

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

MODUL 12

MAPS



Disusun Oleh :

Muhammad Abdul Aziz / 2211104026

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

GUIDED

File main.dart :

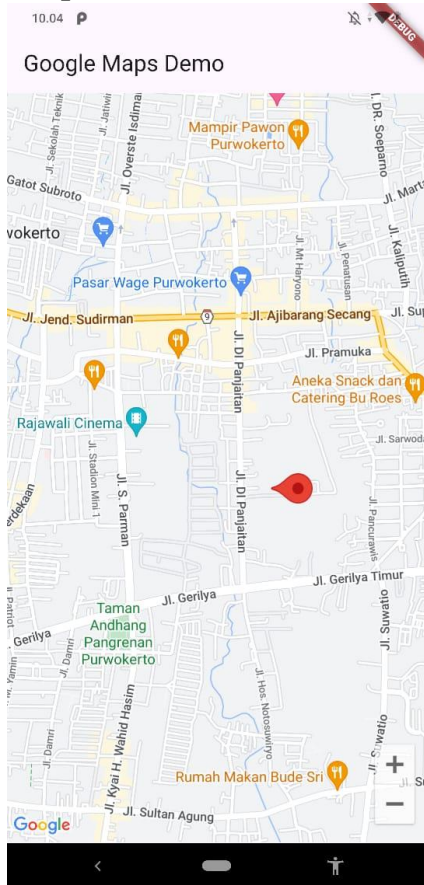


```
1  import 'package:flutter/material.dart';
2  import 'package:guided_pert12/homepage.dart';
3
4  void main() {
5    runApp(const MyApp());
6  }
7
8  class MyApp extends StatelessWidget {
9    const MyApp({super.key});
10
11    @override
12    Widget build(BuildContext context) {
13      return MaterialApp(
14        title: 'Google Maps Demo',
15        theme: ThemeData(
16          primarySwatch: Colors.blue,
17        ),
18        home: MapsScreen(),
19      );
20    }
21  }
22
```

File homepage.dart :

```
1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3
4 class MapsScreen extends StatefulWidget {
5   @override
6   _MapsScreenState createState() => _MapsScreenState();
7 }
8
9 class _MapsScreenState extends State<MapsScreen> {
10   static final LatLng _kMapCenter = LatLng(-7.4347652, 109.2500561);
11   static final CameraPosition _kInitialPosition = CameraPosition(
12     target: _kMapCenter,
13     zoom: 11.0,
14   );
15
16   late GoogleMapController _mapController;
17
18   @override
19   Widget build(BuildContext context) {
20     return Scaffold(
21       appBar: AppBar(
22         title: Text('Google Maps Demo'),
23       ),
24       body: GoogleMap(
25         initialCameraPosition: _kInitialPosition,
26         onMapCreated: (GoogleMapController controller) {
27           _mapController = controller;
28         },
29         myLocationEnabled: true, // Menampilkan Lokasi pengguna
30         markers: _createMarker(), // Menambahkan marker
31       ),
32     );
33   }
34
35   Set<Marker> _createMarker() {
36     return {
37       Marker(
38         markerId: MarkerId("marker_1"),
39         position: _kMapCenter,
40         infoWindow: InfoWindow(title: 'Marker 1'),
41         rotation: 90,
42       ),
43       Marker(
44         markerId: MarkerId("marker_2"),
45         position: LatLng(-6.9733165, 107.6281415),
46         infoWindow: InfoWindow(title: 'Marker 2'),
47       ),
48     };
49   }
50 }
```

Output :



Penjelasan :

Kode di atas adalah implementasi aplikasi Flutter sederhana yang menggunakan pustaka `google_maps_flutter` untuk menampilkan peta Google Maps. Aplikasi ini terdiri dari dua file utama: `main.dart` dan `MapsScreen`. Pada `main.dart`, aplikasi diinisialisasi dengan menggunakan widget `MyApp`, yang mengatur tema aplikasi dan menjadikan `MapsScreen` sebagai halaman awal.

Pada `MapsScreen`, peta ditampilkan dengan posisi awal yang telah ditentukan menggunakan `CameraPosition`, dengan pusat koordinat di lokasi tertentu di Indonesia. Fungsi `_createMarker` digunakan untuk membuat dua marker pada peta, masing-masing dilengkapi dengan informasi tambahan (`InfoWindow`). Fitur seperti `myLocationEnabled` juga diaktifkan, memungkinkan peta untuk menampilkan lokasi pengguna saat ini. Marker pertama memiliki rotasi khusus, sementara marker kedua ditempatkan di lokasi berbeda. Aplikasi ini juga menangani pengontrol peta melalui `GoogleMapController` untuk interaksi lebih lanjut.

UNGUIDED

File main.dart :



```
1  import 'package:flutter/material.dart';
2  import 'package:cobaaaaaaaaaaa/homepage.dart';
3
4  void main() {
5      runApp(const MainApp());
6  }
7
8  class MainApp extends StatelessWidget {
9      const MainApp({super.key});
10
11      @override
12      Widget build(BuildContext context) {
13          return const MaterialApp(home: HomePage());
14      }
15  }
16
```

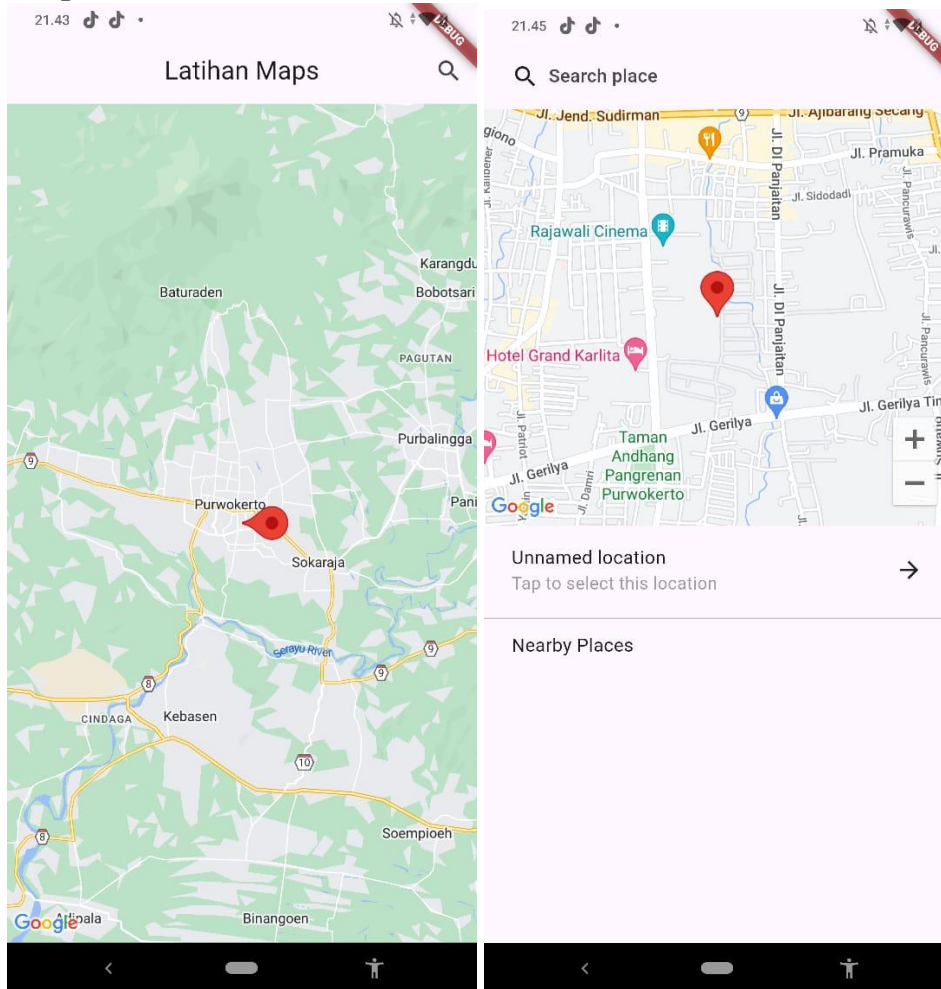
File homepage.dart :

```

1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3 import 'package:place_picker/place_picker.dart';
4
5 class HomePage extends StatefulWidget {
6   const HomePage({super.key});
7
8   @override
9   State<HomePage> createState() => _HomePageState();
10 }
11
12 class _HomePageState extends State<HomePage> {
13   static const LatLng _initialMapCenter = LatLng(-7.4352631,109.2465177);
14   static final CameraPosition _initialCameraPosition = CameraPosition(
15     target: _initialMapCenter,
16     zoom: 11.0,
17   );
18
19   late GoogleMapController _mapController;
20
21   // Fungsi untuk menangani pembentukan controller Google Maps
22   void _onMapCreated(GoogleMapController controller) {
23     _mapController = controller;
24   }
25
26   // Fungsi untuk membuat marker default
27   Set<Marker> _createMarker() {
28     return {
29       Marker(
30         markerId: const MarkerId("marker_1"),
31         position: _initialMapCenter,
32         infoWindow: const InfoWindow(title: 'kampus'),
33         rotation: 90,
34       ),
35     };
36   }
37
38   // Fungsi untuk membuka Place Picker
39   void _openPlacePicker() async {
40     try {
41       LocationResult? result = await Navigator.of(context).push(
42         MaterialPageRoute(
43           builder: (context) => PlacePicker(
44             "AIzaSyCj_C0Tc29ZQ4lRmCTTDGt7hmuXL3e3pTg", // Ganti dengan API Key Anda
45             displayLocation: _initialMapCenter, // Lokasi awal pada Place Picker
46           ),
47         ),
48       );
49
50       if (result != null && result.latlng != null) {
51         print("Place Picked: ${result.formattedAddress}");
52
53         // Memindahkan kamera ke Lokasi yang dipilih
54         _mapController.animateCamera(
55           CameraUpdate.newLatLng(result.latlng!),
56         );
57
58         // Menambahkan marker pada Lokasi yang dipilih
59         setState(() {
60           _createMarker().add(
61             Marker(
62               markerId: const MarkerId("picked_location"),
63               position: result.latlng!,
64               infoWindow: InfoWindow(
65                 title: "Picked Location",
66                 snippet: result.formattedAddress,
67               ),
68             ),
69           );
70         });
71       }
72     } catch (e) {
73       print("Error picking place: $e");
74     }
75   }
76
77   @override
78   Widget build(BuildContext context) {
79     return Scaffold(
80       appBar: AppBar(
81         title: const Text('Latihan Maps'),
82         centerTitle: true,
83         actions: [
84           IconButton(
85             onPressed: _openPlacePicker,
86             icon: const Icon(Icons.search),
87           ),
88         ],
89       ),
90       body: GoogleMap(
91         initialCameraPosition: _initialCameraPosition,
92         myLocationEnabled: true,
93         onMapCreated: _onMapCreated,
94         zoomControlsEnabled: false, // Menghilangkan kontrol zoom default
95         markers: _createMarker(),
96       ),
97     );
98   }
99
100   @override
101   void dispose() {
102     _mapController.dispose(); // Membersihkan controller saat widget dihancurkan
103     super.dispose();
104   }
105 }
106

```

Output :



Penjelasan :

Kode di atas adalah implementasi aplikasi Flutter yang memanfaatkan pustaka `google_maps_flutter` dan `place_picker` untuk menampilkan Google Maps dengan fitur interaktif. Aplikasi dimulai melalui `MainApp`, yang menampilkan halaman utama `HomePage`. Pada halaman ini, peta ditampilkan dengan posisi awal yang ditentukan menggunakan `CameraPosition`. Marker default ditambahkan pada lokasi kampus dengan informasi tambahan pada jendela info (`InfoWindow`).

Fitur utama dari aplikasi ini adalah `Place Picker`, yang memungkinkan pengguna memilih lokasi secara interaktif melalui tampilan peta khusus. Ketika pengguna memilih lokasi, kamera peta secara otomatis dipindahkan ke lokasi tersebut, dan marker baru ditambahkan pada posisi yang dipilih. Selain itu, aplikasi menyediakan tombol pencarian pada `AppBar` untuk mengakses fitur `Place Picker`. Fungsi seperti `myLocationEnabled` juga diaktifkan untuk menunjukkan lokasi pengguna saat ini, dan kontrol zoom default dinonaktifkan untuk antarmuka yang lebih bersih. `Controller` peta diatur untuk memastikan pengelolaan sumber daya yang baik dengan membersihkannya saat widget dihancurkan.