

LAPORAN PRAKTIKUM MOBILE PROGRAMMING

(Akses Lokasi dengan GPS di Flutter)



Nama : Andini Aulia Putri

NIM : 230605110059

Kelas : E

Tanggal : 06 November 2025

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
GANJIL 2025/2026**

I. Tujuan

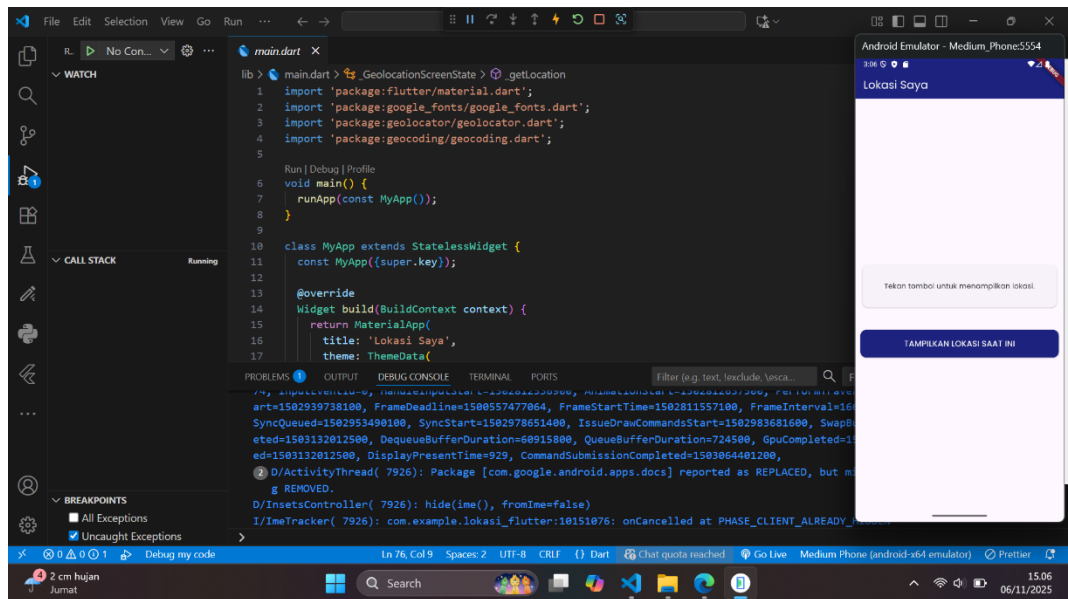
1. Memahami konsep pengambilan lokasi perangkat menggunakan GPS pada Flutter.
2. Mengakses koordinat latitude dan longitude perangkat secara realtime.
3. Mengubah koordinat menjadi informasi alamat (reverse geocoding) seperti kelurahan, kecamatan dan kota.
4. Menangani izin (permission) lokasi pada Android.
5. Menampilkan informasi lokasi pada tampilan aplikasi Flutter secara interaktif dan responsif.

II. Langkah Kerja

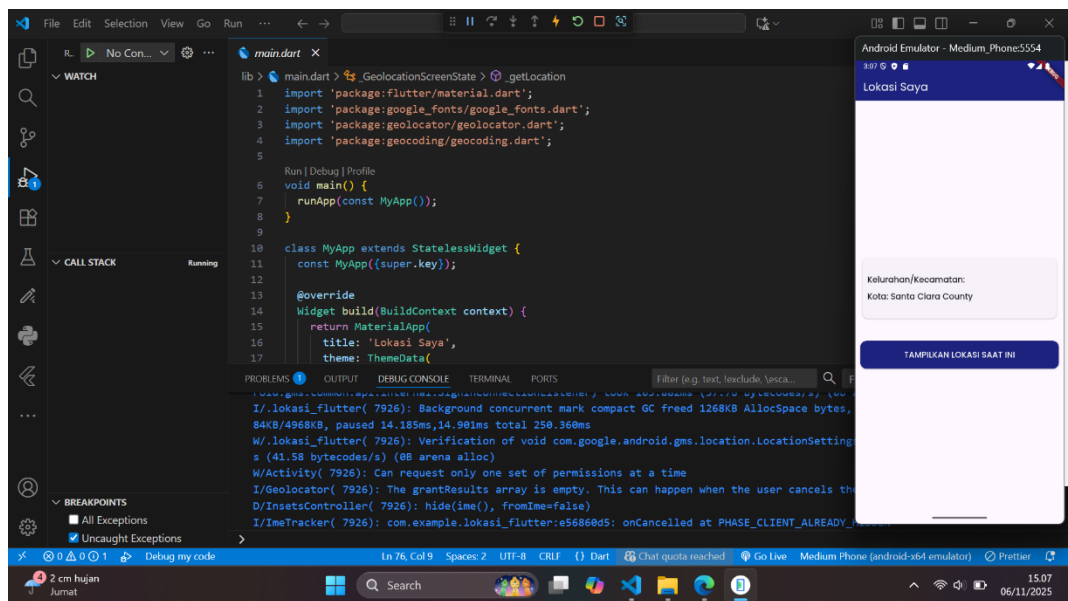
1. Buat proyek baru dengan nama lokasi_flutter.
2. Buat kelas MyApp menggunakan MaterialApp dan jadikan LocationScreen sebagai halaman utama aplikasi.
3. Buat kelas LocationScreen sebagai turunan StatefulWidget dan tambahkan variabel untuk menyimpan data lokasi, status loading, dan pesan error.
4. Buat fungsi _getLocation() di dalam kelas _GeolocationScreenState sebelum method build().
5. Buat tampilan UI di method build() menggunakan Scaffold, Card, dan ElevatedButton. Gunakan logika _isLoading, _errorMessage, dan data lokasi untuk menampilkan konten yang sesuai.
6. Jalankan programnya.

III. Screenshot Hasil

1. Screenshot 1: Halaman awal



2. Screenshot 2: Halamn menampilkan lokasi



IV. Kesimpulan

1. Penggunaan Paket Geolocator dan Geocoding:

- Aplikasi Flutter dapat menampilkan lokasi perangkat secara real-time dengan memanfaatkan paket **Geolocator** untuk mengambil koordinat GPS

dan **Geocoding** untuk mengubah koordinat tersebut menjadi informasi alamat seperti kecamatan dan kota.

2. **Pengelolaan Izin Lokasi:**

- Untuk mengakses lokasi perangkat, aplikasi harus meminta izin dari pengguna. Penambahan izin **ACCESS_FINE_LOCATION** dan **ACCESS_COARSE_LOCATION** pada **AndroidManifest.xml** memastikan aplikasi bisa mendapatkan akses GPS dengan benar.

3. **Pembuatan Antarmuka Interaktif:**

- Widget **Scaffold**, **Card**, dan **ElevatedButton** digunakan untuk membangun antarmuka yang rapi dan interaktif. Tampilan dapat menyesuaikan kondisi aplikasi, seperti saat memuat data, terjadi kesalahan, atau data lokasi sudah tersedia.

4. **Fungsi `_getLocation()`:**

- Fungsi ini menggabungkan pengecekan status layanan lokasi, permintaan izin, pengambilan posisi, dan reverse geocoding. Aplikasi menangani error dengan baik, sehingga menjadi lebih stabil dan aman.

5. **Penggunaan `setState()` untuk Pembaruan Tampilan:**

- Dengan **setState()**, aplikasi secara otomatis memperbarui tampilan ketika data berubah. Setiap perubahan pada status **_isLoading**, **_kecamatan**, **_kota**, atau **_errorMessage** langsung memperbarui tampilan UI, sehingga informasi yang ditampilkan selalu terbaru dan sesuai dengan kondisi saat ini.