

| PROYEK INDIVIDU MATA KULIAH | | |
|-----------------------------|---|--|
| Nama Mata Kuliah | : | Pengembangan Aplikasi Perangkat Bergerak |
| Kelas | : | TIF - B |
| Nama Mahasiswa | : | Nadhif Rif'at Rasendriya / 235150201111074 |
| Nama Dosen Pengampu | : | Ir. Adam Hendra Brata, S.Kom., M.T., M.Sc. |

Bab 1 Android Studio

The screenshot displays the Android Studio IDE. The top toolbar shows various icons for file operations, running, and debugging. The left sidebar contains the 'Android' view, showing the project structure with folders like 'com.example.travelupa', 'res', and 'values'. The main editor window shows the 'MainActivity.kt' file, which is a Kotlin class implementing the 'LandingScreen' interface. The code includes a 'fun LandingScreen(onStartClick: () -> Unit, modifier: Modifier = Modifier)' function that sets up a 'Column' layout. Inside the column, there's a 'Text' widget with the text 'Selamat Datang di Travelupa!' and a 'Spacer' widget. Below the spacer, there's another 'Text' widget with the text 'Solusi buat kamu yang lupa kemana-mana.'. The bottom status bar shows the time as 7:49 and the date as 16-Dec-23.

Preview of the app interface:

Selamat Datang di Travelupa!

Solusi buat kamu yang lupa kemana-mana.

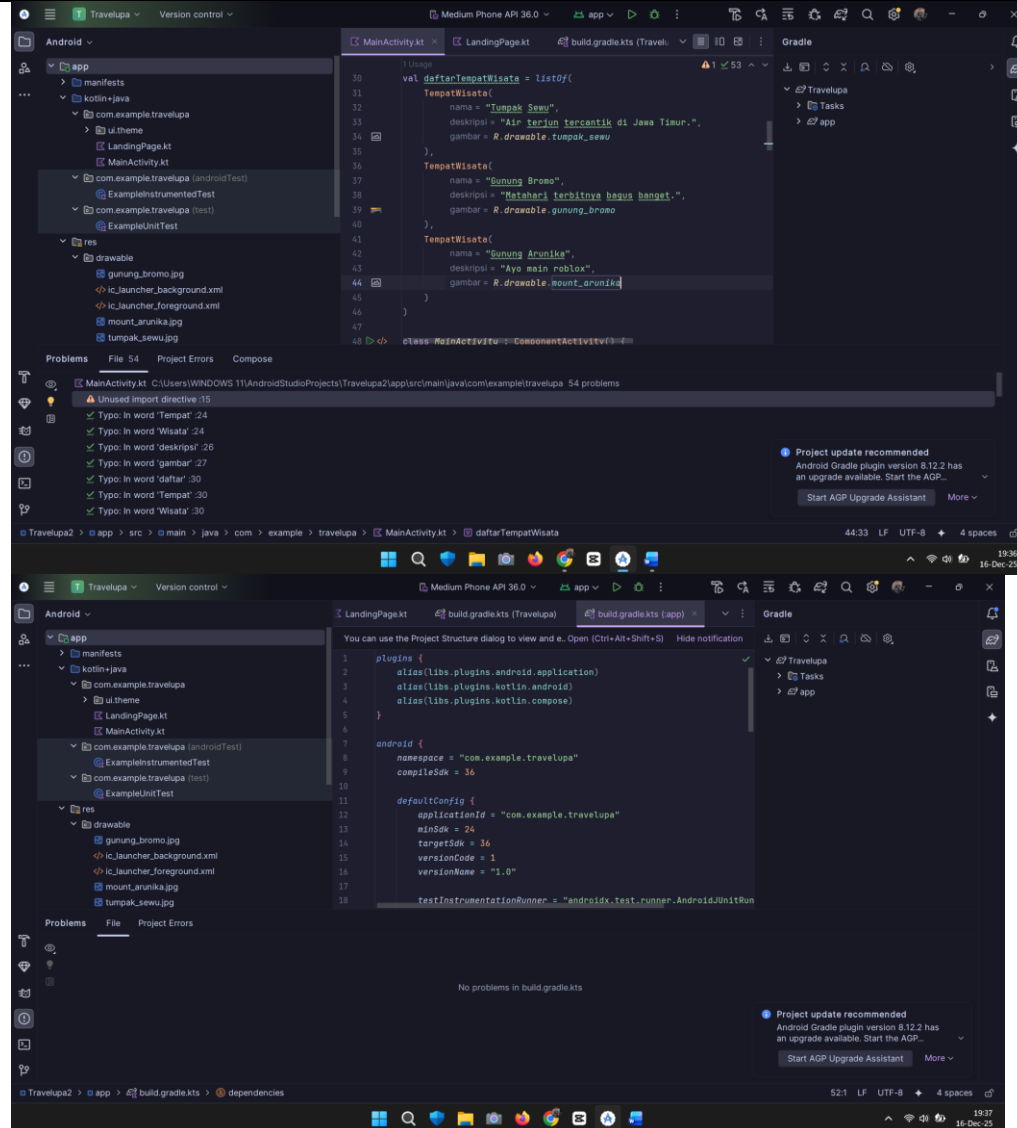
Mulai

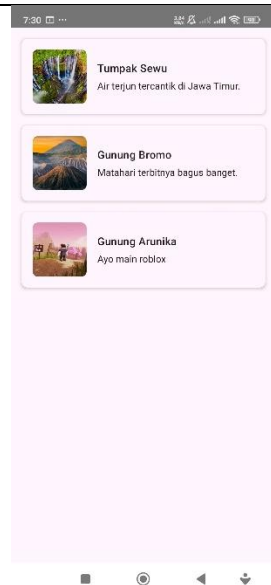
pada tahap awal ini, saya melakukan instalasi dan konfigurasi lingkungan kerja menggunakan Android Studio. Saya membuat *project* baru bernama **"Travelupa"** dengan konfigurasi *Minimum SDK API 24 (Android 7.0)* dan menggunakan *Build Configuration Language* berbasis Kotlin DSL agar sesuai dengan standar pengembangan Android modern.

Bab 2 Jetpack Compose 1

Implementasi dasar UI *Hello World* atau tampilan statis pada bab ini telah saya lebur dan kembangkan langsung menjadi antarmuka yang lebih kompleks pada Bab 3. Hal ini dilakukan untuk efisiensi penulisan kode dan langsung menerapkan struktur *layout* yang dinamis.

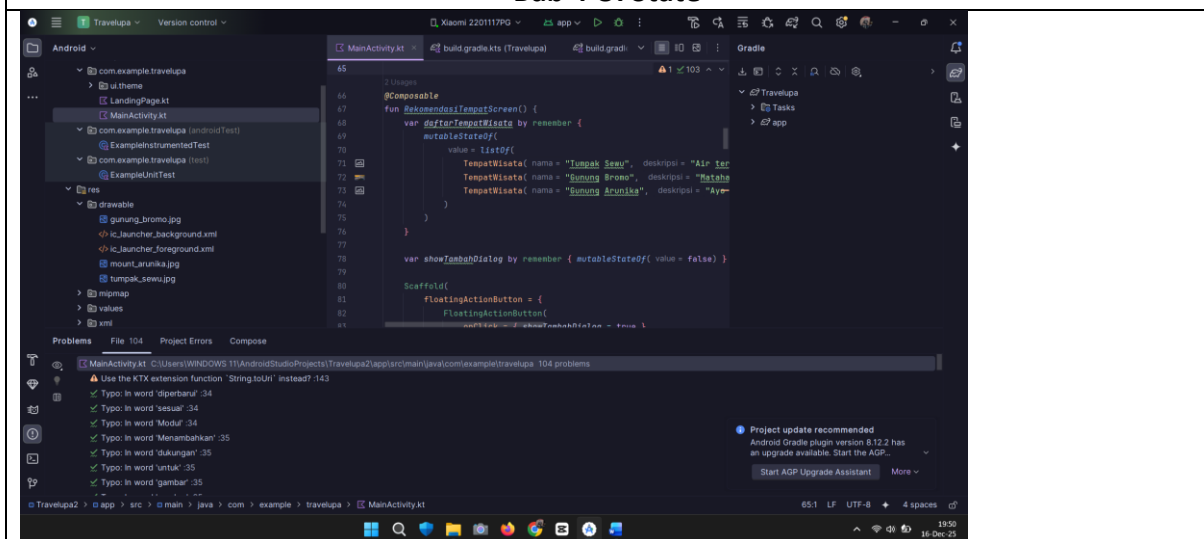
Bab 3 Jetpack Compose 2

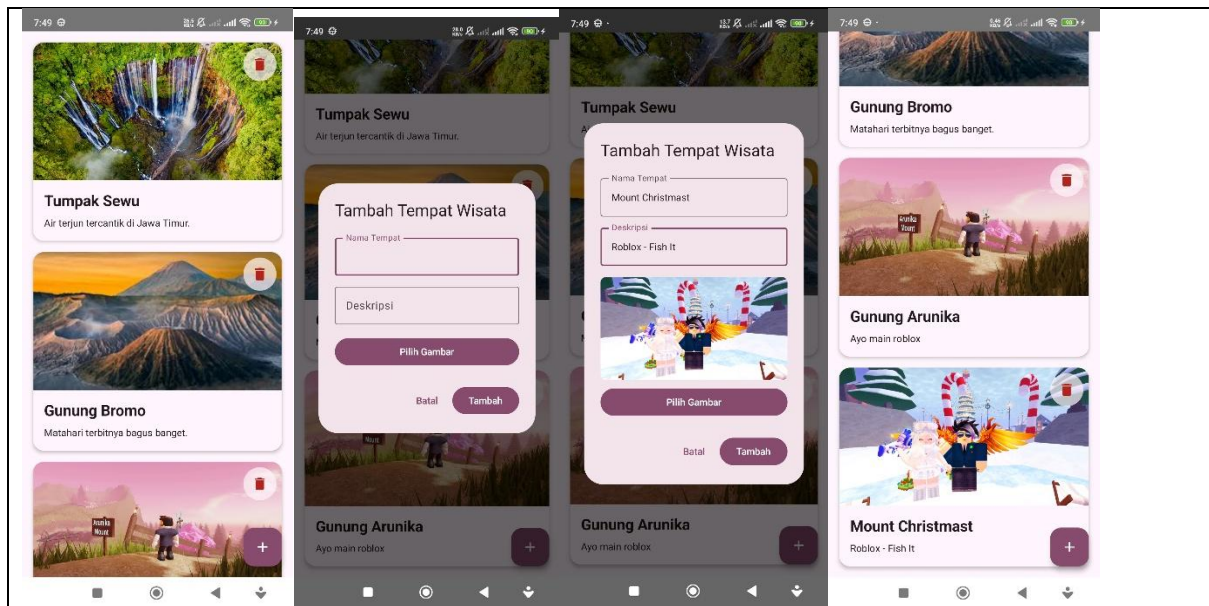




Pada bab ini, saya membangun tampilan utama aplikasi (*Home Screen*) menggunakan **Jetpack Compose**. Saya memanfaatkan komponen `LazyColumn` untuk membuat daftar tempat wisata yang dapat di-*scroll* secara efisien. Setiap *item* wisata ditampilkan menggunakan komponen `Card` yang membungkus `Image` dan `Text` agar tampilan terlihat rapi dan menarik.

Bab 4 UI State

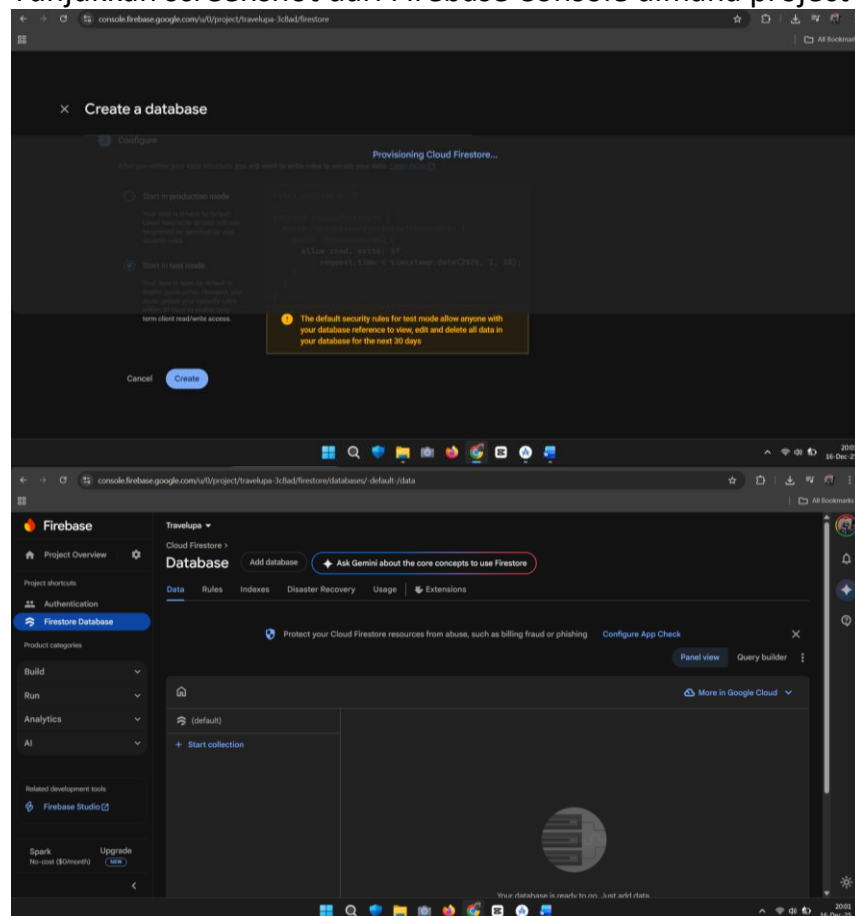


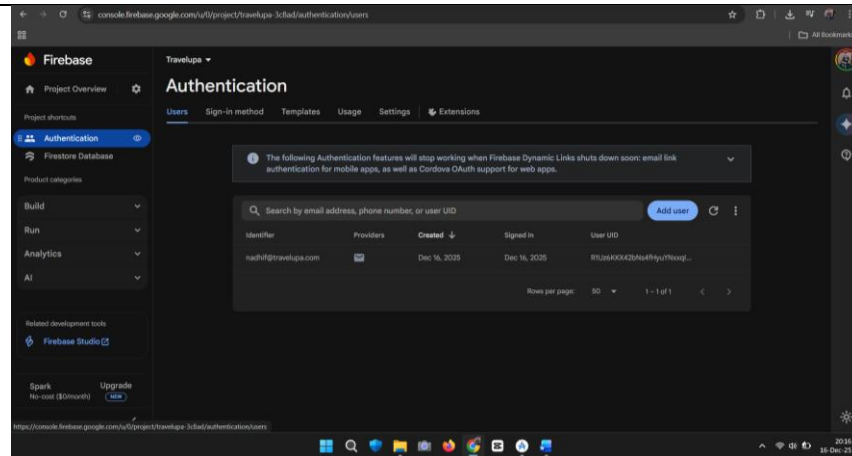


Di sini saya menerapkan konsep *State Management* agar aplikasi menjadi interaktif. Saya menggunakan `useState` dan `remember` untuk memantau perubahan data. Fitur yang saya tambahkan meliputi **Floating Action Button (FAB)** yang memunculkan *Dialog* input untuk menambah data wisata baru, serta tombol hapus pada setiap item *card* yang dapat memperbarui tampilan *list* secara *real-time*.

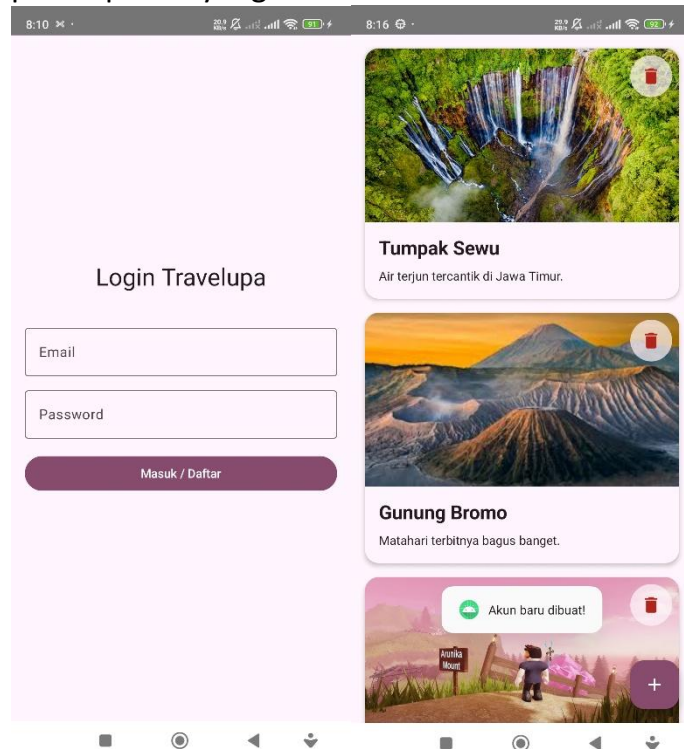
Bab 5 Firebase dan REST API

Tunjukkan screenshot dari Firebase Console dimana project dibuat



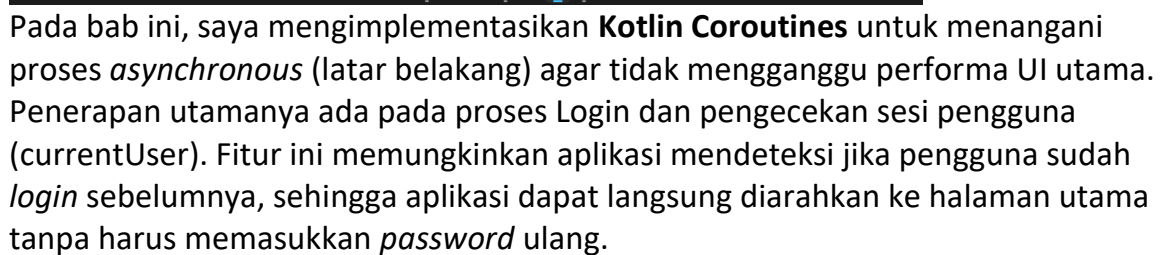


Tunjukkan screenshot dari UI tampilan yang menunjukkan penerapan Firebase pada aplikasi yang sudah dibuat



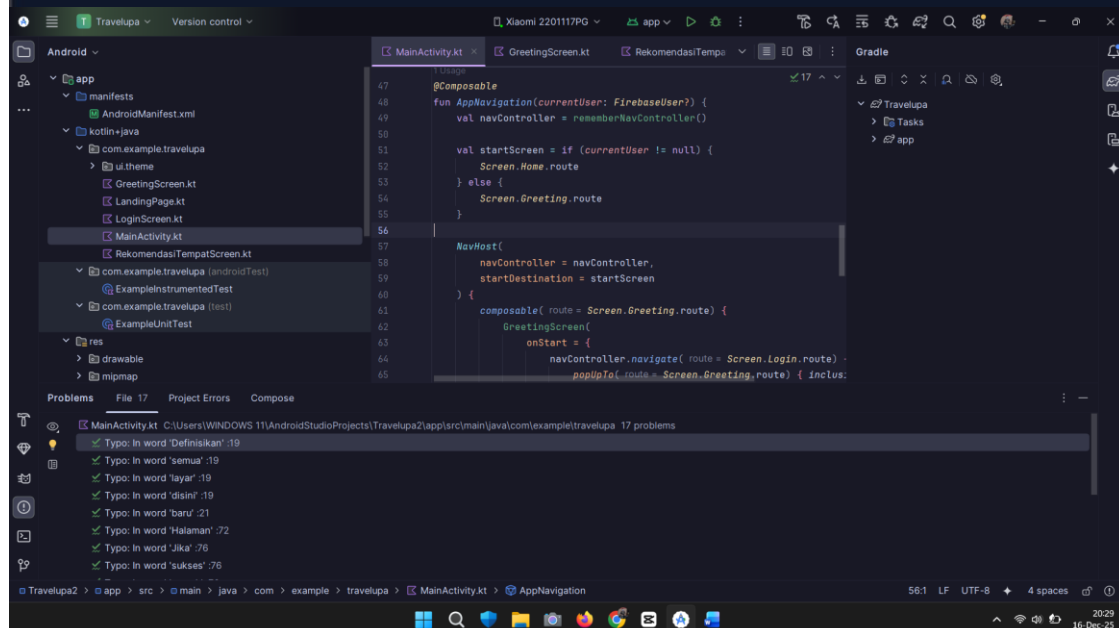
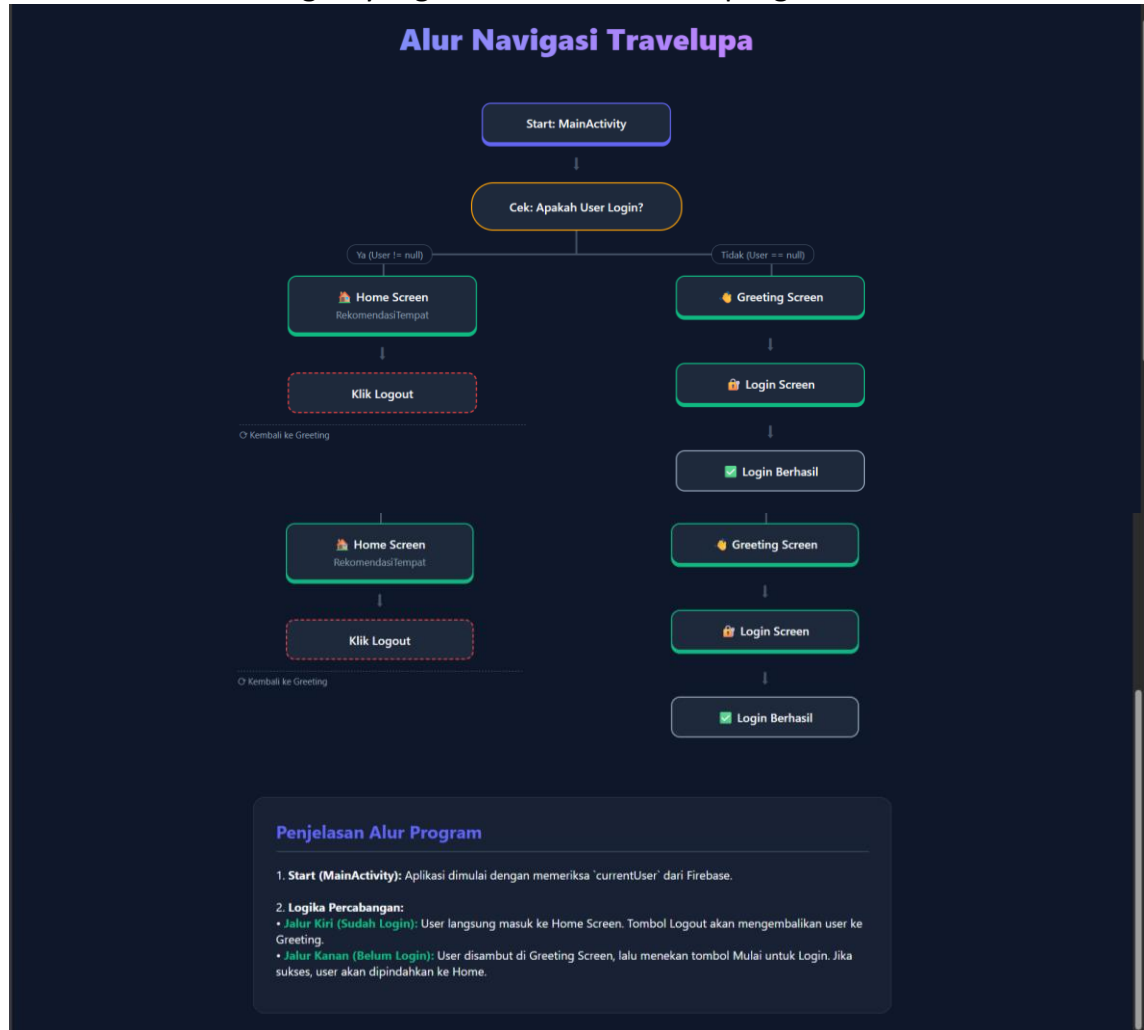
Saya menghubungkan aplikasi Travelupa dengan layanan **Google Firebase**. Langkah yang saya lakukan meliputi konfigurasi google-services.json, mengaktifkan fitur **Firebase Authentication** untuk membuat halaman Login pengguna, serta menyiapkan **Cloud Firestore** sebagai basis data *cloud* untuk menyimpan informasi nama dan deskripsi tempat wisata agar dapat diakses secara *online*.

Tunjukkan screenshot dari Android Studio yang menunjukkan kode yang dibuat untuk menunjukkan penerapan Kotlin Coroutines pada aplikasi yang sudah dibuat

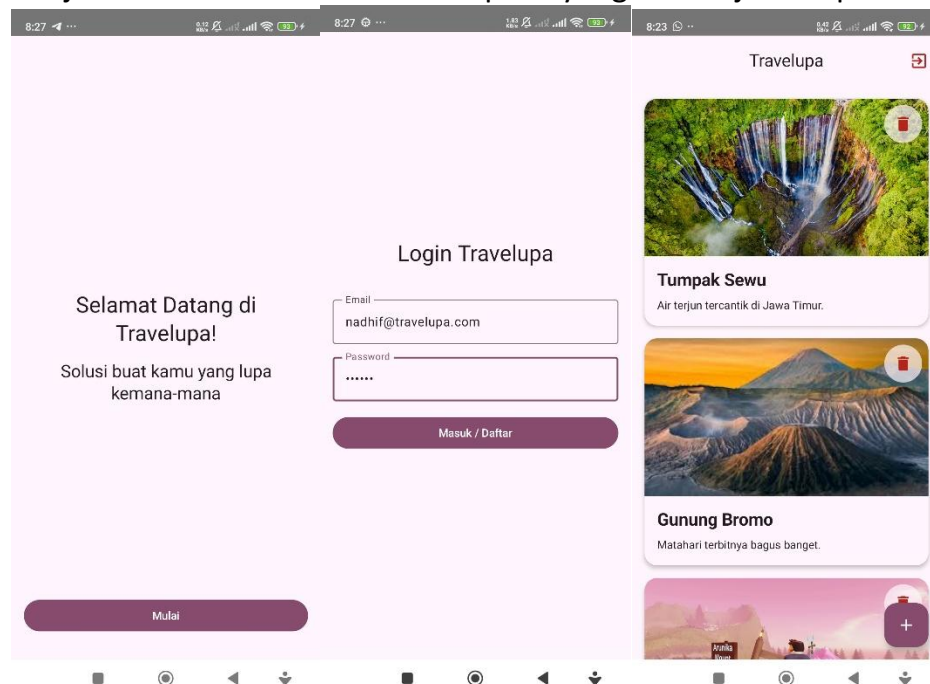


Bab 7 Jetpack Navigation

Gambarkan alur navigasi yang sudah dibuat dalam program



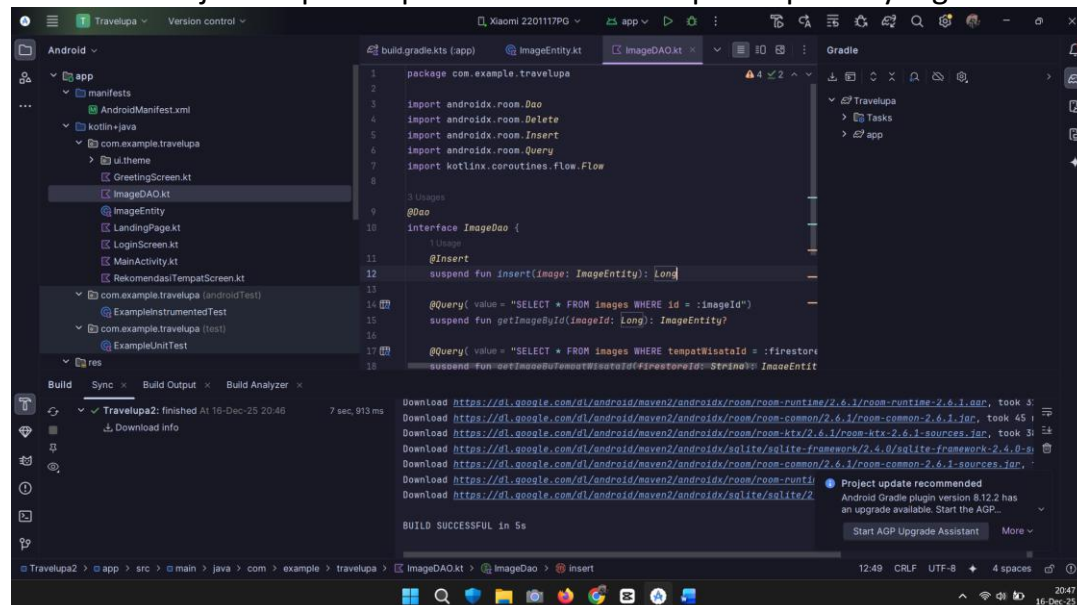
Tunjukkan screenshot dari UI tampilan yang menunjukkan penerapan Navigation

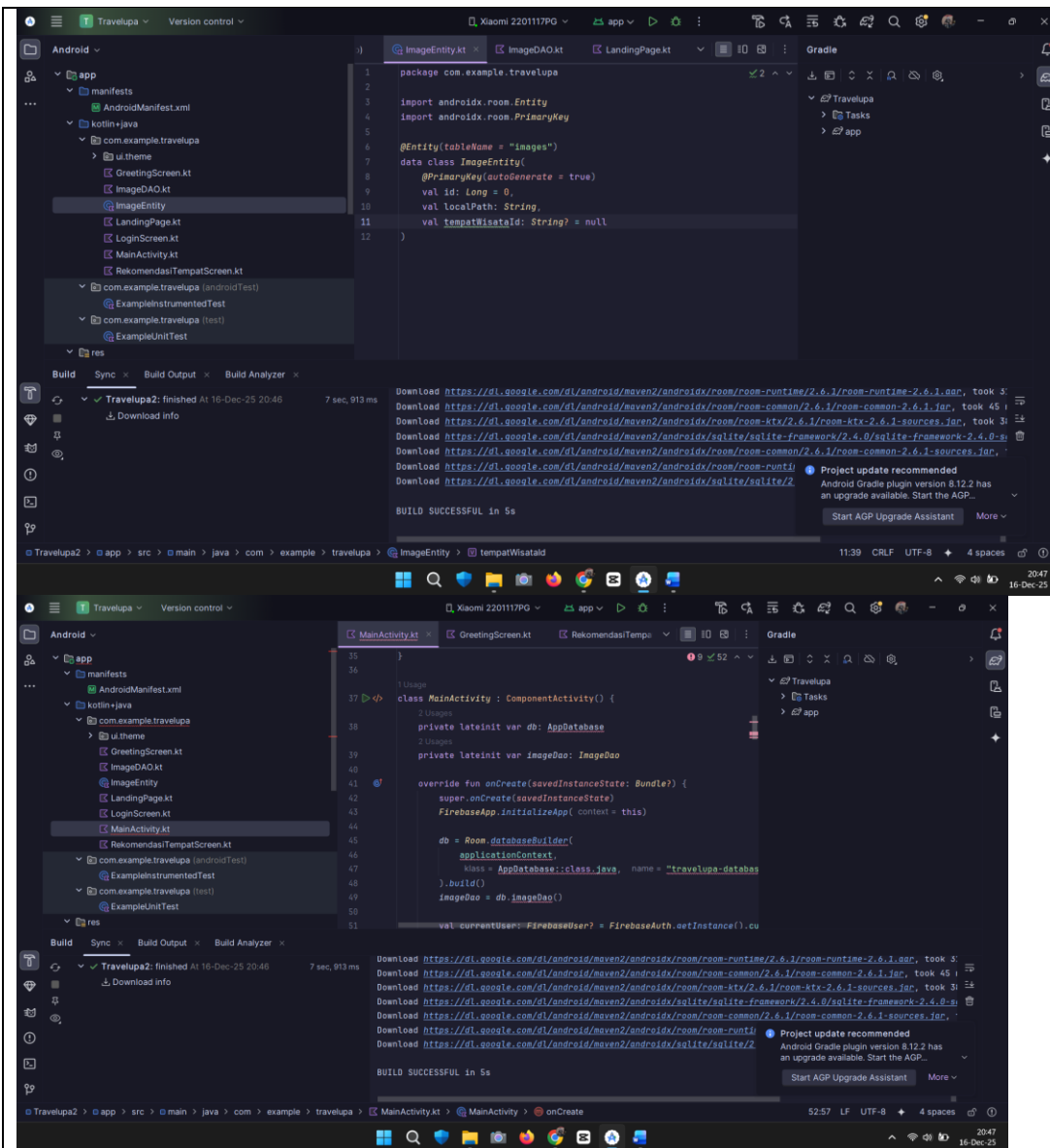


Saya menyusun alur navigasi antar layar menggunakan **Jetpack Navigation Component**. Saya membuat NavHost yang mengatur rute perpindahan dari **Greeting Screen** (Halaman Sapaan) -> **Login Screen** -> **Home Screen**. Saya juga menambahkan logika *PopUpTo* untuk membersihkan riwayat navigasi agar pengguna tidak kembali ke halaman login saat menekan tombol *back* di halaman utama.

Bab 8 Room Database

Tunjukkan screenshot dari Android Studio yang menunjukkan kode yang dibuat untuk menunjukkan penerapan Room Database pada aplikasi yang sudah dibuat

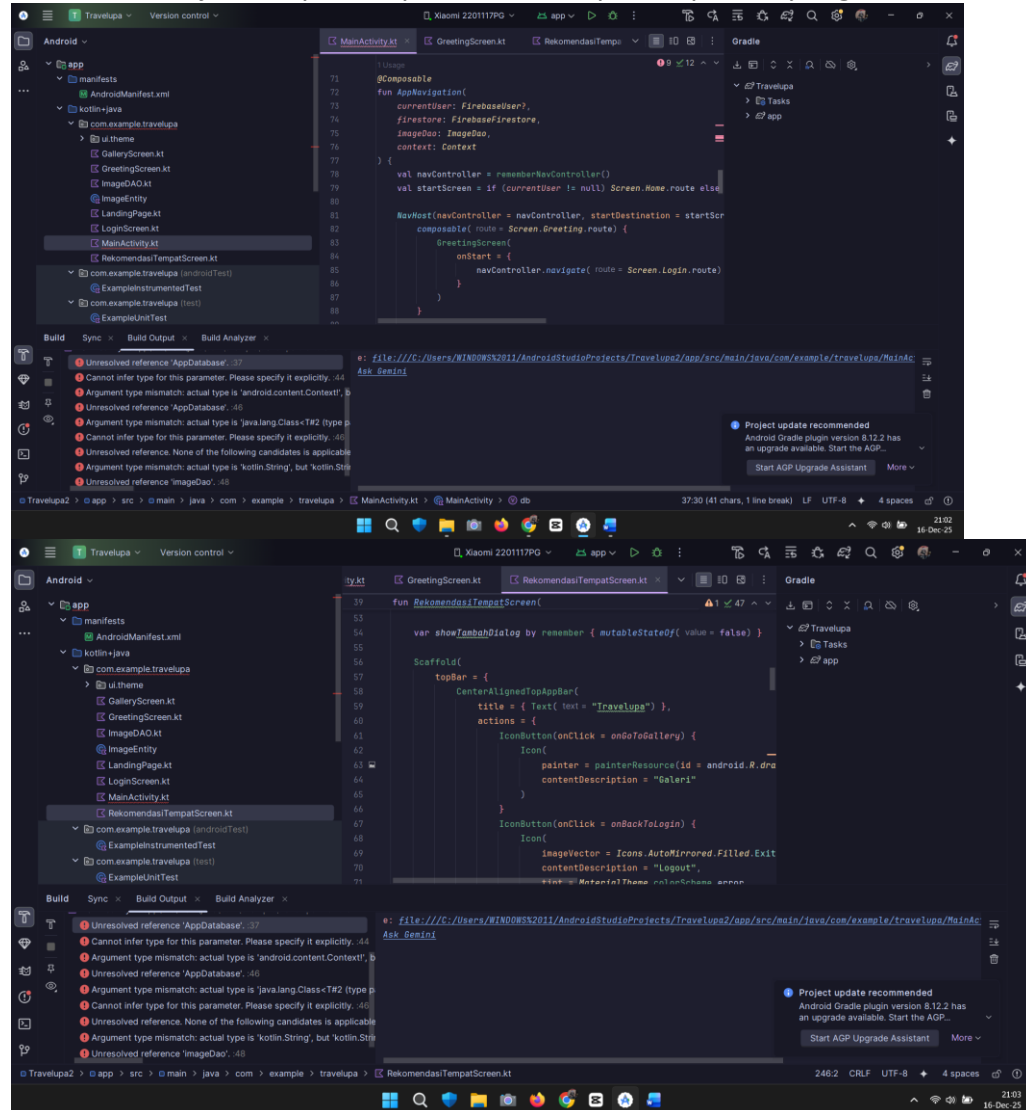




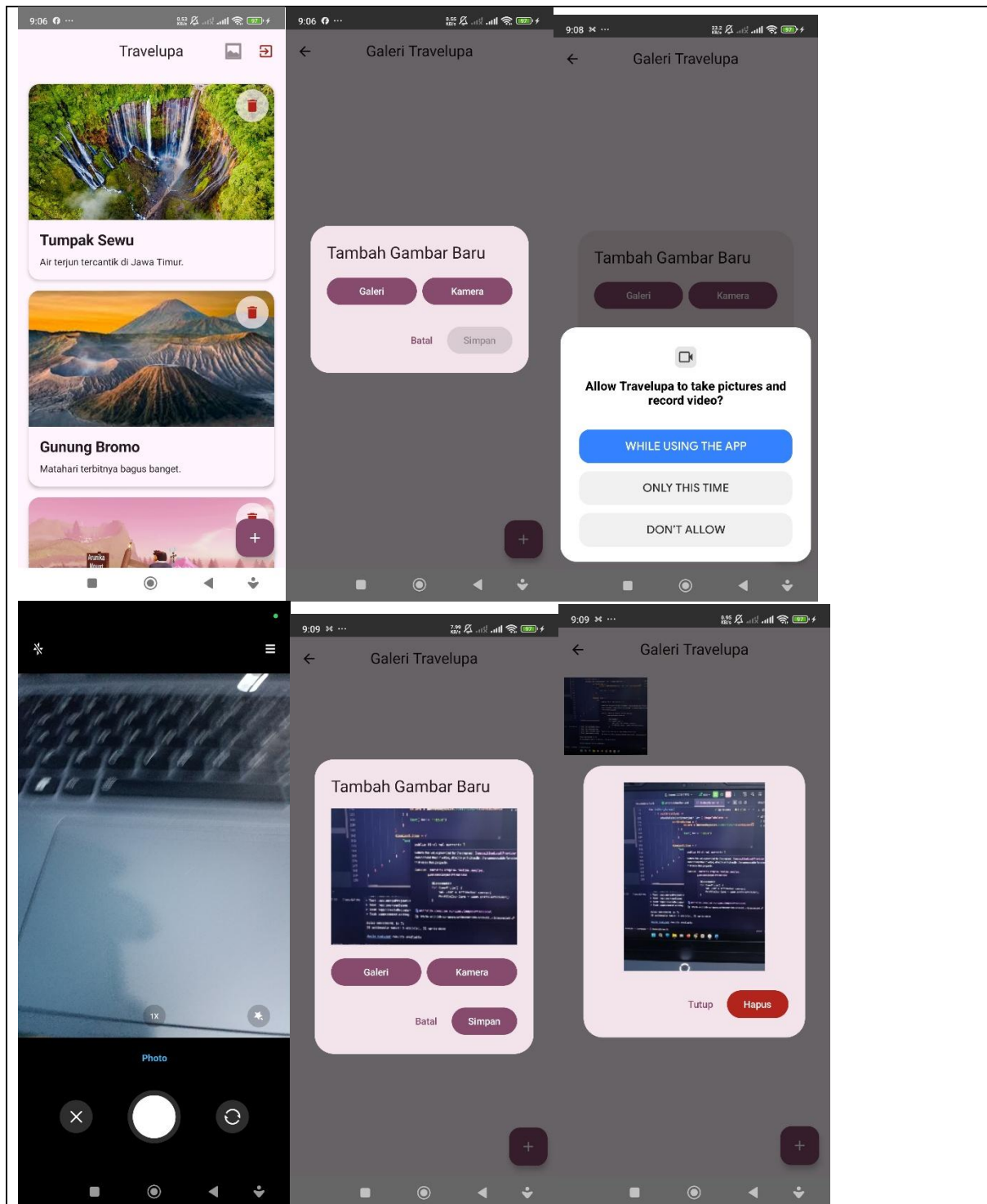
Untuk meningkatkan kemampuan aplikasi dalam menyimpan data gambar secara persisten (tetap ada meski aplikasi ditutup), saya mengimplementasikan **Room Database** (SQLite). Saya membuat struktur tabel (Entity), akses data (DAO), dan inisialisasi database (AppDatabase). Saya juga membuat logika sinkronisasi dimana gambar disimpan secara fisik di penyimpanan lokal perangkat, dan *path*-nya dicatat di Room serta disinkronkan ke Firestore.

Bab 9 CameraX

Tunjukkan screenshot dari Android Studio yang menunjukkan kode yang dibuat untuk menunjukkan penerapan CameraX pada aplikasi yang sudah dibuat



Tunjukkan screenshot dari UI tampilan yang menunjukkan penerapan CameraX



Pada bab terakhir ini, saya menambahkan fitur multimedia dengan mengintegrasikan akses kamera dan galeri. Saya membuat halaman baru yaitu **GalleryScreen** yang menampilkan *grid* foto-foto wisata. Saya menggunakan *ActivityResult Contracts* untuk meminta izin kamera (*Runtime Permission*) dan memungkinkan pengguna mengambil foto langsung dari aplikasi atau memilihnya dari galeri HP, yang kemudian tersimpan otomatis ke dalam database aplikasi.

| Link Project File |
|---|
| <p>Tuliskan alamat link dari Repository project yang sudah dibuat (GitHub, Gitlab, dll)</p> <p>Pastikan dapat dibuka secara publik untuk penilaian</p> <p>https://github.com/nadhif-royal/Travelupa/</p> |