PEMROGRAMAN PERANGKAT BERGERAK

MODUL X!!
MAPS

Disusun oleh:
Salman Alfa R
2211104056

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

Tujuan Praktikum
Mahasiswa memahami cara mengintegrasikan Google Maaps ke dalam project Flutter. Mahasiswa dapat mengimplementasikan Google Maps API untuk menampilkan lokasi

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 Delevopers 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 bersedia 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. Hubrid 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 {
    GMSServices.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.

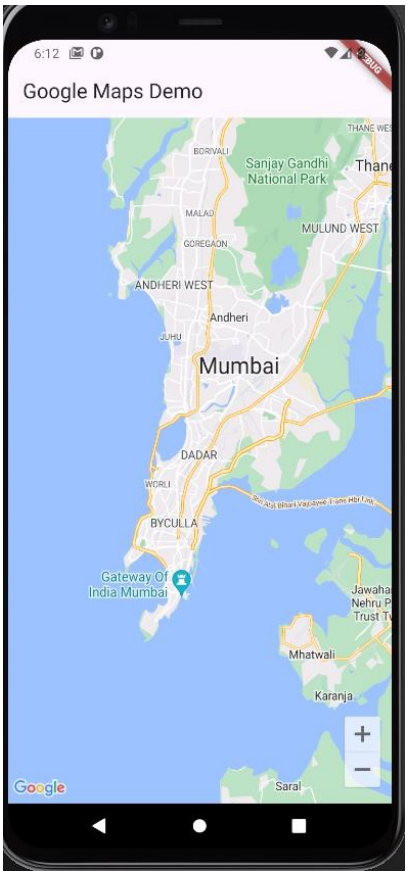
5. Buat fungsi _kMapCenter dan _kInitialPosition dengan kode sebagai berikut:

```
      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 myLocationEnable menjad 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),
    ),
  };
}
```

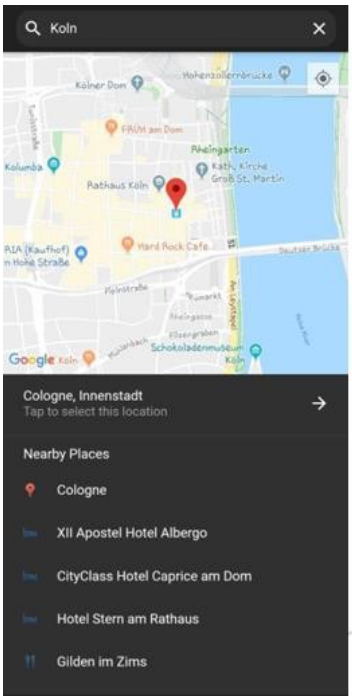
E. Place Picker

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:

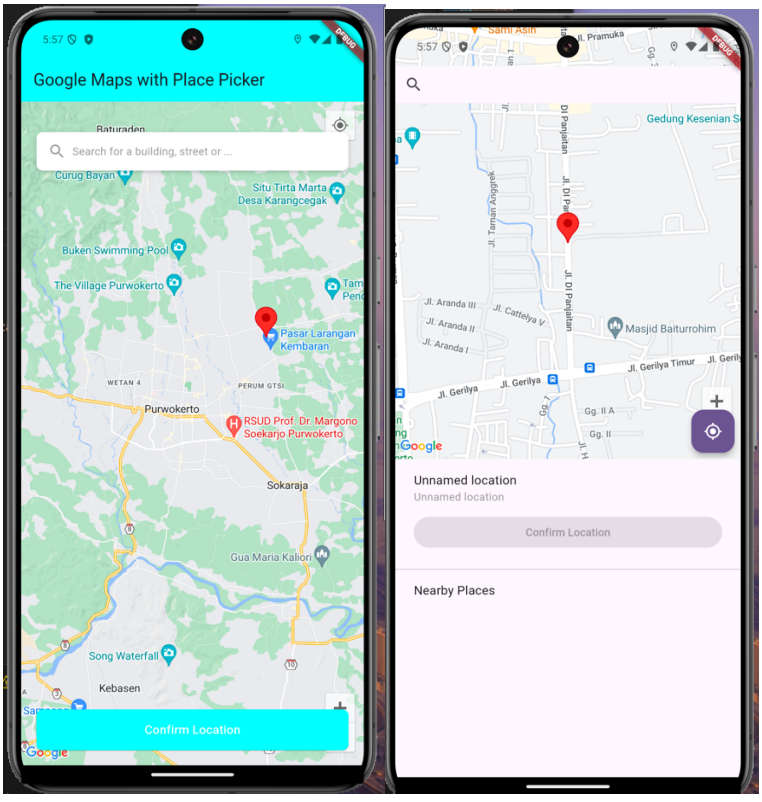
```
void showPlacePicker() async {
  LocationResult result = await Navigator.of(context).push(MaterialPageRoute(
      builder: (context) => PlacePicker(
            "YOUR API KEY",
            displayLocation: customLocation,
          )));

  // 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.

ScreenShoot:



SourceCode
Maps.dart

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

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

  @override
  _MapsScreenState createState() => _MapsScreenState();
}

class _MapsScreenState extends State<MapsScreen> {
  LatLng _kMapCenter = LatLng(-7.435268, 109.248834);
  CameraPosition _kInitialPosition = const CameraPosition(
    target: LatLng(-7.435268, 109.248834),
    zoom: 11.0,
  );

  LatLng? _selectedLocation; // Lokasi yang dipilih

  // Menampilkan Place Picker
  void showPlacePicker() async {
    LocationResult? result = await Navigator.of(context).push(
      MaterialPageRoute(
        builder: (context) => PlacePicker(
          apiKey: "AIzaSyC_qyRFY4Awl1M0-tUFGj_LV4NdADkEZB4", // Ganti dengan API Key
Anda
          initialLocation: _kMapCenter, // Lokasi awal Place Picker
          myLocationButtonEnabled: true,
          myLocationEnabled: true,
        ),
      ),
    );

    if (result != null) {
      setState(() {
        _kMapCenter = result.latLng!;
        _selectedLocation = result.latLng!;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Google Maps with Place Picker"),
        backgroundColor: Colors.cyanAccent,
      ),
```

```
    body: Stack(
      children: [
        GoogleMap(
          initialCameraPosition: _kInitialPosition,
          myLocationEnabled: true,
          markers: {
            Marker(
              markerId: const MarkerId("selected_location"),
              position: _selectedLocation ?? _kMapCenter,
              infoWindow: const InfoWindow(title: 'Selected Location'),
            ),
          },
          onTap: (LatLng location) {
            setState(() {
              _selectedLocation = location;
            });
          },
        ),
        // Search bar di bagian atas
        Positioned(
          top: 40,
          left: 20,
          right: 20,
          child: GestureDetector(
            onTap: showPlacePicker,
            child: Container(
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.circular(8),
                boxShadow: [
                  BoxShadow(
                    color: Colors.black.withOpacity(0.2),
                    blurRadius: 4,
                    offset: Offset(0, 2),
                  ),
                ],
              ),
              padding: const EdgeInsets.symmetric(horizontal: 15),
              height: 50,
              child: Row(
                children: const [
                  Icon(Icons.search, color: Colors.grey),
                  SizedBox(width: 8),
                  Expanded(
                    child: Text(
                      "Search for a building, street or ...",
                      style: TextStyle(color: Colors.grey),
                    ),
                  ),
                ],
              ),
            ),
          ),
```

```
          ),
        ),
        // Tombol konfirmasi lokasi di bagian bawah
        Positioned(
          bottom: 20,
          left: 20,
          right: 20,
          child: ElevatedButton(
            onPressed: _selectedLocation != null
                ? () {
                    if (kDebugMode) {
                      print("Confirmed location: $_selectedLocation");
                    }
                  }
                : null,
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.cyanAccent,
              padding: const EdgeInsets.symmetric(vertical: 15),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8),
              ),
            ),
            child: const Text(
              "Confirm Location",
              style: TextStyle(
                color: Colors.white,
                fontSize: 16,
              ),
            ),
          ),
        ),
      ],
    ),
  );
 }
}
```

Main.dart

```
import 'package:flutter/material.dart';
import 'package:praktikum_12/Maps.dart';


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

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

  // This widget is the root of your application.
  @override
```

```
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a purple toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can be
        // tested with just a hot reload.
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MapsScreen(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

  final String title;

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

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      // This call to setState tells the Flutter framework that something has
      // changed in this State, which causes it to rerun the build method below
```

```
    // so that the display can reflect the updated values. If we changed
    // _counter without calling setState(), then the build method would not be
    // called again, and so nothing would appear to happen.
    _counter++;
  });
}

@override
Widget build(BuildContext context) {
  // This method is rerun every time setState is called, for instance as done
  // by the _incrementCounter method above.
  //
  // The Flutter framework has been optimized to make rerunning build methods
  // fast, so that you can just rebuild anything that needs updating rather
  // than having to individually change instances of widgets.
  return Scaffold(
    appBar: AppBar(
      // TRY THIS: Try changing the color here to a specific color (to
      // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar
      // change color while the other colors stay the same.
      backgroundColor: Theme.of(context).colorScheme.inversePrimary,
      // Here we take the value from the MyHomePage object that was created by
      // the App.build method, and use it to set our appbar title.
      title: Text(widget.title),
    ),
    body: Center(
      // Center is a layout widget. It takes a single child and positions it
      // in the middle of the parent.
      child: Column(
        // Column is also a layout widget. It takes a list of children and
        // arranges them vertically. By default, it sizes itself to fit its
        // children horizontally, and tries to be as tall as its parent.
        //
        // Column has various properties to control how it sizes itself and
        // how it positions its children. Here we use mainAxisAlignment to
        // center the children vertically; the main axis here is the vertical
        // axis because Columns are vertical (the cross axis would be
        // horizontal).
        //
        // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
        // action in the IDE, or press "p" in the console), to see the
        // wireframe for each widget.
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          const Text(
            'You have pushed the button this many times:',
          ),
          Text(
            '$_counter',
            style: Theme.of(context).textTheme.headlineMedium,
          ),
        ],
```

```
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ), // This trailing comma makes auto-formatting nicer for build methods.
    );
  }
}
```

Deskripsi Program

Program ini adalah aplikasi pemetaan berbasis Flutter yang menggunakan Google Maps dan Place Picker. Aplikasi ini memungkinkan pengguna untuk menampilkan peta interaktif dan memilih lokasi tertentu. Peta ditampilkan menggunakan widget GoogleMap, dengan lokasi awal yang telah ditentukan. Pengguna dapat mengetuk peta untuk memilih lokasi secara manual atau menggunakan Place Picker melalui search bar untuk mencari lokasi tertentu. Place Picker memanfaatkan Google Places API untuk menampilkan detail lokasi, seperti nama atau alamat, tetapi jika detail tidak tersedia, lokasi akan ditampilkan sebagai "Unnamed location." Setelah lokasi dipilih, pengguna dapat mengonfirmasi pilihan mereka menggunakan tombol "Confirm Location," yang mencetak koordinat lokasi ke konsol. Program ini cocok untuk aplikasi yang membutuhkan fitur pemilihan lokasi, seperti aplikasi pengiriman barang atau navigasi. Namun, aplikasi ini bergantung pada konfigurasi API Key yang benar untuk menampilkan detail lokasi secara akurat.

Note: Jangan lupa sertakan source code, screenshoot output, dan deskripsi program. Kreatifitas menjadi nilai tambah.