

**LAPORAN PRAKTIKUM  
PEMROGRAMAN MOBILE  
MODUL 3**



**BUILD A SCROLLABLE LIST**

**Oleh:**

**Avantio Fierza Patria NIM. 2310817310001**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS LAMBUNG MANGKURAT  
APRIL 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 : Avantio Fierza Patria  
NIM : 2310817310001

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.....	i
DAFTAR ISI .....	ii
DAFTAR GAMBAR.....	iii
DAFTAR TABEL .....	iv
Soal Praktikum: .....	1
A. Source Code menggunakan Jetpack Compose .....	2
B. Source Code menggunakan XML .....	3
C. Output Program .....	4
D. Pembahasan .....	6
E. Tautan GIT .....	7

## DAFTAR GAMBAR

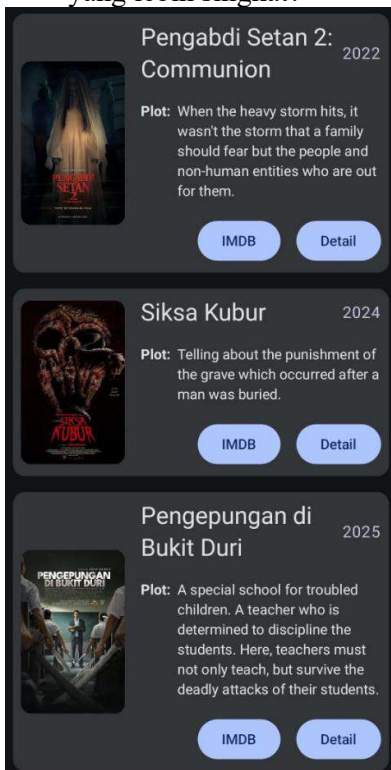
Gambar 1. 1 Contoh UI List .....	1
Gambar 1. 2 Contoh UI Detail.....	2
Gambar 1. 3 Screenshot Tampilan UI List.....	4
Gambar 1. 4 Screenshot Tampilan UI Detail .....	5

## DAFTAR TABEL

Tabel 1. 1 Source Code Jawaban Soal 1 MainActivity.kt Jetpack Compose .....	2
Tabel 1. 2 Tabel 1. 2 Source Code Jawaban Soal 1 MainActivity.kt XML .....	3
Tabel 1. 3 Source Code Jawaban Soal 1 activity_maint .XML .....	3

### Soal Praktikum:

1. 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
2. Mengapa RecyclerView masih digunakan, padahal RecyclerView memiliki kode yang panjang dan bersifat boiler-plate, dibandingkan LazyColumn dengan kode yang lebih singkat?



Gambar 1. 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 1. 2 Contoh UI Detail

#### A. Source Code menggunakan Jetpack Compose MainActivity.kt

```
1 package com.example.scrollablelist
2
3 import android.os.Bundle
4 import androidx.activity.ComponentActivity
5 import androidx.activity.compose.setContent
6 import com.example.scrollablelist.navigation.NavGraph
7 import com.example.scrollablelist.ui.theme.ScrollableListTheme
8
9 class MainActivity : ComponentActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContent {
13             ScrollableListTheme {
14                 NavGraph()
15             }
16         }
17     }
18 }
```

Tabel 1. 1 Source Code Jawaban Soal 1 MainActivity.kt Jetpack Compose

## B. Source Code menggunakan XML

### MainActivity.kt

```

1 package com.example.scrollablelist2
2
3 import android.os.Bundle
4 import androidx.appcompat.app.AppCompatActivity
5 import
6 com.example.scrollablelist2.databinding.ActivityMainBinding
7
8 class MainActivity : AppCompatActivity() {
9     private lateinit var binding: ActivityMainBinding
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         binding = ActivityMainBinding.inflate(layoutInflater)
14         setContentView(binding.root)
15     }
16 }

```

Tabel 1. 2 Tabel 1. 2 Source Code Jawaban Soal 1 MainActivity.kt XML

### activity\_main.xml

```

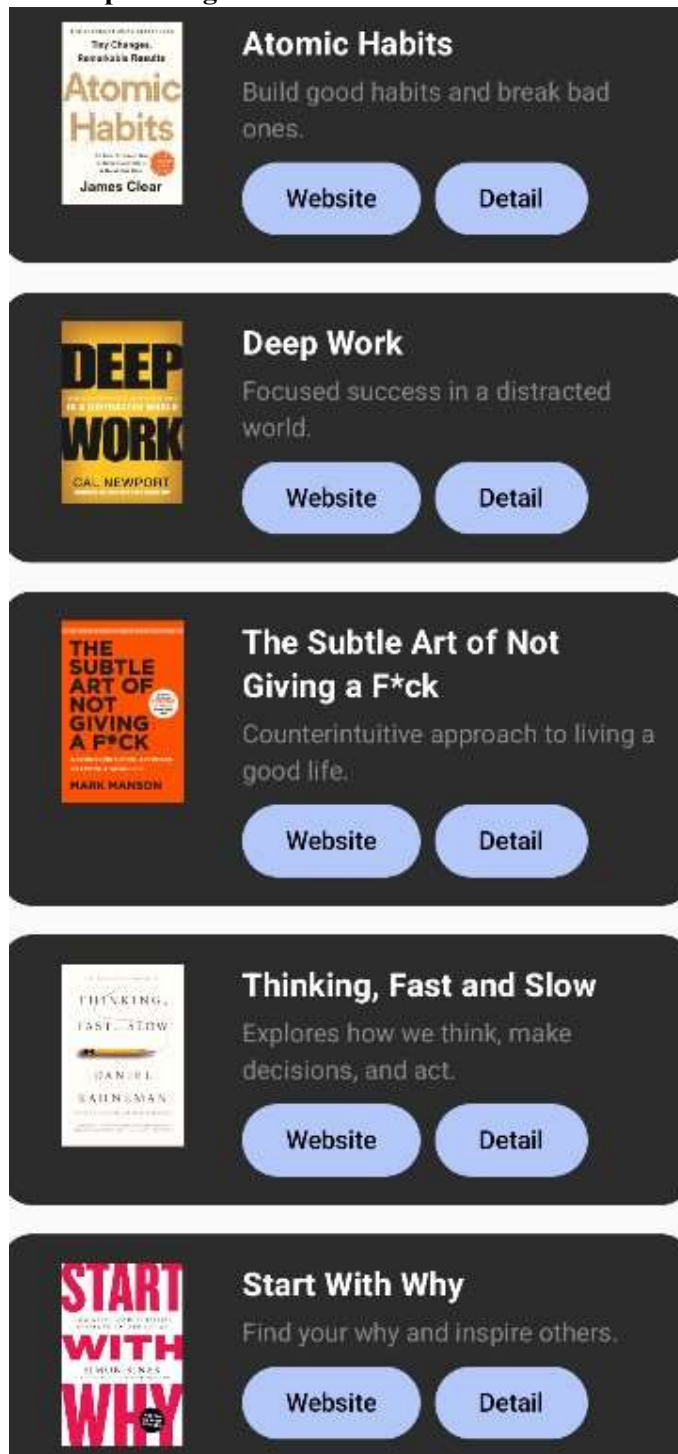
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/mainLayout"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     tools:context=".MainActivity">
10
11     <!-- Tempat untuk menampilkan fragment -->
12     <androidx.fragment.app.FragmentContainerView
13         android:id="@+id/nav_host_fragment"
14
15         android:name="androidx.navigation.fragment.NavHostFragment"
16         android:layout_width="0dp"
17         android:layout_height="0dp"
18         app:defaultNavHost="true"
19         app:navGraph="@navigation/nav_graph"
20         app:layout_constraintTop_toTopOf="parent"
21         app:layout_constraintBottom_toBottomOf="parent"
22         app:layout_constraintStart_toStartOf="parent"
23         app:layout_constraintEnd_toEndOf="parent" />
24 </androidx.constraintlayout.widget.ConstraintLayout>

```

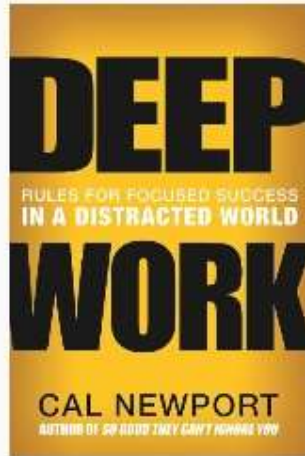
Tabel 1. 3 Source Code Jawaban Soal 1 activity\_maint .XML



## C. Output Program



Gambar 1. 3 Screenshot Tampilan UI List



## **Deep Work**

Deep Work is a book about the benefits of intense focus and how to systematically train your mind and habits to achieve it.

*Gambar 1. 4 Screenshot Tampilan UI Detail*

## D. Pembahasan

### MainActivity.kt Jetpack Compose:

- `ComponentActivity` digunakan karena Jetpack Compose tidak memerlukan `AppCompatActivity`.
- `setContent { ... }` adalah titik masuk untuk Compose UI.
- `NavGraph()` adalah fungsi Compose yang mengatur navigasi antar layar (HomeScreen, DetailScreen).
- Tidak ada layout XML: Semua UI dibangun langsung dengan fungsi-fungsi Compose (`@Composable`).
- Tema (`ScrollableListTheme`) diterapkan langsung lewat kode.

### MainActivity.kt XML:

- `AppCompatActivity` digunakan karena UI berbasis pada XML dan `ViewBinding` atau `findViewById`.
- `setContentView(...)` memuat layout dari file XML (`activity_main.xml`).
- Navigasi dan fragment biasanya dikelola melalui `NavHostFragment`, `FragmentManager`, dan `NavController`.

## Alasan RecyclerView Masih Digunakan

### 1. Kompatibilitas dan Basis Kode Lama (Legacy Projects)

Banyak aplikasi besar dan perusahaan sudah lama menggunakan *RecyclerView*, dan mereka tidak bisa langsung memigrasi ke Compose karena:

- Ukuran proyek yang besar
- Risiko tinggi perubahan UI/UX
- Tim belum semua menguasai Compose

### 2. Ekosistem yang Sudah Matang

*RecyclerView* punya:

- Dukungan kuat dari berbagai pustaka eksternal
- Banyak contoh, dokumentasi, Stack Overflow support
- Komponen tambahan seperti *ItemTouchHelper*, *DiffUtil*, dll.

### 3. Kontrol Lebih Rinci

Dengan *RecyclerView.Adapter*, kamu bisa:

- Mengelola jenis tampilan berbeda (multiple view types)
- Menyesuaikan performa scrolling dengan granularitas tinggi
- Menggunakan fitur seperti prefetching, caching, animasi kompleks

#### 4. Stabilitas Produksi

◆ *RecyclerView* telah teruji di banyak aplikasi produksi selama bertahun-tahun, sehingga dianggap stabil dan aman.

#### 5. Beberapa Developer Belum Pindah ke Compose

◆ Compose masih terbilang baru (stabil sejak 2021). Adopsi Compose masih berlangsung secara bertahap:

- Sebagian tim masih menggunakan View Binding/XML
- Butuh pelatihan ulang dan refactor besar

Kenapa LazyColumn Tetap Lebih Baik untuk Proyek Baru?

- Kode lebih pendek, bersih, dan deklaratif
- Tidak perlu *ViewHolder*, *Adapter*, *notifyDataSetChanged*, dsb
- Lebih mudah mengimplementasikan animasi dan state
- Integrasi erat dengan Compose UI lainnya

#### E. Tautan GIT

**<https://github.com/gr1ff0m/Pemrograman-Mobile-Praktikum>**