1 D4-TEKNIK KOMPUTER C

Android Activities dan Intents

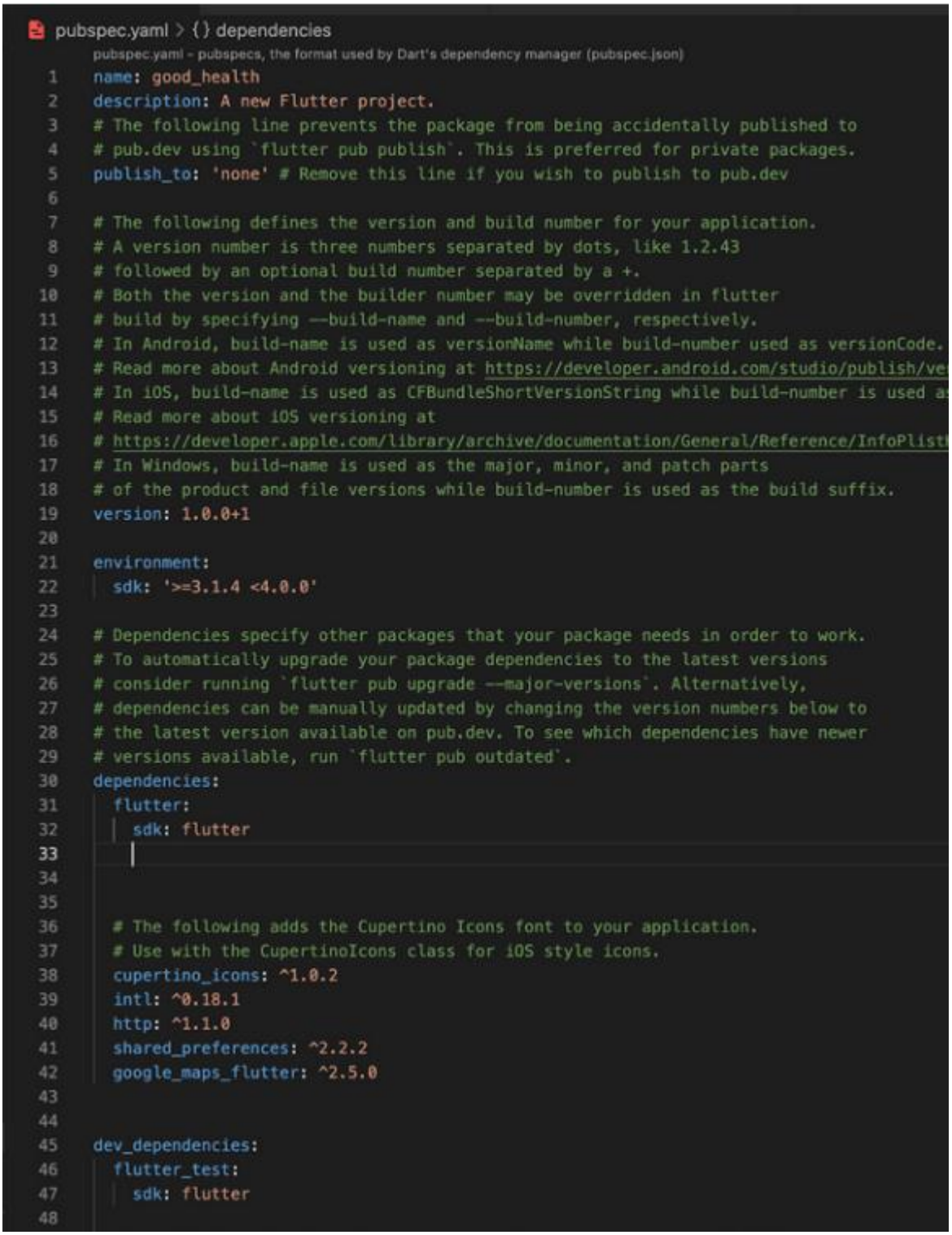
Nama	: Qaedy Qashmal Robbani
NRP	: 3224600063
Kelas	: 1 D4 Teknik Komputer C
Mata Kuliah	: Bahasa Penrograman
Dosen	: Maretha Ruswiansari S.ST., M.T
Hari/Tanggal Praktikum	: Selasa, 11 November 2025

1. Tujuan

- Mahasiswa dapat menjelaskan penerapan basis data dalam aplikasi
- Mahasiswa dapat mengembangkan *backend* dan *frontend* aplikasi
- Mahasiswa dapat menganalisis cara kerja aplikasi

2. Prosedur

1. Buka VS Code, pilih **View > Command Palette** seperti berikut
2. Beri nama *project*, misal **good_health**, kemudian tunggu hingga project selesai disiapkan
3. Cari pada *project folder* ‘good_health’ untuk file **pubspec.yaml**, dimana file ini berisi metadata tentang *package* yang akan digunakan pada *project*. Tambahkan beberapa baris setelah ‘cupertino.icons: ’, kemudian lanjutkan dengan *save file* dan tunggu proses sinkronisasi selesai.



pubspec.yaml > {} dependencies

pubspec.yaml - pubspecs, the format used by Dart's dependency manager (pubspec.json)

```yaml
name: good_health
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/ve
# In iOS, build-name is used as CFBundleShortVersionString while build-number is used a
# Read more about iOS versioning at
# https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlist
# 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: '>=3.1.4 <4.0.0'

# 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: ^1.0.2
  intl: ^0.18.1
  http: ^1.1.0
  shared_preferences: ^2.2.2
  google_maps_flutter: ^2.5.0


dev_dependencies:
  flutter_test:
    sdk: flutter
```

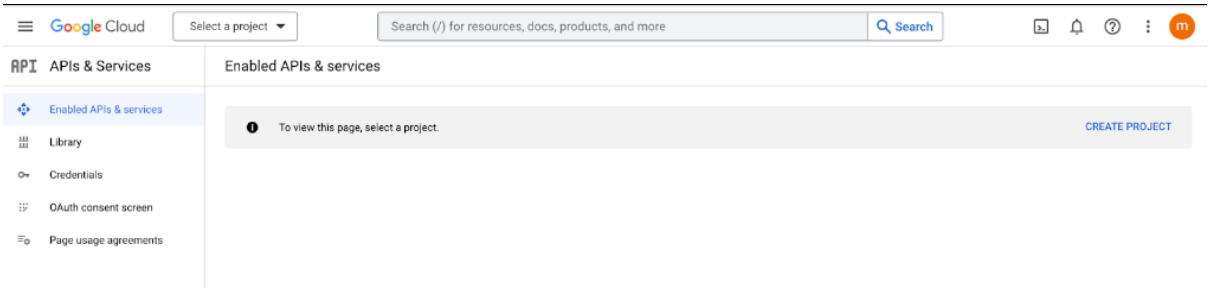
```
cupertino_icons: ^1.0.2
intl: ^0.18.1
http: ^1.1.0
shared_preferences: ^2.2.2
google_maps_flutter: ^2.5.0
```

4. Pada pengembangan aplikasi ini membutuhkan Google Maps API *Setup*, sehingga perlu melakukan *setup* API *key* agar bisa menampilkan *maps* dalam *widget* :

   a. Android

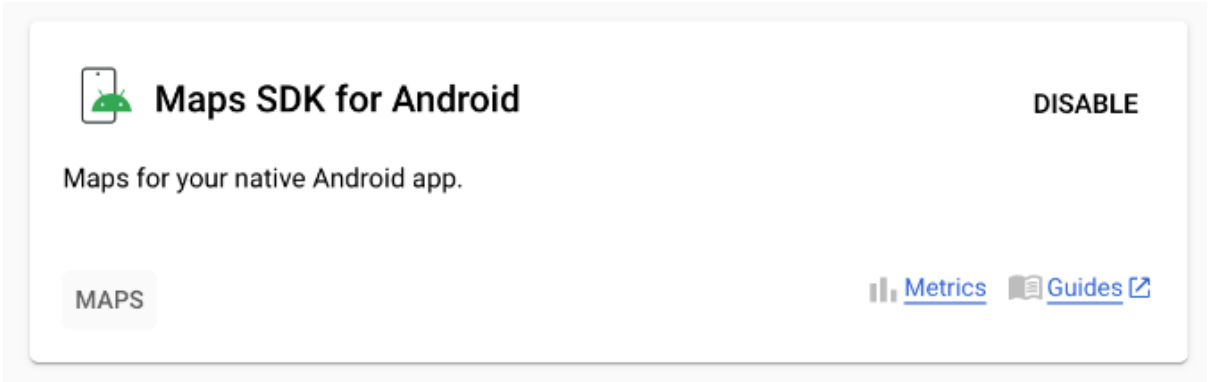
   - Buka browser dan menuju ke halaman https://console.cloud.google.com/

   - Pilih APIs & Services > Enabled APIs & Services



   - Selanjutnya pilih CREATE PROJECT, isikan Good Health sebagai nama *project*

   - Kemudian Pilih Maps SDK for Android dan klik *enable*



   - Pilih Keys & Credentials > CREATE CREDENTIALS > API Key, kemudian *copy* API Key tersebut

   b. iOS

   - Untuk iOS langkahnya sama, namun perlu mengaktifkan Maps SDK for iOS

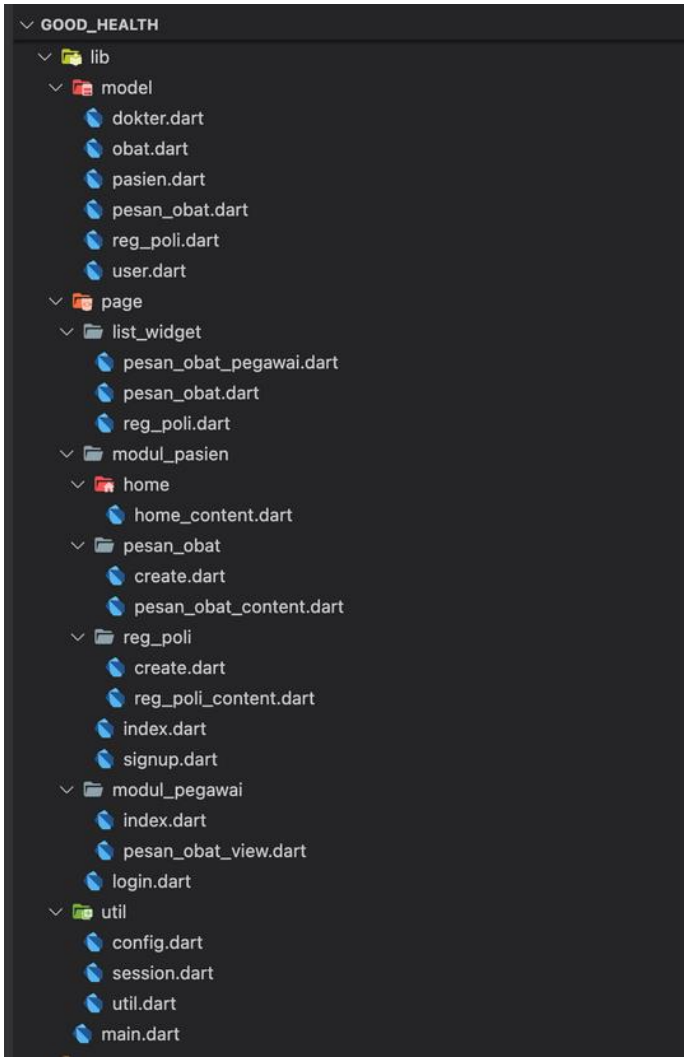
5. Selanjutnya buka *project* good_health > android > app > src > main > AndroidManifest.xml, kemudian tambahkan beberapa baris berikut :

```
android > app > src > main > AndroidManifest.xml
1   anifest xmlns:android="http://schemas.android.com/apk/res/android">
2
3
4     <uses-permission android:name="android.permission.INTERNET" />
5     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
6     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
7     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
8
9
10    <application
11        android:label="good_health"
12        android:name="${applicationName}"
13        android:icon="@mipmap/ic_launcher">
14
15        <meta-data android:name="com.google.android.geo.API_KEY"
16             android:value="AIzaSyC6AwyUtSVQ2Lf_hPAq4HVBVx5MgEhGpmU"/>
17
18        <activity
19            android:name=".MainActivity"
20            android:exported="true"
21            android:launchMode="singleTop"
```

6. Sedangkan untuk iOS dapat membuka *project* good_health > ios > Runner > Base.lproj > AppDelegate.Swift, kemudian tambahkan :

```
ios > Runner > AppDelegate.swift
 1  import UIKit
 2  import Flutter
 3  import GoogleMaps
 4
 5  @UIApplicationMain
 6  @objc class AppDelegate: FlutterAppDelegate {
 7    override func application(
 8      _ application: UIApplication,
 9      didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
10    ) -> Bool {
11      GMSServices.provideAPIKey("AIzaSyC6AwyUtSVQ2Lf_hPAq4HVBVx5MgEhGpmU");
12      GeneratedPluginRegistrant.register(with: self)
13      return super.application(application, didFinishLaunchingWithOptions: launchOptions)
14    }
15  }
16
```

7. Selanjutkan terkait *project organizer,* semua *source code* akan dibuat di dalam folder **lib,** seperti berikut :

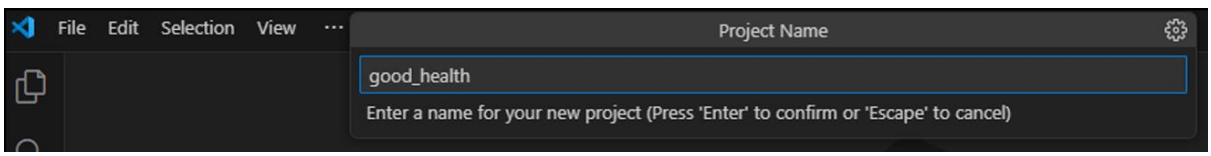


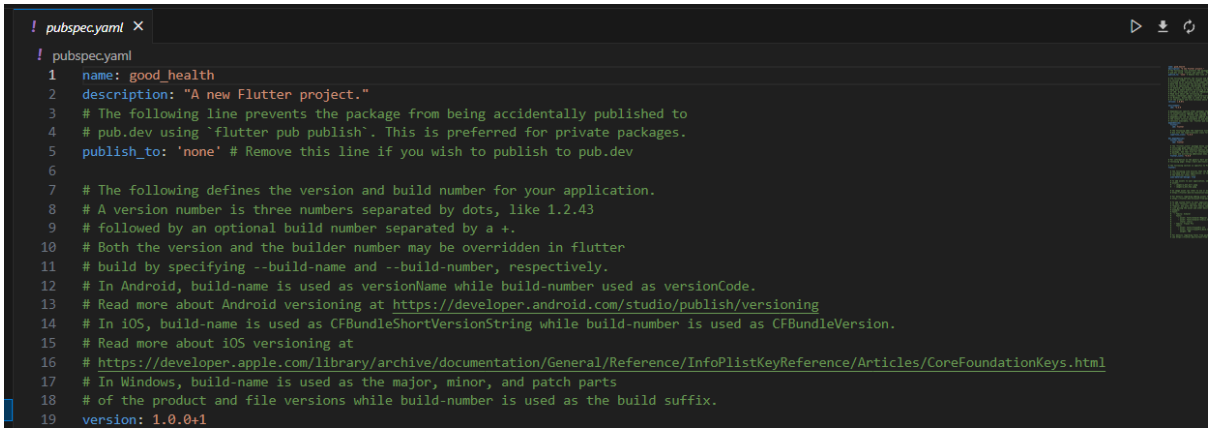
8. Selanjutnya buat folder baru dengan nama **model**, dimana dalam folder baru ini perlu membuat 6 file baru yaitu : **dokter.dart, obat.dart, pasien.dart, pesan_obat.dart, reg_poli.dart, user.dart**

9. Untuk *session and utility,* maka perlu membuat folder **util**, dimana perlu menambah 2 file baru **session.dart** dan **util.dart**
10. Berikutnya kita akan membuat file **main.dart** dan edit file
11. Untuk membuat *login interface*, buat folder baru **page**, dan buat file **login.dart**
12. Untuk membuat *register interface*, kita akan membuat membuat folder **modul_pasien** dan buat file baru **signup.dart**
13. Nantinya user pasien akan melihat 3 menu yaitu Home, Registrasi Poli, dan Pesan Obat, maka kita buat 3 folder baru yaitu : **reg_poli,home,** dan **modul_pasien,** kemudian di dalam folder **modul_pasien**, buatlah file **index.dart**
14. User tidak hanya pasien tetapi juga pegawai, maka kita buat folder baru bernama **modul_pegawai,** yang didalamnya akan terdapat **file index.dart**
15. Selanjutnya, kita perlu membuat folder baru dengan nama **list_widget** di dalam folder **page,** kemudian buat file baru **pesan_obat_pegawai.dart, pesan_obat.dart, dan reg_poli.dart**
16. Dalam folder **modul_pegawai,** buat file baru dengan nama **pesan_obat_view.dart**
17. Dalam folder **modul pasien,** kita buat folder baru **reg_poli** tambahkan file baru **reg_poli_content.dart**
18. Untuk *widget,* kita perlu membuat file baru **reg_poli.dart** dalam folder **list_widget**
19. Selanjutnya tambahkan file dengan nama **create.dart** dalam folder **reg_poli**
20. Buat file baru dengan nama **pesan_obat_content.dart** dalam folder **pesan_obat**
21. Buat file baru dengan nama **pesan_obat.dart** di dalam folder **list_widget**
22. Tambahkan file dengan nama **create.dart** dalam folder **pesan_obat**

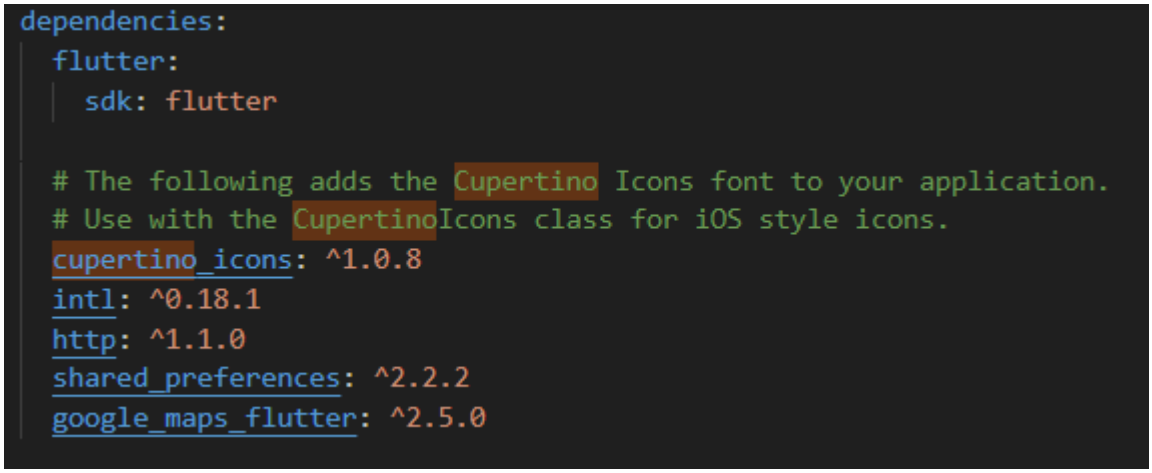
3. Hasil Percobaan

3.1 VS code



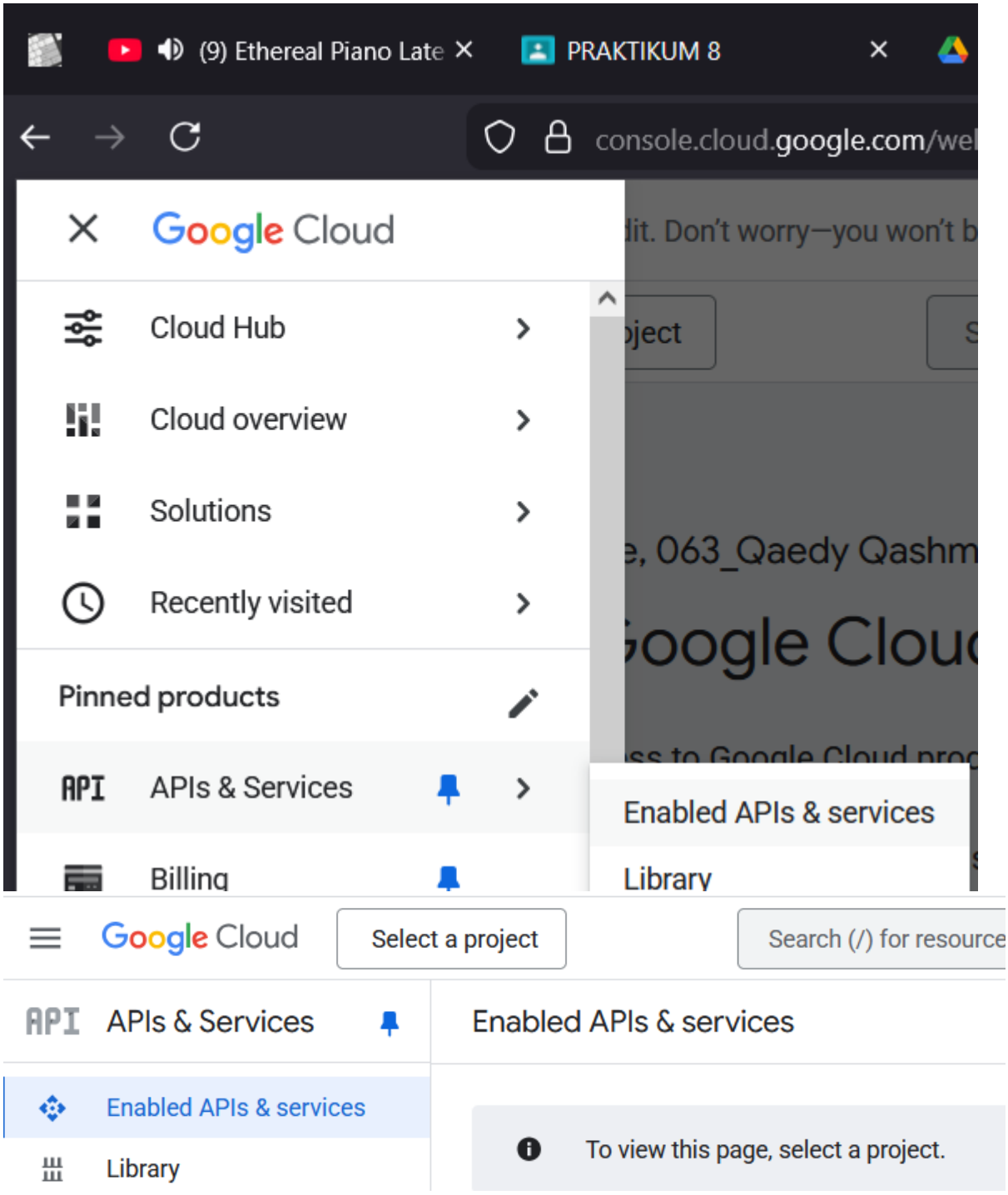


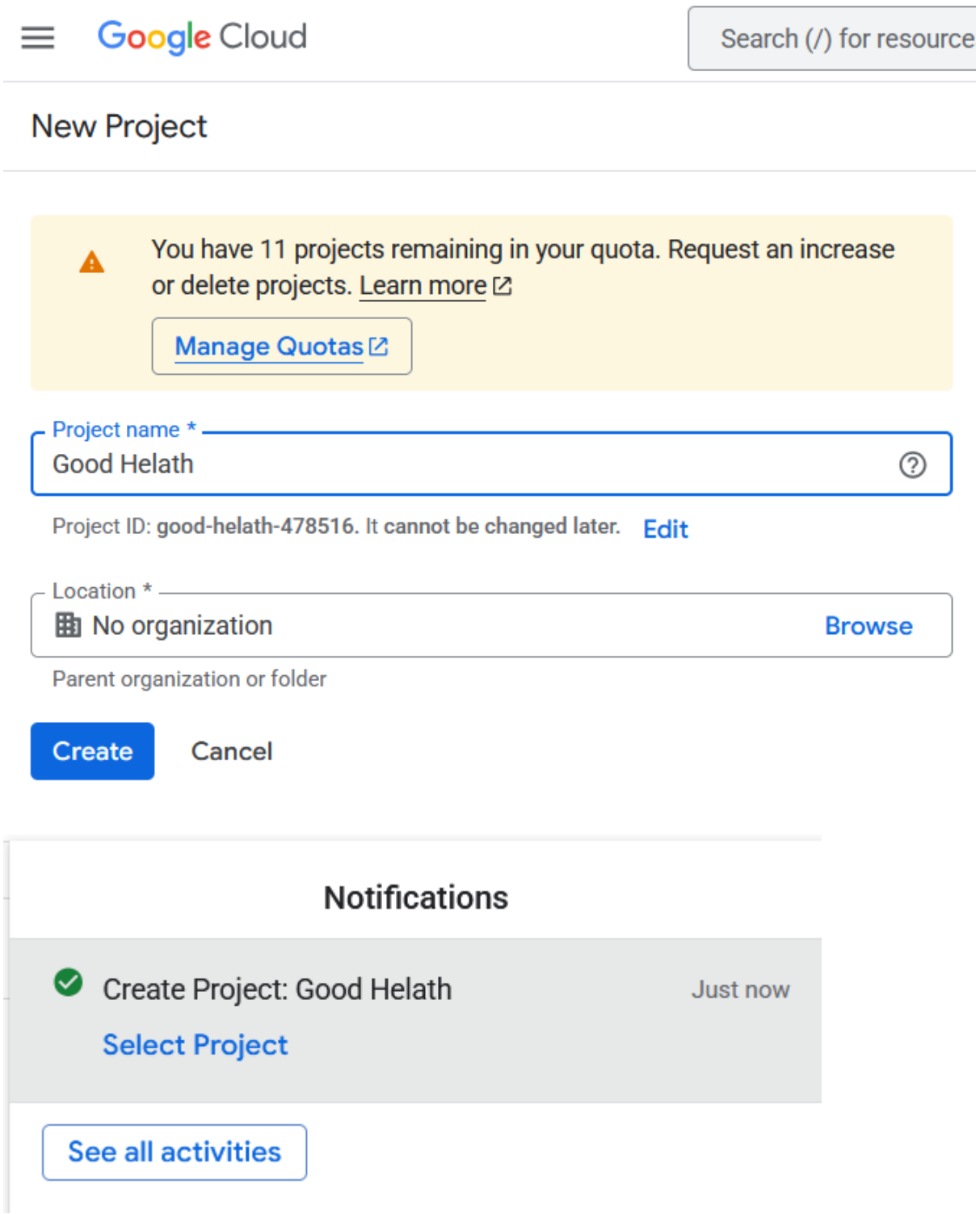
```
name: good_health
description: "A new Flutter project."
# The following line prevents the package from being accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for private packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# 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/Reference/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
```



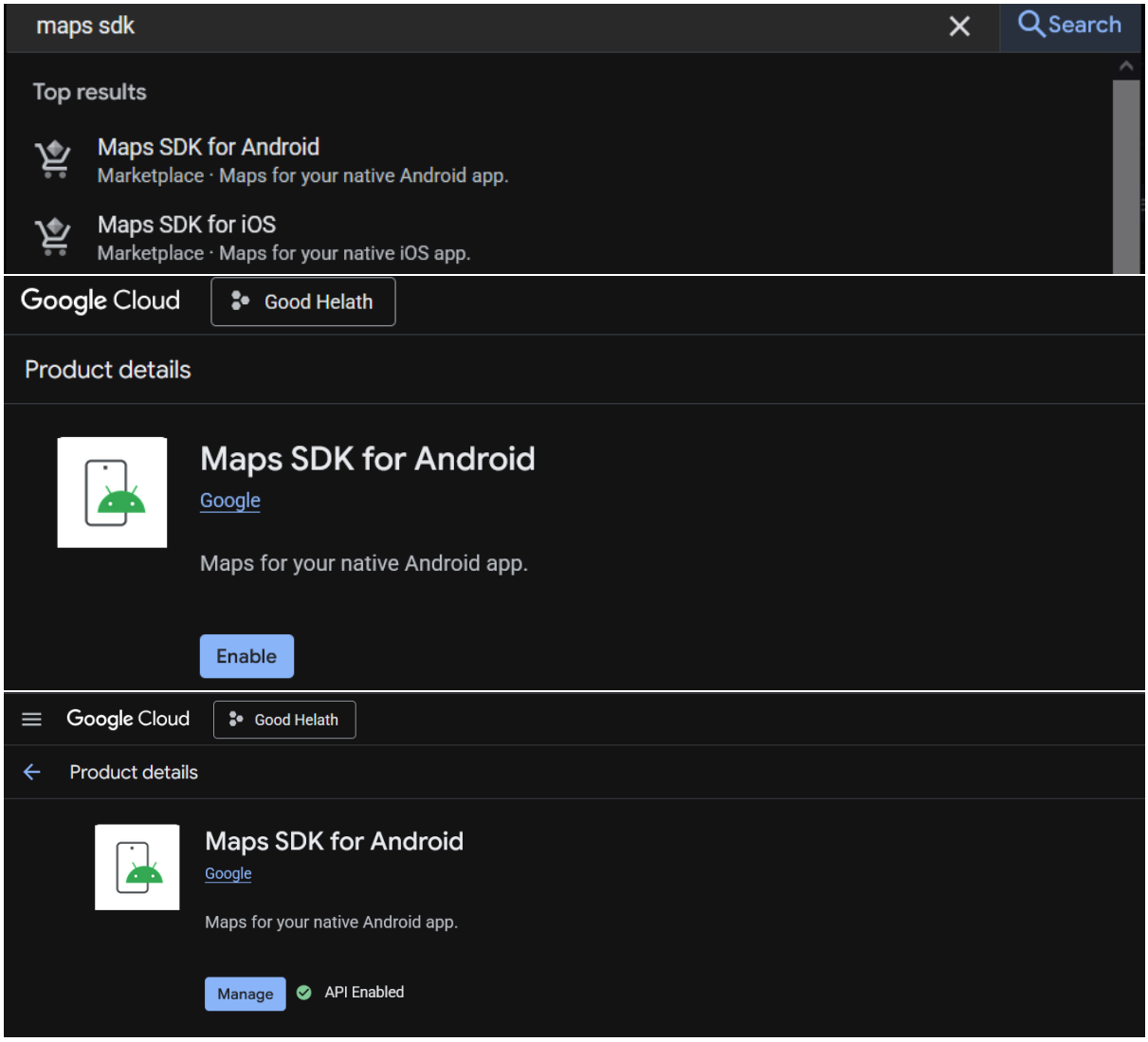
```
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  intl: ^0.18.1
  http: ^1.1.0
  shared_preferences: ^2.2.2
  google_maps_flutter: ^2.5.0
```

3.2 Setup API

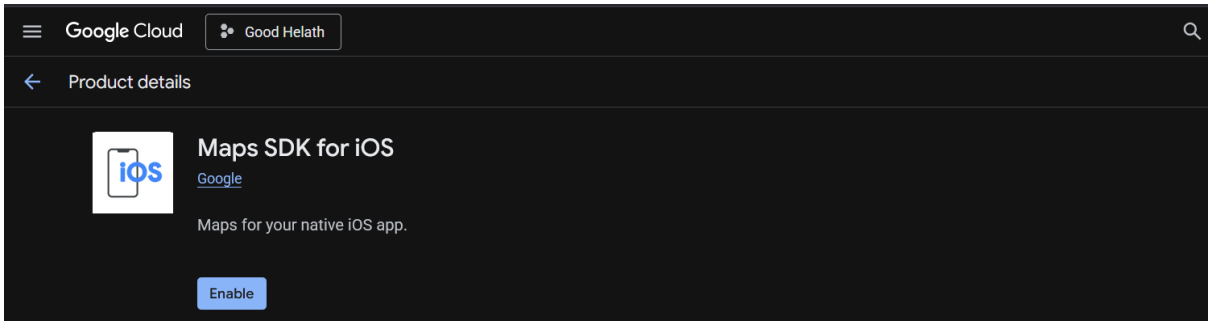

(9) Ethereal Piano Late
PRAKTIKUM 8
console.cloud.google.com/wel
Google Cloud
Cloud Hub
Cloud overview
Solutions
Recently visited
Pinned products
APIs & Services
Billing
Enabled APIs & services
Library
Google Cloud
Select a project
Search (/) for resource
APIs & Services
Enabled APIs & services
Library
To view this page, select a project.

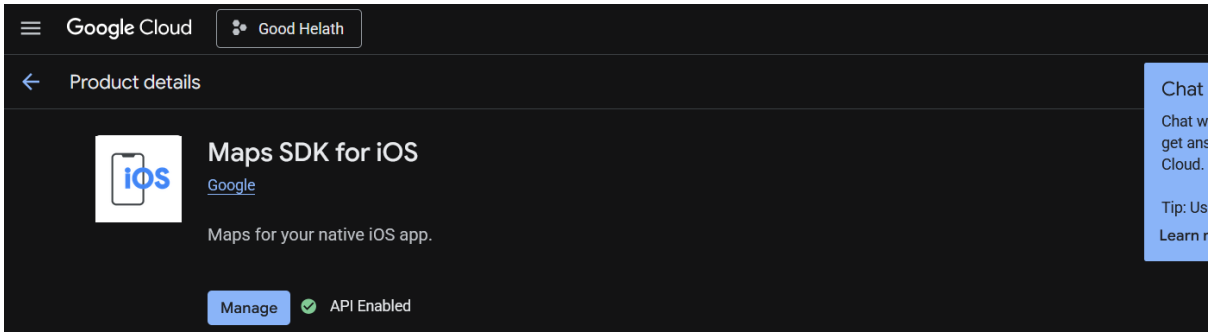

Google Cloud
Search (/) for resource
New Project
You have 11 projects remaining in your quota. Request an increase or delete projects. Learn more
Manage Quotas
Project name *
Good Helath
Project ID: good-helath-478516. It cannot be changed later. Edit
Location *
No organization
Browse
Parent organization or folder
Create
Cancel
Notifications
Create Project: Good Helath
Just now
Select Project
See all activities

3.3 Maps SDK for Android

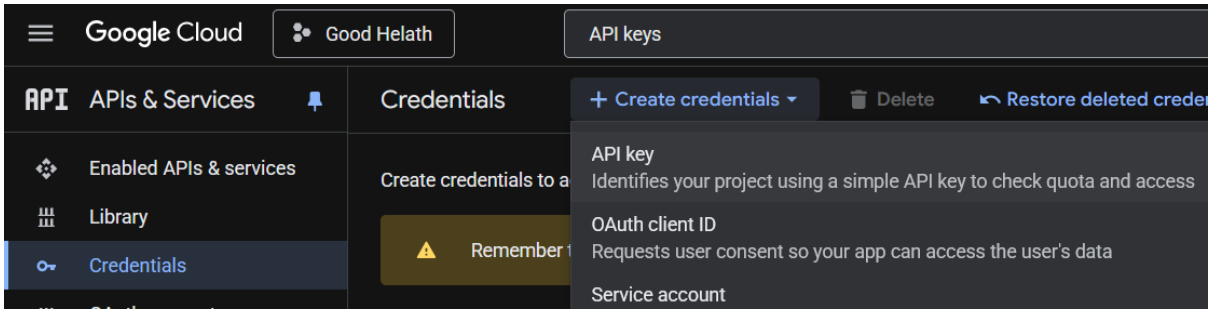


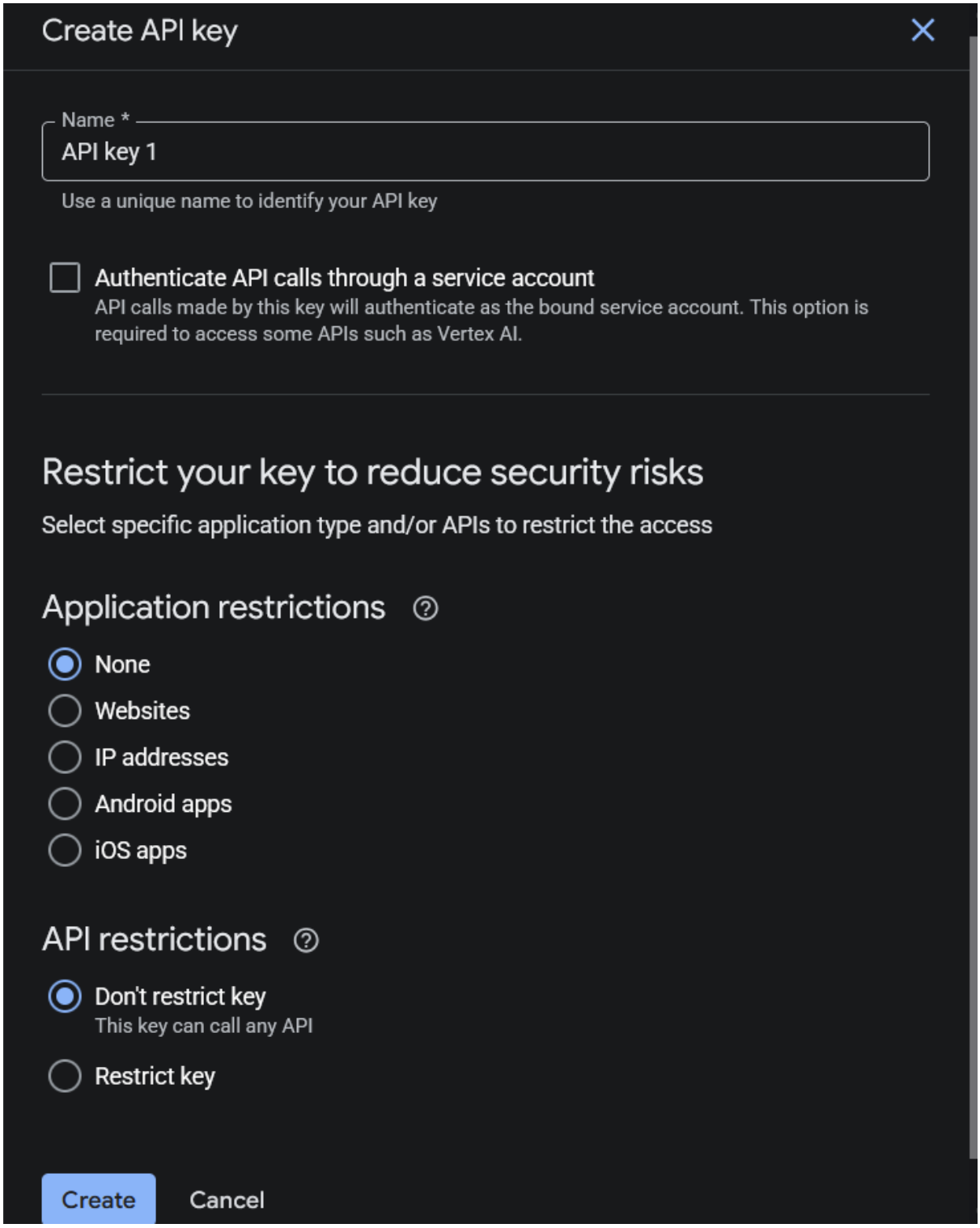
3.4 Maps SDK for iOS



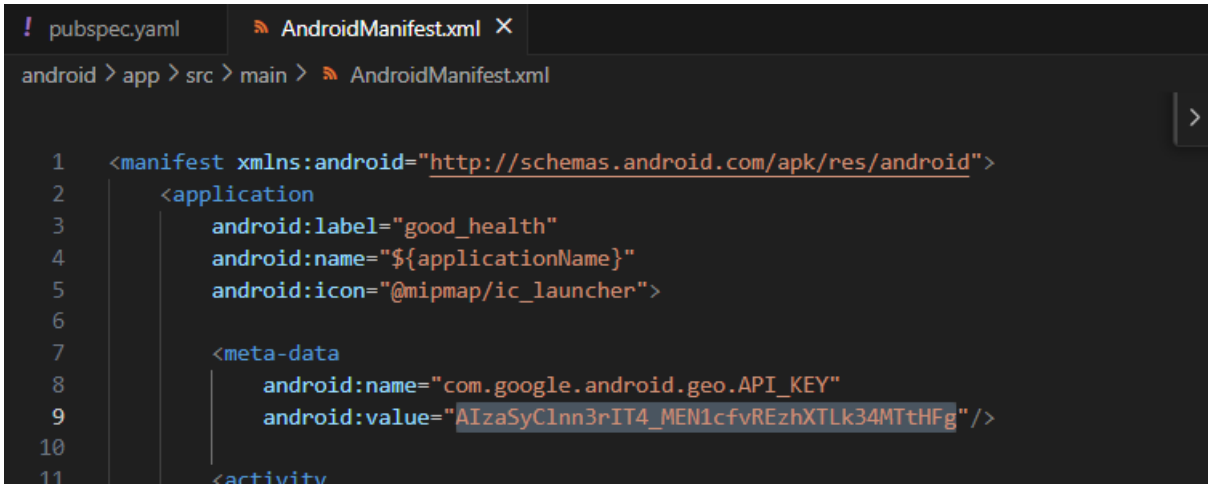


3.5 Create API Keys

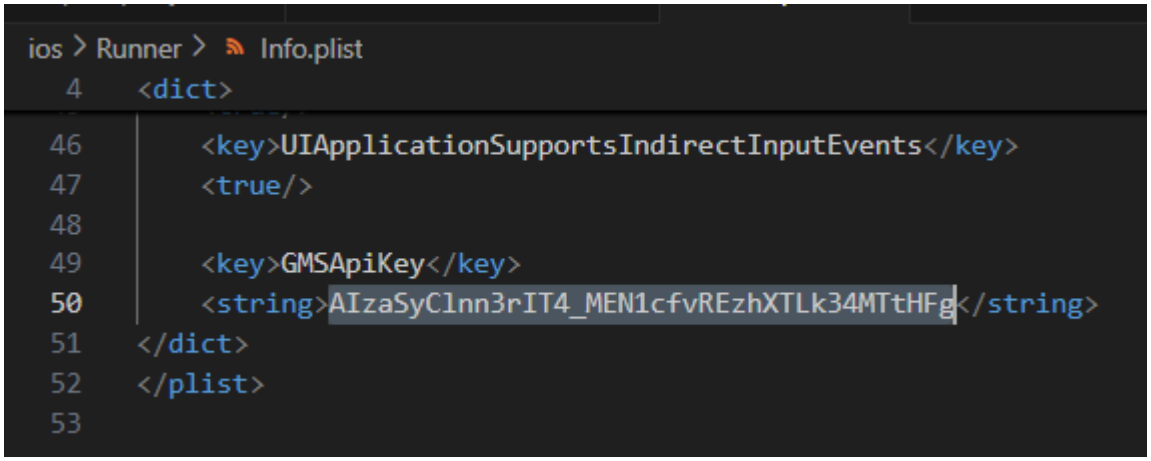



Create API key
Name *
API key 1
Use a unique name to identify your API key
Authenticate API calls through a service account
API calls made by this key will authenticate as the bound service account. This option is required to access some APIs such as Vertex AI.
Restrict your key to reduce security risks
Select specific application type and/or APIs to restrict the access
Application restrictions
None
Websites
IP addresses
Android apps
iOS apps
API restrictions
Don't restrict key
This key can call any API
Restrict key
Create
Cancel

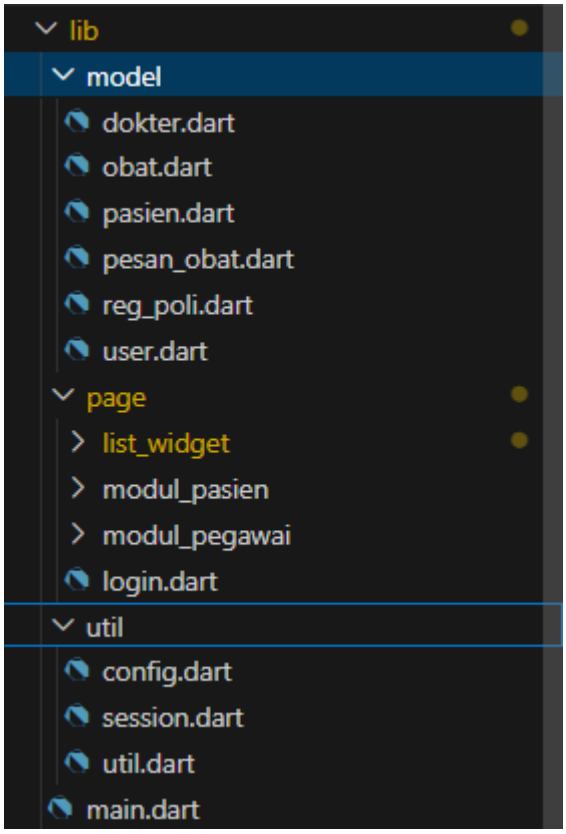
3.6 Copy API key ke AndroidManifest.xml (Android)



3.7 Copy API key ke AppDelegate.swift (IOS)(according to ChatGPT)



3.8 Isi folder lib

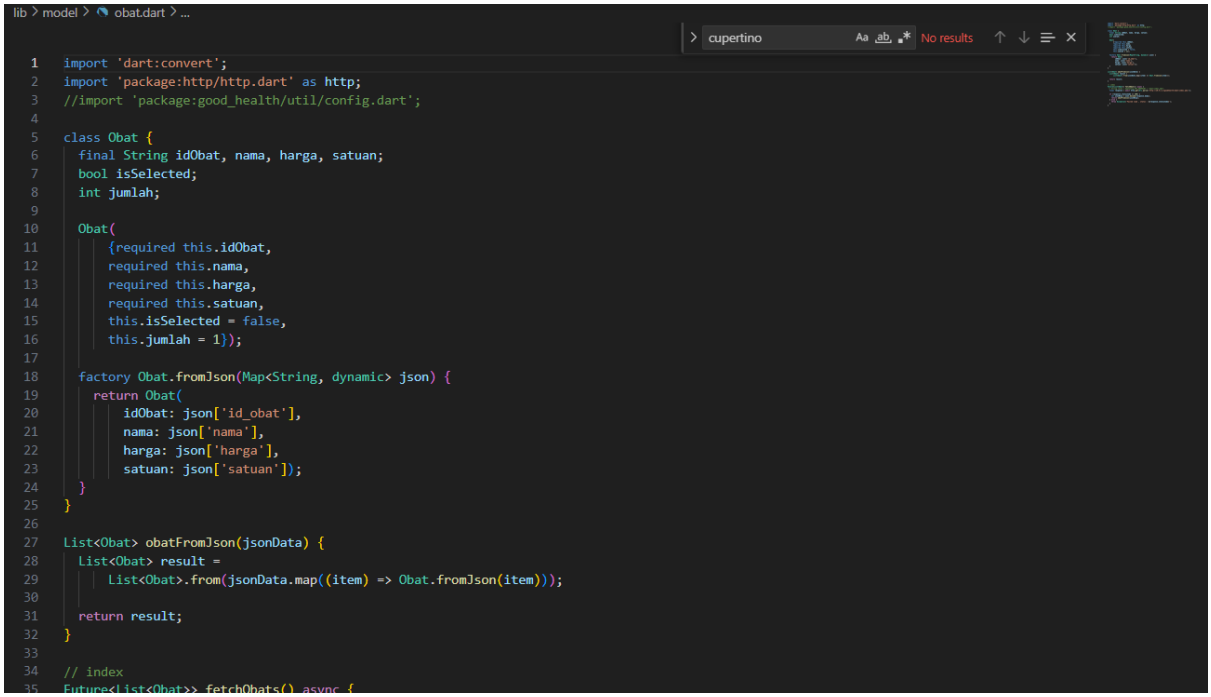


3.9 Doctor.dart

lib > model > dokter.dart > ...

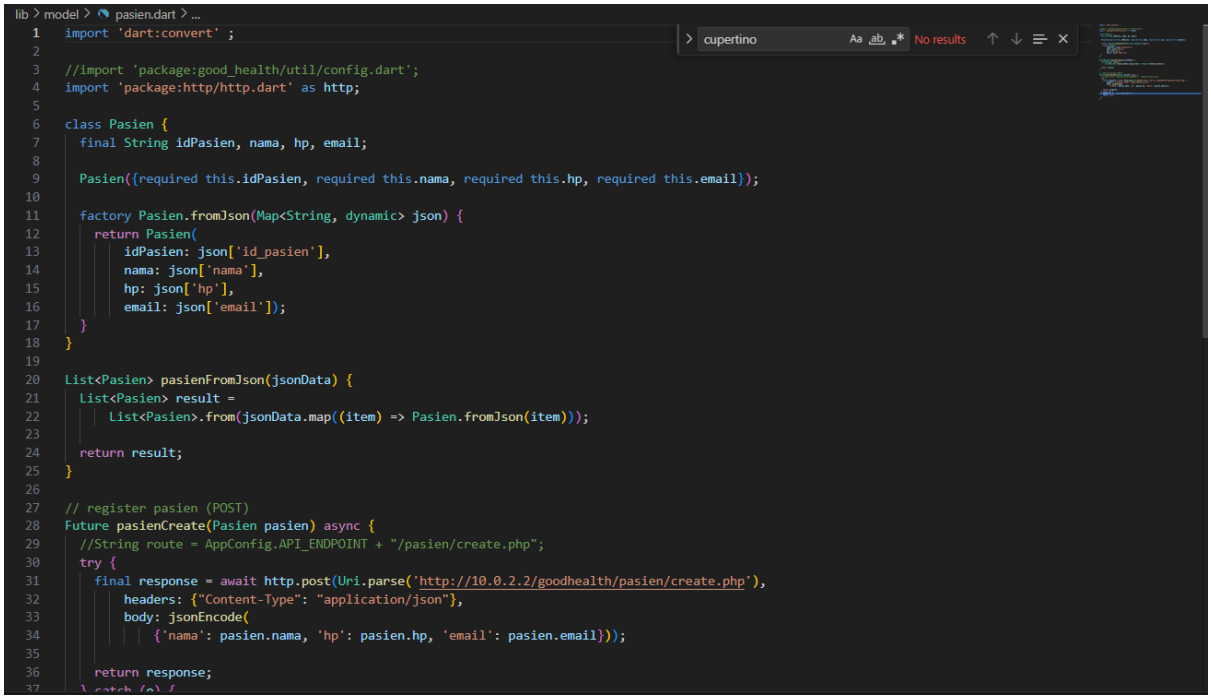
```dart
import 'dart:convert';
import 'package:http/http.dart' as http;
//import 'package:good_health/util/config.dart';

class Dokter {
  final String idDokter, nama, hp;

  Dokter({required this.idDokter, required this.nama, required this.hp});

  factory Dokter.fromJson(Map<String, dynamic> json) {
    return Dokter(
        idDokter: json['id_dokter'], nama: json['nama'], hp: json['hp']);
  }
}

List<Dokter> dokterFromJson(jsonData) {
  List<Dokter> result =
      List<Dokter>.from(jsonData.map((item) => Dokter.fromJson(item)));

  return result;
}

// index
Future<List<Dokter>> fetchDokters() async {
  //String route = AppConfig.API_ENDPOINT + "dokter/index.php";
  final response = await http.get(Uri.parse('http://10.0.2.2/goodhealth/dokter/index.php'));

  if (response.statusCode == 200) {
    var jsonResp = json.decode(response.body);
    return dokterFromJson(jsonResp);
  } else {
    //throw Exception('Failed load $route, status : ${response.statusCode}');
    throw Exception('Failed load, status : ${response.statusCode}');
  }
}
```

3.10 Obat.dart



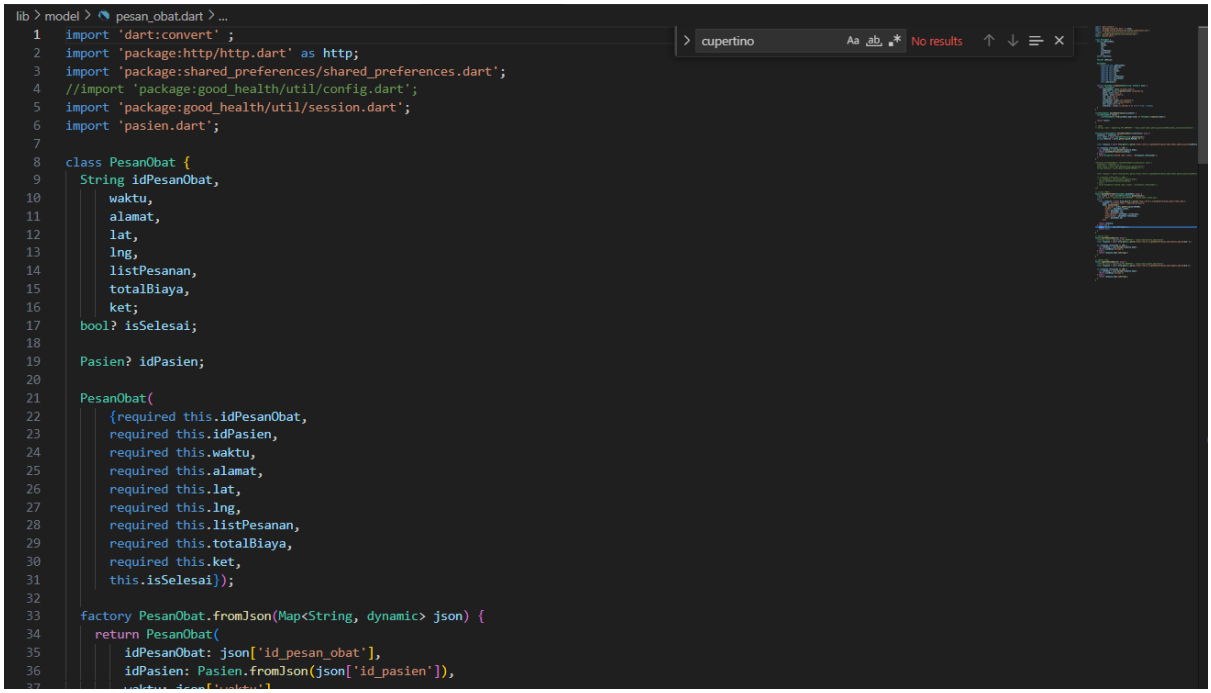
```dart
import 'dart:convert';
import 'package:http/http.dart' as http;
//import 'package:good_health/util/config.dart';

class Obat {
  final String idObat, nama, harga, satuan;
  bool isSelected;
  int jumlah;

  Obat(
      {required this.idObat,
      required this.nama,
      required this.harga,
      required this.satuan,
      this.isSelected = false,
      this.jumlah = 1});

  factory Obat.fromJson(Map<String, dynamic> json) {
    return Obat(
        idObat: json['id_obat'],
        nama: json['nama'],
        harga: json['harga'],
        satuan: json['satuan']);
  }
}

List<Obat> obatFromJson(jsonData) {
  List<Obat> result =
      List<Obat>.from(jsonData.map((item) => Obat.fromJson(item)));

  return result;
}

// index
Future<List<Obat>> fetchObats() async {
```

3.11 Pasien.dart



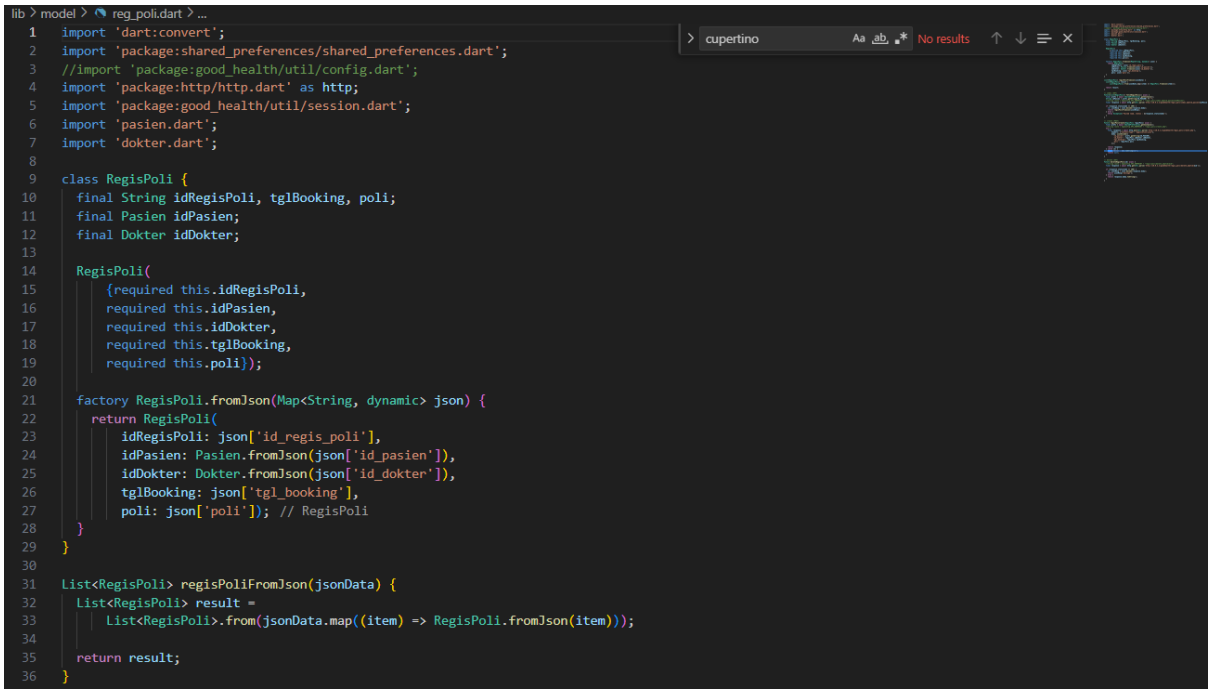
```dart
import 'dart:convert' ;

//import 'package:good_health/util/config.dart';
import 'package:http/http.dart' as http;

class Pasien {
  final String idPasien, nama, hp, email;

  Pasien({required this.idPasien, required this.nama, required this.hp, required this.email});

  factory Pasien.fromJson(Map<String, dynamic> json) {
    return Pasien(
        idPasien: json['id_pasien'],
        nama: json['nama'],
        hp: json['hp'],
        email: json['email']);
  }
}

List<Pasien> pasienFromJson(jsonData) {
  List<Pasien> result =
      List<Pasien>.from(jsonData.map((item) => Pasien.fromJson(item)));

  return result;
}

// register pasien (POST)
Future pasienCreate(Pasien pasien) async {
  //String route = AppConfig.API_ENDPOINT + "/pasien/create.php";
  try {
    final response = await http.post(Uri.parse('http://10.0.2.2/goodhealth/pasien/create.php'),
        headers: {"Content-Type": "application/json"},
        body: jsonEncode(
            {'nama': pasien.nama, 'hp': pasien.hp, 'email': pasien.email}));

    return response;
```

3.12 Pesan_obat.dart



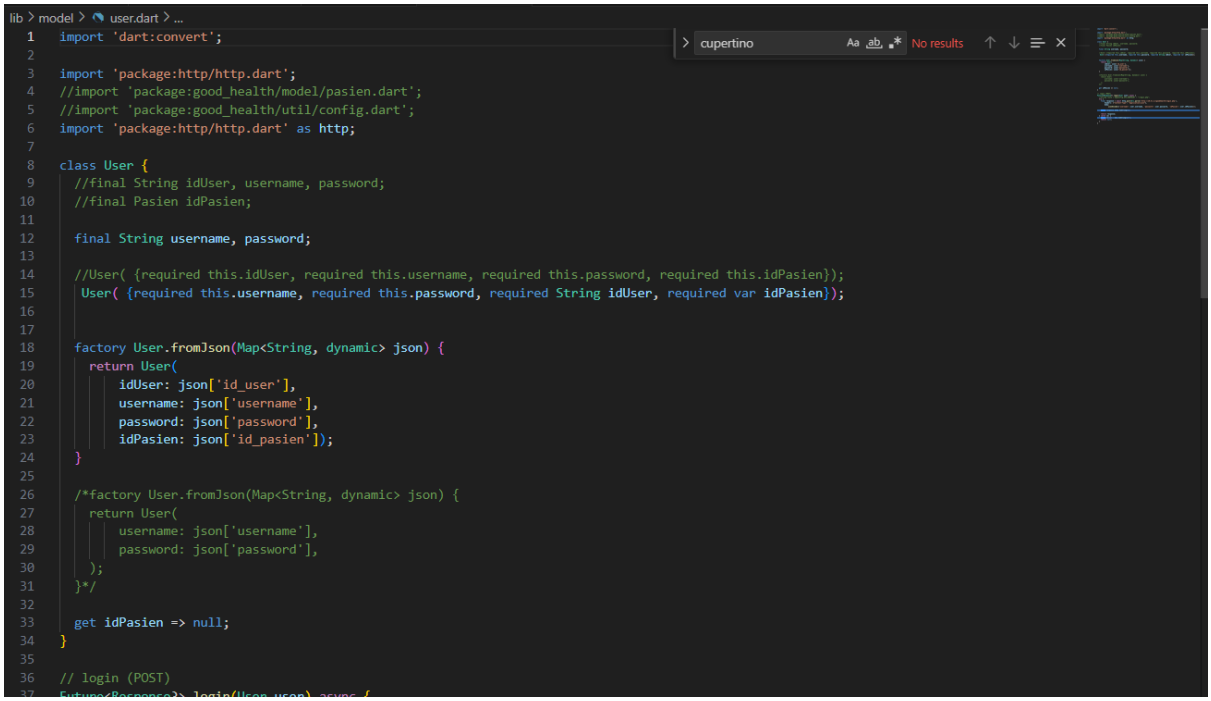
```dart
import 'dart:convert' ;
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';
//import 'package:good_health/util/config.dart';
import 'package:good_health/util/session.dart';
import 'pasien.dart';

class PesanObat {
  String idPesanObat,
      waktu,
      alamat,
      lat,
      lng,
      listPesanan,
      totalBiaya,
      ket;
  bool? isSelesai;

  Pasien? idPasien;

  PesanObat(
      {required this.idPesanObat,
      required this.idPasien,
      required this.waktu,
      required this.alamat,
      required this.lat,
      required this.lng,
      required this.listPesanan,
      required this.totalBiaya,
      required this.ket,
      this.isSelesai});

  factory PesanObat.fromJson(Map<String, dynamic> json) {
    return PesanObat(
        idPesanObat: json['id_pesan_obat'],
        idPasien: Pasien.fromJson(json['id_pasien']),
```

3.13 Reg_poli.dart



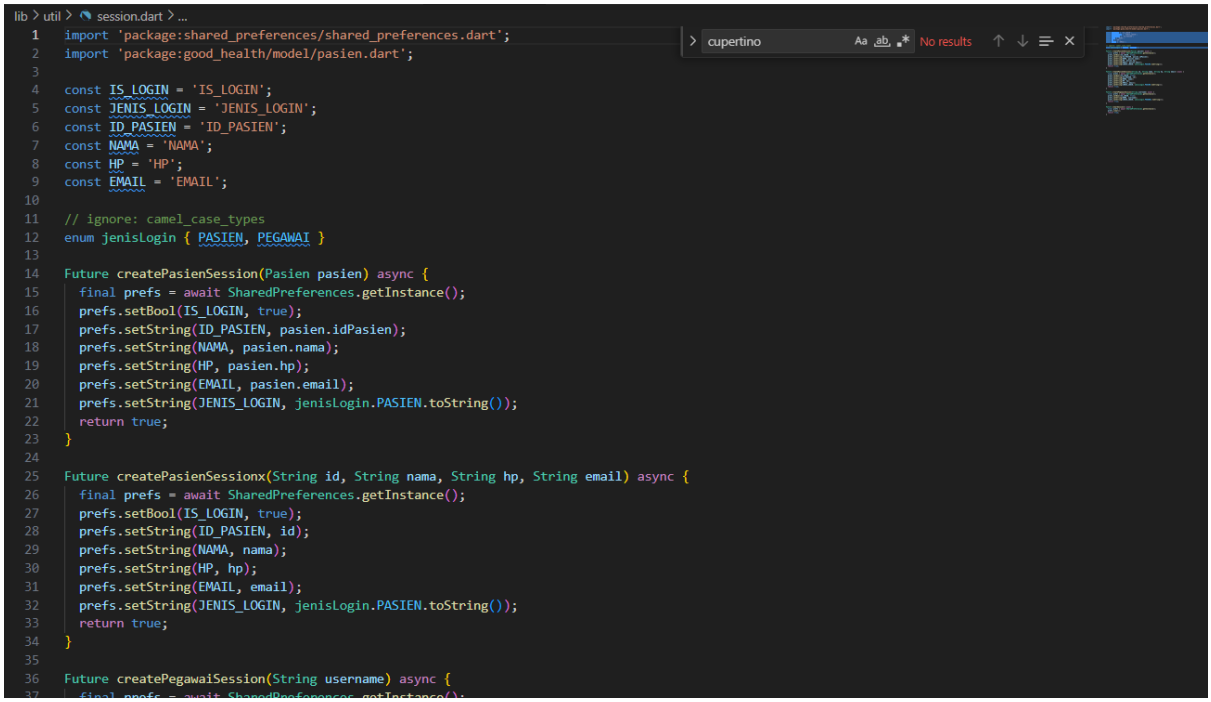
```dart
import 'dart:convert';
import 'package:shared_preferences/shared_preferences.dart';
//import 'package:good_health/util/config.dart';
import 'package:http/http.dart' as http;
import 'package:good_health/util/session.dart';
import 'pasien.dart';
import 'dokter.dart';

class RegisPoli {
  final String idRegisPoli, tglBooking, poli;
  final Pasien idPasien;
  final Dokter idDokter;

  RegisPoli(
      {required this.idRegisPoli,
      required this.idPasien,
      required this.idDokter,
      required this.tglBooking,
      required this.poli});

  factory RegisPoli.fromJson(Map<String, dynamic> json) {
    return RegisPoli(
        idRegisPoli: json['id_regis_poli'],
        idPasien: Pasien.fromJson(json['id_pasien']),
        idDokter: Dokter.fromJson(json['id_dokter']),
        tglBooking: json['tgl_booking'],
        poli: json['poli']); // RegisPoli
  }
}

List<RegisPoli> regisPoliFromJson(jsonData) {
  List<RegisPoli> result =
      List<RegisPoli>.from(jsonData.map((item) => RegisPoli.fromJson(item)));

  return result;
}
```

3.14 User.dart



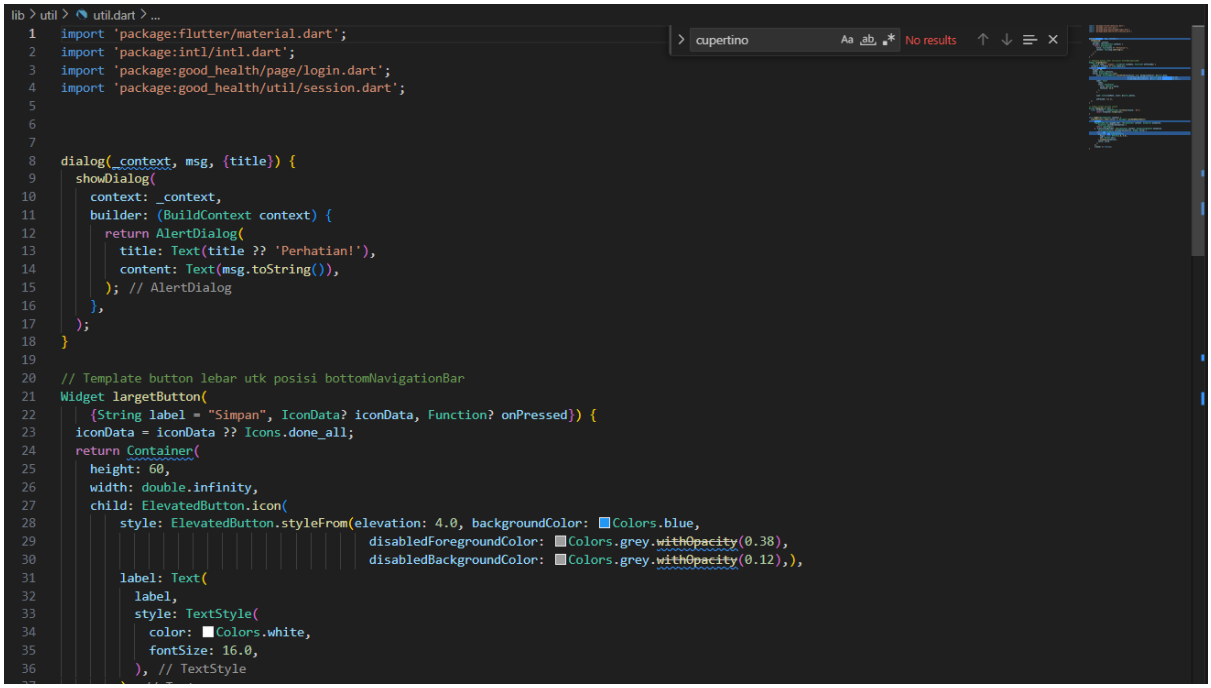
```
lib > model > user.dart > ...
1   import 'dart:convert';
2
3   import 'package:http/http.dart';
4   //import 'package:good_health/model/pasien.dart';
5   //import 'package:good_health/util/config.dart';
6   import 'package:http/http.dart' as http;
7
8   class User {
9     //final String idUser, username, password;
10    //final Pasien idPasien;
11
12    final String username, password;
13
14    //User( {required this.idUser, required this.username, required this.password, required this.idPasien});
15    User( {required this.username, required this.password, required String idUser, required var idPasien});
16
17
18    factory User.fromJson(Map<String, dynamic> json) {
19      return User(
20          idUser: json['id_user'],
21          username: json['username'],
22          password: json['password'],
23          idPasien: json['id_pasien']);
24    }
25
26    /*factory User.fromJson(Map<String, dynamic> json) {
27      return User(
28          username: json['username'],
29          password: json['password'],
30      );
31    }*/
32
33    get idPasien => null;
34  }
35
36  // login (POST)
```

3.15 Session.dart



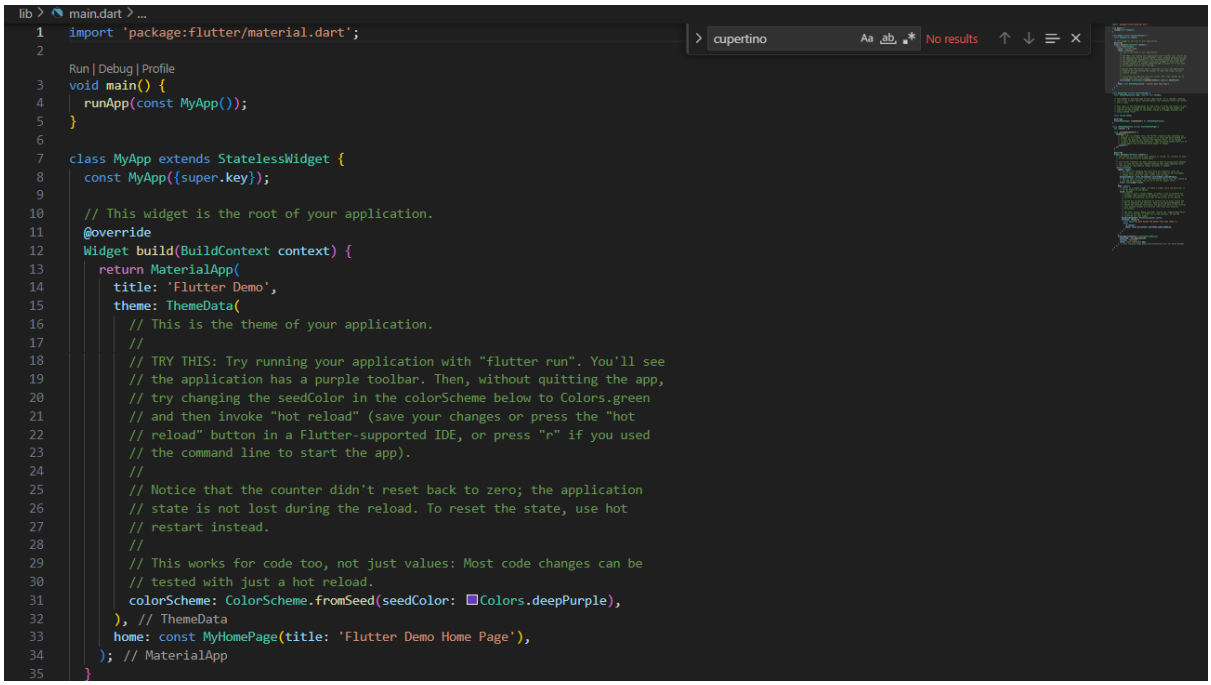
```
lib > util > session.dart > ...
1   import 'package:shared_preferences/shared_preferences.dart';
2   import 'package:good_health/model/pasien.dart';
3
4   const IS_LOGIN = 'IS_LOGIN';
5   const JENIS_LOGIN = 'JENIS_LOGIN';
6   const ID_PASIEN = 'ID_PASIEN';
7   const NAMA = 'NAMA';
8   const HP = 'HP';
9   const EMAIL = 'EMAIL';
10
11  // ignore: camel_case_types
12  enum jenisLogin { PASIEN, PEGAWAI }
13
14  Future createPasienSession(Pasien pasien) async {
15    final prefs = await SharedPreferences.getInstance();
16    prefs.setBool(IS_LOGIN, true);
17    prefs.setString(ID_PASIEN, pasien.idPasien);
18    prefs.setString(NAMA, pasien.nama);
19    prefs.setString(HP, pasien.hp);
20    prefs.setString(EMAIL, pasien.email);
21    prefs.setString(JENIS_LOGIN, jenisLogin.PASIEN.toString());
22    return true;
23  }
24
25  Future createPasienSessionx(String id, String nama, String hp, String email) async {
26    final prefs = await SharedPreferences.getInstance();
27    prefs.setBool(IS_LOGIN, true);
28    prefs.setString(ID_PASIEN, id);
29    prefs.setString(NAMA, nama);
30    prefs.setString(HP, hp);
31    prefs.setString(EMAIL, email);
32    prefs.setString(JENIS_LOGIN, jenisLogin.PASIEN.toString());
33    return true;
34  }
35
36  Future createPegawaiSession(String username) async {
```

3.16 Util.dart



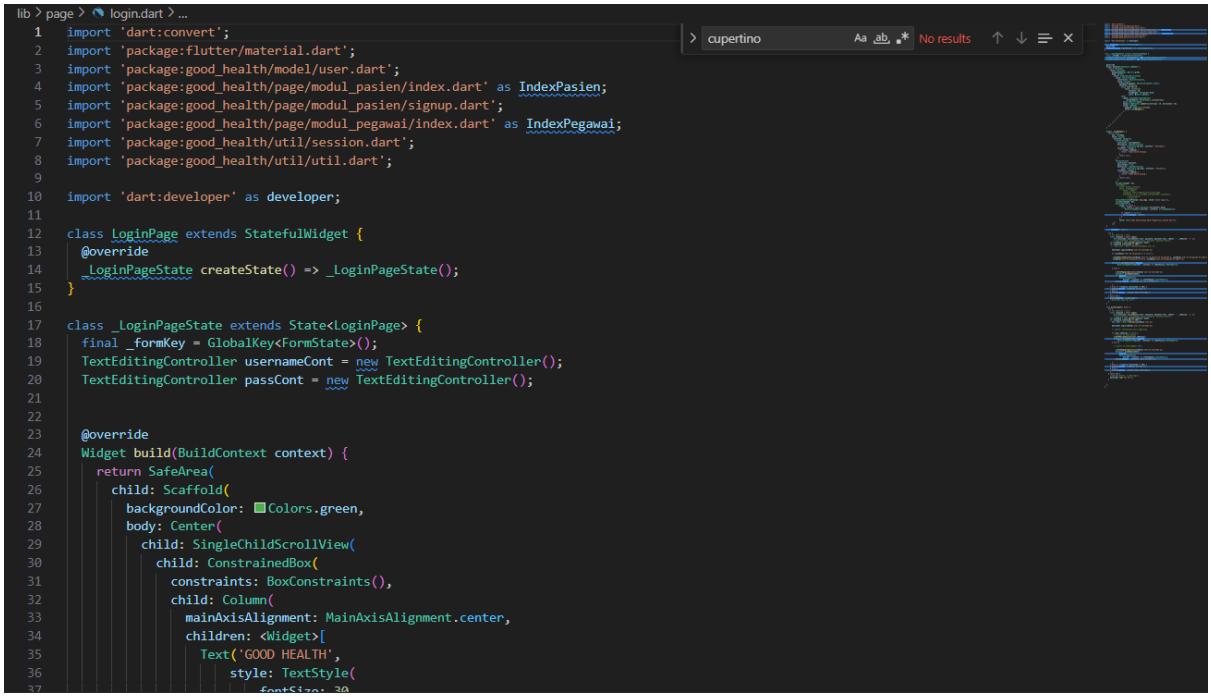
```dart
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'package:good_health/page/login.dart';
import 'package:good_health/util/session.dart';



dialog(_context, msg, {title}) {
  showDialog(
    context: _context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text(title ?? 'Perhatian!'),
        content: Text(msg.toString()),
      ); // AlertDialog
    },
  );
}

// Template button lebar utk posisi bottomNavigationBar
Widget largetButton(
    {String label = "Simpan", IconData? iconData, Function? onPressed}) {
  iconData = iconData ?? Icons.done_all;
  return Container(
    height: 60,
    width: double.infinity,
    child: ElevatedButton.icon(
        style: ElevatedButton.styleFrom(elevation: 4.0, backgroundColor: Colors.blue,
                                        disabledForegroundColor: Colors.grey.withOpacity(0.38),
                                        disabledBackgroundColor: Colors.grey.withOpacity(0.12),),
        label: Text(
          label,
          style: TextStyle(
            color: Colors.white,
            fontSize: 16.0,
          ), // TextStyle
```

3.17 Main.dart



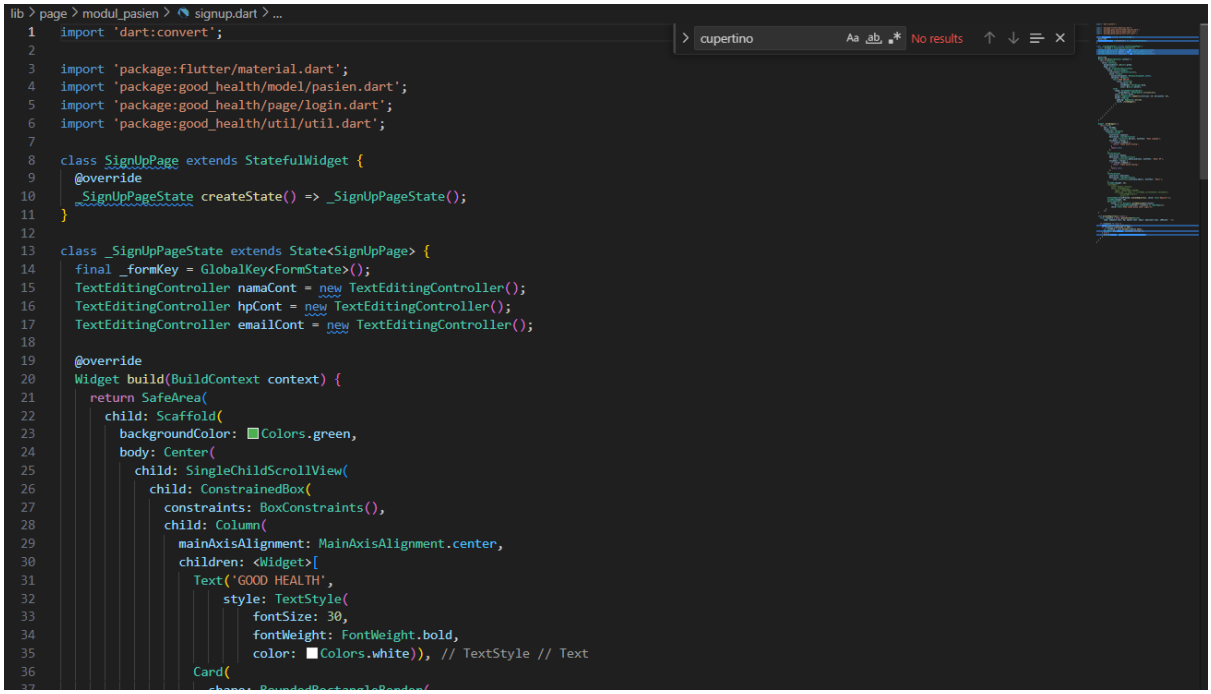
```dart
import 'package:flutter/material.dart';

Run | Debug | Profile
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),
      ), // ThemeData
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    ); // MaterialApp
  }
```

3.18 Login.dart



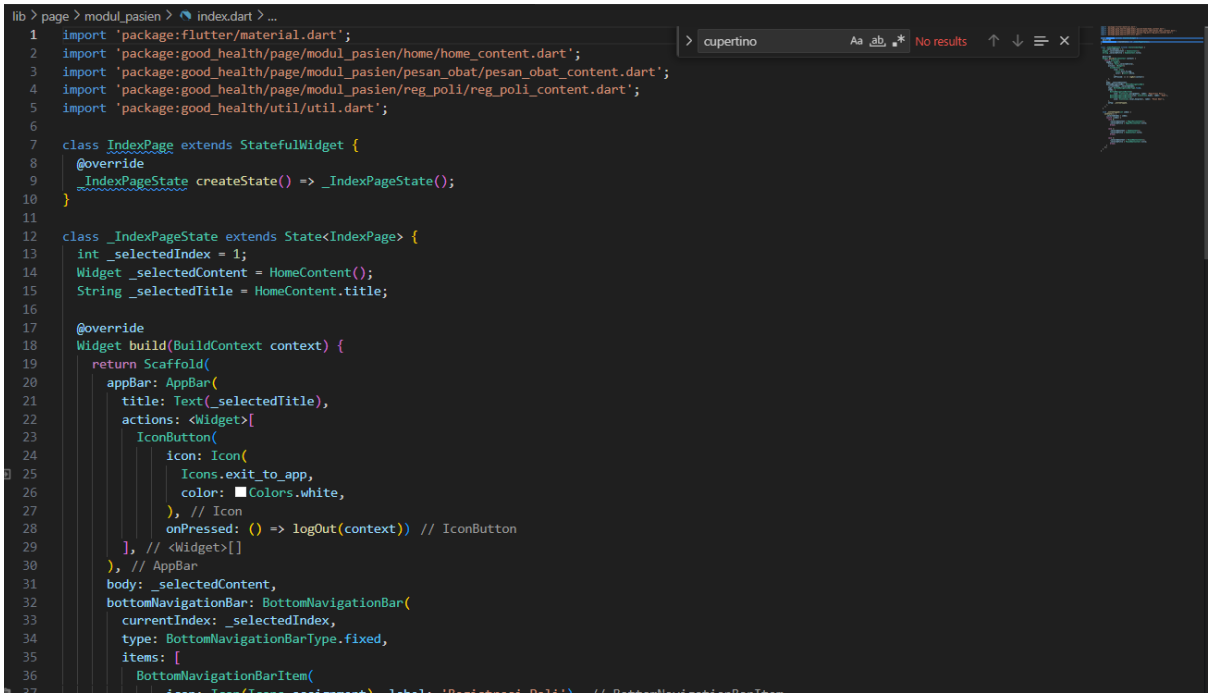
```dart
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:good_health/model/user.dart';
import 'package:good_health/page/modul_pasien/index.dart' as IndexPasien;
import 'package:good_health/page/modul_pasien/signup.dart';
import 'package:good_health/page/modul_pegawai/index.dart' as IndexPegawai;
import 'package:good_health/util/session.dart';
import 'package:good_health/util/util.dart';

import 'dart:developer' as developer;

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final _formKey = GlobalKey<FormState>();
  TextEditingController usernameCont = new TextEditingController();
  TextEditingController passCont = new TextEditingController();


  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        backgroundColor: Colors.green,
        body: Center(
          child: SingleChildScrollView(
            child: ConstrainedBox(
              constraints: BoxConstraints(),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                  Text('GOOD HEALTH',
                      style: TextStyle(
```

3.19 Signup.dart



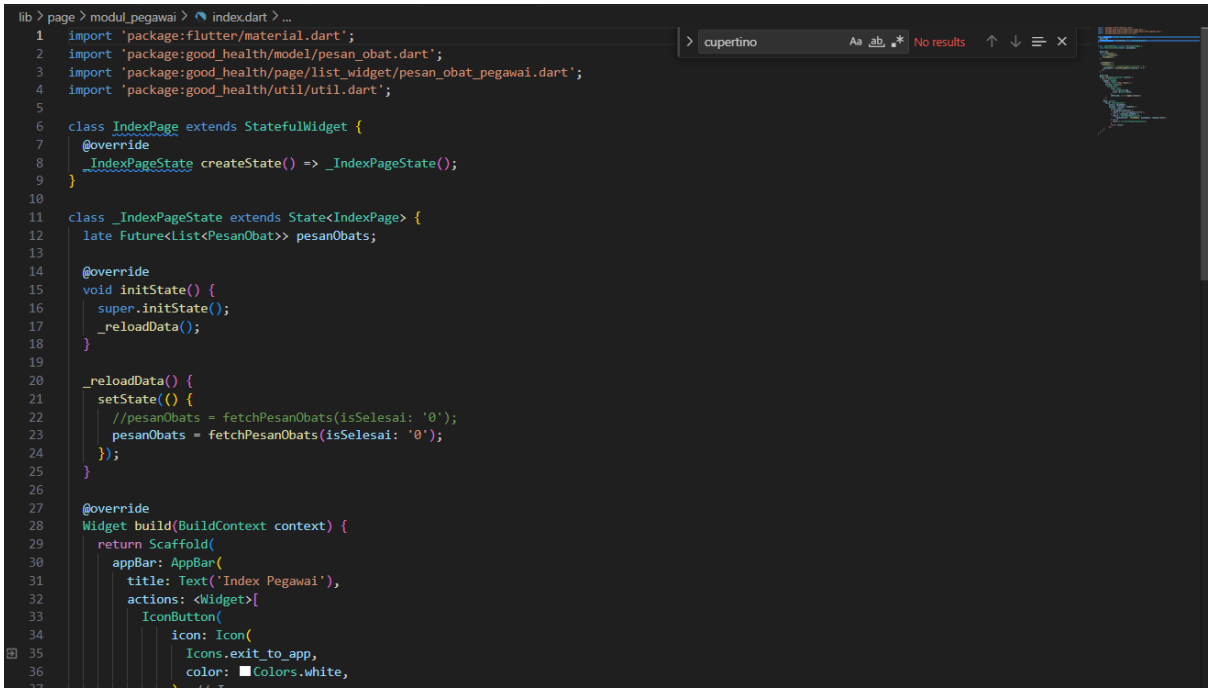
```dart
import 'dart:convert';

import 'package:flutter/material.dart';
import 'package:good_health/model/pasien.dart';
import 'package:good_health/page/login.dart';
import 'package:good_health/util/util.dart';

class SignUpPage extends StatefulWidget {
  @override
  _SignUpPageState createState() => _SignUpPageState();
}

class _SignUpPageState extends State<SignUpPage> {
  final _formKey = GlobalKey<FormState>();
  TextEditingController namaCont = new TextEditingController();
  TextEditingController hpCont = new TextEditingController();
  TextEditingController emailCont = new TextEditingController();

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        backgroundColor: Colors.green,
        body: Center(
          child: SingleChildScrollView(
            child: ConstrainedBox(
              constraints: BoxConstraints(),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                  Text('GOOD HEALTH',
                      style: TextStyle(
                          fontSize: 30,
                          fontWeight: FontWeight.bold,
                          color: Colors.white)), // TextStyle // Text
                  Card(
```

3.20 Index.dart dalam modul_pasien



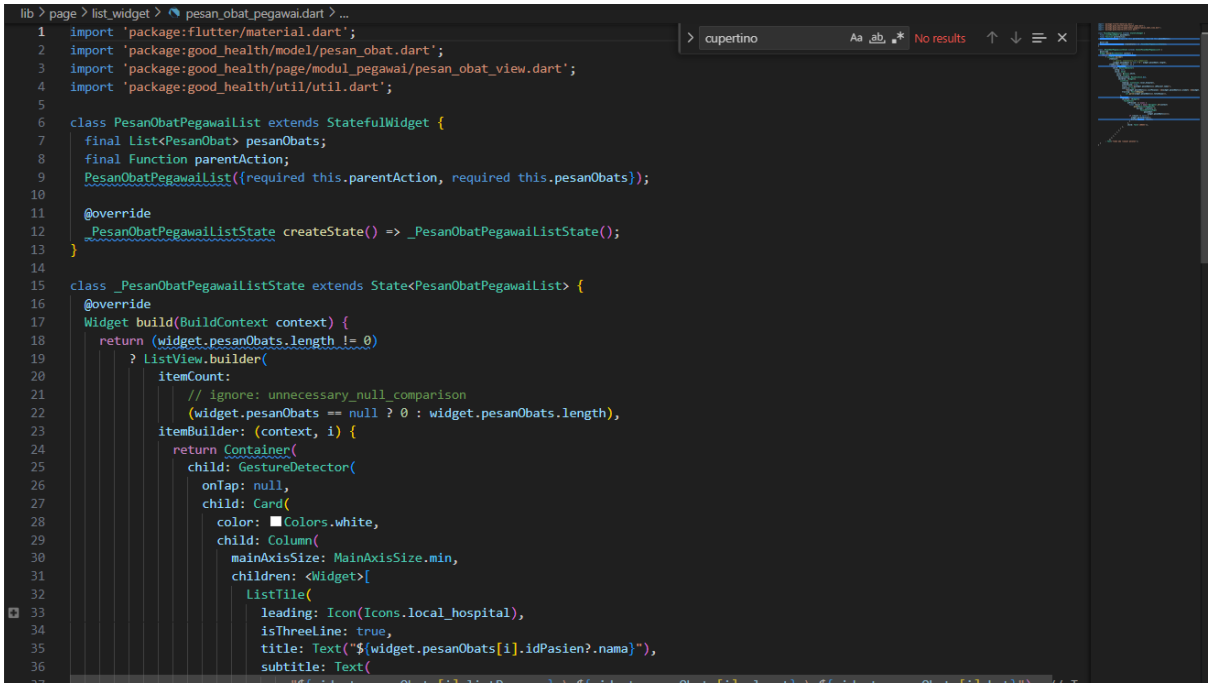
```dart
import 'package:flutter/material.dart';
import 'package:good_health/page/modul_pasien/home/home_content.dart';
import 'package:good_health/page/modul_pasien/pesan_obat/pesan_obat_content.dart';
import 'package:good_health/page/modul_pasien/reg_poli/reg_poli_content.dart';
import 'package:good_health/util/util.dart';

class IndexPage extends StatefulWidget {
  @override
  _IndexPageState createState() => _IndexPageState();
}

class _IndexPageState extends State<IndexPage> {
  int _selectedIndex = 1;
  Widget _selectedContent = HomeContent();
  String _selectedTitle = HomeContent.title;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(_selectedTitle),
        actions: <Widget>[
          IconButton(
              icon: Icon(
                Icons.exit_to_app,
                color: Colors.white,
              ), // Icon
              onPressed: () => logOut(context)) // IconButton
        ], // <Widget>[]
      ), // AppBar
      body: _selectedContent,
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _selectedIndex,
        type: BottomNavigationBarType.fixed,
        items: [
          BottomNavigationBarItem(
```

3.21 Index.dart modul_pegawai



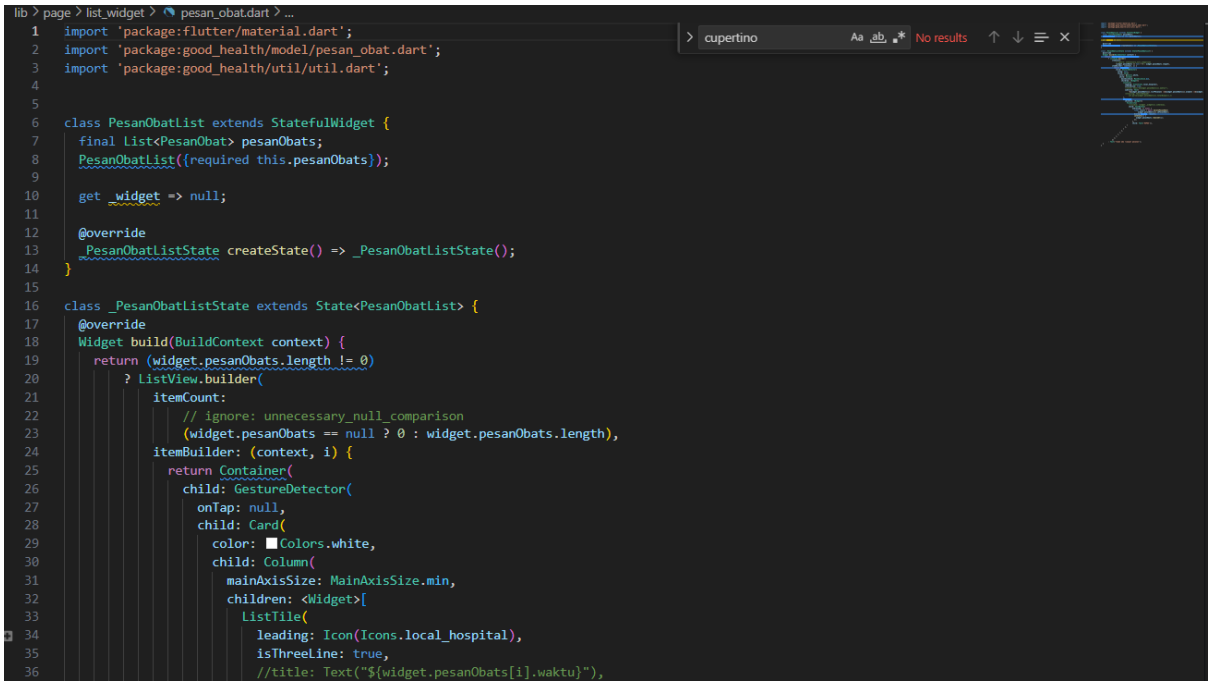
```dart
import 'package:flutter/material.dart';
import 'package:good_health/model/pesan_obat.dart';
import 'package:good_health/page/list_widget/pesan_obat_pegawai.dart';
import 'package:good_health/util/util.dart';

class IndexPage extends StatefulWidget {
  @override
  _IndexPageState createState() => _IndexPageState();
}

class _IndexPageState extends State<IndexPage> {
  late Future<List<PesanObat>> pesanObats;

  @override
  void initState() {
    super.initState();
    _reloadData();
  }

  _reloadData() {
    setState(() {
      //pesanObats = fetchPesanObats(isSelesai: '0');
      pesanObats = fetchPesanObats(isSelesai: '0');
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Index Pegawai'),
        actions: <Widget>[
          IconButton(
              icon: Icon(
                Icons.exit_to_app,
                color: Colors.white,
```

3.22 Pesan_obat_pegawai.dart



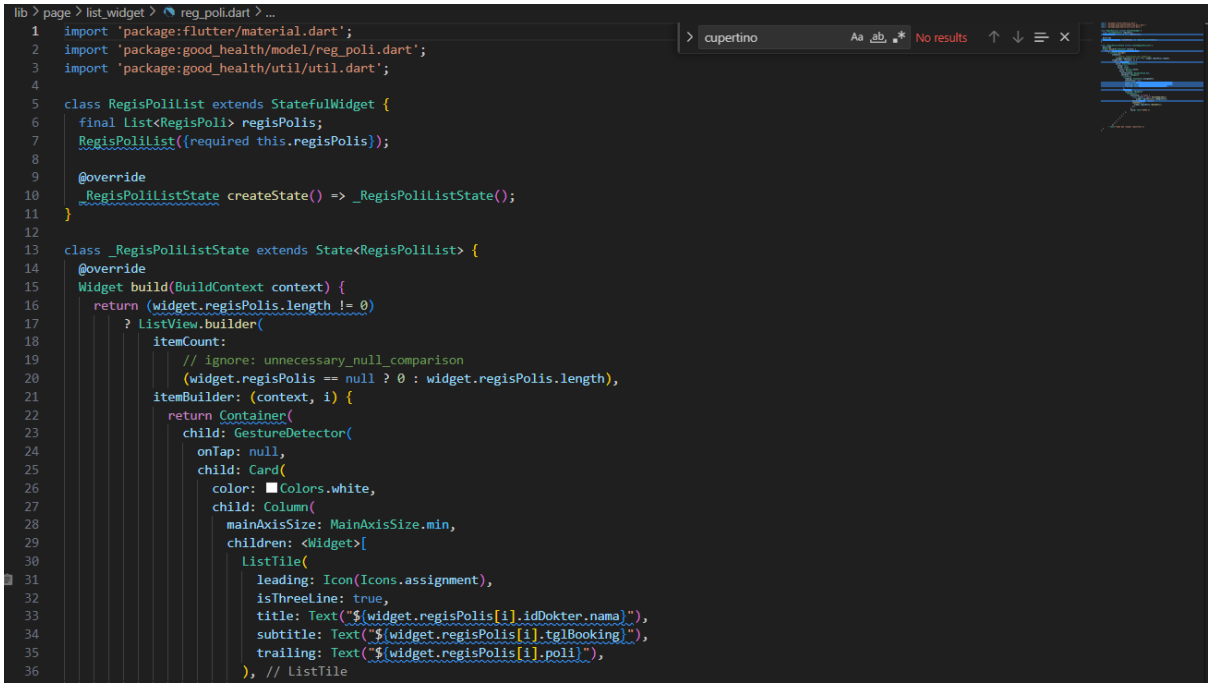
```dart
import 'package:flutter/material.dart';
import 'package:good_health/model/pesan_obat.dart';
import 'package:good_health/page/modul_pegawai/pesan_obat_view.dart';
import 'package:good_health/util/util.dart';

class PesanObatPegawaiList extends StatefulWidget {
  final List<PesanObat> pesanObats;
  final Function parentAction;
  PesanObatPegawaiList({required this.parentAction, required this.pesanObats});

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

class _PesanObatPegawaiListState extends State<PesanObatPegawaiList> {
  @override
  Widget build(BuildContext context) {
    return (widget.pesanObats.length != 0)
        ? ListView.builder(
            itemCount:
                // ignore: unnecessary_null_comparison
                (widget.pesanObats == null ? 0 : widget.pesanObats.length),
            itemBuilder: (context, i) {
              return Container(
                child: GestureDetector(
                  onTap: null,
                  child: Card(
                    color: Colors.white,
                    child: Column(
                      mainAxisSize: MainAxisSize.min,
                      children: <Widget>[
                        ListTile(
                          leading: Icon(Icons.local_hospital),
                          isThreeLine: true,
                          title: Text("${widget.pesanObats[i].idPasien?.nama}"),
                          subtitle: Text(
```

3.23 pesan_obat.dart



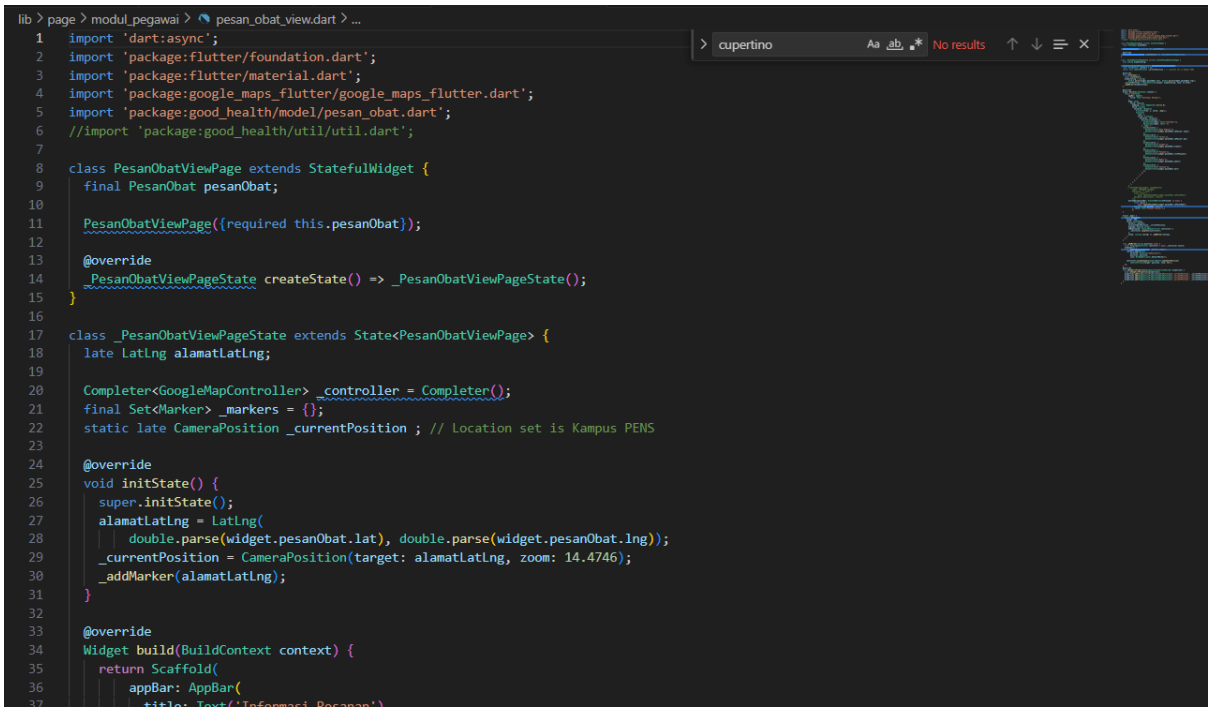
```dart
import 'package:flutter/material.dart';
import 'package:good_health/model/pesan_obat.dart';
import 'package:good_health/util/util.dart';


class PesanObatList extends StatefulWidget {
  final List<PesanObat> pesanObats;
  PesanObatList({required this.pesanObats});

  get _widget => null;

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

class _PesanObatListState extends State<PesanObatList> {
  @override
  Widget build(BuildContext context) {
    return (widget.pesanObats.length != 0)
        ? ListView.builder(
            itemCount:
                // ignore: unnecessary_null_comparison
                (widget.pesanObats == null ? 0 : widget.pesanObats.length),
            itemBuilder: (context, i) {
              return Container(
                child: GestureDetector(
                  onTap: null,
                  child: Card(
                    color: Colors.white,
                    child: Column(
                      mainAxisSize: MainAxisSize.min,
                      children: <Widget>[
                        ListTile(
                          leading: Icon(Icons.local_hospital),
                          isThreeLine: true,
                          //title: Text("${widget.pesanObats[i].waktu}"),
```

3.24 reg_poli.dart



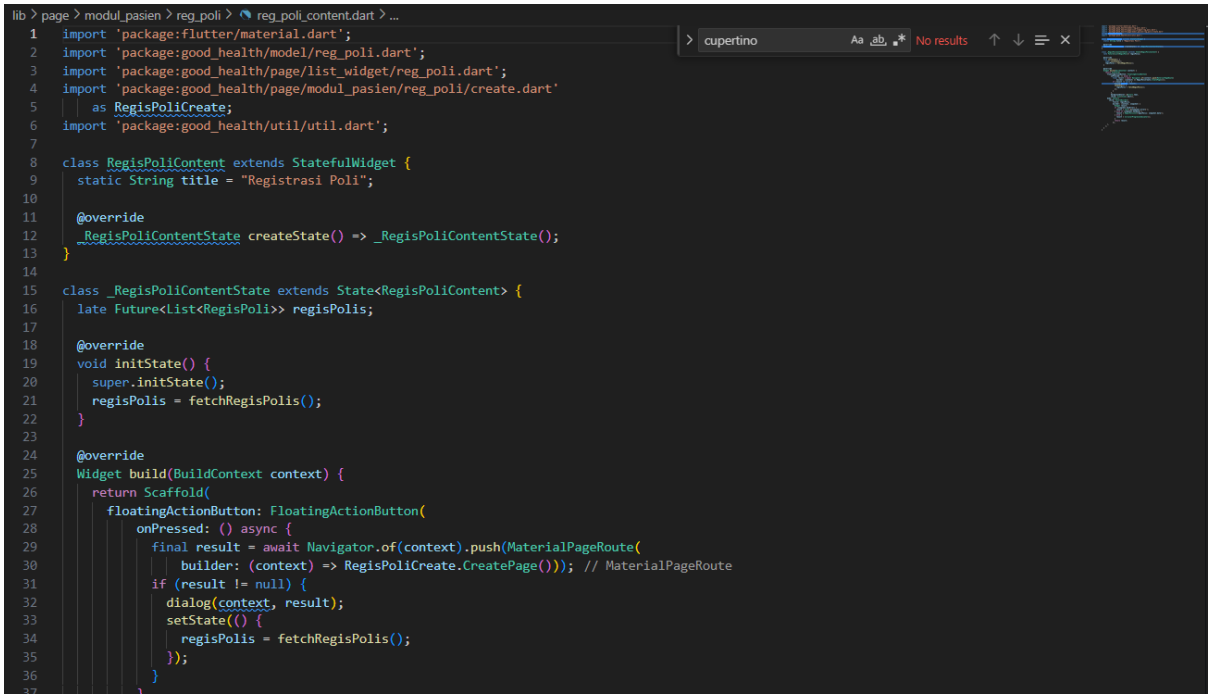
```dart
import 'package:flutter/material.dart';
import 'package:good_health/model/reg_poli.dart';
import 'package:good_health/util/util.dart';

class RegisPoliList extends StatefulWidget {
  final List<RegisPoli> regisPolis;
  RegisPoliList({required this.regisPolis});

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

class _RegisPoliListState extends State<RegisPoliList> {
  @override
  Widget build(BuildContext context) {
    return (widget.regisPolis.length != 0)
        ? ListView.builder(
            itemCount:
                // ignore: unnecessary_null_comparison
                (widget.regisPolis == null ? 0 : widget.regisPolis.length),
            itemBuilder: (context, i) {
              return Container(
                child: GestureDetector(
                  onTap: null,
                  child: Card(
                    color: Colors.white,
                    child: Column(
                      mainAxisSize: MainAxisSize.min,
                      children: <Widget>[
                        ListTile(
                          leading: Icon(Icons.assignment),
                          isThreeLine: true,
                          title: Text("${widget.regisPolis[i].idDokter.nama}"),
                          subtitle: Text("${widget.regisPolis[i].tglBooking}"),
                          trailing: Text("${widget.regisPolis[i].poli}"),
                        ), // ListTile
```

3.25 pesan_obat_view.dart



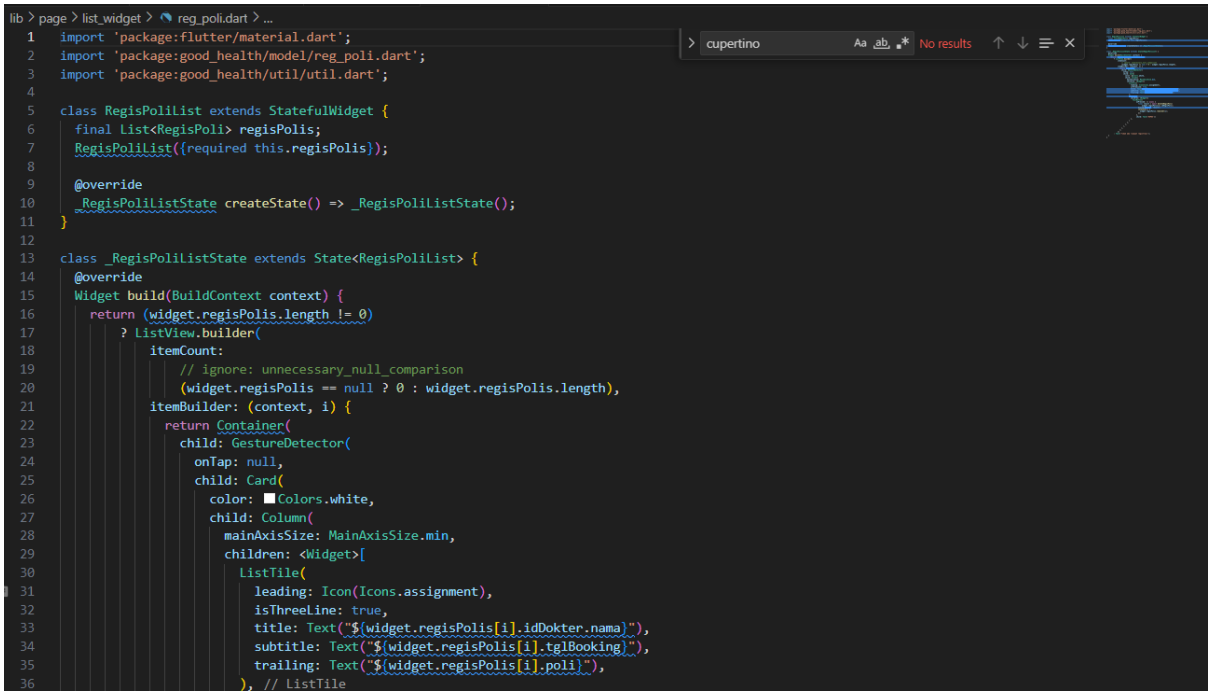
```dart
import 'dart:async';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:good_health/model/pesan_obat.dart';
//import 'package:good_health/util/util.dart';

class PesanObatViewPage extends StatefulWidget {
  final PesanObat pesanObat;

  PesanObatViewPage({required this.pesanObat});

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

class _PesanObatViewPageState extends State<PesanObatViewPage> {
  late LatLng alamatLatLng;

  Completer<GoogleMapController> _controller = Completer();
  final Set<Marker> _markers = {};
  static late CameraPosition _currentPosition ; // Location set is Kampus PENS

  @override
  void initState() {
    super.initState();
    alamatLatLng = LatLng(
        double.parse(widget.pesanObat.lat), double.parse(widget.pesanObat.lng));
    _currentPosition = CameraPosition(target: alamatLatLng, zoom: 14.4746);
    _addMarker(alamatLatLng);
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('Informasi Pesanan')
```

3.26 reg_poli_content.dart



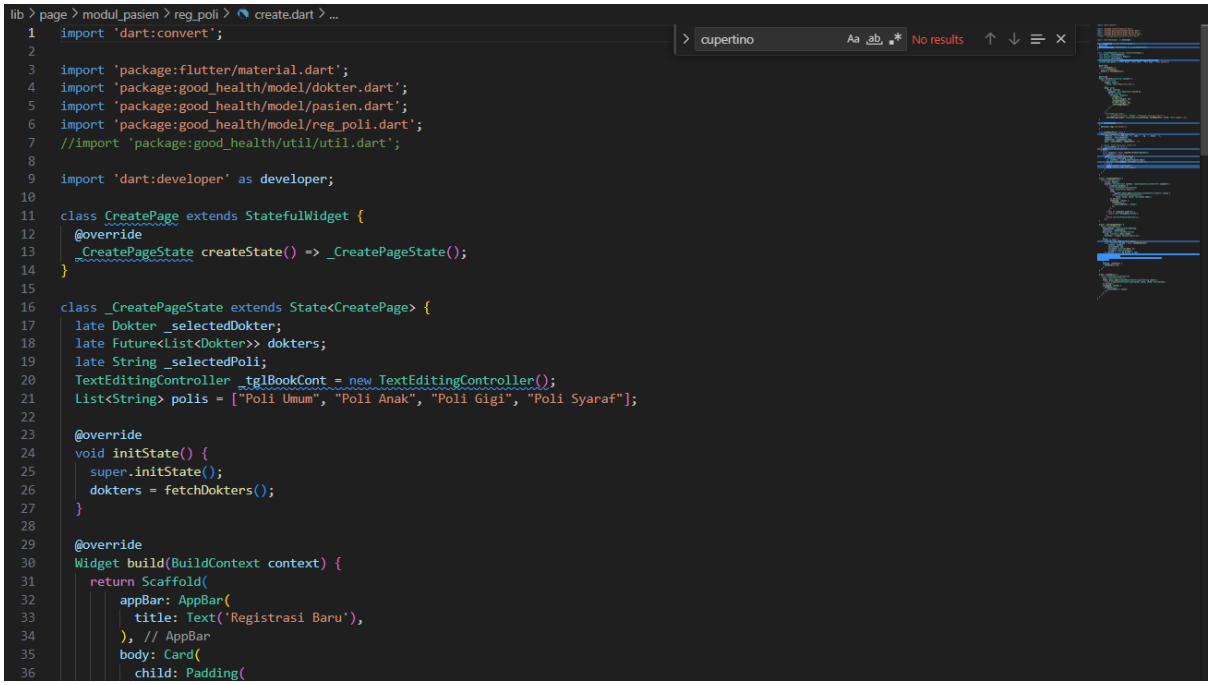
```dart
import 'package:flutter/material.dart';
import 'package:good_health/model/reg_poli.dart';
import 'package:good_health/page/list_widget/reg_poli.dart';
import 'package:good_health/page/modul_pasien/reg_poli/create.dart'
    as RegisPoliCreate;
import 'package:good_health/util/util.dart';

class RegisPoliContent extends StatefulWidget {
  static String title = "Registrasi Poli";

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

class _RegisPoliContentState extends State<RegisPoliContent> {
  late Future<List<RegisPoli>> regisPolis;

  @override
  void initState() {
    super.initState();
    regisPolis = fetchRegisPolis();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: FloatingActionButton(
          onPressed: () async {
            final result = await Navigator.of(context).push(MaterialPageRoute(
                builder: (context) => RegisPoliCreate.CreatePage())); // MaterialPageRoute
            if (result != null) {
              dialog(context, result);
              setState(() {
                regisPolis = fetchRegisPolis();
              });
            }
```

3.27 reg_poli.dart dalam folder list_widget



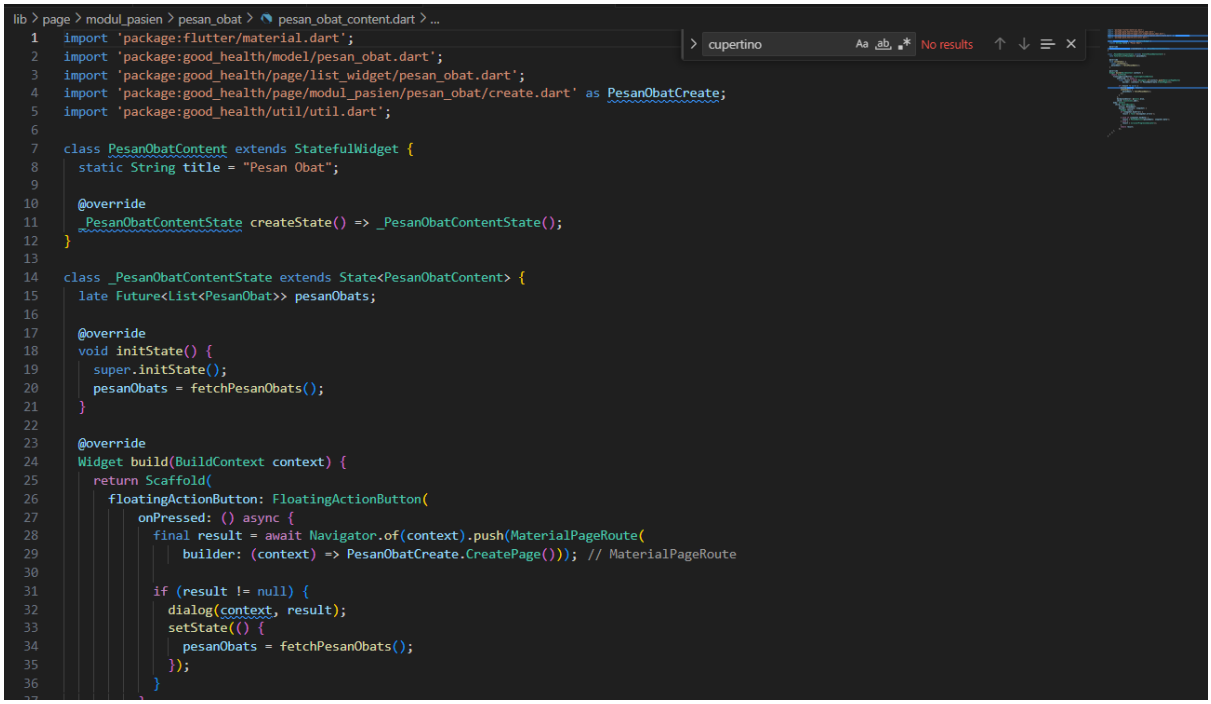
```dart
import 'package:flutter/material.dart';
import 'package:good_health/model/reg_poli.dart';
import 'package:good_health/util/util.dart';

class RegisPoliList extends StatefulWidget {
  final List<RegisPoli> regisPolis;
  RegisPoliList({required this.regisPolis});

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

class _RegisPoliListState extends State<RegisPoliList> {
  @override
  Widget build(BuildContext context) {
    return (widget.regisPolis.length != 0)
        ? ListView.builder(
            itemCount:
                // ignore: unnecessary_null_comparison
                (widget.regisPolis == null ? 0 : widget.regisPolis.length),
            itemBuilder: (context, i) {
              return Container(
                child: GestureDetector(
                  onTap: null,
                  child: Card(
                    color: Colors.white,
                    child: Column(
                      mainAxisSize: MainAxisSize.min,
                      children: <Widget>[
                        ListTile(
                          leading: Icon(Icons.assignment),
                          isThreeLine: true,
                          title: Text("${widget.regisPolis[i].idDokter.nama}"),
                          subtitle: Text("${widget.regisPolis[i].tglBooking}"),
                          trailing: Text("${widget.regisPolis[i].poli}"),
                        ), // ListTile
```

3.28 create.dart dalam folder reg_poli



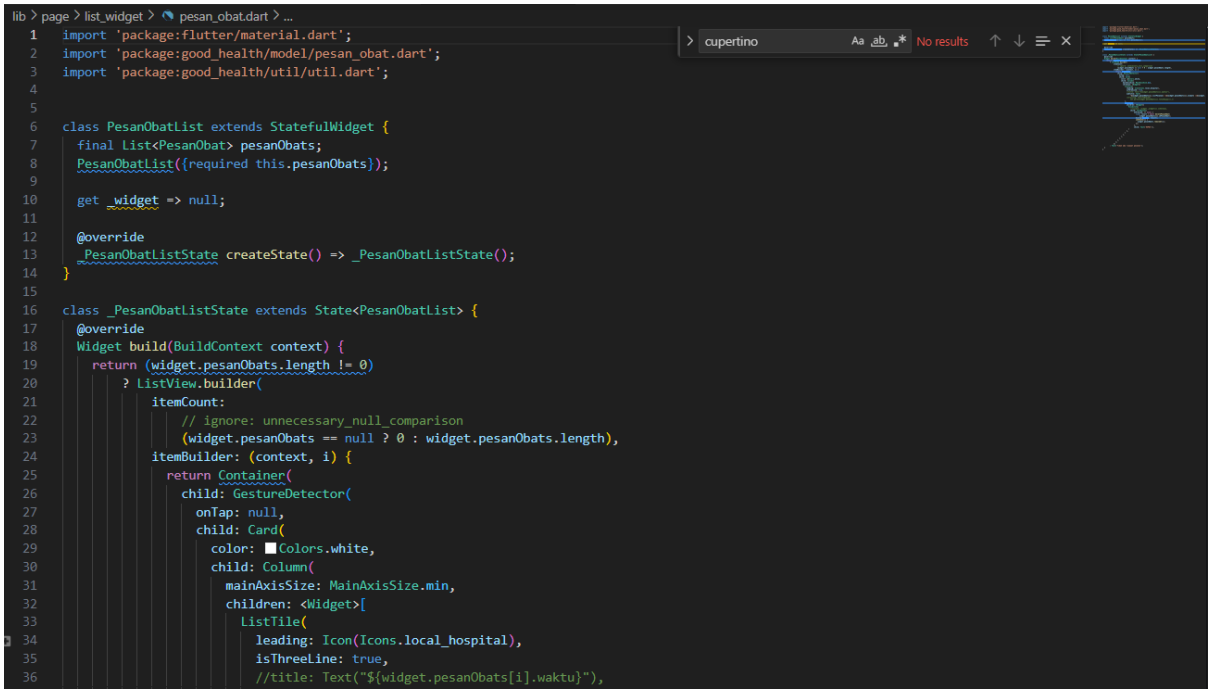
```
lib > page > modul_pasien > reg_poli > create.dart > ...
1  import 'dart:convert';
2
3  import 'package:flutter/material.dart';
4  import 'package:good_health/model/dokter.dart';
5  import 'package:good_health/model/pasien.dart';
6  import 'package:good_health/model/reg_poli.dart';
7  //import 'package:good_health/util/util.dart';
8
9  import 'dart:developer' as developer;
10
11 class CreatePage extends StatefulWidget {
12   @override
13   _CreatePageState createState() => _CreatePageState();
14 }
15
16 class _CreatePageState extends State<CreatePage> {
17   late Dokter _selectedDokter;
18   late Future<List<Dokter>> dokters;
19   late String _selectedPoli;
20   TextEditingController _tglBookCont = new TextEditingController();
21   List<String> polis = ["Poli Umum", "Poli Anak", "Poli Gigi", "Poli Syaraf"];
22
23   @override
24   void initState() {
25     super.initState();
26     dokters = fetchDokters();
27   }
28
29   @override
30   Widget build(BuildContext context) {
31     return Scaffold(
32         appBar: AppBar(
33           title: Text('Registrasi Baru'),
34         ), // AppBar
35         body: Card(
36           child: Padding(
```

3.29 pesan_obat_content.dart dalam folder pesan_obat



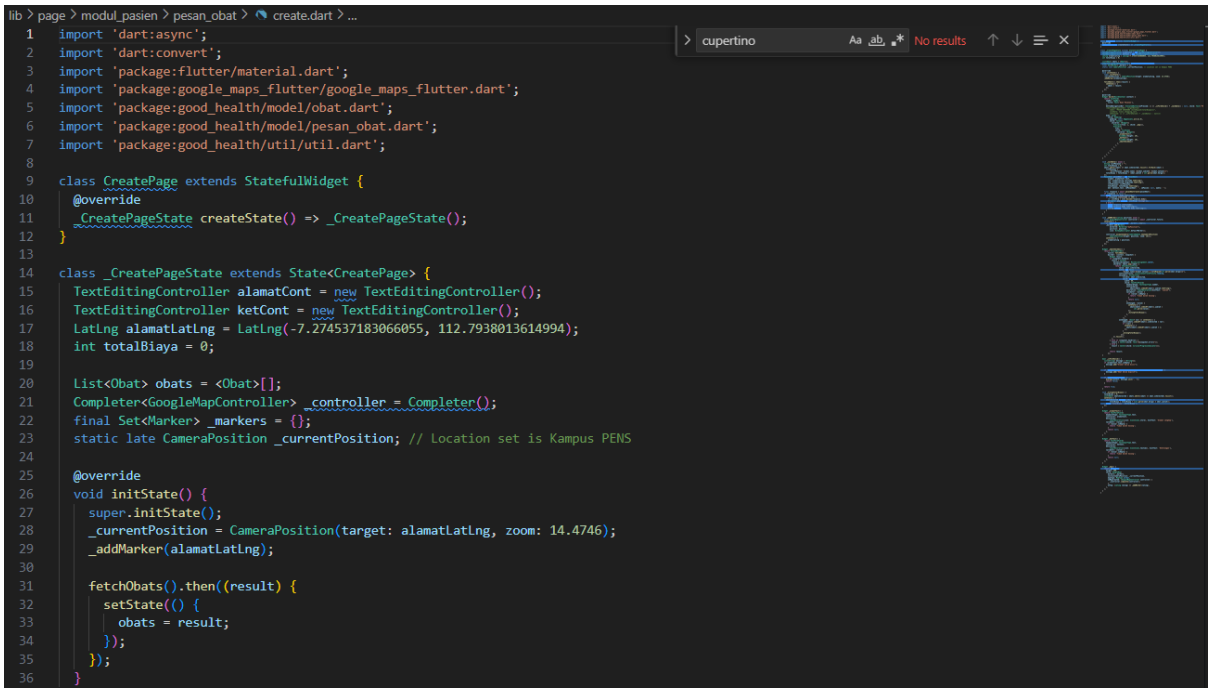
```
lib > page > modul_pasien > pesan_obat > pesan_obat_content.dart > ...
1  import 'package:flutter/material.dart';
2  import 'package:good_health/model/pesan_obat.dart';
3  import 'package:good_health/page/list_widget/pesan_obat.dart';
4  import 'package:good_health/page/modul_pasien/pesan_obat/create.dart' as PesanObatCreate;
5  import 'package:good_health/util/util.dart';
6
7  class PesanObatContent extends StatefulWidget {
8    static String title = "Pesan Obat";
9
10   @override
11   _PesanObatContentState createState() => _PesanObatContentState();
12 }
13
14 class _PesanObatContentState extends State<PesanObatContent> {
15   late Future<List<PesanObat>> pesanObats;
16
17   @override
18   void initState() {
19     super.initState();
20     pesanObats = fetchPesanObats();
21   }
22
23   @override
24   Widget build(BuildContext context) {
25     return Scaffold(
26       floatingActionButton: FloatingActionButton(
27           onPressed: () async {
28             final result = await Navigator.of(context).push(MaterialPageRoute(
29                 builder: (context) => PesanObatCreate.CreatePage())); // MaterialPageRoute
30
31             if (result != null) {
32               dialog(context, result);
33               setState(() {
34                 pesanObats = fetchPesanObats();
35               });
36             }
```

3.30 pesan_obat.dart di dalam folder list_widget



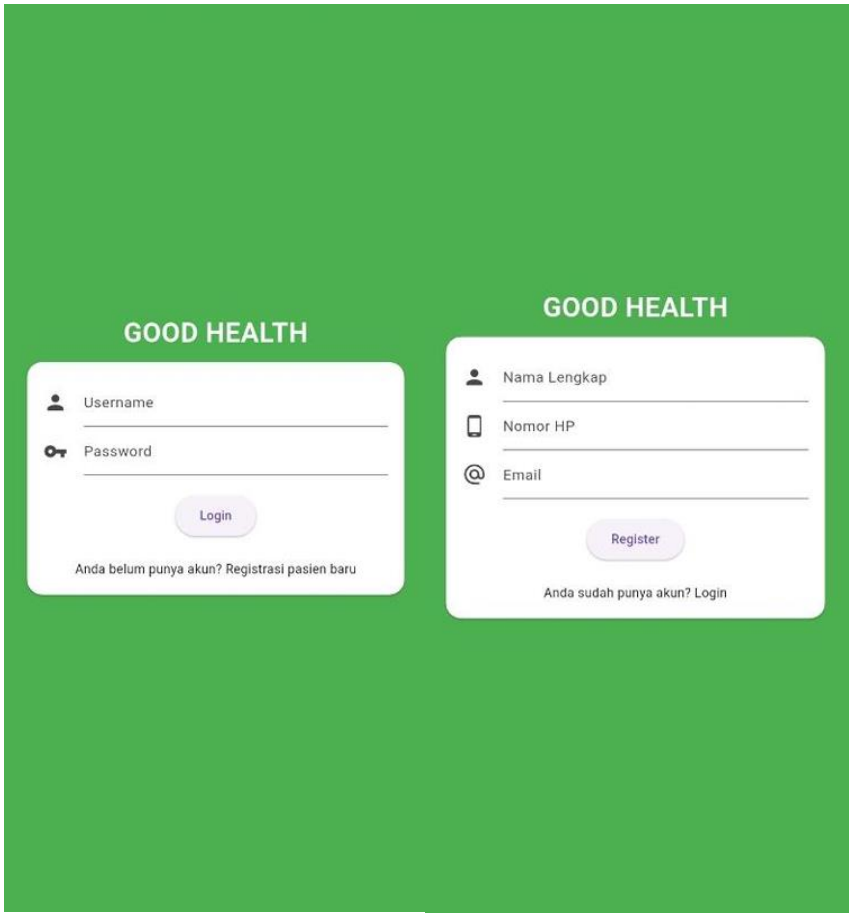
```
lib > page > list_widget > pesan_obat.dart > ...
1  import 'package:flutter/material.dart';
2  import 'package:good_health/model/pesan_obat.dart';
3  import 'package:good_health/util/util.dart';
4
5
6  class PesanObatList extends StatefulWidget {
7    final List<PesanObat> pesanObats;
8    PesanObatList({required this.pesanObats});
9
10   get _widget => null;
11
12   @override
13   _PesanObatListState createState() => _PesanObatListState();
14 }
15
16 class _PesanObatListState extends State<PesanObatList> {
17   @override
18   Widget build(BuildContext context) {
19     return (widget.pesanObats.length != 0)
20         ? ListView.builder(
21             itemCount:
22                 // ignore: unnecessary_null_comparison
23                 (widget.pesanObats == null ? 0 : widget.pesanObats.length),
24             itemBuilder: (context, i) {
25               return Container(
26                 child: GestureDetector(
27                   onTap: null,
28                   child: Card(
29                     color: Colors.white,
30                     child: Column(
31                       mainAxisSize: MainAxisSize.min,
32                       children: <Widget>[
33                         ListTile(
34                           leading: Icon(Icons.local_hospital),
35                           isThreeLine: true,
36                           //title: Text("${widget.pesanObats[i].waktu}"),
```

3.31 create.dart dalam folder pesan_obat



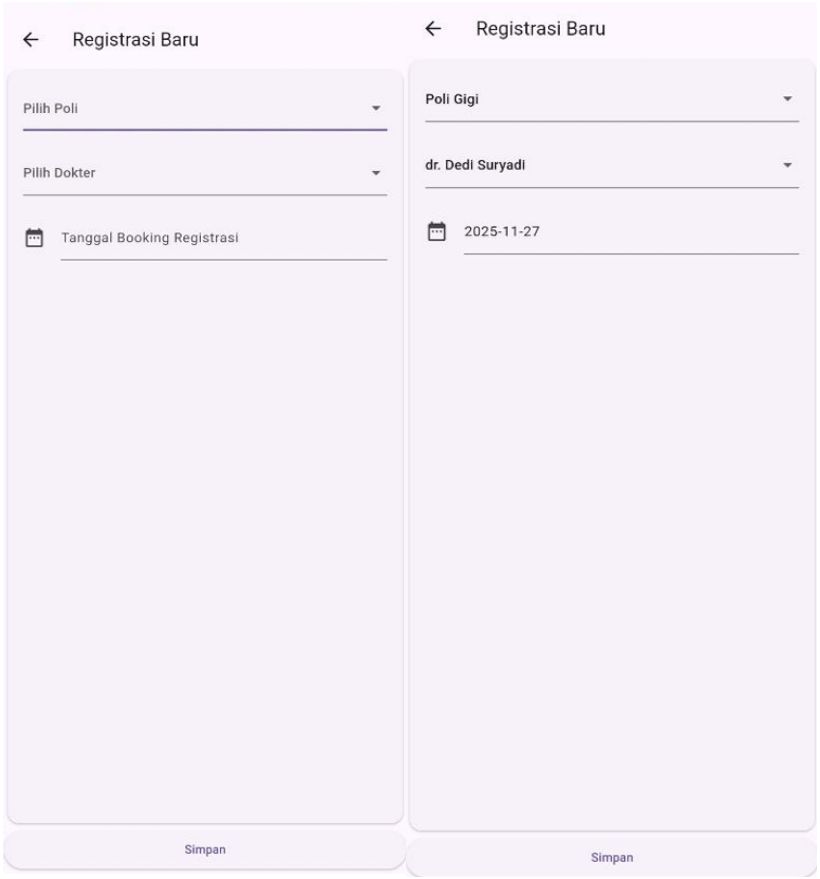
```
lib > page > modul_pasien > pesan_obat > create.dart > ...
1  import 'dart:async';
2  import 'dart:convert';
3  import 'package:flutter/material.dart';
4  import 'package:google_maps_flutter/google_maps_flutter.dart';
5  import 'package:good_health/model/obat.dart';
6  import 'package:good_health/model/pesan_obat.dart';
7  import 'package:good_health/util/util.dart';
8
9  class CreatePage extends StatefulWidget {
10   @override
11   _CreatePageState createState() => _CreatePageState();
12 }
13
14 class _CreatePageState extends State<CreatePage> {
15   TextEditingController alamatCont = new TextEditingController();
16   TextEditingController ketCont = new TextEditingController();
17   LatLng alamatLatLng = LatLng(-7.274537183066055, 112.7938013614994);
18   int totalBiaya = 0;
19
20   List<Obat> obats = <Obat>[];
21   Completer<GoogleMapController> _controller = Completer();
22   final Set<Marker> _markers = {};
23   static late CameraPosition _currentPosition; // Location set is Kampus PENS
24
25   @override
26   void initState() {
27     super.initState();
28     _currentPosition = CameraPosition(target: alamatLatLng, zoom: 14.4746);
29     _addMarker(alamatLatLng);
30
31     fetchObats().then((result) {
32       setState(() {
33         obats = result;
34       });
35     });
36   }
```

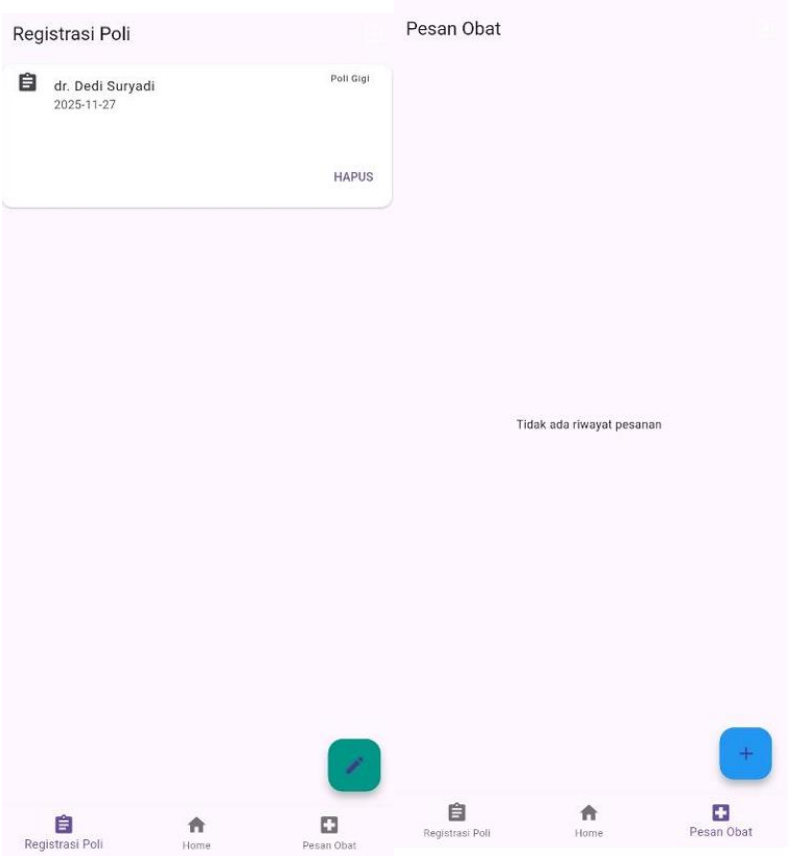
3.32 Jalan program

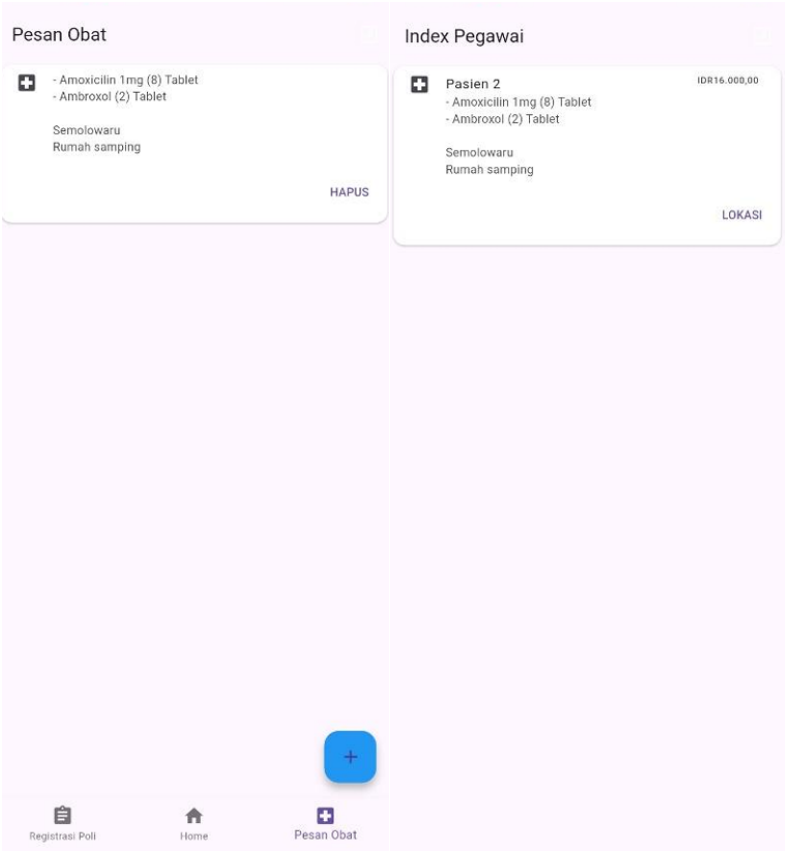


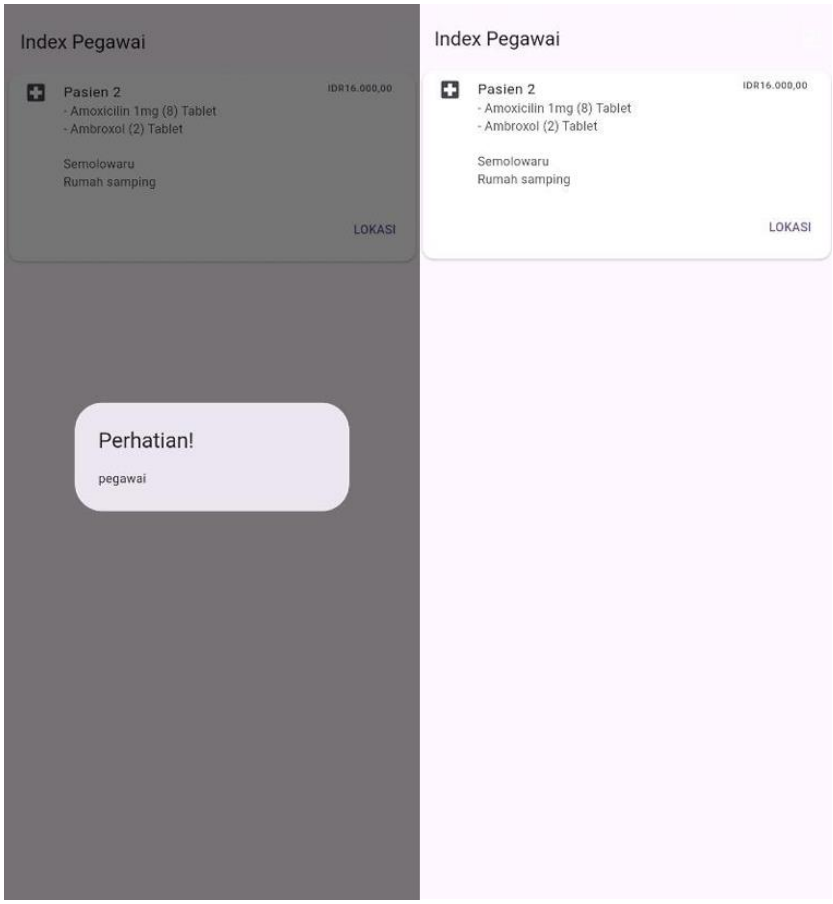

Home Pasien
Selamat Datang
Aplikasi GOOD HEALTH
Registrasi Poli
Home
Pesan Obat

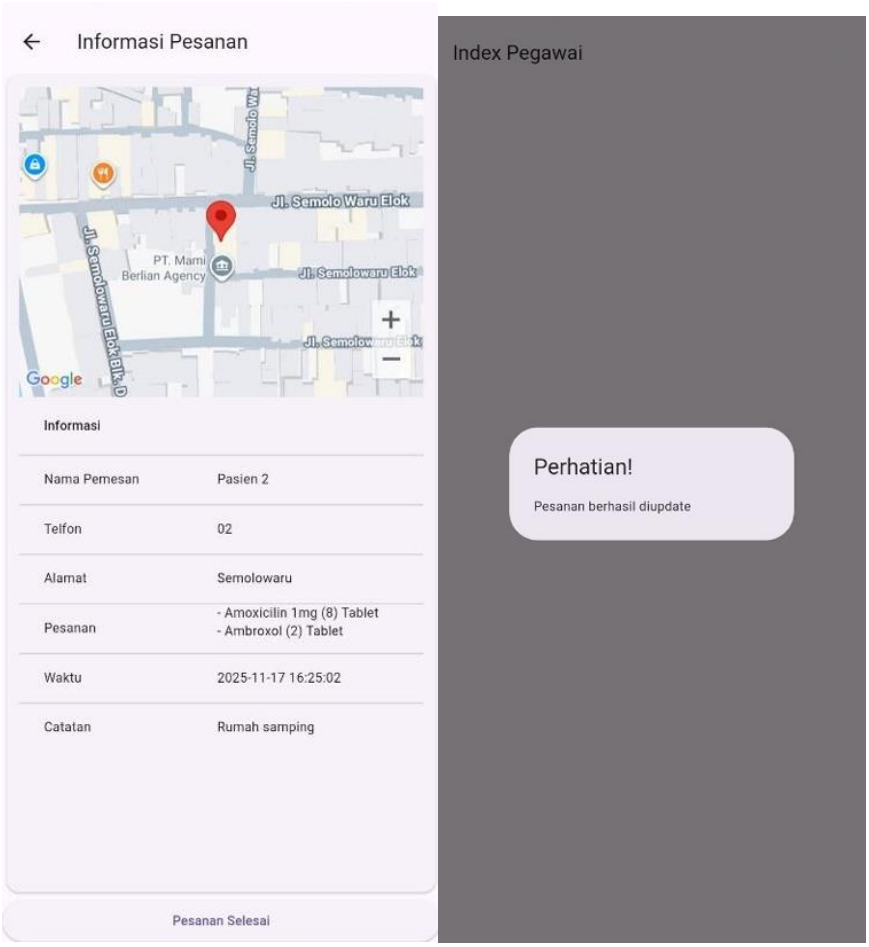

Registrasi Poli
Tidak ada riwayat registrasi
Registrasi Poli
Home
Pesan Obat


Registrasi Baru
Pilih Poli
Pilih Dokter
Tanggal Booking Registrasi
Simpan
Registrasi Baru
Poli Gigi
dr. Dedi Suryadi
2025-11-27
Simpan


Registrasi Poli
dr. Dedi Suryadi
2025-11-27
Poli Gigi
HAPUS
Registrasi Poli
Home
Pesan Obat
Pesan Obat
Tidak ada riwayat pesanan
Registrasi Poli
Home
Pesan Obat


Pesan Obat
- Amoxicilin 1mg (8) Tablet
- Ambroxol (2) Tablet
Semolowaru
Rumah samping
HAPUS
Registrasi Poli
Home
Pesan Obat
Index Pegawai
Pasien 2
IDR16.000,00
- Amoxicilin 1mg (8) Tablet
- Ambroxol (2) Tablet
Semolowaru
Rumah samping
LOKASI


Index Pegawai
Pasien 2
IDR16.000,00
- Amoxicilin 1mg (8) Tablet
- Ambroxol (2) Tablet
Semolowaru
Rumah samping
LOKASI
Perhatian!
pegawai
Index Pegawai
Pasien 2
IDR16.000,00
- Amoxicilin 1mg (8) Tablet
- Ambroxol (2) Tablet
Semolowaru
Rumah samping
LOKASI



4. Analisa

Pada praktikum kali ini, saya mencoba membangun sebuah aplikasi Android menggunakan framework **Flutter**, yang menggunakan bahasa pemrograman **Dart**. Langkah pertama adalah membuat proyek Flutter baru. Setelah proyek berhasil dibuat, file konfigurasi **pubspec.yaml** saya edit untuk memasukkan berbagai library online yang diperlukan, seperti Google Maps, protokol HTTP, ikon tambahan, dan lainnya.

Agar aplikasi dapat menggunakan layanan Google Maps, diperlukan aktivasi API melalui Google Cloud. Setelah API key untuk Android dan iOS didapatkan, key tersebut dimasukkan ke dalam file konfigurasi masing-masing platform, yaitu **AndroidManifest.xml** untuk Android dan **AppDelegate.swift** untuk iOS. Kemudian, folder **lib** diisi dengan tampilan frontend yang sudah disiapkan sebelumnya di classroom.

Sebelum beralih ke bagian antarmuka, terlebih dahulu saya melakukan inisialisasi model untuk koneksi ke database yang sudah dibuat di praktikum sebelumnya.

Proses ini menggunakan protokol HTTP, dan jumlah serta nama model disesuaikan dengan struktur tabel di database. Pada folder **util**, terdapat file utilitas yang berisi komponen atau fungsi yang bisa dipanggil dari berbagai bagian aplikasi, serta file **session** yang bertanggung jawab mengelola login dan logout baik untuk pengguna pasien maupun pegawai. File **main.dart** kemudian saya edit untuk membangun struktur dasar aplikasi, seperti menentukan halaman awal, warna tema, dan konfigurasi global lainnya.

Selanjutnya, pada folder **page**, terdapat tiga subfolder dan satu file utama. Subfolder tersebut adalah **list_widget**, **modul_pasien**, dan **modul_pegawai**, sedangkan file utamanya adalah **login.dart**. Sesuai pengaturan di main.dart, halaman login inilah yang pertama kali ditampilkan kepada pengguna selama tidak ada session aktif. Pada halaman ini, pengguna diminta memasukkan username dan password sesuai akun yang pernah didaftarkan. Jika belum memiliki akun, pengguna dapat menuju halaman registrasi dengan menekan tombol *“Registrasi pasien baru”*.

Halaman registrasi akun terdapat pada **lib/page/modul_pasien/sign_up.dart**. Di halaman tersebut, pengguna diminta mengisi nama lengkap, nomor HP, dan email yang akan digunakan untuk login. Setelah registrasi berhasil, pengguna dapat langsung masuk dengan menggunakan nomor HP sebagai password awal.

Setelah login, pengguna (pasien) akan diarahkan ke **dashboard pasien**, yang menyediakan fitur registrasi poli dan pemesanan obat. Pada halaman registrasi poli, aplikasi menampilkan jadwal poli berdasarkan model **reg_poli**: mulai dari nama dokter, tanggal poli, hingga jenis poli. Untuk menambah registrasi baru, pengguna dapat menekan tombol berlogo pensil di kanan bawah. Pengguna kemudian memilih jenis poli, dokter, serta tanggal kunjungan. Daftar poli bersifat hardcoded, sedangkan daftar dokter diambil dari model **dokter** yang terhubung langsung ke database. Setelah semua dipilih, pengguna dapat menekan **“Simpan”** untuk menambahkan jadwal poli.

Pada menu **Pesan Obat**, pengguna dapat melihat daftar pesanan obat mulai dari jenis dan jumlah obat hingga alamat dan status pemesanan, yang semuanya diambil dari model **obat**. Untuk membuat pesanan baru, pengguna menekan tombol “+”. Di halaman pemesanan obat, tersedia tampilan Google Maps yang sudah diintegrasikan, kolom alamat, kolom keterangan, serta daftar obat lengkap dengan checkbox dan jumlah pesanan. Semua nama obat diambil dari model obat yang terhubung ke database. Setelah data diisi, aplikasi akan menampilkan total harga di bagian bawah. Pengguna kemudian dapat menekan **“Pesan Sekarang”** untuk mengirim pesanan kepada pegawai.

Selain dashboard pasien, aplikasi ini juga menyediakan **dashboard pegawai**, yang hanya dapat diakses menggunakan email dan password pegawai. Pada dashboard ini, pegawai dapat melihat pesanan obat dari para pasien, termasuk nama pasien, total pembayaran, daftar dan jumlah obat, serta alamat pengiriman yang diambil dari model **pesan_obat**. Pegawai juga dapat melihat detail pesanan dan lokasi pasien melalui Google Maps dengan menekan tombol **“Lokasi”**. Setelah obat dikirim dan pasien menerimanya, pegawai dapat menekan tombol **“Pesanan Selesai”**, yang akan menghapus data pemesanan dari database melalui model pesan_obat.

5. Kesimpulan

Flutter adalah framework yang banyak digunakan untuk mengembangkan aplikasi Android karena kemudahan dan fleksibilitasnya. Pada praktikum ini, saya melakukan percobaan untuk menjalankan sebuah aplikasi Flutter yang sudah dilengkapi dengan frontend siap pakai serta backend yang berasal dari praktikum sebelumnya.

Aplikasi ini memiliki beberapa fitur utama, mulai dari login dan registrasi akun, dashboard pasien yang menyediakan layanan registrasi poli dan pemesanan obat, hingga dashboard pegawai untuk memproses pesanan obat yang dibuat oleh pasien.

Secara keseluruhan, aplikasi dapat berjalan dengan baik tanpa muncul error, sehingga seluruh rangkaian praktikum dapat diselesaikan dengan lancar.