

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 5**



CONNECT TO THE INTERNET

Oleh:

Siti Ratna Dwinta Sari

NIM. 2310817120002

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2024**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN I
MODUL 5

Laporan Praktikum Pemrograman Mobile Modul 5: Connect to the Internet ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Siti Ratna Dwinta Sari
NIM : 2310817120002

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Zulfa Auliya Akbar
NIM. 2210817210026

Muti`a Maulida S.Kom M.T.I
NIP. 198810272019032013

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1.....	6
A. Source Code	6
B. Output Program.....	32
C. Pembahasan	35
D. Tautan Git	42

DAFTAR GAMBAR

Gambar 1. Tampilan Home	32
Gambar 2. Tampilan Favorites	32
Gambar 3. Tampilan Home saat tidak ada jaringan	33
Gambar 4. Tampilan Favorites saat tidak ada jaringan, masih bisa diakses	33
Gambar 5. Tampilan Detail	34
Gambar 6. Tampilan Wiki	34

DAFTAR TABEL

Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt	6
Tabel 2. Source Code Jawaban Soal 1 DetailFragment.kt	7
Tabel 3. Source Code Jawaban Soal 1 HomeFragment.kt	9
Tabel 4. Source Code Jawaban Soal 1 CountryAdapter.kt	11
Tabel 5. Source Code Jawaban Soal 1 DreamsTravelApp.kt.....	12
Tabel 6. Source Code Jawaban Soal 1 FavoriteFragment.kt.....	13
Tabel 7. Source Code Jawaban Soal 1 UiState.kt	15
Tabel 8. Source Code Jawaban Soal 1 CountryViewModel.kt	16
Tabel 9. Source Code Jawaban Soal 1 viewmodel/CountryViewModelFactory.kt	18
Tabel 10. Source Code Jawaban Soal 1 data/local/AppDataBase.kt.....	19
Tabel 11. Source Code Jawaban Soal 1 data/local/CountryDao.kt	19
Tabel 12. Source Code Jawaban Soal 1 data/local/CountryEntity.kt	20
Tabel 13. Source Code Jawaban Soal 1 data/local/CountryRepository.kt	21
Tabel 14. Source Code Jawaban Soal 1 data/remote/ApiService.kt.....	22
Tabel 15. Source Code Jawaban Soal 1 data/remote/CountryResponse.kt	22
Tabel 16. Source Code Jawaban Soal 1 data/remote/RetrofitClient.kt	23
Tabel 17. Source Code Jawaban Soal 1 layout/activity_main.xml	24
Tabel 18. Source Code Jawaban Soal 1 layout/fragment_detail.xml	24
Tabel 19. Source Code Jawaban Soal 1 layout/fragment_favorite.xml	26
Tabel 20. Source Code Jawaban Soal 1 layout/fragment_home.xml	27
Tabel 21. Source Code Jawaban Soal 1 item_negara.xml.....	28
Tabel 22. Source Code Jawaban Soal 1 navigation/nav_graph.xml.....	31

SOAL 1

Lanjutkan aplikasi Android yang sudah dibuat pada Modul 4 dengan menambahkan modifikasi sesuai ketentuan berikut:

- a Gunakan networking library seperti Retrofit atau Ktor agar aplikasi dapat mengambil data dari remote API. Dalam penggunaan networking library, sertakan generic response untuk status dan error handling pada API dan Flow untuk data stream.
- b Gunakan KotlinX Serialization sebagai library JSON.
- c Gunakan library seperti Coil atau Glide untuk image loading.
- d API yang digunakan pada modul ini bebas, contoh API gratis The Movie Database (TMDB) API yang menampilkan data film. Berikut link dokumentasi API: <https://developer.themoviedb.org/docs/getting-started>
- e Implementasikan konsep data persistence (misalnya offline-first app, pengaturan dark/light mode, fitur favorite, dll)
- f Gunakan caching strategy pada Room..
- g Untuk Modul 5, bebas memilih UI yang ingin digunakan, antara berbasis XML atau Jetpack Compose.

A. Source Code

1. MainActivity.kt

Tabel 1. Source Code Jawaban Soal 1 MainActivity.kt

1	package com.example.dreamstravel
2	
3	import android.os.Bundle
4	import androidx.appcompat.app.AppCompatActivity
5	import androidx.navigation.fragment.NavHostFragment
6	import androidx.navigation.ui.setupWithNavController
7	import com.example.dreamstravel.databinding.ActivityMainBinding
8	
9	class MainActivity : AppCompatActivity() {
10	
11	private lateinit var binding: ActivityMainBinding
12	

```

13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         binding = ActivityMainBinding.inflate(layoutInflater)
16         setContentView(binding.root)
17
18         val navHostFragment =
supportFragmentManager.findFragmentById(R.id.nav_host_fragment)
as NavHostFragment
19         val navController = navHostFragment.navController
20
21 binding.bottomNavigation.setupWithNavController(navController)
22     }
23 }

```

2. DetailFragment.kt

Tabel 2. Source Code Jawaban Soal 1 DetailFragment.kt

```

1 package com.example.dreamstravel
2
3 import android.os.Bundle
4 import android.view.LayoutInflater
5 import android.view.View
6 import android.view.ViewGroup
7 import androidx.core.view.isVisible
8 import androidx.fragment.app.Fragment
9 import androidx.fragment.app.activityViewModels
10 import androidx.lifecycle.LifecycleScope
11 import androidx.navigation.fragment.NavArgs
12 import coil.load
13 import
com.example.dreamstravel.databinding.FragmentDetailBinding
14 import com.example.dreamstravel.viewmodel.CountryViewModel
15 import
com.example.dreamstravel.viewmodel.CountryViewModelFactory
16 import kotlinx.coroutines.launch
17
18 class DetailFragment : Fragment() {
19
20     private var _binding: FragmentDetailBinding? = null
21     private val binding get() = _binding!!
22
23     private val args: DetailFragmentArgs by navArgs()
24

```

```

25     private val viewModel: CountryViewModel by
activityViewModels {
26         CountryViewModelFactory(
27             requireActivity().application,
28             (requireActivity().application as
DreamsTravelApp).repository
29         )
30     }
31
32     override fun onCreateView(
33         inflater: LayoutInflater, container: ViewGroup?,
34         savedInstanceState: Bundle?
35     ): View {
36         _binding = FragmentDetailBinding.inflate(inflater,
container, false)
37         return binding.root
38     }
39
40     override fun onViewCreated(view: View,
savedInstanceState: Bundle?) {
41         super.onViewCreated(view, savedInstanceState)
42
43         binding.progressBar.visibility = View.VISIBLE
44         binding.contentGroup.visibility = View.GONE
45
46         viewModel.loadCountryByName(args.name)
47
48         viewLifecycleOwner.lifecycleScope.launch {
49             viewModel.countryDetail.collect { country ->
50                 binding.progressBar.visibility = View.GONE
51
52                 country?.let {
53                     binding.contentGroup.visibility =
View.VISIBLE
54                     binding.tvDetailName.text = it.nama
55                     binding.tvDetailAlasan.text =
it.alasan
56
57                     binding.imgDetailPhoto.load(it.imageUrl) {
58                         placeholder(R.drawable.ic_launcher_background)
59                         error(R.drawable.ic_launcher_foreground)
60                     }
61                 }
62             }

```


63	}
64	}
65	
66	override fun onDestroyView() {
67	super.onDestroyView()
68	_binding = null
69	}
70	}

3. HomeFragment.kt

Tabel 3. Source Code Jawaban Soal 1 HomeFragment.kt

1	package com.example.dreamstravel
2	
3	import android.content.Intent
4	import android.os.Bundle
5	import android.util.Log
6	import android.view.LayoutInflater
7	import android.view.View
8	import android.view.ViewGroup
9	import androidx.core.net.toUri
10	import androidx.core.view.isVisible
11	import androidx.fragment.app.Fragment
12	import androidx.fragment.app.activityViewModels
13	import androidx.lifecycle.LifecycleScope
14	import androidx.navigation.fragment.findNavController
15	import androidx.recyclerview.widget.LinearLayoutManager
16	import com.example.dreamstravel.databinding.FragmentHomeBinding
17	import com.example.dreamstravel.viewmodel.CountryViewModel
18	import
	com.example.dreamstravel.viewmodel.CountryViewModelFactory
19	import kotlinx.coroutines.launch
20	
21	class HomeFragment : Fragment() {
22	private var _binding: FragmentHomeBinding? = null
23	private val binding get() = _binding!!
24	
25	private val viewModel: CountryViewModel by
	activityViewModels {
26	CountryViewModelFactory(
27	requireActivity().application,
28	(requireActivity().application as
	DreamsTravelApp).repository
29)
30	}

```

31     private lateinit var adapter: CountryAdapter
32
33     override fun onCreateView(inflater: LayoutInflater,
34 container: ViewGroup?, savedInstanceState: Bundle?): View {
35         _binding = FragmentHomeBinding.inflate(inflater,
36 container, false)
37         return binding.root
38     }
39
40     override fun onViewCreated(view: View, savedInstanceState:
41 Bundle?) {
42         super.onViewCreated(view, savedInstanceState)
43         setupRecyclerView()
44
45         lifecycleScope.launch {
46             viewModel.listCountryState.collect { state ->
47                 when (state) {
48                     is UiState.Loading -> {
49                         binding.progressBar.isVisible = true
50                         binding.rvNegara.isVisible = false
51                         binding.tvError.isVisible = false
52                     }
53                     is UiState.Success -> {
54                         binding.progressBar.isVisible = false
55                         binding.rvNegara.isVisible = true
56                         binding.tvError.isVisible = false
57                         adapter.updateData(state.data)
58                     }
59                     is UiState.Error -> {
60                         binding.progressBar.isVisible = false
61                         binding.rvNegara.isVisible = false
62                         binding.tvError.isVisible = true
63                         binding.tvError.text = state.message
64                     }
65                 }
66             }
67         }
68
69         private fun setupRecyclerView() {
70             adapter = CountryAdapter(
71                 countries = emptyList(),
72                 onDetailClick = { country ->
73                     country.nama?.let { name ->
74                         val action =
75                         HomeFragmentDirections.actionHomeFragmentToDetailFragment(name)
76                         findNavController().navigate(action)
77                     }
78                 }
79             )
80         }
81     }
82 }

```

```

74         }
75     },
76     onWikiClick = { country ->
77         val intent = Intent(Intent.ACTION_VIEW,
country.wikipediaUrl.toUri())
78         startActivity(intent)
79     },
80     onFavoriteClick = { country ->
81         viewModel.toggleFavorite(country)
82     }
83 )
84 binding.rvNegara.layoutManager =
LinearLayoutManager(requireContext())
85 binding.rvNegara.adapter = adapter
86 }
87
88 override fun onDestroyView() {
89     super.onDestroyView()
90     _binding = null
91 }
92 }

```

4. CountryAdapter.kt

Tabel 4. Source Code Jawaban Soal 1 CountryAdapter.kt

```

1 package com.example.dreamstravel
2
3 import android.annotation.SuppressLint
4 import android.view.LayoutInflater
5 import android.view.ViewGroup
6 import androidx.recyclerview.widget.RecyclerView
7 import coil.load
8 import com.example.dreamstravel.data.local.CountryEntity
9 import com.example.dreamstravel.databinding.ItemNegaraBinding
10
11 class CountryAdapter(
12     private var countries: List<CountryEntity>,
13     private val onDetailClick: (CountryEntity) -> Unit,
14     private val onWikiClick: (CountryEntity) -> Unit,
15     private val onFavoriteClick: (CountryEntity) -> Unit
16 ) : RecyclerView.Adapter<CountryAdapter.CountryViewHolder>() {
17
18     inner class CountryViewHolder(val binding:
ItemNegaraBinding) : RecyclerView.ViewHolder(binding.root)
19     override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): CountryViewHolder {

```

	<pre> val binding = 20 ItemNegaraBinding.inflate(LayoutInflater.from(parent.context), parent, false) return CountryViewHolder(binding) 21 } 22 23 override fun onBindViewHolder(holder: CountryViewHolder, 24 position: Int) { val country = countries[position] 25 with(holder.binding) { 26 tvItemName.text = country.nama 27 tvAlasan.text = country.alasan 28 29 imgItemPhoto.load(country.imageUrl) { 30 placeholder(R.drawable.ic_launcher_background) 31 error(R.drawable.ic_launcher_foreground) 32 } 33 34 buttonDetail.setOnClickListener { 35 onDetailClick(country) } btnWiki.setOnClickListener { onWikiClick(country) 36 } 37 btnFavorite.setImageResource(38 if (country.isFavorite) R.drawable.ic_favorite 39 else R.drawable.ic_favorite_border 40) 41 btnFavorite.setOnClickListener { 42 onFavoriteClick(country) } 43 } 44 45 override fun getItemCount(): Int = countries.size 46 47 @SuppressWarnings("NotifyDataSetChanged") 48 fun updateData(newList: List<CountryEntity>) { 49 countries = newList 50 notifyDataSetChanged() 51 } 52 } 53 </pre>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5. DreamsTravelApp.kt

Tabel 5. Source Code Jawaban Soal 1 DreamsTravelApp.kt

1	package com.example.dreamstravel
---	----------------------------------

2	
3	import android.app.Application
4	import
	com.example.dreamstravel.data.local.CountryRepository
5	import com.example.dreamstravel.data.local.CountryDatabase
6	
7	class DreamsTravelApp : Application() {
8	val database by lazy {
9	CountryDatabase.getInstance(this) }
	val repository by lazy {
10	CountryRepository(database.countryDao()) }
11	}

6. FavoriteFragment.kt

Tabel 6. Source Code Jawaban Soal 1 FavoriteFragment.kt

1	package com.example.dreamstravel
2	
3	import android.content.Intent
4	import android.os.Bundle
5	import android.util.Log
6	import android.view.LayoutInflater
7	import android.view.View
8	import android.view.ViewGroup
9	import androidx.core.net.toUri
10	import androidx.core.view.isVisible
11	import androidx.fragment.app.Fragment
12	import androidx.fragment.app.activityViewModels
13	import androidx.lifecycle.LifecycleScope
14	import androidx.navigation.fragment.findNavController
15	import androidx.recyclerview.widget.LinearLayoutManager
16	import
	com.example.dreamstravel.databinding.FragmentFavoriteBinding
17	import com.example.dreamstravel.viewmodel.CountryViewModel
18	import com.example.dreamstravel.viewmodel.CountryViewModelFactory
19	import kotlinx.coroutines.launch
20	
21	class FavoriteFragment : Fragment() {
22	private var _binding: FragmentFavoriteBinding? = null
23	private val binding get() = _binding!!
24	
25	private val viewModel: CountryViewModel by activityViewModels
26	{
27	CountryViewModelFactory(
28	requireActivity().application,

29	(requireActivity().application	as
	DreamsTravelApp).repository	
30)	
31	}	
32	private lateinit var adapter: CountryAdapter	
33		
	override fun onCreateView(inflater: LayoutInflater,	
34	container: ViewGroup?, savedInstanceState: Bundle?): View {	
	_binding = FragmentFavoriteBinding.inflate(inflater,	
35	container, false)	
36	return binding.root	
37	}	
38		
39	override fun onViewCreated(view: View, savedInstanceState:	
40	Bundle?) {	
	super.onViewCreated(view, savedInstanceState)	
41	setupRecyclerView()	
42		
43	viewModel.loadFavoriteCountries()	
44		
45	lifecycleScope.launch {	
46	viewModel.favoriteListState.collect { state ->	
47	when (state) {	
48	is UiState.Loading -> {	
49	binding.progressBar.isVisible = true	
50	binding.rvNegara.isVisible = false	
51	binding.tvInfo.isVisible = false	
52	}	
53	is UiState.Success -> {	
54	binding.progressBar.isVisible = false	
55	if (state.data.isEmpty()) {	
56	binding.rvNegara.isVisible = false	
57	binding.tvInfo.isVisible = true	
58	binding.tvInfo.text = "Belum ada	
59	negara favorit"	
	} else {	
60	binding.rvNegara.isVisible = true	
61	binding.tvInfo.isVisible = false	
62	adapter.updateData(state.data)	
63	}	
64	}	
65	is UiState.Error -> {	
66	binding.progressBar.isVisible = false	
67	binding.rvNegara.isVisible = false	
68	binding.tvInfo.isVisible = true	
69	binding.tvInfo.text = state.message	
70	}	

```

71         }
72     }
73 }
74 }
75
76 private fun setupRecyclerView() {
77     adapter = CountryAdapter(
78         countries = emptyList(),
79         onDetailClick = { country ->
80
81             country.nama?.let { name ->
82                 val action = FavoriteFragmentDirections.actionFavoriteFragmentToDetailFragment(
83                     name)
84                 findNavController().navigate(action)
85             },
86             onWikiClick = { country ->
87                 val intent = Intent(Intent.ACTION_VIEW,
88                     country.wikiUrl.toUri())
89                 startActivity(intent)
90             },
91             onFavoriteClick = { country ->
92                 viewModel.toggleFavorite(country)
93             }
94         )
95     binding.rvNegara.layoutManager =
96     LinearLayoutManager(requireContext())
97     binding.rvNegara.adapter = adapter
98 }
99
100 override fun onDestroyView() {
101     super.onDestroyView()
102     _binding = null
103 }
104 }

```

7. UiState.kt

Tabel 7. Source Code Jawaban Soal 1 UiState.kt

```

1 package com.example.dreamstravel
2
3 sealed class UiState<out T> {
4     object Loading : UiState<Nothing>()
5
6     data class Success<out T>(val data: T) : UiState<T>()

```

7	
8	data class Error(val message: String) :
9	UiState<Nothing>() }

8. viewmodel/CountryViewModel.kt

Tabel 8. Source Code Jawaban Soal 1 CountryViewModel.kt

1	package com.example.dreamstravel.viewmodel
2	
3	import android.app.Application
4	import android.util.Log
5	import androidx.lifecycle.AndroidViewModel
6	import androidx.lifecycle.ViewModelScope
7	import com.example.dreamstravel.UiState
8	import com.example.dreamstravel.data.local.CountryEntity
9	import com.example.dreamstravel.data.local.CountryRepository
10	import kotlinx.coroutines.flow.MutableStateFlow
11	import kotlinx.coroutines.flow.StateFlow
12	import kotlinx.coroutines.launch
13	import java.io.IOException
14	
15	class CountryViewModel(16 application: Application, 17 private val repository: CountryRepository 18) : AndroidViewModel(application) { 19 20 private val _listCountryState = MutableStateFlow<UiState<List<CountryEntity>>>>(UiState.Loading) 21 val listCountryState: StateFlow<UiState<List<CountryEntity>>>> get() = _listCountryState 22 23 private val _favoriteListState = MutableStateFlow<UiState<List<CountryEntity>>>>(UiState.Loading) 24 val favoriteListState: StateFlow<UiState<List<CountryEntity>>>> get() = _favoriteListState 25 26 private val _countryDetail = MutableStateFlow<CountryEntity?>(null) 27 val countryDetail: StateFlow<CountryEntity?> get() = _countryDetail 28 29 init { 30 loadCountries()


```

31     }
32
33     fun loadCountries() {
34         _listCountryState.value = UiState.Loading
35         viewModelScope.launch {
36             try {
37
38                 val remoteData =
39 repository.fetchCountriesFromApi()
40                 repository.insertAll(remoteData) //
41 Simpan/Update ke database
42
43                 repository.getAllCountries().collect {
44 countries ->
45                 _listCountryState.value =
46 UiState.Success(countries)
47             }
48             } catch (e: IOException) {
49                 _listCountryState.value = UiState.Error("Gagal
50 terhubung ke server. Periksa koneksi internet.")
51             } catch (e: Exception) {
52                 _listCountryState.value =
53 UiState.Error("Terjadi kesalahan: ${e.message}")
54             }
55         }
56     }
57
58     fun loadFavoriteCountries() {
59         _favoriteListState.value = UiState.Loading
60         viewModelScope.launch {
61             try {
62                 repository.getFavoriteCountries().collect {
63 favorites ->
64                 _favoriteListState.value =
65 UiState.Success(favorites)
66             }
67             } catch (e: Exception) {
68                 _favoriteListState.value = UiState.Error("Gagal
69 memuat daftar favorit: ${e.message}")
70             }
71         }
72     }
73
74     fun loadCountryByName(countryName: String) {
75         viewModelScope.launch {
76             repository.getCountryByName(countryName).collect {
77 country ->
78                 _countryDetail.value = country

```

```

68         }
69     }
70 }
71
72 fun toggleFavorite(country: CountryEntity) {
73     viewModelScope.launch {
74         val updated = country.copy(isFavorite =
!country.isFavorite)
75         repository.updateCountry(updated)
76
77         if (_favoriteListState.value is UiState.Success) {
78             loadFavoriteCountries()
79         }
80     }
81 }
82 }

```

9. viewmodel/CountryViewModelFactory.kt

Tabel 9. Source Code Jawaban Soal 1 viewmodel/CountryViewModelFactory.kt

```

1 package com.example.dreamstravel.viewmodel
2
3 import android.app.Application
4 import androidx.lifecycle.ViewModel
5 import androidx.lifecycle.ViewModelProvider
6 import
com.example.dreamstravel.data.local.CountryRepository
7
8 class CountryViewModelFactory(
9     private val application: Application,
10    private val repository: CountryRepository
11 ) : ViewModelProvider.Factory {
12
13     override fun <T : ViewModel> create(modelClass:
Class<T>): T {
14         if
(modelClass.isAssignableFrom(CountryViewModel::class.java))
15         {
16             @Suppress("UNCHECKED_CAST")
17             return CountryViewModel(application,
repository) as T
18         }
19         throw IllegalArgumentException("Unknown ViewModel
class")

```

20	}
----	---

10. data/local/AppDataBase.kt

Tabel 10. Source Code Jawaban Soal 1 data/local/AppDataBase.kt

1	package com.example.dreamstravel.data.local
2	
3	import android.content.Context
4	import androidx.room.Database
5	import androidx.room.Room
6	import androidx.room.RoomDatabase
7	
8	@Database(entities = [CountryEntity::class], version = 2, exportSchema = false)
9	abstract class CountryDatabase : RoomDatabase() {
10	
11	abstract fun countryDao(): CountryDao
12	
13	companion object {
14	@Volatile
15	private var INSTANCE: CountryDatabase? = null
16	
17	fun getInstance(context: Context): CountryDatabase
18	{
19	return INSTANCE ?: synchronized(this) {
20	context.applicationContext,
21	CountryDatabase::class.java,
22	"country_database"
23).fallbackToDestructiveMigration(false).build()
24	INSTANCE = instance
25	instance
26	}
27	}
28	}
29	}

11. data/local/CountryDao.kt

Tabel 11. Source Code Jawaban Soal 1 data/local/CountryDao.kt

1	package com.example.dreamstravel.data.local
2	
3	import androidx.room.Dao

```

4 import androidx.room.Insert
5 import androidx.room.OnConflictStrategy
6 import androidx.room.Query
7 import kotlinx.coroutines.flow.Flow
8 import androidx.room.Update
9
10 @Dao
11 interface CountryDao {
12
13     @Query("SELECT * FROM country")
14     fun getAllCountries(): Flow<List<CountryEntity>>
15
16     @Query("SELECT * FROM country WHERE nama = :name LIMIT
17 1")
18     fun getCountryByName(name: String):
19     Flow<CountryEntity?>
20
21     @Insert(onConflict = OnConflictStrategy.REPLACE)
22     suspend fun insertAll(countries: List<CountryEntity>)
23
24     @Query("DELETE FROM country")
25     suspend fun deleteAll()
26
27     @Query("SELECT * FROM country WHERE isFavorite = 1 ORDER
28 BY nama ASC")
29     fun getFavoriteCountries(): Flow<List<CountryEntity>>
30
31     @Update
32     suspend fun updateCountry(country: CountryEntity)
33 }

```

12. data/local/CountryEntity.kt

Tabel 12. Source Code Jawaban Soal 1 data/local/CountryEntity.kt

```

1 package com.example.dreamstravel.data.local
2 import androidx.room.Entity
3 import androidx.room.PrimaryKey
4
5 @Entity(tableName = "country")
6 data class CountryEntity(
7     @PrimaryKey(autoGenerate = true)
8     val id: Int = 0,
9     val nama: String?,
10    val alasan: String,
11    val imageUrl: String?,
12    val wikiUrl: String,

```

13	val isFavorite: Boolean = false
14)

13. data/local/CountryRepository.kt

Tabel 13. Source Code Jawaban Soal 1 data/local/CountryRepository.kt

1	package com.example.dreamstravel.data.local
2	
3	import
	com.example.dreamstravel.data.remote.CountryResponse
4	import com.example.dreamstravel.data.remote.RetrofitClient
5	import kotlinx.coroutines.flow.Flow
6	
7	class CountryRepository(private val countryDao:
	CountryDao) {
8	
9	fun getAllCountries(): Flow<List<CountryEntity>> {
10	return countryDao.getAllCountries()
11	}
12	
13	fun getCountryByName(name: String):
	Flow<CountryEntity?> { // Pastikan return type bisa null
14	return countryDao.getCountryByName(name)
15	}
16	
17	suspend fun insertAll(countries: List<CountryEntity>)
	{
18	countryDao.insertAll(countries)
19	}
20	
21	suspend fun updateCountry(country: CountryEntity) {
22	countryDao.updateCountry(country)
23	}
24	
25	fun getFavoriteCountries(): Flow<List<CountryEntity>>
	{
26	return countryDao.getFavoriteCountries()
27	}
28	
29	suspend fun fetchCountriesFromApi():
	List<CountryEntity> {
30	val response: List<CountryResponse> =
	RetrofitClient.apiService.getAllCountries()
31	
32	return response.mapNotNull { countryResponse ->
33	val countryName = countryResponse.name?.common

34	val flagUrl = countryResponse.flags?.png
35	val region = countryResponse.region
36	
37	if (countryName.isNullOrEmpty()
	flagUrl.isNullOrEmpty()) {
38	null
39	} else {
40	CountryEntity(
41	nama = countryName,
42	tahun = "2025",
43	alasan = region ?: "No region",
44	imageUrl = flagUrl,
45	wikiUrl =
	"https://en.wikipedia.org/wiki/\${countryName.replace(" ",
	"_")}"
46)
47	}
48	}
49	}
50	}

14. data/remote/ApiService.kt

Tabel 14. Source Code Jawaban Soal 1 data/remote/ApiService.kt

1	package com.example.dreamstravel.data.remote
2	
3	import retrofit2.http.GET
4	import retrofit2.http.Query
5	
6	interface ApiService {
7	@GET("all")
8	suspend fun getAllCountries(
9	@Query("fields") fields: String =
	"name,flags,region"
10): List<CountryResponse>
11	}

15. data/remote/CountryResponse.kt

Tabel 15. Source Code Jawaban Soal 1 data/remote/CountryResponse.kt

1	package com.example.dreamstravel.data.remote
2	import kotlinx.serialization.Serializable
3	
4	@Serializable
5	data class CountryResponse(
6	val name: Name?,

7	val flags: Flags?,
8	val region: String?
9)
10	
11	@Serializable
12	data class Name(val common: String?)
13	
14	@Serializable
15	data class Flags(val png: String?)

16. data/remote/RetrofitClient.kt

Tabel 16. Source Code Jawaban Soal 1 data/remote/RetrofitClient.kt

1	package com.example.dreamstravel.data.remote
2	
3	import
	com.jakewharton.retrofit2.converter.kotlinx.serialization.asConverterFactory
4	import kotlinx.serialization.ExperimentalSerializationApi
5	import kotlinx.serialization.json.Json
6	import okhttp3.MediaType.Companion.toMediaType
7	import okhttp3.OkHttpClient
8	import okhttp3.logging.HttpLoggingInterceptor
9	import retrofit2.Retrofit
10	
11	@OptIn(ExperimentalSerializationApi::class)
12	object RetrofitClient {
13	private val json = Json { ignoreUnknownKeys = true }
14	
15	private val loggingInterceptor = HttpLoggingInterceptor().apply {
16	level = HttpLoggingInterceptor.Level.BODY // BODY akan menampilkan semua detail
17	}
18	
19	private val client = OkHttpClient.Builder()
20	.addInterceptor(loggingInterceptor)
21	.build()
22	
23	val apiService: ApiService by lazy {
24	Retrofit.Builder()
25	.baseUrl("https://restcountries.com/v3.1/")
26	.client(client)
27	
	.addConverterFactory(json.asConverterFactory("application/json".toMed
28	.build())
29	.create(ApiService::class.java)
30	}

31	}
----	---

17. layout/activity_main.xml

Tabel 17. Source Code Jawaban Soal 1 layout/activity_main.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
3	xmlns:android="http://schemas.android.com/apk/res/android"
4	xmlns:app="http://schemas.android.com/apk/res-auto"
5	xmlns:tools="http://schemas.android.com/tools"
6	android:layout_width="match_parent"
7	android:layout_height="match_parent"
8	tools:context=".MainActivity">
9	<androidx.fragment.app.FragmentContainerView
10	android:id="@+id/nav_host_fragment"
11	
12	android:name="androidx.navigation.fragment.NavHostFragment"
13	android:layout_width="0dp"
14	android:layout_height="0dp"
15	app:defaultNavHost="true"
16	app:layout_constraintBottom_toTopOf="@id/bottom_navigation"
17	app:layout_constraintEnd_toEndOf="parent"
18	app:layout_constraintStart_toStartOf="parent"
19	app:layout_constraintTop_toTopOf="parent"
20	app:navGraph="@navigation/nav_graph" />
21	
22	<com.google.android.material.bottomnavigation.BottomNavigationView
23	android:id="@+id/bottom_navigation"
24	android:layout_width="0dp"
25	android:layout_height="wrap_content"
26	app:layout_constraintBottom_toBottomOf="parent"
27	app:layout_constraintEnd_toEndOf="parent"
28	app:layout_constraintStart_toStartOf="parent"
29	app:menu="@menu/bottom_nav_menu"
30	app:itemIconTint="@color/bottom_nav_color_selector"
31	app:itemTextColor="@color/bottom_nav_color_selector" />
32	</androidx.constraintlayout.widget.ConstraintLayout>

18. layout/fragment_detail.xml

Tabel 18. Source Code Jawaban Soal 1 layout/fragment_detail.xml

1	<?xml version="1.0" encoding="utf-8"?>
---	----------------------------------------


```

2 <ScrollView
3     xmlns:android="http://schemas.android.com/apk/res/android"
4         xmlns:app="http://schemas.android.com/apk/res-auto"
5         xmlns:tools="http://schemas.android.com/tools"
6         android:layout_width="match_parent"
7         android:layout_height="match_parent"
8         android:fillViewport="true">
9
10     <androidx.constraintlayout.widget.ConstraintLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         tools:context=".DetailFragment">
14
15         <ProgressBar
16             android:id="@+id/progressBar"
17             style="?android:attr/progressBarStyle"
18             android:layout_width="wrap_content"
19             android:layout_height="wrap_content"
20             android:visibility="gone"
21             app:layout_constraintBottom_toBottomOf="parent"
22             app:layout_constraintEnd_toEndOf="parent"
23             app:layout_constraintStart_toStartOf="parent"
24             app:layout_constraintTop_toTopOf="parent"
25             tools:visibility="visible" />
26
27         <androidx.constraintlayout.widget.Group
28             android:id="@+id/content_group"
29             android:layout_width="wrap_content"
30             android:layout_height="wrap_content"
31
32     app:constraint_referenced_ids="imgDetailPhoto,tvDetailName,
33     tvDetailAlasan" />
34
35     <ImageView
36         android:id="@+id/imgDetailPhoto"
37         android:layout_width="200dp"
38         android:layout_height="200dp"
39         android:layout_marginTop="32dp"
40         android:contentDescription="@string/img_desc"
41         app:layout_constraintEnd_toEndOf="parent"
42         app:layout_constraintStart_toStartOf="parent"
43         app:layout_constraintTop_toTopOf="parent"
44         tools:src="@tools:sample/avatars" />
45
46     <TextView
47         android:id="@+id/tvDetailName"
48         android:layout_width="0dp"

```

```

47         android:layout_height="wrap_content"
48         android:layout_marginStart="16dp"
49         android:layout_marginTop="16dp"
50         android:layout_marginEnd="16dp"
51         android:textAlignment="center"
52         android:textSize="24sp"
53         android:textStyle="bold"
54         app:layout_constraintEnd_toEndOf="parent"
55         app:layout_constraintStart_toStartOf="parent"
56
57     app:layout_constraintTop_toBottomOf="@id/imgDetailPhoto"
58         tools:text="Country Name" />
59
60     <TextView
61         android:id="@+id/tvDetailAlasan"
62         android:layout_width="0dp"
63         android:layout_height="wrap_content"
64         android:layout_margin="16dp"
65         android:textSize="16sp"
66         app:layout_constraintEnd_toEndOf="parent"
67         app:layout_constraintStart_toStartOf="parent"
68
69     app:layout_constraintTop_toBottomOf="@id/tvDetailName"
70         tools:text="Reason text goes here..." />
71
72 </androidx.constraintlayout.widget.ConstraintLayout>
73 </ScrollView>

```

19. layout/fragment_favorite.xml

Tabel 19. Source Code Jawaban Soal 1 layout/fragment_favorite.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".FavoriteFragment">
9
10    <androidx.recyclerview.widget.RecyclerView
11        android:id="@+id/rvNegara"
12        android:layout_width="0dp"
13        android:layout_height="0dp"
14        android:clipToPadding="false"
15        android:padding="8dp"

```

15	app:layout_constraintBottom_toBottomOf="parent"
16	app:layout_constraintEnd_toEndOf="parent"
17	app:layout_constraintStart_toStartOf="parent"
18	app:layout_constraintTop_toTopOf="parent"
19	tools:listitem="@layout/item_negara" />
20	
21	<ProgressBar
22	android:id="@+id/progressBar"
23	style="?android:attr/progressBarStyle"
24	android:layout_width="wrap_content"
25	android:layout_height="wrap_content"
26	android:visibility="gone"
27	app:layout_constraintBottom_toBottomOf="parent"
28	app:layout_constraintEnd_toEndOf="parent"
29	app:layout_constraintStart_toStartOf="parent"
30	app:layout_constraintTop_toTopOf="parent"
31	tools:visibility="visible" />
32	
33	<TextView
34	android:id="@+id/tv_info"
35	android:layout_width="wrap_content"
36	android:layout_height="wrap_content"
37	android:text="Belum ada negara favorit"
38	android:visibility="gone"
39	app:layout_constraintBottom_toBottomOf="parent"
40	app:layout_constraintEnd_toEndOf="parent"
41	app:layout_constraintStart_toStartOf="parent"
42	app:layout_constraintTop_toTopOf="parent"
43	tools:visibility="visible" />
44	
45	</androidx.constraintlayout.widget.ConstraintLayout>

20. layout/fragment_home.xml

Tabel 20. Source Code Jawaban Soal 1 layout/fragment_home.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<androidx.constraintlayout.widget.ConstraintLayout
	xmlns:android="http://schemas.android.com/apk/res/android"
3	xmlns:app="http://schemas.android.com/apk/res-auto"
4	android:layout_width="match_parent"
5	android:layout_height="match_parent"
6	xmlns:tools="http://schemas.android.com/tools">
7	
8	<androidx.recyclerview.widget.RecyclerView
9	android:id="@+id/rvNegara"
10	android:layout_width="0dp"

11	android:layout_height="0dp"
12	android:clipToPadding="false"
13	android:padding="8dp"
14	android:contentDescription="cute"
15	app:layout_constraintTop_toTopOf="parent"
16	app:layout_constraintBottom_toBottomOf="parent"
17	app:layout_constraintStart_toStartOf="parent"
18	app:layout_constraintEnd_toEndOf="parent"
19	tools:listitem="@layout/item_negara"
20	tools:ignore="HardcodedText" />
21	
22	<ProgressBar
23	android:id="@+id/progressBar"
24	style="?android:attr/progressBarStyle"
25	android:layout_width="wrap_content"
26	android:layout_height="wrap_content"
27	android:visibility="gone"
28	app:layout_constraintBottom_toBottomOf="parent"
29	app:layout_constraintEnd_toEndOf="parent"
30	app:layout_constraintStart_toStartOf="parent"
31	app:layout_constraintTop_toTopOf="parent"
32	tools:visibility="visible" />
33	
34	<TextView
35	android:id="@+id/tv_error"
36	android:layout_width="wrap_content"
37	android:layout_height="wrap_content"
38	android:text="Error Message"
39	android:visibility="gone"
40	app:layout_constraintBottom_toBottomOf="parent"
41	app:layout_constraintEnd_toEndOf="parent"
42	app:layout_constraintStart_toStartOf="parent"
43	app:layout_constraintTop_toTopOf="parent"
44	tools:visibility="visible" />
45	</androidx.constraintlayout.widget.ConstraintLayout>

21. item_negara.xml

Tabel 21. Source Code Jawaban Soal 1 item_negara.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	xmlns:tools="http://schemas.android.com/tools"
6	android:layout_width="match_parent"
	android:layout_height="wrap_content"

```

7      android:layout_margin="8dp"
8      app:cardCornerRadius="12dp"
9      app:cardElevation="4dp">
10
11      <androidx.constraintlayout.widget.ConstraintLayout
12          android:layout_width="match_parent"
13          android:layout_height="wrap_content"
14          android:padding="12dp">
15
16          <com.google.android.material.imageview.ShapeableImageView
17              android:id="@+id/img_item_photo"
18              android:layout_width="100dp"
19              android:layout_height="100dp"
20              android:scaleType="centerCrop"
21              android:contentDescription="@string/img_desc"
22              app:layout_constraintStart_toStartOf="parent"
23              app:layout_constraintTop_toTopOf="parent"
24              app:layout_constraintBottom_toBottomOf="parent"
25          />
26
27          <TextView
28              android:id="@+id/tv_item_name"
29              android:layout_width="0dp"
30              android:layout_height="wrap_content"
31              android:textSize="18sp"
32              android:textStyle="bold"
33              android:textColor="@android:color/black"
34              android:layout_marginStart="8dp"
35
36          app:layout_constraintStart_toEndOf="@id/img_item_photo"
37              app:layout_constraintTop_toTopOf="parent"
38              app:layout_constraintEnd_toEndOf="parent"
39              tools:text="Jepang" />
40
41          <TextView
42              android:id="@+id/tv_alasan"
43              android:layout_width="0dp"
44              android:layout_height="wrap_content"
45              android:layout_marginTop="4dp"
46              android:textColor="@android:color/darker_gray"
47              android:textSize="14sp"
48
49          app:layout_constraintEnd_toEndOf="@id/tv_item_name"
50
51          app:layout_constraintStart_toStartOf="@id/tv_item_name"

```

```

49 app:layout_constraintTop_toBottomOf="@id/tv_item_name"
50     tools:ignore="TextContrastCheck"
51     tools:text="Asia Timur" />
52
53     <Button
54         android:id="@+id/button_detail"
55         android:layout_width="wrap_content"
56         android:layout_height="wrap_content"
57         android:text="@string/btn_detail"
58         android:layout_marginTop="8dp"
59 app:layout_constraintTop_toBottomOf="@id/tv_alasan"
60 app:layout_constraintStart_toStartOf="@id/tv_item_name"
61     app:layout_constraintBottom_toBottomOf="parent"
62 />
63
64     <Button
65         android:id="@+id/btn_wiki"
66         android:layout_width="wrap_content"
67         android:layout_height="wrap_content"
68         android:text="@string/btn_wiki"
69         android:layout_marginStart="8dp"
70         android:layout_marginTop="8dp"
71 app:layout_constraintTop_toBottomOf="@id/tv_alasan"
72 app:layout_constraintStart_toEndOf="@id/button_detail"
73     app:layout_constraintBottom_toBottomOf="parent"
74 />
75
76     <ImageButton
77         android:id="@+id/btn_favorite"
78         android:layout_width="48dp"
79         android:layout_height="48dp"
80 android:background="?attr/selectableItemBackgroundBorderless"
81         android:contentDescription="@string/favorite"
82         android:tint="@android:color/holo_red_dark"
83         android:padding="8dp"
84         android:src="@drawable/ic_favorite_border"
85         app:layout_constraintEnd_toEndOf="parent"
86         app:layout_constraintTop_toTopOf="parent"
87         tools:ignore="UseAppTint" />
88
89 </androidx.constraintlayout.widget.ConstraintLayout>

```

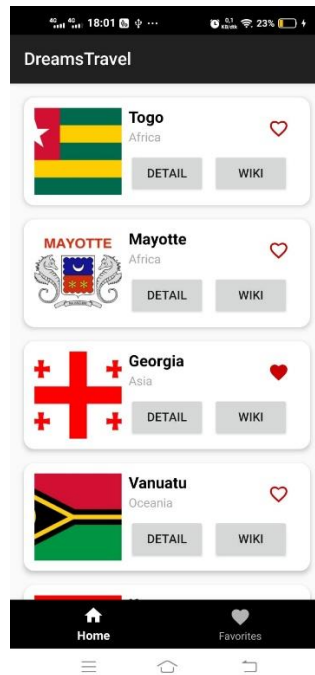
88	</androidx.cardview.widget.CardView>
----	--------------------------------------

22. navigation/nav_graph.xml

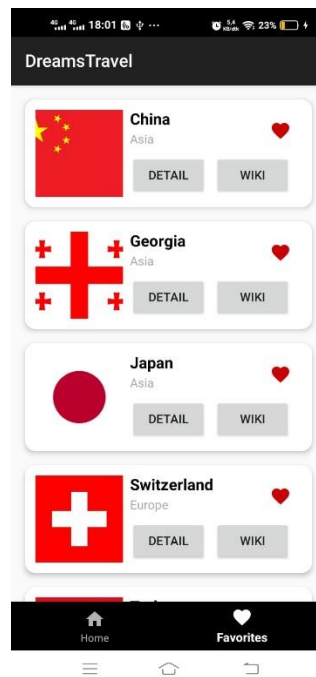
Tabel 22. Source Code Jawaban Soal 1 navigation/nav_graph.xml

1	<?xml version="1.0" encoding="utf-8"?>
2	<navigation
	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:id="@+id/nav_graph"
6	app:startDestination="@id/homeFragment">
7	
8	<fragment
9	android:id="@+id/homeFragment"
10	
	android:name="com.example.dreamstravel.HomeFragment"
11	android:label="Home"
12	tools:layout="@layout/fragment_home">
13	<action
14	
	android:id="@+id/action_homeFragment_to_detailFragment"
15	app:destination="@id/detailFragment" />
16	</fragment>
17	<fragment
18	android:id="@+id/favoriteFragment"
19	
	android:name="com.example.dreamstravel.FavoriteFragment"
20	android:label="Favorites"
21	tools:layout="@layout/fragment_favorite">
22	<action
23	
	android:id="@+id/action_favoriteFragment_to_detailFragment"
24	app:destination="@id/detailFragment" />
25	</fragment>
26	<fragment
27	android:id="@+id/detailFragment"
28	
	android:name="com.example.dreamstravel.DetailFragment"
29	android:label="Detail"
30	tools:layout="@layout/fragment_detail">
31	<argument
32	android:name="name"
33	app:argType="string" />
34	</fragment>
35	</navigation>

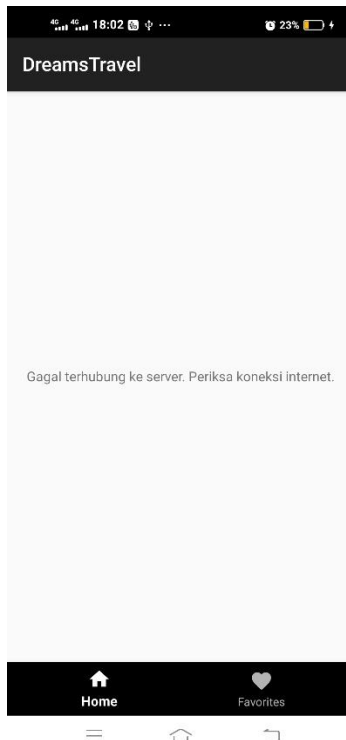
B. Output Program



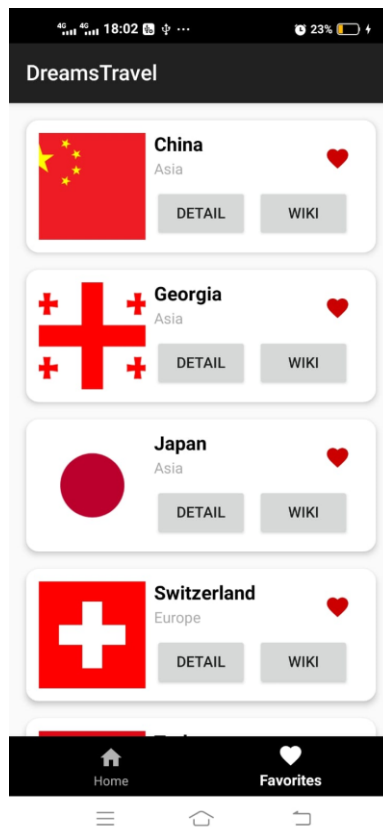
Gambar 1. Tampilan Home



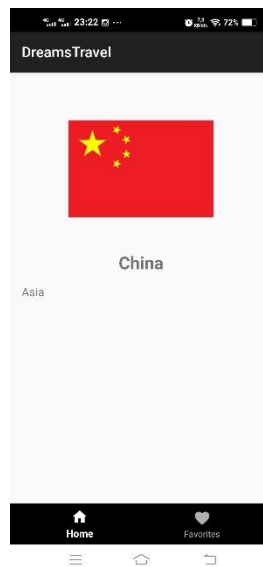
Gambar 2. Tampilan Favorites



Gambar 3. Tampilan Home saat tidak ada jaringan



Gambar 4. Tampilan Favorites saat tidak ada jaringan, masih bisa diakses



Gambar 5. Tampilan Detail



Gambar 6. Tampilan Wiki

C. Pembahasan

1. MainActivity.kt:

Kode di atas adalah bagian dari aplikasi Android yang menggunakan bahasa Kotlin. Kelas `MainActivity` mewarisi `AppCompatActivity` dan berfungsi sebagai aktivitas utama aplikasi. Di dalam metode `onCreate`, layout utama diatur menggunakan `View Binding` (`ActivityMainBinding`). Kemudian, aplikasi mengambil `NavHostFragment` dari layout yang berfungsi sebagai wadah navigasi antar fragment. Setelah mendapatkan `NavController` dari fragment tersebut, `bottomNavigation` dihubungkan dengan sistem navigasi menggunakan `setupWithNavController`, sehingga menu navigasi bawah dapat digunakan untuk berpindah antar halaman (fragment) dalam aplikasi.

2. DetailFragment.kt

Kode di atas adalah `DetailFragment`, yaitu salah satu tampilan halaman dalam aplikasi Android yang menampilkan detail sebuah negara. Fragment ini menggunakan `View Binding` (`FragmentDetailBinding`) untuk mengakses elemen-elemen UI. Data negara yang akan ditampilkan diterima melalui args (menggunakan `navArgs`). `ViewModel` (`CountryViewModel`) digunakan untuk mengelola dan mengambil data, yang dihubungkan ke Activity dengan bantuan `CountryViewModelFactory`. Saat tampilan dibuat, fragment menampilkan `ProgressBar` sambil mengambil data berdasarkan nama negara. Setelah data berhasil dimuat, progress bar disembunyikan, dan data seperti nama, alasan, serta gambar negara ditampilkan di layar. Gambar dimuat menggunakan library `Coil`. `View Binding` dihapus saat tampilan dihancurkan untuk menghindari memory leak.

3. HomeFragment.kt

Kode di atas adalah `HomeFragment`, yaitu tampilan utama aplikasi yang menampilkan daftar negara menggunakan `RecyclerView`. Fragment ini memanfaatkan `View Binding` (`FragmentHomeBinding`) dan `CountryViewModel` untuk mengambil data. Saat tampilan dibuat, fragment mempersiapkan `RecyclerView` dan memantau perubahan data dari `ViewModel` menggunakan `collect`. Jika data sedang dimuat, maka akan ditampilkan `ProgressBar`; jika berhasil, daftar negara akan

muncul; dan jika gagal, akan tampil pesan error. Adapter memiliki tiga aksi: melihat detail negara (navigasi ke DetailFragment), membuka Wikipedia negara, dan menandai sebagai favorit. Binding dibersihkan saat tampilan dihancurkan untuk mencegah kebocoran memori.

4. CountryAdapter.kt

Kode di atas adalah `CountryAdapter`, yaitu kelas adaptor untuk `RecyclerView` yang menampilkan daftar negara dalam aplikasi. Setiap item dalam daftar menggunakan layout `item_negara.xml` yang diakses melalui View Binding. Adaptor ini menerima daftar negara (`countries`) dan tiga fungsi callback: untuk melihat detail, membuka Wikipedia, dan menandai favorit. Di dalam `onBindViewHolder`, data negara seperti nama, alasan, dan gambar ditampilkan, lalu tombol-tombol diberi aksi yang sesuai. Gambar dimuat menggunakan library `Coil`, dan ikon favorit berubah tergantung statusnya. Metode `updateData` digunakan untuk memperbarui data daftar negara dan menyegarkan tampilan.

5. DreamsTravelApp.kt

Kode di atas adalah kelas `DreamsTravelApp` yang merupakan turunan dari `Application` dan berfungsi sebagai titik awal global untuk mengelola data di aplikasi. Di dalamnya, objek `database` dan `repository` dibuat secara **lazy** (hanya dibuat saat dibutuhkan). `database` adalah instance dari `CountryDatabase`, dan `repository` adalah `CountryRepository` yang mengambil data dari DAO (`countryDao`). Dengan cara ini, database dan repository bisa diakses dari bagian lain aplikasi secara efisien dan terpusat.

6. FavoriteFragment.kt

Kode di atas adalah `FavoriteFragment`, yaitu tampilan yang menampilkan daftar negara favorit pengguna dalam aplikasi. Fragment ini menggunakan View Binding (`FragmentFavoriteBinding`) dan `CountryViewModel` untuk mengambil data dari database. Saat tampilan dibuat, fragment memanggil `loadFavoriteCountries()` untuk memuat data favorit, lalu mengamati perubahan datanya menggunakan `collect`. Jika sedang memuat, akan tampil `ProgressBar`; jika sukses dan data kosong, akan muncul pesan bahwa belum ada negara favorit; jika sukses dan data ada, daftar negara ditampilkan; jika gagal, akan muncul pesan error. `RecyclerView` menampilkan

daftar negara favorit, dengan tiga aksi: lihat detail, buka Wikipedia, dan hapus dari favorit. Binding dibersihkan saat tampilan dihancurkan untuk mencegah memory leak.

7. UiState.kt

Kode di atas adalah class ``UiState``, sebuah **`**sealed class**`** yang digunakan untuk merepresentasikan tiga kondisi umum dalam pengambilan data di aplikasi, yaitu: ``Loading``, ``Success``, dan ``Error``. ``Loading`` menandakan data sedang dimuat, ``Success`` menampung data yang berhasil diambil, dan ``Error`` menyimpan pesan kesalahan jika terjadi kegagalan. Dengan menggunakan ``UiState``, alur logika aplikasi jadi lebih terstruktur dan mudah dikelola saat menampilkan status data ke pengguna.

8. viewmodel/CountryViewModel.kt

Kode di atas adalah ``CountryViewModel``, kelas yang berfungsi sebagai jembatan antara UI dan data dalam arsitektur MVVM. ViewModel ini mengelola data negara yang diambil dari ``CountryRepository``. Terdapat tiga ``StateFlow`` untuk memantau status: ``listCountryState`` untuk semua negara, ``favoriteListState`` untuk negara favorit, dan ``countryDetail`` untuk detail satu negara. Saat ViewModel dibuat (``init``), data negara dimuat dari API dan disimpan ke database lokal. Fungsi ``loadCountries()``, ``loadFavoriteCountries()``, dan ``loadCountryByName()`` digunakan untuk mengambil data sesuai kebutuhan. Fungsi ``toggleFavorite()`` digunakan untuk menandai atau menghapus negara dari favorit, lalu memperbarui daftar favorit jika perlu. Semua proses berjalan secara asinkron dengan ``viewModelScope`` menggunakan coroutine.

9. viewmodel/CountryViewModelFactory.kt

Kode di atas adalah ``CountryViewModelFactory``, sebuah kelas yang digunakan untuk membuat instance dari ``CountryViewModel`` dengan parameter tambahan, yaitu ``Application`` dan ``CountryRepository``. Karena ``ViewModel`` bawaan Android tidak mendukung konstruktor dengan parameter, maka dibutuhkan ``ViewModelProvider.Factory`` untuk menyediakannya. Dalam metode ``create``, jika ``modelClass`` cocok dengan ``CountryViewModel``, maka objek tersebut dibuat dan dikembalikan. Jika tidak cocok, maka akan dilemparkan exception. Factory ini

memungkinkan ViewModel digunakan dengan dependensi yang dibutuhkan secara benar dalam arsitektur MVVM.

10. data/local/AppDataBase.kt

Kode di atas adalah `CountryDatabase`, yaitu kelas abstrak yang mewakili database lokal menggunakan Room di Android. Kelas ini mendefinisikan satu entitas `CountryEntity` dan satu DAO `countryDao()` untuk mengakses data. Di dalamnya terdapat `companion object` untuk membuat database sebagai singleton, sehingga hanya ada satu instance database yang aktif selama aplikasi berjalan. Fungsi `getInstance()` memastikan database dibuat sekali dan bisa diakses dari berbagai tempat. `fallbackToDestructiveMigration(false)` berarti jika versi database berubah dan tidak ada migrasi yang disediakan, maka akan terjadi error (tidak otomatis hapus data).

11. data/local/CountryDao.kt

Kode di atas adalah `CountryDao`, yaitu interface yang berisi perintah-perintah untuk mengakses database Room secara langsung. DAO ini mengelola data dari tabel `country`. Fungsi `getAllCountries()` mengambil semua data negara secara real-time menggunakan `Flow`. `getCountryByName(name)` mengambil satu data negara berdasarkan nama. `insertAll()` digunakan untuk menyimpan daftar negara ke database, dan akan mengganti data lama jika ada konflik. `deleteAll()` menghapus semua data negara. `getFavoriteCountries()` mengambil daftar negara yang ditandai sebagai favorit. Terakhir, `updateCountry()` memperbarui data suatu negara di database. Semua operasi berjalan dengan coroutine (`suspend`), sehingga tidak mengganggu UI.

12. data/local/CountryEntity.kt

Kode di atas adalah data class `CountryEntity` yang digunakan sebagai representasi sebuah tabel bernama `country` di database lokal Room. Setiap objek dari kelas ini mewakili satu baris data negara. Properti `id` adalah primary key yang nilainya dibuat otomatis. `nama`, `tahun`, `alasan`, `imageUrl`, dan `wikiUrl` menyimpan informasi tentang nama negara, tahun dikunjungi atau diinginkan, alasan, gambar, dan link Wikipedia. Properti `isFavorite` adalah penanda apakah negara tersebut ditandai

sebagai favorit atau tidak. Kelas ini menjadi model utama untuk menyimpan dan mengambil data negara dalam aplikasi.

13. data/local/CountryRepository.kt

Kode di atas adalah implementasi dari `CountryRepository`, yaitu class yang bertanggung jawab untuk mengelola data negara dari dua sumber: database lokal (`CountryDao`) dan API online (`RetrofitClient`). Method `getAllCountries`, `getCountryByName`, dan `getFavoriteCountries` digunakan untuk mengambil data dari database secara real-time (menggunakan Flow). Method `insertAll` dan `updateCountry` digunakan untuk menyimpan dan memperbarui data negara di database. Method `fetchCountriesFromApi` mengambil data negara dari API, memetakan respon tersebut menjadi objek `CountryEntity`, lalu menyiapkan datanya agar bisa disimpan ke database. Kode ini menjembatani sumber data luar dan lokal agar bisa diakses oleh ViewModel.

14. data/remote/ApiService.kt

Kode di atas adalah interface `ApiService` yang digunakan untuk mendefinisikan endpoint API menggunakan Retrofit. Method `getAllCountries()` akan melakukan request GET ke endpoint `all` dan mengambil data negara dari internet. Parameter `fields` digunakan untuk membatasi data yang diambil, yaitu hanya `name`, `flags`, dan `region`, agar response lebih ringan. Method ini bersifat `suspend`, artinya harus dipanggil di dalam `coroutine` karena merupakan operasi jaringan yang bersifat `asynchronous` (tidak dilakukan di `main thread`). Data yang diterima akan berupa daftar objek `CountryResponse`.

15. data/remote/CountryResponse.kt

Kode ini berisi model data untuk membaca respons dari API tentang data negara. Kelas `CountryResponse` mewakili data utama yang berisi tiga properti: `name` (objek `Name?`), `flags` (objek `Flags?`), dan `region` (nama wilayah). Kelas `Name` hanya memiliki satu properti `common` yang berisi nama umum negara, sedangkan kelas `Flags` menyimpan URL gambar bendera dalam format PNG. Semua kelas ditandai dengan `@Serializable`, artinya data ini bisa diubah otomatis dari JSON ke objek Kotlin saat mengambil data dari internet menggunakan Kotlin Serialization.

16. data/remote/RetrofitClient.kt

Kode ini adalah konfigurasi `RetrofitClient` yang digunakan untuk mengambil data negara dari internet menggunakan Retrofit. Di dalamnya, `Json` disiapkan agar dapat mengabaikan data yang tidak dikenal dari respons API. Kemudian, `HttpLoggingInterceptor` ditambahkan untuk mencatat semua detail permintaan dan respons HTTP ke log, yang membantu saat debugging. `OkHttpClient` digunakan sebagai client untuk Retrofit. Lalu, `Retrofit` disiapkan dengan base URL `https://restcountries.com/v3.1/`, client, dan converter `Kotlinx Serialization` untuk mengubah data JSON menjadi objek Kotlin. Objek `apiService` digunakan untuk memanggil endpoint API yang didefinisikan di `ApiService`.

17. layout/activity_main.xml

Kode XML ini adalah layout utama untuk aplikasi Android yang menggunakan `ConstraintLayout` sebagai wadah utama. Di dalamnya terdapat dua komponen utama:

1. `FragmentContainerView` dengan ID `nav_host_fragment`, berfungsi sebagai tempat untuk menampilkan fragment yang dikelola oleh `Navigation Component`. Ini memungkinkan navigasi antar layar (fragment) dalam aplikasi.
2. `BottomNavigationView` dengan ID `bottom_navigation`, adalah menu navigasi di bagian bawah layar yang memungkinkan pengguna berpindah antar halaman utama aplikasi dengan ikon dan teks.

Layout ini mengatur agar `FragmentContainerView` memenuhi ruang dari atas hingga tepat di atas `BottomNavigationView`, sementara `BottomNavigationView` menempel di bagian bawah layar. Atribut seperti `app:navGraph` dan `app:menu` menghubungkan ke file navigasi dan menu masing-masing.

18. layout/fragment_detail.xml

Kode XML ini merupakan desain tampilan detail pada aplikasi Android yang dibungkus dengan `ScrollView` agar kontennya bisa digulir jika melebihi ukuran layar. Di dalamnya terdapat `ConstraintLayout` sebagai wadah utama yang menyusun beberapa elemen UI. Pertama, terdapat `ProgressBar` yang berguna untuk menampilkan indikator loading saat data sedang dimuat, namun secara default disembunyikan. Selanjutnya, terdapat `Group` yang mengelompokkan tiga elemen utama: gambar, nama, dan alasan, sehingga bisa dikontrol tampilannya secara

bersamaan. Elemen ImageView digunakan untuk menampilkan gambar, sedangkan dua TextView masing-masing menampilkan nama secara mencolok dan alasan dalam bentuk paragraf. Semua elemen diatur secara responsif di tengah layar dengan margin yang cukup agar tampilan tetap rapi dan mudah dibaca.

19. layout/fragment_favorite.xml

Kode XML ini merupakan tampilan untuk halaman favorit dalam aplikasi Android, yang menggunakan ConstraintLayout sebagai wadah utama. Di dalamnya terdapat tiga elemen penting. Pertama, RecyclerView dengan ID `rvNegara` digunakan untuk menampilkan daftar negara favorit secara scrollable, dan setiap itemnya mengikuti layout dari `item_negara`. Kedua, ada ProgressBar yang ditampilkan saat data sedang dimuat, namun secara default disembunyikan. Ketiga, terdapat TextView dengan pesan “Belum ada negara favorit” yang akan muncul jika daftar favorit kosong. Semua elemen diposisikan di tengah layar menggunakan constraint, dan disiapkan agar tampilannya responsif dan informatif.

20. layout/fragment_home.xml

Kode XML ini adalah layout untuk halaman yang menampilkan daftar negara dalam aplikasi Android, menggunakan ConstraintLayout sebagai wadah utama. Di dalamnya terdapat tiga komponen utama. Pertama, RecyclerView dengan ID rvNegara yang digunakan untuk menampilkan daftar negara secara scrollable dan rapi, dengan padding 8dp dan setiap item-nya mengikuti layout item_negara. Kedua, ProgressBar yang berada di tengah layar dan hanya akan muncul saat data sedang dimuat. Ketiga, TextView dengan ID tv_error yang digunakan untuk menampilkan pesan kesalahan jika terjadi error saat memuat data, namun disembunyikan secara default. Semua elemen diposisikan di tengah dan memenuhi layar dengan constraint yang menghubungkan ke parent.

21. item_negara.xml

Kode XML ini merupakan layout untuk satu item dalam daftar (list) yang ditampilkan menggunakan CardView agar terlihat seperti kartu dengan sudut membulat dan bayangan (elevation). Di dalam CardView terdapat ConstraintLayout yang menyusun elemen-elemen secara rapi. Komponen utama di dalamnya adalah ShapeableImageView untuk menampilkan foto negara, diikuti dua TextView untuk

menampilkan nama negara dan alasannya. Lalu ada dua Button: satu untuk melihat detail dan satu lagi untuk membuka Wikipedia. Terakhir, ada ImageButton di pojok kanan atas untuk memberi tanda favorit, menggunakan ikon hati. Semua elemen diatur agar tampil sejajar dan responsif dengan tampilan yang bersih dan mudah dibaca.

22. navigation/nav_graph.xml

Kode XML ini adalah file navigasi untuk aplikasi Android yang menggunakan Navigation Component. Di dalamnya didefinisikan tiga fragment utama: HomeFragment, FavoriteFragment, dan DetailFragment. Fragment awal yang akan ditampilkan saat aplikasi dibuka adalah HomeFragment, karena ditentukan sebagai startDestination. Dari HomeFragment dan FavoriteFragment, pengguna bisa berpindah ke DetailFragment melalui aksi (action) navigasi yang sudah ditentukan. DetailFragment juga menerima argumen bernama "name" dengan tipe data string, yang biasanya digunakan untuk menampilkan detail berdasarkan item yang dipilih. Navigasi ini memungkinkan perpindahan antar layar (fragment) secara terstruktur dan mudah dikontrol.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/annacoded/Pemrograman-Mobile>