LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 5

MENGAMBIL DATA DARI INTERNET

Oleh:

Anis Hanifa **NIM. 2010817320010**

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS LAMBUNG MANGKURAT

JUNI 2022

LEMBAR PENGESAHAN

LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE

MODUL 5

Laporan Praktikum Pemrograman Mobile Modul 5: Mengambil Data dari Internet ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Prakitkum ini dikerjakan oleh:

Nama Praktikan : Anis Hanifa

NIM : 2010817320010

Menyetujui, Asisten Praktikum	Mengetahui, Dosen Penanggung Jawab Praktikum
Rezi Rahdianor NIM. 1810817210019	Andreyan Rizky Baskara, S.Kom., M.Kom. NIP. 19930703 201903 1 011

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

SOAL 1

Buatlah sebuah aplikasi Android sederhana untuk menampilkan data dari Internet melalui Public API

1. Daftar Public API yang dapat digunakan dapat dilihat pada link berikut: https://github.com/public-apis/public-apis (dapat juga mengambil diluar dari link tersebut)
2. Pada saat dijalankan, aplikasi akan terhubung dengan Internet untuk menarik data dari **Public API** tersebut
3. Gunakan library tambahan yaitu **Retrofit** untuk mempermudah proses koneksi internet
4. Gunakan library tambahan yaitu **Mochi** untuk mempermudah proses data JSON
5. Data tersebut kemudian ditampilkan dalam bentuk **RecyclerView**
6. Masing-masing data di RecyclerView tersebut dapat diklik untuk menampilkan detailnya
7. Gunakan **LiveData** dan **ViewModel** untuk mempertahankan state dari aplikasi pada saat Configuration Changes
8. Saat pengguna merotasi tampilan handphone dari **Portrait** menjadi **Landscape** maka tampilan data yang sudah ada tidak boleh hilang

A. Source Code Kotlin

Tabel 1 FishServiceApi.kt

```
1   package com.example.publicapi.network
2
3   import com.squareup.moshi.Moshi
4   import com.squareup.moshi.kotlin.reflect.KotlinJsonAdapterFactory
5   import retrofit2.Retrofit
6   import retrofit2.converter.moshi.MoshiConverterFactory
7   import retrofit2.http.GET
8
9   private const val BASE_URL = "https://www.fishwatch.gov/api/species"
10
11  private val moshi = Moshi.Builder()
12      .add(KotlinJsonAdapterFactory())
13      .build()
14
15  private val retrofit = Retrofit.Builder()
16      .addConverterFactory(MoshiConverterFactory.create(moshi))
17      .baseUrl(BASE_URL)
18      .build()
19
20  interface FishServiceApi {
21      @GET("species")
22      suspend fun getFish() : FishProperty
23  }
24
25  object FishApi{
26      val retrofitServiceApi : FishServiceApi by lazy {
27          retrofit.create(FishServiceApi::class.java)
28      }
29  }
```

Tabel 2 FishProperty.kt

```
1   package com.example.publicapi.network
2
3   data class FishProperty (
4       val results : List<FishItems>? = null,
5       )
6
```

```
7   data class FishItems (
8       val speciesName: String? = null,
9       val scientificName: String? = null,
10      val image: String? = null,
11      val aliases: String? = null,
12      val status: String? = null
13      )
```

Tabel 3 FishViewModel.kt

```
1   package com.example.publicapi.ui
2
3   import androidx.lifecycle.LiveData
4   import androidx.lifecycle.MutableLiveData
5   import androidx.lifecycle.ViewModel
6   import androidx.lifecycle.viewModelScope
7   import com.example.publicapi.network.FishApi
8   import com.example.publicapi.network.FishItems
9   import kotlinx.coroutines.launch
10
11  enum class FishApiStatus { LOADING, ERROR, DONE}
12
13  class FishViewModel: ViewModel() {
14      private val _status = MutableLiveData<FishApiStatus>()
15      val status: LiveData<FishApiStatus> = _status
16
17      private val _Fishes = MutableLiveData<List<FishItems>?> ()
18      val Fishes: MutableLiveData<List<FishItems>?> = _Fishes
19
20      private val _Fish = MutableLiveData<FishItems>()
21      val Fish: LiveData<FishItems> = _Fish
22
23      fun getFishList() {
24          viewModelScope.launch {
25              _status.value = FishApiStatus.LOADING
26              try {
27                  _Fishes.value = FishApi.retrofitServiceApi.getFish().results
28                  _status.value = FishApiStatus.DONE
29              } catch (e: Exception) {
30                  _Fishes.value = listOf()
31                  _status.value = FishApiStatus.ERROR
32              }
33          }
34      }
35
36      fun onFishClicked(Fish: FishItems) {
37          _Fish.value = Fish
38      }
39  }
```

Tabel 4 FishListFragment.kt

```
1   package com.example.publicapi.ui
2
3   import android.os.Bundle
4   import android.view.LayoutInflater
5   import android.view.View
6   import android.view.ViewGroup
7   import androidx.appcompat.app.AppCompatActivity
8   import androidx.fragment.app.Fragment
9   import androidx.fragment.app.activityViewModels
10  import androidx.navigation.fragment.findNavController
11  import com.example.publicapi.R
12  import com.example.publicapi.databinding.FragmentFishListBinding
13
14  class FishListFragment: Fragment() {
15      private val viewModel: FishViewModel by activityViewModels()
16
17      override fun onCreateView(
18          inflater: LayoutInflater,
19          container: ViewGroup?,
20          savedInstanceState: Bundle?
21      ): View? {
```

```
22          val binding = FragmentFishListBinding.inflate(inflater)
23          viewModel.getFishList()
24          binding.lifecycleOwner = this
25          binding.viewModel = viewModel
26          binding.recyclerView.adapter = FishListAdapter(FishListener { fish ->
27              viewModel.onFishClicked(fish)
28              findNavController()
29                  .navigate(R.id.action_FishListFragment_to_FishDetailFragment)
30          })
31          (activity   as   AppCompatActivity).supportActionBar?.title   =   "The   Fishes
32  Profiles"
33
34          return binding.root
35      }
36  }
```

Tabel 5 FishDetailFragment.kt

```
1   package com.example.publicapi.ui
2
3   import android.os.Bundle
4   import android.view.LayoutInflater
5   import android.view.MenuItem
6   import android.view.View
7   import android.view.ViewGroup
8   import androidx.fragment.app.Fragment
9   import androidx.fragment.app.activityViewModels
10  import androidx.navigation.fragment.findNavController
11  import com.example.publicapi.R
12  import com.example.publicapi.databinding.FragmentFishDetailBinding
13
14  class FishDetailFragment: Fragment() {
15      private val viewModel: FishViewModel by activityViewModels()
16
17      override fun onCreateView(
18          inflater: LayoutInflater,
19          container: ViewGroup?,
20          savedInstanceState: Bundle?
21      ): View? {
22          val binding = FragmentFishDetailBinding.inflate(inflater)
23
24          binding.lifecycleOwner = this
25          binding.viewModel = viewModel
26
27          return binding.root
28      }
29
30      override fun onCreate(savedInstanceState: Bundle?) {
31          super.onCreate(savedInstanceState)
32          setHasOptionsMenu(true)
33      }
34
35      override fun onOptionsItemSelected(item: MenuItem): Boolean {
36          when(item.itemId){
37              android.R.id.home                                                    ->
38  findNavController().navigate(R.id.action_FishDetailFragment_to_FishListFragment)
39          }
40          return true
41      }
42  }
```

Tabel 6 FishListAdapter.kt

```
1   package com.example.publicapi.ui
2
3   import android.view.LayoutInflater
4   import android.view.ViewGroup
5   import androidx.recyclerview.widget.DiffUtil
6   import androidx.recyclerview.widget.ListAdapter
7   import androidx.recyclerview.widget.RecyclerView
8   import com.example.publicapi.databinding.ListViewItemBinding
9   import com.example.publicapi.network.FishItems
10
```

```
11  class FishListAdapter ( val clickListener: FishListener) :
12      ListAdapter<FishItems, FishListAdapter.FishViewHolder>(DiffCallback) {
13
14      class FishViewHolder(
15          var binding: ListViewItemBinding
16      ) : RecyclerView.ViewHolder(binding.root) {
17          fun bind(clickListener: FishListener, Fish: FishItems) {
18              binding.fish = Fish
19              binding.clickListener = clickListener
20              binding.executePendingBindings()
21          }
22      }
23
24      companion object DiffCallback: DiffUtil.ItemCallback<FishItems>() {
25          override fun areItemsTheSame(oldItem: FishItems, newItem: FishItems): Boolean
26  {
27              return oldItem.speciesName == newItem.speciesName
28          }
29
30          override  fun  areContentsTheSame(oldItem:  FishItems,  newItem:  FishItems):
31  Boolean {
32              return oldItem.image == newItem.image
33                      && oldItem.scientificName == newItem.scientificName
34                      && oldItem.aliases == newItem.aliases
35                      && oldItem.status == newItem.status
36          }
37      }
38      override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): FishViewHolder
39  {
40          val layoutInflater = LayoutInflater.from(parent.context)
41          return FishViewHolder(
42              ListViewItemBinding.inflate(layoutInflater, parent, false)
43          )
44      }
45      override fun onBindViewHolder(holder: FishViewHolder, position: Int) {
46          val Fish = getItem(position)
47          holder.bind(clickListener, Fish)
48      }
49  }
50
51  class FishListener(val clickListener: (Fish: FishItems) -> Unit) {
52      fun onClick(Fish: FishItems) = clickListener(Fish)
53  }
```

Tabel 7 BindingAdapters.kt

```
1   package com.example.publicapi
2
3   import android.view.View
4   import android.widget.ImageView
5   import androidx.core.net.toUri
6   import androidx.databinding.BindingAdapter
7   import androidx.recyclerview.widget.RecyclerView
8   import coil.load
9   import com.example.publicapi.network.FishItems
10  import com.example.publicapi.ui.FishApiStatus
11  import com.example.publicapi.ui.FishListAdapter
12
13  @BindingAdapter("listData")
14  fun bindRecyclerView(recyclerView: RecyclerView, data: List<FishItems>?){
15      val adapter = recyclerView.adapter as FishListAdapter
16      adapter.submitList(data)
17  }
18
19  @BindingAdapter("imageUrl")
20  fun bindImage(imgView: ImageView, imgUrl: String?) {
21      imgUrl?.let {
22          val imgUri = imgUrl.toUri().buildUpon().scheme("https").build()
23          imgView.load(imgUri) {
24              placeholder(R.drawable.loading_animation)
25              error(R.drawable.ic_broken_image)
26          }
27      }
28  }
29
```

```
@BindingAdapter("apiStatus")
fun bindStatus(statusImageView: ImageView, status: FishApiStatus?) {
    when(status) {
        FishApiStatus.LOADING -> {
            statusImageView.visibility = View.VISIBLE
            statusImageView.setImageResource(R.drawable.loading_animation)
        }
        FishApiStatus.DONE -> {
            statusImageView.visibility = View.GONE
        }
        FishApiStatus.ERROR -> {
            statusImageView.visibility = View.VISIBLE
            statusImageView.setImageResource(R.drawable.ic_connection_error)
        }
    }
}
```

Tabel 8 MainActivity.kt

```
package com.example.publicapi

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.navigation.NavController
import androidx.navigation.fragment.NavHostFragment
import androidx.navigation.ui.NavigationUI

class MainActivity : AppCompatActivity() {
    private lateinit var navController: NavController

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val                              navHostFragment                                =
supportFragmentManager.findFragmentById(R.id.nav_host_fragment) as NavHostFragment
        navController = navHostFragment.navController
        NavigationUI.setupActionBarWithNavController(this, navController)
    }
}
```

B. Source Code XML

Tabel 9 list_view_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <data>
        <variable
            name="Fish"
            type="com.example.publicapi.network.FishItems" />
        <variable
            name="clickListener"
            type="com.example.publicapi.ui.FishListener" />
    </data>

    <com.google.android.material.card.MaterialCardView
        android:id="@+id/card"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="@{() -> clickListener.onClick(Fish)}"
        android:layout_margin="8dp"
        app:cardElevation="8dp"
        app:cardCornerRadius="8dp">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="20dp">
```

```
27
28              <ImageView
29                  android:id="@+id/picture"
30                  android:layout_width="match_parent"
31                  android:layout_height="500px"
32                  android:scaleType="centerCrop"
33                  app:imageUrl="@{Fish.image}"
34                  app:layout_constraintEnd_toEndOf="parent"
35                  app:layout_constraintStart_toStartOf="parent"
36                  app:layout_constraintTop_toTopOf="parent"/>
37
38              <TextView
39                  android:id="@+id/speciesName"
40                  android:layout_width="wrap_content"
41                  android:layout_height="wrap_content"
42                  android:layout_marginTop="10dp"
43                  android:fontFamily="serif"
44                  android:padding="5dp"
45                  android:paddingTop="5dp"
46                  android:text="@{Fish.speciesName}"
47                  android:textAppearance="?attr/textAppearanceHeadline6"
48                  app:layout_constraintStart_toStartOf="parent"
49                  app:layout_constraintTop_toBottomOf="@+id/picture"
50                  tools:text="Species Name" />
51          </androidx.constraintlayout.widget.ConstraintLayout>
52
53      </com.google.android.material.card.MaterialCardView>
54  </layout>
```

Tabel 10 fragment_fish_list.xml

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <layout
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
7       <data>
8           <variable
9               name="viewModel"
10              type="com.example.publicapi.ui.FishViewModel" />
11      </data>
12
13      <androidx.constraintlayout.widget.ConstraintLayout
14          android:layout_width="match_parent"
15          android:layout_height="match_parent">
16
17          <androidx.recyclerview.widget.RecyclerView
18              android:id="@+id/recycler_view"
19              android:layout_width="match_parent"
20              android:layout_height="wrap_content"
21              android:padding="8dp"
22              app:listData="@{viewModel.Fishes}"
23              app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
24              app:layout_constraintStart_toStartOf="parent"
25              app:layout_constraintTop_toTopOf="parent"
26              tools:listitem="@layout/list_view_item" />
27
28          <ImageView
29              android:id="@+id/status_image"
30              android:layout_width="wrap_content"
31              android:layout_height="wrap_content"
32              app:apiStatus="@{viewModel.status}"
33              app:layout_constraintBottom_toBottomOf="parent"
34              app:layout_constraintLeft_toLeftOf="parent"
35              app:layout_constraintRight_toRightOf="parent"
36              app:layout_constraintTop_toTopOf="parent" />
37      </androidx.constraintlayout.widget.ConstraintLayout>
38
39  </layout>
```

Tabel 11 fragment_fish_detail.xml

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <layout
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
7       <data>
8           <variable
9               name="viewModel"
10              type="com.example.publicapi.ui.FishViewModel" />
11
12          <variable
13              name="image"
14              type="com.example.publicapi.network.FishProperty" />
15      </data>
16      <ScrollView
17          android:layout_width="match_parent"
18          android:layout_height="match_parent" >
19
20          <androidx.constraintlayout.widget.ConstraintLayout
21              android:layout_width="match_parent"
22              android:layout_height="wrap_content"
23              android:padding="20dp">
24
25              <TextView
26                  android:id="@+id/speciesName"
27                  android:layout_width="match_parent"
28                  android:layout_height="wrap_content"
29                  android:padding="8dp"
30                  android:text="@{viewModel.Fish.speciesName}"
31                  android:textSize="36dp"
32                  android:textAppearance="?attr/textAppearanceHeadline6"
33                  app:layout_constraintStart_toStartOf="parent"
34                  app:layout_constraintTop_toTopOf="parent"
35                  tools:text="Species Name"/>
36
37              <ImageView
38                  android:id="@+id/thumbnail"
39                  android:layout_width="match_parent"
40                  android:layout_height="500px"
41                  android:scaleType="centerCrop"
42                  app:imageUrl="@{viewModel.Fish.image}"
43                  android:padding="6dp"
44                  app:layout_constraintStart_toStartOf="parent"
45                  app:layout_constraintTop_toBottomOf="@+id/speciesName"
46                  android:src="@drawable/ic_launcher_background"/>
47
48              <TextView
49                  android:id="@+id/scientificName"
50                  android:layout_width="match_parent"
51                  android:layout_height="wrap_content"
52                  android:padding="4dp"
53                  android:textSize="20dp"
54                  android:text="@{`Scientific Name : ` + viewModel.Fish.scientificName}"
55                  app:layout_constraintStart_toStartOf="parent"
56                  app:layout_constraintTop_toBottomOf="@+id/thumbnail"
57                  tools:text="Scientific Name"/>
58
59              <TextView
60                  android:id="@+id/aliases"
61                  android:layout_width="match_parent"
62                  android:layout_height="wrap_content"
63                  android:padding="4dp"
64                  android:textSize="20dp"
65                  android:text="@{`Spesies Aliases: ` + viewModel.Fish.aliases}"
66                  app:layout_constraintStart_toStartOf="parent"
67                  app:layout_constraintTop_toBottomOf="@+id/scientificName"
68                  tools:text="Species Aliases"/>
69
70              <TextView
71                  android:id="@+id/status"
72                  android:layout_width="match_parent"
73                  android:layout_height="wrap_content"
74                  android:padding="4dp"
```

```
                android:textSize="20dp"
                android:text="@{`Status: ` + viewModel.Fish.status}"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/aliases"
                tools:text="Status"/>

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

C. Output Program

(Aplikasi tidak dapat dijalankan karena terdapat masalah pada SDK Android Studio)

D. Pembahasan

Pada praktikum modul kelima, dibuatlah aplikasi Android sederhana yang dapat menampilkan data dari Internet melalui Public API yang telah disediakan dalam daftar. Aplikasi Android akan menarik data dari Public API yang digunakan, selain itu digunakan library tambahan yaitu Retrofit dan Mochi untuk mempermudah proses koneksi Internet dan data JSON. Aplikasi Android ditampilkan dalam bentuk RecyclerView dan menggunakan LiveData serta ViewModel untuk mempertahankan kondisi dari aplikasi pada saat perubahan konfigurasi.

Aplikasi Android yang dibuat menggunakan Public API dari FishWatch (fishwatch.gov) yaitu https://www.fishwatch.gov/api/species. Dimana Aplikasi Android menampilkan data profil dari ikan-ikan yang terdapat pada FishWatch, yang berisi nama spesies, nama ilmiah, spesies alias dan status dari ikan, serta gambar dari ikan yang bersangkutan.

E. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/anishanifa/praktikummobile2/tree/main/modul5