

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



BUILD A SCROLLABLE LIST

Oleh:

Rifky Putra Mahardika NIM. 2310817210023

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
MODUL 3

Laporan Praktikum Pemrograman Mobile Modul 3: Build a Scrollable List ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Rifky Putra Mahardika
NIM : 2310817210023

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar
NIM. 2210817210026

Muti`a Maulida S.Kom M.T.I
NIP. 198810272019032013

DAFTAR ISI

LEMBAR PENGESAHAN.....	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code.....	8
B. Output Program	22
C. Pembahasan	24
D. Tautan Git	29

DAFTAR GAMBAR

Gambar 1. Tampilan Awal Aplikasi.....	7
Gambar 2. Tampilan Aplikasi Setelah Dijalankan	8
Gambar 3. Screenshot Hasil Jawaban Soal 1 Tampilan Awal	22
Gambar 4. Screenshot Hasil Jawaban Soal 1 Saat Tip Dihitung.....	23
Gambar 5. Screenshot Hasil Jawaban Soal 1 Saat Tip Dibulat	Error! Bookmark not defined.
Gambar 6. Screenshot Hasil Jawaban Soal 1 Toast.....	Error! Bookmark not defined.
Gambar 7. Screenshot Hasil Jawaban Soal 1 Mode Landscape	24

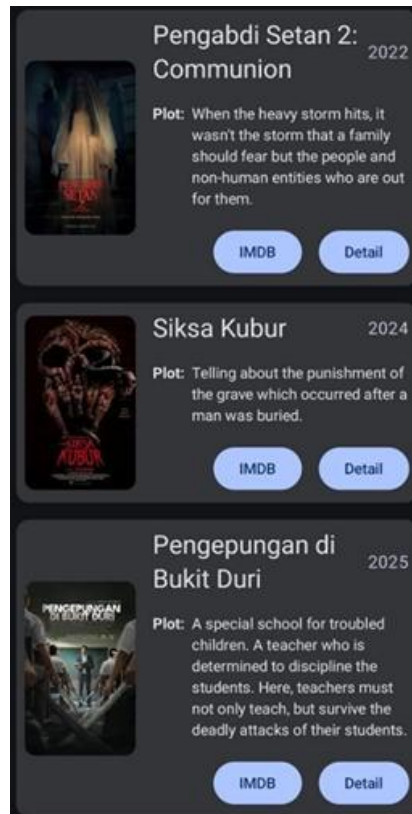
DAFTAR TABEL

Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt	10
Tabel 2. Source Code Jawaban Soal 1 TipCalculator.kt	14

SOAL PRAKTIKUM

1. Buatlah sebuah aplikasi Android menggunakan XML atau Jetpack Compose yang dapat menampilkan list dengan ketentuan berikut:
 1. List menggunakan fungsi RecyclerView (XML) atau LazyColumn (Compose)
 2. List paling sedikit menampilkan 5 item. Tema item yang ingin ditampilkan bebas
 3. item pada list menampilkan teks dan gambar sesuai dengan contoh di bawah
 4. Terdapat 2 button dalam list, dengan fungsi berikut:
 - a. Button pertama menggunakan intent eksplisit untuk membuka aplikasi atau browser lain
 - b. Button kedua menggunakan Navigation component/intent untuk membuka laman detail item
 5. Sudut item pada list dan gambar di dalam list melengkung atau rounded corner menggunakan Radius
 6. Saat orientasi perangkat berubah/dirotasi, baik ke portrait maupun landscape, aplikasi responsif dan dapat menunjukkan list dengan baik. Data di dalam list tidak boleh hilang
 7. Aplikasi menggunakan arsitektur single activity (satu activity memiliki beberapa fragment)
 8. Aplikasi berbasis XML harus menggunakan ViewBinding
2. Mengapa RecyclerView masih digunakan, padahal RecyclerView memiliki kode yang panjang dan bersifat boiler-plate, dibandingkan LazyColumn dengan kode yang lebih singkat?

UI item list harus berisi 1 gambar, 2 button (intent eksplisit dan navigasi), dan 2 baris teks dan setiap baris memiliki 2 teks yang berbeda. Diusahakan agar desain UI item list menyerupai UI berikut:



Gambar 1. Tampilan List Aplikasi

Desain UI laman detail bebas, tetapi diusahakan untuk mengikuti kaidah desain Material Design dan data item ditampilkan penuh di laman detail seperti contoh berikut:



Gambar 2. Tampilan Detail Aplikasi

A. Source Code

1. MainActivity.kt

```

1 package com.presca.modul3
2
3 import android.content.Intent
4 import android.net.Uri
5 import android.os.Bundle
6 import androidx.activity.ComponentActivity
7 import androidx.activity.compose.setContent
8 import androidx.activity.enableEdgeToEdge
9 import androidx.compose.runtime.Composable
10 import androidx.compose.ui.platform.LocalContext
11 import androidx.navigation.NavHostController
12 import androidx.navigation.compose.NavHost
13 import androidx.navigation.compose.composable
14 import androidx.navigation.compose.rememberNavController

```



```

15 import com.presca.modul3.ui.theme.Modul3Theme
16
17 class MainActivity : ComponentActivity() {
18     override fun onCreate(savedInstanceState: Bundle?) {
19         super.onCreate(savedInstanceState)
20         enableEdgeToEdge()
21         setContent {
22             Modul3Theme {
23                 val navController = rememberNavController()
24                 AppNavigation(navController)
25             }
26         }
27     }
28 }
29
30 @Composable
31 fun AppNavigation(navController: NavHostController) {
32     val context = LocalContext.current
33
34     NavHost(
35         navController = navController,
36         startDestination = "music_list"
37     ) {
38         composable("music_list") {
39             MusicListScreen(
40                 onMusicClick = { music ->
41                     navController.navigate("detail/${music.id}")
42                 },
43                 onExternalClick = { url ->
44                     val intent = Intent(Intent.ACTION_VIEW,
45                         Uri.parse(url))
46                     context.startActivity(intent)
47                 }
48             )
49         }
50     }
51 }

```

47)
48	}
49	
50	composable("detail/{musicId}") { backStackEntry ->
51	val musicId =
	backStackEntry.arguments?.getString("musicId")?.toIntOrNull()
	val music = musicList.find { it.id == musicId }
52	music?.let {
53	MusicDetailScreen(music = it, navController =
54	navController)
	}
55	}
56	}
57	}
58	
59	

Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt

2. MusicData.kt

1	package com.presca.modul3
2	
3	data class Music(
4	val id: Int,
5	val title: String,
6	val year: String,
7	val imageUrl: String,
8	val externalUrl: String,
9	val description: String
10)
11	
12	val musicList = listOf(
13	Music(
14	id = 1,

15	title = "What is Love?",
16	year = "9 April 2018",
17	imageUrl = "https://i.postimg.cc/5tpLc5DY/What-Is-Love-Online-Cover.webp",
18	externalUrl =
	"https://en.wikipedia.org/wiki/What_Is_Love%3F_(Twice_song)",
19	description = "Lagu \"What is Love?\" menceritakan tentang rasa penasaran seorang gadis muda yang belum pernah merasakan cinta sejati. Ia hanya mengetahui tentang cinta dari film, drama, dan buku, sehingga muncul keinginan kuat untuk benar-benar memahami dan merasakannya sendiri. Lirik lagu ini menggambarkan perasaan ingin tahu dan harapan akan datangnya cinta yang indah seperti yang sering digambarkan dalam kisah romantis di layar kaca."
20),
21	Music(
22	id = 2,
23	title = "Fancy",
24	year = "22 April 2019",
25	imageUrl =
	"https://upload.wikimedia.org/wikipedia/id/0/09/Twice_-_Fancy_You.png",
26	externalUrl = "https://twice.fandom.com/wiki/Fancy",
27	description = "\"FANCY\" bercerita tentang perasaan jatuh cinta yang begitu kuat, penuh keberanian, dan tidak takut mengambil risiko. Lagu ini menggambarkan momen ketika seseorang naksir atau menyukai orang lain dengan sangat intens, bahkan meski terasa berbahaya seperti duri mawar, perasaan itu tetap terasa manis dan menyenangkan. TWICE mengekspresikan keinginan untuk mengungkapkan cinta secara langsung, tanpa ragu, dan berani mengambil langkah pertama. Lirik seperti \"Aku menyukaimu, aku menyukaimu, menyukaimu\" dan \"Pegang lebih kuat, ambil tanganku. Ini akan sedikit berbahaya, bahkan lebih berbahaya lagi, sayang\" menegaskan

	keberanian dalam menghadapi cinta yang penuh tantangan."
28),
29	Music(
30	id = 3,
31	title = "The Feels",
32	year = "1 Oktober 2021",
33	imageUrl =
	"https://upload.wikimedia.org/wikipedia/en/5/50/Twice_-
	_The_Feels.png",
34	externalUrl =
	"https://twice.fandom.com/wiki/The_Feels",
35	description = "\"The Feels\" menggambarkan perasaan
	jatuh cinta yang kuat namun penuh rasa malu, seperti yang
	sering dialami remaja pada cinta pertama. Lagu ini bercerita
	tentang dua orang yang saling menyukai, namun keduanya masih
	malu-malu untuk mengungkapkan perasaan mereka secara
	langsung. TWICE mengekspresikan kegembiraan, rasa penasaran,
	dan degup jantung yang tak tertahankan saat jatuh cinta,
	sekaligus memberi kode kepada orang yang disukai agar lebih
	berani mengungkapkan perasaannya."
36),
37	Music(
38	id = 4,
39	title = "Likey",
40	year = "30 Oktober 2017",
41	imageUrl =
	"https://i.scdn.co/image/ab67616d0000b2731f21c24e81a9d0d4a30b
	e533",
42	externalUrl = "https://twice.fandom.com/wiki/Likey",
43	description = "\"Likey\" menceritakan tentang
	perasaan suka dan jatuh cinta yang membuat seseorang merasa
	bersemangat sekaligus sedikit canggung. Lagu ini
	menggambarkan bagaimana sang tokoh utama membangun
	kepercayaan diri untuk mengungkapkan perasaannya kepada orang

	yang disukai, meski masih ada rasa malu dan ragu. Selain itu, lagu ini juga mengangkat tema tentang bagaimana seseorang ingin tampil menarik dan disukai, terutama di era media sosial, sehingga ada tekanan untuk selalu menunjukkan sisi terbaiknya agar mendapatkan \"like\" atau perhatian dari orang lain. Melalui liriknya, TWICE menyampaikan pesan tentang keberanian dalam mencintai dan menjadi diri sendiri, tanpa harus terlalu terpengaruh oleh penilaian orang lain."
44),
45	Music(
46	id = 5,
47	title = "Cheer Up",
48	year = "24 April 2016",
49	imageUrl =
	"https://i.scdn.co/image/ab67616d0000b273acf4830dde5e17d356b80ae8",
50	externalUrl =
	"https://en.wikipedia.org/wiki/Cheer_Up_(song)",
51	description = "Lagu ini mengisahkan tentang seorang perempuan yang sedang dalam tahap pendekatan dengan pria yang sangat posesif, di mana ia meminta sedikit ruang agar proses pendekatan menjadi lebih menyenangkan dan tidak menekan. Melalui liriknya, \"Cheer Up\" menyampaikan pesan untuk tetap semangat dan memberikan dukungan kepada orang yang disukai, meskipun ada rasa canggung dan ketidakpastian dalam hubungan yang sedang berkembang. Lagu ini juga menonjolkan karakter ceria dan energik khas TWICE, dengan melodi yang catchy dan koreografi yang dinamis."
52),
53	Music(
54	id = 6,
55	title = "TT",
56	year = "24 Oktober 2016",
57	imageUrl =

	"https://i.scdn.co/image/ab67616d0000b273387444ab2fc1f08dfe7915ab",
58	externalUrl =
	"https://en.wikipedia.org/wiki/TT_(song)",
59	description = "Lagu \"TT\" menggambarkan perasaan cinta yang membingungkan dan penuh gejolak. Liriknya menceritakan tentang seorang perempuan yang merasa kesal dan bingung terhadap perasaan cintanya yang semakin menggebu-gebu meskipun ia berusaha menjauh. Perasaan tersebut membuatnya merasa seperti \"TT\" - sebuah ekspresi emotikon menangis yang juga menjadi simbol khas lagu ini. Lagu ini mengungkapkan dilema antara ingin mendekat dan sekaligus merasa canggung atau takut dalam menghadapi cinta."
60)
61)

Tabel 2. Source Code Jawaban Soal 1 MusicData.kt

3. MusicListScreen.kt

1	package com.presca.modul3
2	
3	import androidx.compose.foundation.layout.*
4	import androidx.compose.foundation.lazy.LazyColumn
5	import androidx.compose.foundation.lazy.items
6	import androidx.compose.foundation.shape.RoundedCornerShape
7	import androidx.compose.material3.*
8	import androidx.compose.runtime.Composable
9	import androidx.compose.ui.Modifier
10	import androidx.compose.ui.draw.clip
11	import androidx.compose.ui.unit.dp
12	import androidx.compose.ui.unit.sp
13	import androidx.compose.ui.text.font.FontWeight
14	import

	com.bumptechn.glide.integration.compose.ExperimentalGlideComposeApi
15	import com.bumptechn.glide.integration.compose.GlideImage
16	
17	@OptIn(ExperimentalGlideComposeApi::class, ExperimentalMaterial3Api::class)
18	@Composable
19	fun MusicListScreen(20 onMusicClick: (Music) -> Unit, 21 onExternalClick: (String) -> Unit 22) { 23 Scaffold(24 topBar = { 25 TopAppBar(26 title = { Text("List of Twice's Best Songs") } 27) 28 } 29) { padding -> 30 LazyColumn(31 contentPadding = padding, 32 modifier = Modifier 33 .fillMaxSize() 34 .padding(8.dp) 35) { 36 items(musicList) { music -> 37 Card(38 modifier = Modifier 39 .fillMaxWidth() 40 .padding(vertical = 10.dp), 41 shape = RoundedCornerShape(16.dp), 42 elevation = CardDefaults.cardElevation(6.dp) 43) {

44	Row(modifier = Modifier.padding(12.dp))
	{
45	GlideImage(
46	model = music.imageUrl,
47	contentDescription =
	music.title,
48	modifier = Modifier
49	.size(120.dp)
50	.clip(RoundedCornerShape(12.dp))
51)
52	Spacer(Modifier.width(16.dp))
53	Column(modifier =
	Modifier.weight(1f)) {
54	Text(
55	music.title,
56	fontSize = 18.sp,
57	style =
	MaterialTheme.typography.titleMedium
58)
59	Text(
60	"Tanggal Rilis:
	`\${music.year}`,
61	fontSize = 14.sp,
62	color =
	MaterialTheme.colorScheme.onBackground.copy(alpha = 0.6f)
63)
64	Spacer(Modifier.height(8.dp))
65	Text(
66	text = "Tentang Lagu Ini:",
67	fontSize = 12.sp,
68	style =
	MaterialTheme.typography.bodySmall.copy(fontWeight =
	FontWeight.Bold)

69)
70	Text(
71	text = music.description,
72	fontSize = 12.sp,
73	maxLines = 3,
74	style =
	MaterialTheme.typography.bodySmall
75)
76	Spacer(Modifier.height(12.dp))
77	Row(
78	modifier =
	Modifier.fillMaxWidth(),
79	horizontalArrangement =
	Arrangement.spacedBy(8.dp)
80) {
81	Button(
82	onClick = {
	onExternalClick(music.externalUrl) },
83	modifier =
	Modifier.weight(1f)
84) {
85	Text("Info")
86	}
87	Button(
88	onClick = {
	onMusicClick(music) },
89	modifier =
	Modifier.weight(1f)
90) {
91	Text("Detail")
92	}
93	}
94	}
95	}

96	}
97	}
98	}
99	}
100	}

Tabel 3. Source Code Jawaban Soal 1 MusicListScreen.kt

4. MusicDetailScreen.kt

1	package com.presca.modul3
2	
3	import androidx.compose.foundation.layout.*
4	import androidx.compose.foundation.lazy.LazyColumn
5	import androidx.compose.foundation.lazy.items
6	import androidx.compose.material3.*
7	import androidx.compose.material.icons.Icons
8	import androidx.compose.material.icons.filled.ArrowBack
9	import androidx.compose.runtime.Composable
10	import androidx.compose.ui.Modifier
11	import androidx.compose.ui.unit.dp
12	import androidx.navigation.NavController
13	import
	com.bumptech.glide.integration.compose.ExperimentalGlideComposeApi
14	import com.bumptech.glide.integration.compose.GlideImage
15	import androidx.compose.foundation.shape.RoundedCornerShape
16	import androidx.compose.ui.draw.clip
17	
18	@OptIn(ExperimentalGlideComposeApi::class,
	ExperimentalMaterial3Api::class)
19	@Composable
20	fun MusicDetailScreen(music: Music, navController:
	NavController) {

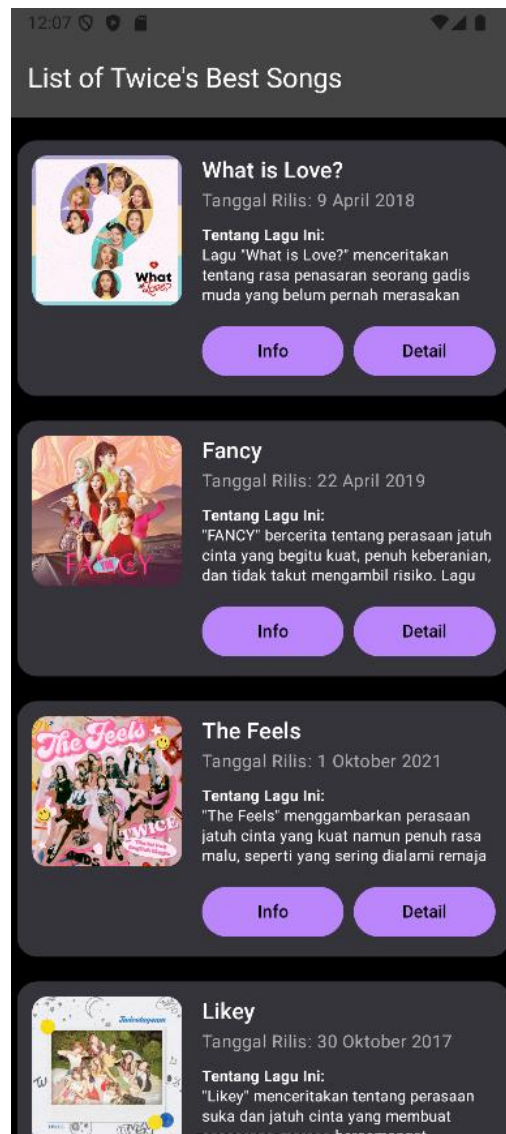
21	Scaffold(
22	topBar = {
23	AppBar(
24	title = {
25	Text(
26	text = music.title,
27	color =
	MaterialTheme.colorScheme.onPrimary
28)
29	},
30	navigationIcon = {
31	IconButton(
32	onClick = {
	navController.popBackStack() },
33	colors =
	IconButtonDefaults.iconButtonColors(
34	contentColor =
	MaterialTheme.colorScheme.onPrimary
35)
36) {
37	Icon(
38	imageVector =
	Icons.Filled.ArrowBack,
39	contentDescription = "Back"
40)
41	}
42	},
43	colors = TopAppBarDefaults.topAppBarColors(
44	containerColor =
	MaterialTheme.colorScheme.surface
45)
46)
47	},
48	containerColor =

	MaterialTheme.colorScheme.background
49) { padding ->
50	LazyColumn(
51	modifier = Modifier
52	.padding(padding)
53	.padding(16.dp)
54	.fillMaxSize(),
55	verticalArrangement =
	Arrangement.spacedBy(12.dp)
56) {
57	item {
58	GlideImage(
59	model = music.imageUrl,
60	contentDescription = music.title,
61	modifier = Modifier
62	.fillMaxWidth()
63	.height(400.dp)
64	.clip(RoundedCornerShape(16.dp))
65)
66	}
67	item {
68	Text("Judul: \${music.title}", style =
	MaterialTheme.typography.titleLarge)
69	}
70	item {
71	Text("Tanggal Rilis: \${music.year}")
72	}
73	item {
74	Text("Tentang Lagu Ini:", style =
	MaterialTheme.typography.titleMedium)
75	}
76	item {
77	Text(
78	text = music.description,

79	style =
	MaterialTheme.typography.bodyMedium
80)
81	}
82	}
83	}
84	}
85	

Tabel 4. Source Code Jawaban Soal 1 MusicDetailScreen.kt

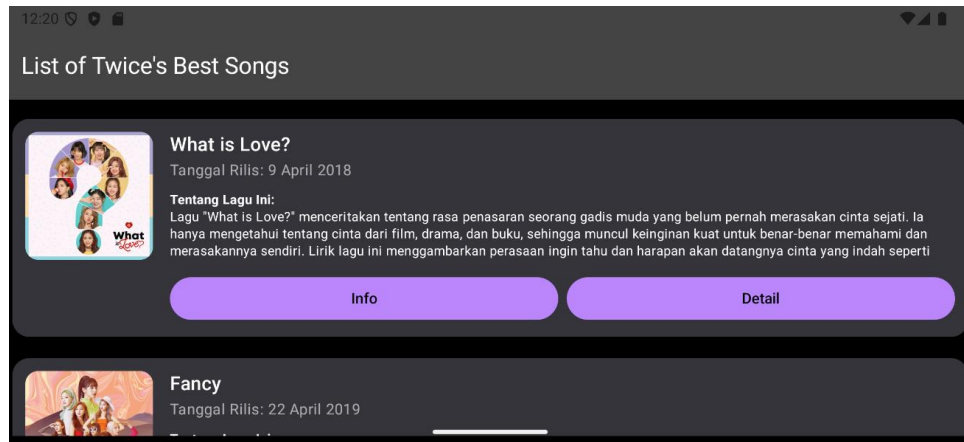
B. Output Program



Gambar 3. Screenshot Hasil Jawaban Soal 1 Tampilan List



Gambar 4. Screenshot Hasil Jawaban Soal 1 Tampilan Detail



Gambar 5. Screenshot Hasil Jawaban Soal 1 Mode Landscape

C. Pembahasan

1) Berikut adalah penjelasan untuk soal nomor 1:

1. MainActivity.kt:

- Pada baris [1], dideklarasikan nama package file Kotlin yang dikelompokkan file ini ke dalam package `com.presca.modul3`.
- Pada baris [3] hingga [15], `import` adalah perintah yang digunakan untuk mengimpor kelas, fungsi, atau objek dari package lain tanpa harus menyebutkan path lengkapnya.
- Pada baris [17], `class MainActivity : AppCompatActivity()` ini digunakan sebagai titik awal aplikasi yang akan mengatur tampilan aplikasi.
- Pada baris [18], `onCreate()` merupakan siklus hidup (lifecycle) dari sebuah activity, yang dimana fungsi ini dipanggil pertama kali saat Activity dibuat.
- Pada baris [19], `super.onCreate(savedInstanceState)` ini memanggil `onCreate()` agar proses inisialisasi standar tetap berjalan.
- Pada baris [20], `enableEdgeToEdge()` digunakan untuk mengaktifkan rendering hingga ke edge layar.
- Pada baris [21], `setContent{...}` digunakan untuk menetapkan tampilan UI dari aplikasi, dan diterapkan tampilan dari `Modul3Theme`.
- Pada baris [22], diterapkan tema khusus aplikasi dari `Modul3Theme`.

- Pada baris [23], `rememberNavController()` digunakan untuk membuat controller navigasi yang mengatur perpindahan antar-screen.
- Pada baris [24], `AppNavigation(navController)` berisi logika navigasi aplikasi.
- Pada baris [31], `AppNavigation` adalah fungsi compose yang digunakan untuk menangani semua screen dan navigasi antar-screen.
- Pada baris [32], `LocalContext.current` ini berisi akses Context Android dari dalam Compose.
- Pada baris [34], `NavHost` merupakan container dari navigasi.
- Pada baris [36], `startDestination` ini digunakan untuk mengatur screen awal adalah "music_list".
- Pada baris [38], didefinisikan "music_list" untuk menampilkan daftar musik.
- Pada baris [39], `MusicListScreen(...)` merupakan fungsi `@Composable` yang bertanggung jawab menampilkan daftar musik.
- Pada baris [40] dan [41], fungsi `onMusicClick` ini akan terpanggil ketika pengguna menekan button "Detail" dan akan menampilkan detail musik.
- Pada baris [43] hingga [45], fungsi `onExternalClick` ini akan terpanggil ketika pengguna menekan button "Info" dan halaman akan berpindah membuka URL di browser.
- Pada baris [50], `composable("detail/{musicId}")` ini digunakan untuk mendefinisikan route untuk tampilan detail musik.
- Pada baris [51], `val music = musicList.find { it.id == musicId }` ini akan mencari music yang sesuai dari `musicList`, yang dimana isi list ini adalah berisi data lagu.
- Pada baris [52], Jika `music` ditemukan (tidak null), maka tampilkan `MusicDetailScreen`. Informasi ini ditampilkan sesuai dengan ID.

2. MusicData.kt:

- Pada baris [1], dideklarasikan nama package file Kotlin yang dikelompokkan file ini ke dalam package `com.presca.modul3`.
- Pada baris [3], didefinisikan data dengan nama `Music`.
- Pada baris [4], didefinisikan juga `id` dalam bentuk integer.
- Pada baris [5] hingga [9], didefinisikan `title`, `year`, `imageUrl`, `externalUrl`, dan `description` dalam bentuk string.
- Pada baris [12], `musicList` ini digunakan untuk membuat list dari musik yang nantinya akan ditampilkan pada `MusicListScreen`.
- Pada baris [13] hingga [19], diisi dari list musik pertama dengan format `id` (sesuai urutan lagu), `title` (judul lagu), `year` (tanggal rilis), `imageUrl` (gambar album), `externalUrl` (link yang dapat dibuka untuk informasi lebih jelas di browser, dan `description` (untuk penjelasan lebih detailnya).
- Pada baris [21] hingga [59], sesuaikan struktur sebelumnya untuk mengisi sisa lagu yang ingin dibuat ke dalam list.

3. **MusicDetailScreen.kt:**

- Pada baris [1], dideklarasikan nama package file Kotlin yang dikelompokkan file ini ke dalam package `com.presca.modul3`.
- Pada baris [3] hingga [16], `import` adalah perintah yang digunakan untuk mengimpor kelas, fungsi, atau objek dari package lain tanpa harus menyebutkan path lengkapnya.
- Pada baris [18], `@OptIn(...)` ini digunakan untuk mengindikasikan jika kita menggunakan API eksperimental (Glide Compose dan Material3).
- Pada baris [20], `MusicDetailScreen` dibuat dengan parameter `music` untuk data musik, dan `navController` untuk navigasi antar screen.
- Pada baris [21], `Scaffold` digunakan untuk membuat struktur layout utama.
- Pada baris [22], `topBar` digunakan untuk mendefinisikan bar di bagian atas.

- Pada baris [24], `title` nantinya pada bagian ini digunakan untuk menampilkan judul lagu pada bar atas, dan `text` nantinya digunakan untuk menampilkan teks pada judul.
- Pada baris [30], bagian `navigationIcon` ini digunakan untuk mengatur tombol navigasi (seperti tombol kembali).
- Pada baris [32], `onClick` pada bagian ini, `navController.popBackStack()` digunakan untuk kembali ke screen sebelumnya dalam stack navigasi
- Pada baris [38], pada bagian `Icon`, `imageVector = Icons.Filled.ArrowBack`: ini artinya kita menggunakan ikon panah kembali sebagai icon dari tombol kembali.
- Pada baris [43] dan [44], `colors` pada bagian ini digunakan untuk mengatur warna pada top bar.
- Pada baris [48], `containerColor` digunakan untuk mengatur warna latar belakang pada scaffold.
- Pada baris [50], `LazyColumn` merupakan sebuah composable yang digunakan untuk menampilkan daftar item yang dapat digulir secara vertikal.
- Pada baris [51] hingga [54], `modifier` pada bagian ini digunakan untuk mengatur layout dari `LazyColumn`.
- Pada baris [55], `Arrangement.spacedBy(12.dp)` ini digunakan untuk memberi jarak 12dp secara vertikal antar item.
- Pada baris [57], `item` pada bagian ini digunakan untuk mengatur ukuran dan bentuk dari cover musik.
- Pada baris [58], `GlideImage` ini merupakan komponen yang digunakan untuk menampilkan gambar dari URL menggunakan library `Glide`.
- Pada baris [67] hingga [79], `item` pada bagian tersebut mengatur kembali untuk tampilan teks dari judul, tanggal rilis, tentang lagu ini, dan teks deskripsi.

4. **MusicListScreen.kt:**

- Pada baris [1], dideklarasikan nama package file Kotlin yang dikelompokkan file ini ke dalam package `com.presca.modul3`.
- Pada baris [3] hingga [15], `import` adalah perintah yang digunakan untuk mengimpor kelas, fungsi, atau objek dari package lain tanpa harus menyebutkan path lengkapnya.
- Pada baris [18], `@OptIn(...)` ini digunakan untuk mengindikasikan jika kita menggunakan API eksperimental (Glide Compose dan Material3).
- Pada baris [19] hingga [21], `MusicListScreen` dibuat dengan parameter callback `onMusicClick` dan `onExternalClick` untuk interaksi pengguna.
- Pada baris [23], `Scaffold` digunakan untuk membuat struktur layout utama.
- Pada baris [24], `topBar` digunakan untuk mendefinisikan bar di bagian atas.
- Pada baris [26], `title` nantinya pada bagian ini digunakan untuk menampilkan judul lagu pada bar atas, dan `text` nantinya digunakan untuk menampilkan teks pada bar atas.
- Pada baris [30], `LazyColumn` merupakan sebuah composable yang digunakan untuk menampilkan daftar item yang dapat digulir secara vertikal.
- Pada baris [32] hingga [34], `modifier` pada bagian ini digunakan untuk mengatur layout dari `LazyColumn`.
- Pada baris [36], `items(musicList)` ini merupakan loop melalui daftar musik untuk membuat item.
- Pada baris [37], `Card` digunakan sebagai komponen Material Design yang digunakan untuk menampilkan konten terkelompok.
- Pada baris [39], `fillMaxWidth()` ini digunakan agar lebar mengisi parent.
- Pada baris [41], `RoundedCornerShape(16.dp)` ini digunakan untuk mengatur sudut kelengkungan dengan radius 16dp
- Pada baris [42], `cardElevation(6.dp)` digunakan untuk mengatur efek bayangan dengan elevasi 6dp.

- Pada baris [44], `Row` digunakan untuk layout horizontal untuk gambar dan juga teks informasi.
- Pada baris [45], `GlideImage` digunakan untuk menampilkan gambar cover dari URL.
- Pada baris [54], `Text` pada bagian ini akan mengatur tampilan dari judul.
- Pada baris [65] hingga [74], `Text` pada bagian tersebut mengatur kembali untuk tampilan teks dari tanggal rilis, tentang lagu ini, dan teks deskripsi.
- Pada baris [77] hingga [91], bagian ini digunakan untuk mengatur tampilan dari button “Info” dan “Detail” serta fungsi dari button tersebut pada saat diklik.

2) `RecyclerView` masih banyak digunakan karena `RecyclerView` stabil, fleksibel, dan juga mempunyai ekosistem library yang matang. Meskipun penulisan kodenya yang panjang dan bersifat boiler-plate, `RecyclerView` mendukung berbagai fitur lanjutan seperti multiple view types, custom item decoration, dan juga animasi kompleks. Selain itu, banyak aplikasi lama yang masih menggunakan XML sehingga tetap mengandalkan `RecyclerView` untuk mempertahankan kompatibilitas dan performa. `LazyColumn` belum sepenuhnya menggantikan fleksibilitas `RecyclerView` dalam beberapa kasus yang spesifik.

Kelebihan `RecyclerView` dibanding `LazyColumn`:

- Stabil dan digunakan secara luas dalam produksi.
- Mendukung multiple view types dan pengaturan layout yang kompleks.
- Kompatibel dengan pendekatan UI berbasis XML yang masih digunakan secara luas.
- Banyak library pihak ketiga yang sudah terintegrasi dengan `RecyclerView`.
- Performa sangat baik dalam menangani daftar data besar dengan efisiensi tinggi.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Prescaa/Kuliah/tree/master/Praktikum%20Mobile/MODUL%203>