

LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3



Build a Scrollable List

Oleh:

Muhammad Rizki Saputra NIM. 2310817310014

PROGRAM STUDI TEKNOLOGI INFORMASI FAKULTAS
TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2024

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE 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 : Muhammad Rizki Saputra
NIM : 2310817310014

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 GAMBAR	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	7
B. Output Program	21
C. Pembahasan	23
D. Tautan Git	26

DAFTAR GAMBAR

Gambar 1 Screenshot Hasil Jawaban Soal 1	21
Gambar 2 Screenshot Hasil Jawaban Soal 1	22

DAFTAR TABEL

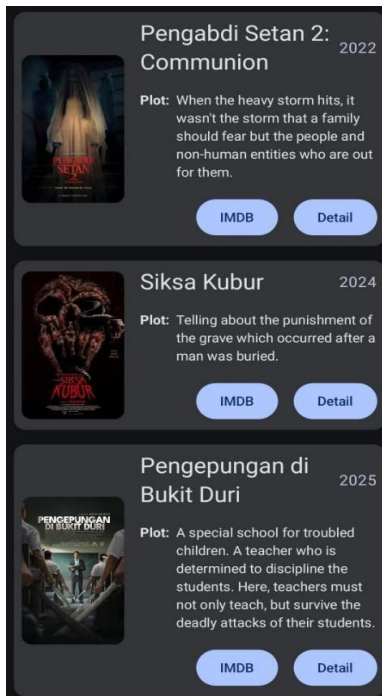
Table 1 Source Code DinosaurRepositoryImpl.kt.....	7
Table 2 Source Code LocalDinosaurDataSource.kt	9
Table 3Source Code Dinosaur.kt	10
Table 4 Source Code DinosaurRepository.kt.....	10
Table 5 Source Code GetDinosaurUseCase.kt.....	10
Table 6 Source Code AppNavigation.kt	11
Table 7 Source Code DinoDetailScreen.kt	13
Table 8 Source Code DinoListScreen.kt.....	18
Table 9 Source Code DinoListViewModel.kt.....	19
Table 10 Source Code MainActivity.kt	20

SOAL 1

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
Item pada list menampilkan teks dan gambar sesuai dengan contoh di bawah 4.
Terdapat 2 button dalam list, dengan fungsi berikut:
 - Button pertama menggunakan intent eksplisit untuk membuka aplikasi atau browser lain
 - 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:



A. Source Code

- DinosaurRepositoryImpl.kt

1	package com.example.modul3.data.repository
2	
3	import
4	com.example.modul3.data.datasource.LocalDinosaurDataSource
5	import com.example.modul3.domain.model.Dinosaur
6	import com.example.modul3.domain.repository.DinosaurRepository
7	
8	
9	class DinosaurRepositoryImpl : DinosaurRepository {
10	override fun getDinosaurs(): List<Dinosaur> {
11	return LocalDinosaurDataSource.getDinosaurs()
12	}
13	}

Table 1 Source Code DinosaurRepositoryImpl.kt

- LocalDinosaurDataSource

1	package com.example.modul3.data.datasource
2	
3	import com.example.modul3.R
4	import com.example.modul3.domain.model.Dinosaur
5	
6	object LocalDinosaurDataSource {
7	fun getDinosaurs(): List<Dinosaur> {
8	return listOf(
9	Dinosaur(
10	"Tyrannosaurus Rex",
11	""
12	Tyrannosaurus Rex adalah predator terbesar di
13	zamannya...
14	""
15	"Predator terbesar di zamannya dengan rahang
16	yang sangat kuat.",
17	"Akhir Periode Kapur",
18	R.drawable.tyrannosaurusrex,
19	"Karnivora",
20	"https://en.wikipedia.org/wiki/Tyrannosaurus"
21),
22	Dinosaur(
23	"Triceratops",
24	""
25	Triceratops adalah dinosaurus herbivora
26	besar...
27	""
28	"Herbivora besar dengan tiga tanduk ikonik di
29	wajahnya.",
30	"Akhir Periode Kapur",

```

31         R.drawable.triceratops,
32         "Herbivora",
33         "https://en.wikipedia.org/wiki/Triceratops"
34     ),
35     // ... Lanjutkan untuk semua dinosaurus lainnya
36     seperti kode di jawaban sebelumnya ...
37     // Pastikan semua 10 dinosaurus memiliki data
38     shortDesc dan period.
39     // (Saya persingkat agar tidak terlalu panjang,
40     gunakan kode lengkap dari jawaban sebelumnya)
41     Dinosaur(
42         "Velociraptor",
43         ""Velociraptor adalah dinosaurus
44 kecil..."".trimIndent(),
45         "Karnivora kecil yang terkenal karena kecepatan
46 dan kecerdasannya.",
47         "75-71 Juta Tahun Lalu",
48         R.drawable.velociraptor, "Karnivora",
49         "https://en.wikipedia.org/wiki/Velociraptor"
50     ),
51     Dinosaur(
52         "Stegosaurus",
53         ""Stegosaurus adalah dinosaurus
54 herbivora..."".trimIndent(),
55         "Herbivora dengan piring tulang besar di
56 punggung dan ekor berduri.",
57         "Periode Jurassic Akhir",
58         R.drawable.stegosaurus, "Herbivora",
59         "https://en.wikipedia.org/wiki/Stegosaurus"
60     ),
61     Dinosaur(
62         "Brachiosaurus",
63         ""Brachiosaurus adalah salah satu sauropoda
64 terbesar..."".trimIndent(),
65         "Sauropoda raksasa dengan leher sangat panjang
66 dan kaki depan yang tinggi.",
67         "Periode Jurassic",
68         R.drawable.brachiosaurus, "Herbivora",
69         "https://en.wikipedia.org/wiki/Brachiosaurus"
70     ),
71     Dinosaur(
72         "Spinosaurus",
73         ""Spinosaurus adalah dinosaurus karnivora
74 terbesar..."".trimIndent(),
75         "Karnivora semi-akuatik dengan layar punggung
76 yang khas.",
77         "112-93 Juta Tahun Lalu",
78         R.drawable.spinosaurus, "Karnivora",
79         "https://en.wikipedia.org/wiki/Spinosaurus"
80     ),
81     Dinosaur(
        "Ankylosaurus",

```


82	"""Ankylosaurus adalah dinosaurus
83	herbivora...""".trimIndent(),
84	"Herbivora berlapis zirah dengan ekor besar
85	berbentuk gada.",
86	"Akhir Periode Kapur",
87	R.drawable.ankylosaurus, "Herbivora",
88	"https://en.wikipedia.org/wiki/Ankylosaurus"
89),
90	Dinosaur(
91	"Allosaurus",
92	"""Allosaurus adalah predator
93	utama...""".trimIndent(),
94	"Predator puncak dari periode Jurassic, sering
95	disebut 'singa Jurassic'.",
96	"Periode Jurassic Akhir",
97	R.drawable.allosaurus, "Karnivora",
98	"https://en.wikipedia.org/wiki/Allosaurus"
99),
100	Dinosaur(
101	"Diplodocus",
102	"""Diplodocus adalah sauropoda
103	raksasa...""".trimIndent(),
104	"Salah satu dinosaurus terpanjang dengan leher
105	dan ekor super panjang.",
106	"Periode Jurassic",
107	R.drawable.diplodocus, "Herbivora",
108	"https://en.wikipedia.org/wiki/Diplodocus"
109),
110	Dinosaur(
111	"Parasaurolophus",
112	"""Parasaurolophus adalah dinosaurus
113	herbivora...""".trimIndent(),
114	"Herbivora dengan jambul panjang berongga yang
115	khas di kepalanya.",
116	"Akhir Periode Kapur",
117	R.drawable.parasaurolophus, "Herbivora",
118	"https://en.wikipedia.org/wiki/Parasaurolophus"
119)
120)
121	}
122	}

Table 2 Source Code LocalDinosaurDataSource.kt

- Dinosaur

1	package com.example.modul3.domain.model
2	
3	data class Dinosaur(
4	val name: String,
5	val description: String,
6	val shortDesc: String,
7	val period: String,
8	val imageRes: Int,
9	val type: String,
10	val wikiUrl: String
11)

Table 3Source Code Dinosaur.kt

- DinosaurRepository

1	package com.example.modul3.domain.repository
2	
3	import com.example.modul3.domain.model.Dinosaur
4	
5	interface DinosaurRepository {
6	fun getDinosaurs(): List<Dinosaur>
7	}

Table 4 Source Code DinosaurRepository.kt

- GetDinosaurUseCase

1	package com.example.modul3.domain.usecase
2	
3	import com.example.modul3.domain.model.Dinosaur
4	import com.example.modul3.domain.repository.DinosaurRepository
5	
6	class GetDinosaursUseCase(private val repository:
7	DinosaurRepository) {
8	operator fun invoke(): List<Dinosaur> {
9	return repository.getDinosaurs()
10	}
11	}

Table 5 Source Code GetDinosaurUseCase.kt

- AppNavigation.kt

```
1 package com.example.modul3.presentation.navigation
2
3 import androidx.compose.runtime.Composable
4 import androidx.compose.ui.Modifier
5 import androidx.navigation.NavHostController
6 import androidx.navigation.compose.NavHost
7 import androidx.navigation.compose.composable
8 import
9 com.example.modul3.presentation.dino_detail.DinoDetailScreen
10 import com.example.modul3.presentation.dino_list.DinoListScreen
11 import java.net.URLDecoder
12 import java.nio.charset.StandardCharsets
13
14 @Composable
15 fun AppNavigation(navController: NavHostController, modifier:
16 Modifier = Modifier) {
17     NavHost(
18         navController = navController,
19         startDestination = "dino_list",
20         modifier = modifier
21     ) {
22         composable("dino_list") {
23             DinoListScreen(navController)
24         }
25         composable("detail/{name}/{desc}/{imgRes}") { backStack
26 ->
27             val name = backStack.arguments?.getString("name")
28             ?: ""
29             val encodedDesc =
30 backStack.arguments?.getString("desc") ?: ""
31             val desc = URLDecoder.decode(encodedDesc,
32 StandardCharsets.UTF_8.toString())
33             val imgRes =
34 backStack.arguments?.getString("imgRes")?.toIntOrNull()
35
36             if (imgRes != null) {
37                 DinoDetailScreen(name, desc, imgRes,
38 navController)
39             } else {
40
41             }
42         }
43     }
44 }
```

Table 6 Source Code AppNavigation.kt

- DinoDetailScreen.kt

```
1 package com.example.modul3.presentation.dino_detail
2
3 import androidx.compose.foundation.Image
4 import androidx.compose.foundation.layout.*
5 import androidx.compose.foundation.rememberScrollState
6 import androidx.compose.foundation.shape.RoundedCornerShape
7 import androidx.compose.foundation.verticalScroll
8 import androidx.compose.material3.Button
9 import androidx.compose.material3.Text
10 import androidx.compose.runtime.Composable
11 import androidx.compose.ui.Alignment
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.text.font.FontWeight
17 import androidx.compose.ui.text.style.TextAlign
18 import androidx.compose.ui.unit.dp
19 import androidx.compose.ui.unit.sp
20 import androidx.navigation.NavController
21
22 @Composable
23 fun DinoDetailScreen(
24     name: String,
25     desc: String,
26     imgRes: Int,
27     navController: NavController
28 ) {
29     Column(
30         modifier = Modifier
31             .padding(16.dp)
32             .verticalScroll(rememberScrollState())
33             .fillMaxSize()
34     ) {
35         Image(
36             painter = painterResource(id = imgRes),
37             contentDescription = null,
38             modifier = Modifier
39                 .fillMaxWidth()
40                 .height(250.dp)
41                 .clip(RoundedCornerShape(20.dp)),
42             contentScale = ContentScale.Crop
43         )
44         Spacer(modifier = Modifier.height(16.dp))
45         Text(
46             name,
47             fontSize = 22.sp,
48             fontWeight = FontWeight.Bold,
```

49	modifier = Modifier.fillMaxWidth(),
50	textAlign = TextAlign.Center
51)
52	Spacer(modifier = Modifier.height(12.dp))
53	Text(desc, fontSize = 14.sp, textAlign =
54	TextAlign.Justify)
55	Spacer(modifier = Modifier.height(16.dp))
56	Button(
57	onClick = { navController.popBackStack() }, //
58	Lebih baik menggunakan popBackStack
59	modifier =
60	Modifier.align(Alignment.CenterHorizontally)
61) {
62	Text("Kembali")
63	}
64	}
65	}

Table 7 Source Code DinoDetailScreen.kt

- DinoListScreen.kt

1	package com.example.modul3.presentation.dino_list
2	
3	import android.content.Intent
4	import android.net.Uri
5	import androidx.compose.foundation.Image
6	import androidx.compose.foundation.background
7	import androidx.compose.foundation.layout.*
8	import androidx.compose.foundation.lazy.grid.GridCells
9	import androidx.compose.foundation.lazy.grid.LazyVerticalGrid
10	import androidx.compose.foundation.lazy.grid.items
11	import androidx.compose.foundation.shape.RoundedCornerShape
12	import androidx.compose.material3.Button
13	import androidx.compose.material3.ButtonDefaults
14	import androidx.compose.material3.Card
15	import androidx.compose.material3.CardDefaults
16	import androidx.compose.material3.DropdownMenu
17	import androidx.compose.material3.DropdownMenuItem
18	import androidx.compose.material3.OutlinedTextField
19	import androidx.compose.material3.Text
20	import androidx.compose.runtime.Composable
21	import androidx.compose.runtime.collectAsState
22	import androidx.compose.runtime.getValue
23	import androidx.compose.runtime.mutableStateOf
24	import androidx.compose.runtime.remember
25	import androidx.compose.runtime.setValue
26	import androidx.compose.ui.Alignment
27	import androidx.compose.ui.Modifier

```

28 import androidx.compose.ui.draw.clip
29 import androidx.compose.ui.graphics.Color
30 import androidx.compose.ui.layout.ContentScale
31 import androidx.compose.ui.platform.LocalContext
32 import androidx.compose.ui.res.painterResource
33 import androidx.compose.ui.text.font.FontWeight
34 import androidx.compose.ui.text.style.TextOverflow
35 import androidx.compose.ui.unit.dp
36 import androidx.compose.ui.unit.sp
37 import androidx.lifecycle.viewmodel.compose.viewModel
38 import androidx.navigation.NavController
39 import java.net.URLEncoder
40 import java.nio.charset.StandardCharsets
41
42 @Composable
43 fun DinoListScreen(
44     navController: NavController,
45     vm: DinoListViewModel = viewModel()
46 ) {
47     val uiState by vm.uiState.collectAsState()
48     val context = LocalContext.current
49     val types = listOf("Semua", "Herbivora", "Karnivora")
50
51     Column(
52         modifier = Modifier
53             .background(Color(0xFFF4F1DE))
54             .fillMaxSize()
55     ) {
56         Box(
57             modifier = Modifier
58                 .fillMaxWidth()
59                 .background(Color(0xFF81B29A))
60                 .padding(16.dp)
61         ) {
62             Text(
63                 text = "Jenis-Jenis Dinosaur",
64                 fontSize = 24.sp,
65                 fontWeight = FontWeight.Bold,
66                 color = Color.White,
67                 modifier =
68                 Modifier.align(Alignment.CenterStart)
69             )
70         }
71
72         OutlinedTextField(
73             value = uiState.searchQuery,
74             onValueChange = { vm.onSearchQueryChanged(it) },
75             label = { Text("Cari dinosaur...") },
76             shape = RoundedCornerShape(24.dp),
77             modifier = Modifier
78                 .fillMaxWidth()
79                 .padding(16.dp)

```

```

79         )
80
81     Row(
82         modifier = Modifier
83             .fillMaxWidth()
84             .padding(horizontal = 16.dp, vertical = 8.dp),
85         horizontalArrangement = Arrangement.SpaceBetween,
86         verticalAlignment = Alignment.CenterVertically
87     ) {
88         Text("Filter berdasarkan tipe:", fontSize = 14.sp)
89         FilterDropDownMenu(
90             options = types,
91             selectedOption = uiState.selectedFilter,
92             onOptionSelected = { vm.onFilterChanged(it) }
93         )
94     }
95
96     LazyVerticalGrid(
97         columns = GridCells.Fixed(2),
98         contentPadding = PaddingValues(16.dp),
99         verticalArrangement = Arrangement.spacedBy(16.dp),
100        horizontalArrangement =
101        Arrangement.spacedBy(16.dp),
102    ) {
103        items(uiState.filteredDinosaurs) { dino ->
104            Card(
105                modifier = Modifier.fillMaxWidth(),
106                colors =
107                CardDefaults.cardColors(containerColor = Color(0xFFFFF7F0)),
108                elevation =
109                CardDefaults.cardElevation(defaultElevation = 4.dp),
110                shape = RoundedCornerShape(16.dp)
111            ) {
112                Column {
113                    Image(
114                        painter = painterResource(id =
115                        dino.imageRes),
116                        contentDescription = dino.name,
117                        contentScale = ContentScale.Crop,
118                        modifier = Modifier
119                            .fillMaxWidth()
120                            .height(120.dp)
121                    )
122                    .clip(RoundedCornerShape(topStart = 16.dp, topEnd = 16.dp))
123                )
124                Column(modifier =
125                Modifier.padding(12.dp)) {
126                    Text(
127                        text = dino.name,
128                        fontSize = 16.sp,
129                        fontWeight = FontWeight.Bold,
130                        color = Color(0xFF3D405B),

```

```

130                                     maxLines = 1,
131                                     overflow =
132 TextOverflow.Ellipsis
133                                     )
134                                     Text(
135                                         text = dino.period,
136                                         fontSize = 12.sp,
137                                         color = Color.Gray
138                                     )
139                                     Spacer(modifier =
140 Modifier.height(8.dp))
141
142                                     Text(
143                                         text = dino.shortDesc,
144                                         fontSize = 13.sp,
145                                         color = Color.DarkGray,
146                                         maxLines = 3,
147                                         overflow =
148 TextOverflow.Ellipsis,
149                                     modifier =
150 Modifier.heightIn(min = 50.dp)
151                                     )
152                                     Spacer(modifier =
153 Modifier.height(12.dp))
154
155                                     Row(
156                                         modifier =
157 Modifier.fillMaxWidth(),
158                                         horizontalArrangement =
159 Arrangement.SpaceBetween
160                                     ) {
161                                         Button(
162                                             onClick = {
163                                                 val encodedDesc =
164 URLEncoder.encode(dino.description,
165 StandardCharsets.UTF_8.toString())
166
167 navController.navigate("detail/${dino.name}/${encodedDesc}/${dino
168 .imageRes}")
169                                             },
170                                             colors =
171 ButtonDefaults.buttonColors(containerColor =
172 Color(0xFF81B29A)),
173                                             shape =
174 RoundedCornerShape(12.dp),
175                                             modifier =
176 Modifier.weight(1f),
177                                             contentPadding =
178 PaddingValues(horizontal = 4.dp, vertical = 8.dp)
179                                         ) {
180                                             Text("Info Detail", color =
Color.White, fontSize = 12.sp)

```



```

181     }
182     Spacer(modifier =
183 Modifier.width(8.dp))
184     Button(
185         onClick = {
186             val intent =
187 Intent(Intent.ACTION_VIEW, Uri.parse(dino.wikiUrl))
188
189 context.startActivity(intent)
190         },
191         colors =
192 ButtonDefaults.buttonColors(containerColor =
193 Color(0xFFE07A5F)),
194         shape =
195 RoundedCornerShape(12.dp),
196         modifier =
197 Modifier.weight(1f),
198         contentPadding =
199 PaddingValues(horizontal = 4.dp, vertical = 8.dp)
200     ) {
201         Text("Wikipedia", color =
202 Color.White, fontSize = 12.sp)
203     }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212
213
214 @Composable
215 fun FilterDropdownMenu(
216     options: List<String>,
217     selectedOption: String,
218     onOptionSelected: (String) -> Unit
219 ) {
220     var expanded by remember { mutableStateOf(false) }
221     Box {
222         Button(
223             onClick = { expanded = true },
224             colors = ButtonDefaults.buttonColors(containerColor
225 = Color(0xFF81B29A))
226         ) {
227             Text(selectedOption, color = Color.White)
228         }
229         DropdownMenu(
230             expanded = expanded,
231             onDismissRequest = { expanded = false }
232         ) {

```

232	options.forEach { label ->
233	DropDownMenuItem(
234	text = { Text(label) },
235	onClick = {
236	onOptionSelected(label)
237	expanded = false
238	}
239)
240	}
241	}
242	}
243	}

Table 8 Source Code DinoListScreen.kt

- DinoListViewModel.kt

1	package com.example.modul3.presentation.dino_list
2	
3	import androidx.lifecycle.ViewModel
4	import
5	com.example.modul3.data.repository.DinosaurRepositoryImpl
6	import com.example.modul3.domain.model.Dinosaur
7	import com.example.modul3.domain.usecase.GetDinosaursUseCase
8	import kotlinx.coroutines.flow.MutableStateFlow
9	import kotlinx.coroutines.flow.StateFlow
10	import kotlinx.coroutines.flow.asStateFlow
11	import kotlinx.coroutines.flow.update
12	
13	
14	data class DinoListUiState(
15	val allDinosaurs: List<Dinosaur> = emptyList(),
16	val filteredDinosaurs: List<Dinosaur> = emptyList(),
17	val searchQuery: String = "",
18	val selectedFilter: String = "Semua"
19)
20	
21	
22	class DinoListViewModel : ViewModel() {
23	
24	private val getDinosaursUseCase =
25	GetDinosaursUseCase(DinosaurRepositoryImpl())
26	
27	private val _uiState = MutableStateFlow(DinoListUiState())
28	val uiState: StateFlow<DinoListUiState> =
29	_uiState.asStateFlow()
30	
31	init {
32	loadDinosaurs()
33	}

34	
35	private fun loadDinosaurs() {
36	val dinosaurs = getDinosaursUseCase()
37	_uiState.update {
38	it.copy(
39	allDinosaurs = dinosaurs,
40	filteredDinosaurs = dinosaurs
41)
42	}
43	}
44	
45	fun onSearchQueryChanged(query: String) {
46	_uiState.update { it.copy(searchQuery = query) }
47	filterDinosaurs()
48	}
49	
50	fun onFilterChanged(filter: String) {
51	_uiState.update { it.copy(selectedFilter = filter) }
52	filterDinosaurs()
53	}
54	
55	private fun filterDinosaurs() {
56	val currentState = _uiState.value
57	val filteredList = currentState.allDinosaurs.filter {
58	dinosaur ->
59	val matchesSearch =
60	dinosaur.name.contains(currentState.searchQuery, ignoreCase =
61	true)
62	val matchesFilter = currentState.selectedFilter ==
63	"Semua" dinosaur.type == currentState.selectedFilter
64	matchesSearch && matchesFilter
65	}
66	_uiState.update { it.copy(filteredDinosaurs =
67	filteredList) }
68	}
69	}

Table 9 Source Code DinoListViewModel.kt

- MainActivity.kt

1	package com.example.modul3.presentation
2	
3	import android.os.Bundle
4	import androidx.activity.ComponentActivity
5	import androidx.activity.compose.setContent
6	import androidx.activity.enableEdgeToEdge
7	import androidx.compose.foundation.layout.padding
8	import androidx.compose.material3.*

```

9  import androidx.compose.ui.Modifier
10 import androidx.navigation.compose.rememberNavController
11 import com.example.modul3.presentation.navigation.AppNavigation
12
13 class MainActivity : ComponentActivity() {
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         enableEdgeToEdge()
17         setContent {
18             MaterialTheme {
19                 val navController = rememberNavController()
20                 Scaffold { padding ->
21                     AppNavigation(
22                         navController = navController,
23                         modifier = Modifier.padding(padding)
24                     )
25                 }
26             }
27         }
28     }
29 }
30

```

Table 10 Source Code MainActivity.kt

B. Output Program



Gambar 1 Screenshot Hasil Jawaban Soal 1



Gambar 2 Screenshot Hasil Jawaban Soal 1

C. Pembahasan

- Dinosaur.kt:

File ini menginisiasi sebuah data class bernama **Dinosaur**. Kelas ini berfungsi sebagai model data atau cetak biru untuk objek dinosaurus di dalam aplikasi. Setiap objek Dinosaur akan memiliki lima properti: name (nama dinosaurus), description (deskripsi panjang), imageRes (ID resource gambar dari drawable), type (jenisnya, seperti Karnivora atau Herbivora), dan wikiUrl (URL ke halaman Wikipedia-nya) dan lain lain

- LocalDinosaurDataSource.kt

File ini berfungsi sebagai sumber data lokal dan statis untuk aplikasi. Di dalamnya terdapat satu fungsi, **getDinosaurs()**, yang membuat dan mengembalikan sebuah List dari objek Dinosaur. Semua data dinosaurus, termasuk nama, deskripsi, gambar, dan URL, didefinisikan secara *hardcoded* di sini. Anggapanya adalah database dari aplikasi dinosaurus.

- DinosaurRepository.kt

File ini berfungsi sebagai interface, lalu mendeklarasikan satu fungsi abstrak, **getDinosaurs()**, yang mewajibkan setiap kelas yang mengimplementasikannya untuk menyediakan cara mengembalikan sebuah List<Dinosaur>. Tujuan utama dari interface ini adalah untuk menciptakan abstraksi, memisahkan logika bisnis dari cara spesifik pengambilan data, sehingga di masa depan sumber data dapat diganti (misalnya, dari lokal ke internet) tanpa mengubah bagian lain dari aplikasi.

- DinosaurRepositoryImpl.kt

File ini berisi implementasi konkret dari DinosaurRepository. kelas ini menggunakan kata kunci **override** untuk menyediakan implementasi fungsi **getDinosaurs()**, lalu file ini memiliki cara dengan memanggil metode **LocalDinosaurDataSource.getDinosaurs()** dan langsung mengembalikan hasilnya.

- GetDinosaurUseCase.kt

File ini berguna untuk mendapatkan daftar dinosaurus, lalu class ini bergantung kepada DinosaurRepository yang diterimanya melalui constructor.

- DinoListViewModel.kt

Kelas ini menggunakan **StateFlow** untuk menyimpan dan mengekspos **DinoListUiState**, sebuah data class yang menampung semua status layar seperti daftar lengkap dinosaurus, daftar yang telah difilter, query pencarian, dan jenis filter. Saat **ViewModel** pertama kali dibuat, dan class ini langsung memuat data dinosaurus melalui **GetDinosaursUseCase**.

- DinoListScreen.kt

File ini berisi Composable yang bertanggung jawab untuk membangun antarmuka pengguna (UI) dari daftar dinosaurus. Fungsi ini mengamati `uiState` dari `DinoListViewModel` menggunakan `collectAsState`, sehingga UI akan otomatis diperbarui setiap kali ada perubahan data. Lalu disini menggunakan `LazyVerticalGrid` untuk menampilkan item secara efisien dalam format dua kolom. Setiap item di dalam grid adalah sebuah Card yang berisi gambar, teks, serta dua tombol dengan fungsi berbeda: satu tombol memicu navigasi ke `DinoDetailScreen` melalui `navController`, dan satu lagi memicu `Intent` untuk membuka browser ke halaman Wikipedia.

- DinoDetailScreen.kt

File ini bertujuan untuk menampilkan halaman detail dari satu dinosaurus yang dipilih. Fungsi ini menerima data seperti nama, deskripsi, dan ID gambar sebagai parameter yang dilewatkan melalui proses navigasi. Tampilan UI-nya disusun secara vertikal menggunakan `Column`, menampilkan `Image` dari dinosaurus, diikuti oleh `Text` untuk nama dan deskripsi lengkap. Dan juga terdapat button kembali yang memanggil `navController.popBackStack()`

- AppNavigation.kt

File ini bertujuan untuk mengatur navigasi dalam aplikasi menggunakan `Jetpack Navigation Compose`. Di dalamnya, Composable `AppNavigation` mendefinisikan sebuah `NavHost` yang menjadi wadah untuk semua rute atau tujuan navigasi. `startDestination` diatur ke `"dino_list"` sebagai layar awal. Terdapat dua rute utama: `"dino_list"` untuk `DinoListScreen`, dan `"detail/{name}/{desc}/{imgRes}"` untuk `DinoDetailScreen`, di mana rute kedua ini menggunakan placeholder (`{...}`) untuk menerima argumen.

- MainActivity.kt

File ini adalah titik awal pada aplikasi ini, Di dalam fungsi `onCreate`, ia menggunakan `setContent` untuk membangun seluruh antarmuka pengguna dengan `Jetpack Compose`. Pada bagian ini menginisialisasi `rememberNavController()` untuk membuat dan mengingat `NavController` yang akan mengelola semua perpindahan layar. Struktur dasar aplikasi diatur menggunakan `Scaffold`, dan komponen `AppNavigation` dipanggil di dalamnya.

2. **RecyclerView** masih banyak digunakan karena dua alasan utama: untuk **proyek yang sudah ada (legacy)** yang dibangun dengan sistem XML, dan ketika developer membutuhkan **kustomisasi tingkat lanjut** pada perilaku list yang belum tentu mudah dilakukan di **LazyColumn**.

Penjelasan Detail:

- Sebagian besar aplikasi Android yang ada saat ini dibangun menggunakan sistem **View berbasis XML**, bukan Compose.
- **RecyclerView** memberikan kontrol penuh atas animasi item (menambah, menghapus, memindahkan) melalui `RecyclerView.ItemAnimator`. Developer bisa membuat animasi yang sangat kompleks dan spesifik.

A. Tautan Git

<https://github.com/Yoruuu00/PemrogramanMobile/tree/main/Modul3>