

**TUGAS PENDAHULUAN  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XII  
MAPS AND 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**

## 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.

**Jawab:**

Nama packagenya adalah google\_maps\_flutter. Berikut adalah langkah-langkah untuk menambahkan package google\_maps\_flutter:

➤ **Tambahkan package ke pubspec.yaml:**

```
dependencies:  
  google_maps_flutter: ^2.2.5
```

➤ **Konfigurasi API Key:**

- Buka file android/app/src/main/AndroidManifest.xml.
- Tambahkan baris berikut di dalam elemen <application>:  
xml

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

➤ **Tambahkan izin lokasi (opsional):**

Jika aplikasi memerlukan akses lokasi, tambahkan izin di file AndroidManifest.xml (untuk Android) dan Info.plist (untuk iOS).

➤ **Gunakan widget GoogleMap:**

- Import package di file Dart: dart

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

- Gunakan widget GoogleMap dalam layout:

```
GoogleMap(  
  initialCameraPosition: CameraPosition(  
    target: LatLng(-7.797068, 110.370529), // Koordinat  
    Yogyakarta  
    zoom: 14.0,  
  ),  
);
```

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

**Jawab:**

Alasan menambahkan API Key:

1. **Identifikasi aplikasi:** API Key digunakan oleh Google untuk mengenali dan mengautentikasi aplikasi yang mengakses layanan Google Maps.
2. **Pengaturan penggunaan:** API Key membantu Google membatasi dan memantau penggunaan layanan (kuota API).
3. **Keamanan:** API Key memastikan bahwa hanya aplikasi yang memiliki akses yang diotorisasi yang dapat menggunakan fitur Google Maps.

Tempat mengatur API Key dalam aplikasi Flutter:

1. **Untuk Android:** API Key diatur di file AndroidManifest.xml, di dalam elemen <application>:

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

2. **Untuk iOS:** API Key diatur di file AppDelegate.swift atau AppDelegate.m menggunakan:

```
GMSServices.provideAPIKey("YOUR_API_KEY_HERE")
```

## 2. Menampilkan Google Maps

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

**Jawab:**

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: GoogleMapScreen(),
    );
  }
}

class GoogleMapScreen extends StatefulWidget {
```

```

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

class _GoogleMapScreenState extends State<GoogleMapScreen> {
  // Inisialisasi posisi awal kamera
  final CameraPosition _initialCameraPosition = CameraPosition(
    target: LatLng(-7.797068, 110.370529), // Koordinat Yogyakarta
    zoom: 14.0,
  );

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Google Map Example"),
      ),
      body: GoogleMap(
        initialCameraPosition: _initialCameraPosition,
      ),
    );
  }
}

```

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

**Jawab:**

Untuk menentukan posisi awal kamera (camera position) pada Google Maps di Flutter, gunakan properti `initialCameraPosition` dalam widget `GoogleMap`. Posisi ini ditentukan dengan objek `CameraPosition`, yang membutuhkan properti berikut:

1. **target**: Lokasi pusat peta dalam format `LatLng(latitude, longitude)`.
2. **zoom**: Tingkat pembesaran peta (semakin tinggi nilainya, semakin dekat pandangan peta).

Contoh Kode:

```

CameraPosition _initialCameraPosition = CameraPosition(
  target: LatLng(-7.797068, 110.370529), // Yogyakarta
  zoom: 14.0,
);

```

Kemudian pasang ke widget GoogleMap:

```
GoogleMap(  
    initialCameraPosition: _initialCameraPosition,  
);
```

- c. Sebutkan properti utama dari widget GoogleMap dan fungsinya.

**Jawab:**

Berikut properti utama dari widget GoogleMap beserta fungsinya:

**1. initialCameraPosition**

**Fungsi:** Menentukan posisi awal kamera pada peta saat widget pertama kali dirender.

**2. mapType**

**Fungsi:** Menentukan jenis tampilan peta (normal, satelit, terrain, atau hybrid). Contoh:

```
mapType: MapType.satellite,
```

**3. markers**

**Fungsi:** Menampilkan penanda (marker) di lokasi tertentu. Contoh:

```
markers: {  
    Marker(  
        markerId: MarkerId("1"),  
        position: LatLng(-7.797068, 110.370529),  
        infoWindow: InfoWindow(title: "Yogyakarta"),  
    ),  
},
```

**4. onMapCreated**

**Fungsi:** Callback yang dipanggil saat peta selesai dimuat. Biasanya digunakan untuk mendapatkan referensi ke kontrol peta.

**5. myLocationEnabled**

**Fungsi:** Menampilkan ikon lokasi pengguna di peta jika diatur ke true.

**6. zoomControlsEnabled**

**Fungsi:** Menampilkan kontrol pembesaran peta (tombol zoom in/out).

**7. polylines**

**Fungsi:** Menampilkan garis lintasan (polylines) di atas peta.

**8. circles**

**Fungsi:** Menambahkan lingkaran ke peta, misalnya untuk radius area tertentu.

```

GoogleMap(
  initialCameraPosition: CameraPosition(
    target: LatLng(-7.797068, 110.370529),
    zoom: 14.0,
  ),
  mapType: MapType.normal,
  markers: {
    Marker(
      markerId: MarkerId("1"),
      position: LatLng(-7.797068, 110.370529),
      infoWindow: InfoWindow(title: "Yogyakarta"),
    ),
  },
  myLocationEnabled: true,
  zoomControlsEnabled: true,
);

```

### 3. Menambahkan Marker

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

**Jawab:**

```

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: GoogleMapScreen(),
    );
  }
}

class GoogleMapScreen extends StatefulWidget {
  @override
  _GoogleMapScreenState createState() => _GoogleMapScreenState();
}

class _GoogleMapScreenState extends State<GoogleMapScreen> {
  final CameraPosition _initialCameraPosition = CameraPosition(
    target: LatLng(-6.2088, 106.8456), // Posisi Jakarta
    zoom: 14.0,
  );

  final Set<Marker> _markers = {};

```

```

@override
void initState() {
  super.initState();
  _addMarker(); // Menambahkan marker saat layar dimuat
}

void _addMarker() {
  setState(() {
    _markers.add(
      Marker(
        markerId: MarkerId("marker_jakarta"),
        position: LatLng(-6.2088, 106.8456), // Koordinat Jakarta
        infoWindow: InfoWindow(
          title: "Jakarta",
          snippet: "Ibu Kota Indonesia",
        ),
      ),
    );
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Menambahkan Marker"),
    ),
    body: GoogleMap(
      initialCameraPosition: _initialCameraPosition,
      markers: _markers,
    ),
  );
}

```

- b. Bagaimana cara menampilkan info window saat marker diklik?

Untuk menampilkan **info window** saat marker diklik, Anda hanya perlu menambahkan properti **infoWindow** pada objek Marker. Info window ini akan otomatis muncul ketika marker diklik.

**Jawab:**

Contoh implementasi:

```

Marker(
  markerId: MarkerId("marker_jakarta"),
  position: LatLng(-6.2088, 106.8456),
  infoWindow: InfoWindow(
    title: "Jakarta",
    snippet: "Ibu Kota Indonesia",
  ),
)

```

Pada kode di atas:

- **title:** Menampilkan judul pada info window.
- **snippet:** Menampilkan deskripsi singkat atau detail tambahan.

#### 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.

**Jawab:**

**Place Picker** adalah fitur yang memungkinkan pengguna untuk memilih lokasi dari peta secara interaktif. Di Flutter, Place Picker biasanya digunakan dengan memanfaatkan Google Maps API atau library lain untuk menyediakan tampilan peta yang interaktif.

1. **Cara kerja Place Picker di Flutter:**

- Integrasi Google Maps API untuk menampilkan peta.
- Pengguna dapat menavigasi dan memilih lokasi yang diinginkan.
- Lokasi yang dipilih kemudian dikembalikan dalam bentuk koordinat geografis (latitude dan longitude).
- Informasi ini dapat digunakan untuk keperluan lain, seperti menyimpan lokasi atau menampilkan alamat.

2. **Nama package untuk implementasi Place Picker di Flutter:** Salah satu package yang populer adalah:

- `place_picker`
- `google_maps_flutter` (digabungkan dengan layanan Geocoding jika diperlukan).

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

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: PlacePickerExample(),
    );
  }
}

class PlacePickerExample extends StatelessWidget {
  final String googleMapsApiKey =
    "YOUR_GOOGLE_MAPS_API_KEY"; // Ganti dengan API key Anda
```



```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text("Place Picker Example")),
    body: Center(
      child: ElevatedButton(
        onPressed: () async {
          // Menampilkan Place Picker
          LocationResult result = await Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => PlacePicker(googleMapsApiKey),
            ),
          );

          // Menampilkan latitude dan longitude yang dipilih
          if (result != null) {
            print("Latitude: ${result.latLng?.latitude}");
            print("Longitude: ${result.latLng?.longitude}");
          }
        },
        child: Text("Pilih Lokasi"),
      ),
    ),
  );
}

```

#### Penjelasan Kode:

1. **Library yang digunakan:** place\_picker untuk integrasi Google Maps dan pemilihan lokasi.
2. **Langkah-langkah dalam kode:**
  - Tekan tombol untuk membuka Place Picker.
  - Pilih lokasi dari peta.
  - Lokasi dipilih dalam bentuk objek LocationResult, yang berisi data latLng (latitude dan longitude).
3. **Catatan:** Pastikan Anda mengaktifkan Google Places API dan Maps API di Google Cloud Console serta mengganti YOUR\_GOOGLE\_MAPS\_API\_KEY dengan API Key Anda.