

PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK

MODUL XII

MAPS AND PLACES



Disusun Oleh:

Nadia Putri Rahmaniar / 2211104012

S1 SE-06-01

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

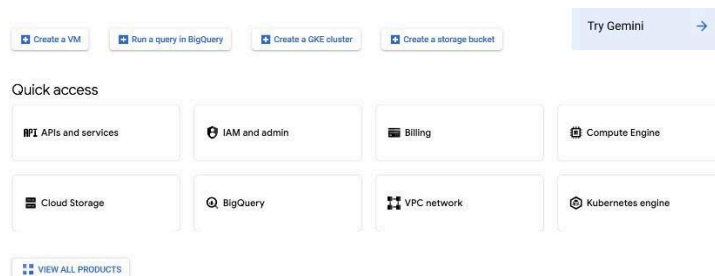
PURWOKERTO 2024

GUIDED

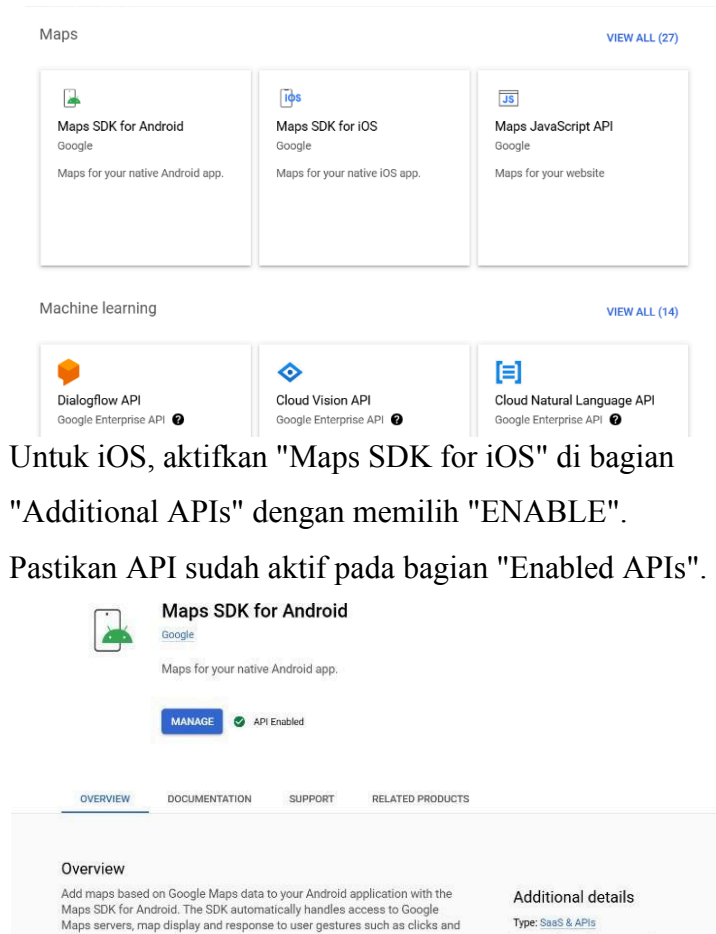
A. Google Maps API

Google Maps API adalah layanan dari Google yang dirancang untuk membantu pengembang membuat aplikasi dengan fitur peta atau maps. Melalui Google Maps API, Anda dapat menambahkan marker, menggunakan fitur rute, mencari lokasi, dan masih banyak lagi. Implementasi Google Maps API di Flutter bisa dilakukan dengan menggunakan package Google Maps. Berikut langkah-langkah menambahkan Google Maps API:

1. Dapatkan API key melalui tautan berikut:
<https://cloud.google.com/maps-platform>.
2. Aktifkan Google Maps SDK untuk setiap platform yang akan digunakan:
 - Akses [Google Developers Console](#).
 - Pilih project yang ingin menggunakan Google Maps.
 - Buka menu navigasi, lalu pilih "Google Maps".
 - Klik "APIs" di bawah menu Google Maps.



- Untuk Android, aktifkan "Maps SDK for Android" di bagian "Additional APIs" dengan memilih "ENABLE".



- Untuk iOS, aktifkan "Maps SDK for iOS" di bagian "Additional APIs" dengan memilih "ENABLE".
- Pastikan API sudah aktif pada bagian "Enabled APIs".

Untuk informasi lebih detail, kunjungi [Panduan Google Maps API](#).

3. Android

- Set minSdkVersion di android/app/build.gradle:

```
defaultConfig {
    // TODO: Specify your own unique Application ID (https://developer.android.com
    applicationId = "com.example.pertemuan_12"
    // You can update the following values to match your application needs.
    // For more information, see: https://flutter.dev/to/review-gradle-config.
    minSdkVersion 20
    targetSdk = flutter.targetSdkVersion
    versionCode = flutter.versionCode
    versionName = flutter.versionName
}
```

- Tambahkan API key pada manifest aplikasi

android/app/src/main/AndroidManifest.xml

```
<meta-data android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyBsO5wGvORisdNM4M3rChwVc_cQWc3V_tA"/>
<activity
```

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`

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  # The following adds the Cupertino Icons font to your application.  
  # Use with the CupertinoIcons class for iOS style icons.  
  cupertino_icons: ^1.0.8  
  google_maps_flutter: ^2.10.0
```

3. Selanjutnya, import packages ke dalam file Dart
4. Lalu, tambahkan widget `GoogleMap` ke file Dart. `GoogleMap` diberi `_kInitialPosition`, yang dimana untuk menyimpan lokasi default saat aplikasi dijalankan atau dimuat.

```
body: GoogleMap(  
  initialCameraPosition: _kInitialPosition,  
  myLocationEnabled: true,  
), // GoogleMap
```

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

File homepage.dart

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

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

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  static final LatLng _kMapCenter =
    LatLng(19.018255973653343, 72.84793849278007);
  static final CameraPosition _kInitialPosition =
    CameraPosition(target: _kMapCenter, zoom: 11.0, tilt: 0, bearing: 0);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Google Mapas in FLutter Demo'),
        centerTitle: true,
        backgroundColor: const Color.fromARGB(255, 255, 255, 255),
      ),
      body: GoogleMap(
        initialCameraPosition: _kInitialPosition,
        //onMapCreated: onMapCreated,
        myLocationEnabled: true,
      ),
    );
  }
}
```

File main.dart

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

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

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

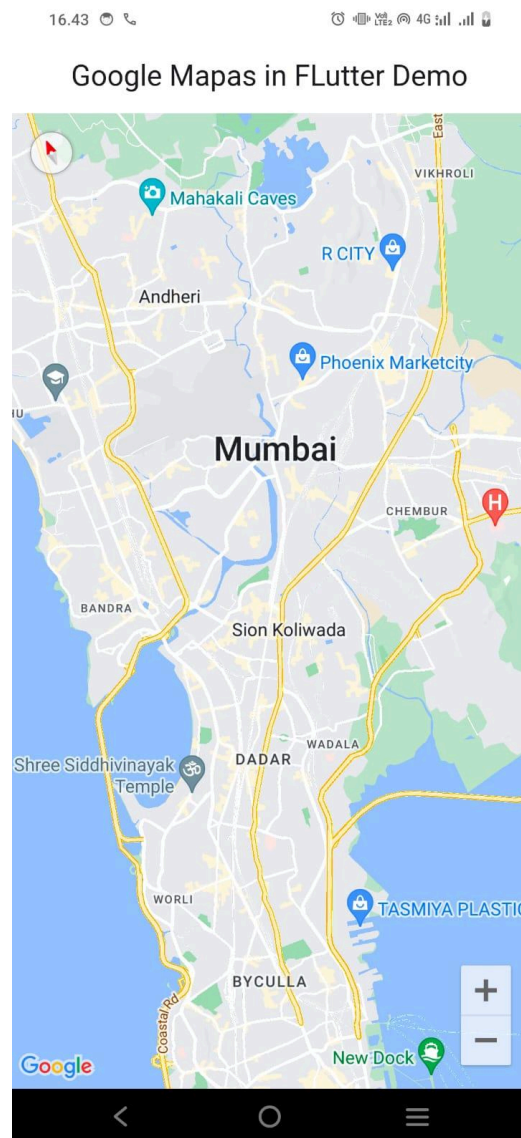
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Modul 12',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
    );
  }
}
```

```

    ),
    home: MyHomePage(),
  );
}
}

```

Output Program



C. Menambahkan Akses ke Manifest

```

body: GoogleMap(
  initialCameraPosition: _kInitialPosition,
  myLocationEnabled: true,

```

UNGUIDED

1. Menambahkan Marker pada Google Maps

Marker digunakan untuk menandai lokasi tertentu pada peta. Berikut adalah kode lengkap untuk membuat marker pada map, sehingga file homepage.dart menjadi seperti berikut.

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

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

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  static final LatLng _kMapCenter =
  LatLng(19.018255973653343,
    72.84793849278007); // Koordinat yang digunakan
  (contohnya lokasi di Mumbai)

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

  // API Key untuk Google Maps
  final String apiKey = 'YOUR_GOOGLE_MAPS_API_KEY';

  // Membuat marker
  Set<Marker> _createMarker() {
    return {
```

```

Marker(
    markerId: MarkerId("marker_1"),
    position: _kMapCenter,
    infoWindow: InfoWindow(title: 'Marker 1'),
    rotation: 0,
),
Marker(
    markerId: MarkerId("marker_2"),
    position: LatLng(-6.9733165, 107.6281415), //
Koordinat contoh lainnya
),
};
}

// Fungsi untuk menampilkan Place Picker
void _showPlacePicker() async {
    // Menampilkan Place Picker dan menangkap hasilnya
    LocationResult? result = await
Navigator.of(context).push(MaterialPageRoute(
    builder: (context) => PlacePicker(
        apiKey: apiKey, // API key untuk Google Maps
        onPlacePicked: (LocationResult result) {
            // Menampilkan alamat tempat yang dipilih
            debugPrint("Place picked:
${result.formattedAddress}");
        },
        initialLocation: _kMapCenter, // Menampilkan
lokasi awal
        searchInputConfig: const SearchInputConfig(
            padding: EdgeInsets.symmetric(horizontal: 16.0,
vertical: 8.0),
            autofocus: false,
            textDirection: TextDirection.ltr,

```



```

        ),
        searchInputDecorationConfig: const
SearchInputDecorationConfig(
        hintText: "Search for a building, street or
...",
        ),
    ),
    ));

    // Pengecekan jika result tidak ada
    if (result != null) {
        debugPrint("Place picked:
${result.formattedAddress}");
    } else {
        debugPrint("No place selected.");
    }
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text('Google Maps with Place Picker
Demo'),
            centerTitle: true,
            backgroundColor: const Color.fromARGB(255, 255,
255, 255),
            actions: [
                IconButton(
                    icon: const Icon(Icons.search),
                    onPressed:
                        _showPlacePicker, // Menampilkan Place
Picker saat ikon pencarian ditekan

```

```
        ),  
        ],  
    ),  
    body: GoogleMap(  
        initialCameraPosition: _kInitialPosition,  
        myLocationEnabled: true,  
        markers: _createMarker(), // Menambahkan marker  
pada peta  
    ),  
);  
}  
}
```

Hasil Output Program

09.07

4G

Google Maps with Place Picke...



Deskripsi Program

1. Fungsi `_createMarker()`:

- Digunakan untuk membuat dan mengembalikan kumpulan Marker yang akan ditampilkan di peta.
- Marker pertama ada di `_kMapCenter` dengan info window berjudul "Marker 1".
- Marker kedua berada di koordinat lain, yaitu `LatLng(-6.9733165, 107.6281415)`.

2. Properti markers pada `GoogleMap`: Menggunakan hasil dari fungsi `_createMarker()` untuk menambahkan marker ke peta.

3. Pengaturan Awal Peta: Peta diatur untuk memiliki posisi awal berdasarkan `_kInitialPosition`.

Setelah kode dijalankan, peta akan menampilkan dua marker di lokasi yang telah ditentukan.

2. Place Picker

Place picker adalah plugin yang digunakan untuk memberikan informasi tentang lokasi yang ditunjuk pada peta. Sebelum menggunakannya, pastikan Anda telah menambahkan API Google Maps dan melakukan pengaturan yang sesuai dengan langkah-langkah yang dijelaskan pada subbab sebelumnya. Setelah pengaturan selesai, langkah berikutnya adalah menambahkan package place picker ke dalam file `pubspec.yaml`, seperti berikut:

```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.8
google_maps_flutter: ^2.10.0
place_picker_google: ^0.0.13
```

Source Code

```
// Fungsi untuk menampilkan Place Picker
void _showPlacePicker() async {
  // Menampilkan Place Picker dan menangkap hasilnya
  LocationResult? result = await
Navigator.of(context).push(MaterialPageRoute(
  builder: (context) => PlacePicker(
    apiKey: apiKey, // API key untuk Google Maps
    onPlacePicked: (LocationResult result) {
      // Menampilkan alamat tempat yang dipilih
      debugPrint("Place picked: ${result.formattedAddress}");
```

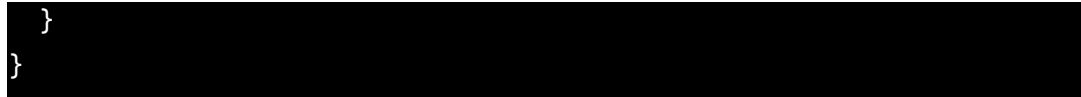
```

    },
    initialLocation: _kMapCenter, // Menampilkan lokasi awal
    searchInputConfig: const SearchInputConfig(
      padding: EdgeInsets.symmetric(horizontal: 16.0, vertical:
8.0),
      autofocus: false,
      textDirection: TextDirection.ltr,
    ),
    searchInputDecorationConfig: const
SearchInputDecorationConfig(
      hintText: "Search for a building, street or ...",
    ),
  ),
));

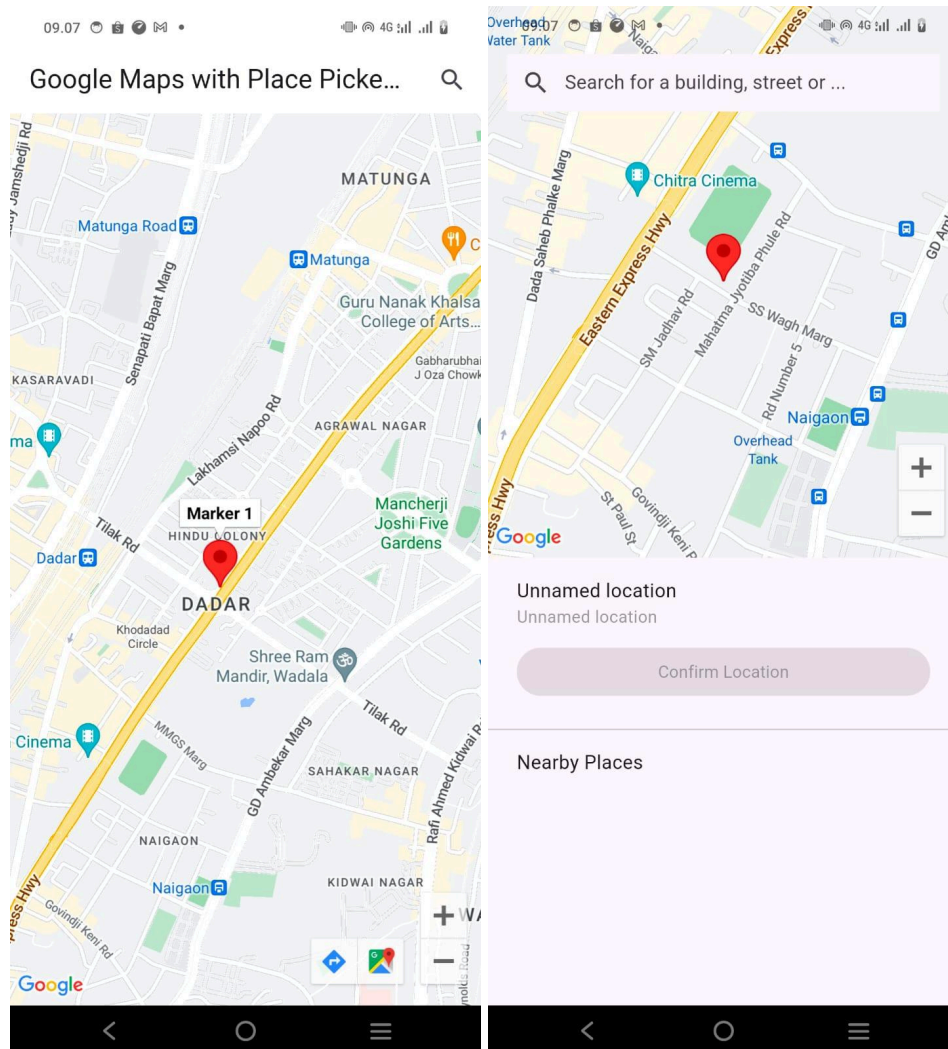
// Pengecekan jika result tidak ada
if (result != null) {
  debugPrint("Place picked: ${result.formattedAddress}");
} else {
  debugPrint("No place selected.");
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Google Maps with Place Picker Demo'),
      centerTitle: true,
      backgroundColor: const Color.fromARGB(255, 255, 255, 255),
      actions: [
        IconButton(
          icon: const Icon(Icons.search),
          onPressed:
            _showPlacePicker, // Menampilkan Place Picker saat
ikon pencarian ditekan
        ),
      ],
    ),
    body: GoogleMap(
      initialCameraPosition: _kInitialPosition,
      myLocationEnabled: true,
      markers: _createMarker(), // Menambahkan marker pada peta
    ),
  );
}

```



Hasil Program



Deskripsi Program

Aplikasi ini memanfaatkan widget GoogleMap dari paket `google_maps_flutter` untuk menampilkan peta dan sejumlah marker di lokasi tertentu. Selain itu, aplikasi ini menyertakan fitur pencarian lokasi menggunakan `PlacePicker` dari paket `place_picker_google`, yang memudahkan pengguna dalam memilih lokasi. Tombol pencarian di app bar digunakan untuk mengaktifkan fungsi `PlacePicker`. Berikut adalah penjelasan cara kerjanya:

Menampilkan Place Picker

Ketika pengguna menekan ikon pencarian (`IconButton` dengan `Icons.search`) di app bar, fungsi `_showPlacePicker()` dipanggil. Fungsi ini memanfaatkan `Navigator.push` untuk membuka halaman `PlacePicker`, yang memungkinkan pengguna mencari dan

memilih lokasi langsung dari peta.

Menerima Hasil dari Place Picker

PlacePicker memerlukan beberapa parameter, seperti apiKey untuk autentikasi ke Google Maps API dan initialLocation sebagai lokasi awal peta (biasanya diatur pada koordinat default aplikasi). Setelah pengguna memilih lokasi, fungsi onPlacePicked dipanggil, mengembalikan objek LocationResult yang berisi informasi detail tentang tempat tersebut, termasuk alamat terformat.

Menangani Hasil Pilihan

Ketika lokasi telah dipilih, alamat dari lokasi tersebut ditampilkan di konsol menggunakan debugPrint. Jika pengguna tidak memilih lokasi, maka pesan "No place selected" akan muncul di konsol.