

LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN MOBILE



Oleh:

Muhammad Ammarin Ihsan NIM. 2010817210002

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
2022**

LEMBAR PENGESAHAN

LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN MOBILE

Laporan Akhir Praktikum Pemrograman Mobile ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Muhammad Ammarin Ihsan
NIM : 2010817210002

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Rezi Rahdianor
NIM. 1810817210019

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 199307032019031011

DAFTAR ISI

LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN MOBILE	0
LEMBAR PENGESAHAN	1
DAFTAR ISI	2
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
MODUL 1 KOTLIN BASIC	6
SOAL 1	7
A. Source Code.....	7
B. Output Program.....	8
C. Pembahasan	8
D. Tautan Git	8
SOAL 2.....	9
A. Source Code.....	9
B. Output Program.....	9
C. Pembahasan	10
D. Tautan Git	10
SOAL 3.....	11
A. Source Code.....	11
B. Output Program.....	11
C. Pembahasan	12
D. Tautan Git	12
SOAL 4.....	13
A. Source Code.....	13
B. Output Program.....	13
C. Pembahasan	14
D. Tautan Git	14
MODUL 2 ANDROID BASICS IN KOTLIN	15
SOAL.....	16
A. Source Code.....	19
B. Pembahasan	20

C. Hasil Screenshot	21
D. Tautan Git	22
MODUL 3 ANDROID LAYOUT.....	23
SOAL.....	24
A. Source Code.....	27
B. Pembahasan	28
C. Hasil Screenshot	29
D. TAUTAN GIT	30
MODUL 4 NAVIGATION	31
SOAL.....	32
A. Sourcecode.....	33
B. Pembahasan	47
C. Hasil Screenshot	48
D. Tautan Github	52
MODUL 5 MENGAMBIL DATA DARI INTERNET	53
SOAL.....	54
A. Source Code.....	54
B. Pembahasan	60
C. Hasil Screenshot	61
D. Tautan Git	62

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Modul 1 Soal 1	8
Gambar 2. Screenshot Hasil Jawaban Modul 1 Soal 2.1	9
Gambar 3. Screenshot Hasil Jawaban Modul 1 Soal 2.2	9
Gambar 4. Screenshot Hasil Jawaban Modul 1 Soal 3.1	11
Gambar 5. Screenshot Hasil Jawaban Modul 1 Soal 3.2	12
Gambar 6. Screenshot Hasil Jawaban Modul 1 Soal 4	13
Gambar 7. Tampilan Awal Aplikasi Modul 2	16
Gambar 8. Tampilan Dadu Setelah Di Roll Modul 2	17
Gambar 9. Tampilan Roll Dadu Double Modul 2	18
Gambar 10. Screenshot Tampilan Roll Dadu Double Modul 2	22
<i>Gambar 11. Tampilan Awal Aplikasi Modul 3</i>	<i>24</i>
<i>Gambar 12. Tampilan User Setelah Memilih Mata Uang dan Mengisi Nilainya Modul 3</i>	<i>25</i>
<i>Gambar 13. Hasil Konversi Mata Uang ke Rupiah Modul 3</i>	<i>26</i>
Gambar 14. Material Palette	26
Gambar 15. Hasil Tampilan Awal Aplikasi Modul 3	29
Gambar 16. Hasil Tampilan User Setelah Memilih Mata Uang dan Mengisi Nilainya Modul 3	29
Gambar 17. Hasil Konversi Mata Uang ke Rupiah Modul 3	30
Gambar 18. Screenshot Daftar Sourcecode Modul 4	47
Gambar 19. Screenshot Modul 4 Masuk Aplikasi Mode Light & Dark	48
Gambar 20. Tampilan Awal Modul 4 Mode Light & Dark	49
Gambar 21. Screenshot ketika mengklik item dan masuk ke menu detail home	50
Gambar 22. Tampilan Modul 4 Navigation Drawer	51
Gambar 23. Tampilan Detail Gallery Modul 4	52
Gambar 24. Tampilan Gallery Modul 4	52
Gambar 25. Source Code Modul 5 MainActivity.kt	54
Gambar 26. Source Code Modul 5 BindingAdapter.kt	55

DAFTAR TABEL

Table 1. Ketentuan Input dan Output Modul 1 Soal 1	7
Table 2. Source Code Modul 1 PRAK101.kt	7
Table 3. Ketentuan Input dan Output Modul 1 Soal 2	9
Table 4. Source Code Modul 1 PRAK102.kt	9
Table 5. Ketentuan Input dan Output Modul 1 Soal 3	11
Table 6. Source Code Modul 1PRAK103.kt	11
Table 7. Source Code Modul 1 PRAK104.kt	13
Table 8. Sourcecode Modul 2 MainActivity.kt	20
Table 9. Source Code Modul 3 MainActivity.kt	28
Table 10. Source Code Modul 4 MainActivity.kt	34
Table 11. Source Code Modul 4 SplashActivity.kt	34
Table 12. Source Code Modul 4 GallerData.kt	35
Table 13. Source Code Modul 4 HomeData.kt	35
Table 14. Source Code Modul 4 NewsData.kt	35
Table 15. Source Code Modul 4 GalleryAdapter.kt	36
Table 16. Source Code Modul 4 HomeAdapter.kt	37
Table 17. Source Code Modul 4 NewsAdapter.kt	39
Table 18. Source Code Modul 4 GalleryDetailActivity	39
Table 19. Source Code GalleryFragment.kt	40
Table 20. Source Code GalleryViewModel.kt	40
Table 21. Source Code Modul 4 HomeDetailActivity.kt	41
Table 22. Source Code Modul 4 HomeFragment.kt	42
Table 23. Source Code Modul 4 HomeViewModel.kt	44
Table 24. Source Code Modul 4 NewsDetailActivity.kt	44
Table 25. Source Code Modul 4 NewsFragment.kt	45
Table 26. Source Code Modul 4 NewsViewModel.kt	46
Table 27. Source Code Modul 5 Hololive.kt	56
Table 28. Source Code Modul 5 HololistViewModel	57
Table 29. Source Code Modul 5 HololistFragment.kt	57
Table 30. Source Code Modul 5 HololistAdapter.kt	59
Table 31. Source Code Modul 5 HoloDetailFragment.kt	59
Table 32. Daftar Source Code Modul 5	60
Table 33. Tampilan Awal Modul 5	61
Table 34. Tampilan Detail Hololist Modul 5	62

MODUL 1 KOTLIN BASIC



Oleh:

Muhammad Ammarin Ihsan NIM. 2010817210002

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MARET 2022**

SOAL 1

Buatlah sebuah program yang menerima input berupa waktu (Pagi, Siang, Sore, Malam), nama anda, umur anda, dan suhu tubuh anda (boleh random di range 35-37) kemudian mengeluarkan output sebagaimana dicontohkan dibawah ini

Input	Output
Waktu Sekarang: Pagi Nama Anda: Muhammad Alkaff Umur Anda: 35 Suhu Tubuh Anda: 36.4	Selamat Pagi, Muhammad Alkaff. Umur anda 35 Tahun. Suhu Tubuh anda 36.4 derajat Celcius.

Table 1. Ketentuan Input dan Output Modul 1 Soal 1

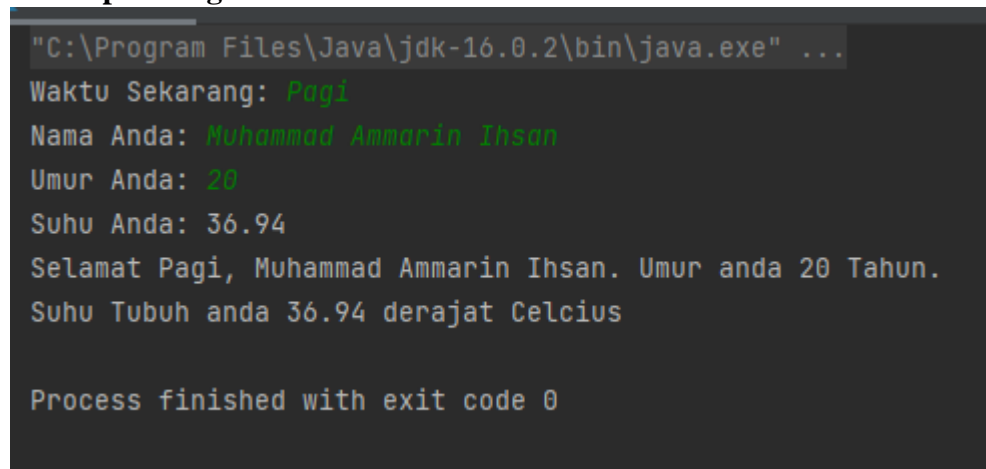
Simpan dengan nama file: PRAK101.kt dan push ke repository kalian masing-masing.

A. Source Code

1	package modul1
2	
3	import kotlin.random.Random
4	
5	fun main(){
6	print("Waktu Sekarang: ")
7	val waktu = readLine()
8	print("Nama Anda: ")
9	val nama = readLine()
10	print("Umur Anda: ")
11	val umur = readLine()
12	var suhu = String.format("%.2f",
13	Random.nextDouble(35.00, 37.00))
14	print("Suhu Anda: \$suhu")
15	
16	println()
17	println("Selamat Pagi, \${nama}. Umur anda \${umur}
18	Tahun.")
19	println("Suhu Tubuh anda \${suhu} derajat Celcius")
20	}

Table 2. Source Code Modul 1 PRAK101.kt

B. Output Program

A screenshot of a terminal window showing the output of a Java program. The window title is '"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...'. The output text is: 'Waktu Sekarang: Pagi', 'Nama Anda: Muhammad Ammarin Ihsan', 'Umur Anda: 20', 'Suhu Anda: 36.94', 'Selamat Pagi, Muhammad Ammarin Ihsan. Umur anda 20 Tahun.', 'Suhu Tubuh anda 36.94 derajat Celcius', and 'Process finished with exit code 0'.

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...  
Waktu Sekarang: Pagi  
Nama Anda: Muhammad Ammarin Ihsan  
Umur Anda: 20  
Suhu Anda: 36.94  
Selamat Pagi, Muhammad Ammarin Ihsan. Umur anda 20 Tahun.  
Suhu Tubuh anda 36.94 derajat Celcius  
  
Process finished with exit code 0
```

Gambar 1. Screenshot Hasil Jawaban Modul 1 Soal 1

C. Pembahasan

Pada baris [1] merupakan lokasi file,

Pada baris [3] mengimport fungsi random milik kotlin

Pada baris [5] adalah fungsi main, baris [6] adalah cetak teks tanpa membuat baris baru.

Pada baris [7] merupakan variabel yang berisikan input dari user.

Pada baris [12] merandom suhu antara 35 – 37 ke dalam tipe double

Baris [16] mencetak garis baru yang kosong(stringnya)

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/FollowID/praktikummobile2/tree/main/modul1>

SOAL 2

Buatlah sebuah program dengan menerapkan function (fungsi) untuk dapat menerjemahkan fungsi matematika dibawah ini kedalam sebuah kode program.

$$f(x) = 2x^2 + 5x - 8$$

Input	Output
Nilai x = 2	10
Input	Output
Nilai x = 4	44

Table 3. Ketentuan Input dan Output Modul 1 Soal 2

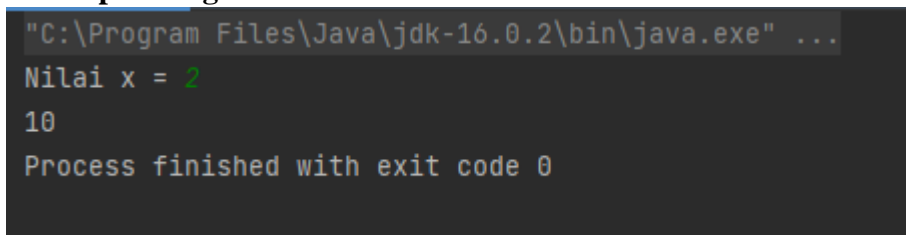
Simpan dengan nama file: PRAK102.kt dan push ke repository kalian masing-masing.

A. Source Code

```
1 package modul1
2
3 fun main(){
4     print("Nilai x = ")
5     val x = Integer.valueOf(readLine())
6     hitung(x)
7 }
8
9 fun hitung(x: Int){
10     val hasil = 2 * (x * x) + 5 * x - 8
11     print(hasil)
12 }
```

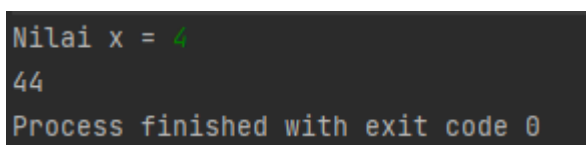
Table 4. Source Code Modul 1 PRAK102.kt

B. Output Program



```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Nilai x = 2
10
Process finished with exit code 0
```

Gambar 2. Screenshot Hasil Jawaban Modul 1 Soal 2.1



```
Nilai x = 4
44
Process finished with exit code 0
```

Gambar 3. Screenshot Hasil Jawaban Modul 1 Soal 2.2

C. Pembahasan

Pada baris [1] merupakan lokasi file, baris [3] merupakan fungsi main.

Pada baris [5] memasukkan input user bertipe integer ke dalam variabel x.

Baris [6] memanggil fungsi hitung() dengan memasukkan variable x

Pada baris [9] fungsi hitung dengan mengambil nilai x bertipe int

Baris [10] memasukkan perhitungan ke dalam variable hasil

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/FollowID/praktikummobile2/tree/main/modul1>

SOAL 3

Buatlah sebuah program yang dapat menampilkan 5 bilangan kelipatan dari bilangan yang telah diinput dan habis dibagi 2 atau 3.

Input	Output
Input bilangan = 5	10 15 20 30 40
Input	Output
Nilai bilangan = 7	14 21 28 42 56

Table 5. Ketentuan Input dan Output Modul 1 Soal 3

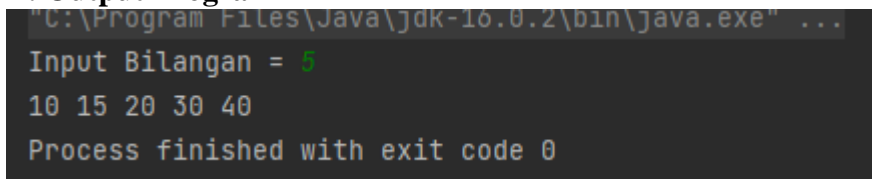
Simpan dengan nama file: PRAK103. kt dan push ke repository kalian masing-masing.

A. Source Code

```
1 package modul1
2
3 fun main(){
4     print("Input Bilangan = ")
5     var bilangan = Integer.valueOf(readLine())
6     var kelipatan = bilangan
7     var banyakOutput = 5
8     var i = 0
9
10    while (i < banyakOutput){
11        bilangan += kelipatan
12        if (bilangan %2 == 0 || bilangan %3 == 0){
13            print("${bilangan} ")
14        }else {
15            banyakOutput++
16        }
17        i++
18    }
19 }
```

Table 6. Source Code Modul 1PRAK103.kt

B. Output Program



```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Input Bilangan = 5
10 15 20 30 40
Process finished with exit code 0
```

Gambar 4. Screenshot Hasil Jawaban Modul 1 Soal 3.1

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...  
Input Bilangan = 7  
14 21 28 42 56  
Process finished with exit code 0
```

Gambar 5. Screenshot Hasil Jawaban Modul 1 Soal 3.2

C. Pembahasan

Pada baris [1] merupakan lokasi file, baris [3] adalah fungsi main

Pada baris [10] akan mengulang koding yang berada di dalamnya sebanyak 5x.

Baris [11] menambahkan bilangan kelipatan

Pada baris [12] akan mencetak jika angka habis dibagi 2 atau 3. Jika tidak maka akan menambah looping hingga angka dapat habis dibagi 2 atau 3.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/FulowID/praktikummobile2/tree/main/modul1>

SOAL 4

Buatlah sebuah program dengan menerapkan konsep Class dan Object pada Kotlin. Class dan Object yang dibuat disesuaikan dengan ketentuan berikut:

- NIM dengan akhiran 0/1 : Class dan Object dengan Tema **Pekerjaan**
- NIM dengan akhiran 2/3 : Class dan Object dengan Tema **Otomotif**
- NIM dengan akhiran 4/5 : Class dan Object dengan Tema **Olahraga**
- NIM dengan akhiran 6/7 : Class dan Object dengan Tema **Hewan**
- NIM dengan akhiran 8/9 : Class dan Object dengan Tema **Teknologi**

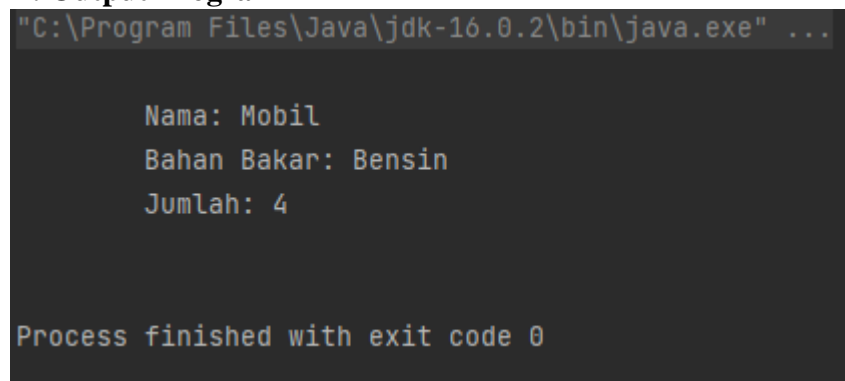
Simpan dengan nama file: PRAK104.kt dan push ke repository kalian masing-masing.

A. Source Code

```
1 package modul1
2
3 fun main(){
4     val otomotif = Otomotif()
5     otomotif.tampil()
6 }
7
8 class Otomotif(var nama: String = "Mobil", var bahanBakar:
9 String = "Bensin", var jumlah: Int = 4){
10     fun tampil(){
11         println("""
12         Nama: $nama
13         Bahan Bakar: $bahanBakar
14         Jumlah: $jumlah
15         """)
16     }
17 }
```

Table 7. Source Code Modul 1 PRAK104.kt

B. Output Program



```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...

    Nama: Mobil
    Bahan Bakar: Bensin
    Jumlah: 4

Process finished with exit code 0
```

Gambar 6. Screenshot Hasil Jawaban Modul 1 Soal 4

C. Pembahasan

Pada baris [1] merupakan lokasi file, baris [3] adalah fungsi main

Pada baris [4] menginstansiasi class otomotif

Baris [5] memanggil fungsi tampil yang ada di class Otomotif

Pada baris [8] class Otomotif dengan constructornya

Pada baris [10] adalah fungsi tampil()

Baris [11] Print dengan petik 3 artinya akan memprint semua string yang ada di dalamnya

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/FulowID/praktikummobile2/tree/main/modul1>

MODUL 2
ANDROID BASICS IN KOTLIN



Oleh:

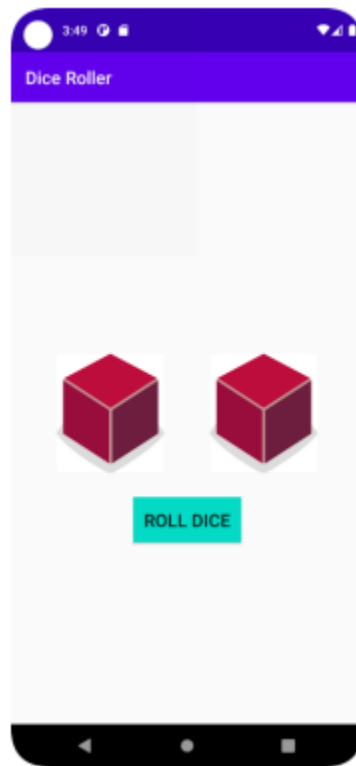
Muhammad Ammarin Ihsan NIM. 2010817210002

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MARET 2022

SOAL

Buatlah sebuah aplikasi yang dapat menampilkan 2 (dua) buah dadu yang dapat berubah-ubah tampilannya pada saat user menekan tombol “Roll Dice”. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

1. Tampilan awal aplikasi setelah dijalankan akan menampilkan 2 buah dadu kosong seperti dapat dilihat pada Gambar 1.



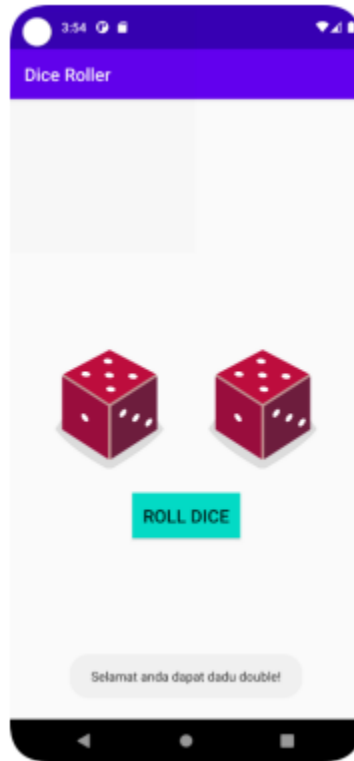
Gambar 7. Tampilan Awal Aplikasi Modul 2

2. Setelah user menekan tombol “Roll Dice” maka masing-masing dadu akan memunculkan sisi dadu masing-masing dengan angka antara 1 s/d 6. Apabila user mendapatkan nilai dadu yang berbeda antara Dadu 1 dengan Dadu 2 maka akan menampilkan pesan “Anda belum beruntung!” seperti dapat dilihat pada Gambar 2.



Gambar 8. Tampilan Dadu Setelah Di Roll Modul 2

3. Apabila user mendapatkan nilai dadu yang sama antara Dadu 1 dan Dadu 2 atau nilai double, maka aplikasi akan menampilkan pesan “Selamat anda dapat dadu double!” seperti dapat dilihat pada Gambar 3.



Gambar 9. Tampilan Roll Dadu Double Modul 2

4. Upload aplikasi yang telah anda buat kedalam repository github ke dalam folder Module 2 dalam bentuk project. Jangan lupa untuk melakukan Clean Project sebelum mengupload pekerjaan anda pada repo.

A. Source Code

```
1 package com.example.diceroller
2
3 import android.os.Bundle
4 import android.widget.Button
5 import android.widget.ImageView
6 import android.widget.Toast
7 import androidx.appcompat.app.AppCompatActivity
8
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13         val rollButton: Button = findViewById(R.id.button)
14
15         rollButton.setOnClickListener { rollDice() }
16
17         val diceImage: ImageView = findViewById(R.id.imageView)
18         val diceImage2: ImageView = findViewById(R.id.imageView2)
19
20         diceImage.setImageResource(R.drawable.dice_0)
21         diceImage2.setImageResource(R.drawable.dice_0)
22     }
23
24     private fun rollDice() {
25         val dice = Dice(6)
26         val diceRoll = dice.roll()
27         val diceRoll2 = dice.roll()
28
29         val diceImage: ImageView = findViewById(R.id.imageView)
30         val diceImage2: ImageView = findViewById(R.id.imageView2)
31
32         val drawableResources = when (diceRoll) {
33             1 -> R.drawable.dice_1
34             2 -> R.drawable.dice_2
35             2 -> R.drawable.dice_2
36             3 -> R.drawable.dice_3
37             4 -> R.drawable.dice_4
38             5 -> R.drawable.dice_5
39             else -> R.drawable.dice_6
40         }
41
42         val drawableResources2 = when (diceRoll2) {
43             1 -> R.drawable.dice_1
44             2 -> R.drawable.dice_2
45             2 -> R.drawable.dice_2
46             3 -> R.drawable.dice_3
47             4 -> R.drawable.dice_4
48             5 -> R.drawable.dice_5
49             else -> R.drawable.dice_6
50         }
51
52         diceImage.setImageResource(drawableResources)
53         diceImage2.setImageResource(drawableResources2)
```

54	
55	diceImage.contentDescription = diceRoll.toString()
56	diceImage2.contentDescription = diceRoll2.toString()
57	
58	if (diceRoll == diceRoll2){
59	Toast.makeText(applicationContext, "Selamat anda dapat
60	dadu double!", Toast.LENGTH_SHORT).show()
62	}
63	else {
64	Toast.makeText(applicationContext, "Anda belum
65	beruntung!", Toast.LENGTH_SHORT).show()
66	}
67	}
68	}
69	
70	class Dice(private val numSides: Int) {
71	fun roll(): Int {
72	return (1..numSides).random()
73	}
74	}

Table 8. Sourcecode Modul 2 MainActivity.kt

B. Pembahasan

Pada baris [13] akan mencari button dengan id “button” dan memasukkannya ke dalam variabel rollButton.

Pada baris [15] setOnClickListener berfungsi jika user menekan button tersebut dan akan menjalankan fungsi rollDice().

Pada baris [17] dan [18] mencari gambar dengan id imageView dan imageView2 ke dalam variabel.

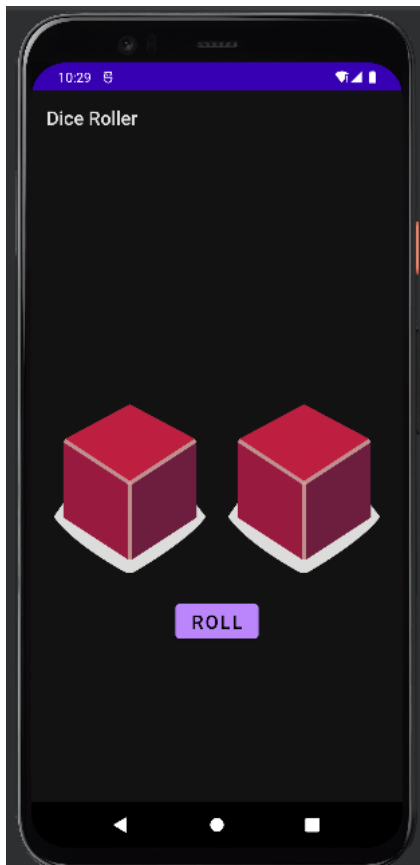
Pada baris [20] dan [21] akan menset gambar pada kedua ImageView dengan gambar dadu kosong.

Pada baris [24] adalah fungsi rollDice di sini akan dijalankan operasi pergantian gambar sesuai dengan hasil gambar dari class Dice.

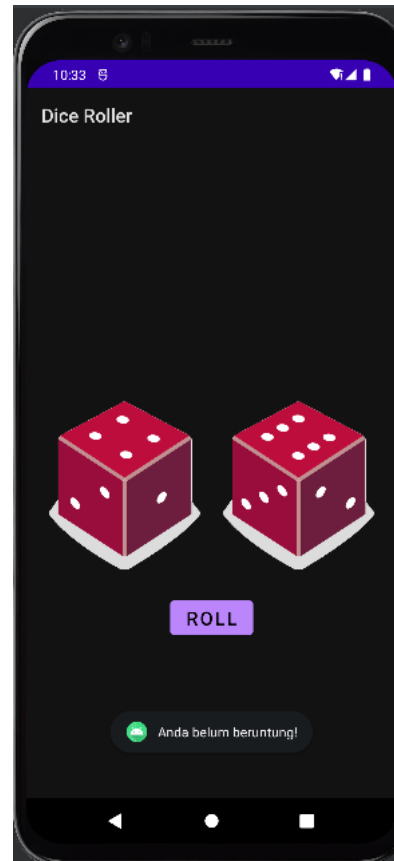
Pada baris [58] terdapat kondisi yakni jika nilai kedua dadu sama maka akan menampilkan teks yang berupa toast “Selamat anda dapat dadu double!” dan jika nilai kedua dadu tidak sama maka akan menampilkan pesan toast “Anda belum beruntung!”, toast adalah feedback sederhana atau umpan balik tentang suatu operasi dalam bentuk popup kecil.

Pada baris [70] adalah class Dice yang di dalamnya terdapat fungsi roll() dimana fungsi tersebut merandom nilai dari 1 sampai dengan nilai yang ditentukan dari bentuk dadu, pada kasus ini menggunakan dadu 6.

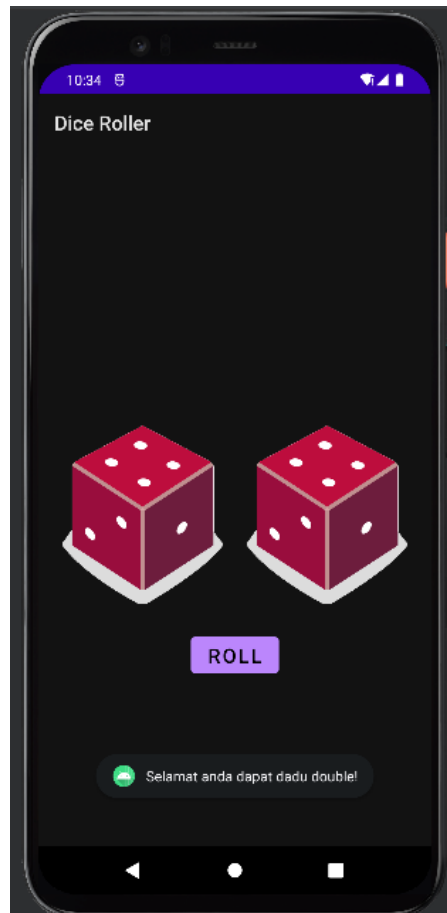
C. Hasil Screenshot



*Gambar 10. Screenshot Tampilan Awal
Aplikasi Modul 2*



*Gambar 11. Screenshot Tampilan Dadu
Setelah Di Roll Modul 2*



Gambar 10. Screenshot Tampilan Roll Dadu Double Modul 2

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/FulowID/praktikummobile2/tree/main/modul2>

MODUL 3

ANDROID LAYOUT



Oleh:

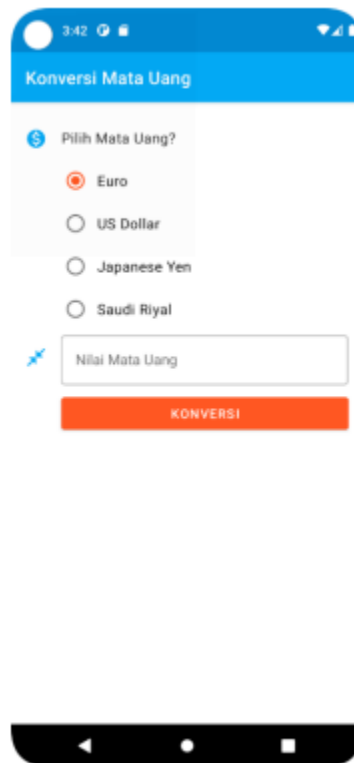
Muhammad Ammarin Ihsan NIM. 2010817210002

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2022**

SOAL

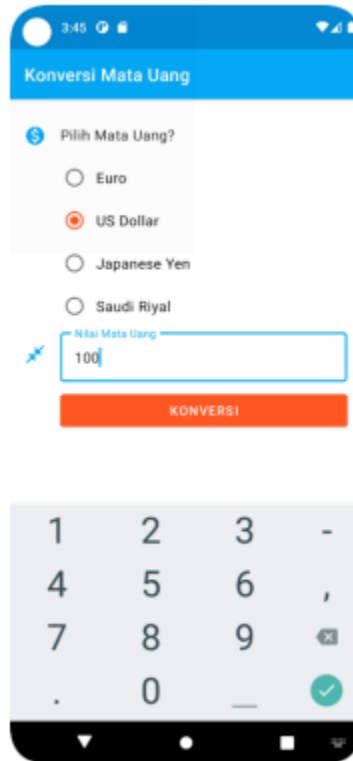
Buatlah sebuah aplikasi yang dapat melakukan konversi mata uang Rupiah terhadap 4 (empat) mata uang lainnya yaitu Euro, US Dollar, Japanese Yen, dan Saudi Riyal. Aturan aplikasi yang akan dibangun adalah sebagaimana berikut:

1. Tampilan awal aplikasi setelah dijalankan akan menunjukkan tampilan seperti dapat dilihat pada Gambar 1.



Gambar 11. Tampilan Awal Aplikasi Modul 3

2. User memilih satu diantara 4 (empat) mata uang yang disediakan yaitu Euro, US Dollar, Japanese Yen, dan Saudi Riyal untuk dikonversi nilainya ke dalam Rupiah. Kemudian user mengisi nilai mata uang pilihannya tersebut untuk dapat dikonversi ke Rupiah.



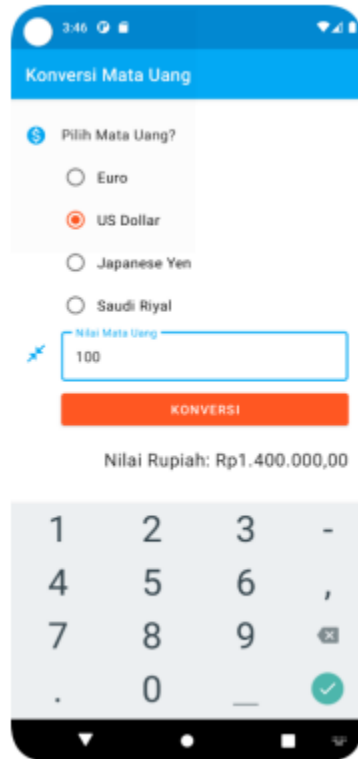
Gambar 12. Tampilan User Setelah Memilih Mata Uang dan Mengisi Nilainya Modul 3

3. User kemudian menekan tombol Konversi yang akan menghasilkan nilai konversi antara mata uang pilihannya ke dalam Rupiah seperti terlihat pada Gambar 3 di halaman selanjutnya.
4. Percantik tampilan aplikasi anda dengan menggunakan prinsip Material Design. Salah satu website yang dapat membantu mempermudah generate color sesuai dengan prinsip Material Design adalah <https://www.materialpalette.com/> seperti dapat terlihat pada Gambar 4 di halaman berikutnya.
5. Sebagai bantuan, gunakan kode dibawah ini untuk dapat mengkonversi nilai yang dihasilkan dalam format Rupiah.

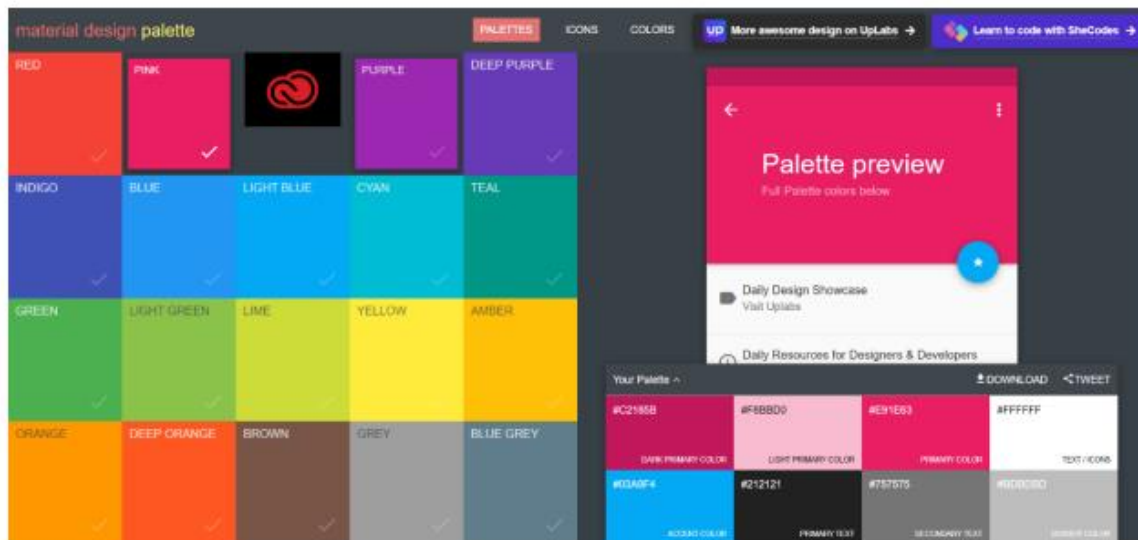
```
val indonesianLocale = Locale("in", "ID")
val formattedTip =
    NumberFormat.getCurrencyInstance(indonesianLocale).format(
        conversion)

```

6. Upload aplikasi yang telah anda buat kedalam repository github ke dalam **folder Module 3 dalam bentuk project**. Jangan lupa untuk melakukan **Clean Project** sebelum mengupload pekerjaan anda pada repo.



Gambar 13. Hasil Konversi Mata Uang ke Rupiah Modul 3



Gambar 14. Material Palette

A. Source Code

```
1 package com.example.konversimatauang
2
3 import android.content.Context
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.view.KeyEvent
7 import android.view.View
8 import android.view.inputmethod.InputMethodManager
9 import androidx.core.view.isVisible
10 import
11 com.example.konversimatauang.databinding.ActivityMainBinding
12 import java.text.NumberFormat
13 import java.util.*
14
15 class MainActivity : AppCompatActivity() {
16
17     private lateinit var idBind: ActivityMainBinding
18
19     override fun onCreate(savedInstanceState: Bundle?) {
20         super.onCreate(savedInstanceState)
21         setContentView(R.layout.activity_main)
22         idBind = ActivityMainBinding.inflate(layoutInflater)
23         setContentView(idBind.root)
24         idBind.nilaiMataUang.setOnKeyListener { view, keyCode, _
25     -> handleKeyEvent(view, keyCode) }
26
27         idBind.textRupiah.isVisible = false
28         idBind.convertButton.setOnClickListener { mengkonversi()
29     }
30     }
31
32     private fun mengkonversi() {
33         val stringInTextField =
34     idBind.nilaiMataUang.text.toString()
35         val nilai = stringInTextField.toDoubleOrNull()
36         if (nilai == null) {
37             idBind.textRupiah.text = " "
38             return
39         }
40
41         val konversi = when
42     (idBind.currencyGroup.checkedRadioButtonId) {
43             R.id.euro_button -> 15886.76
44             R.id.usd_button -> 14376.05
45             R.id.yen_button -> 117.34
46             else -> 3832.17
47         }
48
49         val rupiah = nilai * konversi
50         val indonesianLocale = Locale("in", "ID")
51         val formatRupiah =
52     NumberFormat.getCurrencyInstance(indonesianLocale).format(rupiah)
53         idBind.textRupiah.text = getString(R.string.rupiah_value,
54     formatRupiah)
```

55	
56	idBind.textRupiah.isVisible = true
57	}
58	
59	private fun handleKeyEvent(view: View, keyCode: Int): Boolean
60	{
62	if (keyCode == KeyEvent.KEYCODE_ENTER) {
63	val inputMethodManager =
64	getService(Context.INPUT_METHOD_SERVICE) as
65	InputMethodManager
66	
67	inputMethodManager.hideSoftInputFromWindow(view.windowToken, 0)
68	return true
69	}
70	return false
71	}
72	}

Table 9. Source Code Modul 3 MainActivity.kt

B. Pembahasan

[17] membuat deklarasi binding dengan nama idBind.

[24] pada teks input nilai mata uang ketika menekan keyboard maka akan mempass variabel view dan keycode ke fungsi handleKeyEvent.

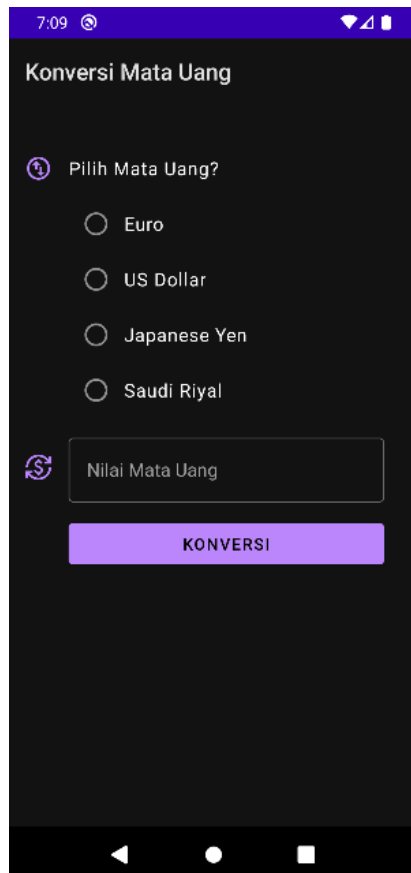
[32] function mengkonversi nilai.

[41] statement when untuk mengecek pilihan radio button yang dipilih dan nilainya tukarnya

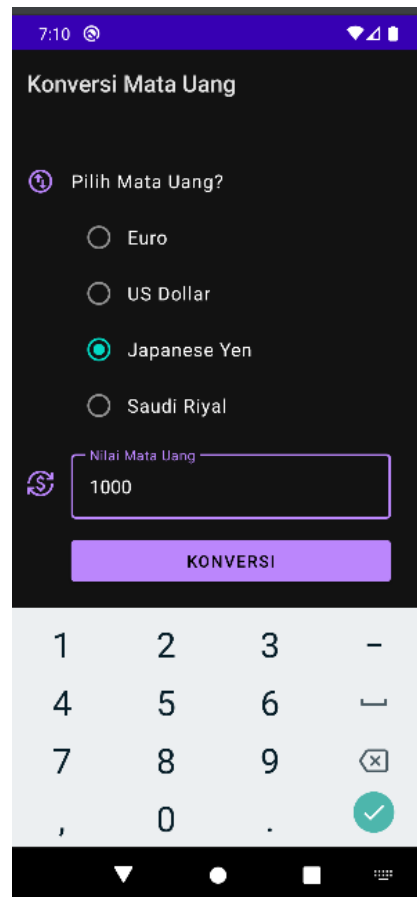
[52] memformat text menjadi format rupiah.

[59] fuction handleKeyEvent dipanggil di onCreate, berisikan kondisi jika user menekan enter pada keyboard maka secara otomatis keyboard akan dihide.

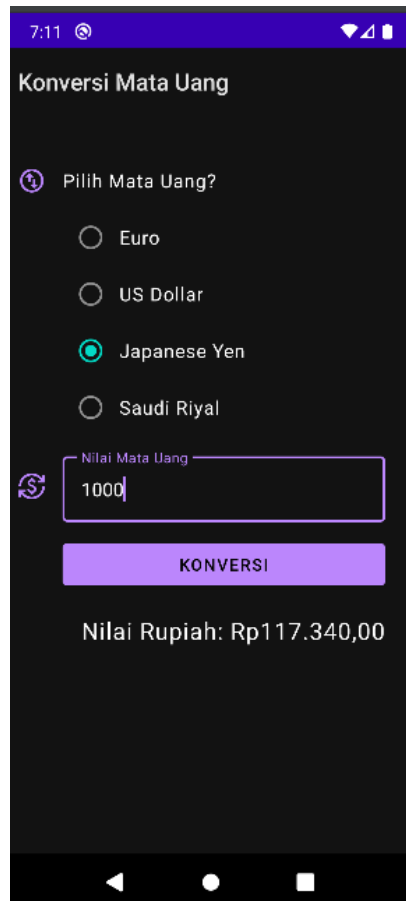
c. Hasil Screenshot



Gambar 15. Hasil Tampilan Awal Aplikasi Modul 3



Gambar 16. Hasil Tampilan User Setelah Memilih Mata Uang dan Mengisi Nilainya Modul 3



Gambar 17. Hasil Konversi Mata Uang ke Rupiah Modul 3

D. TAUTAN GIT

<https://github.com/FulowID/praktikummobile2/modul3>

MODUL 4 NAVIGATION



Oleh:

Muhammad Ammarin Ihsan NIM. 2010817210002

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2022**

SOAL

Buatlah sebuah aplikasi Android sederhana dengan spesifikasi sebagai berikut:

1. Tema aplikasi disesuaikan dengan NIM mahasiswa masing-masing:
 - NIM dengan akhiran 0/1: Class dan Object dengan Tema Pekerjaan
 - NIM dengan akhiran 2/3: Class dan Object dengan Tema Otomotif
 - NIM dengan akhiran 4/5: Class dan Object dengan Tema Olahraga
 - NIM dengan akhiran 6/7: Class dan Object dengan Tema Hewan
 - NIM dengan akhiran 8/9: Class dan Object dengan Tema Teknologi
2. Aplikasi tersebut memiliki elemen berikut:
 - Activity
 - Fragment
 - Intent
 - Navigation Drawer
 - Options Menu
 - RecyclerView
3. Tampilan awal aplikasi menampilkan TextView dan ImageView dengan ucapan selamat datang (format bebas)
4. Menu Navigasi memiliki minimal 3 buah menu yang masing-masing menu dapat menampilkan Fragment yang berisi RecyclerView yang berbeda-beda sesuai dengan tema yang telah ditentukan
5. Data yang digunakan untuk masing-masing RecyclerView dapat menggunakan data statis berupa List.
6. Item pada RecyclerView dapat diklik untuk menampilkan data yang telah dipilih oleh user.
7. Aplikasi juga memiliki Options Menu yang dapat membuka halaman Settings untuk mengubah bahasa yang digunakan (locale) menjadi Bahasa Inggris atau Bahasa Indonesia.
8. Perubahan bahasa dapat diperlihatkan pada tampilan halaman selamat datang dan menu yang berubah sesuai Bahasa yang dipilih oleh user pada menu Setting.

A. Sourcecode

Main Activity.kt	
1	package com.example.mobil
2	
3	import android.content.Intent
4	import android.os.Bundle
5	import android.view.Menu
6	import android.view.MenuItem
7	import androidx.appcompat.app.AppCompatActivity
8	import androidx.drawerlayout.widget.DrawerLayout
9	import androidx.navigation.findNavController
10	import androidx.navigation.fragment.NavHostFragment
11	import androidx.navigation.ui.AppBarConfiguration
12	import androidx.navigation.ui.navigateUp
13	import androidx.navigation.ui.setupActionBarWithNavController
14	import androidx.navigation.ui.setupWithNavController
15	import com.example.mobil.databinding.ActivityMainBinding
16	import com.google.android.material.navigation.NavigationView
17	
18	
19	class MainActivity : AppCompatActivity() {
20	private lateinit var appBarConfiguration: AppBarConfiguration
21	private lateinit var binding: ActivityMainBinding
22	
23	override fun onCreate(savedInstanceState: Bundle?) {
24	super.onCreate(savedInstanceState)
25	
26	binding = ActivityMainBinding.inflate(layoutInflater)
27	setContentView(binding.root)
28	
29	setSupportActionBar(binding.appBarMain.toolbar)
30	
31	val drawerLayout: DrawerLayout = binding.drawerLayout
32	val navView: NavigationView = binding.navView
33	
34	val navHostFragment = supportFragmentManager
35	.findFragmentById(R.id.nav_host_fragment_content_main) as
36	NavHostFragment
37	val navController = navHostFragment.navController
38	appBarConfiguration = AppBarConfiguration(
39	setOf(
40	R.id.nav_home, R.id.nav_gallery, R.id.nav_news
41), drawerLayout
42)
43	setupActionBarWithNavController(navController,
44	appBarConfiguration)
45	navView.setupWithNavController(navController)
46	}
47	
48	override fun onCreateOptionsMenu(menu: Menu): Boolean {
49	menuInflater.inflate(R.menu.main, menu)
50	return true
51	}
52	

53	override fun onOptionsItemSelected(item: MenuItem): Boolean {
54	return when (item.itemId) {
55	R.id.action_settings -> {
56	startActivity(Intent(android.provider.Settings.ACTION_LOCALE_SETTINGS))
57	true
58	}
59	else -> super.onOptionsItemSelected(item)
60	}
61	}
62	
63	
64	
65	override fun onSupportNavigateUp(): Boolean {
66	val navController =
67	findNavController(R.id.nav_host_fragment_content_main)
68	return navController.navigateUp(appBarConfiguration)
69	super.onSupportNavigateUp()
70	}
71	}
72	

Table 10. Source Code Modul 4 MainActivity.kt

SplashActivity.kt	
1	package com.example.mobil
2	
3	import android.content.Intent
4	import androidx.appcompat.app.AppCompatActivity
5	import android.os.Bundle
6	import android.os.Handler
7	import android.os.Looper
8	
9	class SplashActivity : AppCompatActivity() {
10	override fun onCreate(savedInstanceState: Bundle?) {
11	super.onCreate(savedInstanceState)
12	setContentView(R.layout.activity_splash)
13	
14	actionBar?.hide()
15	Handler(Looper.getMainLooper()).postDelayed({
16	startActivity(Intent(this, MainActivity::class.java))
17	finish()
18	}, 3000)
19	}
20	}

Table 11. Source Code Modul 4 SplashActivity.kt

GalleryData.kt	
1	package com.example.mobil.model
2	
3	import androidx.annotation.DrawableRes
4	
5	data class GalleryData(
6	@DrawableRes val imageResourceId: Int
7)

Table 12. Source Code Modul 4 GallerData.kt

HomeData.kt	
1	package com.example.mobil.model
2	
3	import androidx.annotation.DrawableRes
4	import androidx.annotation.StringRes
5	
6	data class HomeData(
7	@StringRes val descriptionResourceId: Int,
8	@StringRes val dateResourceId: Int,
9	@StringRes val nameResourceId: Int,
10	@DrawableRes val imageResourceId: Int
11)

Table 13. Source Code Modul 4 HomeData.kt

NewsData.kt	
1	package com.example.mobil.model
2	
3	import androidx.annotation.DrawableRes
4	import androidx.annotation.StringRes
5	
6	data class HomeData(
7	@StringRes val descriptionResourceId: Int,
8	@StringRes val dateResourceId: Int,
9	@StringRes val nameResourceId: Int,
10	@DrawableRes val imageResourceId: Int
11)

Table 14. Source Code Modul 4 NewsData.kt

GalleryAdapter.kt	
1	package com.example.mobil.adapter
2	
3	import android.content.Context
4	import android.content.Intent
5	import android.view.LayoutInflater
6	import android.view.View
7	import android.view.ViewGroup
8	import android.widget.ImageView
9	import androidx.recyclerview.widget.RecyclerView
10	import com.example.mobil.R
11	import com.example.mobil.model.GalleryData
12	import com.example.mobil.ui.gallery.GalleryDetailActivity
13	import com.example.mobil.ui.gallery.GalleryViewModel
14	
15	class GalleryAdapter(
16	private val context: Context,
17	private val dataset: List<GalleryData>

```

18 ) : RecyclerView.Adapter<GalleryAdapter.GalleryViewHolder>() {
19     private val viewModel = GalleryViewModel()
20
21
22     class GalleryViewHolder(val view: View) :
23     RecyclerView.ViewHolder(view) {
24         val imageView: ImageView =
25         view.findViewById(R.id.item_gallery_image)
26     }
27
28     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
29     GalleryViewHolder {
30         val adapterLayout = LayoutInflater
31             .from(parent.context)
32             .inflate(R.layout.gallery_item, parent, false)
33         return GalleryViewHolder(adapterLayout)
34     }
35
36     override fun onBindViewHolder(holder: GalleryViewHolder, position:
37     Int) {
38         val item = dataset[position]
39         holder.imageView.setImageResource(item.imageResourceId)
40
41         holder.view.setOnClickListener {
42             viewModel.setData(item)
43             val intent = Intent(context,
44             GalleryDetailActivity::class.java).apply {
45                 putExtra("image", viewModel.image.value)
46             }
47             context.startActivity(intent)
48         }
49     }
50
51     override fun getItemCount() = dataset.size
52
53
54 }

```

Table 15. Source Code Modul 4 GalleryAdapter.kt

GalleryAdapter.kt	
1	package com.example.mobil.adapter
2	
3	import android.content.Context
4	import android.content.Intent
5	import android.view.LayoutInflater
6	import android.view.View
7	import android.view.ViewGroup
8	import android.widget.ImageView
9	import android.widget.TextView
10	import androidx.recyclerview.widget.RecyclerView
11	import com.example.mobil.R
12	import com.example.mobil.model.HomeData
13	import com.example.mobil.ui.home.HomeDetailActivity
14	import com.example.mobil.ui.home.HomeViewModel
15	
16	class HomeAdapter(
17	private val context: Context,

```

18     private val dataset: List<HomeData>
19 ) : RecyclerView.Adapter<HomeAdapter.HomeViewHolder>() {
20     private val viewModel = HomeViewModel()
21
22     class HomeViewHolder(val view: View) :
23 RecyclerView.ViewHolder(view) {
24         val imageView: ImageView = view.findViewById(R.id.newsimage)
25         val nameText: TextView = view.findViewById(R.id.newsTitle)
26         val dateText: TextView = view.findViewById(R.id.newsDesc)
27         val descText: TextView = view.findViewById(R.id.homeTextDesc)
28     }
29
30     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
31 HomeViewHolder {
32         val adapterLayout = LayoutInflater
33             .from(parent.context)
34             .inflate(R.layout.home_item, parent, false)
35         return HomeViewHolder(adapterLayout)
36     }
37
38     override fun onBindViewHolder(holder: HomeViewHolder, position:
39 Int) {
40         val item = dataset[position]
41         holder.nameText.text =
42 context.resources.getString(item.nameResourceId)
43         holder.dateText.text =
44 context.resources.getString(item.dateResourceId)
45         holder.imageView.setImageResource(item.imageResourceId)
46         holder.descText.text =
47 context.resources.getString(item.descriptionResourceId)
48
49         holder.view.setOnClickListener {
50             viewModel.setData(item, context)
51             val intent = Intent(context,
52 HomeDetailActivity::class.java).apply {
53                 putExtra("image", viewModel.image.value)
54                 putExtra("name", viewModel.name.value)
55                 putExtra("date", viewModel.date.value)
56                 putExtra("desc", viewModel.desc.value)
57             }
58             context.startActivity(intent)
59         }
60     }
61
62     override fun getItemCount() = dataset.size
63
64 }

```

Table 16. Source Code Modul 4 HomeAdapter.kt

NewsAdapter.kt

```
1 package com.example.mobil.adapter
2
3 import com.example.mobil.ui.news.NewsDetailActivity
4 import com.example.mobil.ui.news.NewsViewModel
5 import android.content.Context
6 import android.content.Intent
7 import android.view.LayoutInflater
8 import android.view.View
9 import android.view.ViewGroup
10 import android.widget.ImageView
11 import android.widget.TextView
12 import androidx.recyclerview.widget.RecyclerView
13 import com.example.mobil.R
14 import com.example.mobil.model.NewsData
15
16 class NewsAdapter(
17     private val context: Context,
18     private val dataset: List<NewsData>
19 ) : RecyclerView.Adapter<NewsAdapter.NewsViewHolder>() {
20     private val viewModel = NewsViewModel()
21
22     class NewsViewHolder(val view: View) :
23     RecyclerView.ViewHolder(view) {
24         val imageView: ImageView = view.findViewById(R.id.newsimage)
25         val nameText: TextView = view.findViewById(R.id.newsTitle)
26         val descText: TextView = view.findViewById(R.id.newsDesc)
27     }
28
29     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
30     NewsViewHolder {
31         val adapterLayout = LayoutInflater
32             .from(parent.context)
33             .inflate(R.layout.news_item, parent, false)
34         return NewsViewHolder(adapterLayout)
35     }
36
37     override fun onBindViewHolder(holder: NewsViewHolder, position:
38     Int) {
39         val item = dataset[position]
40         holder.nameText.text = context.resources.getString(item.title)
41         holder.imageView.setImageResource(item.image)
42         holder.descText.text =
43         context.resources.getString(item.description)
44
45         holder.view.setOnClickListener {
46             viewModel.setData(item, context)
47             val intent = Intent(context,
48             NewsDetailActivity::class.java).apply {
49                 putExtra("image", viewModel.image.value)
50                 putExtra("name", viewModel.name.value)
51                 putExtra("desc", viewModel.desc.value)
52             }
53             context.startActivity(intent)
54         }
55     }
56 }
```

57	override fun getItemCount() = dataset.size
58	
59	}

Table 17. Source Code Modul 4 NewsAdapter.kt

GalleryDetailActivity.kt	
1	package com.example.mobil.ui.gallery
2	
3	import android.os.Bundle
4	import androidx.appcompat.app.AppCompatActivity
5	import com.example.mobil.databinding.ActivityGalleryDetailBinding
6	
7	class GalleryDetailActivity : AppCompatActivity() {
8	private var _binding: ActivityGalleryDetailBinding? = null
9	private val binding get() = _binding!!
10	
11	companion object {
12	const val EXTRA_IMAGE = "image"
13	}
14	
15	override fun onCreate(savedInstanceState: Bundle?) {
16	super.onCreate(savedInstanceState)
17	
18	_binding = ActivityGalleryDetailBinding.inflate(layoutInflater)
19	val view = binding.root
20	setContentView(view)
21	
22	val image = binding.detailImage
23	image.setImageResource(intent.getIntExtra(EXTRA_IMAGE, 0))
24	}
25	}

Table 18. Source Code Modul 4 GalleryDetailActivity

GalleryFragment.kt	
1	package com.example.mobil.ui.gallery
2	
3	import android.os.Bundle
4	import android.view.LayoutInflater
5	import android.view.View
6	import android.view.ViewGroup
7	import androidx.fragment.app.Fragment
8	import androidx.recyclerview.widget.RecyclerView
9	import com.example.mobil.R
10	import com.example.mobil.adapter.GalleryAdapter
11	import com.example.mobil.databinding.FragmentGalleryBinding
12	
13	class GalleryFragment : Fragment(R.layout.fragment_gallery) {
14	
15	private var _binding: FragmentGalleryBinding? = null
16	private val binding get() = _binding!!
17	private lateinit var recyclerView: RecyclerView
18	
19	override fun onCreateView(
20	inflater: LayoutInflater,
21	container: ViewGroup?,
22	savedInstanceState: Bundle?
23): View {


```

24         _binding = FragmentGalleryBinding.inflate(inflater, container,
25 false)
26         val view: View = binding.root
27         return view
28     }
29
30     override fun onCreateView(view: View, savedInstanceState: Bundle?)
31 {
32         recyclerView = binding.galleryRV
33         recyclerView.adapter = GalleryAdapter(requireContext(),
34 GalleryViewModel().loadData())
35     }
36
37     override fun onDestroyView() {
38         super.onDestroyView()
39         _binding = null
40     }
41 }

```

Table 19. Source Code GalleryFragment.kt

GalleryViewModel.kt	
1	package com.example.mobil.ui.gallery
2	
3	import androidx.lifecycle.LiveData
4	import androidx.lifecycle.MutableLiveData
5	import androidx.lifecycle.ViewModel
6	import com.example.mobil.R
7	import com.example.mobil.model.GalleryData
8	
9	class GalleryViewModel : ViewModel() {
10	
11	private val _image = MutableLiveData<Int>()
12	val image: LiveData<Int> = _image
13	
14	fun loadData(): List<GalleryData> {
15	return listOf(
16	GalleryData(R.drawable.vehicle01),
17	GalleryData(R.drawable.vehicle02),
18	GalleryData(R.drawable.vehicle03),
19	GalleryData(R.drawable.vehicle04),
20	GalleryData(R.drawable.vehicle05),
21	GalleryData(R.drawable.vehicle06),
22	GalleryData(R.drawable.vehicle07),
23	GalleryData(R.drawable.vehicle08),
24	GalleryData(R.drawable.vehicle09),
25	GalleryData(R.drawable.vehicle10)
26)
27	}
28	
29	fun setData(data: GalleryData) {
30	_image.value = data.imageResourceId
31	}
32	}

Table 20. Source Code GalleryViewModel.kt

HomeDetailActivity.kt	
1	package com.example.mobil.ui.home
2	
3	import android.os.Bundle
4	import androidx.appcompat.app.AppCompatActivity
5	import com.example.mobil.databinding.ActivityHomeDetailBinding
6	
7	class HomeDetailActivity : AppCompatActivity() {
8	private var _binding: ActivityHomeDetailBinding? = null
9	private val binding get() = _binding!!
10	
11	companion object {
12	const val EXTRA_IMAGE = "image"
13	const val EXTRA_NAME = "name"
14	const val EXTRA_DATE = "date"
15	const val EXTRA_DESCRIPTION = "description"
16	}
17	
18	override fun onCreate(savedInstanceState: Bundle?) {
19	super.onCreate(savedInstanceState)
20	
21	_binding = ActivityHomeDetailBinding.inflate(layoutInflater)
22	val view = binding.root
23	setContentView(view)
24	
25	val image = binding.imageHomeDetail
26	image.setImageResource(intent.getIntExtra(EXTRA_IMAGE, 0))
27	
28	val name = binding.nameHomeDetail
29	name.text = intent.getStringExtra(EXTRA_NAME)
30	
31	val date = binding.dateHomeDetail
32	date.text = intent.getStringExtra(EXTRA_DATE)
33	
34	val description = binding.descHomeDetail
35	description.text = intent.getStringExtra(EXTRA_DESCRIPTION)
36	}
37	}

Table 21. Source Code Modul 4 HomeDetailActivity.kt

HomeFragment.kt	
1	package com.example.mobil.ui.home
2	
3	import android.os.Bundle
4	import android.view.LayoutInflater
5	import android.view.View
6	import android.view.ViewGroup
7	import androidx.fragment.app.Fragment
8	import androidx.recyclerview.widget.RecyclerView
9	import com.example.mobil.adapter.HomeAdapter
10	import com.example.mobil.databinding.FragmentHomeBinding
11	
12	class HomeFragment : Fragment() {
13	
14	private var _binding: FragmentHomeBinding? = null
15	private val binding get() = _binding!!

16	private lateinit var recyclerView: RecyclerView
17	
18	override fun onCreateView(
19	inflater: LayoutInflater,
20	container: ViewGroup?,
21	savedInstanceState: Bundle?
22): View {
23	
24	_binding = FragmentHomeBinding.inflate(inflater, container,
25	false)
26	val root: View = binding.root
27	return root
28	}
29	
30	override fun onViewCreated(view: View, savedInstanceState: Bundle?)
31	{
32	recyclerView = binding.homeRV
33	recyclerView.adapter = HomeAdapter(requireContext(),
34	HomeViewModel().loadData())
35	}
36	
37	override fun onDestroyView() {
38	super.onDestroyView()
39	_binding = null
40	}
41	}

Table 22. Source Code Modul 4 HomeFragment.kt

HomeViewModel.kt	
1	package com.example.mobil.ui.home
2	
3	import android.content.Context
4	import androidx.lifecycle.LiveData
5	import androidx.lifecycle.MutableLiveData
6	import androidx.lifecycle.ViewModel
7	import com.example.mobil.R
8	import com.example.mobil.model.HomeData
9	
10	class HomeViewModel : ViewModel() {
11	private val _image = MutableLiveData<Int>()
12	val image: LiveData<Int> = _image
13	
14	private val _name = MutableLiveData<String>()
15	val name: LiveData<String> = _name
16	
17	private val _date = MutableLiveData<String>()
18	val date: LiveData<String> = _date
19	
20	private val _desc = MutableLiveData<String>()
21	val desc: LiveData<String> = _desc
22	
23	fun loadData(): List<HomeData> {
24	return listOf(
25	HomeData(
26	R.string.desc_vehicle01,
27	R.string.date_vehicle01,
28	R.string.name_vehicle01,

```
29         R.drawable.vehicle01
30     ),
31     HomeData (
32         R.string.desc_vehicle02,
33         R.string.date_vehicle02,
34         R.string.name_vehicle02,
35         R.drawable.vehicle02
36     ),
37     HomeData (
38         R.string.desc_vehicle03,
39         R.string.date_vehicle03,
40         R.string.name_vehicle03,
41         R.drawable.vehicle03
42     ),
43     HomeData (
44         R.string.desc_vehicle04,
45         R.string.date_vehicle04,
46         R.string.name_vehicle04,
47         R.drawable.vehicle04
48     ),
49     HomeData (
50         R.string.desc_vehicle05,
51         R.string.date_vehicle05,
52         R.string.name_vehicle05,
53         R.drawable.vehicle05
54     ),
55     HomeData (
56         R.string.desc_vehicle06,
57         R.string.date_vehicle06,
58         R.string.name_vehicle06,
59         R.drawable.vehicle06
60     ),
62     HomeData (
63         R.string.desc_vehicle07,
64         R.string.date_vehicle07,
65         R.string.name_vehicle07,
66         R.drawable.vehicle07
67     ),
68     HomeData (
69         R.string.desc_vehicle08,
70         R.string.date_vehicle08,
71         R.string.name_vehicle08,
72         R.drawable.vehicle08
73     ),
74     HomeData (
75         R.string.desc_vehicle09,
76         R.string.date_vehicle09,
77         R.string.name_vehicle09,
78         R.drawable.vehicle09
79     ),
80     HomeData (
81         R.string.desc_vehicle10,
82         R.string.date_vehicle10,
83         R.string.name_vehicle10,
84         R.drawable.vehicle10
85     )
86 )
```

87	}
88	
89	fun setData(data: HomeData, context: Context) {
90	_name.value = context.getString(data.nameResourceId)
91	_image.value = data.imageResourceId
92	_date.value = context.getString(data.dateResourceId)
93	_desc.value = context.getString(data.descriptionResourceId)
94	}
95	}

Table 23. Source Code Modul 4 HomeViewModel.kt

NewsDetailActivity.kt	
1	package com.example.mobil.ui.news
2	
3	import android.os.Bundle
4	import androidx.appcompat.app.AppCompatActivity
5	import com.example.mobil.databinding.ActivityNewsDetailBinding
6	
7	class NewsDetailActivity : AppCompatActivity() {
8	private var _binding: ActivityNewsDetailBinding? = null
9	private val binding get() = _binding!!
10	
11	companion object {
12	const val EXTRA_IMAGE = "image"
13	const val EXTRA_NAME = "name"
14	const val EXTRA_DESCRIPTION = "description"
15	}
16	
17	override fun onCreate(savedInstanceState: Bundle?) {
18	super.onCreate(savedInstanceState)
19	
20	_binding = ActivityNewsDetailBinding.inflate(layoutInflater)
21	val view = binding.root
22	setContentView(view)
23	
24	val image = binding.imageNewsDetail
25	image.setImageResource(intent.getIntExtra(EXTRA_IMAGE, 0))
26	
27	val name = binding.titleDetailNews
28	name.text = intent.getStringExtra(EXTRA_NAME)
29	
30	val description = binding.descriptionDetailNews
31	description.text = intent.getStringExtra(EXTRA_DESCRIPTION)
32	}
33	}

Table 24. Source Code Modul 4 NewsDetailActivity.kt

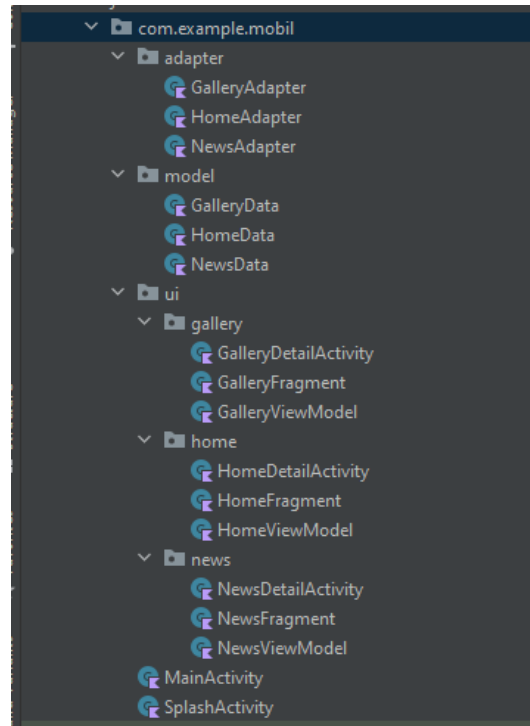
NewsFragment.kt	
1	package com.example.mobil.ui.news
2	
3	import android.os.Bundle
4	import android.view.LayoutInflater
5	import android.view.View
6	import android.view.ViewGroup
7	import androidx.fragment.app.Fragment
8	import androidx.recyclerview.widget.RecyclerView
9	import com.example.mobil.adapter.NewsAdapter
10	import com.example.mobil.databinding.FragmentNewsBinding
11	
12	class NewsFragment : Fragment() {
13	
14	private var _binding: FragmentNewsBinding? = null
15	private val binding get() = _binding!!
16	private lateinit var recyclerView: RecyclerView
17	
18	override fun onCreateView(
19	inflater: LayoutInflater,
20	container: ViewGroup?,
21	savedInstanceState: Bundle?
22): View {
23	
24	_binding = FragmentNewsBinding.inflate(inflater, container,
25	false)
26	val root: View = binding.root
27	return root
28	}
29	
30	override fun onViewCreated(view: View, savedInstanceState: Bundle?)
31	{
32	recyclerView = binding.newsRV
33	recyclerView.adapter = NewsAdapter(requireContext(),
34	NewsViewModel().loadData())
35	}
36	
37	override fun onDestroyView() {
38	super.onDestroyView()
39	_binding = null
40	}
41	}

Table 25. Source Code Modul 4 NewsFragment.kt

NewsViewModel.kt	
1	package com.example.mobil.ui.news
2	
3	import android.content.Context
4	import androidx.lifecycle.LiveData
5	import androidx.lifecycle.MutableLiveData
6	import androidx.lifecycle.ViewModel
7	import com.example.mobil.R
8	import com.example.mobil.model.NewsData
9	
10	class NewsViewModel : ViewModel() {
11	private val _image = MutableLiveData<Int>()
12	val image: LiveData<Int> = _image
13	
14	private val _name = MutableLiveData<String>()
15	val name: LiveData<String> = _name
16	
17	private val _desc = MutableLiveData<String>()
18	val desc: LiveData<String> = _desc
19	
20	
21	fun loadData(): List<NewsData> {
22	return listOf(
23	NewsData(R.string.news_title1, R.string.news_desc1,
24	R.drawable.news01),
25	NewsData(R.string.news_title2, R.string.news_desc2,
26	R.drawable.news02),
27	NewsData(R.string.news_title3, R.string.news_desc3,
28	R.drawable.news03),
29	NewsData(R.string.news_title4, R.string.news_desc4,
30	R.drawable.news04)
31)
32	}
33	
34	fun setData(data: NewsData, context: Context) {
35	_name.value = context.getString(data.title)
36	_desc.value = context.getString(data.description)
37	_image.value = data.image
38	}
39	
40	}

Table 26. Source Code Modul 4 NewsViewModel.kt

B. Pembahasan



Gambar 18. Screenshot Daftar Sourcecode Modul 4

[MainActivity.kt] Berfungsi sebagai controller dari navigasi.

[SplashActivity.kt] Berfungsi sebagai splash screen ketika pertama kali membuka aplikasi.

[Model] Berfungsi untuk menyimpan data class.

[GalleryAdapter.kt] Berfungsi untuk membuat recyclerview untuk view gallery.

[HomeAdapter.kt] Berfungsi untuk membuat recyclerview untuk view Home.

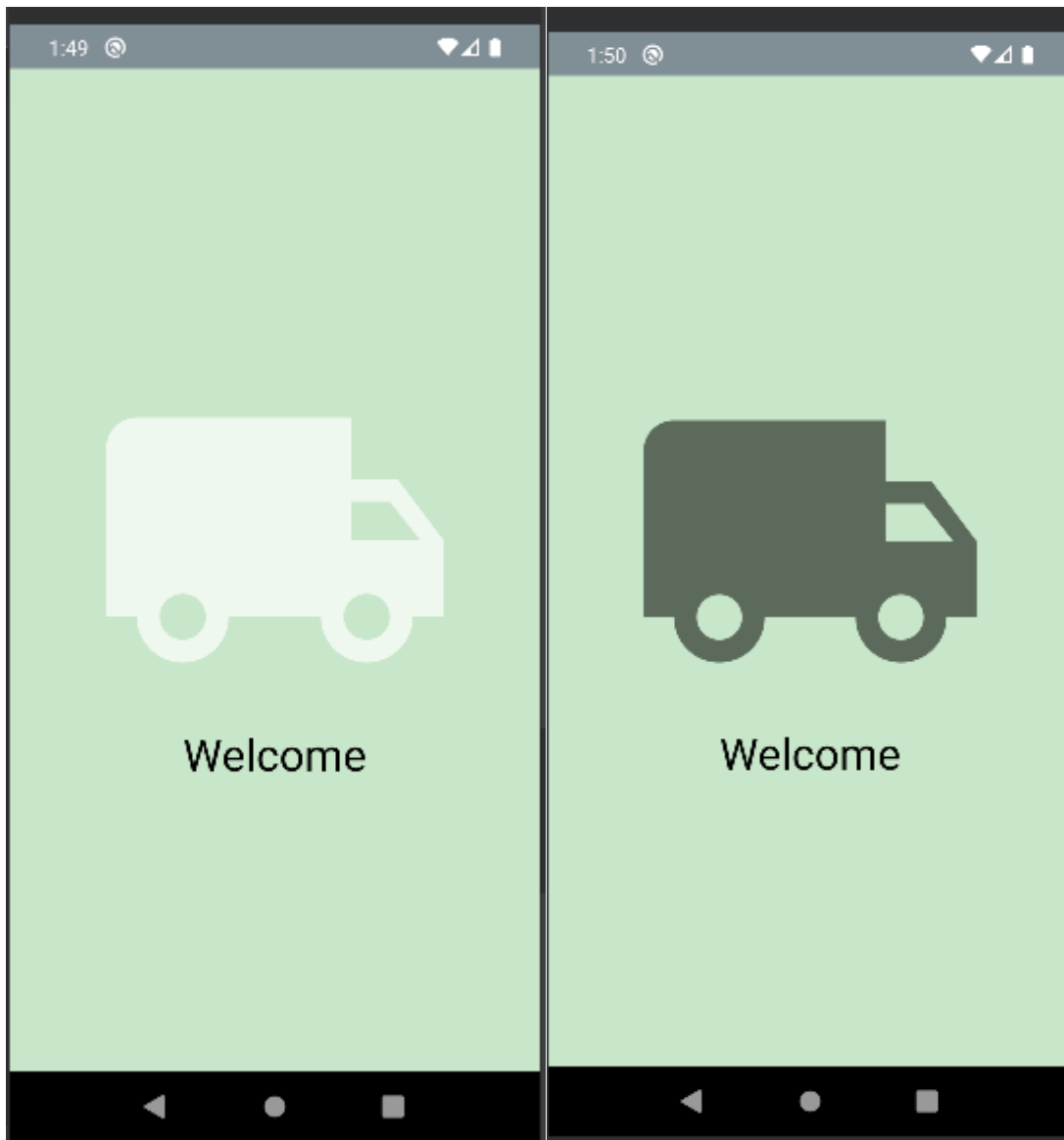
[NewsAdapter.kt] Berfungsi untuk membuat recyclerview untuk view News.

[GalleryDetailActivity.kt, HomeDetailActivity.kt, NewsDetailActivity.kt] Berfungsi sebagai controller objek yang diambil dari item pada recyclerview.

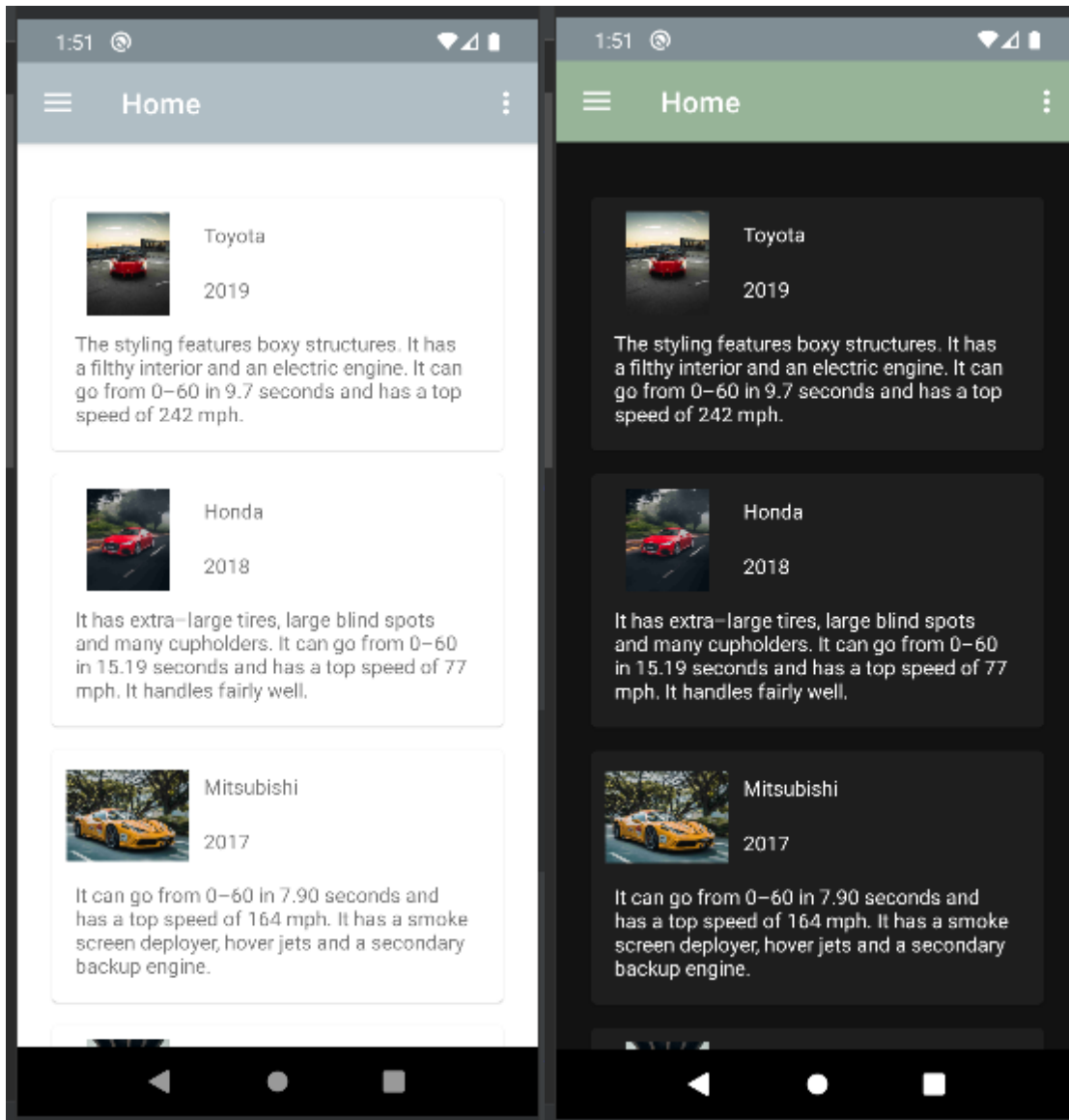
[GalleryFragment.kt, HomeFragment.kt, NewsFragment.kt] Berfungsi untuk menampilkan view fragment.

[GalleryViewModel.kt, HomeViewModel.kt, NewsViewModel.kt] Berfungsi untuk menyimpan livedata.

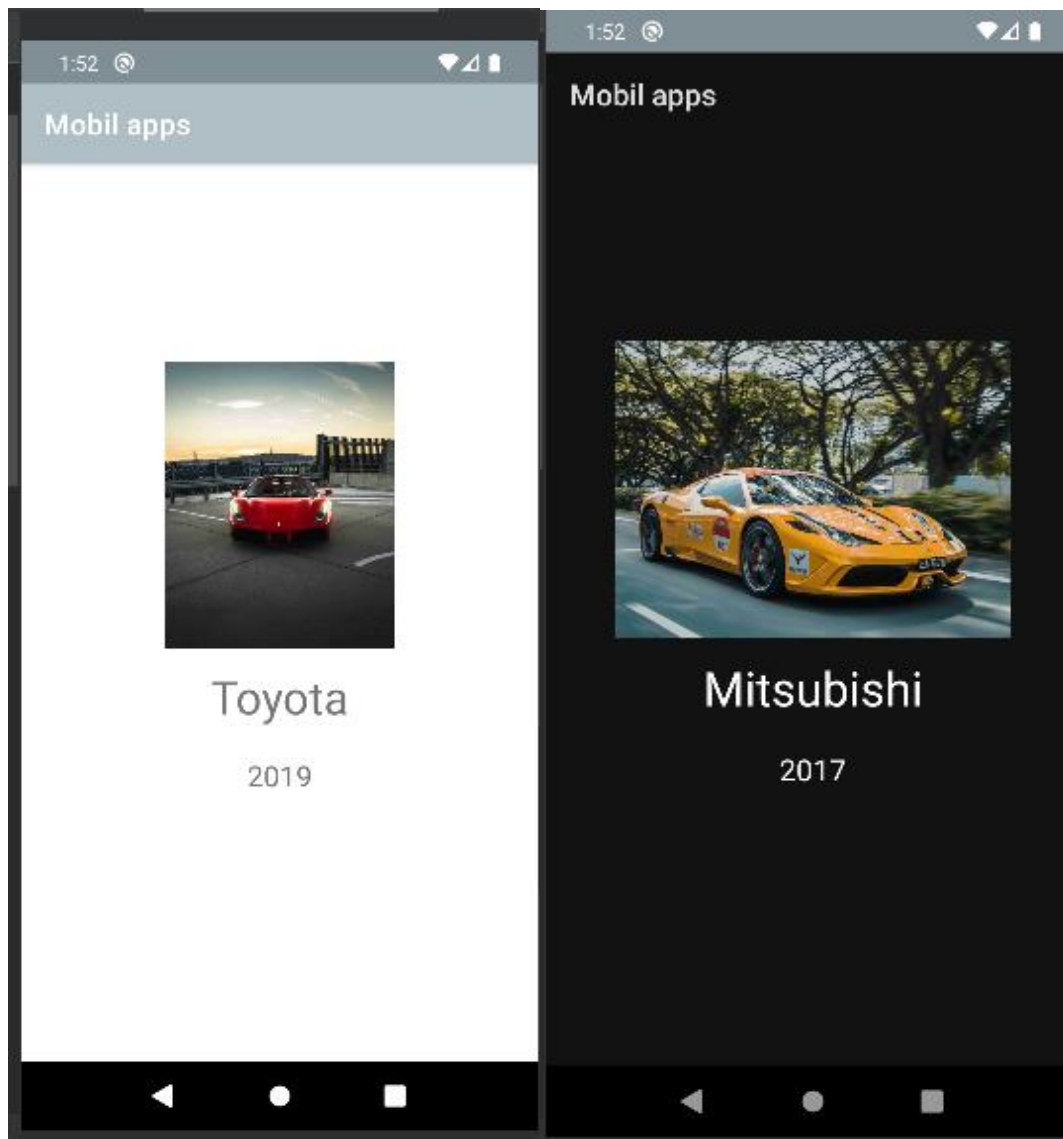
C. Hasil Screenshot



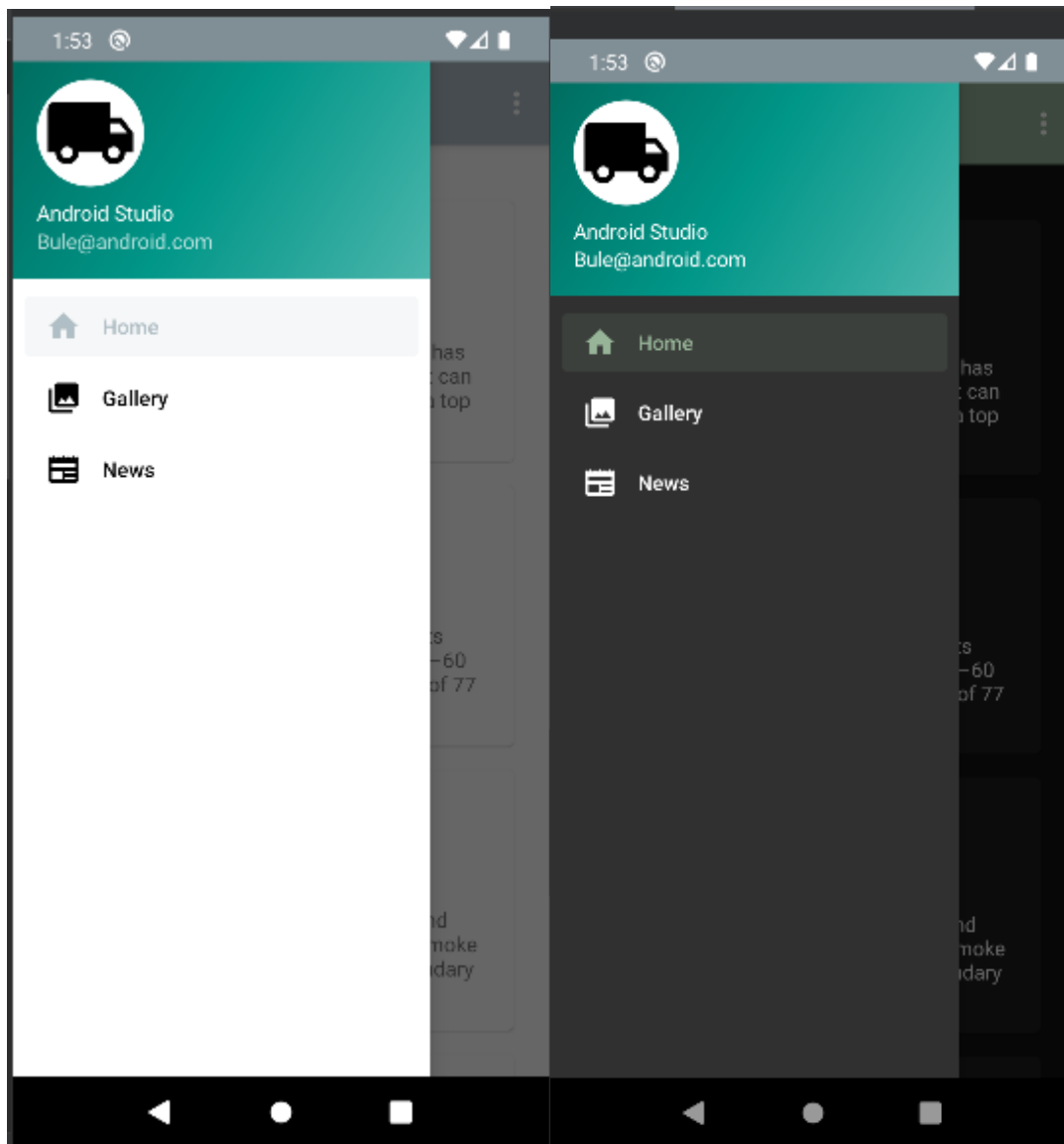
Gambar 19. Screenshot Modul 4 Masuk Aplikasi Mode Light & Dark



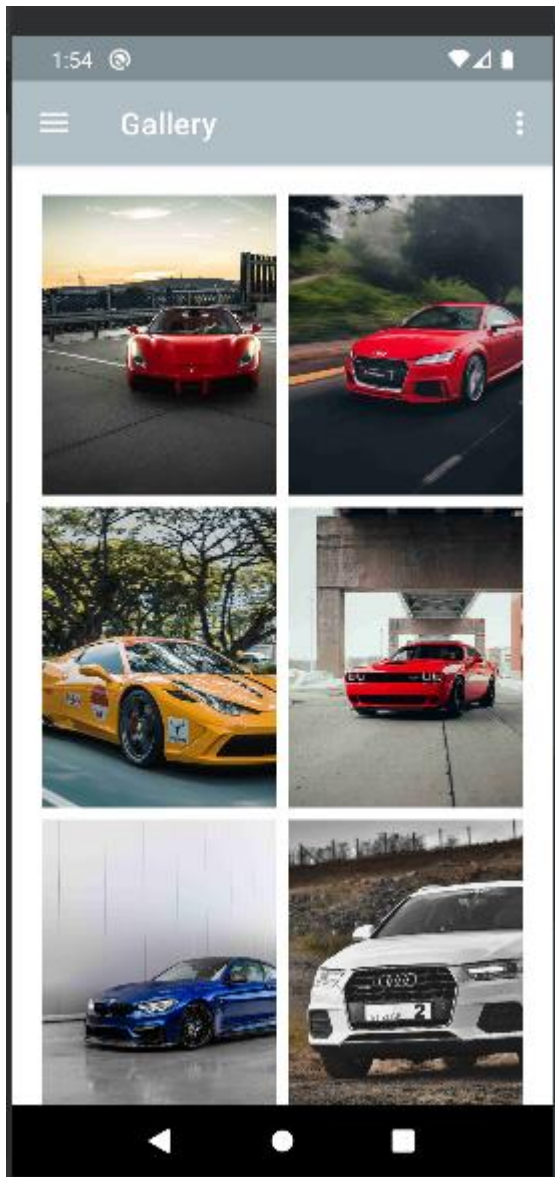
Gambar 20. Tampilan Awal Modul 4 Mode Light & Dark



Gambar 21. Screenshot ketika mengklik item dan masuk ke menu detail home



Gambar 22. Tampilan Modul 4 Navigation Drawer



Gambar 24. Tampilan Gallery Modul 4



Gambar 23. Tampilan Detail Gallery Modul 4

D. Tautan Github

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/FullowID/praktikummobile2/tree/main/modul4>

MODUL 5
MENGAMBIL DATA DARI INTERNET



Oleh:

Muhammad Ammarin Ihsan NIM. 2010817210002

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2022

SOAL

Buatlah sebuah aplikasi Android sederhana dengan spesifikasi sebagai berikut:

1. Daftar Public API yang dapat digunakan dapat dilihat pada link berikut: <https://github.com/public-apis/public-apis> (dapat juga mengambil diluar dari link tersebut).
2. Pada saat dijalankan, aplikasi akan terhubung dengan Internet untuk menarik data dari Public API tersebut.
3. Gunakan library tambahan yaitu Retrofit untuk mempermudah proses koneksi internet.
4. Gunakan library tambahan yaitu Moshi untuk mempermudah proses data JSON.
5. Data tersebut kemudian ditampilkan dalam bentuk RecyclerView.
6. Masing-masing data di RecyclerView tersebut dapat diklik untuk menampilkan detailnya.
7. Gunakan LiveData dan ViewModel untuk mempertahankan state dari aplikasi pada saat Configuration Changes.
8. Saat pengguna merotasi tampilan handphone dari Portrait menjadi Landscape maka tampilan data yang sudah ada tidak boleh hilang.

A. Source Code

MainActivity.kt	
1	package com.prakmodul5.hololive
2	
3	import androidx.appcompat.app.AppCompatActivity
4	import android.os.Bundle
5	import androidx.navigation.NavController
6	import androidx.navigation.fragment.NavHostFragment
7	import androidx.navigation.ui.NavigationUI
8	
9	class MainActivity : AppCompatActivity() {
10	private lateinit var navController: NavController
11	
12	override fun onCreate(savedInstanceState: Bundle?) {
13	super.onCreate(savedInstanceState)
14	setContentView(R.layout.activity_main)
15	val navHostFragment =
16	
17	supportFragmentManager.findFragmentById(R.id.nav_host_fragment) as
18	NavHostFragment
19	navController = navHostFragment.navController
20	NavigationUI.setupActionBarWithNavController(this,
21	navController)
22	}
23	}

Gambar 25. Source Code Modul 5 MainActivity.kt

BindingAdapter.kt	
1	package com.prakmodul5.hololive
2	
3	import android.view.View
4	import android.widget.ImageView
5	import androidx.core.net.toUri
6	import androidx.databinding.BindingAdapter
7	import androidx.recyclerview.widget.RecyclerView
8	import coil.load
9	import com.prakmodul5.hololive.network.ChannelsItem
10	import com.prakmodul5.hololive.ui.HoloListAdapter
11	import com.prakmodul5.hololive.ui.HololiveApiStatus
12	
13	@BindingAdapter("listData")
14	fun bindRecyclerView(recyclerView: RecyclerView, data:
15	List<ChannelsItem>?) {
16	val adapter = recyclerView.adapter as HoloListAdapter
17	adapter.submitList(data)
18	}
19	
20	@BindingAdapter("imageUrl")
21	fun bindImage(imgView: ImageView, imgUrl: String?) {
22	imgUrl?.let {
23	val imgUri = imgUrl.toUri().buildUpon().scheme("https").build()
24	imgView.load(imgUri) {
25	placeholder(R.drawable.loading_animation)
26	error(R.drawable.ic_broken_image)
27	}
28	}
29	}
30	
31	@BindingAdapter("hololiveApiStatus")
32	fun bindStatus(statusImageView: ImageView, status: HololiveApiStatus?)
33	{
34	when (status) {
35	HololiveApiStatus.LOADING -> {
36	statusImageView.visibility = View.VISIBLE
37	
38	statusImageView.setImageResource(R.drawable.loading_animation)
39	}
40	HololiveApiStatus.ERROR -> {
41	statusImageView.visibility = View.VISIBLE
42	
43	statusImageView.setImageResource(R.drawable.ic_connection_error)
44	}
45	HololiveApiStatus.DONE -> {
46	statusImageView.visibility = View.GONE
47	}
48	}
49	}

Gambar 26. Source Code Modul 5 BindingAdapter.kt

Hololive.kt	
1	package com.prakmodul5.hololive.network
2	
3	data class Hololive(
4	val channels: List<ChannelsItem>? = null,
5)
6	
7	data class ChannelsItem(
8	val name: String? = null,
9	val description: String? = null,
10	val photo: String? = null,
11	val id: Int? = null
12)

Table 27. Source Code Modul 5 Hololive.kt

HololistViewModel.kt	
1	package com.prakmodul5.hololive.ui
2	
3	import android.util.Log
4	import androidx.lifecycle.LiveData
5	import androidx.lifecycle.MutableLiveData
6	import androidx.lifecycle.ViewModel
7	import androidx.lifecycle.viewModelScope
8	import com.prakmodul5.hololive.network.ChannelsItem
9	import com.prakmodul5.hololive.network.HoloApi
10	import kotlinx.coroutines.launch
11	
12	enum class HololiveApiStatus { LOADING, ERROR, DONE }
13	class HoloListViewModel : ViewModel() {
14	private val _status = MutableLiveData<HololiveApiStatus>()
15	val status: LiveData<HololiveApiStatus> = _status
16	
17	private val _hololives = MutableLiveData<List<ChannelsItem>?>()
18	val hololives: MutableLiveData<List<ChannelsItem>?> = _hololives
19	
20	private val _hololive = MutableLiveData<ChannelsItem>()
21	val hololive: LiveData<ChannelsItem> = _hololive
22	
23	init {
24	getHololiveData()
25	}
26	
27	fun getHololiveData() {
28	viewModelScope.launch {
29	_status.value = HololiveApiStatus.LOADING
30	try {
31	_hololives.value =
32	HoloApi.retrofitService.getData().channels
33	_status.value = HololiveApiStatus.DONE
34	} catch (e: Exception) {
35	_status.value = HololiveApiStatus.ERROR
36	_hololives.value = listOf()
37	e.message?.let { Log.i("Pesn Error", it) }
38	}
39	}
40	}

41	
42	fun onHololiveItemClicked(holo: ChannelsItem) {
43	_hololive.value = holo
44	}
45	}

Table 28. Source Code Modul 5 HololistViewModel

HololistFragment.kt	
1	package com.prakmodul5.hololive.ui
2	
3	import android.os.Bundle
4	import android.view.LayoutInflater
5	import android.view.View
6	import android.view.ViewGroup
7	import androidx.appcompat.app.AppCompatActivity
8	import androidx.fragment.app.Fragment
9	import androidx.fragment.app.activityViewModels
10	import androidx.navigation.fragment.findNavController
11	import androidx.recyclerview.widget.LinearLayoutManager
12	import
13	com.google.android.material.divider.MaterialDividerItemDecoration
14	import com.prakmodul5.hololive.R
15	import com.prakmodul5.hololive.databinding.FragmentHoloListBinding
16	
17	class HoloListFragment : Fragment() {
18	private val viewModel: HoloListViewModel by activityViewModels()
19	
20	override fun onCreateView(
21	inflater: LayoutInflater,
22	container: ViewGroup?,
23	savedInstanceState: Bundle?
24): View? {
25	val binding = FragmentHoloListBinding.inflate(inflater)
26	viewModel.getHololiveData()
27	binding.lifecycleOwner = viewLifecycleOwner
28	binding.viewModel = viewModel
29	binding.rv.adapter = HoloListAdapter(HololiveListener { holo ->
30	viewModel.onHololiveItemClicked(holo)
31	findNavController()
32	
33	.navigate(R.id.action_holoListFragment_to_holoDetailFragment)
34	})
35	
36	(activity as AppCompatActivity).supportActionBar?.title = "Data
37	Vtuber Hololive"
38	binding.rv.addItemDecoration(
39	MaterialDividerItemDecoration(
40	requireContext(),
41	LinearLayoutManager.VERTICAL
42)
43)
44	
45	
46	return binding.root
47	}
48	}

Table 29. Source Code Modul 5 HololistFragment.kt

HololistAdapter.kt

```
1 package com.prakmodul5.hololive.ui
2
3 import android.view.LayoutInflater
4 import android.view.ViewGroup
5 import androidx.recyclerview.widget.DiffUtil
6 import androidx.recyclerview.widget.ListAdapter
7 import androidx.recyclerview.widget.RecyclerView
8 import com.prakmodul5.hololive.databinding.ItemRvBinding
9 import com.prakmodul5.hololive.network.ChannelsItem
10
11 class HoloListAdapter(private val clickListener: HololiveListener) :
12     ListAdapter<ChannelsItem,
13     HoloListAdapter.HoloViewHolder>(DiffCallback) {
14     class HoloViewHolder(
15         var binding: ItemRvBinding
16     ) : RecyclerView.ViewHolder(binding.root) {
17         fun bind(clickListener: HololiveListener, hololive:
18     ChannelsItem) {
19             binding.holoItem = hololive
20             binding.clickListener = clickListener
21             binding.executePendingBindings()
22         }
23     }
24
25     companion object DiffCallback :
26     DiffUtil.ItemCallback<ChannelsItem>() {
27         override fun areItemsTheSame(oldItem: ChannelsItem, newItem:
28     ChannelsItem): Boolean {
29             return oldItem.id == newItem.id
30         }
31
32         override fun areContentsTheSame(oldItem: ChannelsItem, newItem:
33     ChannelsItem): Boolean {
34             return oldItem.name == newItem.name
35         }
36     }
37
38     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
39     HoloViewHolder {
40         val inflater = LayoutInflater.from(parent.context)
41         return HoloViewHolder(
42             ItemRvBinding.inflate(inflater, parent, false)
43         )
44     }
45
46     override fun onBindViewHolder(holder: HoloViewHolder, position:
47     Int) {
48         val holo = getItem(position)
49         holder.binding.bind(clickListener, holo)
50     }
51
52 }
53
54 class HololiveListener(val clickListener: (holo: ChannelsItem) -> Unit)
55 {
56 }
```

57	fun onClick(holo: ChannelsItem) = clickListener(holo)
58	}

Table 30. Source Code Modul 5 HololistAdapter.kt

HoloDetailFragment.kt	
1	package com.prakmodul5.hololive.ui
2	
3	import android.os.Bundle
4	import android.view.LayoutInflater
5	import android.view.MenuItem
6	import android.view.View
7	import android.view.ViewGroup
8	import androidx.appcompat.app.AppCompatActivity
9	import androidx.fragment.app.Fragment
10	import androidx.fragment.app.activityViewModels
11	import androidx.navigation.fragment.findNavController
12	import com.prakmodul5.hololive.R
13	import com.prakmodul5.hololive.databinding.FragmentHoloDetailBinding
14	
15	class HoloDetailFragment : Fragment() {
16	private val viewModel: HoloListViewModel by activityViewModels()
17	
18	override fun onCreateView(
19	inflater: LayoutInflater,
20	container: ViewGroup?,
21	savedInstanceState: Bundle?
22): View? {
23	val binding = FragmentHoloDetailBinding.inflate(inflater)
24	binding.lifecycleOwner = this
25	binding.viewModel = viewModel
26	
27	(activity as AppCompatActivity).supportActionBar?.title =
28	viewModel.hololive.value?.name
29	return binding.root
30	}
31	
32	override fun onCreate(savedInstanceState: Bundle?) {
33	super.onCreate(savedInstanceState)
34	setHasOptionsMenu(true)
35	
36	}
37	
38	override fun onOptionsItemSelected(item: MenuItem): Boolean {
39	when (item.itemId) {
40	android.R.id.home ->
41	findNavController().navigate(R.id.action_holoDetailFragment_to_holoListFragment)
42	}
43	return true
44	}
45	}

Table 31. Source Code Modul 5 HoloDetailFragment.kt

B. Pembahasan

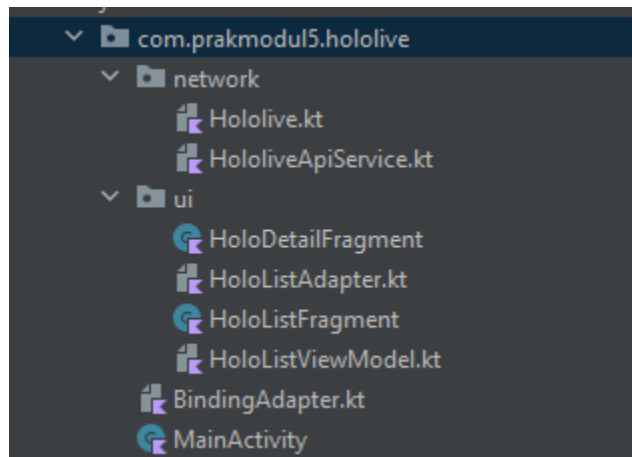


Table 32. Daftar Source Code Modul 5

[MainActivity.kt] Berfungsi sebagai controller navigasi.

[BindingAdapter.kt] Berfungsi sebagai adapter binding yang menyimpan fungsi untuk dipanggil pada layout xml.

[Hololive.kt] Berfungsi sebagai data class untuk menjadi pencetak objek dari api yang dipanggil.

[HoloListFragment.kt] Merupakan Controller untuk fragment daftar yang berisi recycler view.

[HoloListAdapter.kt] Berfungsi untuk membuat recycler view.

[HoloListViewModel.kt] Berfungsi untuk menyimpan live data.

[HoloListDetailFragment.kt] Berfungsi sebagai controller detail dari item recycler view.

C. Hasil Screenshot



Table 33. Tampilan Awal Modul 5



Table 34. Tampilan Detail Hololist Modul 5

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/FollowID/praktikummobile2/tree/main/modul5>