

LAPORAN TEORI MOBILE PROGRAMMING
MODUL 12



Nama : Erai Bagusalim
NIM : 240605110088
Kelas : Mobile Programming B
Tanggal : 1 November 2025

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

i. Tujuan

1. Memahami cara pengambilan lokasi perangkat menggunakan GPS di Flutter.
2. Penerapan paket geolocator dan geocoding.
3. Penanganan izin lokasi dan error.
4. Pembuatan antarmuka interaktif yang menampilkan informasi Lokasi secara real time.

ii. Langkah Kerja

1. Penambahan izin lokasi di AndroidManifest.xml.
2. Penambahan dependensi pada file pubspec.yaml.
3. Pembuatan UI menggunakan Scaffold, Card, dan ElevatedButton.
4. Implementasi fungsi `_getLocation()` beserta logika pengecekan layanan lokasi, permintaan izin, pengambilan posisi, dan reverse geocoding.
5. Penggunaan `setState()` untuk memperbarui tampilan berdasarkan perubahan data lokasi.

iii. Screenshot Hasil

a. Kode Program

Main.dart

```
● ● ●
1 import 'package:flutter/material.dart';
2 import 'package:google_fonts/google_fonts.dart';
3 import 'package:geolocator/geolocator.dart';
4 import 'package:geocoding/geocoding.dart';
5
6 void main() {
7   runApp(const MyApp());
8 }
9
10 class MyApp extends StatelessWidget {
11   const MyApp({super.key});
12
13   @override
14   Widget build(BuildContext context) {
15     return MaterialApp(
16       title: 'Lokasi Saya',
17       theme: ThemeData(
18         primarySwatch: Colors.indigo,
19         textTheme: GoogleFonts.poppinsTextTheme(
20           Theme.of(context).textTheme,
21         ),
22       ),
23       home: const GeolocationScreen(),
24     );
25   }
26 }
27
28 class GeolocationScreen extends StatefulWidget {
29   const GeolocationScreen({super.key});
30
31   @override
32   State<GeolocationScreen> createState() => _GeolocationScreenState();
33 }
34
35 class _GeolocationScreenState extends State<GeolocationScreen> {
36   String? _kecamatan;
37   String? _kota;
38   bool _isLoading = false;
39   String? _errorMessage;
```

```
● ● ●
1 Future<void> _getLocation() async {
2   setState(() {
3     _isLoading = true;
4     _errorMessage = null;
5     _kecamatan = null;
6     _kota = null;
7   });
8
9   try {
10     bool serviceEnabled = await Geolocator.isLocationServiceEnabled();
11     if (!serviceEnabled) {
12       throw Exception('Layanan lokasi tidak aktif. Mohon aktifkan GPS.');
13     }
14
15     LocationPermission permission = await Geolocator.checkPermission();
16     if (permission == LocationPermission.denied) {
17       permission = await Geolocator.requestPermission();
18       if (permission == LocationPermission.denied) {
19         throw Exception('Izin lokasi ditolak oleh pengguna.');
20       }
21     }
22     if (permission == LocationPermission.deniedForever) {
23       throw Exception('Izin lokasi ditolak permanen. Anda harus mengubahnya di pengaturan aplikasi.');
24     }
25
26     Position position = await Geolocator.getCurrentPosition(desiredAccuracy: LocationAccuracy.high);
27
28     List<Placemark> placemarks = await placemarkFromCoordinates(position.latitude, position.longitude);
29
30     if (placemarks.isNotEmpty) {
31       Placemark place = placemarks[0];
32       setState(() {
33         _kecamatan = place.subLocality;
34         _kota = place.subAdministrativeArea;
35       });
36     } else {
37       throw Exception('Tidak dapat menemukan informasi alamat.');
38     }
39   } catch (e) {
40     setState(() {
41       _errorMessage = e.toString().replaceAll("Exception: ", " ");
42     });
43   } finally {
44     setState(() {
45       _isLoading = false;
46     });
47   }
48 }
```

```
1  @override
2  Widget build(BuildContext context) {
3    return Scaffold(
4      appBar: AppBar(
5        title: const Text(
6          'Lokasi Saya',
7          style: TextStyle(
8            color: Colors.white,
9            ),
10         ),
11        backgroundColor: const Color(0xFF1A237E),
12      ),
13      body: Padding(
14        padding: const EdgeInsets.all(10.0),
15        child: Column(
16          mainAxisAlignment: MainAxisAlignment.center,
17          crossAxisAlignment: CrossAxisAlignment.stretch,
18          children: [
19            Card(
20              elevation: 4,
21              shape: RoundedRectangleBorder(
22                borderRadius: BorderRadius.circular(12)),
23              child: Container(
24                padding: const EdgeInsets.symmetric(vertical: 32, horizontal: 10),
25                child: _isLoading
26                  ? const Center(child: CircularProgressIndicator())
27                  : (_errorMessage != null)
28                    ? Text(_errorMessage!,
29                      textAlign: TextAlign.center,
30                      style: const TextStyle(color: Colors.red, fontSize: 16))
31                    : (_kecamatan == null && _kota == null)
32                      ? const Text(
33                        'Tekan Tombol untuk menampilkan lokasi Anda.',
34                        textAlign: TextAlign.center,
35                      )
36                      : Column(
37                        crossAxisAlignment: CrossAxisAlignment.start,
38                        children: [
39                          Text(
40                            'Kelurahan/Kecamatan: $_kecamatan ',
41                            style: const TextStyle(
42                              fontSize: 16,
43                              fontWeight: FontWeight.bold
44                            ),
45                          ),
46                          const SizedBox(height: 8),
47                          Text(
48                            'Kota: $_kota ',
49                            style: const TextStyle(
50                              fontSize: 16,
51                              fontWeight: FontWeight.bold
52                            ),
53                          ),
54                        ],
55                      ),
56                    ),
57                  const SizedBox(height: 40),
58                  ElevatedButton(
59                    onPressed: _isLoading ? null : _getLocation,
60                    style: ElevatedButton.styleFrom(
61                      backgroundColor: const Color(0xFF1A237E),
62                      padding: const EdgeInsets.symmetric(vertical: 16),
63                      shape: RoundedRectangleBorder(
64                        borderRadius: BorderRadius.circular(8),
65                      ),
66                      elevation: 5.0,
67                    ),
68                    child: const Text(
69                      'TAMPILKAN LOKASI SAATINI',
70                      style: TextStyle(
71                        fontSize: 16,
72                        color: Colors.white,
73                      ),
74                    ),
75                  ),
76                ],
77              ),
78            ),
79          );
80        );
81      );
82    }

```

b. Penjelasan Kode Program

- Cara Kerja `_getLocation()` dalam Menampilkan Lokasi

Fungsi `_getLocation()` adalah pusat proses pengambilan lokasi pada aplikasi ini. Fungsi ini dijalankan saat pengguna menekan tombol “TAMPILKAN LOKASI SAAT INI”.

Saat fungsi ini dipanggil, hal pertama yang dilakukan adalah mengubah kondisi aplikasi menjadi loading. Ini ditandai dengan:

- `_isLoading = true` → indikator loading muncul
- `_errorMessage = null` → pesan error lama dihapus
- `_kecamatan` dan `_kota` direset agar tidak menampilkan data lama

Selanjutnya, aplikasi akan:

- Mengecek apakah GPS aktif
- Mengecek dan meminta izin Lokasi
- Mengambil koordinat posisi pengguna
- Mengubah koordinat menjadi alamat (kecamatan dan kota) menggunakan geocoding

Jika semua proses berhasil, maka nilai `_kecamatan` dan `_kota` akan langsung diisi dengan data yang didapat. Namun jika terjadi kesalahan (GPS mati, izin ditolak, atau alamat tidak ditemukan), maka pesan kesalahan akan disimpan dalam `_errorMessage`.

Di akhir proses, `_isLoading` dikembalikan menjadi `false`, sehingga loading otomatis berhenti.

- Peran Variabel dalam Menampilkan Data Secara Dinamis

1. `_isLoading`

Variabel ini berfungsi sebagai penanda proses sedang berlangsung atau tidak.

- Saat bernilai `true` → indikator loading (`CircularProgressIndicator`) ditampilkan

- Saat bernilai false → hasil lokasi atau pesan ditampilkan
- Tombol juga dibuat tidak bisa ditekan saat loading agar tidak terjadi proses ganda

2. _kecamatan

Variabel ini menyimpan nama kelurahan/kecamatan hasil dari lokasi pengguna.

- Jika masih null → layar menampilkan teks petunjuk
- Jika sudah terisi → nama kecamatan langsung muncul di layar

3. _kota

Variabel ini menyimpan nama kota atau kabupaten.

- Ditampilkan bersamaan dengan kecamatan setelah lokasi berhasil didapat

4. _errorMessage

Variabel ini digunakan untuk menyimpan pesan kesalahan.

- Jika terjadi error → teks merah akan muncul di dalam Card
- Jika tidak terjadi kesalahan → variabel ini tetap null dan tidak ditampilkan

- **Bagaimana setState() Memperbarui Tampilan Secara Otomatis**

setState() berfungsi sebagai penghubung antara data dan tampilan.

Setiap kali nilai variabel diubah di dalam setState():

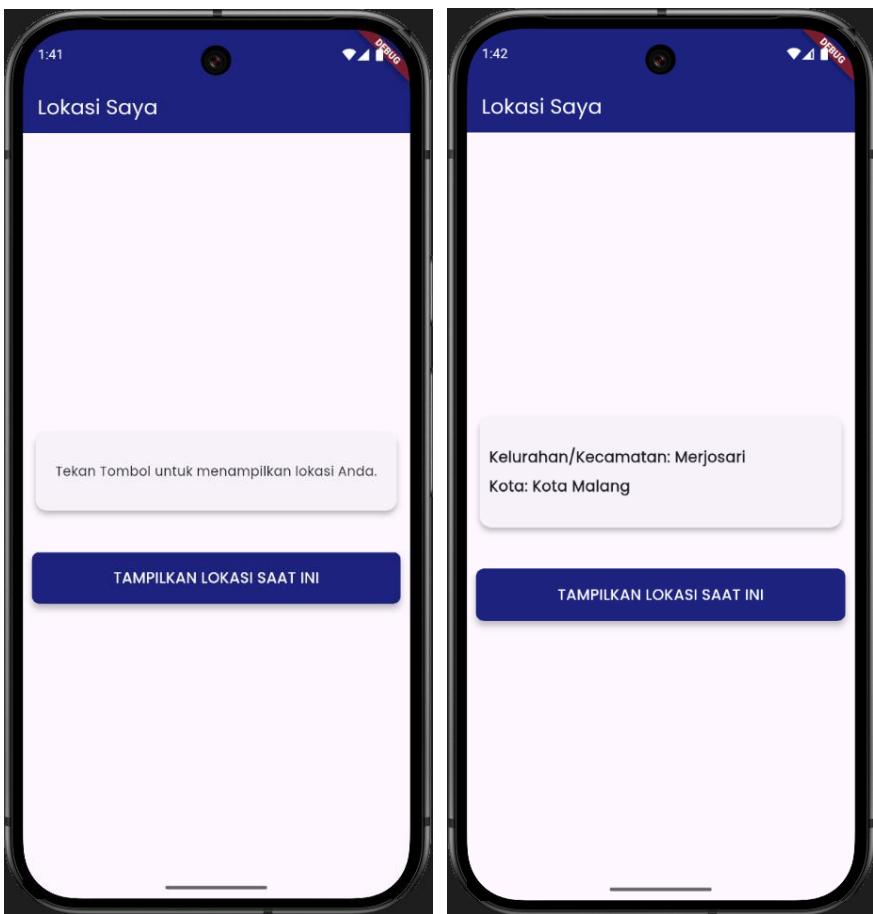
1. Data diperbarui
2. Flutter otomatis memanggil ulang build()
3. Tampilan langsung menyesuaikan kondisi terbaru

Contohnya:

- Saat _isLoading berubah → loading muncul atau hilang
- Saat _kecamatan dan _kota terisi → hasil lokasi langsung tampil
- Saat _errorMessage terisi → pesan error langsung muncul

Semua perubahan ini terjadi otomatis tanpa perlu refresh manual.

Output :



iv. Kesimpulan

Berdasarkan hasil penggerjaan, aplikasi berhasil menampilkan informasi lokasi pengguna secara real-time dengan memanfaatkan sensor GPS dan proses geocoding. Pembaruan tampilan dilakukan secara otomatis menggunakan `setState()`, sehingga setiap perubahan data seperti status loading, hasil lokasi, maupun pesan error langsung terlihat pada antarmuka tanpa perlu refresh manual. Selain itu, aplikasi juga telah menerapkan penanganan izin lokasi dan error dengan baik, sehingga dapat memberikan notifikasi yang jelas kepada pengguna ketika GPS tidak aktif, izin ditolak, atau data lokasi gagal diperoleh. Dengan demikian, aplikasi ini sudah bekerja secara dinamis, responsif, dan user-friendly.