

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XII
MAPS & PLACES**



Disusun Oleh :

Devrin Anggun Saputri / 2211104001

SE0601

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

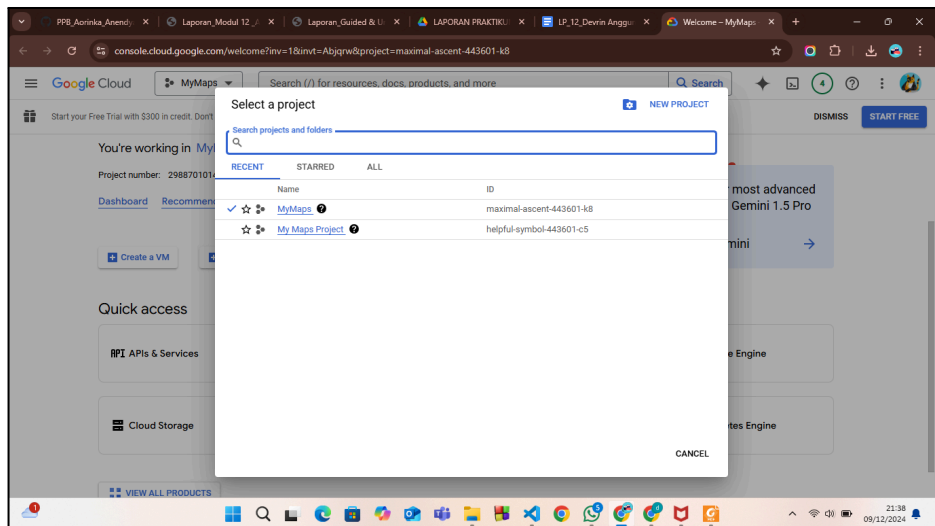
2024

GUIDED

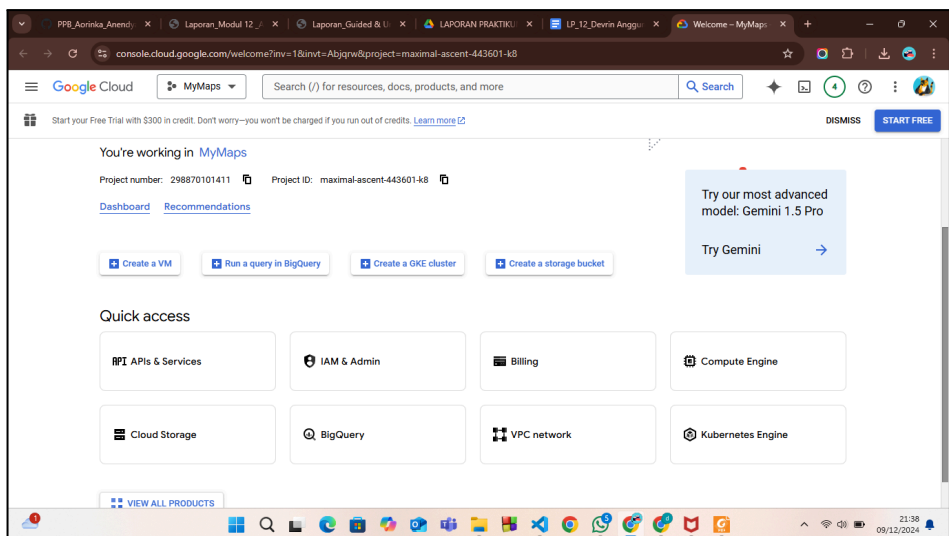
A. GOOGLE MAPS API

Google Maps API merupakan salah satu layanan dari Google untuk membantu developer menciptakan aplikasi yang menggunakan fitur peta atau maps. Pada Google Maps API kita dapat memasang marker, menggunakan fitur route, mencari tempat, dan masih banyak lagi. Cara implementasi Google API pada flutter dapat dilakukan dengan menggunakan packages Google Maps. Tahapan dalam menambahkan Google Maps API dapat mengikuti langkah-langkah berikut :

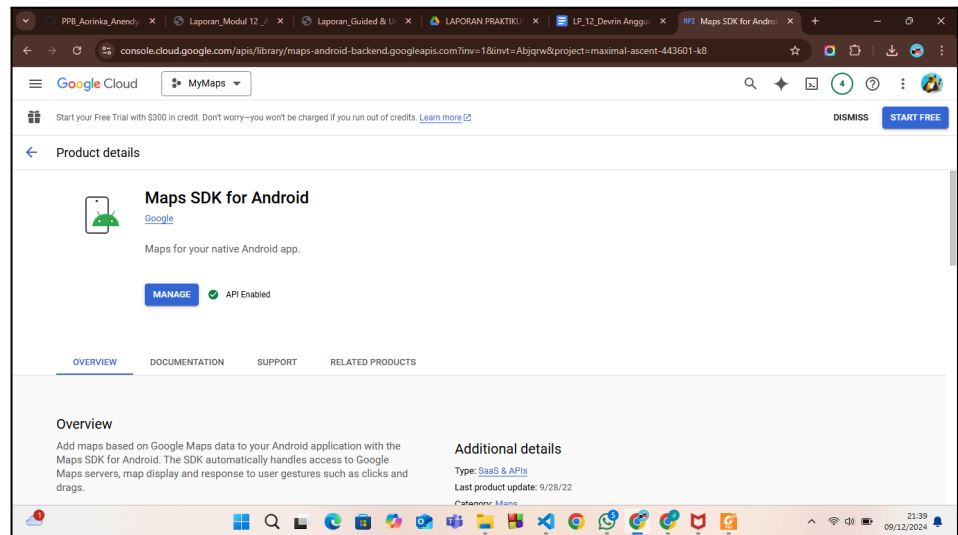
1. Dapatkan API key melalui link berikut
<https://cloud.google.com/maps-platform/>
2. Selanjutnya, enable Google Map SDK di tiap platform yang akan menggunakan Google Maps.
 - a. Pergi ke [Google Developers Console](#)
 - b. Pilih project yang ingin menggunakan Google Maps



- 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>

3. Android

- a. Set minSdkVersion di android/app/build.gradle:

```
minSdkVersion 20
targetSdk = flutter.targetSdkVersion
versionCode = flutter.versionCode
versionName = flutter.versionName
```

- b. Tambahkan API key pada manifest aplikasi android/app/src/main/AndroidManifest.xml

```
<meta-data android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyBu41DG5kKf_JYioaJgsAb55tvYZVrbIMk"/>
<activity
```

B. Menambah Packages Google Maps

Setelah mengikuti langkah diatas, sekarang adalah langkah-langkah menambahkan Google Maps ke layar aplikasi Flutter:

1. Pergi ke <https://www.pub.dev> lalu cari packages Google Maps. Nama packagesnya adalah google_maps_flutter.
2. Cari versi yang paling terbaru lalu tambahkan pada file pubspec.yaml

```
dependencies:
  flutter:
    sdk: flutter

# The following adds the Cupertino
# Use with the CupertinoIcons
cupertino_icons: ^1.0.8
google_maps_flutter: ^2.10.0
```

3. Selanjutnya, import packages ke dalam file Dart

```
1 import 'package:flutter/material.dart';
2 import 'package:praktikum_12/homepage.dart';
```

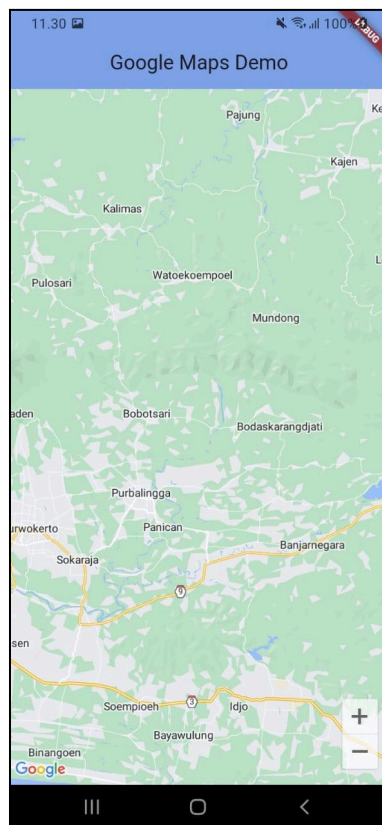
4. Lalu, tambahkan widget GoogleMap ke file Dart

```
23 body: GoogleMap(
24   initialCameraPosition: _kInitialPosition,
25   myLocationEnabled: true,
```

5. Buat fungsi `_kMapCenter` dan `_kInitialPosition` dengan kode sebagai berikut:

```
9 class _MapsScreenState extends State<MapsScreen> {
10   static final LatLng _kMapCenter = LatLng(-7.3548884, 109.0485449);
11   static final CameraPosition _kInitialPosition = CameraPosition(
12     target: _kMapCenter,
13     zoom: 11.0,
14   );
```

6. Berikut adalah tampilan kode yang lengkap
7. Output Tampilan:



C. Menambahkan Akses di Manifest

Secara default, peta akan menampilkan lokasi yang telah ditentukan pada parameter `initialCameraPosition` di widget. Untuk menampilkan lokasi pengguna, ubah pengaturan `myLocationEnabled` menjadi `true`. Berikut adalah kode untuk menampilkan lokasi saat ini:

```
body: GoogleMap(
  initialCameraPosition: _kInitialPosition,
  myLocationEnabled: true,
```

D. Menambahkan Marker pada Google Maps

Marker merupakan cara untuk menunjukkan lokasi tertentu. Untuk membuat marker pada map, berikut adalah barisan kodenya:

```
Set<Marker> _createMarker() {  
  return {  
    Marker(  
      markerId: MarkerId("marker 1"),  
      position: _kMapCenter,  
      infoWindow: InfoWindow(title: 'Marker 1'),  
      rotation: 0,  
    ), // Marker  
    Marker(  
      markerId: MarkerId("marker 2"),  
      position: LatLng(31.6680341,115.6758447), // Koordinat contoh lainnya  
    ), // Marker  
  };  
}
```

UNGUIDED

Tugas Mandiri (Unguided)

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

Note: Jangan lupa sertakan source code, screenshot output, dan deskripsi program. Kreativitas menjadi nilai tambah.

Source Code:

- main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:praktikum_12/homepage.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: 'Flutter Demo',
16       theme: ThemeData(),
17       home: MyHomePage(),
18     );
19   }
20 }
```

- homepage.dart

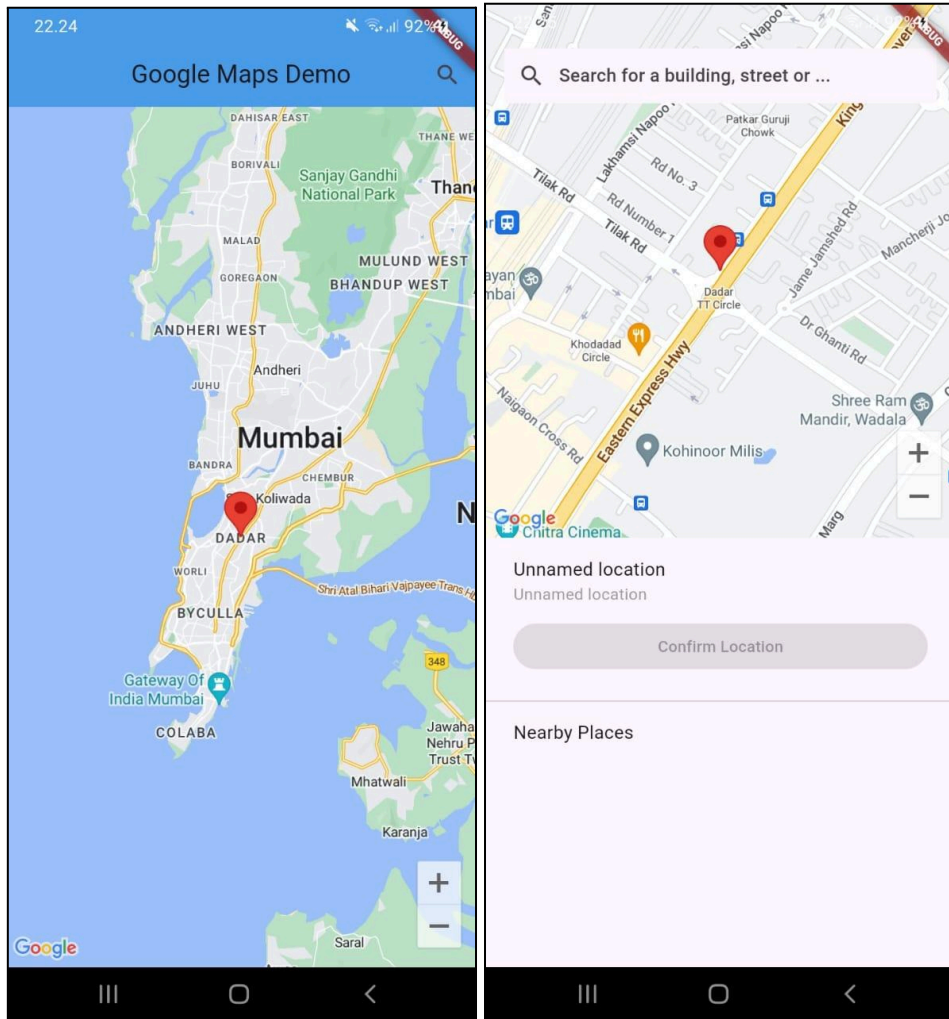
```
1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3 import 'package:place_picker_google/place_picker_google.dart';
4
5 class MyHomePage extends StatefulWidget {
6   const MyHomePage({super.key});
7
8   @override
9   State<MyHomePage> createState() => _MyHomePageState();
10 }
11
12 class _MyHomePageState extends State<MyHomePage> {
13   static final LatLng _kMapCenter = LatLng(19.018255973653343,
14     72.84793849278007); // Koordinat yang digunakan (contohnya lokasi di Mumbai)
15   static final CameraPosition _kInitialPosition =
16     CameraPosition(target: _kMapCenter, zoom: 11.0, tilt: 0, bearing: 0);
17 }
```

```

18 // API Key untuk Google Maps
19 final String apiKey = 'YOUR_GOOGLE_MAPS_API_KEY';
20
21 // Membuat marker
22 Set<Marker> _createMarker() {
23   return {
24     Marker(
25       markerId: MarkerId("marker_1"),
26       position: _kMapCenter,
27       infoWindow: InfoWindow(title: 'Marker 1'),
28       rotation: 0,
29     ),
30     Marker(
31       markerId: MarkerId("marker_2"),
32       position: LatLng(-6.9733165, 107.6281415), // Koordinat contoh lainnya
33     ),
34   };
35 }
36
37 // Fungsi untuk menampilkan Place Picker
38 void _showPlacePicker() async {
39   // Menampilkan Place Picker dan menangkap hasilnya
40   LocationResult? result = await Navigator.of(context).push(MaterialPageRoute(
41     builder: (context) => PlacePicker(
42       apiKey: apiKey, // API key untuk Google Maps
43       onPlacePicked: (LocationResult result) {
44         // Menampilkan alamat tempat yang dipilih
45         debugPrint("Place picked: ${result.formattedAddress}");
46       },
47       initialLocation: _kMapCenter, // Menampilkan lokasi awal
48       searchInputConfig: const SearchInputConfig(
49         padding: EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),
50         autofocus: false,
51         textDirection: TextDirection.ltr,
52       ),
53       searchInputDecorationConfig: const SearchInputDecorationConfig(
54         hintText: "Search for a building, street or ...",
55       ),
56     ),
57   ));
58
59   // Pengecekan jika result tidak ada
60   if (result != null) {
61     debugPrint("Place picked: ${result.formattedAddress}");
62   } else {
63     debugPrint("No place selected.");
64   }
65 }
66
67 @override
68 Widget build(BuildContext context) {
69   return Scaffold(
70     appBar: AppBar(
71       title: const Text('Google Maps Demo'),
72       centerTitle: true,
73       backgroundColor: const Color.fromARGB(255, 76, 157, 233),
74       actions: [
75         IconButton(
76           icon: const Icon(Icons.search),
77           onPressed:
78             _showPlacePicker, // Menampilkan Place Picker saat ikon pencarian ditekan
79         ),
80       ],
81     ),
82     body: GoogleMap(
83       initialCameraPosition: _kInitialPosition,
84       myLocationEnabled: true,
85       markers: _createMarker(), // Menambahkan marker pada peta
86     ),
87   );
88 }
89 }

```

Output:



Deskripsi Program:

Program ini menggunakan Google Maps API dan Google Places API untuk memungkinkan pengguna memilih lokasi. Fitur utama program adalah Place Picker, yang memanfaatkan widget untuk membantu pengguna mencari dan memilih lokasi dengan antarmuka yang mudah digunakan. Program ini juga menampilkan marker di peta menggunakan widget GoogleMap sebagai petunjuk visual lokasi tertentu.

Beberapa fungsi penting yang digunakan dalam program ini antara lain: fungsi `showPlacePicker` yang menampilkan Place Picker dan mengembalikan koordinat lokasi yang dipilih, fungsi `getPlaceName` yang memanggil Google Geocoding API untuk mendapatkan nama lokasi berdasarkan koordinat, dan fungsi `fetchNearbyPlaces` untuk mengambil daftar tempat terdekat dari Google Places API yang kemudian ditampilkan di layar. Marker awal ditambahkan ke peta dengan fungsi `_createMarker`, yang dipanggil saat program diinisialisasi.