

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



Build a Scrollable List

Oleh:

Muhammad Rizky

NIM. 2310817310011

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 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 : Muhammad Rizky
NIM : 2310817310011

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar
NIM. 2210817210026

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	5
SOAL 1	6
A. Source Code.....	6
B. Output Program	15
C. Pembahasan	17
D. Tautan Git	24
SOAL 2.....	25
A. Jawaban	25

DAFTAR GAMBAR

Gambar 1. Screenshot Halaman Awal.....	15
Gambar 2. Screenshot Ketika Tombol BWF Ditekan Mengarah Ke Website	16
Gambar 3. Screenshot Ketika Tombol Detail Ditekan	16

DAFTAR TABEL

Table 1. Source Code MainActivity.kt.....	6
Table 2. Source Code DataTournament.kt	7
Table 3. Source Code NaavGraph.kt	9
Table 4. Souce Code ListScreen.kt.....	10
Table 5. Source Code DetailScreen.kt.....	12

SOAL 1

Soal Praktikum:

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

A. Source Code

1. MainActivity.kt

Table 1. Source Code MainActivity.kt

1	package com.example.modul_3
2	
3	import android.os.Bundle
4	import androidx.activity.ComponentActivity
5	import androidx.activity.compose.setContent
6	import androidx.compose.runtime.Composable
7	import androidx.compose.ui.tooling.preview.Preview
8	import com.example.modul_3.navigation.NavGraph
9	
10	class MainActivity : ComponentActivity() {
11	override fun onCreate(savedInstanceState: Bundle?) {
12	super.onCreate(savedInstanceState)

13	setContent {
14	NavGraph()
15	}
16	}
17	}
18	
19	@Preview(showBackground = true)
20	@Composable
21	fun Preview() {
22	NavGraph()
23	}

2. DataTournament.kt

Table 2. Source Code DataTournament.kt

1	package com.example.modul_3.data
2	
3	data class ListItem(
4	val title: String, val desc: String, val imageUrl: String,
5	val linkUrl: String)
6	
7	val sampleItems = listOf(
8	ListItem("Indonesia Open", "Indonesia Open adalah
9	sebuah kejuaraan bulu tangkis internasional di Indonesia.
10	Kejuaraan ini diselenggarakan oleh Persatuan Bulu Tangkis
11	Seluruh Indonesia dengan edisi terkini berlangsung di
12	Istora Gelora
13	Bung Karno, Jakarta.", "https://img.bwfbadminton.com/image/u
14	pload/v1746_607228/assets/tournaments/logo/8E66BAA2-CF0F
15	-4FFA-8798-FF700D1CDBEC.png", "https://bwfworldtour.bwfb
16	adminton.com/tournament/5236/indonesia-open-2025/results/2
17	025-06-03"),
18	ListItem("Yonex All England Open", "Turnamen bulutangkis
19	tertua di dunia yang diadakan di Inggris sejak 1899.
20	Saat ini menjadi bagian dari BWF World Tour Super 1000
21	dan sangat bergengsi karena Sejarah
22	dan prestisenya.", "https://img.bwfbadminton.com/image
23	/upload
24	/v1734076435/assets/tournaments/logo/F6304F7C-D298-4E3D
25	AA62A6B933CBB8C7.png", "https://bwfworldtour.bwfbadminton.com/
26	tournament/5238/yonex-all-england-open-badminton-champions
27	hips-2025/results/podium/"),
28	ListItem("Victor China Open", "Turnamen bulutangkisintern
29	asional bergengsi yang diadakan setiap tahun di Tiongkok.
30	Merupakan bagian dari BWF World Tour Super 1000 (bersama
31	All England dan Indonesia Open).
32	Menarik banyak pemain top dunia karena

33	hadiah besar dan poin peringkat tinggi.", " https://img.bwfbadminton.com/image/upload/v1746149516/assets/tournaments/logo/5A59A1E7-1366-49BA-863D-33FB844D51C0.png ", " https://bwfworldtour.bwfbadminton.com/tournament/5281/victor-china-open-2025/results/2025-07-22 "),
34	
35	ListItem("Daihatsu Indonesia Master", "Indonesia Master adalah sebuah kejuaraan bulu tangkis internasional di Indonesia. Kejuaraan ini diselenggarakan oleh Persatuan Bulu Tangkis Seluruh Indonesia dengan edisi terkini berlangsung di Istora Gelora Bung Karno, Jakarta.", " https://img.bwfbadminton.com/image/upload/v1731292348/assets/tournaments/logo/C166BF11-2128-4F8C-B1ED-F30565DE9089.png ", " https://bwfworldtour.bwfbadminton.com/tournament/5234/daihatsu-indonesia-masters-2025/results/podium/ "),
36	
37	ListItem("HSBC BWF World Tour Final", "Turnamen penutup musim bulutangkis yang hanya mempertemukan 8 pemain atau pasangan terbaik dunia dari setiap sektor berdasarkan perolehan poin sepanjang musim BWF World Tour. Turnamen ini sangat bergengsi dan berhadiah besar, setara dengan kejuaraan elit lainnya.", " https://img.bwfbadminton.com/image/upload/v1734514049/assets/tournaments/logo/F34EBDDDB-AA5D-4F1E-A14F-A94473DD1295.png ", " https://bwfworldtourfinals.bwfbadminton.com/results/5259/hsbc-bwf-world-tour-finals-2025/draw/ "),
38	
39	ListItem("TotalEnergies BWF Sudirman Cup Finals", "Sudirman Cup adalah kejuaraan bulu tangkis internasional untuk nomor beregu campuran, mempertandingkan nomor tunggal putra, tunggal putri, ganda putra, ganda putri, dan ganda campuran. Kejuaraan ini digelar setiap dua tahun sekali.", " https://img.bwfbadminton.com/image/upload/v1734511970/assets/tournaments/logo/C9F72B3B-2C6A-4D63-A782-78968933BDE9.png ", " https://bwfsudirmancup.bwfbadminton.com/results/5260/totalenergies-bwf-sudirman-cup-finals-2025/podium "),
40	
41	ListItem("TotalEnergies BWF Sudirman Cup Finals", "Sudirman Cup adalah kejuaraan bulu tangkis inte
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	
74	
75	
76	
77	
78	
79	

80	rnasional untuk nomor beregu campuran, mempertandingkan
81	nomor tunggal putra, tunggal putri, ganda putra, ganda
82	putri, dan ganda campuran. Kej
83	uaraan ini digelar setiap dua tahun sekali.", "https://
84	img.bwfbadminton.com/image/upload/v1
85	734511970/assets/tournaments/logo/C9F72B3B-2C6A-4D63-A782-
86	78968933BDE9.png", "https://bwfsudirmancup.bwfbadm
87	inton.com/results/5260/totalenergies-bwf-sudirman-cup-finals
88	-2025/podium"),
89	ListItem("TotalEnergies BWF Thomas & Uber Cup Finals",
90	"Thomas Cup dan Uber Cup adalah kejuaraan bulutangkis
91	beregu putra dan putri antarnegara yang diselenggarakan
92	dua tahun sekali oleh BWF. Setiap tim terdiri dari pemain
93	tunggal dan ganda, dan bertanding untuk menjadi tim
94	nasional putra dan putri terbaik di dunia",
95	"https://bwfthomasubercups.bwfbadminton.com/wp-conten
96	t/plugins/bwf-menu-
97	system/images/TUCTournamentLogoLandscape.png",
98	"https://bwfthomasubercups.bwfbadminton.com/results/5
99	147/totalenergies-bwf-thomas-uber-cup-finals-2024/podium"),
100)

3. NavGraph.kt

Table 3. Source Code NaavGraph.kt

1	package com.example.modul_3.navigation
2	
3	import androidx.compose.runtime.Composable
4	import androidx.navigation.NavType
5	import androidx.navigation.compose.NavHost
6	import androidx.navigation.compose.composable
7	import androidx.navigation.compose.rememberNavController
8	import androidx.navigation.navArgument
9	import com.example.modul_3.screen.DetailScreen
10	import com.example.modul_3.screen.ListScreen
11	
12	@Composable
13	fun NavGraph() {
14	val navController = rememberNavController()
15	
16	NavHost(navController = navController,
17	startDestination = "list") {
18	composable("list") {
19	ListScreen(navController)
20	}
21	composable(

22	"detail/{itemTitle}",
23	arguments = listOf(navArgument("itemTitle") {
24	type = NavType.StringType })
25) { backStackEntry ->
26	val itemTitle =
27	backStackEntry.arguments?.getString("itemTitle") ?: "No
28	Title"
29	DetailScreen(itemTitle)
30	}
31	}
32	}
33	

4. ListScreen.kt

Table 4. Source Code ListScreen.kt

1	package com.example.modul_3.screen
2	
3	import android.content.Intent
4	import android.net.Uri
5	import androidx.compose.foundation.layout.Arrangement
6	import androidx.compose.foundation.layout.Column
7	import androidx.compose.foundation.layout.Row
8	import androidx.compose.foundation.layout.Spacer
9	import androidx.compose.foundation.layout.fillMaxSize
10	import androidx.compose.foundation.layout.fillMaxWidth
11	import androidx.compose.foundation.layout.padding
12	import androidx.compose.foundation.layout.size
13	import androidx.compose.foundation.layout.width
14	import androidx.compose.foundation.lazy.LazyColumn
15	import androidx.compose.foundation.lazy.items
16	import
17	androidx.compose.foundation.shape.RoundedCornerShape
18	import androidx.compose.material3.Button
19	import androidx.compose.material3.Card
20	import androidx.compose.material3.MaterialTheme
21	import androidx.compose.material3.Text
22	import androidx.compose.runtime.Composable
23	import androidx.compose.ui.Alignment
24	import androidx.compose.ui.Modifier
25	import androidx.compose.ui.draw.clip
26	import androidx.compose.ui.layout.ContentScale
27	import androidx.compose.ui.platform.LocalContext
28	import androidx.compose.ui.unit.dp
29	import androidx.navigation.NavController
30	import coil.compose.AsyncImage
31	import com.example.modul_3.data.sampleItems

```

32
33 @Composable
34 fun ListScreen(navController: NavController) {
35     val context = LocalContext.current
36
37     LazyColumn(
38         modifier = Modifier
39             .fillMaxSize()
40             .padding(8.dp)
41     ) {
42         items(sampleItems) { item ->
43             Card(
44                 shape = RoundedCornerShape(16.dp),
45                 modifier = Modifier
46                     .fillMaxWidth()
47                     .padding(8.dp)
48             ) {
49                 Row(
50                     modifier = Modifier
51                         .padding(16.dp)
52                         .fillMaxWidth(),
53                     verticalAlignment =
54 Alignment.CenterVertically
55                 ) {
56                     AsyncImage(
57                         model = item.imageUrl,
58                         contentDescription = item.title,
59                         modifier = Modifier
60                             .size(100.dp)
61
62                     .clip(RoundedCornerShape(12.dp)),
63                     contentScale = ContentScale.Crop
64                 )
65
66                 Spacer(modifier =
67 Modifier.width(16.dp))
68
69                 Column(
70                     modifier = Modifier.weight(1f),
71                     verticalArrangement =
72 Arrangement.spacedBy(8.dp)
73                 ) {
74                     Text(text = item.title, style =
75 MaterialTheme.typography.titleMedium)
76                     Text(text = item.desc, style =
77 MaterialTheme.typography.bodySmall)
78

```

79	Row (
80	horizontalArrangement =
81	Arrangement.spacedBy(8.dp)
82) {
83	Button(onClick = {
84	val intent =
85	Intent(Intent.ACTION_VIEW, Uri.parse(item.linkUrl)).apply
86	{
87	
88	addFlags(Intent.FLAG_ACTIVITY_NEW_TASK)
89	}
90	
91	context.startActivity(intent)
92)) {
93	Text("BWF")
94	}
95	
96	Button(onClick = {
97	
98	navController.navigate("detail/\${item.title}")
99)) {
100	Text("Detail")
101	}
102	}
103	}
104	}
105	}
106	}
107	}
108	}

5. DetailScreen.kt

Table 5. Source Code DetailScreen.kt

1	package com.example.modul_3.screen
2	
3	import androidx.compose.foundation.layout.Box
4	import androidx.compose.foundation.layout.Column
5	import androidx.compose.foundation.layout.Spacer
6	import androidx.compose.foundation.layout.fillMaxSize
7	import androidx.compose.foundation.layout.fillMaxWidth
8	import androidx.compose.foundation.layout.height
9	import androidx.compose.foundation.layout.padding
10	import androidx.compose.foundation.layout.wrapContentSize
11	import
12	androidx.compose.foundation.shape.RoundedCornerShape

```

13 import androidx.compose.material3.Card
14 import androidx.compose.material3.MaterialTheme
15 import androidx.compose.material3.Text
16 import androidx.compose.runtime.Composable
17 import androidx.compose.ui.Alignment
18 import androidx.compose.ui.Modifier
19 import androidx.compose.ui.draw.clip
20 import androidx.compose.ui.graphics.Color
21 import androidx.compose.ui.layout.ContentScale
22 import androidx.compose.ui.text.font.FontWeight
23 import androidx.compose.ui.unit.dp
24 import coil.compose.AsyncImage
25 import com.example.modul_3.data.sampleItems
26
27 @Composable
28 fun DetailScreen(itemTitle: String) {
29     val item = sampleItems.firstOrNull { it.title ==
30 itemTitle }
31
32     item?.let {
33         Box(
34             contentAlignment = Alignment.Center,
35             modifier =
36 Modifier.fillMaxSize().padding(16.dp)
37         ) {
38             Column(horizontalAlignment =
39 Alignment.CenterHorizontally) {
40                 Card(
41                     shape = RoundedCornerShape(16.dp),
42                     modifier = Modifier
43                         .fillMaxWidth()
44                         .height(250.dp)
45                         .clip(RoundedCornerShape(16.dp))
46                 ) {
47                     AsyncImage(
48                         model = it.imageUrl,
49                         contentDescription = itemTitle,
50                         modifier = Modifier
51                             .fillMaxSize()
52
53 .clip(RoundedCornerShape(16.dp)),
54                         contentScale = ContentScale.Crop
55                     )
56                 }
57
58                 Spacer(modifier = Modifier.height(24.dp))
59

```

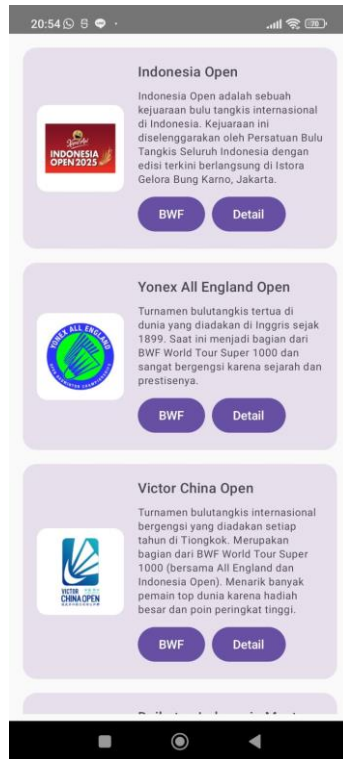
```

60         Text(
61             text = it.title,
62             style                                     =
63     MaterialTheme.typography.titleMedium.copy(
64                 fontWeight = FontWeight.Bold,
65                 color                                     =
66     MaterialTheme.colorScheme.primary
67             ),
68             modifier = Modifier
69                 .fillMaxWidth()
70                 .padding(horizontal = 16.dp)
71
72     .align(Alignment.CenterHorizontally)
73     )
74
75     Spacer(modifier = Modifier.height(8.dp))
76
77     Text(
78         text = "Description:",
79         style                                     =
80     MaterialTheme.typography.bodyMedium.copy(
81                 fontWeight = FontWeight.SemiBold,
82                 color                                     =
83     MaterialTheme.colorScheme.secondary
84             ),
85             modifier                                     =
86     Modifier.padding(horizontal = 16.dp)
87             )
88
89     Text(
90         text = it.desc,
91         style                                     =
92     MaterialTheme.typography.bodyLarge,
93         modifier                                     =
94     Modifier.padding(horizontal = 16.dp)
95             )
96     }
97 }
98 } ?: run {
99     Text(
100         text = "Item tidak ditemukan",
101         style                                     =
102     MaterialTheme.typography.bodyLarge.copy(
103                 fontWeight = FontWeight.Bold,
104                 color = Color.Red
105             ),
106         modifier = Modifier

```

107	<code>.fillMaxSize()</code>
108	<code>.wrapContentSize(Alignment.Center)</code>
109	<code>)</code>
110	<code>}</code>
111	<code>}</code>

B. Output Program



Gambar 1. Screenshot Halaman Awal



Gambar 2. Screenshot Ketika Tombol BWF Ditekan Mengarah Ke Website



Gambar 3. Screenshot Ketika Tombol Detail Ditekan

C. Pembahasan

1. MainActivity.kt:

Line 1, Mendeklarasikan nama package `com.example.modul_3` sebagai struktur direktori tempat file berada dalam proyek

Line 3, Mengimpor `Bundle` dari `android.os` untuk menyimpan dan mengirim data antar komponen saat aktivitas dibuat atau dijalankan kembali

Line 4, Mengimpor `ComponentActivity` dari `androidx.activity` sebagai kelas dasar untuk aktivitas yang menggunakan Jetpack Compose

Line 5, Mengimpor fungsi `setContent` dari `androidx.activity.compose` untuk menetapkan isi UI aktivitas menggunakan deklarasi Compose

Line 6, Mengimpor anotasi `@Composable` dari `androidx.compose.runtime` untuk menandai fungsi sebagai fungsi komposisi dalam Jetpack Compose

Line 7, Mengimpor anotasi `@Preview` dari `androidx.compose.ui.tooling.preview` untuk menampilkan pratinjau komponen Compose di Android Studio

Line 8, Mengimpor `NavGraph` dari `com.example.modul_3.navigation` yang berisi pengaturan navigasi antar layar dalam aplikasi

Line 10, Mendeklarasikan kelas `MainActivity` sebagai turunan dari `ComponentActivity`, menjadi entry point utama aplikasi

Line 11, Fungsi `onCreate` dioverride dari `ComponentActivity` dan dipanggil saat aktivitas pertama kali dibuat

Line 12, Memanggil `super.onCreate(savedInstanceState)` untuk memastikan inisialisasi komponen aktivitas dari superclass

Line 13, Menggunakan fungsi `setContent` untuk menetapkan isi UI menggunakan Jetpack Compose

Line 14, Memanggil `MainActivity`

Line 19, Anotasi `@Preview(showBackground = true)` digunakan untuk menampilkan pratinjau UI dengan latar belakang putih

Line 20, Mendeklarasikan fungsi `Preview()` sebagai fungsi Compose untuk menampilkan pratinjau dari `NavGraph` di Android Studio

Line 21, Fungsi NavGraph() dipanggil di dalam fungsi Preview agar struktur navigasi bisa dilihat dalam mode pratinjau

Line 22, Kurung kurawal penutup untuk menutup fungsi Preview

2. DataTournament.kt

Line 1, Mendeklarasikan nama package com.example.modul_3.data sebagai lokasi file dalam struktur proyek

Line 3, Mendefinisikan data class ListItem dengan empat properti: title, desc, imageUrl, dan linkUrl yang bertipe String untuk merepresentasikan data turnamen

Line 5, Mendeklarasikan variabel sampleItems bertipe List yang berisi kumpulan objek ListItem

Line 6-14, Menambahkan objek ListItem pertama dengan informasi tentang turnamen Indonesia Open meliputi judul, deskripsi, URL gambar, dan link situs

Line 15-21, Menambahkan objek ListItem kedua tentang Yonex All England Open dengan data lengkap seperti objek sebelumnya

Line 22-28, Menambahkan objek ListItem ketiga tentang Victor China Open, turnamen bergengsi di Tiongkok

Line 29-35, Menambahkan objek ListItem keempat mengenai Daihatsu Indonesia Master yang diselenggarakan di Jakarta

Line 36-42, Menambahkan objek ListItem kelima tentang HSBC BWF World Tour Final, turnamen penutup musim untuk delapan pemain atau pasangan terbaik

Line 43-49, Menambahkan objek ListItem keenam mengenai TotalEnergies BWF Sudirman Cup Finals, turnamen beregu campuran

Line 50-56, Menambahkan kembali objek ListItem yang identik dengan sebelumnya yaitu tentang TotalEnergies BWF Sudirman Cup Finals

Line 57-63, Menambahkan objek ListItem kedelapan tentang TotalEnergies BWF Thomas & Uber Cup Finals, kejuaraan beregu putra dan putri antarnegara

3. NavGraph.kt

Line 1, Mendeklarasikan nama package com.example.modul_3.navigation sebagai lokasi file dalam struktur proyek

Line 3, Mengimpor Composable dari `androidx.compose.runtime` untuk menandai fungsi sebagai bagian dari UI deklaratif Jetpack Compose

Line 4, Mengimpor NavType dari `androidx.navigation` untuk menentukan tipe data dari argumen navigasi

Line 5, Mengimpor NavHost dari `androidx.navigation.compose` sebagai kontainer utama untuk mendefinisikan rute navigasi

Line 6, Mengimpor composable dari `androidx.navigation.compose` untuk mendefinisikan layar yang dapat dinavigasikan dalam NavHost

Line 7, Mengimpor `rememberNavController` untuk membuat dan mengingat instance NavController sebagai pengatur navigasi

Line 8, Mengimpor `navArgument` dari `androidx.navigation` untuk mendefinisikan argumen yang digunakan pada rute tertentu

Line 9, Mengimpor `DetailScreen` dari `package com.example.modul_3.screen` sebagai layar detail suatu item

Line 10, Mengimpor `ListScreen` dari `package com.example.modul_3.screen` sebagai layar daftar awal item

Line 12, Mendeklarasikan fungsi `NavGraph()` dengan anotasi `@Composable` untuk membangun UI menggunakan Jetpack Compose

Line 13, Membuat variabel `navController` menggunakan `rememberNavController()` untuk mengatur navigasi

Line 15, Menggunakan NavHost untuk menentukan rute navigasi dengan `navController` dan `startDestination` yang mengarah ke "list"

Line 16, Mendefinisikan rute `composable("list")` untuk memanggil fungsi `ListScreen` saat rute ini aktif

Line 17, Memanggil fungsi `ListScreen(navController)` untuk menampilkan daftar item dengan dukungan navigasi

Line 18, Mendefinisikan rute `composable("detail/{itemTitle}")` dengan parameter `itemTitle` yang diterima dari layar sebelumnya

Line 19, Menggunakan `navArgument("itemTitle")` untuk mendeklarasikan argumen bertipe String yang diperlukan oleh rute detail

Line 20, Menandai blok lambda dari rute detail untuk menerima `backStackEntry` sebagai data dari rute sebelumnya

Line 21, Mengambil nilai dari argumen `itemTitle` melalui `backStackEntry` atau menggunakan nilai default "No Title" jika tidak ditemukan

Line 22, Memanggil fungsi `DetailScreen(itemTitle)` untuk menampilkan layar detail berdasarkan argumen yang diterima

4. ListScreen.kt

Pada line 1, dideklarasikan nama package `com.example.modul_3.screen` sebagai lokasi file ini berada dalam struktur proyek.

Pada line 3, diimpor `Intent` dari Android untuk membuat aksi berpindah ke aplikasi lain (misal membuka URL di browser).

Pada line 4, diimpor `Uri` dari Android untuk menangani data URI pada `Intent`.

Pada line 5–14, diimpor berbagai komponen layout dari Jetpack Compose seperti `Column`, `Row`, `Spacer`, `LazyColumn`, dll, yang digunakan untuk menyusun tampilan UI secara deklaratif.

Pada line 15, diimpor `RoundedCornerShape` untuk membuat sudut komponen berbentuk membulat.

Pada line 16–18, diimpor komponen Material3 seperti `Button`, `Card`, dan `Text` untuk membangun antarmuka material design.

Pada line 19, diimpor anotasi `@Composable` agar fungsi dapat digunakan sebagai komponen UI dalam Compose.

Pada line 20–24, diimpor utilitas Compose tambahan, seperti `Modifier`, `Alignment`, `ContentSize`, dan unit pengukuran `dp`.

Pada line 25, diimpor `LocalContext` untuk mengakses `Context` dari Android dalam komposisi.

Pada line 26, diimpor `NavController` dari `androidx.navigation` untuk menangani navigasi antar layar.

Pada line 27, diimpor `AsyncImage` dari library `Coil` untuk memuat gambar dari URL secara asynchronous.

Pada line 28, diimpor `sampleItems` dari `package com.example.modul_3.data` sebagai data yang akan ditampilkan dalam daftar.

Pada line 30, dideklarasikan fungsi `ListScreen(navController: NavController)` sebagai layar daftar yang bisa dinavigasikan dan menerima parameter `NavController`.

Pada line 31, dibuat variabel `context` dari `LocalContext.current` untuk mendapatkan `context` saat ini dari komposisi.

Pada line 33, digunakan `LazyColumn` sebagai komponen scrollable untuk menampilkan daftar secara efisien.

Pada line 34–36, diberikan `Modifier` untuk mengisi seluruh ukuran dan memberikan `padding 8dp` ke `LazyColumn`.

Pada line 37, digunakan fungsi `items(sampleItems)` untuk mengiterasi semua data dari `sampleItems`.

Pada line 38, dideklarasikan `Card` untuk setiap item yang dibungkus dalam desain kartu.

Pada line 39–41, diberikan bentuk sudut membulat dan `padding` pada `Card` agar tampilannya lebih rapi dan modern.

Pada line 42, dideklarasikan `Row` untuk menampilkan gambar dan teks secara horizontal sejajar.

Pada line 43–45, digunakan `Modifier` untuk `padding` dan memenuhi lebar maksimum, serta menyetel perataan vertikal ke tengah (`Alignment.CenterVertically`).

Pada line 46, digunakan `AsyncImage` untuk memuat gambar dari URL item.

Pada line 47–51, diberikan atribut model gambar, deskripsi konten, ukuran, `clipping bundar`, dan skala `isi crop` agar gambar tampil dengan proporsi yang bagus.

Pada line 53, digunakan `Spacer` untuk memberi jarak horizontal antara gambar dan teks.

Pada line 55, dibuat `Column` untuk menampilkan teks dan tombol secara vertikal.

Pada line 56, digunakan `Modifier.weight(1f)` agar kolom mengambil sisa ruang, dan `Arrangement.spacedBy(8.dp)` untuk memberi jarak antar elemen di dalam kolom.

Pada line 57, ditampilkan judul item dengan style `titleMedium` dari `MaterialTheme`.

Pada line 58, ditampilkan deskripsi item dengan style `bodySmall`.

Pada line 60, dibuat `Row` lagi untuk menampung dua tombol dalam satu baris horizontal.

Pada line 61, digunakan `Arrangement.spacedBy(8.dp)` agar tombol memiliki jarak antar satu sama lain.

Pada line 62–66, tombol "BWF" dibuat, yang ketika ditekan akan membuka link URL menggunakan `Intent.ACTION_VIEW`.

Pada line 67, tombol "Detail" dibuat, yang ketika ditekan akan berpindah ke layar detail berdasarkan judul item.

Pada line 68, menggunakan `navController.navigate("detail/${item.title}")` untuk melakukan navigasi dan mengirim judul item sebagai argumen.

Pada line 69, teks tombol "Detail" ditampilkan.

5. DetailScreen.kt

Pada line 1, dideklarasikan nama package `com.example.modul_3.screen` sebagai lokasi file ini berada dalam struktur proyek.

Pada line 3–19, diimpor berbagai komponen dari Jetpack Compose seperti `Box`, `Column`, `Spacer`, `Card`, dan lain-lain yang akan digunakan untuk layout dan desain UI.

Pada line 20, diimpor `AsyncImage` dari `Coil` untuk menampilkan gambar secara asynchronous.

Pada line 21, diimpor `sampleItems` dari package `com.example.modul_3.data`, yang berisi data item yang akan ditampilkan pada layar detail.

Pada line 23, dideklarasikan fungsi `DetailScreen(itemTitle: String)` yang menerima parameter `itemTitle` untuk mencari item sesuai judulnya.

Pada line 24, digunakan `sampleItems.firstOrNull { it.title == itemTitle }` untuk mencari item dengan judul yang sesuai dalam daftar `sampleItems`. Jika tidak ditemukan, item akan bernilai `null`.

Pada line 26–27, menggunakan `item?.let { ... }` untuk memeriksa apakah item ditemukan. Jika ditemukan, blok kode di dalam `let` akan dieksekusi.

Pada line 28, menggunakan `Box` untuk menyusun konten secara pusat, dengan `contentAlignment = Alignment.Center` untuk memastikan semua konten berada di tengah layar. `Modifier.fillMaxSize().padding(16.dp)` digunakan agar `Box` mengisi seluruh layar dengan padding 16dp.

Pada line 30–31, menggunakan `Column` untuk menampilkan elemen-elemen secara vertikal dengan perataan horizontal di tengah.

Pada line 32, mendeklarasikan Card untuk menampilkan gambar dengan bentuk sudut membulat dan ukuran tinggi 250dp.

Pada line 34, digunakan AsyncImage untuk memuat gambar dari URL yang ada pada item. ContentScale.Crop digunakan untuk memotong gambar agar mengisi ukuran yang tersedia.

Pada line 38, digunakan Spacer(modifier = Modifier.height(24.dp)) untuk memberi jarak vertikal sebesar 24dp setelah gambar.

Pada line 40–41, digunakan Text untuk menampilkan judul item dengan gaya MaterialTheme.typography.titleMedium, memberi bobot font Bold, dan warna primer dari tema material.

Pada line 43, digunakan Spacer(modifier = Modifier.height(8.dp)) untuk memberi jarak vertikal setelah judul item.

Pada line 45, digunakan Text untuk menampilkan label "Description:" dengan gaya MaterialTheme.typography.bodyMedium, bobot font SemiBold, dan warna sekunder dari tema material.

Pada line 47, digunakan Text untuk menampilkan deskripsi item, menggunakan gaya MaterialTheme.typography.bodyLarge.

Pada line 50, digunakan } untuk menutup blok item?.let { ... } yang berarti tampilan detail sudah selesai ditampilkan.

Pada line 52, digunakan blok ?: run { ... } untuk menangani kasus ketika item tidak ditemukan. Jika item tidak ditemukan, akan menampilkan teks "Item tidak ditemukan".

Pada line 54, digunakan Text untuk menampilkan pesan error dengan gaya MaterialTheme.typography.bodyLarge, bobot font Bold, dan warna merah, serta memastikan teks ditampilkan di tengah layar dengan Modifier.fillMaxSize().wrapContentSize(Alignment.Center).

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/kkeshiiian/Pemrograman-Mobile>

SOAL 2

Mengapa RecyclerView masih digunakan, padahal RecyclerView memiliki kode yang panjang dan bersifat boiler-plate, dibandingkan LazyColumn dengan kode yang lebih singkat?

A. Jawaban

Pertama, banyak proyek Android masih menggunakan View System, terutama aplikasi yang sudah lama dikembangkan, sehingga migrasi ke Jetpack Compose belum menjadi prioritas. RecyclerView sudah lama menjadi komponen andalan dan memiliki ekosistem pendukung yang sangat matang, seperti integrasi dengan DiffUtil, Paging, dan berbagai LayoutManager yang memberikan fleksibilitas tinggi dalam menampilkan data dalam berbagai format. Selain itu, RecyclerView memberikan kontrol dan kustomisasi lebih banyak terhadap perilaku tampilan, animasi, dan performa, yang belum sepenuhnya bisa digantikan oleh LazyColumn dalam beberapa kasus. Jetpack Compose sendiri masih dalam tahap penyempurnaan dan meskipun produktivitas lebih tinggi dalam pengembangan UI, penggunaannya lebih cocok untuk proyek baru yang sepenuhnya mengadopsi arsitektur deklaratif.