

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XII  
MAPS PLACE FIREBASE NOTIFICATION**



**Disusun Oleh :**

**Zivana Afra Yulianto / 2211104039**

**SE-06-02**

**Asisten Praktikum :**

**Muhammad Faza Zulian Gesit Al Barru**

**Aisyah Hasna Aulia**

**Dosen Pengampu :**

**Yudha Islami Sulistya, S.Kom., M.Cs**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## **BAB I**

### **PENDAHULUAN**

#### **A. DASAR TEORI**

Peta digital adalah representasi geografis yang digunakan dalam layanan berbasis lokasi. Google Maps API menyediakan fitur seperti tampilan peta, marker, dan pencarian lokasi melalui Google Places API. Dalam Flutter, integrasi peta dilakukan menggunakan Google Maps Flutter Plugin, yang mendukung navigasi, lokasi pengguna, dan interaksi peta.

#### **B. MAKSUD DAN TUJUAN**

Maksud:

Memahami implementasi peta digital menggunakan Google Maps dan Places API.

Tujuan:

1. Mengintegrasikan Google Maps ke dalam Flutter.
2. Menampilkan lokasi pengguna dan marker.
3. Menggunakan Places API untuk pencarian lokasi.

## BAB II IMPLEMENTASI

### A. GUIDED

#### Code main.dart

```
import 'package:flutter/material.dart';
import 'package:praktikum/homepage.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

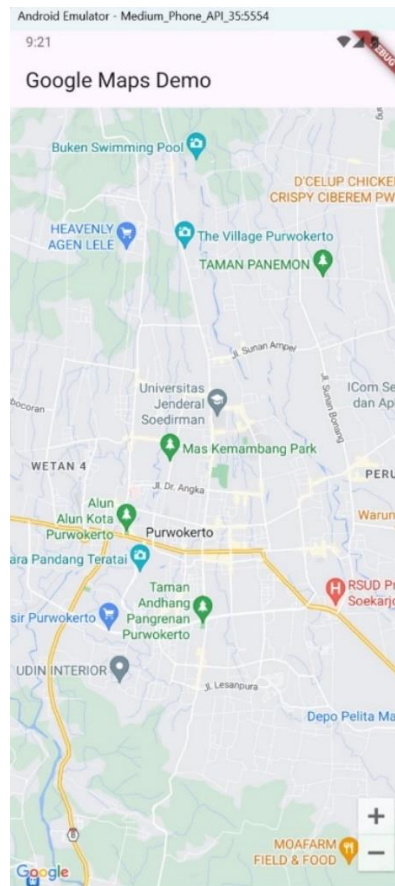
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: MapsScreen(),
    );
  }
}
```

#### Code homepage.dart :

```
import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';

class MapsScreen extends StatefulWidget {
  @override
  _MapsScreenState createState() => _MapsScreenState();
}

class _MapsScreenState extends State<MapsScreen> {
  static final LatLng _kMapCenter =
    LatLng(19.018255973653343, 72.84793849278007);
  static final CameraPosition _kInitialPosition = CameraPosition(
    target: _kMapCenter,
    zoom: 11.0,
  );
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Google Maps Demo'),
      ),
      body: GoogleMap(
        initialCameraPosition: _kInitialPosition,
        myLocationEnabled: true,
      ),
    );
  }
}
```



- Fungsi Utama:
  - Aplikasi menampilkan peta Google Maps yang dimulai pada posisi awal dengan koordinat yang telah ditentukan (`LatLng(19.018255973653343, 72.84793849278007)`).
  - Peta dapat menampilkan lokasi pengguna dengan mengaktifkan fitur `myLocationEnabled`.
- Kelas Utama:
  - `MapsScreen`: Merupakan layar utama aplikasi yang menampilkan peta.
- Detail Implementasi:
  - `_kMapCenter`: Titik pusat awal peta, yang diatur menggunakan koordinat lintang dan bujur.
  - `_kInitialPosition`: Posisi awal kamera peta dengan tingkat zoom 11.0.
  - `GoogleMap Widget`: Komponen peta dari pustaka `Google Maps Flutter` yang menyediakan berbagai fitur peta.
    - Parameter `initialCameraPosition` menentukan lokasi awal kamera.
    - Parameter `myLocationEnabled` diatur menjadi `true`, sehingga lokasi pengguna dapat ditampilkan di peta jika izin lokasi diberikan.

## B. UNGUIDED

### Code main.dart :

```
import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'src/locations.dart' as locations;

void main() {
  runApp(const MyApp());
}

class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  final Map<String, Marker> _markers = {};
  Future<void> _onMapCreated(GoogleMapController controller) async {
    final googleOffices = await locations.getGoogleOffices();
    setState(() {
      _markers.clear();
      for (final office in googleOffices.offices) {
        final marker = Marker(
          markerId: MarkerId(office.name),
          position: LatLng(office.lat, office.lng),
          infoWindow: InfoWindow(
            title: office.name,
            snippet: office.address,
          ),
        );
        _markers[office.name] = marker;
      }
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Google Office Locations'),
          backgroundColor: Colors.green[700],
        ),
        body: GoogleMap(
          onMapCreated: _onMapCreated,
          initialCameraPosition: const CameraPosition(
            target: LatLng(0, 0),
            zoom: 2,
          ),
          markers: _markers.values.toSet(),
        ),
      ),
    );
  }
}
```

## Code lib/src/location.dart

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:json_annotation/json_annotation.dart';
import 'package:flutter/services.dart' show rootBundle;

part 'locations.g.dart';

@JsonSerializable()
class LatLng {
  LatLng({
    required this.lat,
    required this.lng,
  });

  factory LatLng.fromJson(Map<String, dynamic> json) => _$LatLngFromJson(json);
  Map<String, dynamic> toJson() => _$LatLngToJson(this);

  final double lat;
  final double lng;
}

@JsonSerializable()
class Region {
  Region({
    required this.coords,
    required this.id,
    required this.name,
    required this.zoom,
  });

  factory Region.fromJson(Map<String, dynamic> json) => _$RegionFromJson(json);
  Map<String, dynamic> toJson() => _$RegionToJson(this);

  final LatLng coords;
  final String id;
  final String name;
  final double zoom;
}

@JsonSerializable()
class Office {
  Office({
    required this.address,
    required this.id,
    required this.image,
    required this.lat,
    required this.lng,
    required this.name,
    required this.phone,
    required this.region,
  });

  factory Office.fromJson(Map<String, dynamic> json) => _$OfficeFromJson(json);
  Map<String, dynamic> toJson() => _$OfficeToJson(this);

  final String address;
  final String id;
  final String image;
  final double lat;
  final double lng;
  final String name;
  final String phone;
  final String region;
}
```

```

@JsonSerializable()
class Locations {
  Locations({
    required this.offices,
    required this.regions,
  });

  factory Locations.fromJson(Map<String, dynamic> json) =>
    _$LocationsFromJson(json);
  Map<String, dynamic> toJson() => _$LocationsToJson(this);

  final List<Office> offices;
  final List<Region> regions;
}

Future<Locations> getGoogleOffices() async {
  const googleLocationsURL = 'https://about.google/static/data/locations.json';

  // Retrieve the locations of Google offices
  try {
    final response = await http.get(Uri.parse(googleLocationsURL));
    if (response.statusCode == 200) {
      return Locations.fromJson(json.decode(response.body));
    }
  } catch (e) {
    print(e);
  }

  // Fallback for when the above HTTP request fails.
  return Locations.fromJson(
    json.decode(
      await rootBundle.loadString('assets/locations.json'),
    ),
  );
}

```

### Kemuddian pada terminal :

```

$ flutter pub run build_runner build --delete-conflicting-outputs
$ mkdir assets
$ cd assets
$ curl -o locations.json https://about.google/static/data/locations.json

```

### Pada pubspec.yaml :

```

flutter:
  uses-material-design: true

assets:
  - assets/locations.json

```

## Screenshoot Output



## Deskripsi Program

### Model Data (Menggunakan `json_serializable`)

- **LatLng:** Menyimpan koordinat geografis (latitude dan longitude).
- **Region:** Menyimpan informasi tentang wilayah, termasuk koordinat, ID, nama, dan tingkat zoom peta.
- **Office:** Menyimpan informasi kantor, seperti alamat, ID, gambar, koordinat, nama, nomor telepon, dan wilayah terkait.
- **Locations:** Menyimpan daftar **kantor** dan **wilayah**.

## 2. Fungsi `getGoogleOffices`

- **API Request:** Mengambil data lokasi kantor Google dari URL API <https://about.google/static/data/locations.json>.
- **Error Handling:** Jika permintaan API gagal, data akan diambil dari file lokal `assets/locations.json` sebagai cadangan.
- **Deserialisasi:** Data JSON yang diterima diubah menjadi objek `Locations` menggunakan model data yang sudah didefinisikan.

## BAB III KESIMPULAN

### KESIMPULAN



Google Maps Flutter mempermudah pengembangan aplikasi berbasis lokasi dengan fitur interaktif seperti pencarian lokasi (Places API) dan lokasi pengguna. Praktikum ini menunjukkan bagaimana peta digital dapat digunakan untuk navigasi, pelacakan, dan pencarian lokasi secara efisien.