

# **Laporan Project Mata Kuliah Mobile Computing**



Disusun Oleh :

Tia Anata (2207411012)

**TI-4A  
TEKNIK INFORMATIKA  
TEKNIK INFORMATIKA DAN KOMPUTER  
POLITEKNIK NEGERI JAKARTA  
2024**

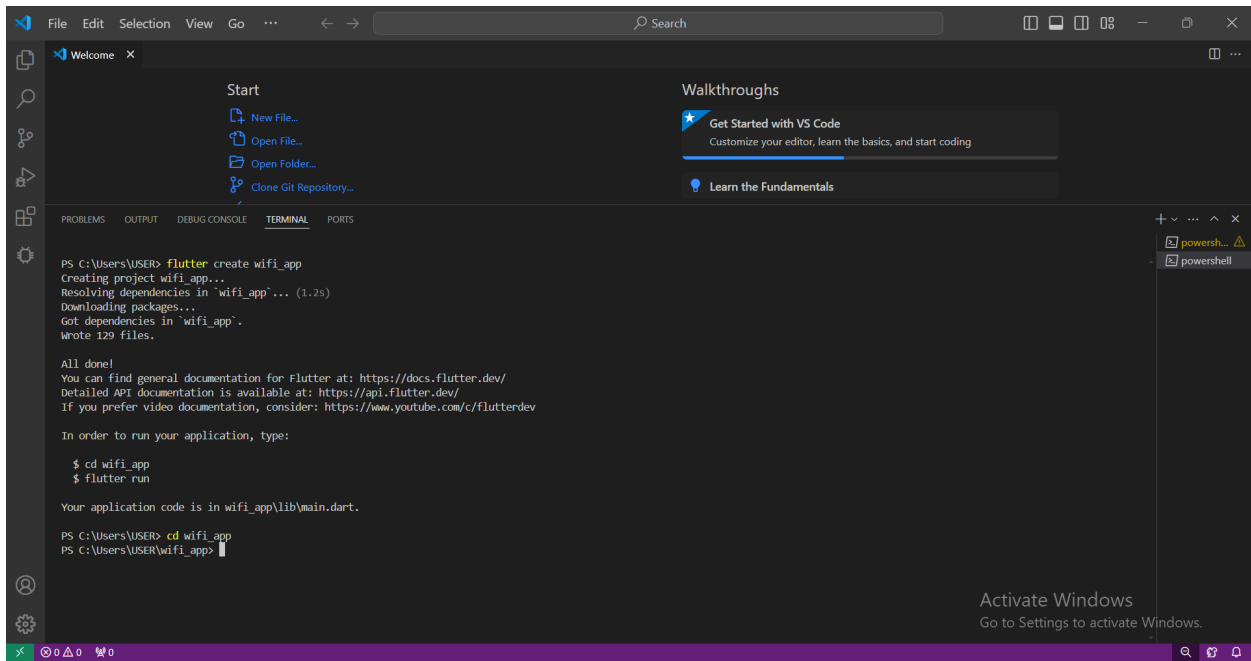
## Prompt VS Code to install Flutter

1. Launch VS Code.
2. To open the **Command Palette**, press **Control + Shift + P**.
3. In the **Command Palette**, type `flutter`.
4. Select **Flutter: New Project**.
5. VS Code prompts you to locate the Flutter SDK on your computer.
  1. If you have the Flutter SDK installed, click **Locate SDK**.
  2. If you do not have the Flutter SDK installed, click **Download SDK**.

This option sends you the Flutter install page if you have not installed Git for Windows as directed in the [development tools prerequisites](#).
6. When prompted **Which Flutter template?**, ignore it. Press **Esc**. You can create a test project after checking your development setup.

Foto diatas adalah cara untuk mengunduh flutter. Langkah pertama instal flutter, choco, dan java terlebih dahulu setelah itu di viscode pilih tanda titik tiga - new terminal dan ketik “flutter create wifi\_app” dan kemudian enter

### 1. Bagian wifi:



```
File Edit Selection View Go ... Search
Welcome x

Start
New File...
Open File...
Open Folder...
Clone Git Repository...

Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\USER> flutter create wifi_app
Creating project wifi_app...
Resolving dependencies in 'wifi_app'... (1.2s)
Downloading packages...
Got dependencies in 'wifi_app'.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/Flutterdev

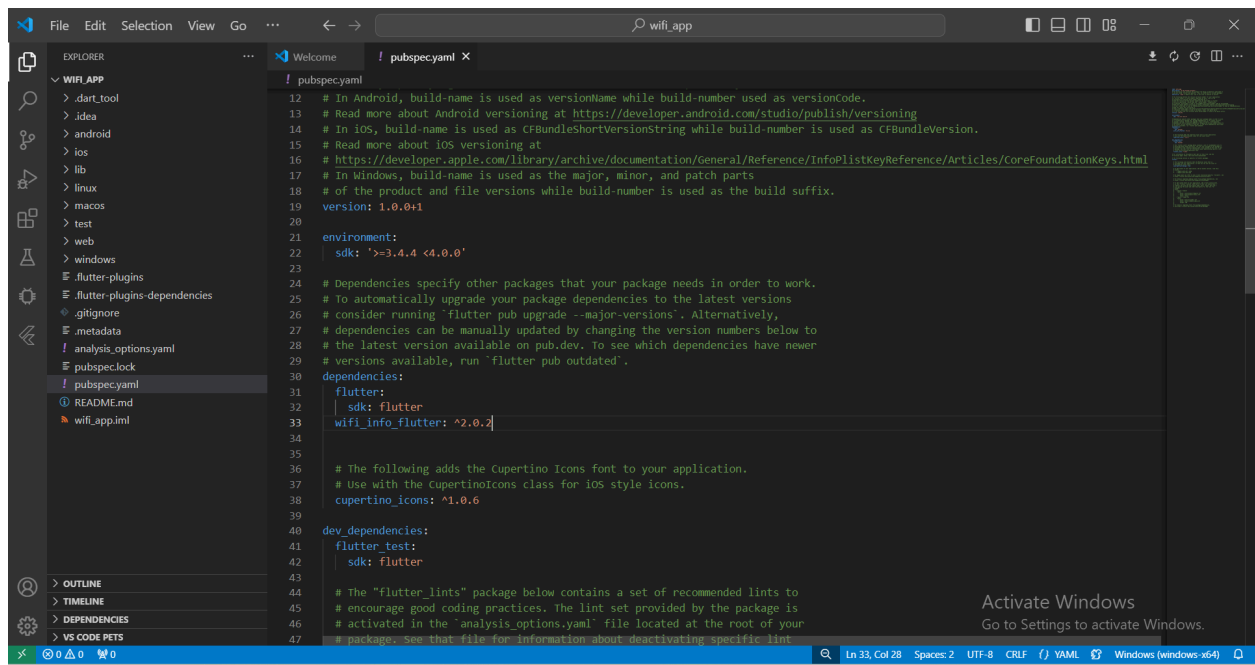
In order to run your application, type:

$ cd wifi_app
$ flutter run

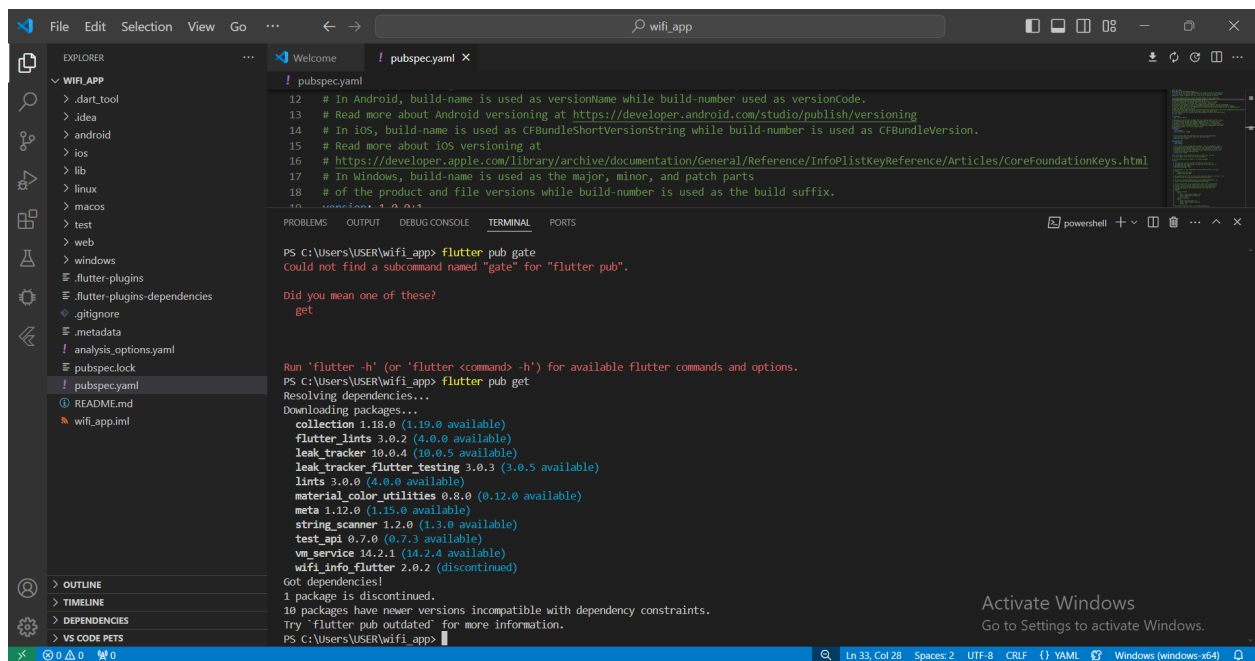
Your application code is in wifi_app\lib\main.dart.

PS C:\Users\USER> cd wifi_app
PS C:\Users\USER>wifi_app>
```

Setelah berhasil buka file - new file - ke folder C - users - user dan cari wifi\_app dan klik open folder.

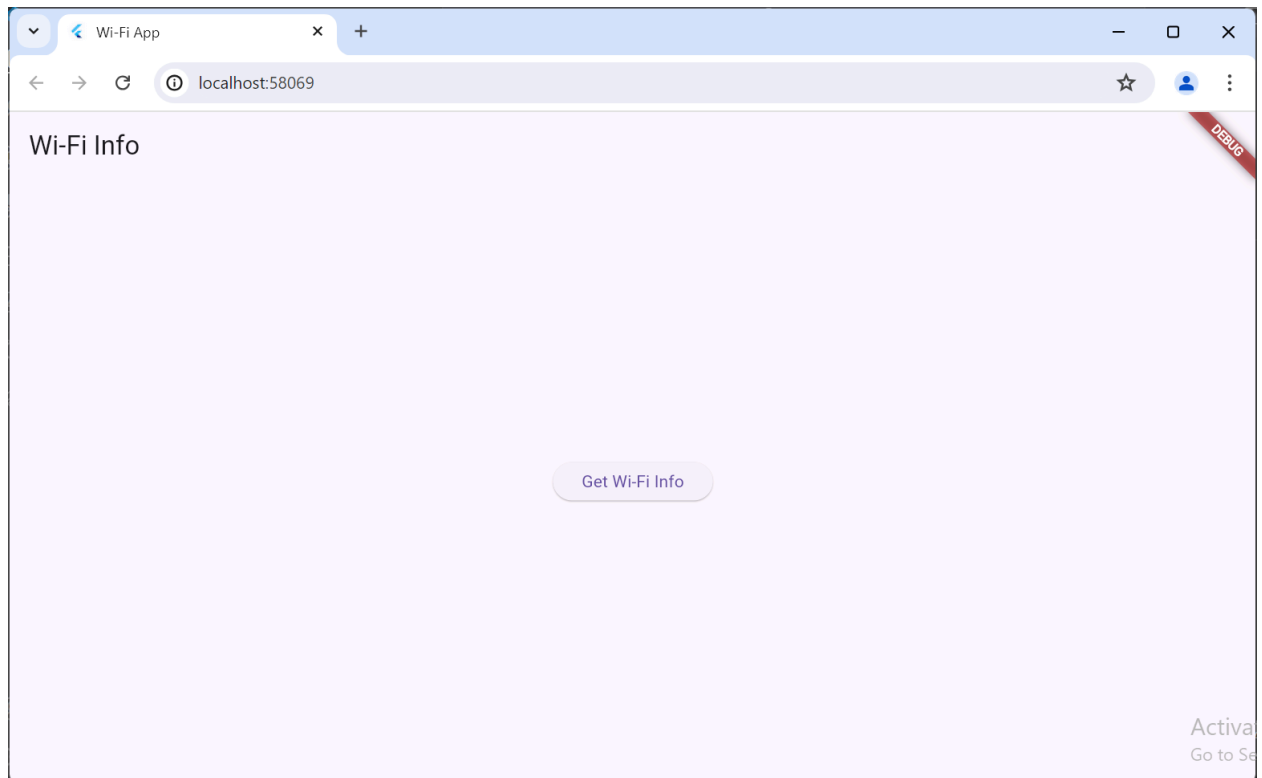


Setelah terbuka pilih file `pubspec.yaml` dan masukkan “`wifi_info_flutter: ^2.0.2` dan `ctrl + s` untuk menyimpan.

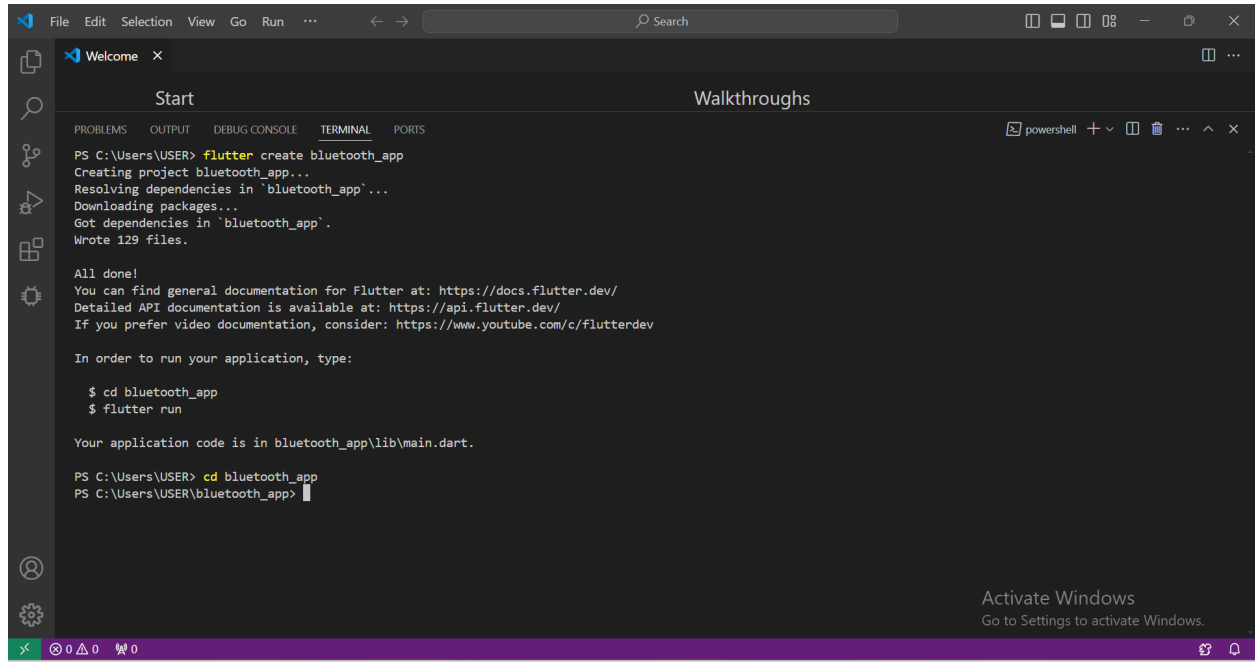


Setelah itu pilih tanda titik tiga di atas - terminal - new terminal - dan ketik “flutter pub get” dan enter. Lalu setelah itu ketik lagi “terminal run” dan enter lagi.

Hasil output:



## 2. Bagian Bluetooth :



```
PS C:\Users\USER> flutter create bluetooth_app
Creating project bluetooth_app...
Resolving dependencies in `bluetooth_app`...
Downloading packages...
Got dependencies in `bluetooth_app`.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev

In order to run your application, type:

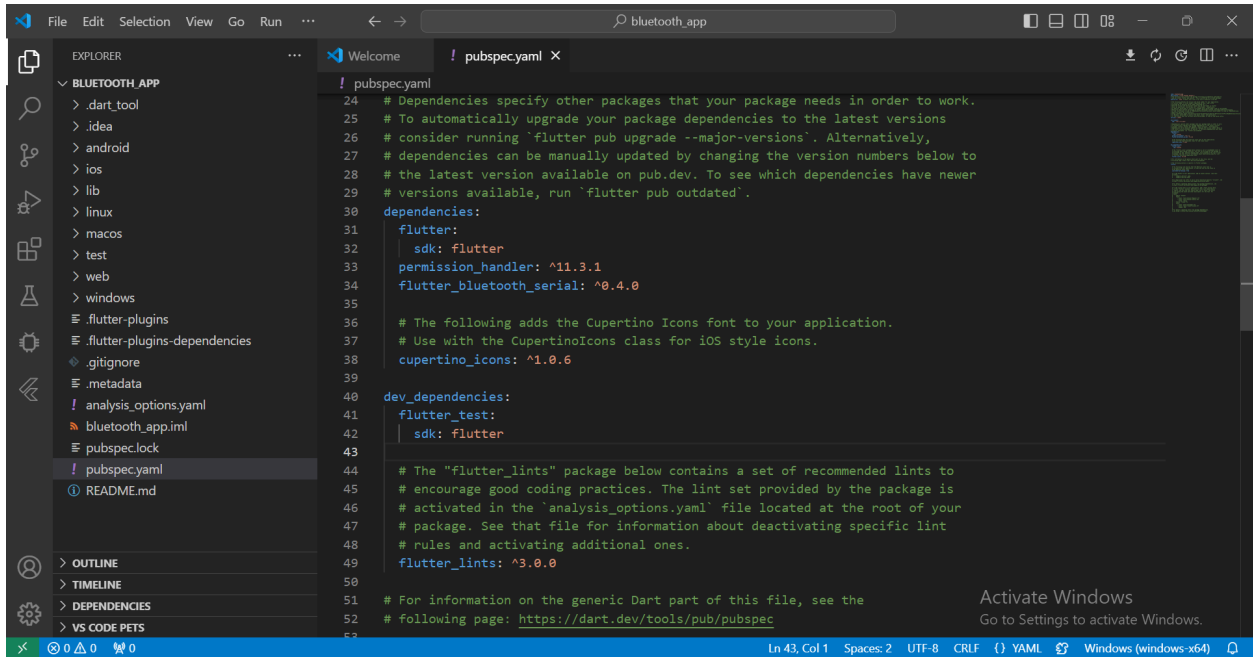
$ cd bluetooth_app
$ flutter run

Your application code is in bluetooth_app\lib\main.dart.

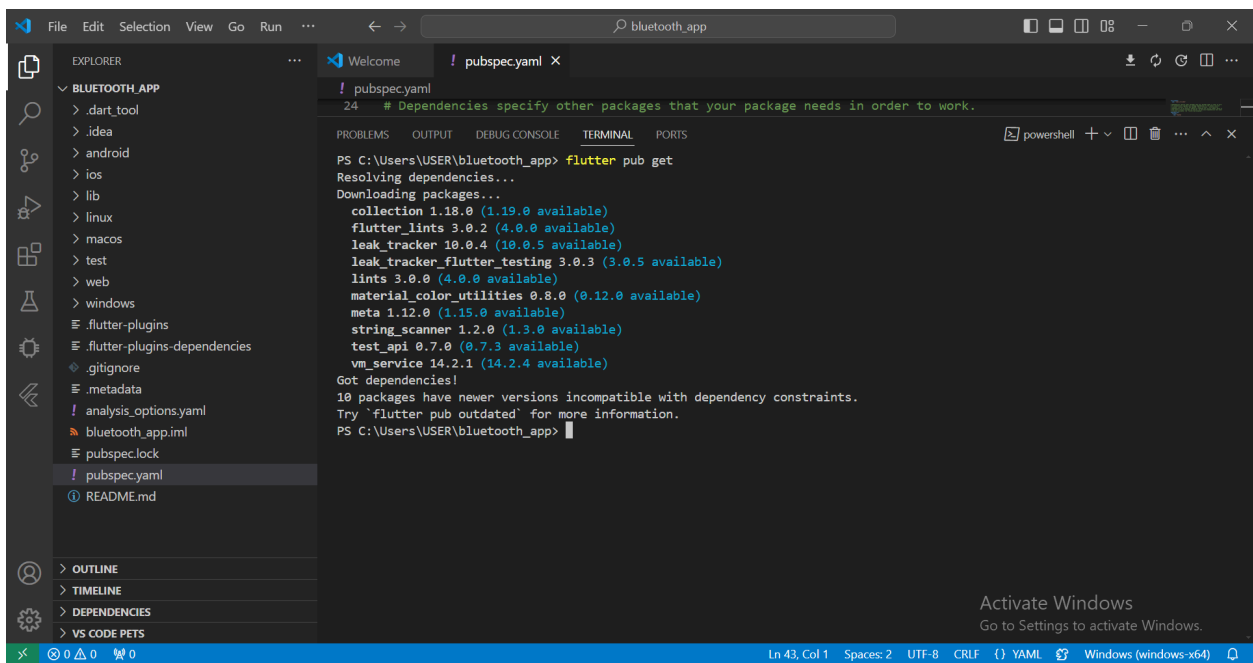
PS C:\Users\USER> cd bluetooth_app
PS C:\Users\USER\bluetooth_app>
```

Untuk bagian bluetooth langkah menginstallnya juga sama seperti wifi. File - new windows - terminal - new terminal - lalu ketik “flutter create bluetooth\_app” dan enter.

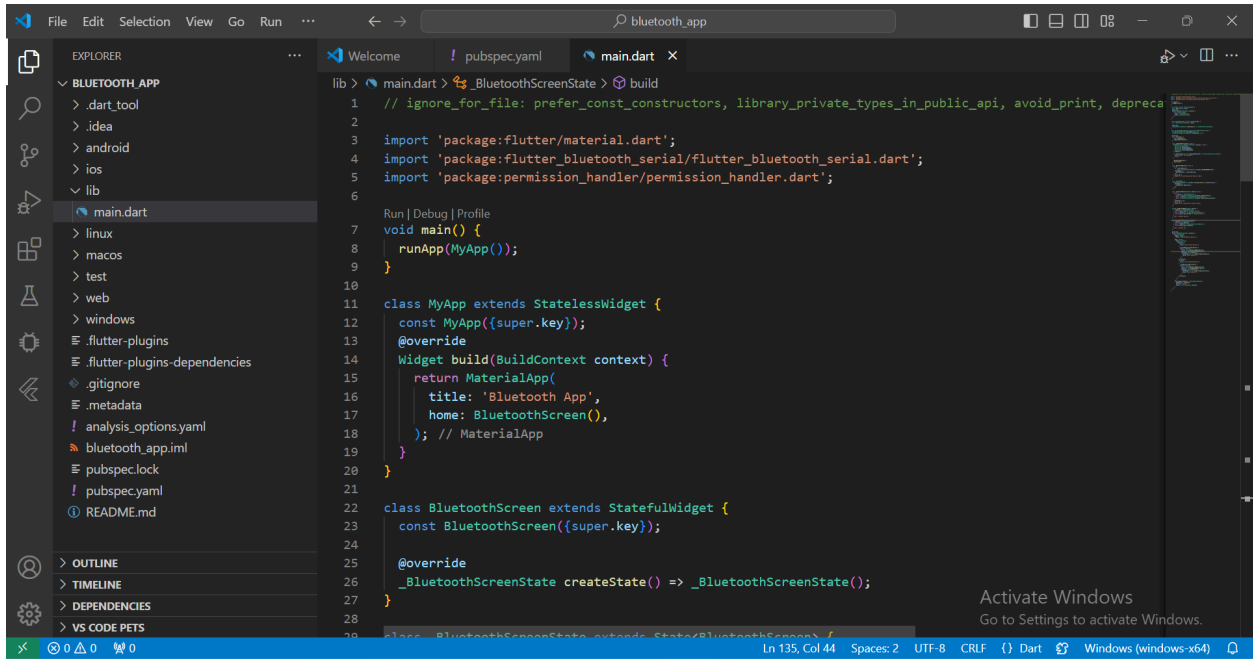
Selanjutnya ketik “cd bluetooth\_app” dan enter. Jika sudah buka file C-USERS-USER- dan cari bluetooth\_app”



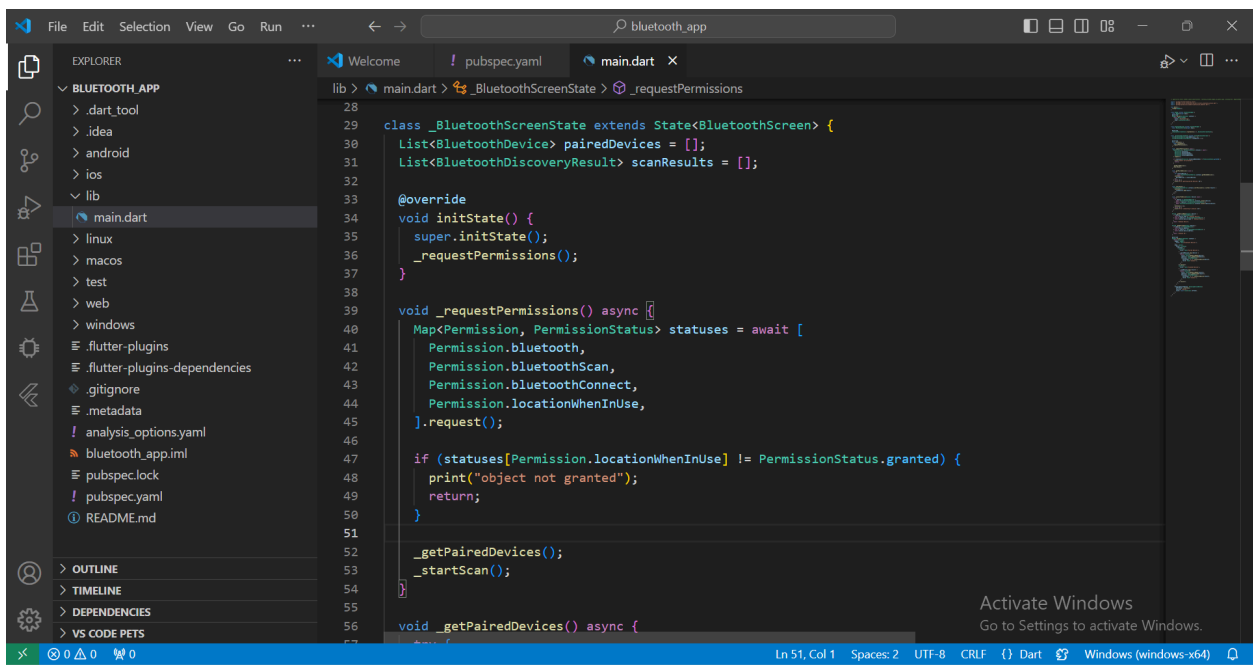
Pada file pubspec.yaml tambahkan code yang ada di line 33-34 dan sejajarkan dengan dependencies.



Klik tanda tiga lagi - terminal - new terminal lalu ketik “flutter pub get” dan enter.



Pada folder lib - main.dart lalu masukkan code yang tertera digambar. Lalu ctrl + S



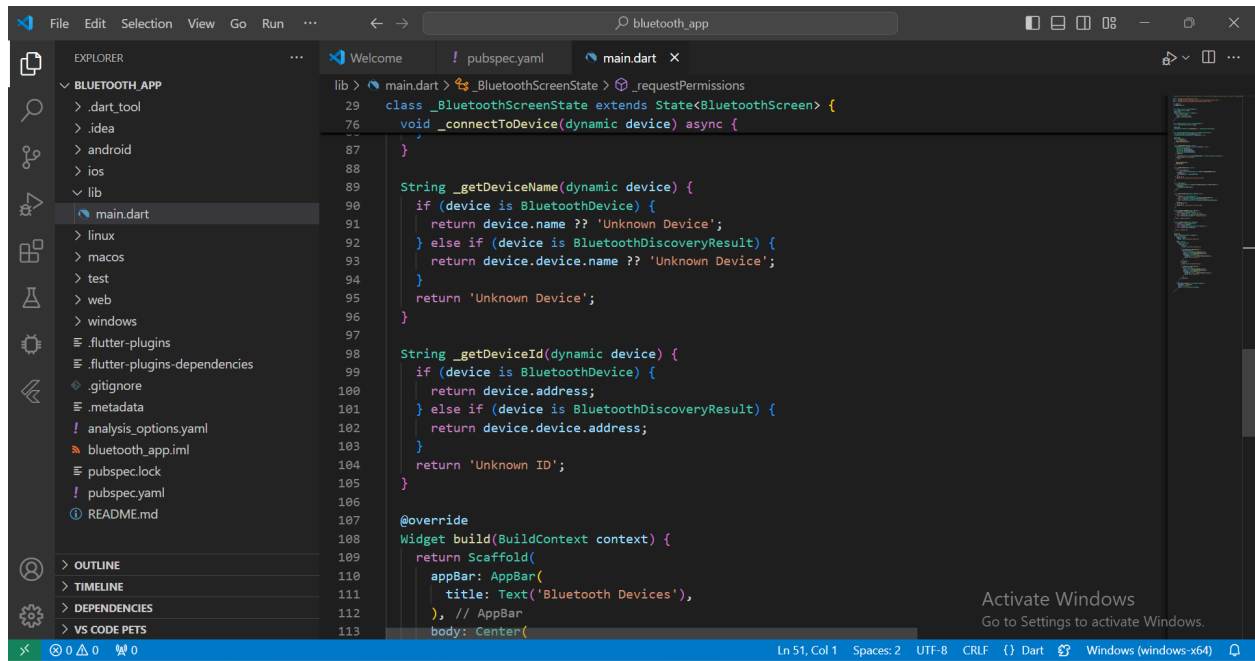
```
lib > main.dart > _BluetoothScreenState > _requestPermissions
29 class _BluetoothScreenState extends State<BluetoothScreen> {
56 void _getPairedDevices() async {
57   try {
58     var classicDevices =
59       await FlutterBluetoothSerial.instance.getBondedDevices();
60     setState(() {
61       pairedDevices = classicDevices;
62     });
63   } catch (e) {
64     print("Error getting paired devices: $e");
65   }
66 }
67
68 void _startScan() {
69   FlutterBluetoothSerial.instance.startDiscovery().listen((result) {
70     setState(() {
71       scanResults.add(result);
72     });
73   });
74 }
75
76 void _connectToDevice(dynamic device) async {
77   try {
78     if (device is BluetoothDevice) {
79       await FlutterBluetoothSerial.instance.connect(device);
80     } else if (device is BluetoothDiscoveryResult) {
81       await FlutterBluetoothSerial.instance.connect(device.device);
82     }
83     setState(() {});
84   } catch (e) {
85     print("Error connecting to device: $e");
86   }
87 }
```

Activate Windows  
Go to Settings to activate Windows.

```
lib > main.dart > _BluetoothScreenState > _requestPermissions
29 class _BluetoothScreenState extends State<BluetoothScreen> {
56 void _getPairedDevices() async {
57   try {
58     var classicDevices =
59       await FlutterBluetoothSerial.instance.getBondedDevices();
60     setState(() {
61       pairedDevices = classicDevices;
62     });
63   } catch (e) {
64     print("Error getting paired devices: $e");
65   }
66 }
67
68 void _startScan() {
69   FlutterBluetoothSerial.instance.startDiscovery().listen((result) {
70     setState(() {
71       scanResults.add(result);
72     });
73   });
74 }
75
76 void _connectToDevice(dynamic device) async {
77   try {
78     if (device is BluetoothDevice) {
79       await FlutterBluetoothSerial.instance.connect(device);
80     } else if (device is BluetoothDiscoveryResult) {
81       await FlutterBluetoothSerial.instance.connect(device.device);
82     }
83     setState(() {});
84   } catch (e) {
85     print("Error connecting to device: $e");
86   }
87 }
```

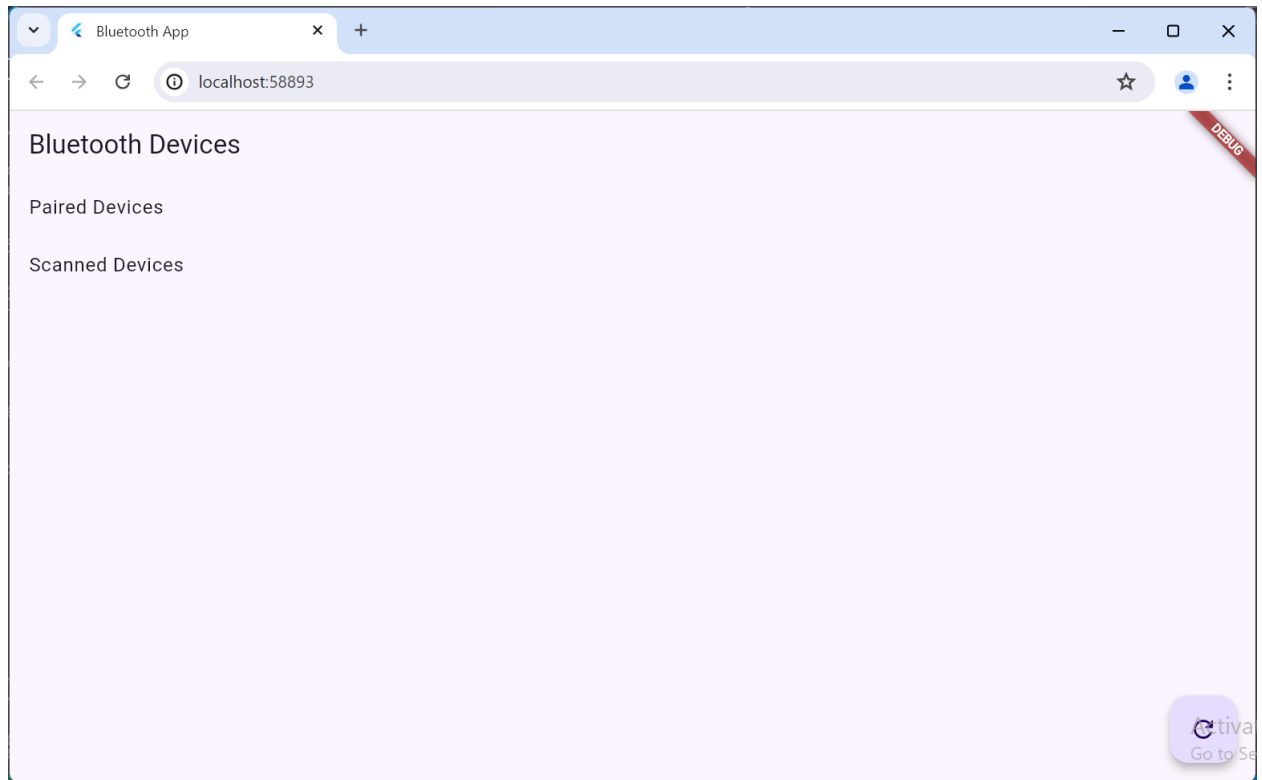
Activate Windows  
Go to Settings to activate Windows.





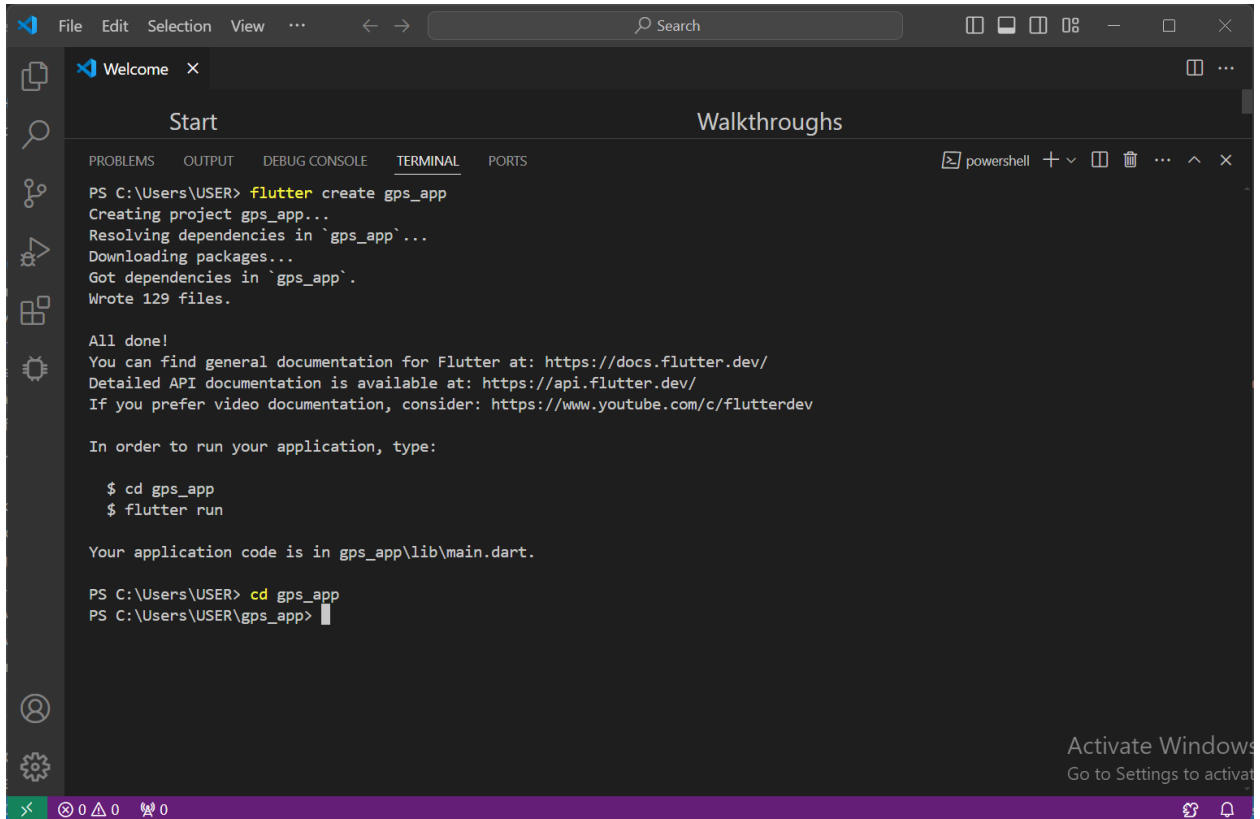
Jika sudah selesai lakukan langkah-langkah seperti `wifi_app`. Klik tanda tiga diatas - terminal - new terminal - dan ketik “flutter run” lalu enter

## Hasil Output:



Note: ketika ingin melakukan “Run” harus menyalakan bluetooth yang ada pada laptop. Jika perangkat laptop pernah terhubung pada hp atau perangkat lain selain earphone akan ada tulisannya pada hasil output.

### 3. Bagian Gps



The screenshot shows the Visual Studio Code interface with the terminal window open. The terminal output indicates that the Flutter project 'gps\_app' has been successfully created. The steps shown are: running 'flutter create gps\_app', resolving dependencies, downloading packages, and writing 129 files. The terminal also provides links to Flutter documentation and instructions on how to run the application. The user has navigated to the 'gps\_app' directory using 'cd gps\_app'.

```
PS C:\Users\USER> flutter create gps_app
Creating project gps_app...
Resolving dependencies in `gps_app`...
Downloading packages...
Got dependencies in `gps_app`.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev

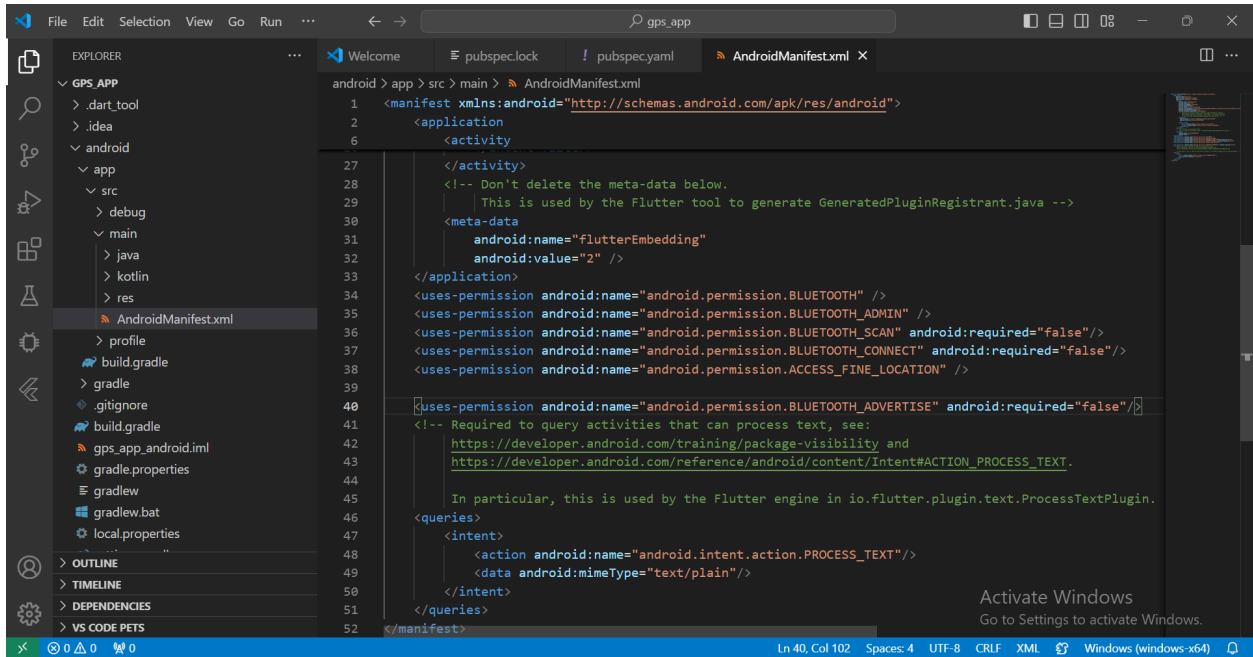
In order to run your application, type:

$ cd gps_app
$ flutter run

Your application code is in gps_app\lib\main.dart.

PS C:\Users\USER> cd gps_app
PS C:\Users\USER\gps_app>
```

Pilih tanda titik tiga diatas - terminal - new terminal - ketik “flutter create gps\_app” lalu enter. Kemudian dibawahnya Ketik lagi “cd gps\_app” lalu enter



**Klik android - app - src - main - androidManifest.xml dibagian bawah application masukkan code berikut ini.**

```
<uses-permission android:name="android.permission.BLUETOOTH" />
  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"
/>
  <uses-permission android:name="android.permission.BLUETOOTH_SCAN"
android:required="false"/>
  <uses-permission
android:name="android.permission.BLUETOOTH_CONNECT"
android:required="false"/>
  <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />

  <uses-permission
android:name="android.permission.BLUETOOTH_ADVERTISE"
android:required="false"/>
```

**Jika sudah ke bagian meta-data dan tambahkan code berikut.**

```
<meta-data
```

```
android:name="com.google.android.geo.API_KEY"  
android:value="AIzaSyBQEuIXMJseti4EmnL1LvIizXQfCTrPeF8"/>
```

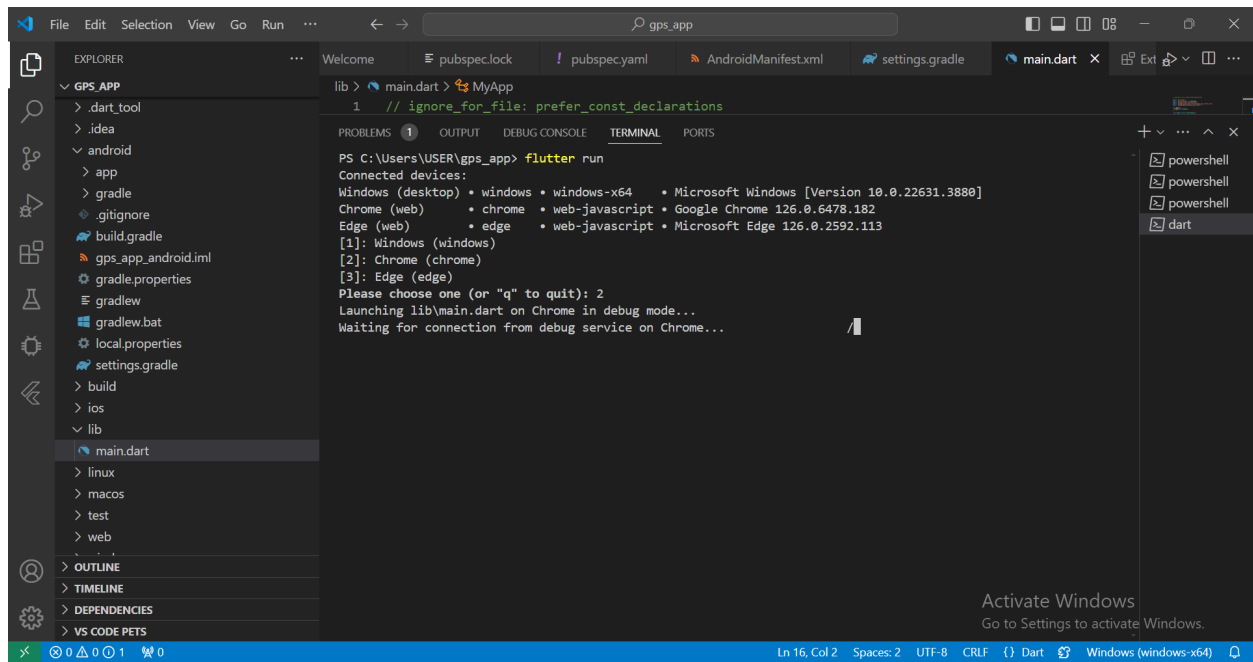
```
<uses-permission android:name="android.permission.INTERNET"/>
```

```
<uses-permission
```

```
android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
<uses-permission
```

```
android:name="android.permission.ACCESS_COARSE_LOCATION" />
```



Output:

