

**TUGAS PENDAHULUAN  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XII  
MAPS & PLACES**



**Disusun Oleh :**

**Dhiya Ulhaq R 2211104053**

**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**

## TUGAS PENDAHULUAN

### SOAL

#### 1. Menambahkan Google Maps Package

- 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.
- Mengapa kita perlu menambahkan API Key, dan di mana API Key tersebut diatur dalam aplikasi Flutter?

**Jawab :**

- Package yang digunakan untuk mengintegrasikan Google Maps di Flutter adalah `google_maps_flutter`.

Tambahkan dependency di `pubspec.yaml` :

```
30 dependencies:
31   flutter:
32     sdk: flutter
33   cupertino_icons: ^1.0.8
34   google_maps_flutter: ^2.10.0
```

Lalu tambahkan permission di `AndroidManifest.xml`

```
android > app > src > main > AndroidManifest.xml
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
3   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

- Mengautentikasi aplikasi ke layanan Google Maps

- Melacak penggunaan API dan menghitung biaya
- Mengakses fitur-fitur premium Google Maps
- Keamanan aplikasi dari penyalahgunaan

API Key diatur di:

Android: di file `android/app/src/main/AndroidManifest.xml`

```
32 <meta-data
33     android:name="flutterEmbedding"
34     android:value="2" />
35 </application>
```

#### 2. Menampilkan Google Maps

- a. Tuliskan kode untuk menampilkan Google Map di Flutter menggunakan widget `GoogleMap`.

```
lib > screens > google_maps_screen.dart > GoogleMapsScreen
1  import 'package:flutter/material.dart';
2  import 'package:google_maps_flutter/google_maps_flutter.dart';
3
4  class GoogleMapsScreen extends StatefulWidget {
5    const GoogleMapsScreen({Key? key}) : super(key: key);
6
7    @override
8    State<GoogleMapsScreen> createState() => _GoogleMapsScreenState();
9  }
10 class _GoogleMapsScreenState extends State<GoogleMapsScreen> {
11   late GoogleMapController mapController;
12
13   static const CameraPosition initialPosition = CameraPosition(
14     target: LatLng(-6.2088, 106.8456),
15     zoom: 14.0,
16   ); // CameraPosition
17
18   @override
19   Widget build(BuildContext context) {
20     return Scaffold(
21       appBar: AppBar(
22         title: const Text('Google Maps Demo'),
23       ), // AppBar
24       body: GoogleMap(
25         initialCameraPosition: initialPosition,
26         onMapCreated: (GoogleMapController controller) {
27           mapController = controller;
28         },
29         zoomControlsEnabled: true,
30         myLocationButtonEnabled: true,
31         myLocationEnabled: true,
32         mapType: MapType.normal,
33         compassEnabled: true,
34       ), // GoogleMap
35     ); // Scaffold
36   }
37 }
```

- b. Bagaimana cara menentukan posisi awal kamera (camera position) pada Google Maps di Flutter?

```
13 CameraPosition initialPosition = CameraPosition(
14   target: LatLng(-6.2088, 106.8456), // Koordinat lokasi
15   zoom: 14.0, // Level zoom (0-20)
16   tilt: 0.0, // Kemiringan kamera (0-90 derajat)
17   bearing: 0.0 // Rotasi kamera (0-360 derajat)
```

- c. Sebutkan properti utama dari widget `GoogleMap` dan fungsinya.
- *initialCameraPosition*: Menentukan posisi awal kamera saat peta dimuat
  - *onMapCreated*: Callback yang dipanggil saat peta selesai dibuat
  - *myLocationEnabled*: Mengaktifkan titik lokasi pengguna saat ini
  - *myLocationButtonEnabled*: Menampilkan tombol untuk ke lokasi pengguna
  - *mapType*: Jenis tampilan peta (normal, satellite, hybrid, terrain)
  - *markers*: Kumpulan marker yang ditampilkan di peta
  - *zoomControlsEnabled*: Menampilkan tombol kontrol zoom
  - *compassEnabled*: Menampilkan kompas saat peta dirotasi

### 3. Menambahkan Marker

- a. Tuliskan kode untuk menambahkan marker di posisi tertentu (latitude: -6.2088, longitude: 106.8456) pada Google Maps.

```
22 void _addMarker() {
23   setState(() {
24     markers.add(
25       Marker(
26         markerId: const MarkerId('custom_location'),
27         position: const LatLng(-6.2088, 106.8456),
```

Bagaimana cara menampilkan info window saat marker diklik?

```
28 // Menambahkan info window yang muncul saat marker diklik
29 infoWindow: const InfoWindow(
30   title: 'Lokasi Kustom',
31   snippet: 'Ini adalah lokasi yang ditandai',
32 ), // InfoWindow
33 // Mengaktifkan info window dapat diklik
34 onTap: () {
35   print('Marker diklik!');
36 },
```

### 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.

Jawaban :

Place Picker adalah fitur yang memungkinkan pengguna memilih lokasi menggunakan search box dengan auto-complete. Package yang digunakan adalah `google_places_flutter`. Cara kerjanya:

- Pengguna mengetik nama lokasi di search box
- Google Places API memberikan suggestions
- Saat lokasi dipilih, kita mendapatkan detail lokasi termasuk koordinat
- Kita dapat memindahkan kamera ke lokasi yang dipilih

- b. Tuliskan kode untuk menampilkan Place Picker, lalu kembalikan lokasi yang dipilih oleh pengguna dalam bentuk latitude dan longitude.

```

1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3 import 'package:google_places_flutter/google_places_flutter.dart';
4 import 'package:google_places_flutter/model/prediction.dart';
5
6 class PlacePickerImplementation extends StatefulWidget {
7   const PlacePickerImplementation({Key? key}) : super(key: key);
8
9   @override
10  State<PlacePickerImplementation> createState() => _PlacePickerImplementationState();
11 }
12
13 class _PlacePickerImplementationState extends State<PlacePickerImplementation> {
14   LatLng? selectedLocation;
15   late GoogleMapController mapController;
16
17   @override
18   Widget build(BuildContext context) {
19     return Column(
20       children: [
21         PlacesAutocompleteField(
22           apiKey: 'YOUR_API_KEY',
23           onPlaceSelected: (Prediction? prediction) {
24             if (prediction != null) {
25               setState(() {
26                 selectedLocation = LatLng(
27                   double.parse(prediction.lat ?? '-6.2088'),
28                   double.parse(prediction.lng ?? '106.8456'),
29                 ); // LatLng
30               });
31
32               // Tampilkan koordinat yang dipilih
33               print('Latitude: ${selectedLocation!.latitude}');
34               print('Longitude: ${selectedLocation!.longitude}');
35
36               // Pindahkan kamera ke lokasi yang dipilih
37               mapController.animateCamera(

```

```

38                 CameraUpdate.newLatLng(selectedLocation!),
39               );
40             }
41           },
42         ),
43         Expanded(
44           child: GoogleMap(
45             initialCameraPosition: const CameraPosition(
46               target: LatLng(-6.2088, 106.8456),
47               zoom: 14.0,
48             ), // CameraPosition
49             onMapCreated: (controller) => mapController = controller,
50             markers: selectedLocation != null ? {
51               Marker(
52                 markerId: const MarkerId('selected_place'),
53                 position: selectedLocation!,
54               ), // Marker
55             } : {},
56           ), // GoogleMap
57         ), // Expanded
58       ],
59     ); // Column
60   }
61 }

```