

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 4**



**View Model and Debugging**

**Oleh:**

**Natalie Grace Katiandagho**

**NIM. 2310817120003**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS LAMBUNG MANGKURAT  
MEI 2025**

**LEMBAR PENGESAHAN**  
**LAPORAN PRAKTIKUM PEMROGRAMAN I**  
**MODUL 4**

Laporan Praktikum Pemrograman Mobile Modul 4:

View Model and Debugging ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Natalie Grace Katiandagho

NIM : 2310817120003

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Natalie Grace Katiandagho  
NIM. 2310817120003

Muti`a Maulida S.Kom M.T.I  
NIP. 19881027 201903 20 13

## DAFTAR ISI

LEMBAR PENGESAHAN.....	2
DAFTAR ISI .....	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL .....	<b>Error! Bookmark not defined.</b>
SOAL 1 .....	6
A. Source Code.....	6
B. Output Program.....	23
C. Pembahasan .....	24

## DAFTAR GAMBAR

Tabel 1. Source Code Jawaban Soal 1.....	<b>Error! Bookmark not defined.</b>
--	-------------------------------------

## DAFTAR GAMBAR

Tabel 1. Manhwa.kt Modul 4 .....	6
Tabel 2. ManhwaRepository.kt .....	7
Tabel 3. MainActivity.kt Modul 4.....	13
Tabel 4. ManhwaViewModel.kt.....	16
Tabel 5. ViewModelFactory.kt.....	17
Tabel 6. DetailScreen.kt .....	18
Tabel 7. ListScreen.kt Modul 4 .....	20
Tabel 8. Theme.kt Modul 4 .....	21

## SOAL 1

Lanjutkan aplikasi Android berbasis XML dan Jetpack Compose yang sudah dibuat pada Modul 3 dengan menambahkan modifikasi sesuai ketentuan berikut:

- a. Buatlah sebuah ViewModel untuk menyimpan dan mengelola data dari list item. Data tidak boleh disimpan langsung di dalam Fragment atau Activity.
- b. Gunakan ViewModelFactory dalam pembuatan ViewModel
- c. Gunakan StateFlow untuk mengelola event onClick dan data list item dari ViewModel ke Fragment
- d. gunakan logging untuk event berikut:
  - a. Log saat data item masuk ke dalam list
  - b. Log saat tombol Detail dan tombol Explicit Intent ditekan
  - c. Log data dari list yang dipilih ketika berpindah ke halaman Detail
- e. Gunakan tool Debugger di Android Studio untuk melakukan debugging pada aplikasi. Cari setidaknya satu breakpoint yang relevan dengan aplikasi. Lalu, gunakan fitur Step Into, Step Over, dan Step Out. Setelah itu, jelaskan fungsi Debugger, cara menggunakan Debugger, serta fitur Step Into, Step Over, dan Step Out

### A. Source Code

Tabel 1. Manhwa.kt Modul 4

1	package com.example.gracemanhwa_picks.data
2	
3	data class Manhwa(
4	val id: Int,
5	val title: String,
6	val author: String,
7	val description: String,

8	val imageRes: Int,
9	val url: String
10	)

*Tabel 2.ManhwaRepository.kt*

01	package com.example.gracemanhwa_picks.data
2	
3	import com.example.gracemanhwa_picks.R
4	
5	class ManhwaRepository {
6	fun getManhwas(): List<Manhwa> {
7	return listOf(
8	Manhwa(
9	id = 1,
10	title = "Solo Leveling",
11	author = "Chu-Gong",
12	description = "In a world where hunters
	– human warriors who possess supernatural abilities –
	must battle deadly monsters to protect all mankind from
	certain annihilation, a notoriously weak hunter named
	Sung Jin-woo finds himself in a seemingly endless
	struggle for survival. "
13	+ "One day, after narrowly
	surviving an overwhelmingly powerful double dungeon
	that nearly wipes out his entire party, a mysterious
	program called the System chooses him as its sole
	player and in turn, gives him the unique ability to
	level up in strength. "
14	+ "This is something no other
	hunter is able to do, as a hunter's abilities are set

	once they awaken. Jinwoo then sets out on a journey as he fights against all kinds of enemies, both man and monster, to discover the secrets of the dungeons and the true source of his powers. "
15	<div> <div></div> <div>+ "He soon discovers that he has been chosen to inherit the position of Shadow Monarch, essentially turning him into an immortal necromancer who has absolute rule over the dead. "</div> </div>
16	<div> <div></div> <div>+ "He is the only Monarch who fights to save humanity, as the other Monarchs are all trying to kill him and wipe out the humans.",</div> </div>
17	<div> <div> <div>imageRes = R.drawable.solo_leveling,</div> <div>url =</div> </div> <div>"https://www.tappytoon.com/en/book/solo-leveling-official"</div> </div>
18	<div> <div></div> <div>),</div> </div>
19	<div> <div></div> <div>Manhwa(</div> </div>
20	<div> <div></div> <div>id = 2,</div> </div>
21	<div> <div></div> <div>title = "Omniscient Reader",</div> </div>
22	<div> <div></div> <div>author = "SingNSong",</div> </div>
23	<div> <div></div> <div>description = "Kim Dokja is a young man leading a simple life, who has been the sole reader of a novel \"Three Ways to Survive in a Ruined World\" for 13 years of his life. "</div> </div>
24	<div> <div></div> <div>+ "As he was reading the novel's final chapter, reality and the world of fiction started to merge, allowing him to appear at the beginning point of the story. "</div> </div>
25	<div> <div></div> <div>+ "Being the only person who knew how the world could end, Kim Dokja is determined to create a different ending by solving and conquering</div> </div>



	various challenges, known as scenarios, which are operated by dokkaebi.",
26	<pre>                 imageRes = R.drawable.omniscient_reader, </pre>
27	<pre>                 url = "https://www.webtoons.com/en/action/omniscient- reader/list?title_no=2154"             ), </pre>
28	<pre>             Manhwa( </pre>
29	<pre>                 id = 3, </pre>
30	<pre>                 title = "The Beginning After the End", </pre>
31	<pre>                 author = "TurtleMe", </pre>
32	<pre>                 description = "It follows the life of the late King Grey after his untimely and mysterious death. Reborn as Arthur Leywin, he seeks to correct his past mistakes in the vibrant new continent of Dicathen, a world of magic and fantastical creatures. " </pre>
33	<pre>                     + "Equipped with the knowledge of a powerful king in his mid-thirties, Arthur navigates his new life as the magic-wielding child of two retired adventurers and gains purpose through each of his new experiences—something he lacked in his previous life. " </pre>
34	<pre>                     + "When a kind dragon sacrifices her life to protect him, Arthur resolves to live a sincere, kind, and courageous life with those he loves. With the help of a lost elf princess and the Elven Kingdom of Elenoir, Arthur begins his long journey to find his true place in the world.\n\n" </pre>
35	<pre>                     + "As the years pass, Arthur becomes more and more comfortable in this world, </pre>

	positioning himself as a young, but respected figure. However, deja-vu strikes as a war brews between Dicathen and the Vritra, a clan of banished deities now ruling over a faraway continent. "
36	+ "Arthur must rise as a leader, despite his fear of becoming the war-hardened monster he once was in his past life. "
37	+ "As the war rages on, Arthur discovers that he was not reborn to this world by chance...nor was he the only one.",
38	imageRes = R.drawable.tbate,
39	url = "https://tapas.io/series/tbate- comic/info"
40	),
41	Manhwa( id = 4, title = "Eleceed", author = "Jeho Son", description = "Jiwoo is a kind-hearted young man who harnesses the lightning-quick reflexes of a cat to secretly make the world a better place - one saved little child or foster pet at a time. "
46	+ "Kayden is a secret agent on the run, who finds himself stuck in the body of a...um...decidedly fat old fluffy cat. "
47	+ "Together, armed with Jiwoo's superpowers and Kayden's uber-smarts, they're out to fight those forces who would let evil rule this world. "
48	+ "That is, if they can stand each other long enough to get the job done.",

49	imageRes = R.drawable.eleceed,
50	url =
	"https://www.webtoons.com/en/action/eleceed/list?title_no=1571"
51	),
52	Manhwa(
53	id = 5,
54	title = "Killer Peter",
55	author = "Kim Junghyun",
56	description = "On the surface, Glory Hound is a simple human rights organization. In reality, the organization has some of the best assassins in the world, in charge of performing legendary.\n\n"
57	+ "One of their best members was simply known as Apostle Peter, and he retired in protest of the new leader, Raphael. However, resignations were not accepted, and Peter was soon ambushed. Despite his best efforts, he dies.\n\n"
58	+ "Instead of dying, though, Peter miraculously found himself back in his teenage body. He doesn't know why, but he knows one thing: he will destroy Glory Hound.",
59	imageRes = R.drawable.killer_peter,
	url =
	"https://www.webtoons.com/en/action/killer-peter/list?title_no=5816"
60	),
	Manhwa(
61	id = 6,
62	title = "Player Who Can't Level Up",

63	author = "GaVinGe",
64	description = "When Kim Kigyu received his invitation to become a player (a unique-ability player, at that), he thought his struggles were over. But no matter how hard he tries, he just can't seem to get past level 1! "
65	+ "After five years of working as a guide on the lower floors of the tower, he's finally discovered his ability to link with "Egos" and raise his stats. "
66	+ "As his new skills unlock adventures in unexplored gates, Kigyu gets his chance to defy expectations and show the world that rank isn't everything.",
67	imageRes = R.drawable.player_cant_level_up,
68	url = "https://tapas.io/episode/2414063"
69	),
70	Manhwa(
71	id = 7,
72	title = "SSS-Class Revival Hunter",
73	author = "Shinnoa",
74	description = "After the Tower suddenly appeared, individuals who wished to pursue their personal values began to inhabit it, coming to be called \"hunters.\" "
75	+ "Everyone had their own goals, but only a chosen few were acknowledged and given powerful skills by the mysterious structure. "
76	+ "Kim Gong-Ja, a weak F-Class

	hunter without any skills, is envious of those who were blessed by the Tower. "
77	<pre> + "Letting his jealousy overcome him one day, Gong-Ja abruptly receives a S- Class skill that allows him to copy a skill from someone else—after they kill him.\n\n" </pre>
78	<pre> + "Sooner than he likes, Gong- Ja gets to test his newly acquired ability on the legendary hunter known as the Flame Emperor. " </pre>
79	<pre> + "As he is dying, Gong-Ja learns the evil truth about the man he once admired the most. " </pre>
80	<pre> + "Receiving another potent skill that allows him to revive and go back in time by 24 hours, Gong-Ja devises a plan to travel 11 years into the past to eliminate the Flame Emperor and cement himself as the world's best hunter.", </pre>
81	<pre> imageRes = R.drawable.sss_class_hunter, </pre>
82	<pre> url = "https://www.mangaread.org/manga/sss-class-suicide- hunter/" </pre>
83	<pre> ) </pre>
84	<pre> ) </pre>
85	<pre> } </pre>
86	<pre> } </pre>

*Tabel 3. MainActivity.kt Modul 4*

01	package com.example.gracemanhwa_picks
2	
3	import android.os.Bundle

```
4 import androidx.activity.ComponentActivity
5 import androidx.activity.compose.setContent
6 import androidx.activity.viewModels
7 import androidx.compose.runtime.Composable
8 import androidx.navigation.NavType
9 import androidx.navigation.compose.NavHost
10 import androidx.navigation.compose.composable
11 import
12 androidx.navigation.compose.rememberNavController
13 import androidx.navigation.navArgument
14 import
15 com.example.gracemanhwa_picks.data.ManhwaRepository
16 import com.example.gracemanhwa_picks.ui.DetailScreen
17 import com.example.gracemanhwa_picks.ui.ListScreen
18 import
19 com.example.gracemanhwa_picks.ui.GracemanhwaPicksTheme
20 import
21 com.example.gracemanhwa_picks.ui.ViewModel.ManhwaViewMo
22 del
23 import
24 com.example.gracemanhwa_picks.ui.ViewModel.ViewModelFac
25 tory
26
27 class MainActivity : ComponentActivity() {
28     private val viewModel: ManhwaViewModel by
29     viewModels {
30         ViewModelFactory(ManhwaRepository())
31     }
32
33     override fun onCreate(savedInstanceState: Bundle?)
```

```

29 {
30     super.onCreate(savedInstanceState)
31     setContent {
32         GracemanhwaPicksTheme {
33             ManhwaApp(viewModel = viewModel)
34         }
35     }
36 }
37
38
39 @Composable
40 fun ManhwaApp(viewModel: ManhwaViewModel) {
41     val navController = rememberNavController()
42     NavHost(navController = navController,
43 startDestination = "list") {
44         composable("list") {
45             ListScreen(navController = navController,
46 viewModel = viewModel)
47         }
48         composable(
49             route = "detail/{id}",
50             arguments = listOf(navArgument("id") { type
51 = NavType.IntType })
52         ) { backStackEntry ->
53             val id =
54 backStackEntry.arguments?.getInt("id") ?: 0
55             viewModel.getManhwaById(id)?.let { item ->
56                 DetailScreen(item = item)
57             }
58         }
59     }
60 }

```

56	}
57	}

*Tabel 4. ManhwaViewModel.kt*

1	package com.example.gracemanhwa_picks.ui.ViewModel
2	
3	import android.util.Log
4	import androidx.lifecycle.ViewModel
5	import com.example.gracemanhwa_picks.data.Manhwa
6	import
7	com.example.gracemanhwa_picks.data.ManhwaRepository
8	import kotlinx.coroutines.flow.MutableStateFlow
9	import kotlinx.coroutines.flow.StateFlow
10	
11	class ManhwaViewModel(private val repository:
12	ManhwaRepository): ViewModel() {
13	
14	private val _manhwas =
	MutableStateFlow<List<Manhwa>>(emptyList())
15	val manhwas: StateFlow<List<Manhwa>> get() =
	_manhwas
16	
17	init {
18	loadManhwas()
19	}
20	
21	private fun loadManhwas() {
22	_manhwas.value = repository.getManhwas()
23	Log.d("Manhwa ViewModel", "Manhwa data
	loaded into the list.")



24	}
25	
26	fun getManhwaById(id: Int): Manhwa? {
27	val manhwa = _manhwas.value.firstOrNull {
	<b>it</b> .id == id}
28	if (manhwa != null) {
29	Log.d("Manhwa ViewModel", "Navigating
	to Detail for: \${manhwa.title}")
30	}
31	return manhwa
32	}
33	}

*Tabel 5. ViewModelFactory.kt*

01	package com.example.gracemanhwa_picks.ui.ViewModel
2	
3	import androidx.lifecycle.ViewModel
4	import androidx.lifecycle.ViewModelProvider
5	import
6	com.example.gracemanhwa_picks.data.ManhwaRepository
7	
8	class ViewModelFactory(private val repository:
9	ManhwaRepository) : ViewModelProvider.Factory {
10	override fun <T : ViewModel> create(modelClass:
11	Class<T>): T {
12	if
13	(modelClass.isAssignableFrom(ManhwaViewModel::class.java
	a)) {
14	@Suppress("UNCHECKED_CAST")
15	return ManhwaViewModel(repository) as T

16	}
17	throw IllegalArgumentException("Unknown ViewModel class")
18	}
19	}

*Tabel 6. DetailScreen.kt*

01	package com.example.gracemanhwa_picks.ui
2	
3	import androidx.compose.foundation.Image
4	import androidx.compose.foundation.layout.*
5	import
6	androidx.compose.foundation.rememberScrollState
7	import
8	androidx.compose.foundation.shape.RoundedCornerShape
9	import androidx.compose.foundation.verticalScroll
10	import androidx.compose.material3.*
11	import androidx.compose.runtime.Composable
12	import androidx.compose.ui.Modifier
13	import androidx.compose.ui.draw.clip
14	import androidx.compose.ui.layout.ContentScale
15	import androidx.compose.ui.res.painterResource
16	import androidx.compose.ui.unit.dp
17	import com.example.gracemanhwa_picks.data.Manhwa
18	
19	@Composable
20	fun DetailScreen(item: Manhwa) {
21	val scrollState = rememberScrollState()
22	
23	Column (

```
24         modifier = Modifier
25             .fillMaxSize()
26             .verticalScroll(scrollState)
27             .padding(16.dp)
28     ) {
29         Image(
30             painter = painterResource(id =
item.imageRes),
31             contentDescription = item.title,
32             modifier = Modifier
33                 .fillMaxWidth()
34                 .height(250.dp)
35                 .clip(RoundedCornerShape(16.dp)),
36             contentScale = ContentScale.Crop
37         )
38         Spacer(modifier = Modifier.height(16.dp))
39         Text(item.title, style =
MaterialTheme.typography.headlineMedium)
40         Text("By ${item.author}", style =
MaterialTheme.typography.titleMedium)
41         Spacer(modifier = Modifier.height(8.dp))
42         Divider()
43         Spacer(modifier = Modifier.height(8.dp))
44         Text(item.description, style =
MaterialTheme.typography.bodyLarge)
45     }
46 }
```

Tabel 7. ListScreen.kt Modul 4

01	package com.example.gracemanhwa_picks.ui
2	
3	import androidx.compose.foundation.Image
4	import androidx.compose.foundation.layout.*
5	import
6	androidx.compose.foundation.rememberScrollState
7	import
8	androidx.compose.foundation.shape.RoundedCornerShape
9	import androidx.compose.foundation.verticalScroll
10	import androidx.compose.material3.*
11	import androidx.compose.runtime.Composable
12	import androidx.compose.ui.Modifier
13	import androidx.compose.ui.draw.clip
14	import androidx.compose.ui.layout.ContentScale
15	import androidx.compose.ui.res.painterResource
16	import androidx.compose.ui.unit.dp
17	import com.example.gracemanhwa_picks.data.Manhwa
18	
19	@Composable
20	fun DetailScreen(item: Manhwa) {
21	val scrollState = rememberScrollState()
22	
23	Column(
24	modifier = Modifier
25	.fillMaxSize()
26	.verticalScroll(scrollState)
27	.padding(16.dp)
28	) {
29	Image(
30	painter = painterResource(id =

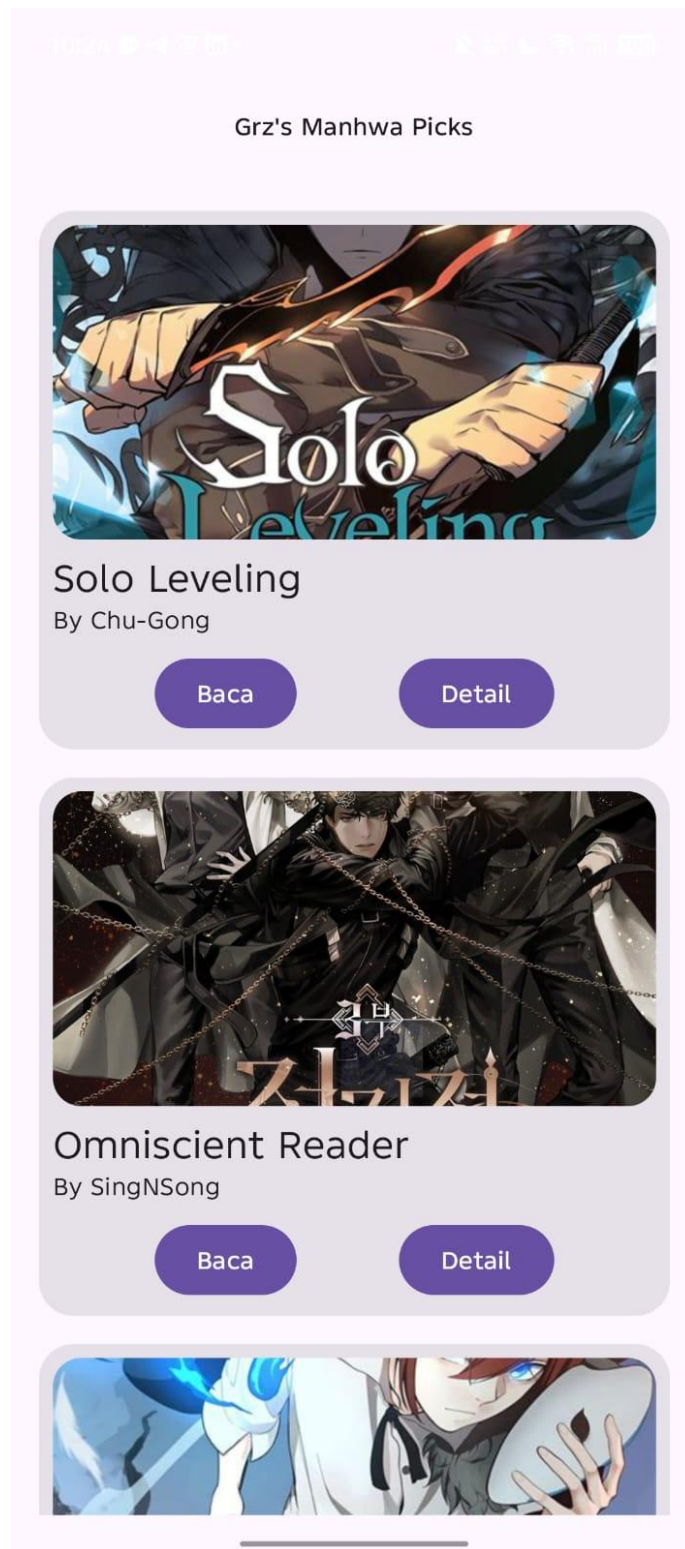
	item.imageRes),
31	contentDescription = item.title,
32	modifier = Modifier
33	.fillMaxWidth()
34	.height(250.dp)
35	.clip(RoundedCornerShape(16.dp)),
36	contentScale = ContentScale.Crop
37	)
38	Spacer(modifier = Modifier.height(16.dp))
39	Text(item.title, style =
	MaterialTheme.typography.headlineMedium)
40	Text("By \${item.author}", style =
	MaterialTheme.typography.titleMedium)
41	Spacer(modifier = Modifier.height(8.dp))
42	Divider()
43	Spacer(modifier = Modifier.height(8.dp))
44	Text(item.description, style =
45	MaterialTheme.typography.bodyLarge)
46	}
47	}

*Tabel 8. Theme.kt Modul 4*

01	package com.example.gracemanhwa_picks.ui
2	
3	import androidx.compose.material3.*
4	import androidx.compose.runtime.Composable
5	
6	@Composable

7	fun GracemanhwaPicksTheme(content: @Composable () -
	> Unit) {
8	MaterialTheme(
9	colorScheme = <i>lightColorScheme()</i> ,
10	typography = Typography(),
11	content = content
12	)
13	}

## B. Output Program



Gambar 1. Output Program Modul 4

### C. Pembahasan

Manhwa.kt untuk definisi struktur data untuk objek manhwa. Pada baris pertama hingga ketiga, package dan import tidak dituliskan karena file ini tidak memerlukan import eksternal. Deklarasi data class Manhwa dimulai pada baris keempat dengan parameter seperti id, title, author, description, imageRes, dan url. Tipe data yang digunakan seluruhnya bersifat eksplisit, seperti Int untuk id dan imageRes, serta String untuk properti lainnya. Struktur ini memungkinkan penyimpanan berbagai informasi penting yang akan digunakan dalam tampilan daftar dan detail manhwa.

ManhwaRepository.kt digunakan sebagai penyedia data manhwa. Package dideklarasikan pada awal, dan import untuk resource R digunakan agar dapat mengakses gambar manhwa dari drawable. Fungsi utama di dalam file ini adalah getManhwas() yang berada pada baris keenam. Fungsi ini mengembalikan list yang terdiri dari beberapa objek Manhwa. Masing-masing objek Manhwa diinisialisasi dengan data seperti judul, penulis, deskripsi panjang, resource gambar, dan URL. Setiap manhwa memiliki deskripsi unik yang menjelaskan latar belakang cerita dan karakter utama, serta tujuan atau konflik dalam kisah tersebut. Data yang tersedia sangat lengkap, mendukung kebutuhan aplikasi untuk menampilkan informasi manhwa secara mendetail.

MainActivity.kt memulai aktivitas utama aplikasi Android. Package serta import berbagai komponen Compose dan ViewModel dideklarasikan di awal. Komponen ComponentActivity diturunkan ke class MainActivity, dan pada baris kesebelas dilakukan inisialisasi viewModel menggunakan delegate by viewModels, disertai factory ViewModelFactory yang menerima instance ManhwaRepository. Dalam fungsi onCreate, method setContent dipanggil untuk mengatur konten tampilan dengan tema GracemanhwaPicksTheme. Fungsi ManhwaApp digunakan sebagai root composable. Di dalam fungsi ManhwaApp, controller navigasi diciptakan menggunakan rememberNavController. Route "list" diatur untuk menampilkan ListScreen, sementara route "detail/{id}" akan mengambil argumen id dan menampilkan DetailScreen berdasarkan item yang sesuai dari ViewModel. Pendekatan ini mendukung arsitektur single-activity dengan navigasi antar composable yang efisien.

ManhwaViewModel.kt berisi logika ViewModel untuk mengelola data manhwa. Package dan import mendeklarasikan penggunaan ViewModel, logging, dan StateFlow. Kelas ManhwaViewModel menerima parameter repository untuk mengakses data. Di dalamnya, variabel \_manhwas bertipe MutableStateFlow digunakan untuk menyimpan data secara mutable, sedangkan variabel publik manhwas bertipe StateFlow hanya menyediakan akses baca. Proses pengambilan data dimulai dari fungsi init yang langsung memanggil loadManhwas(). Fungsi ini mengambil data dari repository dan mengisi \_manhwas dengan daftar manhwa. Logging dilakukan untuk mencatat bahwa data telah dimuat. Fungsi getManhwaById menerima parameter id dan mencari objek manhwa dengan ID tersebut. Jika ditemukan, logging mencatat navigasi ke halaman detail berdasarkan judul manhwa yang dipilih.



ViewModelFactory.kt bertugas menyediakan mekanisme pembuatan ViewModel dengan parameter. Package dan import mendefinisikan penggunaan ViewModel dan ViewModelProvider. Kelas ViewModelFactory menerima parameter repository. Implementasi fungsi create digunakan untuk menghasilkan objek ManhwaViewModel jika modelClass yang diminta sesuai. Jika tidak cocok, maka dilempar pengecualian, memastikan ViewModel dapat menerima parameter eksternal dengan cara yang sesuai standar Android.

DetailScreen.kt merupakan composable untuk menampilkan halaman detail manhwa. Pada bagian awal, dilakukan import terhadap komponen layout seperti Column, Image, Text, dan Spacer, serta fungsi pendukung seperti painterResource dan ContentScale. Fungsi DetailScreen menerima parameter item bertipe Manhwa. Scroll state disiapkan agar tampilan dapat digulir secara vertikal. Layout utama menggunakan Column dengan modifier untuk ukuran penuh, scrollable, dan padding. Di dalamnya, gambar manhwa ditampilkan menggunakan Image dengan bentuk rounded dan skala cropping agar memenuhi tampilan horizontal. Selanjutnya, ditampilkan judul manhwa dengan gaya teks headline, disusul nama penulis dengan gaya teks title. Setelah itu, deskripsi panjang manhwa ditampilkan dengan pemisah berupa Divider dan jarak antar elemen dengan Spacer.

ListScreen.kt seharusnya berisi tampilan daftar manhwa, namun isi dari file yang ada adalah salinan DetailScreen.kt, menunjukkan kemungkinan kesalahan duplikasi konten saat penulisan laporan. Karena strukturnya identik, penjelasannya pun sama seperti pada DetailScreen, dan seharusnya diganti dengan kode yang menampilkan daftar manhwa dalam bentuk list item yang dapat ditekan untuk berpindah ke detail.

Theme.kt berfungsi untuk mengatur tema aplikasi. Package dan import mengarah ke penggunaan Material3 dan composable function. Fungsi GracemanhwaPicksTheme menerima lambda composable sebagai parameter. Di dalamnya, MaterialTheme digunakan dengan skema warna terang dan tipografi default. Fungsi ini membungkus semua konten aplikasi agar tetap konsisten secara tampilan dan nuansa visual.

## **Jawaban Soal 2**

Dalam arsitektur aplikasi Android, Application class merupakan komponen inti yang pertama kali diinisialisasi oleh sistem saat aplikasi dijalankan. Kelas ini hanya dibuat satu kali dan tetap aktif selama siklus hidup aplikasi berlangsung, sehingga sangat cocok digunakan untuk mengelola inisialisasi global. Salah satu fungsi utamanya adalah melakukan konfigurasi awal terhadap komponen atau pustaka yang digunakan di seluruh aplikasi, seperti Firebase, Retrofit, Room, atau library logging. Selain itu, Application class juga dapat dimanfaatkan untuk menyimpan state atau objek yang bersifat global, seperti repository, container dependency injection, atau konfigurasi tertentu yang diperlukan oleh berbagai aktivitas dan komponen UI.

Dengan membuat subclass dari Application dan meng-override fungsi onCreate(), proses inisialisasi yang penting dapat dilakukan sebelum aktivitas atau fragment pertama ditampilkan. Contohnya, repository atau service bisa diinisialisasi satu kali di sini, lalu disuntikkan ke dalam ViewModel melalui ViewModelFactory. Application juga bisa digunakan untuk memantau perilaku aplikasi secara keseluruhan, seperti mencatat log aktivitas, mendeteksi crash, atau mengelola analitik pengguna. Agar class ini dikenali oleh sistem Android, nama subclass Application perlu ditulis di atribut android:name pada file AndroidManifest.xml.

### **Tautan Git**

Berikut adalah tautan untuk semua source code yang telah dibuat.

[natnutnot/PrakMobile at master](#)