

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 3**



**BUILD A SCROLLABLE LIST**

**Oleh:**

**Damarjati Suryo Laksono**

**NIM. 2310817210014**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS LAMBUNG MANGKURAT  
MEI 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 : Damarjati Suryo Laksono

NIM : 2310817210014

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.....	8
B. Output Program .....	26
C. Pembahasan .....	28
SOAL 2.....	36
D. Tautan Git .....	37

## DAFTAR GAMBAR

Gambar 1 Contoh UI List .....	7
Gambar 2 Gambar UI Detail .....	8
Gambar 1 Screenshot Hasil Jawaban Soal 1 Awalan .....	26
Gambar 2 Screenshot Hasil Jawaban Soal 1 .....	27
Gambar 3 Screenshot Hasil Jawaban Soal 1 .....	27

## DAFTAR TABEL

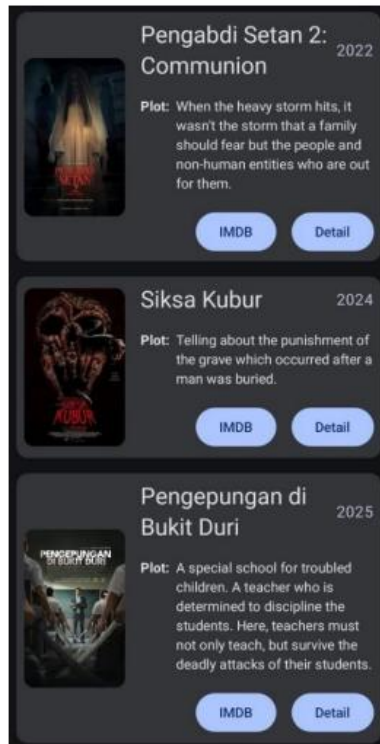
Tabel 1. 1 Source Code Jawaban Soal 1 .....	<b>Error! Bookmark not defined.</b>
Tabel 1. 2 Source Code Jawaban Soal 1 .....	<b>Error! Bookmark not defined.</b>
Tabel 1. 3 Source Code Jawaban Soal 1 .....	12
Tabel 1. 4 Source Code Jawaban Soal 1 .....	15
Tabel 1. 5 Source Code Jawaban Soal 1 .....	15

## SOAL 1

### Soal Praktikum:

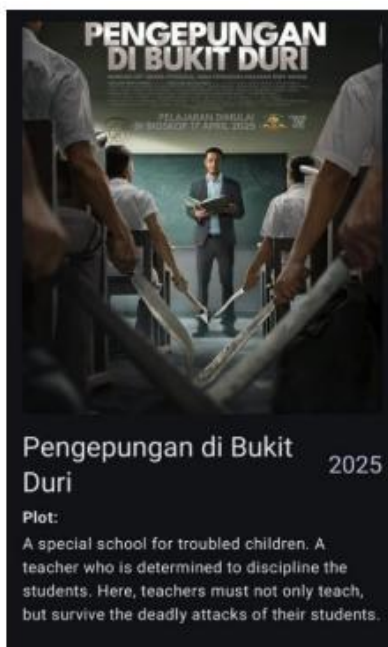
1. Buatlah sebuah aplikasi Android menggunakan XML atau Jetpack Compose yang dapat menampilkan list dengan ketentuan berikut:
  1. List menggunakan fungsi RecyclerView (XML) atau LazyColumn (Compose)
  2. List paling sedikit menampilkan 5 item. Tema item yang ingin ditampilkan bebas
  3. Item pada list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

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



*Gambar 1 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:



## A. Source Code

### 1. MainActivity.kt

```
1 package com.example.myrecylerview
2
3 import android.os.Bundle
4 import android.widget.Toast
5 import androidx.appcompat.app.AppCompatActivity
6 import androidx.appcompat.app.AppCompatActivity
7 import
8     androidx.core.splashscreen.SplashScreen.Companion.installSpla
9 shScreen
10 import
11 com.example.myrecylerview.databinding.ActivityMainBinding
12
13 class MainActivity : AppCompatActivity() {
14
15     private lateinit var binding: ActivityMainBinding
16
17     override fun onCreate(savedInstanceState: Bundle?) {
18
19         AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_
20 NIGHT_NO)
21         installSplashScreen()
22
23         super.onCreate(savedInstanceState)
24         binding = ActivityMainBinding.inflate(layoutInflater)
25         setContentView(binding.root)
26
27         binding.btnMenu.setOnClickListener {
28             Toast.makeText(this, "This Button is used for
29 Decoration", Toast.LENGTH_SHORT).show()
```



28	}
29	}
30	}

*Tabel 1 Source Code Jawaban Soal 1*

## 2. DetailFragment.kt

1	package com.example.myrecyclerview
2	
3	import android.content.Intent
4	import android.os.Bundle
5	import android.view.LayoutInflater
6	import android.view.View
7	import android.view.ViewGroup
8	import androidx.fragment.app.Fragment
9	import androidx.navigation.fragment.findNavController
	import
10	com.example.myrecyclerview.databinding.FragmentDetailBinding
11	
12	class DetailFragment : Fragment() {
13	
14	private var _binding: FragmentDetailBinding? = null
15	private val binding get() = _binding!!
16	
17	override fun onCreateView(
18	inflater: LayoutInflater, container: ViewGroup?,
19	savedInstanceState: Bundle?
20	): View {
	_binding = FragmentDetailBinding.inflate(inflater,
21	container, false)
22	return binding.root
23	}
24	
25	override fun onViewCreated(view: View,

```

26 savedInstanceState: Bundle?) {
27     super.onViewCreated(view, savedInstanceState)
28
29     val args =
30 DetailFragmentArgs.fromBundle(requireArguments())
31     val photo = args.extraPhoto
32     val link = args.extraLink
33     val detail = args.extraDetail
34
35     binding.detailImage.setImageResource(photo)
36     binding.detailDescription.text = detail
37
38     binding.btnBack.setOnClickListener {
39         findNavController().navigateUp()
40     }
41
42     binding.btnShare.setOnClickListener {
43         val shareText = buildString {
44             append("Check this out!\n\n$detail\n\nMore
45 info: $link")
46         }
47
48         val shareIntent =
49 Intent(Intent.ACTION_SEND).apply {
50             type = "text/plain"
51             putExtra(Intent.EXTRA_TEXT, shareText)
52         }
53
54         startActivity(Intent.createChooser(shareIntent,
55 "Share via"))
56     }
57 }
58
59 override fun onDestroyView() {

```

60	<code>super.onDestroyView()</code>
61	<code>_binding = null</code>
62	<code>}</code>
63	<code>}</code>

*Tabel 2 Source Code Jawaban Soal 1*

### 3. MyAdapter.kt

1	<code>package com.example.myrecyclerview</code>
2	
3	<code>import android.view.LayoutInflater</code>
4	<code>import android.view.ViewGroup</code>
5	<code>import androidx.recyclerview.widget.RecyclerView</code>
6	<code>import com.example.myrecyclerview.databinding.ItemLayoutBinding</code>
7	
8	<code>class MyAdapter(</code>
9	<code>    private val listCharacter: ArrayList&lt;MyData&gt;,</code>
10	<code>    private val onYTClick: (String) -&gt; Unit,</code>
11	<code>    private val onDetailClick: (String, Int, String) -&gt; Unit</code>
12	<code>) : RecyclerView.Adapter&lt;MyAdapter.ListViewHolder&gt;() {</code>
13	
14	<code>    class ListViewHolder(val binding: ItemLayoutBinding) :</code>
15	<code>RecyclerView.ViewHolder(binding.root)</code>
16	
17	<code>        override fun onCreateViewHolder(parent: ViewGroup,</code>
18	<code>viewType: Int): ListViewHolder {</code>
19	<code>            val binding =</code>
20	<code>ItemLayoutBinding.inflate(LayoutInflater.from(parent.context),</code>
21	<code>parent, false)</code>
22	<code>            return ListViewHolder(binding)</code>
23	<code>        }</code>
24	
25	<code>        override fun getItemCount(): Int = listCharacter.size</code>

26	
27	override fun onBindViewHolder(holder: ListViewHolder,
28	position: Int) {
29	val (name, link, photo, detail, subtext) =
30	listCharacter[position]
31	with(holder.binding) {
32	textTitle.text = name
33	textDescription.text = subtext
34	itemImage.setImageResource(photo)
35	btnWebsite.setOnClickListener { onYTClick(link) }
36	btnDetails.setOnClickListener {
37	onDetailClick(detail, photo, link) }
38	}
39	}
40	}

*Tabel 3 Source Code Jawaban Soal 1*

#### 4. MyFragment.kt

1	package com.example.myrecylerview
2	
	import android.content.Intent
3	import android.net.Uri
4	import android.os.Bundle
5	import android.view.LayoutInflater
6	import android.view.View
7	import android.view.ViewGroup
8	import androidx.fragment.app.Fragment
9	import androidx.navigation.fragment.findNavController
10	import androidx.recyclerview.widget.LinearLayoutManager
11	import
12	com.example.myrecylerview.databinding.FragmentMyBinding

```

13
14
15 class MyFragment : Fragment() {
16
17     private var _binding: FragmentMyBinding? = null
18     private val binding get() = _binding!!
19
20     private lateinit var characterAdapter: MyAdapter
21     private val list = ArrayList<MyData>()
22
23     override fun onCreateView(
24         inflater: LayoutInflater, container: ViewGroup?,
25         savedInstanceState: Bundle?
26     ): View {
27         _binding = FragmentMyBinding.inflate(inflater,
28 container, false)
29         list.clear()
30         list.addAll(getListCharacter())
31         setupRecyclerView()
32         return binding.root
33     }
34
35     private fun setupRecyclerView() {
36         characterAdapter = MyAdapter(
37             list,
38             onYTClick = { link ->
39                 val intent = Intent(Intent.ACTION_VIEW,
40 Uri.parse(link))
41                 startActivity(intent)
42             },
43             onDetailClick = { detail, photo, link ->
44                 val action = MyFragmentDirections
45                     .actionMyFragmentToDetailFragment(photo,
46 link, detail)

```

```

47         findNavController().navigate(action)
48     }
49 )
50
51     binding.rvCharacter.apply {
52         layoutManager =
53     LinearLayoutManager(requireContext())
54         adapter = characterAdapter
55         setHasFixedSize(true)
56     }
57 }
58
59     private fun getListCharacter(): ArrayList<MyData> {
60         val dataName =
61     resources.getStringArray(R.array.data_name)
62         val dataLink =
63     resources.getStringArray(R.array.data_link)
64         val dataSubtext =
65     resources.getStringArray(R.array.data_subtext)
66         val dataPhoto =
67     resources.obtainTypedArray(R.array.data_photo)
68         val dataDetail =
69     resources.getStringArray(R.array.data_detail)
70         val listCharacter = ArrayList<MyData>()
71         for (i in dataName.indices) {
72             val character = MyData(
73                 name = dataName[i],
74                 link = dataLink[i],
75                 subtext = dataSubtext[i],
76                 detail = dataDetail[i],
77                 photo = dataPhoto.getResourceId(i, -1)
78             )
79             listCharacter.add(character)
80         }

```

81	dataPhoto.recycle()
82	return listCharacter
83	}
84	
85	override fun onDestroyView() {
86	super.onDestroyView()
87	_binding = null
88	}
89	}

*Tabel 4 Source Code Jawaban Soal 1*

## 5. MyData.kt

1	package com.example.myrecylerview
2	
	import android.os.Parcelable
3	import kotlinx.parcelize.Parcelize
4	
5	@Parcelize
6	data class MyData(
7	val name: String,
8	val link: String,
9	val photo: Int,
10	val detail: String,
11	val subtext: String
12	
13	):Parcelable

*Tabel 5 Source Code Jawaban Soal 1*

## 6. activity\_main.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout

	xmlns:android="http://schemas.android.com/apk/res/android"
3	xmlns:app="http://schemas.android.com/apk/res-auto"
4	xmlns:tools="http://schemas.android.com/tools"
5	android:layout_width="match_parent"
6	android:layout_height="match_parent"
7	android:background="@android:color/white"
8	tools:context=".MainActivity">
9	
10	<com.google.android.material.appbar.MaterialToolbar
11	android:id="@+id/toolbar"
12	android:layout_width="match_parent"
13	android:layout_height="?attr/actionBarSize"
14	android:background="@android:color/white"
15	android:elevation="4dp"
16	app:layout_constraintEnd_toEndOf="parent"
17	app:layout_constraintStart_toStartOf="parent"
18	app:layout_constraintTop_toTopOf="parent"
19	app:title="Owi Core"
20	app:titleTextColor="@android:color/black"
21	
22	app:titleTextAppearance="@style/TextAppearance.MaterialCompo
23	nts.Headline6">
24	
25	<ImageView
26	android:id="@+id/btn_menu"
27	android:layout_width="24dp"
28	android:layout_height="24dp"
29	android:layout_gravity="end"
30	android:layout_marginEnd="16dp"
31	android:src="@drawable/ic_more_vert"
32	app:tint="@android:color/black" />
33	
34	</com.google.android.material.appbar.MaterialToolbar>
35	



36	<androidx.fragment.app.FragmentContainerView
37	android:id="@+id/nav_host_fragment"
38	
39	android:name="androidx.navigation.fragment.NavHostFragment"
40	android:layout_width="0dp"
41	android:layout_height="0dp"
42	app:layout_constraintTop_toBottomOf="@id/toolbar"
43	app:layout_constraintBottom_toBottomOf="parent"
44	app:layout_constraintStart_toStartOf="parent"
45	app:layout_constraintEnd_toEndOf="parent"
46	app:navGraph="@navigation/nav_graph"
47	app:defaultNavHost="true" />
48	
49	</androidx.constraintlayout.widget.ConstraintLayout>

*Tabel 6 Source Code Jawaban Soal 1*

## 7. fragment\_detail.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.core.widget.NestedScrollView
3	xmlns:android="http://schemas.android.com/apk/res/android"
4	xmlns:app="http://schemas.android.com/apk/res-auto"
5	xmlns:tools="http://schemas.android.com/tools"
6	android:layout_width="match_parent"
7	android:layout_height="match_parent"
8	android:fillViewport="true">
9	
10	<androidx.constraintlayout.widget.ConstraintLayout
11	android:layout_width="match_parent"
12	android:layout_height="wrap_content"
13	android:padding="16dp">
14	

15	<ImageButton
16	android:id="@+id/btn_back"
17	android:layout_width="48dp"
18	android:layout_height="48dp"
19	
20	android:background="?selectableItemBackgroundBorderless"
21	android:src="@drawable/ic_arrow_back"
22	app:layout_constraintStart_toStartOf="parent"
23	app:layout_constraintTop_toTopOf="parent"
24	app:tint="@android:color/black" />
25	
26	
27	<com.google.android.material.imageview.ShapeableImageView
28	android:id="@+id/detail_image"
29	android:layout_width="200dp"
30	android:layout_height="200dp"
31	android:scaleType="centerCrop"
32	app:layout_constraintEnd_toEndOf="parent"
33	app:layout_constraintStart_toStartOf="parent"
34	app:layout_constraintTop_toTopOf="parent"
35	app:layout_constraintVertical_bias="0"
36	android:layout_marginTop="32dp"
37	android:elevation="4dp"
38	
39	android:transitionName="shared_image_container"/>
40	
41	<LinearLayout
42	android:layout_width="match_parent"
43	android:layout_height="wrap_content"
44	android:orientation="vertical"
45	android:padding="24dp"
46	app:layout_constraintEnd_toEndOf="parent"
47	app:layout_constraintStart_toStartOf="parent"
48	

```

49 app:layout_constraintTop_toBottomOf="@id/detail_image"
50     android:layout_marginTop="24dp">
51
52     <androidx.cardview.widget.CardView
53         android:layout_width="match_parent"
54         android:layout_height="wrap_content"
55         app:cardCornerRadius="16dp"
56         app:cardElevation="4dp"
57
58         app:cardBackgroundColor="@android:color/white"
59         android:layout_marginBottom="16dp">
60
61         <TextView
62             android:id="@+id/detail_description"
63             android:layout_width="match_parent"
64             android:layout_height="wrap_content"
65             android:textSize="16sp"
66             android:lineSpacingMultiplier="1.2"
67             android:textColor="@android:color/black"
68             android:padding="24dp"
69             tools:text="Lorem ipsum dolor sit amet,
70 consectetur adipiscing elit. Sed do eiusmod tempor
71 incididunt ut labore et dolore magna aliqua. Ut enim ad
72 minim veniam, quis nostrud exercitation ullamco laboris nisi
73 ut aliquip ex ea commodo consequat."/>
74
75     </androidx.cardview.widget.CardView>
76
77     <LinearLayout
78         android:layout_width="match_parent"
79         android:layout_height="wrap_content"
80         android:orientation="horizontal"
81         android:gravity="center_horizontal"
82         android:spacing="16dp">

```

83	
84	
85	<com.google.android.material.button.MaterialButton
86	android:id="@+id/btn_share"
87	
88	style="@style/Widget.Material3.Button.OutlinedButton"
89	android:layout_width="match_parent"
90	android:layout_height="48dp"
91	android:text="Share"
92	app:icon="@drawable/ic_share"
93	app:iconTint="@null"
94	app:cornerRadius="8dp"/>
95	
96	</LinearLayout>
97	
98	</LinearLayout>
99	
100	</androidx.constraintlayout.widget.ConstraintLayout>
101	
102	</androidx.core.widget.NestedScrollView>

*Tabel 7 Source Code Jawaban Soal 1*

## 8. fragment\_my.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
	xmlns:android="http://schemas.android.com/apk/res/android"
3	xmlns:app="http://schemas.android.com/apk/res-auto"
4	xmlns:tools="http://schemas.android.com/tools"
5	android:layout_width="match_parent"
6	android:layout_height="match_parent"
7	tools:context=".MyFragment">
8	
9	
10	<androidx.recyclerview.widget.RecyclerView

11	android:id="@+id/rv_character"
12	android:layout_width="0dp"
13	android:layout_height="0dp"
14	app:layout_constraintBottom_toBottomOf="parent"
15	app:layout_constraintEnd_toEndOf="parent"
16	app:layout_constraintStart_toStartOf="parent"
17	app:layout_constraintTop_toTopOf="parent" />
18	</androidx.constraintlayout.widget.ConstraintLayout>

*Tabel 8 Source Code Jawaban Soal 1*

## 9. item\_layout.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.cardview.widget.CardView
3	xmlns:android="http://schemas.android.com/apk/res/android"
4	xmlns:app="http://schemas.android.com/apk/res-auto"
5	xmlns:tools="http://schemas.android.com/tools"
6	android:layout_width="match_parent"
7	android:layout_height="wrap_content"
8	android:layout_margin="2dp"
9	app:cardCornerRadius="12dp"
10	app:cardElevation="4dp"
11	app:cardUseCompatPadding="true">
12	
13	<androidx.constraintlayout.widget.ConstraintLayout
14	android:layout_width="match_parent"
15	android:layout_height="wrap_content"
16	android:padding="16dp">
17	
18	<ImageView
19	android:src="@drawable/pakdhe_1"
20	android:id="@+id/item_image"
21	android:layout_width="64dp"
22	android:layout_height="64dp"

23	android:scaleType="centerCrop"
24	app:layout_constraintBottom_toBottomOf="parent"
25	app:layout_constraintStart_toStartOf="parent"
26	app:layout_constraintTop_toTopOf="parent"
27	android:layout_marginEnd="16dp"/>
28	
29	<LinearLayout
30	android:layout_width="0dp"
31	android:layout_height="wrap_content"
32	android:orientation="vertical"
33	
34	app:layout_constraintBottom_toBottomOf="@id/item_image"
35	
36	app:layout_constraintEnd_toStartOf="@+id/button_group"
37	
38	app:layout_constraintStart_toEndOf="@id/item_image"
39	
40	app:layout_constraintTop_toTopOf="@id/item_image">
41	
42	<TextView
43	android:id="@+id/text_title"
44	android:layout_width="match_parent"
45	android:layout_height="wrap_content"
46	android:layout_marginStart="8dp"
47	android:ellipsize="end"
48	android:maxLines="1"
49	android:textColor="@android:color/black"
50	android:textSize="18sp"
51	android:textStyle="bold"
52	tools:text="Main Title" />
53	
54	<TextView
55	android:id="@+id/text_description"
56	android:layout_width="match_parent"

```

57         android:layout_height="wrap_content"
58         android:layout_marginStart="8dp"
59         android:layout_marginTop="4dp"
60         android:ellipsize="end"
61         android:maxLines="2"
62
63     android:textColor="@android:color/darker_gray"
64         android:textSize="14sp"
65         tools:text="Secondary description text that
66     can span multiple lines" />
67     </LinearLayout>
68
69     <LinearLayout
70         android:id="@+id/button_group"
71         android:layout_width="wrap_content"
72         android:layout_height="wrap_content"
73         android:orientation="vertical"
74         android:gravity="end"
75         android:spacing="8dp"
76         app:layout_constraintBottom_toBottomOf="parent"
77         app:layout_constraintEnd_toEndOf="parent"
78         app:layout_constraintTop_toTopOf="parent">
79
80
81     <com.google.android.material.button.MaterialButton
82         android:id="@+id/btn_details"
83
84     style="@style/Widget.Material3.Button.OutlinedButton"
85         android:layout_width="wrap_content"
86         android:layout_height="36dp"
87         android:minWidth="64dp"
88         app:icon="@drawable/ic_info_outline"
89         app:iconPadding="0dp"
90         app:iconGravity="textStart"

```

91	app:cornerRadius="8dp"/>
92	
93	
94	<com.google.android.material.button.MaterialButton
95	android:id="@+id/btn_website"
96	
97	style="@style/Widget.Material3.Button.TonalButton"
98	android:layout_width="wrap_content"
99	android:layout_height="36dp"
100	android:minWidth="64dp"
101	app:icon="@drawable/ic_open_in_browser"
102	app:iconPadding="0dp"
103	app:iconGravity="textStart"
104	app:cornerRadius="8dp"/>
105	</LinearLayout>
106	
107	</androidx.constraintlayout.widget.ConstraintLayout>
108	</androidx.cardview.widget.CardView>

*Tabel 9 Source Code Jawaban Soal 1*

## 10. nav\_graph.xml

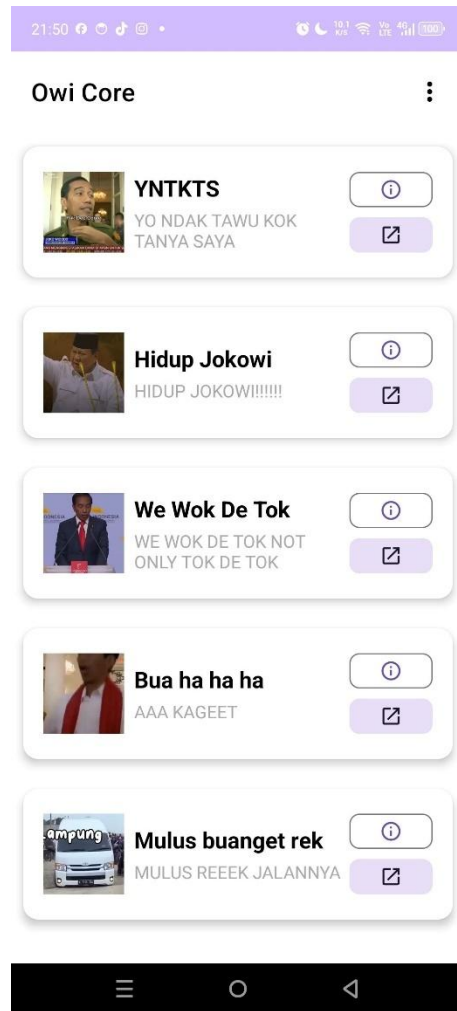
1	<?xml version="1.0" encoding="utf-8"?>
2	<navigation
3	xmlns:android="http://schemas.android.com/apk/res/android"
4	xmlns:app="http://schemas.android.com/apk/res-auto"
5	android:id="@+id/nav_graph"
6	app:startDestination="@id/myFragment">
7	
8	<fragment
9	android:id="@+id/myFragment"
10	android:name="com.example.myrecylerview.MyFragment"
11	android:label="MyFragment" >
12	<action
13	
14	android:id="@+id/action_myFragment_to_detailFragment"



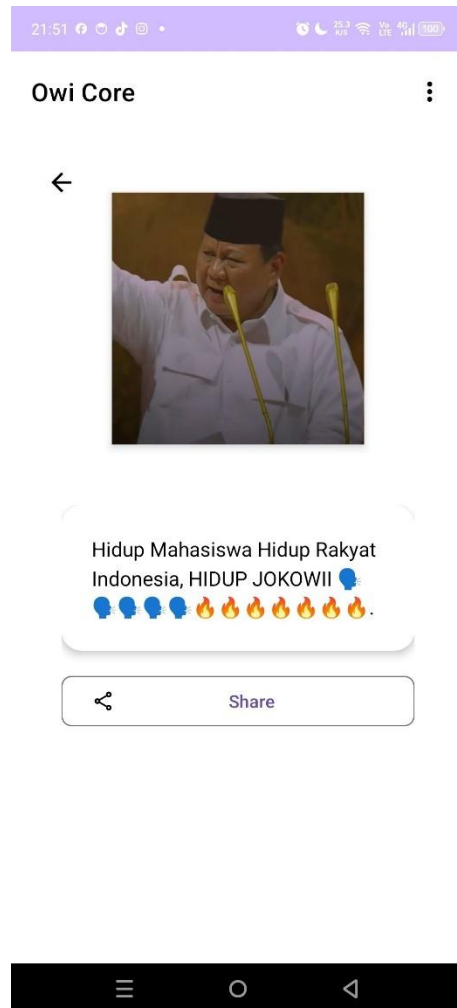
15	app:destination="@id/detailFragment" />
16	</fragment>
17	
18	<fragment
19	android:id="@+id/detailFragment"
20	
21	android:name="com.example.myrecylerview.DetailFragment"
22	android:label="Detail">
23	
24	<argument
25	android:name="extraPhoto"
26	app:argType="integer" />
27	
28	<argument
29	android:name="extraLink"
30	app:argType="string" />
31	
32	<argument
33	android:name="extraDetail"
34	app:argType="string" />
35	</fragment>
36	
37	</navigation>

*Tabel 10 Source Code Jawaban Soal 1*

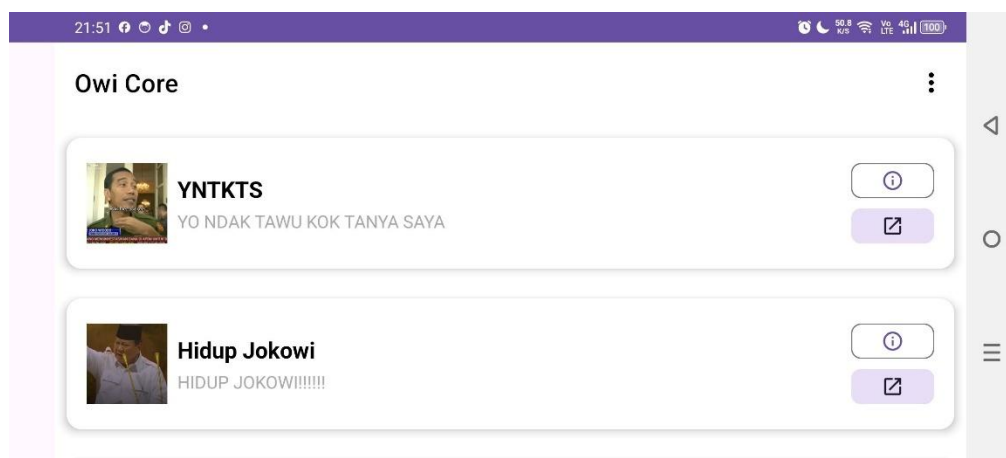
## B. Output Program



Gambar 3 Screenshot Hasil Jawaban Soal 1



Gambar 4 Screenshot Hasil Jawaban Soal 1



Gambar 5 Screenshot Hasil Jawaban Soal 1

## **C. Pembahasan**

### **1. MainActivity.kt**

Pada modul kali ini MainActivity hanya digunakan sebagai wadah untuk fragment-fragment yang ada, sehingga kode pada MainActivity berfokus hanya pada instalasi splash screen, dan juga karena start destination sudah di set pada MyFragment, jadi secara keseluruhan MainActivity ada hanya sebagai pelengkap dari android manifest itu sendiri.

### **2. DetailFragment.kt**

Overall kegunaan fragment ini hanya sebagai penunjuk detail saja, pada filenya logic yang digunakan hanya menerima data yang di passing oleh adapter seperti pada line [28]-[31], dan melakukan pergantian image atau ui dari fragment\_detail seperti pada line [33]-[38] saja, dan juga beberapa button untuk share, dan kembali pada line [40]-[50].

### **3. MyAdapter.kt**

Oke, Logic dari RecyclerView dimulai pada file ini, dari baris [9] hingga [12], di mana terdapat inisialisasi beberapa parameter, yaitu listCharacter, onYTClick, dan onDetailClick. Parameter listCharacter merupakan data utama yang akan ditampilkan dalam RecyclerView. Parameter onYTClick dan onDetailClick adalah fungsi lambda yang akan dieksekusi ketika tombol tertentu ditekan.

Selanjutnya, pada baris [14] hingga [22], terdapat struktur utama dari RecyclerView Adapter. Pada bagian ini, kelas ViewHolder digunakan untuk mendefinisikan bahwa setiap elemen yang ditampilkan merupakan bagian dari RecyclerView. Kelas ini menerima parameter berupa ItemLayoutBinding, yang menghubungkan layout XML dengan kode Kotlin.

Fungsi onCreateViewHolder digunakan untuk membuat tampilan item menggunakan LayoutInflater, dengan menghubungkannya ke ItemLayoutBinding.

Ini berarti adapter akan membuat tampilan sesuai template yang sudah didefinisikan di file XML.

Fungsi `getItemCount` mengembalikan jumlah item yang ada dalam `listCharacter`, yang berarti menyesuaikan jumlah data yang akan ditampilkan dalam daftar.

Pada bagian akhir, yaitu baris [23] sampai selesai, terdapat fungsi `onBindViewHolder`. Fungsi ini bertanggung jawab untuk binding data ke dalam tampilan. Artinya, setiap elemen seperti `textTitle`, `textDescription`, dan `itemImage` akan diisi dengan data dari objek yang sesuai di dalam `listCharacter`. Selain itu, fungsi ini juga menetapkan aksi ketika tombol `btnWebsite` dan `btnDetails` ditekan, dengan menjalankan fungsi `onYTClick` dan `onDetailClick` yang sudah didefinisikan sebelumnya.

#### **4. MyFragment.kt**

Oke, pada file ini, `MyFragment` merupakan fragment pertama yang akan dijumpai saat membuka aplikasi, karena sudah diatur di `nav_graph`. Pada baris 17 hingga 21 merupakan template (sering disebut boilerplate) untuk inisialisasi binding dan adapter.

Selanjutnya, pada baris 23 sampai 32 terdapat fungsi `onCreateView`. Pertama, fungsi ini akan melakukan clear terhadap list yang sudah ada. Setelah itu, data baru akan dimasukkan ke dalam `listCharacter`, dan kemudian akan memanggil fungsi `setupRecyclerView`.

Pada baris 34 sampai 52 terdapat logika dari tombol-tombol yang ada. Pada `onYTClick`, jika tombol ditekan, maka akan dilakukan navigasi ke link yang telah disediakan menggunakan `Intent`. Sementara itu, pada `onDetailClick`, data akan dipassing ke `DetailFragment` menggunakan `MyFragmentDirections`.

Kemudian, terdapat `binding.rvCharacter` yang digunakan untuk menerapkan `RecyclerView`, dengan menggunakan `LinearLayoutManager` dan menetapkan adapter.

Pada baris 55 hingga 74, terdapat fungsi `getListCharacter` yang digunakan untuk menjalankan `RecyclerView` itu sendiri. Fungsi ini mengambil data dari `resources`, seperti array nama, link, subtext, photo, dan detail, lalu dimasukkan ke dalam `ArrayList` bertipe `MyData`. Setelah selesai, `TypedArray` akan di-recycle.

Terakhir, pada fungsi `onDestroyView`, dilakukan pembersihan terhadap objek binding agar tidak terjadi memory leak.

## 5. `MyData.kt`

Wadah untuk melakukan pendefinisian tipe variable yang digunakan, serta penyimpanan data itu sendiri

## 6. `activity_main.xml`

File ini merupakan layout utama yang digunakan oleh `MainActivity`. Layout ini menggunakan `ConstraintLayout` sebagai elemen root, yang memungkinkan setiap elemen diatur secara fleksibel berdasarkan hubungan antar elemen atau terhadap parent-nya.

Pertama, terdapat `MaterialToolbar` dari `com.google.android.material.appbar.MaterialToolbar`, yang berfungsi sebagai app bar atau toolbar utama. Toolbar ini memiliki lebar `match_parent` dan tinggi berdasarkan atribut `?attr/actionBarSize`. Latar belakangnya berwarna putih dan diberi elevation sebesar 4dp agar terlihat bayangan (shadow) pada perangkat dengan material design. Judul dari toolbar ini adalah "Owi Core", dengan warna teks hitam dan gaya teks sesuai dengan `TextAppearance.MaterialComponents.Headline6`.

Di dalam `MaterialToolbar`, terdapat satu buah `ImageView` dengan id `btn_menu`. Elemen ini memiliki ukuran lebar dan tinggi 24dp, dengan posisi gravitasi ke end (kanan), serta diberi margin end sebesar 16dp. Ikon yang digunakan diambil dari `drawable/ic_more_vert`, dan diberi warna menggunakan atribut tint hitam.

Setelah itu, terdapat `FragmentContainerView` dari `androidx.fragment.app.FragmentContainerView`, yang digunakan sebagai wadah untuk menampilkan fragment yang aktif. Elemen ini memiliki id `nav_host_fragment`

dan menggunakan NavHostFragment sebagai kelas utama yang menjalankan sistem navigasi. Ukuran lebar dan tingginya diatur menggunakan 0dp karena akan disesuaikan oleh constraint. Komponen ini berada di bawah toolbar, dan dikaitkan ke semua sisi parent untuk memenuhi seluruh layar yang tersisa. Properti `app:navGraph` menunjukkan `nav_graph` yang digunakan untuk mengatur alur navigasi antar fragment, dan `app:defaultNavHost` disetel ke `true` agar fragment ini bertindak sebagai host utama navigasi.

## 7. `fragment_detail.xml`

Layout ini menggunakan `NestedScrollView` dari `androidx.core.widget.NestedScrollView` sebagai elemen utama. Elemen ini memungkinkan konten yang lebih panjang dari layar untuk digulir secara vertikal. Properti `fillViewport` disetel ke `true` agar isi konten mengisi seluruh area tampilan saat belum bisa digulir.

Di dalamnya terdapat `ConstraintLayout` yang berfungsi untuk mengatur posisi setiap elemen UI secara fleksibel. Layout ini memiliki padding sebesar 16dp dan tinggi `wrap_content`, sehingga akan menyesuaikan dengan isi yang dimuat.

Elemen pertama adalah `ImageButton` dengan id `btn_back`. Tombol ini berfungsi sebagai tombol kembali. Lebar dan tingginya 48dp, tidak memiliki latar belakang tetap (`borderless`) karena menggunakan `?selectableItemBackgroundBorderless`, dan menggunakan ikon dari `drawable/ic_arrow_back`. Tombol ini diletakkan di kiri atas dengan menggunakan `layout_constraintStart_toStartOf` dan `layout_constraintTop_toTopOf` parent. Warna ikonnya diatur menggunakan `app:tint` hitam.

Berikutnya adalah `ShapeableImageView` dari `com.google.android.material.imageview.ShapeableImageView` yang berfungsi untuk menampilkan gambar utama atau gambar detail. Elemen ini memiliki ukuran 200dp x 200dp, dengan `scaleType` `centerCrop` agar gambar terpotong sesuai proporsi

tampilan. Gambar ini ditempatkan di tengah horizontal dan di bagian atas layout, dengan margin atas sebesar 32dp dan bayangan sebesar 4dp melalui elevation. Selain itu, elemen ini diberi properti `transitionName` yaitu `shared_image_container` untuk mendukung animasi transisi antar fragment.

Setelah itu terdapat `LinearLayout` yang orientasinya vertikal dan posisinya berada di bawah `detail_image`. Elemen ini memiliki padding sebesar 24dp dan margin atas 24dp. Di dalamnya terdapat dua elemen utama: `CardView` dan `LinearLayout` horizontal.

Pertama, `CardView` dari `androidx.cardview.widget.CardView` digunakan untuk menampilkan deskripsi dalam bentuk `TextView`. Kartu ini memiliki sudut membulat sebesar 16dp, bayangan sebesar 4dp, dan latar belakang putih. Di dalamnya, `TextView` dengan id `detail_description` menampilkan teks deskripsi dengan ukuran huruf 16sp, jarak antar baris sebesar 1.2, warna teks hitam, dan padding 24dp di semua sisi.

Kedua, terdapat `LinearLayout` horizontal yang digunakan untuk menempatkan tombol di bagian bawah konten. Elemen ini memiliki orientasi horizontal dan properti gravity disetel ke `center_horizontal` untuk memusatkan isinya. Properti spacing diatur sebesar 16dp, walaupun atribut ini tidak berlaku langsung pada `LinearLayout` standar dan biasanya perlu ditangani melalui margin antar elemen secara manual atau menggunakan spacing dari library tambahan.

Di dalamnya terdapat `MaterialButton` dari `com.google.android.material.button.MaterialButton` dengan id `btn_share`. Tombol ini menggunakan gaya `Widget.Material3.Button.OutlinedButton`, memiliki lebar `match_parent`, tinggi 48dp, dan menampilkan teks "Share" serta ikon dari `drawable/ic_share`. Ikon tidak diberi warna tambahan karena `iconTint` disetel ke null. Sudut tombol dibulatkan dengan `cornerRadius` sebesar 8dp.



Secara keseluruhan, layout ini dirancang untuk menampilkan halaman detail dengan gambar utama di atas, deskripsi di tengah, dan tombol aksi di bagian bawah, semuanya dalam struktur yang dapat digulir secara vertikal.

## **8. fragment\_my.xml**

File layout ini menggunakan `ConstraintLayout` dari `androidx.constraintlayout.widget.ConstraintLayout` sebagai elemen utama. Layout ini memiliki lebar dan tinggi `match_parent`, artinya akan mengisi seluruh ruang dari layar. Properti `tools:context` diatur ke `.MyFragment`, yang memberi petunjuk kepada Android Studio bahwa layout ini digunakan oleh kelas `MyFragment`. Atribut ini hanya berfungsi saat preview, tidak memengaruhi saat runtime.

Di dalamnya hanya terdapat satu komponen utama yaitu `RecyclerView` dari `androidx.recyclerview.widget.RecyclerView` dengan id `rv_character`. Komponen ini bertugas untuk menampilkan daftar data dalam bentuk list yang bisa digulir. Ukuran lebar dan tingginya diatur ke `0dp` karena akan diatur menggunakan `constraint`. Properti `layout_constraintTop_toTopOf`, `layout_constraintBottom_toBottomOf`, `layout_constraintStart_toStartOf`, dan `layout_constraintEnd_toEndOf` digunakan untuk mengaitkan semua sisi `RecyclerView` ke parent-nya, yaitu `ConstraintLayout`, sehingga elemen ini akan memenuhi seluruh ruang yang tersedia.

Layout ini bersifat minimal dan hanya bertugas menyediakan wadah untuk menampilkan daftar item dalam fragment. Semua logika pemrosesan data dan penanganan klik dilakukan di kelas `MyFragment`.

## **9. Item\_layout.xml**

Layout ini menggunakan `CardView` dari `androidx.cardview.widget.CardView` sebagai elemen utama, yang berfungsi untuk membungkus setiap item dalam daftar agar memiliki tampilan seperti kartu. Atribut `layout_width` diatur ke `match_parent` dan `layout_height` ke `wrap_content`, artinya

kartu akan mengisi lebar penuh tapi tingginya akan menyesuaikan kontennya. Properti `cardCornerRadius` diatur ke 12dp agar memiliki sudut melengkung, `cardElevation` sebesar 4dp untuk memberi efek bayangan, dan `cardUseCompatPadding` digunakan agar padding kompatibel dengan perangkat lama.

Di dalam `CardView` terdapat `ConstraintLayout`, yang bertugas untuk mengatur posisi elemen-elemen di dalam kartu secara fleksibel. Layout ini memiliki padding 16dp agar konten tidak menempel langsung ke tepi kartu.

Elemen pertama adalah `ImageView` dengan id `item_image`. Gambar ini memiliki ukuran tetap 64dp x 64dp, dengan `scaleType` diatur ke `centerCrop` agar gambar memenuhi area tanpa merusak rasio. Gambar ini diposisikan secara vertikal di tengah dan di sisi kiri kartu menggunakan constraint ke parent.

Berikutnya adalah `LinearLayout` yang menampung dua `TextView`, yaitu `text_title` dan `text_description`. Layout ini diletakkan di sebelah kanan gambar menggunakan constraint ke id `item_image`, dan di sebelah kiri tombol-tombol menggunakan constraint ke `button_group`. Layout ini disusun secara vertikal.

`text_title` berfungsi menampilkan judul item, memiliki ukuran teks 18sp, maksimal 1 baris dan akan terpotong dengan `ellipsize` jika terlalu panjang. `text_description` menampilkan deskripsi tambahan dengan ukuran teks 14sp, maksimal 2 baris.

Terakhir, terdapat `LinearLayout` lain dengan id `button_group` yang menampung dua tombol dari `MaterialButton`. Layout ini disusun secara vertikal, dan diletakkan di paling kanan kartu. Tombol pertama adalah `btn_details`, menggunakan style `OutlinedButton` dan menampilkan ikon `ic_info_outline`. Tombol kedua adalah `btn_website`, menggunakan style `TonalButton` dengan ikon `ic_open_in_browser`. Kedua tombol ini memiliki tinggi tetap 36dp dan lebar minimum 64dp. Ikon berada di sebelah kiri teks tombol sesuai dengan `iconGravity` yang diatur ke `textStart`.

Secara keseluruhan, layout ini dirancang untuk menampilkan item dalam bentuk kartu yang berisi gambar, judul, deskripsi, dan dua tombol aksi.

## 10. nav\_graph.xml

file navigation graph yang digunakan untuk mengatur alur navigasi antar fragment dalam aplikasi. Root tag-nya adalah navigation dari `androidx.navigation`, dan berfungsi sebagai container dari seluruh rute navigasi. Atribut `app:startDestination` diatur ke `@id/myFragment`, artinya aplikasi akan membuka `myFragment` sebagai tampilan awal.

Di dalamnya terdapat dua elemen fragment.

Fragment pertama memiliki id `@+id/myFragment`, dengan atribut `android:name` menunjuk ke `com.example.myrecyclerview.MyFragment`, yaitu kelas yang akan digunakan saat rute ini dipanggil. Label-nya diatur menjadi `MyFragment`, dan di dalamnya terdapat elemen action.

Elemen action berfungsi untuk mendeklarasikan arah navigasi dari satu fragment ke fragment lain. Dalam hal ini, `action_myFragment_to_detailFragment` menunjuk ke `@id/detailFragment`, yang berarti ketika aksi ini dipanggil, aplikasi akan berpindah dari `MyFragment` ke `DetailFragment`.

Fragment kedua adalah `detailFragment`, dengan atribut `android:name` menunjuk ke `com.example.myrecyclerview.DetailFragment`. Label-nya diatur menjadi `Detail`.

Di dalam `detailFragment`, terdapat tiga elemen argument. Elemen ini digunakan untuk mendeklarasikan argumen yang perlu dikirimkan saat navigasi ke `detailFragment` dilakukan:

- `extraPhoto`, dengan tipe integer
- `extraLink`, dengan tipe string
- `extraDetail`, dengan tipe string

Argumen-argumen ini digunakan agar data dapat dipassing dari `MyFragment` ke `DetailFragment`, seperti gambar, deskripsi detail, dan link eksternal.

## **SOAL 2**

Menurut saya kenapa masih banyak yang memakai RecyclerView meski kode-nya lebih panjang karena alasan praktis dan teknis. Pertama, backward compatibility dan banyak aplikasi besar masih mengandalkan RecyclerView (XML) untuk daftar data besar karena migrasi penuh ke Compose sangat kompleks. Selain itu, RecyclerView menyediakan kontrol tingkat rendah yang lebih besar (misal LayoutManager kustom, animasi item kompleks, dan tipe tampilan item beragam) yang belum sepenuhnya didukung oleh LazyColumn. developer juga sering melakukan adopsi bertahap atau hybrid: Compose dirancang interoperabel dengan UI lama, sehingga bagian aplikasi yang kompleks tetap menggunakan RecyclerView sambil menulis fitur baru dengan LazyColumn. Terakhir, RecyclerView sudah sangat matang (banyak library pihak ketiga dan komunitas luas), sehingga komponen ini tetap menjadi pilihan andal di aplikasi mapan

## **Tautan Git**

<https://github.com/au290/Kuliah/tree/main/Pemro%20Mobile/Modul%203>