

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



Build a Scrollable List

Oleh:

Alya Rosaan NIM. 2310817320006

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2024**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
MODUL 3

Laporan Praktikum Pemrograman Mobile Modul 2: Build a Scrollable List ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Alya Rosaan
NIM : 2310817320006

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Salsabila Syifa
NIM. 2010817320004

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 01 011

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
A. Source Code.....	6
B. Output Program	11
C. Pembahasan	12
D. Source Code.....	13
E. Output Program	20
F. Pembahasan	20
G. Tautan Git	22

DAFTAR GAMBAR

Gambar 1 Screenshot Output Compose.....	11
Gambar 2 Screenshot Output XML.....	20

DAFTAR TABEL

Tabel 1. Source Code Compose	11
Tabel 2 Source Code XML.....	19

Soal Praktikum

1. Buatlah sebuah aplikasi Android menggunakan XML dan Jetpack Compose yang dapat menampilkan list dengan ketentuan berikut:

1. List menggunakan fungsi RecyclerView (XML) dan 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 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?

Jetpack Compose

A. Source Code

MainActivity.kt

1	package com.example.myanimelistapp
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.layout.*
11	import androidx.compose.foundation.lazy.LazyColumn
12	import androidx.compose.foundation.lazy.items
13	import androidx.compose.foundation.shape.RoundedCornerShape
14	import androidx.compose.material3.*
15	import androidx.compose.runtime.Composable
16	import androidx.compose.ui.Modifier

```

17 import androidx.compose.ui.draw.clip
18 import androidx.compose.ui.layout.ContentScale
19 import androidx.compose.ui.platform.LocalContext
20 import androidx.compose.ui.res.painterResource
21 import androidx.compose.ui.text.style.TextOverflow
22 import androidx.compose.ui.tooling.preview.Preview
23 import androidx.compose.ui.unit.dp
24 import androidx.navigation.NavType
25 import androidx.navigation.NavHostController
26 import androidx.navigation.compose.*
27 import androidx.navigation.navArgument
28 import com.example.myanimelistapp.ui.theme.MyAnimeListAppTheme
29
30 class MainActivity : ComponentActivity() {
31     override fun onCreate(savedInstanceState: Bundle?) {
32         super.onCreate(savedInstanceState)
33         enableEdgeToEdge()
34         setContent {
35             MyAnimeListAppTheme {
36                 val navController = rememberNavController()
37                 NavHost(navController = navController,
38 startDestination = "list") {
39
40                     // Rute detail menerima 3 argumen
41                     composable(
42                         "detail/{title}/{detail}/{imageRes}",
43                         arguments = listOf(
44                             navArgument("title") { type =
45 NavType.StringType },
46                             navArgument("detail") { type =
47 NavType.StringType },
48                             navArgument("imageRes") { type =
49 NavType.IntType }
50                         )
51                     ) { backStackEntry ->
52                         val title =
53 backStackEntry.arguments?.getString("title") ?: ""
54                         val detail =
55 backStackEntry.arguments?.getString("detail") ?: ""
56                         val imageRes =
57 backStackEntry.arguments?.getInt("imageRes") ?: R.drawable.naruto
58                         DetailScreen(title, detail, imageRes)
59                     }
60
61                     composable("list") {
62                         ItemList(navController)
63                     }
64                 }
65             }
66 }

```

```

67
68 // === Data class ===
697 data class ListItem(
0     val title: String,
71     val subtitle: String,
72     val detail: String,
73     val imageRes: Int,
74     val url: String
75 )
76
77 // === Sample data ===
78 val sampleItems = listOf(
    ListItem("Naruto", "\"Naruto Uzumaki adalah seorang ninja remaja
yang bercita-cita menjadi Hokage, pemimpin desa Konoha. Ia menyimpan
kekuatan rubah berekor sembilan dalam tubuhnya, dan harus menghadapi
penolakan, musuh kuat, serta perjalanan keras untuk membuktikan
dirinya.\"\\n\", \"Naruto Uzumaki adalah seorang ninja remaja yang
bercita-cita menjadi Hokage, pemimpin desa Konoha. Ia menyimpan
kekuatan rubah berekor sembilan dalam tubuhnya, dan harus menghadapi
penolakan, musuh kuat, serta perjalanan keras untuk membuktikan
dirinya.\"\\n\", R.drawable.naruto,
\"https://myanimelist.net/anime/20/Naruto\"),
79     ListItem("One Piece", "\"Monkey D. Luffy berlayar bersama kru
Bajak Laut Topi Jerami untuk mencari harta karun legendaris One Piece.
Dengan kemampuan buah iblis dan semangat pantang menyerah, mereka
menjelajahi lautan, melawan bajak laut, dan mengejar mimpi menjadi
Raja Bajak Laut.\"\\n\", \"Monkey D. Luffy berlayar bersama kru Bajak
Laut Topi Jerami untuk mencari harta karun legendaris One Piece.
Dengan kemampuan buah iblis dan semangat pantang menyerah, mereka
menjelajahi lautan, melawan bajak laut, dan mengejar mimpi menjadi
Raja Bajak Laut.\"\\n\", R.drawable.op,
\"https://myanimelist.net/anime/21/One_Piece\"),
80     ListItem("Attack on Titan", "\"Dalam dunia di mana umat manusia
terancam punah oleh para Titan pemakan manusia, Eren Yeager bersumpah
untuk membalas dendam setelah desanya dihancurkan. Ia bergabung dengan
militer dan mengungkap misteri kelam di balik keberadaan Titan dan
sejarah manusia.\"\\n\", \"Dalam dunia di mana umat manusia terancam
punah oleh para Titan pemakan manusia, Eren Yeager bersumpah untuk
membalas dendam setelah desanya dihancurkan. Ia bergabung dengan
militer dan mengungkap misteri kelam di balik keberadaan Titan dan
sejarah manusia.\"\\n\", R.drawable.aot,
\"https://myanimelist.net/anime/16498/Shingeki_no_Kyojin\"),
81     ListItem("Demon Slayer", "\"Tanjiro Kamado menjadi pembasmi iblis
setelah keluarganya dibantai dan adiknya berubah menjadi iblis.
Bersama teman-temannya, Tanjiro menghadapi berbagai iblis berbahaya
 demi mencari cara menyelamatkan adiknya dan membalas kejahatan yang
terjadi.\"\\n\", \"Tanjiro Kamado menjadi pembasmi iblis setelah
keluarganya dibantai dan adiknya berubah menjadi iblis. Bersama teman-
temannya, Tanjiro menghadapi berbagai iblis berbahaya demi mencari
cara menyelamatkan adiknya dan membalas kejahatan yang terjadi.\"\\n\",
R.drawable.kny,
\"https://myanimelist.net/anime/38000/Kimetsu_no_Yaiba\"),

```



```

82      ListItem("Jujutsu Kaisen", "\Yuji Itadori, seorang siswa SMA
dengan kekuatan fisik luar biasa, tanpa sengaja memakan jari iblis
terkutuk Sukuna. Ia bergabung dengan sekolah Jujutsu untuk
mengendalikan kekuatan itu dan melawan kutukan yang mengancam umat
manusia.\n", "\Yuji Itadori, seorang siswa SMA dengan kekuatan
fisik luar biasa, tanpa sengaja memakan jari iblis terkutuk Sukuna. Ia
bergabung dengan sekolah Jujutsu untuk mengendalikan kekuatan itu dan
melawan kutukan yang mengancam umat manusia.\n", R.drawable.jjk,
"https://myanimelist.net/anime/40748/Jujutsu_Kaisen")
)

83  // === List Screen ===
84  @Composable
85  fun ItemList(navController: NavHostController) {
86      val context = LocalContext.current
87      LazyColumn(modifier = Modifier.padding(8.dp)) {
88          items(sampleItems) { item ->
89              Card(
90                  shape = RoundedCornerShape(16.dp),
91                  modifier = Modifier
92                      .padding(8.dp)
93                      .fillMaxWidth()
94                      .clip(RoundedCornerShape(16.dp))
95              ) {
96                  Row(modifier = Modifier.padding(16.dp)) {
97                      // Gambar di kiri (portrait)
98                      Image(
99                          painter = painterResource(id = item.imageRes),
100                         contentDescription = null,
101                         contentScale = ContentScale.Crop,
102                         modifier = Modifier
103                             .width(150.dp)
104                             .height(250.dp)
105                             .clip(RoundedCornerShape(12.dp))
106                     )
107
108                     Spacer(modifier = Modifier.width(16.dp))
109
110                     // Konten teks dan tombol di kanan
111                     Column(
112                         modifier = Modifier
113                             .weight(1f)
114                             .fillMaxHeight(),
115                         verticalArrangement =
Arrangement.spacedBy(8.dp)
116                     ) {
117                         Text(item.title, style =
MaterialTheme.typography.titleMedium)
118                         Text(
119                             item.subtitle,
120                             style =
MaterialTheme.typography.bodySmall,

```

```

120                 maxLines = Int.MAX_VALUE,
121                 overflow = TextOverflow.Visible
122             )
123
124             Column(verticalArrangement =
125                 Arrangement.spacedBy(8.dp)) {
126                 Button(
127                     onClick = {
128                         // Kirim imageRes ke halaman
129                         detail
130                         navController.navigate("detail/${item.title}/${item.detail}/${item.imageRes}")
131                     },
132                     modifier = Modifier.widthIn(min =
133                         100.dp, max = 140.dp)
134                 ) {
135                     Text("Detail", style =
136                         MaterialTheme.typography.labelLarge)
137                 }
138                 Button(
139                     onClick = {
140                         val intent =
141                         Intent(Intent.ACTION_VIEW, Uri.parse(item.url))
142                         context.startActivity(intent)
143                     },
144                     modifier = Modifier.widthIn(min =
145                         100.dp, max = 140.dp)
146                 ) {
147                     Text("Open URL", style =
148                         MaterialTheme.typography.labelLarge)
149                 }
150             }
151         }
152     }
153 }
154
155 // === Detail Screen ===
156 @Composable
157 fun DetailScreen(title: String, detail: String, imageRes: Int) {
158     Column(
159         modifier = Modifier
160             .fillMaxSize()
161             .padding(16.dp)
162     ) {
163         Image(
164             painter = painterResource(id = imageRes),
165             contentDescription = null,

```

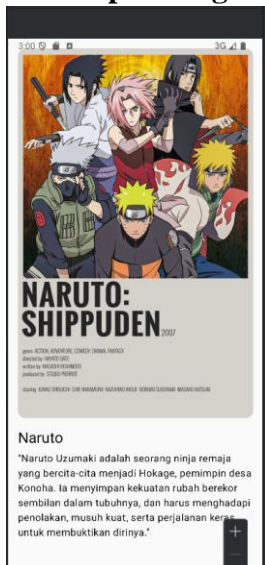
```

165         contentScale = ContentScale.Crop,
166         modifier = Modifier
167             .fillMaxWidth()
168             .height(600.dp)
169             .clip(RoundedCornerShape(12.dp))
170     )
171
172     Spacer(modifier = Modifier.height(16.dp))
173
174     Text(text = title, style =
MaterialTheme.typography.headlineSmall)
175     Spacer(modifier = Modifier.height(8.dp))
176     Text(
177         text = detail,
178         style = MaterialTheme.typography.bodyLarge,
179         maxLines = Int.MAX_VALUE,
180         overflow = TextOverflow.Visible
181     )
182 }
183 }
184
185 @Preview(showBackground = true)
186 @Composable
187 fun DefaultPreview() {
188     MyAnimeListAppTheme {
189         ItemList(rememberNavController())
190     }
191 }

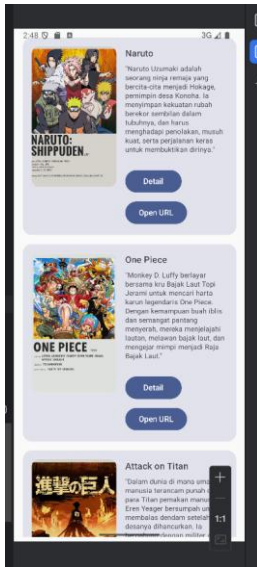
```

Tabel 1. Source Code Compose

B. Output Program



Gambar 1 Screenshot Output Compose



C. Pembahasan

MainActivity.kt:

1. `package com.example.myanimelistapp`
- Menentukan paket utama dari aplikasi ini.
- 2-19. `import ...`
- Mengimpor berbagai library Android dan Jetpack Compose yang diperlukan seperti UI, Navigasi, Layout, Gambar, dsb.
21. `class MainActivity : ComponentActivity()`
- Mendefinisikan activity utama yang akan dijalankan saat aplikasi dibuka.
- 22-24. `override fun onCreate(...) { ... }`
- Fungsi lifecycle utama saat Activity dibuat.
23. `enableEdgeToEdge()`
- Mengaktifkan tampilan layar penuh hingga ke tepi layar.
24. `setContent { ... }`
- Menentukan isi UI yang akan dirender dengan Jetpack Compose.
25. `MyAnimeListAppTheme { ... }`
- Menerapkan tema kustom aplikasi.
26. `val navController = rememberNavController()`
- Membuat controller untuk navigasi antar composable.
- 27-45. `NavHost(navController...) { ... }`
- Menentukan rute navigasi:
- "list" -> menampilkan daftar anime
- "detail/{title}/{detail}/{imageRes}" -> menampilkan halaman detail anime
- 29-35. `composable("detail/...") { ... }`
- Halaman detail, menerima parameter: title, detail, imageRes.
- 36-39. Ambil argumen dari `BackStackEntry`
- Mengambil data yang dikirim dari halaman sebelumnya.

```

40. DetailScreen(...)
    - Menampilkan halaman detail berdasarkan data yang dikirim.
42-44. composable("list") { ItemList(navController) }
    - Halaman daftar anime.
47-54. data class ListItem(...)
    - Model data untuk setiap item anime.
57-70. val sampleItems = listOf(...)
    - Daftar data statis anime.
73-121. @Composable fun ItemList(...)
    - Fungsi composable untk menampilkan daftar anime.
75. val context = LocalContext.current
    - Mendapatkan context untuk keperluan intent.
76. LazyColumn(...)
    - List vertikal yang bisa di-scroll.
77. items(sampleItems) { item -> ... }
    - Iterasi tiap anime dan tampilkan dalam bentuk Card.

79-83. Card(...)
    - Menampilkan kotak berisi konten item.
84-109. Row, Image, Column
    - Menata layout gambar dan teks secara horizontal.
98-105. Tombol Detail dan Open URL
    - Navigasi ke detail & membuka URL dengan Intent.
124-136. @Composable fun DetailScreen(...)
    - Menampilkan detail anime: gambar besar, judul, dan deskripsi.
139-142. @Preview(...)
    - Untuk melihat pratinjau tampilan daftar di Android Studio .

```

XML

D. Source Code

MainActivity.kt

```

1 package com.example.myanimelistapp
2
3 import android.content.Intent
4 import android.net.Uri
5 import android.os.Bundle
6 import androidx.appcompat.app.AppCompatActivity
7 import androidx.recyclerview.widget.LinearLayoutManager
8 import androidx.recyclerview.widget.RecyclerView
9 import com.example.myanimelistappxml.R
10
11 class MainActivity : AppCompatActivity() {
12
13     private val animeList = listOf(
14         ListItem("Naruto", "Seorang ninja muda yang bermimpi
15 menjadi Hokage", R.drawable.naruto,

```

```

16         "Naruto adalah anime populer tentang ninja muda dengan
17 semangat juang tinggi.", "https://naruto.com"),
18         ListItem("One Piece", "Monkey D. Luffy berlayar bersama
19 kru Bajak Laut Topi Jerami untuk mencari harta karun legendaris
20 One Piece. Dengan kemampuan buah iblis dan semangat pantang
21 menyerah, mereka menjelajahi lautan, melawan bajak laut, dan
22 mengejar mimpi menjadi Raja Bajak Laut.", R.drawable.naruto,
23         "Monkey D. Luffy berlayar bersama kru Bajak Laut Topi
24 Jerami untuk mencari harta karun legendaris One Piece. Dengan
25 kemampuan buah iblis dan semangat pantang menyerah, mereka
26 menjelajahi lautan, melawan bajak laut, dan mengejar mimpi menjadi
27 Raja Bajak Laut.", "https://naruto.com"),
28         ListItem("Attack On Titan", "Dalam dunia di mana umat
29 manusia terancam punah oleh para Titan pemakan manusia, Eren
30 Yeager bersumpah untuk membalas dendam setelah desanya
31 dihancurkan. Ia bergabung dengan militer dan mengungkapkan misteri
32 kelam di balik keberadaan Titan dan sejarah manusia.",
33         R.drawable.naruto,
34         "Dalam dunia di mana umat manusia terancam punah oleh
35 para Titan pemakan manusia, Eren Yeager bersumpah untuk membalas
36 dendam setelah desanya dihancurkan. Ia bergabung dengan militer
37 dan mengungkapkan misteri kelam di balik keberadaan Titan dan sejarah
38 manusia.",
39         "https://myanimelist.net/anime/16498/Shingeki_no_Kyojin"),
40         ListItem("Demon Slayer", "Tanjiro Kamado menjadi pembasmi
41 iblis setelah keluarganya dibantai dan adiknya berubah menjadi
42 iblis. Bersama teman-temannya, Tanjiro menghadapi berbagai iblis
43 berbahaya demi mencari cara menyelamatkan adiknya dan membalas
44 kejahatan yang terjadi.", R.drawable.naruto,
45         "Tanjiro Kamado menjadi pembasmi iblis setelah
46 keluarganya dibantai dan adiknya berubah menjadi iblis. Bersama
47 teman-temannya, Tanjiro menghadapi berbagai iblis berbahaya demi
48 mencari cara menyelamatkan adiknya dan membalas kejahatan yang
49 terjadi.",
50         "https://myanimelist.net/anime/38000/Kimetsu_no_Yaiba"),
51         ListItem("Jujutsu Kaisen", "Yuji Itadori, seorang siswa
52 SMA dengan kekuatan fisik luar biasa, tanpa sengaja memakan jari
53 iblis terkutuk Sukuna. Ia bergabung dengan sekolah Jujutsu untuk
54 mengendalikan kekuatan itu dan melawan kutukan yang mengancam umat
55 manusia.", R.drawable.naruto,
56         "Yuji Itadori, seorang siswa SMA dengan kekuatan fisik
57 luar biasa, tanpa sengaja memakan jari iblis terkutuk Sukuna. Ia
58 bergabung dengan sekolah Jujutsu untuk mengendalikan kekuatan itu
59 dan melawan kutukan yang mengancam umat manusia.",
60         "https://myanimelist.net/anime/40748/Jujutsu_Kaisen"),
61     )
62     override fun onCreate(savedInstanceState: Bundle?) {
63         super.onCreate(savedInstanceState)
64         setContentView(R.layout.activity_main)
65
66         val recyclerView: RecyclerView =
67             findViewById(R.id.recyclerView)

```

```

65         recyclerView.layoutManager = LinearLayoutManager(this)
66         recyclerView.adapter = AnimeAdapter(animeList,
67             onDetailClick = { item ->
68                 val intent = Intent(this,
69 DetailActivity::class.java).apply {
69                     putExtra("title", item.title)
70                     putExtra("desc", item.detailDesc)
71                     putExtra("imageResId", item.imageResId)
72                 }
73                 startActivity(intent)
74             },
75             onOpenUrlClick = { item ->
76                 val intent = Intent(Intent.ACTION_VIEW,
77 Uri.parse(item.url))
78                 startActivity(intent)
79             }
80         )
    }
}

```

ListItem.kt

```

1 package com.example.myanimelistapp
2
3 data class ListItem(
4     val title: String,
5     val subtitle: String,
6     val imageResId: Int,
7     val detailDesc: String,
8     val url: String
9 )

```

AnimeAdapter.kt

```

1 package com.example.myanimelistapp
2
3 import android.view.LayoutInflater
4 import android.view.View
5 import android.view.ViewGroup
6 import android.widget.Button
7 import android.widget.ImageView
8 import android.widget.TextView
9 import androidx.recyclerview.widget.RecyclerView
10 import com.example.myanimelistapp.xml.R
11
12 class AnimeAdapter(
13     private val animeList: List<ListItem>,
14     private val onDetailClick: (ListItem) -> Unit,
15     private val onOpenUrlClick: (ListItem) -> Unit

```

```

16 ) : RecyclerView.Adapter<AnimeAdapter.AnimeViewHolder>() {
17
18     class AnimeViewHolder(view: View) :
19     RecyclerView.ViewHolder(view) {
20         val imgAnime: ImageView = view.findViewById(R.id.imgAnime)
21         val tvTitle: TextView = view.findViewById(R.id.tvTitle)
22         val tvSubtitle: TextView =
23         view.findViewById(R.id.tvSubtitle)
24         val btnDetail: Button = view.findViewById(R.id.btnDetail)
25         val btnOpenUrl: Button =
26         view.findViewById(R.id.btnOpenUrl)
27     }
28
29     override fun onCreateViewHolder(parent: ViewGroup, viewType:
30     Int): AnimeViewHolder {
31         val view =
32         LayoutInflater.from(parent.context).inflate(R.layout.item_anime,
33         parent, false)
34         return AnimeViewHolder(view)
35     }
36
37     override fun onBindViewHolder(holder: AnimeViewHolder,
38     position: Int) {
39         val item = animeList[position]
40         holder.imgAnime.setImageResource(item.imageResId)
41         holder.tvTitle.text = item.title
42         holder.tvSubtitle.text = item.subtitle
43         holder.btnDetail.setOnClickListener { onDetailClick(item)
44     }
45         holder.btnOpenUrl.setOnClickListener {
46         onOpenUrlClick(item) }
47     }
48
49     override fun getItemCount() = animeList.size
50 }

```

DetailActivity.kt

```

1 package com.example.myanimelistapp
2
3 import android.os.Bundle
4 import android.widget.ImageView
5 import android.widget.TextView
6 import androidx.appcompat.app.AppCompatActivity
7 import com.example.myanimelistapp.xml.R
8
9 class DetailActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_detail)

```



```

13
14         val img = findViewById<ImageView>(R.id.imgDetail)
15         val tvTitle = findViewById<TextView>(R.id.tvDetailTitle)
16         val tvDesc = findViewById<TextView>(R.id.tvDetailDesc)
17
18         val title = intent.getStringExtra("title")
19         val desc = intent.getStringExtra("desc")
20         val imageResId = intent.getIntExtra("imageResId", 0)
21
22         tvTitle.text = title
23         tvDesc.text = desc
24         img.setImageResource(imageResId)
25     }
26 }

```

Activity_main.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <androidx.recyclerview.widget.RecyclerView
9         android:id="@+id/recyclerView"
10        android:layout_width="0dp"
11        android:layout_height="0dp"
12        android:padding="8dp"
13        app:layout_constraintTop_toTopOf="parent"
14        app:layout_constraintBottom_toBottomOf="parent"
15        app:layout_constraintStart_toStartOf="parent"
16        app:layout_constraintEnd_toEndOf="parent"/>
17
18 </androidx.constraintlayout.widget.ConstraintLayout>

```

item_anime.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:orientation="horizontal"
5     android:padding="8dp"
6     android:layout_width="match_parent"
7     android:layout_height="wrap_content">
8
9     <ImageView
10        android:id="@+id/imgAnime"
11        android:layout_width="100dp"

```

12	android:layout_height="150dp"
13	android:scaleType="centerCrop"
14	android:layout_marginEnd="8dp"/>
15	
16	<LinearLayout
17	android:orientation="vertical"
18	android:layout_width="0dp"
19	android:layout_height="match_parent"
20	android:layout_weight="1">
21	
22	<TextView
23	android:id="@+id/tvTitle"
24	android:textStyle="bold"
25	android:textSize="18sp"
26	android:layout_width="wrap_content"
27	android:layout_height="wrap_content"/>
28	
29	<TextView
30	android:id="@+id/tvSubtitle"
31	android:textSize="14sp"
32	android:maxLength="4"
33	android:ellipsize="end"
34	android:layout_width="wrap_content"
35	android:layout_height="wrap_content"/>
36	
37	<Button
38	android:id="@+id/btnDetail"
39	android:text="Detail"
40	android:layout_width="wrap_content"
41	android:layout_height="wrap_content"/>
42	
43	<Button
44	android:id="@+id/btnOpenUrl"
45	android:text="Open URL"
46	android:layout_width="wrap_content"
47	android:layout_height="wrap_content"/>
48	</LinearLayout>
49	</LinearLayout>
50	

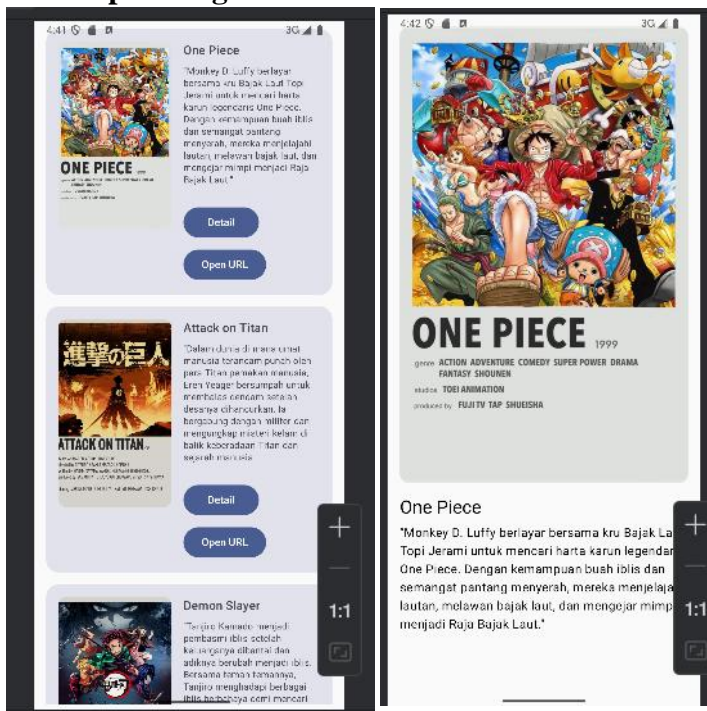
activity_detail.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<ScrollView
3	xmlns:android="http://schemas.android.com/apk/res/android"
4	android:layout_width="match_parent"
5	android:layout_height="match_parent"
6	android:padding="16dp">
7	
8	<LinearLayout

9	android:orientation="vertical"
10	android:layout_width="match_parent"
11	android:layout_height="wrap_content">
12	
13	<ImageView
14	android:id="@+id/imgDetail"
15	android:layout_width="match_parent"
16	android:layout_height="400dp"
17	android:scaleType="centerCrop"/>
18	
19	<TextView
20	android:id="@+id/tvDetailTitle"
21	android:textSize="20sp"
22	android:textStyle="bold"
23	android:layout_marginTop="16dp"
24	android:layout_width="wrap_content"
25	android:layout_height="wrap_content"/>
26	
27	<TextView
28	android:id="@+id/tvDetailDesc"
29	android:layout_width="wrap_content"
30	android:layout_height="wrap_content"/>
31	</LinearLayout>
32	</ScrollView>
33	
34	

Tabel 2 Source Code XML

E. Output Program



Gambar 2 Screenshot Output XML

F. Pembahasan

MainActivity.kt:

1. `package com.example.myanimelistapp`
 - Menyatakan file ini bagian dari package tersebut.
2. `import ...`
 - Mengimpor berbagai kelas Android seperti Intent, Uri, RecyclerView.
3. `class MainActivity : AppCompatActivity()`
 - Kelas utama turunan dari AppCompatActivity.
4. `val animeList = listOf(...)`
 - List data anime yang berisi title, subtitle, gambar, deskripsi, dan url.
5. `override fun onCreate()`
 - Metode awal saat Activity dijalankan.
6. `setContentView(R.layout.activity_main)`
 - Mengatur layout utama yang digunakan.
7. `val recyclerView: RecyclerView = findViewById(R.id.recyclerView)`
 - Mengakses RecyclerView dari layout.
8. `recyclerView.layoutManager = LinearLayoutManager(this)`
 - Menyusun item secara vertikal.
9. `recyclerView.adapter = AnimeAdapter(...)`
 - Menghubungkan adapter dan callback tombol.

ListItem.kt

10. `data class ListItem(...)`

- Struktur data untuk menyimpan data anime.

AnimeAdapter.kt

11. `class AnimeAdapter(...) : RecyclerView.Adapter`
 - Adapter untuk mengatur data anime dalam RecyclerView.
12. `class AnimeViewHolder(view: View)`
 - ViewHolder untuk menyimpan elemen UI per item.
13. `onCreateViewHolder()`
 - Membuat view item dari XML.
14. `onBindViewHolder()`
 - Mengisi data ke setiap elemen UI.

DetailActivity.kt

15. `class DetailActivity : AppCompatActivity()`
 - Menampilkan detail anime yang dikirim dari MainActivity.
16. `intent.getStringExtra / getIntentExtra`
 - Mengambil data yang dikirim lewat Intent.
 - Menampilkan title, desc, dan image di layout activity_detail.

activity_main.xml:

1. `<?xml version="1.0" encoding="utf-8"?>`
`<androidx.constraintlayout.widget.ConstraintLayout ... >`
 - Menggunakan ConstraintLayout sebagai root layout, memungkinkan elemen UI diposisikan fleksibel dan responsif.
2. `<androidx.recyclerview.widget.RecyclerView ... />`
 - RecyclerView digunakan untuk menampilkan daftar (list) item anime.
 - `android:id="@+id/recyclerView"`: ID unik untuk RecyclerView ini.
 - `layout_width & layout_height` diatur 0dp agar mengikuti constraint.
 - `Padding 8dp` untuk memberi jarak dari tepi.
 - `app:layout_constraintTop_toTopOf="parent", dst:` mengikat RecyclerView ke semua sisi parent agar memenuhi layar.

item_anime.xml:

3. `<LinearLayout ... orientation="horizontal">`
 - Mengatur item dalam arah horizontal (gambar di kiri, teks di kanan).
4. `<ImageView android:id="@+id/imgAnime" ... />`
 - Gambar anime dengan lebar 100dp dan tinggi 150dp, crop agar pas.
5. `<LinearLayout orientation="vertical">`
 - Menampung teks dan tombol secara vertikal.
6. `<TextView android:id="@+id/tvTitle" ... />`
 - Menampilkan judul anime dengan teks tebal dan besar.
7. `<TextView android:id="@+id/tvSubtitle" ... />`
 - Menampilkan deskripsi singkat (subtitle), maksimal 4 baris.
8. `<Button android:id="@+id/btnDetail" text="Detail" />`
 - Tombol untuk membuka detail anime.

9. `<Button android:id="@+id/btnOpenUrl" text="Open URL" />`
- Tombol untuk membuka link anime di browser.

activity_detail.xml:

10. `<ScrollView ... >`
- Membuat tampilan bisa discroll jika konten panjang.
11. `<LinearLayout orientation="vertical">`
- Layout vertikal menampung gambar dan teks.
12. `<ImageView android:id="@+id/imgDetail" ... />`
- Gambar anime di detail, lebar penuh dan tinggi 400dp.
13. `<TextView android:id="@+id/tvDetailTitle" ... />`
- Judul anime pada halaman detail.
14. `<TextView android:id="@+id/tvDetailDesc" ... />`
- Deskripsi lengkap anime.
- `</LinearLayout>`
`</ScrollView>`

G. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/ALYAROSAAN/Pemrograman-Mobile-Modul3>