LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3

Build a Scrollable List

Oleh:

Harry Pratama Yunus NIM. 2310817210010

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 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 : Harry Pratama Yunus
NIM : 2310817210010

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

DAFTAR GAMBAR

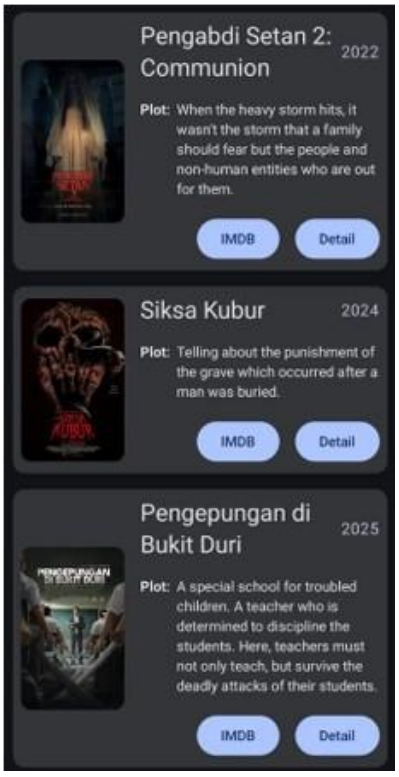
DAFTAR TABEL

SOAL 1

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

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 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 List

A. Source Code

XML :

Affirmation.kt

Table 1 source code soal 1 Affirmation.kt (xml)

```
1  package com.pemrogamanmobile.myfavoritegames.data
2
3  import java.io.Serializable
4
5  data class Affirmation(
6          val detailResourceId: Int,
7          val imageResourceId: Int,
8          val steamLinkResourceId: Int,
9          val yearResourceId: Int,
10         val titleResourceId: Int
11 ) : Serializable
```

DataSource.kt

Table 2 source code soal 1 DataSource.kt (xml)

```
1  package com.pemrogamanmobile.myfavoritegames.data
2
3  import com.pemrogamanmobile.myfavoritegames.R
4
5  object DataSource {
6      fun loadGames(): List<Affirmation> = listOf(
7          Affirmation(R.string.detail1, R.drawable.image1, R.string.link1, R.string.year1,
       R.string.title1),
8          Affirmation(R.string.detail2, R.drawable.image2, R.string.link2, R.string.year2,
       R.string.title2),
9          Affirmation(R.string.detail3, R.drawable.image3, R.string.link3, R.string.year3,
       R.string.title3),
10         Affirmation(R.string.detail4, R.drawable.image4, R.string.link4, R.string.year4,
       R.string.title4),
11         Affirmation(R.string.detail5, R.drawable.image5, R.string.link5, R.string.year5,
       R.string.title5),
12         Affirmation(R.string.detail6, R.drawable.image6, R.string.link6, R.string.year6,
       R.string.title6),
13         Affirmation(R.string.detail7, R.drawable.image7, R.string.link7, R.string.year7,
       R.string.title7),
14         Affirmation(R.string.detail8, R.drawable.image8, R.string.link8, R.string.year8,
       R.string.title8),
15         Affirmation(R.string.detail9, R.drawable.image9, R.string.link9, R.string.year9,
       R.string.title9),
16         Affirmation(R.string.detail10, R.drawable.image10, R.string.link10,
       R.string.year10, R.string.title10)
17     )
18 }
```

AffirmationAdapter.kt

Table 3 source code soal 1 AffirmationAdapter.kt (xml)

```
1   package com.pemrogamanmobile.myfavoritegames.ui
2
3   import android.content.Intent
4   import android.net.Uri
5   import android.view.LayoutInflater
6   import android.view.ViewGroup
7   import androidx.recyclerview.widget.RecyclerView
8   import com.pemrogamanmobile.myfavoritegames.data.Affirmation
9   import
    com.pemrogamanmobile.myfavoritegames.databinding.ItemAffirmationBinding
10
11  class AffirmationAdapter(
12      private val items: List<Affirmation>,
```

```
13      private val onDetailClick: (Affirmation) -> Unit
14  ) : RecyclerView.Adapter<AffirmationAdapter.ViewHolder>() {
15
16      inner class ViewHolder(val binding: ItemAffirmationBinding)
17          : RecyclerView.ViewHolder(binding.root)
18
19      override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
        ViewHolder {
20          val binding = ItemAffirmationBinding.inflate(
21              LayoutInflater.from(parent.context), parent, false
22          )
23          return ViewHolder(binding)
24      }
25
26      override fun onBindViewHolder(holder: ViewHolder, position: Int) {
27          val item = items[position]
28          with(holder.binding) {
29              imageView.setImageResource(item.imageResourceId)
30              titleText.text   = holder.itemView.context.getString(item.titleResourceId)
31              yearText.text    =
        holder.itemView.context.getString(item.yearResourceId)
32              detailText.text  =
        holder.itemView.context.getString(item.detailResourceId)
33
34              viewButton.setOnClickListener {
35                  val url =
        holder.itemView.context.getString(item.steamLinkResourceId)
36                  holder.itemView.context.startActivity(
37                      Intent(Intent.ACTION_VIEW, Uri.parse(url))
38                  )
39              }
40              detailButton.setOnClickListener {
41                  onDetailClick(item)
42              }
43          }
44      }
45
46      override fun getItemCount(): Int = items.size
47  }
```

AffirmationListFragment.kt

Table 4 source code soal 1 AffirmationListFragment.kt (xml)

```
1   package com.pemrogamanmobile.myfavoritegames.ui
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 androidx.navigation.fragment.findNavController
9  import com.pemrogamanmobile.myfavoritegames.data.DataSource
10 import
   com.pemrogamanmobile.myfavoritegames.databinding.FragmentAffirmationListBinding
11
12 class AffirmationListFragment : Fragment() {
13     private var _binding: FragmentAffirmationListBinding? = null
14     private val binding get() = _binding!!
15
16     override fun onCreateView(
17         inflater: LayoutInflater, container: ViewGroup?, saved: Bundle?
18     ) = FragmentAffirmationListBinding.inflate(inflater, container, false)
19         .also { _binding = it }
20         .root
21
22     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
23         val games = DataSource.loadGames()
24         binding.recyclerView.adapter = AffirmationAdapter(games) { affirmation ->
25             val action = AffirmationListFragmentDirections
26                 .actionListToDetail(affirmation)
27             findNavController().navigate(action)
28         }
29     }
30
31     override fun onDestroyView() {
32         super.onDestroyView()
33         _binding = null
34     }
35 }
```

DetailFragment.kt

Table 5 source code soal 1 DetailFragment.kt (xml)

```
1  package com.pemrogamanmobile.myfavoritegames.ui
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 androidx.navigation.fragment.navArgs
9  import com.pemrogamanmobile.myfavoritegames.databinding.FragmentDetailBinding
```

```
10
11  class DetailFragment : Fragment() {
12      private var _binding: FragmentDetailBinding? = null
13      private val binding get() = _binding!!
14
15      private val args: DetailFragmentArgs by navArgs()
16
17      override fun onCreateView(
18          inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?
19      ) = FragmentDetailBinding.inflate(inflater, container, false)
20          .also { _binding = it }
21          .root
22
23      override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
24          val affirmation = args.affirmation
25          with(binding) {
26              detailImageView.setImageResource(affirmation.imageResourceId)
27              detailTitleText.text  = getString(affirmation.titleResourceId)
28              detailYearText.text   = getString(affirmation.yearResourceId)
29              detailDescText.text   = getString(affirmation.detailResourceId)
30          }
31      }
32
33      override fun onDestroyView() {
34          super.onDestroyView()
35          _binding = null
36      }
37  }
```

MainActivity.kt

Table 6 source code soal 1 mainactivity.kt (xml)

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

```
15  }
```

activity_main.xml

Table 7 source code soal 1 acivity_main.xml (xml)

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <androidx.fragment.app.FragmentContainerView
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       xmlns:app="http://schemas.android.com/apk/res-auto"
5       android:id="@+id/nav_host_fragment"

6   android:name="androidx.navigation.fragment.NavHostFragment"
7       android:layout_width="match_parent"
8       android:layout_height="match_parent"
9       app:navGraph="@navigation/nav_graph"
10      app:defaultNavHost="true"/>
```

Fragment_affirmation_list.xml

Table 8 source code soal 1 fragment_affirmation_list.xml (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:background="#000000"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent">
8
9     <androidx.recyclerview.widget.RecyclerView
10      android:id="@+id/recyclerView"
11      android:contentDescription="@string/affirmation_list_desc"
12      android:layout_width="0dp"
13      android:layout_height="0dp"
14      android:padding="8dp"
15      app:layoutManager="LinearLayoutManager"
16      app:layout_constraintTop_toTopOf="parent"
17      app:layout_constraintBottom_toBottomOf="parent"
18      app:layout_constraintStart_toStartOf="parent"
19      app:layout_constraintEnd_toEndOf="parent"/>
20  </androidx.constraintlayout.widget.ConstraintLayout>
```

Fragment_detail.xml

Table 9 source code soal 1 fragment_detail.xml (xml)

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     android:background="#131418"
```

```
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <LinearLayout
9        android:orientation="vertical"
10       android:padding="16dp"
11       android:layout_width="match_parent"
12       android:layout_height="wrap_content">
13
14       <com.google.android.material.imageview.ShapeableImageView
15          android:id="@+id/detailImageView"
16          android:layout_width="match_parent"
17          android:layout_height="400dp"
18          android:scaleType="centerCrop"

19    app:shapeAppearanceOverlay="@style/ShapeAppearance.App.Rounded6dp" />
20
21       <LinearLayout
22          android:layout_width="match_parent"
23          android:layout_height="wrap_content"
24          android:orientation="horizontal"
25          android:gravity="center_vertical"
26          android:layout_marginTop="12dp">
27
28          <TextView
29             android:id="@+id/detailTitleText"
30             android:layout_width="0dp"
31             android:layout_weight="1"
32             android:layout_height="wrap_content"
33             android:textColor="#FFFFFF"
34             android:textSize="26sp"
35             android:textStyle="bold"/>
36
37          <TextView
38             android:id="@+id/detailYearText"
39             android:layout_width="wrap_content"
40             android:layout_height="wrap_content"
41             android:textColor="#7C7C86"
42             android:textSize="14sp"/>
43       </LinearLayout>
44
45       <TextView
46          android:layout_marginTop="8dp"
47          android:layout_width="wrap_content"
48          android:layout_height="wrap_content"
49          android:text="tentang game ini:"
50          android:textColor="#FFFFFF"
```

```
51          android:textSize="14sp"
52          android:textStyle="bold"/>
53
54      <TextView
55          android:id="@+id/detailDescText"
56          android:layout_marginTop="4dp"
57          android:layout_width="match_parent"
58          android:layout_height="wrap_content"
59          android:textColor="#FFFFFF"
60          android:textSize="14sp"/>
61    </LinearLayout>
62 </ScrollView>
```

Item_affirmation.kt

Table 10 source code soal 1 item_affirmation.xml (xml)

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.cardview.widget.CardView
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="wrap_content"
7     android:layout_margin="4dp"
8     app:cardCornerRadius="6dp"
9     app:cardBackgroundColor="#343539">
10
11    <LinearLayout
12       android:layout_width="match_parent"
13       android:layout_height="wrap_content"
14       android:orientation="horizontal"
15       android:padding="8dp"
16       android:gravity="center_vertical">
17
18       <com.google.android.material.imageview.ShapeableImageView
19          android:id="@+id/imageView"
20          android:layout_width="100dp"
21          android:layout_height="140dp"
22          android:layout_marginEnd="12dp"
23          android:scaleType="centerCrop"

24 app:shapeAppearanceOverlay="@style/ShapeAppearance.App.Rounded6dp"
25          />
26
27       <LinearLayout
28          android:layout_width="0dp"
29          android:layout_weight="1"
30          android:orientation="vertical"
31          android:layout_height="wrap_content">
```

```
32
33          <LinearLayout
34              android:layout_width="match_parent"
35              android:layout_height="wrap_content"
36              android:orientation="horizontal"
37              android:gravity="center_vertical">
38
39              <TextView
40                  android:id="@+id/titleText"
41                  android:layout_width="0dp"
42                  android:layout_weight="1"
43                  android:layout_height="wrap_content"
44                  android:textColor="@android:color/white"
45                  android:textSize="20sp"
46                  android:textStyle="bold"/>
47
48              <TextView
49                  android:id="@+id/yearText"
50                  android:layout_width="wrap_content"
51                  android:layout_height="wrap_content"
52                  android:textColor="#7C7C86"
53                  android:textSize="14sp"/>
54          </LinearLayout>
55
56          <TextView
57              android:layout_marginTop="8dp"
58              android:layout_width="wrap_content"
59              android:layout_height="wrap_content"
60              android:text="tentang game ini:"
61              android:textColor="@android:color/white"
62              android:textSize="14sp"
63              android:textStyle="bold"/>
64
65          <TextView
66              android:id="@+id/detailText"
67              android:layout_marginTop="4dp"
68              android:layout_width="wrap_content"
69              android:layout_height="wrap_content"
70              android:textColor="@android:color/white"
71              android:textSize="14sp"/>
72
73          <LinearLayout
74              android:layout_width="match_parent"
75              android:layout_height="wrap_content"
76              android:orientation="horizontal"
77              android:gravity="end"
78              android:layout_marginTop="12dp">
```

```
79
80              <Button
81                 android:id="@+id/viewButton"
82                 android:layout_width="100dp"
83                 android:layout_height="44dp"
84                 android:text="View"
85                 android:backgroundTint="#b0c4ff"
86                 android:textColor="#000"/>
87
88              <Space                                    android:layout_width="8dp"
    android:layout_height="wrap_content"/>
89
90              <Button
91                 android:id="@+id/detailButton"
92                 android:layout_width="100dp"
93                 android:layout_height="44dp"
94                 android:text="Detail"
95                 android:backgroundTint="#b0c4ff"
96                 android:textColor="#000"/>
97           </LinearLayout>
98        </LinearLayout>
99     </LinearLayout>
100 </androidx.cardview.widget.CardView>
```

strings.xml

Table 11 source code soal 1 strings.xml (xml)

```
1    <resources>
2       <string name="app_name">My favorite games</string>
3
4       <string name="affirmation_list_desc">List of favorite games</string>
5
6       <string name="title1">Terraria</string>
7       <string name="title2">Mobile Legends</string>
8       <string name="title3">Firewatch</string>
9       <string name="title4">The Elder Scrolls : Skyrim</string>
10      <string name="title5">Monster Hunter World</string>
11      <string name="title6">Genshin Impact</string>
12      <string name="title7">Delta Force</string>
13      <string name="title8">Magic Chess Go Go</string>
14      <string name="title9">STAR WARS Jedi: Fallen Order</string>
15      <string name="title10">Minecraft</string>
16
17      <string name="year1">2011</string>
18      <string name="year2">2016</string>
19      <string name="year3">2016</string>
20      <string name="year4">2016</string>
```

```
    <string name="year5">2018</string>
    <string name="year6">2020</string>
    <string name="year7">2025</string>
    <string name="year8">2025</string>
    <string name="year9">2019</string>
    <string name="year10">2011</string>

    <string name="detail1">Dig, fight, explore, build! Nothing is impossible in this action-packed adver
Pack also available!</string>
    <string name="detail2">Bergabung dengan teman Anda di Mobile Legends:
Bang Bang, Game MOBA 5v5 terbaru, dan bertarung melawan Player sungguhan!
Pilih Hero favorit Anda dan bentuk tim yang sempurna dengan teman Anda!
Matchmaking 10 detik, pertandingan 10 menit. Laning, Jungling, Team Fight, dan
menghancurkan Turret, semua kesenangan dari Game aksi dan MOBA PC berada
dalam genggaman Anda! Tunjukkan semangat Esports Anda!</string>
    <string name="detail3">Firewatch is a single-player first-person mystery set in the Wyoming wi
your only emotional lifeline is the person on the other end of a handheld radio.</string>
    <string name="detail4">Winner of more than 200 Game of the Year Awards, The Elder Scrolls V:
Edition brings the epic fantasy to life in stunning detail. The Special Edition includes the critically accl
add-ons with all-new features.</string>
    <string name="detail5">Welcome to a new world! In Monster Hunter: World, the latest installme
you can enjoy the ultimate hunting experience, using everything at your disposal to hunt monsters
teeming with surprises and excitement.</string>
    <string name="detail6">Genshin Impact adalah game RPG Open World terbaru. Kamu adalah seora
yang menjelajah sebuah dunia fantasi yang teramat luas, kamu dapat menjelajahi tujuh bangsa, be
karakter berkemampuan dan berkepribadian unik, melawan musuh-musuh kuat, dan berkelana men
yang hilang. Di dunia yang luas ini, biarkan rasa keingintahuan membawamu menguak miste
tersembunyi, sampai akhirnya kamu dapat bertemu kembali dengan saudaramu, dan menyaksikan
perjalananmu.</string>
    <string name="detail7">Pertempuran Epic Warfare 24vs24, Rasakan map besar, puluhan senjata,
permainan menarik, dan update yang terus menerus! Warfare membawa kalian ke medan perang dina
kekacauan dan kehancuran environment. Baik kalian menembaki musuh di darat, laut, dan udara, atau
rekan satu tim sebagai medic, pilih role kalian dan raih kemenangan!</string>
    <string name="detail8">Magic Chess: Go Go - Game strategi multiplayer baru yang terinspira
Legends: Bang Bang. Dengan gameplay seperti catur, game ini kasual dan mudah dimainkan kapan sa
bersama teman-teman! Di sini, kemenangan bergantung pada strategi dan sedikit keberuntungan daripa
pengendalian mikro. Di setiap babak, kamu bisa mengendalikan Commander untuk membeli dan men
menyusun Sinergi, menggunakan equipment, dan menempatkan hero dengan cerdik untuk meng
Kalahkan 7 player lain secara bertahap untuk memenangkan pertandingan.</string>
    <string name="detail9">A galaxy-spanning adventure awaits in Star Wars Jedi: Fallen Order, a 3r
adventure title from Respawn. An abandoned Padawan must complete his training, develop new
abilities, and master the art of the lightsaber - all while staying one step ahead of the Empire.</string>
    <string name="detail10">Bangun, jelajahi, bertahan hidup! Main dengan teman dan buat duniamu
dunia terbuka dengan membangun, membuat item, dan bertahan hidup di game sandbox memb
Kumpulkan sumber daya, coba bertahan hidup di malam hari, dan rancang petualangan epik balok den
dan buat item sesuka hati di dunia yang terbuka sepenuhnya tempat kamu bisa bermain dengan tem
```

```
38      kota balok, membuka peternakan, menambang jauh ke bawah tanah, menghadapi musuh misteriu
39      bereksperimen sejauh imajinasi membawamu!</string>
40
41          <string name="link1">https://store.steampowered.com/app/105600/Terraria/</string>
42          <string name="link2">https://play.google.com/store/apps/details?id=com.mobile.legends</string>
            <string name="link3">https://store.steampowered.com/app/383870/Firewatch/</string>
43          <string
44      name="link4">https://store.steampowered.com/app/489830/The_Elder_Scrolls_V_Skyrim_Special_Ed
45          <string name="link5">https://store.steampowered.com/app/582010/Monster_Hunter_World/</string
46          <string name="link6">https://play.google.com/store/apps/details?id=com.miHoYo.GenshinImpact<
47          <string name="link7">https://play.google.com/store/apps/details?id=com.garena.game.df</string>
48          <string name="link8">https://play.google.com/store/apps/details?id=com.mobilechess.gp</string>
49          <string name="link9">https://store.steampowered.com/app/1172380/STAR_WARS_Jedi_Fallen_O
            <string name="link10">https://play.google.com/store/apps/details?id=com.mojang.minecraftpe</str
        </resources>
```

stlyes.xml

Table 12 source code soal 1 style.xml (xml)

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <resources>
3       <style name="ShapeAppearance.App.Rounded6dp" parent="">
4           <item name="cornerFamily">rounded</item>
5           <item name="cornerSize">6dp</item>
6       </style>
7   </resources>
```

COMPOSE :

MainActivity.kt

Table 13 source code soal 1 mainactivity.kt (compose)

```
1   package com.pemrogamanmobile.myfavoritegames
2
3   import android.os.Bundle
4   import androidx.activity.ComponentActivity
5   import androidx.activity.compose.setContent
6   import androidx.compose.material3.Surface
7   import androidx.compose.ui.graphics.Color
8   import androidx.navigation.NavType
9   import androidx.navigation.compose.NavHost
10  import androidx.navigation.compose.composable
11  import androidx.navigation.compose.rememberNavController
12  import androidx.navigation.navArgument
13  import
    com.pemrogamanmobile.myfavoritegames.ui.theme.MyFavoriteGames
    Theme
14
15  class MainActivity : ComponentActivity() {
```

```
16      override fun onCreate(savedInstanceState: Bundle?) {
17          super.onCreate(savedInstanceState)
18          setContent {
19              MyFavoriteGamesTheme {
20                  Surface(color = Color(0xFF131418)) {
21                      val            navController            =
22  rememberNavController()
23
24                      NavHost(navController   =   navController,
25  startDestination = "home") {
26                          composable("home") {
27                              HomeScreen(onDetailClick    =    {
28  affirmation ->
29                                  navController.navigate(
30
31  "detail/${affirmation.titleResourceId}/${affirmation.imageRes
32  ourceId}/${affirmation.yearResourceId}/${affirmation.detailRe
33  sourceId}"
34                                  )
35                              })
36                          }
37
38                          composable(
39                              route                              =
40  "detail/{titleResId}/{imageResId}/{yearResId}/{detailResId}",
41                              arguments = listOf(
42                                  navArgument("titleResId")    {
43  type = NavType.IntType },
44                                  navArgument("imageResId")    {
45  type = NavType.IntType },
46                                  navArgument("yearResId")     {
47  type = NavType.IntType },
48                                  navArgument("detailResId")   {
49  type = NavType.IntType }
50                              )
51                          ) { backStackEntry ->
52                              val          titleResId          =
53  backStackEntry.arguments?.getInt("titleResId") ?: 0
54                              val          imageResId          =
55  backStackEntry.arguments?.getInt("imageResId") ?: 0
56                              val          yearResId           =
57  backStackEntry.arguments?.getInt("yearResId") ?: 0
58                              val         detailResId          =
59  backStackEntry.arguments?.getInt("detailResId") ?: 0
60
61                              DetailScreen(
62                                  titleResId = titleResId,
63                                  imageResId = imageResId,
64                                  yearResId = yearResId,
65                                  detailResId = detailResId
66                              )
67                          }
```

```
                        }
                    }
                }
            }
        }
}
```

HomeScreen.kt

Table 14 source code soal 1 HomeScreen.kt (compose)

```
package com.pemrogamanmobile.myfavoritegames

import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier

import
com.pemrogamanmobile.myfavoritegames.data.Datasource
import
com.pemrogamanmobile.myfavoritegames.model.Affirmation

@Composable
fun HomeScreen(
    modifier: Modifier = Modifier,
    onDetailClick: (Affirmation) -> Unit
) {
    val affirmations = Datasource().loadAffirmations()

    LazyColumn(modifier = modifier) {
        items(affirmations) { affirmation ->
            AffirmationCard(affirmation  =  affirmation,
onDetailClick = onDetailClick)
        }
    }
}
```

DetailScreen.kt

Table 15 source code soal 1 DetailScreen.kt (compose)

```
package com.pemrogamanmobile.myfavoritegames

import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import
androidx.compose.foundation.shape.RoundedCornerShape
```

```
6  import androidx.compose.material3.*
7  import androidx.compose.runtime.Composable
8  import androidx.compose.ui.Alignment
9  import androidx.compose.ui.Modifier
10 import androidx.compose.ui.draw.clip
11 import androidx.compose.ui.graphics.Color
12 import androidx.compose.ui.res.painterResource
13 import androidx.compose.ui.res.stringResource
14 import androidx.compose.ui.unit.dp
15 import androidx.compose.ui.layout.ContentScale
16 import androidx.compose.ui.text.TextStyle
17 import androidx.compose.ui.text.font.FontWeight
18 import androidx.compose.ui.unit.sp
19
20 @Composable
21 fun DetailScreen(titleResId: Int, imageResId: Int,
   yearResId: Int, detailResId: Int) {
22     Surface(
23         modifier = Modifier.fillMaxSize(),
24         color = Color(0xFF131418)
25     ) {
26         Column(modifier = Modifier.padding(16.dp)) {
27
28             Image(
29                 painter = painterResource(imageResId),
30                 contentDescription                    =
   stringResource(detailResId),
31                 modifier = Modifier
32                     .fillMaxWidth()
33                     .height(400.dp)
34                     .clip(RoundedCornerShape(6.dp)),
35                 contentScale = ContentScale.Crop
36             )
37
38             Column(modifier = Modifier.fillMaxWidth()) {
39                 Row(
40                     modifier = Modifier.fillMaxWidth(),
41                     horizontalArrangement                =
   Arrangement.SpaceBetween
42                 ) {
43                     Text(
44                         text                             =
   stringResource(titleResId),
45                         style  =  TextStyle(fontSize  =
   26.sp,  fontWeight  =  FontWeight.Bold,  color  =
   Color.White),
46                         modifier = Modifier.weight(1f)
```

```
47                      )
48                      Text(
49                          text                                =
    stringResource(yearResId),
50                          style   =   TextStyle(fontSize   =
    16.sp, color = Color(0xFF7C7C86)),
51                          modifier                            =
    Modifier.align(Alignment.CenterVertically)
52                      )
53                  }
54
55                  Spacer(modifier = Modifier.height(8.dp))
56
57                  Text(
58                      text = "tentang game ini:",
59                      style = TextStyle(fontSize = 14.sp,
    fontWeight = FontWeight.Bold, color = Color.White)
60                  )
61
62                  Text(
63                      text = stringResource(detailResId),
64                      style = TextStyle(fontSize = 14.sp,
    color = Color.White),
65                      modifier  =  Modifier.padding(top  =
    4.dp)
66                  )
67              }
68          }
69      }
70  }
```

AffirmationCard.kt

Table 16 source code soal 1 AffirmationCard.kt (compose)

```
1   package com.pemrogamanmobile.myfavoritegames
2
3   import android.content.Intent
4   import androidx.compose.foundation.Image
5   import androidx.compose.foundation.layout.*
    import
6   androidx.compose.foundation.shape.RoundedCornerShape
7   import androidx.compose.material3.*
8   import androidx.compose.runtime.Composable
9   import androidx.compose.ui.Alignment
10  import androidx.compose.ui.Modifier
11  import androidx.compose.ui.draw.clip
```

```
12 import androidx.compose.ui.graphics.Color
13 import androidx.compose.ui.platform.LocalContext
14 import androidx.compose.ui.res.painterResource
15 import androidx.compose.ui.res.stringResource
16 import androidx.compose.ui.text.TextStyle
17 import androidx.compose.ui.text.font.FontWeight
18 import androidx.compose.ui.unit.dp
19 import androidx.compose.ui.unit.sp
20 import androidx.core.net.toUri
21 import
   com.pemrogamanmobile.myfavoritegames.model.Affirmation
22 import androidx.compose.ui.layout.ContentScale
23
24 @Composable
25 fun AffirmationCard(
26     affirmation: Affirmation,
27     modifier: Modifier = Modifier,
28     onDetailClick: (Affirmation) -> Unit
29 ) {
30     val context = LocalContext.current
31
32     Card(
33         modifier = modifier.padding(12.dp),
34         colors = CardDefaults.cardColors(containerColor
   = Color(0xFF343539))
35     ) {
36         Row(modifier = Modifier.fillMaxWidth()) {
37
38             Image(
39                 painter                                =
   painterResource(affirmation.imageResourceId),
40                 contentDescription                     =
   stringResource(affirmation.titleResourceId),
41                 modifier = Modifier
42                     .padding(start = 12.dp)
43                     .width(100.dp)
44                     .height(140.dp)
45                     .clip(RoundedCornerShape(6.dp))
46                     .align(Alignment.CenterVertically),
47                 contentScale = ContentScale.Crop
48             )
49
50             Column(
51                 modifier = Modifier
52                     .padding(16.dp)
53                     .fillMaxWidth(),
54
```

```
                verticalArrangement =
55  Arrangement.SpaceBetween
56          ) {
57              Row(
58                  modifier = Modifier.fillMaxWidth(),
                    horizontalArrangement =
59  Arrangement.SpaceBetween,
60                  verticalAlignment = Alignment.Top
61              ) {
62                  Text(
                        text =
63  context.getString(affirmation.titleResourceId),
                        style = TextStyle(fontSize =
    26.sp, fontWeight = FontWeight.Bold, color =
64  Color.White),
65                      modifier = Modifier.weight(1f)
66                  )
67
                    Spacer(modifier =
68  Modifier.width(8.dp))
69
70                  Text(
                        text =
71  context.getString(affirmation.yearResourceId),
                        style = TextStyle(fontSize =
72  16.sp, color = Color(0xFF7C7C86)),
                        modifier = Modifier
73
    .wrapContentWidth(Alignment.End)
74
75  .align(Alignment.CenterVertically)
76                  )
77              }
78
                Spacer(modifier =
79  Modifier.height(8.dp))
80
                Column(modifier =
81  Modifier.fillMaxWidth()) {
82                  Text(
                        text = "tentang game ini:",
                        style = TextStyle(fontSize =
83  14.sp, fontWeight = FontWeight.Bold, color =
    Color.White)
                    )
84                  Text(
85
```

```
86                          text                                    =
    context.getString(affirmation.detailResourceId),
87                          style  =  TextStyle(fontSize  =
    14.sp, color = Color.White),
88                          modifier = Modifier.padding(top
    = 4.dp)
89                      )
90                  }
91
92                  Spacer(modifier                                 =
    Modifier.height(12.dp))
93
94                  Row(
95                      modifier = Modifier.fillMaxWidth(),
96                      horizontalArrangement                       =
    Arrangement.End,
97                          verticalAlignment                       =
    Alignment.CenterVertically
98                  ) {
99                      Button(
100                         onClick = {
101                             val          url          =
    context.getString(affirmation.steamLinkResourceId)
102                             val        intent         =
    Intent(Intent.ACTION_VIEW, url.toUri())

103 context.startActivity(intent)
104                         },
105                         modifier                                =
    Modifier.width(100.dp).height(44.dp),
106                         colors                                  =
    ButtonDefaults.buttonColors(containerColor                  =
    Color(0xFFb0c4ff), contentColor = Color.White)
107                     ) {
108                         Text("View",       style       =
    TextStyle(fontSize = 16.sp, color = Color.Black))
109                     }
110
111                     Spacer(modifier                             =
    Modifier.width(8.dp))
112
113                     Button(
114                         onClick          =          {
    onDetailClick(affirmation) },
115                         modifier                                =
    Modifier.width(100.dp).height(44.dp),
116
```

```
                        colors                                  =
    ButtonDefaults.buttonColors(containerColor                  =
117 Color(0xFFb0c4ff), contentColor = Color.White)
118                     ) {
                            Text("Detail",      style       =
119 TextStyle(fontSize = 16.sp, color = Color.Black))
120                     }
121                 }
122             }
123         }
124     }
    }
```

Datasource.kt

Table 17 source code soal 1 Datasource.kt (compose)

```
1   package com.pemrogamanmobile.myfavoritegames.data
2   import com.pemrogamanmobile.myfavoritegames.R
3   import
    com.pemrogamanmobile.myfavoritegames.model.Affirmation
4
5   class Datasource {
6       fun loadAffirmations(): List<Affirmation> {
7           return listOf<Affirmation>(
8               Affirmation(R.string.detail1,
    R.drawable.image1,    R.string.link1,    R.string.year1,
    R.string.title1),
9               Affirmation(R.string.detail2,
    R.drawable.image2,    R.string.link2,    R.string.year2,
    R.string.title2),
10              Affirmation(R.string.detail3,
    R.drawable.image3,    R.string.link3,    R.string.year3,
    R.string.title3),
11              Affirmation(R.string.detail4,
    R.drawable.image4,    R.string.link4,    R.string.year4,
    R.string.title4),
12              Affirmation(R.string.detail5,
    R.drawable.image5,    R.string.link5,    R.string.year5,
    R.string.title5),
13              Affirmation(R.string.detail6,
    R.drawable.image6,    R.string.link6,    R.string.year6,
    R.string.title6),
                Affirmation(R.string.detail7,
    R.drawable.image7,    R.string.link7,    R.string.year7,
14  R.string.title7),
```

```
            Affirmation(R.string.detail8,
R.drawable.image8,    R.string.link8,    R.string.year8,
R.string.title8),
            Affirmation(R.string.detail9,
R.drawable.image9,    R.string.link9,    R.string.year9,
R.string.title9),
            Affirmation(R.string.detail10,
R.drawable.image10,  R.string.link10,  R.string.year10,
R.string.title10))
    }
}

```

strings.xml

Table 18 source code soal 1 strings.xml (compose)

```
   <resources>
       <string name="app_name">my favorite games</string>

       <string name="title1">Terraria</string>
       <string name="title2">Mobile Legends</string>
       <string name="title3">Firewatch</string>
       <string   name="title4">The   Elder   Scrolls   :
   Skyrim</string>
       <string       name="title5">Monster       Hunter
   World</string>
       <string name="title6">Genshin Impact</string>
       <string name="title7">Delta Force</string>
       <string name="title8">Magic Chess Go Go</string>
       <string  name="title9">STAR  WARS  Jedi:  Fallen
   Order</string>
       <string name="title10">Minecraft</string>

       <string name="year1">2011</string>
       <string name="year2">2016</string>
       <string name="year3">2016</string>
       <string name="year4">2016</string>
       <string name="year5">2018</string>
       <string name="year6">2020</string>
       <string name="year7">2025</string>
       <string name="year8">2025</string>
       <string name="year9">2019</string>
       <string name="year10">2011</string>


```

```
    <string name="detail1">Dig, fight, explore, build!
27  Nothing is impossible in this action-packed adventure
    game. Four Pack also available!</string>
    <string name="detail2">Bergabung dengan teman Anda
    di Mobile Legends: Bang Bang, Game MOBA 5v5 terbaru,
    dan bertarung melawan Player sungguhan! Pilih Hero
    favorit Anda dan bentuk tim yang sempurna dengan teman
    Anda! Matchmaking 10 detik, pertandingan 10 menit.
    Laning, Jungling, Team Fight, dan menghancurkan
    Turret, semua kesenangan dari Game aksi dan MOBA PC
28  berada dalam genggaman Anda! Tunjukkan semangat
    Esports Anda!</string>
    <string name="detail3">Firewatch is a single-
    player first-person mystery set in the Wyoming
29  wilderness, where your only emotional lifeline is the
    person on the other end of a handheld radio.</string>
    <string name="detail4">Winner of more than 200
    Game of the Year Awards, The Elder Scrolls V: Skyrim
    Special Edition brings the epic fantasy to life in
    stunning detail. The Special Edition includes the
30  critically acclaimed game and add-ons with all-new
    features.</string>
    <string name="detail5">Welcome to a new world! In
    Monster Hunter: World, the latest installment in the
    series, you can enjoy the ultimate hunting experience,
    using everything at your disposal to hunt monsters in
31  a new world teeming with surprises and
    excitement.</string>
    <string name="detail6">Genshin Impact adalah game
    RPG Open World terbaru. Kamu adalah seorang pengembara
    yang menjelajah sebuah dunia fantasi yang teramat luas,
    kamu dapat menjelajahi tujuh bangsa, bertemu berbagai
    karakter berkemampuan dan berkepribadian unik, melawan
    musuh-musuh kuat, dan berkelana mencari saudaramu yang
    hilang. Di dunia yang luas ini, biarkan rasa
    keingintahuan membawamu menguak misteri-misteri yang
    tersembunyi, sampai akhirnya kamu dapat bertemu
32  kembali dengan saudaramu, dan menyaksikan akhir dari
    kisah perjalananmu.</string>
    <string name="detail7">Pertempuran Epic Warfare
    24vs24, Rasakan map besar, puluhan senjata, berbagai
    mode permainan menarik, dan update yang terus menerus!
    Warfare membawa kalian ke medan perang dinamis yang
    penuh kekacauan dan kehancuran environment. Baik
    kalian menembaki musuh di darat, laut, dan udara, atau
33  menyelamatkan rekan satu tim sebagai medic, pilih role
    kalian dan raih kemenangan!</string>
```

```
    <string name="detail8">Magic Chess: Go Go - Game strategi multiplayer baru yang terinspirasi dari Mobile Legends: Bang Bang. Dengan gameplay seperti catur, game ini kasual dan mudah dimainkan kapan saja, di mana saja bersama teman-teman! Di sini, kemenangan bergantung pada strategi dan sedikit keberuntungan daripada keterampilan pengendalian mikro. Di setiap babak, kamu bisa mengendalikan Commander untuk membeli dan meningkatkan Hero, menyusun Sinergi, menggunakan equipment, dan menempatkan hero dengan cerdik untuk
34  mengalahkan lawan. Kalahkan 7 player lain secara bertahap untuk memenangkan pertandingan.</string>
    <string name="detail9">A galaxy-spanning adventure awaits in Star Wars Jedi: Fallen Order, a 3rd person action-adventure title from Respawn. An abandoned Padawan must complete his training, develop new powerful Force abilities, and master the art of the
35  lightsaber - all while staying one step ahead of the Empire.</string>
    <string name="detail10">Bangun, jelajahi, bertahan hidup! Main dengan teman dan buat duniamu sendiri! Jelajahi dunia terbuka dengan membangun, membuat item, dan bertahan hidup di game sandbox membangun terbaik. Kumpulkan sumber daya, coba bertahan hidup di malam hari, dan rancang petualangan epik balok demi balok. Jelajahi dan buat item sesuka hati di dunia yang terbuka sepenuhnya tempat kamu bisa bermain dengan teman, membangun kota balok, membuka peternakan, menambang jauh ke bawah tanah, menghadapi musuh misterius, atau sekadar bereksperimen sejauh imajinasi membawamu!</string>
36
    <string
    name="link1">https://store.steampowered.com/app/10560
37  0/Terraria/</string>
    <string
    name="link2">https://play.google.com/store/apps/detail
38  s?id=com.mobile.legends</string>
    <string
    name="link3">https://store.steampowered.com/app/383870
39  /Firewatch/</string>
    <string
    name="link4">https://store.steampowered.com/app/489830
40  /The_Elder_Scrolls_V_Skyrim_Special_Edition/</string>
    <string
    name="link5">https://store.steampowered.com/app/582010
41  /Monster_Hunter_World/</string>
```

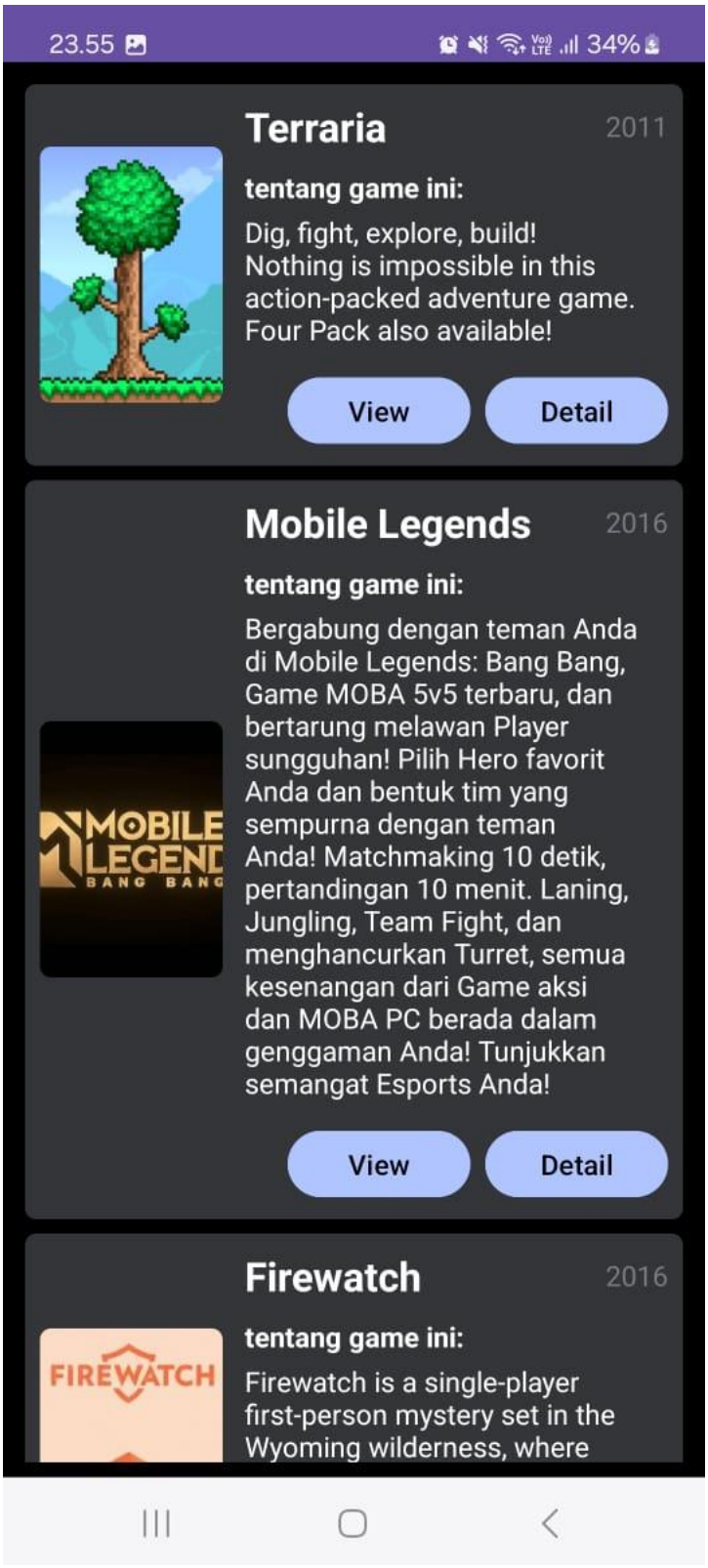
```
        <string
    name="link6">https://play.google.com/store/apps/detail
42  s?id=com.miHoYo.GenshinImpact</string>
        <string
    name="link7">https://play.google.com/store/apps/detail
43  s?id=com.garena.game.df</string>
        <string
    name="link8">https://play.google.com/store/apps/detail
44  s?id=com.mobilechess.gp</string>
        <string
    name="link9">https://store.steampowered.com/app/117238
45  0/STAR_WARS_Jedi_Fallen_Order/</string>
        <string
    name="link10">https://play.google.com/store/apps/detai
46  ls?id=com.mojang.minecraftpe</string>
    </resources>
```

Affirmation.kt

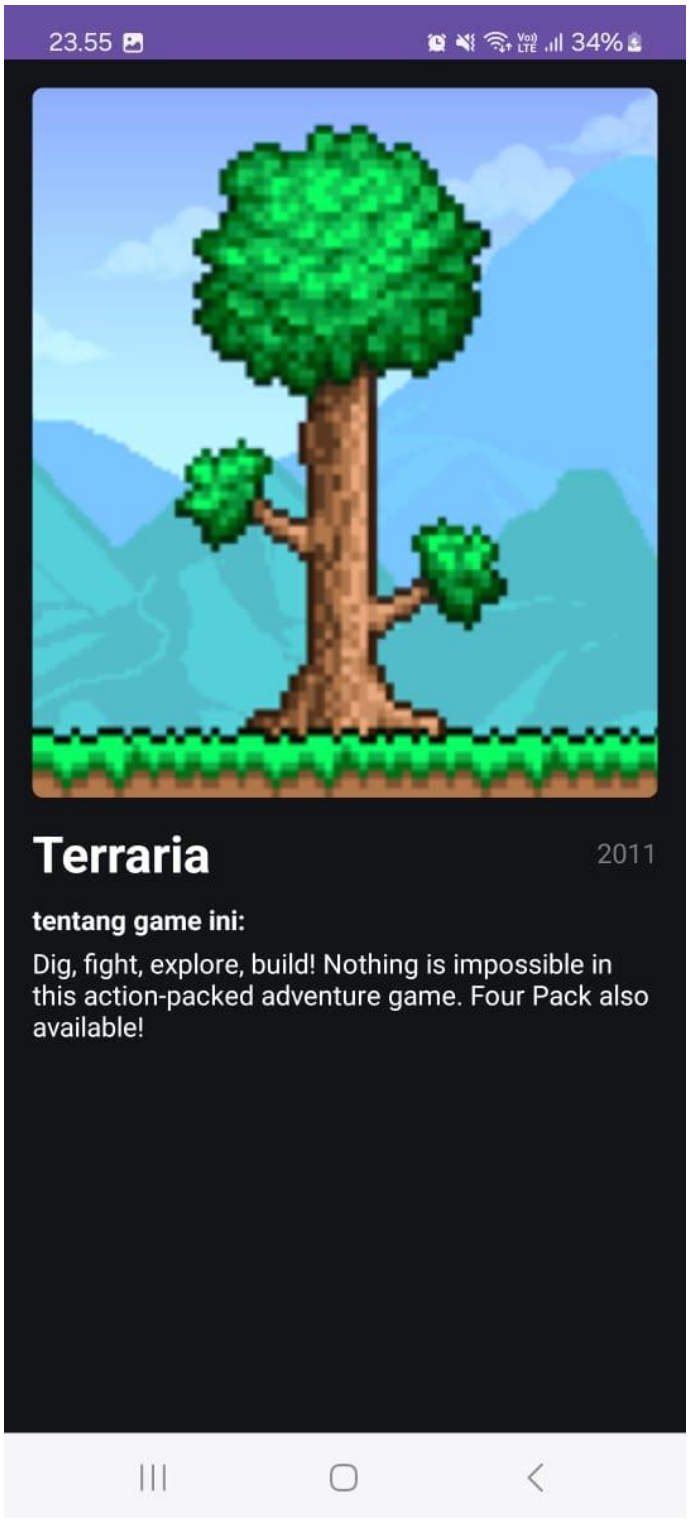
Table 19 source code soal 1 Affirmation.kt (compose)

```
1   package com.pemrogamanmobile.myfavoritegames.model
2   import androidx.annotation.DrawableRes
3   import androidx.annotation.StringRes
4
5   data class Affirmation(
6       @StringRes val detailResourceId: Int,
7       @DrawableRes val imageResourceId: Int,
8       @StringRes val steamLinkResourceId: Int,
9       @StringRes val yearResourceId: Int,
10      @StringRes val titleResourceId: Int,
11  )
```

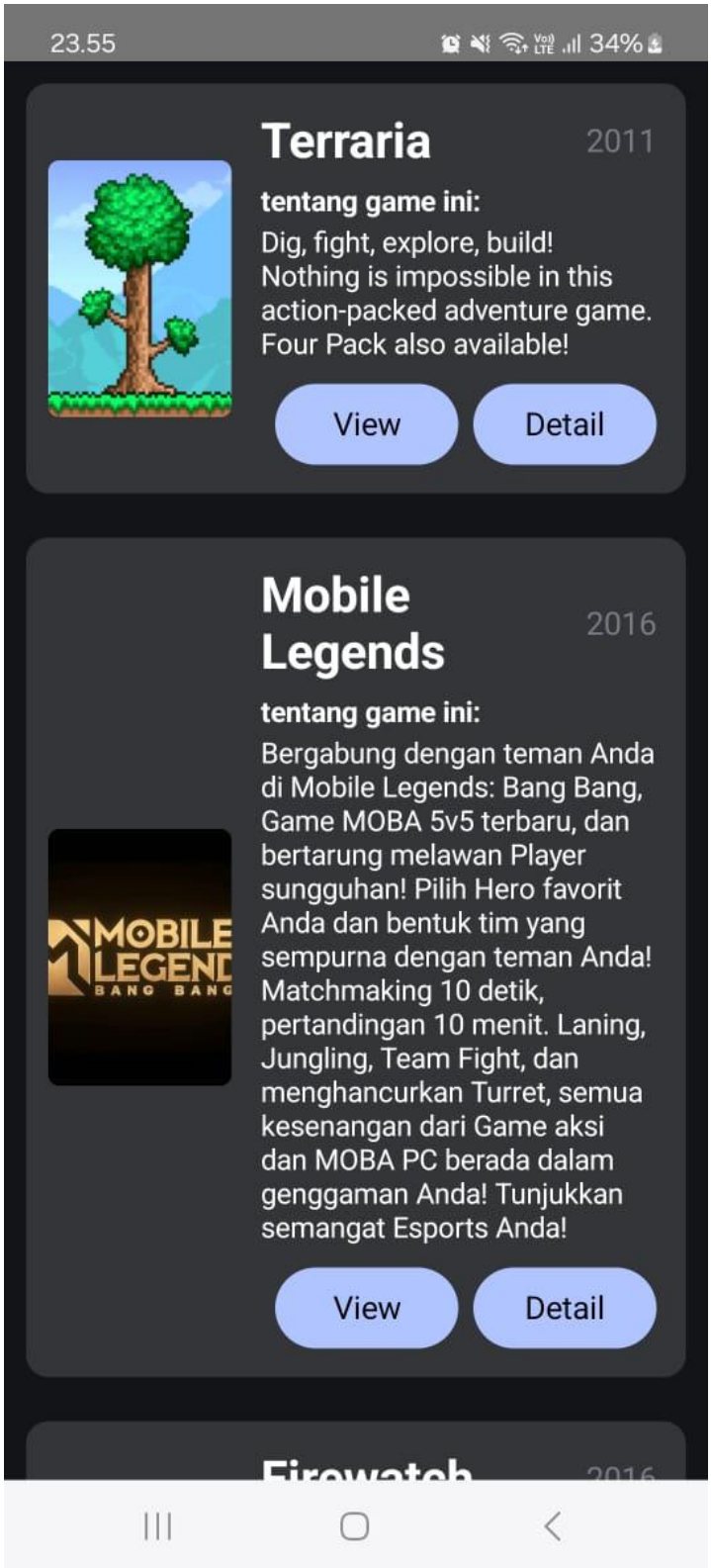
B. Output Program



Gambar 3. Screenshot Hasil Jawaban Soal 1 XML bagian a



Gambar 4. Screenshot Hasil Jawaban Soal 1 XML bagian b



Gambar 5. Screenshot Hasil Jawaban Soal 1 compose bagian a

Gambar 6. Screenshot Hasil Jawaban Soal 1 compose bagian b

C. Pembahasan

XML :

Affirmation.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames.data yang berfungsi sebagai nama paket dari aplikasi dan tempat menyimpan file ini agar terorganisir sesuai struktur proyek Android.

Pada baris ke [3] terdapat import java.io.Serializable yang berfungsi untuk mengimport kelas Serializable, agar objek dari kelas ini bisa dikirim antar activity melalui intent.

Pada baris ke [5] terdapat data class Affirmation(yang berfungsi untuk membuat kelas data bernama Affirmation, yang secara otomatis menyediakan fungsi seperti toString(), equals(), hashCode(), dan copy().

Pada baris ke [6] sampai [10] berfungsi untuk mendeklarasikan properti dalam kelas Affirmation, yaitu detailResourceId, imageResourceId, steamLinkResourceId, yearResourceId, dan titleResourceId yang masing-masing bertipe Int dan berfungsi sebagai ID dari resource yang menyimpan informasi detail teks, gambar game, tautan Steam, tahun rilis, serta judul game yang akan digunakan pada aplikasi.

Pada baris ke [11] terdapat) : Serializable yang berfungsi untuk menjadikan kelas Affirmation sebagai objek yang dapat diserialisasi agar bisa dikirim melalui Intent.

Datasource.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames.data yang berfungsi sebagai nama paket dari aplikasi dan tempat menyimpan file ini agar terorganisir sesuai struktur proyek Android.

Pada baris ke [3] terdapat import com.pemrogamanmobile.myfavoritegames.R yang berfungsi untuk mengakses resource seperti string dan drawable yang terdapat di dalam file res aplikasi Android ini.

Pada baris ke [5] terdapat object DataSource yang berfungsi untuk mendefinisikan object singleton bernama DataSource, artinya hanya ada satu instance dari objek ini yang akan digunakan sepanjang aplikasi berjalan.

Pada baris ke [6] terdapat fun loadGames(): List<Affirmation> = listOf(yang berfungsi untuk mendeklarasikan fungsi bernama loadGames yang mengembalikan daftar dari objek Affirmation dalam bentuk List.

Pada baris ke [7] sampai [16] berfungsi untuk mengisi list tersebut dengan sepuluh objek Affirmation yang masing-masing berisi lima parameter, yaitu detail, image, link ke Steam, tahun rilis, dan judul game. Kelima parameter tersebut merujuk pada ID resource yang disimpan dalam file strings.xml dan drawable di folder res. Setiap baris merepresentasikan satu game favorit yang akan ditampilkan di aplikasi.

Pada baris ke [17] terdapat tanda) yang berfungsi untuk menutup fungsi listOf yang telah dibuat sebelumnya.

AffirmationAdapter.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames.ui yang berfungsi sebagai nama paket dari file ini dan menunjukkan bahwa file ini berada dalam folder ui (user interface).

Pada baris ke [3] sampai [9], berfungsi untuk mengimpor beberapa kelas yang digunakan, yaitu Intent dan Uri untuk membuka tautan eksternal, LayoutInflater dan ViewGroup untuk membuat tampilan, RecyclerView untuk membuat daftar scrollable,

Affirmation sebagai data model, dan ItemAffirmationBinding untuk menghubungkan data dengan layout XML menggunakan View Binding.

Pada baris ke [11] terdapat class AffirmationAdapter(yang berfungsi untuk mendeklarasikan kelas AffirmationAdapter sebagai adapter untuk RecyclerView.

Pada baris ke [12] sampai [13] berfungsi untuk mendefinisikan dua parameter konstruktor utama adapter, yaitu items sebagai list dari objek Affirmation, dan onDetailClick sebagai lambda function yang dijalankan ketika tombol detail ditekan.

Pada baris ke [14] terdapat) : RecyclerView.Adapter<AffirmationAdapter.ViewHolder>() yang berfungsi untuk menyatakan bahwa kelas ini merupakan subclass dari RecyclerView.Adapter dan akan menggunakan inner class ViewHolder untuk menampung tampilan item.

Pada baris ke [16] terdapat inner class ViewHolder(val binding: ItemAffirmationBinding) yang berfungsi untuk mendefinisikan inner class ViewHolder yang menyimpan objek binding terhadap layout XML item.

Pada baris ke [17] terdapat : RecyclerView.ViewHolder(binding.root) yang berfungsi untuk menghubungkan root view dari layout item dengan superclass ViewHolder.

Pada baris ke [19] terdapat fungsi onCreateViewHolder yang berfungsi untuk membuat ViewHolder baru setiap kali RecyclerView membutuhkan item baru untuk ditampilkan.

Pada baris ke [20] sampai [21] digunakan LayoutInflater.from(parent.context) untuk mengakses context dan ItemAffirmationBinding.inflate untuk mengikat tampilan.

Pada baris ke [23] mengembalikan ViewHolder dengan binding tersebut.

Pada baris ke [26] terdapat fungsi onBindViewHolder yang berfungsi untuk menampilkan data ke dalam setiap item yang ada di RecyclerView.

Pada baris ke [27] terdapat val item = items[position] yang berfungsi untuk mengambil data Affirmation berdasarkan posisi item.

Pada baris ke [28] terdapat with(holder.binding) untuk mempermudah akses komponen dalam layout XML.

Pada baris ke [29] sampai [32] berfungsi untuk mengatur isi komponen seperti imageView, titleText, yearText, dan detailText dengan data dari objek Affirmation menggunakan getString() untuk mengambil string dari ID resource.

Pada baris ke [34] sampai [39] berfungsi untuk mengatur tombol viewButton agar saat diklik akan membuka link Steam yang sesuai dengan data item menggunakan Intent dengan action ACTION_VIEW dan Uri.parse().

Pada baris ke [40] sampai [42] berfungsi untuk mengatur tombol detailButton agar saat diklik akan menjalankan fungsi onDetailClick yang sebelumnya dikirim dari parameter konstruktor adapter.

Pada baris ke [46] terdapat override fun getItemCount(): Int = items.size yang berfungsi untuk mengembalikan jumlah total item dalam list yang akan ditampilkan oleh RecyclerView.

AffirmationListFragment.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames.ui yang berfungsi sebagai nama paket dari file ini dan menunjukkan bahwa file ini berada dalam folder ui (user interface).

Pada baris ke [3] sampai [10], berfungsi untuk mengimport beberapa kelas yang akan digunakan, yaitu Bundle, LayoutInflater, View, ViewGroup, Fragment, findNavController,

DataSource, dan FragmentAffirmationListBinding. Import ini mendukung pengelolaan tampilan fragment, navigasi antar fragment, pengambilan data dari DataSource, dan binding layout XML dengan View.

Pada baris ke [12] terdapat class AffirmationListFragment : Fragment() yang berfungsi untuk mendeklarasikan kelas AffirmationListFragment sebagai subclass dari Fragment, artinya kelas ini merepresentasikan salah satu tampilan layar (screen) dalam aplikasi.

Pada baris ke [13] terdapat private var _binding: FragmentAffirmationListBinding? = null yang berfungsi untuk mendeklarasikan variabel _binding yang menyimpan objek binding dari layout fragment ini, namun masih bernilai null untuk mencegah memory leak.

Pada baris ke [14] terdapat private val binding get() = _binding!! yang berfungsi untuk membuat properti akses terhadap _binding dengan jaminan bahwa nilainya tidak null, menggunakan not-null assertion (!!).

Pada baris ke [16] sampai [19] terdapat fungsi onCreateView yang berfungsi untuk meng-inflate layout fragment_affirmation_list.xml menggunakan FragmentAffirmationListBinding, dan mengatur nilai _binding.

Pada baris ke [20] berfungsi untuk mengembalikan root sebagai tampilan utama dari fragment ini.

Pada baris ke [22] terdapat fungsi onViewCreated yang berfungsi untuk menangani logika setelah tampilan fragment berhasil dibuat.

Pada baris ke [23] terdapat val games = DataSource.loadGames() yang berfungsi untuk mengambil daftar data Affirmation dari DataSource.

Pada baris ke [24] berfungsi untuk mengatur adapter dari recyclerView di layout fragment ini dengan objek AffirmationAdapter, dan juga menetapkan lambda function yang akan dipanggil saat tombol detail ditekan.

Pada baris ke [25] sampai [26] terdapat val action = AffirmationListFragmentDirections.actionListToDetail(affirmation) yang berfungsi untuk membuat objek action berdasarkan safe args yang membawa data affirmation ke fragment detail.

Pada baris ke [27] terdapat findNavController().navigate(action) yang berfungsi untuk menavigasi pengguna ke fragment detail menggunakan NavController dan action yang telah dibuat.

Pada baris ke [31] sampai [34] terdapat fungsi onDestroyView yang berfungsi untuk menghapus referensi binding saat tampilan fragment dihancurkan, guna mencegah memory leak dengan mengatur _binding = null.

DetailFragment.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames.ui yang berfungsi sebagai nama paket dari file ini dan menunjukkan bahwa file ini berada di dalam folder ui.

Pada baris ke [3] sampai [9], berfungsi untuk mengimport beberapa kelas yang digunakan, yaitu Bundle, LayoutInflater, View, ViewGroup, Fragment, navArgs, dan FragmentDetailBinding. Import ini mendukung pembuatan tampilan fragment, navigasi menggunakan Safe Args, dan binding layout XML ke dalam kode.

Pada baris ke [11] terdapat class DetailFragment : Fragment() yang berfungsi untuk mendeklarasikan kelas DetailFragment sebagai subclass dari Fragment, yang merupakan bagian dari tampilan aplikasi.

Pada baris ke [12] terdapat private var _binding: FragmentDetailBinding? = null yang berfungsi untuk mendeklarasikan variabel _binding sebagai objek binding yang akan menghubungkan kode dengan layout fragment. Nilai awalnya adalah null untuk menghindari memory leak.

Pada baris ke [13] terdapat private val binding get() = _binding!! yang berfungsi untuk membuat properti binding yang pasti tidak null saat digunakan, menggunakan not-null assertion (!!).

Pada baris ke [15] terdapat private val args: DetailFragmentArgs by navArgs() yang berfungsi untuk mengambil data dari argument navigasi menggunakan Safe Args, yaitu data Affirmation yang dikirim dari fragment sebelumnya.

Pada baris ke [17] sampai [21] terdapat fungsi onCreateView yang berfungsi untuk meng-inflate layout fragment_detail.xml menggunakan View Binding dan mengembalikan tampilan utama dari fragment.

Pada baris ke [23] terdapat fungsi onViewCreated yang berfungsi untuk mengatur tampilan fragment setelah layout selesai dibuat.

Pada baris ke [24] terdapat val affirmation = args.affirmation yang berfungsi untuk mengambil objek Affirmation dari argumen Safe Args yang dikirim melalui navigasi.

Pada baris ke [25] sampai [30] berfungsi untuk mengatur konten layout berdasarkan data dari objek Affirmation, seperti gambar, judul, tahun rilis, dan deskripsi dengan menggunakan getString() berdasarkan ID resource masing-masing.

Pada baris ke [33] sampai [36] terdapat fungsi onDestroyView yang berfungsi untuk menghapus referensi binding saat tampilan fragment dihancurkan, untuk mencegah memory leak dengan mengatur _binding = null.

MainActivity.kt:

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames yang berfungsi sebagai nama paket utama dari aplikasi ini dan menunjukkan bahwa file ini berada di bagian utama dari struktur folder aplikasi.

Pada baris ke [3] sampai [5], berfungsi untuk mengimpor beberapa kelas yang akan digunakan, yaitu Bundle dari Android untuk menyimpan data saat pembuatan activity, AppCompatActivity sebagai superclass activity, dan ActivityMainBinding untuk menghubungkan layout XML activity ke dalam kode melalui View Binding.

Pada baris ke [7] terdapat class MainActivity : AppCompatActivity() yang berfungsi untuk mendeklarasikan MainActivity sebagai activity utama aplikasi dan mewarisi perilaku dari AppCompatActivity.

Pada baris ke [8] terdapat private lateinit var binding: ActivityMainBinding yang berfungsi untuk mendeklarasikan variabel binding yang akan diinisialisasi nanti dan digunakan untuk mengakses komponen dalam layout activity melalui View Binding.

Pada baris ke [10] sampai [14] terdapat fungsi onCreate yang merupakan lifecycle method yang dijalankan saat activity pertama kali dibuat.

Pada baris ke [11] terdapat super.onCreate(savedInstanceState) yang berfungsi untuk memanggil implementasi onCreate dari superclass (AppCompatActivity).

Pada baris ke [12] terdapat binding = ActivityMainBinding.inflate(layoutInflater) yang berfungsi untuk menginisialisasi objek binding menggunakan layout inflater agar layout XML bisa digunakan dalam kode.

Pada baris ke [13] terdapat setContentView(binding.root) yang berfungsi untuk menampilkan layout utama activity ke layar dengan menggunakan root dari objek binding.

activity_main.xml:

Pada baris ke [1], terdapat deklarasi <?xml version="1.0" encoding="utf-8"?> yang berfungsi untuk menyatakan versi XML yang digunakan serta karakter encoding yang dipakai, yaitu UTF-8.

Pada baris ke [2], terdapat elemen FragmentContainerView yang berasal dari pustaka androidx.fragment.app. Elemen ini digunakan sebagai wadah untuk menampilkan fragmen.

Pada baris ke [3], terdapat deklarasi namespace xmlns:android dengan nilai http://schemas.android.com/apk/res/android, yang digunakan untuk mendefinisikan atribut-atribut khusus Android pada elemen-elemen XML.

Pada baris ke [4], terdapat deklarasi namespace tambahan xmlns:app dengan nilai http://schemas.android.com/apk/res-auto, yang memungkinkan penggunaan atribut kustom dari pustaka AndroidX.

Pada baris ke [5], terdapat atribut android:id="@+id/nav_host_fragment", yang menetapkan ID unik untuk elemen FragmentContainerView ini sebagai nav_host_fragment.

Pada baris ke [6], atribut android:name="androidx.navigation.fragment.NavHostFragment" mendefinisikan bahwa elemen ini adalah NavHostFragment, yang merupakan komponen utama untuk navigasi menggunakan Jetpack Navigation.

Pada baris ke [7], atribut android:layout_width="match_parent" digunakan untuk mengatur lebar elemen agar sesuai dengan lebar layar perangkat.

Pada baris ke [8], atribut android:layout_height="match_parent" digunakan untuk mengatur tinggi elemen agar sesuai dengan tinggi layar perangkat.

Pada baris ke [9], atribut app:navGraph="@navigation/nav_graph" menghubungkan elemen ini dengan file navigasi bernama nav_graph, yang berisi konfigurasi alur navigasi aplikasi.

Pada baris ke [10], atribut app:defaultNavHost="true" menetapkan elemen ini sebagai host navigasi utama, sehingga mampu menangani tindakan Back pada perangkat secara otomatis.

fragment_affirmation_list.xml

Pada baris ke [1], terdapat deklarasi <?xml version="1.0" encoding="utf-8"?> yang berfungsi untuk menyatakan versi XML yang digunakan serta karakter encoding yang dipakai, yaitu UTF-8.

Pada baris ke [2], terdapat elemen root ConstraintLayout dari pustaka androidx.constraintlayout.widget. Elemen ini digunakan sebagai layout utama yang memungkinkan penempatan elemen-elemen anak dengan aturan constraint.

Pada baris ke [3], terdapat deklarasi namespace xmlns:android dengan nilai http://schemas.android.com/apk/res/android, yang digunakan untuk mendefinisikan atribut-atribut khusus Android pada elemen-elemen XML.

Pada baris ke [4], terdapat deklarasi namespace tambahan xmlns:app dengan nilai http://schemas.android.com/apk/res-auto, yang memungkinkan penggunaan atribut kustom dari pustaka AndroidX.

Pada baris ke [5], atribut android:background="#000000" digunakan untuk mengatur latar belakang layout dengan warna hitam.

Pada baris ke [6], atribut android:layout_width="match_parent" digunakan untuk mengatur lebar layout agar sesuai dengan lebar layar perangkat.

Pada baris ke [7], atribut android:layout_height="match_parent" digunakan untuk mengatur tinggi layout agar sesuai dengan tinggi layar perangkat.

Pada baris ke [9], terdapat elemen RecyclerView dari pustaka androidx.recyclerview.widget. Elemen ini digunakan untuk menampilkan daftar data dengan performa tinggi.

Pada baris ke [10], atribut android:id="@+id/recyclerView" menetapkan ID unik untuk elemen ini sebagai recyclerView.

Pada baris ke [11], atribut android:contentDescription="@string/affirmation_list_desc" memberikan deskripsi konten yang bersifat aksesibilitas untuk RecyclerView, menggunakan nilai dari sumber string bernama affirmation_list_desc.

Pada baris ke [12] sampai [13], terdapat atribut android:layout_width="0dp" dan android:layout_height="0dp" mengatur lebar dan tinggi elemen ini agar dapat diatur sepenuhnya oleh constraint.

Pada baris ke [14], atribut android:padding="8dp" menambahkan jarak dalam sebesar 8dp di seluruh sisi elemen ini.

Pada baris ke [15], atribut app:layoutManager="LinearLayoutManager" menetapkan tata letak RecyclerView menggunakan LinearLayoutManager, yang menyusun elemen secara vertikal atau horizontal.

Pada baris ke [15] hingga [18], terdapat atribut constraint yang menghubungkan sisi atas, bawah, kiri, dan kanan RecyclerView dengan sisi parent, sehingga elemen ini berada di tengah layout secara proporsional.

fragment_detail.xml

Pada baris ke [1], terdapat deklarasi <?xml version="1.0" encoding="utf-8"?> yang berfungsi untuk menyatakan versi XML yang digunakan serta karakter encoding yang dipakai, yaitu UTF-8.

Pada baris ke [2], terdapat elemen root ScrollView yang berasal dari pustaka bawaan Android. Elemen ini digunakan sebagai wadah untuk memungkinkan konten di dalamnya dapat digulir secara vertikal. deklarasi namespace xmlns:android dengan nilai http://schemas.android.com/apk/res/android digunakan untuk mendefinisikan atribut-atribut khusus Android pada elemen-elemen XML.

Pada baris ke [3], deklarasi namespace tambahan xmlns:app dengan nilai http://schemas.android.com/apk/res-auto memungkinkan penggunaan atribut kustom dari pustaka AndroidX atau Material.

Pada baris ke [4], atribut android:background="#131418" mengatur latar belakang elemen dengan warna abu-abu gelap (#131418).

Pada baris ke [5] dan [6], atribut android:layout_width="match_parent" dan android:layout_height="match_parent" digunakan untuk mengatur elemen agar mengisi seluruh lebar dan tinggi layar perangkat.

Pada baris ke [8] sampai [9], terdapat elemen LinearLayout dengan orientasi vertikal yang berfungsi sebagai kontainer utama untuk menampung elemen-elemen lainnya.

Pada baris ke [10] sampai [12], atribut android:padding="16dp" menambahkan jarak sebesar 16dp di semua sisi elemen ini untuk memberikan ruang antara konten dan tepi layar.

Pada baris ke [14] hingga [19], terdapat elemen ShapeableImageView dari pustaka Material Components. Elemen ini menampilkan gambar dengan ID detailImageView, lebar penuh, tinggi 400dp, skala centerCrop, dan sudut membulat dengan gaya yang diatur oleh @style/ShapeAppearance.App.Rounded6dp.

Pada baris ke [21], terdapat LinearLayout horizontal yang digunakan untuk menempatkan teks secara berdampingan.

Pada baris ke [25], atribut android:gravity="center_vertical" digunakan untuk menyelaraskan konten di dalam elemen secara vertikal di tengah.

Pada baris ke [26], atribut android:layout_marginTop="12dp" menambahkan jarak sebesar 12dp antara elemen ini dan elemen di atasnya.

Pada baris ke [28] sampai [35], terdapat elemen TextView dengan ID detailTitleText, yang digunakan untuk menampilkan judul. Elemen ini menggunakan warna teks putih, ukuran teks 26sp, dan gaya teks tebal (bold).

Pada baris ke [37] hingga [42], terdapat elemen TextView dengan ID detailYearText, yang digunakan untuk menampilkan tahun. Elemen ini menggunakan warna teks abu-abu terang, ukuran teks 14sp, dan tinggi sesuai konten.

Pada baris ke [45] hingga [52], terdapat elemen TextView tanpa ID yang menampilkan teks “tentang game ini:” sebagai label. Elemen ini menggunakan warna teks putih, ukuran teks 14sp, dan gaya teks tebal.

Pada baris ke [54] hingga [60], terdapat elemen TextView dengan ID detailDescText, yang digunakan untuk menampilkan deskripsi detail. Elemen ini menggunakan warna teks putih, ukuran teks 14sp, dan margin atas sebesar 4dp untuk memberi jarak dari elemen di atasnya.

item_affirmation.xml

Pada baris ke [1], terdapat deklarasi <?xml version="1.0" encoding="utf-8"?> yang berfungsi untuk menyatakan versi XML yang digunakan serta karakter encoding yang dipakai, yaitu UTF-8.

Pada baris ke [2], terdapat elemen root CardView dari pustaka androidx.cardview.widget. Elemen ini digunakan untuk menampilkan konten dalam bentuk kartu dengan sudut membulat dan latar belakang yang dapat disesuaikan.

Pada baris ke [3], deklarasi namespace xmlns:android dengan nilai http://schemas.android.com/apk/res/android digunakan untuk mendefinisikan atribut-atribut khusus Android pada elemen-elemen XML.

Pada baris ke [4], deklarasi namespace tambahan xmlns:app dengan nilai http://schemas.android.com/apk/res-auto memungkinkan penggunaan atribut kustom dari pustaka AndroidX.

Pada baris ke [5] hingga [6], atribut android:layout_width="match_parent" dan android:layout_height="wrap_content" digunakan untuk mengatur elemen agar memenuhi lebar layar perangkat dengan tinggi yang menyesuaikan kontennya.

Pada baris ke [7], atribut android:layout_margin="4dp" menambahkan jarak luar sebesar 4dp di seluruh sisi elemen ini.

Pada baris ke [8], atribut app:cardCornerRadius="6dp" digunakan untuk mengatur radius sudut kartu menjadi 6dp agar tampak membulat.

Pada baris ke [9], atribut app:cardBackgroundColor="#343539" mengatur warna latar belakang kartu dengan warna abu-abu gelap (#343539).

Pada baris ke [11], terdapat elemen LinearLayout dengan orientasi horizontal yang digunakan untuk menata elemen-elemen anak secara berdampingan.

Pada baris ke [15], atribut android:padding="8dp" menambahkan jarak dalam sebesar 8dp di seluruh sisi elemen ini.

Pada baris ke [18] hingga [25], terdapat elemen ShapeableImageView dari pustaka Material Components, yang menampilkan gambar dengan ID imageView. Elemen ini memiliki lebar 100dp, tinggi 140dp, margin kanan 12dp, skala gambar centerCrop, dan sudut membulat sesuai dengan gaya @style/ShapeAppearance.App.Rounded6dp.

Pada baris ke [27] sampai [37], terdapat LinearLayout vertikal yang digunakan untuk menata elemen-elemen teks dan tombol secara bertumpuk di dalamnya.

Pada baris ke [29], atribut android:layout_weight="1" digunakan untuk membuat elemen ini mengisi ruang yang tersisa dalam LinearLayout horizontal induknya.

Pada baris ke [39] hingga [53], terdapat elemen TextView untuk menampilkan judul dan tahun. Teks judul menggunakan ID titleText, warna putih, ukuran teks 20sp, dan gaya tebal, sedangkan teks tahun menggunakan ID yearText, warna abu-abu terang, dan ukuran teks 14sp.

Pada baris ke [56] hingga [63], terdapat elemen TextView tanpa ID yang menampilkan teks "tentang game ini:" sebagai label dengan warna putih, ukuran teks 14sp, dan gaya tebal.

Pada baris ke [65] hingga [71], terdapat elemen TextView dengan ID detailText yang digunakan untuk menampilkan deskripsi tambahan dengan warna putih, ukuran teks 14sp, dan margin atas 4dp.

Pada baris ke [73] hingga [78], terdapat LinearLayout horizontal yang digunakan untuk menempatkan dua tombol di sisi kanan bawah kartu.

Pada baris ke [80] hingga [86], terdapat tombol pertama dengan ID viewButton yang memiliki teks "View", lebar 100dp, tinggi 44dp, warna latar biru terang (#b0c4ff), dan warna teks hitam.

Pada baris ke [88], terdapat elemen Space dengan lebar 8dp yang digunakan untuk memberi jarak antara kedua tombol.

Pada baris ke [90] hingga [96], terdapat tombol kedua dengan ID detailButton yang memiliki teks "Detail", lebar dan tinggi yang sama seperti tombol pertama, warna latar biru terang, dan warna teks hitam.

strings.xml

Pada baris ke [1], terdapat elemen root <resources> yang digunakan untuk menampung semua sumber daya string dalam aplikasi Android.

Pada baris ke [2], terdapat elemen <string> dengan atribut name="app_name" yang menyimpan nama aplikasi, yaitu "My favorite games".

Pada baris ke [4], terdapat elemen <string> dengan atribut name="affirmation_list_desc" yang berfungsi sebagai deskripsi aksesibilitas untuk daftar permainan favorit, yaitu "List of favorite games".

Pada baris ke [6] sampai [15], terdapat sepuluh elemen <string> dengan atribut name="titleX" yang masing-masing menyimpan judul permainan favorit berdasarkan nomor urut, seperti "Terraria", "Mobile Legends", dan lainnya.

Pada baris ke [17] sampai [26], terdapat sepuluh elemen <string> dengan atribut name="yearX" yang masing-masing menyimpan tahun rilis permainan berdasarkan nomor urut, seperti "2011" untuk year1 dan "2020" untuk year6.

Pada baris ke [28] sampai [37], terdapat sepuluh elemen <string> dengan atribut name="detailX" yang masing-masing menyimpan deskripsi detail permainan berdasarkan nomor urut. Contohnya, detail1 berisi deskripsi "Terraria", dan detail6 berisi deskripsi "Genshin Impact".

Pada baris ke [39] sampai [48], terdapat sepuluh elemen <string> dengan atribut name="linkX" yang masing-masing menyimpan URL terkait permainan berdasarkan nomor urut. Contohnya, link1 berisi URL ke halaman Steam "Terraria", dan link6 berisi URL ke halaman Google Play "Genshin Impact".

styles.xml

Pada baris ke [1], terdapat deklarasi <?xml version="1.0" encoding="utf-8"?> yang berfungsi untuk menyatakan versi XML yang digunakan serta karakter encoding yang dipakai, yaitu UTF-8.

Pada baris ke [2], terdapat elemen root <resources> yang digunakan untuk menampung semua sumber daya gaya (style) dalam aplikasi Android.

Pada baris ke [3], terdapat elemen <style> dengan atribut name="ShapeAppearance.App.Rounded6dp" yang mendefinisikan gaya bernama ShapeAppearance.App.Rounded6dp. Gaya ini digunakan untuk mengatur tampilan sudut elemen tertentu menjadi membulat dengan radius 6dp. atribut parent="" tidak memiliki nilai tertentu, menunjukkan bahwa gaya ini tidak mewarisi properti dari gaya lain.

Pada baris ke [4], terdapat elemen <item> dengan atribut name="cornerFamily" dan nilai rounded, yang menentukan bahwa elemen-elemen yang menggunakan gaya ini akan memiliki sudut membulat.

Pada baris ke [5], terdapat elemen <item> dengan atribut name="cornerSize" dan nilai 6dp, yang menentukan ukuran radius sudut membulat sebesar 6dp.

Pada baris ke [6], elemen </style> menutup definisi gaya ShapeAppearance.App.Rounded6dp.

Pada baris ke [7], elemen </resources> menutup elemen root <resources>.

COMPOSE :

Datasource.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames.data yang berfungsi sebagai deklarasi nama paket tempat file ini disimpan dalam struktur proyek aplikasi Android.

Pada baris ke [2] berfungsi untuk mengimpor file R, yaitu file auto-generated yang berisi referensi ke semua sumber daya (seperti string, drawable, dan layout) dalam aplikasi.

Pada baris ke [3] digunakan untuk mengimpor kelas Affirmation dari package model, yaitu model data yang digunakan untuk merepresentasikan setiap item dalam daftar favorit.

Pada baris ke [5] terdapat deklarasi class Datasource yang berfungsi sebagai sumber data untuk aplikasi ini.

Pada baris ke [6] sampai [7] terdapat fungsi loadAffirmations() yang akan mengembalikan sebuah List<Affirmation>, yaitu daftar data yang berisi 10 objek Affirmation.

Pada baris ke [8] sampai [17] setiap item Affirmation diisi menggunakan referensi dari resource R.string, R.drawable, dan R.string lainnya. Setiap objek menyimpan detail teks, gambar, link, tahun rilis, dan judul dari game favorit.

Affirmation.kt:

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames.model yang berfungsi sebagai deklarasi nama paket tempat file ini disimpan, dan menunjukkan bahwa file ini berada di dalam struktur folder model.

Pada baris ke [2] sampai [3], dilakukan impor terhadap anotasi @DrawableRes dan @StringRes dari AndroidX. Anotasi ini digunakan untuk memberikan informasi bahwa parameter yang digunakan harus berupa ID dari resource drawable dan string, sehingga membantu Android Studio melakukan validasi dan memberikan peringatan bila terjadi kesalahan tipe resource.

Pada baris ke [5] terdapat deklarasi data class Affirmation yang merupakan sebuah kelas data yang berfungsi untuk menyimpan informasi dari satu item game favorit. Dengan menggunakan data class, Kotlin secara otomatis akan menyediakan fungsi toString, equals, dan hashCode yang mempermudah pengelolaan data dalam list atau tampilan antarmuka.

Pada baris ke [6] sampai [10] terdapat lima properti yang dideklarasikan di dalam konstruktor Affirmation, yaitu detailResourceId, imageResourceId, steamLinkResourceId, yearResourceId, dan titleResourceId. Semuanya bertipe Int karena merujuk pada resource ID yang didefinisikan dalam file res/values/strings.xml dan res/drawable. Properti detailResourceId, steamLinkResourceId, yearResourceId, dan titleResourceId masing-masing diberi anotasi @StringRes untuk menandakan bahwa nilai tersebut adalah ID dari

string resource. Sedangkan imageResourceId menggunakan anotasi @DrawableRes karena nilainya berupa ID dari gambar (drawable).

AffirmationCard.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames yang berfungsi sebagai nama paket dari aplikasi dan tempat menyimpan file.

Pada baris ke [3] sampai [22], dilakukan impor berbagai komponen penting dari Android Jetpack Compose seperti layout, gambar, teks, warna, bentuk sudut, serta intent untuk membuka halaman baru. Impor ini juga mencakup LocalContext dan stringResource untuk mengambil data dari resource string dan drawable berdasarkan ID.

Pada baris ke [25] sampai [29], terdapat fungsi composable bernama AffirmationCard yang menerima tiga parameter, yaitu objek affirmation dari tipe data Affirmation, modifier yang bersifat opsional dengan nilai default Modifier, dan onDetailClick sebagai fungsi lambda yang dijalankan saat tombol Detail ditekan.

Pada baris ke [30] terdapat deklarasi val context = LocalContext.current yang digunakan untuk mengambil context saat ini, yang diperlukan ketika akan membuat intent untuk membuka halaman link.

Pada baris ke [32] sampai [123] digunakan untuk membentuk struktur tampilan kartu menggunakan komponen Card dari Material3 dengan padding 12.dp dan warna latar belakang abu-abu gelap.

Pada baris ke [36] digunakan Row agar isi kartu tersusun secara horizontal.

Pada baris ke [38] sampai [48] digunakan Image untuk menampilkan gambar dari game, dengan pengambilan resource dari affirmation.imageResourceId. Gambar diberi padding kiri, ukuran lebar 100.dp dan tinggi 140.dp, serta sudut yang dibulatkan menggunakan clip(RoundedCornerShape(6.dp)). Gambar juga disejajarkan secara vertikal di tengah dan menggunakan ContentScale.Crop agar gambar menyesuaikan ukuran kotaknya.

Pada baris ke [50] sampai [121] digunakan Column sebagai kolom utama di sebelah kanan gambar yang berisi judul, tahun rilis, deskripsi singkat, dan dua tombol. Kolom diberi padding 16.dp dan diatur agar memenuhi lebar yang tersedia.

Pada baris ke [56] sampai [76] digunakan Row di bagian atas kolom untuk menampilkan judul game dan tahun rilis secara berdampingan. Judul menggunakan fontSize 26.sp dan FontWeight.Bold, sedangkan tahun rilis menggunakan warna teks abu-abu muda dan ukuran 16.sp.

Pada baris ke [80] hingga [90] digunakan Column yang menampilkan label teks "tentang game ini:" dan dilanjutkan dengan teks deskripsi dari resource affirmation.detailResourceId. Seluruh teks menggunakan warna putih dengan ukuran font 14.sp.

Pada baris ke [94] hingga [120] digunakan Row untuk menyusun dua buah tombol secara horizontal di bagian kanan bawah kartu. Tombol pertama diberi label “View” dan ketika ditekan akan membuka link Steam dari affirmation.steamLinkResourceId menggunakan Intent.ACTION_VIEW dan method startActivity. Tombol kedua diberi label “Detail” dan ketika ditekan akan menjalankan lambda onDetailClick untuk menampilkan informasi lebih lengkap. Kedua tombol memiliki ukuran lebar 100.dp dan tinggi 44.dp, serta menggunakan warna latar biru muda dengan teks berwarna hitam.

DetailScreen.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames yang berfungsi sebagai penamaan paket dari aplikasi dan sebagai tempat penyimpanan file DetailScreen.kt.

Pada baris ke [3] sampai [18], berfungsi untuk mengimpor berbagai komponen penting dari Jetpack Compose seperti Image, Column, Row, Text, Surface, Modifier, Color, dan lainnya yang digunakan untuk membuat tampilan layar detail game.

Pada baris ke [20] terdapat anotasi @Composable yang menunjukkan bahwa fungsi DetailScreen merupakan fungsi composable, yaitu fungsi yang digunakan untuk membangun UI di Jetpack Compose.

Pada baris ke [21] terdapat deklarasi fungsi DetailScreen dengan parameter titleResId, imageResId, yearResId, dan detailResId bertipe Int, yang digunakan untuk menampilkan detail game berdasarkan resource ID yang diberikan.

Pada baris ke [22] sampai [25] digunakan Surface sebagai wadah utama tampilan yang memenuhi ukuran layar penuh dan memiliki warna latar belakang Color(0xFF131418).

Pada baris ke [26] terdapat Column yang digunakan untuk menyusun komponen secara vertikal, dengan jarak padding sebesar 16dp di sekeliling konten.

Pada baris ke [28] sampai [36] digunakan Image untuk menampilkan gambar game berdasarkan imageResId. Gambar memenuhi lebar penuh (fillMaxWidth), tinggi 400dp, memiliki sudut membulat, dan menggunakan ContentScale.Crop agar gambar menyesuaikan ukuran.

Pada baris ke [38] sampai [53] digunakan Column dan Row untuk menampilkan judul dan tahun rilis game. Judul menggunakan ukuran font 26sp dan tebal dengan warna putih, sedangkan tahun menggunakan ukuran font 16sp dengan warna abu-abu keunguan.

Pada baris ke [55] berfungsi untuk menjadi Spacer untuk memberi jarak vertikal 8dp antara bagian judul dan deskripsi.

Pada baris ke [57] sampai [66] digunakan Text untuk menampilkan label "tentang game ini:" dan isi detail deskripsi dari game. Label ditampilkan dalam font 14sp tebal berwarna putih, sedangkan isi detail ditampilkan dalam font 14sp biasa juga berwarna putih, dengan padding atas sebesar 4dp.

HomeScreen.kt

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames yang berfungsi untuk menentukan lokasi file ini dalam struktur proyek.

Pada baris ke [3] sampai [6] digunakan import untuk memanggil fungsi dan komponen penting dari Jetpack Compose, yaitu LazyColumn, items, dan Composable, yang digunakan untuk membuat tampilan daftar secara efisien dan dapat diskrol.

Pada baris ke [8] sampai [9] diimpor Datasource dan Affirmation dari package data dan model, yang masing-masing menyediakan data dummy dan struktur data untuk ditampilkan di layar.

Pada baris ke [11] terdapat anotasi @Composable yang menunjukkan bahwa HomeScreen adalah fungsi composable yang digunakan untuk membuat UI layar utama dari aplikasi.

Pada baris ke [12] sampai [15] terdapat deklarasi fungsi HomeScreen dengan parameter modifier (opsional untuk pengaturan tampilan) dan onDetailClick, yaitu lambda function yang menangani aksi ketika pengguna mengklik salah satu item untuk melihat detailnya.

Pada baris ke [16] dipanggil fungsi loadAffirmations() dari Datasource() untuk memuat daftar data Affirmation.

Pada baris ke [18] sampai [22] digunakan LazyColumn untuk menampilkan daftar secara vertikal dan efisien. Setiap item dalam daftar affirmations akan ditampilkan dengan memanggil fungsi AffirmationCard, dan saat diklik, akan memanggil lambda onDetailClick dengan data item tersebut.

MainActivity.kt:

Pada baris ke [1] terdapat package com.pemrogamanmobile.myfavoritegames yang berfungsi sebagai nama paket dari aplikasi dan tempat menyimpan file.

Pada baris ke [3] sampai [13], berfungsi untuk mengimpor beberapa komponen penting dari Android seperti Bundle, ComponentActivity, setContent, Surface, Color, dan komponen-komponen navigasi dari Jetpack Compose yang digunakan dalam aplikasi ini.

Pada baris ke [15] terdapat class MainActivity : ComponentActivity() yang berfungsi untuk membuat class MainActivity sebagai activity utama aplikasi ini.

Pada baris ke [16] sampai [19] berfungsi untuk membuat fungsi onCreate yang akan dijalankan saat activity dimulai, lalu menampilkan tampilan aplikasi menggunakan tema MyFavoriteGamesTheme.

Pada baris ke [20] sampai [54] digunakan untuk menampilkan Surface sebagai latar belakang aplikasi dengan warna Color(0xFF131418).

Pada baris ke [21] terdapat deklarasi navController menggunakan rememberNavController() yang berfungsi untuk mengatur navigasi antar layar dalam aplikasi.

Pada baris ke [23] sampai [53] digunakan untuk membuat NavHost dengan startDestination bernama "home", yang berisi dua composable: "home" dan "detail".

Pada baris ke [25] sampai [29] digunakan untuk menampilkan HomeScreen pada rute "home", dengan parameter onDetailClick yang menerima objek affirmation lalu menavigasi

ke layar detail sambil membawa beberapa nilai resource ID seperti title, image, year, dan detail.

Pada baris ke [32] sampai [40] digunakan untuk menampilkan composable "detail" dengan parameter yang dikirim dari navigasi berupa titleResId, imageResId, yearResId, dan detailResId, yang semuanya bertipe Int.

Pada baris ke [40] sampai [51] digunakan untuk mengambil nilai-nilai argumen dari backStackEntry dan menampilkannya pada DetailScreen.

strings.xml

Pada baris ke [1], terdapat elemen root <resources> yang digunakan untuk menampung semua sumber daya string dalam aplikasi Android.

Pada baris ke [2], terdapat elemen <string> dengan atribut name="app_name" yang menyimpan nama aplikasi, yaitu "My favorite games".

Pada baris ke [4] sampai [13], terdapat sepuluh elemen <string> dengan atribut name="titleX" yang masing-masing menyimpan judul permainan favorit berdasarkan nomor urut, seperti "Terraria", "Mobile Legends", dan lainnya.

Pada baris ke [15] sampai [24], terdapat sepuluh elemen <string> dengan atribut name="yearX" yang masing-masing menyimpan tahun rilis permainan berdasarkan nomor urut, seperti "2011" untuk year1 dan "2020" untuk year6.

Pada baris ke [26] sampai [35], terdapat sepuluh elemen <string> dengan atribut name="detailX" yang masing-masing menyimpan deskripsi detail permainan berdasarkan nomor urut. Contohnya, detail1 berisi deskripsi "Terraria", dan detail6 berisi deskripsi "Genshin Impact".

Pada baris ke [37] sampai [46], terdapat sepuluh elemen <string> dengan atribut name="linkX" yang masing-masing menyimpan URL terkait permainan berdasarkan nomor urut. Contohnya, link1 berisi URL ke halaman Steam "Terraria", dan link6 berisi URL ke halaman Google Play "Genshin Impact".

SOAL 2

Soal Praktikum:

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

A. Pembahasan

RecyclerView masih umum digunakan padahal memiliki kode yang panjang dan bersifat boiler-plate dibandingkan LazyColumn dengan kode yang lebis singkat dikarenakan beberapa hal berikut :

1. Kompabilitas dan dukungannya yang cukup luas
   RecyclerView masih ada di dalam Pustaka android.rcyclerview dan didukung secara luas dalam berbagai versi aplikasi android. Selain itu, aplikasi yang dibuat sebelum mengenal jetpack compose banyak menggunakan Recylerview sehingga contohnya banyak tersedia.
2. Fleksibel dan memiliki kendali penuh
   RecyclerView memberikan pengguna kemampuan unuk mengontrol tampilannya,secara fleksibel contohnya sendiri seperti mengatur tata letak, animasinya, atau prilaku scroll.
3. Dukungan yang sangat stabil
   RecyclerView merupakan komponen yang stabil digunakan karena telah melewati banyak ujicoba,sehingga banyak digunakan dalam aplikasi produksi. Selain itu RecylcerView bisa dimasukkan Solusi atau liblary pihak ketiga, dan dapat teritegrasi dengan baik.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/Harry154git/Pemrogaman_mobile/tree/main/Modul_3