

Link GitHub : https://github.com/HitoKawiswara/LAB_WEEK_12

Commit 2 :

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `MainActivity.kt` file, which contains code for setting up a RecyclerView with a MovieAdapter. The preview window on the right shows a list of movie posters, including "Nahual", "Wicked: For Good", "The Gentlemen", and "War of the Worlds".

```
package com.example.test_lab_week_12

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.RecyclerView
import com.example.test_lab_week_12.model.Movie
import com.google.android.material.snackbar.Snackbar
import java.util.Calendar // use java.util.Calendar

class MainActivity : AppCompatActivity() {

    private val movieAdapter by lazy {
        MovieAdapter(clickListener = object : MovieAdapter.ClickListener {
            override fun onMovieClick(movie: Movie) {
                openMovieDetails(movie)
            }
        })
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val recyclerView: RecyclerView = findViewById(R.id.recyclerView)
        recyclerView.adapter = movieAdapter
        movieAdapter.setMovies(movies)
    }

    val movieRepository = (application as MovieRepository)
}
```

Commit 3 :

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `MovieViewModel.kt` file, which contains code for managing movie data using coroutines and StateFlow. The preview window on the right shows a list of movie posters, including "THE SHADOW'S EDGE", "Bureau 749", "The Family Plan 2", and "Frankenstein". A tooltip at the bottom right indicates the install was successful.

```
package com.example.test_lab_week_12

import androidx.lifecycle.ViewModel
import androidx.lifecycle.ViewModelScope
import com.example.test_lab_week_12.model.Movie
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.StateFlow
import kotlinx.coroutines.launch

class MovieViewModel(
    private val movieRepository: MovieRepository
) : ViewModel() {

    private val _popularMovies = MutableStateFlow<List<Movie>>(
        value = emptyList()
    )
    val popularMovies: StateFlow<List<Movie>> = _popularMovies

    private val _error = MutableStateFlow<String>(
        value = ""
    )
    val error: StateFlow<String> = _error

    init {
        // supaya langsung load saat ViewModel dibuat
        fetchPopularMovies()
    }

    private fun fetchPopularMovies() {
        viewModelScope.launch(Dispatchers.IO) {
            _popularMovies.value = movieRepository.fetchPopularMovies()
        }
    }
}
```