



# Android Navigation and Interface Design 1

PBP, SEMESTER GANJIL T.A. 2022-  
2023



Pemrograman Berbasis Platform  
PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ATMA JAYA YOGYAKARTA

## TUJUAN

Setelah menyelesaikan modul ini, praktikan diharapkan mampu:

1. Membuat dan menggunakan activity dalam suatu UI dan aplikasi.
2. Menggabungkan beberapa activity dalam sebuah aplikasi.
3. Menggunakan intent untuk memanggil suatu activity.
4. Memahami penggunaan syntax import library.

## DASAR TEORI

### A. Activity

Suatu User Interface (UI) aplikasi ditampilkan dalam layar sebuah perangkat melalui Activity. Pada dasarnya 1 Activity digunakan untuk 1 UI aplikasi (tampilan bagi penggunanya). Ketika aplikasi digunakan, di dalam memori terdapat tumpukan Activity atau “stack of Activities”. Ketika berpindah dari suatu UI ke UI lain, Activity dari UI yang selanjutnya akan diletakkan (PUSH) di tumpukan/Stack yang paling atas. Sehingga Activity yang ditampilkan di layar adalah Activity yang paling atas. Activity akan dikeluarkan (POP) dari tumpukan/Stack dengan menekan tombol Back yang kemudian akan ditampilkan Activity yang sebelumnya dan melanjutkan (resume) state dari Activity tersebut.

### B. Intent

Istilah ‘Intent’ berasal dari kata “intention”. Intent adalah deskripsi abstrak dari suatu operasi / task yang akan dilakukan. Intent dapat digunakan dengan startActivityForResult untuk memanggil suatu Activity, dengan broadcastIntent untuk mengirim sesuatu ke semua komponen BroadcastReceiver, dan startService(Intent) atau bindService(Intent, ServiceConnection) untuk berkomunikasi dengan suatu background Service. Dengan kata lain, intent dapat membantu mengkomunikasikan atau menghubungkan suatu bagian dengan bagian lain dari sebuah aplikasi. Dalam hal ini, bagian aplikasi yang dimaksud adalah sebuah Activity.

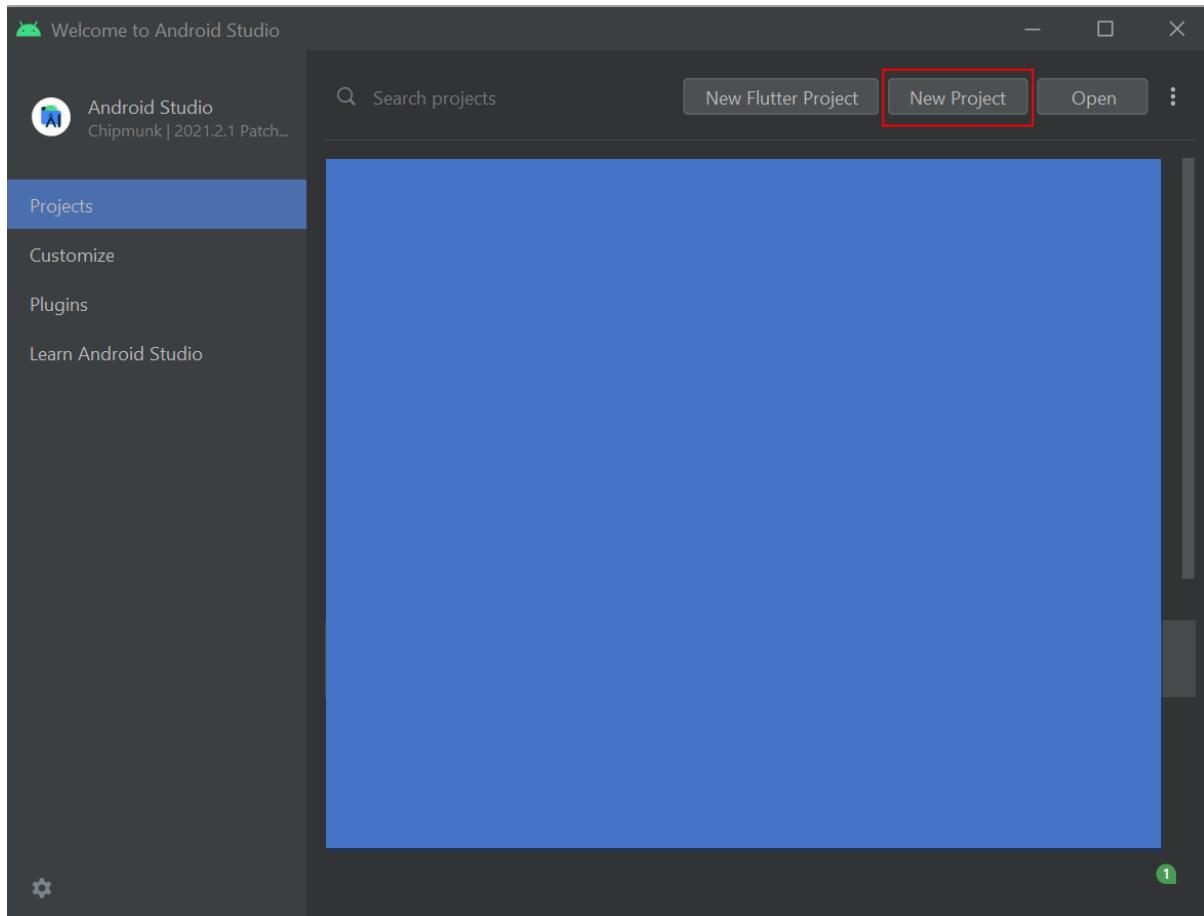
## GUIDED 1 – HELLO WORLD

Poin yang dipelajari dalam Guided 1 ini, yaitu:

1. Memahami cara membuat projek.
2. Memahami struktur folder dan file dalam sebuah projek.
3. Memahami prosedur dasar untuk menjalankan activity.

Pada guided 1 ini, kita akan mencoba memahami activity dengan membuat aplikasi Hello World pada platform Android 4.0.1. Silahkan ikuti langkah – langkah berikut ini :

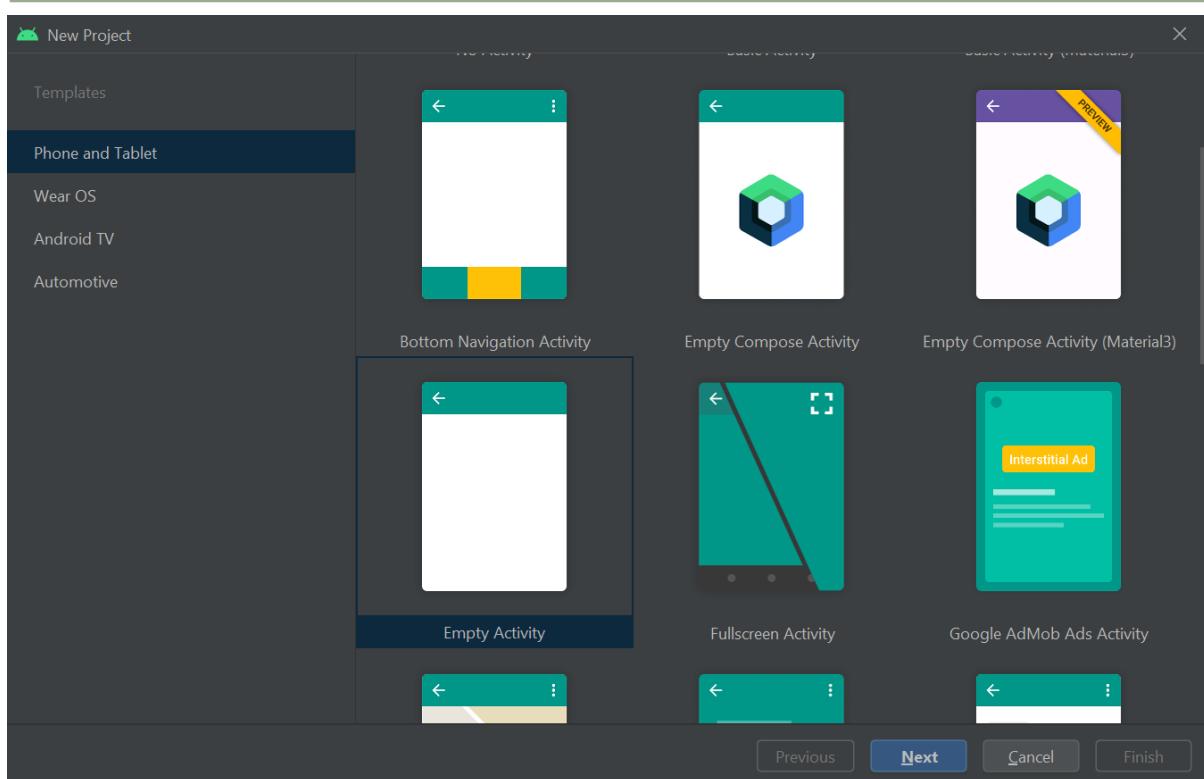
1. Buka aplikasi Android Studio, pilih **New Project**.



Gambar 1. Klik New Project

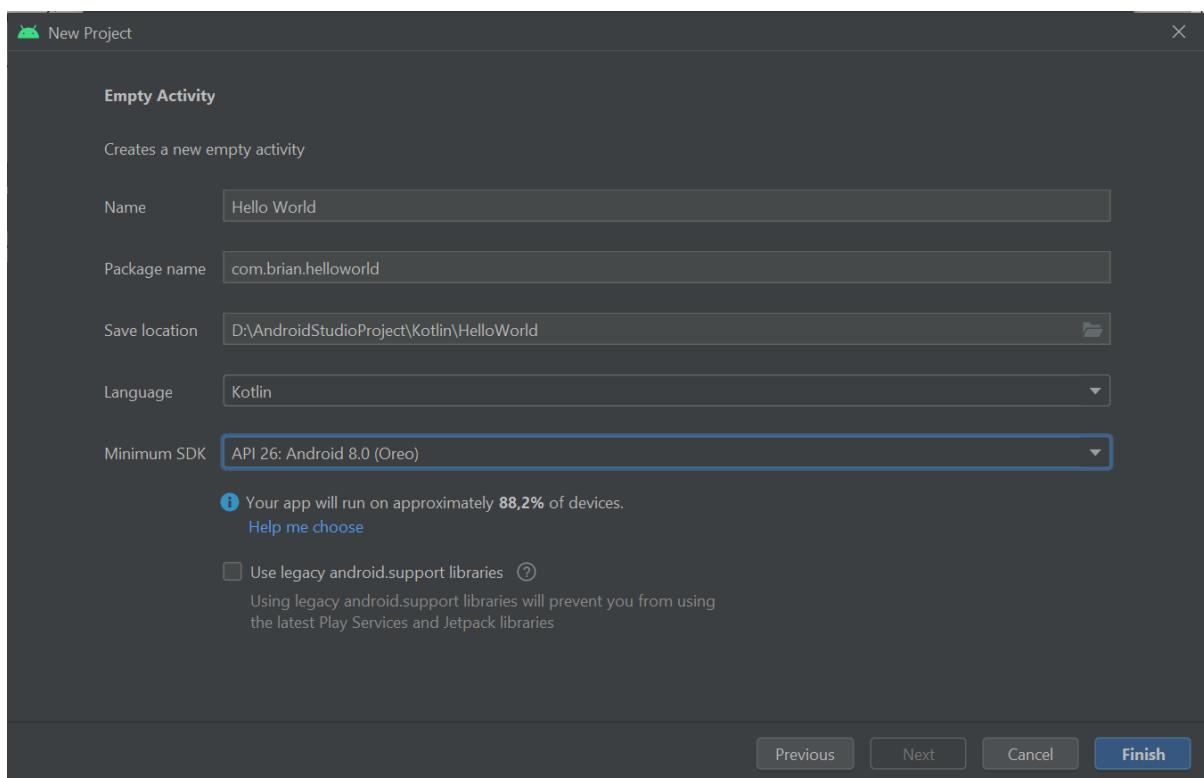
2. Kemudian, pilih Template **Empty Activity**, lalu klik **Next**.

## ANDROID NAVIGATION AND INTERFACE DESIGN 1



Gambar 2. Klik Empty Activity

3. Kemudian isikan kolom **Name**, **Package Name**, dan pilih lokasi penyimpanan projek (**Save Location**). Lalu, pilih **Language Kotlin** dan **Minimum SDK** . Klik **Finish**.



Gambar 3. Isikan kolom yang ada sesuai dengan kebutuhan

**NOTE:**

**Application Name** adalah nama untuk aplikasi yang akan dibuat.

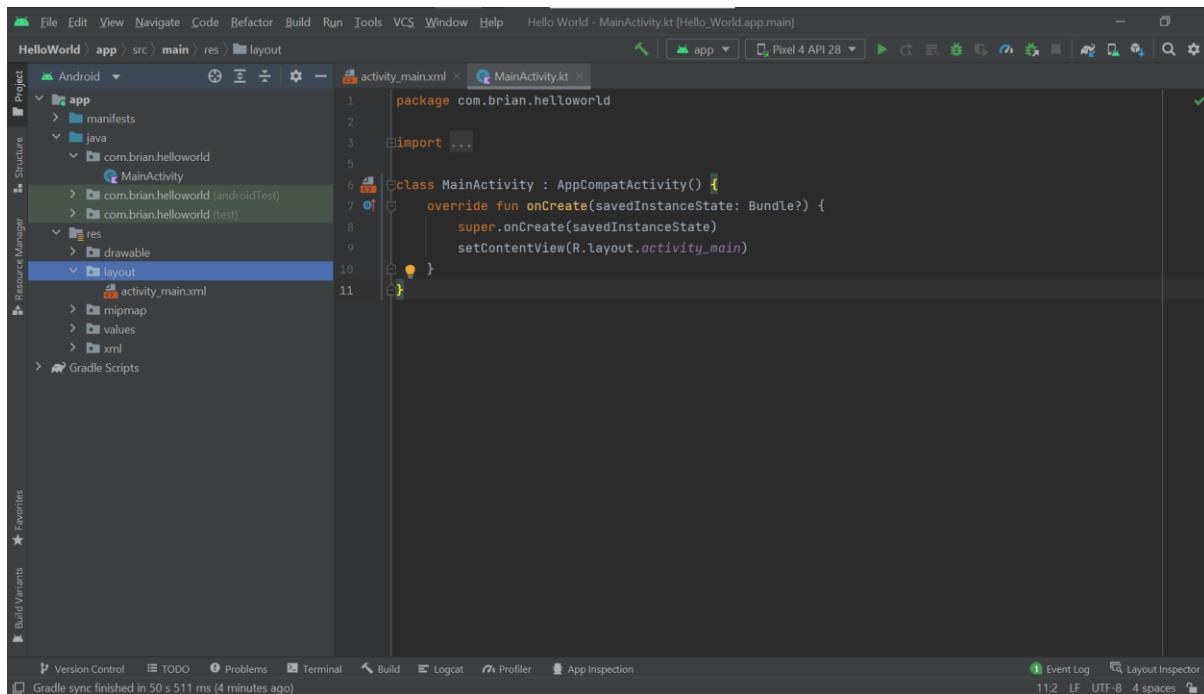
**Package Name** adalah nama dari package yang akan dibuat, biasanya diisikan dengan nama dari domain website perusahaan kita.

**Project Location** adalah nama yang nantinya digunakan lokasi menyimpan data di memori internal ketika aplikasi dijalankan di perangkat Android.

**Language** adalah bahasa pemrograman yang dapat digunakan dalam membuat projek aplikasi android. Disini terdapat pilihan bahasa pemrograman Java dan Kotlin.

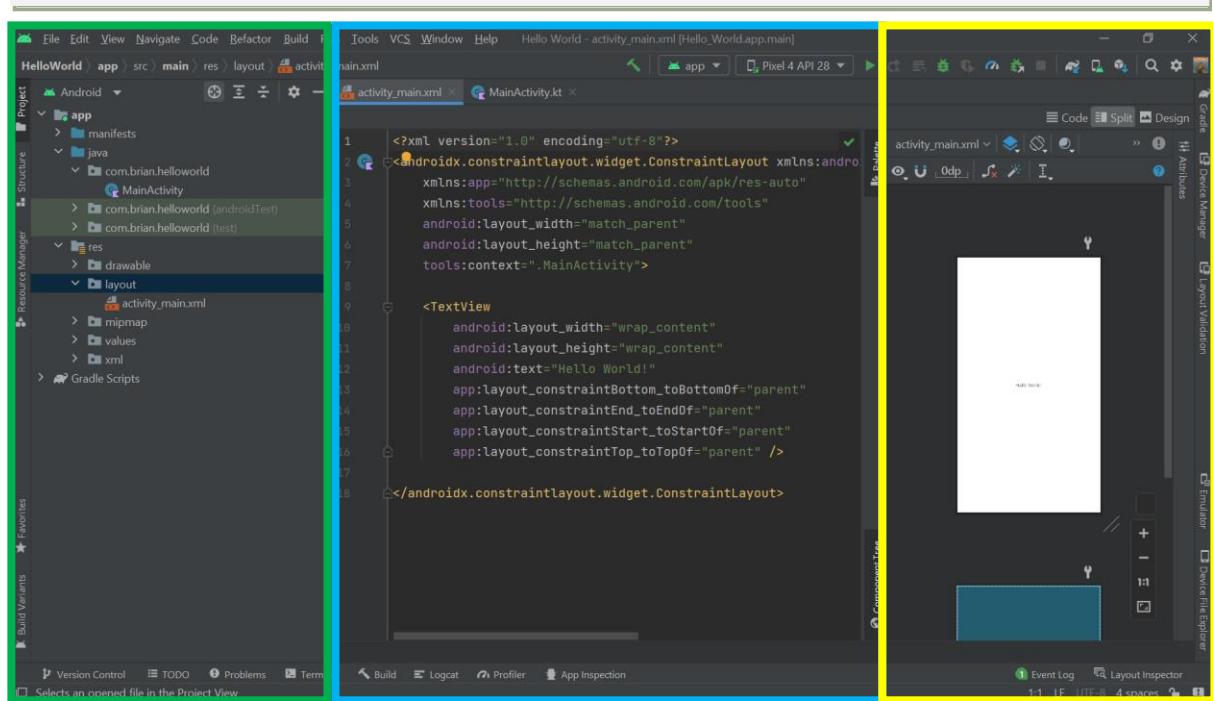
**Minimum SDK** merupakan versi minimal dari device android yang akan menjalankan aplikasi android yang dibuat. Disarankan untuk minimal menggunakan API 26: Android 8.0 (Oreo) agar perbedaan requirements dengan versi terbaru dari android dapat diminimalisir.

4. Setelah kita klik **Finish**, maka akan muncul tampilan (lembar kerja) seperti pada gambar 4. Disini terlihat bahwa pada projek otomatis dibuat activity default bernama MainActivity.kt dan UI nya dengan nama activity\_main.xml.

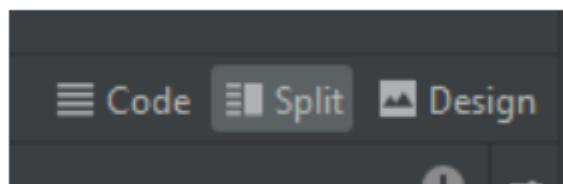


Gambar 4. Tampilan MainActivity.kt

## ANDROID NAVIGATION AND INTERFACE DESIGN 1



Gambar 5. Activity\_main.xml



Gambar 6. Opsi tampilan xml

### NOTE :

Ketika kita klik pada bagian MainActivity.kt, akan muncul tampilan seperti pada gambar 4. MainActivity.kt berisikan code dari fungsi-fungsi yang ada pada aplikasi. Kita juga dapat menambahkan activity baru pada folder yang sama dengan folder tempat MainActivity.kt dibuat. Cara pembuatan activity baru, akan dijelaskan pada GUIDED 2.

Ketika kita klik pada bagian activity\_main.xml, akan muncul tampilan seperti pada gambar 5. activity\_main.xml berisikan code untuk tampilan user interface kepada pengguna. User interface tersebut dapat kita desain sedemikian rupa sesuai dengan kebutuhan dari aplikasi.

Berikut penjelasan tampilan lembar kerja yang ada pada gambar 6, disini digunakan tampilan Split (pilihan tampilan lembar kerja ada di bagian kanan atas projek):

- Pada bagian kiri (kotak hijau) merupakan Jendela Tools dimana terdapat struktur file dari projek kita.
- Pada bagian tengah (kotak biru) merupakan Jendela editor adalah tempat Anda membuat dan memodifikasi kode xml/ kode UI yang dibuat. Bergantung pada jenis file saat ini, editor dapat berubah. Misalnya, ketika melihat file tata letak, editor menampilkan Layout Editor.

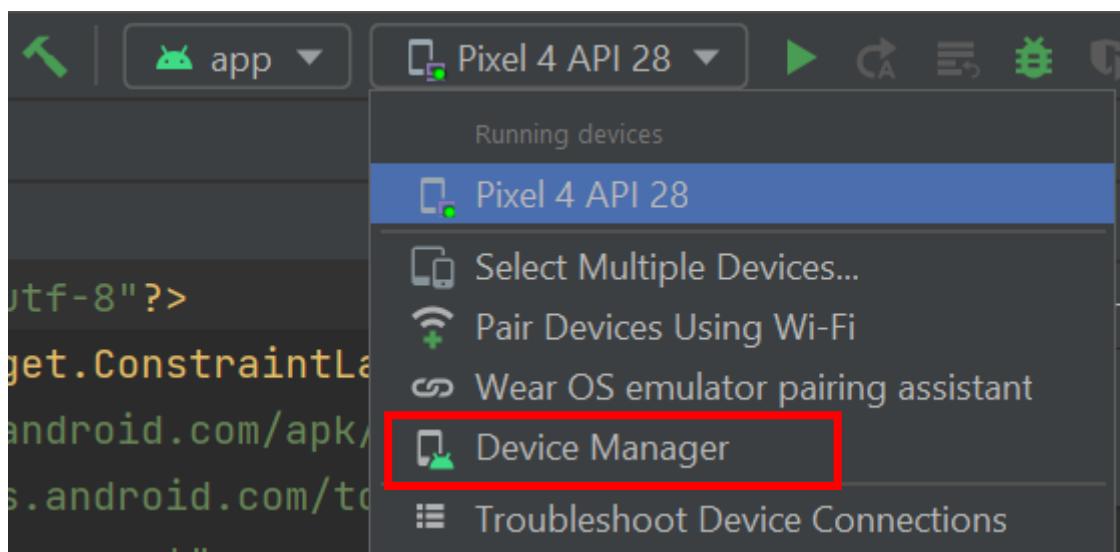
- c. Pada bagian kanan (kotak kuning) merupakan Jendela Preview dimana pada halaman ini hasil design kita disimulasikan.
  
- 5. Selesai. Projek Hello World siap untuk dijalankan dengan menekan tombol seperti pada gambar 7 dan akan menghasilkan tampilan seperti gambar 9 dibawah ini. Untuk melihat hasil projek kita caranya adalah dengan menekan tombol run pada bagian tool bar android studio, kemudian pilih emulator/device yang tersedia.



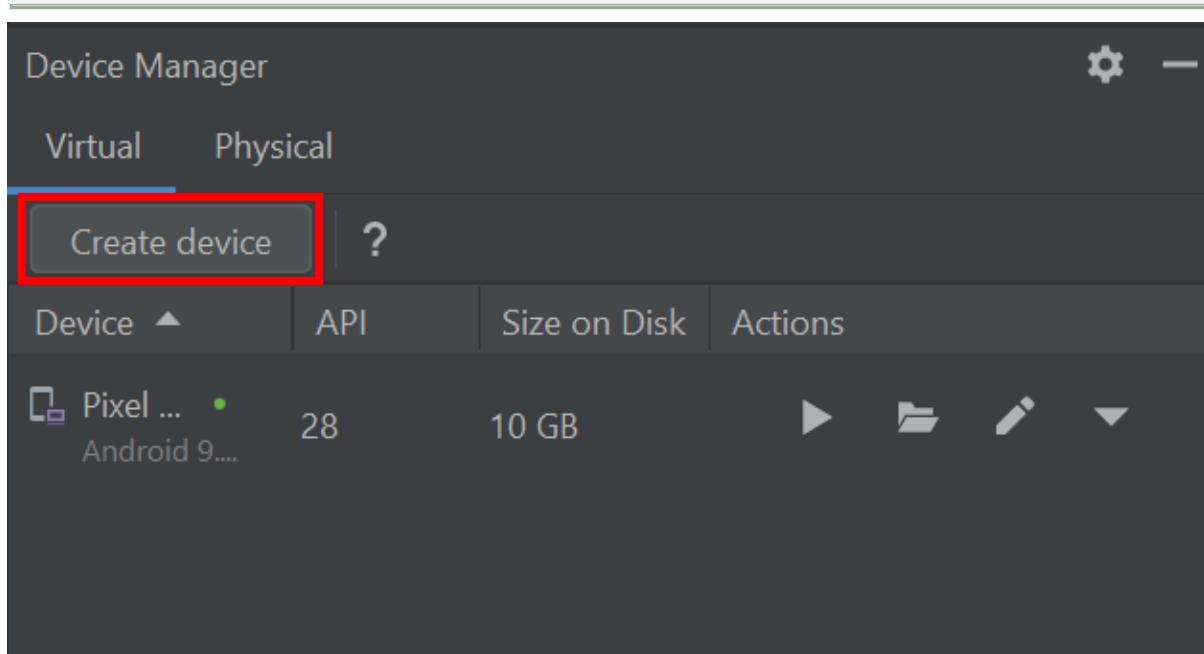
Gambar 7. Tombol Run Aplikasi

**NOTE :**

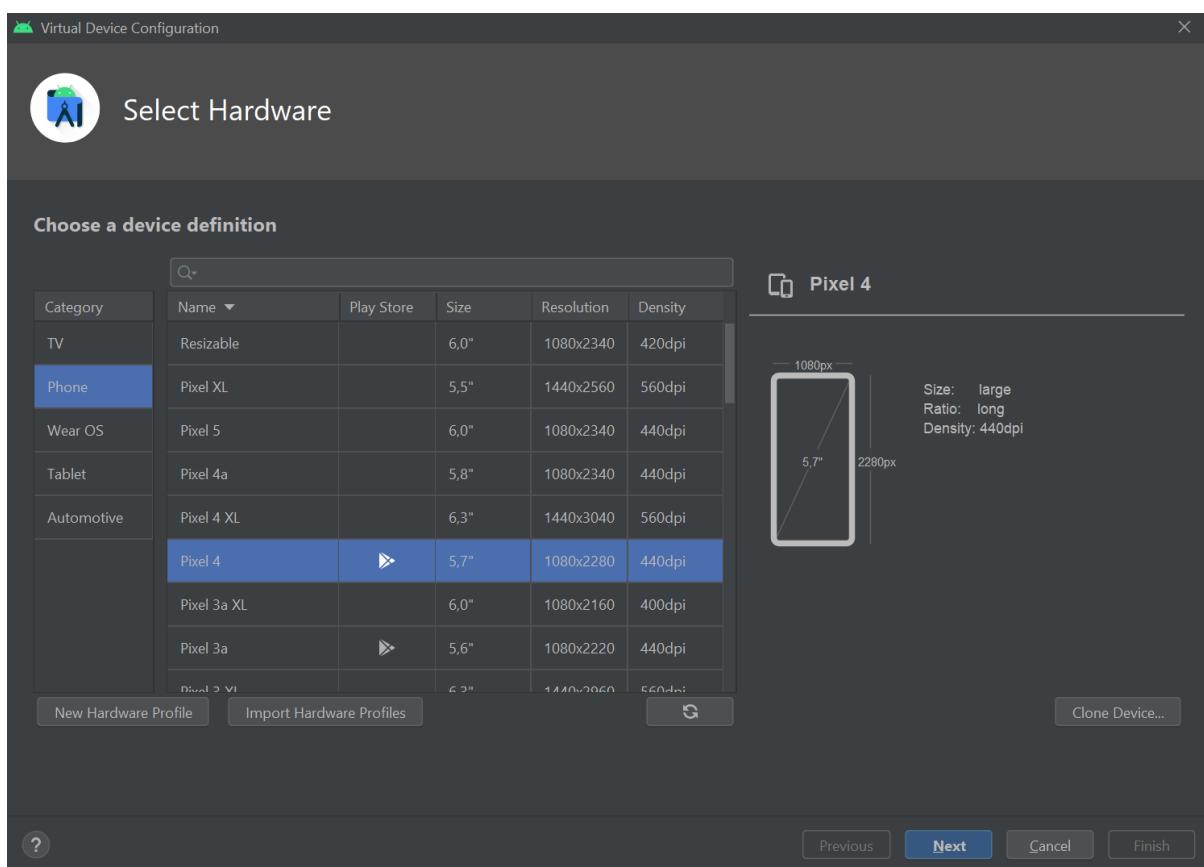
Untuk menjalankan aplikasi, dapat dipilih emulator device yang tersedia pada bagian Open Device Manager seperti yang ditunjukkan pada gambar 8, lalu Create Virtual Device seperti pada gambar 9. Emulator device dapat dipilih sesuai dengan kebutuhan aplikasi seperti gambar 10 dan Minimum SDK yang sudah dipilih pada saat awal pembuatan projek android studio gambar 11. Layar selanjutnya memuat verifikasi data emulator yang akan dibuat. Pada bagian ini dapat diatur nama emulator, orientasi emulator dan device frame (gambar 12). Emulator dapat digunakan ketika sudah didownload dan akan tersedia pada list available devices.



Gambar 8. Klik Device Manager

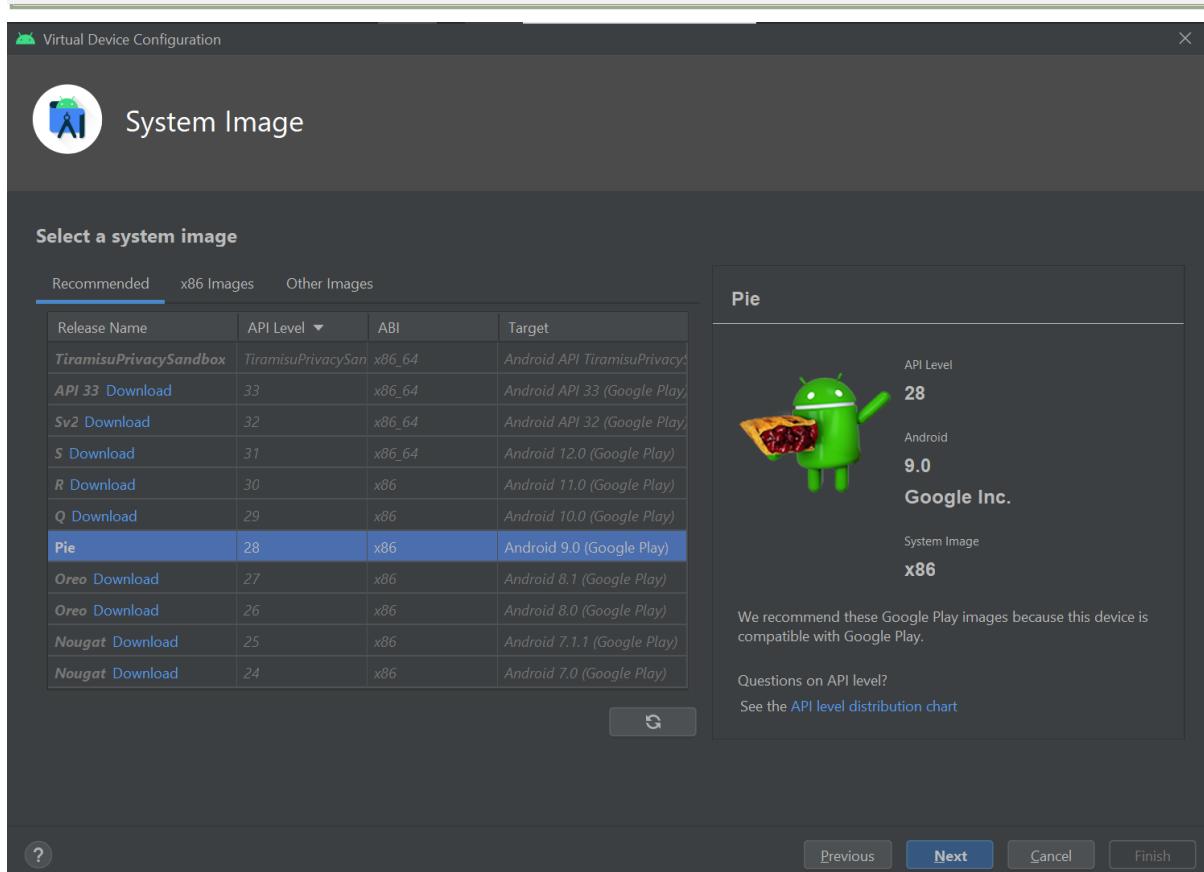


Gambar 9. Klik Create Device

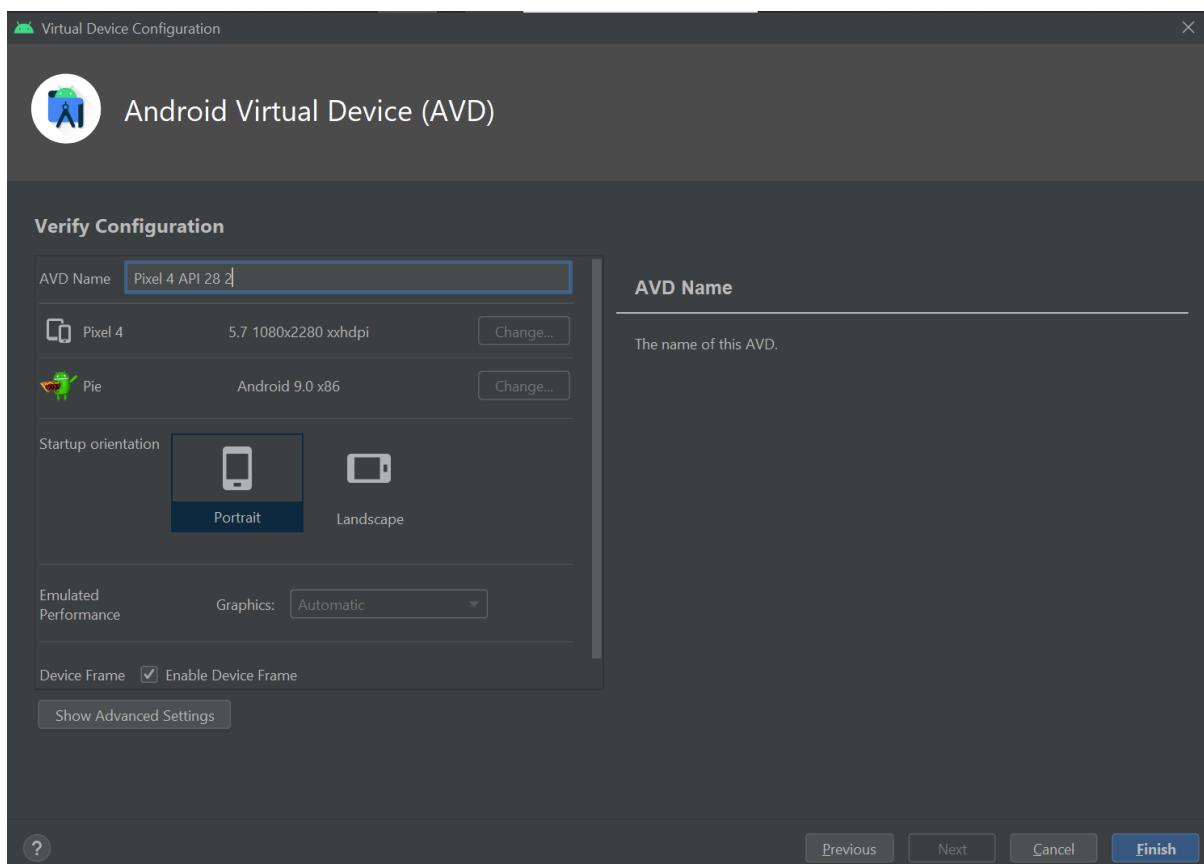


Gambar 10. Pilih hardware emulator

## ANDROID NAVIGATION AND INTERFACE DESIGN 1



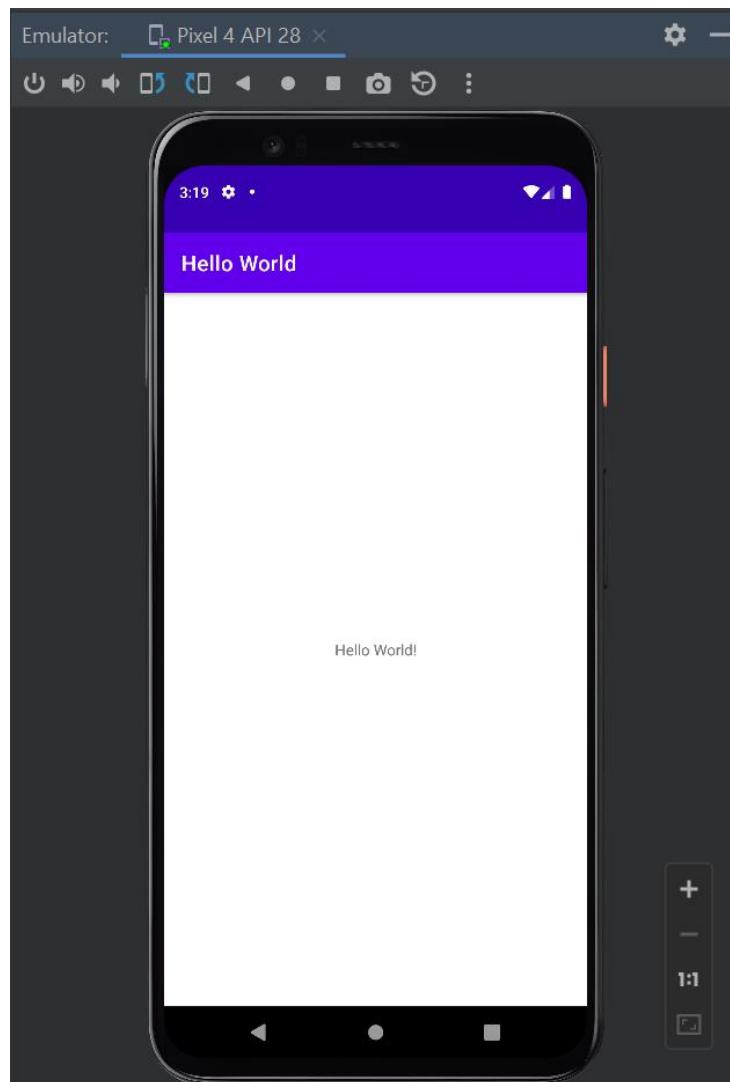
Gambar 11. Pilihlah System Image yang tersedia atau mendownload sesuai dengan kebutuhan



Gambar 12. Verify Configuration AVD

Selain menggunakan emulator, aplikasi juga dapat langsung dijalankan pada device android yang kita miliki dengan menghubungkan device dan laptop menggunakan kabel data. Pada gambar 8, akan terlihat android device yang siap untuk digunakan. Perlu diketahui, jika menggunakan device android (bukan emulator), maka untuk dapat menampilkan layar device pada laptop, diperlukan aplikasi tambahan untuk screen mirroring, seperti A Power Mirror.

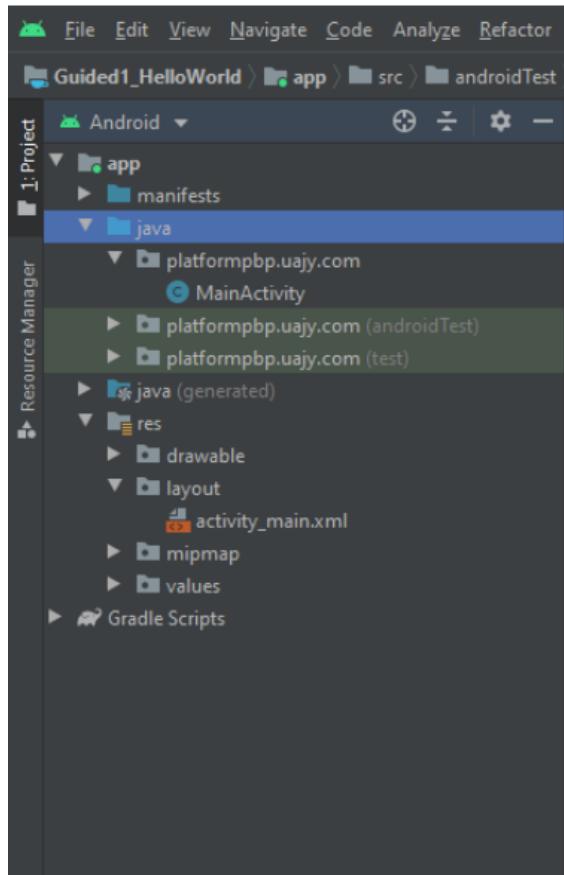
Berikut hasil dari Guided Hello World.



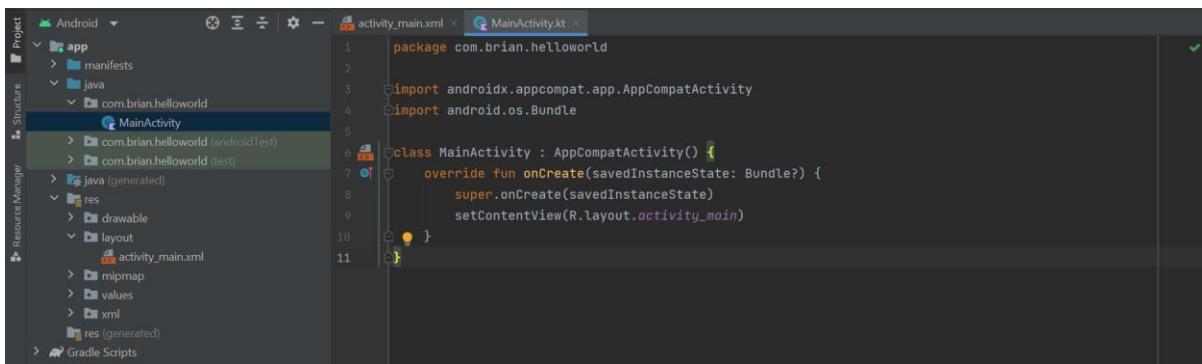
Gambar 13. Hasil GD Hello world

## PEMBAHASAN GUIDED 1

Pada guided 1 di atas, anda telah mencoba untuk membuat sebuah activity dengan menggunakan Android Studio. Berikut susunan folder – folder yang ada pada sebuah projek android studio :



1. Folder manifests yang didalamnya terdapat file `AndroidManifest.xml` yang mencatat semua activity, services, dan permission yang kita buat pada projek aplikasi android.
2. Folder java berisi nama kelas, kelas tersebut bisa berupa activity (yang terhubung dengan UI) maupun kelas yang tidak terhubung dengan UI yang telah anda buat.
3. Folder res (resource) berisi banyak folder, yaitu :
  - a. Drawable yang digunakan untuk menyimpan icon / gambar yang digunakan oleh aplikasi.
  - b. Layout : berisi layout / UI aplikasi.
  - c. Menu : berisi file xml menu yang digunakan pada aplikasi.
  - d. Values : berisi file xml variabel yang digunakan pada aplikasi dan themes.
4. Kemudian, bila kelas activity dibuka, maka akan muncul tampilan code seperti Gambar 14.



Gambar 14

Kemudian, agar aplikasi Hello World bisa dijalankan, maka dilakukan override method `onCreate`. Di dalam method `onCreate`, dilakukan pemanggilan method `setContentView` untuk menampilkan UI layout yang dipilih sehingga ketika aplikasi Hello World dijalankan, UI yang ditampilkan pertama kali adalah UI dari layout yang bernama `activity_main` (ini mengacu pada penamaan file `activity_main.xml` yang ada pada folder `layout`).

## GUIDED 2 – INTENT

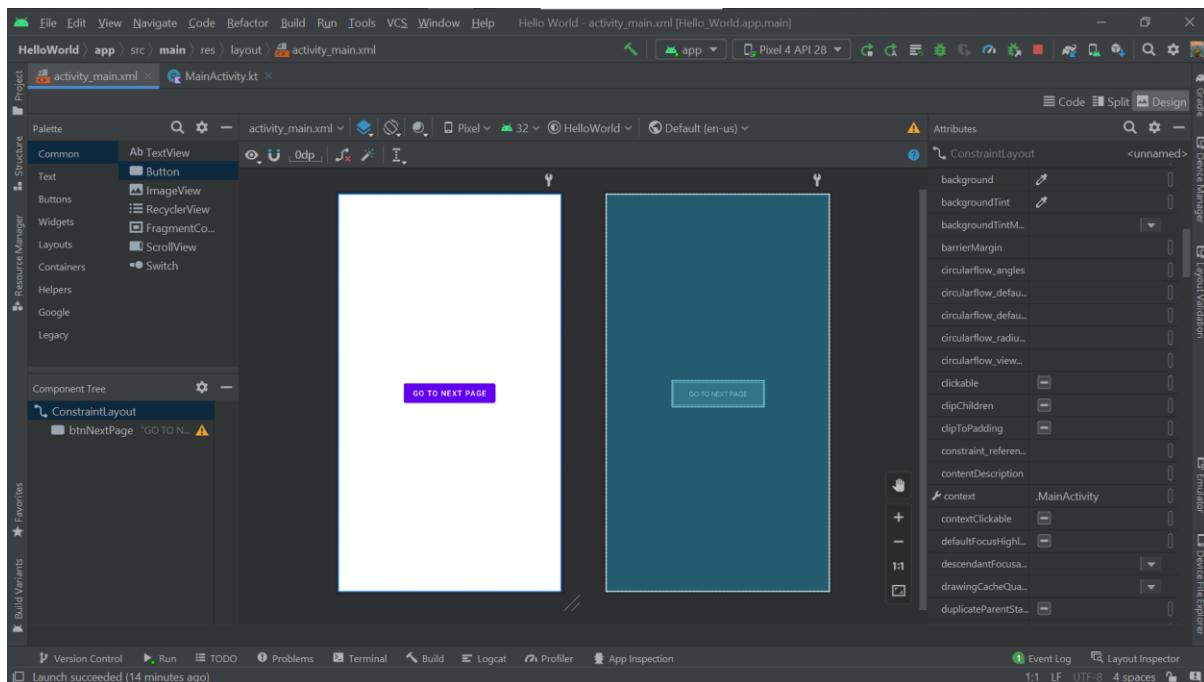
Poin yang dipelajari dalam Guided 2 ini, yaitu :

1. Memahami penggunaan Intent.
2. Memahami cara membuat layout dan class baru.
3. Memahami penggunaan widget dan listener button.
4. Memahami penggunaan import library.

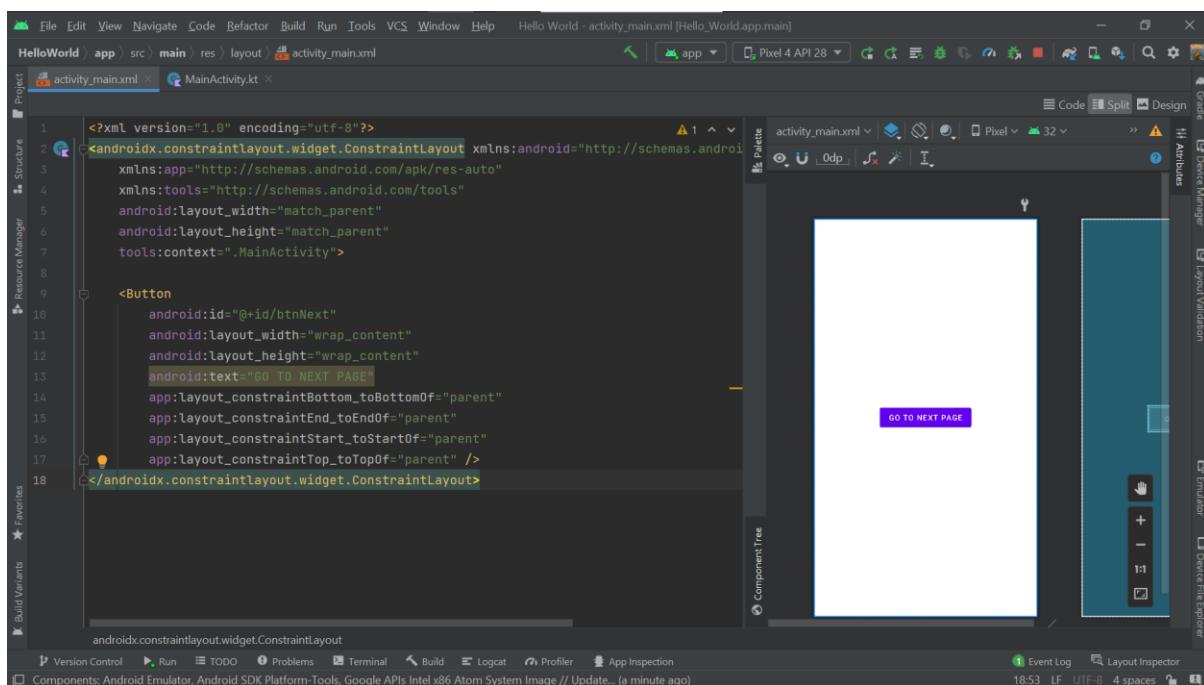
Pada guided ini, anda akan mencoba mmebuat 2 buah activity dengan intent dan 1 widget berupa button. Apabila button di activity pertama di klik, maka akan masuk ke activity kedua dan sebaliknya. Silahkan ikuti langkah – langkah berikut ini :

1. Buat sebuah projek dengan spesifikasi projek sebagai berikut :
  - a. Template : Empty
  - b. Activity Name : Guided2.Intent
  - c. Package Name : (bebas) **usahakan berbeda dengan projek yang lainnya, agar pada device, aplikasi dapat dibedakan (tidak tertumpuk)**
  - d. Language : Kotlin
  - e. Minimum SDK : API 26: Android 8.0 (Oreo)
2. Tambahkan sebuah button dengan id btnNext pada file activity\_main.xml yang terletak di res/layout/activity\_main.xml. Seperti pada guided 1, file activity\_main.xml merupakan UI default aplikasi yang mengacu pada kelas MainActivity.kt. Untuk mengakses xml dari folder layout, klik tab yang terletak di sebelah kiri di tab Project saat projek selesai di buat. Kemudian ketikkan potongan code pada gambar 10.  
**Hint :** Anda dapat menggunakan widget dengan cara melakukan drag and drop pada bagian tab design (gambar 15).

## ANDROID NAVIGATION AND INTERFACE DESIGN 1



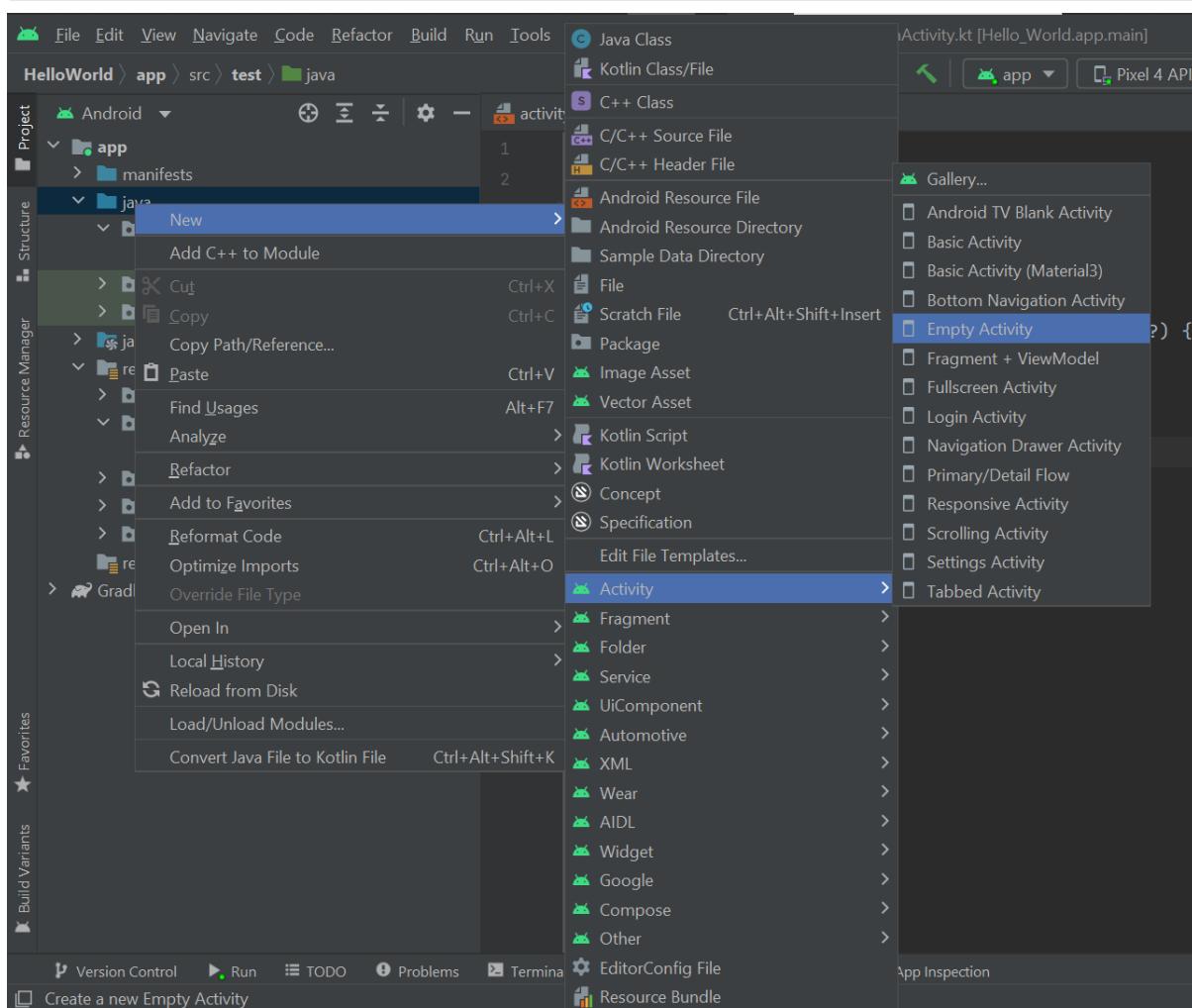
Gambar 15 Tampilan design



Gambar 16. Tampilan Code xml

3. Kemudian, buatlah activity baru dengan cara : klik kanan pada folder java → New → Activity → Empty Activity (gambar 17). Beri nama activity baru tersebut dengan NextActivity. Kemudian klik Finish.

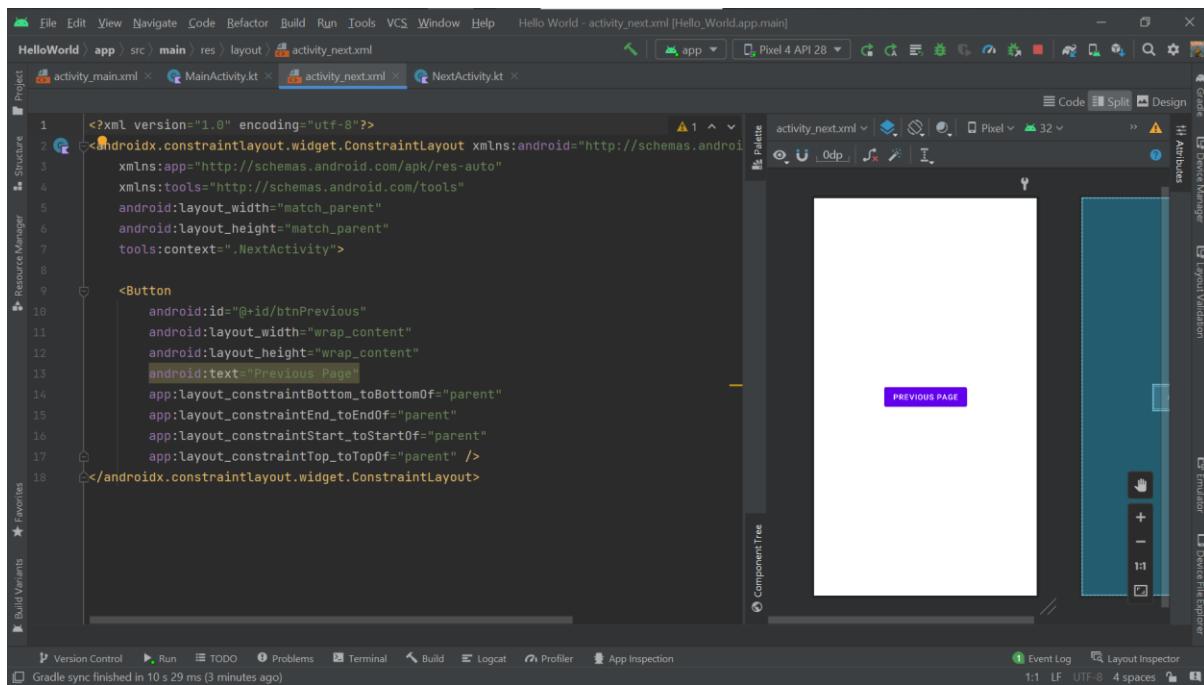
## ANDROID NAVIGATION AND INTERFACE DESIGN 1



Gambar 17. Pembuatan Activity Baru

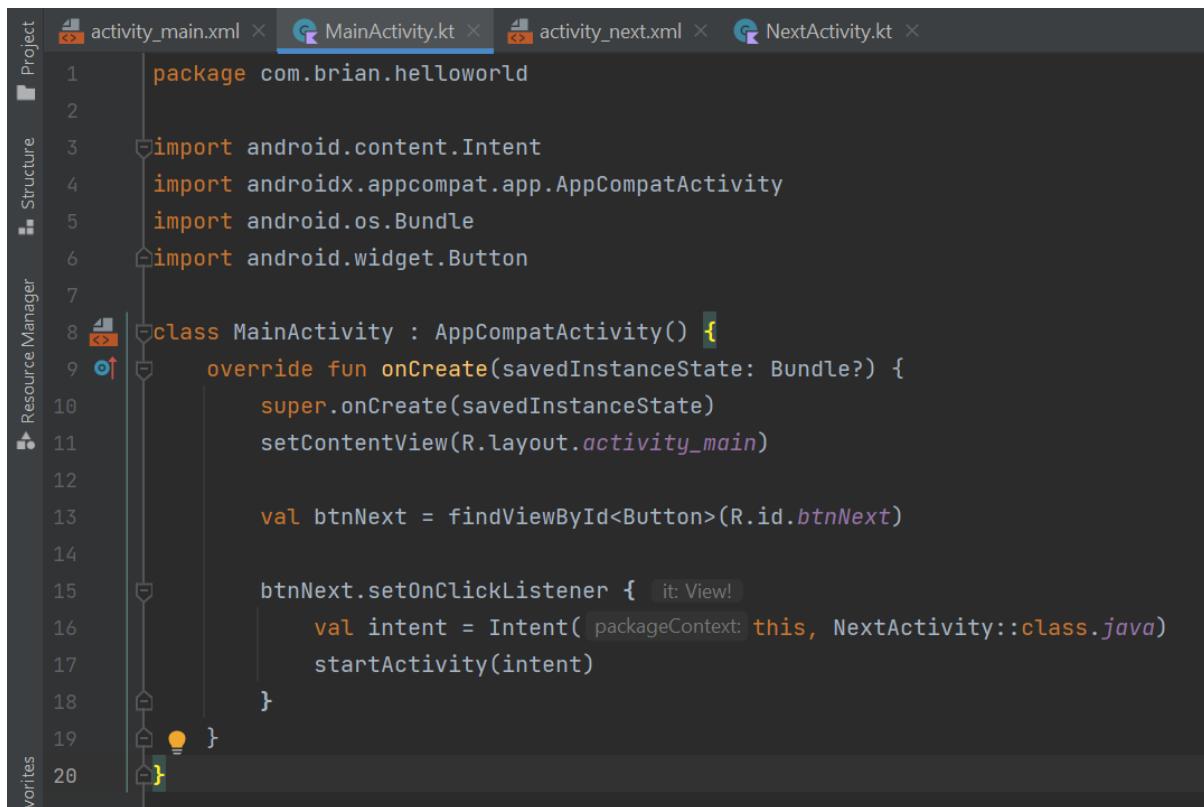
4. Selanjutnya, masuk ke file `activity_next.xml` yang baru saja terbuat dan buat layout atau ketikan potongan code di gambar 18 ini.

## ANDROID NAVIGATION AND INTERFACE DESIGN 1



Gambar 18. Layout untuk Previous Page

5. Buka file MainActivity.kt, kemudian tambahkan code dibawah ini :



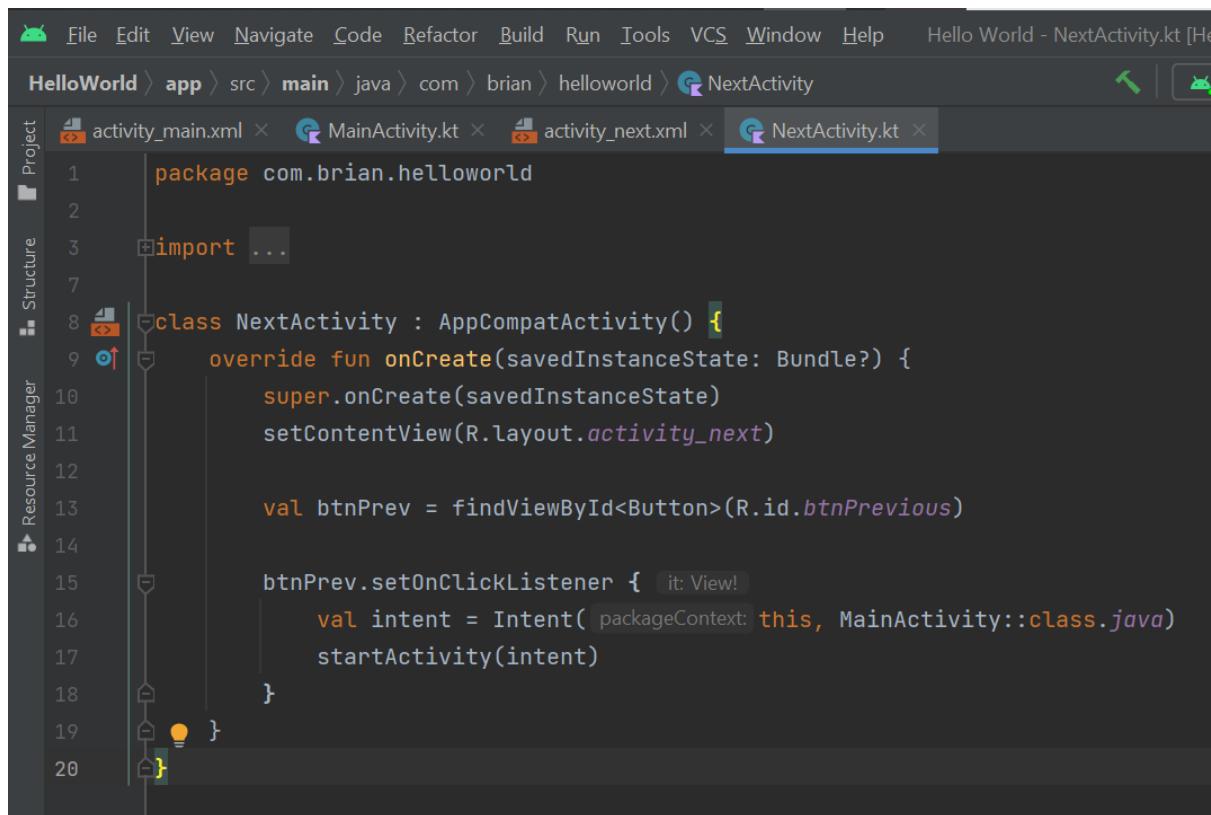
Gambar 19. Code Intent ke activity next

### NOTE (PENJELASAN CODE PADA KELAS ActivityMain.kt) :

Pada code di file ActivityMain.kt terdapat method setContentView(...) yang digunakan untuk mengatur view dari activity tersebut, yaitu layout pertama yang anda buat (activity\_main).

Kemudian, untuk dapat menggunakan button next yang ada di layout tersebut, maka perlu ada pendeklarasian objek button. Objek button dapat dideklarasikan dengan berbagai cara, salah satunya seperti pada code pada gambar 19. Fungsi dari findViewById adalah untuk mengarahkan variabel btnNext dengan button next yang ada pada layout xml, dengan cara memanggil id dari button Next. Kemudian agar button bisa diklik, maka digunakan method setOnClickListener(...). Method ini berfungsi sebagai listener dari event click yang dihasilkan oleh button. Ketika mendeklarasikan method ini, anda harus mengimplementasikan "abstract method" yang dimiliki, yaitu method onClick(...). Di dalam method onClick(...) ada deklarasi intent yang mengacu pada class ActivityMain.kt (this) dan NextActivity.kt. Code startActivityForResult (intent); digunakan untuk mengaktifkan intent dan mengarahkan ke activity kelas yang dipanggil.

#### 6. Kemudian, pada kelas NextActivity.kt tambahkan code berikut :



```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Hello World - NextActivity.kt [Help]
HelloWorld > app > src > main > java > com > brian > helloworld > NextActivity
Project activity_main.xml < MainActivity.kt < activity_next.xml < NextActivity.kt
1 package com.brian.helloworld
2
3 import ...
4
5 class NextActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_next)
9
10        val btnPrev = findViewById<Button>(R.id.btnPrevious)
11
12        btnPrev.setOnClickListener { it: View! -
13            val intent = Intent( packageContext: this, MainActivity::class.java)
14            startActivity(intent)
15        }
16    }
17
18 }
19
20

```

Gambar 20. Code Intent ke Activity Main

#### 7. Kemudian buka file AndroidManifest.xml, pilih folder manifest pada bagian kiri projek anda. Lalu cek apakah setiap activity yang sudah dibuat sudah terdaftar seperti code pada gambar 21:

## ANDROID NAVIGATION AND INTERFACE DESIGN 1

The screenshot shows the Android Studio interface with the project 'HelloWorld' open. The left sidebar displays the project structure, including the 'app' module with its manifest files. The main editor window shows the XML code for the manifest:

```
<application>
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="Hello World"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.HelloWorld"
    tools:targetApi="31">
        <activity
            android:name=".NextActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

The code highlights the `android:exported="true"` attribute for the `MainActivity`.

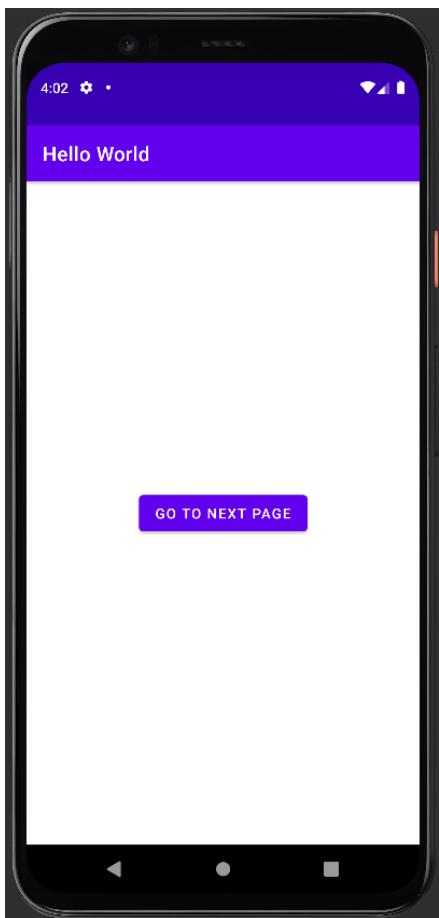
Gambar 21. Android Manifest

**NOTE :** `MainActivity` merupakan class yang akan ditampilkan pertama kali saat aplikasi dijalankan. Hal tersebut disebabkan karena class `MainActivity` diberi code.

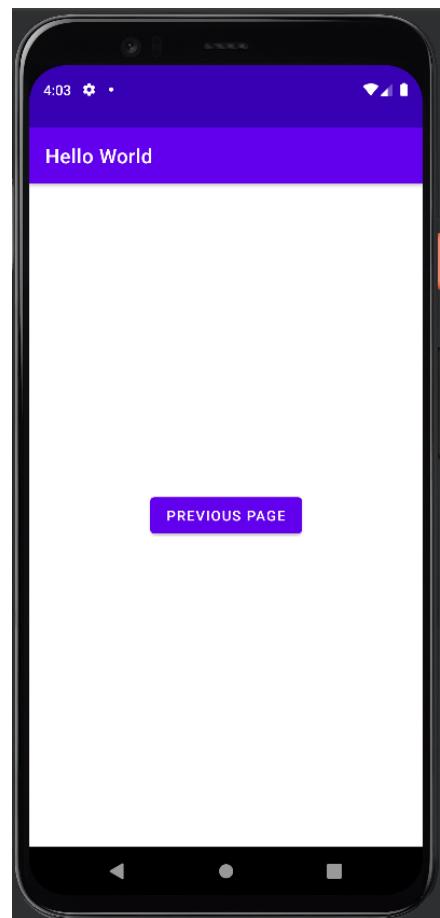
### PENTING!!

- **Pastikan semua activity yang ada terdaftar pada android manifest.**
- **Activity yang terdaftar pada merupakan activity yang pertama kali dijalankan.**

8. Selesai. Jalankan projek dengan langkah seperti yang telah dijelaskan pada guided 1. Maka akan muncul tampilan sebagai berikut:



Gambar 22. Main page



Gambar 23. Next Page

NOTE :

Gambar 22 → Tampilan Awal dan tampilan ketika button Previous Page di klik.

Gambar 23 → Tampilan setelah button Go To Next Page di klik.

## GUIDED 3 – STRING EXTRA

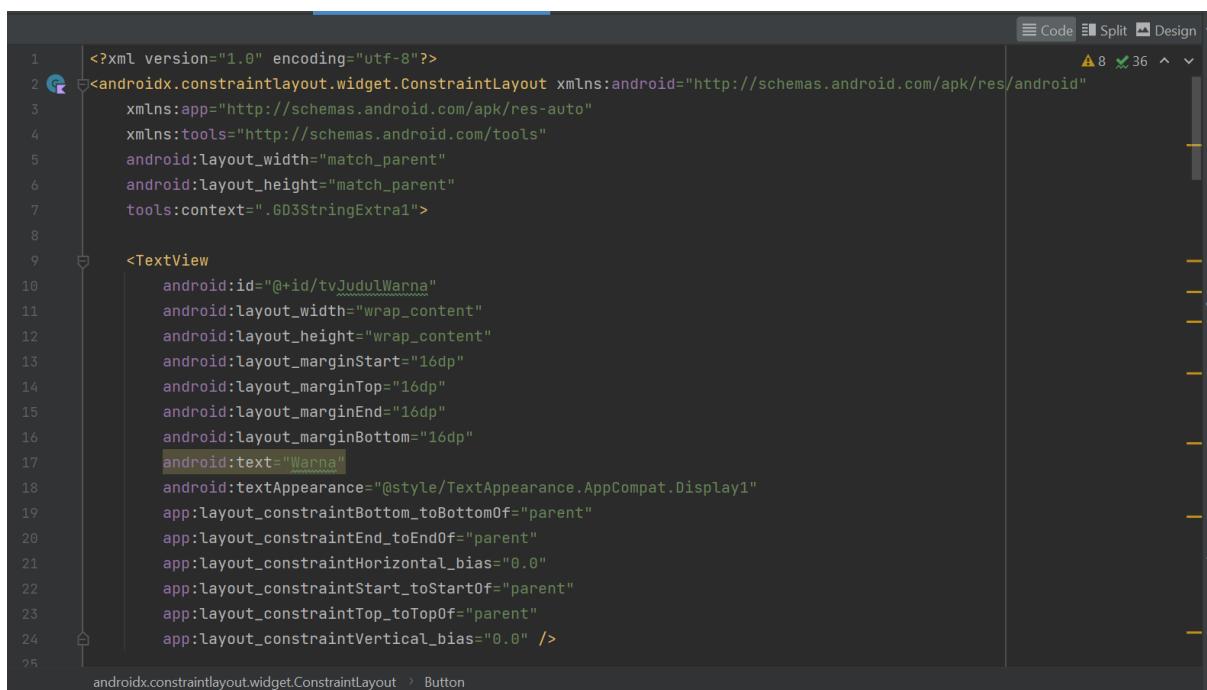
Poin yang dipelajari dalam Guided 3 ini, yaitu:

1. Memahami penggunaan intent dan pengiriman pesan atau message melalui intent.
2. Memahami cara mendaftarkan activity di AndroidManifest.xml (secara manual).

Pada guided ini, Anda akan mencoba membuat 3 buah activity dengan Intent dan mengirimkan message melalui Intent. Silahkan ikuti langkah – langkah berikut ini :

1. Buat sebuah projek dengan spesifikasi projek sebagai berikut :
  - a. Template : Empty
  - b. Activity Name : Guided3\_StringExtra
  - c. Package Name : (bebas) usahakan berbeda dengan projek yang lainnya, agar pada device, aplikasi dapat dibedakan (tidak tertumpuk)
  - d. Language : Kotlin
  - e. Minimum SDK : API 26: Android 8.0 (Oreo)
2. Kemudian, tambahkan 2 buah textView, 1 buah radio button group, 1 buah textfield, dan 2 buah button pada layout activity\_main.xml. Button pertama akan mengirimkan message melalui intent kepada activity 2 dan button kedua akan mengirimkan message melalui intent kepada activity
3. Ketikan sesuai dengan potongan code di bawah ini atau buatlah layout seperti gambar 24 :

**Hint :** untuk mempercepat pengetikan code, Anda cukup mengetik tulisan berwarna putih untuk setiap komponen dari widget yang ada. Misalkan untuk code “android:layout\_width=”wrap\_content” Anda cukup mengetikkan layout\_width lalu Enter, maka text akan secara otomatis muncul pada activity\_main.xml.



```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".GD3StringExtra1">
8
9     <TextView
10        android:id="@+id/tvJudulWarna"
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        android:layout_marginStart="16dp"
14        android:layout_marginTop="16dp"
15        android:layout_marginEnd="16dp"
16        android:layout_marginBottom="16dp"
17        android:text="Warna"
18        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
19        app:layout_constraintBottom_toBottomOf="parent"
20        app:layout_constraintEnd_toEndOf="parent"
21        app:layout_constraintHorizontal_bias="0.0"
22        app:layout_constraintStart_toStartOf="parent"
23        app:layout_constraintTop_toTopOf="parent"
24        app:layout_constraintVertical_bias="0.0" />
25

```

## ANDROID NAVIGATION AND INTERFACE DESIGN 1

```
26 <RadioGroup  
27     android:id="@+id/radioGroup"  
28     android:layout_width="0dp"  
29     android:layout_height="wrap_content"  
30     android:layout_marginStart="16dp"  
31     android:layout_marginTop="16dp"  
32     android:layout_marginEnd="16dp"  
33     android:layout_marginBottom="16dp"  
34     app:layout_constraintBottom_toBottomOf="parent"  
35     app:layout_constraintEnd_toEndOf="parent"  
36     app:layout_constraintStart_toStartOf="parent"  
37     app:layout_constraintTop_toBottomOf="@+id/tvJudulWarna"  
38     app:layout_constraintVertical_bias="0.0" >  
39  
40     <RadioButton  
41         android:id="@+id/rbMerah"  
42         android:layout_width="match_parent"  
43         android:layout_height="wrap_content"  
44         android:text="Merah"  
45         android:textColor="#FF0000" />  
46
```

```
47     <RadioButton  
48         android:id="@+id/rbHijau"  
49         android:layout_width="match_parent"  
50         android:layout_height="wrap_content"  
51         android:text="Hijau"  
52         android:textColor="#00FF00" />  
53  
54     <RadioButton  
55         android:id="@+id/rbBiru"  
56         android:layout_width="match_parent"  
57         android:layout_height="wrap_content"  
58         android:text="Biru"  
59         android:textColor="#0000FF" />  
60     </RadioGroup>  
61  
62     <Button  
63         android:id="@+id/btnPilihWarna"  
64         android:layout_width="0dp"  
65         android:layout_height="wrap_content"  
66         android:layout_marginStart="16dp"  
67         android:layout_marginTop="16dp"  
68         android:layout_marginEnd="16dp"  
69         android:layout_marginBottom="16dp"  
70         android:text="PILIH WARNA"  
71         android:onClick="sendMessage" />
```

```
71         android:onClick="sendMessage"  
72         app:layout_constraintBottom_toBottomOf="parent"  
73         app:layout_constraintEnd_toEndOf="parent"  
74         app:layout_constraintStart_toStartOf="parent"  
75         app:layout_constraintTop_toBottomOf="@+id/radioGroup"  
76         app:layout_constraintVertical_bias="0.0" />  
77  
78     <TextView  
79         android:id="@+id/tvJudulAngka"  
80         android:layout_width="wrap_content"  
81         android:layout_height="wrap_content"  
82         android:layout_marginStart="16dp"  
83         android:layout_marginTop="32dp"  
84         android:layout_marginEnd="16dp"  
85         android:layout_marginBottom="16dp"  
86         android:text="Ketik Angka"  
87         android:textAppearance="@style/TextAppearance.AppCompat.Display1"  
88         app:layout_constraintBottom_toBottomOf="parent"  
89         app:layout_constraintEnd_toEndOf="parent"  
90         app:layout_constraintHorizontal_bias="0.0"  
91         app:layout_constraintStart_toStartOf="parent"  
92         app:layout_constraintTop_toBottomOf="@+id/btnPilihWarna"  
93         app:layout_constraintVertical_bias="0.0" />  
94
```

## ANDROID NAVIGATION AND INTERFACE DESIGN 1

The screenshot shows the Android Studio code editor with two tabs open, both displaying XML layout code for an Android application.

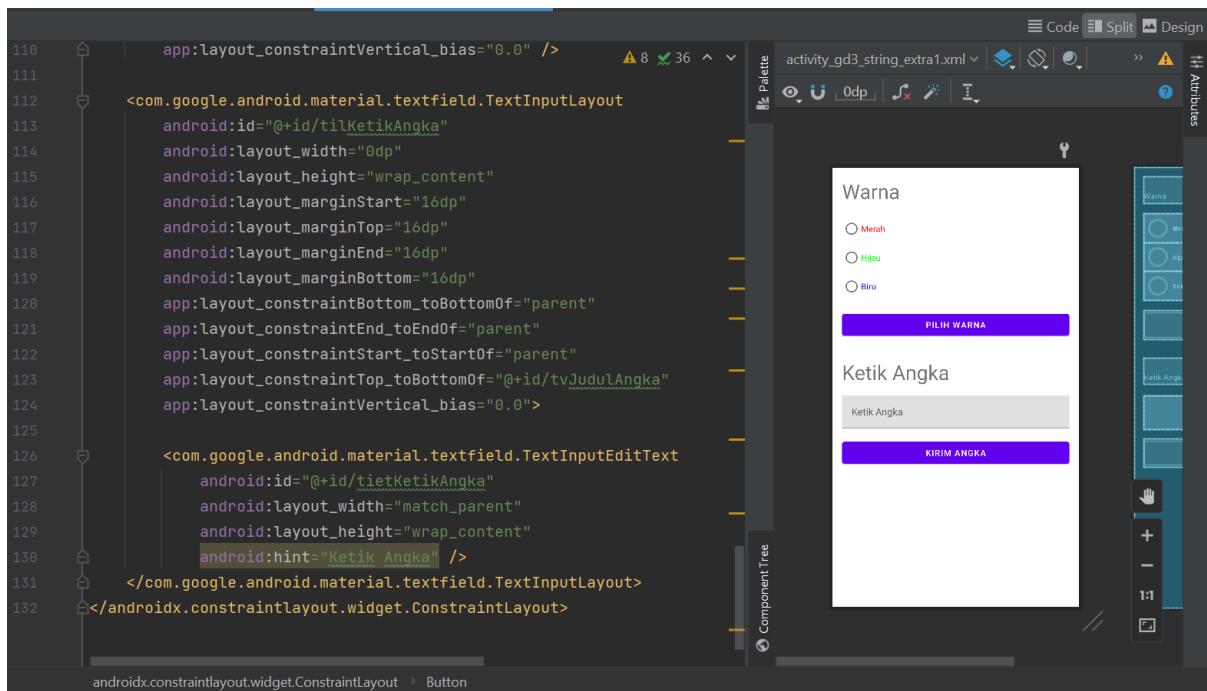
**Top Tab (Line 95 to 111):**

```
95     <Button  
96         android:id="@+id/btnKirimAngka"  
97         android:layout_width="0dp"  
98         android:layout_height="wrap_content"  
99         android:layout_marginStart="16dp"  
100        android:layout_marginTop="16dp"  
101        android:layout_marginEnd="16dp"  
102        android:layout_marginBottom="16dp"  
103        android:text="Kirim Angka"  
104        android:onClick="sendNumberMessage"  
105        app:layout_constraintBottom_toBottomOf="parent"  
106        app:layout_constraintEnd_toEndOf="parent"  
107        app:layout_constraintHorizontal_bias="1.0"  
108        app:layout_constraintStart_toStartOf="parent"  
109        app:layout_constraintTop_toBottomOf="@+id/tilKetikAngka"  
110        app:layout_constraintVertical_bias="0.0" />  
111  
112     <com.google.android.material.textfield.TextInputLayout  
113         android:id="@+id/tilKetikAngka"  
114         android:layout_width="0dp"  
115         android:layout_height="wrap_content"  
116         android:layout_marginStart="16dp"  
117         android:layout_marginTop="16dp"  
118         android:layout_marginEnd="16dp"  
119         android:layout_marginBottom="16dp"  
120     androidx.constraintlayout.widget.ConstraintLayout > Button  
121  
122         app:layout_constraintVertical_bias="0.0" />
```

**Bottom Tab (Line 110 to 132):**

```
110     <com.google.android.material.textfield.TextInputLayout  
111         android:id="@+id/tilKetikAngka"  
112         android:layout_width="0dp"  
113         android:layout_height="wrap_content"  
114         android:layout_marginStart="16dp"  
115         android:layout_marginTop="16dp"  
116         android:layout_marginEnd="16dp"  
117         android:layout_marginBottom="16dp"  
118         app:layout_constraintBottom_toBottomOf="parent"  
119         app:layout_constraintEnd_toEndOf="parent"  
120         app:layout_constraintStart_toStartOf="parent"  
121         app:layout_constraintTop_toBottomOf="@+id/tvJudulAngka"  
122         app:layout_constraintVertical_bias="0.0" >  
123  
124     <com.google.android.material.textfield.TextInputEditText  
125         android:id="@+id/tietKetikAngka"  
126         android:layout_width="match_parent"  
127         android:layout_height="wrap_content"  
128         android:hint="Ketik Angka" />  
129     </com.google.android.material.textfield.TextInputLayout>  
130  
131 </androidx.constraintlayout.widget.ConstraintLayout>  
132  
133 androidx.constraintlayout.widget.ConstraintLayout > Button
```

## ANDROID NAVIGATION AND INTERFACE DESIGN 1



Gambar 24. Layout GD 3

4. Langkah selanjutnya, buatlah sebuah class dengan nama `DisplayMessage.kt`, kemudian tambahkan code seperti ini :

```

activity_main.xml × MainActivity.kt × DisplayMessage.kt × DisplayNumber.kt ×
1 package com.brian.helloworld
2
3 import ...
4
5
6
7
8
9 class DisplayMessage : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12
13         / val intent: Intent = intent .../
14     }
15 }
```

Gambar 25. DisplayMessage.kt

**\*Code “/ val intent: Intent = intent .../” tidak perlu diketikkan dulu**

**Hint :** untuk mempercepat pengetikan code untuk method `onCreate`, dapat menggunakan shortcut Alt+Insert, lalu pilih `OverrideMethods...`

5. Lalu, buatlah sebuah class lagi dengan nama `DisplayNumber.kt`, kemudian tambahkan code seperti ini :

```

1 package com.brian.helloworld
2
3 import ...
4
5 class DisplayNumber : AppCompatActivity() {
6
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9
10        / val intent: Intent = intent .../
11    }
12
13 }

```

Gambar 26. DisplayNumber.kt

\*Code “/ val intent: Intent = intent .../” tidak perlu diketikkan dulu

- Setelah itu, modifikasi kelas MainActivity.kt dengan menambahkan potongan code sebagai berikut: \*GD3StringExtra1.kt adalah MainActivity.kt

```

1 package com.brian.helloworld
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.view.View
6 import android.widget.Button
7 import android.widget.RadioButton
8 import android.widget.RadioGroup
9 import androidx.appcompat.app.AppCompatActivity
10 import com.google.android.material.textfield.TextInputEditText
11
12 class GD3StringExtra1 : AppCompatActivity() {
13
14     private lateinit var etAngka : TextInputEditText
15     private var btnRGrup : RadioGroup? = null
16     private lateinit var btnRButton : RadioButton
17
18     override fun onCreate(savedInstanceState: Bundle?) {
19         super.onCreate(savedInstanceState)
20         setContentView(R.layout.activity_gd3_string_extra1)
21
22     }
23
24     fun sendMessage(view: View){
25         val intent = Intent( packageContext: this, DisplayMessage::class.java)
26         val message: String
27         btnRGrup = findViewById(R.id.radioGroup)

```

```

25     val intent = Intent( packageName: this, DisplayMessage::class.java)
26     val message: String
27     btnRGrup = findViewById(R.id.radioGroup)
28     val rbc: Int = btnRGrup!!.checkedRadioButtonId
29     btnRButton = findViewById(rbc)
30     message = btnRButton.text.toString()
31
32     intent.putExtra(EXTRA_MESSAGE, message)
33     startActivity(intent)
34 }
35
36 fun sendNumberMessage(view: View){
37     val intent = Intent( packageName: this, DisplayNumber::class.java)
38     etAngka = findViewById(R.id.tietKetikAngka)
39     val angka: Int = (etAngka.text.toString()).toInt()
40
41     intent.putExtra( name: "inputanAngka", angka.toString())
42     startActivity(intent)
43 }
44
45 //An object declaration inside a class can be marked with the companion keyword
46 companion object {
47     const val EXTRA_MESSAGE = "MESSAGE"
48 }
49

```

#### **NOTE (PENJELASAN CODE PADA KELAS MainActivity.kt):**

EXTRA\_MESSAGE merupakan key yang digunakan sebagai keyword untuk menambahkan message yang akan dikirim melalui intent pada method sendMessage. Method sendMessage merupakan method untuk menjalankan Intent dan memanggil kelas DisplayMessage.kt. di dalam method terdapat code intent.putExtra(EXTRA\_MESSAGE, message); yang digunakan untuk menambahkan data yang akan dikirim ke activity DisplayMessage.kt. Dalam hal ini data adalah isi dari variabel message tersebut. Method sendNumberMessage merupakan method untuk menjalankan Intent dan memanggil kelas DisplayNumber.kt. di dalam method terdapat code intent.putExtra("inputanAngka", String.valueOf(angka)); yang digunakan untuk menambahkan data yang akan dikirim ke activity DisplayNumber.kt. Dalam hal ini digunakan pesan "inputanAngka" untuk mengirimkan nilai dari variabel angka. Terdapat sedikit perbedaan dengan method sendMessage, dikarenakan pada method sendNumberMessage, pesan yang dikirim berupa angka integer bukan string.

7. Kemudian, modifikasi code pada override method onCreate(...) pada kelas DisplayMessage.kt. Code pada kelas tersebut menjadi sebagai berikut :

```

1 package com.brian.helloworld
2
3 import ...
4
5 class DisplayMessage : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8
9         val intent: Intent = intent
10        val message: String = intent.getStringExtra(GD3StringExtra1.EXTRA_MESSAGE).toString()
11
12        //membuat textView
13        val textView = TextView( context: this)
14        textView.setTextSize(40F)
15        textView.setText("Anda memilih warna " + message)
16
17        if(message=="Merah"){
18            textView.setTextColor(Color.parseColor( colorString: "#FF0000"))
19        } else if (message=="Hijau"){
20            textView.setTextColor(Color.parseColor( colorString: "#00FF00"))
21        } else {
22            textView.setTextColor(Color.parseColor( colorString: "#00FFFF"))
23        }
24        setContentView(textView)
25    }
26
27 }
28
29
30

```

8. Lalu, modifikasi code pada override method `onCreate(...)` pada kelas `DisplayNumber.kt`.  
Code pada kelas tersebut menjadi sebagai berikut :

```

1 package com.brian.helloworld
2
3 import ...
4
5 class DisplayNumber : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8
9         val intent: Intent = intent
10        val message: String? = intent.getStringExtra( name: "inputanAngka")
11
12        val textView = TextView( context: this)
13        textView.setTextSize(40F)
14        textView.setText("Angka yang anda inputkan adalah "+message)
15
16        setContentView(textView)
17
18    }
19
20
21
22
23

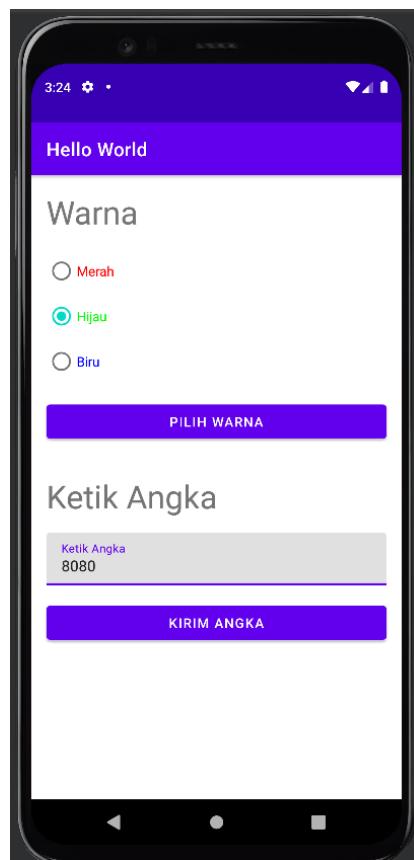
```

9. Setelah itu daftarkan kelas Activity `DisplayMessage.kt` dan `DisplayNumber.kt` pada `AndroidManifest.xml` sebagai berikut :

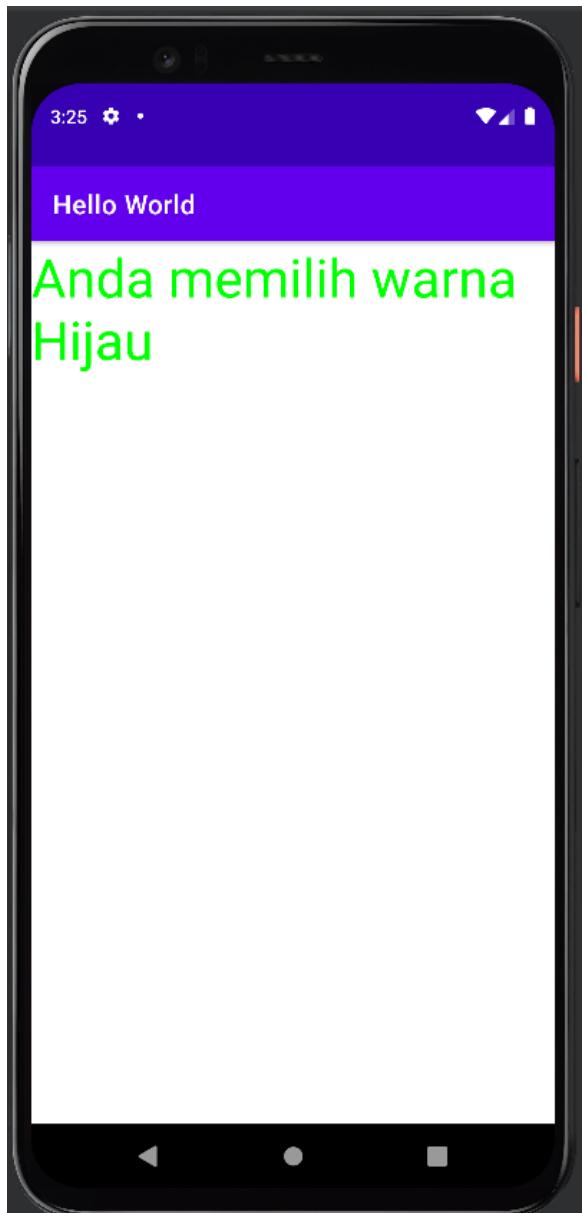
```
<activity
    android:name=".DisplayMessage"
    android:exported="false"
    android:label="Hello World" />
<activity
    android:name=".DisplayNumber"
    android:exported="false"
    android:label="Hello World" />
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

10. impan dan jalankan projek. Cara untuk menjalankan projek sudah dijelaskan di guided sebelumnya.



Gambar 27. Tampilan awal project, dengan pilihan warna hijau dan angka 8080



Gambar 28. Hasil button pilih warna



Gambar 29. Hasil button kirim angka

## GUIDED 4 – BUNDLE

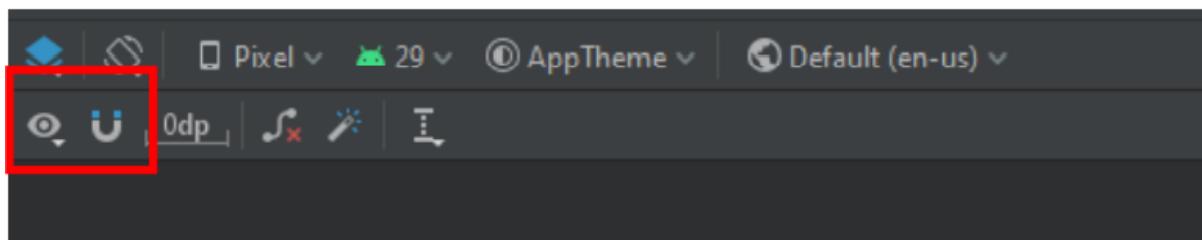
Poin yang dipelajari dalam Guided 4 ini, yaitu :

1. Memahami penggunaan bundle dan pengiriman pesan atau message melalui bundle.

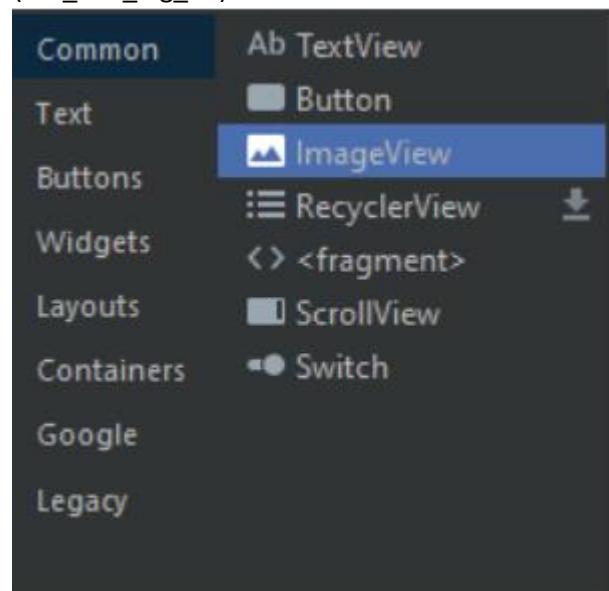
Bundle pada umumnya digunakan sebagai alat bantu untuk melakukan parsing data (memindahkan data) dari satu activity ke activity lain, biasanya digunakan untuk memindahkan beberapa variable dalam satu activity sekaligus. Contohnya adalah saat memindahkan data profil diri dari activity input data diri ke activity tampil data diri.

Sebelum mencoba membuat guided ini, berikut sekilas tentang constraint layout, yang merupakan salah satu metode baru dalam android studio untuk membuat tampilan layout. Pada dasarnya constraint layout adalah metode yang paling gampang dalam membuat layout, karena dengan metode ini kita hanya perlu melakukan drag and drop widget atau komponen layout yang akan kita gunakan. Tetapi ada beberapa ketentuan yang harus diperhatikan dalam menggunakan constraint layout. Berikut penjelasan singkat cara penggunaan **constraint layout (PENJELASAN INI TIDAK TERMASUK DALAM GUIDED)** :

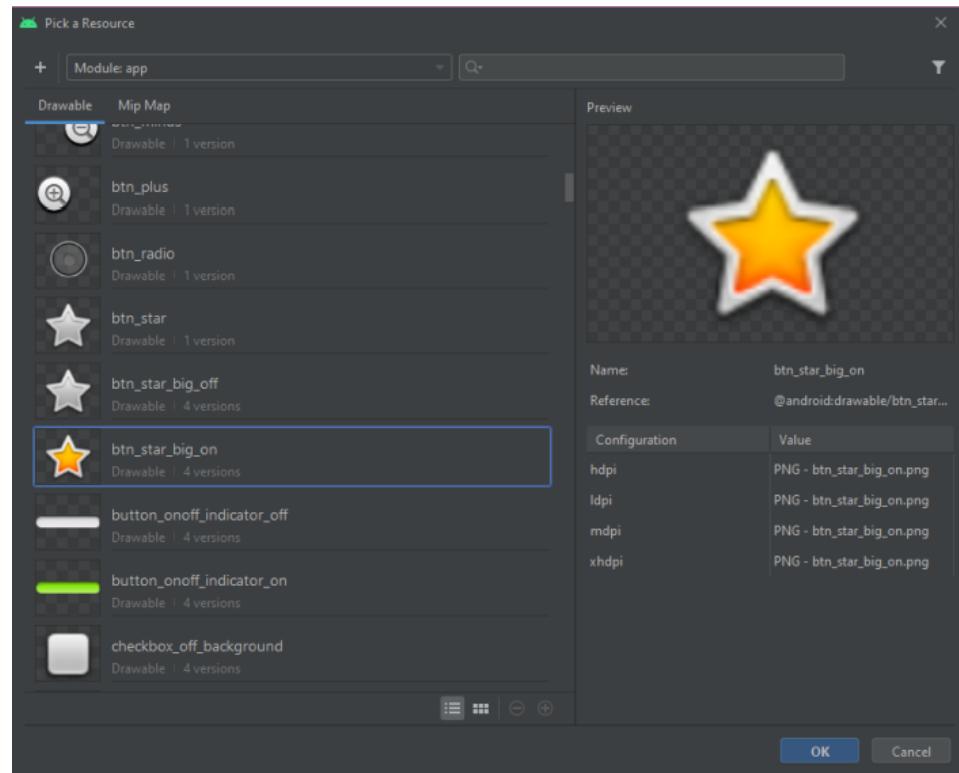
1. Dibagian atas preview windows pada file .xml, terdapat icon berbentuk mata dan magnet, yang fungsinya adalah (mata) menampilkan line constraint yang ada, sedangkan magnet berfungsi sebagai auto connect constraint yang ada, sehingga memudahkan penggunaan constraint.



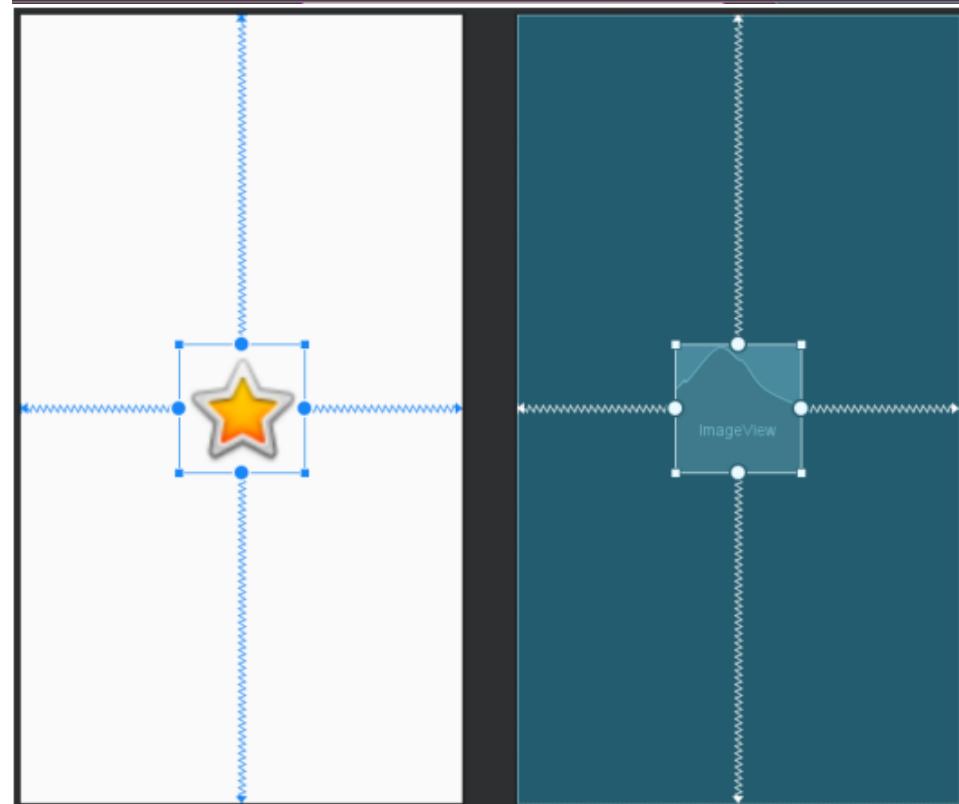
2. Click and drag, imageView ke tengah jendela kerja, dan kemudian pilih gambar yang tersedia(btn\_star\_big\_on).



a.



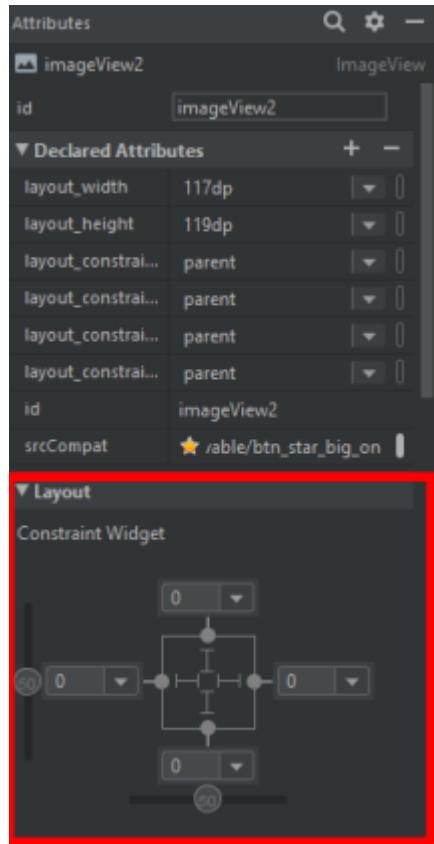
b.



c.

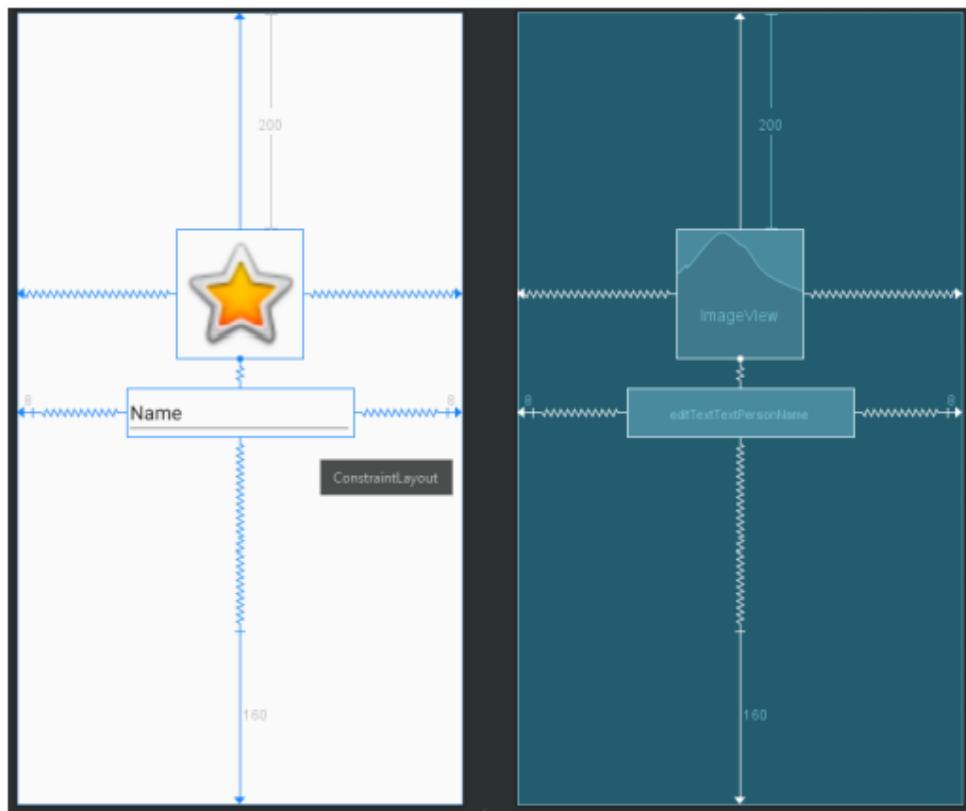
- d. Jika menggunakan auto-connect, maka tampilan seharusnya seperti gambar diatas. Dengan fitur tersebut setiap titik constraint pada widget akan otomatis tersambung ke empat arah (atas, kiri, kanan dan bawah).
- e. Titik-titik constraint tersebut berguna untuk menyimpan posisi dari widget tersebut. Jadi pastikan setiap komponen layout memiliki titik constraint agar memiliki posisi di layout.

- f. Jika kita hendak mengatur posisi widget kita dapat menggunakan tab properties yang terdapat dibagian kanan jendela kerja, dengan menggeser lingkaran yang berisikan posisi widget secara horizontal maupun vertikal.

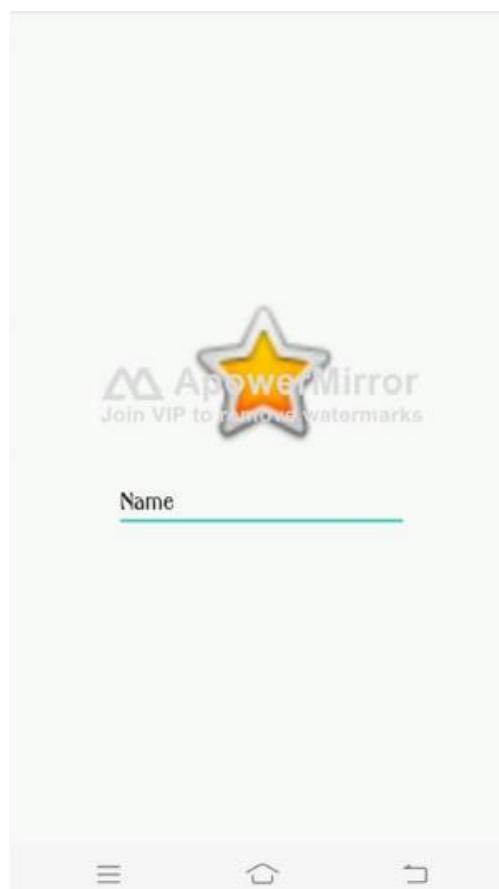


g.

3. Selanjutnya coba tambahkan Text --> Plain Text ke dalam jendela kerja, kemudian atur titik constraint atas plain text, agar menunjuk ke titik constraint bawah image view dengan cara klik titik atas plain text dan kemudian drag pointer mouse hingga menyentuh titik bawah constraint image view. Dan jangan lupa atur titik constraint plain text lainnya, sehingga seperti gambar ini :



- Run aplikasi, jika tampilan sudah sesuai seperti di jendela kerja maka layout sudah benar.

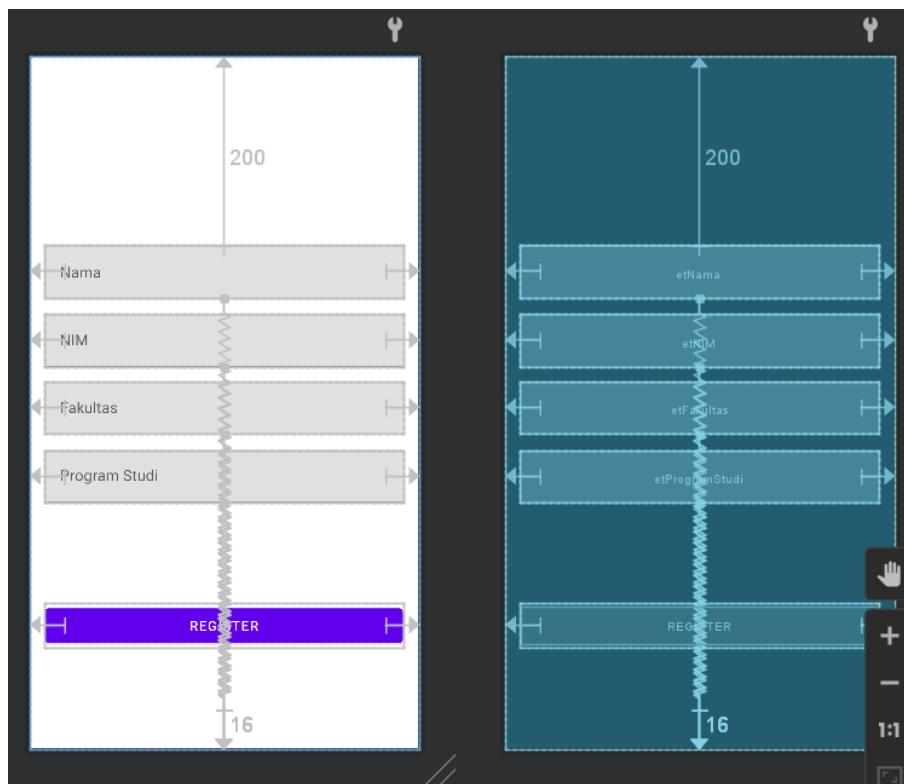


5. Jika masih kurang paham silahkan buka link ini untuk mempelajari constraint layout : <https://developer.android.com/training/constraintlayout/index.html>
6. Untuk penjelasan metode layout lainnya akan di jelaskan pada modul selanjutnya.

## GUIDED 4

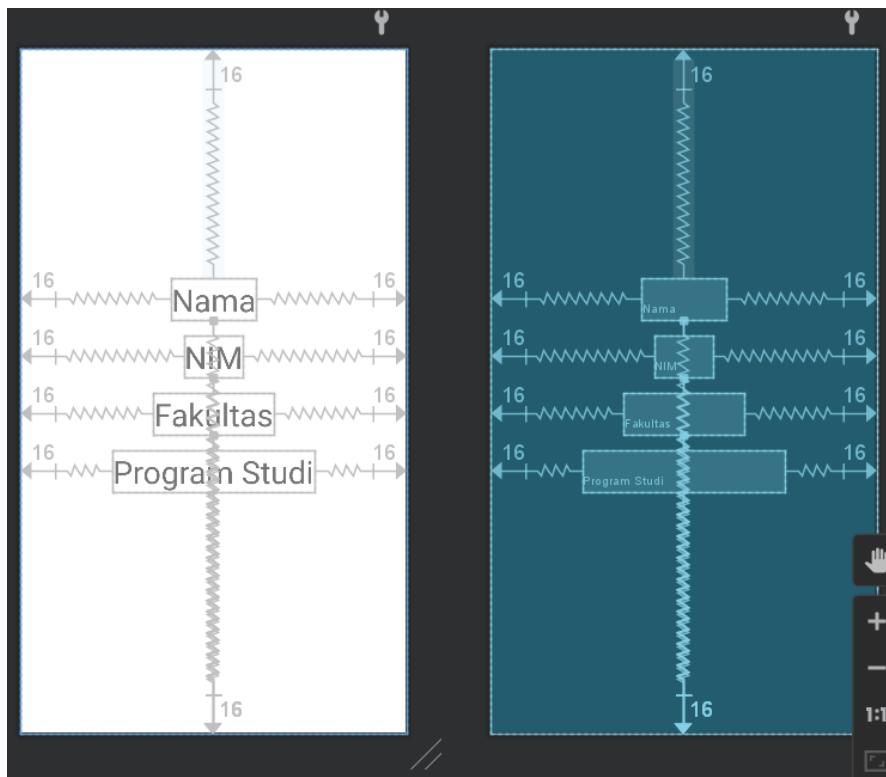
Pada guided 4 ini anda akan mencoba membuat 2 buah activity dengan intent dan mengirimkan message melalui bundle. Silahkan ikuti langkah – langkah berikut ini :

- Buat sebuah projek dengan spesifikasi projek sebagai berikut :
  - a. Template : Empty
  - b. Activity Name : Guided4\_Bundle
  - c. Package Name : (bebas) usahakan berbeda dengan projek yang lainnya, agar pada device, aplikasi dapat dibedakan (tidak tertumpuk)
  - d. Language : Kotlin
  - e. Minimum SDK : API 26: Android 8.0 (Oreo).
- Buatlah 1 empty activity baru dengan nama NextActivity.kt.
- Kemudian, buat tampilan layout activity\_main.xml seperti gambar 30 dan tampilan layout activity\_next.xml seperti gambar 31.



Gambar 30. Activity\_main.xml

**NOTE :** Terdapat 4 PlainText/EditText dan 1 Button. Dicontoh ini digunakan TextInputEditText, penggunaan Edit Text dibebaskan. Penamaan ID bebas untuk setiap widget.



Gambar 31. Activity\_next.xml

**NOTE :** Terdapat 4 TextView. Penamaan ID bebas untuk setiap widget.

- Selanjutnya buka MainActivity.java kemudian tambahkan code berikut:

```

1 package com.brian.helloworld
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.widget.Button
7 import com.google.android.material.textfield.TextInputEditText
8
9 class GD4Bundle : AppCompatActivity() {
10     private lateinit var nama: TextInputEditText
11     private lateinit var nim : TextInputEditText
12     private lateinit var fakultas: TextInputEditText
13     private lateinit var prodi: TextInputEditText
14     private lateinit var btnRegister: Button
15
16     override fun onCreate(savedInstanceState: Bundle?) {
17         super.onCreate(savedInstanceState)
18         setContentView(R.layout.activity_gd4_bundle)
19
20         nama = findViewById(R.id.etNama)
21         nim = findViewById(R.id.etNIM)
22         fakultas = findViewById(R.id.etFakultas)
23         prodi = findViewById(R.id.etProgramStudi)
24         btnRegister = findViewById(R.id.btnRegister)
25
26         btnRegister.setOnClickListener { it: View!
27             val intent = Intent(packageContext: this, GD4BundleResult::class.java)
    }
}

```

## ANDROID NAVIGATION AND INTERFACE DESIGN 1

```
26     btnRegister.setOnClickListener { it: View! ->
27         val intent = Intent(packageContext: this, GD4BundleResult::class.java)
28         val mBundle = Bundle()
29         //memasukkan data ke dalam bundle
30         mBundle.putString("nama", nama.text.toString())
31         mBundle.putString("nim", nim.text.toString())
32         mBundle.putString("fakultas", fakultas.text.toString())
33         mBundle.putString("prodi", prodi.text.toString())
34         //melakukan intent dengan memanggil bundle
35         intent.putExtra(name: "register", mBundle)
36
37         startActivity(intent)
38     }
39 }
40 }
```

Device File Explorer Emulator

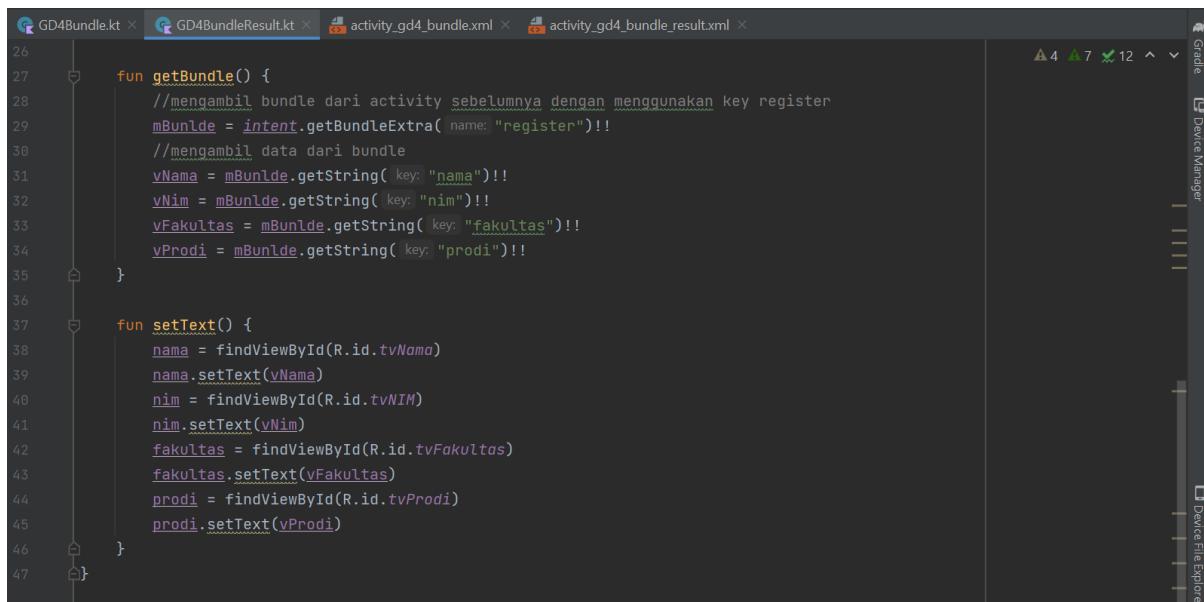
**NOTE :** Perhatikan pada pemberian key tiap variable, pastikan pada saat pemanggilan key pada activity tujuan tidak salah.

Contoh : Misalkan kita akan membundle variable fakultas dari inputan, pada saat pembuatan key kita memberi nama key tersebut “mBundle.putString(“fakultas”, mFakultas.getText().toString());”. Maka pada saat pemanggilan key harus menggunakan key dengan nama “fakultas”.

- Dan untuk NextActivity.java seperti berikut :

```
GD4Bundle.kt x GD4BundleResult.kt x activity_gd4_bundle.xml x activity_gd4_bundle_result.xml
1 package com.brian.helloworld
2
3 import android.os.Bundle
4 import android.widget.TextView
5 import androidx.appcompat.app.AppCompatActivity
6
7 class GD4BundleResult : AppCompatActivity() {
8     private lateinit var nama: TextView
9     private lateinit var nim: TextView
10    private lateinit var fakultas: TextView
11    private lateinit var prodi: TextView
12    lateinit var mBundle: Bundle
13
14    lateinit var vNama: String
15    lateinit var vNim: String
16    lateinit var vFakultas: String
17    lateinit var vProdi: String
18
19    override fun onCreate(savedInstanceState: Bundle?) {
20        super.onCreate(savedInstanceState)
21        setContentView(R.layout.activity_gd4_bundle_result)
22
23        getBundle()
24        setText()
25    }
26
27    fun getBundle() {
```

Gradle Device Manager Device File Explorer Emulator

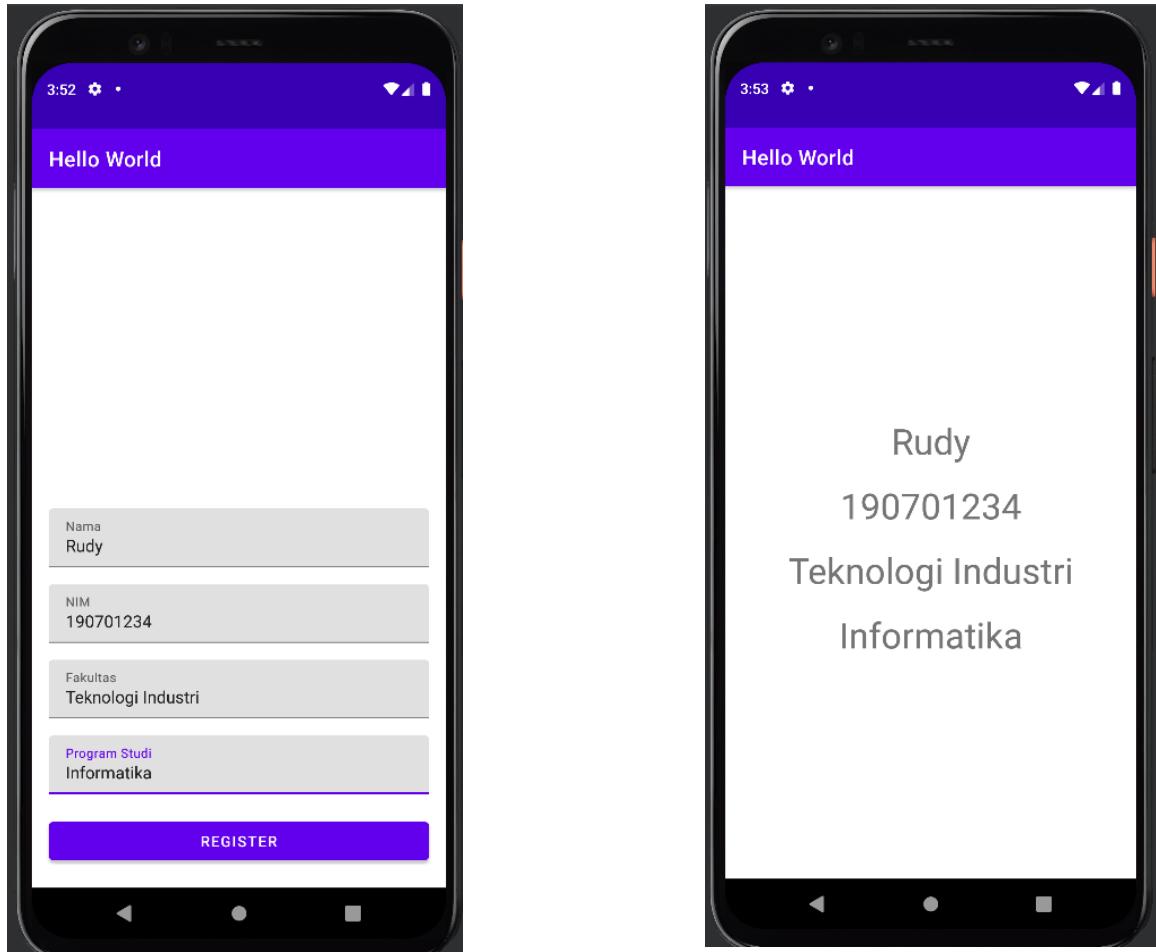


```

26
27     fun getBundle() {
28         //mengambil bundle dari activity sebelumnya dengan menggunakan key register
29         mBunlde = intent.getBundleExtra( name: "register")!!
30         //mengambil data dari bundle
31         vNama = mBunlde.getString( key: "nama")!!
32         vNim = mBunlde.getString( key: "nim")!!
33         vFakultas = mBunlde.getString( key: "fakultas")!!
34         vProdi = mBunlde.getString( key: "prodi")!!
35     }
36
37     fun setText() {
38         nama = findViewById(R.id.tvNama)
39         nama.setText(vNama)
40         nim = findViewById(R.id.tvNIM)
41         nim.setText(vNim)
42         fakultas = findViewById(R.id.tvFakultas)
43         fakultas.setText(vFakultas)
44         prodi = findViewById(R.id.tvProdi)
45         prodi.setText(vProdi)
46     }
47 }

```

- Jika sudah coba run aplikasi.



### PENTING!!!

Ada beberapa aturan yang perlu diperhatikan saat menggunakan bundle, yaitu **pemberian key dan pemanggilan key harus sama**, jika tidak maka akan terjadi error atau data tidak terbaca.

**ATURAN PENGERJAAN GUIDED :**

- Guided dikerjakan di rumah dan jika masih terdapat kendala/ error pada GD, dapat ditanyakan saat perkuliahan.
- Guided dinilai saat perkuliahan berlangsung.
- Penamaan projek guided harus sesuai dengan yang sudah dicontohkan.
- NAMAGUIDED\_XXXX (contoh : Guided1 Intent\_9999)
  - a. NAMAGUIDED → SESUAI DENGAN CONTOH DALAM MODUL INI
  - b. XXXX → 4 DIGIT TERAKHIR NPM

**PENTING!!!**

Kerjakan Guided secara baik dan benar, jangan mengerjakan GD dengan cara copy paste code saja!

Jika ada kendala penggerjaan diusahakan terlebih dulu untuk searching di Internet.

**Selamat mengerjakan!**