

Bab 9

Pemrograman Widget pada Android Studio



Mata Kuliah Mobile Programming

Dosen : Bambang Sugiarto, S.T, M.T

Program Studi S1 Teknik Informatika

Fakultas Teknik

Universitas Sangga Buana YPKP Bandung

*Dirangkum dari berbagai sumber referensi
(hanya untuk penggunaan internal/tidak untuk dipublikasikan)*

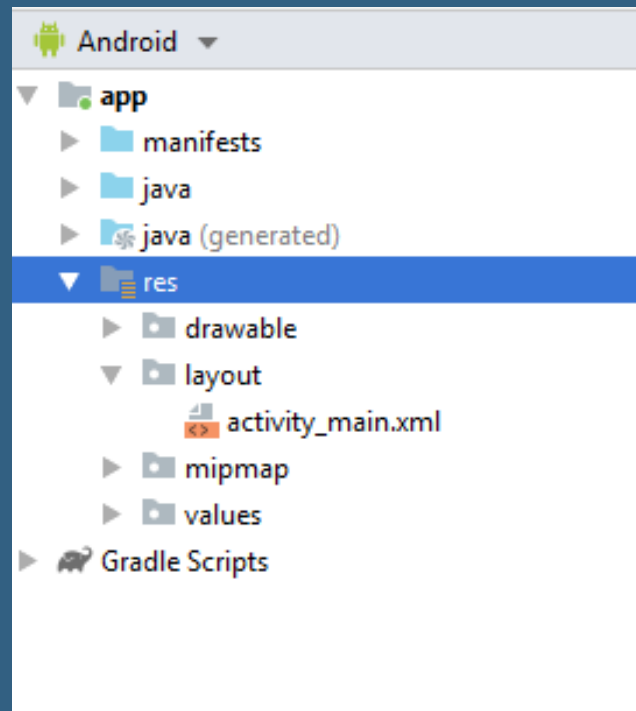
Widget pada Android Studio

- **Widget adalah** komponen dalam bahasa pemrograman Android yang dapat digunakan untuk menampilkan user interface dan menerima interaksi terhadap user serta memudahkan user menggunakan aplikasi.
- Widget mempunyai properti pasti yaitu lebar dan tinggi.
- Setiap widget mempunyai action sendiri-sendiri, sesuai dengan fungsinya.
 - Contoh dari widget yaitu :
 - Button
 - EditText,
 - TextView
 - Dan lain-lain.

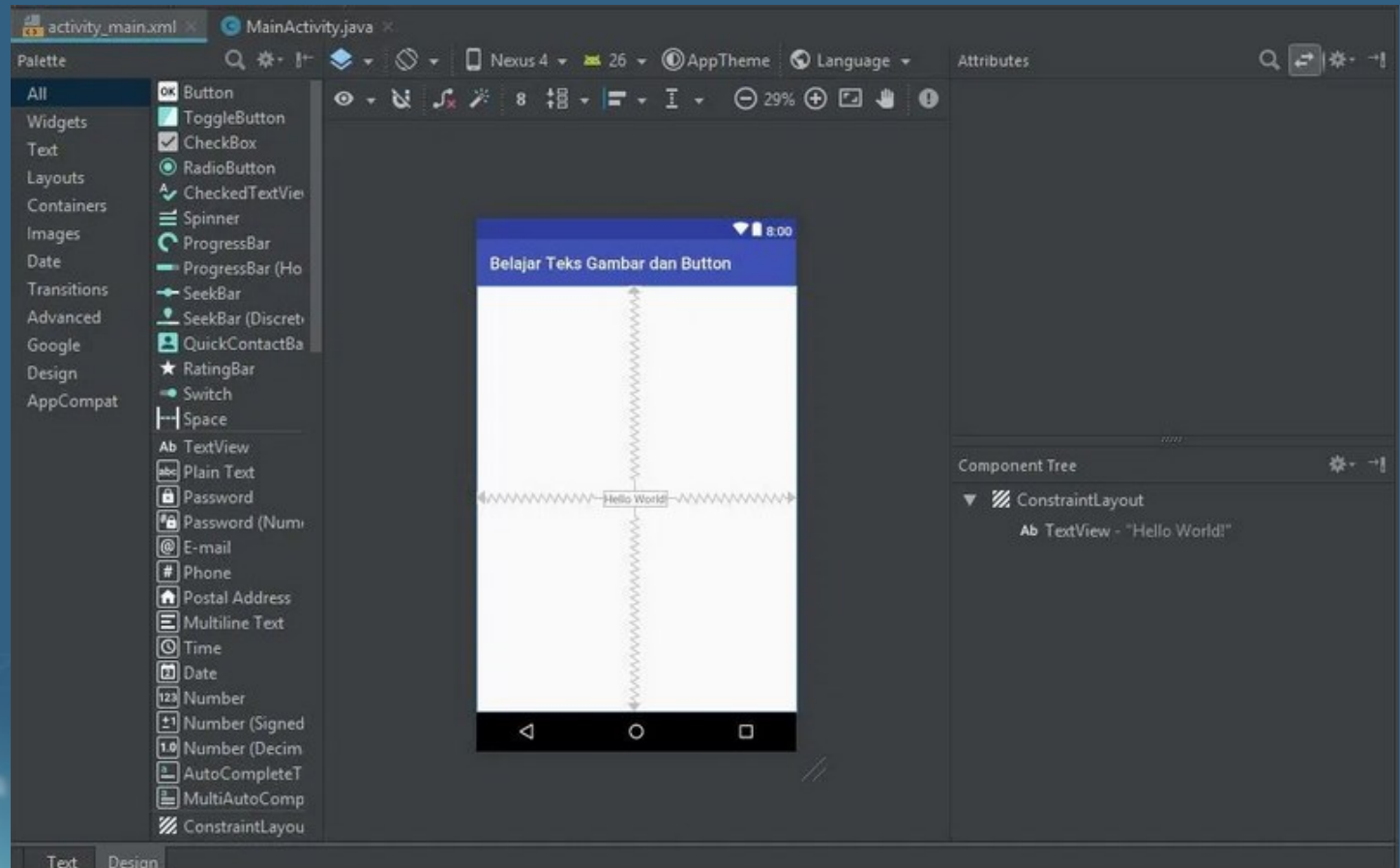


Bekerja dengan Widget

- Untuk latihan membuat Widget, buatlah satu proyek baru seperti pada pertemuan sebelumnya. dengan nama LatihanWidget
- Buka file main.xml yang ada pada folder res/layout.



- Pada jendela bagian kanan akan tampil layout beserta daftar komponen yang dapat digunakan untuk membangun sebuah aplikasi Android



- Kita bisa *drag-and-drop* komponen/widget pada bagian palette ke jendela layout.
- Misal, widget button, radio button , dll kemudian buka pada file main.xml, dan ubah nilai yang ada pada text sesuai dengan keinginan
- Kemudian setelah itu jalankan aplikasi.



Button

- Pada bagian ini, akan dibahas empat jenis *button* yang sering digunakan dalam mengembangkan aplikasi di Android.
- *Button* tersebut antara lain *Basic Button*, *Image Button*, *Toggle Button* dan *Custom Button*.

➤ Basic Button

- ✓ *Button* yang paling dasar di Android adalah `android.widget.Button`.
- ✓ *Class* ini hanya dapat menangani *event* ketika *button* di tekan dengan menggunakan `OnClickListener`.
- ✓ Berikut ini adalah potongan kode definisi pembuatan *button* dalam bentuk xml :



```
<Button  
    android:id="@+id/ccbtn1"  
    android:text="Basic Button"  
    android:typeface="serif"  
    android:textStyle="bold"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"/>
```

- ✓ Berikut ini adalah potongan kode pemanggilan *button* menggunakan code Java:

```
Button btn =  
    (Button) this.findViewById(R.id.ccbtn1);  
    btn.setOnClickListener(new  
        OnClickListener() {  
            public void onClick(View v) {  
                finish();  
            }  
        }  
    );
```



- ✓ Kode diatas menjelaskan tentang bagaimana *event* ketika *button* ditekan.
- ✓ *Event* tersebut didaftarkan pada method `setOnClickListener()` dari `ClassOnClickListener`.
- ✓ Didalam kode, *listener* dibuat untuk menangani *event* ketika *button* ditekan.
- ✓ Ketika *button* ditekan *method* `OnClick()` otomatis langsung dieksekusi.

➤ Image Button

- ✓ Android mendukung *image button* melalui `android.widget.ImageButton`.
- ✓ Penggunaan *image button* hampir sama dengan *basic button*.



- ✓ Berikut adalah potongan kode menggunakan layout xml :

```
<ImageButtonandroid:id="@+id/imageBtn"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

- ✓ *Image* dari *button* juga dapat dibuat secara dinamis melalui *method* `setImageResource()` atau dengan memodifikasi *file xml layout* dengan menambahkan `android: src` yang merujuk pada *ID image* seperti yang terlihat pada potongan kode berikut :

```
<ImageButtonandroid:id="@+id/imageBtn"  
    android:src="@drawable/btnImage"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```



- ✓ Berikut ini adalah potongan kode pemanggilan *image button* menggunakan kode Java:

```
ImageButton btn =  
(ImageButton) this.findViewById(R.id.imageBtn  
);  
btn.setImageResource(R.drawable.icon);
```

➤ Toggle Button

- ✓ *ToggleButton* ini seperti *check box* atau *radio button*. *Button* ini mengarah pada *On* dan *Off*.
- ✓ *ToggleButton* akan menampilkan *green-bar* pada keadaan *On* dan *grayed-outbar* pada keadaan *Off*.
- ✓ Selain itu juga, *button* akan bertuliskan *On* pada saat keadaan *On* dan bertuliskan *Off* pada saat keadaan *Off*.



```
<ToggleButtonandroid:id="@+id/cctglBtn"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Toggle Button"/>
```

- ✓ Selain itu, tulisan *On-Off* tersebut dapat diubah dengan `android:textOn` dan `android:textOff` seperti yang terlihat pada potongan kode berikut :

```
<ToggleButtonandroid:id="@+id/cctglBtn"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:textOn="Run"  
android:textOff="Stop"  
android:text="Toggle Button"/>
```



➤ Custom Button

- ✓ Pada bagian ini, akan dijelaskan bagaimana membuat button dengan menggunakan *custom image*, bukan menggunakan teks.
- ✓ Pembuatan *custom button* menggunakan *Buttonwidget* dan sebuah *file XML* yang mendefinisikan tiga *image* yang berbeda untuk digunakan pada *button* yang berbeda.
- ✓ Berikut dapat dilihat potongan kode pembuatan Custom Button pada *file xml* :

```
<?xmlversion="1.0"encoding="utf-8"?>
```

```
<selectorxmlns:android="http://schemas.android  
.com/apk/res/android">
```

```
<itemandroid:drawable="@drawable/android_press  
ed"
```

```
android:state_pressed="true" />
<itemandroid:drawable="@drawable/android_focused"
android:state_focused="true" />
<itemandroid:drawable="@drawable/android_normal" />
</selector>
```

- ✓ *Source code* di atas mendefinisikan sebuah *single drawable resource* yang berada pada *drawable*
- ✓ Dalam kasus ini nama *file*-nya adalah *android_button.xml* yang akan mengganti *image* berdasarkan *status* yang berlaku pada saat *button* ditekan.
- ✓ `<item>` pertama definisikan *android_pressed.png* sebagai *image* pada saat *button* ditekan (diaktifkan)



- ✓ `<item>` kedua mendefinisikan *android_focused.png* sebagai *image* pada saat button di-*highlight*
- ✓ `<item>` yang ketiga mendefinisasikan *android_normal.png* sebagai *image* pada saat tidak sedang aktif.
- ✓ *File xml* ini merepresentasikan sebuah *single drawable resource* dan pada saat *button* ditekan, *image* akan ditampilkan berdasarkan *state* yang telah diatur pada konfigurasi di atas.
- ✓ Setelah selesai mendefinisikan *selector* di atas, selanjutnya *selector* tersebut akan di panggil dalam layout *file XML* yang akan ditampilkan :



```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:padding="10dp"  
    android:background="@drawable/android_button"  
>
```

- ✓ `android:background` menspesifikasikan *drawableresource* mana yang akan digunakan sebagai *background* (dimana pada saat disimpan dalam `res/drawable/android.xml`, direferensikan sebagai `@drawable/android`). Ini menggantikan *normal backgroundimage* yang digunakan sebagai *button* pada *system* ini.

