

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XII
MAPS & PLACES**



Disusun Oleh :

M. Edgar Nadhif

2211104019

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.

google_maps_flutter.

Langkah-Langkah Menambahkan Google Maps ke Proyek Flutter :

1. Tambahkan Dependency

Buka file pubspec.yaml di proyek Flutter Anda, lalu tambahkan:

google_maps_flutter: versionnya

Kemudian jalankan perintah flutter pub get

2. Tambahkan API Key Google maps

- a. Buat atau akses Google Cloud Console.
- b. Aktifkan Google Maps SDK for Android dan Google Maps SDK for iOS.
- c. Dapatkan **API Key**, lalu tambahkan ke file berikut:

i. **Android:** Tambahkan di

android/app/src/main/AndroidManifest.xml di dalam tag
<application> :

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

3. Konfigurasi Platform Spesifik

Android: Pastikan versi minimum SDK di android/app/build.gradle adalah minSdkVersion 20 atau lebih tinggi.

4. Gunakan Google Maps dalam Widget
- b. Mengapa kita perlu menambahkan API Key, dan di mana API Key tersebut diatur dalam aplikasi Flutter?
- **Autentikasi dan Otorisasi:**
API Key berfungsi untuk mengenali dan memverifikasi aplikasi yang mengakses layanan Google Maps. Ini memastikan bahwa hanya aplikasi yang telah diotorisasi yang dapat menggunakan fitur Google Maps.
 - **Keamanan dan Pembatasan Penggunaan:**
API Key memungkinkan pembatasan akses untuk mencegah penyalahgunaan, seperti pembatasan domain, IP, atau referer. Hal ini melindungi proyek Anda dari penggunaan yang tidak sah

2. Menampilkan Google Maps

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

```
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: Scaffold(
        appBar: AppBar(
          title: Text('Google Map Example'),
        ),
        body: GoogleMap(
          initialCameraPosition: CameraPosition(
```

```

        target: LatLng(-6.200000, 106.816666), // Koordinat Jakarta

        zoom: 14.0, // Zoom level awal

    ),

    mapType: MapType.normal, // Jenis peta

    onMapCreated: (GoogleMapController controller) {

        print("Google Map berhasil dimuat!");

    },

),

),

);

}

}

```

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

Posisi awal kamera ditentukan menggunakan properti `initialCameraPosition` dari widget `GoogleMap`.

Untuk menentukan posisi awal:

Gunakan objek `CameraPosition` untuk mendefinisikan lokasi awal kamera.

Berikan properti berikut :

`target`: Koordinat lokasi awal (menggunakan `LatLng(latitude, longitude)`).

`zoom`: Level zoom awal (semakin besar nilai, semakin dekat).

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

- `initialCameraPosition`
Fungsi: Menentukan posisi awal kamera (latitude, longitude, dan zoom level).
- `mapType`
Fungsi: Menentukan jenis peta yang ditampilkan. Nilai: `normal`, `satellite`, `terrain`, atau `hybrid`.
- `onMapCreated`
- Fungsi: Callback yang dipanggil saat peta selesai dimuat, berguna untuk mendapatkan controller peta.

- `markers`
Fungsi: Menambahkan marker (penanda) pada lokasi tertentu di peta.
- `polylines`
Fungsi: Menambahkan garis poligon untuk menandai rute atau area tertentu di peta.
- `circles`
Fungsi: Menambahkan lingkaran untuk menyoroti area di sekitar koordinat tertentu.
- `zoomControlsEnabled`
Fungsi: Menentukan apakah tombol zoom (+/-) ditampilkan pada peta.
- `myLocationEnabled`
Fungsi: Menampilkan lokasi pengguna (jika izin akses lokasi telah diberikan).
- `onTap`
Fungsi: Callback yang dipanggil saat peta ditekan (berguna untuk interaksi pengguna).
- `onCameraMove`
Fungsi: Callback saat kamera bergerak (misalnya karena drag atau zoom oleh pengguna).

3. Menambahkan Marker

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

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

import 'package:google_maps_flutter/google_maps_flutter.dart';

void main() => runApp(MyApp());

class MyApp extends StatefulWidget {

  @override

  _MyAppState createState() => _MyAppState();

}

class _MyAppState extends State<MyApp> {

  late GoogleMapController _controller;

  // Set berisi marker
```

```

final Set<Marker> _markers = {};

@override

void initState() {
    super.initState();
    _addMarker();
}

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

@override

Widget build(BuildContext context) {
    return MaterialApp(
        home: Scaffold(
            appBar: AppBar(

```

```

        title: Text('Google Maps Marker Example'),
      ),
      body: GoogleMap(
        initialCameraPosition: CameraPosition(
          target: LatLng(-6.2088, 106.8456), // Posisi awal kamera
          zoom: 14.0, // Zoom level
        ),
        markers: _markers, // Tambahkan marker ke Google Map
        onMapCreated: (GoogleMapController controller) {
          _controller = controller;
        },
      ),
    ),
  );
}
}

```

- b. Bagaimana cara menampilkan info window saat marker diklik?
 Info Window dapat ditampilkan dengan menambahkan properti `infoWindow` pada Marker. Info Window akan muncul secara otomatis saat marker diklik. Anda bisa mengatur:
- `title`: Teks utama pada Info Window.
 - `snippet`: Teks tambahan di bawah judul.

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.

Place Picker adalah fitur yang memungkinkan pengguna memilih lokasi di peta, biasanya dilengkapi dengan antarmuka interaktif untuk memilih lokasi secara manual atau menggunakan pencarian. Dalam Flutter, Place Picker membantu mendapatkan detail lokasi, seperti latitude, longitude, dan nama tempat.

Cara Kerjanya di Flutter:

1. Peta ditampilkan melalui widget Place Picker.
2. Pengguna memilih lokasi dengan cara mengklik peta atau mencari nama tempat.
3. Aplikasi mengembalikan data lokasi yang dipilih, seperti koordinat (latitude dan longitude) atau nama tempat.

Nama Package untuk Implementasi Place Picker di Flutter:

Package yang sering digunakan adalah `google_maps_place_picker`.

- 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:google_maps_flutter/google_maps_flutter.dart';
import
  'package:google_maps_place_picker/google_maps_place_picker.da
  rt';

void main() => runApp(MyApp());

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

class PlacePickerExample extends StatelessWidget {
  static const String googleApiKey = "YOUR_GOOGLE_API_KEY";
  // Masukkan API Key
```



```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Place Picker Example'),
    ),
    body: Center(
      child: ElevatedButton(
        onPressed: () async {
          // Navigasi ke halaman Place Picker
          final result = await Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => PlacePicker(
                apiKey: googleApiKey,
                initialPosition: LatLng(-6.2088, 106.8456), // Posisi awal
                (Jakarta)
                useCurrentLocation: true, // Gunakan lokasi saat ini
                onPlacePicked: (result) {
                  Navigator.pop(context, result); // Kembalikan hasil
                },
              ),
            ),
          );

          if (result != null) {
            // Tampilkan hasil lokasi yang dipilih
            print("Latitude: ${result.geometry?.location.lat}");
            print("Longitude: ${result.geometry?.location.lng}");
          }
        },
        child: Text('Pilih Lokasi'),
      ),
    ),
  );
}

```

	}
}	