

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 3**



**BUILD A SCROLLABLE LIST**

**Oleh:**

**Devi Hafida Ariyani**

**NIM. 2310817220018**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS LAMBUNG MANGKURAT  
APRIL 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 : Devi Hafida Ariyani  
NIM : 2310817220018

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Muhammad Raka Azwar  
NIM. 2210817210012

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

## DAFTAR ISI

LEMBAR PENGESAHAN .....	2
DAFTAR ISI .....	3
DAFTAR GAMBAR .....	4
DAFTAR TABEL .....	5
SOAL 1 .....	6
A. SOURCE CODE .....	7
B. OUTPUT PROGRAM .....	18
C. PEMBAHASAN .....	22
D. TAUTAN GIT .....	23
SOAL 2 .....	24

## DAFTAR GAMBAR

Gambar 1. Contoh UI List .....	6
Gambar 2. Contoh UI Detail .....	7
Gambar 3. Soal 1 (Compose).....	18
Gambar 4. Soal 1 (Compose).....	19
Gambar 5. Soal 1 (Compose).....	19
Gambar 6. Soal 1 (XML) .....	20
Gambar 7. Soal 1 (XML) .....	20
Gambar 8. Soal 1 (XML) .....	21
Gambar 9. Soal 1 (XML) .....	21

## DAFTAR TABEL

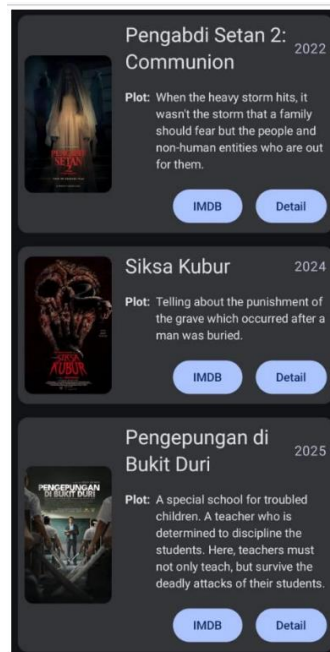
Tabel 1. Source Code Jawaban Soal 1 .....	7
Tabel 2. Source Code Jawaban Soal 1 .....	7
Tabel 3. Source Code Jawaban Soal 1 .....	8
Tabel 4. Source Code Jawaban Soal 1 .....	9
Tabel 5. Source Code Jawaban Soal 1 .....	9
Tabel 6. Source Code Jawaban Soal 1 .....	11
Tabel 7. Source Code Jawaban Soal 1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

## SOAL 1

### Soal Praktikum:

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

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



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

**Compose:**

**MainActivity.kt**

*Tabel 1. Source Code Jawaban Soal 1*

```
package com.example.listfilm

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.navigation.compose.rememberNavController
import com.example.listfilm.ui.screen.AppNavGraph

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            val navController = rememberNavController()
            AppNavGraph(navController)
        }
    }
}
```

**FilmData.kt**

*Tabel 2. Source Code Jawaban Soal 1*

```
package com.example.listfilm.data

import com.example.listfilm.R

val filmList = listOf(
    Film(
        title = "Crash Landing on You",
        description = "Romansa tak terduga antara pewaris Korea Selatan dan tentara Korea Utara.",
    )
)
```

```

        imageResId = R.drawable.crashlandingonyou,
        url = "https://youtu.be/GVQGWgeVc4k?feature=shared",
        genre = "Romance, Drama",
        year = 2019
    ),
    Film(
        title = "Descendants of the Sun",
        description = "Cinta antara dokter dan tentara di zona
perang.",
        imageResId = R.drawable.dots,
        url = "https://youtu.be/wTGwjDqtfzQ?feature=shared",
        genre = "Romance, Action",
        year = 2016
    ),
    Film(
        title = "Goblin",
        description = "Kisah makhluk abadi yang mencari pengantinnya
untuk mengakhiri kutukan.",
        imageResId = R.drawable.goblin, // Hapus ekstensi .jpg karena
di res sudah tanpa ekstensi
        url = "https://youtu.be/S94ukM8C17A?feature=shared",
        genre = "Fantasy, Romance",
        year = 2016
    ),
    Film(
        title = "Resident Playbook",
        description = "Kehidupan para dokter muda di rumah sakit penuh
tantangan.",
        imageResId = R.drawable.residentplaybook,
        url = "https://youtu.be/VTjJfJ5kWxUE?feature=shared",
        genre = "Medical, Drama",
        year = 2020
    ),
    Film(
        title = "Twenty Five Twenty One",
        description = "Cinta dan mimpi yang tumbuh di tengah krisis
tahun 1998.",
        imageResId = R.drawable.twentyfivetwentyone,
        url = "https://youtu.be/PBCXH0skDQ4?feature=shared",
        genre = "Romance, Drama",
        year = 2022
    )
)
)

```

### **Film.kt**

*Tabel 3. Source Code Jawaban Soal 1*

```

package com.example.listfilm.data

data class Film(
    val title: String,
    val year: Int,
    val genre: String,

```



```

        val description: String,
        val imageResId: Int,
        val url: String
    )

```

## FilmDetailScreen.kt

*Tabel 4. Source Code Jawaban Soal 1*

```

package com.example.listfilm.ui.screen

import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import com.example.listfilm.data.Film

@Composable
fun FilmDetailScreen(film: Film) {
    Column(modifier = Modifier.padding(16.dp)) {
        Image(
            painter = painterResource(id = film.imageResId),
            contentDescription = null,
            modifier = Modifier
                .fillMaxWidth()
                .height(200.dp)
                .clip(RoundedCornerShape(12.dp)),
            contentScale = ContentScale.Crop
        )
        Spacer(modifier = Modifier.height(16.dp))
        Text(text = film.title, style = MaterialTheme.typography.headlineSmall)
        Text(text = "${film.genre} • ${film.year}")
        Spacer(modifier = Modifier.height(8.dp))
        Text(text = film.description)
    }
}

```

## FilmListScreen.kt

*Tabel 5. Source Code Jawaban Soal 1*

```

package com.example.listfilm.ui.screen

import android.content.Intent
import android.net.Uri
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*

```

```

import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.navigation.NavHostController
import androidx.navigation.compose.rememberNavController
import androidx.core.net.toUri
import com.example.listfilm.data.filmList

@Composable
fun FilmListScreen(navController: NavHostController) {
    val context = LocalContext.current
    val films = filmList

    LazyColumn(contentPadding = PaddingValues(16.dp)) {
        items(films) { film ->
            Card(
                shape = RoundedCornerShape(16.dp),
                modifier = Modifier
                    .padding(bottom = 16.dp)
                    .fillMaxWidth()
            ) {
                Column(Modifier.padding(16.dp)) {
                    Image(
                        painterResource(id =
film.imageResId),
                        contentDescription = null,
                        modifier = Modifier
                            .fillMaxWidth()
                            .height(200.dp)
                            .clip(RoundedCornerShape(12.dp)),
                        contentScale = ContentScale.Crop
                    )
                    Spacer(Modifier.height(8.dp))
                    Row(
                        horizontalArrangement =
Arrangement.SpaceBetween,
                        modifier = Modifier.fillMaxWidth()
                    ) {
                        Text(text = film.title, fontWeight =
FontWeight.Bold)
                        Text(text = "${film.year}")
                    }
                }
            }
        }
    }

```

```

        Row(
            horizontalArrangement =
Arrangement.SpaceBetween,
            modifier = Modifier.fillMaxWidth()
        ) {
            Text(text = film.genre)
            Text(text = "Korean")
        }
        Spacer(Modifier.height(8.dp))
        Row(
            modifier = Modifier.fillMaxWidth(),
            horizontalArrangement =
Arrangement.SpaceBetween
        ) {
            Button(onClick = {
                val intent = Intent(Intent.ACTION_VIEW,
film.url.toUri())
                context.startActivity(intent)
            }) {
                Text("Open Site")
            }
            Button(onClick = {
navController.navigate("detail/${Uri.encode(film.title)}")
            }) {
                Text("Detail")
            }
        }
    }
}

@Preview(showBackground = true)
@Composable
fun FilmListScreenPreview() {
    val navController = rememberNavController()
    FilmListScreen(navController = navController)
}

```

## NavGraph.kt

*Tabel 6. Source Code Jawaban Soal 1*

```

package com.example.listfilm.ui.screen

import android.net.Uri
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.navigation.NavHostController
import androidx.navigation.compose.NavHost

```

```

import androidx.navigation.compose.composable
import com.example.listfilm.data.filmList

@Composable
fun AppNavGraph(navController: NavHostController) {
    val films = filmList // langsung properti filmList
    NavHost(navController = navController, startDestination = "list")
    {
        composable("list") {
            FilmListScreen(navController)
        }
        composable("detail/{title}") { backStackEntry ->
            val rawTitle = backStackEntry.arguments?.getString("title")
            val decodedTitle = rawTitle?.let { Uri.decode(it) }
            val film = films.find { it.title == decodedTitle }
            if (film != null) {
                FilmDetailScreen(film = film)
            } else {
                Text("Film not found", modifier =
                Modifier.fillMaxSize())
            }
        }
    }
}

```

## XML:

### Activity\_main.xml

*Tabel 7. Source Code Jawaban Soal 1*

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/mainLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvFilm"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="16dp"/>
</LinearLayout>

```

### activity\_detail.xml

*Tabel 8. Source Code Jawaban Soal 1*

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="16dp">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <ImageView
                android:id="@+id/imgDetail"
                android:layout_width="match_parent"
                android:layout_height="351dp"
                android:scaleType="centerCrop" />

            <TextView
                android:id="@+id/tvDetailTitle"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textSize="20sp"
                android:textStyle="bold"
                android:layout_marginTop="16dp" />

            <TextView
                android:id="@+id/tvDetailGenre"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textSize="14sp"
                android:textColor="#666"/>

            <TextView
                android:id="@+id/tvDetailDescription"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="8dp" />
        </LinearLayout>
    </ScrollView>

```

### item\_film.xml

*Tabel 9. Source Code Jawaban Soal 1*

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    card_view:cardCornerRadius="12dp"
    card_view:cardElevation="4dp"
    android:layout_marginBottom="16dp"
    tools:ignore="MissingClass">

```

```

<LinearLayout
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <ImageView
        android:id="@+id/imgFilm"
        android:layout_width="match_parent"
        android:layout_height="259dp"
        android:adjustViewBounds="true"
        android:scaleType="centerCrop" />

    <TextView
        android:id="@+id/tvTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="16sp"
        android:layout_marginTop="8dp"/>

    <TextView
        android:id="@+id/tvInfo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:textColor="#666"/>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:gravity="end">

        <Button
            android:id="@+id/btnOpenSite"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Open Site"/>

        <Button
            android:id="@+id/btnDetail"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Detail"
            android:layout_marginStart="8dp"/>
    </LinearLayout>

</LinearLayout>
</androidx.cardview.widget.CardView>

```

## DetailActivity.kt

Tabel 10. Source Code Jawaban Soal 1

```
package com.example.listfilm

import android.annotation.SuppressLint
import android.os.Bundle
import android.widget.ImageView
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.listfilm.data.filmList

class DetailActivity : AppCompatActivity() {

    @SuppressLint("SetTextI18n")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail)

        val title = intent.getStringExtra("film")
        if (title == null) {
            Toast.makeText(this, "Data film tidak diterima",
                Toast.LENGTH_SHORT).show()
            finish()
            return
        }

        val film = filmList.find { it.title == title }
        if (film == null) {
            Toast.makeText(this, "Film tidak ditemukan",
                Toast.LENGTH_SHORT).show()
            finish()
            return
        }
    }
}
```

```

    }

    findViewById<ImageView>(R.id.imgDetail).setImageResource(film.imageResId)

    findViewById<TextView>(R.id.tvDetailTitle).text = film.title

    findViewById<TextView>(R.id.tvDetailGenre).text = "${film.genre}
    • ${film.year}"

    findViewById<TextView>(R.id.tvDetailDescription).text =
    film.description

    }

}

```

### **Film.kt**

*Tabel 11. Source Code Jawaban Soal 1*

```

package com.example.listfilm.data

data class Film(
    val title: String,
    val year: Int,
    val genre: String,
    val description: String,
    val imageResId: Int,
    val url: String
)

```

### **FilmAdapter.kt**

*Tabel 12. Source Code Jawaban Soal 1*

```

package com.example.listfilm

import android.content.Context
import android.content.Intent
import android.net.Uri
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.*
import androidx.recyclerview.widget.RecyclerView
import com.example.listfilm.data.Film

class FilmAdapter(
    private val context: Context,
    private val listFilm: List<Film>,
    private val onDetailClick: (Film) -> Unit
) : RecyclerView.Adapter<FilmAdapter.FilmViewHolder>() {

    inner class FilmViewHolder(view: View) :
    RecyclerView.ViewHolder(view) {
        val imgFilm: ImageView = view.findViewById(R.id.imgFilm)
        val tvTitle: TextView = view.findViewById(R.id.tvTitle)
    }
}

```



```

        val tvInfo: TextView = view.findViewById(R.id.tvInfo)
        val btnOpenSite: Button = view.findViewById(R.id.btnOpenSite)
        val btnDetail: Button = view.findViewById(R.id.btnDetail)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
    FilmViewHolder {
        val view =
        LayoutInflater.from(parent.context).inflate(R.layout.item_film, parent,
        false)
        return FilmViewHolder(view)
    }

    override fun getItemCount(): Int = listFilm.size

    override fun onBindViewHolder(holder: FilmViewHolder, position: Int)
    {
        val film = listFilm[position]
        holder.imgFilm.setImageResource(film.imageResId)
        holder.tvTitle.text = film.title
        holder.tvInfo.text = "${film.genre} • ${film.year}"

        holder.btnOpenSite.setOnClickListener {
            val intent = Intent(Intent.ACTION_VIEW, Uri.parse(film.url))
            context.startActivity(intent)
        }

        holder.btnDetail.setOnClickListener {
            onDetailClick(film)
        }
    }
}

```

## MainActivity.kt

*Tabel 13. Source Code Jawaban Soal 1*

```

package com.example.listfilm

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.example.listfilm.data.filmList

class MainActivity : AppCompatActivity() {

    private lateinit var rvFilm: RecyclerView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}

```

```

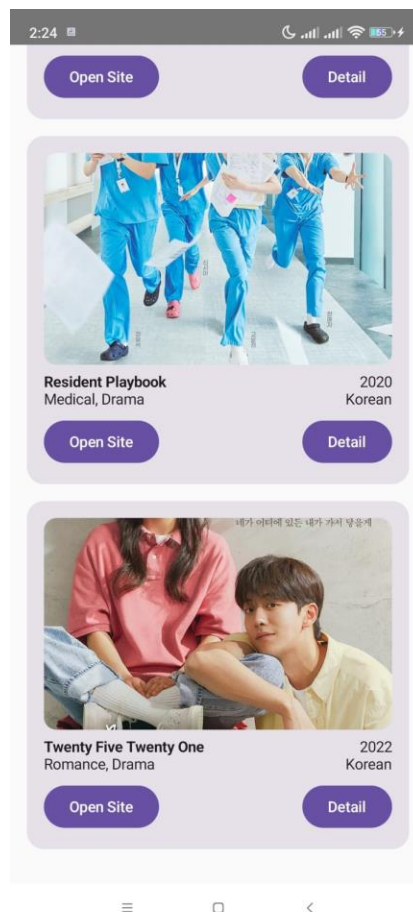
        rvFilm = findViewById(R.id.rvFilm)
        rvFilm.layoutManager = LinearLayoutManager(this)

        val adapter = FilmAdapter(this, filmList) { film ->
            val intent = Intent(this,
DetailActivity::class.java).apply {
                putExtra("film", film.title) // pakai title sebagai ID
            }
            startActivity(intent)
        }

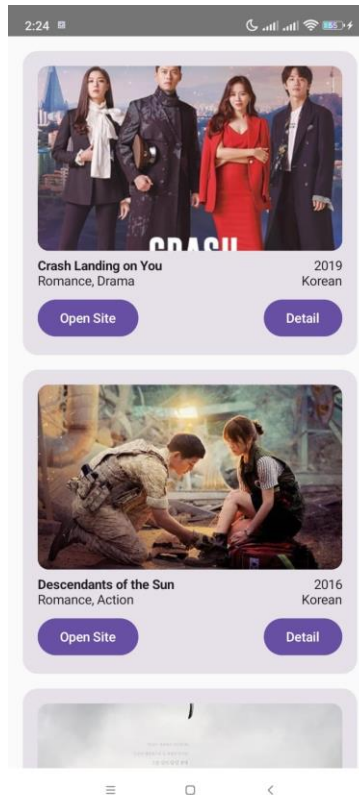
        rvFilm.adapter = adapter
    }
}

```

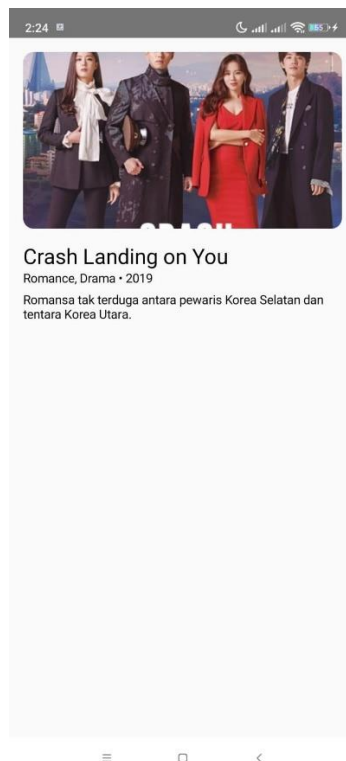
## B. OUTPUT PROGRAM



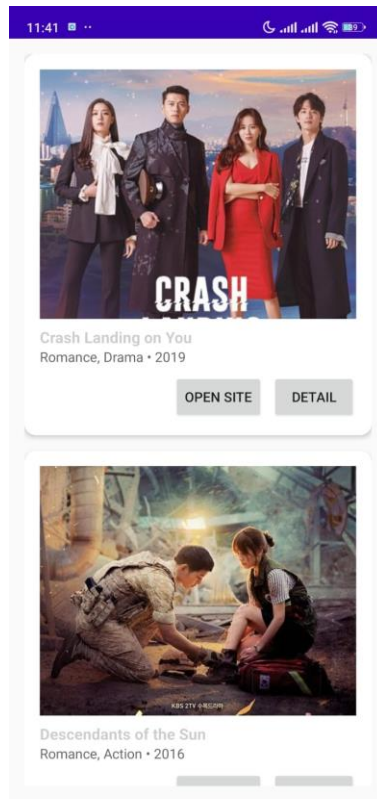
Gambar 3. Soal 1 (Compose)



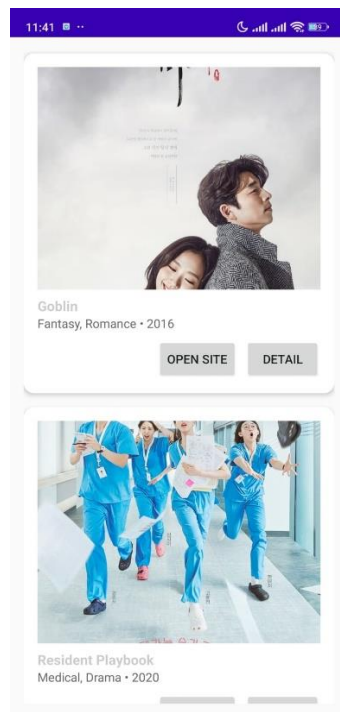
*Gambar 4. Soal 1 (Compose)*



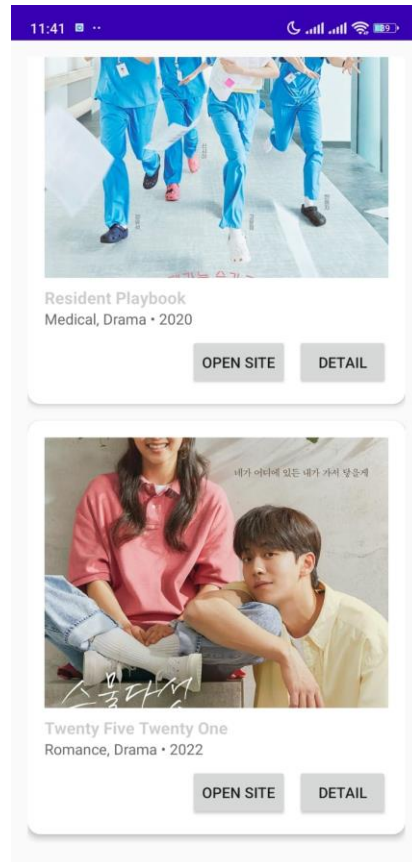
*Gambar 5. Soal 1 (Compose)*



Gambar 6. Soal 1(XML)



Gambar 7. Soal 1 (XML)



Gambar 8. Soal 1 (XML)



Gambar 9. Soal 1 (XML)

## **C. PEMBAHASAN**

### **XML (RecyclerView)**

#### **1. MainActivity.kt (XML)**

File ini merupakan activity utama untuk versi XML. Di dalamnya kamu menggunakan `setContentView(R.layout.activity_main)` untuk menampilkan layout `activity_main.xml`. Kamu membuat list data dari `DummyDataFilm`, lalu mengatur `RecyclerView` dan menghubungkannya ke `FilmAdapter`. Saat item film diklik, kamu mengirim data ke `DetailActivity.kt` melalui `Intent`.

#### **2. FilmAdapter.kt**

File ini adalah adapter kustom untuk `RecyclerView`. Ia menerima `List<Film>` dan membuat tampilan untuk tiap item menggunakan layout `item_film.xml`. Saat user mengklik salah satu film, adapter mengirim data ke `DetailActivity.kt` menggunakan `Intent`, jadi interaktif dan dinamis.

#### **3. DetailActivity.kt**

Activity ini menampilkan detail lengkap dari film yang dipilih. Ia menerima data film (judul, sutradara, tahun, genre, sinopsis, gambar) dari `Intent` dan menampilkannya dalam layout `activity_detail.xml`.

#### **4. activity\_main.xml**

Ini adalah layout utama dari `MainActivity` versi XML. Di dalamnya hanya terdapat satu elemen yaitu `RecyclerView` dengan ID `recyclerView`, tempat daftar film ditampilkan menggunakan adapter.

#### **5. item\_film.xml**

Layout ini digunakan oleh `FilmAdapter` untuk menampilkan satu item film di `RecyclerView`. Layout ini menampilkan gambar film (`ImageView`), judul (`TextView`), sutradara, dan tahun. Dibungkus dalam `CardView` agar terlihat rapi dan modern.

#### **6. activity\_detail.xml**

Layout ini digunakan oleh `DetailActivity` untuk menampilkan detail dari film yang dipilih. Di dalamnya ada `ImageView` untuk gambar film, dan beberapa `TextView` untuk menampilkan judul, sutradara, tahun, genre, dan sinopsis.

#### **7. Film.kt**

File ini adalah data class yang mendefinisikan struktur data film. Ia punya properti seperti judul, sutradara, tahun, genre, sinopsis, dan gambar. Digunakan oleh versi XML (`RecyclerView`) maupun `Jetpack Compose` untuk menyimpan dan mengelola data film.

### **JETPACK COMPOSE**

#### **1. MainActivity.kt (Compose)**

File ini adalah activity utama versi `Jetpack Compose`. Ia tidak menggunakan `setContentView`, melainkan `setContent` dari `Compose`. Ia memanggil fungsi `AppNavGraph()` untuk mengatur navigasi antar screen. File ini adalah titik masuk aplikasi `Compose`.

#### **2. AppNavGraph.kt**

File ini mengatur navigasi antar screen menggunakan NavController dan NavHost. Ia menentukan rute awal (film\_list) dan rute detail (film\_detail/{judul}). Parameter judul dikirim lewat route dan dipakai untuk menampilkan detail film di layar FilmDetailScreen.

3. FilmListScreen.kt

Composable ini menampilkan daftar film menggunakan LazyColumn, yang berfungsi seperti RecyclerView di Compose. Setiap item film ditampilkan dalam Card dengan gambar, judul, sutradara, dan tahun. Ketika diklik, NavController akan berpindah ke FilmDetailScreen.

4. FilmDetailScreen.kt

Composable ini menampilkan informasi detail dari film berdasarkan judul yang dikirim melalui navigasi. Ia mencari data film yang sesuai dari DummyDataFilm, lalu menampilkannya dengan elemen Compose seperti Text, Image, dan Column.

5. Film.kt

File ini adalah data class yang mendefinisikan struktur data film. Ia punya properti seperti judul, sutradara, tahun, genre, sinopsis, dan gambar. Digunakan oleh versi XML (RecyclerView) maupun Jetpack Compose untuk menyimpan dan mengelola data film.

6. FilmData.kt

File FilmData.kt dalam package com.example.listfilm.data berisi data statis berupa daftar film Korea dalam bentuk list objek Film, yang setiap itemnya mencakup judul (title), deskripsi (description), gambar (imageResId), link trailer YouTube (url), genre (genre), dan tahun rilis (year). Data ini disusun dengan listOf() dan disimpan dalam variabel filmList, yang kemudian digunakan sebagai sumber utama untuk menampilkan daftar film baik dalam tampilan XML menggunakan RecyclerView maupun dalam Jetpack Compose dengan LazyColumn. File ini berfungsi sebagai penyedia konten utama yang akan diakses oleh komponen tampilan untuk menampilkan informasi lengkap dari tiap film, termasuk gambar poster, genre, dan link trailer yang relevan.

#### **D. TAUTAN GIT**

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/DeviHafida/Mobile>

## SOAL 2

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

RecyclerView masih digunakan karena memiliki beberapa keunggulan yang tetap relevan dalam pengembangan aplikasi Android, meskipun LazyColumn menawarkan kode yang lebih singkat dalam Jetpack Compose. Berikut beberapa alasan mengapa RecyclerView masih populer:

1. Fleksibilitas dan Kontrol Lebih Detail
  - RecyclerView memungkinkan kustomisasi lebih mendalam, seperti penggunaan ItemDecoration, ItemAnimator, serta berbagai layout manager seperti GridLayoutManager dan StaggeredGridLayoutManager.
  - LazyColumn lebih terbatas dalam hal kustomisasi tampilan dibandingkan RecyclerView.
2. Kompatibilitas dengan View-based UI
  - Banyak aplikasi lama masih menggunakan XML dan View-based UI, sehingga RecyclerView tetap menjadi pilihan utama.
  - LazyColumn hanya bekerja dalam lingkungan Jetpack Compose, sementara RecyclerView dapat digunakan dalam proyek yang belum beralih ke Compose.
3. Efisiensi dalam Pengelolaan Data yang Kompleks
  - RecyclerView lebih optimal untuk skenario data yang kompleks, misalnya daftar dengan banyak tipe item (Multiple View Type).
  - LazyColumn lebih cocok untuk daftar sederhana yang tidak memerlukan banyak interaksi khusus dengan view holder.
4. Dukungan Library dan Integrasi
  - Banyak library pihak ketiga masih bergantung pada RecyclerView, seperti Paging3 untuk pemrosesan data paginasi.
  - Meskipun Compose semakin berkembang, ekosistem RecyclerView masih lebih luas dan mapan.
5. Pengendalian Scroll dan Animasi yang Lebih Kuat
  - RecyclerView mendukung berbagai fitur scroll canggih, termasuk snapping dengan SnapHelper.
  - LazyColumn masih terbatas dalam pengendalian scroll dibandingkan RecyclerView.