

# MOBILE PROGRAMMING



Oleh :

CHALIFA CHAZAR



## MATERI 2

### MEMBUAT ANTARMUKA

#### Tujuan:

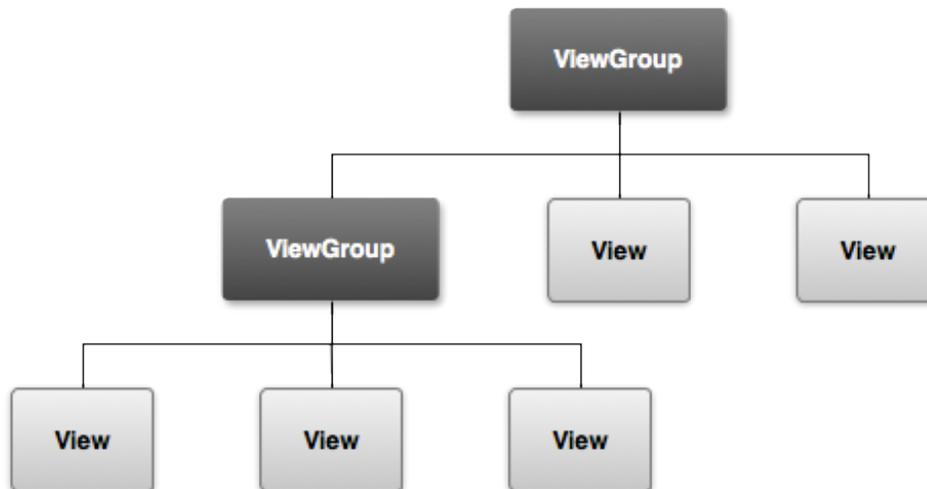
Mahasiswa memahami dan mampu menggunakan beberapa komponen antarmuka untuk mendukung suatu aplikasi Android.

#### Pustaka:

Abdul Kadir, 2013: Pemograman Aplikasi Android. Andi Yogyakarta.

#### 1. Mengetahui Komponen Penyusun Antarmuka

Hirarki *User Interface* (UI) pada aplikasi Android dapat digambarkan sebagai berikut.



Pada dasarnya untuk menyusun sebuah antarmuka (*user interface*) di aplikasi Android, dapat dikelompokkan berdasarkan prinsipnya menjadi tiga kelompok yaitu komponen dasar, komponen pemilih, dan komponen daftar.

- Komponen dasar, contohnya `TextView` untuk menyatakan label, `EditText` untuk menyatakan pemasukan data, dan `Button` untuk menyatakan tombol.
- Komponen pemilih, contohnya `TimePicker` untuk mengatur waktu dan `DatePicker` untuk mengatur tanggal.



- Komponen daftar, contohnya `ListView` dan `SpinnerView` untuk menampilkan sejumlah data.

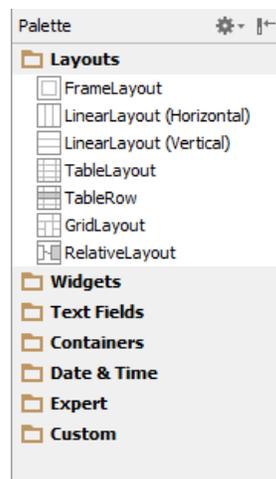
## **Catatan!**

**Linear Layout** adalah layout yang terstruktur. **Linear layout** akan meletakkan elemen yang ada di dalamnya secara berurutan tergantung *orientation*-nya. Jika *orientation* vertical, maka peletakan elemen akan terurut ke bawah. Sedangkan jika *orientation* horizontal, maka peletakan elemennya akan terurut ke samping. Secara default, *orientation* horizontal tidak ditulis di dalam XML.

## 2. Membuat Dua Baris Tulisan

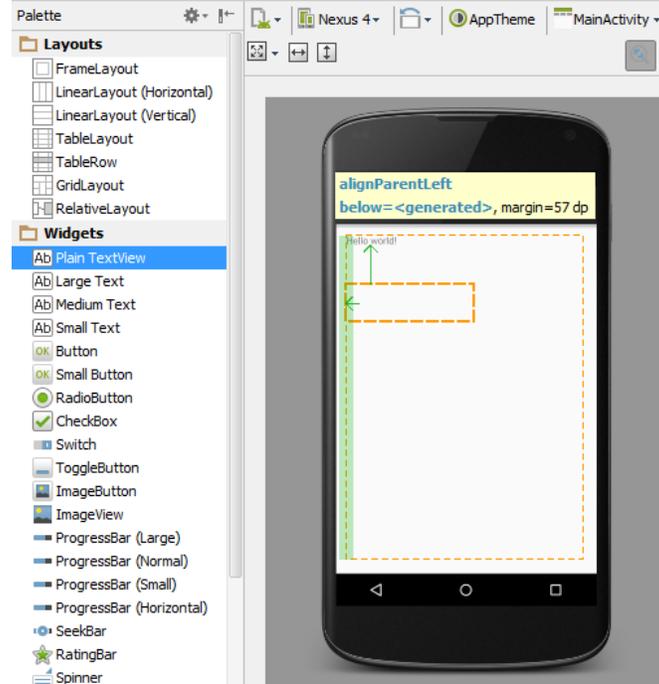
Pada materi sebelumnya, Anda diminta untuk membuat sebuah aplikasi yang menampilkan dua buah text. Baris pertama didapatkan secara default pada program yang menampilkan “Hello world!”.

Untuk menambahkan satu baris baru dapat menggunakan komponen yang terdapat di panel **Palette**. Berikut ini tampilan dari panel **Palette**.

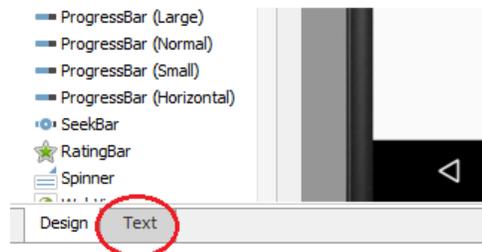


Berikut ini adalah langkah-langkah untuk membuat aplikasi yang menampilkan dua baris tulisan.

1. Buatlah sebuah **New Project** dan beri nama project **Latihan2**.
2. Klik panel palette **Widgets** terlebih dahulu untuk menampilkan isinya.
3. Pilih dan tempatkan **Plain TextView** pada tampilan layar Design.



4. Ketika Anda menambahkan sebuah **TextView** pada layar **Design**, maka secara otomatis pada layar **Text** juga akan ditambahkan **TextView** dalam bentuk kode xml. Perhatikan gambar dibawah ini.



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

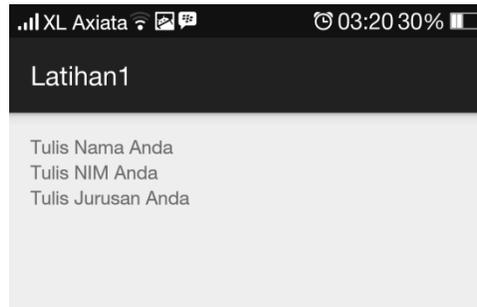
    <TextView android:text="Nama Saya Android" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20dp"
        android:id="@+id/textView"
        android:textColor="@color/abc_primary_text_material_light" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Nim Saya 23512127"
        android:id="@+id/textView1"
        android:layout_below="@+id/textView"
        android:layout_alignParentLeft="true"
        android:layout_marginTop="23dp"
        android:textColor="@color/abc_primary_text_disable_only_material_light"
        android:textSize="@dimen/Large" />

</RelativeLayout>
```



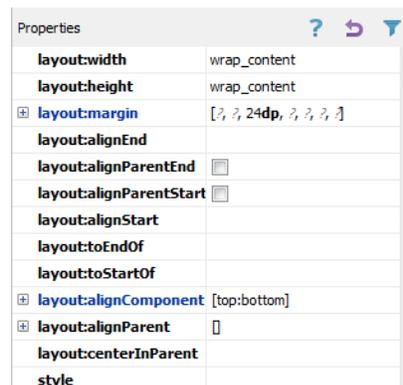
- Ubah isi tulisan menjadi NIM Anda. Untuk mengubah isi tulisan dapat dilakukan dengan menambahkan string baru seperti yang sudah dipelajari pada materi sebelumnya.



### 3. Menagtur Text

Mengatur jenis, ukuran, dan style sebuah teks dapat dilakukan dengan menggunakan panel **Properties**.

Setiap komponen memiliki sebuah panel **Properties** yang bisa diubah sesuai dengan pilihan yang tersedia. Atau dapat juga menambahkan sebuah pilihan baru dengan menggunakan sebuah script.



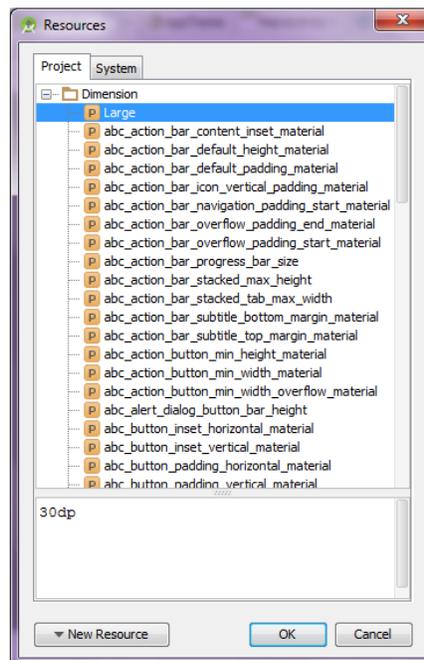
Berikut ini adalah langkah-langkah untuk merubah ukuran teks.

- Klik pada **TextView** pada layar **Design**.
- Klik **textSize** pada panel Properties di sebelah kanan layar Anda.

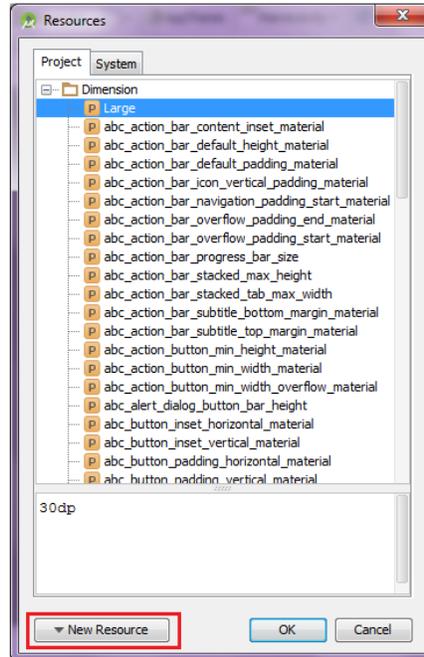


textColor	@color/accent_material_light
textColorHighlight	
textColorHint	
textColorLink	
textIsSelectable	<input type="checkbox"/>
textSize	@dimen/Large <span style="border: 1px solid red; border-radius: 50%; padding: 2px;">...</span>
<b>textStyle</b>	<input type="checkbox"/>
normal	<input type="checkbox"/>
bold	<input type="checkbox"/>
italic	<input type="checkbox"/>

3. Pada saat di-klik akan muncul sebuah dialog box yang menampilkan banyak pilihan resources. Anda dapat memilih **Resources** yang diinginkan.



4. Atau Anda dapat membuat sebuah **Resources** baru

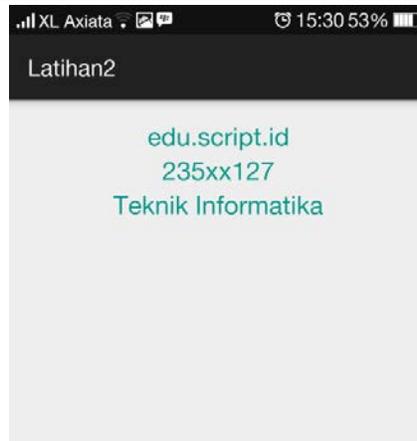


Beberapa properti yang digunakan di komponen melibatkan suatu satuan ukuran, misalnya:

- dp (density-independent pixel). 160dp = 1 inci.
- sp (scale-independent pixel). Dianjurkan untuk menentukan ukuran font.
- pt (point). 1pt = 1/72 inci.
- px (pixel). Menyatakan ukuran piksel yang sebenarnya. Penggunaannya tidak dianjurkan mengingat kemungkinan ukuran layar piranti device yang berbeda-beda.

### **Latihan 3**

Ubah jenis font, ukuran font, warna, dan style text (bold / italic) pada Latihan 2 (pada materi sebelumnya).



## 4. Menampilkan Gambar

Untuk menampilkan gambar pada aplikasi Android tidak jauh berbeda dengan membuat sebuah baris tulisan pada aplikasi. Perbedaannya hanya pada komponen dasar yang digunakannya.

**ImageView** adalah komponen dasar Android yang memiliki fungsi untuk menampilkan gambar. Gambar di komponen **ImageView** bisa diambil di folder **Drawable**.

Beberapa hal yang perlu diperhatikan dalam menggunakan komponen **ImageView** antara lain adalah sebagai berikut.

- **Penamaan**

Aturan penamaan file dalam **Drawable** mengikuti aturan penamaan variable java, yaitu tidak menggunakan spasi, simbol-simbol (-!#\$%\*&), huruf kapital, dst. Perlu diketahui juga bahwa tidak boleh ada dua buah file dengan nama yang sama dalam **Drawable** (walaupun berbeda ekstensi).

Kesalahan penamaan dapat menyebabkan error, karena folder di **Drawable** direferensikan hanya berdasarkan namanya saja tanpa menggunakan ekstensinya.

- **SubFolder**

Pada **Drawable** tidak memperbolehkan adanya subfolder.

- **Jenis File**

Pada umumnya file pada folder **Drawable** hanya berisi gambar atau XML.

- **Pemanggilan File**

Untuk pemanggilan file dalam **Drawable**, dapat dilakukan dengan mudah melalui XML.

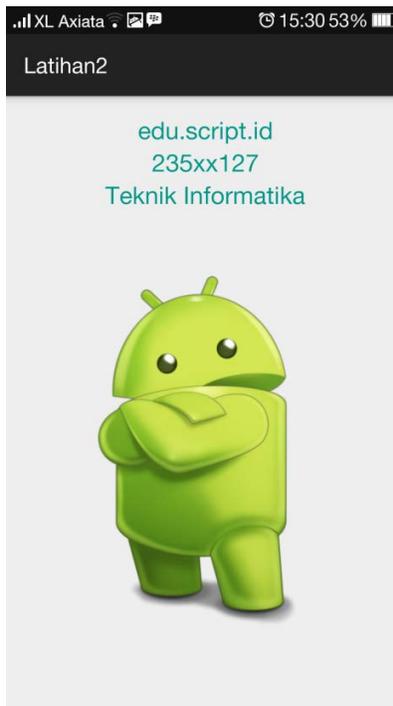
- **Keamanan**



File APK sebenarnya adalah sebuah file ZIP yang menggunakan ekstensi APK. Anda bisa mengekstrak APK dengan menggunakan program seperti 7zip, Winzip atau Winrar. File yang ada dalam **Drawable** sudah diubah menjadi file biner yang tidak bisa dibuka atau dibaca lagi.

### Latihan 4

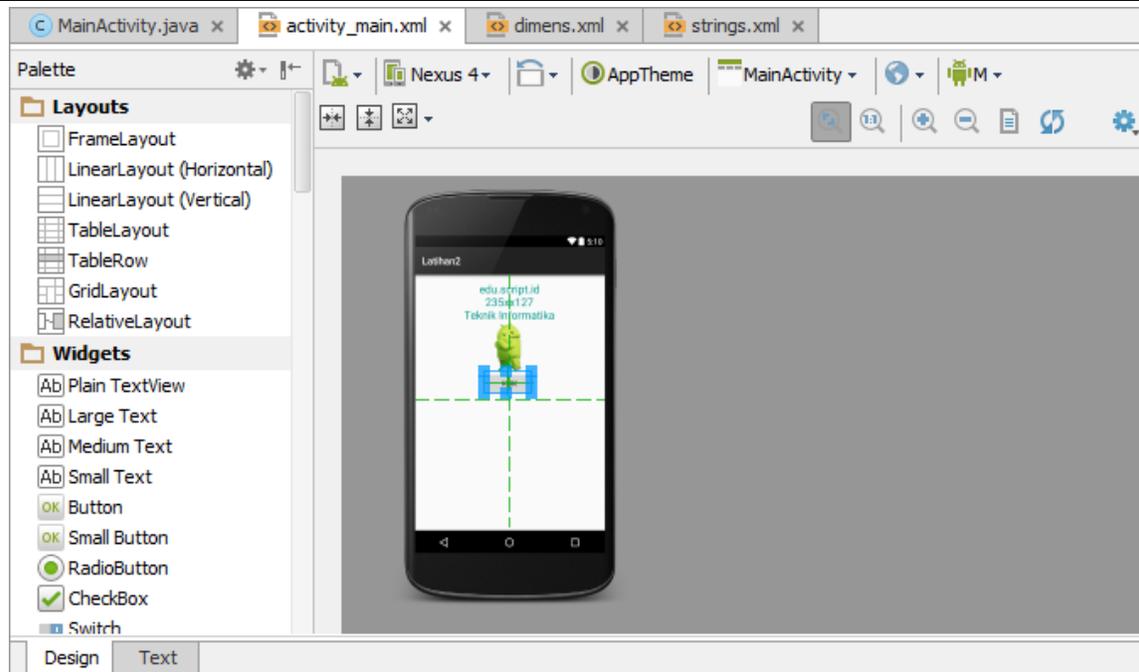
Coba tambahkan gambar pada aplikasi Android Anda sehingga terlihat seperti gambar dibawah ini.



### 5. Menambahkan Button

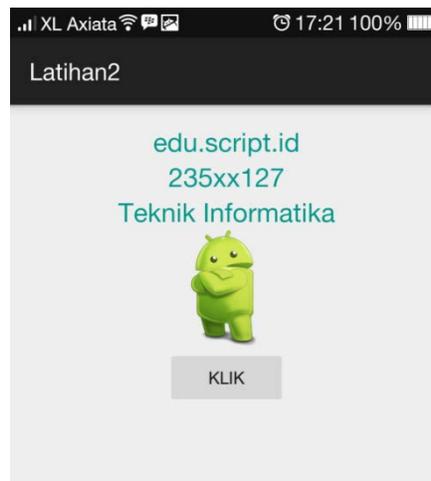
**Button** merupakan salah satu komponen yang sering digunakan sebagai antarmuka. Ketika **button** di-klik akan muncul suatu tindakan/perintah. Berikut ini langkah-langkah untuk menambahkan **button**.

1. Klik pallete **Widgets**
2. Klik **Button**, tarik dan tempatkan **Button** pada layar desain



3. Ganti isi property Text (melalui panel Properties) menjadi: Klik
4. Jalankan program

Berikut ini adalah hasil tampilan ketika dijalankan pada mobile device. Pada saat ini tombol “klik” masih belum berisi aksi tertentu.



### Catatan!

Untuk menampilkan **button** di posisi center dapat menggunakan kode berikut ini:

```
android:gravity="center | center_vertical | center_horizontal"
```