

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XII
MAPS & PLACES**



**Disusun Oleh :
Nadia Putri Rahmانيar / 2211104012
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 package yang digunakan adalah `google_maps_flutter`. Berikut adalah langkah-langkah untuk menambahkan package `google_maps_flutter` ke dalam proyek Flutter:

- Buka file `pubspec.yaml` di proyek Anda, kemudian tambahkan package berikut di bawah `dependencies`:

```
dependencies:  
  google_maps_flutter: ^2.10.0
```

- Setelah itu, jalankan perintah **`flutter pub get`** di terminal untuk mengunduh package
- Tambahkan API Key yang diperlukan untuk menggunakan Google Maps. Untuk Android, tambahkan API Key ke file `android/app/src/main/AndroidManifest.xml` seperti ini:

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

- Lalu tambahkan juga versi minimum SDK di dalam file `android/app/build.gradle`

```
defaultConfig {  
    minSdkVersion 20  
}
```

- Menambahkan widget Google Maps di dalam file dart, misalkan:

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Google Maps Example")),
        body: GoogleMap(
          initialCameraPosition: const CameraPosition(
            target: LatLng(-7.797068, 110.370529), // Posisi awal: Yogyakarta
            zoom: 12.0,
          ),
        ),
      ),
    );
  }
}
```

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

Jawab :

API Key diperlukan untuk mengautentikasi aplikasi saat mengakses layanan Google Maps. Google menggunakan API Key untuk:

1. Identifikasi: Memastikan bahwa permintaan datang dari aplikasi Anda.
2. Hak Akses: Mengontrol fitur Google Maps yang dapat digunakan oleh aplikasi, berdasarkan pengaturan di Google Cloud Console.
3. Pembatasan: Mencegah penyalahgunaan API oleh pihak lain dengan menerapkan pembatasan (misalnya berdasarkan domain atau IP).

Tanpa API Key, aplikasi tidak akan bisa menggunakan fitur Google Maps.

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(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Google Maps Example")),
        body: GoogleMap(
          initialCameraPosition: const CameraPosition(
            target: LatLng(-7.797068, 110.370529), // Posisi awal: Yogyakarta
            zoom: 12.0,
          ),
        ),
      ),
    );
  }
}
```

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

Posisi awal kamera di Google Maps ditentukan melalui properti `initialCameraPosition` dari widget `GoogleMap`. Properti ini membutuhkan objek `CameraPosition` yang mencakup:

- `target`: Objek `LatLng` berisi koordinat geografis (latitude, longitude).
- `zoom`: Tingkat pembesaran peta.

```
CameraPosition(
  target: LatLng(-7.797068, 110.370529), // Posisi Yogyakarta
  zoom: 15.0,
);
```

c. Sebutkan properti utama dari widget GoogleMap dan fungsinya.

Jawab :

- `initialCameraPosition` : digunakan untuk menentukan posisi awal kamera peta. Posisi tersebut mencakup koordinat geografis (latitude dan longitude), tingkat zoom, kemiringan kamera, dan rotasi terhadap utara.
- `markers` : menampilkan marker atau penanda di lokasi tertentu di peta. Markers didefinisikan menggunakan `Set<Marker>`.
- `mapType` : mengatur jenis tampilan peta, seperti peta standar, satelit, gabungan standar dan satelit (hybrid), atau peta medan (terrain).
- `onMapCreated` : Callback ini dipanggil ketika peta selesai dimuat. Fungsi ini memberikan instance `GoogleMapController` untuk mengontrol peta lebih lanjut.
- `myLocationEnabled` : jika diaktifkan, properti ini akan menampilkan lokasi pengguna saat ini di peta, asalkan izin lokasi diberikan.
- `myLocationButtonEnabled` : jika diaktifkan, properti ini akan menampilkan tombol yang memungkinkan pengguna memusatkan kamera ke lokasi mereka saat ini.
- `zoomControlsEnabled` : menentukan apakah kontrol zoom (+/-) ditampilkan di peta.
- `polylines` : memungkinkan penambahan garis atau jalur pada peta. Garis tersebut didefinisikan dengan `Set<Polyline>`.
- `circles` : menambahkan lingkaran ke peta, biasanya untuk menyoroti area tertentu.
- `gestureRecognizers` : mengatur interaksi pengguna dengan peta, seperti kemampuan untuk menggeser, memperbesar, atau mengetuk peta.

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(const MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: const MarkerExampleScreen(),
    );
  }
}

class MarkerExampleScreen extends StatefulWidget {
  const MarkerExampleScreen({super.key});

  @override
  State<MarkerExampleScreen> createState() =>
  _MarkerExampleScreenState();
}

class _MarkerExampleScreenState extends
State<MarkerExampleScreen> {
```

```

final Set<Marker> _markers = {};

@override
void initState() {
    super.initState();

    // Menambahkan marker di posisi tertentu
    _markers.add(
        const 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 Scaffold(
        appBar: AppBar(
            title: const Text("Marker Example"),
        ),
        body: GoogleMap(
            initialCameraPosition: const CameraPosition(
                target: LatLng(-6.2088, 106.8456), // Fokus
awal: Jakarta

                zoom: 12.0,

```

```

    ),
    markers: _markers, // Menambahkan marker ke peta
  ),
);
}
}

```

b. Bagaimana cara menampilkan info window saat marker diklik?

Jawab :

- Tambahkan properti infoWindow pada objek Marker.
 - Properti ini memiliki parameter seperti:
 - title: Menampilkan judul di info window.
 - snippet: Menampilkan deskripsi tambahan di bawah judul.
- Info window akan otomatis muncul ketika marker diklik. Jika Anda ingin menambahkan logika tertentu saat marker diklik, gunakan callback onTap di dalam Marker.

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 pada peta secara interaktif. Biasanya digunakan dalam aplikasi yang membutuhkan lokasi pengguna seperti alamat, titik koordinat, atau tempat tertentu. Cara kerjanya yaitu :

1. Menampilkan peta: Place Picker menggunakan Google Maps API untuk menampilkan peta.
2. Pengguna memilih lokasi: Lokasi dapat dipilih dengan menekan pin atau memindahkan peta ke tempat tertentu.
3. Mengembalikan data lokasi: Setelah lokasi dipilih, aplikasi mengembalikan data berupa koordinat geografis (latitude dan longitude) atau informasi tempat.

Package untuk Implementasi Place Picker: google_maps_place_picker.

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

Jawab :

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: const PlacePickerExample(),
    );
  }
}

class PlacePickerExample extends StatelessWidget {
  const PlacePickerExample({super.key});

  final String apiKey = "YOUR_API_KEY"; // Ganti dengan
API Key Google Anda

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Place Picker Example"),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () async {
            // Navigasi ke Place Picker
            final result = await Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => PlacePicker(
                  apiKey,
                  initialPosition: const LatLng(-6.2088,
106.8456), // Jakarta
                  useCurrentLocation: true, // Gunakan
```

```
lokasi saat ini

        onPlacePicked: (result) {
            print("Alamat:
${result.formattedAddress}");
            print("Latitude:
${result.geometry!.location.lat}");
            print("Longitude:
${result.geometry!.location.lng}");
            Navigator.pop(context,
result.geometry!.location);
        },
    ),
),
);

    if (result != null) {
        print("Lokasi dipilih: $result");
    }
},
child: const Text("Pilih Lokasi"),
),
),
);
}
}
```