LAPORAN GUIDED & UNGUIDED

PEMROGRAMAN PERANGKAT BERGERAK

MODUL XII

MAPS AND PLACES

Disusun Oleh :

Ahmad Junaidi / 2211104002

SE-06-01

Asisten Praktikum :

Ayu Susilowati

Noviana Rizki Anisa Putri

Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

GUIDED

Souce Code AndroidManifest.xml :

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.guided_pert12">

    <!-- Permissions should be outside the <application> tag -->
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"
/>

    <application
        android:label="guided_pert12"
        android:name="${applicationName}"
        android:icon="@mipmap/ic_launcher">

        <meta-data
            android:name="com.google.android.geo.API_KEY"

android:value="AIzaSyACZqm7h0T1hGj2ZW0e54UwjsW9cQwZ09M"
/>

        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:launchMode="singleTop"
            android:taskAffinity=""
            android:theme="@style/LaunchTheme"

android:configChanges="orientation|keyboardHidden|keyboard|screenSi
ze|smallestScreenSize|locale|layoutDirection|fontScale|screenLayout|densi
ty|uiMode"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">

            <meta-data
                android:name="io.flutter.embedding.android.NormalTheme"
                android:resource="@style/NormalTheme" />

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
```

```
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <!-- Don't delete the meta-data below.
             This is used by the Flutter tool to generate
GeneratedPluginRegistrant.java -->
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>

    <!-- Required to query activities that can process text -->
    <queries>
        <intent>
            <action android:name="android.intent.action.PROCESS_TEXT"
/>
            <data android:mimeType="text/plain" />
        </intent>
    </queries>
</manifest>
```

Souce Code main.dart :

```
import 'package:flutter/material.dart';
import 'package:guided_pert12/homepage.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Google Maps Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MapsScreen(),
    );
  }
```

```
}
```

Souce Code homepage.dart:

```
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(-7.4347652, 109.2500561);
  static final CameraPosition _kInitialPosition = CameraPosition(
    target: _kMapCenter,
    zoom: 11.0,
  );

  late GoogleMapController _mapController;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Google Maps Demo'),
      ),
      body: GoogleMap(
        initialCameraPosition: _kInitialPosition,
        onMapCreated: (GoogleMapController controller) {
          _mapController = controller;
        },
        myLocationEnabled: true, // Menampilkan lokasi pengguna
        markers: _createMarker(), // Menambahkan marker
      ),
    );
  }

  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),
      infoWindow: InfoWindow(title: 'Marker 2'),
    ),
  };
 }
}
```

Souce pubspec.yaml:

```
name: guided_pert12
description: "A new Flutter project."
# The following line prevents the package from being accidentally
published to
# pub.dev using `flutter pub publish`. This is preferred for private
packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number for your
application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number
used as versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while
build-number is used as CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Refe
rence/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build
suffix.
version: 1.0.0+1

environment:
  sdk: ^3.5.3
```

```
# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
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
  location: ^7.0.1
  place_picker_google: ^0.0.13

dev_dependencies:
  flutter_test:
    sdk: flutter

  # The "flutter_lints" package below contains a set of recommended lints to
  # encourage good coding practices. The lint set provided by the package is
  # activated in the `analysis_options.yaml` file located at the root of your
  # package. See that file for information about deactivating specific lint
  # rules and activating additional ones.
  flutter_lints: ^4.0.0

# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter packages.
flutter:

  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
```

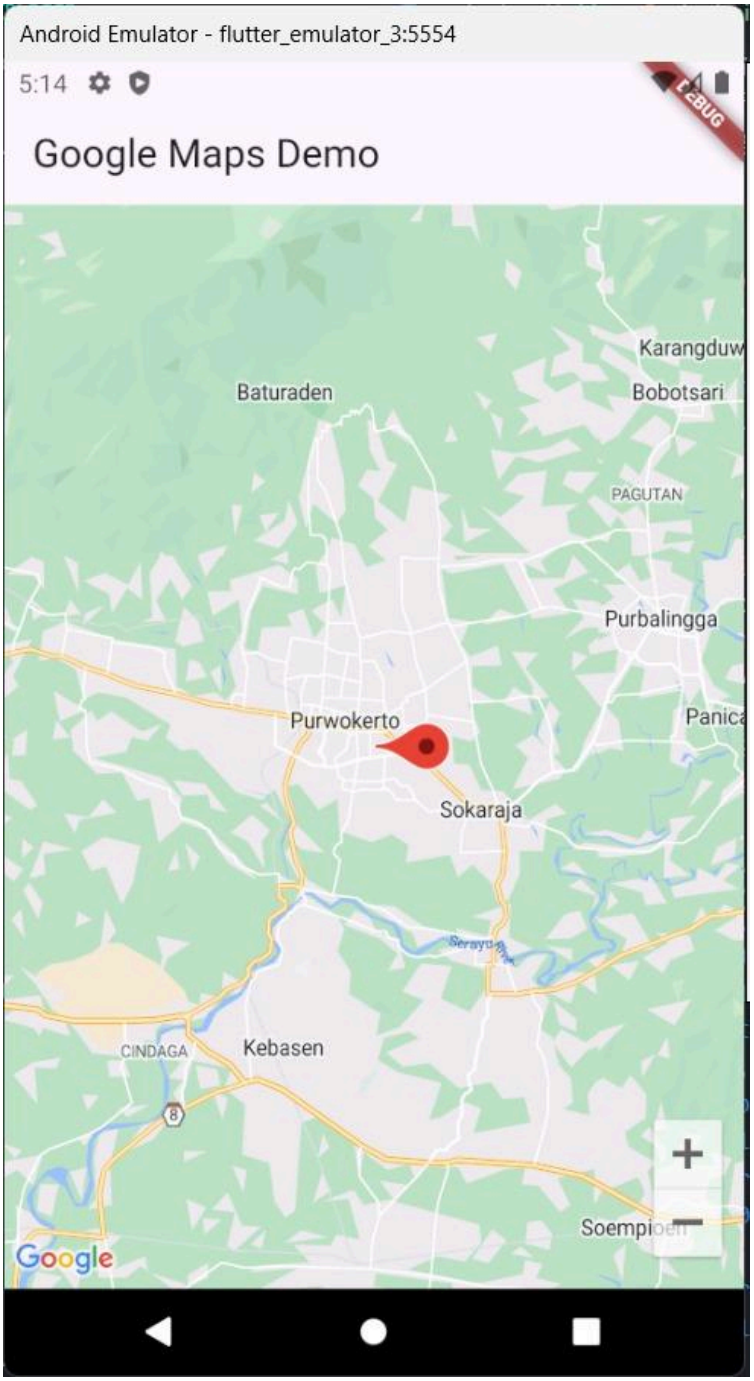
```
  uses-material-design: true

  # To add assets to your application, add an assets section, like this:
  # assets:
  #   - images/a_dot_burr.jpeg
  #   - images/a_dot_ham.jpeg

  # An image asset can refer to one or more resolution-specific "variants",
see
  # https://flutter.dev/to/resolution-aware-images

  # For details regarding adding assets from package dependencies, see
  # https://flutter.dev/to/asset-from-package

  # To add custom fonts to your application, add a fonts section here,
  # in this "flutter" section. Each entry in this list should have a
  # "family" key with the font family name, and a "fonts" key with a
  # list giving the asset and other descriptors for the font. For
  # example:
  # fonts:
  #   - family: Schyler
  #     fonts:
  #       - asset: fonts/Schyler-Regular.ttf
  #       - asset: fonts/Schyler-Italic.ttf
  #         style: italic
  #   - family: Trajan Pro
  #     fonts:
  #       - asset: fonts/TrajanPro.ttf
  #       - asset: fonts/TrajanPro_Bold.ttf
  #         weight: 700
  #
  # For details regarding fonts from package dependencies,
  # see https://flutter.dev/to/font-from-package
```

Souce build.gradle:

```
allprojects {
    repositories {
        google()
        mavenCentral()
    }
}

rootProject.buildDir = "../build"
```

```
subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
}
subprojects {
    project.evaluationDependsOn(":app")
}

tasks.register("clean", Delete) {
    delete rootProject.buildDir
}
```

Output GUIDED:



UNGUIDED

Souce Code AndroidManifest.xml :

```
<manifest
xmlns:android="http://schemas.android.com/apk/res/android">
   <application
      android:label="pertemuan12maps"
      android:name="${applicationName}"
      android:icon="@mipmap/ic_launcher">
      <!-- API MAPS -->
      <meta-data android:name="com.google.android.geo.API_KEY"

android:value="AIzaSyCaF6C9X2P9bNBGl0SEJ3r8ux6jYZF-6mc"/>
      <activity
         android:name=".MainActivity"
         android:exported="true"
         android:launchMode="singleTop"
         android:taskAffinity=""
         android:theme="@style/LaunchTheme"

android:configChanges="orientation|keyboardHidden|keyboard|screenSi
ze|smallestScreenSize|locale|layoutDirection|fontScale|screenLayout|densi
ty|uiMode"
         android:hardwareAccelerated="true"
         android:windowSoftInputMode="adjustResize">
         <!-- Specifies an Android theme to apply to this Activity as soon as
              the Android process has started. This theme is visible to the
user
              while the Flutter UI initializes. After that, this theme continues
              to determine the Window background behind the Flutter UI.
-->
         <meta-data
           android:name="io.flutter.embedding.android.NormalTheme"
           android:resource="@style/NormalTheme"
           />
         <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category
android:name="android.intent.category.LAUNCHER"/>
         </intent-filter>
      </activity>
      <!-- Don't delete the meta-data below.
           This is used by the Flutter tool to generate
GeneratedPluginRegistrant.java -->
```

```
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>
    <!-- Required to query activities that can process text, see:
         https://developer.android.com/training/package-visibility and

https://developer.android.com/reference/android/content/Intent#ACTIO
N_PROCESS_TEXT.

        In particular, this is used by the Flutter engine in
io.flutter.plugin.text.ProcessTextPlugin. -->
    <queries>
        <intent>
            <action
android:name="android.intent.action.PROCESS_TEXT"/>
            <data android:mimeType="text/plain"/>
        </intent>
    </queries>

</manifest>
```

Souce Code main.dart :

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

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

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(home:HomePage());
  }
}
```

Souce Code homepage.dart:

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

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

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  static const LatLng _initialMapCenter = LatLng(-7.4352631,
109.2465177);
  static final CameraPosition _initialCameraPosition = CameraPosition(
    target: _initialMapCenter,
    zoom: 11.0,
  );

  late GoogleMapController _mapController;

  // Fungsi untuk menangani pembentukan controller Google Maps
  void _onMapCreated(GoogleMapController controller) {
    _mapController = controller;
  }

  // Fungsi untuk membuat marker default
  Set<Marker> _createMarker() {
    return {
      Marker(
        markerId: const MarkerId("marker_1"),
        position: _initialMapCenter,
        infoWindow: const InfoWindow(title: 'kampus'),
        rotation: 90,
      ),
    };
  }

  // Fungsi untuk membuka Place Picker
  void _openPlacePicker() async {
    try {
      LocationResult? result = await Navigator.of(context).push(
        MaterialPageRoute(
          builder: (context) => PlacePicker(
            "AIzaSyCJ_C0Tc29ZQ4lRmCTTDGt7hmuXL3e3pTg", // Ganti
```

```
dengan API Key Anda
          displayLocation: _initialMapCenter, // Lokasi awal pada Place
Picker
        ),
      ),
    );

    if (result != null && result.latLng != null) {
      print("Place Picked: ${result.formattedAddress}");

      // Memindahkan kamera ke lokasi yang dipilih
      _mapController.animateCamera(
        CameraUpdate.newLatLng(result.latLng!),
      );

      // Menambahkan marker pada lokasi yang dipilih
      setState(() {
        _createMarker().add(
          Marker(
            markerId: const MarkerId("picked_location"),
            position: result.latLng!,
            infoWindow: InfoWindow(
              title: "Picked Location",
              snippet: result.formattedAddress,
            ),
          ),
        );
      });
    }
  } catch (e) {
    print("Error picking place: $e");
  }
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Latihan Maps'),
      centerTitle: true,
      actions: [
        IconButton(
          onPressed: _openPlacePicker,
          icon: const Icon(Icons.search),
        ),
```

```
        ],
      ),
      body: GoogleMap(
        initialCameraPosition: _initialCameraPosition,
        myLocationEnabled: true,
        onMapCreated: _onMapCreated,
        zoomControlsEnabled: false, // Menghilangkan kontrol zoom default
        markers: _createMarker(),
      ),
    );
  }

  @override
  void dispose() {
    _mapController.dispose(); // Membersihkan controller saat widget
dihancurkan
    super.dispose();
  }
}
```

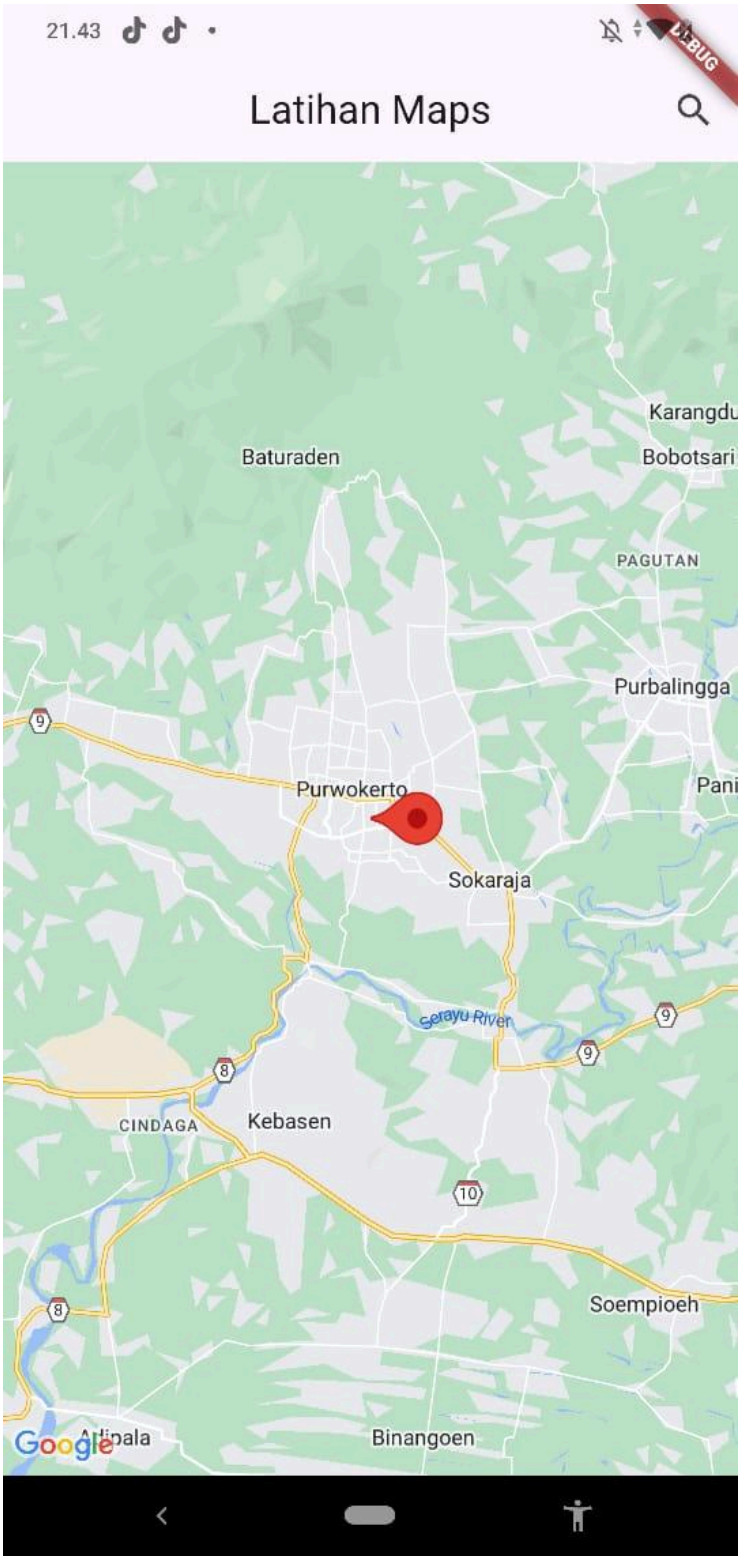
Souce build.gradle:

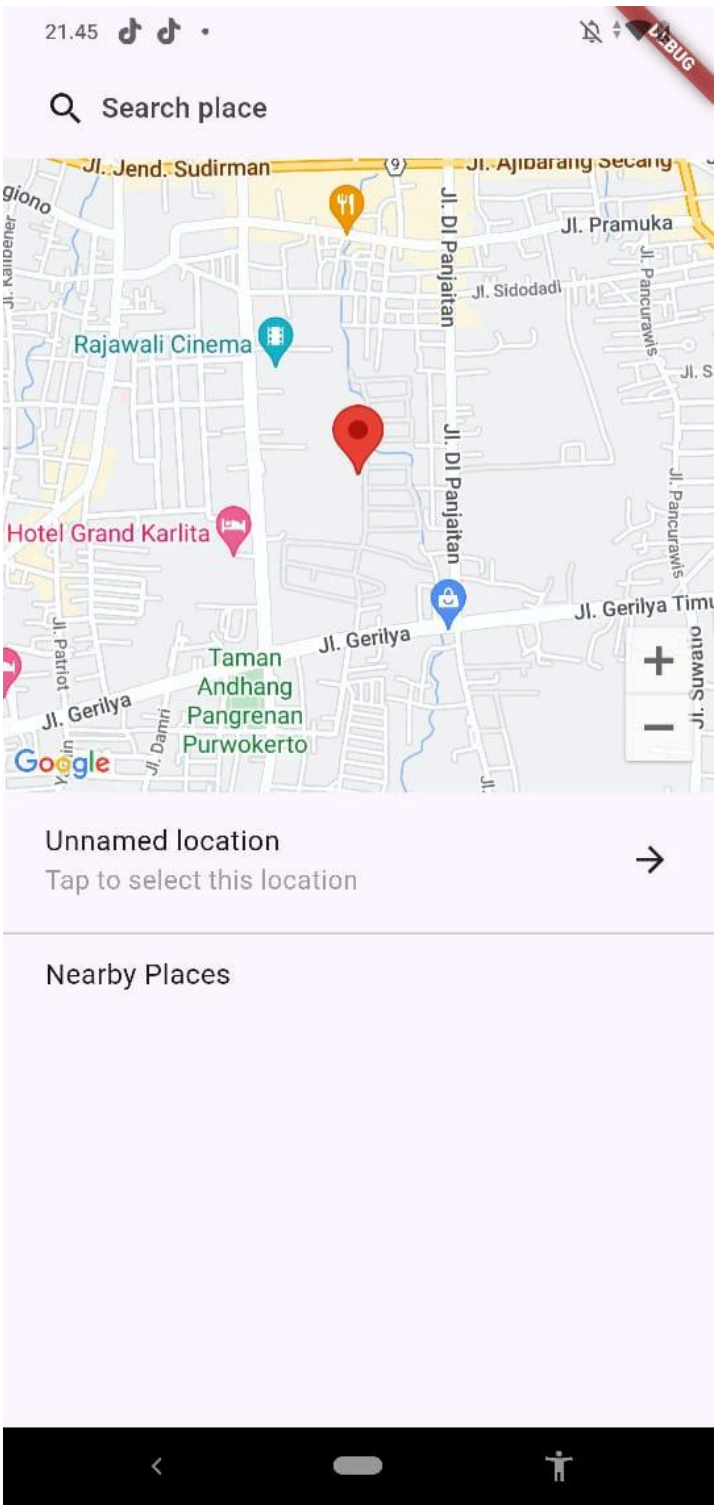
```
allprojects {
    repositories {
        google()
        mavenCentral()
    }
}

rootProject.buildDir = "../build"
subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
}
subprojects {
    project.evaluationDependsOn(":app")
}

tasks.register("clean", Delete) {
    delete rootProject.buildDir
}
```

Souce pubspec.yaml:

```
name: cobaaaaaaaaaaa
description: "A new Flutter project."
publish_to: "none"
version: 1.0.0+1

environment:
  sdk: ^3.5.3

dependencies:
  cupertino_icons: ^1.0.8
  flutter:
    sdk: flutter
  google_maps_flutter: ^2.10.0
  place_picker: ^0.10.0

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^4.0.0

flutter:
  uses-material-design: true
```

Output Programs UNGUIDED




21.45
DEBUG
Search place
Jl. Jend. Sudirman
Jl. Ajibarang Secang
Jl. Pramuka
Jl. Sidodadi
Jl. Pancurawis
Jl. Dl Panjaitan
Rajawali Cinema
Hotel Grand Karlita
Jl. Gerilya Timu
Jl. Gerilya
Jl. Patriot
Taman
Andhang
Pangrenan
Purwokerto
Jl. Gerilya
Jl. Damri
Google
Unnamed location
Tap to select this location
Nearby Places

Deskripsi

ProgramProyek ini menggunakan beberapa dependensi seperti Google Maps Flutter dan Place Picker, yang menunjukkan bahwa aplikasi ini kemungkinan akan memiliki fitur berbasis peta, seperti menampilkan lokasi atau memilih tempat. Proyek juga mendukung desain berbasis Material Design dan menggunakan SDK Flutter versi terbaru (3.5.3).

Informasi Dasar:

- Nama Proyek: `cobaaaaaaaaaaa`
- Deskripsi: Proyek Flutter baru.
- Versi Aplikasi: `1.0.0+1` (versi awal aplikasi).

Lingkungan (Environment):

- Memanfaatkan SDK Flutter versi `^3.5.3`, yang mendukung fitur-fitur terbaru dalam pengembangan aplikasi.

Dependensi Utama:

- flutter: Paket inti yang menyediakan kerangka kerja Flutter.
- cupertino_icons: Digunakan untuk ikon berbasis gaya Cupertino (gaya iOS).
- google_maps_flutter: Digunakan untuk integrasi Google Maps dalam aplikasi, memungkinkan penggunaan fitur peta interaktif.
- place_picker: Untuk memilih lokasi di peta, melengkapi fitur berbasis Google Maps.

Dependensi Pengembangan (Dev Dependencies):

- flutter_test: Digunakan untuk menulis dan menjalankan pengujian aplikasi.
- flutter_lints: Memberikan aturan linting untuk menjaga kualitas kode.

Pengaturan Desain:

- uses-material-design: Mengaktifkan Material Design, memastikan aplikasi memiliki tampilan modern yang konsisten dengan prinsip desain Flutter.

Tujuan Proyek:

Proyek ini tampaknya dirancang untuk membuat aplikasi yang memanfaatkan fitur peta interaktif, seperti menampilkan lokasi, memilih tempat, atau memberikan panduan berbasis lokasi. Cocok untuk aplikasi seperti layanan perjalanan, panduan kota, atau layanan berbasis lokasi lainnya.