LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3

BUILD A SCROLLABLE LIST

Oleh:

Andra Braputra Akbar Saleh NIM. 2310817210001

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 Prakitkum ini dikerjakan oleh:

Nama Praktikan : Andra Braputra Akbar Saleh
NIM : 2310817210001

Menyetujui, Asisten Praktikum	Mengetahui, Dosen Penanggung Jawab Praktikum
Salsabila Syifa NIM. 2010817320004	Andreyan Rizky Baskara, S.Kom., M.Kom. NIP. 19930703 201903 01 011

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

XML

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 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 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

A. Source Code

MainActivity.kt

Tabel 1. MainActivity.kt

```
1   package com.example.scrollablexml
2
3   import android.os.Bundle
4   import android.util.Log
5   import androidx.appcompat.app.AppCompatActivity
6   import com.example.scrollablexml.databinding.ActivityMainBinding
7
8   class MainActivity : AppCompatActivity() {
9
10      private lateinit var binding: ActivityMainBinding
11
12      override fun onCreate(savedInstanceState: Bundle?) {
13          super.onCreate(savedInstanceState)
14
15          binding = ActivityMainBinding.inflate(layoutInflater)
16          setContentView(binding.root)
17
```

```
        val fragmentManager = supportFragmentManager
        val menuFragment = MenuFragment()
        val                          fragment                          =
fragmentManager.findFragmentByTag(MenuFragment::class.java.simpleName)
        if (fragment !is MenuFragment) {
            Log.d("MyFlexibleFragment",    "Fragment    Name    :"    +
MenuFragment::class.java.simpleName)
            fragmentManager
                .beginTransaction()
                .add(binding.fragmentContainer.id,       menuFragment,
MenuFragment::class.java.simpleName)
                .commit()
        }
    }
}
```

activity_main.xml

Tabel 2. Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MenuFragment.kt

Tabel 3. MenuFragment.kt

```
package com.example.scrollablexml

import android.content.Intent
import android.net.Uri
import android.os.Bundle
```

```
6  import android.view.LayoutInflater
7  import android.view.View
8  import android.view.ViewGroup
9  import androidx.fragment.app.Fragment
10 import androidx.recyclerview.widget.LinearLayoutManager
11 import com.example.scrollablexml.databinding.FragmentMenuBinding
12
13 class MenuFragment : Fragment() {
14
15     private var _binding: FragmentMenuBinding? = null
16     private val binding get() = _binding!!
17
18     private lateinit var falloutList: List<FalloutItem>
19
20     override fun onCreateView(
21         inflater: LayoutInflater, container: ViewGroup?,
22         savedInstanceState: Bundle?
23     ): View {
24         _binding = FragmentMenuBinding.inflate(inflater, container, false)
25         return binding.root
26     }
27
28     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
29         super.onViewCreated(view, savedInstanceState)
30
31         falloutList = listOf(
32             FalloutItem(1,                    getString(R.string.judul1),
   getString(R.string.year1),                getString(R.string.deskripsi1),
   getString(R.string.wiki1), getString(R.string.detail1), R.drawable.img1),
             FalloutItem(2,                    getString(R.string.judul2),
   getString(R.string.year2),                getString(R.string.deskripsi2),
   getString(R.string.wiki2), getString(R.string.detail2), R.drawable.img2),
             FalloutItem(3,                    getString(R.string.judul3),
   getString(R.string.year3),                getString(R.string.deskripsi3),
   getString(R.string.wiki3), getString(R.string.detail3), R.drawable.img3),
             FalloutItem(4,                    getString(R.string.judul4),
   getString(R.string.year4),                getString(R.string.deskripsi4),
   getString(R.string.wiki4), getString(R.string.detail4), R.drawable.img4),
             FalloutItem(5,                    getString(R.string.judul5),
   getString(R.string.year5),                getString(R.string.deskripsi5),
   getString(R.string.wikinv),                  getString(R.string.detail5),
   R.drawable.imgnv)
         )

         val adapter = FalloutAdapter(
             falloutList,
             onDetailClick = { item ->
                 val detailFragment = DetailFragment.newInstance(
                     item.title, item.year, item.detail, item.imageResId
                 )

                 parentFragmentManager.beginTransaction()
```

```
                    .replace(R.id.fragment_container, detailFragment)
                    .addToBackStack(null)
                    .commit()
            },
            onWikiClick = { item ->
                val url = item.wiki
                val    browserIntent    =    Intent(Intent.ACTION_VIEW,
Uri.parse(url))
                startActivity(browserIntent)
            }
        )

        binding.rvFallout.layoutManager                                    =
LinearLayoutManager(requireContext())
        binding.rvFallout.adapter = adapter
    }

    override fun onDestroyView() {
        super.onDestroyView()
        _binding = null
    }
}
```

Fragment_menu.xml

Tabel 4. Fragment_menu.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/main"
7       android:layout_width="match_parent"
8       android:layout_height="match_parent"
9       tools:context=".MainActivity">
10
11      <androidx.recyclerview.widget.RecyclerView
12          android:id="@+id/rvFallout"
13          android:layout_width="0dp"
14          android:layout_height="0dp"
15          app:layout_constraintTop_toTopOf="parent"
16          app:layout_constraintBottom_toBottomOf="parent"
17          app:layout_constraintStart_toStartOf="parent"
18          app:layout_constraintEnd_toEndOf="parent"
19          android:background="@color/fallout_bg"/>
20
    </androidx.constraintlayout.widget.ConstraintLayout>
```

DetailFragment.kt

Tabel 5. DetailFragment.kt

```
1   package com.example.scrollablexml
2
3   import android.os.Bundle
4   import android.view.LayoutInflater
5   import android.view.View
6   import android.view.ViewGroup
7   import androidx.fragment.app.Fragment
8   import com.example.scrollablexml.databinding.FragmentDetailBinding
9
10  class DetailFragment : Fragment() {
11
12      private var _binding: FragmentDetailBinding? = null
13      private val binding get() = _binding!!
14
15      override fun onCreateView(
16          inflater: LayoutInflater, container: ViewGroup?,
17          savedInstanceState: Bundle?
18      ): View {
19          _binding  =  FragmentDetailBinding.inflate(inflater,  container,
20  false)
21          return binding.root
22      }
23
24      override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
25          super.onViewCreated(view, savedInstanceState)
26
27          val title = arguments?.getString("title")
28          val year = arguments?.getString("year")
29          val description = arguments?.getString("description")
30          val imageResId = arguments?.getInt("imageResId", 0) ?: 0
31
32          binding.tvTitle.text = title
            binding.tvYear.text = year
            binding.tvDetail.text = description
            binding.detailImage.setImageResource(imageResId)
        }

        override fun onDestroyView() {
            super.onDestroyView()
            _binding = null
        }

        companion object {
            fun newInstance(title: String, year: String, description: String,
    imageResId: Int): DetailFragment {
                val fragment = DetailFragment()
                val bundle = Bundle().apply {
                    putString("title", title)
                    putString("year", year)
```

```
                putString("description", description)
                putInt("imageResId", imageResId)
            }
            fragment.arguments = bundle
            return fragment
        }
    }
}
```

Fragment_menu.xml

Tabel 6. Fragment_menu 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      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:fillViewport="true"
8      android:background="@color/fallout_bg"
9      >
10
11     <ImageView
12         android:id="@+id/detailImage"
13         android:layout_marginTop="15dp"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:contentDescription="@string/img"
17         android:scaleType="centerCrop"
18         android:src="@drawable/img1"
19         app:layout_constraintEnd_toEndOf="parent"
20         app:layout_constraintStart_toStartOf="parent"
           app:layout_constraintTop_toTopOf="parent" />

           <TextView
               android:id="@+id/tvTitle"
               android:layout_width="0dp"
               android:layout_height="wrap_content"
               android:layout_marginTop="16dp"
               android:layout_marginStart="10dp"
               android:text="@string/title"
               android:textColor="@color/fallout_accent"
               android:textSize="22sp"
               android:textStyle="bold"

   app:layout_constraintTop_toBottomOf="@+id/detailImage"
               app:layout_constraintStart_toStartOf="parent"
               app:layout_constraintEnd_toStartOf="@+id/tvYear" />

           <TextView
               android:id="@+id/tvYear"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="10dp"
            android:text="@string/year"
            android:textColor="@color/fallout_text"
            android:textSize="20sp"

app:layout_constraintBaseline_toBaselineOf="@id/tvTitle"
            app:layout_constraintEnd_toEndOf="parent" />

        <TextView
            android:id="@+id/tvDetail"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_marginHorizontal="20dp"
            android:text="@string/description"
            android:textColor="@color/fallout_text"
            android:textSize="16sp"
            app:layout_constraintTop_toBottomOf="@+id/tvTitle"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

FalloutAdapter.kt

Tabel 7. FalloutAdapter.kt

```
1  package com.example.scrollablexml
2
3  import android.view.LayoutInflater
4  import android.view.ViewGroup
5  import androidx.recyclerview.widget.RecyclerView
6  import com.example.scrollablexml.databinding.GameCardBinding
7
8  class FalloutAdapter(
9      private val items: List<FalloutItem>,
10     private val onWikiClick: (FalloutItem) -> Unit,
11     private val onDetailClick: (FalloutItem) -> Unit
12 ) : RecyclerView.Adapter<FalloutAdapter.ViewHolder>() {
13
14     inner class ViewHolder(val binding: GameCardBinding) :
15         RecyclerView.ViewHolder(binding.root)
16
17     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
18 ViewHolder {
19         val                              binding                              =
20 GameCardBinding.inflate(LayoutInflater.from(parent.context),      parent,
21 false)
22         return ViewHolder(binding)
23     }
```

```
24
25      override fun getItemCount(): Int = items.size
26
27      override fun onBindViewHolder(holder: ViewHolder, position: Int) {
28          val item = items[position]
29          with(holder.binding) {
30              tvTitle.text = item.title
31              tvDescription.text = item.description
32              ivImg.setImageResource(item.imageResId)

                btnDetail.setOnClickListener {
                    onDetailClick(item)
                }

                btnWiki.setOnClickListener {
                    onWikiClick(item)
                }
            }
        }
    }
```

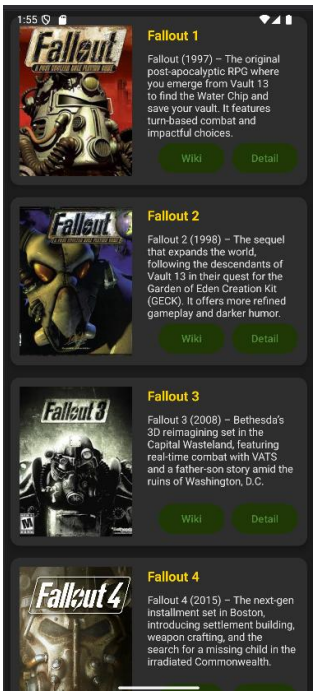
game_card.xml

Tabel 8. game_card.xml

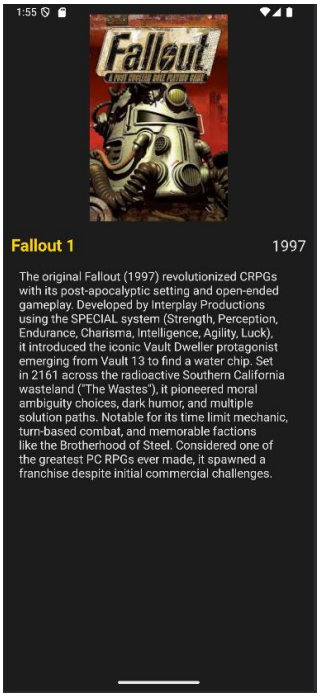
```
1   package com.example.scrollablexml
2
3   import android.view.LayoutInflater
4   import android.view.ViewGroup
5   import androidx.recyclerview.widget.RecyclerView
6   import com.example.scrollablexml.databinding.GameCardBinding
7
8   class FalloutAdapter(
9       private val items: List<FalloutItem>,
10      private val onWikiClick: (FalloutItem) -> Unit,
11      private val onDetailClick: (FalloutItem) -> Unit
12  ) : RecyclerView.Adapter<FalloutAdapter.ViewHolder>() {
13
14      inner class ViewHolder(val binding: GameCardBinding) :
15          RecyclerView.ViewHolder(binding.root)
16
17      override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
18  ViewHolder {
19          val                              binding                              =
20  GameCardBinding.inflate(LayoutInflater.from(parent.context),      parent,
21  false)
22          return ViewHolder(binding)
23      }
24
25      override fun getItemCount(): Int = items.size
26
27      override fun onBindViewHolder(holder: ViewHolder, position: Int) {
```

```
28          val item = items[position]
29          with(holder.binding) {
30              tvTitle.text = item.title
31              tvDescription.text = item.description
32              ivImg.setImageResource(item.imageResId)

                btnDetail.setOnClickListener {
                    onDetailClick(item)
                }

                btnWiki.setOnClickListener {
                    onWikiClick(item)
                }
            }
        }
    }
```

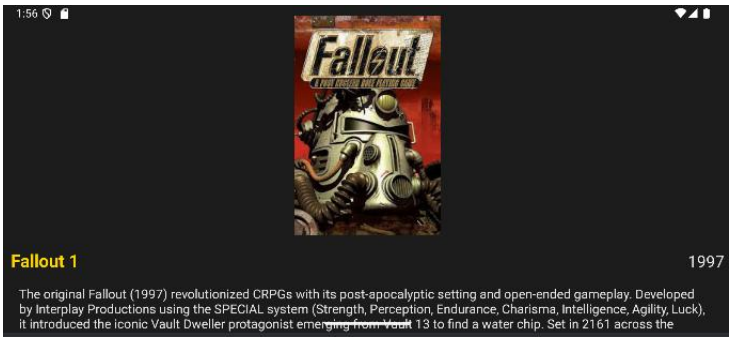
B. Output Program



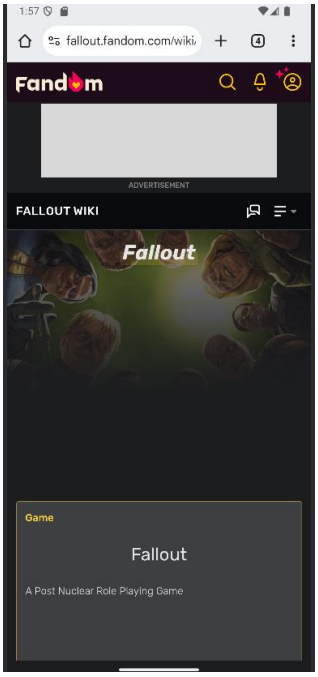
Gambar 1. Screenshot Hasil Jawaban XML



Gambar 2. Screenshot Hasil Jawaban XML



Gambar 3. Screenshot Hasil Jawaban XML



Gambar 4. Screenshot Hasil Jawaban XML

C. Pembahasan

MainActivity.kt:

Line 1**:** Dideklarasikan nama package file Kotlin, yaitu `com.example.scrollablexml`
Line 3**:** Diimport `Bundle` untuk menyimpan dan meneruskan data saat aktivitas dibuat
Line 4**:** Diimport `Log` untuk mencatat log debug selama aplikasi dijalankan
Line 5**:** Diimport `AppCompatActivity` sebagai superclass dari `MainActivity` agar kompatibel dengan versi Android lama
Line 6**:** Diimport `ActivityMainBinding` untuk menghubungkan layout XML `activity_main.xml` menggunakan View Binding
Line 8**:** Dideklarasikan class `MainActivity` yang merupakan turunan dari `AppCompatActivity`
Line 10**:** Dideklarasikan properti `binding` bertipe `ActivityMainBinding` dengan `lateinit` agar dapat diinisialisasi nanti
Line 12**:** Override fungsi `onCreate()` yang dipanggil saat activity pertama kali dimulai
Line 13**:** Memanggil `super.onCreate()` untuk memastikan lifecycle superclass dijalankan
Line 15**:** Inisialisasi `binding` dengan `ActivityMainBinding.inflate(layoutInflater)` untuk mengakses elemen-elemen layout
Line 16**:** Menampilkan isi layout dari `binding.root` ke layar
Line 18**:** Mendeklarasikan variabel `fragmentManager` untuk mengelola fragment
Line 19**:** Membuat instance dari `MenuFragment`
Line 20**:** Mengecek apakah fragment dengan tag `MenuFragment` sudah ada dalam fragment manager

Line 21–22**:** Jika belum ada, maka akan mencetak log nama fragment
Line 23–26**:** Melakukan transaksi untuk menambahkan `MenuFragment` ke dalam `fragment_container`, dan memberi tag dengan nama class-nya

MenuFragment.kt

Line 1**:** Dideklarasikan nama package file Kotlin yaitu `com.example.scrollablexml`
Line 3–4**:** Diimport `Intent` dan `Uri` untuk membuka URL (tautan Wiki) di browser
Line 5–6**:** Diimport class `Bundle`, `LayoutInflater`, `View`, dan `ViewGroup` untuk kebutuhan tampilan fragment
Line 7**:** Diimport `Fragment` sebagai superclass dari `MenuFragment`
Line 8**:** Diimport `LinearLayoutManager` untuk menentukan layout RecyclerView secara vertikal
Line 9**:** Diimport `FragmentMenuBinding` untuk mengakses elemen di `fragment_menu.xml` melalui View Binding
Line 11**:** Dideklarasikan kelas `MenuFragment` yang mewarisi class `Fragment`
Line 13**:** Dideklarasikan variabel `_binding` dengan tipe nullable `FragmentMenuBinding` untuk diinisialisasi di `onCreateView`
Line 14**:** Dideklarasikan properti `binding` yang mengakses `_binding` secara non-null
Line 16**:** Dideklarasikan variabel `falloutList` sebagai list yang berisi data `FalloutItem`
Line 18–21**:** Override fungsi `onCreateView()` yang digunakan untuk meng-inflate layout fragment dan menginisialisasi View Binding
Line 23**:** Override fungsi `onViewCreated()` untuk logika program setelah tampilan berhasil dibuat
Line 25–30**:** Mengisi `falloutList` dengan beberapa item, mengambil data dari `strings.xml` dan gambar dari `drawable`
Line 32**:** Membuat adapter `FalloutAdapter`, menyambungkan list dengan aksi klik tombol detail dan wiki
Line 33–36**:** Saat tombol Detail diklik, akan berpindah ke `DetailFragment` menggunakan data yang dikirim
Line 38–40**:** Saat tombol Wiki diklik, membuka URL menggunakan browser
Line 43**:** Mengatur layout RecyclerView dengan `LinearLayoutManager`
Line 44**:** Menetapkan adapter ke RecyclerView
Line 47–49**:** Membersihkan `_binding` saat view dihancurkan agar tidak terjadi memory leak

DetailFragment.kt

Line 1**:** Dideklarasikan package `com.example.scrollablexml`
Line 3–6**:** Diimport semua class yang diperlukan untuk fragment dan tampilan
Line 7**:** Diimport `FragmentDetailBinding` untuk binding layout `fragment_detail.xml`
Line 9**:** Dideklarasikan class `DetailFragment` sebagai turunan dari `Fragment`
Line 11**:** Dideklarasikan variabel `_binding` bertipe nullable `FragmentDetailBinding`
Line 12**:** Dideklarasikan properti `binding` untuk mengakses `_binding` secara aman

Line 14–17**:** Override fungsi `onCreateView()` untuk meng-inflate layout `fragment_detail.xml`
Line 19**:** Override fungsi `onViewCreated()` untuk logika ketika tampilan sudah dibuat
Line 21–24**:** Mengambil data argumen (judul, tahun, deskripsi, dan gambar) dari bundle yang dikirim
Line 26–29: Menampilkan data yang diterima ke elemen UI (TextView dan ImageView)
Line 31–33: Membersihkan binding saat view dihancurkan
Line 35–41: Companion object dengan fungsi `newInstance()` untuk membuat dan mengirim data ke fragment melalui `Bundle`

FalloutAdapter.kt

Line 1**:** Dideklarasikan package `com.example.scrollablexml`
Line 3–5**:** Diimport class yang diperlukan untuk RecyclerView dan LayoutInflater
Line 6**:** Diimport `GameCardBinding` untuk menghubungkan tampilan layout item list
Line 8–12**:** Dideklarasikan class `FalloutAdapter` dengan konstruktor berisi:
- `items`: list item Fallout
- `onWikiClick`: fungsi lambda untuk event klik tombol Wiki
- `onDetailClick`: fungsi lambda untuk event klik tombol Detail

Line 13–14**:** Dideklarasikan inner class `ViewHolder` untuk menyimpan elemen UI setiap item list
Line 16–18**:** Override fungsi `onCreateViewHolder()` untuk meng-inflate layout item list dari XML `game_card.xml`
Line 20**:** Override fungsi `getItemCount()` untuk menghitung jumlah item dalam list
Line 22–31**:** Override fungsi `onBindViewHolder()` untuk menampilkan data setiap elemen item list
- Line 23–26**:** Mengisi teks dan gambar dari `FalloutItem`
- Line 28**:** Saat tombol `btnDetail` diklik, jalankan fungsi `onDetailClick`
- Line 30**:** Saat tombol `btnWiki` diklik, jalankan fungsi `onWikiClick`

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
https://github.com/Andra-Braputra/PrakMobile/tree/main/Modul%203/ScrollableXML-main

COMPOSE

A. Source Code

MainActivity.kt

Tabel 9. MainActivity.kt

```
1  package com.example.scrollablecompose
2
3  import android.os.Bundle
4  import androidx.activity.ComponentActivity
5  import androidx.activity.compose.setContent
6  import androidx.navigation.NavType
7  import androidx.navigation.compose.NavHost
8  import androidx.navigation.compose.composable
9  import androidx.navigation.compose.rememberNavController
10 import androidx.navigation.navArgument
11
12 class MainActivity : ComponentActivity() {
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         setContent {
16             val navController = rememberNavController()
17             NavHost(
18                 navController = navController,
19                 startDestination = Routes.Menu
20             ) {
21                 composable(Routes.Menu) {
22                     FalloutApp(navController)
23                 }
24                 composable(
25                     route = "detail/{gameId}",
26                     arguments = listOf(navArgument("gameId") { type
27 = NavType.IntType })
28                 ) { backStackEntry ->
29                     val                    gameId                    =
30 backStackEntry.arguments?.getInt("gameId") ?: 0
31                     ShowDetail(gameId = gameId)
32                 }
33             }
34         }
35     }
36 }
```

Detail.kt

Tabel 10. Detail.kt

```
1  package com.example.scrollablecompose
2
3  import androidx.compose.foundation.Image
```

```
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.aspectRatio
import androidx.compose.foundation.layout.fillMaxHeight
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material3.Card
import androidx.compose.material3.CardDefaults
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.remember
import androidx.compose.ui.Modifier
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.colorResource
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import com.example.scrollablecompose.data.Datasource

@Composable
fun ShowDetail(gameId: Int) {
    val          game          =          remember          {
Datasource().loadFallout().firstOrNull { it.id == gameId } }

    if (game == null) {
        Text(
            text = "Game not found",
            modifier = Modifier.padding(16.dp),
            style = MaterialTheme.typography.bodyLarge,
            color = colorResource(R.color.fallout_text)
        )
    } else {
        Column(modifier = Modifier
            .fillMaxHeight()
            .background(color                                    =
colorResource(R.color.fallout_bg))
            .run {
                padding(16.dp).fillMaxWidth()
                    .verticalScroll(rememberScrollState())
```

```
            }

        ) {
            Card(
                modifier = Modifier
                    .fillMaxWidth()
                    .aspectRatio(16f / 9f),
                colors                                          =
CardDefaults.cardColors(containerColor                          =
colorResource(R.color.fallout_card))
            ) {
                Image(
                    painter     =     painterResource(id     =
game.imageResId),
                    contentDescription  =  stringResource(id  =
game.titleResId),
                    modifier = Modifier.fillMaxSize(),
                    contentScale = ContentScale.Fit
                )
            }

            Spacer(modifier = Modifier.height(16.dp))

            Row{
                Text(
                    text      =      stringResource(id      =
game.titleResId),
                    style                                       =
MaterialTheme.typography.headlineLarge,
                    color                                       =
colorResource(R.color.fallout_accent)
                )
                Spacer(modifier = Modifier.weight(1f))
                Text(
                    text = stringResource(id = game.yearId),
                    style                                       =
MaterialTheme.typography.headlineLarge,
                    color                                       =
colorResource(R.color.fallout_text)


                )
            }

            Spacer(modifier = Modifier.height(16.dp))

            Text(
```

```
                    text = stringResource(id = game.detailResId),
                    style = MaterialTheme.typography.bodyLarge,
                    modifier = Modifier.fillMaxWidth(),
                    textAlign = TextAlign.Justify,
                    color = colorResource(R.color.fallout_text)
                )
            }
        }
    }
```

Menu.kt

Tabel 11. Menu.kt

```
1  package com.example.scrollablecompose
2
3  import android.content.Context
4  import android.content.Intent
5  import android.net.Uri
6  import androidx.compose.foundation.Image
7  import androidx.compose.foundation.layout.Column
8  import androidx.compose.foundation.layout.PaddingValues
9  import androidx.compose.foundation.layout.Row
10 import androidx.compose.foundation.layout.WindowInsets
11 import androidx.compose.foundation.layout.asPaddingValues
12 import androidx.compose.foundation.layout.aspectRatio
13 import androidx.compose.foundation.layout.calculateEndPadding
14 import androidx.compose.foundation.layout.calculateStartPadding
15 import androidx.compose.foundation.layout.fillMaxHeight
16 import androidx.compose.foundation.layout.fillMaxSize
17 import androidx.compose.foundation.layout.fillMaxWidth
18 import androidx.compose.foundation.layout.padding
19 import androidx.compose.foundation.layout.safeDrawing
20 import androidx.compose.foundation.layout.statusBarsPadding
21 import androidx.compose.foundation.lazy.LazyColumn
22 import androidx.compose.material3.Button
23 import androidx.compose.material3.ButtonDefaults
24 import androidx.compose.material3.Card
25 import androidx.compose.material3.CardDefaults
26 import androidx.compose.material3.MaterialTheme
27 import androidx.compose.material3.Surface
28 import androidx.compose.material3.Text
29 import androidx.compose.runtime.Composable
30 import androidx.compose.ui.Alignment
   import androidx.compose.ui.Modifier
   import androidx.compose.ui.layout.ContentScale
   import androidx.compose.ui.platform.LocalContext
   import androidx.compose.ui.platform.LocalLayoutDirection
   import androidx.compose.ui.res.colorResource
   import androidx.compose.ui.res.painterResource
```

```
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.navigation.NavController
import com.example.scrollablecompose.data.Datasource
import com.example.scrollablecompose.model.Fallout
import com.example.scrollablecompose.ui.theme.FalloutButton

@Composable
fun FalloutApp(navController: NavController) {
    val layoutDirection = LocalLayoutDirection.current
    val paddingValues = WindowInsets.safeDrawing.asPaddingValues()
    val                          startPadding                                =
paddingValues.calculateStartPadding(layoutDirection)
    val                           endPadding                                 =
paddingValues.calculateEndPadding(layoutDirection)


    Surface(
        modifier = Modifier
            .fillMaxSize()
            .statusBarsPadding()
            .padding(start = startPadding, end = endPadding),
        color = colorResource(R.color.fallout_bg)
    ) {
        GameList(
            gameList = Datasource().loadFallout(),
            onDetailClick = { id ->
                navController.navigate("detail/$id")
            }
        )
    }
}


@Composable
fun GameList(
    gameList: List<Fallout>,
    modifier: Modifier = Modifier,
    onDetailClick: (Int) -> Unit
) {
    LazyColumn(
        modifier = modifier,
        contentPadding = PaddingValues(10.dp)
    ) {
        items(gameList.size) { index ->
            val game = gameList[index]
            FalloutCard(
                fallout = game,
                modifier = Modifier.padding(vertical = 4.dp),
                onDetailClick = { onDetailClick(game.id) }
```

```
            )
        }
    }
}

@Composable
fun FalloutCard(
    fallout: Fallout,
    modifier: Modifier = Modifier,
    onDetailClick: () -> Unit = {}
) {
    val context = LocalContext.current
    val image = painterResource(id = fallout.imageResId)
    val title = stringResource(id = fallout.titleResId)
    val description = stringResource(id = fallout.descriptionResId)
    val wikiUrl = stringResource(id = fallout.wikiResId)

    Card(modifier    =    modifier.fillMaxHeight(),    colors    =
CardDefaults.cardColors(containerColor                                =
com.example.scrollablecompose.ui.theme.FalloutCard) ) {
        Row(modifier = Modifier.fillMaxHeight()) {
            Image(
                painter = image,
                contentDescription = title,
                modifier = Modifier
                    .aspectRatio(3f / 5f)
                    .weight(1.5f),
                contentScale = ContentScale.Crop
            )
            Column(modifier = Modifier.weight(2f)) {
                Text(
                    text = title,
                    style                                             =
MaterialTheme.typography.titleLarge.copy(fontWeight                =
FontWeight.Bold),
                    modifier = Modifier.padding(horizontal = 16.dp,
vertical = 20.dp),
                    color = colorResource(R.color.fallout_text)
                )
                Text(
                    text = description,
                    style = MaterialTheme.typography.bodyLarge,
                    modifier = Modifier.padding(horizontal = 16.dp,
vertical = 8.dp),
                    color = colorResource(R.color.fallout_text)
                )
                Row(
                    modifier = Modifier.padding(bottom = 12.dp),
                    verticalAlignment = Alignment.Bottom
                ) {
                    Button(
                        onClick = {
```

```
                            openWikiForGame(context, wikiUrl)
                        },
                        modifier = Modifier
                            .padding(6.dp)
                            .weight(1f),
                        colors                                          =
ButtonDefaults.buttonColors(containerColor = FalloutButton)
                    ) {
                        Text(
                            modifier = Modifier.fillMaxWidth(),
                            textAlign = TextAlign.Center,
                            text = "Wiki",
                            color                                       =
colorResource(R.color.fallout_button_text)
                        )
                    }
                    Button(
                        onClick = onDetailClick,
                        modifier = Modifier
                            .padding(6.dp)
                            .weight(1f),
                        colors                                          =
ButtonDefaults.buttonColors(containerColor = FalloutButton)
                    ) {
                        Text(
                            modifier = Modifier.fillMaxWidth(),
                            textAlign = TextAlign.Center,
                            text = "Detail",
                            color                                       =
colorResource(R.color.fallout_button_text)
                        )
                    }
                }
            }
        }
    }
}

fun openWikiForGame(context: Context, wikiUrl: String) {
    val intent = Intent(Intent.ACTION_VIEW, Uri.parse(wikiUrl))
    context.startActivity(intent)
}
```

Routes.kt

Tabel 12. MainActivity.kt

```
1  package com.example.scrollablecompose
2
3  object Routes {
4      var Menu = "Menu"
```

```
5 }
```

DataSource.kt

Tabel 13. DataSource.kt

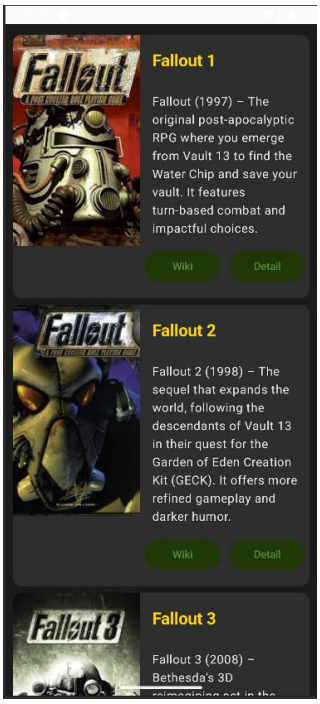
```
1 package com.example.scrollablecompose
2
3 object Routes {
4     var Menu = "Menu"
5 }
```

Fallout.kt

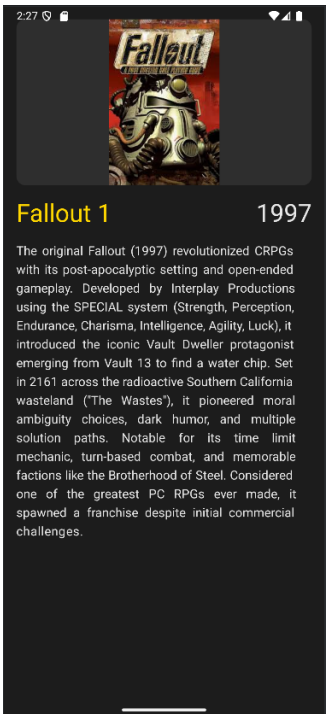
Tabel 14. Fallout.kt

```
1 package com.example.scrollablecompose.model
2
3 import androidx.annotation.DrawableRes
4 import androidx.annotation.StringRes
5

  data class Fallout(
      val id: Int,
      @StringRes val titleResId: Int,
      @StringRes val descriptionResId: Int,
      @DrawableRes val imageResId: Int,
      @StringRes val wikiResId: Int,
      @StringRes val yearId: Int,
      @StringRes val detailResId: Int
  )
```

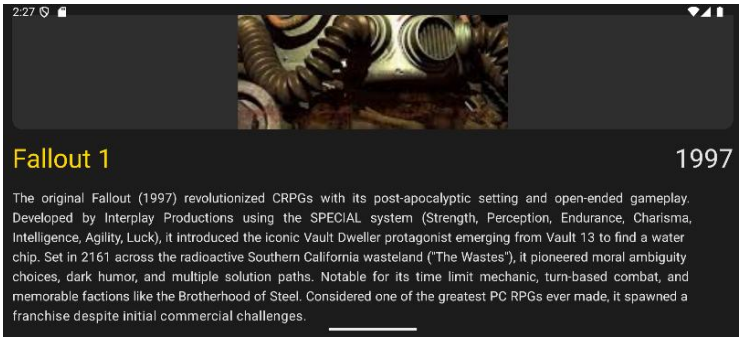
B. Output Program



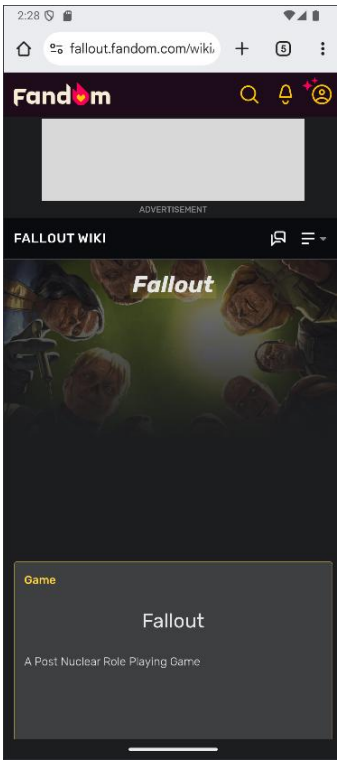
Gambar 4. Source Code Jawaban Compose



Gambar 5. Source Code Jawaban Compose



Gambar 6. Source Code Jawaban Compose



Gambar 7. Source Code Jawaban Compose

C. Pembahasan

MainActivity.kt:

- line 9**,** dideklarasikan kelas bernama `MainActivity` yang mewarisi dari `ComponentActivity`, yaitu base class activity khusus Jetpack Compose.
- line 10**,** terdapat override fungsi `onCreate`, yang dipanggil saat activity pertama kali dibuat.
- line 11**,** dipanggil fungsi `super.onCreate()` untuk menjalankan proses `onCreate` bawaan dari superclass.

- line 12**,** dipanggil fungsi `setContent` untuk memulai UI berbasis Jetpack Compose.
- line 13**,** dibuat variabel `navController` untuk mengatur navigasi antar composable menggunakan Jetpack Navigation.
- line 14**,** digunakan `NavHost` dengan `navController` dan startDestination “Menu” sebagai halaman awal.
- line 15**,** ditambahkan rute “Menu” menggunakan `composable`.
- line 16**,** dipanggil fungsi `FalloutApp(navController)` yang menampilkan daftar game.
- line 18**,** dibuat rute dinamis “detail/{gameId}” dengan argumen `gameId` bertipe `Int`.
- line 22**,** diambil nilai `gameId` dari argumen navigasi, jika null maka default-nya 0.
- line 23**,** dipanggil fungsi `ShowDetail(gameId)` untuk menampilkan detail game berdasarkan ID.

Detail.kt

- line 1, dideklarasikan nama package file Kotlin
- line 32, fungsi `ShowDetail` dideklarasikan dan menerima parameter `gameId` bertipe `Int`
- line 33, `game` diambil dari sumber data berdasarkan `id` menggunakan fungsi `firstOrNull`
- line 35, jika `game` bernilai null maka ditampilkan teks "Game not found"
- line 42, jika `game` tidak null maka akan ditampilkan UI dalam `Column`
- line 43, `Modifier` diatur untuk mengisi tinggi layar dan memberi padding serta scrollable
- line 48, digunakan `Card` untuk menampilkan gambar game
- line 52, digunakan `Image` untuk menampilkan gambar game dari resource ID
- line 58, diberi `Spacer` untuk jarak antar elemen
- line 60, digunakan `Row` untuk menampilkan judul dan tahun
- line 61-64, judul game ditampilkan dengan style headline
- line 65-68, tahun rilis ditampilkan di sisi kanan baris
- line 71, diberi `Spacer` kembali
- line 73-78, ditampilkan deskripsi game dalam paragraf justify

Menu.kt

- line 1, dideklarasikan nama package file Kotlin
- line 41, fungsi `FalloutApp` dideklarasikan dan menerima parameter `navController`
- line 42-44, digunakan `WindowInsets` untuk mengatur padding aman layout
- line 47, digunakan `Surface` sebagai wadah UI utama
- line 48-50, modifier digunakan untuk padding dan mengatur ukuran penuh
- line 51, warna latar belakang diatur menggunakan resource
- line 52, fungsi `GameList` dipanggil untuk menampilkan daftar game
- line 53, saat tombol detail diklik, navigasi dilakukan ke "detail/{id}"

Subfungsi: GameList

- line 60, fungsi `GameList` menerima daftar game dan fungsi callback `onDetailClick`
- line 63, digunakan `LazyColumn` untuk daftar scrollable
- line 65-68, setiap item game di-loop dan dipanggil fungsi `FalloutCard`

Subfungsi: FalloutCard

- line 71, fungsi `FalloutCard` menerima data Fallout dan fungsi klik detail
- line 72-75, diambil data dari resource ID untuk ditampilkan
- line 77, digunakan `Card` sebagai container dari informasi game
- line 78, digunakan `Row` untuk membagi layout horizontal
- line 79-83, gambar game ditampilkan di sisi kiri
- line 84-101, kolom kanan berisi judul, deskripsi, dan dua tombol
- line 88-91, judul game ditampilkan dengan padding
- line 92-95, deskripsi ditampilkan
- line 96-100, dua tombol ditampilkan secara horizontal:
    - Tombol Wiki membuka URL dengan `openWikiForGame`
    - Tombol Detail memanggil fungsi `onDetailClick`

Fungsi bantu: openWikiForGame

- line 104, fungsi `openWikiForGame` menerima context dan URL wiki
- line 105, `Intent` dibuat untuk membuka browser
- line 106, browser dibuka menggunakan `startActivity`

Routes.kt

- line 1, dideklarasikan nama package file Kotlin
- line 3, dideklarasikan object singleton `Routes`
- line 4, variabel `Menu` bertipe string dengan nilai "Menu"

Fallout.kt

- line 1, dideklarasikan nama package file Kotlin
- line 3-4, digunakan anotasi `@DrawableRes` dan `@StringRes` untuk validasi tipe resource
- line 6, dideklarasikan data class `Fallout` sebagai model data game
- line 7, `id` merupakan integer unik untuk setiap game
- line 8, `titleResId` adalah ID dari string title
- line 9, `descriptionResId` adalah ID dari string deskripsi
- line 10, `imageResId` adalah ID dari drawable image
- line 11, `wikiResId` adalah ID dari string URL wiki
- line 12, `yearId` adalah ID dari string tahun rilis
- line 13, `detailResId` adalah ID dari string detail game

DataSource.kt

Object DataSource

- line 8, dideklarasikan object singleton bernama `DataSource`
- line 9, dideklarasikan properti `falloutList` bertipe `List<Fallout>`, yang berisi daftar game Fallout

List Fallout

- line 10–22, objek pertama `Fallout` dibuat dengan properti:
  - `id = 1` (ID unik)
  - `titleResId = R.string.fallout_1_title` (judul Fallout 1)
  - `descriptionResId = R.string.fallout_1_description` (deskripsi pendek)
  - `imageResId = R.drawable.fallout1` (gambar dari drawable)
  - `wikiResId = R.string.fallout_1_wiki` (URL ke Wikipedia Fallout 1)
  - `yearId = R.string.fallout_1_year` (tahun rilis)
  - `detailResId = R.string.fallout_1_detail` (deskripsi panjang/detail)

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
https://github.com/Andra-Braputra/PrakMobile/tree/main/Modul%203/ScrollableCompose-main

SOAL 2

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

A. Jawaban

RecyclerView memberikan fleksibilitas lebih dalam hal menyesuaikan layout item, menangani berbagai tipe item, dan mengintegrasikan animasi kompleks. LazyRow/LazyColumn, meskipun lebih sederhana, mungkin memiliki keterbatasan dalam hal opsi kustomisasi.

RecyclerView sudah memilki komponen yang lebih lama dan sering digunakan pada pengembangan android, sehingga akan lebih mudah dalam mencari bantuan di internet dan sudah banyak bantuan dari forum. LazyColumn yang relatif baru akan lebih sulit.