

LAPORAN PRAKTIKUM

PEMROGRAMAN BERBASIS MOBILE

PERTEMUAN KE-5



Disusun Oleh :

NAMA : Raden Isnawan Argi Aryasatya
NIM : 195410257
JURUSAN : Informatika
JENJANG : S1
KELAS : 5

Laboratorium Terpadu
Sekolah Tinggi Manajemen Informatika Komputer
AKAKOM
YOGYAKARTA

2021

PERTEMUAN KE-5

(KOMPONEN WIDGET RADIOBUTTON DAN CHECKBOX)

TUJUAN

Mahasiswa mampu menggunakan Widget EditText, Button, RadioButton dan CheckBox untuk membuat aplikasi sederhana

DASAR TEORI

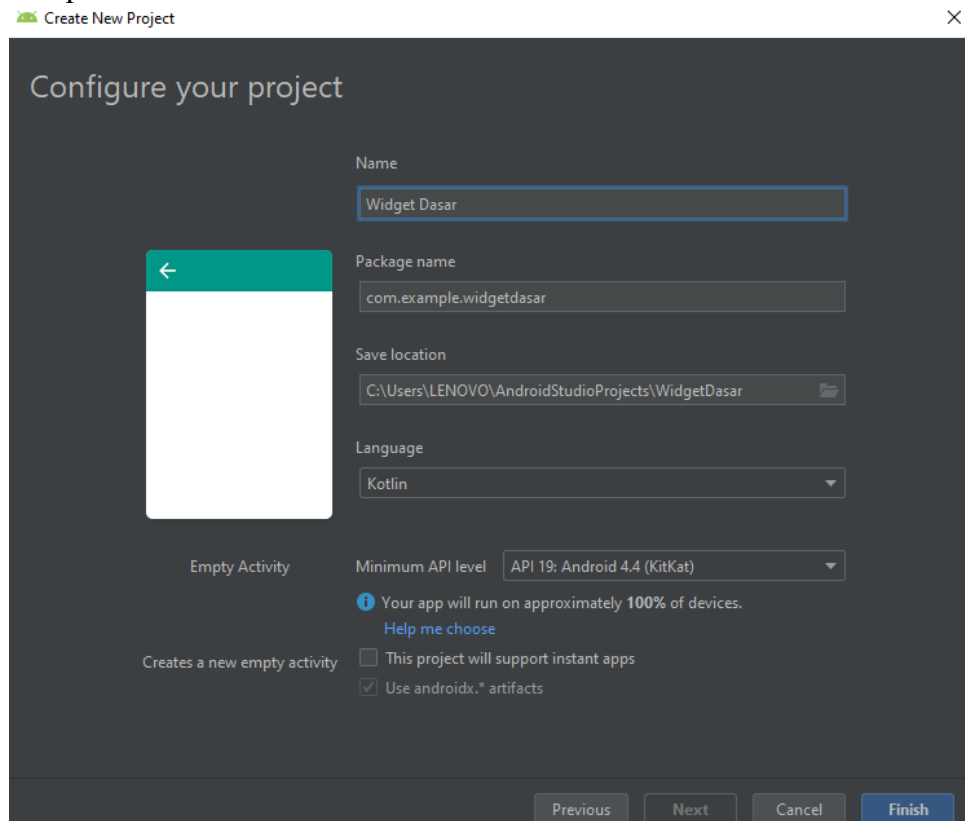
- **Button**
Button terdiri atas teks atau ikon (atau keduanya, teks dan ikon) yang mengkomunikasikan tindakan apa yang terjadi ketika pengguna menyentuhnya. Anda dapat membuat tombol di tata letak dengan tiga cara yaitu dengan teks, ikon, atau keduanya.
- **RadioButton**
Radio buttons memungkinkan user untuk memilih satu opsi dari sejumlah opsi yang tersedia. Sebagai contoh pilihan jenis kelamin, pilihan agama dan lainnya. Untuk menciptakan opsi radio button, buatlah RadioButton dalam layout. Namun, karena radio button bersifat mutually exclusive, Anda harus mengelompokkannya kedalam RadioGroup. Dengan mengelompokkannya, sistem akan memberikan hanya satu radio button yang dapat dipilih dalam satu waktu.
- **CheckBox**
Checkbox memungkinkan user untuk memilih satu atau lebih opsi dari sejumlah opsi yang tersedia. Pada umumnya anda menampilkan setiap opsi checkbox dalam daftar vertikal. Untuk menciptakan opsi checkbox, buatlah CheckBox dalam layout. Karena himpunan opsi checkbox memungkinkan user untuk memilih sejumlah item pilihan sekaligus, setiap checkbox dikelola secara terpisah dan anda harus mendaftarkan click listener untuk setiap checkbox.

PRAKTIK

1. Buatlah project dengan nama Widget Dasar dengan parameter berikut:

Attribute	Value
Template	Empty Activity dalam tab Phone and Tablet
Application Name	Widget Dasar
Company Name android	Com.android.example.widgetdasar (atau domain anda sendiri)
Language	Kotlin
Minimum API level	API 19: Android 4.4 (KitKat) atau API 21: Android 5.0 (Lollipop)
This project will support instant apps	(Biarkan box ini terhapus / tidak terpilih)
Use AndroidX artifacts	Pilih box ini.

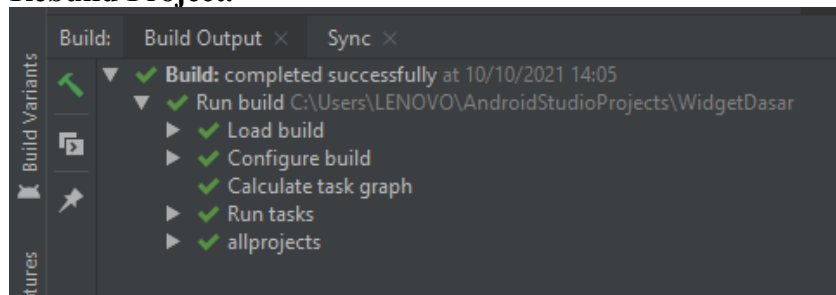
Tampilan di android studio:



Penjelasan:

bagian ini adalah bagian konfigurasi sebelum project dibuat. Di field name kita mengisi nama project kita yaitu “**Widget Dasar**”. Lalu ada sebuah field Package Name, yang kita tulis **com.example.widgetdasar** yang merupakan tempat untuk memberi nama package pada aplikasi yang akan kita buat di project. Semua aplikasi Android memiliki nama paket. Nama paket secara unik mengidentifikasi aplikasi pada perangkat. Nama paket juga bersifat untuk saat aplikasi sudah di-deploy Google Play. Di Save location kita pilih destinasi tempat menyimpan project yaitu **C:\Users\LENOVO\AndroidStudioProjects\WidgetDasar**. Lalu kita pilih bahasa yang akan kita gunakan di project “Widget Dasar” yaitu Kotlin. Setelah itu, di Minimum API Level (API Level adalah nilai integer yang secara unik mengidentifikasi framework revisi API yang ditawarkan oleh versi platform Android), kita memilih **API 19: Android 4.4 (KitKat)**. Terakhir, kita centang box **Use AndroidX artifacts**. AndroidX adalah proyek open source yang digunakan tim Android untuk mengembangkan, menguji, mengemas, versi, dan merilis pustaka dalam Jetpack.

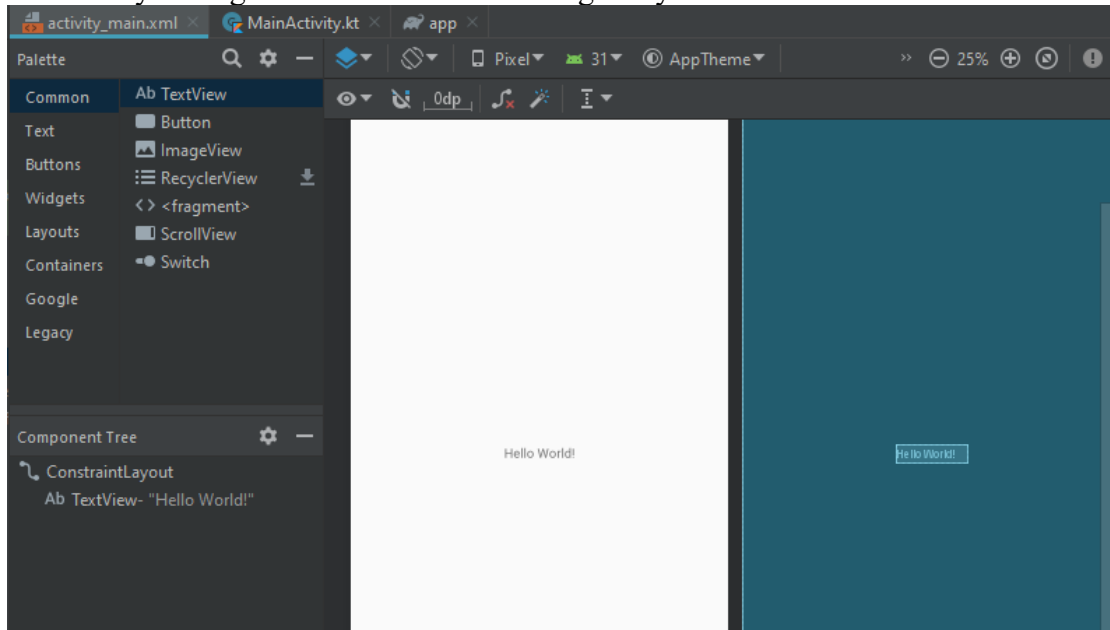
2. Tunggu Android Studio menyelesaikan Gradle build. Jika anda melihat error, pilih **Build > Rebuild Project**.



Penjelasan:

Gradle adalah sebuah program yang berfungsi untuk melakukan build secara otomatis. Karena itu, Gradle sering disebut sebagai build-tool. Build di sini bisa kita artikan sebagai aktivitas melakukan compile dan packaging.

3. Jalankan aplikasi dan tunggu beberapa saat sampai proses build selesai. Anda seharusnya melihat layar dengan **"Hello World!"** di tengah layar.



4. Buka file **activity_main.xml** dalam mode **Text**. Hapuslah widget **TextView** berikut.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

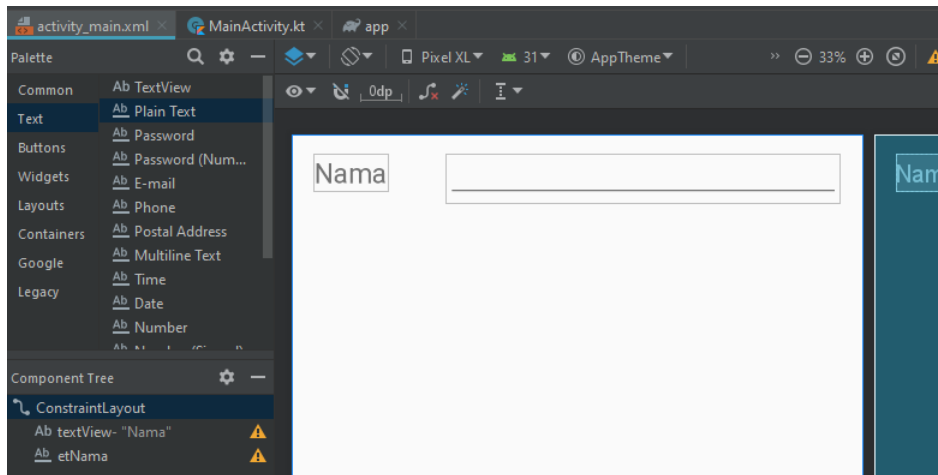
Penjelasan:

File ini adalah file layout yang tersedia di direktori res/layout, yang direferensikan oleh aplikasi kita saat membangun antarmukanya. Di android studio, kita bisa memodifikasi file ini untuk mengubah tata letak aplikasi. Untuk kasus di atas, kita hapus atau hilangkan widget TextView yang tadi memunculkan tulisan "Hello World!".

5. Tambahkan TextView dan EditText sehingga layout menjadi seperti berikut. Atribut android:id untuk EditText diset @+id/etNama

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="14dp"
    android:text="Nama"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/etNama"
    android:layout_width="284dp"
    android:layout_height="35dp"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="16dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />
```



Penjelasan:

TextView menampilkan teks kepada user dan secara opsional memungkinkan user untuk mengeditnya. TextView adalah editor teks lengkap, namun secara oromatis sudah dikonfigurasi untuk tidak mengizinkan pengeditan oleh user. Sementara EditText adalah overlay TextView telah dikonfigurasi supaya dapat diedit. EditText adalah subclass dari TextView menyertakan banyak kemampuan pengeditan. Di dalam kode di atas, kita berikan android:id ke EditText bernama "etNama"

6. Tambahkan TextView dan letakkan di bawah Nama. Tambahkan widget RadioGroup kedalam layout, dan tambahkan kedalam RadioGroup dua buah widget RadioButton, sehingga layout menjadi seperti berikut.



Kode XML untuk TextView "Kelamin" dan dua buah RadioButton adalah:

```

33
34 <TextView
35     android:id="@+id/textView2"
36     android:layout_width="wrap_content"
37     android:layout_height="wrap_content"
38     android:layout_marginStart="16dp"
39     android:layout_marginTop="17dp"
40     android:text="Kelamin"
41     android:textSize="20sp"
42     app:layout_constraintStart_toStartOf="parent"
43     app:layout_constraintTop_toBottomOf="@+id/textView" />
44
45 <RadioGroup
46     android:id="@+id/rg_kelamin"
47     android:layout_width="284dp"
48     android:layout_height="31dp"
49     android:layout_marginStart="8dp"
50     android:layout_marginEnd="16dp"
51     android:orientation="horizontal"
52     app:layout_constraintEnd_toEndOf="parent"
53     app:layout_constraintHorizontal_bias="1.0"
54     app:layout_constraintStart_toEndOf="@+id/textView2"
55     app:layout_constraintTop_toTopOf="@+id/textView2">
56
57     <RadioButton
58         android:id="@+id/rb_pria"
59         android:layout_width="121dp"
60         android:layout_height="wrap_content"
61         android:text="Pria"
62         android:textSize="20sp" />

```

```

62
63     <RadioButton
64         android:id="@+id/rb_wanita"
65         android:layout_width="124dp"
66         android:layout_height="wrap_content"
67         android:text="Wanita"
68         android:textSize="20sp" />
69
70 </RadioGroup>

```

Penjelasan:

Pertama kita membuat textview yang memiliki id “textView2”. Untuk width dan height, kita setting wrap_content yang digunakan agar lebar atau tinggi, dapat mengikuti object didalam komponen tersebut. Kita juga berikan marginStart 16dp dan marginTop 17dp. Sementara untuk text nya kita tuliskan “Kelamin” dengan textSize 20sp. constraintStart nya parent (mengikuti layar atau screen pada perangkat Android), dan constraintBottom nya adalah “textView” yang berarti “textView2” diletakkan berada di bawah “textView”. Lalu kita tambahkan juga RadioGroup. RadioGroup digunakan untuk membuat lingkup multiple-exclusion untuk satu set RadioButton. Radio buttons mengijinkan user untuk memilih satu opsi dari sejumlah opsi yang tersedia.

7. Tambahkan kedalam dua buah **RadioButton** atribut berikut (**android:onClick="onRbClicked"**)

```

<RadioButton
    android:id="@+id/rb_pria"
    android:layout_width="121dp"
    android:layout_height="wrap_content"
    android:text="Pria"
    android:textSize="20sp"
    android:onClick="onRbClicked"/>

<RadioButton
    android:id="@+id/rb_wanita"
    android:layout_width="124dp"
    android:layout_height="wrap_content"
    android:text="Wanita"
    android:textSize="20sp"
    android:onClick="onRbClicked"/>

```

Penjelasan:

onRbClicked merupakan fungsi yang muncul ketika user menyentuh button. Konfigurasi button fungsi tersebut akan dibuat di MainActivity.kt.

8. Tambahkan kedalam kelas **MainActivity** method(fungsi) **onRbClicked** seperti berikut.

```

1 package com.example.widgetdasar
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import android.widget.Toast
7 import android.widget.RadioButton
8 import kotlinx.android.synthetic.main.activity_main.*
9
10 class MainActivity : AppCompatActivity() {
11
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_main)
15     }
16
17     fun onRbClicked(view: View) {
18         if (view is RadioButton) {
19             // is the button now checked?
20             val checked = view.isChecked

```

```

20
21 //when which radio button was clicked
22 when (view.getId()) {
23     R.id.rb_pria ->
24         if (checked) {
25             Toast.makeText(applicationContext,
26                 text: "On click : ${rb_pria}.text",
27                 Toast.LENGTH_SHORT).show()
28         }
29     R.id.rb_wanita ->
30         if (checked) {
31             Toast.makeText(applicationContext,
32                 text: "On click : ${rb_wanita}.text",
33                 Toast.LENGTH_SHORT).show()
34         }
35     }
36 }
37 }

```

Penjelasan:

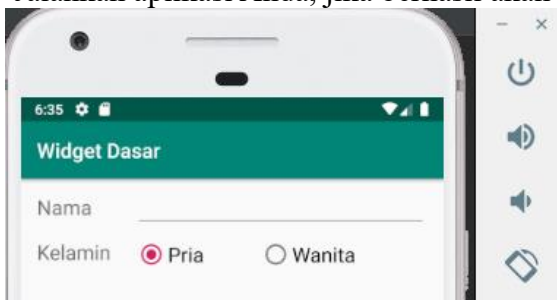
Di file kotlin ini, pertama kita import beberapa paket yang kita butuhkan yaitu:

- androidx.appcompat.app.AppCompatActivity** yang membuat sebuah activity untuk project dan menyediakan berbagai fitur untuk ActionBar.
- android.os.Bundle** untuk menjalankan atau mengoper data antar activity. Nilai yang akan diteruskan dipetakan ke kunci String yang kemudian digunakan dalam activity berikutnya untuk mengambil nilai.
- android.view.View** adalah blok bangunan dasar UI (User Interface) di android. Method ini juga berfungsi untuk melihat view yang telah diklik pengguna
- android.widget.Toast**. Toast adalah tampilan yang berisi pesan kecil berdurasi cepat untuk pengguna.
- android.widget.RadioButton**. RadioButton adalah tombol dua status yang dapat dicheck atau tidak dicheck. Ketika radiobutton tidak dicentang, pengguna dapat menekan atau mengkliknya untuk memeriksanya. Namun, tombol radio tidak dapat di-uncheck oleh pengguna setelah terlanjur dicentang.
- kotlinx.android.synthetic.main.activity_main.***. Synthetics memungkinkan aplikasi pada project untuk mengganti panggilan ke findViewById dengan binding **kotlinx.android.synthetic**

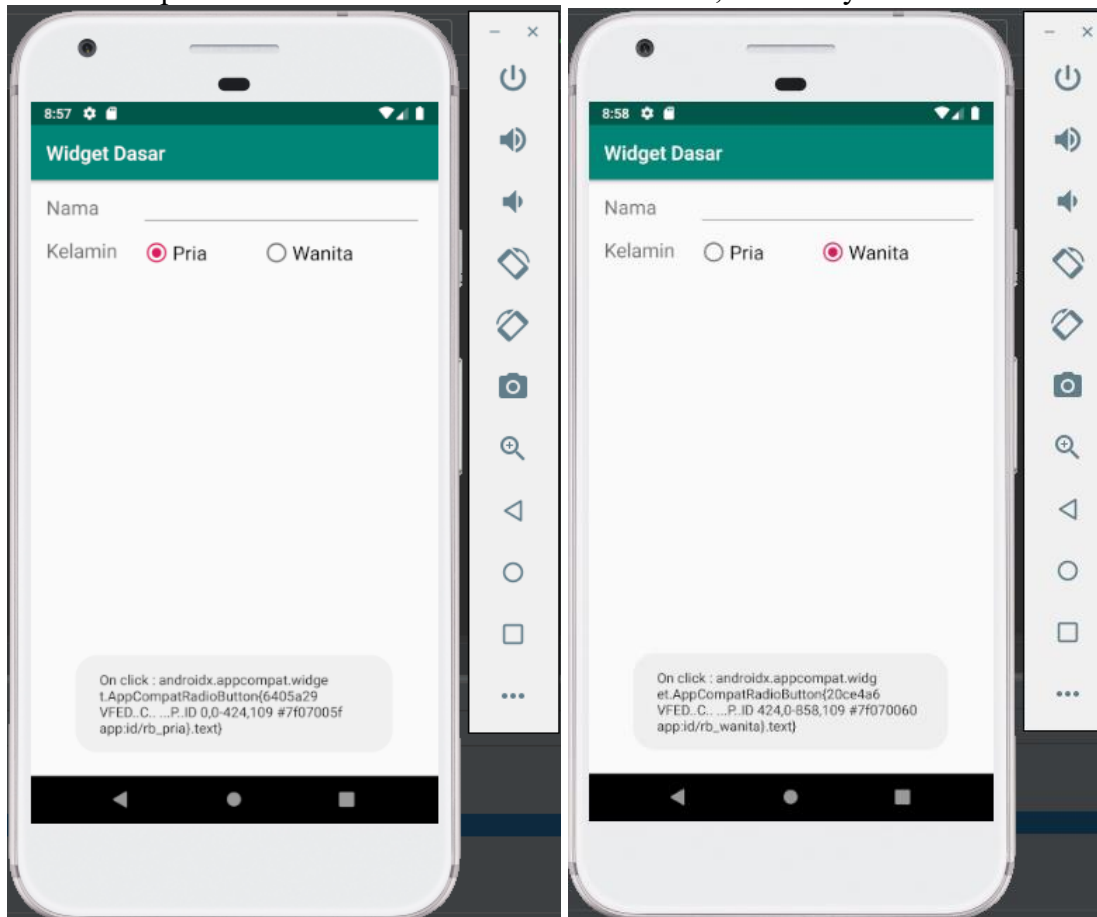
setelah itu kita buat fungsi onCreate yang berguna untuk memulai sebuah aktivitas aplikasi. Dilanjutkan dengan fungsi onRbClicked yang memiliki argumen (view: View) yang berguna sebagai argumen ketika method untuk listener dan view (Button, Spinner, Switch, dll.) digunakan untuk membedakan view mana pada layout yang telah diklik/dipilih. Di dalamnya kita buat beberapa baris kode untuk RadioButton yaitu untuk pria dan wanita.

Di baris kode untuk pria, kita buat argumen **if(checked)** yang artinya “jika tombol disentuh, maka..”, kemudian kita memerintahkan program untuk memunculkan toast ketika kita menyentuh button Pria. Hal tersebut berlaku juga dengan button wanita

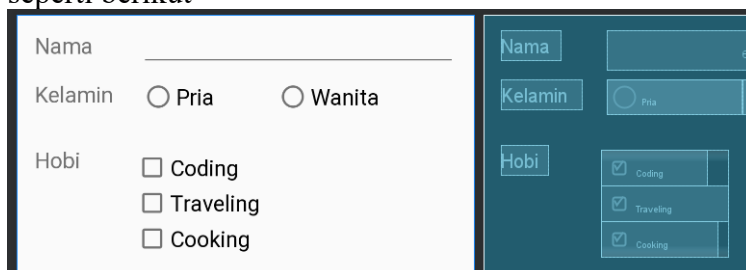
- Jalankan aplikasi Anda, jika berhasil akan terlihat seperti berikut.



10. Sentuhlah / pilihlah kelamin Pria dan kemudian Wanita, seharusnya muncul Toast.



11. Tambahkan kedalam file layout TextView dan tiga buah widget CheckBox, dengan desain seperti berikut



Kode XML untuk desain di atas adalah seperti berikut.

```

activity_main.xml x MainActivity.kt x app x
72
73 <TextView android:id="@+id/textView3"
74         android:layout_width="wrap_content"
75         android:layout_height="wrap_content"
76         android:layout_marginStart="16dp"
77         android:layout_marginTop="32dp"
78         android:text="Hobi"
79         android:textSize="20sp"
80         app:layout_constraintStart_toStartOf="parent"
81         app:layout_constraintTop_toBottomOf="@+id/textView2" />
82
83 <LinearLayout android:layout_width="wrap_content"
84         android:layout_height="wrap_content"
85         android:layout_marginLeft="48dp"
86         android:layout_marginTop="32dp"
87         android:orientation="vertical"
88         app:layout_constraintLeft_toRightOf="@id/textView3"
89         app:layout_constraintTop_toBottomOf="@id/rg_kelamin">

```



```

90
91 <CheckBox android:id="@+id/cb_coding"
92         android:layout_width="wrap_content"
93         android:layout_height="wrap_content"
94         android:textSize="20sp"
95         android:onClick="onCbClicked"
96         android:text="Coding" />
97
98 <CheckBox android:id="@+id/cb_traveling"
99         android:layout_width="wrap_content"
100        android:layout_height="wrap_content"
101        android:textSize="20sp"
102        android:onClick="onCbClicked"
103        android:text="Traveling" />
104
105 <CheckBox android:id="@+id/cb_cooking"
106         android:layout_width="wrap_content"
107         android:layout_height="wrap_content"
108         android:textSize="20sp"
109         android:onClick="onCbClicked"
110         android:text="Cooking" />
111
112 </LinearLayout>

```

Penjelasan:

Kode-kode di atas berguna untuk memunculkan widget berupa checkbox coding, traveling, dan cooking. Untuk bagian width dan height semua menggunakan `wrap_content` yang digunakan agar lebar atau tinggi, dapat mengikuti object didalam komponen tersebut. Ukuran text adalah 20sp. Dan `onClick` nya menggunakan referensi method `onCbClicked` yang ada di `MainActivity.kt`.

12. Tambahkan kedalam kelas **MainActivity** method(fungsi) **onCbClicked** seperti berikut.

```

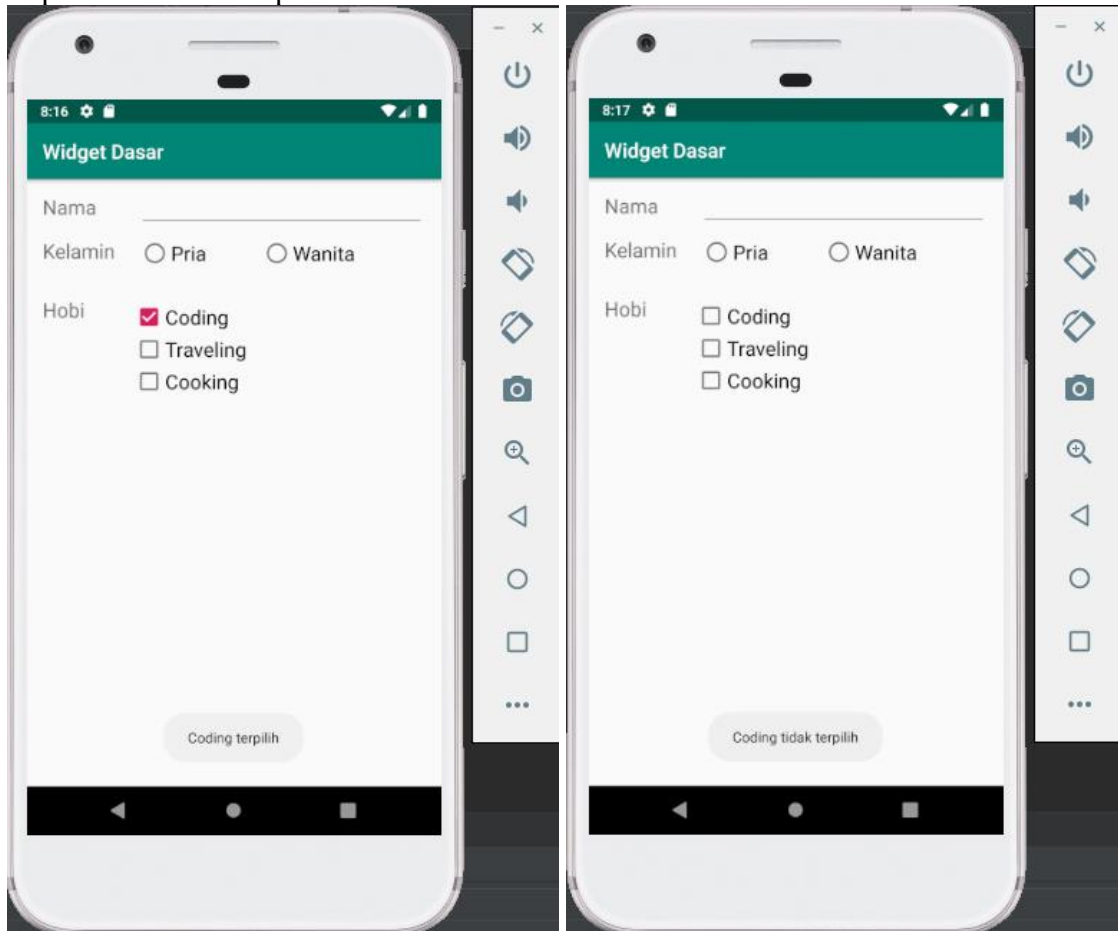
39 fun onCbClicked(view: View) {
40     if (view is CheckBox) {
41         val checked: Boolean = view.isChecked
42         when (view.id) {
43             R.id.cb_coding -> {
44                 if (checked) {
45                     Toast.makeText(applicationContext,
46                                 text: "${cb_coding.text} terpilih",
47                                 Toast.LENGTH_SHORT).show()
48                 } else {
49                     Toast.makeText(applicationContext,
50                                 text: "${cb_coding.text} tidak terpilih",
51                                 Toast.LENGTH_SHORT).show()
52                 }
53             }
54         }
55     }
56 }

```

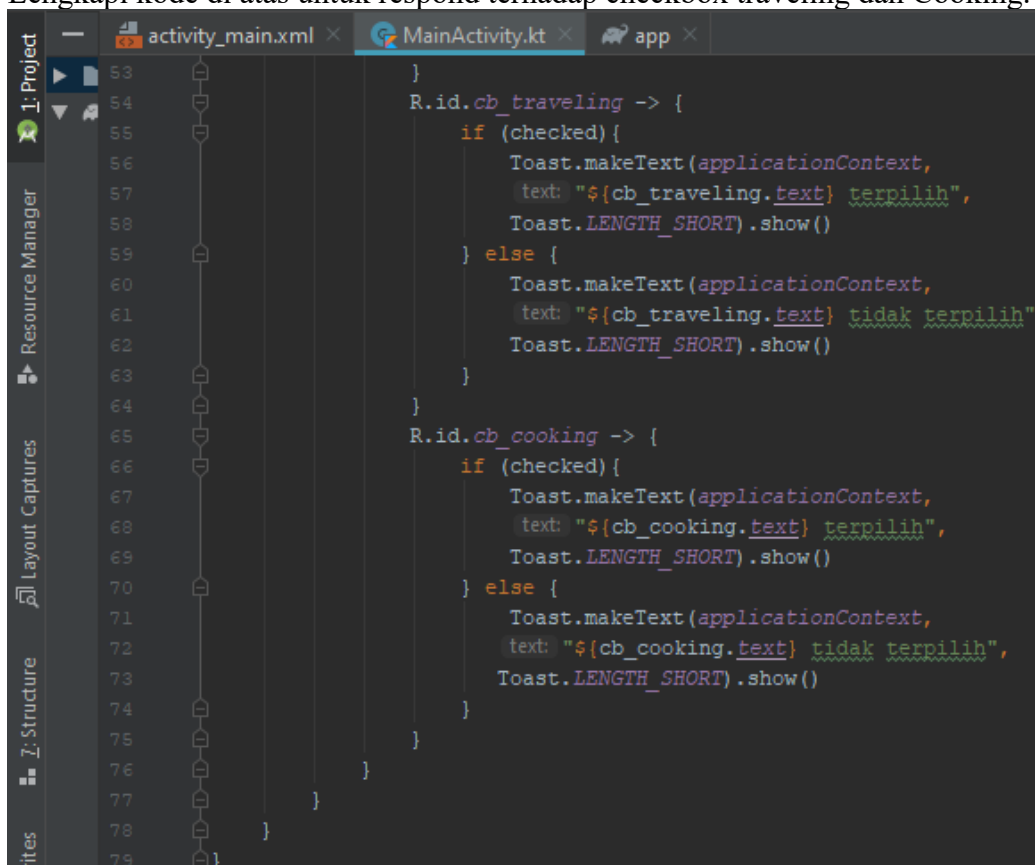
Penjelasan:

`onCbClicked` yang memiliki argumen (`view: View`) yang berguna sebagai argumen ketika method untuk listener dan view (Button, Spinner, Switch, dll.) digunakan. kita buat argumen **`if(checked)`** yang artinya “jika tombol disentuh, maka..”, kemudian kita memerintahkan program untuk memunculkan toast ketika kita menyentuh button coding. Lalu kita gunakan **`Toast.makeText`** untuk membuat text saat user meng-klik checkbox yaitu dengan kode **`"${cb_coding.text} terpilih"`**, dan untuk memunculkan tulisan atau pesan toast tersebut kita gunakan kode **`Toast.LENGTH_SHORT).show()`**. Jika user meng-uncheck checkbox maka kita langsung saja tuliskan beberapa baris kode untuk membuat sebuah pesan toast yaitu **`Toast.makeText(applicationContext, "${cb_coding.text} tidak terpilih"`**, dan untuk menampilkan pesan toast tersebut gunakan **`Toast.LENGTH_SHORT).show()`**

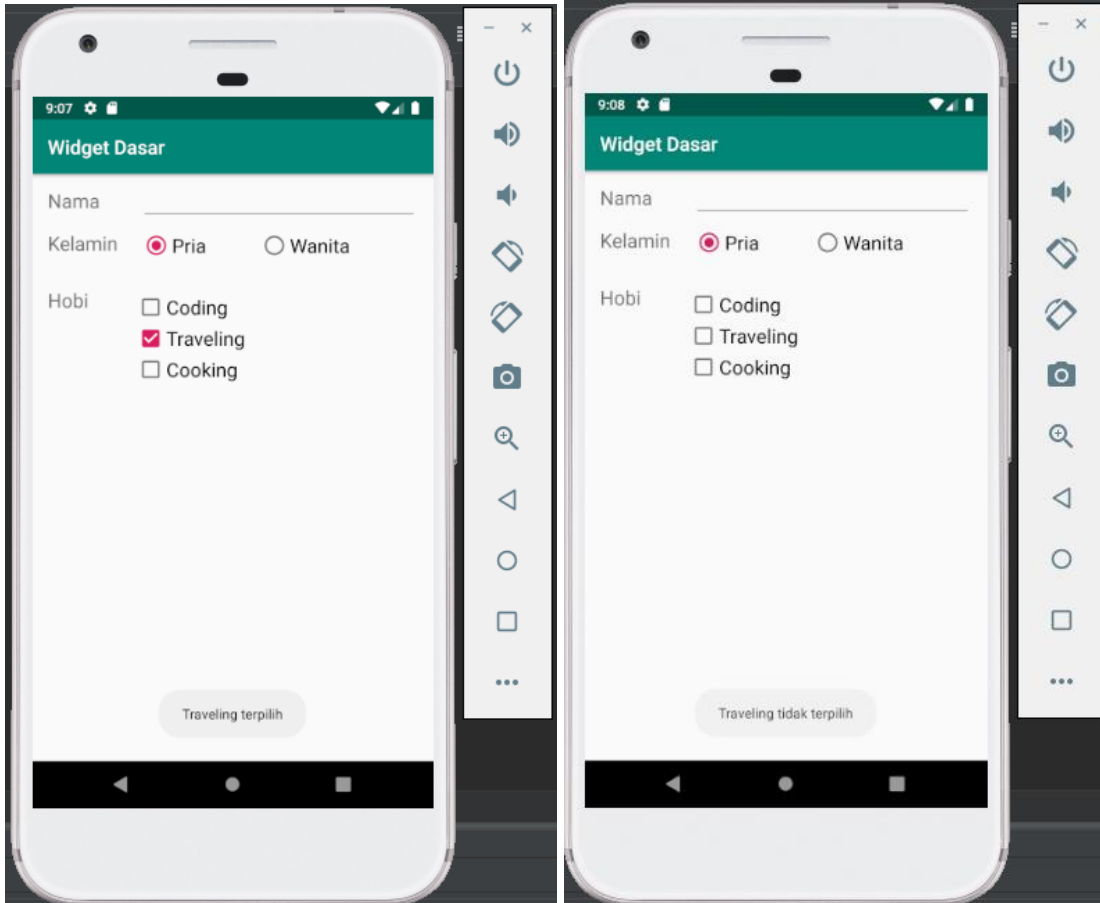
13. Jalankan aplikasi Anda, jika berhasil akan terlihat seperti berikut. Sentuhlah opsi coding agar terpilih atau tidak terpilih.



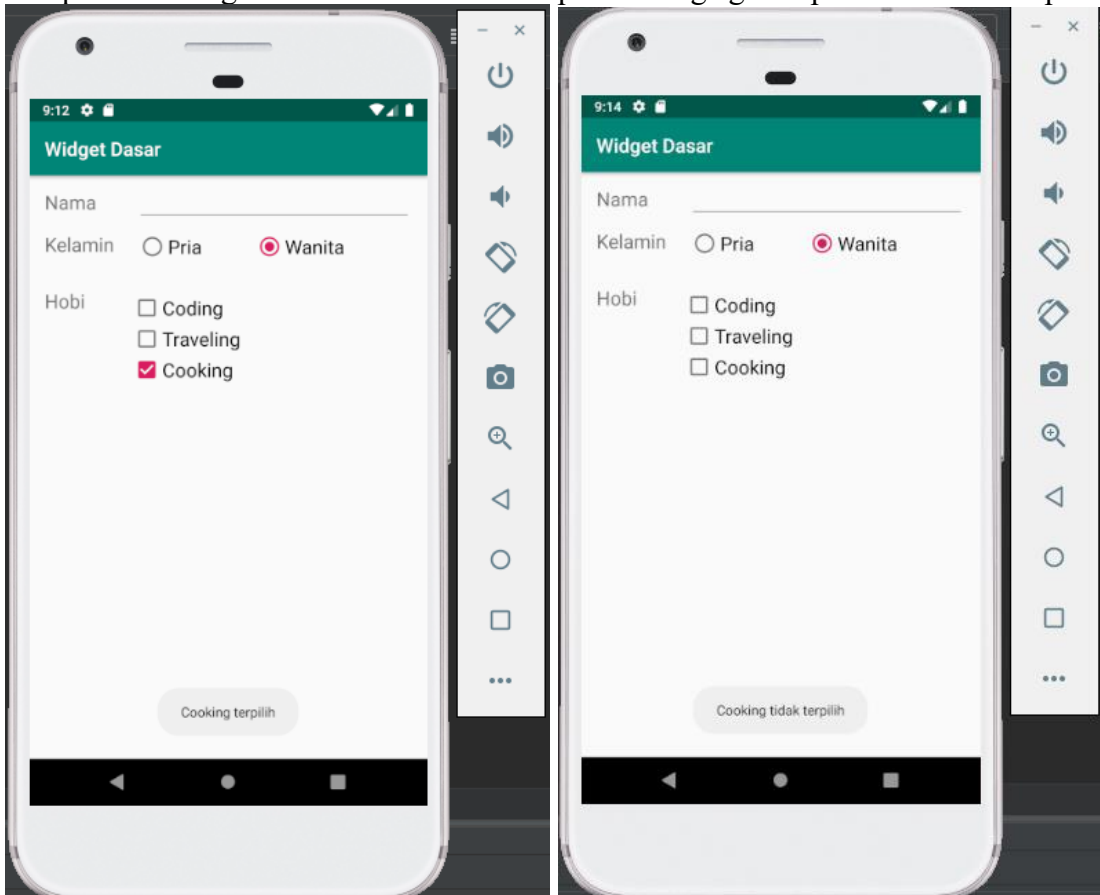
14. Lengkapi kode di atas untuk respond terhadap checkbox traveling dan Cooking.



Tampilan traveling di emulator. Sentuhlah opsi traveling agar terpilih atau tidak terpilih.



Tampilan cooking di emulator. Sentuhlah opsi cooking agar terpilih atau tidak terpilih.



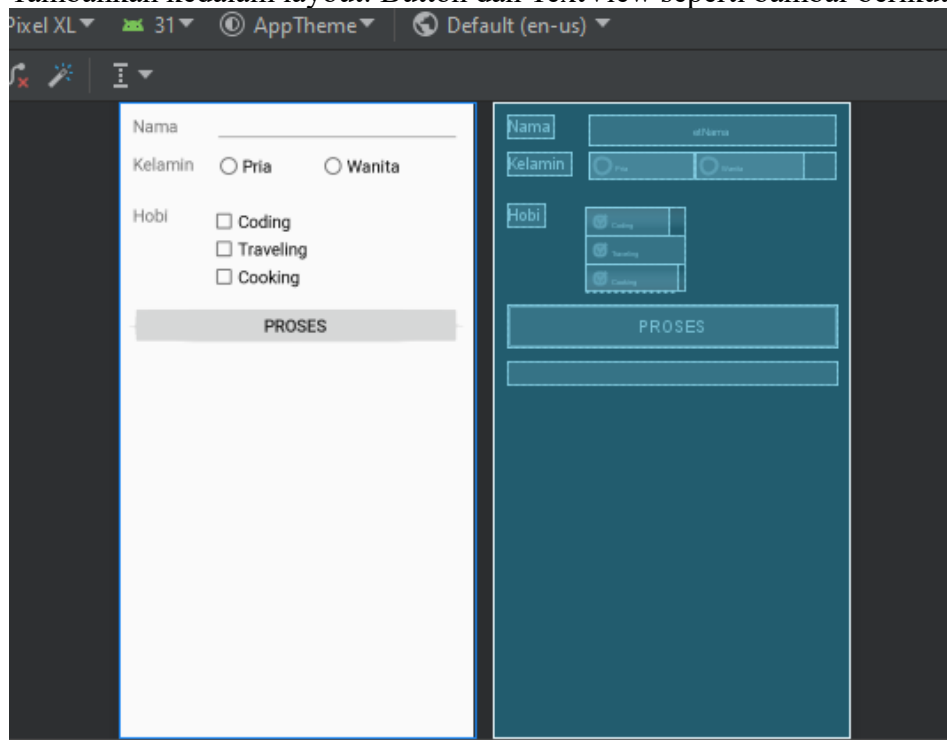
Penjelasan:

Kita memerintahkan program untuk memunculkan toast ketika kita menyentuh radio button traveling. Lalu kita gunakan **Toast.makeText** untuk membuat text saat user meng-klik checkbox yaitu dengan kode **"\${cb_traveling.text} terpilih"**, dan untuk memunculkan pesan toast tersebut kita gunakan kode **Toast.LENGTH_SHORT).show()**. Jika user meng-uncheck checkbox maka kita langsung tuliskan kode untuk membuat pesan toast yaitu **Toast.makeText(applicationContext, "\${cb_traveling.text} tidak terpilih"**, dan untuk menampilkan pesan toast tersebut gunakan **Toast.LENGTH_SHORT).show()**

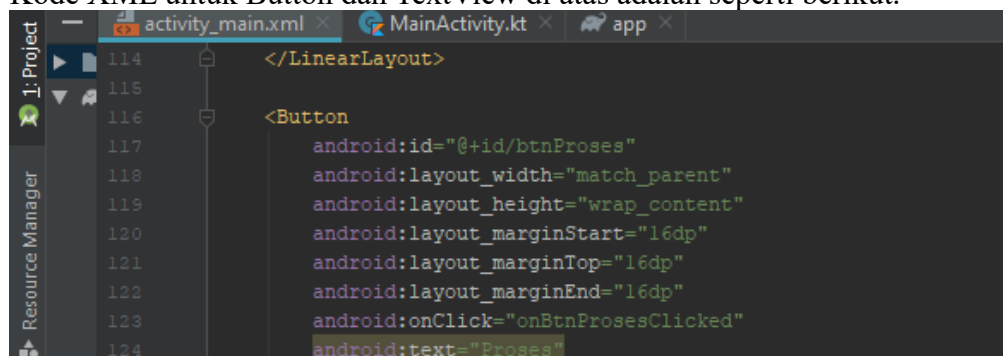
yang kedua kita memerintahkan program untuk memunculkan toast ketika kita menyentuh radio button cooking. Lalu kita gunakan **Toast.makeText** untuk membuat text saat user meng-klik checkbox yaitu dengan kode **"\${cb_cooking.text} terpilih"**, dan untuk memunculkan tulisan atau pesan toast tersebut kita gunakan kode **Toast.LENGTH_SHORT).show()**. Jika user meng-uncheck checkbox maka kita langsung tuliskan kode untuk membuat pesan toast yaitu **Toast.makeText(applicationContext, "\${cb_coocking.text} tidak terpilih"**, dan untuk menampilkan pesan toast tersebut gunakan **Toast.LENGTH_SHORT).show()**

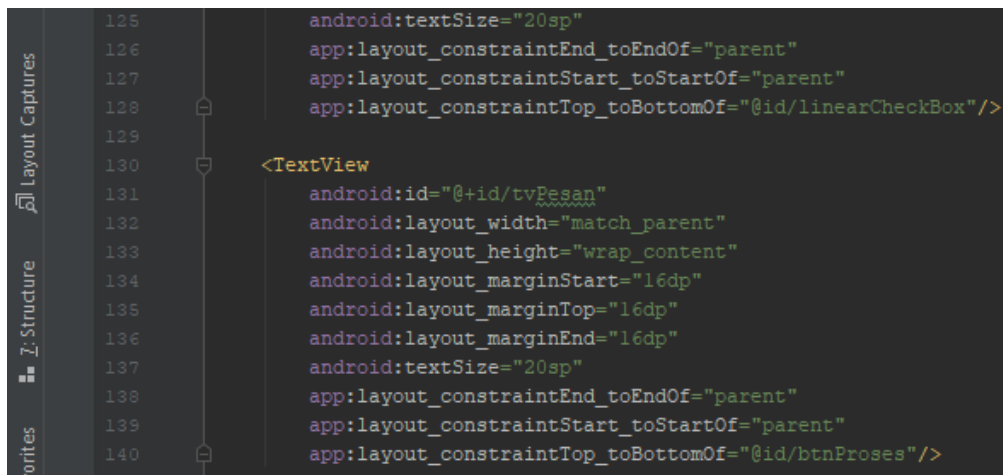
LATIHAN

1. Tambahkan kedalam layout: Button dan TextView seperti bambar berikut.



Kode XML untuk Button dan TextView di atas adalah seperti berikut.





Penjelasan:

widget Button yang kita buat memiliki id btnProses dengan width match_parent yang artinya menyesuaikan lebar layout, dan height wrap_content yang berarti menyesuaikan komponen objek di dalam layout. marginStart, marginTop, dan marginEnd memiliki besar yang sama yaitu 16dp. Kemudian kita beri fungsi onClick bernama onBtnProsesClicked yang nanti akan diproses di MainActivity.kt. Button ini memiliki tulisan "Proses" sebesar 20sp.

Widget selanjutnya adalah TextView yang memiliki id tvPesan dengan width match_parent yang artinya menyesuaikan lebar layout, dan height wrap_content yang berarti menyesuaikan komponen objek di dalam layout. marginStart, marginTop, dan marginEnd memiliki besar yang sama yaitu 16dp. TextView ini tidak kita berikan text karena nanti TextView ini akan otomatis menampung text yang muncul saat kita menekan Button. Text yang akan muncul di TextView memiliki besar 20sp.

2. Tambahkan kedalam kelas **MainActivity** method (fungsi) **onBtnProsesClicked** seperti berikut.

```

}
}
fun onBtnProsesClicked(view: View){
    if(view is Button){
        btnProses.setOnClickListener{ it: View!
            tvPesan.text = "Mbak ${etNama.text}, kamu suka ${cb_coding.text}, ${cb_traveling.text}"
        }
    }
}
}

```

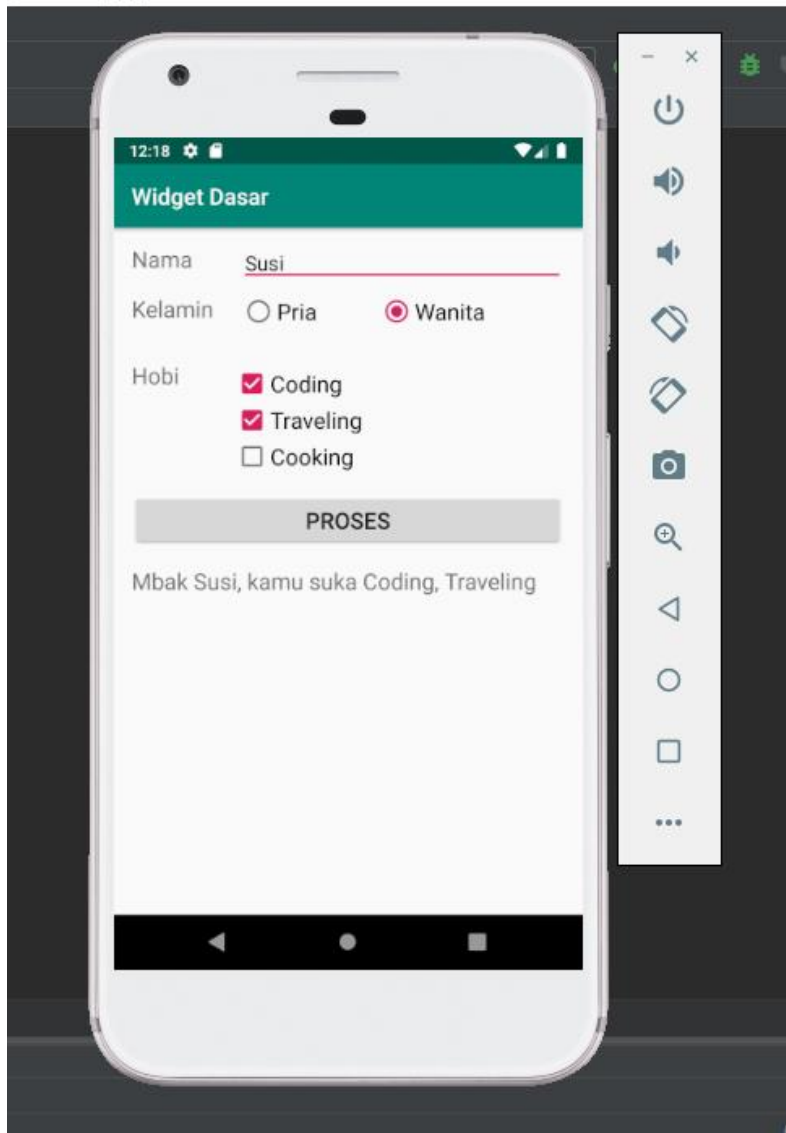
Penjelasan:

Di MainActivity, kita membuat fungsi/method bernama **fun onBtnProsesClicked** yang memiliki argumen (**view: View**) yang berguna untuk inisialisasi widget yang disentuh user. Kemudian kita buat seleksi kondisi berupa if(view is Button) yang menandakan jika widget button disentuh oleh user. Lalu dilanjutkan dengan inisialisasi btnProses dengan salahsatu fungsi built-in yaitu setOnClickListener yang berguna untuk melakukan suatu proses jika user menekan button. Di bawahnya, kita perintahkan supaya muncul pesan/kalimat text di textView tvPesan dengan baris kode **tvPesan.text = "Mbak \${etNama.text}, kamu suka \${cb_coding.text}, \${cb_traveling.text}"**. etNama adalah nama yang kita masukkan di widget TextView petama, cb_coding merupakan radiobutton coding, dan cb_traveling adalah radiobutton traveling. Kemudian , aplikasi langsung saja kita run di salah satu emulator virtual device yaitu Pixel XL API 28.

(screenshot saat run aplikasi di halaman selanjutnya)

Run aplikasi di emulator

_main.xml [app] - Android Studio



KESIMPULAN

Di laporan ini, saya belajar banyak tentang komponen widget RadioButton dan CheckBox. Di pertemuan ini saya juga belajar tentang Button, EditText, TextView secara lebih dalam dan lebih detail. Saat mengerjakan laporan ini, saya berhasil memenuhi tujuan dari dibuatnya laporan ini yaitu mampu menggunakan Widget EditText, Button, RadioButton dan CheckBox untuk membuat aplikasi sederhana. Saya pun belajar tentang fungsi built-in Button seperti onClick untuk respons terhadap suatu event klik, dan OnClickListener untuk mendeklarasikan event handler secara terprogram. Saya juga belajar tentang cara mengelompokkan RadioButton, yaitu dengan RadioGroup. Untuk bagian CheckBox, saya mengetahui jika user juga bisa memilih sejumlah item pilihan sekaligus, dan setiap checkbox dikelola secara terpisah dan kita harus mendaftarkan click listener untuk setiap checkbox. Inti dari laporan ini adalah mengasah kemampuan mahasiswa untuk menggunakan RadioButton dan CheckBox, serta meningkatkan kemampuan beberapa widget yang sudah dibahas di laporan sebelumnya yaitu Button, EditText, dan TextView.

Terima Kasih