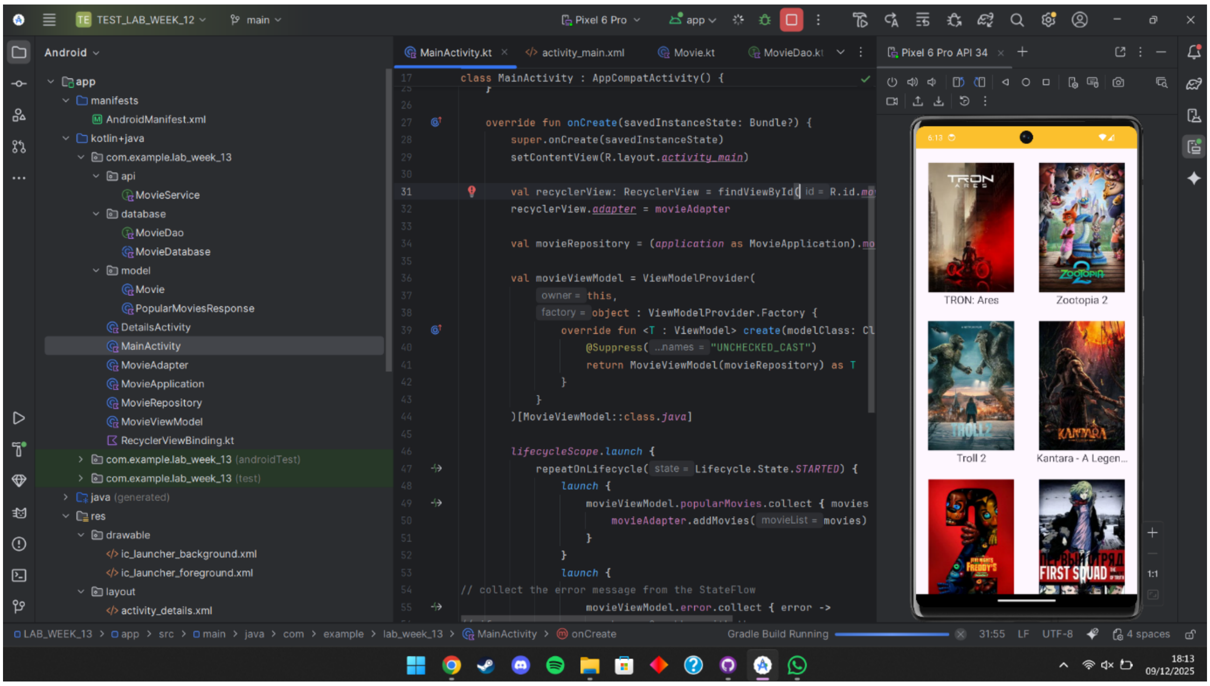
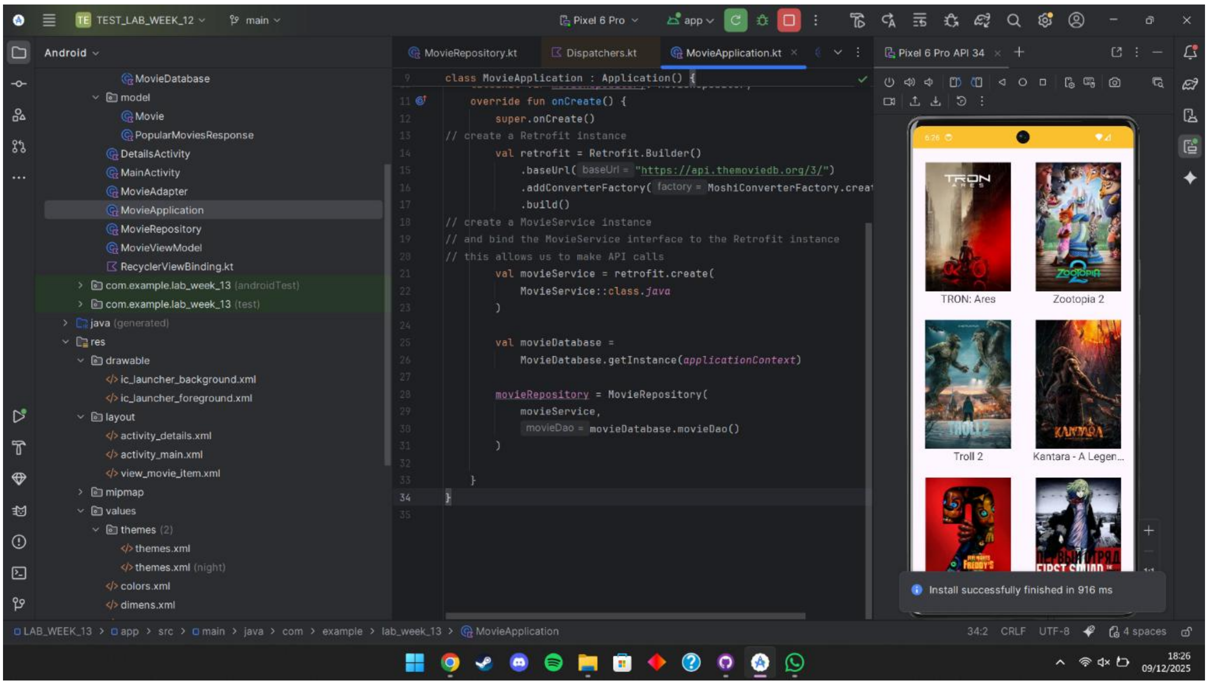
Link GitHub : https://github.com/HitoKawiswara/LAB_WEEK_13

Commit 1 :



```
class MainActivity : AppCompatActivity() {

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

        val recyclerView: RecyclerView = findViewById(R.id.mo
        recyclerView.adapter = movieAdapter

        val movieRepository = (application as MovieApplication).mo

        val movieViewModel = ViewModelProvider(
            this,
            object : ViewModelProvider.Factory {
                override fun <T : ViewModel> create(modelClass: Cl
                    @Suppress("UNCHECKED_CAST")
                    return MovieViewModel(movieRepository) as T
                }
            }
        )[MovieViewModel::class.java]

        lifecycleScope.launch {
            repeatOnLifecycle(Lifecycle.State.STARTED) {
                launch {
                    movieViewModel.popularMovies.collect { movies
                        movieAdapter.addMovies(movies)
                    }
                }
                launch {
// collect the error message from the StateFlow
                    movieViewModel.error.collect { error ->
```

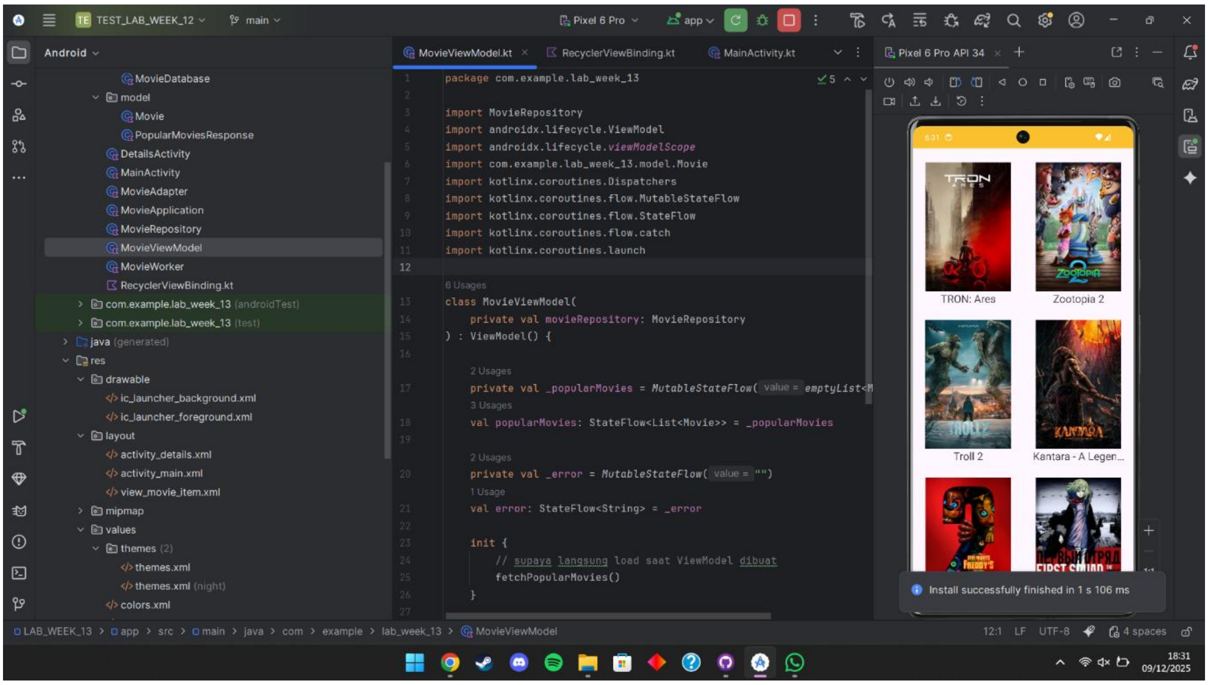
Commit 2 :



```
class MovieApplication : Application() {
    override fun onCreate() {
        super.onCreate()
// create a Retrofit instance
        val retrofit = Retrofit.Builder()
            .baseUrl("https://api.themoviedb.org/3/")
            .addConverterFactory(MoshiConverterFactory.creat
            .build()
// create a MovieService instance
// and bind the MovieService interface to the Retrofit instance
// this allows us to make API calls
        val movieService = retrofit.create(
            MovieService::class.java
        )

        val movieDatabase =
            MovieDatabase.getInstance(applicationContext)

        movieRepository = MovieRepository(
            movieService,
            movieDatabase.movieDao()
        )

    }
}
```

Install successfully finished in 916 ms

Commit 3 :



```
MovieDatabase
model
    Movie
    PopularMoviesResponse
DetailsActivity
MainActivity
MovieAdapter
MovieApplication
MovieRepository
MovieViewModel
MovieWorker
RecyclerViewBinding.kt
com.example.lab_week_13 (androidTest)
com.example.lab_week_13 (test)
java (generated)
res
    drawable
        ic_launcher_background.xml
        ic_launcher_foreground.xml
    layout
        activity_details.xml
        activity_main.xml
        view_movie_item.xml
    mipmap
    values
        themes (2)
            themes.xml
            themes.xml (night)
        colors.xml
```

```kotlin
package com.example.lab_week_13

import MovieRepository
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.example.lab_week_13.model.Movie
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.StateFlow
import kotlinx.coroutines.flow.catch
import kotlinx.coroutines.launch

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

    2 Usages
    private val _popularMovies = MutableStateFlow( value = emptyList<M
    3 Usages
    val popularMovies: StateFlow<List<Movie>> = _popularMovies

    2 Usages
    private val _error = MutableStateFlow( value = "")
    1 Usage
    val error: StateFlow<String> = _error

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

```

Install successfully finished in 1 s 106 ms

LAB_WEEK_13 > app > src > main > java > com > example > lab_week_13 > MovieViewModel
12:1 LF UTF-8 4 spaces
18:31
09/12/2025