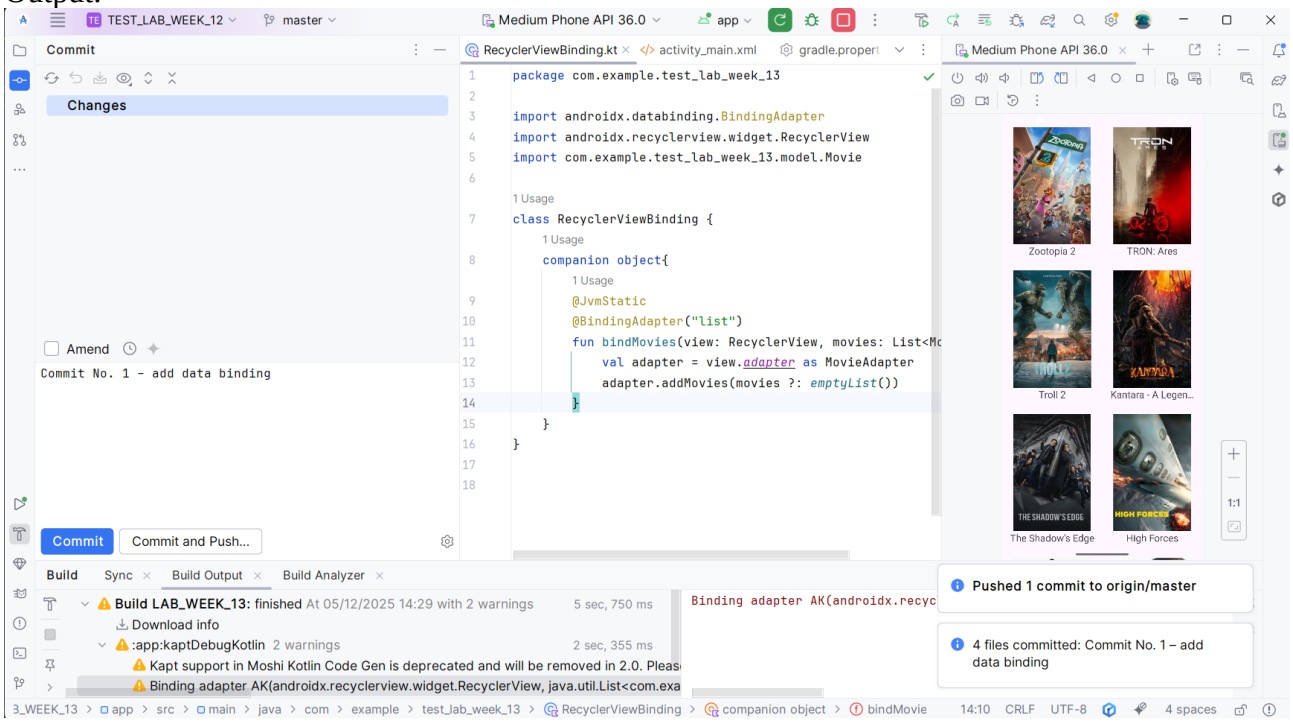
Nama : John Isaac Witness
NIM : 00000088626
Mobile Application Programming Week 13
Link GitHub: https://github.com/akunjone/LAB_WEEK_13

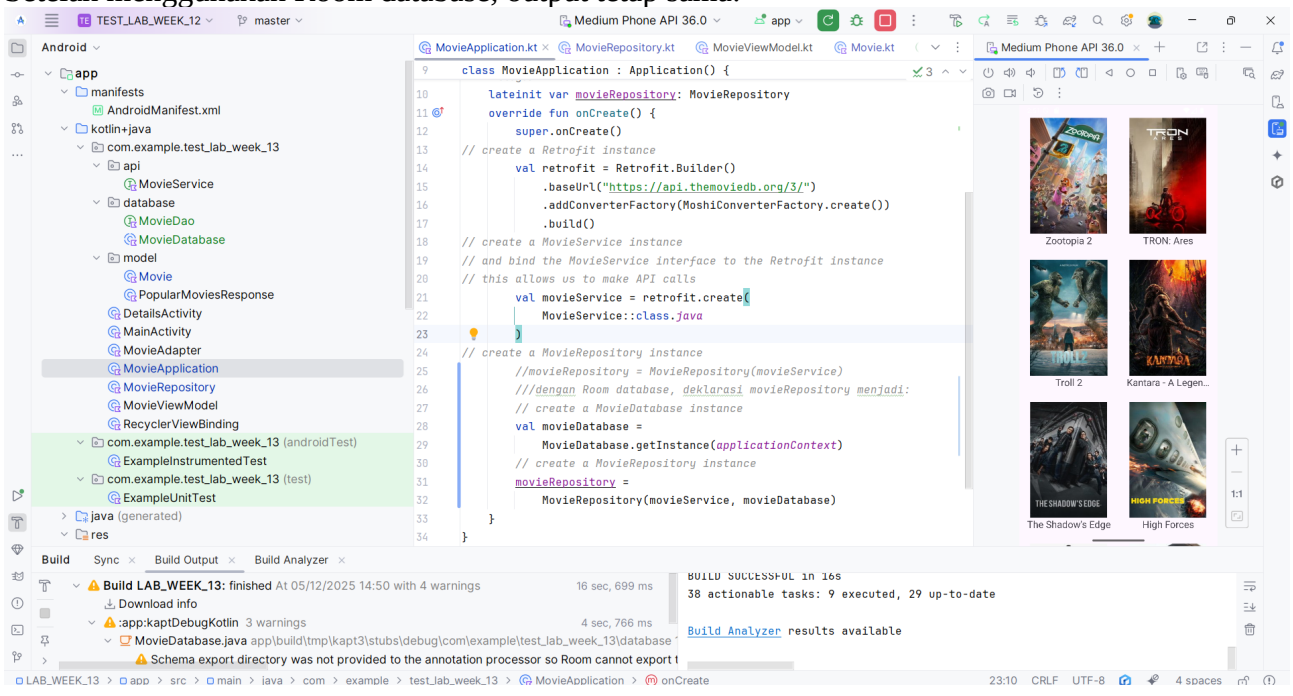
Part 1 - Using Data Binding in Android

Output:

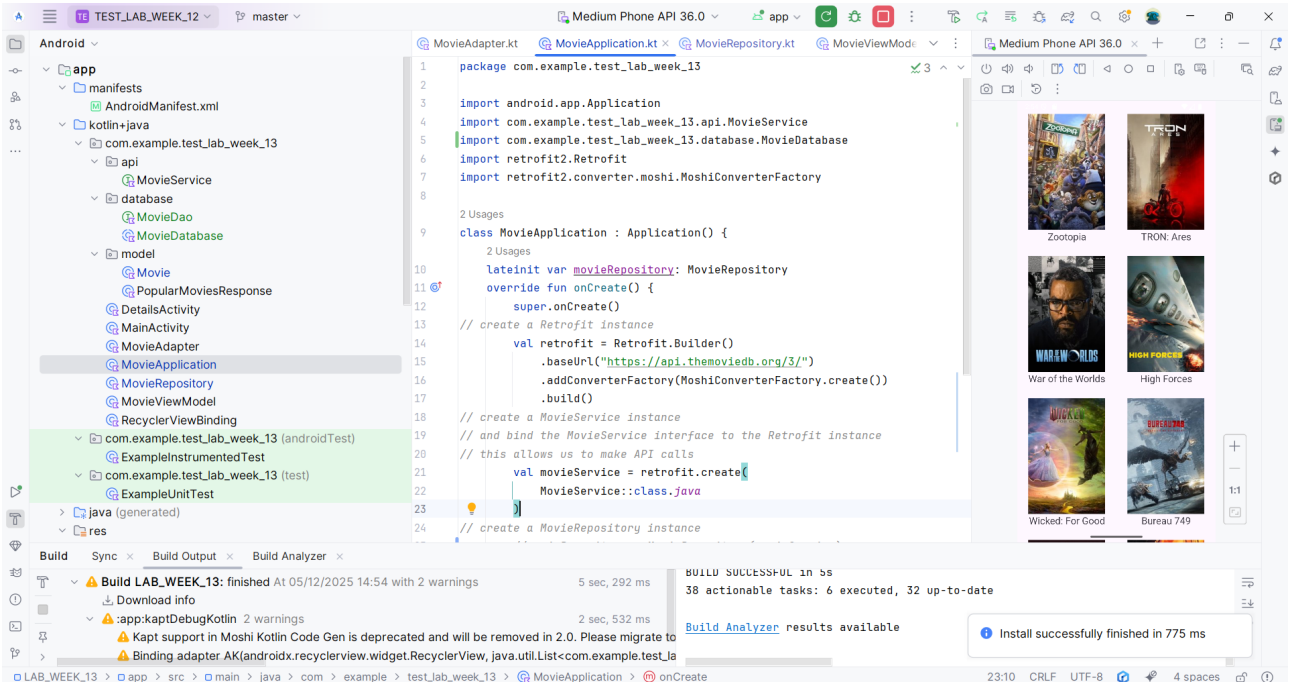


Part 2 - Using Repository with Room

Setelah menggunakan Room database, output tetap sama:



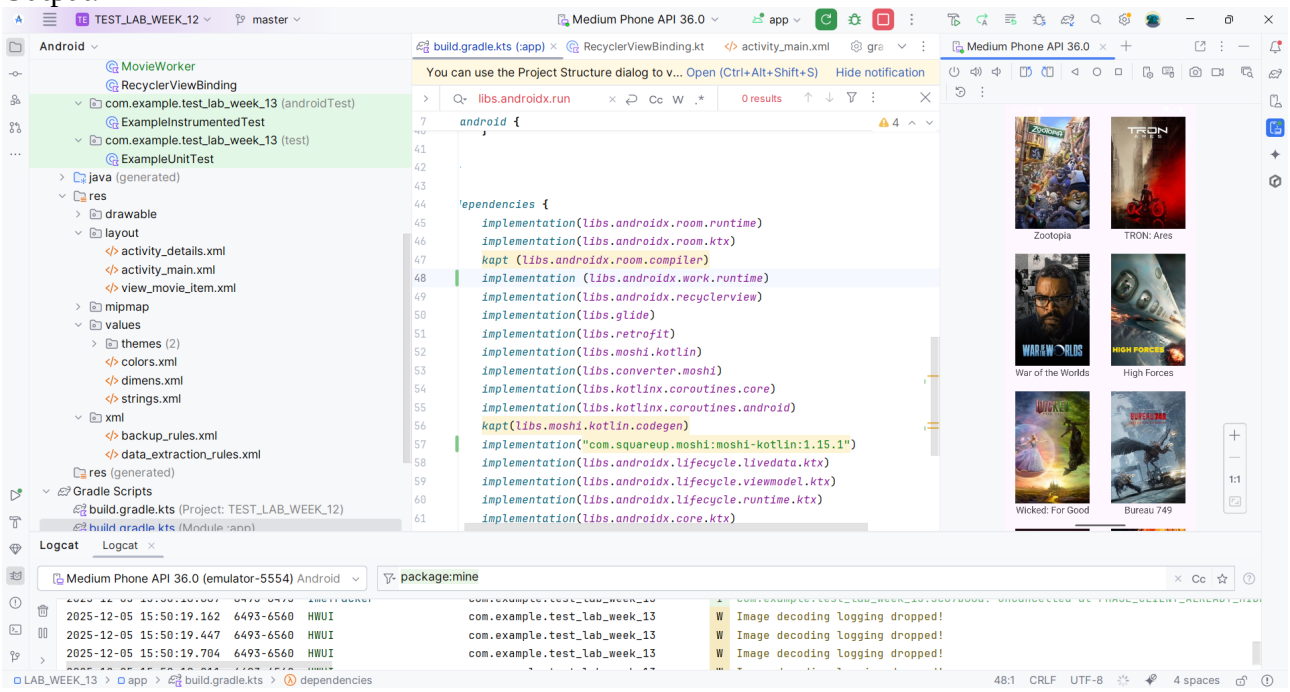
Namun bedanya jika menggunakan Room, semua movie yang didapat disimpan dan dimasukkan kedalam Room. Sehingga jika koneksi terputus, movies tetap ada (namun tidak ter up-to-date dengan waktu saat ini).
Misal koneksi terputus (saya ubah sedikit API-key nya), maka akan tetap muncul movienya:



Part 3 - Using WorkManager

Project di Part 2 sudah hampir selesai, tapi database tidak bisa direfresh datanya ke versi terbaru dari API. Cara menyelesaikan masalah ini adalah dengan mengambil data dari API saat selang waktu tertentu atau saat user tidak membuka aplikasi. Dengan cara ini, data yang sudah dicache akan selalu up-to-date. Kita bisa melakukannya dengan mengatur WorkManager.

Output:



Sama seperti sebelumnya, namun, database sudah ter up-to-date.

ASSIGNMENT

1. Why is MVVM important? Which files represent Model, which files represent View, and which files represent ViewModel?
2. In Part 1, you implemented Data Binding, why is this more efficient than using the normal method?

3. In Part 2, you implemented the Singleton Pattern, why is this important?
4. In Part 2 & 3, you implemented the Repository Pattern, why is this important?
5. In part 3, you implemented the Worker Manager, is there another way to refresh your database with the latest data other than using Worker?

ANSWERS FOR ASSIGNMENT

1. MVVM memisahkan UI dan logic bisnis. Jadi ketika kita ingin melakukan perbaikan untuk UI atau business logic, kita hanya perbaiki yang relevan saja. MVVM menjadi penting, karena ini membuat pekerjaan menjadi lebih mudah untuk menambah fitur baru, test kode baru, mengelola dan menjaga data lebih terarah juga. MVVM menjadi lebih baik juga karena strukturnya jelas dan tugas masing-masing juga jelas.
   - Yang menjadi Model adalah yang ada didalam folder “model”: Movie.kt dan PopularMoviesResponse.kt. Mereka menjadi struktur datanya (data layer).
   - View yang ada untuk layout aplikasi, untuk render data: MainActivity.kt, activity_main.xml, view_movie_item.xml, activity_details.xml.
   - ViewModel yang mengambil data dari Model dan menyediakannya ke View: MovieViewModel.kt
2. Data binding penting karena view model langsung berkomunikasi dengan views tanpa menggunakan yang seperti FindViewById, dll. Tujuannya untuk sinkronisasi secara otomatis antara View dan ViewModel, jadi ketika data pada satu sisi berubah, yang di sisi lainnya juga berubah.
3. Singleton pattern digunakan untuk deklarasi database. Dimana Singleton memastikan hanya ada satu instance database di beberapa thread. Tujuan singleton untuk memastikan data validity dan menghindari Race Condition. Singleton menjadi penting untuk mengurangi penggunaan resource yang mahal saat membuat instance database.
4. Repository pattern untuk menyinkronkan data antara API dengan Room (database). Repository ini memastikan data di database lokal selalu up-to-date dengan data yang ada di web. Ini penting di saat tidak ada internet, atau koneksi terputus, namun tetap membutuhkan info/sesuatu yang up-to-date.
5. Bisa menggunakan Coroutines saat aktif di foreground, Foreground Service untuk tugas yang diharuskan berjalan lama, JobScheduler, atau bisa menggunakan bantuan dari Firebase.

Terima kasih.