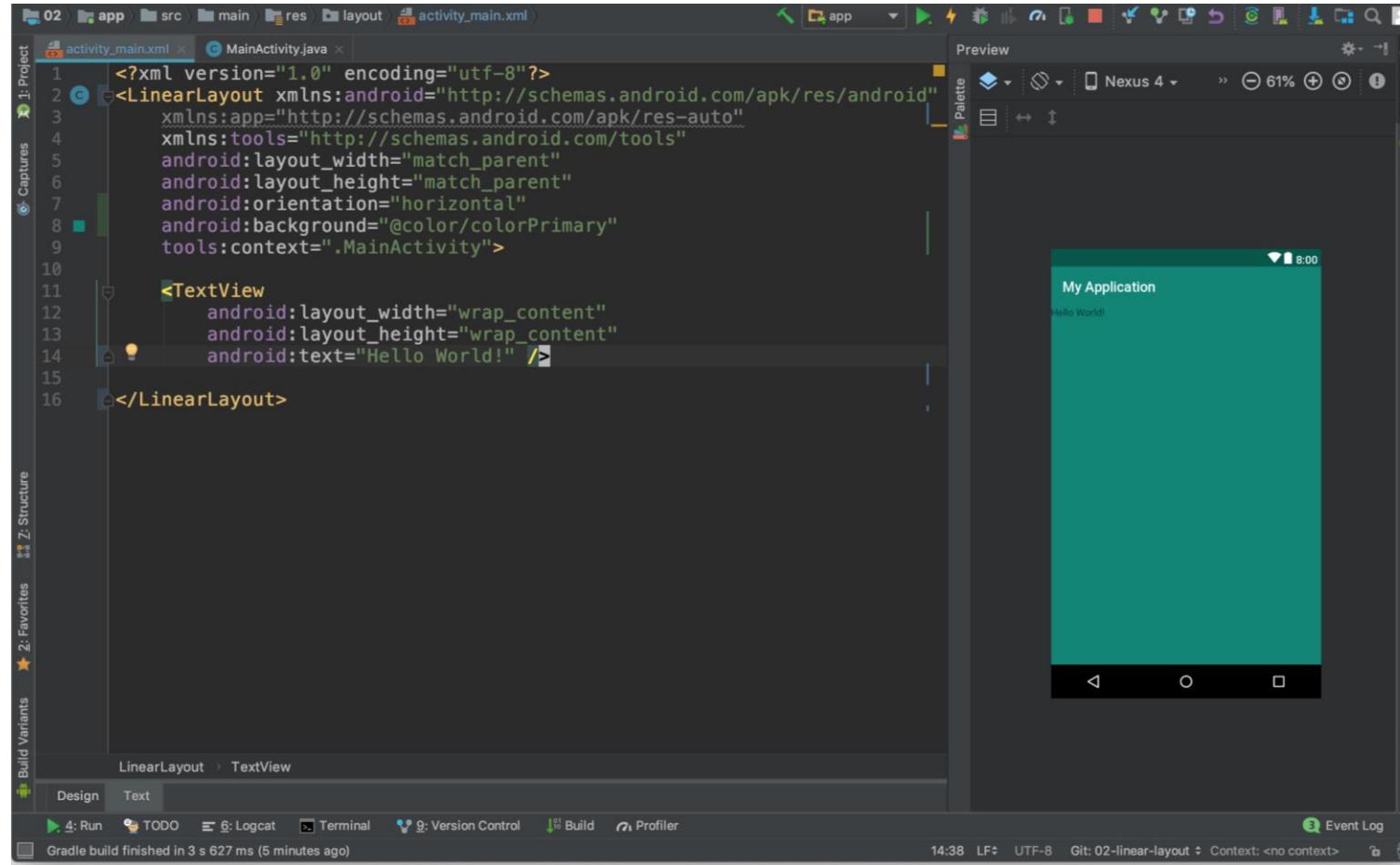
- Kemudian gantilah design mode ke text mode.

## LinearLayout



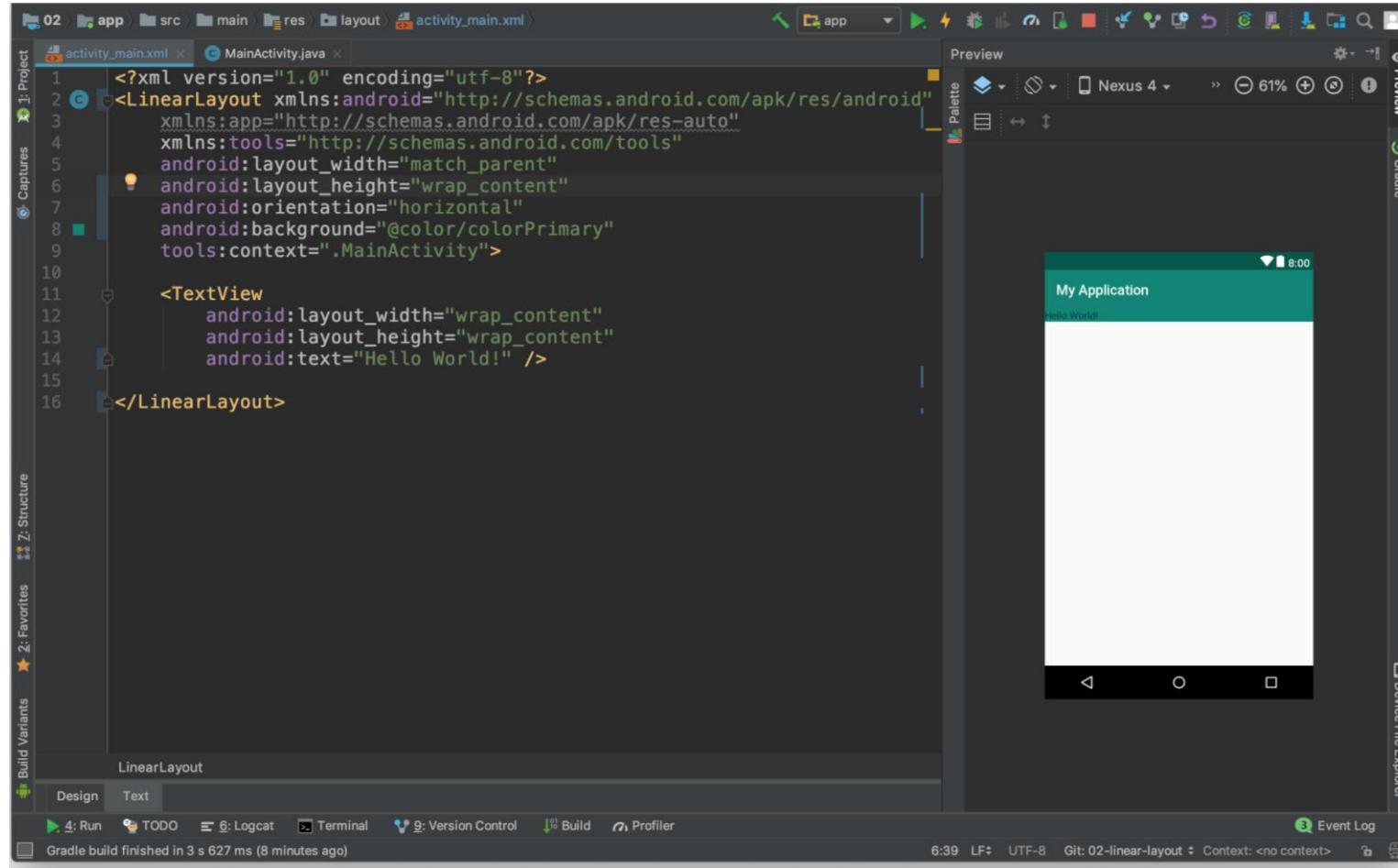
- Tambahkan property `android:orientation="vertical"` pada tag **LinearLayout**

## LinearLayout



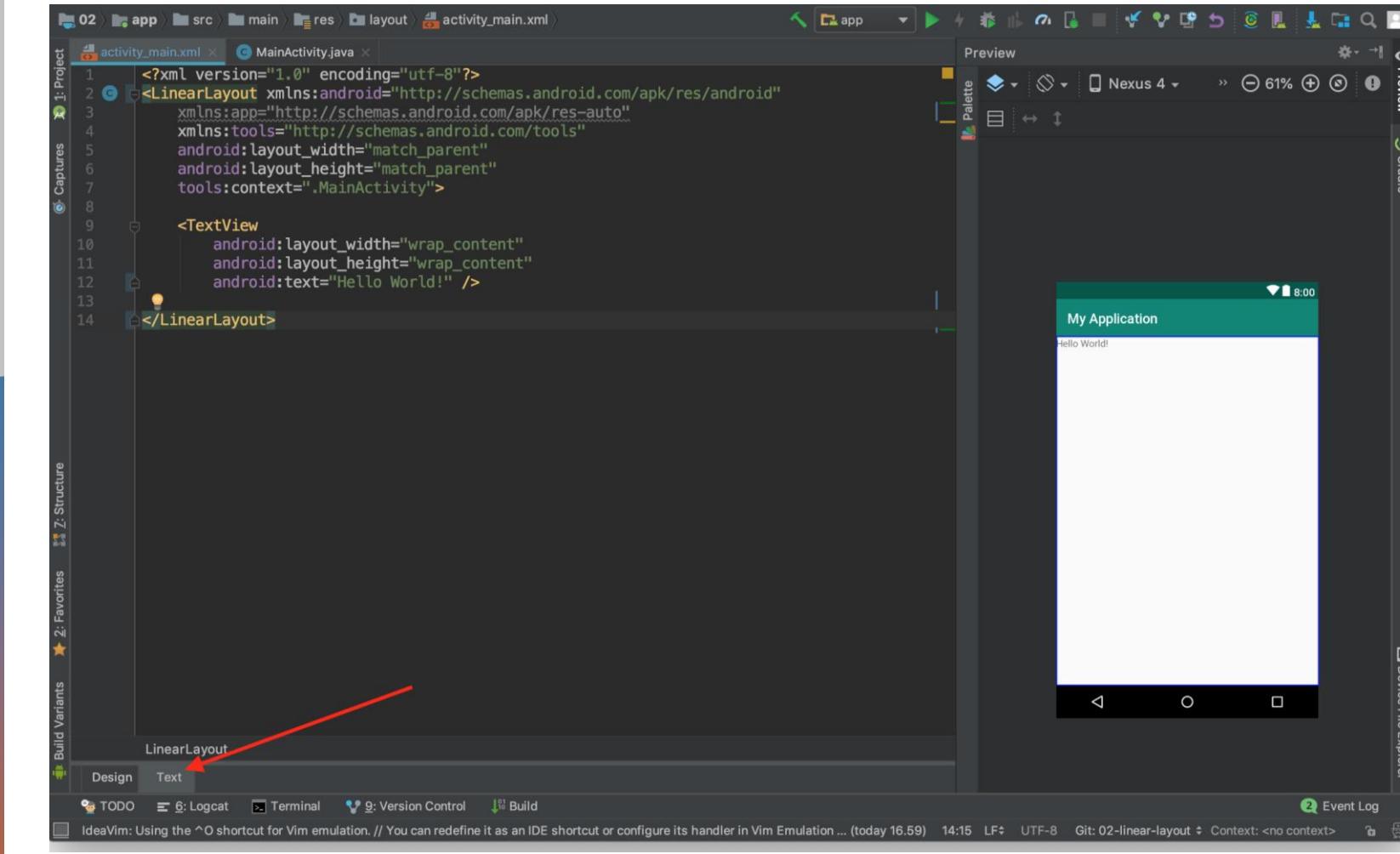
Selanjutnya tambahkan property **background** pada linear layout seperti pada gambar dibawah ini sehingga layout Anda berubah background nya menjadi hijau.

## LinearLayout



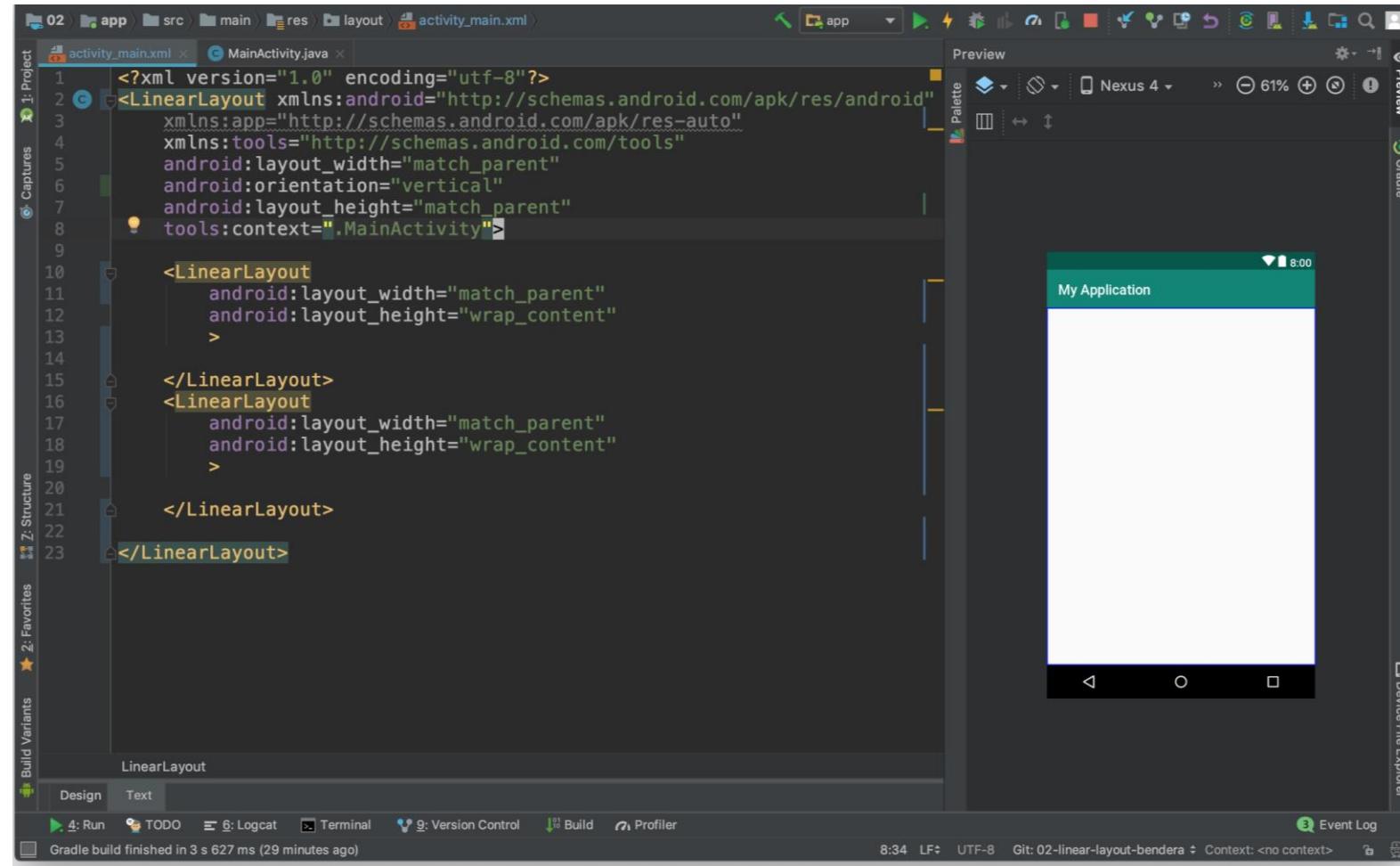
- Kemudian ubahlah nilai dari property `layout_height` menjadi `wrap_content`
- Lakukanlah kompilasi, kemudian simpulkan apakah perbedaan antara `wrap_content` dan `match_parent`

## LinearLayout



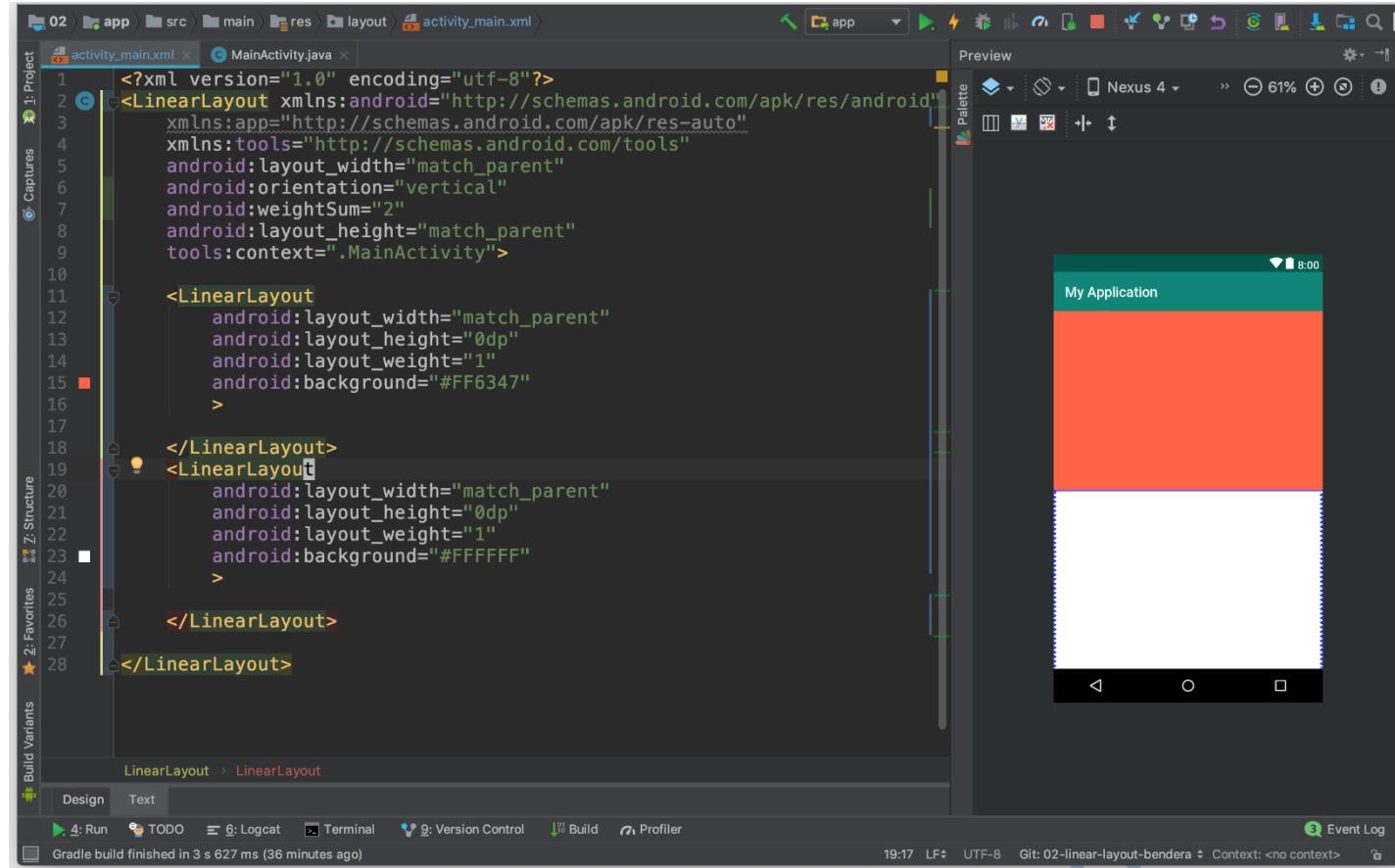
- Buka kembali project sebelumnya, kemudian kembalikan `layout_height` ke `match_parent`, hapus juga property `background` seperti pada gambar disamping.

## LinearLayout



- Hapuslah tag `TextView` kemudian isilah `LinearLayout` dengan dua buah child lain yang juga sebuah `LinearLayout`. Hasilnya akan seperti pada gambar disamping.

## LinearLayout



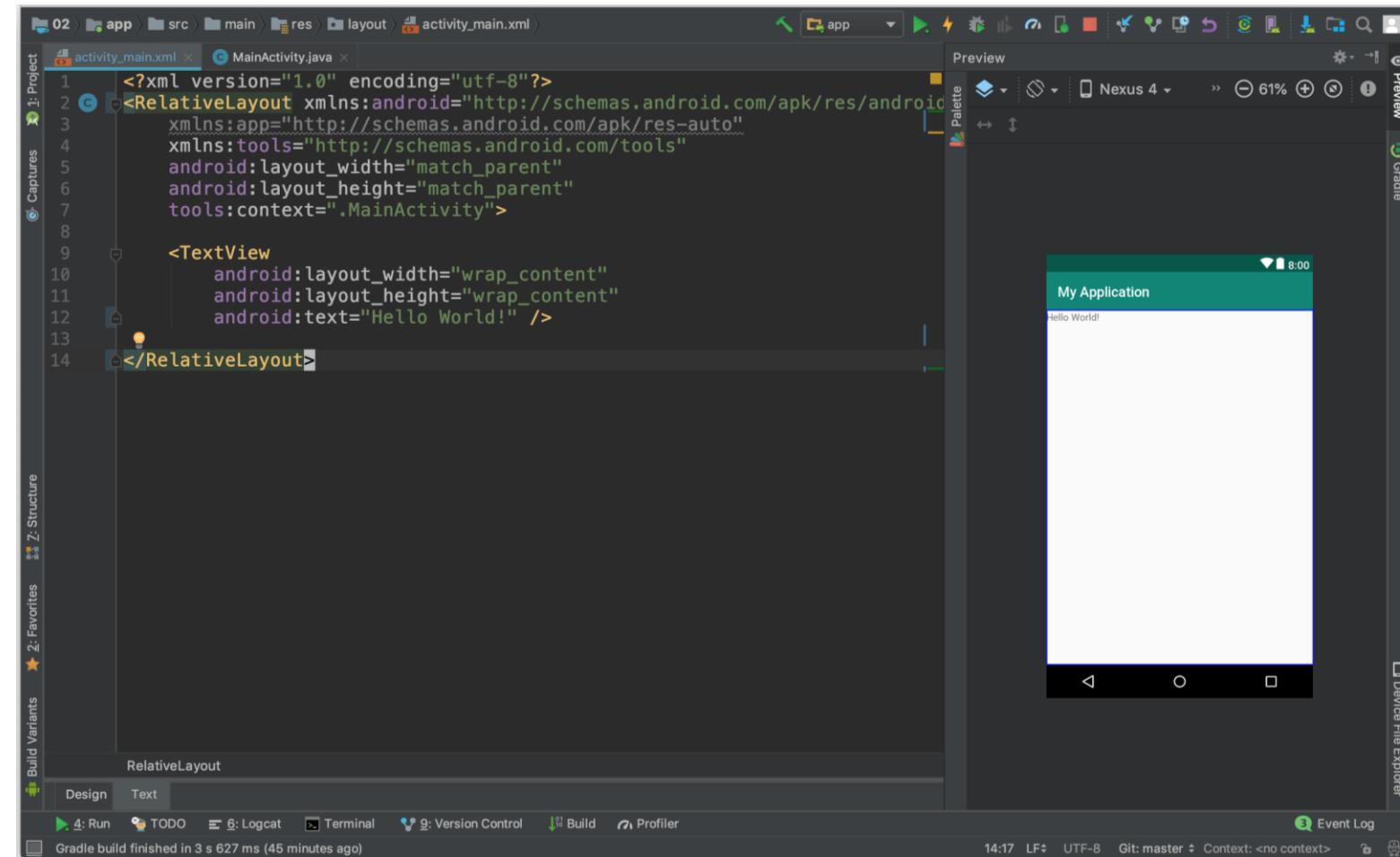
- Jika Anda perhatikan, belum terjadi perubahan apapun pada layout. Untuk itu lanjutkan dengan menambahkan `property background, weight_sum, dan layout_weight` seperti pada gambar disamping.
- Lalu Run/Debug dengan devices masing-masing



# Tugas - Membuat Aplikasi Relativelayout

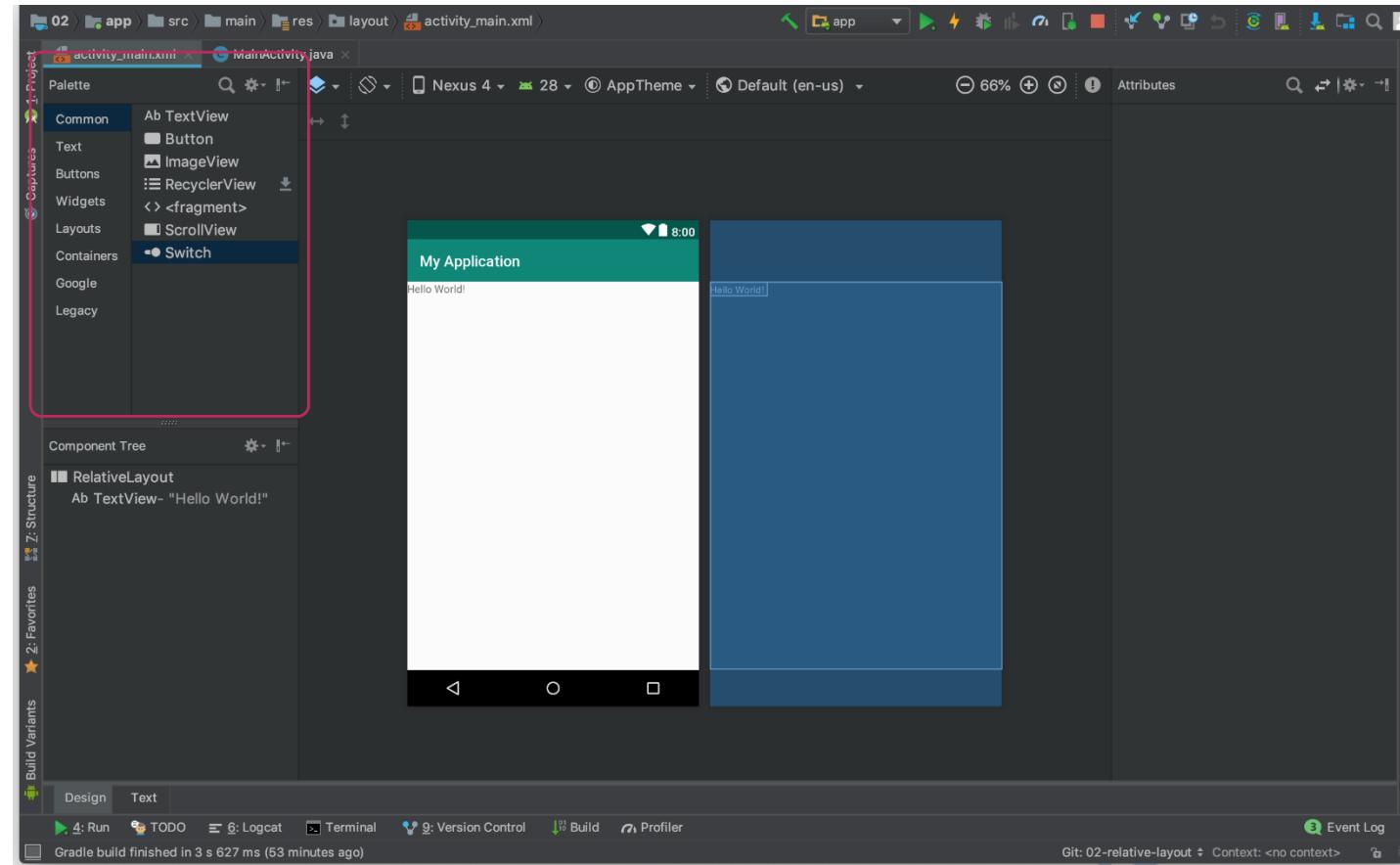


## Relativelayout



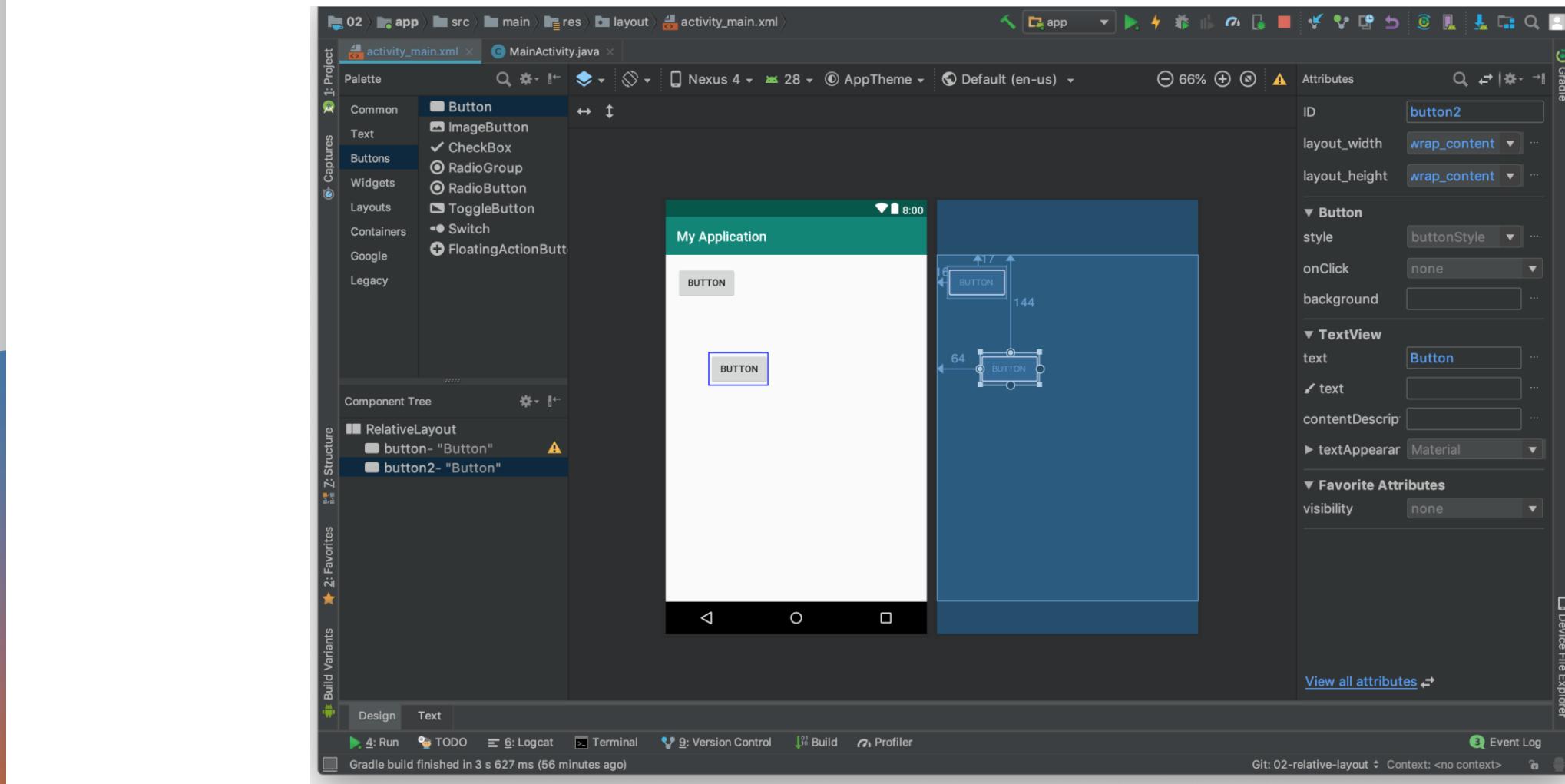
- Bukalah project **Tugas\_RelativeLayout\_Nim**
- kemudian ubah default layout pada `activity_main.xml` menggunakan layout **RelativeLayout**. Sehingga kode program dan tampilan berubah seperti gambar di samping

## Relativelayout

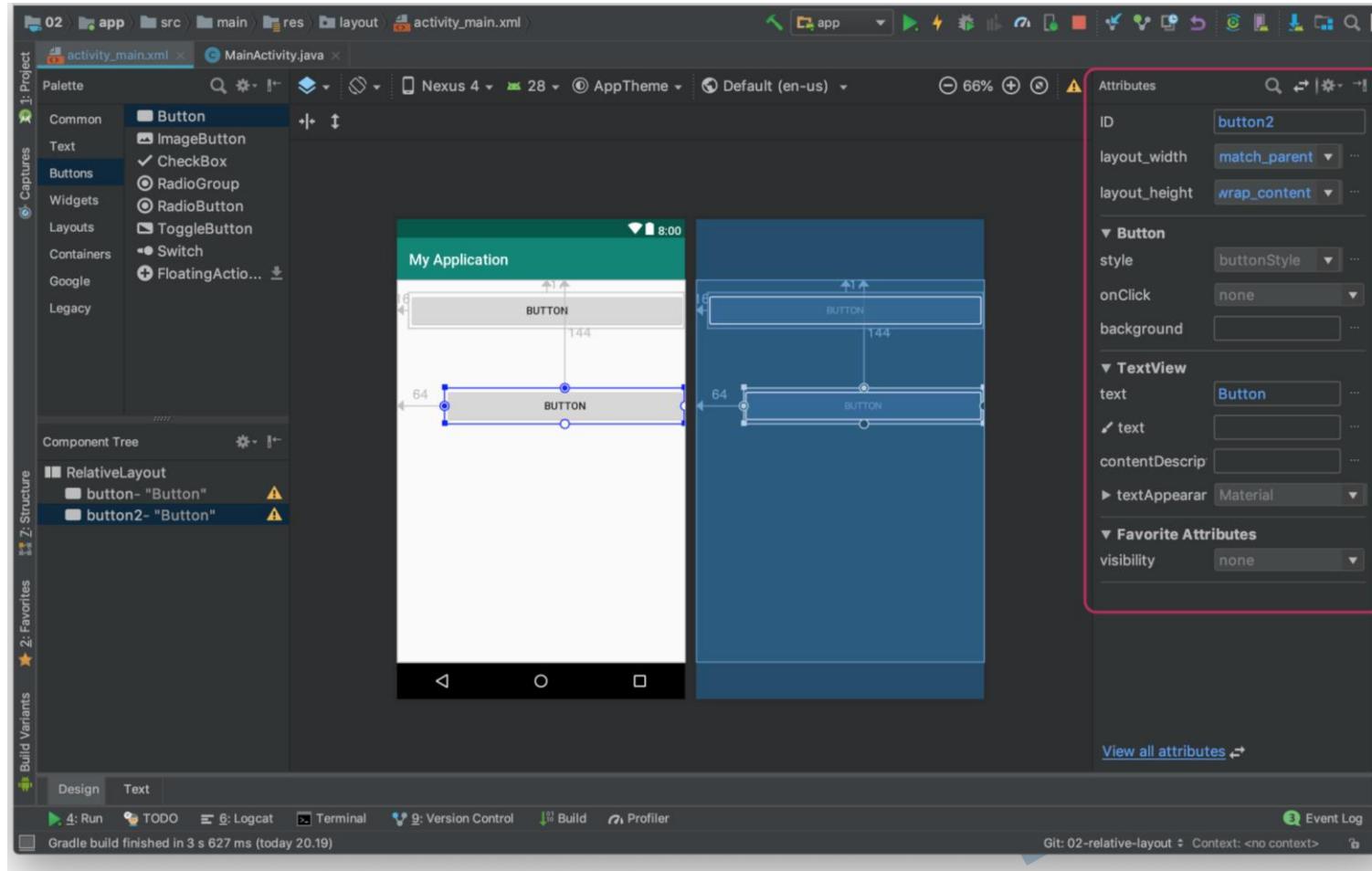


- Untuk mencobanya, tambahkan dua buah button pada layout. Anda dapat menambahkan button dengan menggunakan pallet yang ada di sebelah kiri tampilan design. Kemudian jangan lupa menghapus **TextView "hello world"**.

## Relativelayout

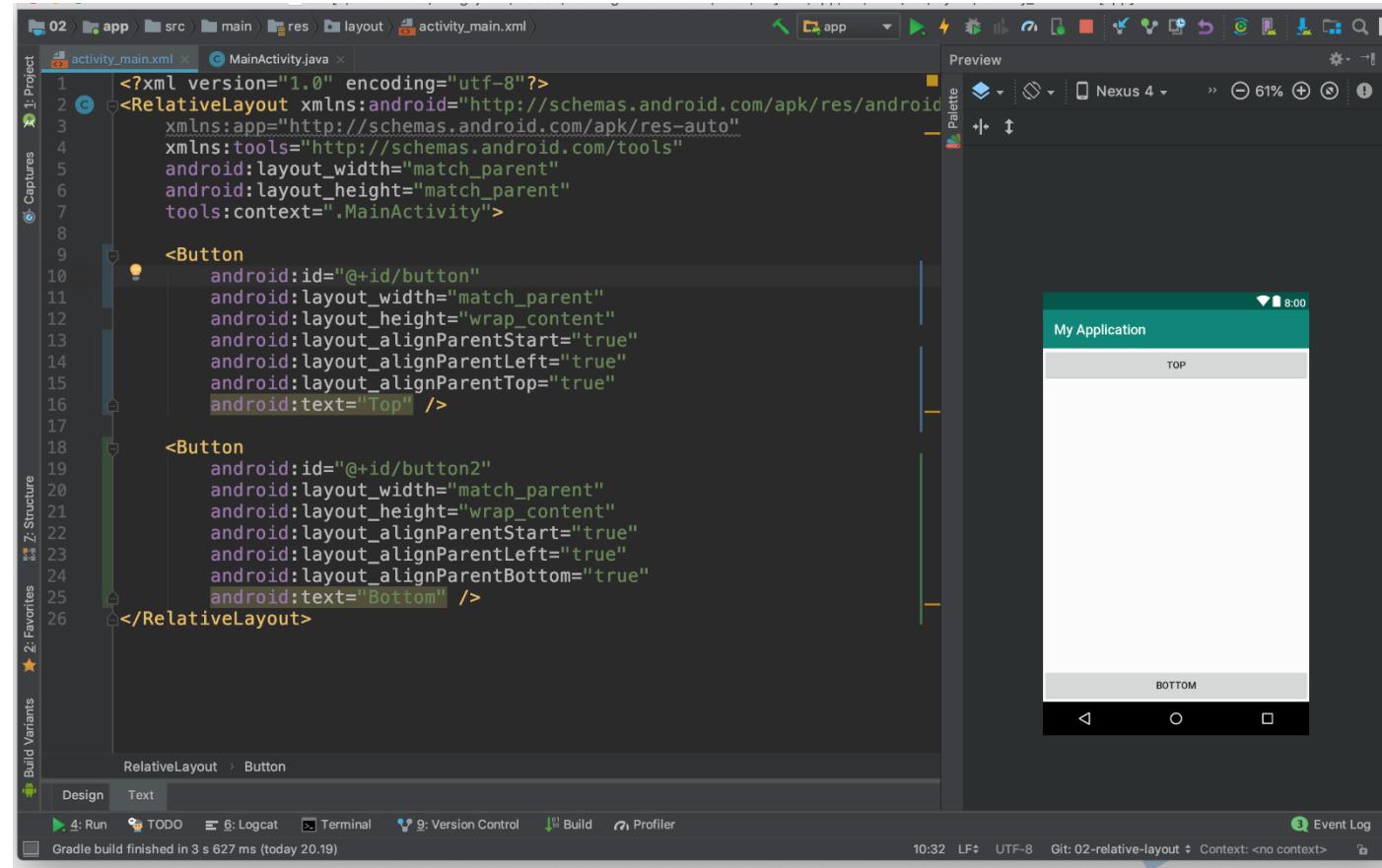


## Relativelayout



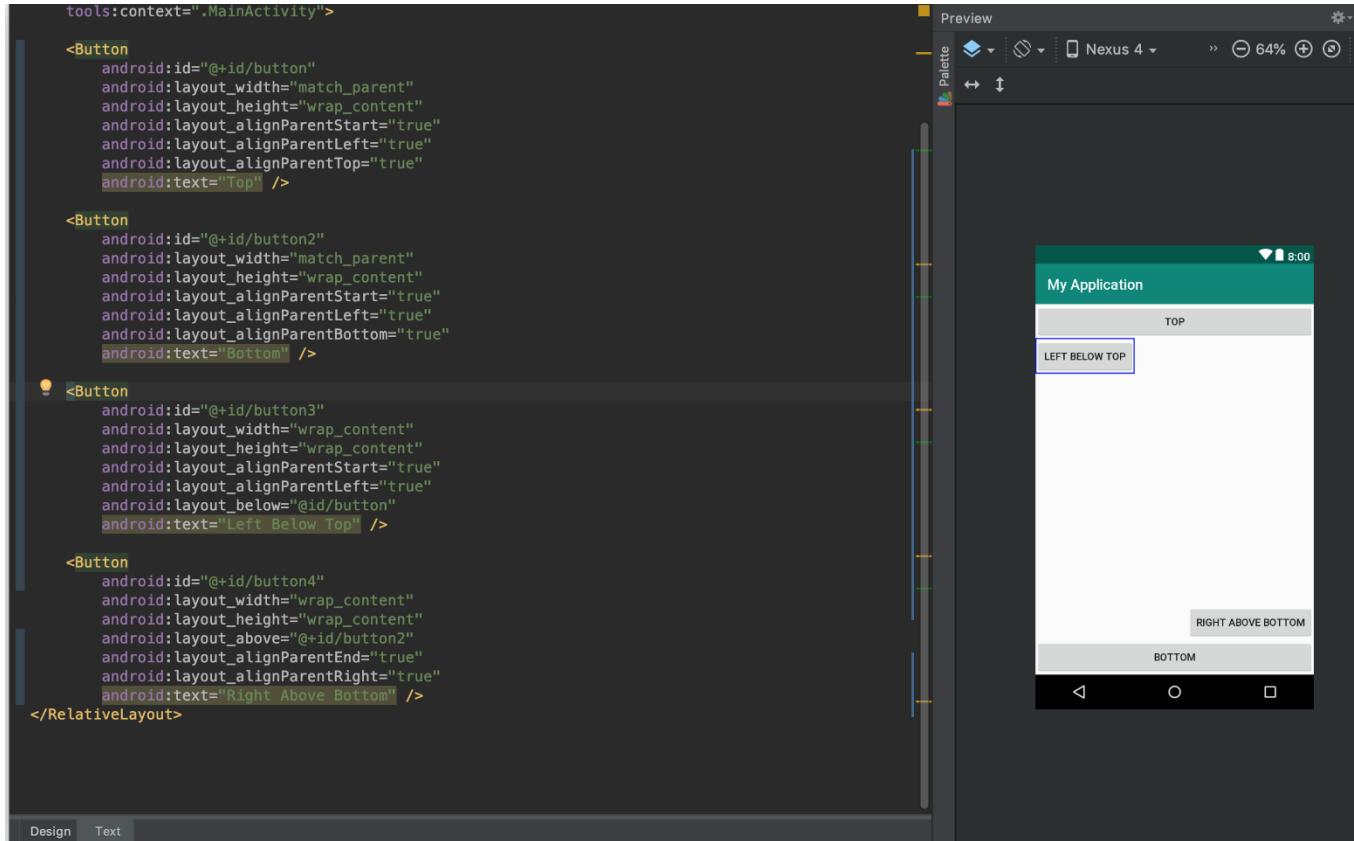
- Ubahlah teks dan lebar dari **button** sehingga menjadi seperti gambar di samping ini.

## Relativelayout



- Selanjutnya, ubahlah tampilan dengan mengubah kode program melalui **text mode** sehingga menjadi seperti gambar di samping ini.

## Relativelayout



- untuk lebih memahami relative layout buatlah layout baru seperti pada gambar disamping ini.
- Gradle/Run dengan devices masing-masing
- Screenrecord hasil dengan tools yang ada pada devices masing-masing, lalu send ke Group Telegram dengan Subjek : Nim\_Nama\_Relativelayout
- Dikumpulkan paling lambat tanggal 20 Oktober 2022

# Any Question ?

