LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3

BUILD A SCROLLABLE LIST

Oleh:

Aiko Anatasha Wendiono NIM. 2310817320013

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 : Aiko Anatasha Wendiono
NIM : 2310817320013

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

DAFTAR GAMBAR

DAFTAR TABEL

SOAL 1

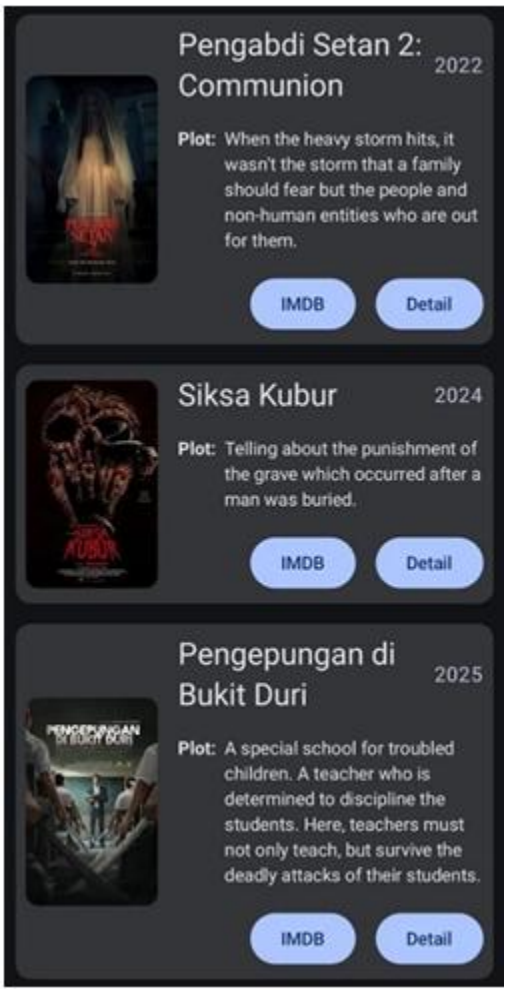
Soal Praktikum:

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

1. List menggunakan fungsi RecylerView (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. Terdpaat 2 button dalam list, dengan fungsi sebagai berikut:
    a. Button pertama menggunakan intent eksplisit untuk membuka apliaksi 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/di rotasi, baik ke potrait maupun landscape, aplikasi responsif dan dpaat 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?

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 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. MainActivity

```
1  package com.example.mlacharacters
2
3  import android.os.Bundle
4  import androidx.activity.ComponentActivity
5  import androidx.activity.compose.setContent
6  import androidx.compose.material3.MaterialTheme
7  import androidx.compose.material3.Surface
8  import
9  androidx.navigation.compose.rememberNavController
10 import
11 com.example.mlacharacters.ui.theme.MLACharactersTheme
12
13 class MainActivity : ComponentActivity() {
14     override fun onCreate(savedInstanceState: Bundle?)
15 {
16         super.onCreate(savedInstanceState)
17         setContent {
18             MLACharactersTheme {
19                 Surface(color                          =
20 MaterialTheme.colorScheme.background) {
21                     val         navController         =
22 rememberNavController()
```

```
                    NavGraph(navController =
navController)
                }
            }
        }
    }
}
```

Tabel 1. Source Code

2. Character

```
package com.example.mlacharacters

data class Character(
    val name: String,
    val alias: String,
    val imageRes: Int,
    val description: String,
    val wikiUrl: String
)
```

Tabel 2. Source Code

3. CharacterDetailScreen.kt

```
package com.example.mlacharacters

import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.material.icons.Icons
import
androidx.compose.material.icons.filled.ArrowBack
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
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.compose.ui.unit.sp
import androidx.navigation.NavController

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun    CharacterDetailScreen(character:    Character,
navController: NavController) {
    Scaffold(
```

```
        containerColor = Color(0xFF1C1C1E),
        topBar = {
            CenterAlignedTopAppBar(
                title = {},
                navigationIcon = {
                    IconButton(onClick       =       {
navController.popBackStack() }) {
                        Icon(
                            imageVector              =
Icons.Default.ArrowBack,
                            contentDescription       =
"Back",
                            tint = Color.White
                        )
                    }
                },
                colors                                =
TopAppBarDefaults.centerAlignedTopAppBarColors(
                    containerColor = Color.Transparent
                )
            )
        }
    ) { innerPadding ->
        LazyColumn(
            modifier = Modifier
                .padding(innerPadding)
                .fillMaxSize()
        ) {
            item {
                Image(
                    painter  =  painterResource(id  =
character.imageRes),
                    contentDescription               =
character.name,
                    contentScale = ContentScale.Crop,
                    modifier = Modifier
                        .fillMaxWidth()
                        .height(500.dp)
                )
            }

            item {
                Column(modifier                      =
Modifier.padding(16.dp)) {
                    Row(
                        modifier                     =
Modifier.fillMaxWidth(),

```

```
                        horizontalArrangement = Arrangement.SpaceBetween
                    ) {
                        Text(
                            text = character.name,
                            color = Color.White,
                            style = MaterialTheme.typography.titleLarge,
                            fontWeight = FontWeight.Bold
                        )
                        Text(
                            text = "2016",
                            color = Color.LightGray,
                            style = MaterialTheme.typography.bodyMedium
                        )
                    }

                    Spacer(modifier = Modifier.height(8.dp))

                    Text(
                        text = "Deskripsi:",
                        color = Color.White,
                        fontWeight = FontWeight.Bold,
                        style = MaterialTheme.typography.bodyMedium
                    )

                    Spacer(modifier = Modifier.height(4.dp))

                    Text(
                        text = character.description,
                        color = Color(0xFFCCCCCC),
                        style = MaterialTheme.typography.bodySmall,
                        lineHeight = 20.sp
                    )
                }
            }
        }
    }
}

```

Tabel 3. Source Code

4. CharacterListScreen.kt

```
1  package com.example.mlacharacters
2
3  import androidx.compose.foundation.Image
4  import androidx.compose.foundation.layout.*
5  import androidx.compose.foundation.lazy.LazyColumn
6  import androidx.compose.foundation.lazy.items
7  import androidx.compose.material3.*
8  import androidx.compose.runtime.Composable
9  import
10 androidx.compose.foundation.shape.RoundedCornerShape
11 import androidx.compose.ui.text.font.FontWeight
12 import androidx.compose.ui.Alignment
13 import androidx.compose.ui.Modifier
14 import androidx.compose.ui.graphics.Color
15 import androidx.compose.ui.layout.ContentScale
16 import androidx.compose.ui.res.painterResource
17 import androidx.compose.ui.unit.dp
18 import androidx.navigation.NavController
19 import android.content.Intent
20 import android.net.Uri
21 import androidx.compose.ui.platform.LocalContext
22
23 @OptIn(ExperimentalMaterial3Api::class)
24 @Composable
25 fun  CharacterListScreen(characters:  List<Character>,
26 navController: NavController) {
27     LazyColumn(
28
29         modifier = Modifier
30             .fillMaxSize()
31             .padding(8.dp),
32
33         contentPadding = PaddingValues(top = 60.dp),
34         verticalArrangement                              =
35 Arrangement.spacedBy(12.dp)
36
37     ) {
38         items(characters) { character ->
39             Card(
40                 shape = RoundedCornerShape(12.dp),
41                 colors                                   =
42 CardDefaults.cardColors(containerColor                   =
43 Color(0xFF2B2B2B)),
44                 elevation                                =
45 CardDefaults.cardElevation(4.dp),
46                 modifier = Modifier.fillMaxWidth()
47
48
49
50
```

```
            ) {
                Row(modifier                                =
Modifier.padding(12.dp)) {
                    Image(
                        painter = painterResource(id =
character.imageRes),
                        contentDescription          =
character.name,
                        contentScale                =
ContentScale.Crop,

                        modifier = Modifier
                            .width(90.dp)
                            .height(140.dp)
                            .padding(end = 12.dp)

                    )

                    Column(modifier                 =
Modifier.weight(1f)) {
                        Row(
                            modifier                =
Modifier.fillMaxWidth(),
                            horizontalArrangement   =
Arrangement.SpaceBetween
                        ) {
                            Text(
                                text = character.name,
                                style               =
MaterialTheme.typography.titleMedium,
                                color = Color.White,
                                fontWeight          =
FontWeight.Bold
                            )
                            Text(
                                text = "2016",
                                style               =
MaterialTheme.typography.labelMedium,
                                color               =
Color.LightGray
                            )
                        }

                        Spacer(modifier             =
Modifier.height(6.dp))

                        Row(

```

```
                                modifier                    =
Modifier.fillMaxWidth(),
                                verticalAlignment           =
Alignment.Top
                            ) {
                                Text(
                                    text = "Deskripsi: ",
                                    color = Color.White,
                                    fontWeight              =
FontWeight.Bold,
                                    style                   =
MaterialTheme.typography.bodyMedium
                                )

                                Spacer(modifier             =
Modifier.width(8.dp))

                                Text(
                                    text                    =
character.description,
                                    color = Color.White,
                                    style                   =
MaterialTheme.typography.bodySmall,
                                    modifier                =
Modifier.weight(1f)
                                )
                            }

                            Spacer(modifier                 =
Modifier.height(10.dp))

                            Row(
                                horizontalArrangement       =
Arrangement.spacedBy(8.dp)
                            ) {
                                val        context          =
LocalContext.current

                                Button(
                                    onClick = {
                                        val    intent       =
Intent(Intent.ACTION_VIEW,
Uri.parse(character.wikiUrl))

context.startActivity(intent)
                                    },

```

```
                                    colors                  =
ButtonDefaults.buttonColors(containerColor              =
Color(0xFF9BB1EB)),
                                    modifier                =
Modifier.weight(1f)
                                ) {
                                    Text("Detail    Hero",
color = Color.White)
                                }

                                Button(
                                    onClick = {

navController.navigate("detail/${character.name}")
                                    },
                                    colors                  =
ButtonDefaults.buttonColors(containerColor              =
Color(0xFF9BB1EB)),
                                    modifier                =
Modifier.weight(1f)
                                ) {
                                    Text("Deskripsi",
color = Color.White)
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}


```

Tabel 4. Source Code

5. NavGraph.kt

```
package com.example.mlacharacters

import androidx.compose.runtime.Composable
import androidx.navigation.NavHostController
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable

@Composable
fun NavGraph(navController: NavHostController) {
    val characters = listOf(

```

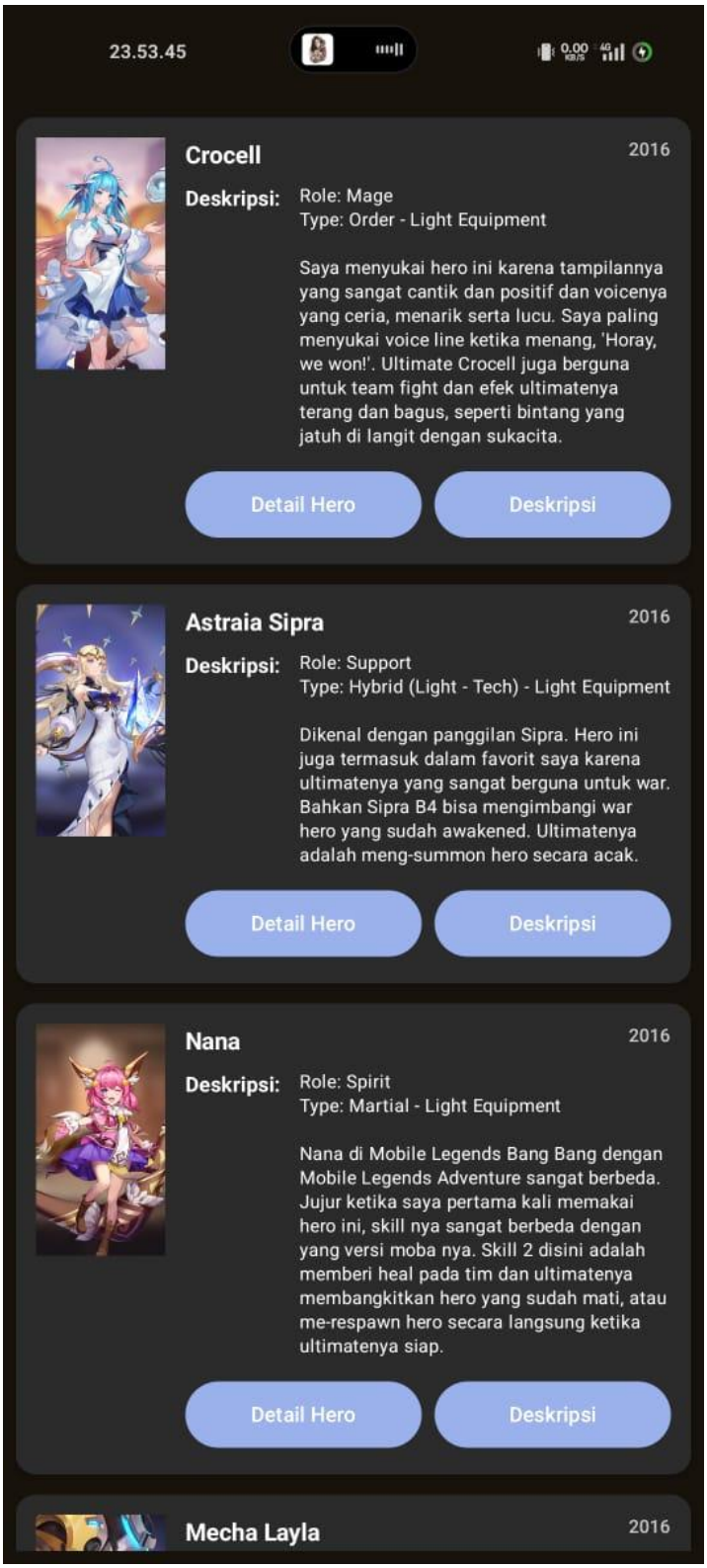
```
        Character("Crocell", "Mage, Order: Light Equipment", R.drawable.crocell, "Role: Mage\nType: Order - Light Equipment\n\nSaya menyukai hero ini karena tampilannya yang sangat cantik dan positif dan voicenya yang ceria, menarik serta lucu. Saya paling menyukai voice line ketika menang, 'Horay, we won!'. Ultimate Crocell juga berguna untuk team fight dan efek ultimatenya terang dan bagus, seperti bintang yang jatuh di langit dengan sukacita.", "https://mla.fandom.com/wiki/Crocell"),
        Character("Astraia Sipra", "Support, Hybrid, Light - Tech: Light Equipment", R.drawable.sipra, "Role: Support\nType: Hybrid (Light - Tech) - Light Equipment\n\nDikenal dengan panggilan Sipra. Hero ini juga termasuk dalam favorit saya karena ultimatenya yang sangat berguna untuk war. Bahkan Sipra B4 bisa mengimbangi war hero yang sudah awakened. Ultimatenya adalah meng-summon hero secara acak.", "https://mla.fandom.com/wiki/Astraia_Sipra"),
        Character("Nana", "Spirit, Martial: Light Equipment", R.drawable.nana, "Role: Spirit\nType: Martial - Light Equipment\n\nNana di Mobile Legends Bang Bang dengan Mobile Legends Adventure sangat berbeda. Jujur ketika saya pertama kali memakai hero ini, skill nya sangat berbeda dengan yang versi moba nya. Skill 2 disini adalah memberi heal pada tim dan ultimatenya membangkitkan hero yang sudah mati, atau me-respawn hero secara langsung ketika ultimatenya siap.", "https://mla.fandom.com/wiki/Nana"),
        Character("Mecha Layla", "Marksman, Hybrid (Light - Tech): Light Equipment", R.drawable.mechalayla, "Role: Marksman\nType: Hybrid (Light - Tech) - Light Equipment\n\nLayla di MLA ada 2 versi, yang pertama adalah Layla dan yang kedua adalah Mecha Layla. Hero menurut saya juga cukup meta karena memiliki kemampuan heal kepada tim, dan pasifnya yang tidak dapat diserang oleh lawan (invisible). Jadi, Mecha Layla akan memberi damage dan heal secara terus-menerus tanpa tereteksi oleh musuh. Namun, ketika semua hero di tim telah lenyap, maka Mecha Layla juga ikut lenyap", "https://mla.fandom.com/wiki/Mecha_Layla"),
        Character("Natsu", "Fighter, Hybrid (Dark - Martial): Medium Equipment", R.drawable.natsu, "Role: Fighter\nType:Hybrid (Dark - Martial) - Medium Equipment\n\nNatsu, hero kolaborasi dengan anime Fairy Tail. Natsu dan Lucy dengan status card SSR, dan Erza dengan status card UR. Saya mendapatkan card Natsu
```

```
secara gratis melalui event, hanya cukup menjalankan quest saya bisa mendapatkan Natsu dengan B5 merah (belum awakened). Yang saya sukai dari hero ini adalah ketika posisi back line diserang, Natsu dengan cepat menghampiri dan menyerang serta ultimatenya yang mendorong semua musuh ke kanan, dimana ini sangat menguntungkan tim terutama jika dikombo dengan Alice atau hero tipe AoE. Entah ini kebetulan atau bukan, tapi saya juga seringkali melihat ketika Natsu mengeluarkan ultimate pada HP di bawah 15%, HP dia akan terisi secara penuh kembali (kekebalan tubuh)",
"https://mla.fandom.com/wiki/Natsu"),
        Character("Naiad Rafaela", "Hybrid (Light - Elemental): Light Equipment", R.drawable.rafaela, "Role:Support\nType: Hybrid (Light - Elemental) - Light Equipment\n\nVersi lain dari hero Rafaela. Saya tidak memperhatikan skill-skillnya apakah berguna untuk war atau tidak. Namun, ketika saya pertama kali melihat hero SSR ini, saya langsung menyukainya karena desainnya yang begitu cantik, anggun, elegan, dan sangat nature (menyatu dengan alam). ",
"https://mla.fandom.com/wiki/Naiad_Rafaela"),
        Character("Shah Torre", "Mage, Chaos: Light Equipment", R.drawable.torre, "Role: Mage\nType: Chaos - Light Equipment\n\nShah Torre, hero yang sangat menyebalkan jika menjadi musuh. Ultimate hero ini menyerap bar energi, dimana ketika bar energi sudah penuh, maka hero akan mengeluarkan ultimate mereka. Namun, dengan ultimate Torre ini maka ultimate hero lawan akan tertunda karena ia menyerap bar energi pada semua hero.",
"https://mla.fandom.com/wiki/Shah_Torre"),
        Character("Forseti", "Marksman, Order: Light Equipment", R.drawable.forseti, "Role: Marksman\nType: Order - Light Equipment\n\nSaya baru tahu ada hero ini ketika Tower of Babel khusus hero Order dibuka. Awalnya saya tidak menaruh harapan yang tinggi kepada hero ini, namun ketika melihat ia MVP di hampir semua match, saya memperhatikan skill-skillnya. Yang menarik dari hero ini adalah ultimatenya, ketika Forseti mengeluarkan ultimate maka ia akan mati. Jadi, hero ini kurang cocok jika tidak ada hero lain yang bisa me-respawn hero tim, seperti Nana atau Singularity Lunox.",
"https://mla.fandom.com/wiki/Forseti"),

    )
    val charMap = characters.associateBy { it.name }

```

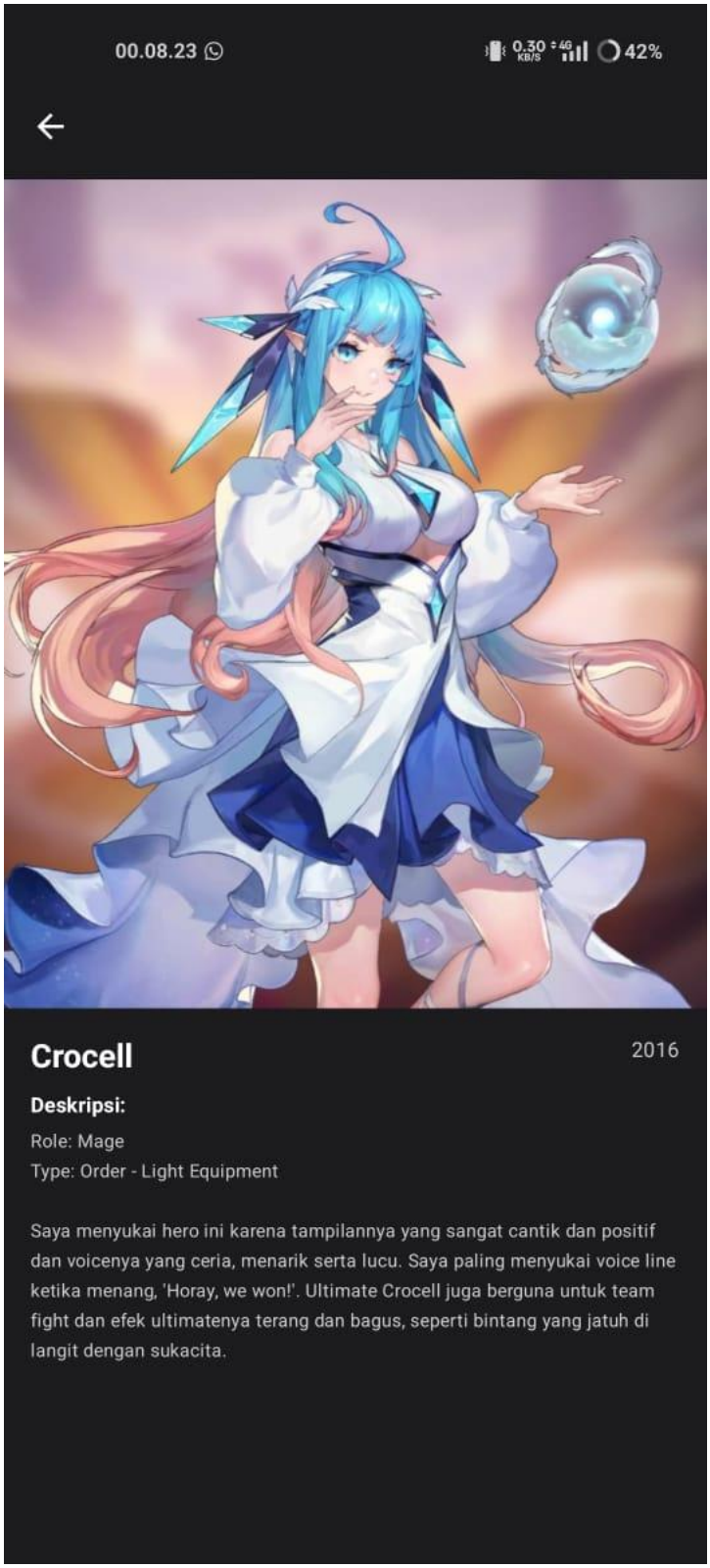
```
117
118     NavHost(navController, startDestination = "list")
119 {
120         composable("list") {
121             CharacterListScreen(characters =
123 characters, navController = navController)
124         }
125         composable("detail/{name}") { backStackEntry -
126 >
127             val name =
128 backStackEntry.arguments?.getString("name")
129             val character = charMap[name]
130             character?.let {
131                 CharacterDetailScreen(character = it,
132 navController = navController)
133             }
134         }
135     }
136 }
137
```

Tabel 5. Source Code

B. Output Program



Gambar 3. Screenshot Hasil Jawaban Soal 1



Gambar 4. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1. MainActivity

Kode `MainActivity` adalah titik awal aplikasi yang menggunakan Jetpack Compose. Di dalam `onCreate`, UI dibangun dengan `setContent`, menerapkan tema `MLACharactersTheme`, dan membungkusnya dalam `Surface` berwarna latar dari tema. Sebuah `NavController` dibuat untuk mengelola navigasi antar layar, lalu `NavGraph(navController)` dipanggil untuk mengatur rute-rute tampilan aplikasi seperti daftar dan detail karakter.

2. Character

Kode tersebut mendefinisikan sebuah `data class` bernama `Character` yang digunakan untuk merepresentasikan data karakter dalam aplikasi Android. Kelas ini memiliki lima properti, yaitu `name` untuk menyimpan nama asli karakter, `alias` untuk nama lain atau julukan, `imageRes` sebagai ID dari resource gambar karakter, `description` untuk menyimpan deskripsi karakter, dan `wikiUrl` sebagai tautan ke halaman wiki karakter tersebut.

3. CharacterDetailScreen.kt

Kode di atas merupakan implementasi tampilan detail karakter dalam aplikasi Android Jetpack Compose. Fungsi `CharacterDetailScreen` menerima data karakter dan `NavController` untuk navigasi. Komponen `Scaffold` digunakan untuk menyusun struktur UI, dengan `TopAppBar` yang memiliki tombol kembali. Isi layar dibungkus dalam `LazyColumn`, yang memungkinkan seluruh konten digulir. Gambar karakter ditampilkan di bagian atas, diikuti oleh informasi seperti nama, tahun, label deskripsi, dan isi deskripsi. Desain ini membuat tampilan detail tetap rapi dan mudah digulir meskipun konten panjang.

4. CharacterListScreen.kt

Kode di atas menampilkan daftar karakter dalam bentuk kartu menggunakan Jetpack Compose. Fungsi `CharacterListScreen` menerima list data `Character` dan `NavController` untuk navigasi. Daftar ini dibungkus dengan `LazyColumn` agar bisa digulir secara efisien. Setiap item ditampilkan dalam `Card` dengan gambar karakter di sebelah kiri dan informasi karakter di sebelah kanan.

Informasi ini mencakup nama, tahun, dan deskripsi. Dua tombol disediakan: satu untuk membuka tautan Wikipedia karakter menggunakan `Intent`, dan satu lagi untuk navigasi ke layar detail karakter. Desain UI dibuat rapi dan responsif dengan pengaturan padding, warna, dan tata letak.

5. **NavGraph.kt**

   Kode `NavGraph` di atas adalah implementasi navigasi menggunakan Jetpack Compose Navigation. Fungsi ini mendefinisikan dua layar utama dalam aplikasi: `CharacterListScreen` untuk menampilkan daftar karakter dan `CharacterDetailScreen` untuk menampilkan detail karakter berdasarkan nama. Data karakter disusun dalam list `characters`, lalu dipetakan menjadi `charMap` agar bisa diakses dengan cepat berdasarkan nama saat navigasi ke detail. Navigasi dimulai dari route `"list"`, dan ketika user memilih karakter, aplikasi berpindah ke route `"detail/{name}"` menggunakan nama sebagai parameter. Ini memungkinkan perpindahan data antar layar secara dinamis berdasarkan pilihan pengguna.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

Laporan-Praktikum-Pemrograman-Mobile/MODUL 3 at main · aikoanatashawendiono/Laporan-Praktikum-Pemrograman-Mobile