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

```kotlin
package com.example.publicapi.network

import com.squareup.moshi.Moshi
import com.squareup.moshi.kotlin.reflect.KotlinJsonAdapterFactory
import retrofit2.Retrofit
import retrofit2.converter.moshi.MoshiConverterFactory
import retrofit2.http.GET

private const val BASE_URL = "https://www.fishwatch.gov/api/species"

private val moshi = Moshi.Builder()
    .add(KotlinJsonAdapterFactory())
    .build()

private val retrofit = Retrofit.Builder()
    .addConverterFactory(MoshiConverterFactory.create(moshi))
    .baseUrl(BASE_URL)
    .build()

interface FishServiceApi {
    @GET("species")
    suspend fun getFish() : FishProperty
}

object FishApi{
    val retrofitServiceApi : FishServiceApi by lazy {
        retrofit.create(FishServiceApi::class.java)
    }
}
```

Tabel 2 FishProperty.kt

```kotlin
package com.example.publicapi.network

data class FishProperty (
    val results : List<FishItems>? = null,
    )

```

```
data class FishItems (
    val speciesName: String? = null,
    val scientificName: String? = null,
    val image: String? = null,
    val aliases: String? = null,
    val status: String? = null
    )
```

Tabel 3 FishViewModel.kt

```
package com.example.publicapi.ui

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.example.publicapi.network.FishApi
import com.example.publicapi.network.FishItems
import kotlinx.coroutines.launch

enum class FishApiStatus { LOADING, ERROR, DONE}

class FishViewModel: ViewModel() {
    private val _status = MutableLiveData<FishApiStatus>()
    val status: LiveData<FishApiStatus> = _status

    private val _Fishes = MutableLiveData<List<FishItems>?> ()
    val Fishes: MutableLiveData<List<FishItems>?> = _Fishes

    private val _Fish = MutableLiveData<FishItems>()
    val Fish: LiveData<FishItems> = _Fish

    fun getFishList() {
        viewModelScope.launch {
            _status.value = FishApiStatus.LOADING
            try {
                _Fishes.value = FishApi.retrofitServiceApi.getFish().results
                _status.value = FishApiStatus.DONE
            } catch (e: Exception) {
                _Fishes.value = listOf()
                _status.value = FishApiStatus.ERROR
            }
        }
    }

    fun onFishClicked(Fish: FishItems) {
        _Fish.value = Fish
    }
}
```

Tabel 4 FishListFragment.kt

```
package com.example.publicapi.ui

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import androidx.fragment.app.activityViewModels
import androidx.navigation.fragment.findNavController
import com.example.publicapi.R
import com.example.publicapi.databinding.FragmentFishListBinding

class FishListFragment: Fragment() {
    private val viewModel: FishViewModel by activityViewModels()

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val binding = FragmentFishListBinding.inflate(inflater)
        viewModel.getFishList()
        binding.lifecycleOwner = this
        binding.viewModel = viewModel
        binding.recyclerView.adapter = FishListAdapter(FishListener { fish ->
            viewModel.onFishClicked(fish)
            findNavController()
                .navigate(R.id.action_FishListFragment_to_FishDetailFragment)
        })
        (activity as AppCompatActivity).supportActionBar?.title = "The Fishes Profiles"

        return binding.root
    }
}
```

Tabel 5 FishDetailFragment.kt

```
package com.example.publicapi.ui

import android.os.Bundle
import android.view.LayoutInflater
import android.view.MenuItem
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.fragment.app.activityViewModels
import androidx.navigation.fragment.findNavController
import com.example.publicapi.R
import com.example.publicapi.databinding.FragmentFishDetailBinding

class FishDetailFragment: Fragment() {
    private val viewModel: FishViewModel by activityViewModels()

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val binding = FragmentFishDetailBinding.inflate(inflater)

        binding.lifecycleOwner = this
        binding.viewModel = viewModel

        return binding.root
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setHasOptionsMenu(true)
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when(item.itemId){
            android.R.id.home -> findNavController().navigate(R.id.action_FishDetailFragment_to_FishListFragment)
        }
        return true
    }
}
```

Tabel 6 FishListAdapter.kt

```
package com.example.publicapi.ui

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.DiffUtil
import androidx.recyclerview.widget.ListAdapter
import androidx.recyclerview.widget.RecyclerView
import com.example.publicapi.databinding.ListViewItemBinding
import com.example.publicapi.network.FishItems

```

```
class FishListAdapter ( val clickListener: FishListener) :
    ListAdapter<FishItems, FishListAdapter.FishViewHolder>(DiffCallback) {

    class FishViewHolder(
        var binding: ListViewItemBinding
    ) : RecyclerView.ViewHolder(binding.root) {
        fun bind(clickListener: FishListener, Fish: FishItems) {
            binding.fish = Fish
            binding.clickListener = clickListener
            binding.executePendingBindings()
        }
    }

    companion object DiffCallback: DiffUtil.ItemCallback<FishItems>() {
        override fun areItemsTheSame(oldItem: FishItems, newItem: FishItems): Boolean
{
            return oldItem.speciesName == newItem.speciesName
        }

        override fun areContentsTheSame(oldItem: FishItems, newItem: FishItems):
Boolean {
            return oldItem.image == newItem.image
                    && oldItem.scientificName == newItem.scientificName
                    && oldItem.aliases == newItem.aliases
                    && oldItem.status == newItem.status
        }
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): FishViewHolder
{
        val layoutInflater = LayoutInflater.from(parent.context)
        return FishViewHolder(
            ListViewItemBinding.inflate(layoutInflater, parent, false)
        )
    }
    override fun onBindViewHolder(holder: FishViewHolder, position: Int) {
        val Fish = getItem(position)
        holder.bind(clickListener, Fish)
    }
}

class FishListener(val clickListener: (Fish: FishItems) -> Unit) {
    fun onClick(Fish: FishItems) = clickListener(Fish)
}
```

Tabel 7 BindingAdapters.kt

```
package com.example.publicapi

import android.view.View
import android.widget.ImageView
import androidx.core.net.toUri
import androidx.databinding.BindingAdapter
import androidx.recyclerview.widget.RecyclerView
import coil.load
import com.example.publicapi.network.FishItems
import com.example.publicapi.ui.FishApiStatus
import com.example.publicapi.ui.FishListAdapter

@BindingAdapter("listData")
fun bindRecyclerView(recyclerView: RecyclerView, data: List<FishItems>?){
    val adapter = recyclerView.adapter as FishListAdapter
    adapter.submitList(data)
}

@BindingAdapter("imageUrl")
fun bindImage(imgView: ImageView, imgUrl: String?) {
    imgUrl?.let {
        val imgUri = imgUrl.toUri().buildUpon().scheme("https").build()
        imgView.load(imgUri) {
            placeholder(R.drawable.loading_animation)
            error(R.drawable.ic_broken_image)
        }
    }
}

```

```
@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">

            <ImageView
                android:id="@+id/picture"
                android:layout_width="match_parent"
                android:layout_height="500px"
                android:scaleType="centerCrop"
                app:imageUrl="@{Fish.image}"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent"/>

            <TextView
                android:id="@+id/speciesName"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:fontFamily="serif"
                android:padding="5dp"
                android:paddingTop="5dp"
                android:text="@{Fish.speciesName}"
                android:textAppearance="?attr/textAppearanceHeadline6"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/picture"
                tools:text="Species Name" />
        </androidx.constraintlayout.widget.ConstraintLayout>

    </com.google.android.material.card.MaterialCardView>
</layout>
```

Tabel 10 fragment_fish_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">

    <data>
        <variable
            name="viewModel"
            type="com.example.publicapi.ui.FishViewModel" />
    </data>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/recycler_view"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="8dp"
            app:listData="@{viewModel.Fishes}"
            app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            tools:listitem="@layout/list_view_item" />

        <ImageView
            android:id="@+id/status_image"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:apiStatus="@{viewModel.status}"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

</layout>
```

Tabel 11 fragment_fish_detail.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">

    <data>
        <variable
            name="viewModel"
            type="com.example.publicapi.ui.FishViewModel" />

        <variable
            name="image"
            type="com.example.publicapi.network.FishProperty" />
    </data>
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

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

            <TextView
                android:id="@+id/speciesName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="8dp"
                android:text="@{viewModel.Fish.speciesName}"
                android:textSize="36dp"
                android:textAppearance="?attr/textAppearanceHeadline6"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent"
                tools:text="Species Name"/>

            <ImageView
                android:id="@+id/thumbnail"
                android:layout_width="match_parent"
                android:layout_height="500px"
                android:scaleType="centerCrop"
                app:imageUrl="@{viewModel.Fish.image}"
                android:padding="6dp"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/speciesName"
                android:src="@drawable/ic_launcher_background"/>

            <TextView
                android:id="@+id/scientificName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="4dp"
                android:textSize="20dp"
                android:text="@{`Scientific Name : ` + viewModel.Fish.scientificName}"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/thumbnail"
                tools:text="Scientific Name"/>

            <TextView
                android:id="@+id/aliases"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="4dp"
                android:textSize="20dp"
                android:text="@{`Spesies Aliases: ` + viewModel.Fish.aliases}"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/scientificName"
                tools:text="Species Aliases"/>

            <TextView
                android:id="@+id/status"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                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