

GUIDED & UNGUIDED
PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK
MODUL XII
MAPS PLACES FIREBASE NOTIFICATION



Disusun Oleh :
Maria Nathasya Desfera Pangestu / 2211104008
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

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  
    }  
}
```

Hal ini agar aplikasinya support dengan 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.userAndroidViewSurface = 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 {  
        GMSServices.provideAPIKey("YOUR KEY HERE")  
        GeneratedPluginRegistrant.register(with: self)  
        return super.application(application, didFinishLaunchingWithOptions:  
        launchOptions)  
    }  
}
```

Langkah tersebut untuk menambahkan Google Maps API ke aplikasi.

B. Menambahkan 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. 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,  
      ),  
    );  
  }  
}
```

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),  
    ),  
  ];  
}
```

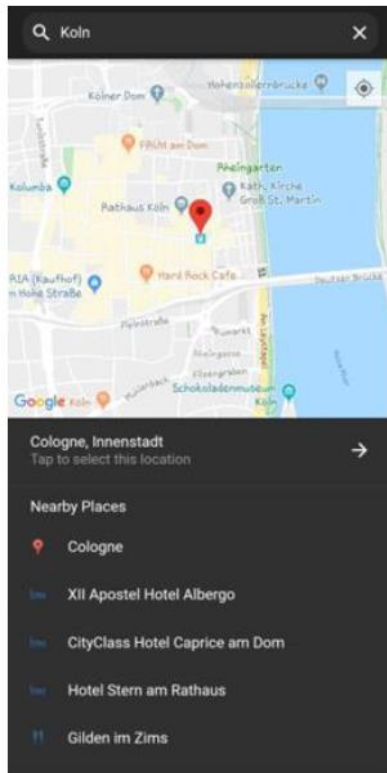
E. Place Picker

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);  
}
```



GUIDED

1. Source code main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:guided/myhomepage.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'Modul 12',
15       theme: ThemeData(
16         colorScheme: ColorScheme.fromSeed(seedColor: const Color.fromARGB(255, 222, 135, 240)),
17         useMaterial3: true,
18       ),
19       home: const MyHomePage(),
20     );
21   }
22 }
```

Source code myhomepage.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3
4 class MyHomePage extends StatefulWidget {
5   const MyHomePage({super.key});
6
7   @override
8   State<MyHomePage> createState() => _MyHomePageState();
9 }
10
11 class _MyHomePageState extends State<MyHomePage> {
12   static final LatLng _kMapCenter =
13     const LatLng(19.018255973653343, 72.84793849278007);
14   static final
15     CameraPosition _kInitialPosition = CameraPosition(target: _kMapCenter, zoom: 11.0, tilt: 0, bearing: 0);
16
17   @override
18   Widget build(BuildContext context) {
19     return Scaffold(
20       appBar: AppBar(
21         title: const Text('Google Maps Demo'),
22         centerTitle: true,
23         backgroundColor: const Color.fromARGB(255, 222, 135, 240),
24       ),
25       body: GoogleMap(initialCameraPosition: _kInitialPosition, myLocationEnabled: true,
26         ), // GoogleMap
27     );
28   }
29 }
```

2. Screenshot tampilan output maps



3. Penjelasan

Program tersebut menampilkan Google Maps dengan memakai `google_maps_flutter`. Dalam kelas `_MyHomePageState` lokasi awal peta ditentukan dengan koordinat yang diatur di variabel `_kMapCenter` dan dikonfigurasi menggunakan `CameraPosition` yang berisikan target untuk lokasi pusat, zoom untuk memperbesar peta, tilt untuk sudut kemiringan kamera, dan bearing untuk orientasi arah. Peta ditampilkan melalui widget `GoogleMap`, dengan lokasi pengguna diaktifkan melalui properti `myLocationEnabled`. Tampilan antarmuka mencakup `AppBar` berjudul "Google Maps Demo" dan peta di bagian body.


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. Kreativitas menjadi nilai tambah.

Jawab

1. Source code main.dart

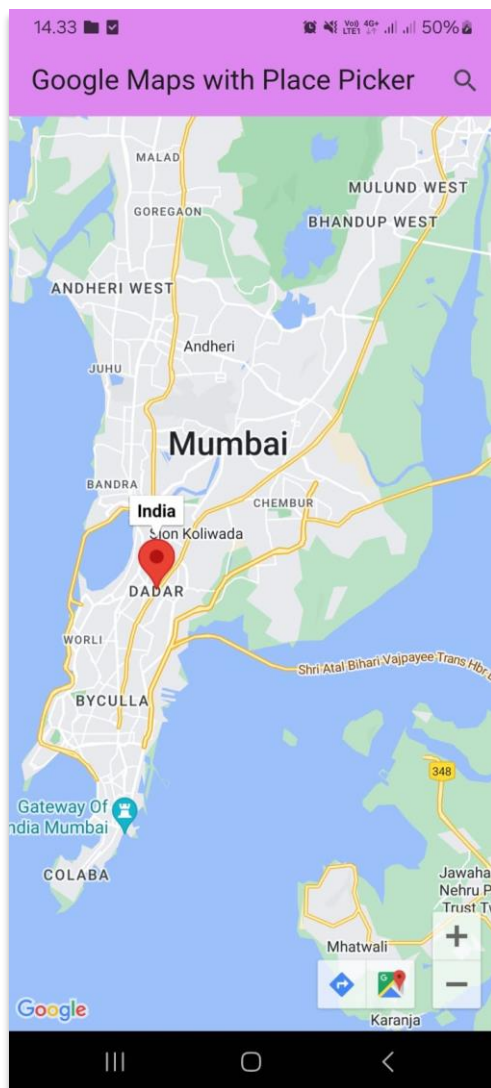


```
1  import 'package:flutter/material.dart';
2  import 'package:unguided/homepage.dart';
3
4  void main() {
5    runApp(const MyApp());
6  }
7
8  class MyApp extends StatelessWidget {
9    const MyApp({super.key});
10
11    @override
12    Widget build(BuildContext context) {
13      return MaterialApp(
14        debugShowCheckedModeBanner: false,
15        title: 'Modul 12',
16        theme: ThemeData(
17          useMaterial3: true,
18        ),
19        home: const MyHomePage(),
20      );
21    }
22  }
23
```

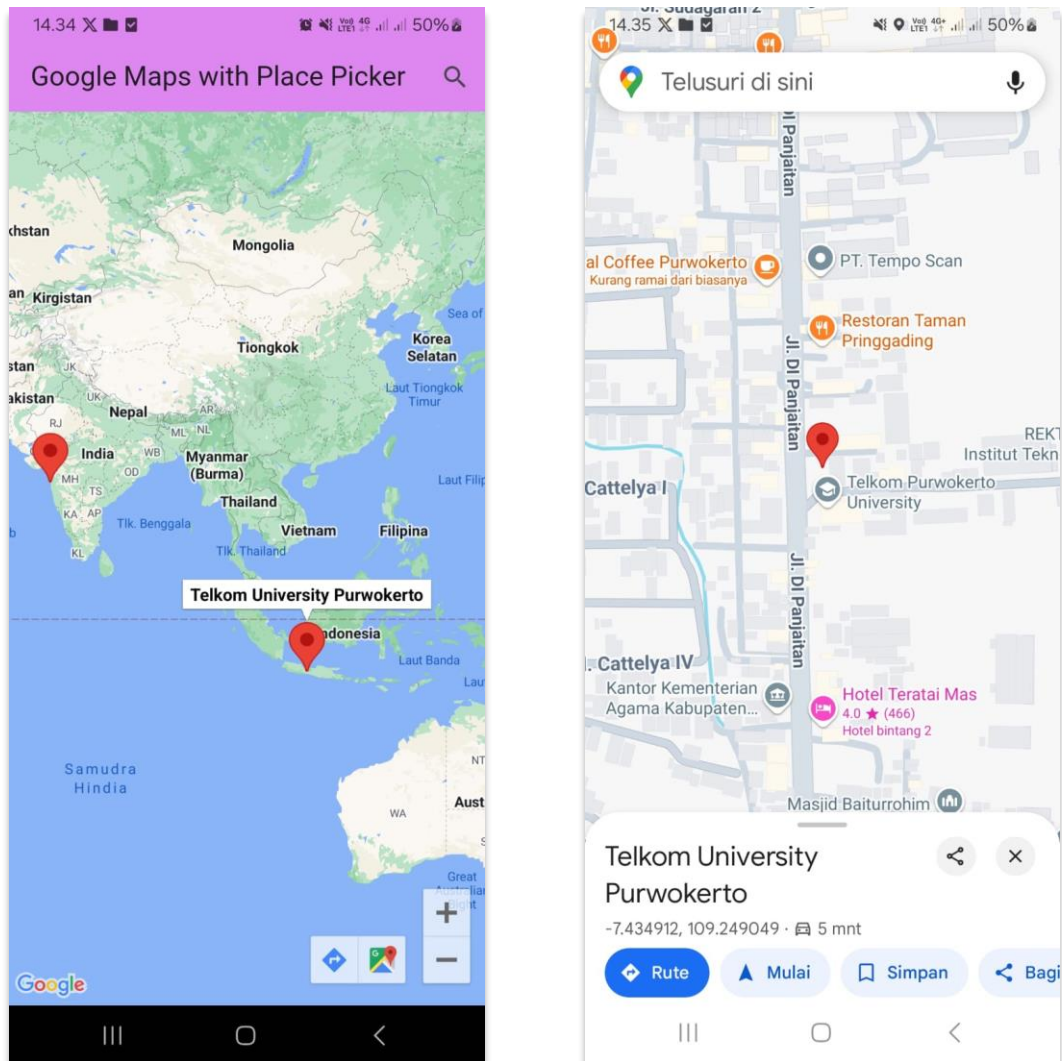
2. Source code homepage

```
1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3 import 'package:place_picker_google/place_picker_google.dart';
4
5 class MyHomePage extends StatefulWidget {
6   const MyHomePage({super.key});
7
8   @override
9   State<MyHomePage> createState() => _MyHomePageState();
10 }
11
12 class _MyHomePageState extends State<MyHomePage> {
13   static final LatLng _kMapCenter = LatLng(19.018255973653343,
14     72.84793849278007); // Koordinat yang digunakan (contohnya lokasi di Mumbai)
15   static final CameraPosition _kInitialPosition =
16     CameraPosition(target: _kMapCenter, zoom: 11.0, tilt: 0, bearing: 0);
17
18   // API Key untuk Google Maps
19   final String apiKey = 'YOUR_GOOGLE_MAPS_API_KEY';
20
21   // Membuat marker
22   Set<Marker> _createMarker() {
23     return {
24       Marker(
25         markerId: MarkerId("India"),
26         position: _kMapCenter,
27         infoWindow: InfoWindow(title: 'India'),
28         rotation: 0,
29       ),
30       Marker(
31         markerId: MarkerId("Telkom University Purwokerto"),
32         position: LatLng(-7.434912017809685,
33           109.24904967880413), // Contoh koordinat lain
34         infoWindow: InfoWindow(title: 'Telkom University Purwokerto')),
35     };
36   }
37
38   // Fungsi untuk menampilkan Place Picker
39   void _showPlacePicker() async {
40     // Menampilkan Place Picker dan menangkap hasilnya
41     LocationResult? result = await Navigator.of(context).push(MaterialPageRoute(
42       builder: (context) => PlacePicker(
43         apiKey: apiKey, // API key untuk Google Maps
44         onPlacePicked: (LocationResult result) {
45           // Menampilkan alamat tempat yang dipilih
46           debugPrint("Place picked: ${result.formattedAddress}");
47         },
48         initialLocation: _kMapCenter, // Menampilkan lokasi awal
49         searchInputConfig: const SearchInputConfig(
50           padding: EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),
51           autofocus: false,
52           textDirection: TextDirection.ltr,
53         ),
54         searchInputDecorationConfig: const SearchInputDecorationConfig(
55           hintText: "Search for a building, street or ...",
56         ),
57       ),
58     ));
59
60     // Pengecekan jika result tidak ada
61     if (result != null) {
62       debugPrint("Place picked: ${result.formattedAddress}");
63     } else {
64       debugPrint("No place selected.");
65     }
66   }
67
68   @override
69   Widget build(BuildContext context) {
70     return Scaffold(
71       appBar: AppBar(
72         title: const Text('Google Maps with Place Picker'),
73         centerTitle: true,
74         backgroundColor: const Color.fromARGB(255, 222, 135, 240),
75         actions: [
76           IconButton(
77             icon: const Icon(Icons.search),
78             onPressed:
79               _showPlacePicker, // Menampilkan Place Picker saat ikon pencarian ditekan
80           ),
81         ],
82       ),
83       body: GoogleMap(
84         initialCameraPosition: _kInitialPosition,
85         myLocationEnabled: true,
86         markers: _createMarker(), // Menambahkan marker pada peta
87       ),
88     );
89   }
90 }
```

3. Hasil output tampilan awal ketika di zoom in



Hasil output tampilan ketika di zoom out dan saat markernya di klik akan menampilkan detail Lokasi



4. Penjelasan

Kode Flutter ini menggunakan `google_maps_flutter` dan `place_picker_google` untuk mengintegrasikan Google Maps dan fitur Place Picker. Menampilkan peta dengan posisi awal di Mumbai dan dua marker yaitu di India dan Universitas Telkom Purwokerto. Pengguna dapat memilih tempat dengan fitur pencarian lokasi yang akan ditampilkan di konsol. Komponen antarmuka Flutter terdiri dari `flutter/material.dart`, `google_maps_flutter` untuk menampilkan peta Google Maps, dan `place_picker_google` untuk menyediakan fitur pencarian lokasi. Kemudian untuk mengubah peta dan data lokasi selama aplikasi berjalan menggunakan widget `stateful`. Lalu `kMapCenter` untuk menentukan koordinat pusat awal peta, misalnya lokasi di Mumbai, dan `kInitialPosition` menentukan posisi awal kamera dengan pengaturan zoom, tilt, dan bearing. Marker dibuat pada peta dengan menggunakan fungsi `_createMarker`. Dengan fitur `myLocationEnabled`, peta mendukung lokasi pengguna.