

**TUGAS UNGUIDED & GUIDED
PEMROGRAMAN PERANGKAT BERGERAK**

MODUL XIII

MAPS AND PLACES



Disusun Oleh :

Nama: Arwin Nabiel Arroffif

NIM: 2211104057

Kelas: 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

MAPS AND PLACES

A. GOOGLE MAPS API

Google Maps API merupakan salah satu layanan dari Google untuk membantu developer menciptakan aplikasi yang menggunakan fitur peta atau maps. Pada Google Maps API kita dapat memasang marker, menggunakan fitur route, mencari tempat, dan masih banyak lagi.

Cara implementasi Google API pada flutter dapat dilakukan dengan menggunakan packages Google Maps. Tahapan dalam menambahkan Google Maps API dapat mengikuti langkah-langkah berikut :

1. Dapatkan API key melalui link berikut <https://cloud.google.com/maps-platform/>
2. Selanjutnya, enable Google Map SDK di tiap platform yang akan menggunakan Google Maps.
 - a. Pergi ke <https://console.cloud.google.com/> (Google Developers Console)
 - b. Pilih project yang ingin menggunakan Google Maps
 - c. Pilih pada navigation menu, lalu pilih “Google Maps”
 - d. Pilih “APIs” di bawah menu Google Maps
 - e. Untuk mengaktifkan Google Maps di Android, pilih “Maps SDK for Android” pada section “Additional APIs”, lalu pilih “ENABLE”
 - f. Untuk mengaktifkan Google Maps di iOS, pilih “Maps SDK for iOS” pada section “Additional APIs”, lalu pilih “ENABLE”
 - g. Pastikan bahwa APIs telah aktif pada section “Enabled APIs”
 - h. Untuk lebih detail bisa cek di <https://developers.google.com/maps/gmp-get-started>
3. Android
 - a. Set minSdkVersion di android/app/build.gradle:

```
android {  
    defaultConfig {  
        minSdkVersion 20  
    }  
}
```

Ini dimaksudkan agar aplikasinya support atau tersedia pada Android SDK 20 atau lebih tinggi.

- b. Tambahkan API key pada manifest aplikasi
android/app/src/main/AndroidManifest.xml

```

        android:name="android.permission.ACCESS_COARSE_LOCATION" />

        <application
            android:label="google_maps_flutter"
            android:icon="@mipmap/ic_launcher">

            <!-- TODO: Add your API key here -->
            <meta-data android:name="com.google.android.geo.API_KEY"
                android:value="YOUR KEY HERE"/>

            <activity>...</activity>
        </application>
    </manifest>

```

4. Hybrid Composition

Untuk menggunakan Hybrid Composition yang digunakan untuk merender GoogleMap pada widget Android, terapkan `AndroidGoogleMapsFlutter.userAndroidViewSurface` ke `true`.

```

if (defaultTargetPlatform == TargetPlatform.android) {
    AndroidGoogleMapsFlutter.useAndroidViewSurface = true;
}

```

5. IOS

Plugin ini membutuhkan iOS 9.0 atau lebih tinggi. Untuk menerapkan, tambahkan API key pada application delegate `ios/Runner/AppDelegate.m`:

```

#include "AppDelegate.h"
#include "GeneratedPluginRegistrant.h"
#import "GoogleMaps/GoogleMaps.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
    didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    [GMSServices provideAPIKey:@"YOUR KEY HERE"];
    [GeneratedPluginRegistrant registerWithRegistry:self];
    return [super application:application
        didFinishLaunchingWithOptions:launchOptions];
}

@end

```

Atau dalam penulisan bahasa Swift, tambahkan API key pada application delegate `ios/Runner/AppDelegate.swift`:

```

import UIKit
import Flutter
import GoogleMaps

@UIApplicationMain
@objc class AppDelegate: FlutterAppDelegate {
  override func application(
    _ application:
    UIApplication,
    didFinishLaunchingWithOptions launchOptions:
    [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
    GMServices.provideAPIKey("YOUR KEY HERE")
    GeneratedPluginRegistrant.register(with: self)
    return
    super.application(application, didFinishLaunchingWithOptions:
    launchOptions)
  }
}

```

Langkah diatas untuk menambahkan Google Maps API ke aplikasi.

B. Menambah Packages Google Maps

Setelah mengikuti langkah diatas, sekarang adalah langkah-langkah menambahkan Google Maps ke layar aplikasi Flutter:

1. Pergi ke <https://www.pub.dev> , lalu cari packages Google Maps. Nama packagesnya adalah `google_maps_flutter`.
2. Cari versi yang paling terbaru lalu tambahkan pada file `pubspec.yaml`
3. Selanjutnya, import packages ke dalam file Dart

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

4. Lalu, tambahkan widget `GoogleMap` ke file Dart

```

GoogleMap(
  initialCameraPosition: _kInitialPosition,
),

```

`GoogleMap` diberi `_kInitialPosition`, yang dimana untuk menyimpan lokasi default saat aplikasi dijalankan atau dimuat.

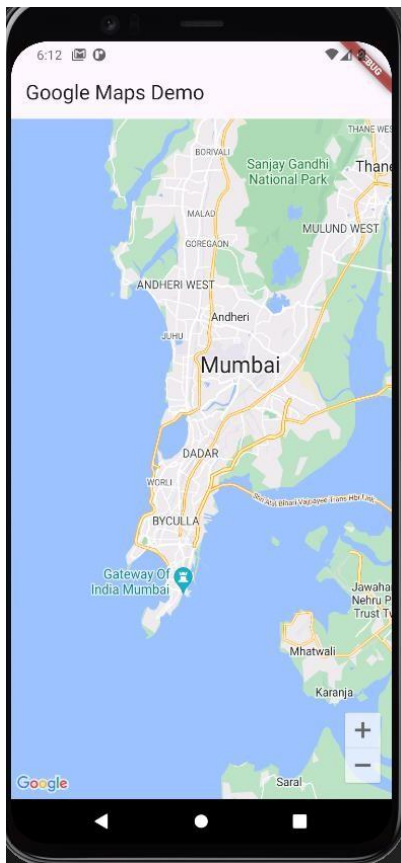
5. Buat fungsi `_kMapCenter` dan `_kInitialPosition` dengan kode sebagai berikut:

```
static final LatLng _kMapCenter =  
    LatLng(19.018255973653343, 72.84793849278007);  
  
static final CameraPosition _kInitialPosition =  
    CameraPosition(target: _kMapCenter, zoom: 11.0, tilt: 0, bearing: 0);
```

6. Berikut adalah tampilan kode yang lengkap

```
import 'package:flutter/material.dart';  
import 'package:google_maps_flutter/google_maps_flutter.dart';  
  
class MapsScreen extends StatefulWidget {  
    @override  
    _MapsScreenState createState() => _MapsScreenState();  
}  
  
class _MapsScreenState extends State<MapsScreen> {  
    static final LatLng _kMapCenter =  
        LatLng(19.018255973653343, 72.84793849278007);  
    static final CameraPosition _kInitialPosition =  
        CameraPosition(  
            target: _kMapCenter, zoom: 11.0,  
        );  
  
    @override  
    Widget build(BuildContext context)  
    {  
        return Scaffold(  
            appBar: AppBar(  
                title: Text('Google Maps Demo'),  
            ),  
            body: GoogleMap(  
                initialCameraPosition: _kInitialPosition,  
                myLocationEnabled: true,  
            ),  
        );  
    }  
}
```

Berikut tampilannya



C. Menambahkan Akses di Manifest

Secara default, map akan menampilkan lokasi yang sudah kita definisikan pada `initialCameraPosition` yang ada pada parameter widget. Jika pengguna ingin menampilkan lokasi mereka, ubah pengaturan `myLocationEnabled` menjadi `true`. Berikut barisan kode untuk menampilkan lokasi kita saat ini:

```
GoogleMap(  
    initialCameraPosition: _kInitialPosition, onMapCreated: onMapCreated,  
    myLocationEnabled: true,  
);
```

D. Menambahkan Marker pada Google Maps

Marker merupakan cara untuk menunjukkan lokasi tertentu. Untuk membuat marker pada map, berikut adalah barisan kodenya:

```
Set<Marker> _createMarker()  
{  
    return {  
        Marker(  
            markerId: MarkerId("marker_1"),  
            position: _kMapCenter,  
        ),  
    };  
}
```

```
        infoWindow: InfoWindow(title: 'Marker  
1'),          rotation: 90),      Marker(  
        markerId: MarkerId("marker_2"),  
        position: LatLng(-6.9733165, 107.6281415, 17),  
    ),  
    };  
}
```

E. Place Picker

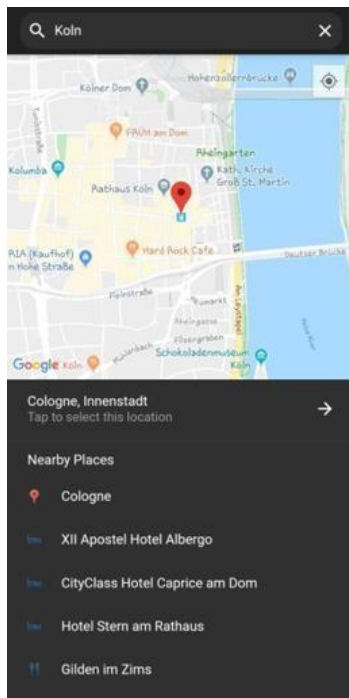
Place picker merupakan plugin untuk memberi informasi terkait lokasi yang sedang ditunjuk oleh map. Untuk menggunakan place picker, terlebih dahulu kita tambahkan API Google Maps dan ubah beberapa pengaturan dengan mengikuti langkah-langkah sebelum sub bab ini.

Jika sudah mengubah pengaturan yang diminta, terlebih dahulu kita tambahkan package ke dalam pubspec.yaml seperti ini:

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

Setelah itu, kita buat method seperti yang akan di bahas pada sub bab ini, lalu gunakan onTap di Button atau InkWell untuk memanggil method tersebut. Berikut barisan kodenya:

```
void showPlacePicker() async {  
    LocationResult result = await Navigator.of(context).push(MaterialPageRoute(  
builder: (context) => PlacePicker(  
        "YOUR API KEY",  
        displayLocation: customLocation,  
    )));  
  
    // Handle the result in your way  
    print(result);  
}
```

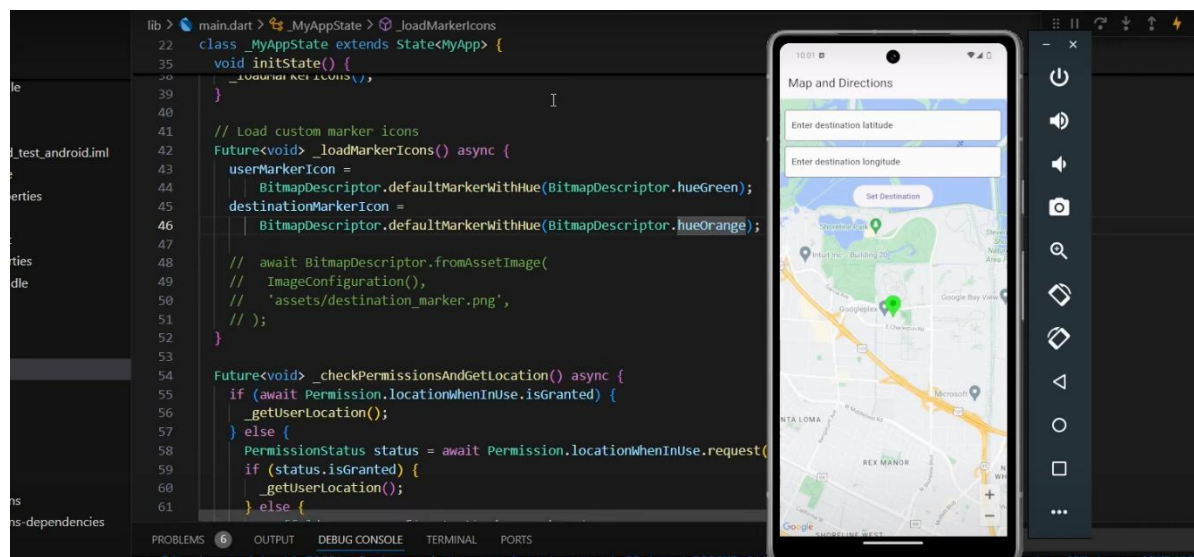


Tugas Mandiri (Unguided)

Dari tugas guided yang telah dikerjakan, lanjutkan hingga ke bagian place picker untuk memberikan informasi mengenai lokasi yang ditunjuk di peta.

Note: Jangan lupa sertakan source code, screenshot output, dan deskripsi program.

Kreatifitas menjadi nilai tambah.



Deskripsi program

Program ini adalah aplikasi Flutter sederhana yang menggunakan Google Maps untuk menampilkan peta dan mencari lokasi berdasarkan nama tempat yang diinputkan oleh pengguna. Aplikasi ini terdiri dari dua file utama, yaitu `main.dart` untuk pengaturan antarmuka utama aplikasi dan `map_screen.dart` untuk logika peta dan pencarian lokasi.

Aplikasi bekerja dengan menampilkan peta awal di lokasi default (Köln, Jerman). Pengguna dapat mengetikkan nama lokasi di kolom pencarian, dan aplikasi akan mencari koordinat lokasi tersebut menggunakan plugin geocoding. Setelah ditemukan, peta akan berpindah secara otomatis ke lokasi yang diinginkan dengan efek animasi. Program memanfaatkan integrasi API Google Maps untuk menampilkan peta, dan `ChangeNotifier` untuk memperbarui UI saat pencarian dilakukan. Aplikasi ini dirancang responsif dan memberikan pengalaman pengguna yang interaktif dan intuitif.

Untuk kodingan ada di folder `unguided`.