

**TUGAS PENDAHULUAN**  
**PEMROGRAMAN PERANGKAT BERGERAK**  
**MODUL XII**  
**MAPS, PLACES DAN FIREBASE NOTIFIKASI**



**Disusun Oleh :**  
**Lintang Suminar Tyas Wening / 2211104009**  
**SE-06-01**

**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**

## TUGAS PENDAHULUAN

### SOAL

1. Menambahkan Google Maps Package
  - a. Apa nama package yang digunakan untuk mengintegrasikan Google Maps di Flutter dan sebutkan langkah-langkah yang diperlukan untuk menambahkan package Google Maps ke dalam proyek Flutter.
  - b. Mengapa kita perlu menambahkan API Key, dan di mana API Key tersebut diatur dalam aplikasi Flutter?
2. Menampilkan Google Maps
  - a. Tuliskan kode untuk menampilkan Google Map di Flutter menggunakan widget GoogleMap.
  - b. Bagaimana cara menentukan posisi awal kamera (camera position) pada Google Maps di Flutter?
  - c. Sebutkan properti utama dari widget GoogleMap dan fungsinya.
3. Menambahkan Marker
  - a. Tuliskan kode untuk menambahkan marker di posisi tertentu (latitude: -6.2088, longitude: 106.8456) pada Google Maps.
  - b. Bagaimana cara menampilkan info window saat marker diklik?
4. Menggunakan Place Picker
  - a. Apa itu Place Picker, dan bagaimana cara kerjanya di Flutter dan sebutkan nama package yang digunakan untuk implementasi Place Picker di Flutter.
  - b. Tuliskan kode untuk menampilkan Place Picker, lalu kembalikan lokasi yang dipilih oleh pengguna dalam bentuk latitude dan longitude.

### JAWAB:

1. Menambahkan Google Maps Package
  - a. Nama Package dan Langkah-langkah Penambahan Google Maps di Flutter  
Package yang umum digunakan untuk mengintegrasikan Google Maps di Flutter adalah `google_maps_flutter`.

Berikut langkah-langkah menambahkan package Google Maps ke proyek Flutter :

- 1) Tambahkan dependency ke dalam file `pubspec.yaml`:

```
dependencies:  
  google_maps: ^8.1.1  
  google_maps_flutter: ^2.10.0
```

- 2) Jalankan perintah `flutter pub get` untuk mengunduh package
- 3) Tambahkan izin lokasi pada file `AndroidManifest.xml` untuk Android :
  - Tambahkan perintah berikut

```
build.gradle  AndroidManifest.xml  homepage.dart  pubspec.yaml  main.dart  
android > app > src > main > AndroidManifest.xml  
1  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />  
2  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

- Masukkan API Key di dalam tag <application>

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

#### 4) Menggunakan widget Google Maps

- Import Library

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

- Tambahkan widget GoogleMap di dalam aplikasi

```
GoogleMap(
  initialCameraPosition: CameraPosition(
    target: LatLng(-6.1751, 106.8650), // Lokasi awal (contoh Jakarta)
    zoom: 12.0,
  ), // CameraPosition
```

- b. Firebase adalah platform yang dikembangkan oleh Google untuk membantu API Key penting untuk mengidentifikasi aplikasi ke Google Cloud Platform dan memungkinkan akses ke layanan seperti Google Maps. Selain itu, API Key mencegah penyalahgunaan, memastikan hanya aplikasi terotorisasi yang dapat menggunakan layanan, dan membantu melacak penggunaan. Pengaturan API Key dilakukan melalui Google Cloud Console dengan membuat project, mengaktifkan layanan seperti Google Maps SDK, dan mengelola API Key di menu Credentials. Akses pengaturan dapat dilakukan di Google Cloud Console.

## 2. Menampilkan Google Maps

### a. Kode untuk Menampilkan Google Map di Flutter

Berikut adalah contoh kode untuk menampilkan Google Map menggunakan widget GoogleMap:

```
1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3
4 void main() => runApp(MyApp());
5
6 class MyApp extends StatelessWidget {
7   @override
8   Widget build(BuildContext context) {
9     return MaterialApp(
10       home: MapScreen(),
11     );
12   }
13 }
14
15 class MapScreen extends StatelessWidget {
16   // Posisi awal kamera
17   static const CameraPosition _initialCameraPosition = CameraPosition(
18     target: LatLng(-6.1751, 106.8650), // Lokasi awal (contoh: Jakarta)
19     zoom: 12.0, // Tingkat zoom awal
20   );
21
22   @override
23   Widget build(BuildContext context) {
24     return Scaffold(
25       appBar: AppBar(
26         title: Text('Google Maps in Flutter'),
27       ),
28       body: GoogleMap(
29         initialCameraPosition: _initialCameraPosition,
30         mapType: MapType.normal, // Jenis peta
31         myLocationEnabled: true, // Menampilkan lokasi pengguna (jika izin diberikan)
32         zoomControlsEnabled: true, // Menampilkan kontrol zoom
33         onMapCreated: (GoogleMapController controller) {
34           // Callback saat peta berhasil dimuat
35           print('Google Map Created');
36         },
37       ),
38     );
39   }
40 }
41
```

b. Menggunakan Posisi Awal Camera

Posisi awal kamera pada Google Maps diatur menggunakan properti `initialCameraPosition` pada widget `GoogleMap`.

```
initialCameraPosition: CameraPosition(  
  target: _mumbaiLocation,  
  zoom: 12.0,  
), // CameraPosition
```

Dalam pengaturan peta, terdapat beberapa parameter penting untuk mengontrol tampilan awal. **Target** digunakan untuk menentukan koordinat lokasi awal pada peta dengan format `LatLng`(latitude, longitude), yang menunjuk ke posisi geografis spesifik. **Zoom** menentukan tingkat kedekatan tampilan peta, di mana semakin besar angka zoom, semakin dekat peta terhadap lokasi tersebut. Selain itu, terdapat parameter opsional seperti **tilt**, yang mengatur kemiringan sudut pandang kamera untuk memberikan efek perspektif, dan **bearing**, yang menentukan rotasi kamera terhadap arah utara dalam satuan derajat, memungkinkan orientasi peta diubah sesuai kebutuhan.

c. Properti Utama Widget `GoogleMap` dan Fungsinya

Properti	Fungsi
<code>initialCameraPosition</code>	Menentukan posisi awal kamera saat peta pertama kali ditampilkan.
<code>mapType</code>	Mengatur jenis peta yang ditampilkan (normal, satellite, hybrid, terrain, none).
<code>myLocationEnabled</code>	Menampilkan ikon lokasi pengguna jika izin akses lokasi telah diberikan
<code>zoomControlsEnabled</code>	Menampilkan kontrol zoom (+/-) pada peta
<code>onMapCreated</code>	Callback yang dipanggil saat Google Maps berhasil dimuat, berguna untuk inisialisasi.
<code>markers</code>	Menampilkan penanda (marker) di lokasi tertentu pada peta menggunakan <code>Set&lt;Marker&gt;</code> .
<code>Polylines</code>	Menampilkan garis lintasan pada peta menggunakan <code>Set&lt;Polyline&gt;</code> .
<code>Circles</code>	Menampilkan lingkaran pada peta menggunakan <code>Set&lt;Circle&gt;</code> .
<code>onTap</code>	Callback yang dipanggil saat pengguna mengetuk peta
<code>onCameraMove</code>	Callback saat kamera (pemandangan peta) digeser atau diperbesar/dikecilkan

### 3. Menambahkan markers

#### a. Kode untuk Menambahkan Marker di Posisi Tertentu

Berikut adalah contoh kode untuk menambahkan marker di lokasi dengan koordinat tertentu (latitude: -6.2088, longitude: 106.8456):

```
1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3
4 void main() => runApp(MyApp());
5
6 class MyApp extends StatelessWidget {
7   @override
8   Widget build(BuildContext context) {
9     return MaterialApp(
10       home: MapScreen(),
11     );
12   }
13 }
14
15 class MapScreen extends StatefulWidget {
16   @override
17   _MapScreenState createState() => _MapScreenState();
18 }
19
20 class _MapScreenState extends State<MapScreen> {
21   static const CameraPosition _initialCameraPosition = CameraPosition(
22     target: LatLng(-6.2088, 106.8456), // Jakarta
23     zoom: 12.0,
24   );
25
26   final Set<Marker> _markers = {};
27
28   @override
29   void initState() {
30     super.initState();
31     _addMarker();
32   }
33
34   void _addMarker() {
35     _markers.add(
36       Marker(
37         markerId: MarkerId('jakarta'),
38         position: LatLng(-6.2088, 106.8456), // Posisi marker
39         infoWindow: InfoWindow(
40           title: 'Jakarta',
41           snippet: 'Ibu Kota Indonesia',
42         ),
43         icon: BitmapDescriptor.defaultMarker, // Warna default marker
44       ),
45     );
46   }
47
48   @override
49   Widget build(BuildContext context) {
50     return Scaffold(
51       appBar: AppBar(
52         title: Text('Google Maps with Marker'),
53       ),
54       body: GoogleMap(
55         initialCameraPosition: _initialCameraPosition,
56         markers: _markers, // Set marker di peta
57         onMapCreated: (GoogleMapController controller) {
58           print('Map Created');
59         },
60       ),
61     );
62   }
63 }
64
```

b. Menampilkan Info Window saat Marker Diklik

Info Window akan muncul otomatis jika infoWindow diatur pada marker. Untuk menyesuaikan atau menambah logika saat marker diklik, gunakan properti onTap pada Marker.

```
markers: {  
  Marker(  
    markerId: const MarkerId('mumbai'),  
    position: _mumbaiLocation,  
    infoWindow: const InfoWindow(title: 'Gateway of India Mumbai'),  
    onTap: () {  
      print('Marker Jakarta diklik!');  
    },  
  ), // Marker
```

InfoWindow muncul otomatis saat marker diklik, kecuali diatur menjadi kosong dengan InfoWindow.noText. Anda bisa menambahkan judul dan deskripsi di dalamnya. Untuk menyembunyikan InfoWindow secara manual, gunakan controller peta dengan hideMarkerInfoWindow(markerId).

4. Menggunakan place picker

a. Apa itu place picker ?

Place Picker adalah fitur yang memungkinkan pengguna memilih lokasi pada peta secara interaktif. Biasanya, Place Picker digunakan dalam aplikasi untuk memudahkan pengguna menentukan lokasi seperti alamat pengiriman, titik pertemuan, atau lokasi favorit.

Cara Kerja Place Picker di Flutter:

- 1) Aplikasi menampilkan peta interaktif menggunakan Google Maps.
- 2) Pengguna memilih lokasi dengan mengetuk atau menyeret pin (marker) di peta.
- 3) Lokasi yang dipilih (dalam bentuk koordinat latitude dan longitude) dikembalikan ke aplikasi.

Nama Package untuk Implementasi Place Picker:

- Salah satu package populer untuk Place Picker di Flutter adalah: flutter\_google\_places atau google\_maps\_place\_picker.

b. Kode untuk Menampilkan Place Picker

Contoh berikut menggunakan google\_maps\_place\_picker untuk menampilkan Place Picker:

Langkah-langkah

- 1) Tambahkan package ke dalam proyek, setelah itu ketik perintah flutter pub get di terminal

```
dependencies:  
  google_maps_place_picker: ^3.0.2
```

- 2) Aktifkan Places API di Google Cloud Console dan pastikan API Key Anda mendukung layanan ini.
- 3) Tambahkan kode berikut untuk menampilkan Place Picker:

```

1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3 import 'package:google_maps_place_picker/google_maps_place_picker.dart';
4
5 void main() => runApp(MyApp());
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: PlacePickerScreen(),
12     );
13   }
14 }
15
16 class PlacePickerScreen extends StatelessWidget {
17   final String googleMapsApiKey = "YOUR_API_KEY"; // Masukkan API Key Google Anda
18
19   @override
20   Widget build(BuildContext context) {
21     return Scaffold(
22       appBar: AppBar(
23         title: Text("Place Picker"),
24       ),
25       body: Center(
26         child: ElevatedButton(
27           onPressed: () async {
28             // Navigasikan ke halaman Place Picker
29             final result = await Navigator.push(
30               context,
31               MaterialPageRoute(
32                 builder: (context) => PlacePicker(
33                   apiKey: googleMapsApiKey,
34                   initialPosition: LatLng(-6.1751, 106.8650), // Lokasi awal (Jakarta)
35                   useCurrentLocation: true, // Gunakan lokasi pengguna jika tersedia
36                   onPlacePicked: (result) {
37                     print("Lokasi dipilih: ${result.geometry?.location.lat}, ${result.geometry?.location.lng}");
38                     Navigator.of(context).pop(result);
39                   },
40                 ),
41               ),
42             );
43
44             if (result != null) {
45               ScaffoldMessenger.of(context).showSnackBar(
46                 SnackBar(
47                   content: Text(
48                     "Lokasi yang dipilih: (${result.geometry?.location.lat}, ${result.geometry?.location.lng})",
49                   ),
50                 ),
51               );
52             }
53           },
54           child: Text("Pilih Lokasi"),
55         ),
56       ),
57     );
58   }
59 }
60

```