LAPORAN TEORI MOBILE PROGRAMMING
MODUL 13

Nama : Firman Fadilah Noor
NIM : 240605110083
Kelas : B
Tanggal : 17 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 real-time.
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 responsive.

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

- Models/
  Post_model.dart

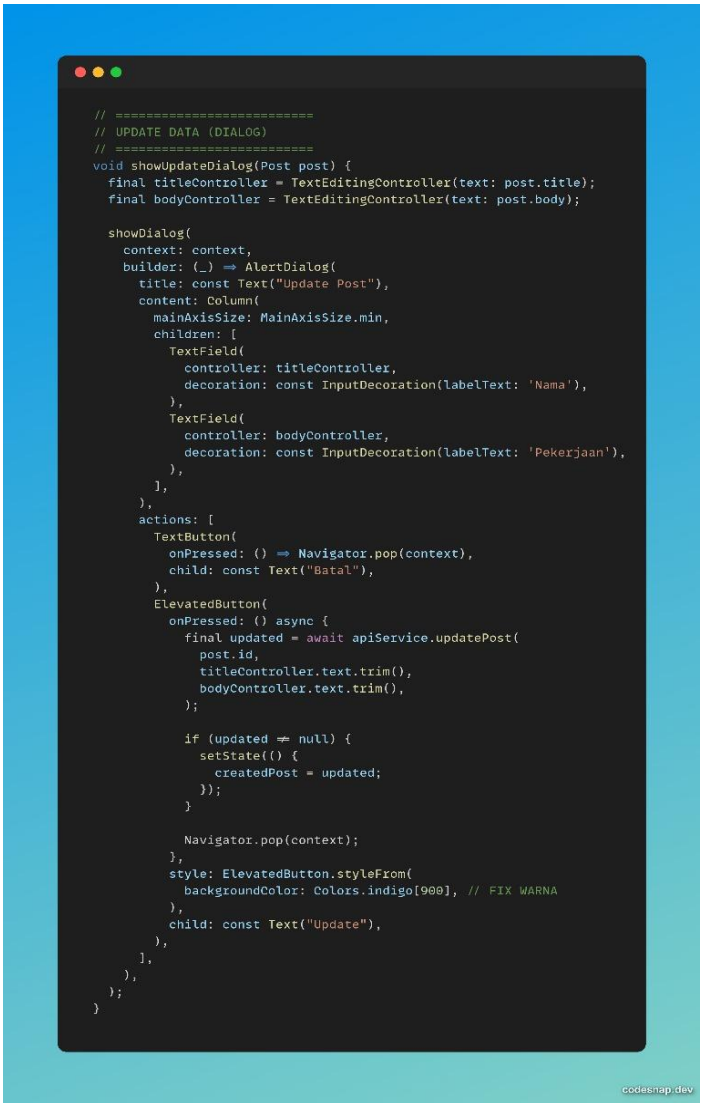
```dart
class Post {
  final int id;
  final int userId;
  final String title;
  final String body;
  final DateTime createdAt;

  Post({
    required this.id,
    required this.userId,
    required this.title,
    required this.body,
    required this.createdAt,
  });

  factory Post.fromJson(Map<String, dynamic> json) {
    return Post(
      id: json['id'],
      userId: json['userId'] ?? 1,
      title: json['title'],
      body: json['body'],
      createdAt: DateTime.now(),
    );
  }
}
```


- Services/
  Api_service.dart

```dart
import 'dart:convert';
import 'package:http/http.dart' as http;
import '../models/post_model.dart';

class ApiService {
  final String baseUrl = "https://jsonplaceholder.typicode.com/posts";

  // CREATE
  Future<Post?> createPost(String title, String body) async {
    try {
      final response = await http.post(
        Uri.parse(baseUrl),
        headers: {"Content-Type": "application/json"},
        body: jsonEncode({"title": title, "body": body, "userId": 1}),
      );

      if (response.statusCode == 201) {
        return Post.fromJson(jsonDecode(response.body));
      }
    } catch (e) {
      return null;
    }
    return null;
  }

  // UPDATE
  Future<Post?> updatePost(int id, String title, String body) async {
    try {
      final response = await http.put(
        Uri.parse("$baseUrl/1"),
        headers: {"Content-Type": "application/json"},
        body: jsonEncode({
          "id": 1,
          "title": title,
          "body": body,
          "userId": 1,
        }),
      );

      if (response.statusCode == 200) {
        return Post.fromJson(jsonDecode(response.body));
      }
    } catch (e) {
      return null;
    }
    return null;
  }

  // DELETE
  Future<bool> deletePost(int id) async {
    try {
      final response = await http.delete(Uri.parse("$baseUrl/$id"));
      return response.statusCode == 200;
    } catch (e) {
      return false;
    }
  }
}
```

- Pages/
  Home_page.dart

```dart
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:intl/intl.dart';

import '../models/post_model.dart';
import '../service/api_service.dart';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class  HomePageState extends State<HomePage> {
  final ApiService apiService = ApiService();
  Post? createdPost;

  // ========================
  // TAMBAH DATA (DIALOG)
  // ========================
  void showAddDialog() {
    final titleController = TextEditingController();
    final bodyController = TextEditingController();

    showDialog(
      context: context,
      builder: ( ) => AlertDialog(
        title: const Text("Tambah Post"),
        content: Column(
          mainAxisSize: MainAxisSize.min,
          children: [
            TextField(
              controller: titleController,
              decoration: const InputDecoration(labelText: 'Nama'),
            ),
            TextField(
              controller: bodyController,
              decoration: const InputDecoration(labelText: 'Pekerjaan'),
            ),
          ],
        ),
        actions: [
          TextButton(
            onPressed: () => Navigator.pop(context),
            child: const Text("Batal"),
          ),
          ElevatedButton(
            onPressed: () async {
              final title = titleController.text.trim();
              final body = bodyController.text.trim();

              if (title.isEmpty || body.isEmpty) return;

              final post = await apiService.createPost(title, body);

              if (post != null) {
                setState(() {
                  createdPost = post;
                });
              }

              Navigator.pop(context);
            },
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.indigo[900], // FIX WARNA
            ),
            child: const Text("Simpan"),
          ),
        ],
      ),
    );
  }
```

```dart
  // ========================
  // UPDATE DATA (DIALOG)
  // ========================
  void showUpdateDialog(Post post) {
    final titleController = TextEditingController(text: post.title);
    final bodyController = TextEditingController(text: post.body);

    showDialog(
      context: context,
      builder: (_) => AlertDialog(
        title: const Text("Update Post"),
        content: Column(
          mainAxisSize: MainAxisSize.min,
          children: [
            TextField(
              controller: titleController,
              decoration: const InputDecoration(labelText: 'Nama'),
            ),
            TextField(
              controller: bodyController,
              decoration: const InputDecoration(labelText: 'Pekerjaan'),
            ),
          ],
        ),
        actions: [
          TextButton(
            onPressed: () => Navigator.pop(context),
            child: const Text("Batal"),
          ),
          ElevatedButton(
            onPressed: () async {
              final updated = await apiService.updatePost(
                post.id,
                titleController.text.trim(),
                bodyController.text.trim(),
              );

              if (updated != null) {
                setState(() {
                  createdPost = updated;
                });
              }

              Navigator.pop(context);
            },
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.indigo[900], // FIX WARNA
            ),
            child: const Text("Update"),
          ),
        ],
      ),
    );
  }
```

```dart
  // ==========================
  // UI
  // ==========================
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Demo CRUD Client-Server"),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: showAddDialog,
        backgroundColor: Colors.indigo[900], // FIX WARNA FAB
        child: const Icon(Icons.add),
      ),
      body: createdPost == null
          ? const Center(
        child: Text(
          "Belum ada data",
          style: TextStyle(fontSize: 18),
        ),
      )
          : ListView(
        padding: const EdgeInsets.all(16),
        children: [
          Card(
            elevation: 8,
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(16),
            ),
            child: Padding(
              padding: const EdgeInsets.all(16),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  // TITLE + MENU
                  Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                      Text(
                        createdPost!.title,
                        style: GoogleFonts.montserrat(
                          fontSize: 22,
                          fontWeight: FontWeight.bold,
                        ),
                      ),
                      PopupMenuButton<String>(
                        itemBuilder: (_) => const [
                          PopupMenuItem(
                            value: "update",
                            child: Text("Update"),
                          ),
                          PopupMenuItem(
                            value: "delete",
                            child: Text("Delete"),
                          ),
                        ],
                        onSelected: (value) {
                          if (value == "update") {
                            showUpdateDialog(createdPost!);
                          } else {
                            confirmDelete(createdPost!.id);
                          }
                        },
                      ),
                    ],
                  ),

                  const SizedBox(height: 10),
                  Text(
                    createdPost!.body,
                    style: const TextStyle(fontSize: 16),
                  ),
                  const SizedBox(height: 12),
                  Text(
                    "id: ${createdPost!.id} | userId: ${createdPost!.userId}",
                  ),
                  const SizedBox(height: 12),
                  Text(
                    DateFormat("dd MMM yyyy, HH:mm")
                        .format(createdPost!.createdAt),
                    style:
                    const TextStyle(fontSize: 12, color: Colors.grey),
                  ),
                ],
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```



```dart
// ==========================
// KONFIRMASI DELETE
// ==========================
void confirmDelete(int id) {
  showDialog(
    context: context,
    builder: (_) => AlertDialog(
      title: const Text("Konfirmasi Hapus"),
      content: const Text("Yakin ingin menghapus data ini?"),
      actions: [
        TextButton(
          onPressed: () => Navigator.pop(context),
          child: const Text("Batal"),
        ),
        ElevatedButton(
          onPressed: () async {
            final success = await apiService.deletePost(id);

            if (success) {
              setState(() => createdPost = null);
            }

            Navigator.pop(context);
          },
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.red,
          ),
          child: const Text("Hapus"),
        ),
      ],
    ),
  );
}
```

b. Penjelasan Kode Program.

1. Penjelasan main.dart

File main.dart adalah titik awal aplikasi berjalan. Di bagian ini, aplikasi Flutter dijalankan melalui fungsi runApp() yang memanggil widget MyApp. Di dalam MyApp, kita menggunakan MaterialApp untuk mengatur tema dan halaman pertama yang muncul. Tema warna dibuat selaras dengan modul, yaitu menggunakan warna indigo sebagai warna utama dan background yang lebih lembut agar tampilannya bersih. Halaman awal aplikasi diarahkan ke HomePage, yang nanti menjadi tempat seluruh proses CRUD (Create, Read, Update, Delete) dilakukan. Intinya, file ini bertugas mengatur tampilan dasar aplikasi sekaligus menentukan halaman mana yang pertama kali ditampilkan.

2. Penjelasan post_model.dart

File ini berisi class Post, yaitu struktur data yang digunakan untuk menyimpan informasi satu data “post”. Di dalam class ini terdapat beberapa atribut seperti id, userId, title, body, dan waktu pembuatan (createdAt). Class ini juga memiliki metode fromJson() yang berfungsi mengubah data JSON dari API menjadi objek Dart. Dengan adanya model ini, pengelolaan data jadi lebih rapi dan terstruktur karena semua bagian aplikasi menggunakan format data yang sama. File ini bisa dianggap sebagai “wadah” data agar mudah dipakai di UI maupun API.

3. Penjelasan api_service.dart

File ini menjadi penghubung antara aplikasi Flutter dan server. Di sini terdapat class ApiService yang berisi fungsi-fungsi untuk mengirim permintaan ke API, seperti menambah data (createPost()), memperbarui data (updatePost()), dan menghapus data (deletePost()). Setiap fungsi mengirim request HTTP ke server menggunakan metode POST, PUT, dan DELETE. Setelah server memberikan respons, fungsi-fungsi ini mengubah data tersebut menjadi objek Post agar bisa digunakan kembali dalam aplikasi. Dengan adanya file ini, logika komunikasi dengan server dipisahkan dari tampilan sehingga kode lebih bersih dan gampang dikelola.

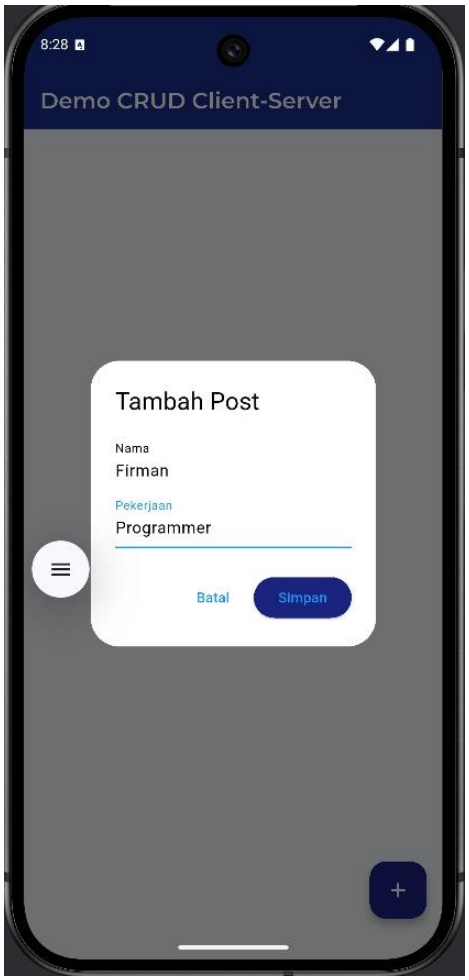
4. Penjelasan home_page.dart

Ini adalah inti dari aplikasi, tempat semua aktivitas CRUD dilakukan dan ditampilkan ke pengguna. Halaman ini menggunakan StatefulWidget agar UI bisa berubah otomatis ketika ada data baru, data diedit, atau data dihapus. Ada tiga dialog penting: dialog untuk menambah data, dialog untuk mengedit data, dan dialog konfirmasi saat menghapus. Setiap dialog menyediakan TextField untuk input dan tombol aksi seperti Simpan, Update, atau Hapus.

Bagian tampilan utama menggunakan widget Card untuk menampilkan data yang sudah dibuat, lengkap dengan judul, isi, waktu dibuat, dan tombol menu untuk

Update atau Delete. Ketika data berubah—misalnya setelah tombol Simpan ditekan—fungsi setState() akan memperbarui UI secara langsung. Halaman ini juga memiliki FloatingActionButton berwarna indigo yang digunakan untuk menambah data baru. Intinya, file home_page.dart mengatur seluruh alur CRUD sekaligus menampilkan hasilnya secara real-time.

Output:



IV. Kesimpulan.

Melalui praktikum Modul 13 ini, saya belajar bahwa mengambil lokasi perangkat di Flutter bukan hanya soal “mengakses GPS”, tetapi tentang bagaimana beberapa komponen saling bekerja sama. Mulai dari izin lokasi yang harus disetujui pengguna, pengecekan apakah layanan GPS aktif, hingga proses mendapatkan koordinat dan mengubahnya menjadi informasi alamat yang lebih mudah dipahami.

Selama praktikum, setiap file memiliki perannya masing-masing. main.dart menjadi pintu masuk aplikasi, post_model.dart membantu merapikan data, api_service.dart mengatur komunikasi dengan server, dan home_page.dart menjadi tempat semua aksi CRUD dijalankan. Dari sini saya semakin memahami bahwa aplikasi yang rapi harus memisahkan antara tampilan, logika, dan data.

Selain itu, proses *setState()* membuat saya melihat bagaimana UI dapat berubah secara langsung ketika ada data baru atau data diperbarui. Hal ini membuat aplikasi terasa lebih hidup dan responsif.

Secara keseluruhan, praktikum ini membantu saya memahami alur kerja pengolahan data lokasi dan bagaimana menampilkannya secara interaktif di aplikasi Flutter. Saya juga semakin terbiasa menulis kode yang terstruktur, mudah dibaca, dan siap dikembangkan lebih lanjut. Dengan demikian, semua tujuan pembelajaran pada modul ini dapat dicapai dengan baik.