

LAPORAN
Jobsheet-12 : Weather app using API integration in Flutter
TUGAS PEMROGRAMAN MOBILE

Dibimbing oleh: Bapak Ade Ismail, S.Kom., M.TI.



Disusun oleh:
ANARADI OCTA LAVECHIA
2241760007 / SIB-3D

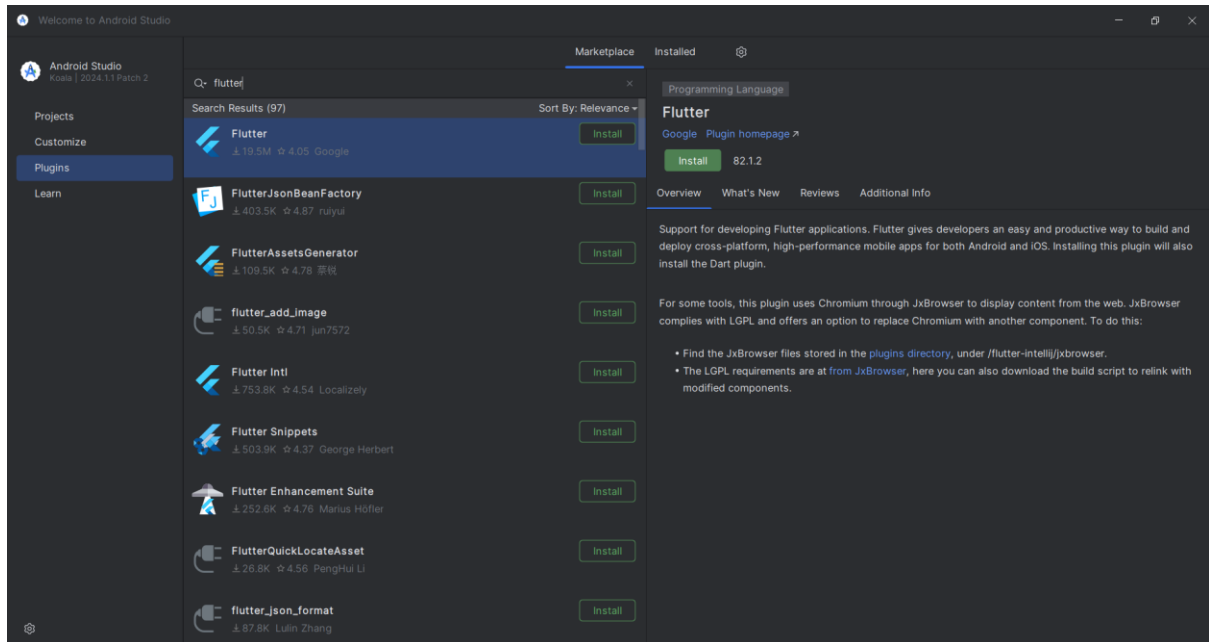
JURUSAN TEKNOLOGI INFORMASI
PRODI D-IV SISTEM INFORMASI BISNIS
POLITEKNIK NEGERI MALANG
2024

Praktikum 1: Menerapkan Control Flows ("if/else")

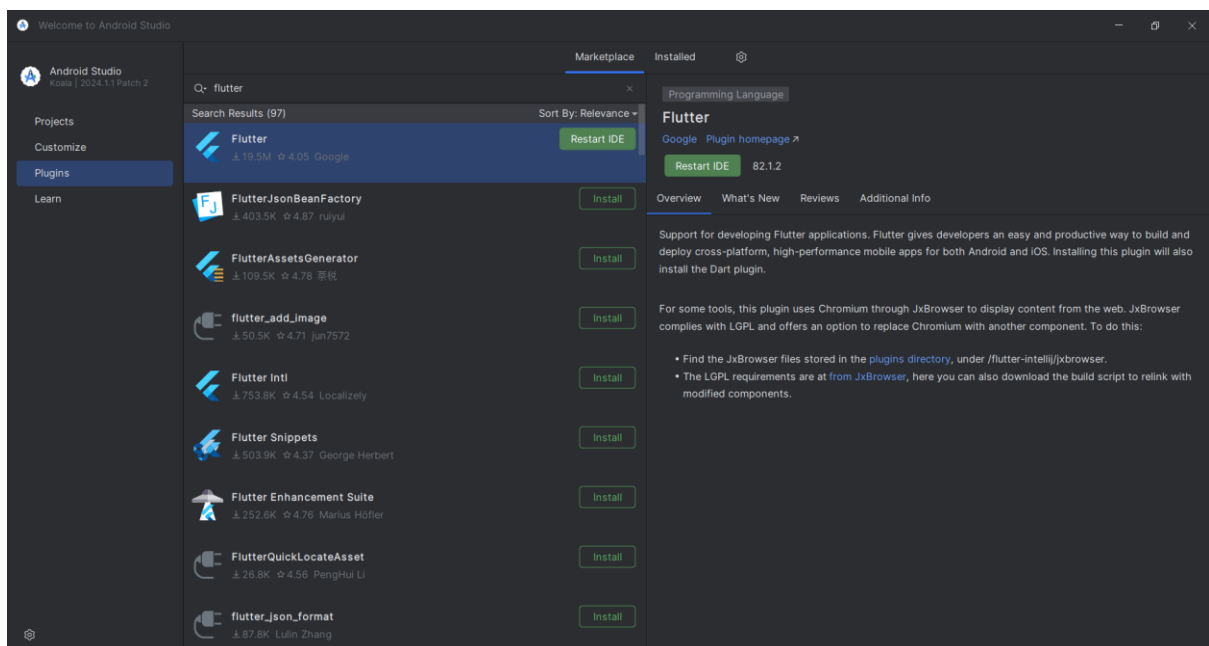
Selesaikan langkah-langkah praktikum berikut ini menggunakan DartPad di browser Anda.

Langkah 1:

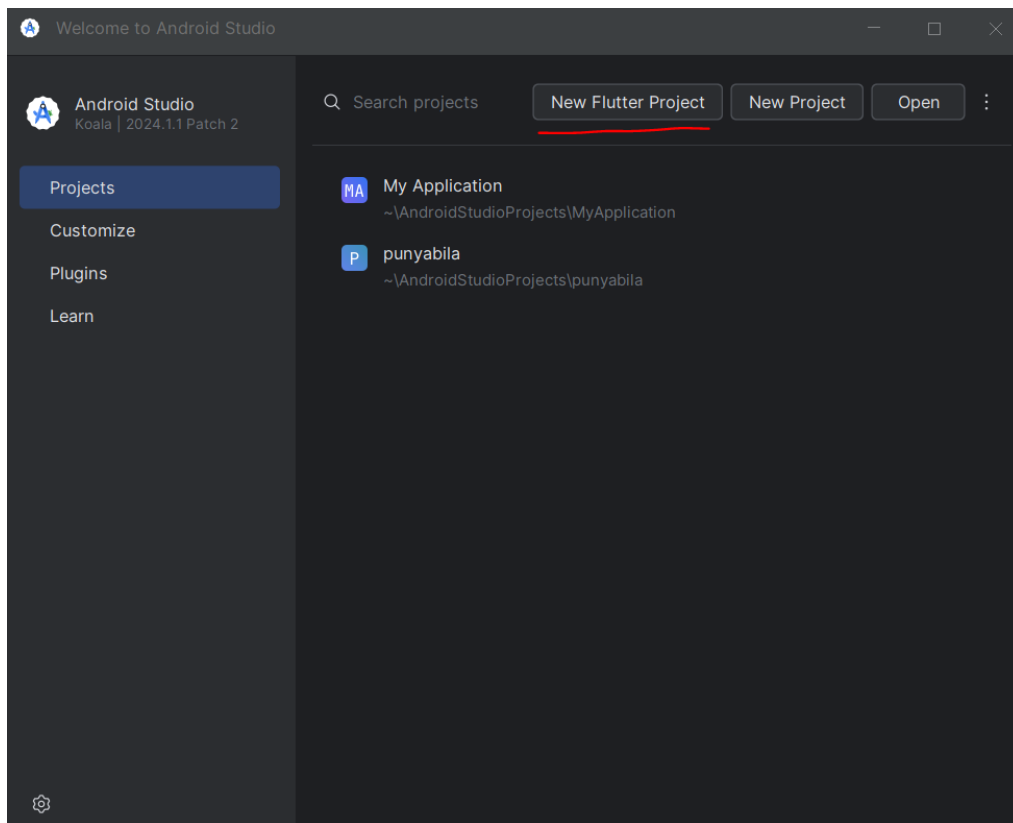
Buka Android Studio sebagai IDE dan pilih “Start a new Flutter project”. Jika pada home project tidak ada menu tersebut maka dapat dipastikan bahwa “Plugin Fultter” sudah terinstal di Android Studio. Jika belum maka dapat dilakukan instalasi pada menu plugins seperti Berikut.



Saat menginstal plugin Flutter maka plugin Dart akan otomatis ikut terinstal. Setelah instalasi selesai, klik **Restart IDE** untuk menerapkan perubahan.

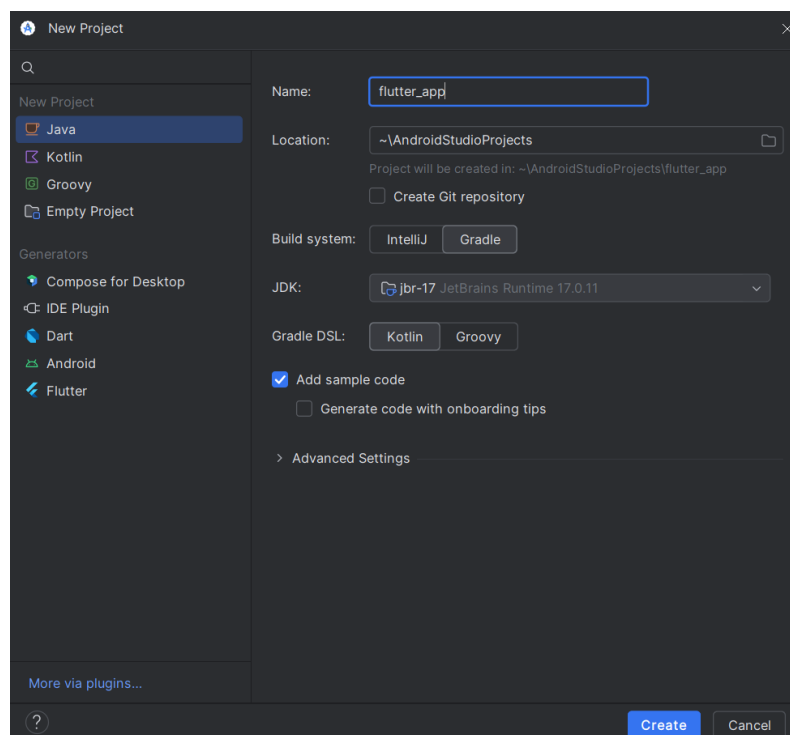


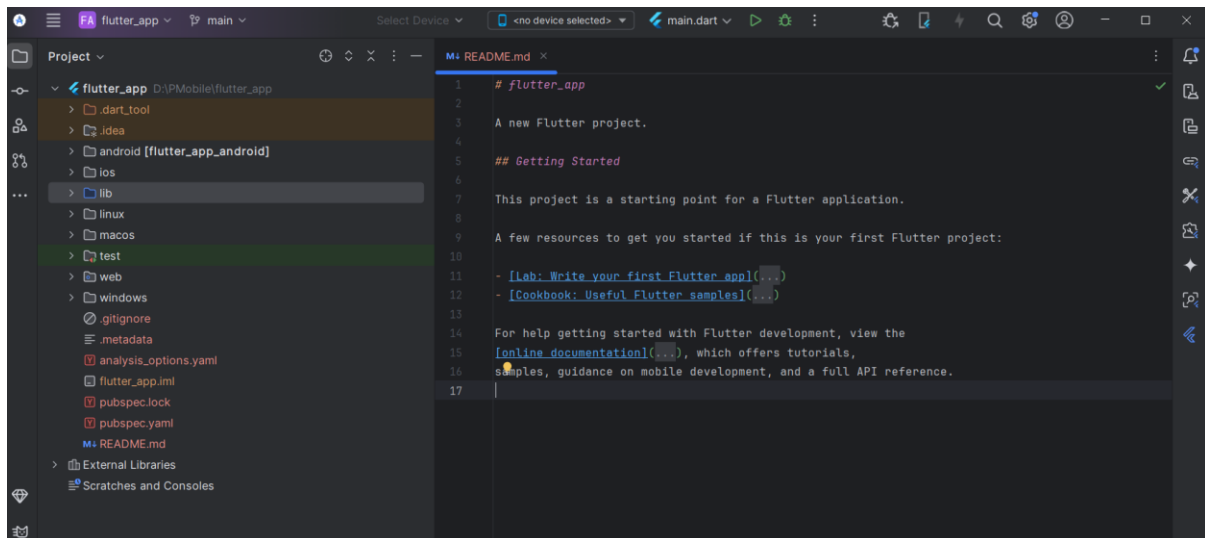
Maka setelah Android Studio terbuka kembali menu **"New Flutter Project"** akan muncul di halaman utama.



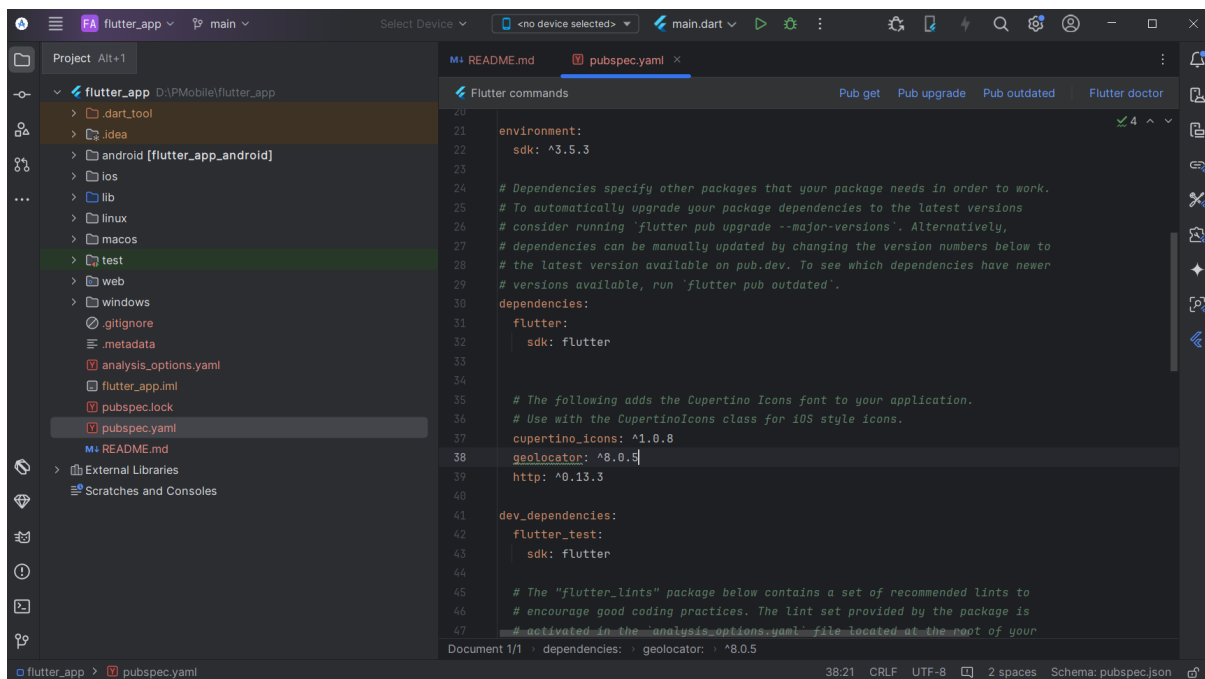
Langkah 2:

Membuka nama project baru

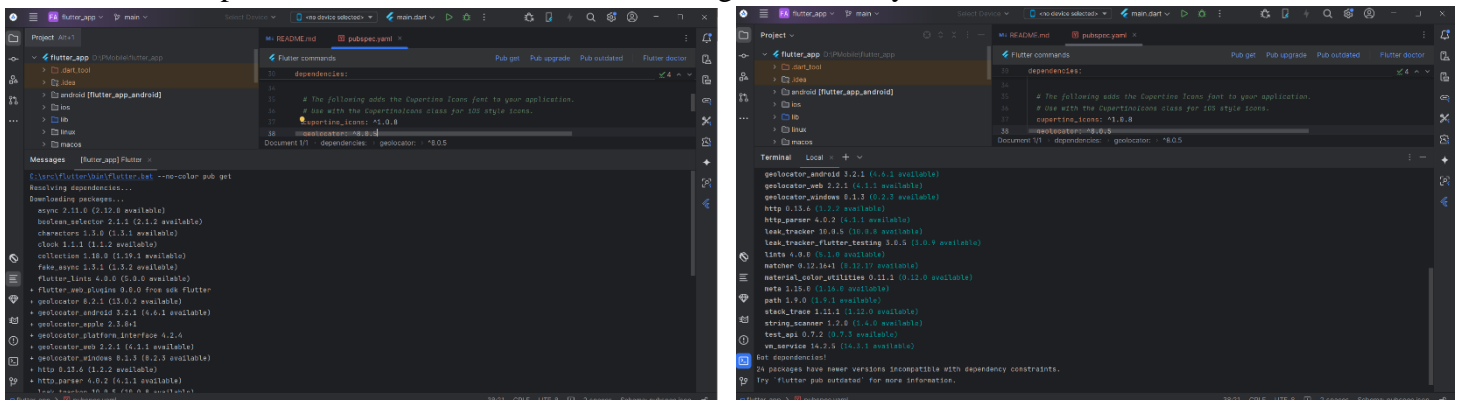




Buka file pubspec.yaml dan tambahkan http untuk mengambil data dari API dan geolocator untuk mendapatkan lokasi.

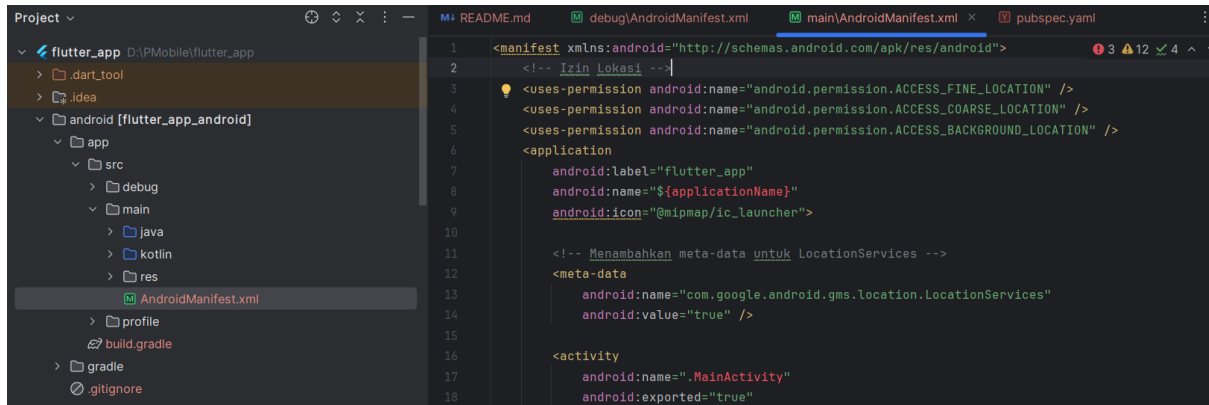


Jalankan perintah flutter pub get untuk mengunduh library.

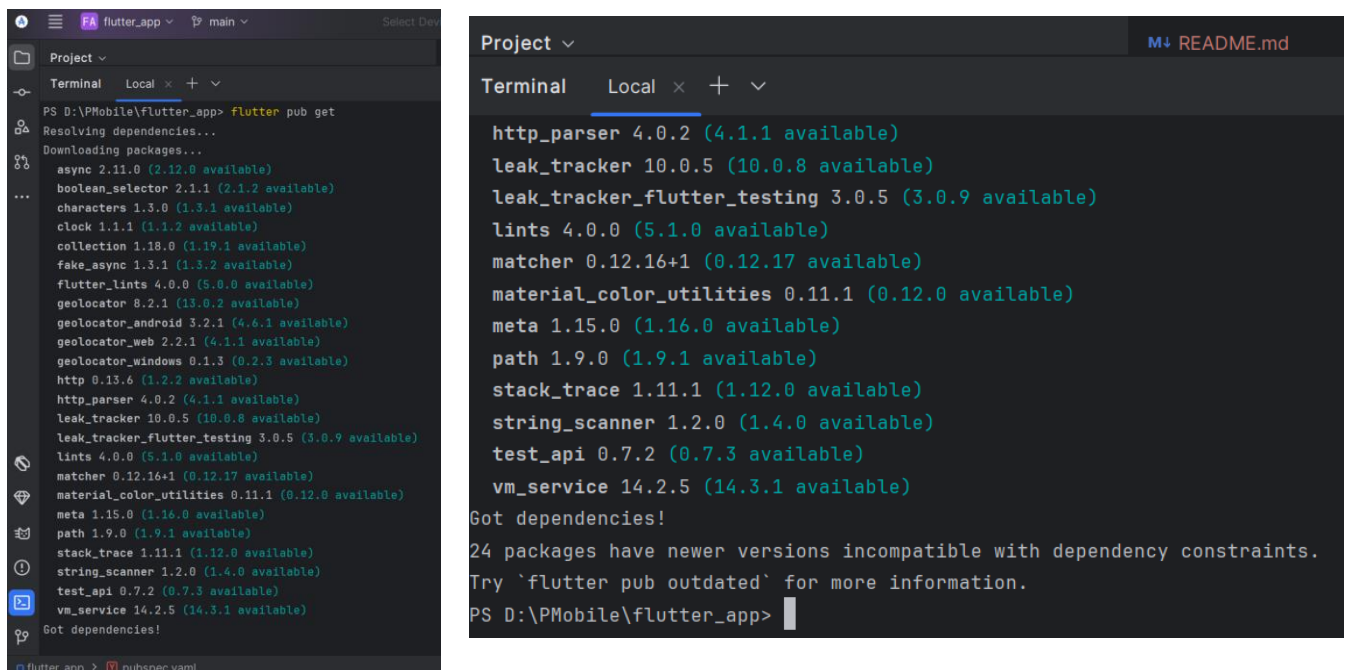


Langkah 3:

Konfigurasi Geolocator dengan menambahkan izin lokasi di file `AndroidManifest.xml` seperti berikut



Jalankan perintah `flutter pub get` untuk mengunduh library



Langkah 4:

Buat Struktur Utama dengan Membuat file yang dibutuhkan seperti `main.dart` sebagai titik masuk aplikasi dan `homescreen.dart` sebagai halaman utama, serta mengubah widget `MyApp` menjadi `MaterialApp` di `main.dart`.

```

lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2 import 'screen/homeScreen.dart';
3
4 Run | Debug | Profile
5 void main() {
6   runApp(const MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   const MyApp({super.key});
11
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: 'Weather App',
16       debugShowCheckedModeBanner: false,
17       theme: ThemeData(
18         colorScheme: ColorScheme.fromSeed(seedColor: Colors.blue),
19         useMaterial3: true,
20       ), // ThemeData
21       home: const HomeScreen(),
22     ); // MaterialApp
23   }
24 }

```

```

lib > screen > homeScreen.dart > _HomeScreenState
1 import 'package:flutter/material.dart';
2 import '/config/location_helper.dart'; // Assuming you have location helper functions
3 import '/services/weather_service.dart'; // Assuming you have a weather service
4 import '/widgets/weather_card.dart'; // Assuming custom widget for displaying weather info
5 import '/widgets/search_field.dart'; // Assuming custom widget for search field
6
7 class HomeScreen extends StatefulWidget {
8   const HomeScreen({super.key});
9
10  @override
11  State<HomeScreen> createState() => _HomeScreenState();
12}
13
14 class _HomeScreenState extends State<HomeScreen> {
15  String cityName = "Loading...";
16  num temperature = 0;
17  num humidity = 0;
18  num windSpeed = 0;
19  num cloudCover = 0;
20  num pressure = 0;
21  bool isLoading = false;
22
23  @override
24  void initState() {
25    super.initState();
26    _fetchWeatherByLocation();
27  }
28
29  void _fetchWeatherByLocation() async {
30    final position = await getCurrentLocation(); // Assuming you have a function to get location
31    if (position != null) {
32      final weatherData = await fetchWeatherByLocation(position); // Assuming function to fetch weather by location
33      if (weatherData != null) {
34        setState(() {
35          cityName = weatherData['name'];
36          temperature = weatherData['main']['temp'] - 273.15; // Convert Kelvin to Celsius
37          humidity = weatherData['main']['humidity'];
38          windSpeed = weatherData['wind']['speed'];
39          cloudCover = weatherData['clouds']['all'];
40          pressure = weatherData['main']['pressure'];
41          isLoading = true;
42        });
43      }
44    }
45  }
46
47  void _fetchWeatherByCity(String city) async {
48    final weatherData = await fetchWeatherByCity(city); // Assuming function to fetch weather by city name
49    if (weatherData != null) {
50      setState(() {
51        cityName = weatherData['name'];
52        temperature = weatherData['main']['temp'] - 273.15; // Convert Kelvin to Celsius
53        humidity = weatherData['main']['humidity'];
54        windSpeed = weatherData['wind']['speed'];
55        cloudCover = weatherData['clouds']['all'];
56        pressure = weatherData['main']['pressure'];
57        isLoading = true;
58      });
59    }
60  }
61
62  // New method to build weather cards with images
63  Widget _buildWeatherCards() {
64    return Column(
65      children: [
66        _buildWeatherCard('Temperature', '${temperature.toStringAsFixed(2)} °C', 'assets/images/thermometer.jpg'),
67        _buildWeatherCard('Pressure', '${pressure.toInt()} hPa', 'assets/images/barometer.png'),
68        _buildWeatherCard('Humidity', '${humidity.toInt()} %', 'assets/images/humidity.png'),
69        _buildWeatherCard('Cloud Cover', '${cloudCover.toInt()} %', 'assets/images/clouds.png'),
70      ],
71    );
72  }
73}

```

```

lib > screen > homeScreen.dart > _HomeScreenState
14 class _HomeScreenState extends State<HomeScreen> {
15   // Helper method to create individual weather cards with images
16   Widget _buildWeatherCard(String title, String value, String imagePath) {
17     return Card(
18       margin: const EdgeInsets.symmetric(vertical: 10),
19       child: Padding(
20         padding: const EdgeInsets.all(10.0),
21         child: Row(
22           mainAxisAlignment: MainAxisAlignment.start,
23           children: [
24             // Image for the weather parameter
25             Image.asset(
26               imagePath,
27               width: 60,
28               height: 60,
29               fit: BoxFit.cover,
30             ), // Image.asset
31             const SizedBox(width: 10),
32             Column(
33               crossAxisAlignment: CrossAxisAlignment.start,
34               children: [
35                 Text(title, style: const TextStyle(fontSize: 18, fontWeight: FontWeight.bold)),
36                 Text(value, style: const TextStyle(fontSize: 16)),
37               ], // Column
38             ), // Row
39           ], // Padding
40         ), // Card
41       );
42
43   // Method for displaying city name with an image
44   Widget _buildCityName() {
45     return Row(
46       children: [
47         // You can add an icon or image next to the city name (outside the card)
48         Image.asset(
49           'assets/images/loc.png', // Example image for the city icon
50           width: 30,
51           height: 30,
52         ), // Image.asset
53         const SizedBox(width: 10),
54         Text(
55           cityName,
56           style: const TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
57         ), // Text
58       ], // Row
59     );
60
61   @override
62   Widget build(BuildContext context) {
63     return Scaffold(
64       appBar: AppBar(
65         title: const Text('Weather App'),
66       ), // AppBar
67       body: Container(
68         decoration: const BoxDecoration(
69           gradient: LinearGradient(
70             colors: [Colors.teal, Colors.pink, Colors.purple],
71             begin: Alignment.bottomLeft,
72             end: Alignment.topRight,
73             stops: [0.3, 0.6, 1.0],
74           ), // LinearGradient
75         ), // BoxDecoration
76         padding: const EdgeInsets.all(16.0),
77         child: Column(
78           mainAxisAlignment: MainAxisAlignment.start,
79           children: [
80             SearchField(onSubmitted: _fetchWeatherByCity, // Custom search field widget
81               const SizedBox(height: 20),
82             ), // Display city name with icon/image beside it
83             const SizedBox(height: 20),
84             _buildWeatherCards(), // Display weather cards with images
85           ], // Column
86         ), // Container
87       ), // Scaffold
88     );
89   }
90}

```

Tampilan awal ketika di run:



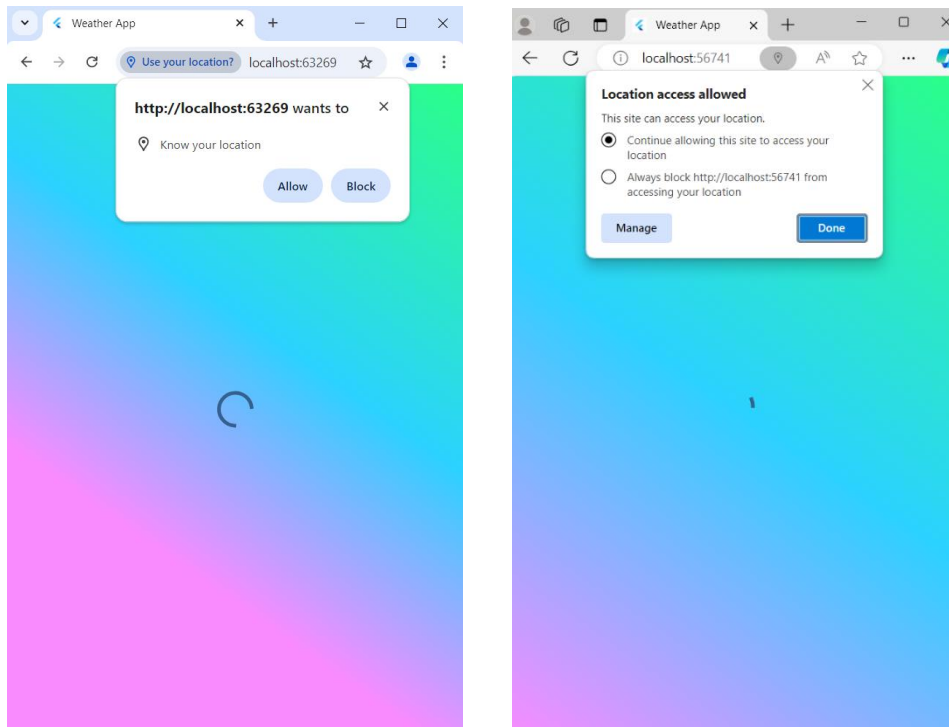
Langkah 5:

Di `homescreen.dart` import `geolocator` lalu buat fungsi untuk mendapatkan koordinat lokasi seperti berikut.

```
lib > screen > 📄 homescreen.dart > 🛠️ _HomeScreenState
1  import 'package:flutter/material.dart';
2  import '/config/location_helper.dart'; // Assu
3  import '/services/weather_service.dart'; // As
4  import '/widgets/weather_card.dart'; // Assumi
5  import '/widgets/search_field.dart'; // Assumi
6  import 'package:geolocator/geolocator.dart';
7
```

```
lib > screen > 📄 homescreen.dart > ...
15  class _HomeScreenState extends State<HomeScreen> {
24  @override
25  void initState() {
26    super.initState();
27    _fetchWeatherByLocation();
28  }
29
30  void _fetchWeatherByLocation() async {
31    final position = await getCurrentLocation(); // Assuming you have a function to get location
32    if (position != null) {
33      final weatherData = await fetchWeatherByLocation(position); // Assuming function to fetch weather by location
34      if (weatherData != null) {
35        setState(() {
36          cityName = weatherData['name'];
37          temperature = weatherData['main']['temp'] - 273.15; // Convert Kelvin to Celsius
38          humidity = weatherData['main']['humidity'];
39          windSpeed = weatherData['wind']['speed'];
40          cloudCover = weatherData['clouds']['all'];
41          pressure = weatherData['main']['pressure'];
42          isLoading = true;
43        });
44      }
45    }
46  }
```

Output yang dihasilkan saat dijalankan



Langkah 6:

Dapatkan API Key di OpenWeatherMap <https://openweathermap.org> pilih paket gratis (Free Plan) lalu daftar akun bila belum memiliki akun dengan mengklik tombol "Sign In"

The screenshot shows the 'openweathermap.org/price' page. It features a navigation bar with links like 'Guide', 'API', 'Dashboard', 'Marketplace', 'Pricing', 'Maps', 'Our Initiatives', 'Partners', 'Blog', 'For Business', 'Sign in', and 'Support'. Below the navigation bar is a section titled 'Current weather and forecasts collection' which contains a table of subscription plans.

Free	Startup	Developer	Professional	Enterprise
Get API key	30 GBP/ month Subscribe	140 GBP/ month Subscribe	370 GBP/ month Subscribe	1500 GBP/ month Subscribe
60 calls/minute 1,000,000 calls/month	600 calls/minute 10,000,000 calls/month	3,000 calls/minute 100,000,000 calls/month	30,000 calls/minute 1,000,000,000 calls/month	200,000 calls/minute 5,000,000,000 calls/month
Current Weather 3-hour Forecast 5 days Hourly Forecast 4 days Daily Forecast 16 days Climatic Forecast 30 days Bulk Download	Current Weather 3-hour Forecast 5 days Hourly Forecast 4 days Daily Forecast 16 days Climatic Forecast 30 days Bulk Download	Current Weather 3-hour Forecast 5 days Hourly Forecast 4 days Daily Forecast 16 days Climatic Forecast 30 days Bulk Download	Current Weather 3-hour Forecast 5 days Hourly Forecast 4 days Daily Forecast 16 days Climatic Forecast 30 days Bulk Download (global cities)	Current Weather 3-hour Forecast 5 days Hourly Forecast 4 days Daily Forecast 16 days Climatic Forecast 30 days Bulk Download (global cities + ZIPs of US, EU, UK)

At the bottom of the page, there is a cookie consent banner that reads: 'We use cookies to personalize content and to analyze our traffic. Please decide if you are willing to accept cookies from our website.' It includes links for 'Advanced Settings', 'Decline', and 'Accept'.

home.openweathermap.org/users/sign_up

OpenWeather

Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business Sign In Support

Create New Account

anaradiocla

nelaporseni@gmail.com

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather [Privacy Policy](#).

☒ I am 16 years old and over

☒ I agree with [Privacy Policy](#), [Terms and conditions of sale](#) and [Websites terms and conditions of use](#)

I consent to receive communications from OpenWeather Group of Companies and their partners.

☒ System news (API usage alert, system update, temporary system shutdown, etc)

☒ Product news (change to price, new product features, etc)

☒ Corporate news (our life, the launch of a new service, etc)

☒ I'm not a robot

[Create Account](#)

Setelah mendaftar maka akan ada perintah membuka gmail untuk verifikasi akun yang didaftarkan kemudian masuk ke akun yang baru dibuat.

home.openweathermap.org

OpenWeather

Weather in your city

Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business anar... Support

We have sent the confirmation link to nelaporseni@gmail.com. Please check your email.

New Products Services API keys Billing plans Payments Block logs My orders My profile Ask a question

https://home.openweathermap.org/users/sign_in

OpenWeather

Weather in your city

Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business Sign In Support

Notice

Your email address has been successfully confirmed.

Sign In To Your Account

anaradiocla

☐ Remember me

[Submit](#)

Not registered? [Create an Account.](#)

Lost your password? [Click here to recover.](#)

Setelah login, buka dashboard akun. Di dashboard, pilih API keys pada menu sebelah kiri akan terdapat API Key default yang disediakan oleh OpenWeatherMap. API key ini sudah aktif dan bisa digunakan untuk membuat aplikasi.



home.openweathermap.org/api_keys

OpenWeather

Weather in your city Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business anara... Su

New Products Services **API keys** Billing plans Payments Block logs My orders My profile Ask a question

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
a527ed6a80b5d3e6fba92330b2f7cc43	Default	Active	 

Create key

API key name

Product Collections

- Current and Forecast APIs
- Historical Weather Data
- Weather Maps
- Weather Dashboard

Subscription

- How to start
- Pricing
- Subscribe for free
- FAQ

Company

OpenWeather is a team of IT experts and data scientists that has been practising deep weather data science since 2014. For each point on the globe, OpenWeather provides historical, current and forecasted weather data via light-

Disini saya ganti nama Default nya menjadi mengikuti nama aplikasinya yaitu Weather App.

home.openweathermap.org/api_keys

OpenWeather



Weather in your city Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business anara... Su

New Products Services **API keys** Billing plans Payments Block logs My orders My profile Ask a question

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Edit API key name

API key name

Key	Name	Status	Actions
a527ed6a80b5d3e6fba92330b2f7cc43	Weather App	Active	 

Create key

API key name

Product Collections

- Current and Forecast APIs
- Historical Weather Data
- Weather Maps
- Weather Dashboard

Subscription

- How to start
- Pricing
- Subscribe for free
- FAQ

Company

OpenWeather is a team of IT experts and data scientists that has been practising deep weather data science since 2014. For each point on the globe, OpenWeather provides historical, current and forecasted weather data via light-

home.openweathermap.org/api_keys

OpenWeather



Weather in your city Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business anara... Su

New Products Services **API keys** Billing plans Payments Block logs My orders My profile Ask a question

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Notice

API key was edited successfully

Key	Name	Status	Actions
a527ed6a80b5d3e6fba92330b2f7cc43	Weather App	Active	 

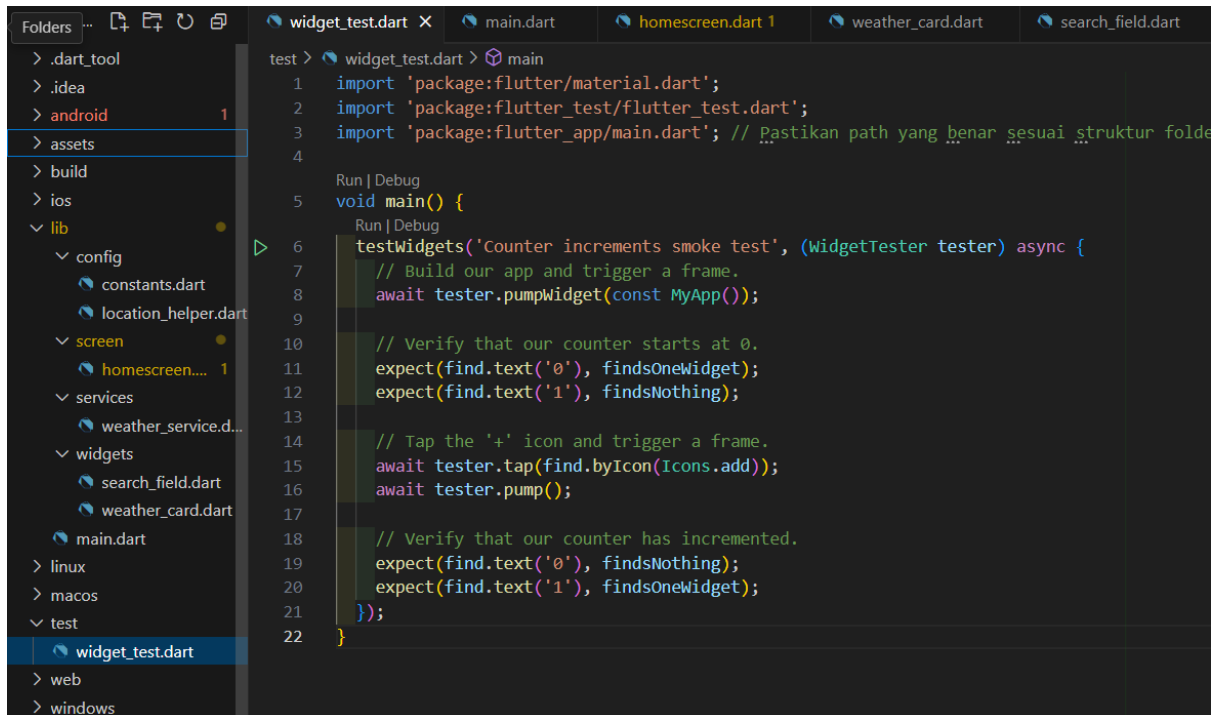
Create key

API key name

Kemudian masukan API Key ke Aplikasi dengan membuat folder baru bernama config yang berisi file constants.dart yang mana dalam file tersebut buat konstanta apiKey seperti berikut

```
> .dart_tool
> .idea
> android 1
lib > config > constants.dart > ...
1 const String apiKey = 'a527ed6a80b5d3e6fba92330b2f7cc43'; // Ganti dengan API Key yang kamu dapat
2 const String domain = 'https://api.openweathermap.org/data/2.5/weather?';
```

Dan perbaiki file widget_test.dart di folder test seperti berikut

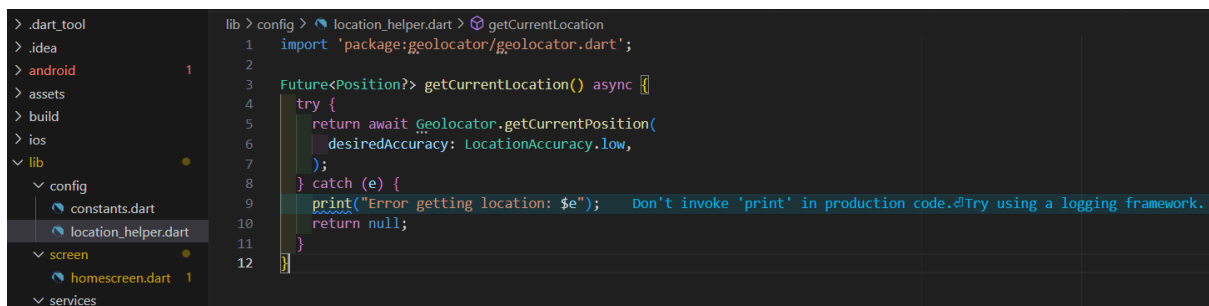


```
test > widget_test.dart > main
1 import 'package:flutter/material.dart';
2 import 'package:flutter_test/flutter_test.dart';
3 import 'package:flutter_app/main.dart'; // Pastikan path yang benar sesuai struktur folder
4
5 void main() {
6   testWidgets('Counter increments smoke test', (WidgetTester tester) async {
7     // Build our app and trigger a frame.
8     await tester.pumpWidget(const MyApp());
9
10    // Verify that our counter starts at 0.
11    expect(find.text('0'), findsOneWidget);
12    expect(find.text('1'), findsNothing);
13
14    // Tap the '+' icon and trigger a frame.
15    await tester.tap(find.byIcon(Icons.add));
16    await tester.pump();
17
18    // Verify that our counter has incremented.
19    expect(find.text('0'), findsNothing);
20    expect(find.text('1'), findsOneWidget);
21  });
22 }
```

Langkah 7:

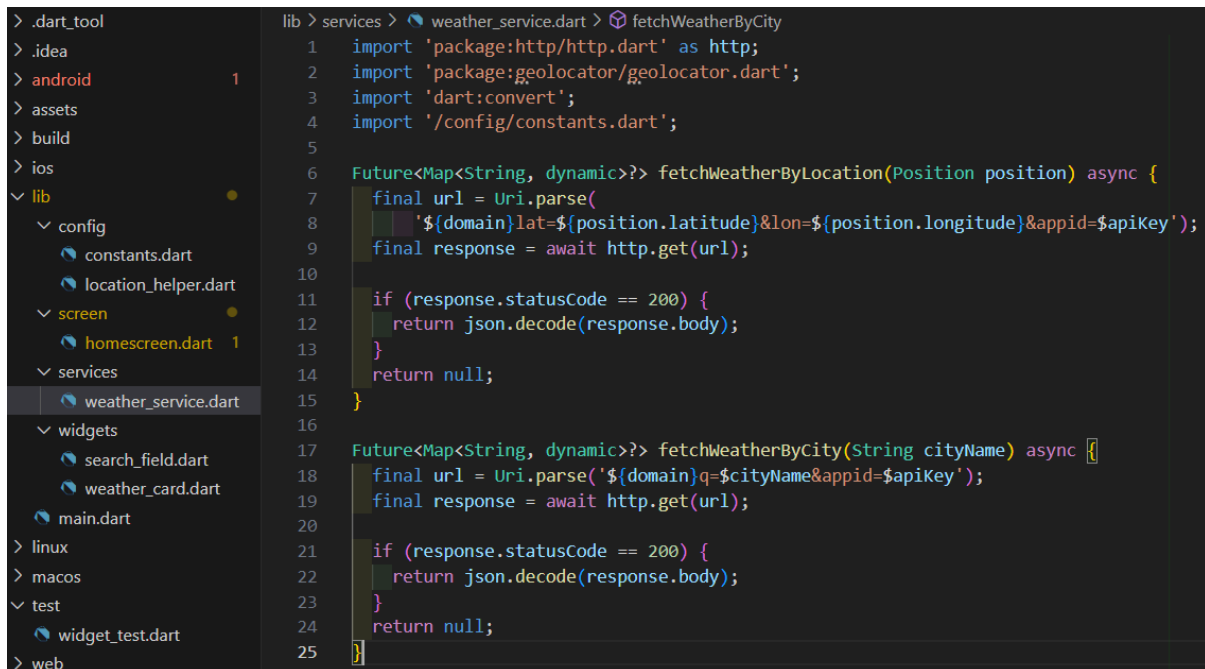
Ambil Data Cuaca gunakan library http untuk mengambil data dari API berdasarkan latitude/longitude, dan berdasarkan nama kota serta decode respons JSON menggunakan dart:convert.

Disini saya membuat file baru dalam folder config bernama location_helper.dart yang isinya seperti berikut



```
lib > config > location_helper.dart > getCurrentLocation
1 import 'package:geolocator/geolocator.dart';
2
3 Future<Position> getCurrentLocation() async {
4   try {
5     return await Geolocator.getCurrentPosition(
6       desiredAccuracy: LocationAccuracy.low,
7     );
8   } catch (e) {
9     print("Error getting location: $e"); // Don't invoke 'print' in production code. Try using a logging framework.
10    return null;
11  }
12 }
```

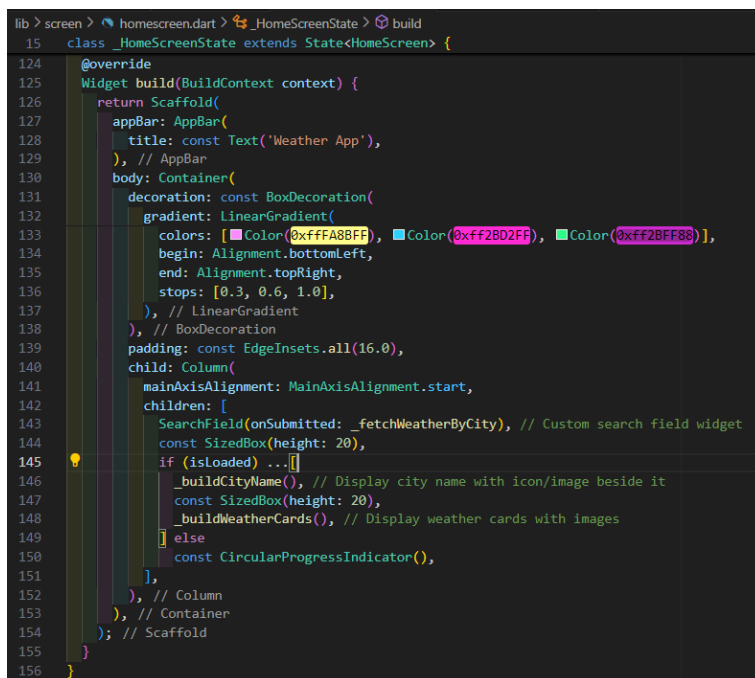
Kemudian saya juga membuat folder `services` yang berisi file `weather_service.dart` untuk mengambil data cuaca berdasarkan lokasi yang inputannya seperti berikut.



```
lib > services > weather_service.dart > fetchWeatherByCity
1  import 'package:http/http.dart' as http;
2  import 'package:geolocator/geolocator.dart';
3  import 'dart:convert';
4  import '/config/constants.dart';
5
6  Future<Map<String, dynamic>?> fetchWeatherByLocation(Position position) async {
7    final url = Uri.parse(
8      '${domain}lat=${position.latitude}&lon=${position.longitude}&appid=$apiKey';
9    final response = await http.get(url);
10
11    if (response.statusCode == 200) {
12      return json.decode(response.body);
13    }
14    return null;
15  }
16
17  Future<Map<String, dynamic>?> fetchWeatherByCity(String cityName) async {
18    final url = Uri.parse('${domain}q=$cityName&appid=$apiKey');
19    final response = await http.get(url);
20
21    if (response.statusCode == 200) {
22      return json.decode(response.body);
23    }
24    return null;
25  }
```

Langkah 8:

Bangun tampilan UI dengan menambahkan background gradasu di `homescreen.dart` pada folder `screen`



```
lib > screen > homescreen.dart > _HomeScreenState > build
15  class _HomeScreenState extends State<HomeScreen> {
16
17    @override
18    Widget build(BuildContext context) {
19      return Scaffold(
20        appBar: AppBar(
21          title: const Text('Weather App'),
22        ), // AppBar
23        body: Container(
24          decoration: const BoxDecoration(
25            gradient: LinearGradient(
26              colors: [Color(0xffffA88F), Color(0xff2BD2FF), Color(0xff2BFF88)],
27              begin: Alignment.bottomLeft,
28              end: Alignment.topRight,
29              stops: [0.3, 0.6, 1.0],
30            ), // LinearGradient
31          ), // BoxDecoration
32          padding: const EdgeInsets.all(16.0),
33          child: Column(
34            mainAxisAlignment: MainAxisAlignment.start,
35            children: [
36              SearchField(onSubmitted: _fetchWeatherByCity), // Custom search field widget
37              const SizedBox(height: 20),
38              if (isLoading) ...[
39                _buildCityName(), // Display city name with icon/image beside it
40                const SizedBox(height: 20),
41                _buildWeatherCards(), // Display weather cards with images
42              ] else
43                const CircularProgressIndicator(),
44            ],
45          ), // Column
46        ), // Container
47      ); // Scaffold
48    }
49  }
```

Langkah 9:

Tambahkan data cuaca menggunakan Widget Text jadi disini saya membuat folder baru di dalam lib yaitu folder widgets yang berisi file weather_card.dart yang berisi data cuaca dan file search_filde.dart yang digunakan widget nya untuk mencari atau menambah data lokasi. Berikut isi dari masing-masing file tersebut

- weather_card.dart

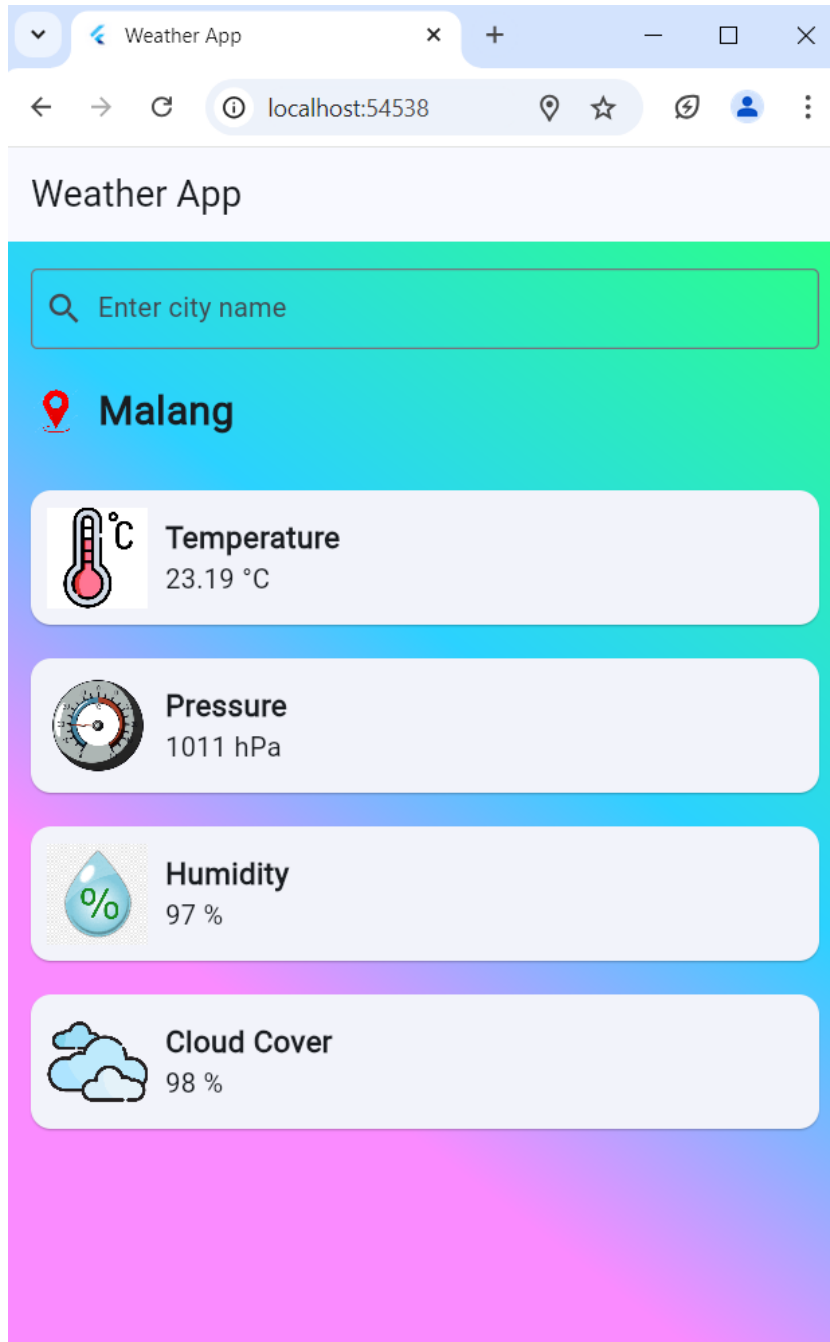
```
lib > widgets > weather_card.dart > WeatherCard
1  import 'package:flutter/material.dart';
2
3  class WeatherCard extends StatelessWidget {
4    final String title;
5    final String value;
6
7    const WeatherCard({required this.title, required this.value, super.key});
8
9    @override
10   Widget build(BuildContext context) {
11     return Card(
12       margin: const EdgeInsets.symmetric(vertical: 8),
13       child: ListTile(
14         title: Text(
15           title,
16           style: const TextStyle(fontWeight: FontWeight.bold),
17         ), // Text
18         trailing: Text(
19           value,
20           style: const TextStyle(fontSize: 16),
21         ), // Text
22       ), // ListTile
23     ); // Card
24   }
25 }
```

- search_filde.dart

```
lib > widgets > search_field.dart > SearchField > build
1  import 'package:flutter/material.dart';
2
3  class SearchField extends StatelessWidget {
4    final Function(String) onSubmitted;
5
6    const SearchField({required this.onSubmitted, super.key});
7
8    @override
9    Widget build(BuildContext context) {
10     return TextField(
11       decoration: const InputDecoration(
12         hintText: "Enter city name",
13         border: OutlineInputBorder(),
14         prefixIcon: Icon(Icons.search),
15       ), // InputDecoration
16       onSubmitted: onSubmitted,
17     ); // TextField
18   }
19 }
```

Langkah 10:

Uji dan Finalisasi. Berikut output yang dihasilkan dari aplikasi yang telah saya buat di atas yang menampilkan cuaca berdasarkan nama kota dan hasil pengujian menunjukan bahwa seluruh fungsi sudah berjalan dengan baik.



Berikut hasil proses berserta hasil pencarian cuaca berdasarkan lokasi.

