

## **MODUL 4**

### **Komponen Widget View**



#### **CAPAIAN PEMBELAJARAN**

---

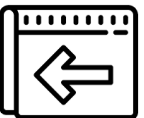
Mahasiswa mampu menggunakan Widget View (masukan) untuk membuat aplikasi sederhana



#### **KEBUTUHAN ALAT/BAHAN/SOFTWARE**

---

1. Android Studio 3.4.
2. Handphone Android versi 7.0 (Nougat)
3. Kabel data USB.
4. Driver ADB.



#### **DASAR TEORI**

---

Paket widget pada dasarnya merupakan visualisasi dari elemen user interface (UI) yang digunakan pada layar aplikasi Android di mana kita dapat merancang sendiri sesuai kebutuhan.

Widget di dalam Android ditampilkan dengan konsep *View*. Di mana aplikasi Android pada umumnya menggunakan widget sebagai Layout XML. Untuk mengimplementasikan widget, selain file kotlin kita juga membutuhkan tambahan dua file. Berikut ini adalah file-file yang umumnya kita butuhkan apabila kita membuat widget:

1. File Kotlin. Berupa file yang mengimplementasikan aksi dari widget. Jika kita mendefinisikan suatu widget beserta posisinya di layar yang didefinisikan dari file XML, kita harus melakukan coding di file kotlin yang dapat mengambil semua nilai atribut dari file layout XML yang didefinisikan.

2. File XML. Sebuah file yang mendefinisikan komponen elemen-elemen XML yang digunakan untuk inisialisasi widget serta atribut yang mendukungnya.
3. Layout XML. File XML menggambarkan atau penambahan keterangan pada layout widget kita.

Komponen widget *TextView* dan *Button* sudah kita bahas pada modul sebelumnya. Beberapa komponen widget akan kita bahas saat ini. *Widget EditText* untuk menuliskan teks ke aplikasi dan akan ditangkap oleh aplikasi untuk diolah. *Widget Image Button* untuk membuat button yang diberi gambar. *Widget Image View* untuk membuat tampilan gambar. Sedangkan widget *RadioButton* / *RadioGroup* biasanya digunakan bersama-sama.

Di dalam satu RadioGroup terdapat beberapa RadioButton. Dan di dalam satu RadioGroup user hanya dapat melakukan satu check/pemilihan RadioButton. Dan yang terakhir widget akan kita bahas CheckBox, pilihan yang dapat dipilih lebih dari satu item.

#### **Event Handling.**

Android dapat menangani **event** dari interaksi dengan pengguna. Saat mempertimbangkan event dalam user interface, pendekatannya adalah menangkap event dari objek **View** tertentu yang digunakan pengguna untuk berinteraksi. Kelas View menyediakan sarana untuk melakukannya.

Dalam berbagai kelas View yang akan digunakan untuk menyusun layout, mungkin dapat dilihat beberapa method callback publik yang tampak berguna untuk kejadian UI. Method ini dipanggil oleh framework Android ketika masing-masing tindakan terjadi pada objek itu. Misalnya, jika View (seperti Button) disentuh, method onTouchEvent() akan dipanggil pada objek itu. Kelas View salah satunya berisi sekumpulan interface bertumpuk dengan callback yang mudah didefinisikan. Antarmuka ini, yang disebut event listener, digunakan untuk melakukan interaksi pengguna dengan UI.

#### **Event listener**

Event listener merupakan antarmuka di kelas View yang berisi method callback tunggal. Method ini akan dipanggil oleh framework Android jika View yang telah didaftarkan dengan listener dipicu oleh interaksi pengguna dengan item dalam UI.

Yang juga disertakan dalam antarmuka event listener adalah method callback berikut ini:

1. Method onClick() dari View.OnClickListener. Ini dipanggil baik saat pengguna menyentuh item (jika dalam mode sentuh), maupun memfokuskan pada item dengan tombol navigasi atau trackball dan menekan tombol "enter" yang sesuai atau menekan trackball.
2. Method onLongClick() dari View.OnLongClickListener. Ini dipanggil baik saat pengguna menyentuh dan menahan item (jika dalam mode sentuh), maupun memfokuskan pada item dengan tombol navigasi atau trackball dan menekan serta menahan tombol "enter" yang sesuai atau menekan dan menahan trackball (selama satu detik).
3. Method onFocusChange() dari View.OnFocusChangeListener. Ini dipanggil saat pengguna menyusuri ke atau dari item, dengan menggunakan tombol navigasi atau trackball.
4. Method onKeyDown() dari View.OnKeyListener. Ini dipanggil saat pengguna memfokuskan pada item dan menekan atau melepas tombol perangkat keras pada perangkat.
5. Method onTouch() dari View.OnTouchListener. Ini dipanggil saat pengguna melakukan tindakan yang digolongkan sebagai peristiwa sentuh, termasuk penekanan, pelepasan, atau isyarat perpindahan pada layar (dalam batasan item itu).
6. Method onCreateContextMenu() dari View.OnCreateContextMenuListener. Ini dipanggil saat Menu Konteks sedang dibuat (akibat "klik lama" terus-menerus).

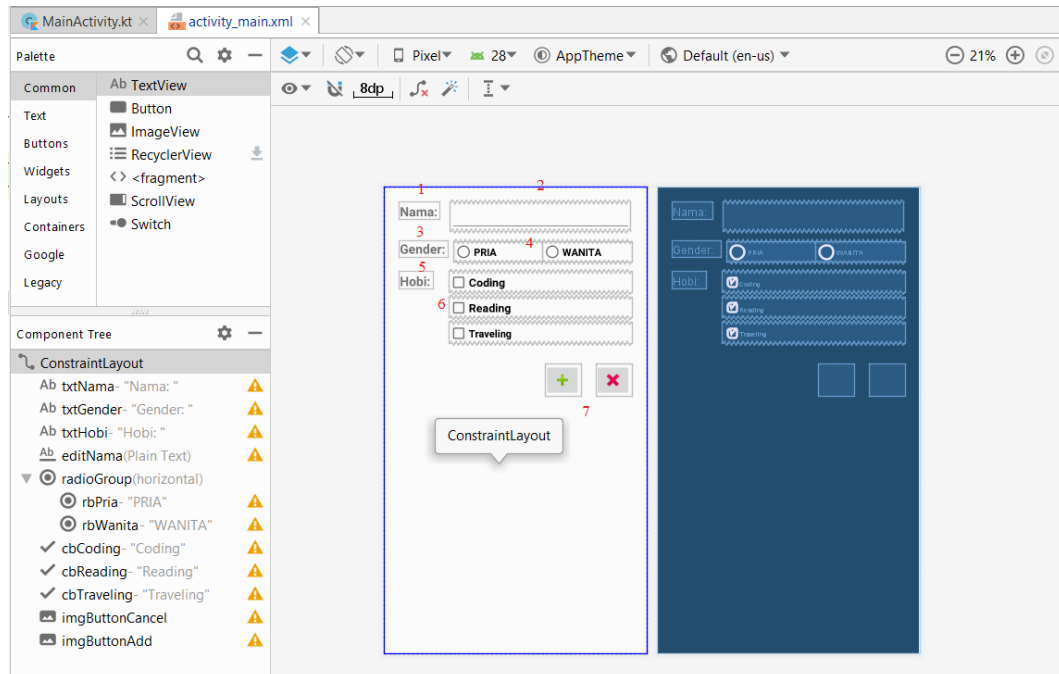


## **PRAKTIK**

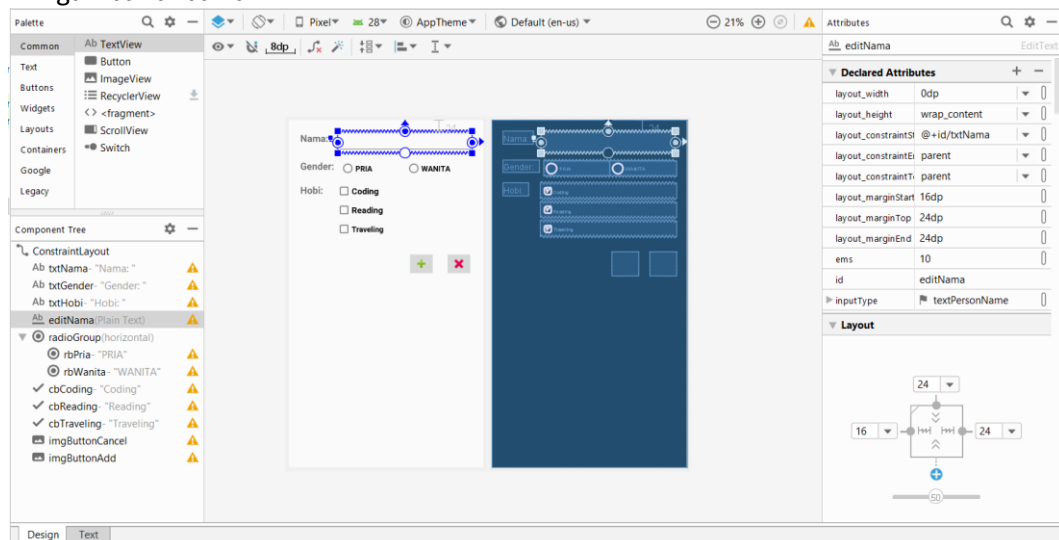
---

1. Buat Project Baru dengan nama **UIComponent**.

- Kemudian, buat desain layout seperti pada gambar dibawah. Gunakan tab design dan klik - drag komponen yang diperlukan ke layar desain.



- Perhatikan, atur atribut komponen widget dengan menggunakan tab attributes. Contoh, untuk komponen widget EditText, pengaturannya adalah seperti pada gambar di bawah.



- Untuk keseluruhan komponen widget, perhatikan nilai atribut di setiap komponen di bawah.
- Nomor 1, 3 dan 5 adalah komponen TextView.

```
<TextView
    android:text="Nama: "
    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:id="@+id/txtNama"
        android:layout_marginTop="24dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginStart="24dp"
        android:textSize="20sp"
        android:textStyle="bold" />
<TextView
    android:text="Gender: "
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/txtGender"
    app:layout_constraintTop_toBottomOf="@+id/txtNama"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_marginStart="24dp"
    android:layout_marginTop="32dp"
    android:textSize="20sp"
    android:textStyle="bold" />
<TextView
    android:text="Hobi: "
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/txtHobi"
    app:layout_constraintTop_toBottomOf="@+id/txtGender"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:textSize="20sp"
    android:textStyle="bold" />

```

6. Nomor 2 adalah komponen EditText.

```

<EditText
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:ems="10"
    android:id="@+id/editNama"
    android:layout_marginTop="24dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toEndOf="@+id/txtNama"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="24dp"
    app:layout_constraintEnd_toEndOf="parent" />

```

7. Nomor 4 ada dua macam komponen, satu buah RadioGroup dan dua buah RadioButton.

```

<RadioGroup
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/editNama"
    app:layout_constraintStart_toEndOf="@+id/txtGender"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginEnd="24dp"
    android:layout_marginTop="16dp"
    android:layout_marginStart="8dp"
    android:orientation="horizontal"
    android:id="@+id/radioGroup">
    <RadioButton
        android:text="PRIA"
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:id="@+id/rbPria"
        android:layout_weight="1"
        android:textSize="16sp"
        android:textStyle="bold"/>
<RadioButton
    android:text="WANITA"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/rbWanita"
    android:layout_weight="1"
    android:textSize="16sp"
    android:textStyle="bold"/>
</RadioGroup>

```

8. Nomor 6 ada 3 komponen CheckBox.

```

<CheckBox
    android:text="Coding"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:id="@+id/cbCoding"
    app:layout_constraintStart_toEndOf="@+id/txtHobi"
    android:layout_marginStart="24dp"
    android:layout_marginTop="16dp"
    app:layout_constraintTop_toBottomOf="@+id/radioGroup"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginEnd="24dp"
    android:textSize="18sp"
    android:textStyle="bold"/>
<CheckBox
    android:text="Reading"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:id="@+id/cbReading"
    android:layout_marginTop="8dp"
    app:layout_constraintTop_toBottomOf="@+id/cbCoding"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginEnd="24dp"
    app:layout_constraintStart_toEndOf="@+id/txtHobi"
    android:layout_marginStart="24dp"
    android:textSize="18sp"
    android:textStyle="bold"/>
<CheckBox
    android:text="Traveling"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:id="@+id/cbTraveling"
    android:layout_marginEnd="24dp"
    app:layout_constraintTop_toBottomOf="@+id/cbReading"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/txtHobi"
    android:layout_marginStart="24dp"
    android:layout_marginTop="8dp"
    android:textSize="16sp"
    android:textStyle="bold"/>

```

9. Nomor 7 adalah 2 komponen ImageButton. Perhatikan bahwa untuk ImageButton diperlukan file gambar, dalam contoh dibawah file gambarnya adalah ic\_delete dan ic\_input\_add.

```

<ImageButton
    android:layout_width="wrap_content"

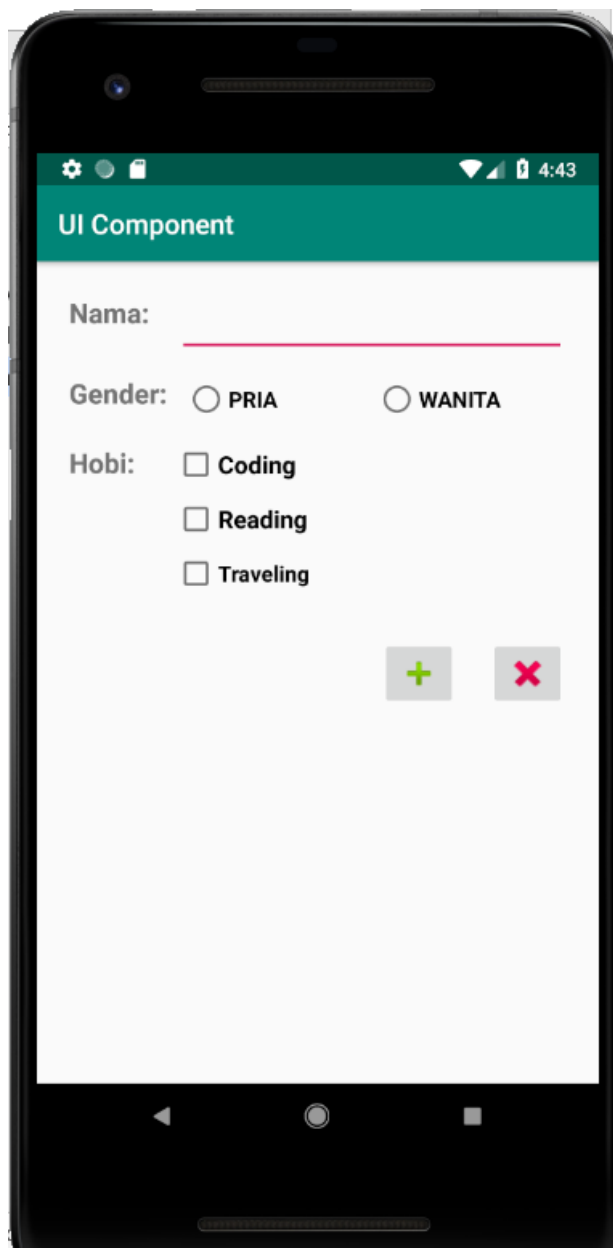
```

```

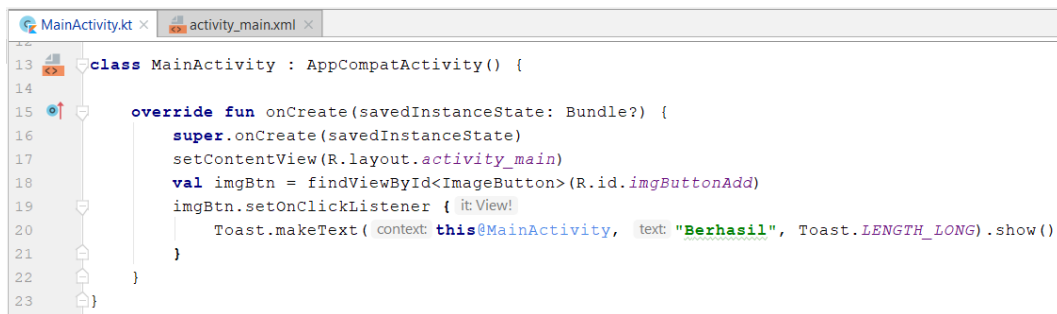
        android:layout_height="wrap_content"
        app:srcCompat="@android:drawable/ic_delete"
        android:id="@+id/imgButtonCancel"
        android:layout_marginTop="32dp"
        app:layout_constraintTop_toBottomOf="@+id/cbTraveling"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginEnd="24dp"/>
<ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@android:drawable/ic_input_add"
        android:id="@+id/imgButtonAdd"
        app:layout_constraintEnd_toStartOf="@+id/imgButtonCancel"
        android:layout_marginEnd="24dp"
        android:layout_marginTop="32dp"
        app:layout_constraintTop_toBottomOf="@+id/cbTraveling"/>

```

10. Setelah selesai desain, jalankan. Hasilnya adalah sebagai berikut.



11. Kita akan menambahkan event handling untuk ImageButton. Tambahkan koding pada MainActivity.kt, sehingga menjadi sebagai berikut.



```

13 class MainActivity : AppCompatActivity() {
14
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         setContentView(R.layout.activity_main)
18         val imgBtn = findViewById<ImageButton>(R.id.imgButtonAdd)
19         imgBtn.setOnClickListener { it: View!
20             Toast.makeText(context: this@MainActivity, text: "Berhasil", Toast.LENGTH_LONG).show()
21         }
22     }
23 }

```

12. Jalankan, dan beri event klik pada ImageButton plus.



## LATIHAN

---

1. Buat project baru, buat antar muka berbeda yang melibatkan komponen-komponen diatas.



## TUGAS

---

1. Analisislah atribut komponen untuk constrain layout.



## REFERENSI

---

1. <https://kotlinlang.org/docs/reference/>
2. <https://developer.android.com/kotlin>
3. <https://developer.android.com/courses/kotlin-android-fundamentals/toc>
4. <https://codelabs.developers.google.com/android-kotlin-fundamentals/>
5. <https://developer.android.com/kotlin/learn>
6. <https://developer.android.com/kotlin/resources>