

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



BUILD A SCROLLABLE LIST

Oleh:

Nur Hikmah

NIM. 2310817120010

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
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 : Nur Hikmah
NIM : 2310817120010

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	22
C. Pembahasan	26
D. Tautan Git	34
SOAL 2.....	35
A. Pembahasan	35

DAFTAR GAMBAR

Gambar 1. Contoh UI List	7
Gambar 2. Contoh UI Detail	7
Gambar 3. Screenshot Hasil Jawaban Soal 1 UI List	22
Gambar 4. Screenshot Hasil Jawaban Soal 1 UI List 10 Item.....	24
Gambar 5. Screenshot Hasil Jawaban Soal 1 Berpindah ke Aplikasi Lain Saat Menekan Tombol Shopee.....	24
Gambar 6. Screenshot Hasil Jawaban Soal 1 UI Detail Saat Menekan Tombol Detail	25

DAFTAR TABEL

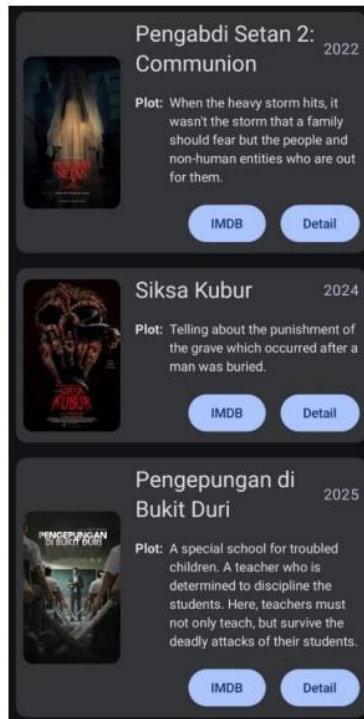
Tabel 1. Source Code Jawaban Soal 1.....	9
Tabel 2. Source Code Jawaban Soal 1.....	10
Tabel 3. Source Code Jawaban Soal 1.....	11
Tabel 4. Source Code Jawaban Soal 1.....	12
Tabel 5. Source Code Jawaban Soal 1.....	12
Tabel 6. Source Code Jawaban Soal 1.....	13
Tabel 7. Source Code Jawaban Soal 1.....	15
Tabel 8. Source Code Jawaban Soal 1.....	16
Tabel 9. Source Code Jawaban Soal 1.....	17
Tabel 10. Source Code Jawaban Soal 1.....	21
Tabel 11. Source Code Jawaban Soal 1.....	21

SOAL 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. Contoh UI Detail

A. Source Code

1. HomeFragment.kt

```
1 package com.example.modul3
2
3 import android.os.Bundle
4 import androidx.fragment.app.Fragment
5 import android.view.LayoutInflater
6 import android.view.View
7 import android.view.ViewGroup
8 import androidx.navigation.fragment.findNavController
9 import androidx.recyclerview.widget.LinearLayoutManager
10 import com.example.modul3.databinding.FragmentHomeBinding
11
12 class HomeFragment : Fragment() {
13     private var _binding: FragmentHomeBinding? = null
14     private val binding get() = _binding!!
15
16     override fun onCreateView(
17         inflater: LayoutInflater, container: ViewGroup?,
18         savedInstanceState: Bundle?
19     ): View {
20         _binding = FragmentHomeBinding.inflate(inflater,
21 container, false)
22         return binding.root
23     }
24
25     override fun onViewCreated(view: View,
26 savedInstanceState: Bundle?) {
27         val listTas = generateTasList()
28         val adapter = AdapterTas(listTas,
29 findNavController())
30
31         binding.recyclerView.apply {
32             layoutManager =
33             LinearLayoutManager(requireContext())
34             this.adapter = adapter
35         }
36
37     override fun onDestroyView() {
38         super.onDestroyView()
39         _binding = null
40     }
41
42     private fun generateTasList(): List<ItemTas> {
43         val context = requireContext()
44         val names =
45         context.resources.getStringArray(R.array.data_name)
46         val harga =
47         context.resources.getStringArray(R.array.data_harga)
48         val deskripsi =
```


44	context.resources.getStringArray(R.array.data_deskripsi)
	val links =
	context.resources.getStringArray(R.array.data_link)
45	
46	val imageIds = listOf(
47	R.drawable.marsya_bag,
48	R.drawable.lyn_bag,
49	R.drawable.reigny_bag,
50	R.drawable.madeline_bag,
51	R.drawable.tote_bag,
52	R.drawable.mini_cendy_bag,
53	R.drawable.amora_bag,
54	R.drawable.gradien_bag,
55	R.drawable.lysmare_bag,
56	R.drawable.delphine_bag
57)
58	
59	val list = mutableListOf<ItemTas>()
60	for (i in names.indices) {
61	list.add(
62	ItemTas(
63	nama = names[i],
64	harga = harga[i],
65	deskripsi = deskripsi[i],
66	gambar = imageIds[i],
67	link = links[i]
68)
69)
70	}
71	return list
72	}
73	}

Tabel 1. Source Code Jawaban Soal 1

2. DetailFragment.kt

1	package com.example.modul3
2	
3	import android.os.Bundle
4	import android.util.Log
5	import android.view.LayoutInflater
6	import android.view.View
7	import android.view.ViewGroup
8	import androidx.fragment.app.Fragment
9	import com.example.modul3.databinding.FragmentDetailBinding
10	
11	class DetailFragment : Fragment() {
12	
13	private var _binding: FragmentDetailBinding? = null
14	private val binding get() = _binding!!
15	

```

16     override fun onCreateView(
17         inflater: LayoutInflater, container: ViewGroup?,
18         savedInstanceState: Bundle?
19     ): View {
20         _binding = FragmentDetailBinding.inflate(inflater,
container, false)
21         return binding.root
22     }
23
24     override fun onViewCreated(view: View,
savedInstanceState: Bundle?) {
25         super.onViewCreated(view, savedInstanceState)
26
27         Log.d("DetailFragment", "Arguments: $arguments")
28         val tas = arguments?.getParcelable<ItemTas>("tas")
29         Log.d("DetailFragment", "Tas received: $tas")
30         tas?.let {
31             binding.tvName.text = it.nama
32             binding.tvHarga.text =
getString(R.string.label_harga) + " " + it.harga
33             binding.tvDeskripsi.text = it.deskripsi
34             binding.imgItemPhoto.setImageResource(it.gambar)
35         }
36     }
37
38     private fun getImageResourceIdByName(name: String): Int {
39         val cleanName = name.replace("@drawable/", "")
40         return resources.getIdentifier(cleanName, "drawable",
requireContext().packageName)
41     }
42
43     override fun onDestroyView() {
44         super.onDestroyView()
45         _binding = null
46     }
47 }

```

Tabel 2. Source Code Jawaban Soal 1

3. ItemTas.kt

```

1 package com.example.modul3
2
3 import android.os.Parcelable
4 import kotlinx.parcelize.Parcelize
5
6 @Parcelize
7 data class ItemTas(
8     val nama: String,
9     val gambar: Int,
10    val harga: String,
11    val deskripsi: String,

```

12	val link: String
13) : Parcelable

Tabel 3. Source Code Jawaban Soal 1

4. AdapterTas.kt

```

1 package com.example.modul3
2
3 import android.content.Intent
4 import android.net.Uri
5 import android.os.Bundle
6 import android.util.Log
7 import android.view.LayoutInflater
8 import android.view.ViewGroup
9 import androidx.navigation.NavController
10 import androidx.recyclerview.widget.RecyclerView
11 import com.example.modul3.databinding.ItemTasBinding
12
13 class AdapterTas(
14     private val listTas: List<ItemTas>,
15     private val navController: NavController
16 ) : RecyclerView.Adapter<AdapterTas.ListViewHolder>() {
17
18     inner class ListViewHolder(val binding: ItemTasBinding) :
19         RecyclerView.ViewHolder(binding.root) {
20
21         fun bind(tas: ItemTas) {
22             binding.tvItemName.text = tas.nama
23             binding.tvItemPrice.text = tas.harga
24             binding.imgItemPhoto.setImageResource(tas.gambar)
25             binding.buttonDetail.setOnClickListener {
26                 Log.d("AdapterTas", "Tombol detail diklik")
27                 val bundle = Bundle().apply {
28                     putParcelable("tas", tas)
29                 }
30                 navController.navigate(R.id.action_homeFragment_to_detailFragment,
31                     bundle)
32                 binding.buttonShopee.setOnClickListener {
33                     val intent = Intent(Intent.ACTION_VIEW,
34                         Uri.parse(tas.link))
35                     binding.root.context.startActivity(intent)
36                 }
37             }
38
39             override fun onCreateViewHolder(parent: ViewGroup, viewType:
40                 Int): ListViewHolder {
41                 val binding =
42                     ItemTasBinding.inflate(LayoutInflater.from(parent.context),
43                         parent, false)

```

41	return ViewHolder(binding)
42	}
43	
44	override fun onBindViewHolder(holder: ViewHolder,
45	position: Int) {
46	holder.bind(listTas[position])
47	}
48	override fun getItemCount(): Int = listTas.size
49	}

Tabel 4. Source Code Jawaban Soal 1

5. MainActivity.kt

1	package com.example.modul3
2	
3	import android.os.Bundle
4	import androidx.appcompat.app.AppCompatActivity
5	import com.example.modul3.databinding.ActivityMainBinding
6	
7	class MainActivity : AppCompatActivity() {
8	private lateinit var binding: ActivityMainBinding
9	override fun onCreate(savedInstanceState: Bundle?) {
10	super.onCreate(savedInstanceState)
11	binding = ActivityMainBinding.inflate(layoutInflater)
12	setContentView(binding.root)
13	}
14	}

Tabel 5. Source Code Jawaban Soal 1

6. fragment_home.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/homeFragment"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context=".HomeFragment">
9
10  <TextView
11    android:layout_width="match_parent"
12    android:layout_height="wrap_content"
13    android:layout_marginTop="0dp"
14    android:background="#D1B3FF"
15    android:padding="15dp"
16    android:text="ShopBag - Your Bag Your Style"
17    android:textStyle="bold"
18    android:textColor="@color/black"
19    android:textSize="25sp" />
20
21  <androidx.recyclerview.widget.RecyclerView
22    android:id="@+id/recyclerView"
23    android:layout_width="match_parent"
24    android:layout_height="match_parent"
25    android:padding="8dp" />
26
27 </LinearLayout>
```

Tabel 6. Source Code Jawaban Soal 1

7. item_tas.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:id="@+id/card_view"
6   android:layout_width="match_parent"
7   android:layout_height="wrap_content"
8   android:layout_gravity="center"
9   android:layout_marginStart="8dp"
10  android:layout_marginTop="4dp"
11  android:layout_marginEnd="8dp"
12  android:layout_marginBottom="4dp"
13  app:cardCornerRadius="12dp"
14  app:cardElevation="10dp">
15
16  <androidx.constraintlayout.widget.ConstraintLayout
```

```

17         android:layout_width="match_parent"
18         android:layout_height="wrap_content"
19         android:background="#F6F6F6"
20         android:padding="20dp">
21
22         <ImageView
23             android:id="@+id/img_item_photo"
24             android:layout_width="135dp"
25             android:layout_height="140dp"
26             android:scaleType="centerCrop"
27             app:layout_constraintStart_toStartOf="parent"
28             app:layout_constraintTop_toTopOf="parent"
29             app:layout_constraintBottom_toBottomOf="parent"
30             android:contentDescription="@string/app_name"
31             tools:src="@drawable/marsya_bag" />
32
33         <TextView
34             android:id="@+id/tv_item_name"
35             android:layout_width="0dp"
36             android:layout_height="wrap_content"
37             android:text="Tas Kulit Elegan"
38             android:textColor="@color/black"
39             android:textStyle="bold"
40             android:textSize="20sp"
41             app:layout_constraintTop_toTopOf="parent"
42             app:layout_constraintStart_toEndOf="@id/img_item_photo"
43             app:layout_constraintEnd_toEndOf="parent"
44             app:layout_constraintHorizontal_bias="0.0"
45             android:layout_marginStart="8dp"
46             tools:ignore="MissingConstraints" />
47
48         <TextView
49             android:id="@+id/tv_item_price"
50             android:layout_width="0dp"
51             android:layout_height="wrap_content"
52             android:text="Rp 350.000"
53             android:textSize="16sp"
54             android:textColor="@color/black"
55             app:layout_constraintTop_toBottomOf="@id/tv_item_name"
56             app:layout_constraintStart_toStartOf="@id/tv_item_name"
57             app:layout_constraintEnd_toEndOf="parent"
58             android:layout_marginTop="4dp" />
59
60         <Button
61             android:id="@+id/button_shopee"
62             android:layout_width="wrap_content"
63             android:layout_height="wrap_content"
64             android:layout_marginStart="8dp"
65             android:layout_marginTop="40dp"
66             android:text="@string/btn_shopee"
67             app:layout_constraintStart_toStartOf="@id/tv_item_name"
68             app:layout_constraintTop_toBottomOf="@id/tv_item_price"/>

```

69	
70	<Button
71	android:id="@+id/button_detail"
72	android:layout_width="wrap_content"
73	android:layout_height="wrap_content"
74	android:layout_marginStart="20dp"
75	android:text="@string/btn_detail"
76	app:layout_constraintStart_toEndOf="@id/button_shopee"
77	app:layout_constraintTop_toTopOf="@id/button_shopee"/>
78	</androidx.constraintlayout.widget.ConstraintLayout>
79	</androidx.cardview.widget.CardView>

Tabel 7. Source Code Jawaban Soal 1

8. fragment_detail.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<ScrollView
	xmlns:android="http://schemas.android.com/apk/res/android"
3	xmlns:tools="http://schemas.android.com/tools"
4	xmlns:app="http://schemas.android.com/apk/res-auto"
5	android:layout_width="match_parent"
6	android:layout_height="match_parent"
7	android:padding="16dp"
8	android:background="#F6F6F6"
9	tools:context=".DetailFragment">
10	
11	<androidx.constraintlayout.widget.ConstraintLayout
12	android:layout_width="match_parent"
13	android:layout_height="wrap_content"
14	android:background="#F6F6F6">
15	
16	<ImageView
17	android:id="@+id/img_item_photo"
18	android:layout_width="350dp"
19	android:layout_height="350dp"
20	android:scaleType="centerCrop"
21	app:layout_constraintTop_toTopOf="parent"
22	app:layout_constraintStart_toStartOf="parent"
23	app:layout_constraintEnd_toEndOf="parent"
24	android:contentDescription="@string/app_name"
25	tools:src="@drawable/marsya_bag" />
26	
27	<TextView
28	android:id="@+id/tv_name"
29	android:layout_width="wrap_content"
30	android:layout_height="wrap_content"
31	android:textStyle="bold"
32	android:textColor="@color/black"
33	android:textSize="25sp"
34	android:layout_marginTop="16dp"
35	app:layout_constraintTop_toBottomOf="@id/img_item_photo"

36	app:layout_constraintStart_toStartOf="parent"
37	app:layout_constraintEnd_toEndOf="parent"
38	tools:text="BOSTANTEN [Marsya Bag]" />
39	
40	<TextView
41	android:id="@+id/tv_harga"
42	android:layout_width="wrap_content"
43	android:layout_height="wrap_content"
44	android:textColor="@android:color/holo_red_dark"
45	android:textStyle="bold"
46	android:textSize="18sp"
47	android:layout_marginTop="8dp"
48	app:layout_constraintTop_toBottomOf="@id/tv_name"
49	app:layout_constraintStart_toStartOf="parent"
50	app:layout_constraintEnd_toEndOf="parent"
51	tools:text="Harga: Rp179.999" />
52	
53	<TextView
54	android:id="@+id/tv_deskripsi"
55	android:layout_width="match_parent"
56	android:layout_height="wrap_content"
57	android:layout_marginTop="30dp"
58	android:textSize="16sp"
59	android:textColor="@color/black"
60	android:background="#F6F6F6"
61	android:textAlignment="center"
62	android:lineSpacingExtra="5dp"
63	android:padding="8dp"
64	app:layout_constraintTop_toBottomOf="@id/tv_harga"
65	app:layout_constraintStart_toStartOf="parent"
66	app:layout_constraintEnd_toEndOf="parent"
67	tools:text="Deskripsi lengkap tentang tas
68	berbahan kulit PU premium, cocok untuk kuliah dan kerja." />
69	</androidx.constraintlayout.widget.ConstraintLayout>
70	</ScrollView>

Tabel 8. Source Code Jawaban Soal 1

9. 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	xmlns:tools="http://schemas.android.com/tools"
5	app:startDestination="@id/homeFragment">
6	
7	<fragment
8	android:id="@+id/homeFragment"
9	android:name="com.example.modul3.HomeFragment"
10	android:label="fragment_home"

11	tools:layout="@layout/fragment_home">
12	<action
13	android:id="@+id/action_homeFragment_to_detailFragment"
14	app:destination="@id/detailFragment" />
15	</fragment>
16	
17	<fragment
18	android:id="@+id/detailFragment"
19	android:name="com.example.modul3.DetailFragment"
20	android:label="fragment_detail"
21	tools:layout="@layout/fragment_detail" />
22	</navigation>

Tabel 9. Source Code Jawaban Soal 1

10. strings.xml

1	<resources>
2	<string name="app_name">MODUL3</string>
3	<string name="btn_detail">Detail</string>
4	<string name="btn_shopee">Shopee</string>
5	
6	<string-array name="data_name">
7	<item>BOSTANTEN [Marsya Bag]</item>
8	<item>BOSTANTEN [LYN Bag]</item>
9	<item>BOSTANTEN [Reigny Bag]</item>
10	<item>BOSTANTEN [Madeline Bag]</item>
11	<item>BOSTANTEN [Totebag]</item>
12	<item>BOSTANTEN [Mini Cendy Bag]</item>
13	<item>BOSTANTEN [Amora Bag]</item>
14	<item>MOSSDOOM [Gradien Hand Bag]</item>
15	<item>MOSSDOOM [Lysmare Bag]</item>
16	<item>MOSSDOOM [Delphine Bag]</item>
17	</string-array>
18	
19	<string-array name="data_photo">
20	<item>@drawable/marsya_bag</item>
21	<item>@drawable/lyn_bag</item>
22	<item>@drawable/reigny_bag</item>
23	<item>@drawable/madeline_bag</item>
24	<item>@drawable/tote_bag</item>
25	<item>@drawable/mini_cendy_bag</item>
26	<item>@drawable/amora_bag</item>
27	<item>@drawable/gradien_bag</item>
28	<item>@drawable/lysmare_bag</item>
29	<item>@drawable/delphine_bag</item>
30	</string-array>
31	
32	<string-array name="data_link">
33	<item>https://id.shp.ee/ixmk3vP</item>
34	<item>https://id.shp.ee/56fWCro</item>
35	<item>https://id.shp.ee/957MHuZ</item>

36	<code><item>https://id.shp.ee/7ARV68N</item></code>
37	<code><item>https://id.shp.ee/quuhB8b</item></code>
38	<code><item>https://id.shp.ee/RPVhVds</item></code>
39	<code><item>https://id.shp.ee/vKZm8c9</item></code>
40	<code><item>https://id.shp.ee/6tzCzpz</item></code>
41	<code><item>https://id.shp.ee/t3osSMq</item></code>
42	<code><item>https://id.shp.ee/GaVtJQ1</item></code>
43	<code></string-array></code>
44	
45	<code><string-array name="data_harga"></code>
46	<code><item>Rp179.999</item></code>
47	<code><item>Rp159.999</item></code>
48	<code><item>Rp189.999</item></code>
49	<code><item>Rp179.999</item></code>
50	<code><item>Rp169.999</item></code>
51	<code><item>Rp159.999</item></code>
52	<code><item>Rp189.999</item></code>
53	<code><item>Rp249.000</item></code>
54	<code><item>Rp189.000</item></code>
55	<code><item>Rp199.000</item></code>
56	<code></string-array></code>
57	
58	<code><string-array name="data_deskripsi"></code>
59	<code><item>(Bostanten) 100% asli. Selamat datang di toko</code> Bostanten, di mana kami berusaha menyediakan produk yang memenuhi kebutuhan kualitas dan efisiensi pelanggan. Beli produk kami dengan mudah dari pusat perbelanjaan Shopee karena kami menjamin keaslian 100%. Bahan terbuat dari kulit PU dengan pilihan warna Coklat, Hitam, Off Putih, Biru dan Coklat, Aprikot dan Hitam, Kuning Biru Pink, Off Putih dan Coklat, serta Off Putih dan Hitam. Berukuran 19.5 cm (Lebar) × 11.5 cm (Tinggi) × 7 cm (Tebal), dengan berat sekitar 0.35 kg. Tas ini memiliki ruang penyimpanan besar, satu kantong, serta tali belakang yang dapat disesuaikan antara 98-122 cm. Cocok digunakan untuk sekolah, kantor, perjalanan, perjalanan bisnis, belanja, pertemuan, makan, kencan, dan acara sosial seperti pesta atau konser. Target penggunaanya adalah wanita, gadis, pekerja kantoran, pelajar, dan remaja.</item>
60	<code><item>(Bostanten) 100% Asli. Selamat datang di toko</code> Bostanten, di mana kami akan berusaha untuk menyediakan produk yang memenuhi kebutuhan kualitas dan efisiensi kepada pelanggan. Beli produk kami dengan mudah dari pusat perbelanjaan Shopee karena kami menjamin keaslian 100%. Tas ini terbuat dari bahan kulit PU dengan pilihan warna Coklat, Off Putih, Hitam, dan Aprikot Coklat. Memiliki dimensi 29 cm (Lebar) × 14 cm (Tinggi) × 9 cm (Tebal), dan berat sekitar 0.34 kg. Bagian dalam terdiri dari ruang penyimpanan besar, satu tas masukan dalam, dan satu tas zipper internal. Pegangan tas memiliki jarak sedang yaitu 31 cm. Sangat cocok digunakan untuk sekolah, kantor, pesta, dan perjalanan.</item>

61	<p><item>(Bostanten) 100% asli. Selamat datang di toko Bostanten, di mana kami akan berusaha untuk menyediakan produk yang memenuhi kebutuhan kualitas dan efisiensi kepada pelanggan. Beli produk kami dengan mudah dari pusat perbelanjaan Shopee karena kami menjamin keaslian 100%. Tas ini menggunakan bahan kulit PU dan tersedia dalam warna Off Putih, Warna Kamel, dan Hitam. Ukurannya adalah 20 cm (Lebar) × 16 cm (Tinggi) × 9 cm (Tebal), dengan berat 0.27 kg. Dilengkapi dengan ruang penyimpanan besar, tas masukan, dan sabuk bahu dengan tinggi tengah sekitar 92-112 cm yang dapat disesuaikan dan dilepas pasang. Pegangan tas berukuran sedang, yaitu 9 cm. Cocok untuk sekolah, kantor, pesta, maupun perjalanan.</item></p>
62	<p><item>(Bostanten) 100% asli. Selamat datang di toko Bostanten, di mana kami akan berusaha untuk menyediakan produk yang memenuhi kebutuhan kualitas dan efisiensi kepada pelanggan. Beli produk kami dengan mudah dari pusat perbelanjaan Shopee karena kami menjamin keaslian 100%. Tas ini terbuat dari bahan kulit PU dan tersedia dalam warna Off Putih, Warna Kamel, Hitam, Abu-abu, dan Coklat. Berukuran 28.5 cm (Lebar) × 21.5 cm (Tinggi) × 7 cm (Tebal), dengan berat 0.29 kg. Bagian dalam terdiri dari ruang penyimpanan besar, satu tas zipper belakang, dan satu saku masukan depan. Sabuk bahunya memiliki tinggi tengah sekitar 0-111 cm dan tidak dapat disesuaikan maupun dilepas. Tas ini cocok digunakan untuk sekolah, kantor, pesta, maupun perjalanan.</item></p>
63	<p><item>(Bostanten) 100% asli. Selamat datang di toko Bostanten, di mana kami akan berusaha untuk menyediakan produk yang memenuhi kebutuhan kualitas dan efisiensi kepada pelanggan. Beli produk kami dengan mudah dari pusat perbelanjaan Shopee karena kami menjamin keaslian 100%. Keserbagunaan, kepraktisan, elegan, dan gaya kedewasaan adalah inti keindahan dari tas ini. Bahan warna solid dan lekukan yang natural memberikan tampilan yang serbaguna serta memiliki plastisitas yang tepat untuk menonjolkan gaya dewasa wanita. Barang pegangan terpopuler di musim ini adalah tote bag-tas jinjing berkapasitas besar yang berkualitas baik dan terbuat dari bahan tahan air. Kapasitasnya sangat luar biasa; Anda bisa memasukkan laptop, buku teks, tablet, dan barang lainnya dengan mudah. Kepraktisannya sempurna dan sangat ramah bagi pelajar, pekerja kantoran, maupun ibu-ibu yang membutuhkan ruang ekstra. Tas ini terbuat dari bahan kulit PU dan tersedia dalam kombinasi warna Off Putih dan Coklat, Off Putih dan Pink, serta Off Putih dan Hijau. Berukuran 40 cm (Lebar) × 27 cm (Tinggi) × 12 cm (Tebal) dengan berat 0.50 kg. Dilengkapi ruang penyimpanan besar, kantong, dan kantong resleting, serta tali belakang sepanjang 64 cm. Tote bag ini bukan sekadar barang sederhana yang bisa dipakai kapan saja, tetapi juga mencerminkan sikap terhadap kehidupan. Cocok digunakan di sekolah, kantor, perjalanan, bisnis, belanja,</p>

	<p>pertemuan, makan, kencan, maupun acara sosial seperti pesta dan konser.</p> <p>Target penggunaanya adalah wanita, gadis, pekerja kantoran, pelajar, dan remaja.</item></p>
64	<p><item>(Bostanten) 100% asli. Selamat datang di toko Bostanten, di mana kami akan berusaha untuk menyediakan produk yang memenuhi kebutuhan kualitas dan efisiensi kepada pelanggan. Beli produk kami dengan mudah dari pusat perbelanjaan Shopee karena kami menjamin keaslian 100%. Tas ini terbuat dari bahan kulit PU dan tersedia dalam berbagai warna, seperti Coklat, Off Putih, Hitam, Aprikot dan Coklat, Off Putih dan Coklat, serta Off Putih dan Hitam. Dengan dimensi 12 cm (Lebar) × 18.5 cm (Tinggi) × 6.5 cm (Tebal) dan berat 0.16 kg, tas ini memiliki ruang penyimpanan besar dan satu kantong tambahan. Dilengkapi dengan tali belakang sepanjang 132 cm, tas ini cocok digunakan di sekolah, kantor, perjalanan, perjalanan bisnis, belanja, pertemuan, makan, kencan, serta berbagai acara sosial seperti pesta dan konser. Target penggunaanya ialah wanita, gadis, pekerja kantoran, pelajar, dan remaja.</item></p>
65	<p><item>(Bostanten) 100% asli. Selamat datang di toko Bostanten, di mana kami akan berusaha untuk menyediakan produk yang memenuhi kebutuhan kualitas dan efisiensi kepada pelanggan. Beli produk kami dengan mudah dari pusat perbelanjaan Shopee karena kami menjamin keaslian 100%. Tas ini terbuat dari bahan kulit PU dan tersedia dalam pilihan warna Off Putih, Hitam, Coklat, dan Aprikot. Dengan dimensi 22.5 cm (Lebar) × 26.5 cm (Tinggi) × 3.5 cm (Tebal) dan berat 0.28 kg, tas ini dilengkapi dengan ruang penyimpanan besar, satu kantong, dan satu kantong resleting. Tali belakangnya memiliki panjang antara 103-124 cm. Tas ini cocok digunakan untuk sekolah, kantor, perjalanan, perjalanan bisnis, belanja, pertemuan, makan, kencan, serta acara sosial seperti pesta dan konser. Target penggunaanya ialah wanita, gadis, pekerja kantoran, pelajar, dan remaja.</item></p>
66	<p><item>Selamat datang di produk MOSSDOOM, dijamin keasliannya 100%! Tas ini terbuat dari bahan kulit PU. Dengan ukuran panjang 19.5 cm, lebar 8 cm, tinggi 16.5 cm, dan dengan berat 0.36 kg. Tersedia hanya warna Pink. Tas ini dilengkapi dengan satu saku penyimpanan besar, satu saku sisipan, satu saku beritsleting, dan tinggi pegangan tas tangan sebesar 7cm. Untuk tali bahu memiliki panjang antara 47.5-59cm (dapat disesuaikan). Semua gambar produk ini diambil dalam kehidupan nyata, jadi dijamin keaslinya.</item></p>
67	<p><item>Selamat datang di official MOSSDOOM store, 100% menjamin keaslian brand produk kami. Tas ini terbuat dari bahan kulit PU. Berukuran 20 cm (Panjang) × 8 cm (Lebar) × 14 cm (Tinggi) dan berat 0.36 kg. Tas ini tersedia dalam pilihan warna Blue, Light Pink, Pink, Brown, Beige White, Black. Tas ini dilengkapi dengan ruang penyimpanan besar, satu saku sisipan, satu saku beresleting, dan dengan tali bahu setinggi kira-kira 106-130cm (yang dapat disesuaikan, bisa dilepas</p>

68	<p>pasang). Semua foto produk kami adalah foto asli, diambil dari store official. Jadi dijamin keasliannya.</item></p> <p><item>Selamat datang di official MOSSDOOM store, 100% menjamin keaslian brand produk kami. Tas ini terbuat dari bahan kulit PU. Dengan ukuran 17 cm (Panjang) x 9 cm (Lebar) x 10 cm (Tinggi) dan berat 0.29 kg. Tersedia dengan beberapa pilhan warna Pink, Blue, Purple, Apricot, Beige White, Black. Tas ini memiliki struktur internal dengan saku penyimpanan besar, tinggi tali bahunya sekitar 110-121 cm (yang dapat disesuaikan dan bisa dilepas pasang). Semua foto produk kami adalah foto asli, diambil dari store official. Jadi dijamin keasliannya.</item></p>
69	</string-array>
70	
71	<string name="label_nama">Nama</string>
72	<string name="label_harga">Harga:</string>
73	
74	<string name="placeholder_nama">Nama Tas</string>
75	<string name="placeholder_harga">Harga Tas</string>
76	</resources>

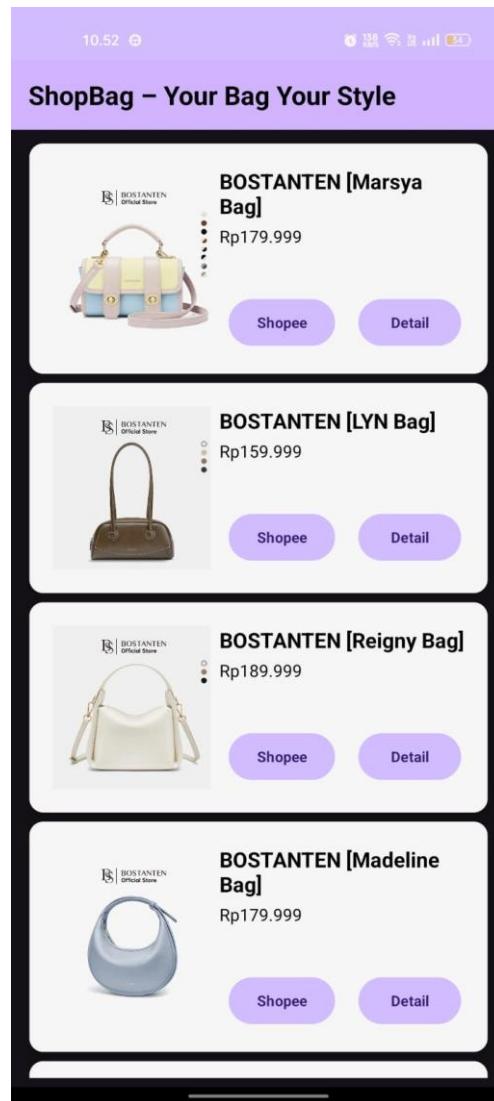
Tabel 10. Source Code Jawaban Soal 1

11. activity_main.xml

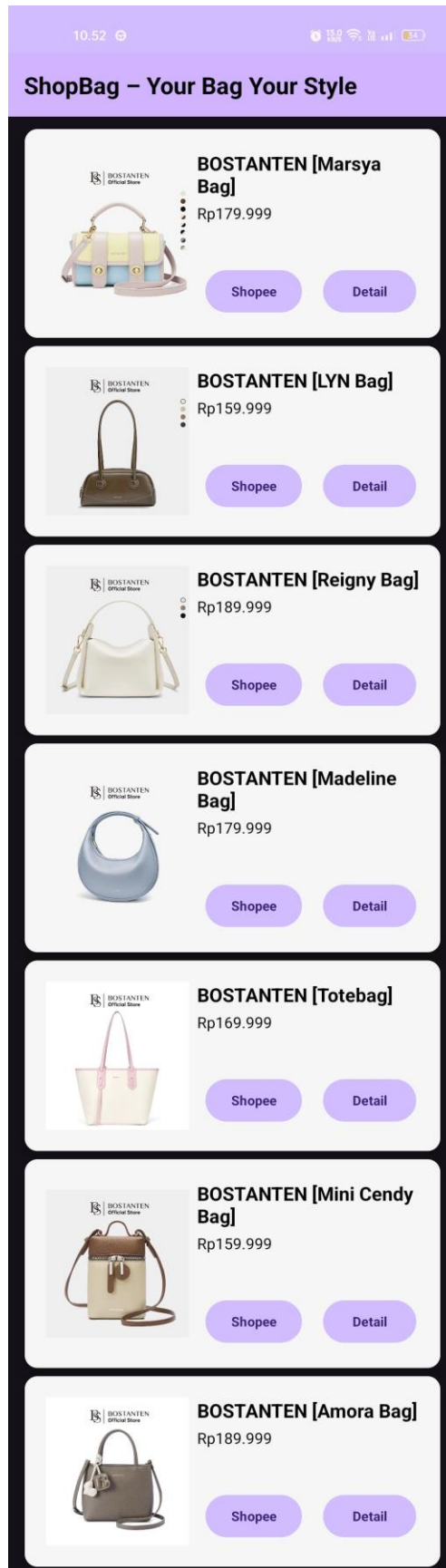
1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
3	xmlns:android="http://schemas.android.com/apk/res/android"
4	xmlns:tools="http://schemas.android.com/tools"
5	xmlns:app="http://schemas.android.com/apk/res-auto"
6	android:layout_width="match_parent"
7	android:layout_height="match_parent"
8	tools:context=".MainActivity"
9	tools:ignore="ExtraText">
10	
11	<androidx.fragment.app.FragmentContainerView
12	android:id="@+id/nav_host_fragme
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 11. Source Code Jawaban Soal 1

B. Output Program



Gambar 3. Screenshot Hasil Jawaban Soal 1 UI List



Gambar 4. Screenshot Hasil Jawaban Soal 1 UI List 10 Item



Gambar 5. Screenshot Hasil Jawaban Soal 1 Berpindah ke Aplikasi Lain Saat Menekan Tombol Shopee



Gambar 6. Screenshot Hasil Jawaban Soal 1 UI Detail Saat Menekan Tombol Detail

C. Pembahasan

1. HomeFragment.kt

Pada line 1, `package com.example.modul3` digunakan untuk menentukan nama package dari aplikasi Android. Package ini adalah namespace unik untuk memisahkan file dalam aplikasi. Kemudian, pada line 3 sampai 10, class-class penting yang digunakan seperti `Fragment`, `LayoutInflater`, `RecyclerView`, dan binding untuk layout `FragmentHome`. Lalu, line 12 membuat fragment bernama `HomeFragment` yang merupakan turunan dari `Fragment`. Line 13 dan 14 menggunakan View Binding agar lebih aman dan mudah mengakses komponen UI. `_binding` digunakan untuk menyimpan objek binding yang diisi saat layout di-inflate. `binding` adalah versi non-null yang digunakan di dalam fragment. Line 16 sampai 22 membuat dan mengembalikan tampilan (UI) fragment. `FragmentHomeBinding.inflate(...)` digunakan untuk meng-inflate layout XML untuk fragment ini dan menyimpannya ke `_binding`. Setelah tampilan dibuat, data tas di-generate dan dimasukkan ke adapter yaitu pada line 24 sampai 32. Adapter diatur untuk `RecyclerView`, menggunakan `LinearLayoutManager` agar tampil vertikal. Line 34 sampai 37, menghapus referensi binding ketika tampilan dihancurkan untuk mencegah memory leak.

Pada line 39 sampai 44, mengambil resource string array dari `res/values/strings.xml` (atau file arrays lain) untuk nama tas, harga, deskripsi, dan link produk. Line 46 sampai 57, menyediakan daftar ID gambar tas dalam bentuk list. Line 59 sampai 73, menggabungkan data teks dan gambar menjadi objek `ItemTas`, lalu dikembalikan sebagai list yang akan ditampilkan di `RecyclerView`.

2. DetailFragment.kt

Pada line 1, `package com.example.modul3` digunakan untuk menentukan nama package dari aplikasi Android. Package ini adalah namespace unik untuk memisahkan file dalam aplikasi. Kemudian, pada line 3 sampai 9, class-class penting yang digunakan seperti `lifecycle fragment`, `log debugging`,

dan View Binding. Lalu, line 12 mendefinisikan fragment detail bernama `DetailFragment` sebagai turunan dari `Fragment`. Line 13 dan 14, view binding digunakan untuk mengakses elemen UI pada layout `fragment_detail.xml`. `_binding` adalah nullable, dan `binding` adalah versi non-null yang digunakan di dalam fragment. Line 16 sampai 22 membuat dan mengembalikan tampilan (UI) fragment. `FragmentDetailBinding.inflate(...)` digunakan untuk meng-inflate layout XML untuk fragment ini dan menyimpannya ke `_binding`. `binding.root` mengembalikan root view yang akan ditampilkan fragment. Line 24 dan 25, melakukan pemanggilan setelah view selesai dibuat. Umumnya dipakai untuk setup logic seperti isi data. Line 27, mencetak data `arguments` ke Logcat untuk debugging. Line 28 dan 29, mengambil data `ItemTas` yang dikirim melalui `Bundle` (via `Navigation Component`) dari fragment sebelumnya dan `ItemTas` harus implement `Parcelable`.

Pada line 30 sampai 36, jika data `tas` tidak null, isi elemen UI (`TextView`, `ImageView`) dengan data dari objek `ItemTas`. Line 38 sampai 41 terdapat fungsi opsional untuk mencari ID drawable berdasarkan nama string. Tidak digunakan langsung di kode ini, tapi berguna jika gambar ingin di-set dari nama string. Line 43 sampai 47, membersihkan binding saat tampilan dihancurkan agar tidak terjadi memory leak.

3. ItemTas.kt

Pada line 1, `package com.example.modul3` digunakan untuk menentukan nama package dari aplikasi Android. Package ini adalah namespace unik untuk memisahkan file dalam aplikasi. Line 3 dan 4 mengimpor `Parcelable`, yang di mana itu interface untuk mengirim data antar komponen Android (seperti antar fragment atau activity) dan `@Parcelize` adalah annotation dari Kotlin untuk membuat proses parceling otomatis tanpa harus menulis `writeToParcel()` dan `CREATOR`. Lalu, line 6 sampai 13 membuat class bisa dikirim lewat `Bundle` dengan mudah dengan cara `@Parcelize.data class` ialah class khusus di Kotlin untuk menyimpan data dengan otomatis menyediakan fungsi seperti `equals()`,

`hashCode()`, dan `toString()`. `ItemTas` menyimpan informasi untuk 1 item tas, nama untuk nama produknya, gambar untuk ID resource drawable gambar produk (`Int`), harga untuk harga produk (sebagai string, bisa mengandung Rp, titik, dsb), deskripsi untuk penjelasan produk di halaman detail, dan `link` adalah URL untuk ke Shopee atau marketplace lainnya. Fungsi utama dari class ini digunakan untuk menyimpan data tas dalam daftar dan mengirim data ke `DetailFragment` melalui `Bundle` secara aman dan efisien.

4. `AdapterTas.kt`

Pada line 1, `package com.example.modul3` digunakan untuk menentukan nama package dari aplikasi Android. Package ini adalah namespace unik untuk memisahkan file dalam aplikasi. Kemudian, pada line 3 sampai 11, class-class penting yang digunakan. Lalu, line 13 sampai 16 mendefinisikan class `AdapterTas`. `AdapterTas` menerima dua parameter, `listTas` ialah daftar objek `ItemTas` untuk ditampilkan dan `navController` untuk melakukan navigasi antar fragment (dalam hal ini ke `DetailFragment`). Adapter ini menghubungkan data dengan tampilan di `RecyclerView`. Line 18 dan 19, inner class `ViewHolder` merepresentasikan tiap item di dalam daftar `RecyclerView` dengan menggunakan View Binding (`ItemTasBinding`) untuk mengakses elemen UI. Line 21 sampai 24, mengisi teks dan gambar dari objek `ItemTas` ke komponen layout (`TextView`, `ImageView`).

Line 25 sampai 31 terdapat proses ketika tombol Detail diklik, maka data `tas` dimasukkan ke dalam `Bundle`, lalu navigasi ke `DetailFragment` dilakukan menggunakan `navController`, dengan membawa `bundle` tersebut. Line 32 sampai 37 melakukan proses pada tombol Shopee, ketika tombol Shopee diklik, maka `intent` dibuka untuk mengunjungi link eksternal (ke halaman Shopee). Line 39 sampai 42, membuat `ViewHolder` baru dari layout XML `item_tas.xml`. line 44 sampai 46, mengisi data ke `ViewHolder` berdasarkan posisi dalam list. Line 48 dan 49, mengembalikan jumlah total item dalam list.

5. MainActivity.kt

Pada line 1, `package com.example.modul3` digunakan untuk menentukan nama package dari aplikasi Android. Package ini adalah namespace unik untuk memisahkan file dalam aplikasi. Kemudian, pada line 3 sampai 5, mengimpor kelas yang dibutuhkan, diantaranya yaitu `Bundle` untuk menyimpan state saat Activity dibuat, `AppCompatActivity` ialah superclass dari semua activity modern, dan `ActivityMainBinding` ialah class auto-generated dari layout `activity_main.xml` (fitur View Binding). Line 7 mendeklarasikan `MainActivity`. `MainActivity` adalah *entry point* aplikasi dan activity utama yang pertama kali ditampilkan. Line 8, mendeklarasikan variabel binding menggunakan View Binding untuk mempermudah akses ke elemen layout tanpa `findViewById()`. Line 9 dan 10 terdapat fungsi lifecycle `onCreate()`, method ini dipanggil saat activity pertama kali dibuat. Line 11, menginisialisasi View Binding untuk `activity_main.xml`. Line 12 sampai 14, menetapkan tampilan layout utama ke root view dari View Binding.

6. fragment_home.xml

Pada line 1 sampai 8 terdapat `LinearLayout` yang di manakomponen dasar dengan orientasi vertikal untuk menyusun anak-anaknya dari atas ke bawah, `match_parent` digunakan untuk memenuhi seluruh lebar dan tinggi layar, dan `tools:context` hanya untuk keperluan preview di Android Studio (menunjukkan layout ini dipakai oleh `HomeFragment`). Line 10 sampai 19, menampilkan judul aplikasi di bagian atas, dengan background ungu (`#D1B3FF`), teks hitam, ukuran besar 25sp, tebal, dan padding 15dp agar teks tidak terlalu mepet. Kemudian, line 21 sampai 27 ialah tempat untuk menampilkan daftar produk tas menggunakan `RecyclerView`, juga menggunakan seluruh sisa ruang layar di bawah `TextView`. `id` tersebut yang diakses di `HomeFragment.kt` untuk menampilkan daftar tas melalui adapter.

7. item_tas.xml

Pada line 1 sampai 14 terdapat `CardView` (root) untuk membungkus seluruh isi item agar terlihat seperti kartu dengan bayangan. `cardCornerRadius="12dp"` untuk mengatur sudut membulat dan `cardElevation="10dp"` untuk memberikan efek bayangan agar item terlihat menonjol. Line 16 sampai 20 terdapat `ConstraintLayout` (Container dalam Card), layout fleksibel untuk mengatur posisi komponen UI berdasarkan kait (constraint) terhadap komponen lain atau parent-nya. Kemudian, line 22 sampai 31 menampilkan foto produk, dengan ukuran tetap `135dp x 140dp`, dan `scaleType="centerCrop"` agar gambar memenuhi tampilan dengan cropping bagian luar. Line 33 sampai 46, menampilkan nama tas. `textStyle="bold", textSize="20sp"` untuk mengatur teks nya agar nama terlihat jelas, dan ditempatkan di sebelah kanan gambar. Lalu, line 48 sampai 58 disitu berfungsi untuk menampilkan harga tas dan ditempatkan di bawah nama produk. Line 60 sampai 68 membuat tombol untuk berpindah ke marketplace Shopee. Dengan posisi di bawah harga dan di sebelah kiri. Constraint ke `@id/tv_item_price` (top) dan Constraint ke `@id/tv_item_name` (start). Line 70 sampai 79 membuat tombol untuk navigasi ke halaman detail produk. Dengan posisi di sebelah kanan tombol Shopee, menyatu sejajar atasnya.

8. fragment_detail.xml

Line 1 sampai 9 terdapat `<ScrollView>` untuk membungkus seluruh isi layout agar bisa di-scroll jika kontennya lebih panjang dari layar. ini sangat berguna untuk tampilan detail yang bisa panjang, seperti pada deskripsi produk. Line 11 sampai 14, `ConstraintLayout` (di dalam `ScrollView`) digunakan untuk menyusun posisi elemen-elemen UI dengan fleksibel. Pada line 16 sampai 25 menampilkan gambar produk ukuran besar (`350dp x 350dp`). `centerCrop` agar gambar tetap proporsional memenuhi area dan diposisikan di bagian atas tengah layout. Kemudian, line 27 sampai 38 menampilkan nama produk dengan ukuran besar (`25sp`) dan bold. Diposisikan di bawah gambar dan disejajarkan ke tengah. Line 40 sampai 51, menampilkan harga dengan warna merah gelap (`holo_red_dark`) dan bold. Diposisikan di bawah nama produk dan tetap di tengah. Lalu, line 53 sampai 70

menampilkan deskripsi lengkap produk. `lineSpacingExtra="5dp"` mengatur agar teks lebih nyaman dibaca. `textAlignment="center"` digunakan untuk isi teks diratakan ke tengah. `padding` dan `marginTop` cukup besar untuk memberi jarak visual. Serta lebar penuh (`match_parent`) agar deskripsi tidak terpotong.

9. nav_graph.xml

Pada line 1 sampai 5, menentukan bahwa fragment pertama kali yang ditampilkan saat app berjalan adalah `HomeFragment`. Line 7 sampai 11, mewakili untuk tampilan utama (daftar tas). Fragment ini berada di package `com.example.modul3`. Layout yang digunakan ialah `fragment_home.xml`. Line 12 sampai 15, mendefinisikan rute navigasi dari `HomeFragment` ke `DetailFragment`, hal ini akan dipicu saat user klik tombol “Detail” pada item tas. Line 17 sampai 22 adalah fragment untuk menampilkan detail dari item tas yang dipilih, dengan menggunakan layout `fragment_detail.xml`.

10. strings.xml

Pada line 1 sampai 4 terdapat `<resources>`, yang di mana semua string dan array harus berada di dalam tag `<resources>` tersebut. Fungsi-fungsinya terdiri dari `app_name` itu nama aplikasi. `btn_detail` dan `btn_shopee` adalah label pada tombol navigasi atau detail dan tombol ke Shopee. `label_nama` dan `label_harga` untuk label yang muncul pada halaman detail produk. `placeholder_nama` dan `placeholder_harga` ialah placeholder atau teks default saat input kosong atau belum ada data. Line 6 sampai 17 terdapat `<string-array name="data_name">`, yang di mana pada array string tersebut berisi nama-nama tas dari dua merek, yaitu BOSTANTEN dan MOSSDOOM. Kemudian, line 19 sampai 30 array string untuk `data_photo`, yang berisi referensi ke gambar tas di folder `res/drawable`, dengan menggunakan notasi `@drawable/nama_gambar`. Line 32 sampai 43 terdapat array string untuk `data_link`, link akan menuju produk yang bersangkutan di marketplace Shopee. Lalu, line 45 sampai 56 terdapat array string `data_harga`, yang di mana akan

menampilkan harga produk dalam Rupiah. Line 58 sampai 76 terdapat array string `data_deskripsi`, setiap `<item>` disitu menjelaskan fitur, ukuran, warna, dan target pengguna dari tas. Informasi tersebut sangat lengkap dan berorientasi pada e-commerce (penjelasan produk seperti di Shopee).

11. `activity_main.xml`

Pada line 1 sampai 9, `ConstraintLayout` adalah layout utama yang digunakan di sini untuk mengatur posisi elemen UI di dalamnya. `ConstraintLayout` memungkinkan penataan elemen UI dengan fleksibilitas tinggi melalui constraint (penghubung). `xmlns:android` adalah deklarasi namespace untuk atribut Android standar. `xmlns:tools` digunakan oleh Android Studio untuk memberikan data atau tips visual saat pengembangan tanpa mempengaruhi aplikasi yang sebenarnya. `tools:context=".MainActivity"` memberikan konteks di mana layout ini digunakan (dalam hal ini, `MainActivity`). `tools:ignore="ExtraText"` memberikan petunjuk kepada Android Studio untuk mengabaikan peringatan tertentu yang tidak penting (misalnya teks ekstra).

Kemudian, line 11 sampai 18 terdapat `FragmentContainerView`, yang di mana komponen ini digunakan untuk menampilkan dan mengelola fragment di dalam activity. `FragmentContainerView` menggantikan penggunaan `FrameLayout` untuk penempatan fragment dan lebih terintegrasi dengan `Navigation Component` Android. `android:id="@+id/nav_host_fragme"` memberikan ID untuk `FragmentContainerView`, yang nantinya bisa digunakan untuk referensi dalam kode.

`android:name="androidx.navigation.fragment.NavHostFragment"` untuk menyatakan bahwa fragment yang akan ditampilkan di dalam `FragmentContainerView` adalah `NavHostFragment`, yang merupakan bagian dari `Navigation Component`. Fragment ini bertugas mengelola navigasi antar fragment. `android:layout_width="match_parent"` dan `android:layout_height="match_parent"` digunakan untuk membuat

FragmentContainerView mengambil ruang penuh dari layar, dengan lebar dan tinggi `match_parent`, sehingga mengisi seluruh layout. `app:navGraph="@navigation/nav_graph"` untuk menyediakan NavGraph yang berisi alur navigasi antar fragment. `@navigation/nav_graph` adalah referensi ke file `nav_graph.xml`, yang mendefinisikan destinasi dan aksi antar fragment dalam aplikasi. `app:defaultNavHost="true"` digunakan untuk menandakan bahwa NavHostFragment ini adalah host utama untuk navigasi di activity ini. Artinya, fragment ini akan menangani back stack navigasi, dan tombol back perangkat akan berfungsi untuk kembali ke fragment sebelumnya (jika ada).

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/nrhikmahhh99/Pemrograman-Mobile>

SOAL 2

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

A. Pembahasan

RecyclerView masih digunakan dalam pengembangan Android meskipun Jetpack Compose dengan LazyColumn menawarkan pendekatan yang lebih modern dan ringkas. Berikut beberapa alasan mengapa RecyclerView masih digunakan:

1. Kompatibilitas dengan Proyek Legacy

Banyak aplikasi Android yang sudah ada dibangun menggunakan sistem tampilan berbasis XML. Mengganti seluruh arsitektur UI ke Jetpack Compose memerlukan waktu dan sumber daya yang signifikan. Oleh karena itu, RecyclerView tetap digunakan untuk mempertahankan stabilitas dan menghindari risiko yang terkait dengan migrasi besar-besaran.

2. Fleksibilitas dan Kustomisasi Tinggi

RecyclerView menawarkan fleksibilitas yang tinggi dalam menampilkan berbagai jenis item dan tata letak yang kompleks. Dengan dukungan untuk berbagai LayoutManager seperti GridLayoutManager dan StaggeredGridLayoutManager, RecyclerView memungkinkan pengembang untuk menciptakan antarmuka pengguna yang kompleks dan dinamis.

3. Performa dan Kontrol yang Lebih Besar

RecyclerView memberikan kontrol penuh kepada pengembang dalam hal pengelolaan tampilan dan data. Dengan menggunakan ViewHolder dan Adapter, pengembang dapat mengoptimalkan performa aplikasi, terutama saat menangani dataset yang besar. Meskipun LazyColumn juga efisien, RecyclerView memungkinkan optimasi manual yang lebih mendalam.

4. Dukungan untuk Fitur-Fitur Lanjutan

RecyclerView mendukung berbagai fitur lanjutan seperti animasi item, drag-and-drop, dan swipe-to-dismiss. Fitur-fitur ini memungkinkan pengembang untuk menciptakan pengalaman pengguna yang interaktif dan responsif, yang mungkin memerlukan upaya tambahan jika diimplementasikan menggunakan LazyColumn.

Referensi:

<https://medium.com/@dobri.kostadinov/jetpack-compose-lazycolumn-vs-view-system-recyclerview-18454b4738c8>

<https://developer.android.com/develop/ui/compose/migrate/migration-scenarios/recycler-view?hl=id>

<https://blog.stackademic.com/lazycolumn-vs-recyclerview-which-one-to-use-in-2025-4fed8226998b>

https://www.linkedin.com/pulse/recyclerview-xml-vs-lazycolumn-jetpack-compose-which-one-levindo-glaef?utm_source