

Penjelasan :

1. HomePage Stateful Widget: Memiliki peta Google Map yang ditampilkan di layar.
2. _initialMapCenter: Lokasi awal peta di set pada koordinat tertentu (Banyumas).
3. GoogleMapController: Digunakan untuk mengontrol peta, seperti memindahkan kamera dan menambah marker.
4. _createMarker(): Fungsi untuk membuat marker awal pada lokasi tertentu dengan info window yang berjudul "Rumah ku istanaku".
5. _openPlacePicker(): Fungsi yang membuka Place Picker. Setelah pengguna memilih lokasi, kamera akan dipindahkan ke lokasi tersebut dan marker baru akan ditambahkan pada lokasi yang dipilih.
6. AppBar: Menampilkan tombol pencarian untuk membuka Place Picker.
7. dispose(): Membersihkan controller peta ketika widget dihancurkan.

Fungsi utama aplikasi ini adalah untuk menampilkan peta dan memungkinkan pengguna memilih lokasi melalui Place Picker dan menambahkan marker pada lokasi yang dipilih.

Sourcecode :

```

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 final LatLng _initialMapCenter = const LatLng(-6.901810, 109.655968);
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: 'Rumah ku istanaku'),
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             "AIzaSyCjC0Tc29ZQ4lRmCTTDGt7hmuXL3e3pTg", // 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       } catch (e) {
72         print("Error picking place: $e");
73       }
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 :

