LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 5

Mengambil Data dari Internet

Oleh:

Ahmad Zaini **NIM. 2010817310001**

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 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 Praktikum ini dikerjakan oleh:

Nama Praktikan : Ahmad Zaini
NIM : 2010817310001

Menyetujui, Asisten Praktikum	Mengetahui, Dosen Penanggung Jawab Praktikum
Muhammad Rizal NIM. 1810817210020	Muhammad Alkaff, S.Kom., M.Kom. NIP. 19860613 201504 1 011

DAFTAR ISI

DAFTAR GAMBAR

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 Network/Movie.kt

```
1   package com.zai.movieku.network
2
3   import java.io.Serializable
4
5   data class Movie(val title:String,
6                    val poster_path:String,
7                    val release_date:String,
8                    val vote_average:Double,
9                    val vote_count:Int,
10                   val overview: String
11  ) : Serializable {}
12  data class MoviesResponse(val page:Int,
13                            val results:List<Movie>,
14                            val total_pages:Int,
15                            val total_results:Int){}
```

B. Source Code Network/MovieApi.kt

```
1  package com.zai.movieku.network
2
3  import
4  com.jakewharton.retrofit2.adapter.kotlin.coroutines.CoroutineCal
5  lAdapterFactory
6  import com.squareup.moshi.Moshi
7  import
8  com.squareup.moshi.kotlin.reflect.KotlinJsonAdapterFactory
9  import kotlinx.coroutines.Deferred
10 import retrofit2.Retrofit
11 import retrofit2.converter.moshi.MoshiConverterFactory
12 import retrofit2.converter.scalars.ScalarsConverterFactory
13 import retrofit2.http.GET
14
15 private const val BASE_URL = "https://api.themoviedb.org/3/"
16
17 val moshi = Moshi.Builder()
18     .add(KotlinJsonAdapterFactory())
19     .build()
20
21 private val retrofit = Retrofit.Builder()
22     .addConverterFactory(ScalarsConverterFactory.create())
23     .addCallAdapterFactory(CoroutineCallAdapterFactory())
24     .addConverterFactory(MoshiConverterFactory.create(moshi))
25     .baseUrl(BASE_URL)
26     .build()
27
28 interface MoviesApiService {
29
30 @GET("discover/movie?api_key=efbf02374fe534aa08c13ca1873bc1f7")
31     fun getMovies(): Deferred<MoviesResponse>
32 }
33
34 object MoviesApi {
35     val retrofitService : MoviesApiService by lazy {
        retrofit.create(MoviesApiService::class.java)
    }
}
```

C. Source Code ui/movie_details/MovieDetailsFragment.kt

```
1  package com.zai.movieku.ui.movie_details
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.appcompat.app.AppCompatActivity
9  import androidx.fragment.app.Fragment
10 import androidx.fragment.app.activityViewModels
11 import androidx.navigation.fragment.findNavController
12 import com.zai.movieku.R
13 import com.zai.movieku.databinding.MainFragmentDetailsBinding
14 import com.zai.movieku.ui.movies.MoviesViewModel
15
16 class MovieDetailsFragment : Fragment() {
17
18     private     val     viewModel     :     MoviesViewModel     by
19 activityViewModels()
20
21     override fun onCreateView(
22         inflater: LayoutInflater,
23         container: ViewGroup?,
24         savedInstancesState: Bundle?
25     ): View? {
26         val                          binding                          =
27 MainFragmentDetailsBinding.inflate(inflater)
28         binding.lifecycleOwner = this
29         binding.viewModel = viewModel
30
31         (activity  as  AppCompatActivity).supportActionBar?.title
32 = viewModel.movie.value?.title
33         return binding.root
34     }
35
36     override fun onCreate(savedInstanceState: Bundle?) {
37         super.onCreate(savedInstanceState)
38         setHasOptionsMenu(true)
39     }
40
41     override  fun  onOptionsItemSelected(item:  MenuItem):  Boolean
42 {
43         when(item.itemId){
44             android.R.id.home                                        ->
```

```
findNavController().navigate(R.id.action_movieDetailFragment_to_
moviesFragment)
        }
        return true
    }
}
```

D. Source Code ui/movies/MovieAdapter.kt

```
package com.zai.movieku.ui.movies

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.ListAdapter
import androidx.recyclerview.widget.DiffUtil
import androidx.recyclerview.widget.RecyclerView
import com.zai.movieku.databinding.MovieRowBinding
import com.zai.movieku.network.Movie

class MovieAdapter(private val clickListener: MoviesListener) :
    ListAdapter<Movie,
MovieAdapter.MovieViewHolder>(DiffCallback)
{
        class MovieViewHolder(
            var binding: MovieRowBinding
        ) : RecyclerView.ViewHolder(binding.root){
            fun   bind(clickListener:   MoviesListener,   movie:
Movie) {
                binding.movie = movie
                binding.clickListener = clickListener
                binding.executePendingBindings()
            }
        }

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

        override fun areContentsTheSame(oldItem: Movie, newItem:
Movie): Boolean {
            return  oldItem.release_date  ==  newItem.release_date
&&  oldItem.overview  ==  newItem.overview  &&  oldItem.poster_path
== newItem.poster_path
        }
```

```
39      }
40
41      override fun onCreateViewHolder(parent: ViewGroup, viewType:
42  Int): MovieViewHolder {
43          val layoutInflater = LayoutInflater.from(parent.context)
44          return MovieViewHolder(
45              MovieRowBinding.inflate(layoutInflater,      parent,
46  false))
47      }
48
49      override   fun   onBindViewHolder(holder:   MovieViewHolder,
50  position: Int) {
51          val movie = getItem(position)
52          holder.bind(clickListener, movie)
53      }
54
55  }
56  class MoviesListener(val clickListener: (movie: Movie) -> Unit)
57  {
58      fun onClick(movie: Movie) = clickListener(movie)
59  }
```

E. Source Code ui/movies/MoviesFragment.kt

```
1   package com.zai.movieku.ui.movies
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.zai.movieku.R
12  import com.zai.movieku.databinding.FragmentMoviesBinding
13
14  class MoviesFragment : Fragment() {
15
16      private      val      viewModel:      MoviesViewModel      by
17  activityViewModels()
18
19      override fun onCreateView(
20          inflater: LayoutInflater,
21          container: ViewGroup?,
22          savedInstanceState: Bundle?
23      ): View? {
```

```
24          val binding = FragmentMoviesBinding.inflate(inflater)
25          viewModel.getMovieList()
26          binding.lifecycleOwner = this
27          binding.viewModel = viewModel
28          binding.recyclerView.adapter                                    =
29  MovieAdapter(MoviesListener { movies ->
30              viewModel.onMovieClicked(movies)
31              findNavController()
32
33  .navigate(R.id.action_moviesFragment_to_movieDetailsFragment)
34          })
35
36          (activity  as  AppCompatActivity).supportActionBar?.title
37  = "MovieKu"
38
39          return binding.root
40      }
41  }
```

F. Source Code ui/movies/MoviesViewModel.kt

```
1   package com.zai.movieku.ui.movies
2
3   import androidx.lifecycle.LiveData
4   import androidx.lifecycle.MutableLiveData
5   import androidx.lifecycle.ViewModel
6   import androidx.lifecycle.viewModelScope
7   import com.zai.movieku.network.Movie
8   import com.zai.movieku.network.MoviesApi
9   import kotlinx.coroutines.launch
10
11  enum class MoviesApiStatus { LOADING, ERROR, DONE }
12
13  class MoviesViewModel: ViewModel(){
14
15      private val _status = MutableLiveData<MoviesApiStatus>()
16      val status: LiveData<MoviesApiStatus> = _status
17
18      private val _movies = MutableLiveData<List<Movie>>()
19      val movies: LiveData<List<Movie>> = _movies
20
21      private val _movie = MutableLiveData<Movie>()
22      val movie: LiveData<Movie> = _movie
23
24      fun listToString(list: List<String>): String {
25          return list.joinToString("\n")
26      }
```

```
27
28      fun getMovieList() {
29          viewModelScope.launch {
30              _status.value = MoviesApiStatus.LOADING
31              try {
32                  _movies.value                                    =
33  MoviesApi.retrofitService.getMovies().await().results
34                  _status.value = MoviesApiStatus.DONE
35              } catch (e: Exception) {
36                  _movies.value = listOf()
37                  _status.value = MoviesApiStatus.ERROR
38              }
39          }
40      }
41
42      fun onMovieClicked(movie: Movie) {
43          _movie.value = movie
44      }
45  }
```

G. Source Code BindingAdapter.kt

```
1   package com.zai.movieku
2
3   import android.view.View
4   import android.widget.ImageView
5   import androidx.databinding.BindingAdapter
6   import androidx.recyclerview.widget.RecyclerView
7   import com.bumptech.glide.Glide
8   import com.bumptech.glide.request.RequestOptions
9   import com.zai.movieku.network.Movie
10  import com.zai.movieku.ui.movies.MovieAdapter
11  import com.zai.movieku.ui.movies.MoviesApiStatus
12
13  private         const         val         BASE_URL_IMAGE         =
14  "https://image.tmdb.org/t/p/w500/"
15
16  @BindingAdapter("listData")
17  fun     bindRecyclerView(recyclerView:     RecyclerView,     data:
18  List<Movie>?){
19      val adapter = recyclerView.adapter as MovieAdapter
20      adapter.submitList(data)
21  }
22
23  @BindingAdapter("imageUrl")
24  fun bindImage(imgView: ImageView, imgUrl: String?){
25      imgUrl?.let{
```

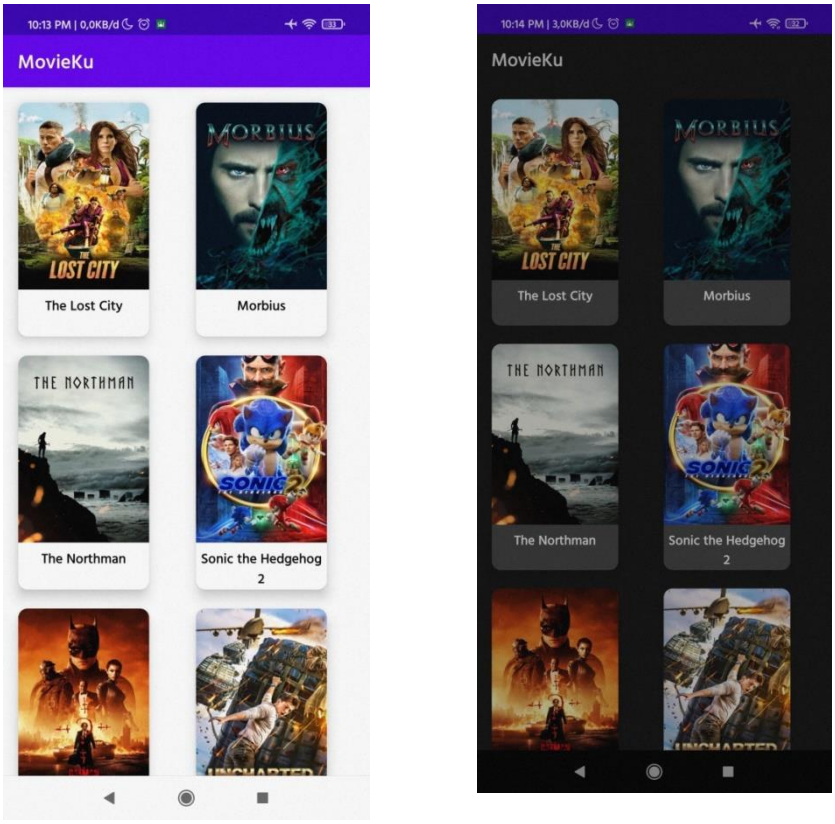
```
        Glide.with(imgView.context)
            .load(BASE_URL_IMAGE + it)
            .apply(
                RequestOptions())
            .into(imgView)
    }
}

@BindingAdapter("apiStatus")
fun bindStatus(statusImageView: ImageView, status:
MoviesApiStatus?) {
    when (status) {
        MoviesApiStatus.LOADING -> {
            statusImageView.visibility = View.VISIBLE
        }
        MoviesApiStatus.DONE -> {
            statusImageView.visibility = View.GONE
        }
        MoviesApiStatus.ERROR -> {
            statusImageView.visibility = View.VISIBLE

statusImageView.setImageResource(R.drawable.ic_connection_error)
        }
    }
}
```

H. Source Code MainActivity.kt

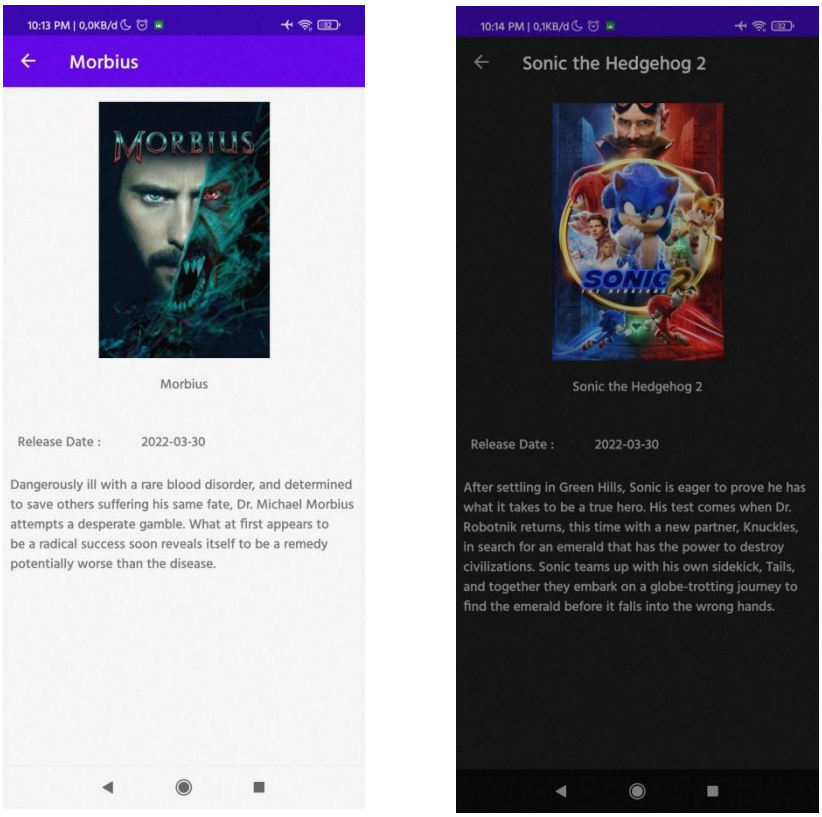
```
package com.zai.movieku

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.movies_host_fragment)
as NavHostFragment
        navController = navHostFragment.navController
        NavigationUI.setupActionBarWithNavController(this,
```

```
20  navController)
21      }
22  }
```

I. Output Program



Gambar 1. Tampilan awal light & darkmode



Gambar 2. Tampilan Detail Fragment dari item yang dipilih dalam light & dark mode

J. Pembahasan

- Pada file network/Movie.kt, berfungi untuk mengkonversi data apa saja yang ingin diambil pada result berbentuk JSON ke kotlin yang selanjutnya akan di handle pada file MovieApi.kt
- Pada file network/MovieApi.kt berfungsi untuk mengambil data dari BASE_URL dan mengkonversi bentuk JSON tadi ke objek,library moshi digunakan agar lebih cepat mengkonversi kalimat JSONnya
- Pada file ui/movie_details/MovieDetailsFragment.kt berfungsi menampilkan detail dari data saat kita mengklik salah satu item,maka akan memulai fragment yang berisi detail informasi item.
- Pada file ui/movies/MovieAdapter.kt berfungsi untuk menghubungkan data ke recyclerview,jadi akan muncul ke layar list dari MovieFragment
- Pada file ui/movies/MoviesFragment.kt berfungsi untuk menampilkan list data yang sudah di dapatkan pada API tadi
- Pada File ui/movies/MoviesViewModel.kt berfungsi untuk memperoleh data dan menyimpan hasil dari API,untuk disesuaikan pada layout..

- Pada file BindingAdapter.kt berfungsi untuk menyesuaikan tampilan,dan binding adapter dapat menimpa data sesuai dengan kondisi yang ada
- Pada file MainActivity.kt berfungsi menampilkan tampilan awal pada saat oncreate,selanjutnya akan menampilkan dari fragment lain.

K. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.
https://github.com/zaaii/Praktikum-Mobile/tree/main/Modul%205