

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



BUILD A SCROLLABLE LIST

Oleh:

Jovan Gilbert Natamasindah NIM. 2310817310002

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN I
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 : Jovan Gilbert Natamasindah
NIM : 2310817310002

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.....	7
B. Output Program	15
C. Pembahasan	16
D. Tautan Git.....	21

DAFTAR GAMBAR

Tabel 1. Source Code Jawaban Soal 1..... **Error! Bookmark not defined.**

DAFTAR TABEL

Tabel 1. Source Code Jawaban Soal 1	Error! Bookmark not defined.
Tabel 2. Source Code Jawaban Soal 1	10

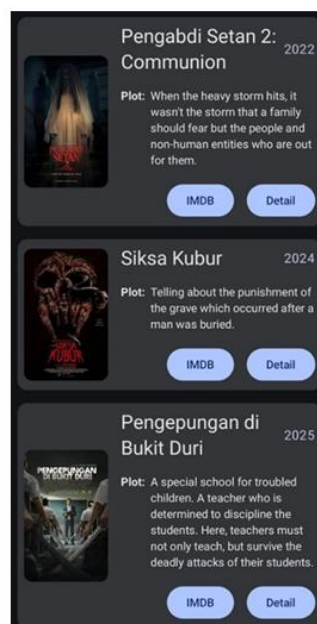
SOAL 1

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

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. Contoh UI List

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. Contoh UI Detail

A. Source Code

1. MainActivity.kt

1	package com.android.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.foundation.Image
10	import androidx.compose.foundation.background
11	import androidx.compose.foundation.layout.Arrangement
12	import androidx.compose.foundation.layout.Column
13	import androidx.compose.foundation.layout.Row
14	import androidx.compose.foundation.layout.Spacer
15	import androidx.compose.foundation.layout.fillMaxSize
16	import androidx.compose.foundation.layout.fillMaxWidth
17	import androidx.compose.foundation.layout.height
18	import androidx.compose.foundation.layout.padding
19	import androidx.compose.foundation.layout.size
20	import androidx.compose.foundation.layout.statusBarsPadding
21	import androidx.compose.foundation.layout.width
22	import androidx.compose.foundation.lazy.LazyColumn
23	import androidx.compose.foundation.rememberScrollState
24	import androidx.compose.foundation.shape.RoundedCornerShape

```

25 import androidx.compose.foundation.verticalScroll
26 import androidx.compose.material3.Button
27 import androidx.compose.material3.Card
28 import androidx.compose.material3.CardDefaults
29 import androidx.compose.material3.ExperimentalMaterial3Api
30 import androidx.compose.material3.Scaffold
31 import androidx.compose.material3.Text
32 import androidx.compose.material3.TopAppBar
33 import androidx.compose.material3.TopAppBarDefaults
34 import androidx.compose.runtime.Composable
35 import androidx.compose.ui.Alignment
36 import androidx.compose.ui.Modifier
37 import androidx.compose.ui.draw.clip
38 import androidx.compose.ui.graphics.Color
39 import androidx.compose.ui.graphics.Shape
40 import androidx.compose.ui.platform.LocalContext
41 import androidx.compose.ui.res.colorResource
42 import androidx.compose.ui.res.painterResource
43 import androidx.compose.ui.text.font.FontWeight
44 import androidx.compose.ui.text.style.TextAlign
45 import androidx.compose.ui.unit.dp
46 import androidx.compose.ui.unit.sp
47 import
    androidx.core.splashscreen.SplashScreen.Companion.installSplashScreen
48 import androidx.navigation.NavController
49 import androidx.navigation.NavHostController
50 import androidx.navigation.compose.NavHost
51 import androidx.navigation.compose.composable
52 import androidx.navigation.compose.rememberNavController
53 import com.android.modul3.ui.theme.Modul3Theme
54 import
    com.bumptech.glide.integration.compose.ExperimentalGlideComposeApi
55 import com.bumptech.glide.integration.compose.GlideImage
56
57 class MainActivity : ComponentActivity() {
58     override fun onCreate(savedInstanceState: Bundle?) {
59         super.onCreate(savedInstanceState)
60         enableEdgeToEdge()
61         setContent {
62             Modul3Theme {
63                 val navController = rememberNavController()
64                 Display(navController)
65             }
66         }
67     }
68 }
69
70 @Composable
71 fun Display(navController: NavHostController) {
72     Scaffold() { innerPadding ->
73         NavHost(
74             navController = navController,

```



```

75         startDestination = "card_list",
76         modifier = Modifier.padding(innerPadding)
77     ) {
78         composable("card_list") {
79             CardList(navController)
80         }
81         composable("detail/{itemTitle}/{itemDesc}/{itemImageUrl}")
82     { backStackEntry ->
83         val itemTitle =
84         backStackEntry.arguments?.getString("itemTitle")!!
85         val itemDesc =
86         backStackEntry.arguments?.getString("itemDesc")!!
87         val itemImageUrl =
88         backStackEntry.arguments?.getString("itemImageUrl")!!
89         DetailPage(itemTitle, itemDesc, itemImageUrl,
90         navController)
91     }
92 }
93
94 @OptIn(ExperimentalGlideComposeApi::class)
95 @Composable
96 fun CardList( navController: NavController) {
97     val context = LocalContext.current
98     val bgcard = Color(0xFF7AE2CF)
99     val bgcolor = Color(0xFFFFFDF6)
100     Title("Maskapai-maskapai Penerbangan di Indonesia")
101     LazyColumn(
102         modifier = Modifier
103             .padding(top = 50.dp)
104             .background(bgcolor)
105     ){
106         items(cardProperties.size) {index ->
107             val property = cardProperties[index]
108             Card(
109                 modifier = Modifier
110                     .fillMaxWidth()
111                     .padding(6.dp)
112                     .height(200.dp),
113                 colors = CardDefaults.cardColors(containerColor =
114                 bgcard),
115             ) {
116                 Row(
117                     modifier = Modifier
118                         .padding(6.dp),
119                 ) {
120                     Img(property.ImageURL, 180)
121                     Column {
122                         Title(property.title)
123                         Desc(if(property.desc.length > 80)
124                         property.desc.take(80) + "..." else property.desc)
125                     }
126                 }
127             }
128         }
129     }
130 }

```

```

120         Spacer(modifier = Modifier.weight(1f))
121         Row(
122             horizontalArrangement = Arrangement.End,
123             modifier = Modifier
124                 .fillMaxWidth()
125         ) {
126             Button(onClick = {
127                 val intent =
128                 Intent(Intent.ACTION_VIEW, Uri.parse(property.Wiki))
129                 context.startActivity(intent)
130                 }, modifier = Modifier.padding(horizontal
131                 = 6.dp),
132                 shape = RoundedCornerShape(8.dp)
133             ) {Text("Wiki") }
134             Button(onClick = {
135                 navController.navigate("detail/${property.title}/${property.desc}/
136                 ${Uri.encode(property.ImageURL)}")
137                 }, shape = RoundedCornerShape(8.dp))
138             {Text("Detail") }
139         }
140     }
141 }
142
143 @OptIn(ExperimentalGlideComposeApi::class)
144 @Composable
145 fun DetailPage(itemTitle: String, itemDesc:String,
146     itemImageURL:String, navController: NavController) {
147     val context = LocalContext.current
148     val namaMaskapai = detailProperties.find{ it.title == itemTitle}
149     Column(modifier = Modifier
150         .padding(16.dp)
151         .verticalScroll(rememberScrollState()),
152         horizontalAlignment = Alignment.CenterHorizontally
153     ) {
154         Img(itemImageURL, 400)
155         Title(itemTitle)
156         Spacer(modifier = Modifier.height(16.dp))
157         namaMaskapai?.let {
158             DetailRow("Tanggal Berdiri", ": ${it.tglBerdiri}")
159             DetailRow("Armada", ": ${it.armada}")
160             DetailRow("Rute Tujuan", ": ${it.rute} ")
161         }
162         Spacer(modifier = Modifier.height(16.dp))
163         Desc(itemDesc)
164         Spacer(modifier = Modifier.height(16.dp))
165         Row(modifier =

```

166	Modifier.fillMaxWidth(),horizontalArrangement =
167	Arrangement.SpaceBetween) {
168	Button(onClick = {
169	val intent = Intent(Intent.ACTION_VIEW,
170	Uri.parse(namaMaskapai?.let {it.website}))
171	context.startActivity(intent)
172	}, modifier = Modifier.padding(horizontal = 6.dp),
173	shape = RoundedCornerShape(8.dp)
174) {Text("Web \$itemTitle") }
175	Button(onClick =
	{navController.navigate("card_list")}, shape =
	RoundedCornerShape(8.dp)) {Text("Kembali") }
	}
	}
	}

Tabel 1. Source Code Jawaban Soal 1 MainActivity

2. AirlineData.kt

1	package com.android.modul3
2	
3	data class cardProperty(
4	val title: String,
5	val desc: String,
6	val ImageURL: String,
7	val Wiki: String
8)
9	
10	val cardProperties = listOf(
11	cardProperty(
12	title = "Garuda Indonesia",
13	desc = "Garuda Indonesia adalah maskapai penerbangan
	nasional Indonesia yang didirikan pada tahun 1949. Berbasis di
	Jakarta, Garuda dikenal dengan layanan penerbangannya yang
	berkualitas tinggi dan keramahannya, mencerminkan budaya
	Indonesia. Maskapai ini mengoperasikan penerbangan domestik
	dan internasional ke berbagai tujuan di Asia, Australia,
	Eropa, dan Timur Tengah. Garuda Indonesia juga merupakan
	anggota dari aliansi penerbangan global SkyTeam dan telah
	beberapa kali meraih penghargaan dunia atas pelayanan
	kabinnya.",
14	ImageURL = "https://cdn.plnspttrs.net/11964/pk-gib-
	garuda-indonesia-mcdonnell-douglas-dc-10-
	30_PlanespottersNet_714188_a594861cb0_o.jpg",
15	Wiki =
	"https://id.wikipedia.org/wiki/Garuda_Indonesia"
16),
17	cardProperty(
18	title = "Lion Air",
19	desc = "Lion Air adalah maskapai penerbangan swasta
	terbesar di Indonesia yang berdiri pada tahun 1999. Fokus
	utamanya adalah layanan penerbangan berbiaya rendah (low-cost

	carrier) dengan rute domestik dan internasional. Lion Air terkenal dengan jaringan penerbangan yang luas dan harga tiket yang kompetitif.",
20	ImageURL = "https://cdn.plnspttrs.net/20446/pk-lhg-lion-air-boeing-747-412_PlanespottersNet_612707_2702071c86_o.jpg",
21	Wiki = "https://id.wikipedia.org/wiki/Lion_Air"
22),
23	cardProperty(
24	title = "Citilink",
25	desc = "Citilink adalah anak perusahaan dari Garuda Indonesia yang beroperasi sebagai maskapai berbiaya rendah. Didirikan pada tahun 2001, Citilink menawarkan penerbangan domestik dan regional dengan konsep layanan yang modern, dinamis, dan lebih santai untuk menarik kalangan muda dan pelancong bisnis.",
26	ImageURL = "https://cdn.plnspttrs.net/42179/pk-gaf-citilink-atr-72-600-72-212a_PlanespottersNet_1657140_83de625cfa_o.jpg",
27	Wiki = "https://id.wikipedia.org/wiki/Citilink"
28),
29	cardProperty(
30	title = "Super Air Jet",
31	desc = "Super Air Jet adalah maskapai baru di Indonesia yang mulai beroperasi pada tahun 2021. Mengusung konsep \"new lifestyle airline\", Super Air Jet fokus melayani segmen anak muda dengan harga terjangkau, desain modern, dan rute-rute domestik populer.",
32	ImageURL = "https://cdn.plnspttrs.net/14570/pk-std-super-air-jet-airbus-a320-232-wl_PlanespottersNet_1755960_137445c980_o.jpg",
33	Wiki = "https://www.superairjet.com/en/about.php"
34),
35	cardProperty(
36	title = "Batik Air",
37	desc = "Batik Air adalah maskapai layanan penuh (full-service) dari Lion Air Group yang didirikan pada tahun 2013. Batik Air menawarkan fasilitas premium seperti makanan dalam penerbangan dan hiburan di kursi, serta menghubungkan berbagai kota besar di Indonesia dan Asia.",
38	ImageURL = "https://cdn.plnspttrs.net/35032/pk-lug-batik-air-airbus-a320-214-wl_PlanespottersNet_1693958_5bbffba586_o.jpg",
39	Wiki = "https://id.wikipedia.org/wiki/Batik_Air"
40),
41	cardProperty(
42	title = "Pelita Air",
43	desc = "Pelita Air adalah maskapai yang awalnya fokus pada penerbangan carter dan layanan energi (minyak dan gas), namun sejak 2022 mulai mengembangkan penerbangan reguler domestik. Sebagai anak usaha Pertamina, Pelita Air membawa konsep layanan penerbangan yang nyaman dan profesional.",

```

44         ImageURL = "https://cdn.plnspttrs.net/22490/pk-pwd-
pelita-air-service-airbus-a320-214-
wl_PlanespottersNet_1698319_901e5c6a31_o.jpg",
45         Wiki = "https://id.wikipedia.org/wiki/Pelita_Air"
46     )
47 )
48
49 data class detailProperty(
50     val title: String,
51     val tglBerdiri: String,
52     val armada: String,
53     val rute: String,
54     val website:String
55 )
56
57 val detailProperties = listOf(
58     detailProperty(
59         title = "Garuda Indonesia",
60         tglBerdiri = "1 Agustus 1947",
61         armada = "77",
62         rute = "Belanda, Thailand, China, Hong Kong, Arab
Saudi, Malaysia, Arab Saudi, Australia, Korea Selatan,
Singapura, Jepang, United Arab Emirates",
63         website = "https://www.garuda-indonesia.com/id/id/",
64     ),
65     detailProperty(
66         title = "Lion Air",
67         tglBerdiri = "15 November 1999",
68         armada = "100",
69         rute = "Arab Saudi, China, Malaysia",
70         website = "https://www.lionair.co.id/",
71     ),
72     detailProperty(
73         title = "Citilink",
74         tglBerdiri = "16 Juli 2001",
75         armada = "57",
76         rute = "Australia, Malaysia, Papua Nugini, Singapura,
Timor Leste, China",
77         website = "https://www.citilink.co.id/",
78     ),
79     detailProperty(
80         title = "Super Air Jet",
81         tglBerdiri = "1 March 2021",
82         armada = "61",
83         rute = "Malaysia",
84         website = "https://checkin.superairjet.com/dx/IUCI/",
85     ),
86     detailProperty(
87         title = "Batik Air",
88         tglBerdiri = "10 Juni 2012",
89         armada = "65",
90         rute = "Australia, Arab Saudi, China, India, Malaysia,

```

91	Singapura, Thailand",
92	website = "https://www.batikair.com.my/",
93),
94	detailProperty(
95	title = "Pelita Air",
96	tglBerdiri = "24 Januari 1970",
97	armada = "33",
98	rute = "Indonesia",
99	website = "https://www.pelita-air.com/",
100)

Tabel 2 Source Code Jawaban Soal 1 AirlineData

3. UiComponent.kt

1	package com.android.modul3
2	
3	import android.content.Intent
4	import android.net.Uri
5	import androidx.compose.foundation.layout.Row
6	import androidx.compose.foundation.layout.fillMaxWidth
7	import androidx.compose.foundation.layout.padding
8	import androidx.compose.foundation.layout.width
9	import androidx.compose.foundation.shape.RoundedCornerShape
10	import androidx.compose.material3.Button
11	import androidx.compose.material3.Text
12	import androidx.compose.runtime.Composable
13	import androidx.compose.ui.Modifier
14	import androidx.compose.ui.draw.clip
15	import androidx.compose.ui.text.font.FontWeight
16	import androidx.compose.ui.text.style.TextAlign
17	import androidx.compose.ui.unit.dp
18	import androidx.compose.ui.unit.sp
19	import
	com.bumptech.glide.integration.compose.ExperimentalGlideComposeApi
20	import com.bumptech.glide.integration.compose.GlideImage
21	
22	@Composable
23	fun Desc(desc: String) {
24	Text(
25	text = desc,
26	fontSize = 14.sp,
27	textAlign = TextAlign.Justify
28)
29	}
30	@OptIn(ExperimentalGlideComposeApi::class)
31	@Composable
32	fun Img(url: String, width: Int) {
33	GlideImage(

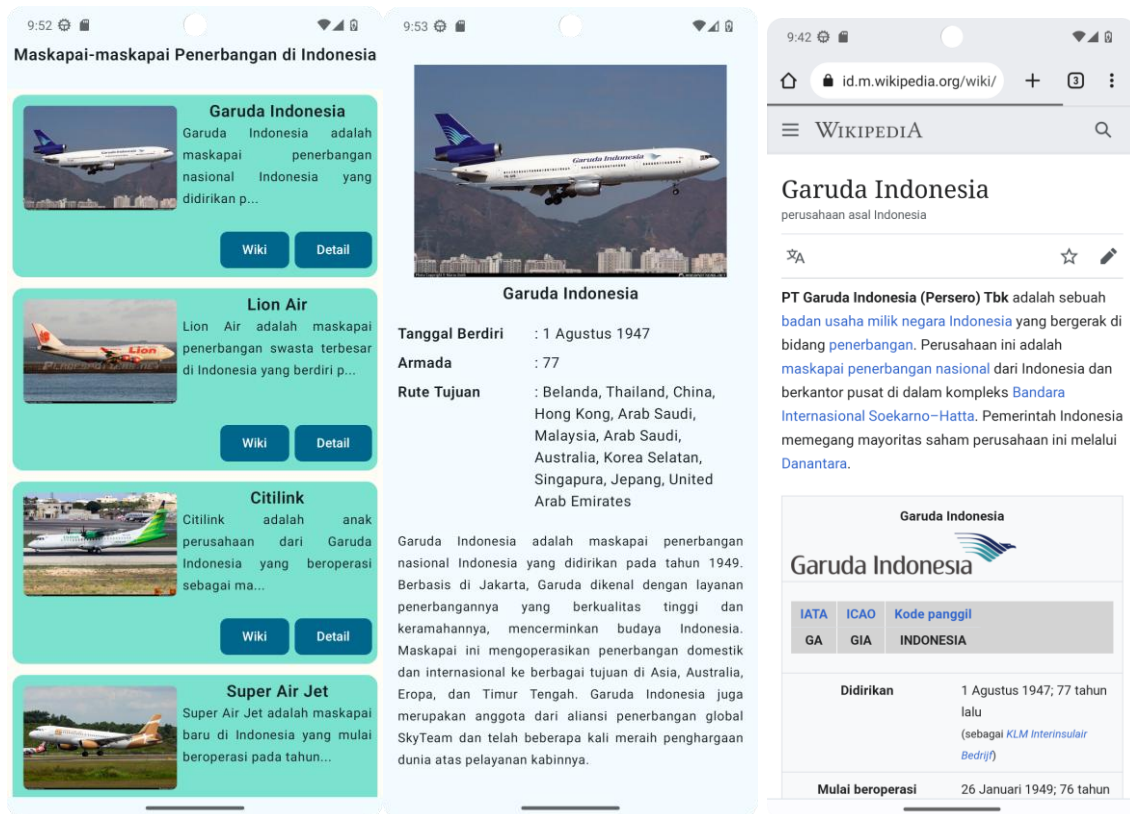
```

34         model = url,
35         contentDescription = "My Image",
36         modifier = Modifier
37             .width(width.dp)
38             .padding(6.dp)
39             .clip(RoundedCornerShape(6.dp)),
40     )
41 }
42 @Composable
43 fun Title(title: String) {
44     Text(
45         text = title,
46         fontSize = 18.sp,
47         textAlign = TextAlign.Center,
48         fontWeight = FontWeight.SemiBold,
49         modifier = Modifier.fillMaxWidth()
50     )
51 }
52
53 @Composable
54 fun DetailRow(label:String, value:String) {
55     Row (
56         modifier = Modifier
57             .fillMaxWidth()
58             .padding(vertical = 4.dp)
59     ) {
60         Text(
61             text = label,
62             modifier = Modifier.width(150.dp),
63             fontWeight = FontWeight.SemiBold
64         )
65         Text(value)
66     }
67 }

```

Tabel 3 Source Code Jawaban Soal 1 UIComponent

B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity.kt:

- Pada baris 1, **package com.android.modul3** pendeklarasian nama package file Kotlin.
- Pada baris 3-55, **import** berfungsi untuk mengimpor package atau kelas untuk mengakses komponen dan fungsi dari android SDK dan jetpack compose.
- Pada baris 57, **class MainActivity : ComponentActivity()**, merupakan titik mula yang menjadi kelas utama dan akan dijalankan pertama kali saat aplikasi dibuka
- Pada baris 58, **override fun onCreate(savedInstanceState: Bundle?)**, berfungsi untuk menimpa (override) fungsi onCreate dari ComponentActivity.
- Pada baris 59, **super.onCreate(savedInstanceState)**, berfungsi untuk memanggil superclass dari fungsi onCreate untuk memastikan bahwa proses inisialisasi standar dari Android dijalankan sebelum logika saya dijalankan.
- Pada baris 60, **enableEdgeToEdge()**, berfungsi agar tampilan aplikasi dapat menggunakan seluruh layar dari status bar sampai navigation bar atau fullscreen layout.

- Pada baris 61, **setContent()**, digunakan untuk menampilkan UI berbasis jetpack compose ke dalam activity
- Pada baris 62, **Modul3Theme**, merupakan fungsi yang berisi tema custom yang membungkus seluruh UI untuk memberikan style yang konsisten.
- Pada baris 63-147, **val** merupakan keyword untuk membuat suatu variable bersifat immutable yang menampung nilai dari berbagai tipe data.
- Pada baris 63, **val NavController = rememberNavController()** berfungsi untuk membuat dan menyimpan instance dari NavController
- Pada baris 64, **Display(navController)**, merupakan fungsi yang dibuat untuk menampilkan aplikasi scrollable list ke dalam activity dan menerima parameter NavController.
- Pada baris 70-144, **@composable**, merupakan annotation untuk menandai bahwa fungsi yang dibuat dibawahnya merupakan fungsi UI di jetpack compose.
- Pada baris 71, **fun Display(navController: NavHostController)**, merupakan fungsi yang dibuat untuk menampilkan semua composable yang ingin ditampilkan ke dalam activity, seperti CardList() dan DetailPage() lalu menerima parameter NavController.
- Pada baris 72, **scaffold()**, merupakan komponen layout yang menjadi kerangka dasar dari tampilan yang mempermudah dalam mengatur strktur UI karena sudah disediakan slot-slot bawannya seperti topBar yang sudah saya isi dengan TopBar() dan content saya isi dengan RollDice().
- Pada baris 73, **NavHost()**, berfungsi untuk menyediakan wadah untuk navigasi antar composable screen berdasarkan rute yang ditentukan
 - Pada baris 74, **NavController** merupakan controller yang bertanggung jawab atas navigasi.
 - Pada baris 75, **startDestination** merupakan rute awal yang ditampilkan ketika aplikasi dijalankan.
 - Pada baris 76, modifier = **Modifier** berfungsi untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.padding* untuk memberikan jarak dalam component.
- Pada baris 73 dan 81, **composable()**, berfungsi untuk mendefinisikan rute, seperti di kasus ini ada “card_list” dan “detail”
- Pada baris 73, **CardList(navController)**, merupakan fungsi untuk menampilkan list yang dapat scroll sekaligus menjadi halaman utama aplikasi.

- pada baris 82-84, **itemTitle**, **itemDesc**, **itemImageURL**
- pada baris 85, **DetailPage(itemTitle, itemDesc, itemImageURL, navController)** merupakan fungsi untuk menampilkan halaman detail dari aplikasi dan menerima beberapa parameter.
- Pada baris 91 dan 153, **@OptIn(ExperimentalGlideComposeApi::class)**, berfungsi untuk memberitahukan compiler bahwa fungsi composable dibawahnya masih menggunakan API yang bersifat eksperimental untuk menghindari error saat kompilasi.
- Pada baris 68, **fun CardList(navController: NavController)**, merupakan fungsi untuk menampilkan card-card dalam jumlah banyak yang dapat di scroll.
- Pada baris 94, **LocalContext.current** berfungsi untuk mendapatkan context dari activity saat ini dalam fungsi composable dan juga digunakan untuk menampilkan Toast sebagai notifikasi dari peringatan jika user salah input.
- Pada baris 97-154, **Title()** merupakan fungsi yang saya buat sendiri untuk memudahkan dalam membuat judul.
- Pada baris 98, **LazyColumn()** berfungsi untuk menampilkan daftar item secara vertical yang dapat di scroll tapi hanya yang terlihat di layar.
 - Pada baris 99, **modifier** merupakan parameter dalam Column() untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.padding* untuk memberikan jarak dalam komponen dan *.background* untuk memberikan warna latar.
- Pada baris 103, **item()** berfungsi untuk membuat item di dalam LazyColumn sebanyak jumlah data.
- pada baris 105, **Card()** berfungsi untuk membuat sebuah kartu.
 - pada baris 106, **modifier** merupakan parameter dalam Column() untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.padding* untuk memberikan jarak dalam component, *.height* untuk tinggi component, dan *.fillMaxWidth()* untuk membuat lebar component lebarnya selayar.
 - pada baris 110, **colors** merupakan parameter dalam Card() untuk mengatur warna background (containerColor).
- Pada baris 112-165, **Row** merupakan komponen layout yang berlawanan dari Column, Dimana jika Column menyusun elemen UI secara Vertical, maka Row menyusun elemen UI secara horizontal.

- Pada baris 112-165, **modifier** berfungsi untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.padding* untuk memberikan jarak dalam component, *.fillMaxWidth()* untuk membuat lebar component lebarnya layar.
 - Pada baris 122 dan 165, **horizontalArrangement** berfungsi untuk memberi jarak maksimal antar elemen.
- Pada baris 116 dan 153, **Img()** merupakan fungsi yang saya buat sendiri untuk memudahkan dalam memuat gambar.
- Pada baris 117 dan 148, **Column()** merupakan komponen layout mirip seperti scaffold, hanya saja Column digunakan untuk menyusun elemen UI secara vertical.
 - Pada baris 148, **modifier** berfungsi untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.padding* untuk memberikan jarak dalam dan *verticalScroll* untuk membuat konten dapat di scroll secara vertical.
 - Pada baris 151, **horizontalAlignment** berfungsi untuk mengatur posisi elemen secara horizontal
- Pada baris 119 dan 162, **Desc()** merupakan fungsi yang saya buat sendiri untuk memudahkan dalam membuat deskripsi.
- Pada baris 120-163, **Spacer()** berfungsi untuk memberikan jarak kosong biasanya secara vertical dalam layout.
- Pada baris 126-172, **Button()** merupakan komponen UI yang berfungsi untuk membuat tombol pada aplikasi.
 - Pada baris 135, **onClick** merupakan parameter dalam Button() untuk memberikan aksi yang akan dijalankan ketika tombol ditekan.
 - Pada baris 146, **modifier** merupakan parameter dalam TextField untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.fillMaxWidth* agar column lebar Button mengisi ukuran layar.
 - Pada baris 148, **colors** merupakan parameter dalam Button() untuk mengatur warna background (containerColor) dan warna teks (titleContentColor).
 - Pada baris 152, **shape** merupakan parameter dalam Button() untuk mengatur sudut agar dapat membulat.

2. AirlineData.kt

- Pada baris 1, **package com.android.modul3** pendeklarasian nama package file Kotlin.
- Pada baris 3 dan 49, **data class** merupakan jenis kelas untuk menyimpan struktur data.

- Pada baris 4-57, **val** merupakan keyword untuk membuat suatu variable bersifat immutable yang menampung nilai dari berbagai tipe data.
- Pada baris 10 dan 57 **listOf** digunakan untuk membuat list dari element dengan sifat immutable.

3. **UIComponent.kt**

- Pada baris 1, **package com.android.modul3** pendeklarasian nama package file Kotlin.
- Pada baris 3-20, **import** berfungsi untuk mengimpor package atau kelas untuk mengakses komponen dan fungsi dari android SDK dan jetpack compose.
- Pada baris 22-53, **@Composable** merupakan annotation untuk menandai bahwa fungsi yang dibuat dibawahnya merupakan fungsi UI di jetpack compose.
- Pada baris 23, **fun Desc(desc: String)** merupakan fungsi yang berisikan Text dengan style tertentu untuk deskripsi.
- pada baris 24-65, **Text()** merupakan komponen UI yang berfungsi menampilkan teks.
 - pada baris 2-61, **text** merupakan isi dari teks yang akan ditampilkan
 - pada baris 26 dan 46, **fontSize** merupakan ukuran dari font yang akan ditampilkan
 - pada baris 27 dan 47, **textAlign** untuk membuat apakah teks berada di kiri, Tengah , atau kanan
 - pada baris 48 dan 63, **fontWeight** merupakan ketebalan dari teks yang akan ditampilkan
 - pada baris 49 dan 62, **modifier** berfungsi untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.fillMaxWidth* agar column lebar Button mengisi ukuran layar.
- Pada baris 30, **@OptIn(ExperimentalGlideComposeApi::class)** berfungsi untuk memberitahukan compiler bahwa fungsi composable dibawahnya masih menggunakan API yang bersifat eksperimental untuk menghindari error saat kompilasi.
- Pada baris 32, **fun Img(url: String, width: Int)** merupakan fungsi yang berisikan gambar dengan style yang sudah ditentukan.
- Pada baris 33, **GlideImage** merupakan library glide yang digunakan untuk menampilkan gambar dari internet melalui URL
 - Pada baris 34, **model** merupakan tempat Dimana URL diletakkan

- Pada baris 35, **contentDescription** merupakan deskripsi dari gambar yang akan ditampilkan
- Pada baris 36, **modifier** merupakan parameter dalam TextField untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.width* untuk lebar, *.height* untuk tinggi, dan *.clip* untuk mengatur lengkungan gambar.
- Pada baris 43, **fun Title(title: String)** merupakan fungsi yang bersikan Text dengan style yang telah ditentukan untuk judul.
- Pada baris 33, **fun DetailRow(label:String, value:String)** merupakan fungsi untuk berisikan sepasang Text untuk halaman detail.
- Pada baris 55, **Row** merupakan komponen layout yang berlawanan dari Column, Dimana jika Column menyusun elemen UI secara Vertical, maka Row menyusun elemen UI secara horizontal.
 - Pada baris 94-124, **modifier** merupakan parameter dalam TextField untuk menerapkan modifikasi terhadap tampilan komponen, seperti *.fillMaxWidth* agar column lebar Row mengisi ukuran layar.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/AppleCraft2005/kuliah-santuy/tree/main/semesterIV/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?

- Karena saat ini masih banyak project-project android yang masih belum melakukan migrasi ke jetpack compose, sehingga masih berbasis XML nah oleh sebab itu RecyclerView masih banyak digunakan. Selain itu juga, di RecyclerView lebih memberikan control kepada pengguna untuk melakukan layoutManager Kustom, Animation transition, Item Decoration, Dimana hal ini belum bisa dilakukan oleh LazyColumn, dan dalam aplikasi yang sudah berskala besar, RecyclerView sering dioptimalkan hingga Tingkat yang sangat rendah yang Dimana hal ini belum bisa dilakukan di LazyColumn.