

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



BUILD A SCROLLABLE LIST

Oleh:

Alysa Armelia

NIM. 2310817120009

**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 : Alysa Armelia
NIM : 2310817120009

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	23
C. Pembahasan	25
SOAL 2.....	32
D. Tautan Git.....	33

DAFTAR GAMBAR

Gambar 1 Contoh UI List	7
Gambar 2 Gambar UI Detail	7
Gambar 3 Screenshot Hasil Jawaban Soal 1	23
Gambar 4 Screenshot Hasil Jawaban Soal 1	23
Gambar 5 Screenshot tombol Detail.....	24
Gambar 6 Screenshot tombol Info.....	24

DAFTAR TABEL

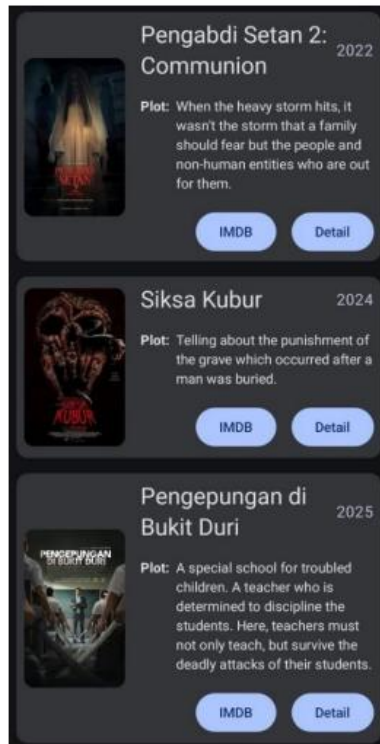
Tabel 1. 1 Source Code Jawaban Soal 1	8
Tabel 1. 2 Source Code Jawaban Soal 1	9
Tabel 1. 3 Source Code Jawaban Soal 1	16
Tabel 1. 4 Source Code Jawaban Soal 1	17
Tabel 1. 5 Source Code Jawaban Soal 1	17
Tabel 1. 6 Source Code Jawaban Soal 1	18
Tabel 1. 7 Source Code Jawaban Soal 1	19
Tabel 1. 8 Source Code Jawaban Soal 1	19
Tabel 1. 9 Source Code Jawaban Soal 1	21
Tabel 1. 10 Source Code Jawaban Soal 1	22

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

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



Gambar 1 Contoh UI List

Desain UI laman detail bebas, tetapi diusahakan untuk mengikuti kaidah desain Material Design dan data item ditampilkan penuh di laman detail seperti contoh berikut:



Gambar 2 Gambar UI Detail

A. Source Code

1. MainActivity.kt

```
1 package com.example.londondestination
2
3 import android.os.Bundle
4 import androidx.appcompat.app.AppCompatActivity
5 import
6 com.example.londondestination.databinding.ActivityMainBinding
7
8 class MainActivity : AppCompatActivity() {
9     private lateinit var binding: ActivityMainBinding
10    override fun onCreate(savedInstanceState: Bundle?) {
11
12        super.onCreate(savedInstanceState)
13        binding = ActivityMainBinding.inflate(layoutInflater)
14        setContentView(binding.root)
15    }
16 }
```

Tabel 1. 1 Source Code Jawaban Soal 1

2. FragmentGuweh.kt

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



```

23
24     override fun onCreateView(view: View,
25 savedInstanceState: Bundle?) {
26         super.onCreateView(view, savedInstanceState)
27         val imageResId = arguments?.getInt("imageResId") ?:
28 R.drawable.ic_launcher_background
29         val nama = arguments?.getString("nama") ?: "Nama
30 tidak tersedia"
31         val deskripsi = arguments?.getString("deskripsi") ?:
32 "Deskripsi tidak tersedia"
33
34         binding.detailImage.setImageResource(imageResId)
35         binding.detailTitle.text = nama
36         binding.detailDescription.text = deskripsi
37     }
38
39     override fun onDestroyView() {
40         super.onDestroyView()
41         _binding = null
42     }
43
44     companion object {
45         fun newInstance(imageResId: Int, nama: String,
46 deskripsi: String): FragmentGuweh {
47             val fragment = FragmentGuweh()
48             val args = Bundle()
49             args.putInt("imageResId", imageResId)
50             args.putString("nama", nama)
51             args.putString("deskripsi", deskripsi)
52             fragment.arguments = args
53             return fragment
54         }
55     }
56 }

```

Tabel 1. 2 Source Code Jawaban Soal 1

3. HomeFragment.kt

```

1     package com.example.londondestination
2
3     import android.os.Bundle
4     import android.view.LayoutInflater
5     import android.view.View
6     import android.view.ViewGroup
7     import androidx.fragment.app.Fragment
8     import androidx.recyclerview.widget.LinearLayoutManager
9     import androidx.navigation.fragment.findNavController
10    import androidx.core.os.bundleOf
11    import

```

	com.example.londondestination.databinding.HomeFragmentBinding
12	
13	class HomeFragment : Fragment() {
14	
15	private var _binding: HomeFragmentBinding? = null
16	private val binding get() = _binding!!
17	
18	private lateinit var adapter: MyAdapter
19	private val myDataList = listOf(
20	MyData(
21	nama = "Natural History Museum",
22	description = "Natural History Museum di London
	adalah destinasi wajib bagi pecinta sains, sejarah alam,
	dan keluarga dengan anak-anak. Museum ini memiliki lebih
	dari 80 juta spesimen, mencakup zoologi, paleontologi,
	botani, dan geologi. Daya tarik utama termasuk rangka
	dinosaurius, seperti Diplodocus dan model animatronik
	Tyrannosaurus Rex. Selain itu, pengunjung bisa melihat
	koleksi batuan langka, meteorit, serta kristal dan
	permata.\n" +
23	"\n" + "Di Earth Galleries, ada pameran
	interaktif tentang geologi bumi, gunung berapi, dan gempa
	bumi. Museum ini juga memiliki zona edukasi yang menarik
	untuk anak-anak. Dibangun dengan arsitektur Romanesque
	bergaya Victoria, bangunannya memukau dengan interior yang
	megah. Yang menarik, museum ini gratis untuk dikunjungi,
	memberikan pengalaman yang tak terlupakan bagi semua
	pengunjung.",
24	descriptionsingkat = "Museum sejarah tentang
	alam",
25	year = 1881,
26	image = R.drawable.naturalhistorymuseum,
27	link =
	"https://en.wikipedia.org/wiki/Natural_History_Museum,_Lond
	on"
28),
29	MyData(
30	nama = "London Eye",
31	description = "London Eye adalah kincir raksasa
	setinggi 135 meter di tepi Sungai Thames, dan menjadi salah
	satu ikon paling terkenal di London. Dari dalam kapsul
	kacanya, pengunjung bisa menikmati panorama kota yang
	menakjubkan, termasuk pemandangan Big Ben, Gedung Parlemen,
	dan Sungai Thames. Pada hari cerah, jarak pandang bisa
	mencapai hampir 40 kilometer.\n" +
32	"\n" + "Waktu terbaik untuk naik adalah
	saat senja, ketika cahaya kota mulai menyala dan langit
	berwarna keemasan karena menciptakan suasana romantis dan
	tenang. Setiap putaran berlangsung sekitar 30 menit,
	memberikan cukup waktu untuk mengagumi pemandangan,
	mengambil foto, atau sekadar menikmati momen dari

	ketinggian. London Eye bukan hanya atraksi wisata, tetapi juga simbol kebanggaan kota yang terus memikat baik turis maupun warga lokal.",
33	descriptionsingkat = "Komedi putar raksasa London",
34	year = 2000,
35	image = R.drawable.londoneye,
36	link =
	"https://en.wikipedia.org/wiki/London_Eye"
37),
38	MyData(
39	nama = "British Museum",
40	description = "British Museum adalah destinasi luar biasa bagi pecinta sejarah dan arkeologi, dengan koleksi global yang mencakup ribuan tahun peradaban manusia. Begitu masuk, suasana elegan langsung membawa pengunjung seakan menjelajahi masa lalu dari mumi Mesir, patung Yunani, artefak Mesopotamia, hingga keramik Tiongkok dan seni Islam. Batu Rosetta adalah salah satu sorotan utama, bersama Patung Ramses II dan reruntuhan Kuil Parthenon yang ikonik.\n" +
41	"\n" + "Setiap koleksi disertai penjelasan informatif yang memudahkan pengunjung memahami konteks sejarahnya. Great Court, dengan atap kaca yang terang dan desain modern, menjadi pusat bangunan yang menawan dan nyaman untuk bersantai. British Museum bukan sekadar tempat melihat artefak, tapi juga ruang kontemplatif yang menghubungkan kita dengan jejak panjang umat manusia dan semuanya bisa dinikmati tanpa biaya masuk.",
42	descriptionsingkat = "Museum koleksi dunia",
43	year = 1753,
44	image = R.drawable.british_museum,
45	link =
	"https://en.wikipedia.org/wiki/British_Museum"
46),
47	MyData(
48	nama = "Big Ben",
49	description = "Big Ben adalah ikon tak tergantikan London, berdiri megah di samping Gedung Parlemen di tepi Sungai Thames. Meski banyak mengira namanya merujuk pada menaranya, Big Ben sebenarnya adalah lonceng besar seberat lebih dari 13 ton di dalam Elizabeth Tower adalah nama resmi menara tersebut, yang diberikan untuk menghormati Ratu Elizabeth II.\n" +
50	"\n" + "Dentang Big Ben punya makna emosional yang dalam, sering terdengar dalam momen penting seperti pergantian tahun atau peringatan nasional, dan bahkan disiarkan BBC sejak 1920-an. Arsitekturnya yang anggun menjadi latar favorit para wisatawan, baik saat disinari mentari pagi maupun diterangi lampu malam hari. Walau akses ke dalam menara terbatas, cukup berdiri di

	dekatnya sudah membuat pengunjung merasa terhubung dengan sejarah dan semangat kota London yang tak lekang waktu",
51	descriptionsingkat = "Jam besar London",
52	year = 1859,
53	image = R.drawable.bigben,
54	link = "https://en.wikipedia.org/wiki/Big_Ben"
55),
56	MyData(
57	nama = "Buckingham Palace",
58	description = "Buckingham Palace adalah simbol monarki Inggris yang berdiri megah di pusat London, berfungsi sebagai kediaman resmi Raja dan pusat berbagai acara kenegaraan. Dengan lebih dari 700 ruangan, istana ini mencerminkan kemewahan dan sejarah yang hidup, bahkan dari luar pagar hitamnya yang ikonik.\n" +
59	"\n" + "Salah satu atraksi utama adalah Upacara Pergantian Penjaga, prosesi tradisional dengan seragam merah dan musik marching band yang menarik ribuan wisatawan setiap harinya. Jika bendera kerajaan berkibar di atas istana, itu menandakan Raja sedang berada di dalam momen sederhana yang membuat banyak orang merasa lebih dekat dengan sejarah kerajaan.\n" +
60	"\n" + "Pada musim panas, beberapa ruang kenegaraan dibuka untuk umum, menampilkan interior menawan lengkap dengan kristal, lukisan klasik, dan kemegahan khas kerajaan. Meski banyak pengunjung hanya melihat dari luar, pesona dan wibawa istana ini menjadikannya salah satu destinasi paling ikonik di London.",
61	descriptionsingkat = "Istana resmi Kerajaan Inggris",
62	year = 1703,
63	image = R.drawable.buckinghampalace,
64	link = "https://en.wikipedia.org/wiki/Buckingham_Palace"
65),
66	MyData(
67	nama = "Tower of London",
68	description = "Tower of London adalah benteng bersejarah di tepi Sungai Thames yang menyimpan kisah dramatis tentang kekuasaan, pengkhianatan, dan warisan kerajaan Inggris. Dulu berfungsi sebagai penjara bagi bangsawan, termasuk Anne Boleyn yang dieksekusi di sana, tempat ini memancarkan nuansa mencekam sekaligus megah, terutama di lokasi-lokasi penting seperti halaman eksekusi.\n" +
69	"\n" + "Namun, sisi gelap itu berpadu dengan kemewahan karena di sinilah Permata Mahkota Inggris disimpan, termasuk mahkota dan tongkat kerajaan yang berkilau menakjubkan. Kontras antara sejarah kelam dan simbol kejayaan membuat kunjungan ke Tower of London terasa seperti menyusuri lorong waktu, menghadirkan pengalaman

70	mendalam yang tak terlupakan.", descriptionsingkat = "Benteng mempunyai banyak sejarah",
71	year = 1066,
72	image = R.drawable.toweroflondon,
73	link = "https://en.wikipedia.org/wiki/Tower_of_London"
74),
75	MyData(76 nama = "Warner Bros. Studio Tour London", 77 description = "Warner Bros. Studio Tour London adalah destinasi impian bagi penggemar Harry Potter, menawarkan pengalaman imersif ke dunia sihir yang sebelumnya hanya bisa dilihat di layar. Begitu masuk, kamu akan dibawa ke set asli seperti Great Hall, Diagon Alley, dan Privet Drive dimana semuanya penuh detail yang membuatmu merasa benar-benar berada di dunia Hogwarts.\n" + 78 "\n" + "Selain menjelajahi lokasi ikonik, pengunjung juga bisa melihat properti film seperti Horcrux, kostum rumah-rumah Hogwarts, hingga proses pembuatan efek visual yang menghidupkan sihir di layar. Setiap sudut studio menyuguhkan keajaiban yang membuat kamu makin menghargai imajinasi dan kerja keras di balik film. Dan tentu saja, mencicipi Butterbeer jadi penutup manis dari kunjungan yang terasa seperti pulang ke dunia masa kecil yang penuh keajaiban.", 79 descriptionsingkat = "Studio Harry Potter London",
80	year = 2012,
81	image = R.drawable.harrypotterstudio,
82	link = "https://www.wbstudiotour.co.uk/"
83),
84	MyData(85 nama = "Hyde Park", 86 description = "Hyde Park adalah oase hijau di tengah London yang menawarkan ketenangan dan ruang bebas bagi siapa saja, dari warga lokal hingga turis. Dengan luas lebih dari 140 hektar, taman ini menjadi tempat ideal untuk jogging, bersepeda, membaca buku, atau sekadar duduk santai di tepi danau Serpentine.\n" + 87 "\n" + "Suasannya santai dan cocok buat piknik, bermain, atau menikmati kopi di bawah rindangnya pepohonan. Salah satu sudut paling unik adalah Speaker's Corner, simbol kebebasan berpendapat di mana siapa pun bisa berbicara di depan umum. Selain sebagai tempat pelarian dari hiruk pikuk kota, Hyde Park juga kerap menjadi lokasi konser besar dan festival, menjadikannya ruang publik yang dinamis dan menyatu dengan jiwa kota London.", 88 descriptionsingkat = "Taman pusat kota", 89 year = 1637, 90 image = R.drawable.hydepark,

91	link =
92	"https://en.wikipedia.org/wiki/Hyde_Park,_London"
93),
94	MyData(
95	nama = "The Sherlock Holmes Museum",
	description = "The Sherlock Holmes Museum di
	221B Baker Street adalah surga bagi penggemar detektif
	legendaris ini, membawa pengunjung langsung ke dunia fiksi
	era Victoria yang terasa hidup. Interiornya merekonstruksi
	ruang kerja dan rumah Holmes secara detail–lengkap dengan
	perapian, kaca pembesar, dan barang-barang pribadi khas
	karakter dalam cerita.\n" +
96	"\n" + "Setiap sudut museum dirancang
	agar kamu seolah benar-benar berada di tengah kisah misteri
	bersama Holmes dan Watson. Lebih dari sekadar pameran,
	museum ini menawarkan pengalaman yang memuaskan rasa ingin
	tahu para penggemar dan pencinta cerita klasik.",
97	descriptionsingkat = "Museum seorang detektif
	yang ikonik",
98	year = 1990,
99	image = R.drawable.thesherlockholmesmuseum,
100	link = "https://www.sherlock-holmes.co.uk/"
101),
102	MyData(
103	nama = "St Paul's Cathedral",
104	description = "St Paul's Cathedral adalah salah
	satu ikon arsitektur paling menakjubkan di London, dikenal
	dengan kubah raksasanya yang mendominasi cakrawala kota.
	Dari luar terlihat megah, tapi keindahan sejatinya baru
	benar-benar terasa saat kamu melangkah masuk–ruang dalamnya
	hening, agung, dan sarat nuansa spiritual. Langit-langit
	tinggi, kaca patri indah, dan cahaya alami menciptakan
	atmosfer yang membuat siapa pun terdiam dalam kekaguman.\n"
	+
105	"\n" + "Salah satu pengalaman paling
	tak terlupakan di sini adalah menaiki ratusan anak tangga
	menuju puncak kubah. Dari atas, kamu bisa menikmati
	panorama kota London yang luas dan penuh sejarah, dari
	Sungai Thames hingga gedung-gedung modern yang berdiri
	berdampingan dengan bangunan bersejarah. Tak heran, St
	Paul's sering menjadi lokasi berbagai momen penting
	nasional, karena tempat ini bukan hanya gereja, tapi simbol
	kekuatan dan keindahan yang hidup modern di kota.",
106	descriptionsingkat = "Katedral bersejarah di
	London",
107	year = 1710,
108	image = R.drawable.stpaulcathedral,
109	link =
	"https://en.wikipedia.org/wiki/St_Paul%27s_Cathedral"
110),
111	MyData(
112	nama = "Camden Market",

```

113         description = "Camden Market adalah salah satu
tempat paling unik dan penuh karakter di London, ideal bagi
kamu yang suka berburu barang-barang tak biasa dan
merasakan suasana kota yang dinamis. Pasar ini dipenuhi
114         toko-toko kecil yang menjual pakaian vintage, aksesor
handmade, dan berbagai barang nyentrik yang sering kali
hanya ada satu di dunia. Suasananya ramai tapi seru, dengan
musik jalanan, aroma makanan internasional, dan gaya busana
pengunjung yang beragam.\n" +
"\n" + "Selain jadi pusat belanja,
Camden juga merupakan ruang ekspresi subkultur alternatif
seperti punk, goth, dan hippie-tempat di mana semua orang
bisa tampil sesuai dirinya sendiri. Kalau lapar, pilihan
kulinernya luar biasa banyak dan menggoda, dari makanan
Asia, Latin, Timur Tengah, sampai fusion kreatif. Setiap
kunjungan ke Camden terasa seperti eksplorasi baru bukan
sekadar belanja, tapi pengalaman hidup kota London yang
bebas, kreatif, dan penuh kejutan.",
115         descriptionsingkat = "Pasar terbesar di
London",
116         year = 1974,
117         image = R.drawable.camdenmarket,
118         link =
"https://en.wikipedia.org/wiki/Camden_Market"
119     )
120 )
121
122     override fun onCreateView(inflater: LayoutInflater,
container: ViewGroup?,
123         savedInstanceState: Bundle?
124     ): View {
125         _binding = HomeFragmentBinding.inflate(inflater,
container, false)
126         return binding.root
127     }
128
129     override fun onViewCreated(view: View,
savedInstanceState: Bundle?) {
130         super.onViewCreated(view, savedInstanceState)
131
132         adapter = MyAdapter(myDataList) { selectedItem ->
133             val bundle = bundleOf(
134                 "imageResId" to selectedItem.image,
135                 "nama" to selectedItem.nama,
136                 "deskripsi" to selectedItem.description
137             )
138         findNavController().navigate(R.id.action_HomeFragment_to_de
tailFragment, bundle)
139     }
140
141     binding.rvCharacter.layoutManager =
142     LinearLayoutManager(requireContext())

```

	binding.rvCharacter.adapter = adapter
143	}
144	
145	override fun onDestroyView() {
146	super.onDestroyView()
147	_binding = null
148	}
149	}

Tabel 1.3 Source Code Jawaban Soal 1

4. MyAdapter.kt

1	package com.example.londondestination
2	
3	import android.content.Intent
4	import android.net.Uri
5	import android.view.LayoutInflater
6	import android.view.ViewGroup
7	import androidx.recyclerview.widget.RecyclerView
8	import
9	com.example.londondestination.databinding.ItemListBinding
10	class MyAdapter(
11	private val destinations: List<MyData>,
12	private val onDetailClick: (MyData) -> Unit
13) : RecyclerView.Adapter<MyAdapter.ViewHolder>() {
14	
15	inner class ViewHolder(val binding: ItemListBinding) :
16	RecyclerView.ViewHolder(binding.root)
17	
18	override fun onCreateViewHolder(parent: ViewGroup,
19	viewType: Int): ViewHolder {
20	val binding = ItemListBinding.inflate(
21	LayoutInflater.from(parent.context),
22	parent,
23	false
24)
25	return ViewHolder(binding)
26	}
27	override fun onBindViewHolder(holder: ViewHolder,
28	position: Int) {
29	val destination = destinations[position]
30	with(holder.binding) {
31	textViewName.text = destination.nama
32	textViewYear.text = destination.year.toString()
33	textViewDesc.text =
34	destination.descriptionsingkat
	imageView.setImageResource(destination.image)

35	buttonLink.setOnClickListener {
36	val context = it .context
37	val intent = Intent(Intent.ACTION_VIEW,
	Uri.parse(destination.link))
38	context.startActivity(intent)
39	}
40	
41	buttonDetail.setOnClickListener {
42	onDetailClick(destination)
43	}
44	}
45	}
46	
47	override fun getItemCount(): Int = destinations.size
48	}

Tabel 1. 4 Source Code Jawaban Soal 1

5. MyData.kt

1	package com.example.londondestination
2	
3	import android.os.Parcelable
4	import kotlinx.parcelize.Parcelize
5	
6	@Parcelize
7	data class MyData(
8	val nama: String,
9	val description: String,
10	val descriptionsingkat: String,
11	val year: Int,
12	val image: Int,
13	val link: String
14): Parcelable

Tabel 1. 5 Source Code Jawaban Soal 1

6. Detail_fragment.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<ScrollView
	xmlns:android="http://schemas.android.com/apk/res/android"
3	xmlns:app="http://schemas.android.com/apk/res-auto"
4	android:id="@+id/detailScrollView"
5	android:layout_width="match_parent"
6	android:layout_height="match_parent"
7	android:padding="16dp">
8	
9	<androidx.cardview.widget.CardView
10	android:layout_width="match_parent"

11	android:layout_height="wrap_content"
12	app:cardCornerRadius="16dp"
13	app:cardElevation="8dp">
14	
15	<LinearLayout
16	android:layout_width="match_parent"
17	android:layout_height="wrap_content"
18	android:orientation="vertical">
19	
20	<ImageView
21	android:id="@+id/detailImage"
22	android:layout_width="match_parent"
23	android:layout_height="200dp"
24	android:scaleType="centerCrop"/>
25	
26	<TextView
27	android:id="@+id/detailTitle"
28	android:layout_width="match_parent"
29	android:layout_height="wrap_content"
30	android:text="Nama Tempat"
31	android:textStyle="bold"
32	android:textSize="20sp"
33	android:padding="16dp" />
34	
35	<TextView
36	android:id="@+id/detailDescription"
37	android:layout_width="match_parent"
38	android:layout_height="wrap_content"
39	android:text="Deskripsi lengkap tempat
40	destinasi."
41	android:paddingStart="16dp"
42	android:paddingEnd="16dp"
43	android:paddingBottom="16dp"
44	android:textSize="16sp" />
45	</LinearLayout>
46	</androidx.cardview.widget.CardView>
47	</ScrollView>

Tabel 1. 6 Source Code Jawaban Soal 1

7. 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:id="@+id/main_layout"
6	android:layout_width="match_parent"
7	android:layout_height="match_parent"

8	tools:context=".MainActivity">
9	
10	
11	<androidx.fragment.app.FragmentContainerView
12	android:id="@+id/fragment_container_view"
13	android:name="androidx.navigation.fragment.NavHostFragment"
14	android:layout_width="match_parent"
15	android:layout_height="match_parent"
16	app:navGraph="@navigation/nav_graph"
17	app:defaultNavHost="true" />
18	</androidx.constraintlayout.widget.ConstraintLayout>

Tabel 1. 7 Source Code Jawaban Soal 1

8. home_fragment.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=".HomeFragment">
8	
9	<androidx.recyclerview.widget.RecyclerView
10	android:id="@+id/rv_character"
11	android:layout_width="0dp"
12	android:layout_height="0dp"
13	app:layout_constraintBottom_toBottomOf="parent"
14	app:layout_constraintEnd_toEndOf="parent"
15	app:layout_constraintStart_toStartOf="parent"
16	app:layout_constraintTop_toTopOf="parent" />
17	</androidx.constraintlayout.widget.ConstraintLayout>

Tabel 1. 8 Source Code Jawaban Soal 1

9. item_list.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.cardview.widget.CardView
	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="wrap_content"
7	android:layout_margin="12dp"
8	app:cardCornerRadius="16dp"
9	app:cardElevation="6dp">
10	
11	<androidx.constraintlayout.widget.ConstraintLayout

```

12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:padding="12dp">
15
16         <ImageView
17             android:id="@+id/imageView"
18             android:layout_width="100dp"
19             android:layout_height="150dp"
20             android:scaleType="centerCrop"
21             app:layout_constraintTop_toTopOf="parent"
22             app:layout_constraintStart_toStartOf="parent" />
23
24         <TextView
25             android:id="@+id/textViewName"
26             android:layout_width="0dp"
27             android:layout_height="wrap_content"
28             android:text="Main Title"
29             android:textStyle="bold"
30             android:textSize="18sp"
31             android:textColor="#000000"
32             android:layout_marginStart="12dp"
33             android:layout_marginTop="8dp"
34             app:layout_constraintStart_toEndOf="@id/imageView"
35             app:layout_constraintEnd_toEndOf="parent"
36             tools:ignore="MissingConstraints" />
37
38         <TextView
39             android:id="@+id/textViewYear"
40             android:layout_width="0dp"
41             android:layout_height="wrap_content"
42             android:text="1990"
43             android:textSize="16sp"
44             android:textColor="#555555"
45             android:layout_marginStart="12dp"
46             android:layout_marginTop="4dp"
47             app:layout_constraintStart_toEndOf="@id/imageView"
48             app:layout_constraintTop_toBottomOf="@id/textViewName"
49             app:layout_constraintEnd_toEndOf="parent" />
50
51         <TextView
52             android:id="@+id/textViewDesc"
53             android:layout_width="0dp"
54             android:layout_height="wrap_content"
55             android:text="Secondary description text that can
span multiple lines."
56             android:textSize="15sp"
57             android:textColor="#555555"
58             android:layout_marginStart="12dp"
59             android:layout_marginTop="8dp"
60             app:layout_constraintStart_toEndOf="@id/imageView"
61             app:layout_constraintTop_toBottomOf="@id/textViewYear"
62             app:layout_constraintEnd_toEndOf="parent" />

```

63	
64	<LinearLayout
65	android:id="@+id/buttonRow"
66	android:layout_width="wrap_content"
67	android:layout_height="wrap_content"
68	android:orientation="horizontal"
69	android:gravity="center"
70	android:layout_marginTop="16dp"
71	app:layout_constraintTop_toBottomOf="@id/imageView"
72	app:layout_constraintBottom_toBottomOf="parent"
73	app:layout_constraintStart_toStartOf="parent"
74	app:layout_constraintEnd_toEndOf="parent">
75	
76	<Button
77	android:id="@+id/buttonLink"
78	android:layout_width="wrap_content"
79	android:layout_height="wrap_content"
80	android:layout_marginEnd="8dp"
81	android:backgroundTint="@color/purple_200"
82	android:text="Detail"
83	android:textAllCaps="false" />
84	
85	<Button
86	android:id="@+id/buttonDetail"
87	android:layout_width="wrap_content"
88	android:layout_height="wrap_content"
89	android:layout_marginStart="8dp"
90	android:backgroundTint="@color/purple_200"
91	android:text="Info"
92	android:textAllCaps="false" />
93	</LinearLayout>
94	
95	</androidx.constraintlayout.widget.ConstraintLayout>
96	</androidx.cardview.widget.CardView>

Tabel 1. 9 Source Code Jawaban Soal 1

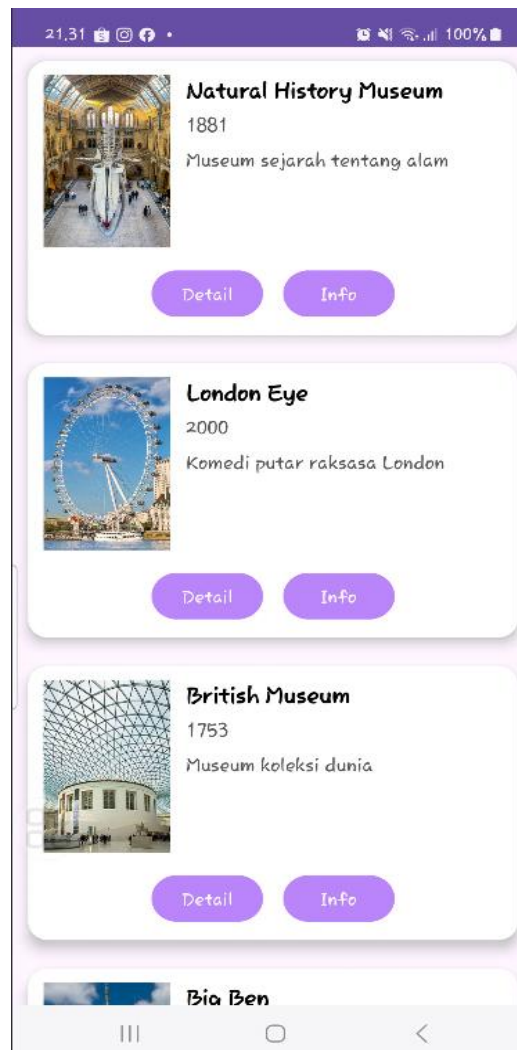
10. nav_graph.xml

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

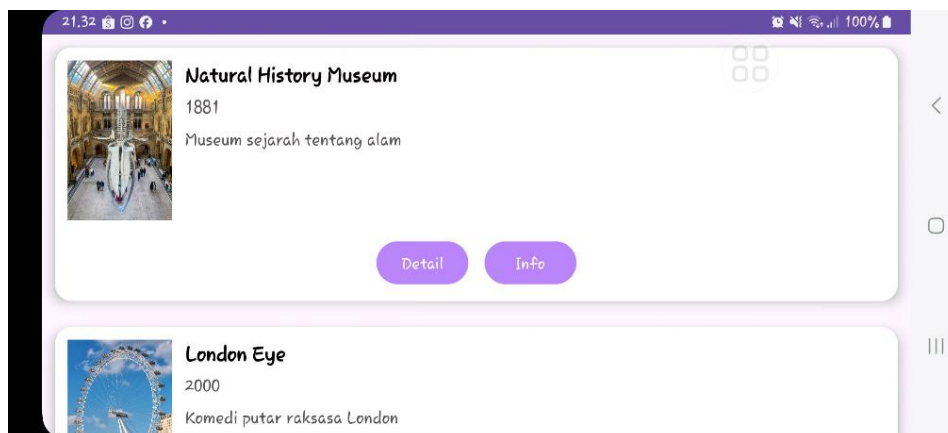
12	android:id="@+id/action_HomeFragment_to_detailFragment"
13	app:destination="@id/detailFragment" />
14	</fragment>
15	
16	<fragment
17	android:id="@+id/detailFragment"
18	android:name="com.example.londondestination.FragmentGuweh"
19	android:label="DetailFragment" >
20	<argument
21	android:name="imageResId"
22	app:argType="integer" />
23	<argument
24	android:name="nama"
25	app:argType="string" />
26	<argument
27	android:name="deskripsi"
28	app:argType="string" />
29	</fragment>
30	
31	</navigation>

Tabel 1. 10 Source Code Jawaban Soal 1

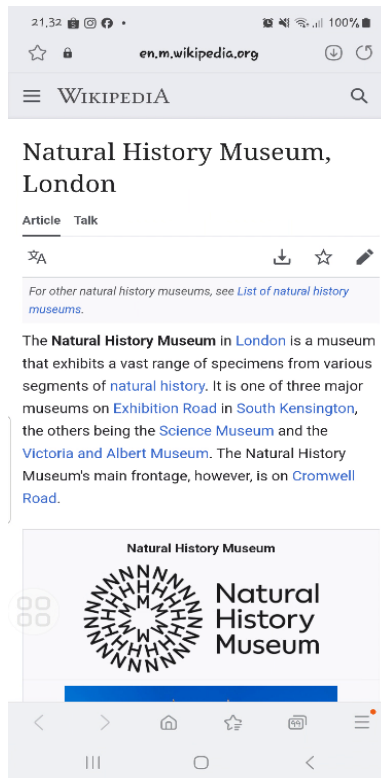
B. Output Program



Gambar 3 Screenshot Hasil Jawaban Soal 1



Gambar 4 Screenshot Hasil Jawaban Soal 1



Gambar 5 Screenshot tombol Detail



Gambar 6 Screenshot tombol Info

C. Pembahasan

1. MainActivity.kt

Di file ini ada MainActivity merupakan class utama dalam aplikasi Android yang digunakan untuk menampilkan antarmuka pengguna dari `activity_main.xml`. Dimana class ini mewarisi dari `AppCompatActivity` adalah versi aktivitas dengan dukungan fitur tambahan. Terdapat `onCreate()`, `ActivityMainBinding` digunakan untuk menghubungkan layout XML dengan kode Kotlin secara langsung melalui View Binding, sehingga memudahkan akses ke elemen UI tanpa menggunakan `findViewById`. Kemudian ada `setContentView(binding.root)` digunakan untuk menampilkan layout tersebut sebagai isi dari aktivitas yang digunakan.

2. FragmentGuweh.kt

Pada class `FragmentGuweh` berguna agar bisa menampilkan detail dari suatu destinasi di aplikasi. Dimana fragment ini memakai View Binding (`DetailFragmentBinding`) untuk mengakses elemen-elemen UI dari layout `detail_fragment.xml` tanpa perlu adanya `findViewById`. Dilihat bagian `onCreateView` yang mana layout di-*inflate* akan dihubungkan menuju fragment melalui binding. Juga, di `onViewCreated` nantinya data seperti gambar (`imageResId`), nama, dan deskripsi diambil dari argumen yang dikirim ke fragment dan ditampilkan ke elemen UI seperti `ImageView` dan `TextView`. Selain itu, ada fungsi `newInstance()` di bagian companion object berguna sebagai pembuatan instance fragment baru sekaligus mengisi argumen yang dibutuhkan. Terakhir, ada `onDestroyView` agar binding dihapus untuk mencegah kebocoran memori.

3. HomeFragment.kt

Pada class `HomeFragment` merupakan bagian dari aplikasi Android yang menampilkan daftar destinasi wisata di London dalam bentuk `RecyclerView`. Dimana fragment ini menggunakan View Binding (`HomeFragmentBinding`)

agar bisa menghubungkan elemen layout dengan kode tanpa harus menggunakan `findViewById`. Kita lihat di `onCreateView` nantinya layout di-*inflate* dan dikembalikan menjadi tampilan utama. Dengan adanya `onViewCreated`, maka fragment dapat mengatur `RecyclerView` menggunakan `LinearLayoutManager` dan adapter (`MyAdapter`) berisi daftar objek `MyData`. Dimana setiap item berisi informasi lengkap seperti nama, deskripsi, gambar, dan link dari destinasi wisata. Saat kita mencoba mengklik salah satu item, maka data dari item tersebut akan terkirim melalui *Bundle* yang menuju fragment detail (`FragmentGuweh`) menggunakan `Navigation Component`. Terakhir, ada `onDestroyView` berguna untuk menghapus binding agar mencegah kebocoran memori. Jadi, fragment ini berfungsi sebagai tampilan utama yang menampilkan seluruh destinasi dalam bentuk daftar.

4. `MyAdapter.kt`

Pada class `MyAdapter` untuk bisa menampilkan daftar destinasi wisata dalam `RecyclerView`. Dimana adapter ini menerima dua parameter yaitu, daftar data (`destinations`) berupa objek `MyData`, dan sebuah fungsi lambda (`onDetailClick`) nantinya dijalankan saat tombol detail kita klik. Di dalamnya terdapat `ViewHolder` yang menggunakan `ItemListAdapter` untuk mengakses elemen-elemen dari layout item seperti nama, tahun, deskripsi singkat, gambar, dan dua tombol. Kemudian, di dalam `onBindViewHolder` setiap data diatur menuju tampilan sesuai posisinya, termasuk menampilkan teks dan gambar. Nah, ketika kita klik nantinya membuka halaman web sesuai URL destinasi melalui `Intent`. Sedangkan bagian tombol detail akan menjalankan fungsi `onDetailClick` dan mengirim data yang dipilih ke fragment detail. Dengan adanya adapter bisa menjembatani data dan tampilan dalam daftar secara efisien, sekaligus menangani interaksi kita sebagai pengguna lebih baik.

5. MyData.kt

Pada data class bernama `MyData` yang berfungsi sebagai model data utama untuk aplikasi destinasi wisata London. Dimana class ini menggunakan anotasi `@Parcelize` yang memberi kemungkinan objek `MyData` bisa dikirim antar komponen Android seperti antar fragment atau activity dengan lebih mudah melalui `Bundle`. Dengan adanya class ini menyimpan enam properti penting untuk tiap destinasi seperti, nama (nama tempat wisata), `description` (deskripsi lengkap), `descriptionsingkat` (deskripsi singkat), `year` (tahun pendirian atau peresmian), `image` (ID dari gambar sumber daya), dan `link` (tautan ke halaman informasi lebih lanjut). Pentingnya struktur ini agar aplikasi bisa menampilkan informasi yang lengkap dan interaktif mengenai setiap lokasi wisata dalam bentuk daftar, serta meneruskan data dengan rapi ke tampilan detail saat kita memilih salah satu destinasi. Jadi, file ini menjadi pondasi data yang akan digunakan oleh adapter dan fragment lainnya di aplikasi ini.

6. Detail_fragment.xml

Pada bagian ini digunakan untuk menampilkan tampilan detail dari sebuah tempat wisata dalam aplikasi. Dimana seluruh konten dibungkus di dalam sebuah elemen `ScrollView` yang memungkinkan kita bisa menggulir layar ke bawah jika isi kontennya lebih panjang dari ukuran layar. Hal ini penting agar seluruh informasi tetap bisa diakses meskipun banyak atau Panjang isinya.

Di dalam `ScrollView` terdapat sebuah `CardView` yang berfungsi memberikan tampilan lebih menarik karena memiliki sudut melengkung (dengan `cardCornerRadius="16dp"`) dan (`cardElevation="8dp"`), sehingga konten tampak seperti kartu sedikit terangkat dari latar belakang membuat tampilan lebih rapi dan enak dilihat.

Isi dari `CardView` diletakkan di dalam `LinearLayout` yang diatur secara vertikal. Di dalam layout ini terdapat tiga komponen utama yaitu, pertama ada `ImageView` menggunakan ID `detailImage` berguna untuk menampilkan gambar tempat wisata yang biasanya gambar ini nantinya ditampilkan dari file `drawable` atau dari sumber lain. Dengan gambar ini disetel menggunakan

`scaleType="centerCrop"` agar mengisi seluruh ruang yang disediakan dengan proporsional.

Selanjutnya, dua `TextView` yang pertama ada `detailTitle` digunakan untuk menampilkan nama tempat dengan ukuran teks yang cukup besar (20sp) dan gaya teks tebal (bold), serta diberi `padding` agar teks tidak menempel langsung ke sisi layar. Yang kedua, `detailDescription` berfungsi untuk menampilkan deskripsi lengkap dari tempat tersebut dengan deskripsi ini menggunakan ukuran teks sedikit lebih kecil (16sp) dan diberi `padding` sisi kiri, kanan, dan bawah agar memudahkan saat membaca. Jadi, secara keseluruhan, layout ini dibuat untuk memberikan tampilan detail tempat wisata secara bersih, informatif, dan nyaman dilihat.

7. `activity_main.xml`

Pada bagian ini berguna untuk mengelola navigasi antar-fragment. Dimana seluruh tampilan dibungkus di dalam `ConstraintLayout` merupakan salah satu jenis layout fleksibel yang memungkinkan pengaturan posisi antar elemen secara dinamis dan efisien. Namun dalam kasus ini, hanya ada satu elemen di dalamnya, jadi `ConstraintLayout` tidak benar-benar dimanfaatkan secara penuh. Dengan elemen utamanya itu `FragmentContainerView` berfungsi sebagai wadah (container) agar bisa menampilkan fragment. Adanya `FragmentContainerView` ini memiliki ID `fragment_container_view` dan diatur untuk mengisi seluruh lebar dan tinggi layar (`match_parent`). Terdapat property berisi `android:name` yang menunjuk ke bagian `androidx.navigation.fragment.NavHostFragment` berarti view ini berperan sebagai host fragment nantinya saat mengelola navigasi.

Adanya atribut `app:navGraph` menunjuk ke file `nav_graph` di direktori `res/navigation` berisi struktur navigasi fragment—seperti daftar tujuan (destination) dan hubungan antar fragment (misalnya aksi berpindah antar fragment). Sementara `app:defaultNavHost="true"` digunakan dalam menyatakan bahwa fragment ini adalah host default agar bisa menangani navigasi sistem, seperti

tombol "Back" di Android. Jadi, keseluruhan bagian ini XML ini menyusun fondasi navigasi fragment berbasis Jetpack Navigation dengan melakukan penempatan satu `NavHostFragment` untuk mengatur transisi antar-fragment di dalam aplikasi tanpa perlu berpindah antar activity yang menjadikan navigasi lebih efisien dan lebih terstruktur.

8. home_fragment.xml

Pada bagian ini adalah layout untuk sebuah Fragment lebih tepatnya `HomeFragment` yang menggunakan `ConstraintLayout` sebagai layout utama. Dimana layout ini hanya memiliki satu elemen di dalamnya, yaitu sebuah `RecyclerView` dengan ID `rv_character` dengan fungsi utama dari `RecyclerView` ini untuk menampilkan daftar item secara efisien dan dapat discroll tergantung pada pengaturan adapter dan layout manager-nya.

Pada pengaturannya `RecyclerView` ini dibuat agar memenuhi seluruh ruang layar karena semua constraint-nya dihubungkan ke tepi-tepi parent layout. Hal ini terlihat dari penggunaan `0dp` untuk `layout_width` dan `layout_height` berarti ukuran akan disesuaikan berdasarkan constraint yang diberikan. Dilihat bagian Constraint-nya menghubungkan atas (Top), bawah (Bottom), kiri (Start), dan kanan (End) ke parent, sehingga `RecyclerView` menutupi seluruh tampilan yang tersedia dalam fragment.

Sementara itu, kita lihat di atribut `tools:context=".HomeFragment"` berguna hanya untuk keperluan preview di Android Studio agar tampilan ini bisa dikenali sedang digunakan `HomeFragment` saat mendesain antarmuka. Jadi, secara keseluruhan layout bagian ini sangat sederhana tapi efektif demi bisa menampilkan daftar yang dinamis, seperti daftar tempat wisata, karakter, atau data lain nantinya diisi melalui adapter dalam kode program aplikasi ini.

9. item_list.xml

Pada layout file untuk komponen tampilan dalam aplikasi Android yang mana layout ini menggunakan `CardView` sebagai wadah utama yang memberikan efek tampilan seperti kartu, lengkap dengan sudut melengkung dan bayangan. Di dalam `CardView` terdapat `ConstraintLayout` digunakan sebagai layout utama agar bisa menyusun elemen-elemen UI dengan fleksibilitas tinggi sesuai posisi satu sama lain. Dimana bagian pertama itu `ImageView` berguna untuk menampilkan gambar, dengan ukurannya ditetapkan 100dp x 150dp dan letaknya di pojok kiri atas tampilan. Gambar diatur agar "crop" ke tengah (`centerCrop`) agar mengisi seluruh ruang.

Selanjutnya, ada tiga `TextView` yang masing-masing menampilkan nama (`textViewName`), tahun atau angkatan (`textViewYear`), dan deskripsi tambahan (`textViewDesc`). Dari semua `TextView` ini diletakkan di sebelah kanan `ImageView` yang disusun secara vertikal satu per satu dari atas ke bawah. Adanya `textViewName` yang menggunakan teks tebal dan ukuran huruf lebih besar cocok untuk judul atau nama utama. Sedangkan `textViewYear` dan `textViewDesc` memiliki ukuran teks yang lebih kecil dan warna abu-abu gelap (`#555555`) untuk memberikan perbedaan teks.

Di bagian paling bawah layout terdapat `LinearLayout` yang berisi dua tombol (`Button`), yaitu tombol "Detail" dan "Info". Dengan layout tombol ini diposisikan di bawah `ImageView` dan diberi margin atas agar tidak menempel langsung. Dimana warna latar tombol diambil dari `@color/purple_200` yang memberi warna ungu dari resources aplikasi. Kedua tombol ini disiapkan untuk tindakan lebih lanjut, misalnya "Detail" untuk membuka tampilan biodata lengkap, dan "Info" untuk menampilkan informasi tambahan.

Jadi, secara keseluruhan, XML ini digunakan untuk item dalam `RecyclerView` karena desainnya ringkas, informatif, dan responsif. Dengan struktur layout sudah rapi sesuai pengaturan `ConstraintLayout` yang fleksibel dan `CardView` mempercantik tampilan menjadi tipe layout biasanya digunakan agar bisa menampilkan daftar data dalam bentuk kartu.

10. nav_graph.xml

Pada file `navigation graph` fungsinya untuk mendefinisikan alur navigasi antar tampilan atau fragment dalam sebuah aplikasi. Di bagian dalamnya akan dideklarasikan dua buah fragment yaitu, `HomeFragment` dan `detailFragment`. Dimana layout navigasi ini untuk aplikasi dapat berpindah dari satu fragment menuju fragment lainnya secara terstruktur dan mudah dikelola.

Pertama, elemen `<navigation>` menyatakan bahwa ini bagian root dari `navigation graph` yang terdapat atribut ini `app:startDestination="@id/HomeFragment"` yang menandakan bahwa saat aplikasi dijalankan, maka tampilan awal (halaman pertama) nantinya ditampilkan itu `HomeFragment`. Dengan fragment ini diberi ID `@+id/HomeFragment` dan terhubung menuju bagian dari class `com.example.londondestination.HomeFragment`.

Di dalam `HomeFragment` terdapat elemen `<action>` dengan ID `action_HomeFragment_to_detailFragment` merupakan aksi navigasi yang berarti jika dipicu (misalnya kita klik tombol), maka aplikasi akan berpindah dari `HomeFragment` menuju `detailFragment`. Nama action ini bisa dipanggil di kode Kotlin atau Java untuk mentrigger perpindahan halaman ke selanjutnya.

Adanya fragment tujuan yaitu `detailFragment` dikaitkan dengan class Java atau Kotlin `com.example.londondestination.FragmentGuweh`. Dimana fragment ini nantinya menerima data melalui tiga buah argument seperti, `imageResId` (tipe integer, biasanya ID gambar dari resource drawable), nama (tipe string), dan deskripsi (juga string) dari ketiga argumen ini memberi kemungkinan fragment tujuan bisa menerima data yang dikirim dari `HomeFragment`, misalnya saat kita klik tombol "Detail". Jadi, file ini memastikan bahwa aplikasi punya alur navigasi yang rapi dari halaman utama ke halaman detail, dan sudah siap menerima data agar bisa ditampilkan.

SOAL 2

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

Jawab:

Menurut saya RecyclerView masih banyak digunakan karena sebagian besar aplikasi Android lama yang masih pakai View XML, bukan Jetpack Compose. Nah, pada LazyColumn itu cuma bisa dipakai semisal kita udah pakai Compose. Jadi, kalau aplikasi yang dibuat itu masih pakai layout XML biasa, pilihan paling bagus dan umum untuk menampilkan daftar data tetap RecyclerView. Dengan konteks memang benar bahwa RecyclerView itu kodenya lebih panjang karena harus bikin Adapter, ViewHolder, dan file layout item-nya, dll. Tetapi, kelebihanannya itu sangat fleksibel dengan bisa menampilkan data yang banyak tanpa bikin aplikasi jadi lemot, dan bisa dikustomisasi banget, misalnya saat kita pakai tombol, gambar, atau animasi di tiap item. Jadi, meskipun lebih ribet karena RecyclerView masih dipakai namun memiliki beberapa kelebihan, sebagai berikut:

- Cocok untuk View XML,
- Sudah stabil dan teruji,
- Banyak tutorial dan contoh yang bisa kita ikuti.

Nah. selama belum pakai Jetpack Compose, maka RecyclerView tetap pilihan yang paling tepat buat menampilkan list data.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/alysaarmelia/AlysaArmelia_2310817120009_Pemrograman_Mobile/tree/dfc1276050282d4aa43523b173d5c8fb0058914f/PRAK_MODUL3