LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3

BUILD A SCROLLABLE LIST

Oleh:

Farisa Adelia NIM. 2110817120010

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2025

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN I
MODUL 2

Laporan Praktikum Pemrograman Mobile Modul 3: Build a Scrollable List ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Prakitkum ini dikerjakan oleh:

Nama Praktikan : Farisa Adelia
NIM : 2110817120010

Menyetujui,
Asisten Praktikum

Zulfa Auliya Akbar
NIM. 2210817210026

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muti`a Maulida S.Kom M.T.I
NIP. 19881027 201903 20 13

DAFTAR ISI

DAFTAR TABEL

SOAL 1

Buatlah sebuah aplikasi Android menggunakan XML atau Jetpack Compose yang dapat menampilkan list

A. Source Code

Tabel 1. Source Code MainActivity.kt

```
// Modul 3 - Build a Scrollable List (Jetpack Compose version)

// MainActivity.kt
package com.example.modul3scrollablelist

import android.content.Intent
import android.net.Uri
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.navigation.NavController
import androidx.navigation.NavHostController
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.rememberNavController
import
com.example.modul3scrollablelist.ui.theme.Modul3ScrollableListTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Modul3ScrollableListTheme {
                Surface(modifier = Modifier.fillMaxSize(), color =
MaterialTheme.colorScheme.background) {
                    MainScreen()
                }
            }
        }
    }
}

data class MovieItem(
    val id: Int,
    val title: String,
    val year: String,
    val plot: String,
    val imageResId: Int,
    val imdbLink: String
)

val sampleMovies = listOf(
    MovieItem(1, "Pengabdi Setan 2: Communion", "2022", "When the
heavy storm hits, it wasn’t the storm that a family should fear but
the people and 'non-human entities' who are out for them.",
R.drawable.pengabdi, "https://www.imdb.com"),
    MovieItem(2, "Siksa Kubur", "2024", "Tells about the punishment
of the grave which occurred after a man was buried.",
R.drawable.siksa, "https://www.imdb.com"),
    MovieItem(3, "Pengepungan di Bukit Duri", "2025", "A special
school for troubled children. A teacher who is determined to
discipline the students. Here, teachers must not only teach, but
survive the deadly attacks of their students.", R.drawable.bukitduri,
"https://www.imdb.com")
)

@Composable
fun MainScreen() {
    val navController = rememberNavController()
    NavHost(navController = navController, startDestination = "list")
{
        composable("list") { MovieListScreen(navController) }
        composable("detail/{movieId}") { backStackEntry ->
            val movieId =
backStackEntry.arguments?.getString("movieId")?.toIntOrNull()
            val movie = sampleMovies.find { it.id == movieId }
            if (movie != null) DetailScreen(movie)
        }
    }
}

@Composable
fun MovieListScreen(navController: NavController) {
    LazyColumn(modifier = Modifier.padding(8.dp)) {
        items(sampleMovies) { movie ->
            Card(
                shape = RoundedCornerShape(12.dp),
                modifier = Modifier
                    .padding(vertical = 8.dp)
                    .fillMaxWidth()
                    .background(MaterialTheme.colorScheme.surface)
            ) {
                Column(
                    modifier = Modifier
                        .padding(12.dp)
                        .fillMaxWidth()
                ) {
                    Image(
                        painter = painterResource(id =
movie.imageResId),
                        contentDescription = movie.title,
                        contentScale = ContentScale.Crop,
                        modifier = Modifier
                            .fillMaxWidth()
                            .height(180.dp)
                            .clip(RoundedCornerShape(8.dp))
                    )
                    Spacer(modifier = Modifier.height(8.dp))
                    Row(Modifier.fillMaxWidth(),
horizontalArrangement = Arrangement.SpaceBetween) {
                        Text(movie.title, fontWeight =
FontWeight.Bold)
                        Text(movie.year)
                    }
                    Spacer(modifier = Modifier.height(4.dp))
                    Text("Plot: ${movie.plot}", maxLines = 3)
                    Spacer(modifier = Modifier.height(8.dp))
                    Row(Modifier.fillMaxWidth(),
horizontalArrangement = Arrangement.SpaceEvenly) {
                        Button(onClick = {
                            val intent = Intent(Intent.ACTION_VIEW,
Uri.parse(movie.imdbLink))

navController.context.startActivity(intent)
                        }) {
                            Text("IMDB")
                        }
                        Button(onClick = {

navController.navigate("detail/${movie.id}")
                        }) {
                            Text("Detail")
                        }
                    }
                }
            }
        }
    }
}

@Composable
fun DetailScreen(movie: MovieItem) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp)
    ) {
        Image(
            painter = painterResource(id = movie.imageResId),
            contentDescription = movie.title,
            contentScale = ContentScale.Crop,
            modifier = Modifier
                .fillMaxWidth()
                .height(250.dp)
                .clip(RoundedCornerShape(12.dp))
        )
        Spacer(modifier = Modifier.height(16.dp))
        Text(movie.title, style =
MaterialTheme.typography.titleLarge)
        Text("${movie.year}", style =
MaterialTheme.typography.labelMedium)
        Spacer(modifier = Modifier.height(8.dp))
        Text("Plot:", fontWeight = FontWeight.Bold)
        Text(movie.plot)
    }
}
```

Tabel 2. Source Code AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Modul3ScrollableList"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.Modul3ScrollableList">
            <intent-filter>
                <action  android:name="android.intent.action.MAIN"  />

                <category
android:name="android.intent.category.LAUNCHER"                     />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

B. Pembahasan

- **Arsitektur Single Activity**

Aplikasi menggunakan NavController dan NavHost untuk menavigasi antar layar (daftar dan detail). Hal ini memungkinkan implementasi **Single Activity Architecture**, yang lebih ringan dan modern.

- **LazyColumn**

- ➢ LazyColumn digunakan untuk membuat list scrollable.
- ➢ Setiap elemen didesain dengan **Card** agar terlihat rapi dan profesional.
- ➢ Gambar diformat menggunakan ContentScale.Crop dan dibungkus dengan RoundedCornerShape.

- **Intent Ekspilisit**
- Tombol “IMDB” menerapkan **explicit intent** untuk membuka link IMDb menggunakan Intent.ACTION_VIEW.

- **Navigasi ke Detail**

- ➢ Tombol “Detail” menavigasi pengguna ke layar DetailScreen dengan navController.navigate("detail/${movie.id}").
- ➢ Data diambil berdasarkan movieId yang diparsing dari argumen navigasi.

- **Layout Responsif**

- ➢ Modifier seperti fillMaxWidth(), padding(), dan clip() digunakan agar tampilan tetap stabil dan menarik pada orientasi portrait maupun landscape.

- **AndroidManifest.xml**

- ➢ Sudah dikonfigurasi dengan benar untuk MainActivity.
- ➢ Menyediakan intent-filter untuk mendeklarasikan MAIN dan LAUNCHER.

SOAL 2

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

JAWABAN : RecyclerView tetap digunakan karena fleksibel, stabil, dan memiliki ekosistem serta fitur lanjutan yang lebih matang, meskipun secara sintaksis lebih kompleks daripada LazyColumn di Jetpack Compose.

LINK GITHUB : https://github.com/farisadelia/Pemrograman-Mobile