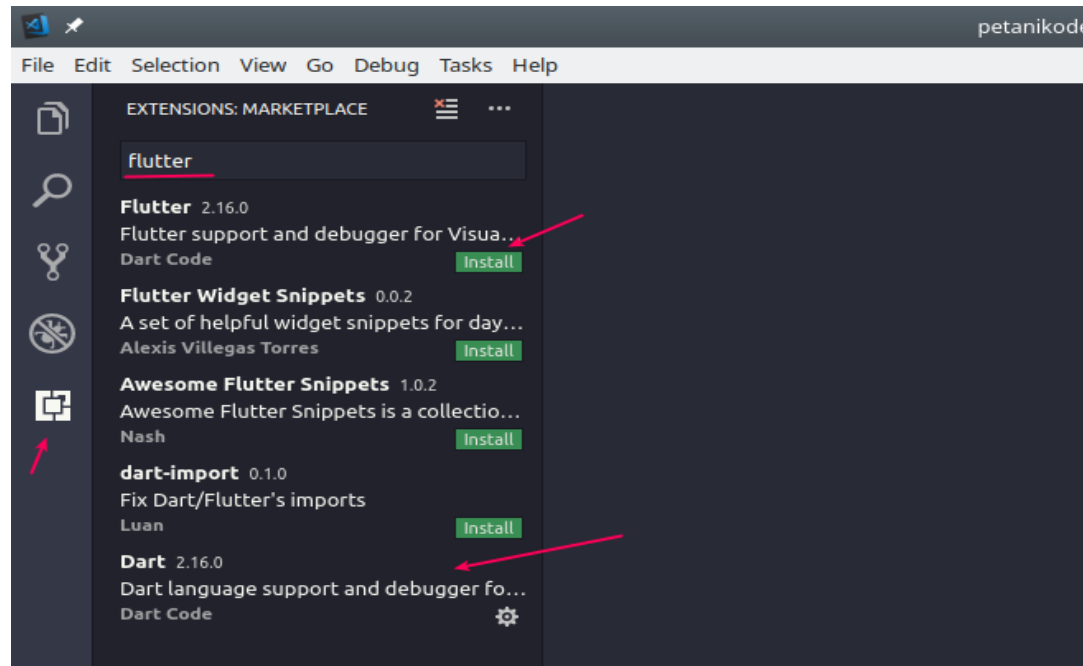


PERTEMUAN 3

Membuat dan Menjalankan Proyek Dengan VSCode dan Handphone Android

Install VSCode Sebagai Alternatif Editor

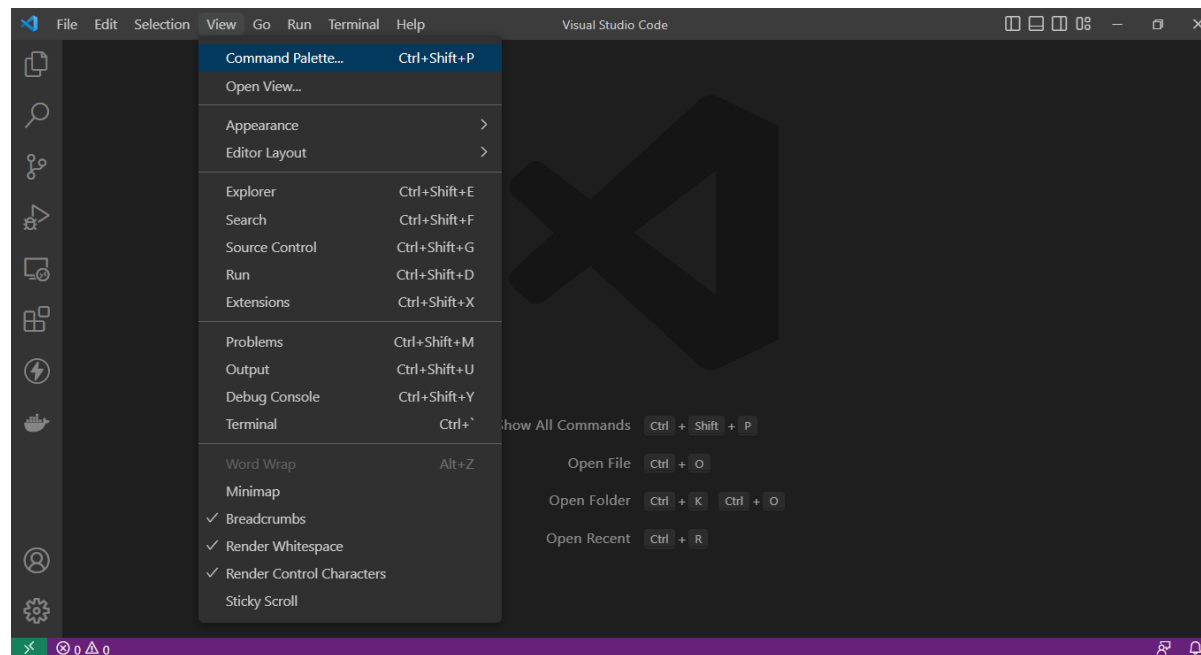
Unduh VSCode pada laman <https://code.visualstudio.com/download> kemudian install. Agar flutter dapat digunakan pada VSCode, perlu diinstall beberapa extension flutter yang dibutuhkan.

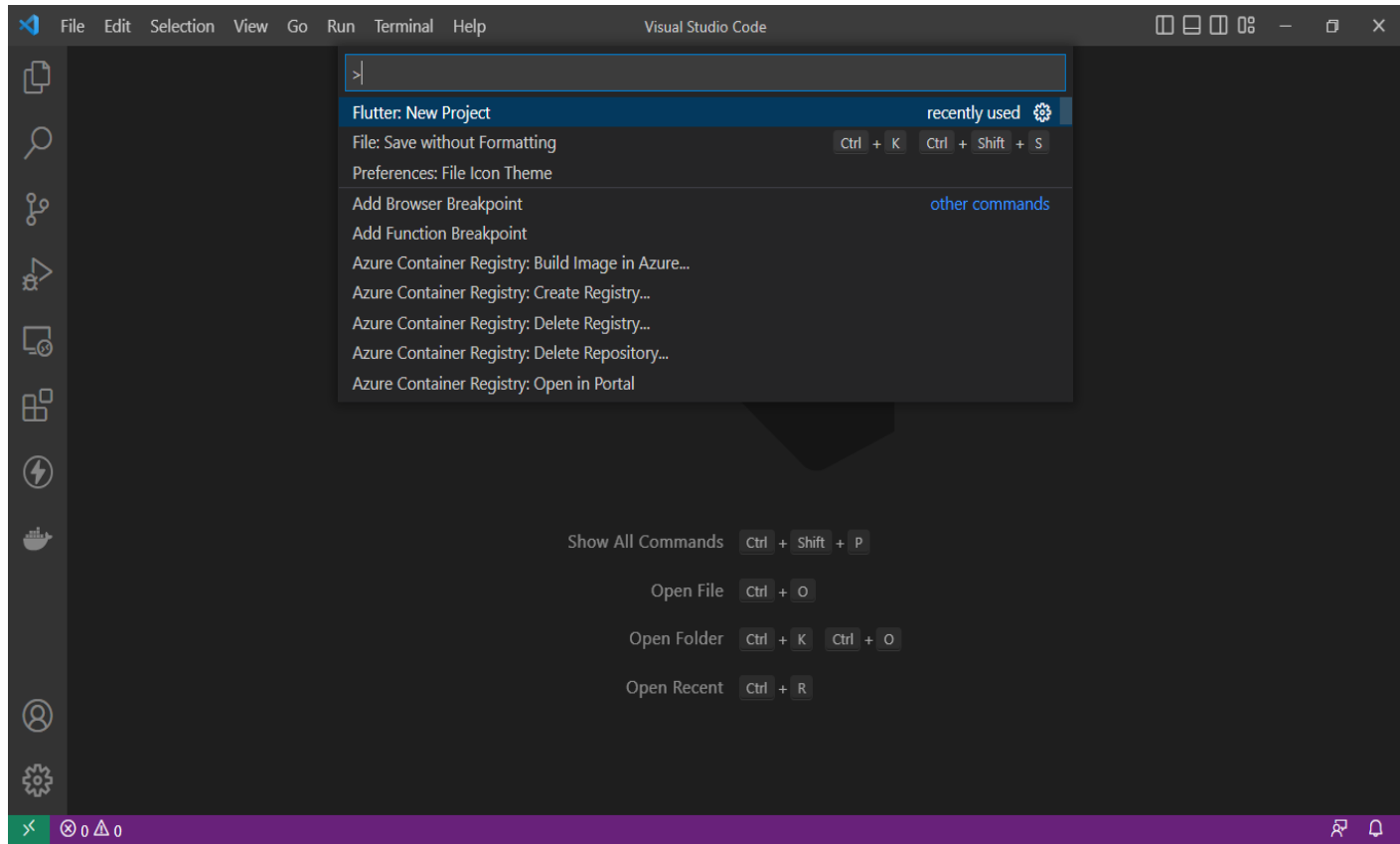


Setelah itu restart/tutup VSCode

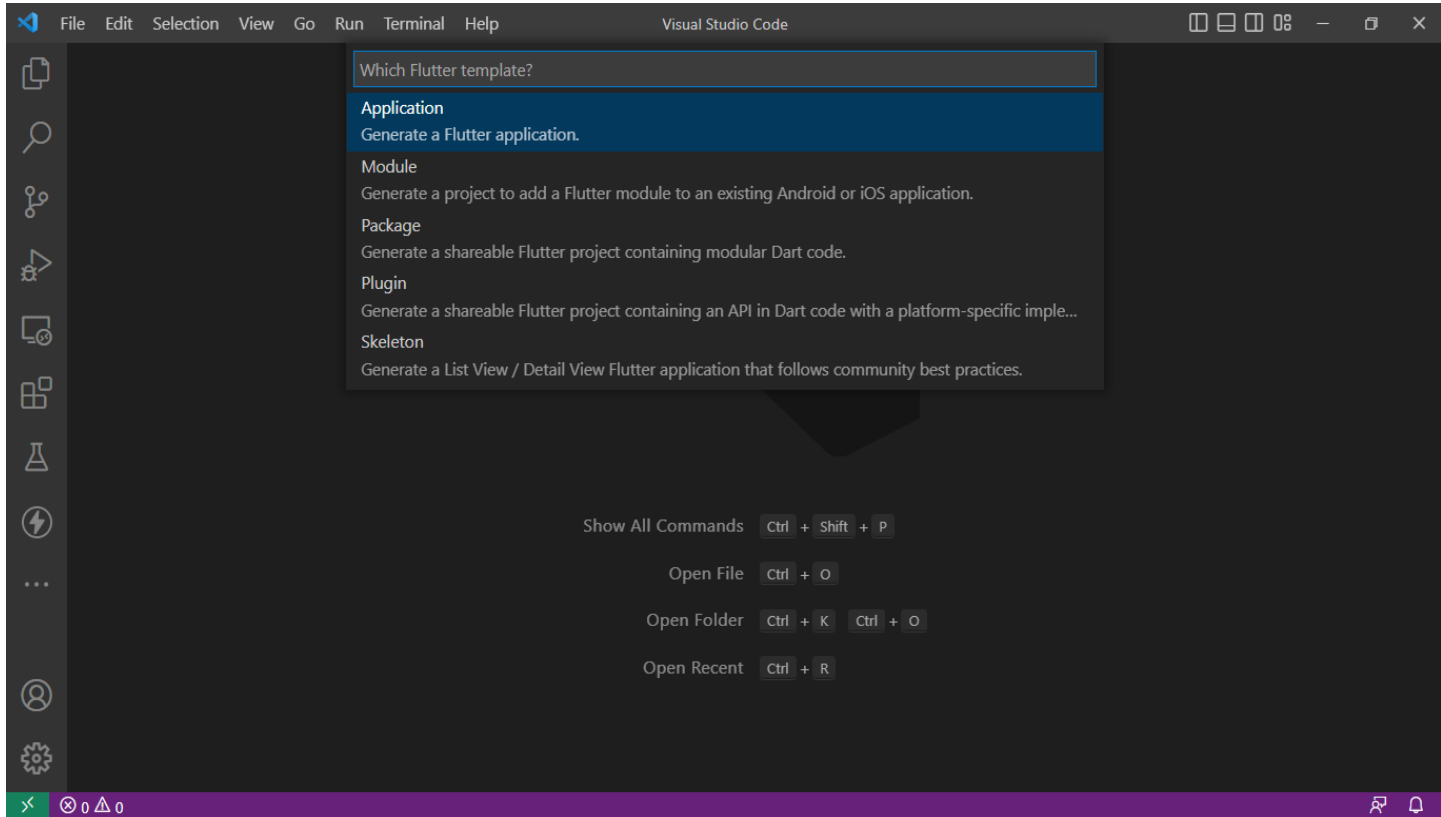
Membuat proyek flutter dengan VSCode

Jalankan VSCode, pada menubar pilih **view -> command Palette...** atau dapat juga dengan shortcut **Ctrl + Shift + P**

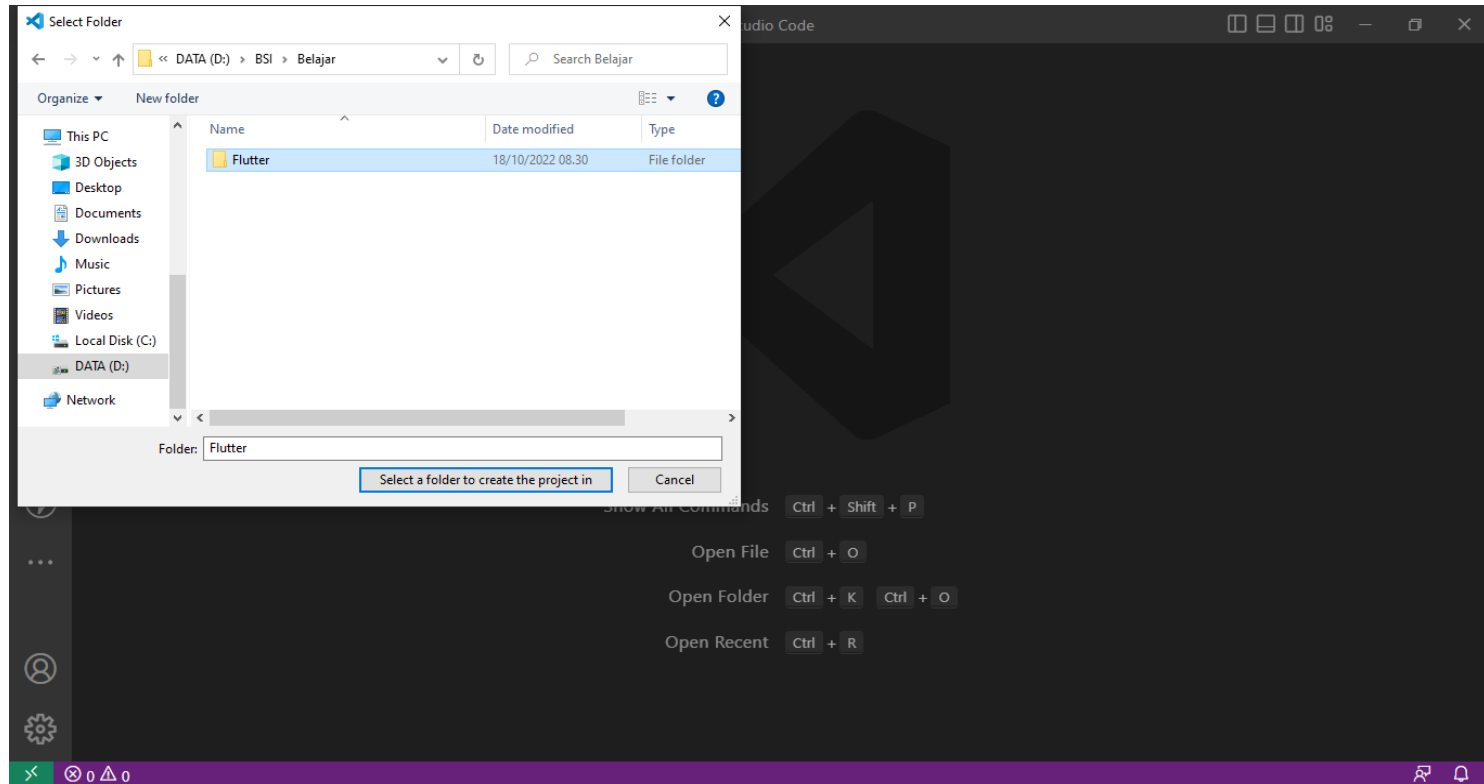




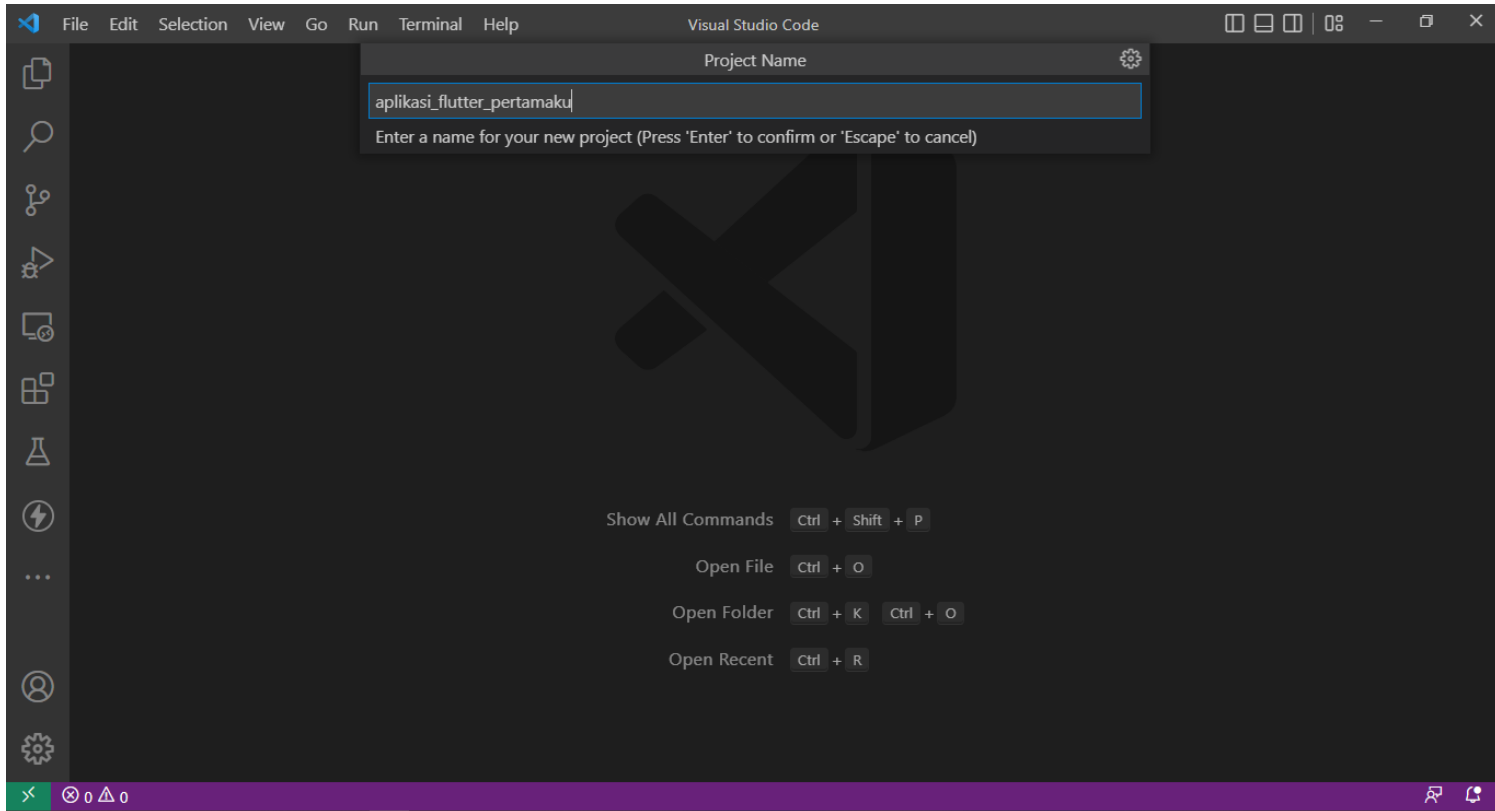
Kemudian ketikkan **flutter** dan pilih **Flutter: New Project**



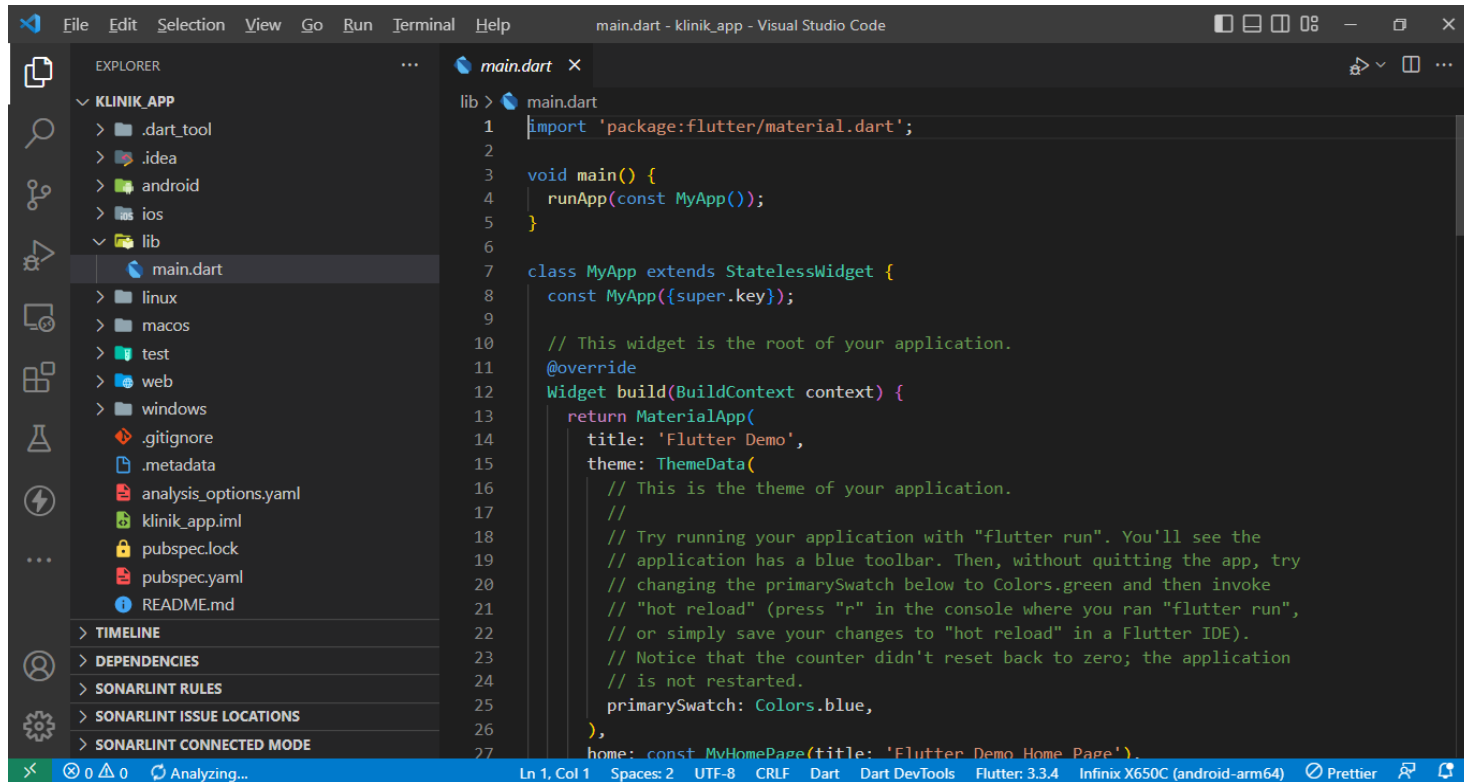
Kemudian pilih **Application**



Pilih folder tempat proyek tersebut



Kemudian tentukan nama proyek flutter yang ingin dibuat dengan nama
aplikasi_flutter_pertamaku



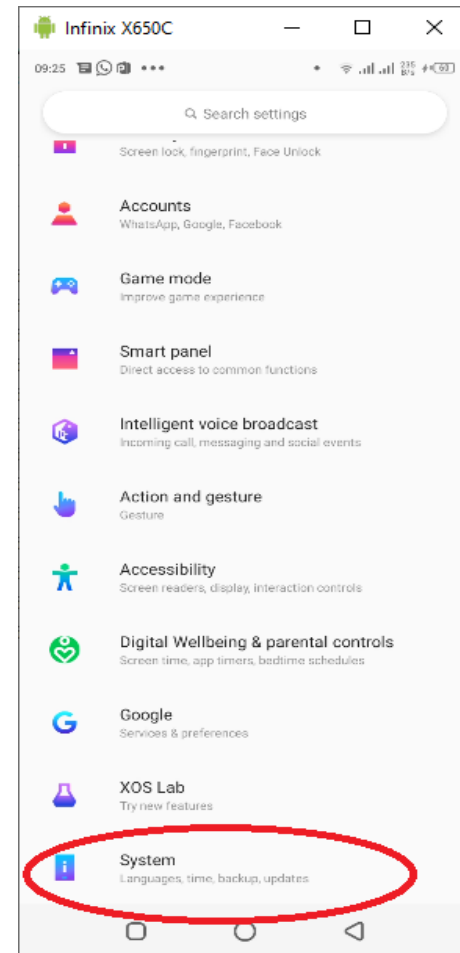
```
lib > main.dart
1  import 'package:flutter/material.dart';
2
3  void main() {
4    runApp(const MyApp());
5  }
6
7  class MyApp extends StatelessWidget {
8    const MyApp({super.key});
9
10   // This widget is the root of your application.
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'Flutter Demo',
15       theme: ThemeData(
16         // This is the theme of your application.
17         //
18         // Try running your application with "flutter run". You'll see the
19         // application has a blue toolbar. Then, without quitting the app, try
20         // changing the primarySwatch below to Colors.green and then invoke
21         // "hot reload" (press "r" in the console where you ran "flutter run",
22         // or simply save your changes to "hot reload" in a Flutter IDE).
23         // Notice that the counter didn't reset back to zero; the application
24         // is not restarted.
25         primarySwatch: Colors.blue,
26       ),
27       home: const MvHomePage(title: 'Flutter Demo Home Page'),
28     );
29   }
30 }
```

Kemudian tekan **Enter** dan tunggu hingga proses unduhan selesai

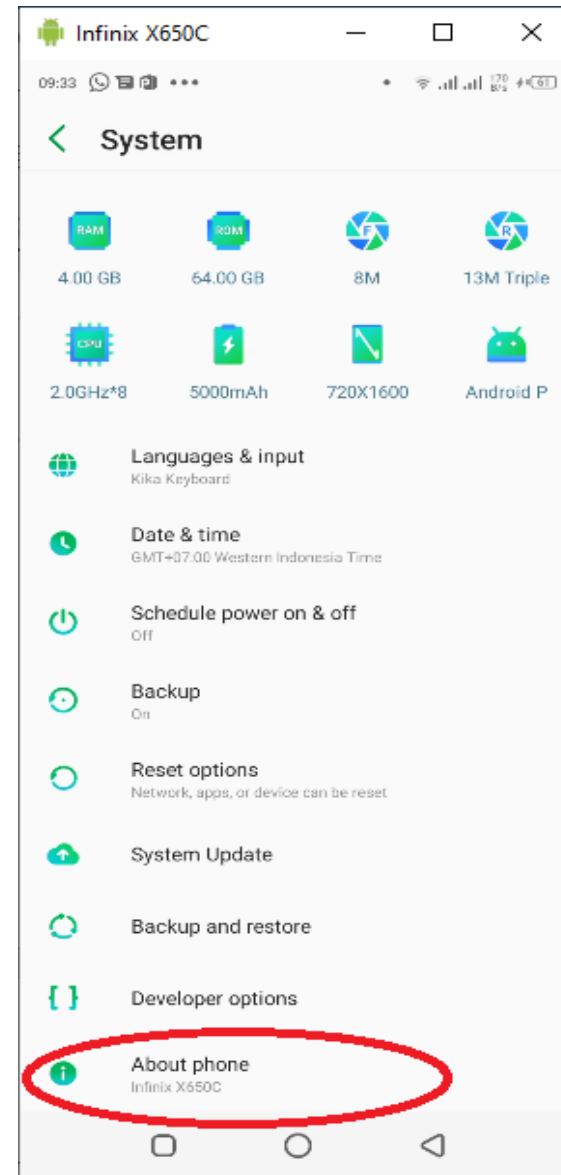
Menjalankan aplikasi dengan Handphone Android

Untuk menjalankan proyek flutter dari VSCode dapat menggunakan Emulator AVD yang telah dibuat sebelumnya menggunakan Android Studio ataupun menggunakan Device Handphone Android langsung.

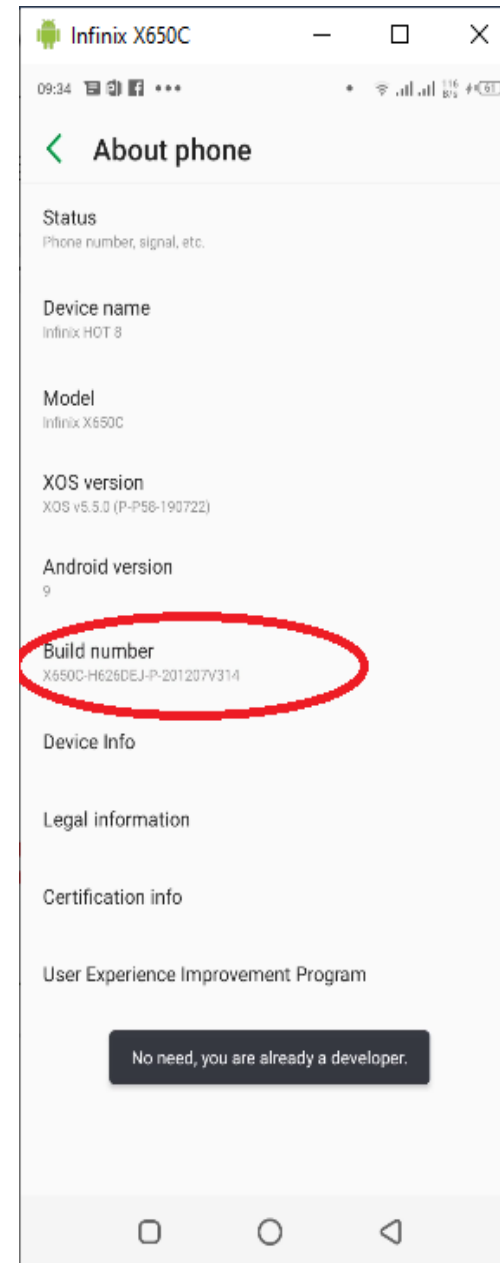
Untuk menggunakan android device secara langsung, pertama aktifkan dulu mode developer dengan cara buka **Setting** kemudian pilih **System** kemudian pilih **About Phone**, untuk masing-masing device mungkin terdapat perbedaan untuk lokasi **About Phone** ada pula yang berada pada **Additional Setting**



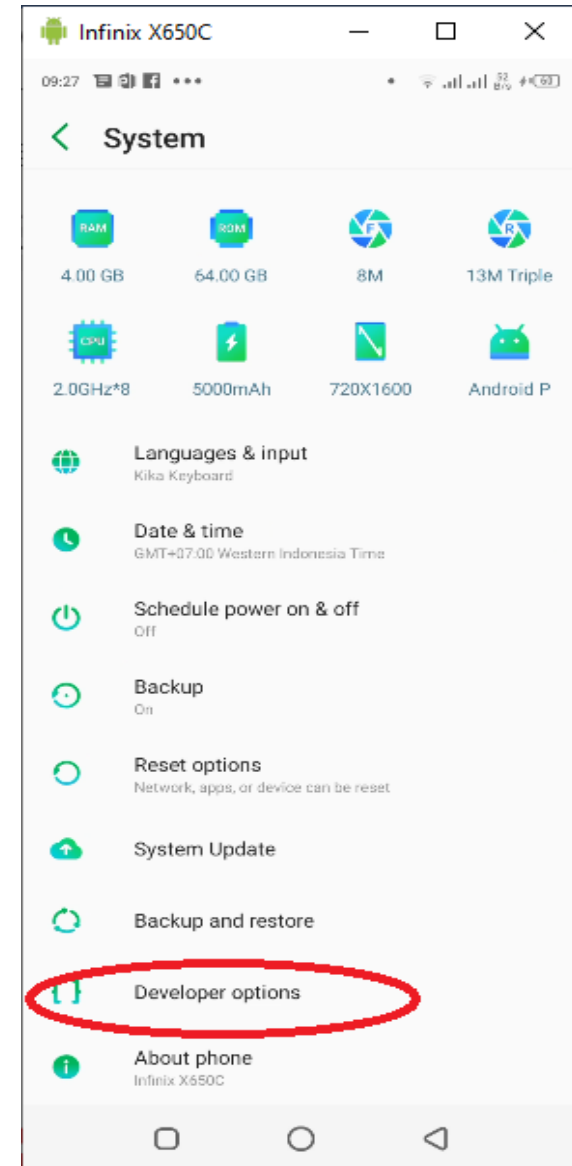
Kemudian pilih **About Phone**



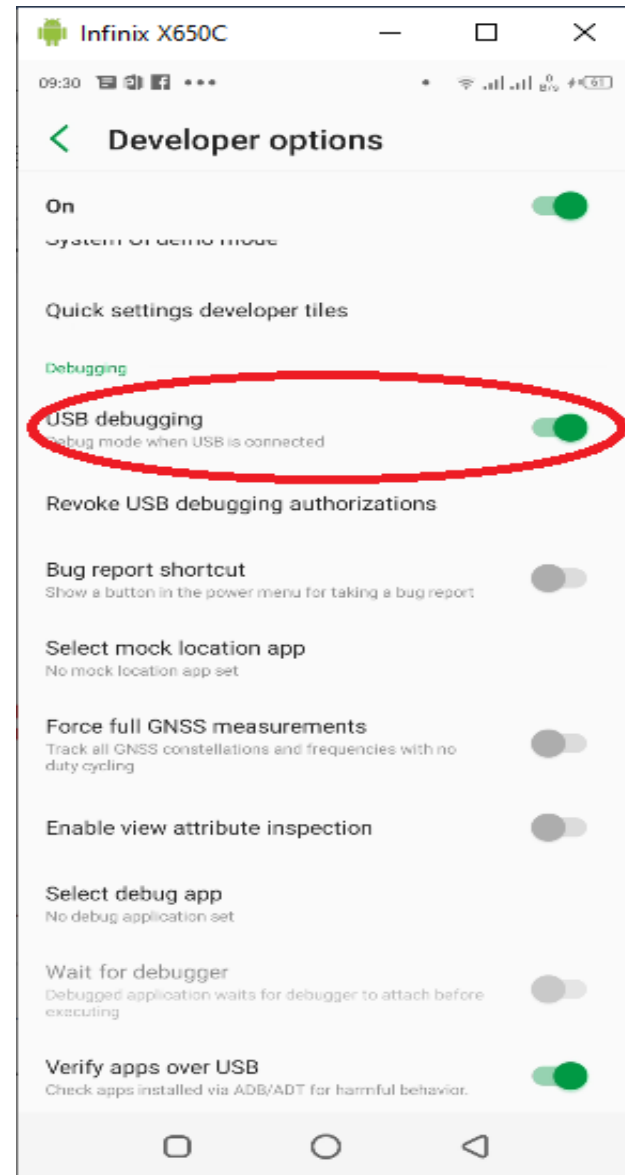
Kemudian ketuk **Build number** beberapa kali, namun ini juga berbeda untuk beberapa versi misalnya untuk Xiaomi dengan mengetuk **MIUI Version** beberapa kali



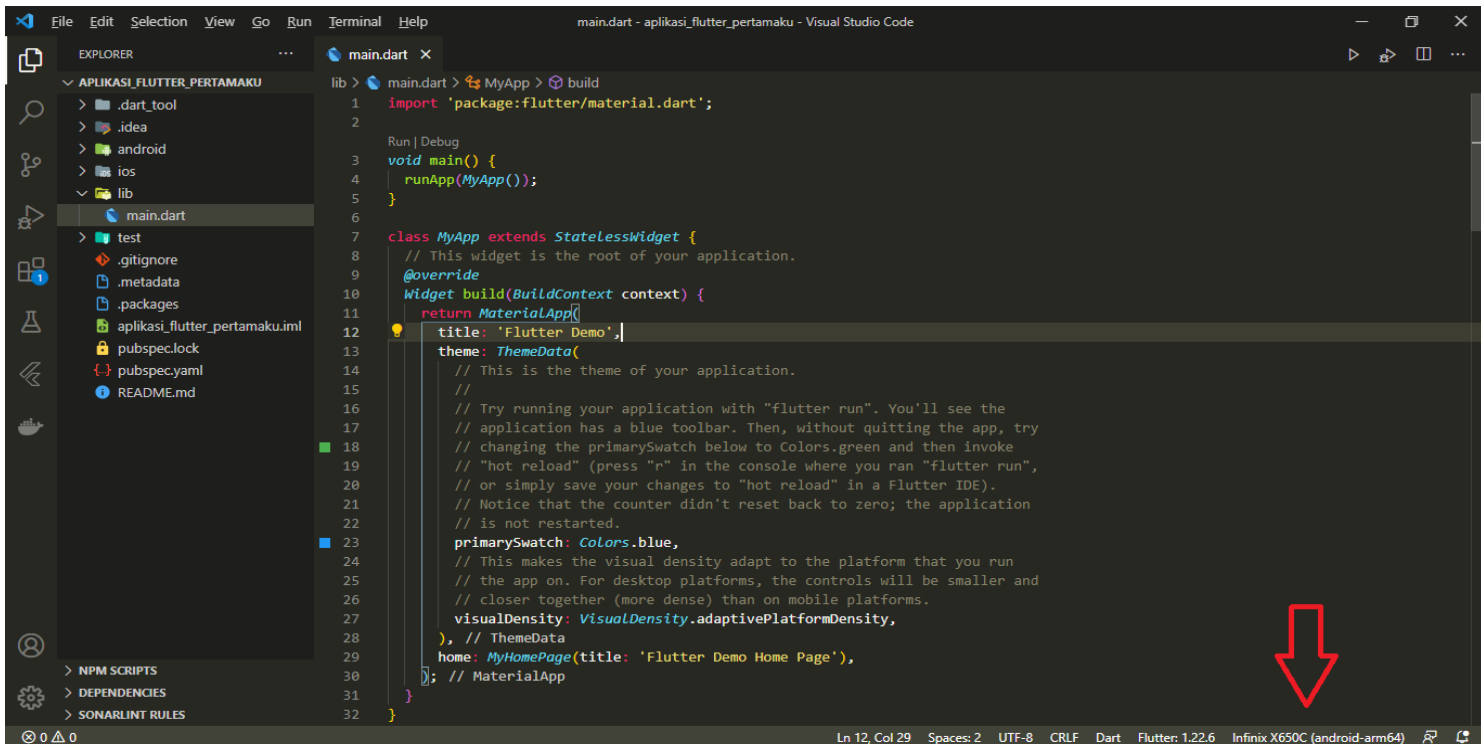
Selanjutnya mengaktifkan USB Debugger dengan cara pilih **Developer Option** pada **System**, **Developer Option** ini akan muncul setelah mode Developer diaktifkan dengan cara diatas



Kemudian aktifkan **USB Debugging**



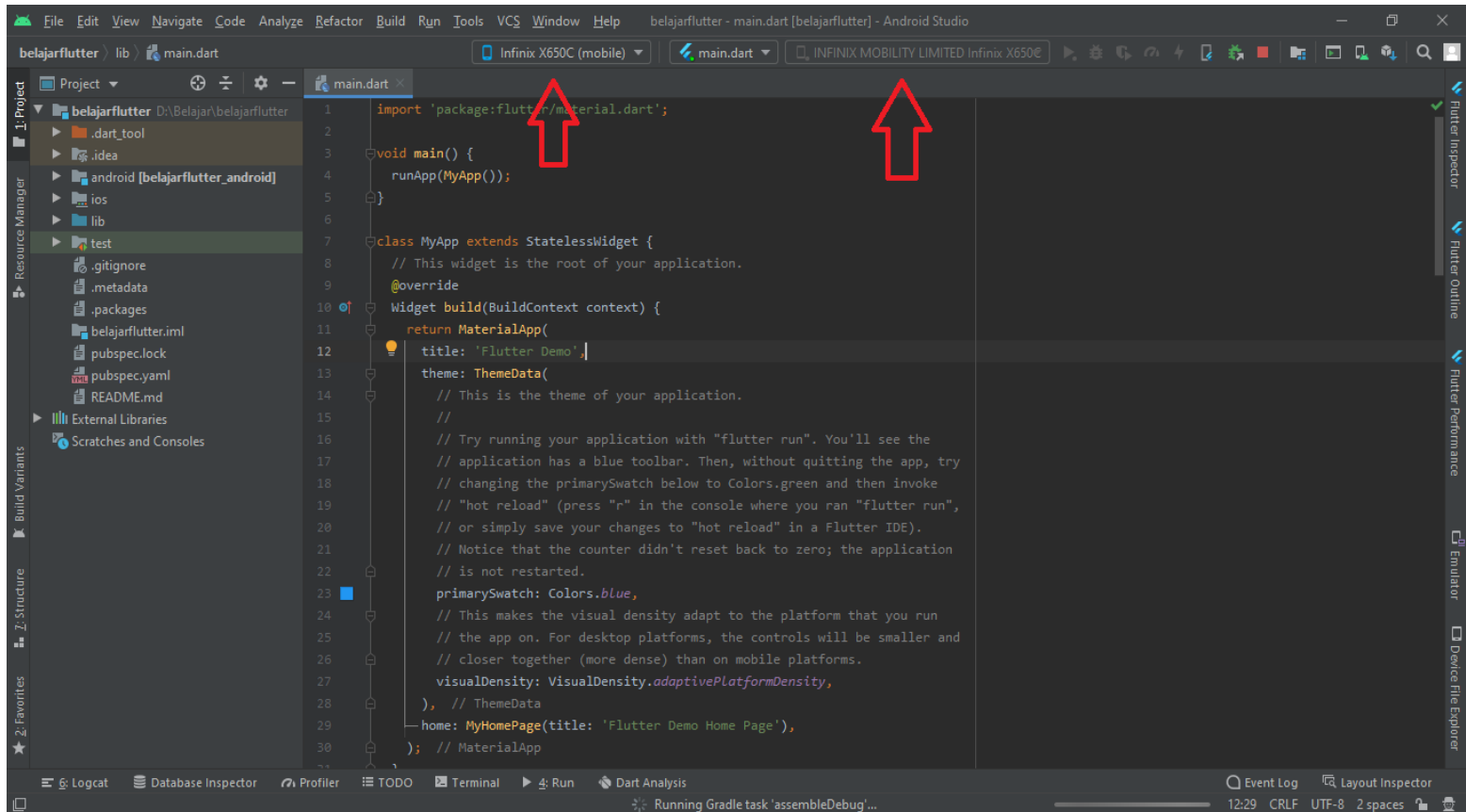
Jika telah selesai, hubungkan Handphone android dengan laptop/komputer dengan kabel data, untuk memeriksa apakah sudah terhubung dengan Handphone, dapat dilihat pada VSCode bagian pojok kanan bawah akan tertera nama device yang terhubung



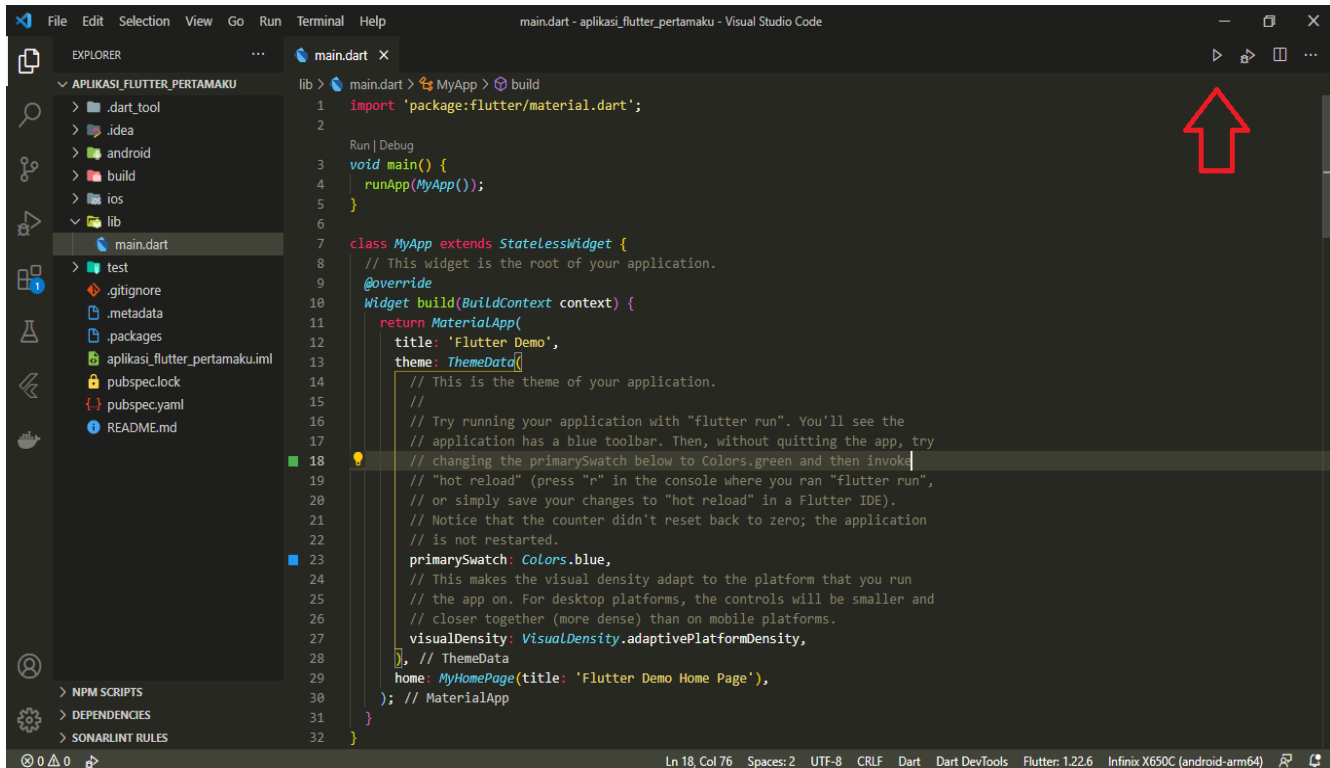
```
lib > main.dart > runApp > build
1 import 'package:flutter/material.dart';
2
3 Run | Debug
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   // This widget is the root of your application.
10  @override
11  Widget build(BuildContext context) {
12    return MaterialApp(
13      title: 'Flutter Demo',
14      theme: ThemeData(
15        // This is the theme of your application.
16        //
17        // Try running your application with "flutter run". You'll see the
18        // application has a blue toolbar. Then, without quitting the app, try
19        // changing the primarySwatch below to Colors.green and then invoke
20        // "hot reload" (press "r" in the console where you ran "flutter run",
21        // or simply save your changes to "hot reload" in a Flutter IDE).
22        // Notice that the counter didn't reset back to zero; the application
23        // is not restarted.
24        primarySwatch: Colors.blue,
25        // This makes the visual density adapt to the platform that you run
26        // the app on. For desktop platforms, the controls will be smaller and
27        // closer together (more dense) than on mobile platforms.
28        visualDensity: VisualDensity.adaptivePlatformDensity,
29      ), // ThemeData
30      home: MyHomePage(title: 'Flutter Demo Home Page'),
31    ); // MaterialApp
32  }
```

Ln 12, Col 29 Spaces: 2 UTF-8 CRLF Dart Flutter: 1.22.6 Infinix X650C (android-arm64)

Atau jika pada Android Studio terletak pada toolbar bagian atas tengah

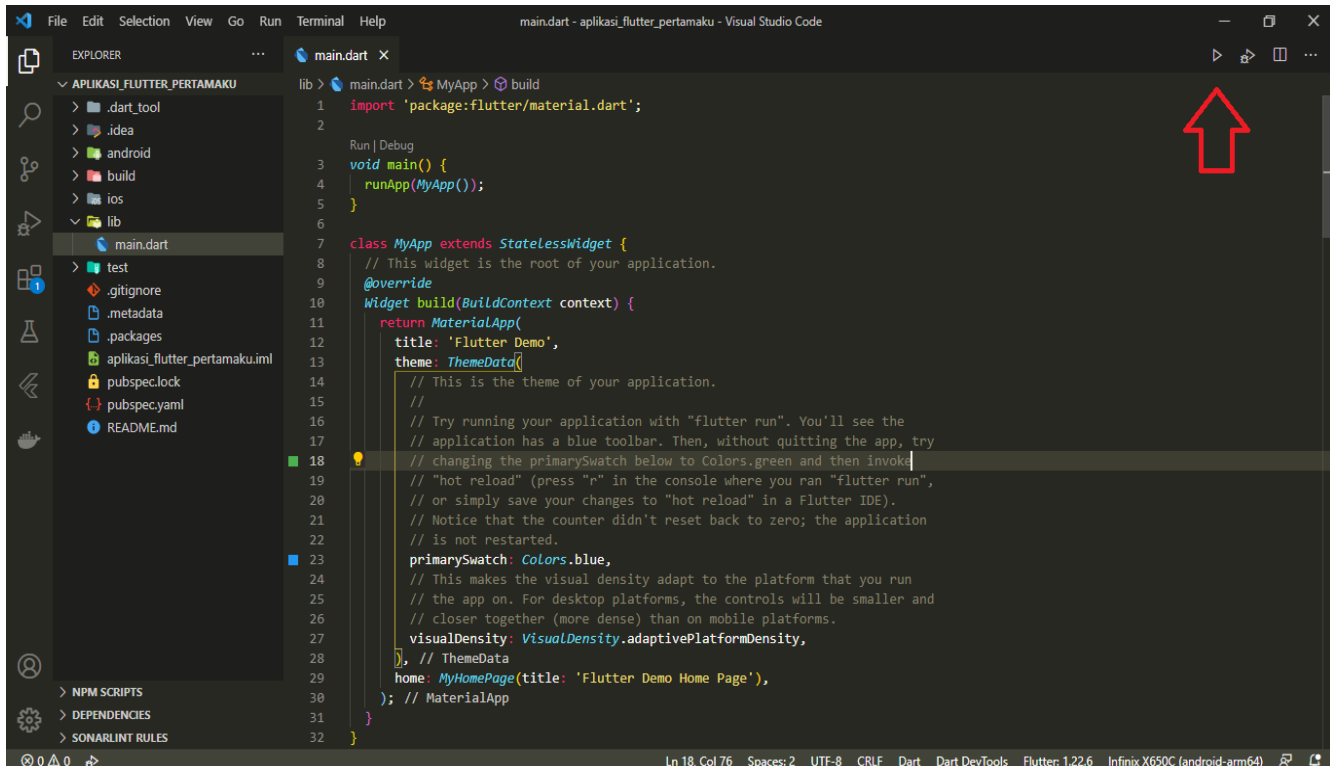


Agar laptop bekerja lebih ringan dapat digunakan Text Editor VSCode dan menjalankan projek langsung menggunakan Handphone Android. Untuk menjalankan projek melalui VSCode dengan klik logo play pada bagian pojok kanan atas

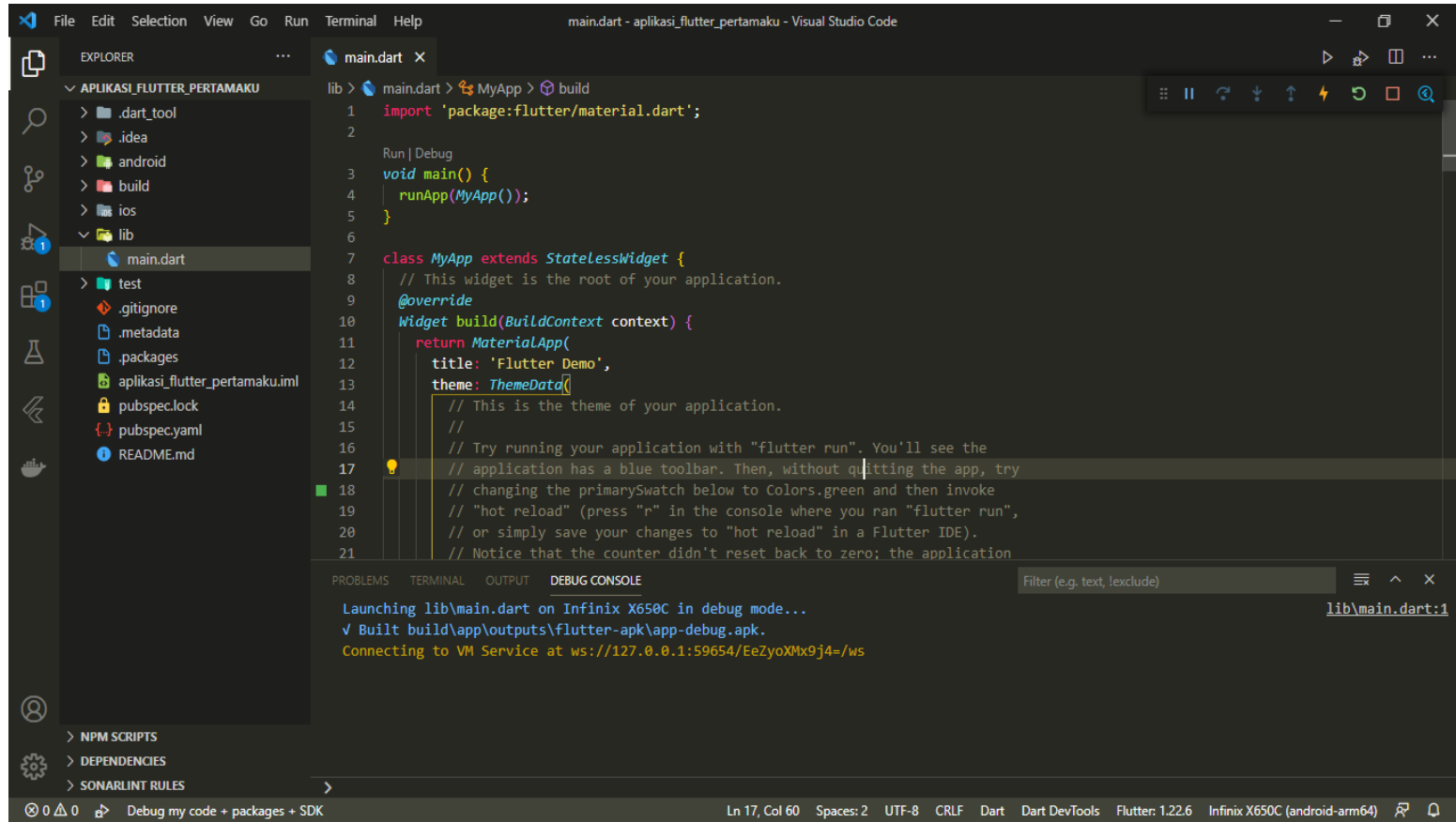


```
lib > main.dart > MyApp > build
1  import 'package:flutter/material.dart';
2
3  Run | Debug
4  void main() {
5    runApp(MyApp());
6  }
7
8  class MyApp extends StatelessWidget {
9    // This widget is the root of your application.
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       title: 'Flutter Demo',
14       theme: ThemeData(
15         // This is the theme of your application.
16         //
17         // Try running your application with "flutter run". You'll see the
18         // application has a blue toolbar. Then, without quitting the app, try
19         // changing the primarySwatch below to Colors.green and then invoke
20         // "hot reload" (press "r" in the console where you ran "flutter run",
21         // or simply save your changes to "hot reload" in a Flutter IDE).
22         // Notice that the counter didn't reset back to zero; the application
23         // is not restarted.
24         primarySwatch: Colors.blue,
25         // This makes the visual density adapt to the platform that you run
26         // the app on. For desktop platforms, the controls will be smaller and
27         // closer together (more dense) than on mobile platforms.
28         visualDensity: VisualDensity.adaptivePlatformDensity,
29       ), // ThemeData
30       home: MyHomePage(title: 'Flutter Demo Home Page'),
31     ); // MaterialApp
32   }
```

Agar laptop bekerja lebih ringan dapat digunakan Text Editor VSCode dan menjalankan proyek langsung menggunakan Handphone Android. Untuk menjalankan proyek melalui VSCode dengan klik logo play pada bagian pojok kanan atas



```
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2
3 Run | Debug
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   // This widget is the root of your application.
10  @override
11  Widget build(BuildContext context) {
12    return MaterialApp(
13      title: 'Flutter Demo',
14      theme: ThemeData(
15        // This is the theme of your application.
16        //
17        // Try running your application with "flutter run". You'll see the
18        // application has a blue toolbar. Then, without quitting the app, try
19        // changing the primarySwatch below to Colors.green and then invoke
20        // "hot reload" (press "r" in the console where you ran "flutter run",
21        // or simply save your changes to "hot reload" in a Flutter IDE).
22        // Notice that the counter didn't reset back to zero; the application
23        // is not restarted.
24        primarySwatch: Colors.blue,
25        // This makes the visual density adapt to the platform that you run
26        // the app on. For desktop platforms, the controls will be smaller and
27        // closer together (more dense) than on mobile platforms.
28        visualDensity: VisualDensity.adaptivePlatformDensity,
29      ), // ThemeData
30      home: MyHomePage(title: 'Flutter Demo Home Page'),
31    ); // MaterialApp
32  }
```



```
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   // This widget is the root of your application.
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Flutter Demo',
13      theme: ThemeData(
14        // This is the theme of your application.
15        //
16        // Try running your application with "flutter run". You'll see the
17        // application has a blue toolbar. Then, without quitting the app, try
18        // changing the primarySwatch below to Colors.green and then invoke
19        // "hot reload" (press "r" in the console where you ran "flutter run",
20        // or simply save your changes to "hot reload" in a Flutter IDE).
21        // Notice that the counter didn't reset back to zero; the application
```

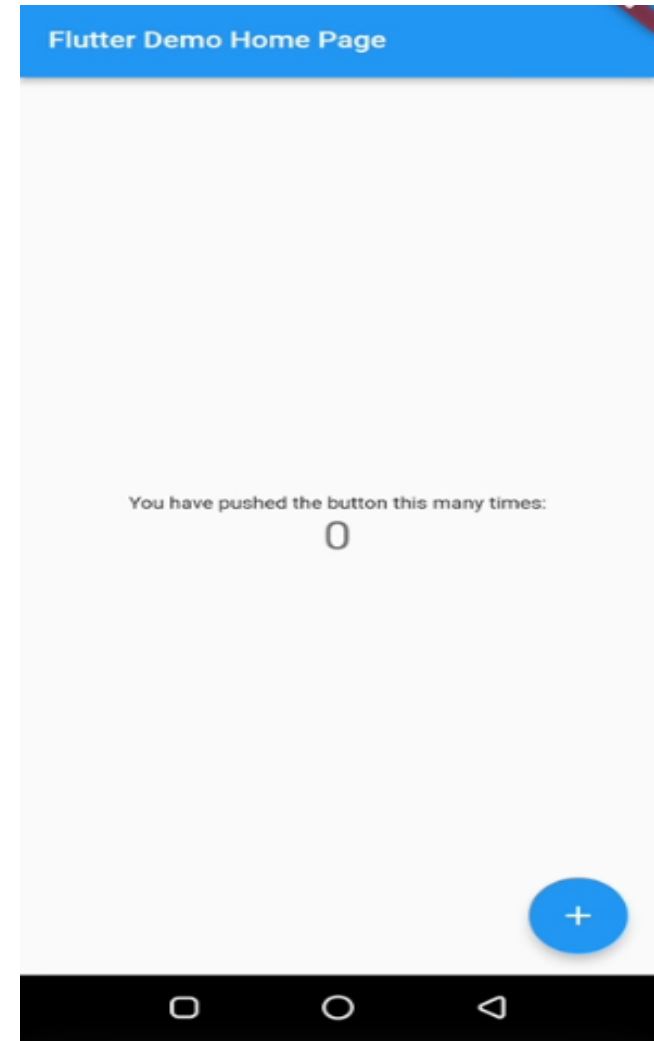
DEBUG CONSOLE

Launching lib\main.dart on Infinix X650C in debug mode...

✓ Built build\app\outputs\flutter-apk\app-debug.apk.

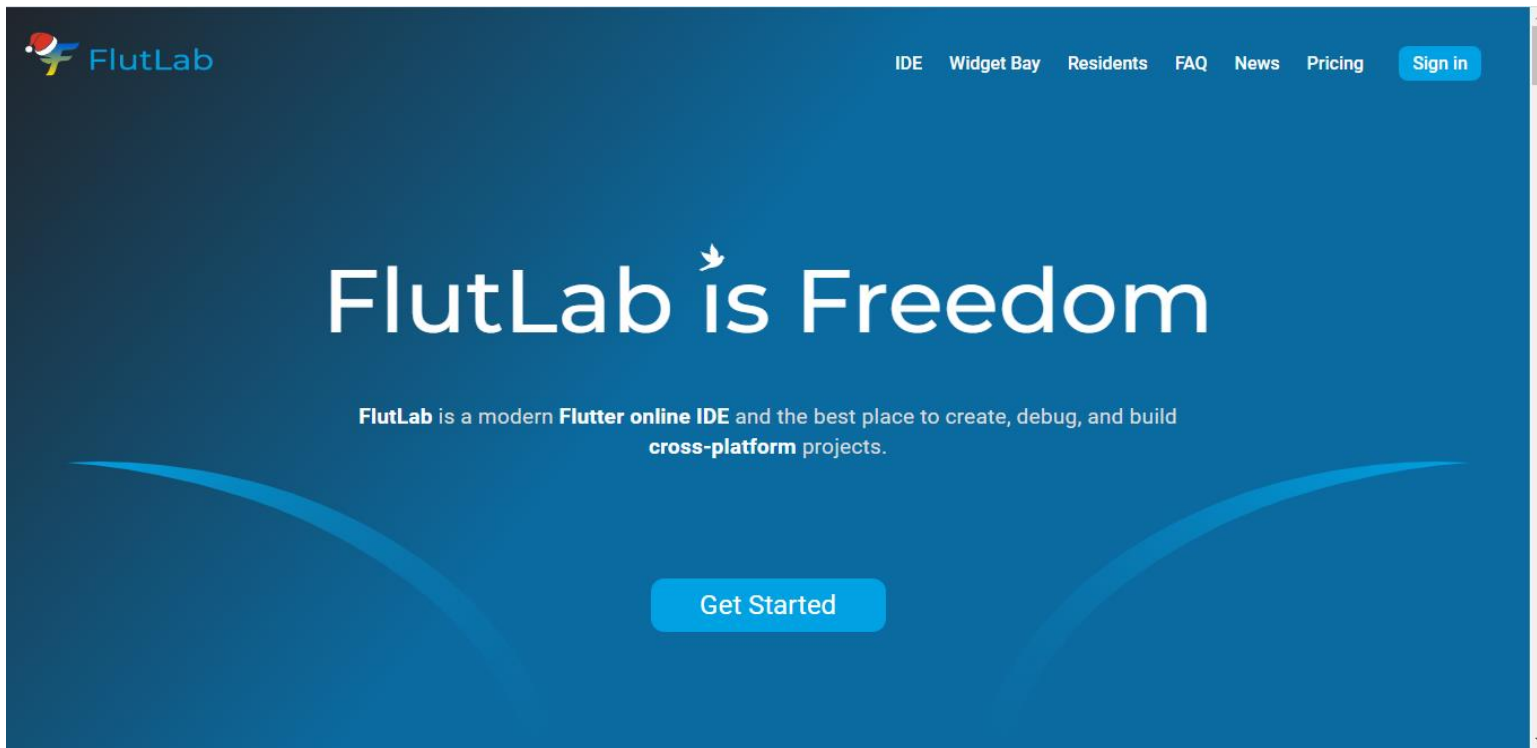
Connecting to VM Service at ws://127.0.0.1:59654/EeZyoXMx9j4=/ws

Adapun tampilannya adalah sebagai berikut

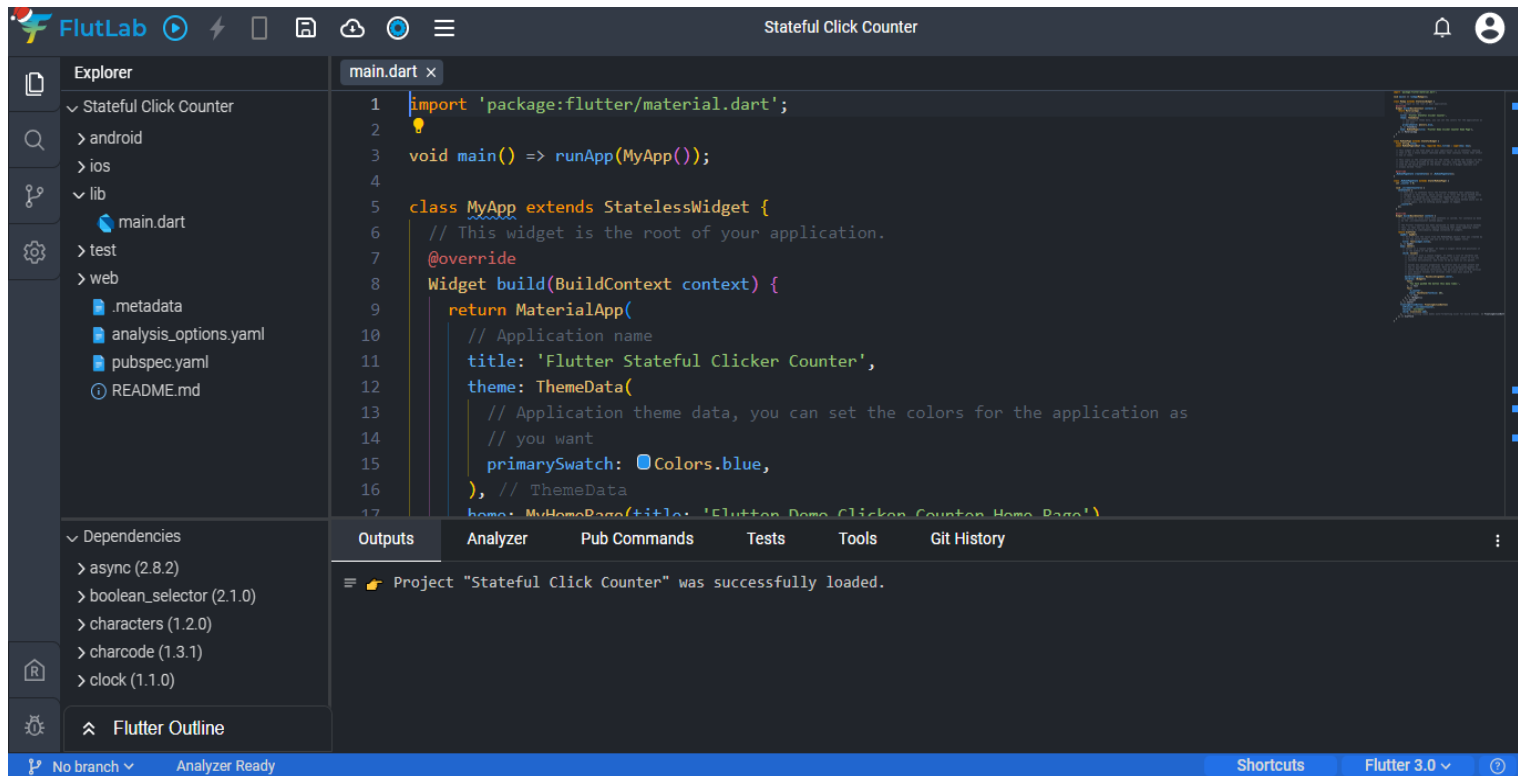


Membuat dan menjalankan Proyek dengan flutlab.io dan Web Emulator

Selain menggunakan Android Studio dan VS Code, untuk melakukan pengkodean flutter dapat pula menggunakan editor online yaitu **flutlab.io**. Namun untuk menggunakan flutlab.io ini memerlukan akses internet

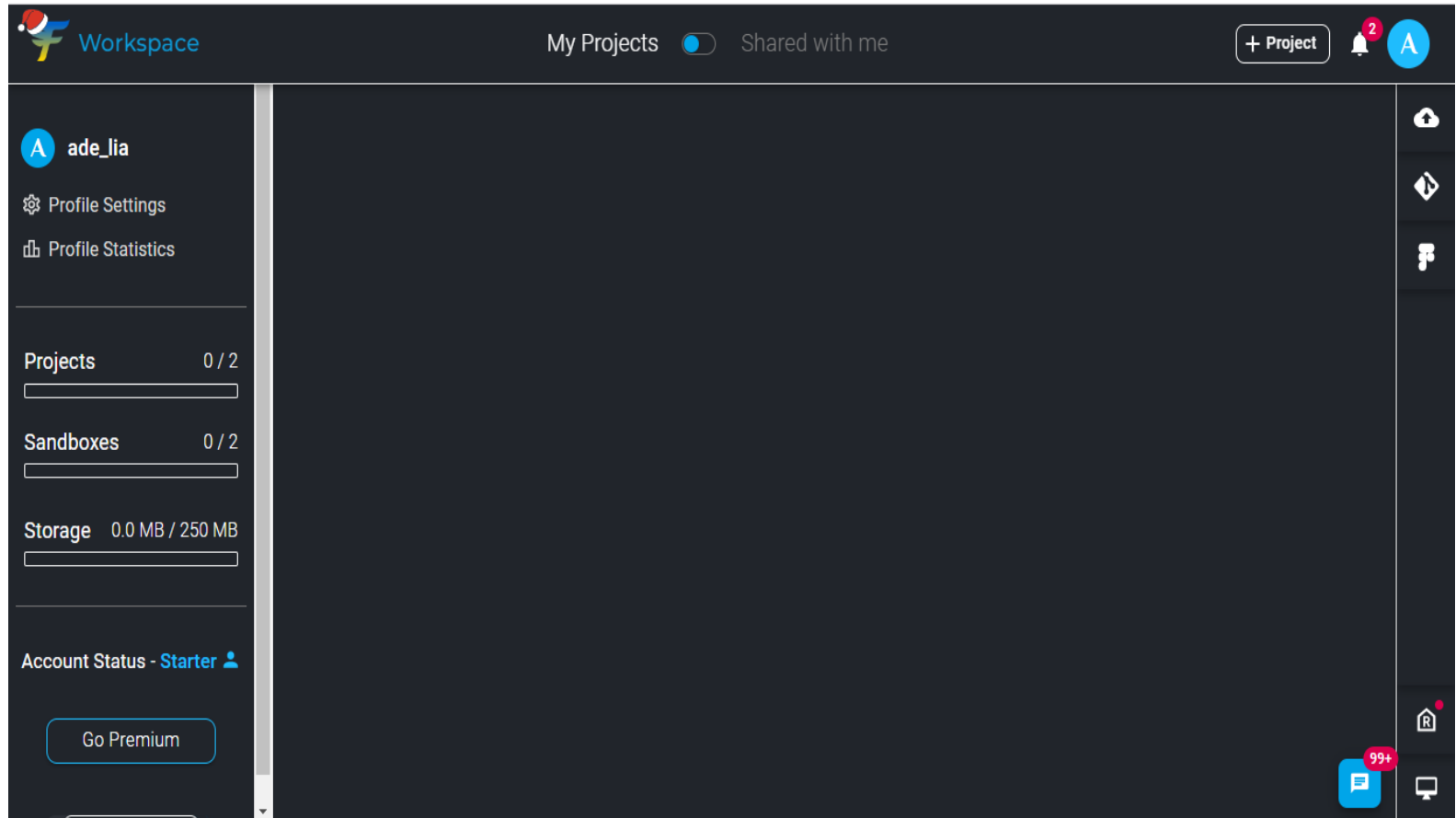


Silahkan buka laman flutlab.io kemudian klik **Get Started** maka akan muncul laman dengan template awal code flutter



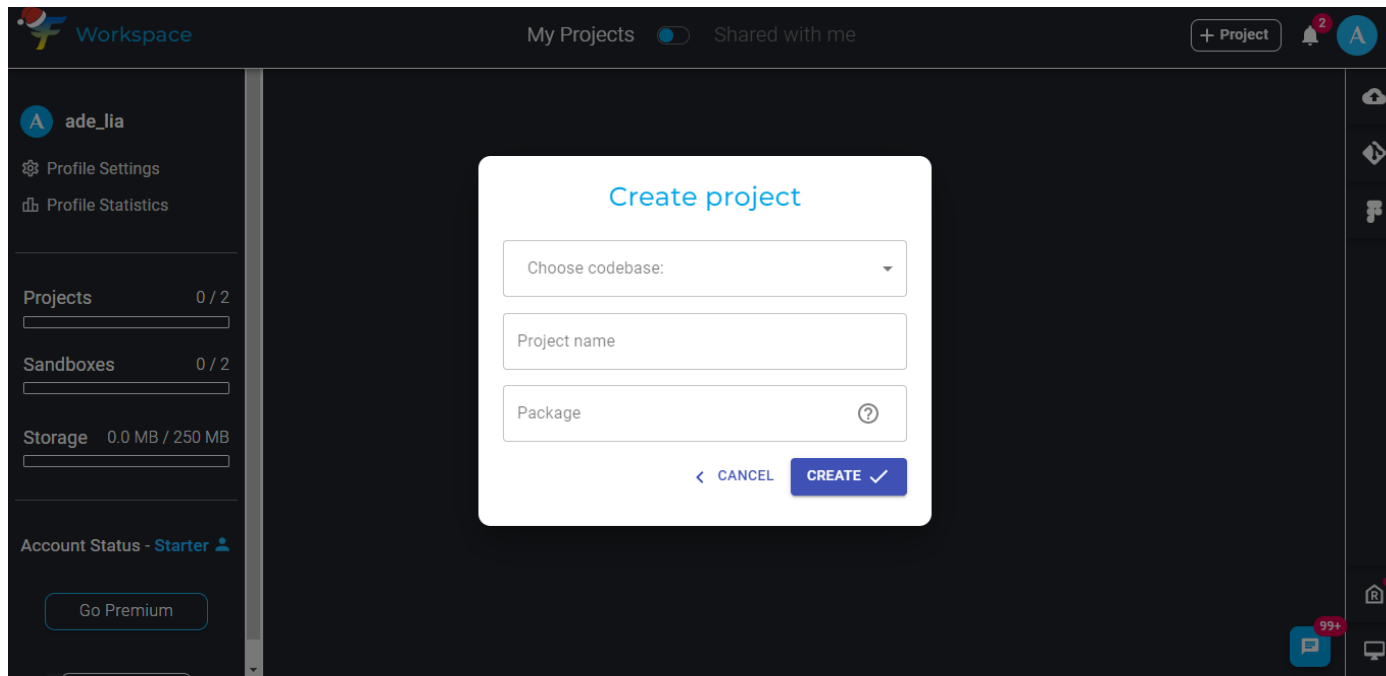
Agar aplikasi yang dibuat dapat tersimpan, lakukan login dengan menekan tombol dengan logo user pada pojok kanan atas, kemudian kita dapat login dengan dua cara, yaitu login dengan google ataupun membuat akun tersendiri pada flutlab.io

Setelah login pada flutlab.io maka akan tampil halaman workspace



Membuat dan Menjalankan Proyek di flutlab.io

Klik tombol **+ Project** yang terdapat pada pojok kanan atas

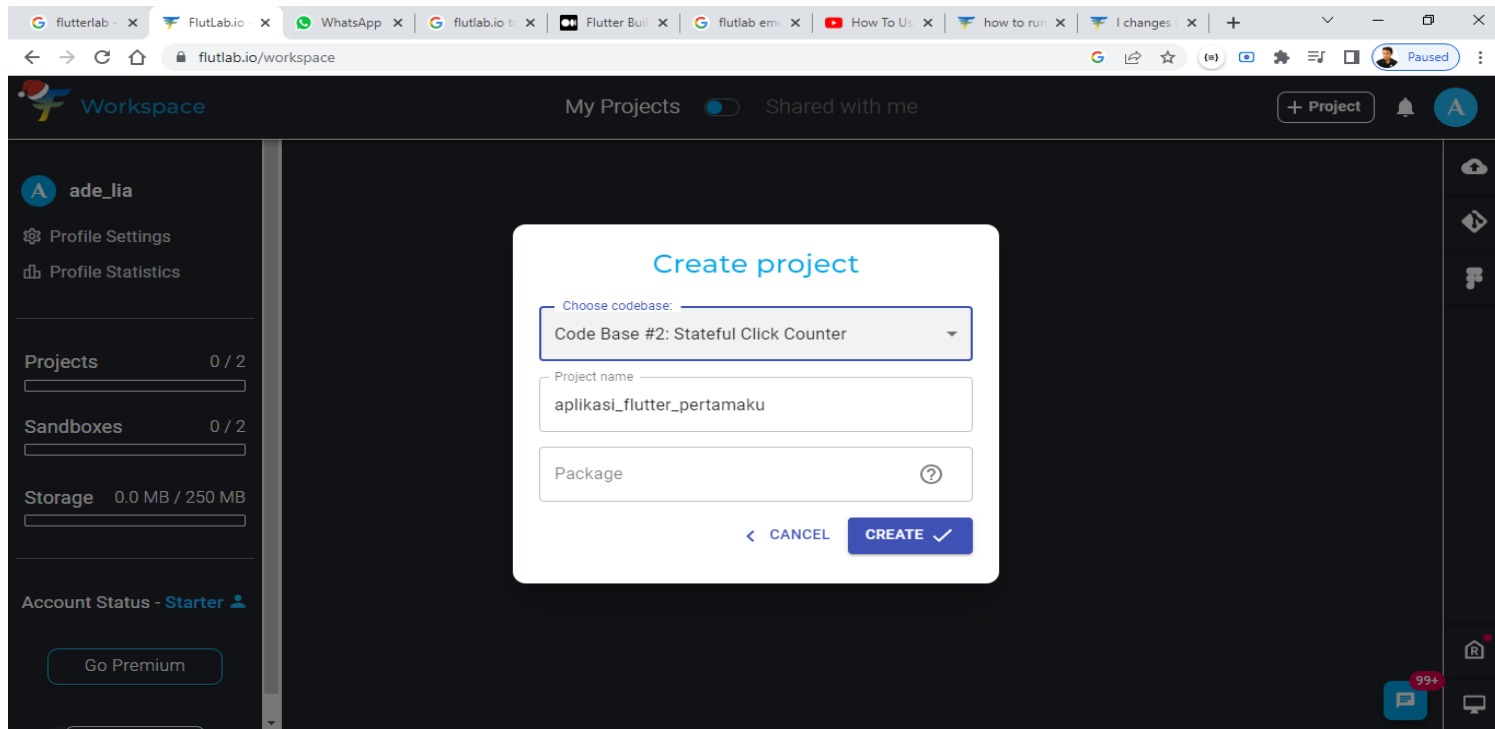


Pada form tersebut isikan dengan data berikut:

Codebase : Code Base #2: Stateful Click Counter

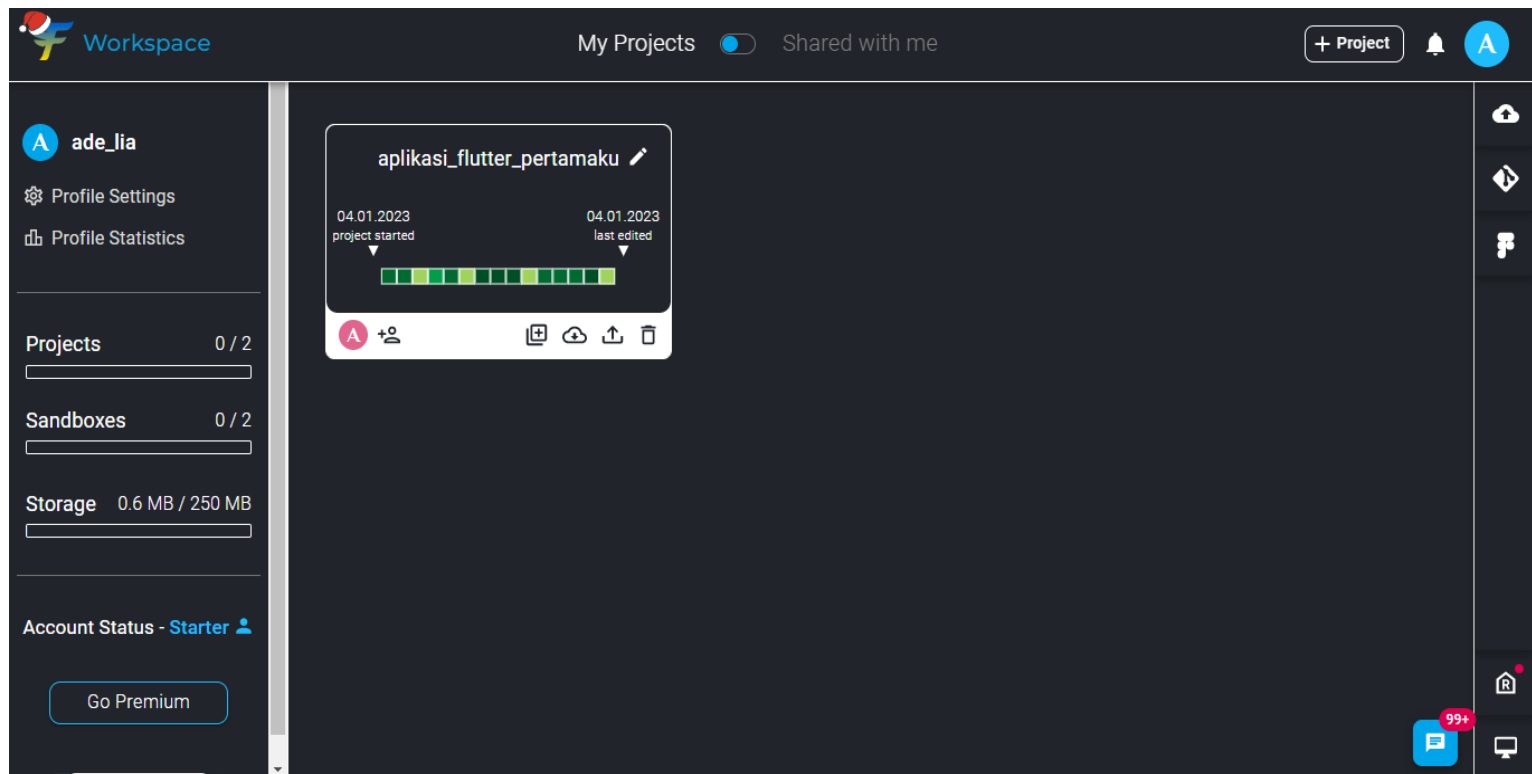
Projek Name : aplikasi_flutter_pertamaku

Untuk bagian **Package** dapat diabaikan dan akan mengambil nama package default yaitu **com.codegemz.flutlab**

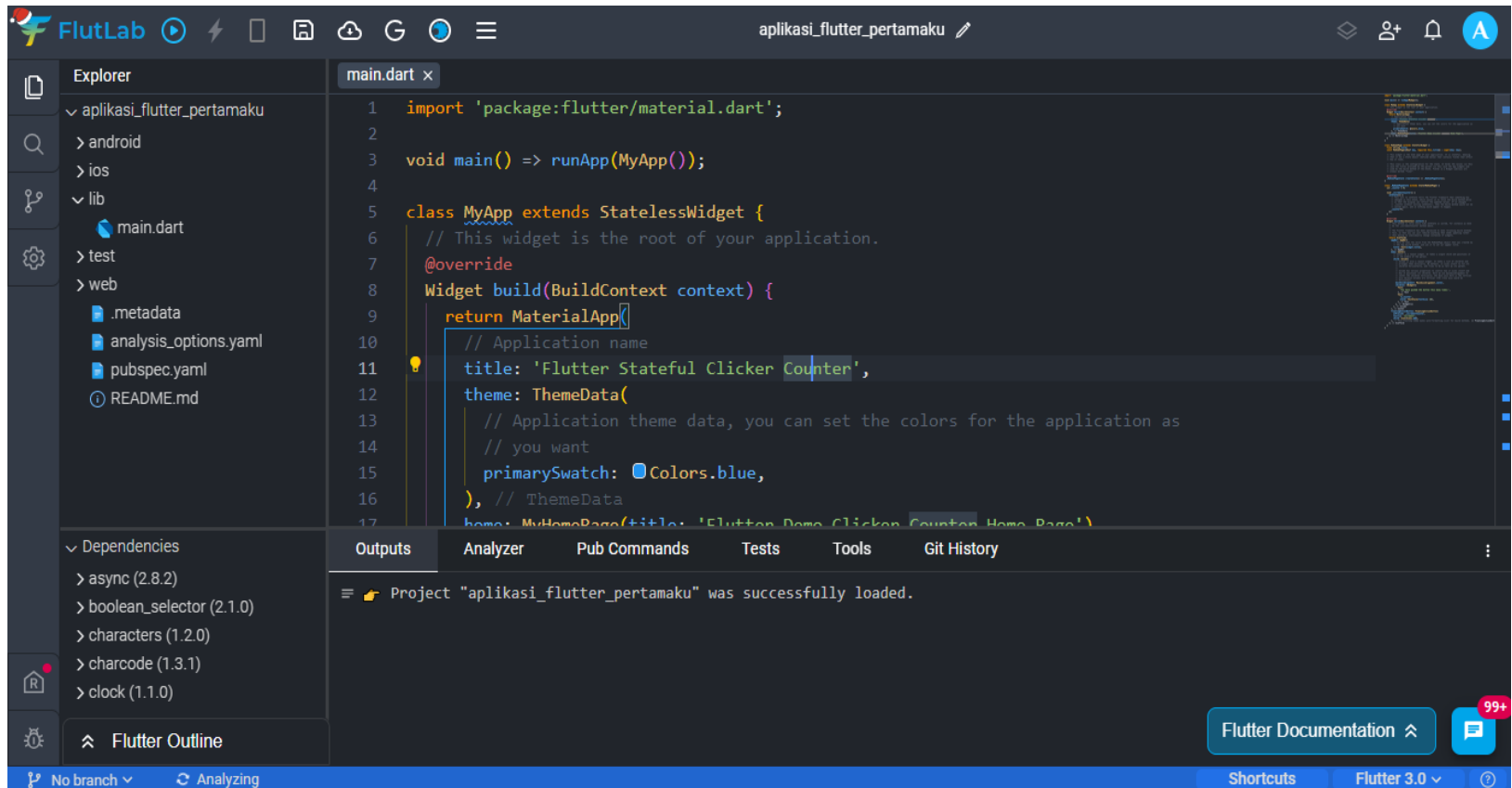


Kemudian klik tombol **CREATE**

Kemudian kita akan diarahkan ke halaman workspace dengan proyek yang telah dibuat sebelumnya. Untuk membuka proyek, klik proyek yang telah dibuat,



maka akan muncul code proyek tersebut




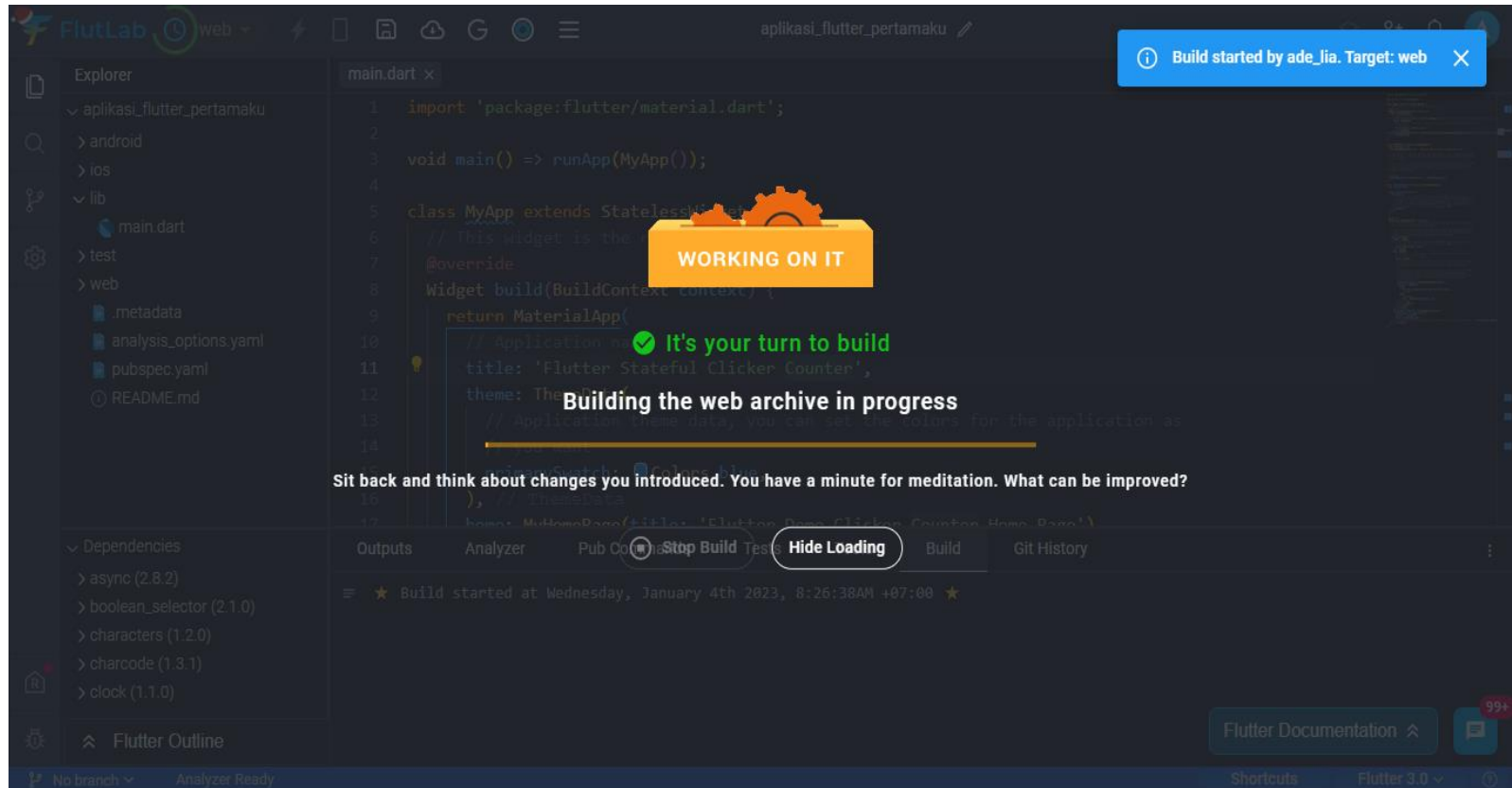
```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatelessWidget {
6   // This widget is the root of your application.
7   @override
8   Widget build(BuildContext context) {
9     return MaterialApp(
10      // Application name
11      title: 'Flutter Stateful Clicker Counter',
12      theme: ThemeData(
13        // Application theme data, you can set the colors for the application as
14        // you want
15        primarySwatch: Colors.blue,
16      ), // ThemeData
17      home: MyHomePage(title: 'Flutter Demo Clicker Counter Home Page')
18    );
19  }
```

Project "aplikasi_flutter_pertamaku" was successfully loaded.

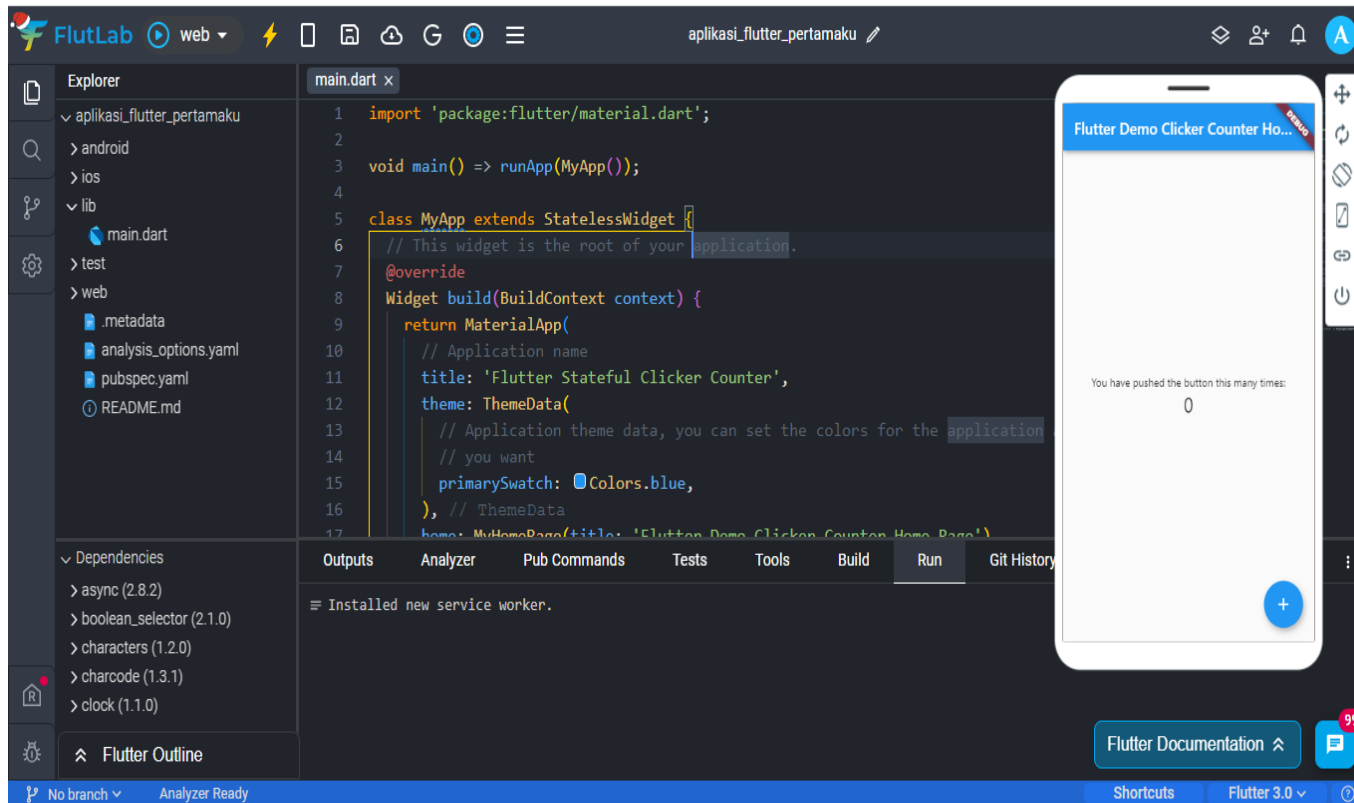
Flutter Documentation 99+

No branch Analyzing Shortcuts Flutter 3.0

Untuk menjalankan proyek, klik tombol  (pastikan opsi yang terpilih adalah **web**) yang terdapat di pojok kiri atas, tunggu hingga proses build selesai.



Setelah selesai, akan muncul tampilan seperti berikut



Setelah melakukan perubahan pada kodingan, kita dapat menyimpan pembaruan dengan shortcut **CTRL + S**