

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



BUILD A SCROLLABLE LIST

Oleh:

Galih Aji Sabdaraya

NIM. 2310817210005

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2025**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
MODUL 3

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

Nama Praktikan : Galih Aji Sabdaraya
NIM : 2310817210005

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar
NIM. 2210817210026

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

DAFTAR ISI

LEMBAR PENGESAHAN.....	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code.....	8
B. Output Program	29
C. Pembahasan	33
SOAL 2.....	43
A. Pembahasan	43
TAUTAN GIT	44

DAFTAR GAMBAR

Gambar 1. Contoh UI List	7
Gambar 2. Contoh UI Detail	7
Gambar 3. Screenshot Tampilan Halaman Home Potrait Aplikasi	29
Gambar 4. Screenshot Tampilan Halaman Detail Potrait Aplikasi	30
Gambar 5. Screenshot Tampilan Halaman Home Landscape Aplikasi.....	31
Gambar 6. Screenshot Tampilan Halaman Detail Landscape Aplikasi.....	31
Gambar 7. Screenshot Membuka Browser F1 Team Setelah Klik “F1 Team”	32

DAFTAR TABEL

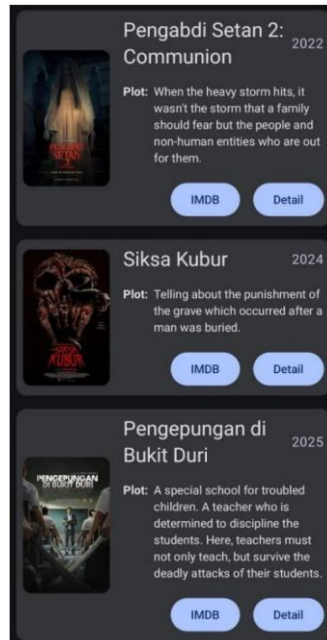
Tabel 1. Source Code MainActivity.kt.....	8
Tabel 2. Source Code F1Team1.kt.....	8
Tabel 3. Source Code HomeFragment.kt	9
Tabel 4. Source Code DetailFragment.kt	11
Tabel 5. Source Code F1TeamViewModel.kt.....	13
Tabel 6. Source Code F1TeamAdapter.kt	15
Tabel 7. Source Code activity_main.xml	17
Tabel 8. Source Code fragment_home.xml	17
Tabel 9. Source Code item_f1.xml	18
Tabel 10. Source Code fragment_detail.xml	21
Tabel 11. Source Code strings.xml.....	24

SOAL 1

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

1. List menggunakan fungsi RecyclerView (XML) atau LazyColumn (Compose)
2. List paling sedikit menampilkan 5 item. Tema item yang ingin ditampilkan bebas
3. Item pada list menampilkan teks dan gambar sesuai dengan contoh di bawah
4. 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/intent untuk membuka laman detail item
5. Sudut item pada list dan gambar di dalam list melengkung atau rounded corner menggunakan Radius
6. Saat orientasi perangkat berubah/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 Detail

A. Source Code

1. MainActivity.kt

Tabel 1. Source Code MainActivity.kt

```
1 package com.example.flteam
2
3 import android.os.Bundle
4 import androidx.appcompat.app.AppCompatActivity
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10    }
11 }
```

2. F1Team.kt

Tabel 2. Source Code F1Team.kt

```
1 package com.example.flteam.model
2
3 import android.os.Parcelable
4 import kotlinx.parcelize.Parcelize
5
6 @Parcelize
7 data class F1Team (
8     val name: String,
9     val base: String,
10    val teamChief: String,
11    val chassis: String,
12    val worldChampion : String,
13    val desc: String,
14    val link: String,
15    val imageUrl: String,
16    val imageUrlDetail: String
17
18 ): Parcelable
```


3. HomeFragment.kt

Tabel 3. Source Code HomeFragment.kt

```
1 package com.example.flteam.ui.fragment
2
3 import android.content.Intent
4 import android.net.Uri
5 import android.os.Bundle
6 import androidx.fragment.app.Fragment
7 import android.view.LayoutInflater
8 import android.view.View
9 import android.view.ViewGroup
10 import androidx.lifecycle.ViewModelProvider
11 import
12     com.example.flteam.databinding.FragmentHomeBinding
13 import com.example.flteam.ui.adapter.FlTeamAdapter
14 import com.example.flteam.viewmodel.FlTeamViewModel
15 import androidx.navigation.fragment.findNavController
16 import androidx.recyclerview.widget.LinearLayoutManager
17
18 class HomeFragment : Fragment() {
19
20     private var _binding: FragmentHomeBinding? = null
21     private val binding get() = _binding!!
22
23     private lateinit var viewModel: FlTeamViewModel
24     private lateinit var adapter: FlTeamAdapter
25
26     override fun onCreateView(
27         inflater: LayoutInflater, container: ViewGroup?,
28         savedInstanceState: Bundle?
29     ): View? {
30         _binding = FragmentHomeBinding.inflate(inflater,
31         container, false)
32         return binding.root
33     }
34
35     override fun onViewCreated(view: View,
36         savedInstanceState: Bundle?) {
37         viewModel =
38         ViewModelProvider(this)[FlTeamViewModel::class.java]
39
40         adapter = FlTeamAdapter(
41             onTeamClick = { team ->
42                 val intent = Intent(Intent.ACTION_VIEW,
43                 Uri.parse(team.link))
```

```

39         startActivity(intent)
40     },
41     onDetailClick = {team ->
42         val action =
HomeFragmentDirections.actionHomeFragmentToDetailFragmen
t(team)
43         findNavController().navigate(action)
44     }
45 )
46
47     binding.rvItem.layoutManager =
LinearLayoutManager(requireContext())
48     binding.rvItem.adapter = adapter
49
50     viewModel.teamList.observe(viewLifecycleOwner)
{team ->
51         adapter.submitList(team)
52     }
53 }
54
55     override fun onDestroyView() {
56         super.onDestroyView()
57         _binding = null
58     }
59 }

```

4. DetailFragment.kt

Tabel 4. Source Code DetailFragment.kt

```
1 package com.example.flteam.ui.fragment
2
3 import android.os.Bundle
4 import androidx.fragment.app.Fragment
5 import android.view.LayoutInflater
6 import android.view.View
7 import android.view.ViewGroup
8 import com.bumptech.glide.Glide
9 import
10 com.example.flteam.databinding.FragmentDetailBinding
11 import com.example.flteam.model.FlTeam
12
13 class DetailFragment : Fragment() {
14
15     private var _binding: FragmentDetailBinding? = null
16     private val binding get() = _binding!!
17
18     private var team: FlTeam? = null
19
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22         arguments?.let {
23             team =
24             DetailFragmentArgs.fromBundle(it).team
25         }
26
27         override fun onCreateView(inflater: LayoutInflater,
28             container: ViewGroup?, savedInstanceState: Bundle?):
29             View? {
30             _binding =
31             FragmentDetailBinding.inflate(inflater, container,
32             false)
33             return binding.root
34         }
35
36         override fun onViewCreated(view: View,
37             savedInstanceState: Bundle?) {
38             team?.let { team ->
39                 binding.itemNameDetail.text = team.name
40                 binding.itemBaseDetail.text = team.base
41                 binding.itemChiefDetail.text =
```

```
team.teamChief
37         binding.itemChassisDetail.text =
team.chassis
38         binding.itemWcDetail.text =
team.worldChampion
39         binding.itemDescDetail.text = team.desc
40
41         Glide.with(requireContext())
42             .load(team.imageUrlDetail)
43             .into(binding.itemImageDetail)
44     }
45 }
46
47 override fun onDestroyView() {
48     super.onDestroyView()
49     _binding = null
50 }
51 }
```

5. F1TeamViewModel.kt

Tabel 5. Source Code F1TeamViewModel.kt

```
1 package com.example.flteam.viewmodel
2
3 import android.app.Application
4 import androidx.lifecycle.AndroidViewModel
5 import androidx.lifecycle.LiveData
6 import androidx.lifecycle.MutableLiveData
7 import com.example.flteam.R
8 import com.example.flteam.model.F1Team
9
10 class F1TeamViewModel(application: Application) :
    AndroidViewModel(application) {
11     private val _teamList =
        MutableLiveData<List<F1Team>>()
12     val teamList: LiveData<List<F1Team>> get() =
        _teamList
13
14     init {
15         val name =
            application.resources.getStringArray(R.array.data_name).
                toList()
16         val base =
            application.resources.getStringArray(R.array.data_base).
                toList()
17         val teamChief =
            application.resources.getStringArray(R.array.data_chief)
                .toList()
18         val chassis =
            application.resources.getStringArray(R.array.data_chassi
                s).toList()
19         val champion =
            application.resources.getStringArray(R.array.data_champi
                on).toList()
20         val desc =
            application.resources.getStringArray(R.array.data_desc).
                toList()
21         val link =
            application.resources.getStringArray(R.array.data_link).
                toList()
22         val image =
            application.resources.getStringArray(R.array.data_image)
                .toList()
23         val imageDetail =
            application.resources.getStringArray(R.array.data_image_
```

```

24 detail).toList()
25     val f1Team = mutableListOf<F1Team>()
26
27     for(i in name.indices) {
28         f1Team.add(
29             F1Team(
30                 name = name[i],
31                 base = base[i],
32                 teamChief = teamChief[i],
33                 chassis = chassis[i],
34                 worldChampion = champion[i],
35                 desc = desc[i],
36                 link = link[i],
37                 imageUrl = image[i],
38                 imageUrlDetail = imageDetail[i]
39             )
40         )
41     }
42
43     _teamList.value = f1Team
44 }
45 }

```

6. F1TeamAdapter.kt

Tabel 6. Source Code F1TeamAdapter.kt

```
1 package com.example.flteam.ui.adapter
2
3 import android.view.LayoutInflater
4 import android.view.ViewGroup
5 import androidx.recyclerview.widget.RecyclerView
6 import com.bumptech.glide.Glide
7 import
8     com.bumptech.glide.load.resource.bitmap.CenterCrop
9     com.bumptech.glide.load.resource.bitmap.RoundedCorners
10 import com.example.flteam.databinding.ItemF1Binding
11 import com.example.flteam.model.F1Team
12
13 class F1TeamAdapter (
14     private val onTeamClick: (F1Team) -> Unit,
15     private val onDetailClick: (F1Team) -> Unit
16 ) :
17     RecyclerView.Adapter<F1TeamAdapter.F1TeamViewHolder>() {
18     private val teamList = mutableListOf<F1Team>()
19
20     fun submitList(list : List<F1Team>){
21         teamList.clear()
22         teamList.addAll(list)
23         notifyDataSetChanged()
24     }
25
26     override fun onCreateViewHolder(parent: ViewGroup,
27         viewType: Int): F1TeamAdapter.F1TeamViewHolder {
28         val binding =
29             ItemF1Binding.inflate(LayoutInflater.from(parent.context),
30                 parent, false)
31         return F1TeamViewHolder(binding)
32     }
33
34     override fun onBindViewHolder(holder:
35         F1TeamAdapter.F1TeamViewHolder, position: Int) {
36         holder.bind(teamList[position])
37     }
38
39     override fun getItemCount(): Int = teamList.count()
```

```

36
37     inner class F1TeamViewHolder(private val binding:
ItemF1Binding) : RecyclerView.ViewHolder(binding.root) {
38         fun bind(team: F1Team) {
39             binding.itemName.text = team.name
40             binding.itemDesc.text = team.desc
41             Glide.with(binding.root.context)
42                 .load(team.imageUrl)
43                 .transform(CenterCrop(),
RoundedCorners(24))
44                 .into(binding.itemImageDetail)
45
46             binding.btnTeam.setOnClickListener
{onTeamClick(team)}
47             binding.btnDetail.setOnClickListener
{onDetailClick(team)}
48         }
49     }
50
51 }

```


7. activity_main.xml

Tabel 7. Source Code activity_main.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.fragment.app.FragmentContainerView
3	xmlns:android="http://schemas.android.com/apk/res/andro id"
4	xmlns:app="http://schemas.android.com/apk/res-auto"
5	xmlns:tools="http://schemas.android.com/tools"
6	android:id="@+id/nav_host_fragment"
7	android:name="androidx.navigation.fragment.NavHostFragm ent"
8	android:layout_width="match_parent"
9	android:layout_height="match_parent"
10	app:defaultNavHost="true"
11	app:navGraph="@navigation/nav_graph"
12	tools:context=".MainActivity" />

8. fragment_home.xml

Tabel 8. Source Code fragment_home.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
	xmlns:android="http://schemas.android.com/apk/res/andro id"
3	xmlns:app="http://schemas.android.com/apk/res-auto"
4	xmlns:tools="http://schemas.android.com/tools"
5	android:layout_width="match_parent"
6	android:layout_height="match_parent"
7	tools:context=".ui.fragment.HomeFragment">
8	
9	<androidx.recyclerview.widget.RecyclerView
10	android:id="@+id/rv_item"
11	android:layout_width="match_parent"
12	android:layout_height="match_parent"
13	android:layout_margin="15dp"
14	app:layout_constraintBottom_toBottomOf="parent"
15	app:layout_constraintEnd_toEndOf="parent"
16	app:layout_constraintHorizontal_bias="0.482"
17	app:layout_constraintStart_toStartOf="parent"
18	app:layout_constraintTop_toTopOf="parent"

19	app:layout_constraintVertical_bias="0.551" />
20	
21	
22	</androidx.constraintlayout.widget.ConstraintLayout>

9. item_f1.xml

Tabel 9. Source Code item_f1.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.cardview.widget.CardView
	xmlns:android="http://schemas.android.com/apk/res/andro
	id"
3	xmlns:tools="http://schemas.android.com/tools"
4	android:layout_width="match_parent"
5	android:layout_height="wrap_content"
6	xmlns:card_view="http://schemas.android.com/apk/res-
	auto"
7	android:layout_gravity="center"
8	android:layout_marginStart="8dp"
9	android:layout_marginTop="4dp"
10	android:layout_marginEnd="8dp"
11	android:layout_marginBottom="4dp"
12	card_view:cardCornerRadius="4dp">
13	
14	<androidx.constraintlayout.widget.ConstraintLayout
15	android:layout_width="match_parent"
16	android:layout_height="wrap_content"
17	android:padding="8dp">
18	
19	<ImageView
20	android:id="@+id/item_image_detail"
21	android:layout_width="110dp"
22	android:layout_height="130dp"
23	android:scaleType="centerCrop"
24	android:layout_marginEnd="8dp"
25	card_view:layout_constraintStart_toStartOf="parent"
26	card_view:layout_constraintTop_toTopOf="parent"
27	card_view:layout_constraintBottom_toBottomOf="parent"/>
28	

```

29         <TextView
30             android:id="@+id/item_name"
31             android:layout_width="0dp"
32             android:layout_height="wrap_content"
33             android:layout_marginStart="8dp"
34             android:layout_marginTop="8dp"
35             android:layout_marginEnd="8dp"
36             android:gravity="start"
37             android:textColor="#000000"
38             android:textSize="16sp"
39             android:textStyle="bold"

40     card_view:layout_constraintEnd_toEndOf="parent"

41     card_view:layout_constraintHorizontal_bias="0.888"

42     card_view:layout_constraintStart_toEndOf="@id/item_image_detail"

43     card_view:layout_constraintTop_toTopOf="parent" />
44
45         <TextView
46             android:id="@+id/item_desc"
47             android:layout_width="0dp"
48             android:layout_height="wrap_content"
49             android:layout_marginStart="8dp"
50             android:layout_marginTop="4dp"
51             android:layout_marginEnd="8dp"
52             android:justificationMode="inter_word"
53             android:ellipsize="end"
54             android:maxLines="4"

55     card_view:layout_constraintBottom_toTopOf="@id/btn_team"

56     card_view:layout_constraintEnd_toEndOf="parent"

57     card_view:layout_constraintStart_toEndOf="@id/item_image_detail"

58     card_view:layout_constraintTop_toBottomOf="@id/item_name" />
59
60         <Button
61             android:id="@+id/btn_detail"
62             android:layout_width="wrap_content"
63             android:layout_height="wrap_content"

```

64	android:layout_marginTop="8dp"
65	android:text="@string/btn_detail"
66	card_view:layout_constraintBottom_toBottomOf="parent"
67	card_view:layout_constraintEnd_toEndOf="parent"
68	card_view:layout_constraintTop_toBottomOf="@id/item_desc" />
69	
70	<Button
71	android:id="@+id/btn_team"
72	android:layout_width="wrap_content"
73	android:layout_height="wrap_content"
74	android:layout_marginTop="8dp"
75	android:layout_marginEnd="8dp"
76	android:text="@string/btn_team"
77	card_view:layout_constraintBottom_toBottomOf="parent"
78	card_view:layout_constraintEnd_toStartOf="@id/btn_detail"
79	card_view:layout_constraintTop_toBottomOf="@id/item_desc" />
80	
81	</androidx.constraintlayout.widget.ConstraintLayout>
82	
83	</androidx.cardview.widget.CardView>

10. fragment_detail.xml

Tabel 10. Source Code fragment_detail.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	xmlns:tools="http://schemas.android.com/tools"
5	android:layout_width="match_parent"
6	android:layout_height="match_parent"
7	tools:context=".ui.fragment.DetailFragment">
8	
9	<androidx.constraintlayout.widget.ConstraintLayout
10	android:layout_width="match_parent"
11	android:layout_height="wrap_content"
12	android:padding="16dp">
13	
14	<ImageView
15	android:id="@+id/item_image_detail"
16	android:layout_width="300dp"
17	android:layout_height="170dp"
18	android:layout_marginTop="32dp"
19	app:layout_constraintEnd_toEndOf="parent"
20	app:layout_constraintStart_toStartOf="parent"
21	app:layout_constraintTop_toTopOf="parent"
	/>
22	
23	<TextView
24	android:id="@+id/item_name_detail"
25	android:layout_width="0dp"
26	android:layout_height="wrap_content"
27	android:layout_marginTop="24dp"
28	android:text="Nama item"
29	android:textAlignment="center"
30	android:textColor="#000000"
31	android:textSize="24sp"
32	android:textStyle="bold"
33	app:layout_constraintEnd_toEndOf="parent"
34	app:layout_constraintStart_toStartOf="parent"
35	app:layout_constraintTop_toBottomOf="@id/item_image_detail" />

```

36
37         <TextView
38             android:id="@+id/item_base_detail"
39             android:layout_width="0dp"
40             android:layout_height="wrap_content"
41             android:layout_marginTop="16dp"
42             android:text="Team base"
43             android:textAlignment="viewStart"
44             android:textSize="16sp"
45             app:layout_constraintEnd_toEndOf="parent"
46 app:layout_constraintStart_toStartOf="parent"
47 app:layout_constraintTop_toBottomOf="@id/item_name_detail" />
48
49         <TextView
50             android:id="@+id/item_chief_detail"
51             android:layout_width="0dp"
52             android:layout_height="wrap_content"
53             android:layout_marginTop="16dp"
54             android:text="Team Principal"
55             android:textAlignment="viewStart"
56             android:textSize="16sp"
57             app:layout_constraintEnd_toEndOf="parent"
58 app:layout_constraintStart_toStartOf="parent"
59 app:layout_constraintTop_toBottomOf="@id/item_base_detail" />
60
61         <TextView
62             android:id="@+id/item_chassis_detail"
63             android:layout_width="0dp"
64             android:layout_height="wrap_content"
65             android:layout_marginTop="16dp"
66             android:text="Chassis"
67             android:textAlignment="viewStart"
68             android:textSize="16sp"
69             app:layout_constraintEnd_toEndOf="parent"
70 app:layout_constraintStart_toStartOf="parent"
71 app:layout_constraintTop_toBottomOf="@id/item_chief_detail" />
72
73         <TextView

```

```

74         android:id="@+id/item_wc_detail"
75         android:layout_width="0dp"
76         android:layout_height="wrap_content"
77         android:layout_marginTop="16dp"
78         android:text="Championship"
79         android:textAlignment="viewStart"
80         android:textSize="16sp"
81         app:layout_constraintEnd_toEndOf="parent"
82         app:layout_constraintStart_toStartOf="parent"
83         app:layout_constraintTop_toBottomOf="@id/item_chassis_
detail" />
84
85         <TextView
86             android:id="@+id/item_desc_detail"
87             android:layout_width="0dp"
88             android:layout_height="wrap_content"
89             android:layout_marginTop="20dp"
90             android:text="Desc"
91             android:textAlignment="viewStart"
92             android:justificationMode="inter_word"
93             android:lineSpacingExtra="4dp"
94             android:textSize="16sp"
95             app:layout_constraintTop_toBottomOf="@id/item_wc_detai
l"
96             app:layout_constraintStart_toStartOf="parent"
97             app:layout_constraintEnd_toEndOf="parent"
98             />
99     </androidx.constraintlayout.widget.ConstraintLayout>
100 </ScrollView>

```

11. strings.xml

Tabel 11. Source Code strings.xml

1	<resources>
2	<string name="app_name">F1 Team</string>
3	<string name="btn_team">F1 Team</string>
4	<string name="btn_detail">Detail</string>
5	
6	
7	<string-array name="data_name">
8	<item>Ferrari</item>
9	<item>Mclaren</item>
10	<item>Mercedes</item>
11	<item>Red Bull Racing</item>
12	<item>Williams</item>
13	
14	</string-array><string-array name="data_base">
15	<item>Base : Maranello, Italy</item>
16	<item>Base : Woking, United Kingdom</item>
17	<item>Base : Brackley, United Kingdom</item>
18	<item>Base : Milton Keynes, United
19	Kingdom</item>
20	<item>Base : Grove, United Kingdom</item>
21	</string-array>
22	<string-array name="data_chief">
23	<item>Team Chief : Frédéric Vasseur</item>
24	<item>Team Chief : Andrea Stella</item>
25	<item>Team Chief : Toto Wolff</item>
26	<item>Team Chief : Christian Horner</item>
27	<item>Team Chief : James Vowles</item>
28	</string-array>
29	
30	<string-array name="data_chassis">
31	<item>Chassis : SF-25</item>
32	<item>Chassis : MCL39</item>
33	<item>Chassis : W16</item>
34	<item>Chassis : RB21</item>
35	<item>Chassis : FW47</item>
36	</string-array>
37	
38	<string-array name="data_champion">
39	<item>World Championship : 16</item>
40	<item>World Championship : 9</item>
41	<item>World Championship : 8</item>

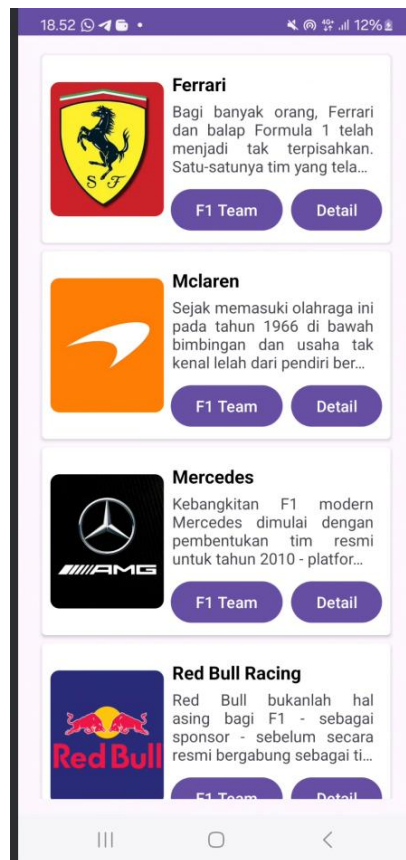
42	<code><item>World Championship : 6</item></code>
43	<code><item>World Championship : 9</item></code>
44	<code></string-array></code>
45	
46	<code><string-array name="data_desc"></code>
47	<code><item></code>
48	Bagi banyak orang, Ferrari dan balap
	Formula 1 telah menjadi tak terpisahkan.
49	Satu-satunya tim yang telah berkompetisi
	di setiap musim sejak kejuaraan dunia dimulai,
50	Ferrari telah berkembang dari mimpi
	sederhana pendirinya, Enzo Ferrari, menjadi salah
	satu merek paling ikonik dan diakui di dunia.
51	Kesuksesan datang dengan cepat melalui
	sosok-sosok seperti Alberto Ascari dan John Surtees,
	dan berlanjut - di tengah masa-masa sulit
52	- dengan Niki Lauda pada tahun 1970-an
	dan kemudian Michael Schumacher pada tahun 2000-an,
53	ketika Ferrari meraih lima gelar ganda
	berturut-turut yang belum pernah terjadi sebelumnya,
54	mengukuhkan status mereka sebagai tim
	paling sukses dan terhiasi dalam sejarah F1
55	<code></item></code>
56	<code><item></code>
57	Sejak memasuki olahraga ini pada tahun
	1966 di bawah bimbingan dan usaha tak kenal lelah
	dari pendiri bernama sama,
58	Bruce, kesuksesan McLaren tidak kurang
	dari menakjubkan.
59	Lima dekade yang gemerlap telah
	menghasilkan tak terhitung jumlahnya kemenangan,
	posisi pole, dan podium,
60	belum lagi sembilan kejuaraan
	konstruktor. Lebih dari itu,
61	beberapa pembalap terbaik dalam olahraga
	ini membuat nama mereka bersama tim ini,
62	termasuk Emerson Fittipaldi, Ayrton
	Senna, Mika Hakkinen, dan Lewis Hamilton
63	<code></item></code>
64	<code><item></code>
65	Kebangkitan F1 modern Mercedes dimulai
	dengan pembentukan tim resmi untuk tahun 2010 -
66	platform untuk kenaikan meteorit ke
	urutan Grand Prix.
67	Tim ini menghasilkan kegembiraan besar
	sejak awal dengan kembalinya sensasional Michael
	Schumacher,

68	tetapi berita utama segera menyusul di lintasan: tiga podium di musim debut mereka, semuanya melalui Nico Rosberg -
69	yang kemudian meraih pole/victory ganda yang luar biasa di China pada tahun 2012.
70	Musim berikutnya dia dipasangkan dengan Lewis Hamilton,
71	duo ini melanjutkan untuk menggelar beberapa pertarungan epik untuk gelar juara saat Silver Arrows
72	menyapu bersih semua di depan mereka dan menjadi salah satu kekuatan paling dominan di era F1 modern -
73	sampai Red Bull datang dan mencuri mahkota itu. Namun demikian,
74	dengan pemenang balapan yang terbukti George Russell kini dipasangkan dengan bintang yang sedang naik daun Kimi Antonelli,
75	Mercedes tetap menjadi salah satu tim yang harus dikalahkan
76	</item>
77	<item>
78	Red Bull bukanlah hal asing bagi F1 - sebagai sponsor - sebelum secara resmi bergabung sebagai tim pabrikan pada tahun 2004.
79	Namun demikian, skala kesuksesan mereka selama dekade berikutnya sangat mengejutkan.
80	Setelah podium pertama pada tahun 2006, tim tersebut menemukan ritme mereka pada tahun 2009,
81	meraih enam kemenangan dan posisi kedua dalam klasemen konstruktor. Selama empat musim berikutnya,
82	mereka menjadi kekuatan yang tak terbendung, meraih gelar ganda berturut-turut antara 2010 dan 2013,
83	dengan Sebastian Vettel muncul sebagai juara kuadruple termuda dalam olahraga ini.
84	Sekarang mereka merebut kembali kejayaan itu dengan bakat yang sama menariknya - seseorang bernama Max Verstappen
85	</item>
86	<item>
87	Didorong oleh semangat dan kecermelangan almarhum Sir Frank Williams,
88	Williams tumbuh dari awal yang sederhana menjadi raksasa Formula 1,
89	tak tertandingi oleh siapa pun kecuali

	Ferrari dan McLaren dalam hal kesuksesan yang berkelanjutan.
90	Selama empat dekade terakhir, tim ini telah mengumpulkan kemenangan Grand Prix dan kejayaan kejuaraan,
91	dan dalam prosesnya telah membina beberapa bakat terbesar dalam olahraga ini, baik di dalam maupun di luar kokpit.
92	Dan, setelah keputusan keluarga Williams untuk mundur setelah penjualan tim kepada Dorilton Capital pada tahun 2020,
93	era baru telah dimulai
94	</item>
95	</string-array>
96	
97	<string-array name="data_link">
98	<item>https://www.formula1.com/en/teams/ferrari</item>
99	<item>https://www.formula1.com/en/teams/mclaren</item>
100	<item>https://www.formula1.com/en/teams/mercedes</item>
101	<item>https://www.formula1.com/en/teams/red-bull-racing</item>
102	<item>https://www.formula1.com/en/teams/williams</item>
103	</string-array>
104	
105	<string-array name="data_image">
106	<item>https://cdn1-production-images-kly.akamaized.net/jMacjCAV_TAZCRybVLNje9eDq0Q=/800x450/smart/filters:quality(75):strip_icc():format(webp)/kly-media-production/medias/1299410/original/085836400_1469592744-logo.jpg</item>
107	<item>https://static.wikia.nocookie.net/f1-formula-1/images/e/ed/McLaren.jpg/revision/latest?cb=20230118201145</item>
108	<item>https://static.wikia.nocookie.net/f1-formula-1/images/1/1e/Mercedes.jpg/revision/latest?cb=20230119001246</item>

109	<item>https://www.logotypes101.com/logos/192/9A4970424CE2C2B5BF5DC41EDF29DE3E/tn_red_bull.png</item>
110	<item>https://preview.redd.it/a-fused-williams-and-atlassian-logo-to-commemorate-the-new-v0-bph9mq8j3vie1.png?width=1080&crop=smart&auto=webp&s=1b5e7228843f0376166dc5028c78b390dd6d83fd</item>
111	</string-array>
112	
113	<string-array name="data_image_detail">
114	<item>https://cdn.ferrari.com/cms/network/media/img/resize/67587a0c4e5f24001074a2ca-scuderia-ferrari-cadillac-2026-news</item>
115	<item>https://d3cm515ijfiu6w.cloudfront.net/wp-content/uploads/2025/01/23171426/mclaren-logo-2024.jpg</item>
116	<item>https://cdn-9.motorsport.com/images/mgl/0RrzmDo0/s8/mercedes-f1-logo-1.jpg</item>
117	<item>https://cdn-5.motorsport.com/images/mgl/Y99JQRbY/s8/red-bull-racing-logo-1.jpg</item>
118	<item>https://upload.wikimedia.org/wikipedia/commons/f/f9/Williamsracing-20200530-0001.jpg</item>
119	</string-array>
120	</resources>

B. Output Program



Gambar 3. Screenshot Tampilan Halaman Home Potrait Aplikasi



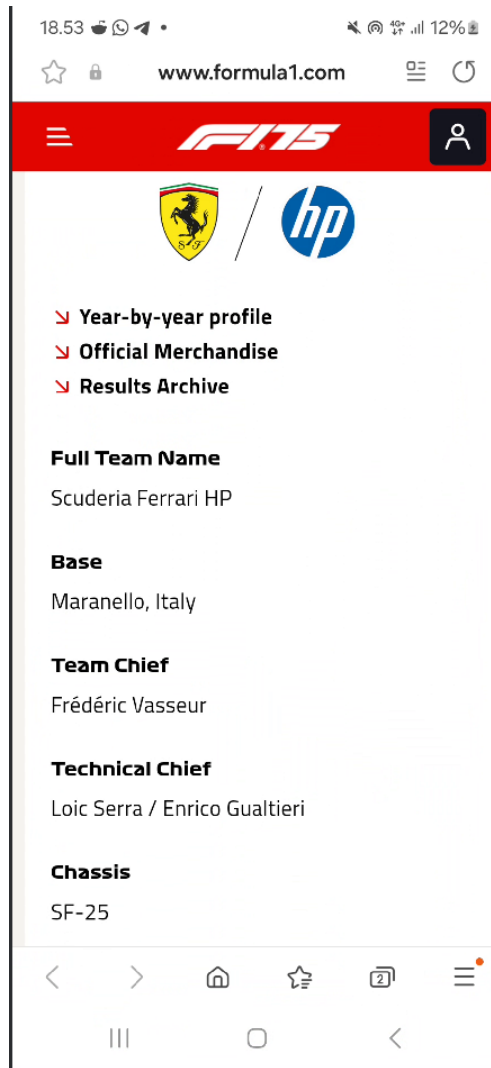
Gambar 4. Screenshot Tampilan Halaman Detail Potrait Aplikasi



Gambar 5. Screenshot Tampilan Halaman Home Landscape Aplikasi



Gambar 6. Screenshot Tampilan Halaman Detail Landscape Aplikasi



Gambar 7. Screenshot Membuka Browser F1 Team Setelah Klik "F1 Team"

C. Pembahasan

1. MainActivity.kt:

- File **MainActivity.kt** adalah kelas utama yang menangani tampilan dan logika UI utama aplikasi F1 Team yang mengikuti arsitektur MVVM (Model-View-ViewModel) dan juga Single Activity. Single Activity berarti beberapa fragment untuk tampilan halaman yang berbeda yang dilakukan hanya pada satu activity.
- Baris [1] adalah deklarasi package menunjukkan bahwa file ini berada dalam package **com.example.f1team**.
- Baris [3] dan baris [4] adalah mengimpor paket – paket yang dibutuhkan dalam kode ini, termasuk fitur **Bundle** untuk menyimpan data saat activity diciptakan, dan **AppCompatActivity** yang mana turunan dari Activity.
- Baris [6], **class MainActivity : AppCompatActivity() {** adalah kelas utama untuk single activity yang dijalankan saat aplikasi dibuka.
- Baris [7] sampai baris [10] adalah **onCreate()** adalah fungsi yang dipanggil saat activity pertama kali atau titik masuk activity. **setContentView(R.layout.activity_main)** adalah menghubungkan Activity dengan layout XML.

2. F1Team.kt

- F1Team.kt adalah kelas data atau model objek tentang informasi tim F1.
- Baris [1], **package com.example.f1team.model** adalah deklarasi package yang menunjukkan bahwa file ini berada di package **model**.
- Baris [3] dan baris [4] adalah mengimpor paket – paket yang digunakan dalam kode ini, yaitu **Parcelable** dan juga **parcelize**.
- Baris [6], anotasi **@Parcelize** digunakan supaya objek **F1Team** dapat dikirim antar komponen Android menggunakan **Bundle** atau **Intent**.
- Baris [7] sampai baris [16] adalah kelas data **F1Team** yang mencakup properti bertipe **String** yang mana masing – masing menyimpan informasi tentang sebuah tim F1. Berikut ini adalah properti **F1Team**: **name** adalah nama tim f1, **base** adalah Lokasi markas tim, **teamChief** adalah nama kepala

tim (*team principal*), *chassis* adalah nama sasis mobil yang digunakan, *worldChampion* adalah jumlah gelar dunia yang dimiliki tim, *desc* adalah deskripsi tim, *link* adalah URL referensi terkait tim, *imageUrl* adalah link gambar utama untuk tampilan list, dan *imageUrlDetail* adalah untuk link gambar tampilan detail.

3. HomeFragment.kt

- File *HomeFragment.kt* adalah fragment yang menampilkan *daftar tim F1* menggunakan RecyclerView sehingga dapat di scroll. Terdapat informasi tim F1 dan juga tombol “*F1 Team*” untuk ke situs F1 tentang team terkait serta tombol detail untuk navigasi ke *DetailFragment* menggunakan Navigation Component.
- Baris [1], *Package com.example.f1team.ui.fragment* adalah menandakan bahwa file ini berada di paket *fragment* dalam paket *ui*.
- Baris [3] sampai baris [15] adalah mengimpor paket – paket yang digunakan dalam kode ini, seperti *Intent*, *Uri*, *Bundle*, *Fragment*, *LayoutInflater*, *View* dan *ViewGroup*, *ViewModel*, *ViewBinding*, *Adapter*, dan *Navigation Component*.
- Baris [17], *class HomeFragment : Fragment()*, adalah deklarasi fragment bagian UI home yang pertama kali dimuat dalam main activity.
- Baris [19] dan baris [20] adalah deklarasi dan inisialisasi *ViewBinding*. *_binding* hanya hidup selama fragment berada di *view lifecycle*.
- Baris [22] dan baris [23] adalah deklarasi *viewModel* yaitu dari kelas *F1TeamViewModel* dan juga *adapter* dari kelas *F1TeamAdapter*. *ViewModel* digunakan untuk menyimpan dan menyediakan data tim F1 dan *Adapter* untuk menampilkan data *F1Team* di *RecyclerView*.
- Baris [25] sampai baris [31] adalah fungsi *onCreateView()* yang mana akan meng-inflate layout XML (*fragment_home.xml*) menggunakan ViewBinding dari *_binding* (*_binding = FragmentHomeBinding.inflate(inflater,*

container, false)). Kemudian, *return binding.root* akan mengembalikan root view dari layout sebagai tampilan fragment.

- Baris [33] sampai baris [45] adalah fungsi *onViewCreated()* yang berisi logika utama dijalankan setelah view terbentuk (*onCreateView*). Kode *viewModel = ViewModelProvider(this)[F1TeamViewModel::class.java]* berarti mengambil instance *F1TeamViewModel* untuk fragment ini dan juga memastikan *ViewModel* mengikuti lifecycle fragment. Kemudian terdapat insialisasi *adapter* berdasarkan kelas *F1TeamAdapter* dengan lambda. *onTeamClick* merupakan aksi untuk membuka link eksternal di browser dengan menggunakan *Intent.ACTION_VIEW*. *onDetailClick* sendiri adalah navigasi ke *DetailFragment* dengan membawa objek *F1Team* menggunakan *Safe Args*.
- Baris [47] dan baris [48] adalah untuk setup *RecyclerView* yang menggunakan *LinearLayoutManager* untuk menampilkan item secara vertikal di *RecyclerView* *rvItem* (*binding.rvItem.layoutManager = LinearLayoutManager(requireContext())*) dan menetapkan adapter yang sudah dikustomisasi (*binding.rvItem.adapter = adapter*).
- Baris [50] sampai baris [52] adalah *Observe LiveData* yang mengamati data dari *ViewModel* (*LiveData<List<F1Team>>*). Saat data berubah, *adapter* akan otomatis menampilkan list baru dengan menggunakan fungsi *submitList()* dari kelas *adapter*.
- Baris [55] sampai baris [58] adalah fungsi *onDestroyView()* yang menghancurkan view fragment. Fungsi ini penting untuk menghindari memory leak dengan mengosongkan (null) *_binding* saat fragment ini hancur.

4. DetailFragment.kt

- File DetailFragment.kt adalah fragment yang menampilkan informasi detail tentang satu tim F1, yang sebelumnya dipilih di *HomeFragment*. Data tim dikirim menggunakan **Safe Args** (*Navigation Component*), lalu ditampilkan di UI.
- Baris [1], *package com.example.f1team.ui.fragment* adalah deklarasi package yang menunjukkan bahwa file ini berada di folder *fragment* dalam dari paket folder *ui*.
- Baris [3] sampai baris [10] adalah mengimpor paket – paket yang digunakan dalam kelas ini, yaitu kelas-kelas dasar *Android Fragment* dan *View*, *Glide*, *FragmentDetailBinding*, dan *F1Team*.
- Baris [13], class DetailFragment : Fragment(), adalah deklarasi fragment bagian UI detail setelah mengklik tombol “*Detail*” dari *HomeFragment*.
- Baris [15] dan baris [16] adalah deklarasi dan inisialisasi *ViewBinding* *_binding* untuk mengakses elemen view secara langsung (*_binding: FragmentDetailBinding? = null*). Variabel *binding* digunakan setelah view di-inflate dan sebelum dihancurkan.
- Baris [18], *private var team: F1Team? = null* adalah menyimpan objek tim (*F1Team*) yang dikirim dari fragment sebelumnya (*HomeFragment*) melalui **Safe Args** yang disimpan ke variabel *team*.
- Baris [20] sampai baris [25] adalah fungsi *onCreate()* untuk titik masuk dari fragment ini. Ketika masuk, fungsi ini akan mendapatkan data *F1Team* dari argument yang dikirim melalui **Safe Args** (*team* dikirim dari *HomeFragment*) dan data tersebut akan disimpan dalam variabel *team* (*team = DetailFragmentArgs.fromBundle(it).team*).
- Baris [27] sampai baris [30] adalah fungsi *onCreateView()* yang mana akan meng-inflate layout *fragment_detail.xml* menggunakan *ViewBinding* dan juga mengembalikan view root dari fragment untuk ditampilkan di layar.
- Baris [32] sampai baris [45] adalah fungsi *onViewCreated()* adalah logika ketika view sudah terbentuk. Jika *team* tidak null, maka data ditampilkan ke UI dengan *TextView* binding yang digunakan untuk menampilkan nama tim,

Lokasi markas, kepala tim, sasis mobil, gelar juara dunia, dan deskripsi tim. Glide sendiri digunakan untuk memuat gambar URL detail ke dalam *imageView*.

- Baris [47] sampai baris [50] adalah fungsi *onDestroyView()* yang membersihkan *_binding* menjadi null saat fragment dihancurkan untuk mencegah memory leak.

5. *F1TeamViewModel.kt*

- File *F1TeamViewModel.kt* adalah **ViewModel** yang berfungsi untuk menyediakan daftar data tim F1 ke UI (*HomeFragment*), mengambil data dari resource XML (*strings.xml*) dalam bentuk array dan mengubahnya menjadi objek *F1Team*, lalu membungkusnya dengan *LiveData*.
- Baris [1], *package com.example.f1team.viewmodel* adalah deklarasi package yang menunjukkan bahwa file ini berada di paket *viewmodel*.
- Baris [3] sampai baris [8] adalah mengimpor paket – paket yang digunakan dalam kelas ini, yaitu *Application* dan *AndroidViewModel*, *LiveData*, *MutableLiveData*, *F1Team*, dan *R*.
- Baris [10], *class F1TeamViewModel(application: Application) : AndroidViewModel(application)* adalah turunan dari *AndroidViewModel* karena membutuhkan akses ke *application.resources (strings.xml)*.
- Baris [11] sampai baris [12] adalah properti *LiveData*. Variabel *_teamList* adalah properti yang bisa diubah dari dalam *ViewModel* menggunakan *MutableLiveData<List<F1Team>>()*. Sedangkan *teamList* adalah versi publik yang hanya bisa dibaca (read-only) oleh UI (seperti *HomeFragment*) menggunakan *LiveData<List<F1Team>>*.
- Baris [14] sampai baris [44] adalah blok *init* untuk mengisi dan mengambil data dari *strings.xml* menggunakan *application.resources.getStringArray()*. Data yang diambil adalah *data_name*, *data_base*, *data_chief*, *data_chassis*, *data_champion*, *data_desc*, *data_link*, *data_image*, dan *data_image*. Semua diubah menjadi list agar bisa diakses per indeks. Terdapat loop *for(i in*

name.indices) yang dilakukan sebanyak jumlah item pada array *name*. Setiap elemen digunakan untuk membuat satu objek *F1Team*, lalu dimasukkan ke list *f1Team*. Setelah semua data dibuat, maka akan disimpan ke dalam *_teamList* (*_teamList.value = f1Team*).

6. F1TeamAdapter.kt

- File *F1TeamAdapter.kt* adalah adapter *RecyclerView* yang digunakan untuk menampilkan daftar objek *F1Team* dalam bentuk list, meng-handle klik dua tombol: " *F1 Team*" (buka link) dan "*Detail*" (navigasi ke detail), dan memuat gambar tim dengan efek *rounded corner* menggunakan *Glide*.
- Baris [1], *package com.example.f1team.ui.adapter* adalah meyimpan adapter di dalam paket *adapter* di dalam paket *ui*.
- Baris [3] sampai baris [10] adalah megimpor paket – paket yang digunakan dalam kelas ini yaitu, *LayoutInflater*, *ViewGroup*, *RecyclerView*, *Glide*, *CenterCrop*, *RoundedCorners*, *ItemF1Binding*, dan *F1Team*.
- Baris [12] sampai baris [16] adalah kelas deklarasi kelas adapter *F1TeamAdapter*. Variabel *onTeamClick* dan *onDetailClick* adalah lambda function yang dikirim dari luar adapter, biasanya dari fragment (misalnya: untuk buka browser dan navigasi).
- Baris [18], *private val teamList = mutableListOf<F1Team>()* adalah menyimpan daftar tim (*teamList*) yang ditampilkan oleh adapter.
- Baris [20] sampai baris [24] adalah fungsi *submitList()* untuk memperbarui data yang ditampilkan dengan membersihkan terlebih dahulu (*teamList.clear()*) dan kemudian menambahkan list ulang (*teamList.addAll(list)*). Setelah itu, memanggil *notifyDataSetChanged()* agar *RecyclerView* me-refresh tampilannya.
- Baris [26] sampai baris [29] adalah fungsi *onCreateViewHolder()* yang mana akan meng-inflate layout untuk satu item list (*item_f1.xml*) dan membungkusnya dalam *ViewHolder*.

- Baris [31] sampai baris [33] adalah ***onBindViewHolder()*** yang mana akan memanggil fungsi ***bind()*** milik ***ViewHolder*** untuk menampilkan data tim pada posisi ***position***.
- Baris [35] adalah fungsi ***getItemCount()*** yang menentukan berapa banyak item dalam daftar dengan menggunakan ***teamList.count()***.
- Baris [37] adalah kelas dalam/inner class ***ViewHolder F1TeamViewHolder*** yang menyimpan referensi ke view setiap item dan menampilkan datanya.
- Baris [38] sampai baris [48] adalah fungsi ***bind()*** yang mengatur binding ***itemName*** dan ***itemDesc*** untuk menampilkan nama dan deskripsi tim, ***itemImageDetail*** yang memuat gambar menggunakan ***Glide*** dengan styling., ***btnTeam*** akan memicu fungsi ***onTeamClick()*** saat diklik (misalnya membuka browser), dan ***btnDetail*** akan memicu fungsi ***onDetailClick()*** saat diklik (navigasi ke detail).

7. File ***activity_main.xml***

- File ***activity_main.xml*** adalah root layout dari ***MainActivity***, yang hanya berisi ***NavHostFragment*** sebagai container untuk fragment-fragment yang ditampilkan melalui ***Navigation Component***.
- Baris [2] ***<FragmentContainerView>*** digunakan sebagai host fragment untuk navigasi.
- Baris [6], ***android:id="@+id/nav_host_fragment"*** adalah ID unik dari container ini yang digunakan oleh ***NavController*** untuk menemukan ***NavHostFragment***.
- Baris [7], ***android:name="androidx.navigation.fragment.NavHostFragment"*** adalah menentukan bahwa ini adalah ***NavHostFragment***, yaitu komponen yang menampilkan fragment dari ***nav_graph***.
- Baris [10], ***app:defaultNavHost="true"*** artinya fragment ini akan menangani aksi ***navigate up*** (tombol back di toolbar).

- Baris [11], `app:navGraph="@navigation/nav_graph"` adalah menghubungkan ke file `nav_graph.xml`, yang berisi skema navigasi (fragment, action, argument, dll).

8. File fragment_home.xml

- File `fragment_home.xml` adalah layout yang digunakan untuk menampilkan daftar tim F1 dalam bentuk **RecyclerView**.
- Layout ini menggunakan **ConstraintLayout** sebagai *layout root* yang fleksibel sehingga memungkinkan untuk mengatur posisi elemen secara relatif terhadap parent atau elemen lainnya.
- Baris [9] sampai baris [19] adalah **RecyclerView** yang Menampilkan list tim F1 yang di-handle oleh **F1TeamAdapter**. Elemen ini memiliki id `rv_item` untuk diakses dari `HomeFragment.kt`. dan juga elemen ini mengisi seluruh area parent layout dengan margin 15dp. **RecyclerView** ditempel ke keempat sisi parent-nya (atas, bawah, kiri, kanan).

9. File item_f1.xml

- File `item_f1.xml` adalah layout ini digunakan untuk menampilkan satu kartu berisi, gambar tim, nama tim, deskripsi singkat, dan dua tombol: “**Detail**” dan “**F1 Team**” yang ditampilkan di **RecyclerView** pada daftar tim F1.
- File ini berisi **CardView** untuk membungkus seluruh tampilan item agar terlihat seperti kartu dengan sudut membulat dan memberikan efek bayangan dan padding otomatis.
- **ConstraintLayout** digunakan untuk layout yang fleksibel untuk menempatkan elemen-elemen di dalam kartu. Semua elemen (gambar, teks, tombol) diatur posisinya secara relatif satu sama lain.
- Baris [19] sampai baris [27] adalah **ImageView** (`item_image_detail`) yang menampilkan gambar tim F1 (dari `imageUrl`). Style **centerCrop** berarti gambar akan dipotong agar memenuhi ukuran view tanpa merusak rasio.

- Baris [29] sampai baris [43] adalah ***TextView (item_name)*** yang menampilkan nama tim, ditempatkan di sebelah kanan gambar (start ke end gambar), dan memiliki huruf tebal dan ukuran 16sp.
- Baris [45] sampai baris [58] adalah ***TextView (item_desc)*** yang menampilkan deskripsi singkat tim dengan maksimal 4 baris. Jika lebih, akan dipotong dengan “.....”. Elemen ini ditempatkan di bawah nama tim dan disamping gambar.
- Baris [60] sampai baris [68] adalah ***Button "F1 Team" (btn_team)*** yang digunakan untuk membuka link resmi atau eksternal dari tim (misalnya situs resmi) dan dihubungkan dengan ***onTeamClick*** di Adapter. Tombol ini diposisikan di sebelah kiri tombol “Detail” dan dibawah deskripsi.
- Baris [70] sampai baris [79] adalah ***Button "Detail" (btn_detail)*** yang digunakan untuk membuka halaman detail dari tim dan dihubungkan dengan ***onDetailClick*** di Adapter. Tombol ini berada di sebelah kanan tombol “***F1 Team***” dan berada di bawah deskripsi.

10. File fragment_detail.xml

- File ***fragment_detail.xml*** adalah layout yang digunakan untuk menampilkan detail lengkap dari sebuah tim F1.
- Layout ini menggunakan ***ScrollView*** untuk membungkus seluruh layout agar bisa di scroll secara vertikal jika isi konten lebih tinggi dari layar. ***ScrollView*** wajib memiliki hanya satu child view, yaitu ***ConstraintLayout*** di dalamnya.
- ***ConstraintLayout*** digunakan sebagai layout utama tempat semua elemen ditaruh dan diatur posisinya secara fleksibel dan menggunakan padding 16dp agar isi tidak menempel ke tepi layar.
- Baris [14] sampai baris [21] adalah ***ImageView (item_image_detail)*** yang menampilkan gambar besar (dari ***imageUrlDetail***) dari tim F1 dan diletakkan di bagian atas (top) dan diposisikan di tengah secara horizontal.
- Baris [23] sampai baris [25] adalah ***TextView (item_name_detail)*** menampilkan nama tim dengan ukuran besar 24sp, bold, dan rata tengah.

- Baris [37] sampai baris [97] adalah *TextView* yang digunakan untuk menampilkan asal tim (markas), team principal, nama sasis mobil, jumlah gelar juara dunia dan deskripsi lengkap tim.

11. File strings.xml

- File *string.xml* adalah resource file string yang berisi data statis untuk aplikasi Android bertema tim Formula 1, termasuk nama tim, informasi detail, gambar, dan deskripsi. Setiap data tim f1 disimpan sebagai string array kemudian diakses oleh *F1TeamViewModel*.

SOAL 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 banyak digunakan di Android, meskipun lebih verbose dan boilerplate dibandingkan LazyColumn di Jetpack Compose, dikarenakan RecyclerView adalah bagian dari View System tradisional yang mana masih banyak digunakan dalam kode legasi di perusahaan. RecyclerView juga sudah matang dan stabil, digunakan selama bertahun-tahun dengan dukungan ekosistem luas sehingga memberi kontrol penuh terhadap lifecycle item, view caching, animasi, dan layouting. Banyak Perusahaan masih memiliki codebase besar berbasis XML dan View System, sehingga mengganti semua ke Compose bukan hal yang mudah dan juga harus bertahap.

TAUTAN GIT

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Kylorts/Pemrograman-mobile/tree/main/Modul%203>