

Android Navigation and Interface Design 2

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:

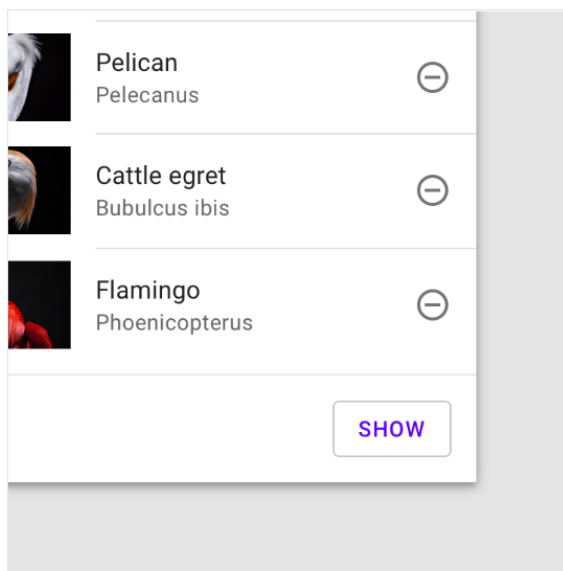
1. Memahami apa yang dimaksudkan dengan material design dan cara mengimplementasikannya.
2. Mampu mengimplementasikan interface sesuai panduan Material Design.
3. Memahami tentang konsep dari recycler view dan cara menggunakannya.
4. Memahami cara menggunakan assets icon pada android studio.
5. Memahami tentang apa itu fragment, dan cara menggunakannya.

TEORI

A. Material Design

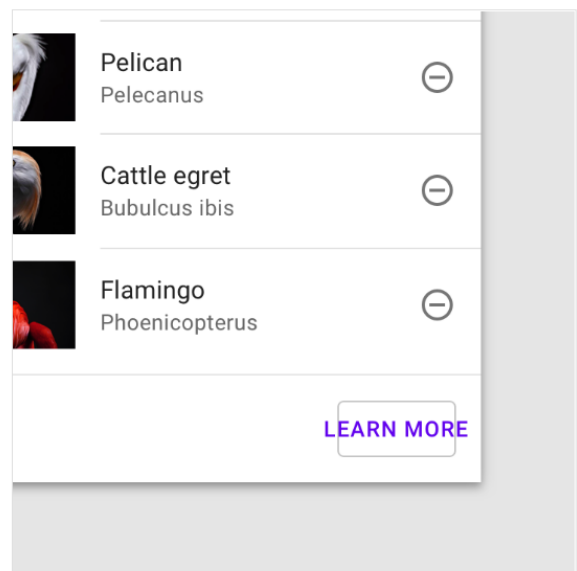
Material design merupakan sebuah sistem design yang dibuat oleh google, sistem design ini memiliki tujuan untuk membantu developer dalam membuat aplikasi dengan kualitas yang tinggi serta memberi experience atau pengalaman yang baik bagi pengguna aplikasi. Material design tersedia untuk Android, IOS, Flutter, & Web. Intinya, material design akan membantu kita sebagai developer untuk membuat tampilan aplikasi yang menarik, serta memiliki UX yang dapat dimengerti dengan mudah oleh penggunanya.

Sebagai sebuah design system, material design akan menyediakan component-component yang nantinya dapat langsung kita pakai di dalam aplikasi android kita misalnya MaterialButton, MaterialTextField, dan lain lain. Selain itu material design juga menyediakan panduan atau tips dalam melakukan design sebuah view atau sebuah UI.



Do

An outlined button's width is dynamically set to fit the text label.



Don't

An outlined button's width shouldn't be narrower than the button's text length.

Berikut merupakan link beberapa contoh dari panduan material design yang di sarankan oleh Google :

1. Panduan untuk Button di android : <https://material.io/components/buttons#hierarchy-and-placement>
2. Panduan untuk TextFields di android : <https://material.io/components/text-fields>

Jika kalian ingin mengetahui lebih lanjut tentang material design, kalian bisa mengunjungi websitenya langsung di material.io.

B. Recycler View

Recycler view merupakan sebuah viewgroup yang berguna untuk mengatur data dalam jumlah banyak (list of data). Recycler view akan membantu dalam membuat satu tampilan lalu digunakan pada banyak data dan ditampilkan dalam bentuk list yang bisa di scroll.

Pada penggunaan recycler view, kita akan terlebih dahulu menentukan bagaimana layout atau tampilan dari item kita. Lalu layout yang telah kita buat dapat kita gunakan dalam menampilkan banyak data dengan 1 layout saja. Selain itu recycler view juga akan membantu kita dalam menyusun datanya dengan menggunakan index.

Ada beberapa komponen yang harus kita ketahui sebelum menggunakan recycler view yaitu :

1. Item Layout : Komponen ini merupakan komponen layout yang kita gunakan untuk menampilkan item kita.
2. DataList : Komponen ini merupakan data yang ingin kita gunakan pada RecyclerView kita.
3. Recycler view & Layout Manager : Komponen ini akan berfungsi untuk mengatur bagaimana kita ingin menampilkan data pada recycler view kita. Beberapa contohnya sendiri misalnya LinearLayoutManager (untuk -menampilkan data secara vertical atau horizontal) dan GridLayoutManager (untuk menampilkan data dalam bentuk baris dan kolom)
4. Adapter : Komponen ini akan berfungsi untuk mengatur data-data yang ada pada recycler view, pada komponen ini nantinya kita akan melakukan bind data kita ke view yang telah kita buat. Bind disini dapat diartikan sebagai implementasikan atau pengisian data kita ke view kita.

C. Fragment

Fragment merupakan salah satu view atau komponen yang ada pada android yang memiliki lifecycle mirip dengan activity. Sehingga nantinya di dalam sebuah fragment akan memiliki 2 file juga yaitu file logic dan file layout. Lalu apa yang membedakan activity dengan fragment ?

Perbedaan Activity dan Fragment

1. Fragment tidak berdiri sendiri, melainkan fragment merupakan sebuah view yang ada dalam activity atau kata lainnya fragment membutuhkan activity.
2. Dalam activity bisa memiliki banyak fragment. Hal ini memungkinkan kita untuk bisa melakukan perpindahan atau pergantian user interface dalam sebuah activity.

3. Perpindahan yang terjadi antara fragment akan lebih cepat dan smooth dibanding dengan perpindahan antara activity.
4. Dalam fragment kita juga bisa menggunakan method yang ada pada activity kita.
5. Biasanya dalam satu activity, kita ingin terdapat beberapa user interface yang berbeda, misalnya kita ingin menyatukan Login Screen dan Register Screen karena menurut kita 2 hal tersebut masih berada dalam context yang sama. Dalam hal ini kita dapat menggunakan Fragment agar hal tersebut bisa terjadi.

Guided

Pada guided ini akan mempelajari mengenai material design, asset icon vector, recycler view, dan fragment.

1. Buatlah projek dengan spesifikasi projek sebagai berikut:
 - a. Template: **Empty Activity**
 - b. Activity Name: **GD3_Y_XXXX**, dengan Y adalah kelas dan X adalah NPM.
 - c. Language: **Kotlin**
 - d. Minimum SDK: **API 26: Android 8.0 (Oreo)**
2. Pada sidebar, klik mouse kanan pada folder **java** -> **com.example.gd_y_xxxx**. Pilih **New** -> **Package**. Lalu beri nama package tersebut dengan nama **entity**.
3. Pada package **entity**. Klik mouse kanan lalu pilih **New** -> **Kotlin Class** dan isikan dengan nama **Mahasiswa** dengan tipe Class lalu enter.
4. Buat juga hal yang sama untuk membuat Kotlin Class baru dengan nama **Dosen**.
5. Ketikan kode berikut. Kode berikut adalah isi dari kelas **Mahasiswa** dan **Dosen** dengan data dummynya.

```
Mahasiswa.kt
1 package com.example.guided3.entity
2
3 class Mahasiswa(var name: String, var IPK: Double, var tahunMasuk: Int) {
4
5     //companion object seperti static di java dan berikut adalah array untuk menyimpan data dummy mahasiswa
6     companion object{
7         @JvmField
8         var listOfMahasiswa = arrayOf(
9             Mahasiswa( name: "Wendy Winata", IPK: 3.5, tahunMasuk: 2018),
10            Mahasiswa( name: "Eras Timothy", IPK: 3.7, tahunMasuk: 2018),
11            Mahasiswa( name: "Jonathan", IPK: 3.8, tahunMasuk: 2018),
12            Mahasiswa( name: "Yosia", IPK: 3.9, tahunMasuk: 2018),
13            Mahasiswa( name: "Yotam", IPK: 4.0, tahunMasuk: 2019)
14        )
15    }
16 }
```

Gambar Kode Kelas Mahasiswa

```
Dosen.kt
1 package com.example.guided3.entity
2
3 class Dosen(var name: String, var pengajar: String) {
4
5     //companion object seperti static di java dan berikut adalah array untuk menyimpan data dummy dosen
6     companion object{
7         @JvmField
8         var listOfDosen = arrayOf(
9             Dosen( name: "Fedelis Brian", pengajar: "Pengajar Kelas A, B, dan D"),
10            Dosen( name: "Thomas Adi", pengajar: "Pengajar Kelas C"),
11        )
12    }
13 }
```

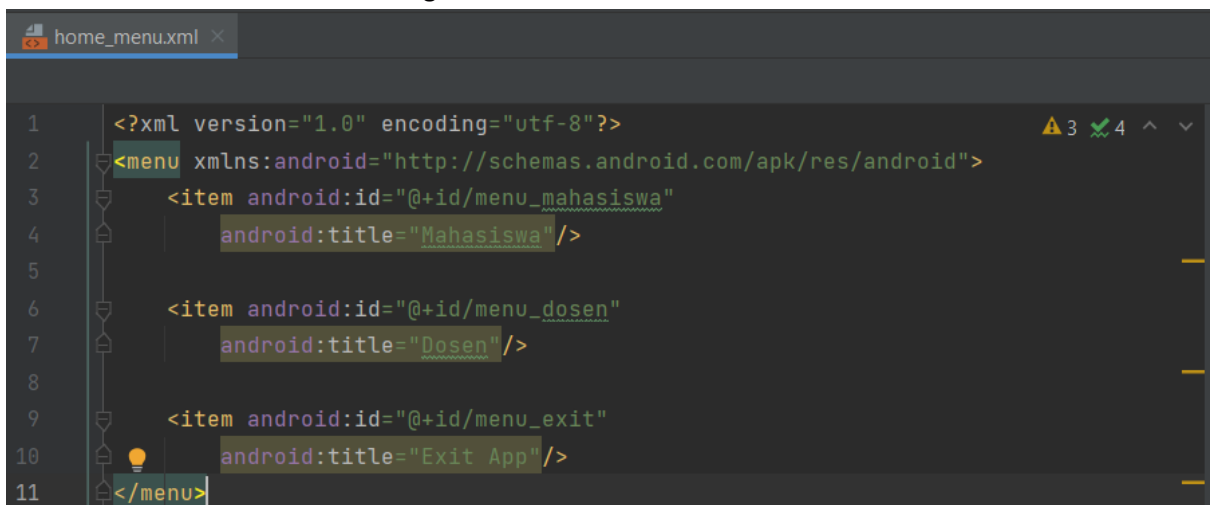
Gambar Kode Kelas Dosen

(Membuat Icon Asstes Vector dan Gambar)

6. Pada bagian ini kita akan membuat icon assets vector yang akan kita gunakan di dalam aplikasi. Pada folder **res -> drawable** klik kanan, lalu **new -> Vector Assets**.
7. Pada Dialog yang muncul, tekan icon yang ada pada baris **Clip Art**.
8. Pada kolom **Search**, tulis **Person**, dan pilih icon **Person** yang muncul, dan klik **OK**.
9. Pada kolom **Name**, ganti namanya menjadi **ic_user_24**. Pilihan yang lain dibiarkan saja, lalu klik **Next** dan **Finish**.
10. Lakukan Step 6-9 kembali namun dengan **icon lock** dan beri nama icon tersebut **ic_lock_24**.
11. Ambil salah satu gambar dari internet (gambar login atau bebas) dan ubah namanya menjadi satu kata dengan ekstensi **.jpg** atau **.png**. Contohnya gambar.png.
12. Drag dan drop gambar tersebut (bisa copy paste) ke **package res -> drawable**.

(Membuat Menu)

13. Pada step ini kita akan membuat sebuah menu yang akan kita pakai di dalam aplikasi kita nanti. Pada **res** klik kanan, lalu **new -> Android Resources File**.
14. Beri nama filenya dengan **home_menu**.
15. Select **Resource type** dengan **Menu**, Lainnya biarkan seperti default.
16. Klik OK, dan file **home_menu.xml** akan terbentuk pada **res -> menu**.
17. Isikan code dalam file tersebut dengan code berikut.



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android">
3      <item android:id="@+id/menu_mahasiswa"
4          android:title="Mahasiswa"/>
5
6      <item android:id="@+id/menu_dosen"
7          android:title="Dosen"/>
8
9      <item android:id="@+id/menu_exit"
10         android:title="Exit App"/>
11 </menu>
    
```

Gambar Kode XML Menu

(Membuat Layout untuk main activity)

18. Pada sidebar, klik package **res -> layout -> activity_main.xml**.
19. Pada layout activity_main.xml akan digunakan untuk tampilan login. Kita akan memakai view yang disediakan oleh **material design**. Perhatikan pada code **Textview**, kita tidak akan memakai **fontSize** lagi, tetapi kita memakai **style** yang disediakan oleh **material design** untuk text judul yaitu **Headline4**. Selain **h4** kita juga bisa menemukan **h1-h6** yang bisa kalian coba sendiri.

ANDROID NAVIGATION AND INTERFACE DESIGN 2

```
activity_main.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      android:id="@+id/mainLayout"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <ImageView
11         android:id="@+id/imageView"
12         android:layout_width="240dp"
13         android:layout_height="240dp"
14         android:layout_marginTop="32dp"
15         android:src="@drawable/unnamed"
16         app:layout_constraintEnd_toEndOf="parent"
17         app:layout_constraintStart_toStartOf="parent"
18         app:layout_constraintTop_toTopOf="parent" />
19
20     <TextView
21         android:id="@+id/textView"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:layout_marginTop="16dp"
25         android:text="Student Login"
26         android:textColor="@color/black"
27         style="@style/TextAppearance.MaterialComponents.Headline4"
28         app:layout_constraintEnd_toEndOf="@+id/imageView"
29         app:layout_constraintStart_toStartOf="@+id/imageView"
30         app:layout_constraintTop_toBottomOf="@+id/imageView" />
31
32     <LinearLayout
33         android:id="@+id/linearLayout2"
34         android:layout_width="match_parent"
35         android:layout_height="wrap_content"
36         android:layout_marginStart="16dp"
37         android:layout_marginEnd="16dp"
38         app:layout_constraintTop_toBottomOf="@+id/textView"
39         android:paddingTop="16dp"
40         android:paddingBottom="16dp"
41         android:orientation="vertical">
42
43         <com.google.android.material.textfield.TextInputLayout
44             android:id="@+id/inputLayoutUsername"
45             android:layout_width="match_parent"
46             android:layout_height="wrap_content"
47             style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
48             android:hint="Username"
49             app:errorEnabled="true"
50             app:startIconDrawable="@drawable/ic_user_24"
51             android:layout_marginBottom="8dp">
52
53             <com.google.android.material.textfield.TextInputEditText
54                 android:layout_width="match_parent"
55                 android:layout_height="wrap_content"
56                 android:inputType="text" />
57
58         </com.google.android.material.textfield.TextInputLayout>
```

```

60 <com.google.android.material.textfield.TextInputLayout
61     android:id="@+id/inputLayoutPassword"
62     android:layout_width="match_parent"
63     android:layout_height="wrap_content"
64     style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
65     android:hint="Password"
66     app:errorEnabled="true"
67     app:startIconDrawable="@drawable/ic_lock_24"
68     app:helperText="Masukan password anda">
69
70     <com.google.android.material.textfield.TextInputEditText
71         android:layout_width="match_parent"
72         android:layout_height="wrap_content"
73         android:inputType="textPassword"
74     />
75
76 </com.google.android.material.textfield.TextInputLayout>
77
78 </LinearLayout>
79
80 <LinearLayout
81     android:id="@+id/linearLayout"
82     android:layout_width="0dp"
83     android:layout_height="wrap_content"
84     android:layout_marginTop="16dp"
85     app:layout_constraintEnd_toEndOf="@+id/linearLayout2"
86     app:layout_constraintStart_toStartOf="@+id/linearLayout2"
87     app:layout_constraintTop_toBottomOf="@+id/linearLayout2">
88
89     <com.google.android.material.button.MaterialButton
90         android:id="@+id/btnClear"
91         style="@style/Widget.MaterialComponents.Button.TextButton"
92         android:layout_width="wrap_content"
93         android:layout_height="wrap_content"
94         android:layout_weight="1"
95         android:text="Clear Field" />
96
97     <com.google.android.material.button.MaterialButton
98         android:id="@+id/btnLogin"
99         android:layout_width="wrap_content"
100        android:layout_height="wrap_content"
101        android:layout_weight="1"
102        android:text="Login" />
103
104 </LinearLayout>
105
106 </androidx.constraintlayout.widget.ConstraintLayout>

```

Gambar Kode activity_main.xml

(Membuat Logic untuk MainActivity)

20. Pada bagian selanjutnya kita akan membuat logic untuk **MainActivity**. Isikan kode MainActivity sebagai berikut. Pada kode bagian ini terdapat warna merah pada bagian **this@HomeActivity**. Tidak usah khawatir karena memang merah karena belum buat activity HomeActivity.
21. Pada baris 60, isikan **username** dengan "admin" dan **password** dengan 4 digit terakhir NPM.


```

1 package com.example.guided3
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.view.View
7 import android.widget.Button
8 import androidx.constraintlayout.widget.ConstraintLayout
9 import com.google.android.material.snackbar.Snackbar
10 import com.google.android.material.textfield.TextInputLayout
11
12 class MainActivity : AppCompatActivity() {
13     // Attribute yang akan kita pakai
14     private lateinit var inputUsername: TextInputLayout
15     private lateinit var inputPassword: TextInputLayout
16     private lateinit var mainLayout: ConstraintLayout
17
18     override fun onCreate(savedInstanceState: Bundle?) {
19         super.onCreate(savedInstanceState)
20         setContentView(R.layout.activity_main)
21
22         // Ubah Title pada App Bar Aplikasi
23         setTitle("User Login")
24
25         // Hubungkan variabel dengan view di layoutnya.
26         inputUsername = findViewById(R.id.inputLayoutUsername)
27         inputPassword = findViewById(R.id.inputLayoutPassword)
28         mainLayout = findViewById(R.id.mainLayout)
29         val btnClear: Button = findViewById(R.id.btnClear)
30         val btnLogin: Button = findViewById(R.id.btnLogin)
31
32         // Aksi btnClear ketika di klik
33         btnClear.setOnClickListener { // Mengosongkan Input
34             inputUsername.getEditText()?.setText("")
35             inputPassword.getEditText()?.setText("")
36
37             // Memunculkan Snackbar
38             Snackbar.make(mainLayout, text: "Text Cleared Success", Snackbar.LENGTH_LONG).show()
39         }
40
41         // Aksi pada btnLogin
42         btnLogin.setOnClickListener(View.OnClickListener { it: View?
43             var checkLogin = false
44             val username: String = inputUsername.getEditText()?.getText().toString()
45             val password: String = inputPassword.getEditText()?.getText().toString()
46
47             // Pengecekan apakah inputan kosong
48             if (username.isEmpty()) {
49                 inputUsername.setError("Username must be filled with text")
50                 checkLogin = false
51             }
52
53             // Pengecekan apakah Inputan kosong
54             if (password.isEmpty()) {
55                 inputPassword.setError("Password must be filled with text")
56                 checkLogin = false
57             }
58         })
59     }
60 }
  
```

```

58
59      // Ganti Password dengan NPM kalian.
60      if (username == "admin" && password == "0350") checkLogin = true
61      if (!checkLogin) return@OnClickListener
62      val moveHome = Intent( packageContext: this@MainActivity, HomeActivity::class.java)
63      startActivity(moveHome)
64    })
65  }
66
  
```

Gambar Kode MainActivity

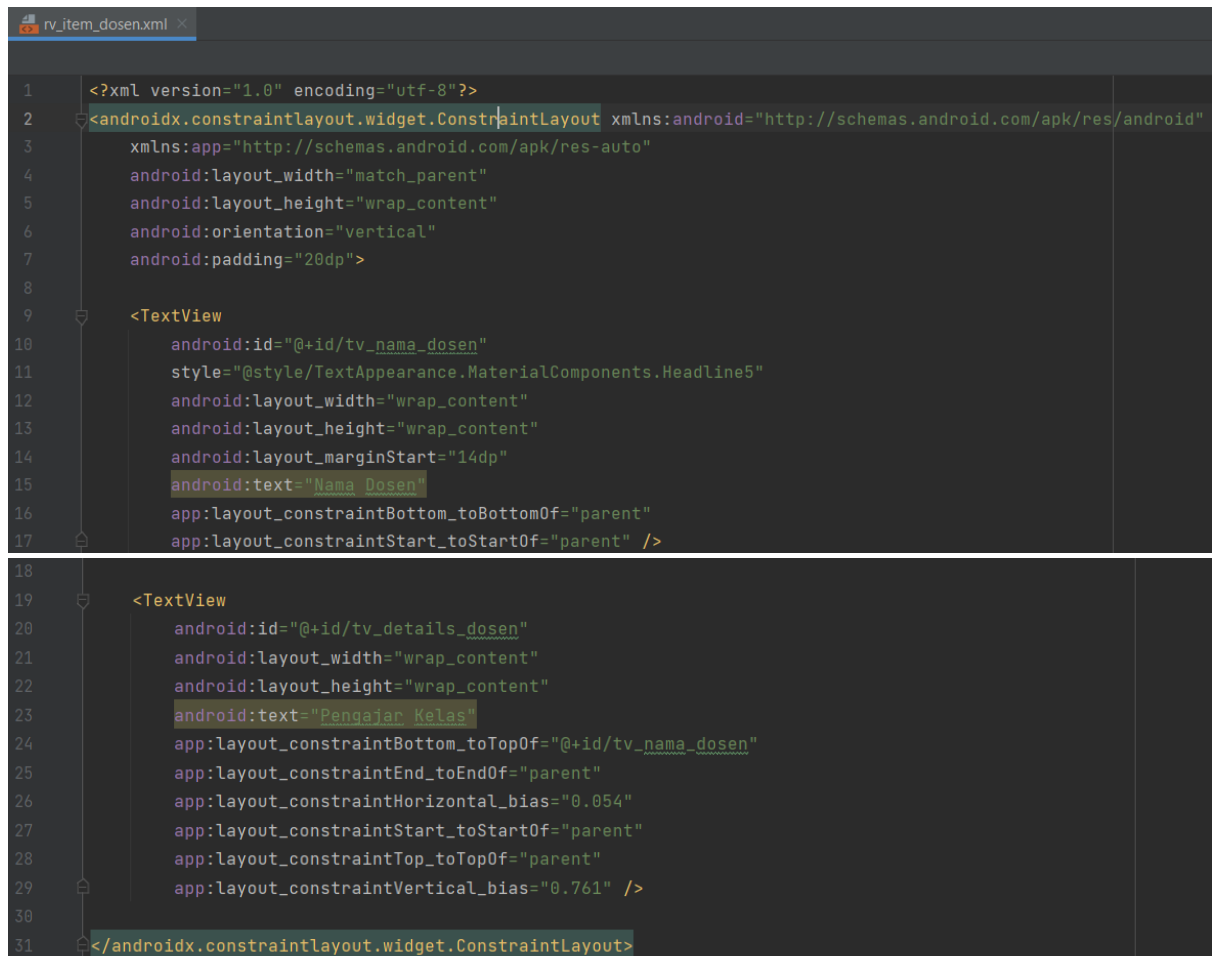
(Membuat Layout untuk item recycler view mahasiswa dan dosen)

22. Berikut kita akan membuat layout untuk item recycler view untuk data mahasiswa dan dosen. Pada res -> layout, klik kanan lalu new -> Layout Resources File. Isikan dengan nama rv_item_mahasiswa lalu ok.
23. Buat hal yang sama namun dengan nama rv_item_dosen.
24. Isikan kode sesuai gambar berikut.

```

rv_item_mahasiswa.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      android:layout_width="match_parent"
5      android:layout_height="wrap_content"
6      android:orientation="vertical"
7      android:padding="20dp">
8
9      <TextView
10         android:id="@+id/tv_nama"
11         style="@style/TextAppearance.MaterialComponents.Headline5"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:layout_marginStart="14dp"
15         android:text="Nama Mahasiswa"
16         app:layout_constraintBottom_toBottomOf="parent"
17         app:layout_constraintStart_toStartOf="parent" />
18
19      <TextView
20         android:id="@+id/tv_details"
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:text="Tahun Masuk - IPK"
24         app:layout_constraintBottom_toTopOf="@+id/tv_nama"
25         app:layout_constraintEnd_toEndOf="parent"
26         app:layout_constraintHorizontal_bias="0.054"
27         app:layout_constraintStart_toStartOf="parent"
28         app:layout_constraintTop_toTopOf="parent"
29         app:layout_constraintVertical_bias="0.761" />
30
31  </androidx.constraintlayout.widget.ConstraintLayout>
  
```

Gambar Kode rv_item_mahasiswa.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      android:layout_width="match_parent"
5      android:layout_height="wrap_content"
6      android:orientation="vertical"
7      android:padding="20dp">
8
9      <TextView
10         android:id="@+id/tv_nama_dosen"
11         style="@style/TextAppearance.MaterialComponents.Headline5"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:layout_marginStart="14dp"
15         android:text="Nama Dosen"
16         app:layout_constraintBottom_toBottomOf="parent"
17         app:layout_constraintStart_toStartOf="parent" />
18
19      <TextView
20         android:id="@+id/tv_details_dosen"
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:text="Pengajar Kelas"
24         app:layout_constraintBottom_toTopOf="@+id/tv_nama_dosen"
25         app:layout_constraintEnd_toEndOf="parent"
26         app:layout_constraintHorizontal_bias="0.054"
27         app:layout_constraintStart_toStartOf="parent"
28         app:layout_constraintTop_toTopOf="parent"
29         app:layout_constraintVertical_bias="0.761" />
30
31 </androidx.constraintlayout.widget.ConstraintLayout>

```

Gambar Kode rv_item_dosen.xml

(Membuat Adapter RecyclerView)

25. Pada tahap ini, kita akan membuat adapter yang berguna untuk mengelola data dummy dari mahasiswa dan dosen untuk setiap item pada recycler view kita.
26. Pada **java -> com.example.gd_y_XXXXX**. Klik kanan lalu **New -> Java Class ->** Isikan dengan **RVMahasiswaAdapter -> OK**.
27. Lakukan langkah 25 untuk membuat hal yang sama namun dengan nama RVDosenAdapter.
28. Isikan kode recyler view tersebut dengan kode ini. Terdapat komen dalam kodenya.

ANDROID NAVIGATION AND INTERFACE DESIGN 2

```
RVMahasiswaAdapter.kt
1 package com.example.guided3
2
3 import android.view.LayoutInflater
4 import android.view.View
5 import android.view.ViewGroup
6 import android.widget.TextView
7 import androidx.recyclerview.widget.RecyclerView
8 import com.example.guided3.entity.Mahasiswa
9
10 // Attribute dan Konstruktor data kita pada adapter
11 class RVMahasiswaAdapter(private val data: Array<Mahasiswa>) : RecyclerView.Adapter<RVMahasiswaAdapter.viewHolder>() {
12
13     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): viewHolder {
14         // Disini kita menghubungkan layout item recycler view kita
15         val itemView = LayoutInflater.from(parent.context).inflate(R.layout.rv_item_mahasiswa, parent, attachToRoot: false)
16         return viewHolder(itemView)
17     }
18
19     override fun onBindViewHolder(holder: viewHolder, position: Int) {
20         // Karena kita sudah mendefinisikan dan menghubungkan view kita,
21         // kita bisa memakai view tersebut dan Melakukan set text pada view tersebut
22         val currentItem = data[position]
23         holder.tvNama.text = currentItem.name
24         holder.tvDetails.text = "${currentItem.tahunMasuk} - ${currentItem.IPK}"
25     }
26
27     override fun getItemCount(): Int {
28         // Disini kita memberitahu jumlah dari item pada recycler view kita.
29         return data.size
30     }
31
32     // Kelas ini berguna untuk menghubungkan view view yang ada pada item di recycler view kita.
33     class viewHolder(itemView : View) : RecyclerView.ViewHolder(itemView) {
34         val tvNama : TextView = itemView.findViewById(R.id.tv_nama)
35         val tvDetails : TextView = itemView.findViewById(R.id.tv_details)
36     }
37 }
```

Gambar Kode RVMahasiswaAdapter

```
RVDosenAdapter.kt
1 package com.example.guided3
2
3 import android.view.LayoutInflater
4 import android.view.View
5 import android.view.ViewGroup
6 import android.widget.TextView
7 import androidx.recyclerview.widget.RecyclerView
8 import com.example.guided3.entity.Dosen
9
10 class RVDosenAdapter(private val data: Array<Dosen>) : RecyclerView.Adapter<RVDosenAdapter.viewHolder>() {
11
12     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): viewHolder {
13         // Disini kita menghubungkan layout item recycler view kita
14         val itemView = LayoutInflater.from(parent.context).inflate(R.layout.rv_item_dosen, parent, attachToRoot: false)
15         return viewHolder(itemView)
16     }
17
18     override fun onBindViewHolder(holder: viewHolder, position: Int) {
19         // Karena kita sudah mendefinisikan dan menghubungkan view kita,
20         // kita bisa memakai view tersebut dan Melakukan set text pada view tersebut
21         val currentItem = data[position]
22         holder.tvNamaDosen.text = currentItem.name
23         holder.tvDetailsDosen.text = currentItem.pengajar
24     }
25 }
```

```

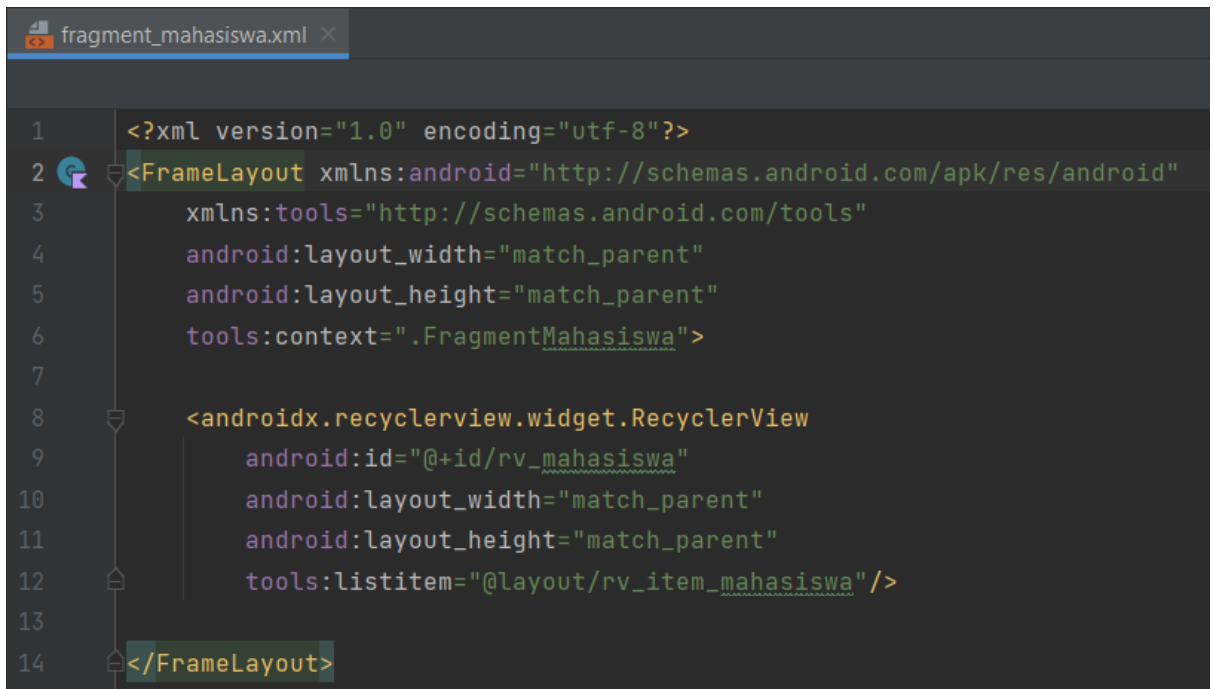
25
26 override fun getItemCount(): Int {
27     // Disini kita memberitahu jumlah dari item pada recycler view kita.
28     return data.size
29 }
30
31 // Kelas ini berguna untuk menghubungkan view view yang ada pada item di recycler view kita.
32 class viewHolder(itemView : View) : RecyclerView.ViewHolder(itemView) {
33     val tvNamaDosen : TextView = itemView.findViewById(R.id.tv_nama_dosen)
34     val tvDetailsDosen : TextView = itemView.findViewById(R.id.tv_details_dosen)
35 }
36 }

```

Gambar Kode RVDosenAdapter

(Membuat Fragment Dosen dan Mahasiswa)

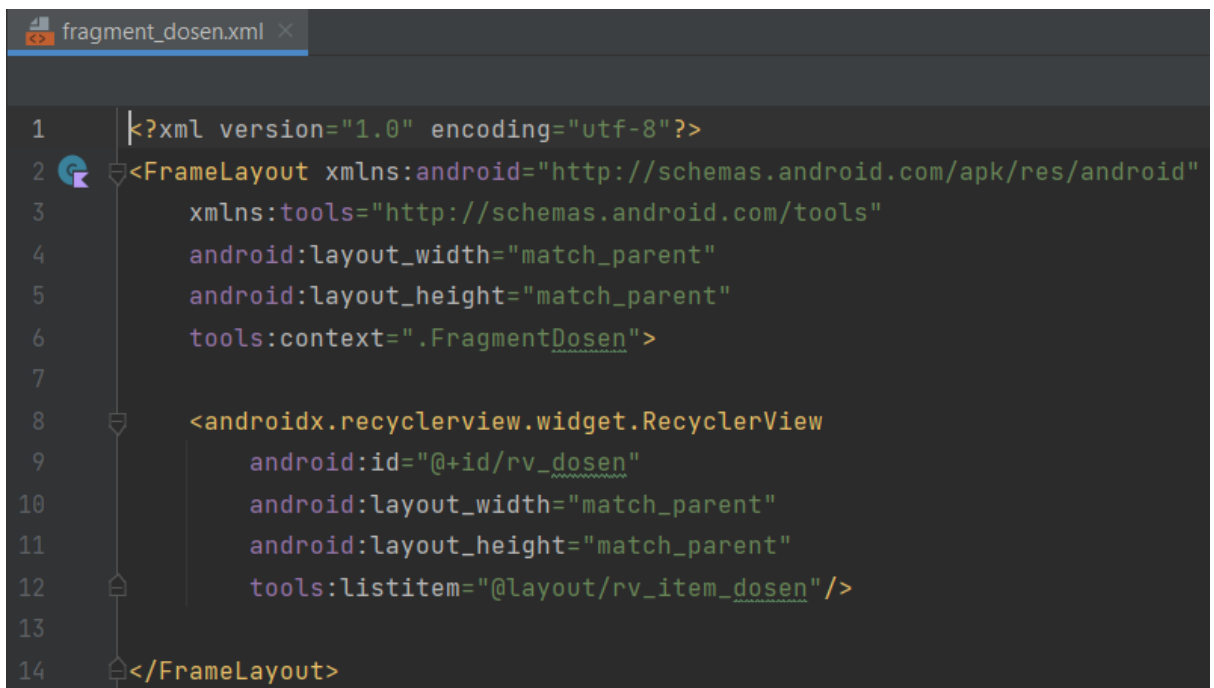
29. Selanjutnya kita akan membuat fragment mahasiswa dan dosen.
30. Pada folder java, klik kanan lalu pilih **new -> fragment -> blank fragment**.
31. Isikan nama fragment dengan **FragmentMahasiswa**.
32. Klik **OK** atau **Finish**. Lalu akan tergenerate 2 file yang akan kita gunakan untuk menaruh layout dan logiknya nanti.
33. Ulang langkah 29-31 untuk membuat FragmentDosen.
34. Isikan file tersebut dengan kode berikut.



```

1      <?xml version="1.0" encoding="utf-8"?>
2      <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3          xmlns:tools="http://schemas.android.com/tools"
4          android:layout_width="match_parent"
5          android:layout_height="match_parent"
6          tools:context=".FragmentMahasiswa">
7
8          <androidx.recyclerview.widget.RecyclerView
9              android:id="@+id/rv_mahasiswa"
10             android:layout_width="match_parent"
11             android:layout_height="match_parent"
12             tools:listitem="@layout/rv_item_mahasiswa"/>
13
14      </FrameLayout>
  
```

Gambar Kode fragment_mahasiswa.xml



```

1      <?xml version="1.0" encoding="utf-8"?>
2      <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3          xmlns:tools="http://schemas.android.com/tools"
4          android:layout_width="match_parent"
5          android:layout_height="match_parent"
6          tools:context=".FragmentDosen">
7
8          <androidx.recyclerview.widget.RecyclerView
9              android:id="@+id/rv_dosen"
10             android:layout_width="match_parent"
11             android:layout_height="match_parent"
12             tools:listitem="@layout/rv_item_dosen"/>
13
14      </FrameLayout>
  
```

Gambar Kode fragment_dosen.xml

```

1  package com.example.guided3
2
3  import android.os.Bundle
4  import androidx.fragment.app.Fragment
5  import android.view.LayoutInflater
6  import android.view.View
7  import android.view.ViewGroup
8  import androidx.recyclerview.widget.LinearLayoutManager
9  import androidx.recyclerview.widget.RecyclerView
10 import com.example.guided3.entity.Mahasiswa
11
12 class FragmentMahasiswa : Fragment() {
13
14     override fun onCreateView(
15         inflater: LayoutInflater, container: ViewGroup?,
16         savedInstanceState: Bundle?
17     ): View? {
18         // Inflate the layout for this fragment
19         // Proses menghubungkan layout fragment_mahasiswa.xml dengan fragment ini
20         return inflater.inflate(R.layout.fragment_mahasiswa, container, attachToRoot: false)
21     }
22
23     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
24         super.onViewCreated(view, savedInstanceState)
25         val layoutManager = LinearLayoutManager(context)
26         val adapter : RVMahasiswaAdapter = RVMahasiswaAdapter(Mahasiswa.listOfMahasiswa)
27
28         // Menghubungkan rvMahasiswa dengan recycler view yang ada pada layout
29         val rvMahasiswa : RecyclerView = view.findViewById(R.id.rv_mahasiswa)
30
31         // Set Layout Manager dari recycler view
32         rvMahasiswa.layoutManager = layoutManager
33
34         //tidak mengubah size recycler view jika terdapat item ditambahkan atau dikurangkan
35         rvMahasiswa.setHasFixedSize(true)
36
37         // Set Adapter dari recycler view.
38         rvMahasiswa.adapter = adapter
39     }
40 }
  
```

Gambar Kode FragmentMahasiswa.kt

```

1  package com.example.guided3
2
3  import android.os.Bundle
4  import androidx.fragment.app.Fragment
5  import android.view.LayoutInflater
6  import android.view.View
7  import android.view.ViewGroup
8  import androidx.recyclerview.widget.LinearLayoutManager
9  import androidx.recyclerview.widget.RecyclerView
10 import com.example.guided3.entity.Dosen
11
12 class FragmentDosen : Fragment() {
13
14     override fun onCreateView(
15         inflater: LayoutInflater, container: ViewGroup?,
16         savedInstanceState: Bundle?
17     ): View? {
18         // Inflate the layout for this fragment
19         // Proses menghubungkan layout fragment_dosen.xml dengan fragment ini
20         return inflater.inflate(R.layout.fragment_dosen, container, attachToRoot: false)
21     }
22
23     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
24         super.onViewCreated(view, savedInstanceState)
25         val layoutManager = LinearLayoutManager(context)
26         val adapter : RV DosenAdapter = RV DosenAdapter(Dosen.listOfDosen)
27
28         // Menghubungkan rvMahasiswa dengan recycler view yang ada pada layout
29         val rvDosen : RecyclerView = view.findViewById(R.id.rv_dosen)
30
31         // Set Layout Manager dari recycler view
32         rvDosen.layoutManager = layoutManager
33
34         //tidak mengubah size recycler view jika terdapat item ditambahkan atau dikurangkan
35         rvDosen.setHasFixedSize(true)
36
37         // Set Adapter dari recycler view.
38         rvDosen.adapter = adapter
39     }
40 }

```

Gambar Kode FragmentDosen.kt

(Membuat Activity Home)

35. Pada langkah ini, kita akan membuat activity baru dengan nama **HomeActivity**. Pada **java** -> **klik kanan** -> **new** -> **activity** -> **Empty Activity**. Isikan nama activity dengan **HomeActivity**. Pastikan bahasa yang dipilih adalah bahasa kotlin, lalu klik **Finish**.
36. Isikan HomeActivity dan activity_home.xml dengan kode berikut.

ANDROID NAVIGATION AND INTERFACE DESIGN 2

```
activity_home.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=".HomeActivity">
8
9     <androidx.fragment.app.FragmentContainerView
10         android:id="@+id/layout_fragment"
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintEnd_toEndOf="parent"
15         app:layout_constraintStart_toStartOf="parent"
16         app:layout_constraintTop_toTopOf="parent" />
17
18 </androidx.constraintlayout.widget.ConstraintLayout>
```

Gambar Kode activity_home.xml

```
HomeActivity.kt
1 package com.example.guided3
2
3 import android.content.DialogInterface
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.view.Menu
7 import android.view.MenuInflater
8 import android.view.MenuItem
9 import androidx.appcompat.app.AlertDialog
10 import androidx.fragment.app.Fragment
11
12 class HomeActivity : AppCompatActivity() {
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         setContentView(R.layout.activity_home)
16
17         // Fragment yang pertama kali terlihat adalah fragment mahasiswa
18         changeFragment(FragmentMahasiswa())
19     }
20
21     // Method untuk mengubah fragment
22     fun changeFragment(fragment: Fragment?) {
23         if (fragment != null) {
24             getSupportFragmentManager()
25                 .beginTransaction()
26                 .replace(R.id.layout_fragment, fragment)
27                 .commit()
28         }
29     }
30 }
```

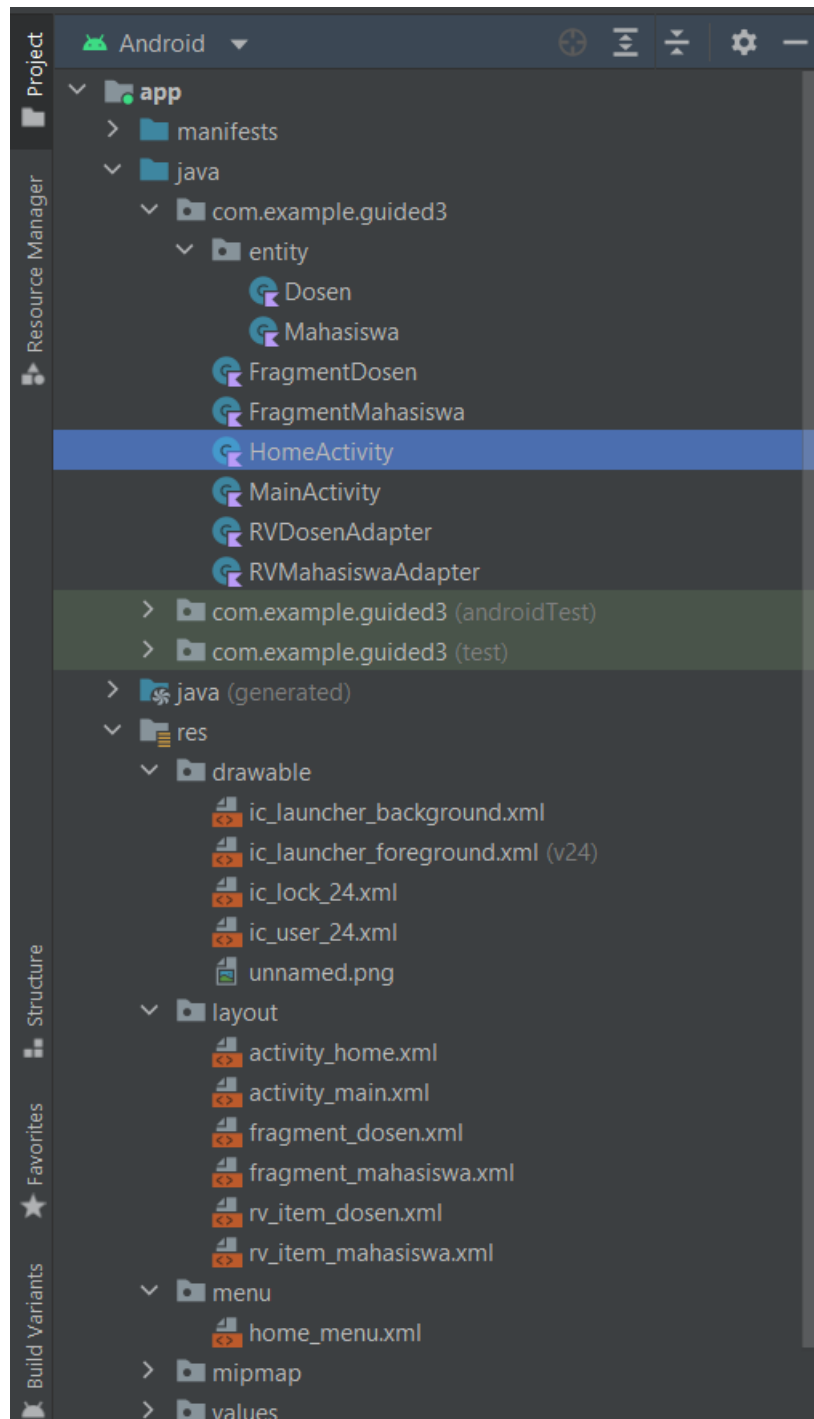
```

30
31 override fun onCreateOptionsMenu(menu: Menu): Boolean {
32     // Disini kita menghubungkan menu yang telah kita buat dengan activity ini
33     val inflater = MenuInflater(context=this)
34     inflater.inflate(R.menu.home_menu, menu)
35     return true
36 }
37
38 override fun onOptionsItemSelected(item: MenuItem): Boolean {
39     if (item.itemId == R.id.menu_mahasiswa) {
40         // Jika menu yang dipilih adalah Mahasiswa maka ganti fragmentnya dengan FragmentMahasiswa
41         changeFragment(FragmentMahasiswa())
42     } else if (item.itemId == R.id.menu_dosen) {
43         // Jika menu yang dipilih adalah Dosen maka ganti fragmentnya dengan FragmentDosen
44         changeFragment(FragmentDosen())
45     } else {
46         // Jika menu yang dipilih adalah menu Exit, maka tampilkan sebuah dialog
47         val builder: AlertDialog.Builder = AlertDialog.Builder(context=this@HomeActivity)
48         builder.setMessage("Are you sure want to exit?")
49         .setPositiveButton(text="YES", object: DialogInterface.OnClickListener {
50             override fun onClick(dialogInterface: DialogInterface, i: Int) {
51                 // Keluar dari aplikasi
52                 finishAndRemoveTask()
53             }
54         })
55         .show()
56     }
57     return super.onOptionsItemSelected(item)
58 }
59 }
  
```

Gambar Kode HomeActivity

(Finishing)

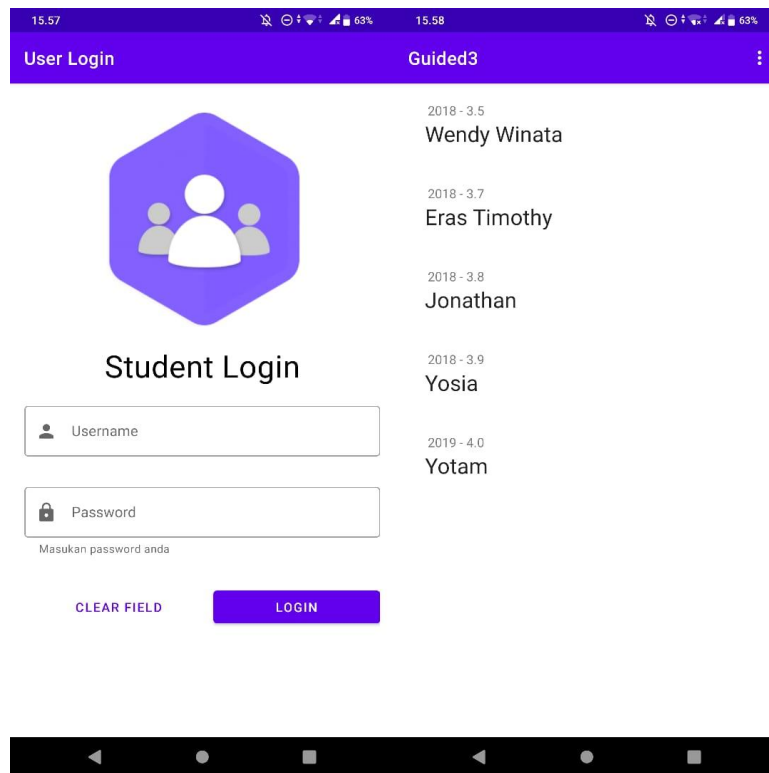
37. Proyek telah selesai. Berikut adalah semua file yang ada di sidebar dalam proyek.



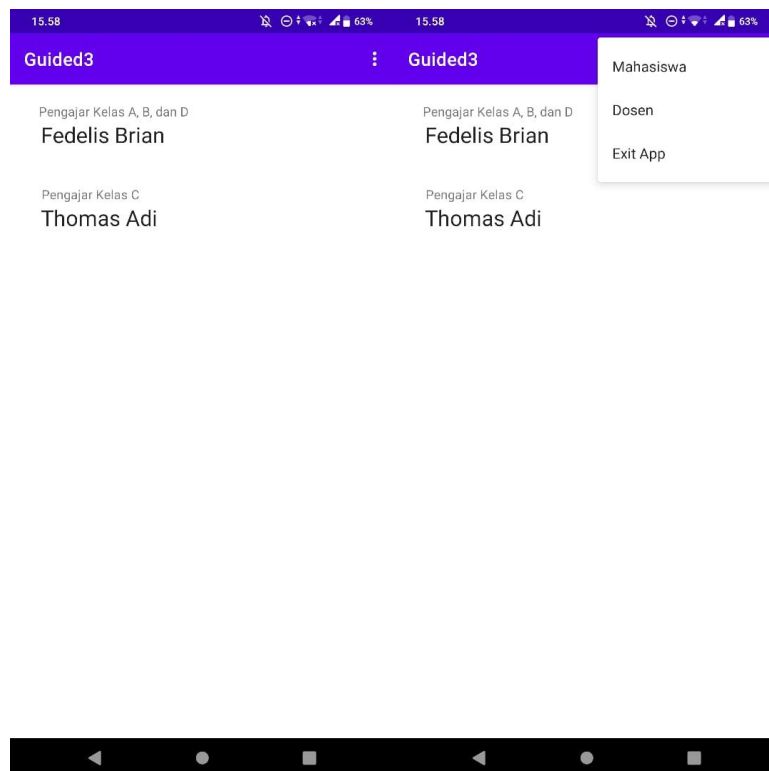
Gambar Kode Sidebar Proyek Guided

38. Pastikan proyek dapat berjalan dengan baik dan tidak ada error sama sekali.
39. Jika terdapat error, silahkan lakukan googling terlebih dahulu untuk belajar bagaimana cara menyelesaikan masalah.
40. Selalu perhatikan comment yang ada pada kode tersebut, agar kalian dapat mengerti apa yang terjadi pada kode tersebut.
41. Jika masih kebingungan, silahkan tanya kakak-kakak asdos atau dosen masing-masing.
42. Semoga beruntung 😊.

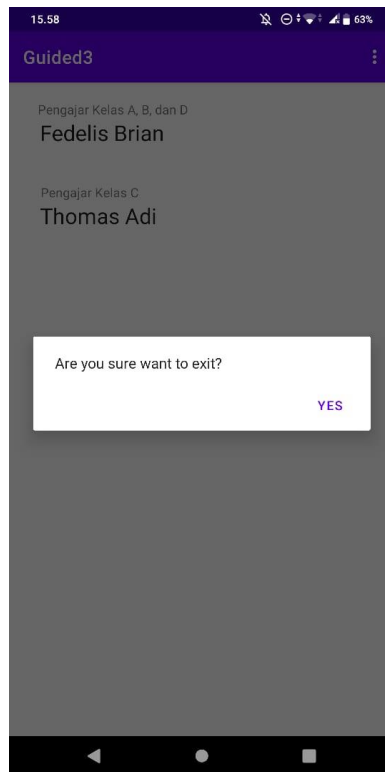
(Tampilan Aplikasinya)



Gambar Tampilan Login dan Menu Mahasiswa



Gambar Tampilan Menu Dosen dan Pilihan Menu-Menu



Gambar Tampilan alertdialog untuk logout