

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK
MODUL XII
Maps, Places, dan Firebase Notifikasi Bagian**



Disusun Oleh :

Dwi Candra Pratama / 2211104032 SE06-02

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**

Guided

MAPS AND PLACES

Tujuan Praktikum

1. Mahasiswa memahami cara mengintegrasikan Google Maaps ke dalam project Flutter.
2. Mahasiswa dapat mengimplementasikan Google Maps API untuk menampilkan lokasi

A. GOOGLE MAPS API

Google Maps API untuk Flutter adalah layanan yang memungkinkan pengembang untuk mengintegrasikan peta interaktif ke dalam aplikasi Flutter mereka. Dengan menggunakan plugin seperti `google_maps_flutter`, pengembang dapat memanfaatkan fitur-fitur canggih dari Google Maps, termasuk penanda lokasi (markers), poligon, polylines, kontrol navigasi, serta kemampuan untuk memanipulasi dan menyesuaikan tampilan peta secara dinamis. API ini juga mendukung berbagai fungsi, seperti mendapatkan lokasi pengguna saat ini, menampilkan rute, dan menyediakan data geospasial. Karena berbasis Flutter, layanan ini kompatibel untuk aplikasi lintas platform (iOS dan Android), sehingga mempermudah proses pengembangan aplikasi dengan antarmuka peta yang interaktif dan responsif. Google Maps API memerlukan konfigurasi kunci API dan pengaturan izin tertentu di platform yang digunakan untuk memastikan integrasi berjalan dengan lancar.

Berikut adalah tahapan agar dalam menambahkan Google Maps API bisa stabil:

1. Dapatkan API key melalui link berikut <https://cloud.google.com/maps-platform/>
2. Selanjutnya, enable Google Map SDK di tiap platform yang akan menggunakan Google Maps.
 - a. Pergi ke <https://console.cloud.google.com/> (Google Delevopers Console)
 - b. Pilih project yang ingin menggunakan Google Maps
 - c. Pilih pada navigation menu, lalu pilih "Google Maps"
 - d. Pilih "APIs" di bawah menu Google Maps
 - e. Untuk mengaktifkan Google Maps di Android, pilih "Maps SDK for Android" pada section "Additional APIs", lalu pilih "ENABLE"
 - f. Untuk mengaktifkan Google Maps di iOS, pilih "Maps SDK for iOS" pada section "Additional APIs", lalu pilih "ENABLE"
 - g. Pastikan bahwa APIs telah aktif pada section "Enabled APIs" h. Untuk lebih detail bisa cek di <https://developers.google.com/maps/gmp-get-started>

Karena saya android jadi Set minSdkVersion di android/app/build.gradle:

```
defaultConfig {  
    // TODO: Specify your own unique Application  
    applicationId = "com.example.pertemuan_12"  
    // You can update the following values to m  
    // For more information, see: https://flutt  
    minSdkVersion 23  
    targetSdk = flutter.targetSdkVersion  
    versionCode = flutter.versionCode  
    versionName = flutter.versionName  
}
```

Serta Tambahkan API key pada manifest aplikasi
(android/app/src/main/AndroidManifest.xml)

```
4      <application
5          android:label="pertemuan_12"
6          android:name="${applicationName}"
7          android:icon="@mipmap/ic_launcher">
8          <meta-data android:name="com.google.android.geo.API_KEY"
9              android:value="AIzaSyD1-yAXZswVWxrIxdfuLxBT-14HJk0XVzo"/>
10         <activity
```

B. Menambah Packages Google Maps

tambahkan Google Maps ke package aplikasi Flutter:

```
37     cupertino_icons: ^1.0.8
38     google_maps_flutter: ^2.10.0
39     place_picker_google: ^0.0.13
```

C. Berikut adalah Penerapan Secara Full saat Praktikum:

Source Code:

Main.dart

```
import 'package:flutter/material.dart';
import 'package:pertemuan_12/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',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: MyMaps(),
    );
  }
}
```

Homepage.dart

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

```

import 'package:google_maps_flutter/google_maps_flutter.dart';

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

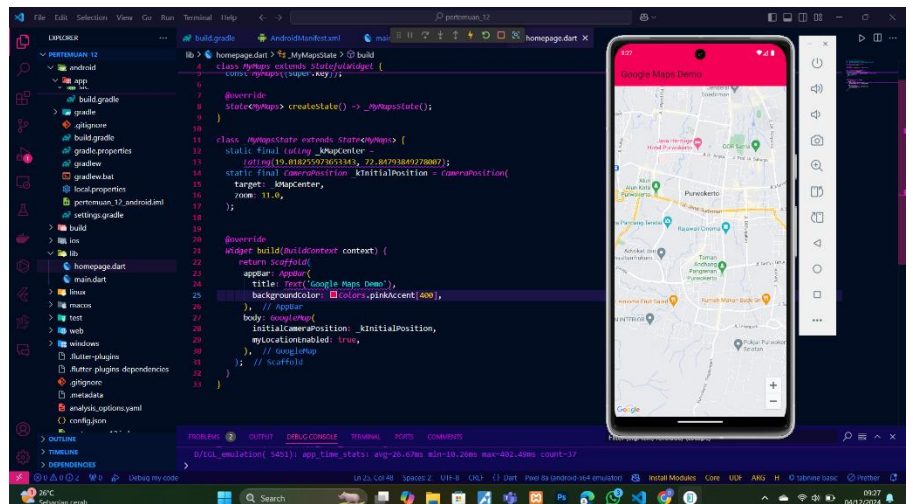
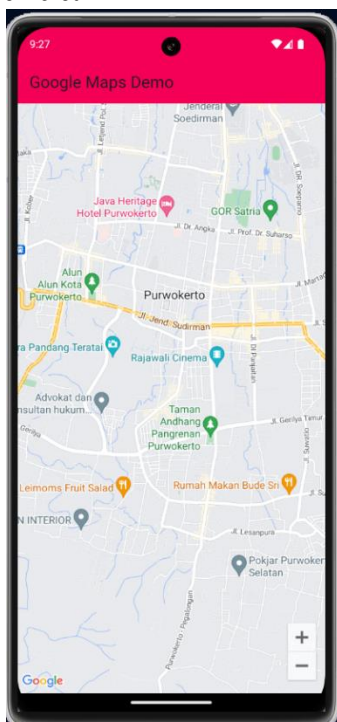
  @override
  State<MyMaps> createState() => _MyMapsState();
}

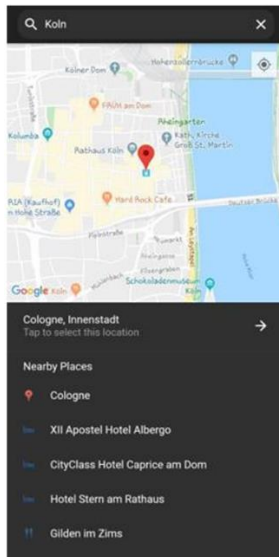
class _MyMapsState extends State<MyMaps> {
  static final LatLng _kMapCenter =
    LatLng(19.018255973653343, 72.84793849278007);
  static final CameraPosition _kInitialPosition = CameraPosition(
    target: _kMapCenter,
    zoom: 11.0,
  );

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Google Maps Demo'),
        backgroundColor: Colors.pinkAccent[400],
      ),
      body: GoogleMap(
        initialCameraPosition: _kInitialPosition,
        myLocationEnabled: true,
      ),
    );
  }
}

```

OutPut:





Tugas Mandiri (Unguided)

Dari tugas guided yang telah dikerjakan, lanjutkan hingga ke bagian place picker untuk memberikan informasi mengenai lokasi yang ditunjuk di peta.

Source Code:

Homepage.dart

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

class AppColors {
  static const Color primary = Color(0xFFE91E63);
  static const Color secondary = Color(0xFFFF06292);
  static const Color background = Color(0xFFFFF8FB);
  static const Color surface = Color(0xFFFFFFFF);
  static const Color text = Color(0xFF880E4F);
  static const Color textSecondary = Color(0xFFD81B60);
  static const Color divider = Color(0xFFF8BBD0);
}

class MyMaps extends StatefulWidget {
  @override
  _MyMapsState createState() => _MyMapsState();
}

class _MyMapsState extends State<MyMaps> with TickerProviderStateMixin {
  static final LatLng _kMapCenter =
    const LatLng(-7.429474, 109.253845);

  static final CameraPosition _kInitialPosition = CameraPosition(
    target: _kMapCenter,
    zoom: 16.0,
  );

  GoogleMapController? _mapController;
```

```

Set<Marker> _markers = {};
late AnimationController _animationController;

final Map<LatLng, Map<String, dynamic>> _placesInfo = {
  const LatLng(-7.429474, 109.253845): {
    'name': 'RS Margono Purwokerto',
    'address': 'Jl. Dr. Gumbreg No.1, Purwokerto',
    'type': 'Hospital',
    'description': 'Rumah sakit umum terbesar di Purwokerto',
    'phone': '+62 281 632708',
    'website': 'www.rsmargono.go.id',
    'operatingHours': '24 Hours',
    'facilities': [
      'Emergency',
      'Inpatient',
      'Pharmacy',
      'Parking',
      'Laboratory'
    ],
    'rating': 4.5
  },
};

@override
void initState() {
  super.initState();
  _initializeMarkers();
  _animationController = AnimationController(
    duration: const Duration(milliseconds: 300),
    vsync: this,
  );
}

@override
void dispose() {
  _animationController.dispose();
  super.dispose();
}

void _initializeMarkers() {
  _markers = _placesInfo.entries.map((entry) {
    return Marker(
      markerId: MarkerId(entry.value['name']),
      position: entry.key,
      onTap: () => _showPlaceInfo(entry.key),
      icon: BitmapDescriptor.defaultMarkerWithHue(BitmapDescriptor.hueAzure),
    );
  }).toSet();
}

```

```

void _showPlaceInfo(LatLng position) {
  _animationController.forward(from: 0.0);
  showModalBottomSheet(
    context: context,
    isScrollControlled: true,
    backgroundColor: Colors.transparent,
    builder: (context) => _buildBottomSheet(position),
  );
}

```

```

Widget _buildInfoCard(IconData icon, String title, String content) {
  return Container(
    margin: const EdgeInsets.only(bottom: 16),
    decoration: BoxDecoration(
      color: AppColors.surface,
      borderRadius: BorderRadius.circular(12),
      boxShadow: [
        BoxShadow(
          color: Colors.black.withOpacity(0.05),
          blurRadius: 10,
          offset: const Offset(0, 2),
        ),
      ],
    ),
    child: ClipRRect(
      borderRadius: BorderRadius.circular(12),
      child: Material(
        color: Colors.transparent,
        child: InkWell(
          onTap: () {},
          child: Padding(
            padding: const EdgeInsets.all(16),
            child: Row(
              children: [
                Container(
                  padding: const EdgeInsets.all(12),
                  decoration: BoxDecoration(
                    color: AppColors.primary.withOpacity(0.1),
                    borderRadius: BorderRadius.circular(10),
                  ),
                  child: Icon(icon, color: AppColors.primary, size: 24),
                ),
                const SizedBox(width: 16),
                Expanded(
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Text(
                        title,
                        style: const TextStyle(

```

```

        color: AppColors.textSecondary,
        fontSize: 14,
        fontWeight: FontWeight.w500,
      ),
    ),
    const SizedBox(height: 4),
    Text(
      content,
      style: const TextStyle(
        color: AppColors.text,
        fontSize: 16,
        fontWeight: FontWeight.w600,
      ),
    ),
  ],
),
),
],
),
),
),
),
),
),
);
}

```

```

Widget _buildBottomSheet(LatLng place) {
  final placeInfo = _placesInfo[place];

  return AnimatedBuilder(
    animation: _animationController,
    builder: (context, child) {
      return Container(
        height: MediaQuery.of(context).size.height * 0.75,
        decoration: const BoxDecoration(
          color: AppColors.background,
          borderRadius: BorderRadius.vertical(top: Radius.circular(28)),
        ),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            Container(
              padding: const EdgeInsets.fromLTRB(24, 12, 24, 24),
              decoration: BoxDecoration(
                color: AppColors.surface,
                borderRadius: const BorderRadius.vertical(top: Radius.circular(28)),
                boxShadow: [
                  BoxShadow(
                    color: Colors.black.withOpacity(0.05),
                    blurRadius: 10,

```



```

        offset: const Offset(0, 2),
      ),
    ],
  ),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Center(
        child: Container(
          width: 40,
          height: 4,
          margin: const EdgeInsets.only(bottom: 16),
          decoration: BoxDecoration(
            color: AppColors.divider,
            borderRadius: BorderRadius.circular(2),
          ),
        ),
      ),
    ],
  ),
  Row(
    children: [
      Expanded(
        child: Text(
          placeInfo?['name'] ?? "",
          style: const TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: AppColors.text,
          ),
        ),
      ),
      Container(
        padding:
          const EdgeInsets.symmetric(horizontal: 12, vertical: 6),
        decoration: BoxDecoration(
          color: AppColors.primary,
          borderRadius: BorderRadius.circular(20),
        ),
        child: Row(
          children: [
            const Icon(Icons.star, color: Colors.white, size: 16),
            const SizedBox(width: 4),
            Text(
              '${placeInfo?['rating']}',
              style: const TextStyle(
                color: Colors.white,
                fontWeight: FontWeight.bold,
              ),
            ),
          ],
        ),
      ),
    ],
  ),

```

```

    ),
  ],
),
const SizedBox(height: 8),
Text(
  placeInfo?['type'] ?? "",
  style: const TextStyle(
    fontSize: 16,
    color: AppColors.primary,
    fontWeight: FontWeight.w500,
  ),
),
],
),
),
Expanded(
  child: SingleChildScrollView(
    physics: const BouncingScrollPhysics(),
    padding: const EdgeInsets.all(24),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        _buildInfoCard(
          Icons.location_on_rounded,
          'Address',
          placeInfo?['address'] ?? "",
        ),
        _buildInfoCard(
          Icons.access_time_rounded,
          'Operating Hours',
          placeInfo?['operatingHours'] ?? "",
        ),
        _buildInfoCard(
          Icons.phone_rounded,
          'Contact',
          placeInfo?['phone'] ?? "",
        ),
        _buildInfoCard(
          Icons.info_outline_rounded,
          'About',
          placeInfo?['description'] ?? "",
        ),
      ],
    ),
  ),
  const Text(
    'Facilities',
    style: TextStyle(
      fontSize: 18,
      fontWeight: FontWeight.bold,
      color: AppColors.text,
    ),
  ),
),

```

```

    ),
  ),
  const SizedBox(height: 16),
  Wrap(
    spacing: 8,
    runSpacing: 8,
    children:
      (placeInfo?['facilities'] as List<String>? ?? [])
        .map((facility) {
      return Chip(
        label: Text(
          facility,
          style: const TextStyle(color: AppColors.primary),
        ),
        backgroundColor: AppColors.primary.withOpacity(0.1),
        padding: const EdgeInsets.symmetric(horizontal: 8),
      );
    }).toList(),
  ),
  const SizedBox(height: 24),

  ElevatedButton(
    onPressed: () {
      _mapController?.animateCamera(
        CameraUpdate.newLatLngZoom(place, 17),
      );
      Navigator.pop(context);
    },
    child: const Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Icon(Icons.navigation_rounded),
        SizedBox(width: 8),
        Text(
          'Navigate',
          style: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.w600,
          ),
        ),
      ],
    ),
    style: ElevatedButton.styleFrom(
      backgroundColor: AppColors.primary,
      padding: const EdgeInsets.symmetric(vertical: 16),
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(12),
      ),
      elevation: 0,
    ),
  ),

```

```

    ),
    ],
  ),
),
],
),
);
},
);
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    extendBodyBehindAppBar: true,
    appBar: AppBar(
      title: const Text(
        'Place Picker Implementation',
        style: TextStyle(
          color: AppColors.text,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
    backgroundColor: AppColors.surface.withOpacity(0.9),
    elevation: 0,
    centerTitle: true,
    shape: const RoundedRectangleBorder(
      borderRadius: BorderRadius.vertical(bottom: Radius.circular(16)),
    ),
  ),
  body: GoogleMap(
    initialCameraPosition: _kInitialPosition,
    myLocationEnabled: true,
    myLocationButtonEnabled: true,
    onMapCreated: (GoogleMapController controller) {
      _mapController = controller;
    },
    markers: _markers,
    mapType: MapType.normal,
  ),
);
}
}
}

```

Main.dart

```

import 'package:flutter/material.dart';
import 'package:pertemuan_12/homepage.dart';

void main() {

```

```

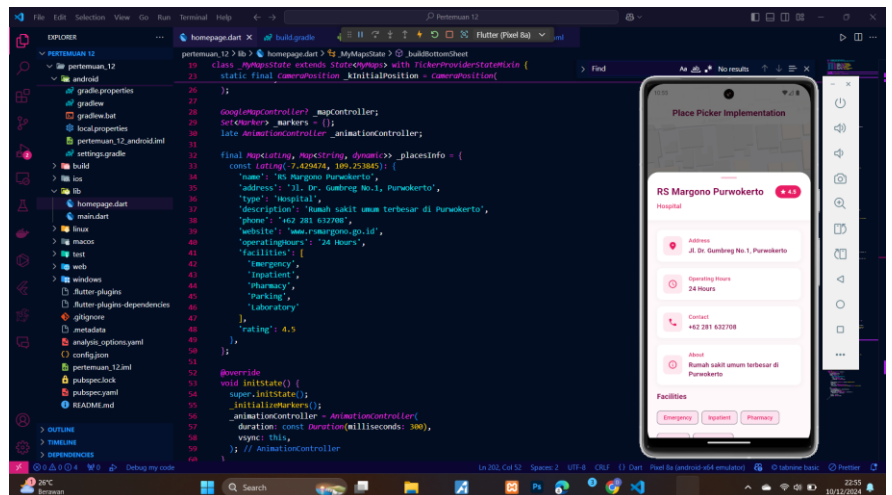
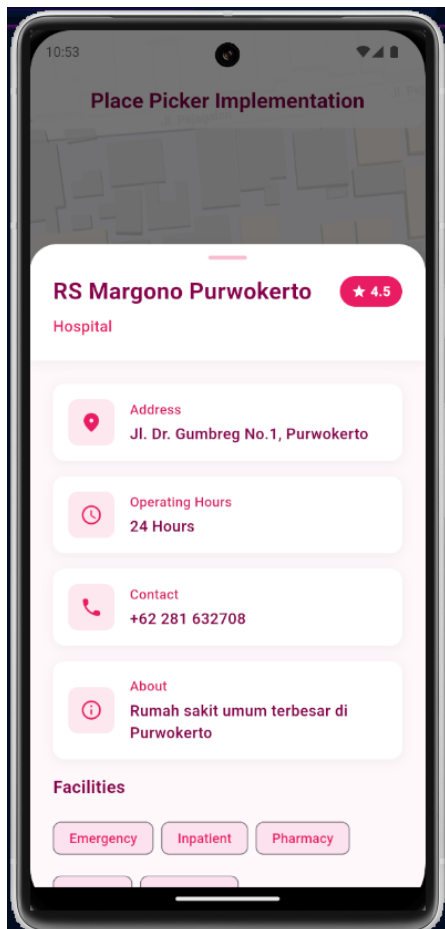
runApp(const MyApp());
}

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',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: MyMaps(),
    );
  }
}

```

OutPut:



Deskripsi Program:

Program di atas adalah implementasi fitur Google Maps menggunakan Flutter yang dirancang untuk menampilkan peta interaktif dengan informasi detail tempat tertentu. Aplikasi ini menggunakan `google_maps_flutter` untuk menampilkan peta dengan marker pada lokasi yang ditentukan, yaitu RS Margono Purwokerto. Marker tersebut menampilkan informasi seperti nama, alamat, tipe, deskripsi, nomor telepon, situs web, jam operasional, fasilitas, dan rating saat disentuh. Informasi tempat ditampilkan dalam bentuk modal bottom sheet yang dapat digeser dan berisi detail lengkap.

Selain itu, aplikasi ini dilengkapi fitur navigasi seperti tombol untuk kembali ke lokasi yang dipilih. Desain antarmuka menggunakan warna yang didefinisikan dalam kelas `AppColors`, menciptakan tema yang konsisten. Program ini memanfaatkan widget animasi untuk memberikan pengalaman yang halus saat bottom sheet muncul. Dengan peta interaktif dan informasi detail yang disajikan, aplikasi ini cocok digunakan untuk berbagai keperluan lokasi dan navigasi di dunia nyata.