MOBILE APPLICATION 2

PERTEMUAN 5



**DESKRIPSI MATAKULIAH**

**Mata Kuliah ini mempelajari konsep bahasa pemrograman Dart, membuat aplikasi mobile dengan flutter menggunakan bahasa Dart, menerapkan widget pada flutter untuk membuat aplikasi mobile**

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami penggunaan Framework Flutter dengan IDE Visual Studio Code untuk membuat aplikasi mobile.**

**DOSEN PENGAMPU**

**Taufikqurrohman, M.Kom**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**(STMIK) WIDYA PRATAMA**

[Pokok bahasan/bab/pertemuan]

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami penggunaan Framework Flutter dengan IDE Visual Studio Code untuk membuat aplikasi mobile.**

**MATERI PEMBELAJARAN**

1. **Android Studio**
2. **Visual Studio Code**
3. **Framework Flutter**

**MATERI**

Sebelum memulai tahapan pembuatan aplikasi, silahkan siapkan dulu aplikasi yang dibutuhkan:

1. JDK 8: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html#license-lightbox>
2. Android Studio: <https://developer.android.com/studio#downloads>
3. Visual Studio Code: [https://code.visualstudio.com/download#](https://code.visualstudio.com/download)
4. Framework Flutter: <https://storage.googleapis.com/flutter_infra/releases/stable/windows/flutter_windows_1.22.2-stable.zip>

Spesifikasi minimum yang disarankan untuk membuat aplikasi mobile dengan lancar:

1. Microsoft® Windows® 7 SP1/8/10 (64-bit)
2. Prosesor Core i3 2.0 GHz
3. 4 GB RAM minimum, 8 GB RAM recommended
4. 2 GB of available disk space minimum
5. Disarankan menggunakan SSD

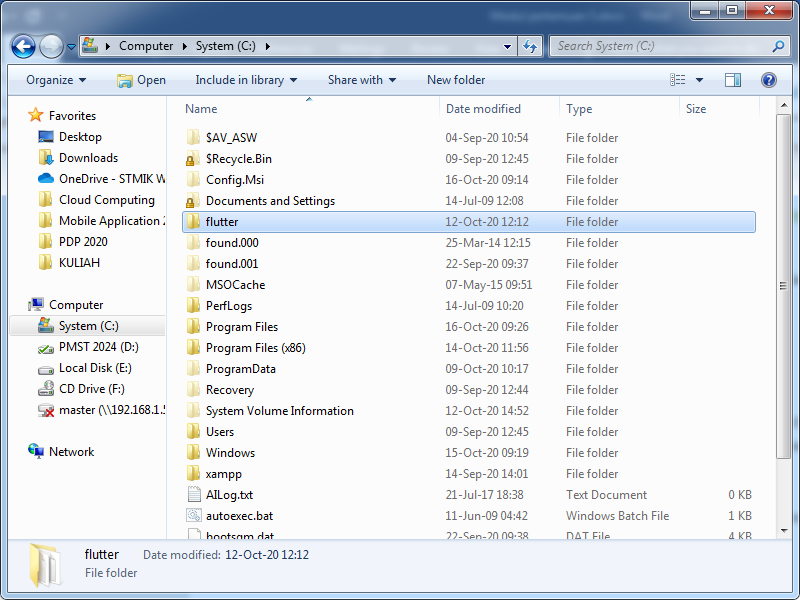
Sekarang kita mulai menyiapkan tools untuk membuat aplikasi mobile dengan Flutter. Silahkan buka halaman <https://flutter.dev/docs/get-started/install/windows>. Berdasarkan halaman tersebut, tahapannya adalah sbb:

1. **Menyiapkan JDK 8**

Silahkan download JDK 8 dari link yang disediakan di atas, kemudian instal, ikuti prosesnya sampai selesai.

1. **Menyiapkan Flutter SDK**

Silahkan download Flutter SDK terbaru dari link yang disediakan di atas. Ekstrak file zip ke C:\flutter.



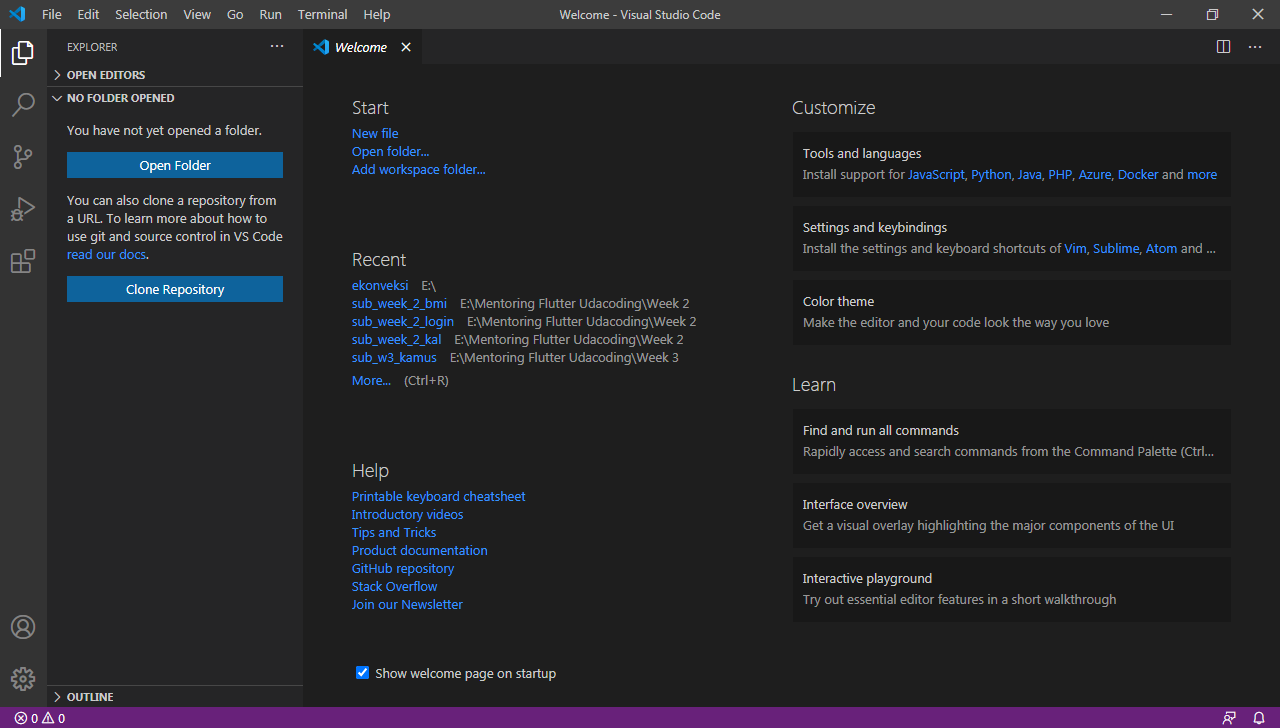
1. **Menyiapkan Andoid Studio**

Silahkan download android studio dari link yang disediakan di atas, kemudian instal, ikuti prosesnya sampai selesai.

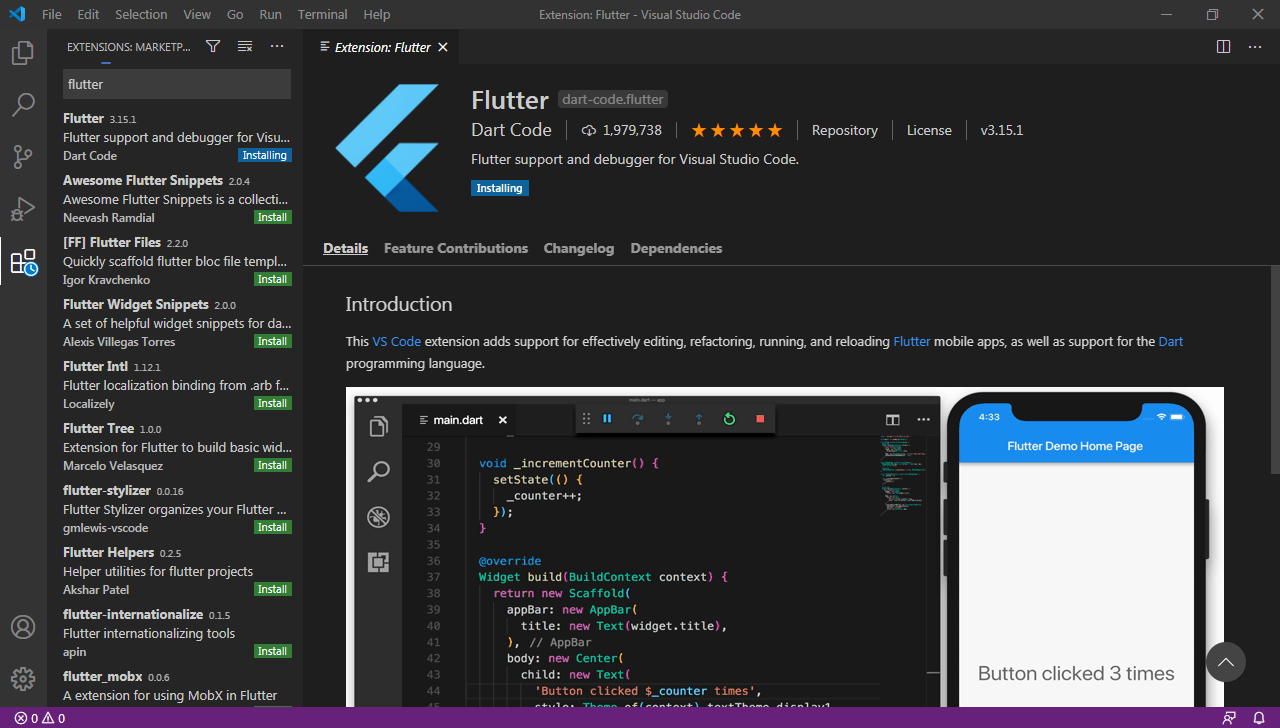
1. **Menyiapkan Visual Studio Code**

Silahkan download visual studio code dari link yang disediakan di atas, kemudian instal, ikuti prosesnya sampai selesai.

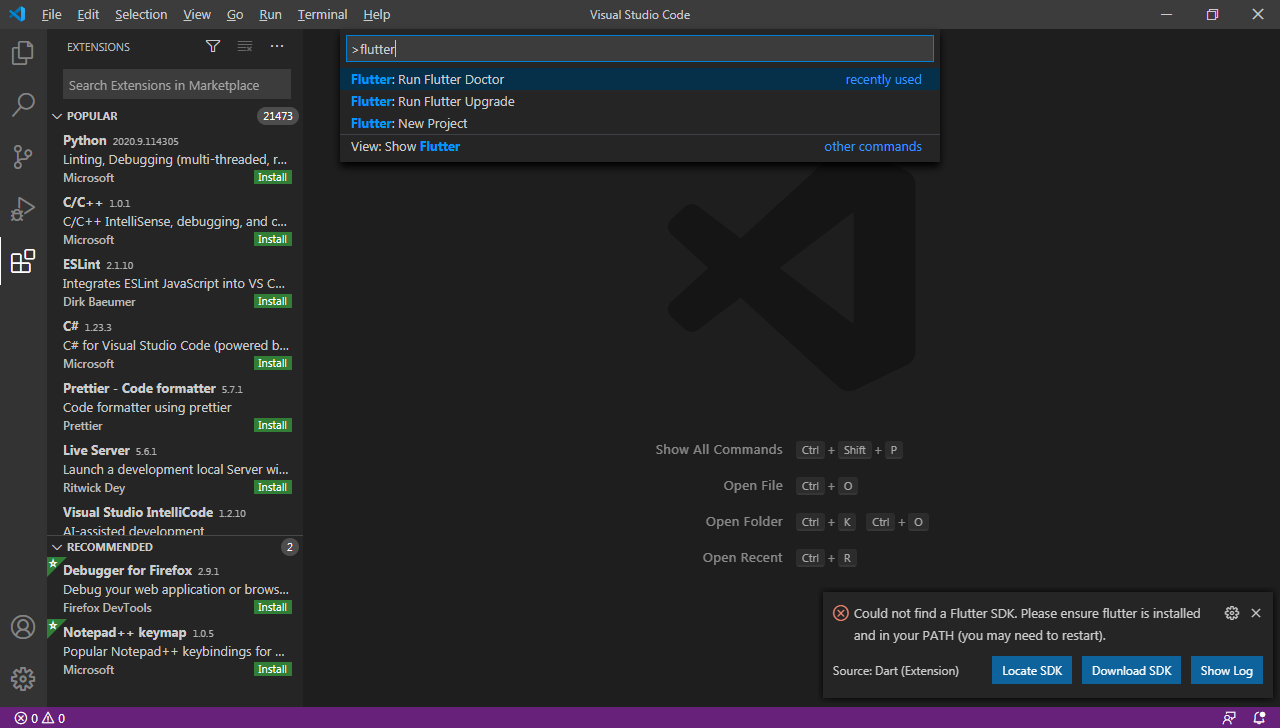
Setelah selesai, silahkan jalankan Visual Studio Code.



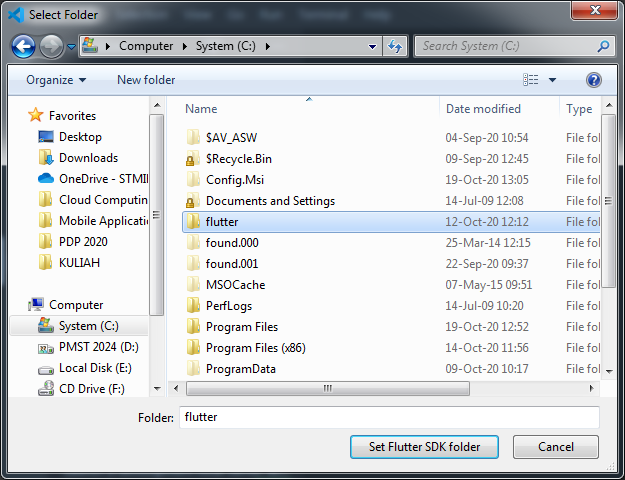
Install extension dart dan flutter. Klik Extension pada menu samping, kemudian search flutter dan selanjutnya install. Dengan menginstall flutter, secara otomatis kita menginstal extension untuk dart.



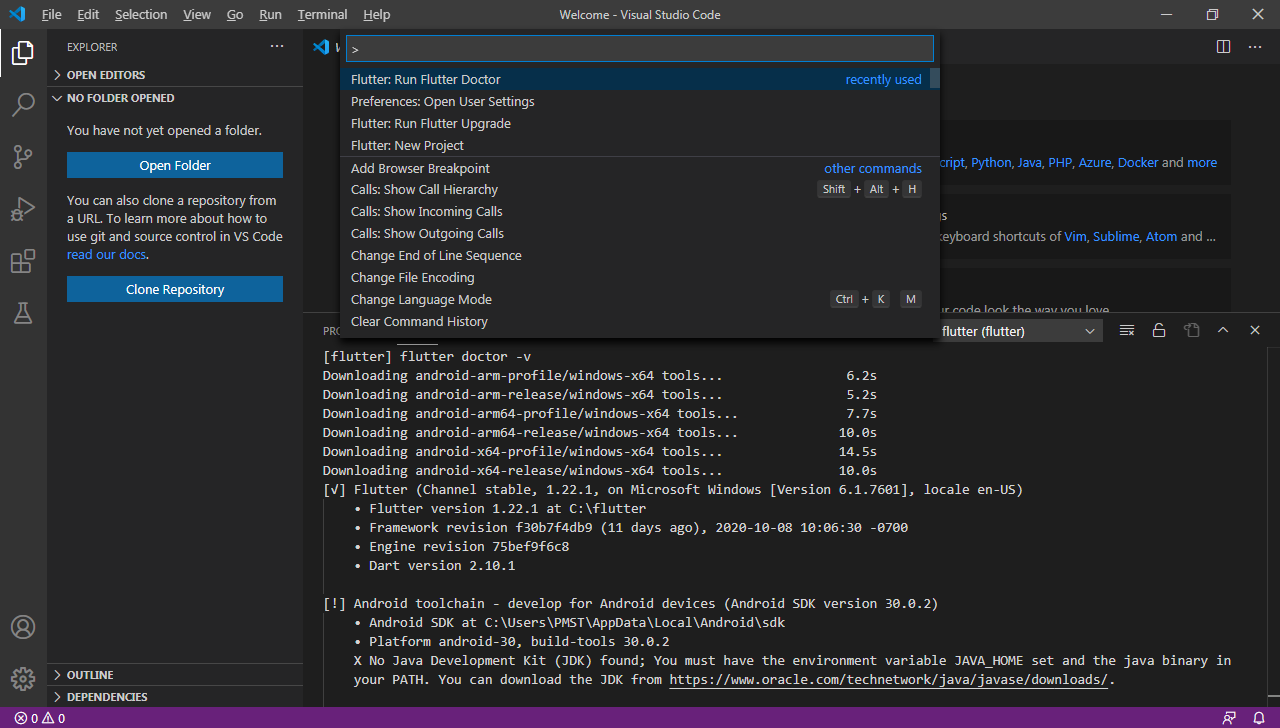
Pilih **View > Command Pallete > ketikkan flutter > Flutter: Run Flutter Doctor**, akan muncul window di pojok kanan bawah, silahkan pilih locate SDK



Arahkan ke folder flutter hasil dari ekstrak sebelumnya yaitu di drive C, kemudian pilih Set Flutter SDK Folder.



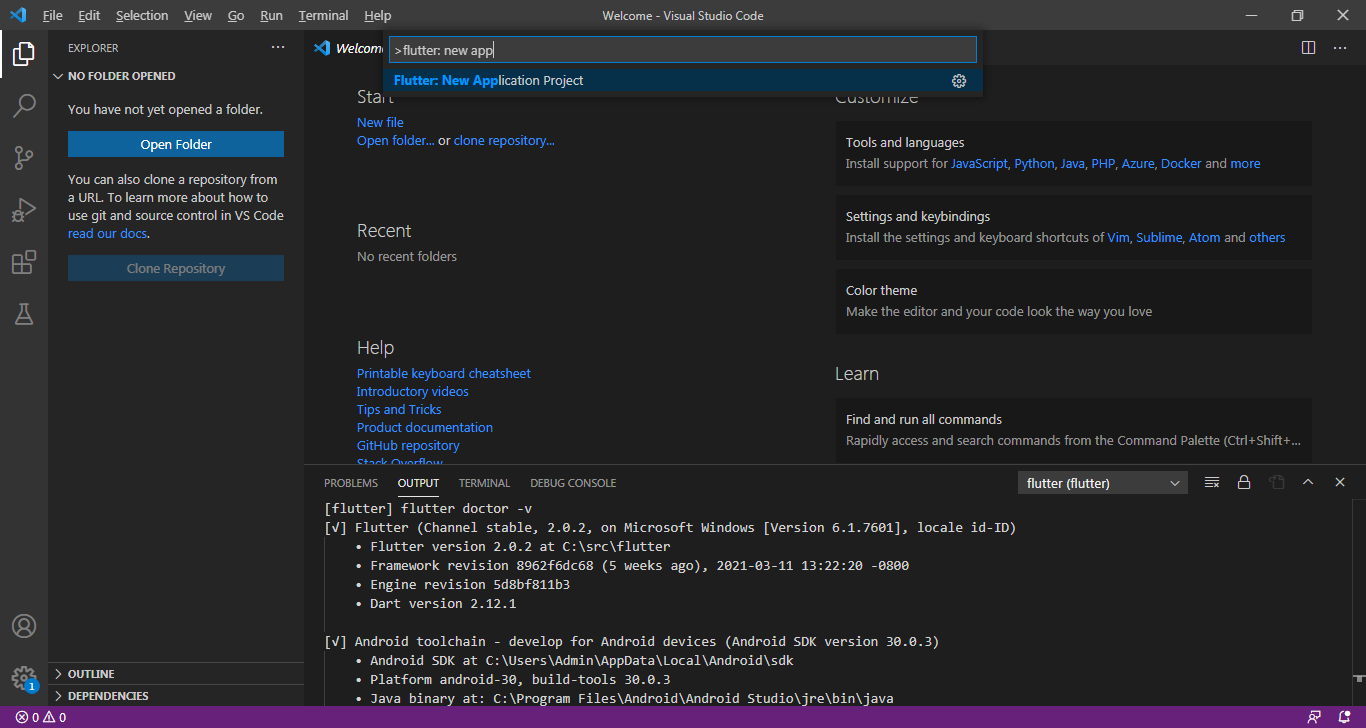
Kemudian close VSCode dan jalankan lagi. Lakukan perintah seperti sebelumnya yaitu **View > Command Pallete > Flutter: Run Flutter Doctor**. Akan ditampilkan versi flutter yang terinstall.



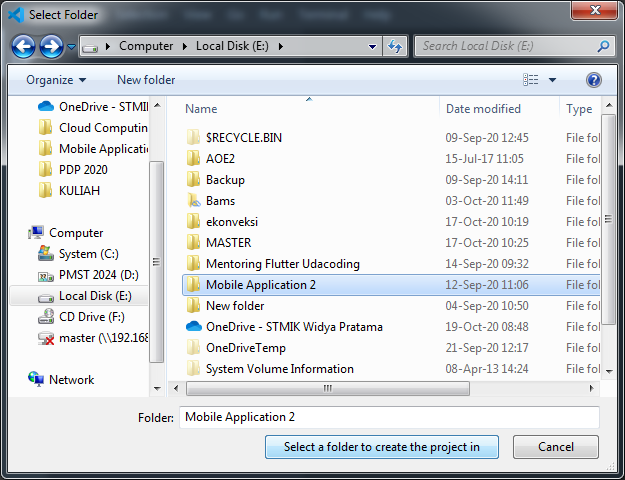
Membuat Aplikasi Mobile dengan Framework Flutter. Langkah-langkahnya adalah sebagai berikut:

1. **Membuat Project Baru**

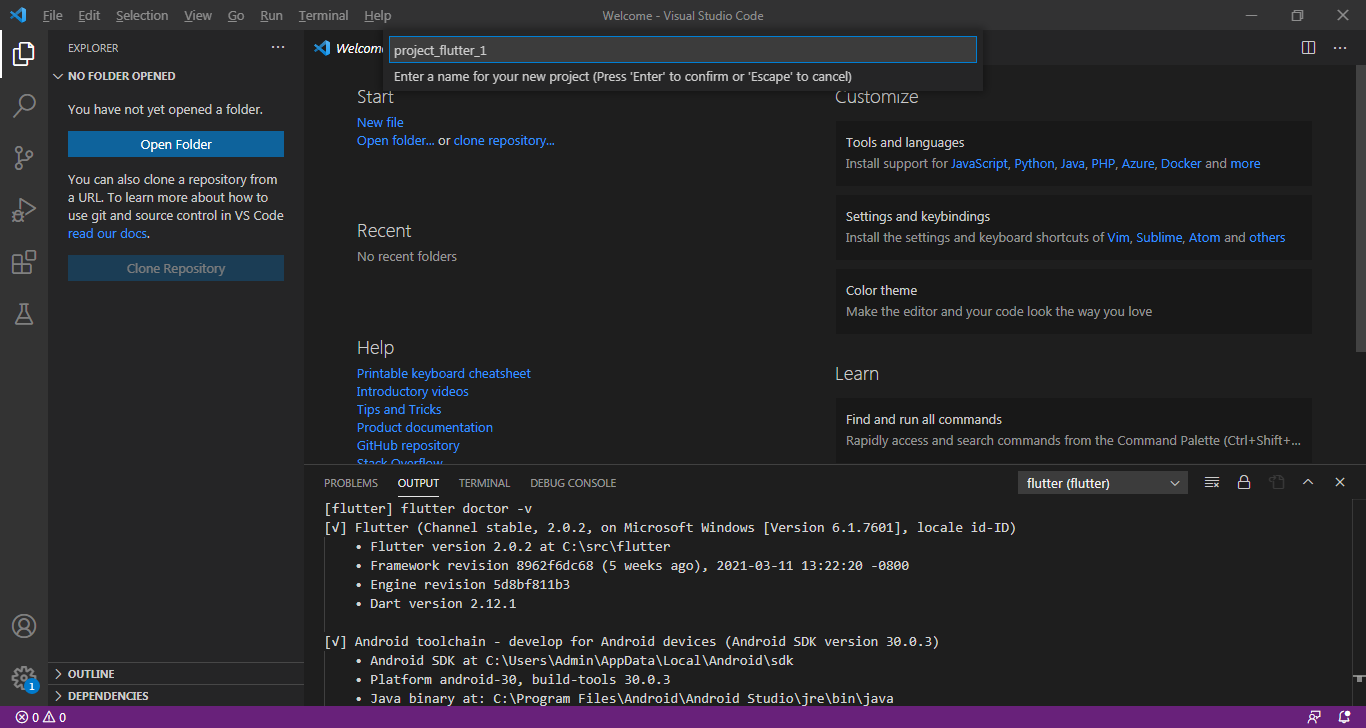
Untuk membuat projek baru silahkan pilih **View > Command Pallete > Flutter: New Application Project**.



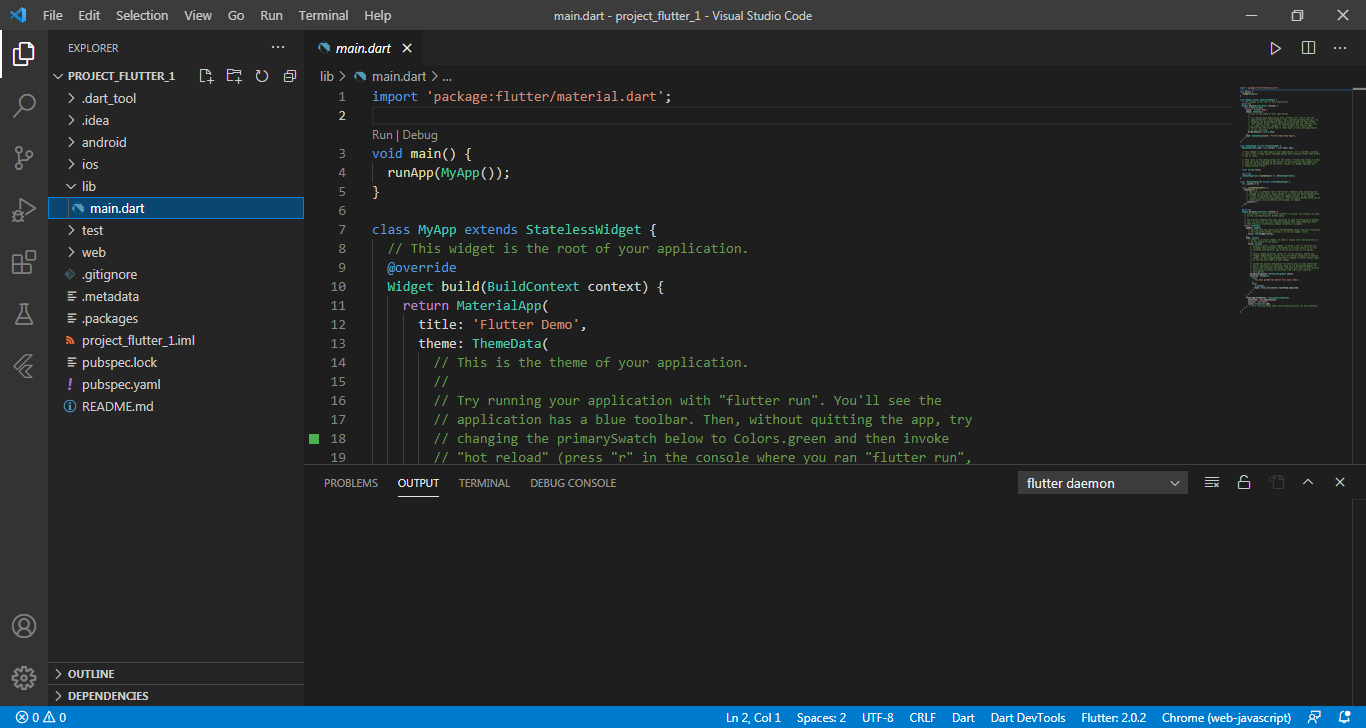
Pilih lokasi penyimpanan projeknya (terserah mau disimpan dimana).



Ketikkan nama projek nya, misal namanya adalah **project\_flutter\_1** kemudian enter.



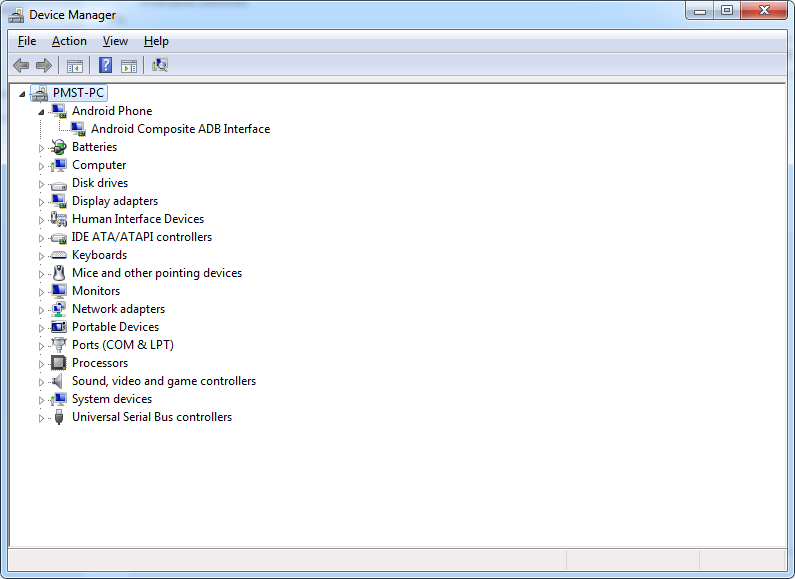
Ini adalah tampilan awal dari projek pertama.



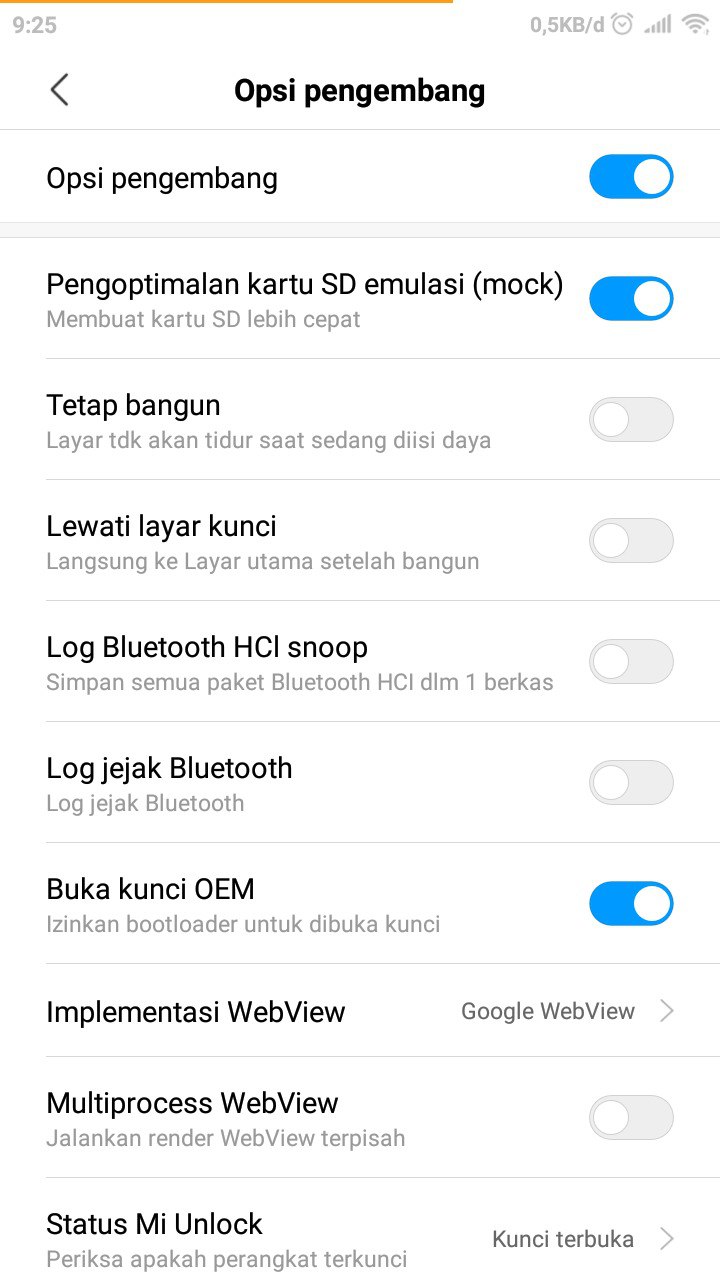
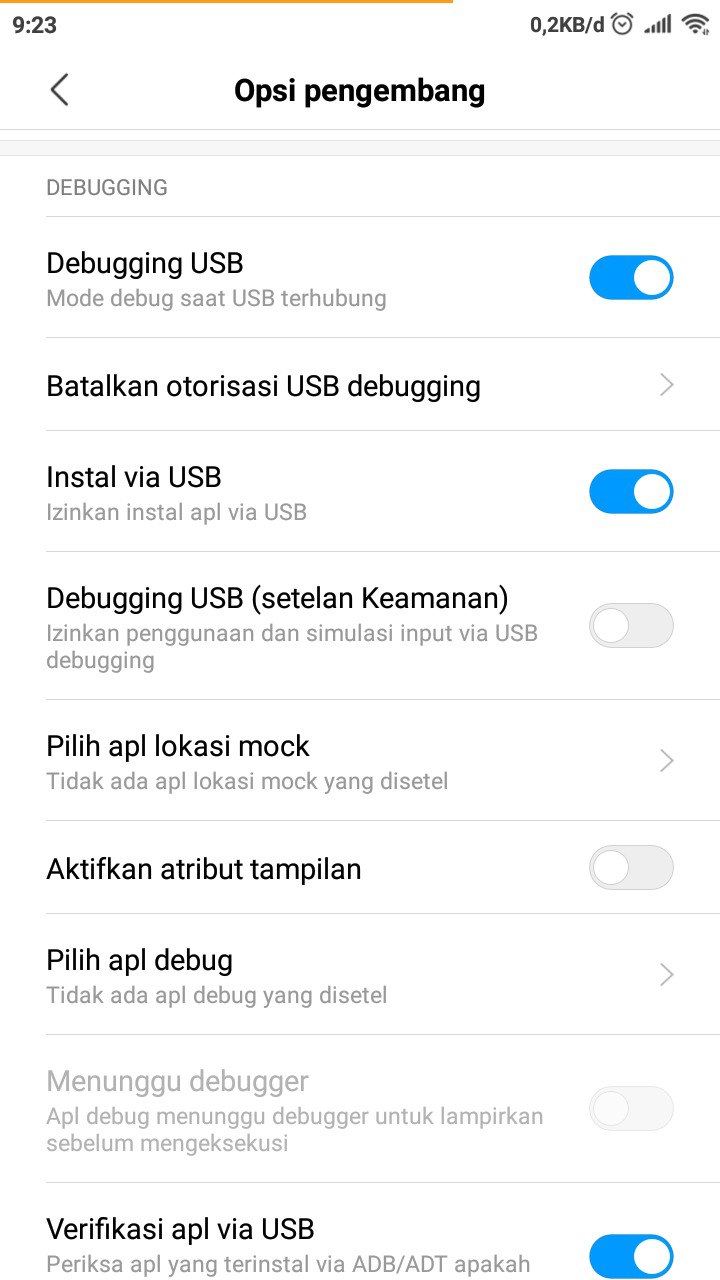
1. **Menjalankan Projek**

Untuk menjalankan projek, ada beberapa pilihan. Yang pertama menggunakan emulator seperti AVD dari android studio atau emulator nox, bluestack, memu. Akan tetapi jika kita menggunakan emulator, maka akan menggunakan RAM komputer kita, sehingga akan menambah berat. Kemudian yang kedua menggunakan perangkat asli. Nah untuk komputer dengan spek minimum, disarankan untuk menggunakan perangkat asli.

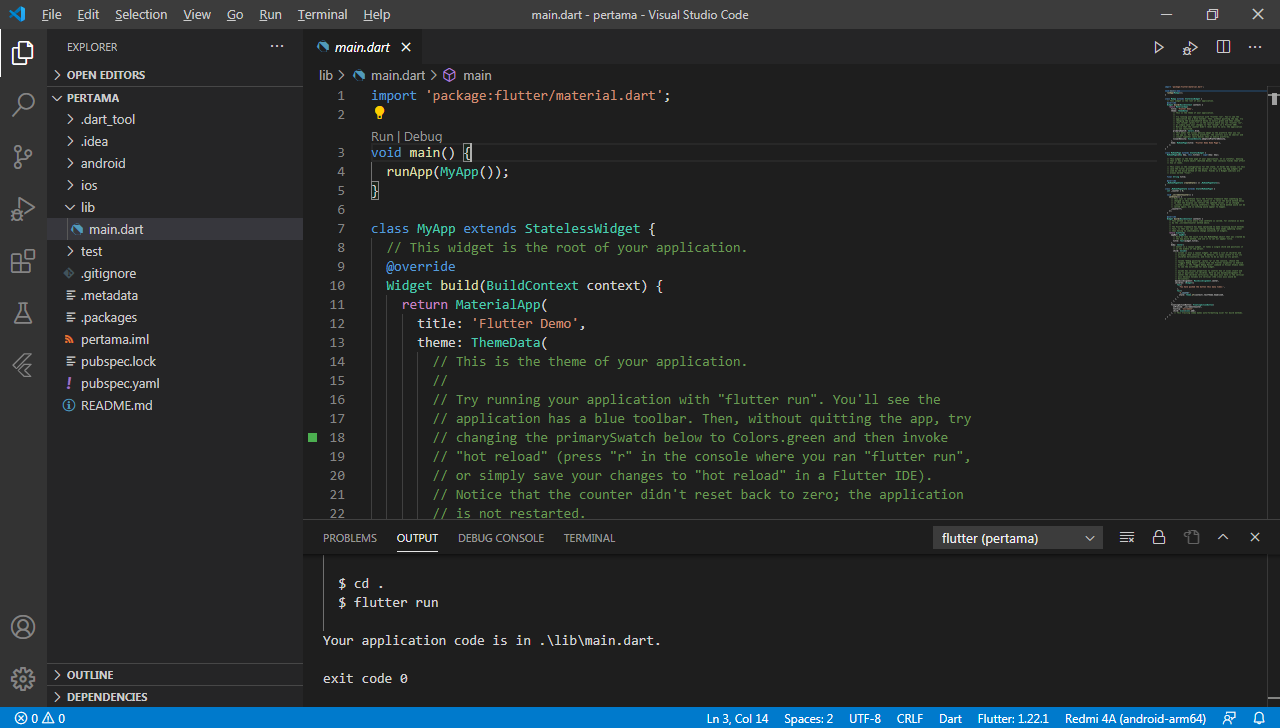
Untuk bisa menggunakan perangkat asli, pertama kita harus install terlebih dahulu driver ADB android kita. Untuk memastikan, lihat di bagian Device Manager.



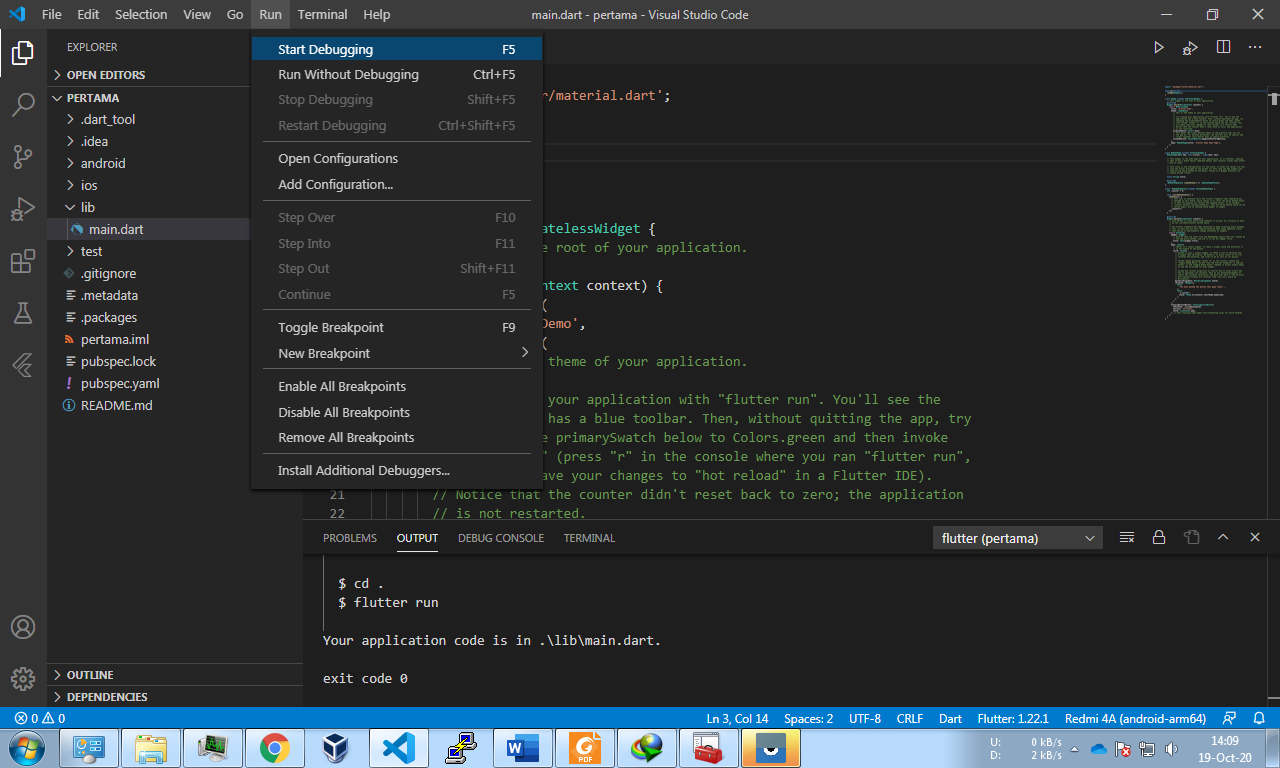
Kemudian pastikan juga untuk mengaktifkan USB Debugging pada perangkat kita. Pertama kita aktifkan dulu **Developer Option** kemudian aktifkan **USB Debugging**.

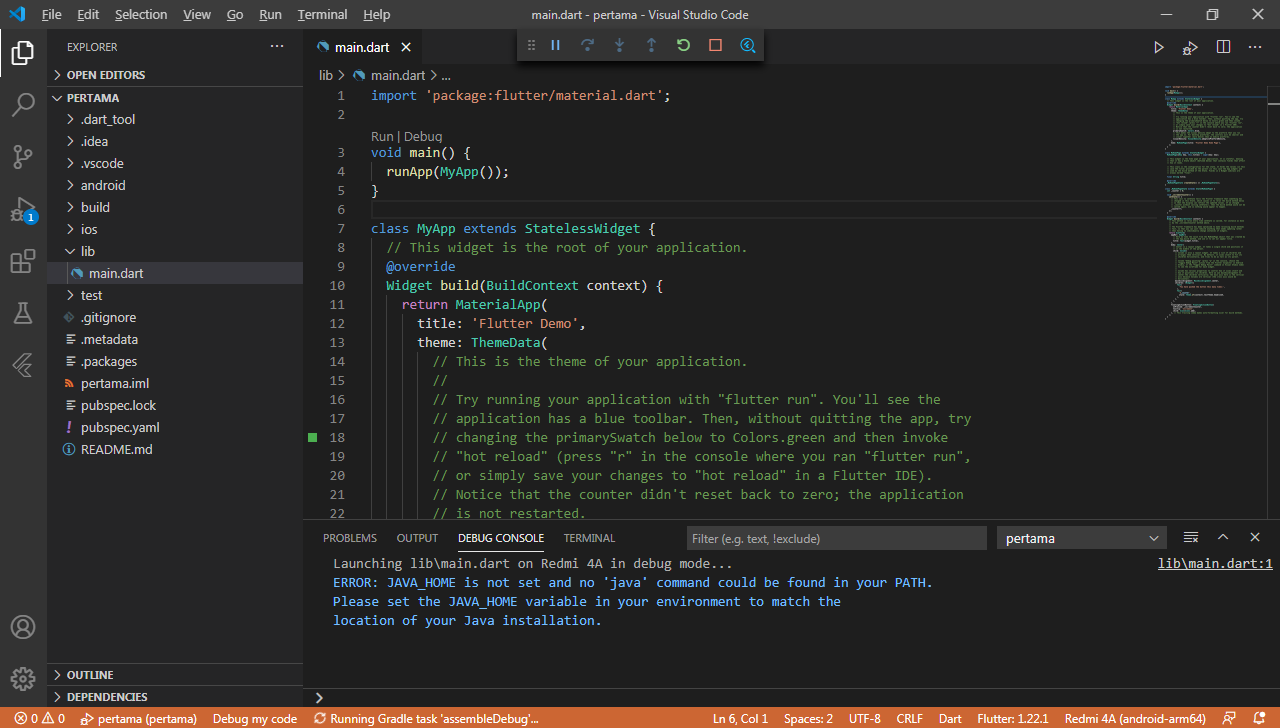
Selanjutnya kembali ke VSCode, di bagian kanan bawah akan muncul perangkat kita.



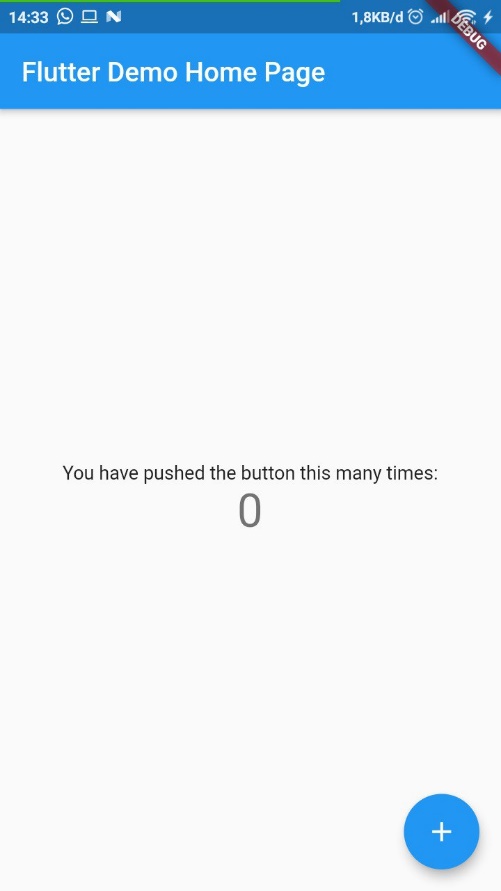
Jika sudah muncul, artinya kita bisa menjalankan aplikasi kita melalui perangkat asli. Untuk menjalankan, silahkan pilih **Run > Start Debugging**.



Pada saat kita running akan muncul tampilan di atas dan muncul running Gradle task di bawah. Tampilan di atas untuk mengontrol aplikasi kita, diantaranya untuk reload aplikasi, restart dan stop aplikasi. Di bagian bawah menginformasikan running aplikasi kita. Di posisi ini kadang akan mendownload file gradle yang sesuai denga settingan aplikasi kita. Di posisi ini juga adalah yang paling berat, karena menggunakan banya resource dari komputer kita. Lamanya proses tergantung dari spesifikasi komputer kita, terutama prosesor, RAM dan Harddisk (lebih baik menggunakan SSD).



Berikut adalah tampilan aplikasi pada saat dijalankan.



**Hal yang perlu diketahui ketika menggunakan flutter yaitu :**

1. **folder lib**

Pada folder lib inilah kode kita akan dibuat.

1. **pubspec.yaml**

Pada flutter semua gambar atau assets yang ditambahkan ke project maka dia wajib

didaftarkan di file pubspec.yaml, file ini juga berfungsi apabila kita ingin menambah

dependensi pada project kita, maka kita perlu menambahkannya disini lalu setelah

itu kita tinggal menjalankan perintah ​flutter pub get​ pada terminal.

1. **build.gradle**

Fungsinya untuk menambah dependensi seperti pada project android kita pada

umumnya.

1. **Info.plist**

Pada file ini berisi konfigurasi permission untuk ios.

**Penjelasan Singkat:**

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

void​ main() => runApp(MyApp());

class​ ​MyApp​ ​extends​ ​StatelessWidget​ {

​*// This widget is the root of your application.*

​@override

Widget build(BuildContext context) {

return​ MaterialApp(

title: ​'Flutter Demo'​,

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: MyHomePage(title: ​'Flutter Demo Home Page'​),

);

}

}

Pada kode tersebut kita melakukan import material.dart fungsinya agar bisa menggunakan widget-widget yang telah disediakan. Lalu pada fungsi main() kita memanggil class MyApp dan melakukan extend StatelessWidget.

class​ ​MyHomePage​ ​extends​ ​StatefulWidget​ {

MyHomePage({Key key, ​this​.title}) : ​super​(key: key);

​final​ ​String​ title;

​@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class​ ​\_MyHomePageState​ ​extends​ ​State​<​MyHomePage​> {

​int​ \_counter = ​0​;

​void​ \_incrementCounter() {

setState(() {

\_counter++;

});

}

​@override

Widget build(BuildContext context) {

​return​ Scaffold(

appBar: AppBar(

title: Text(widget.title),

),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

​'You have pushed the button this many times:'​,

),

Text(

​'$\_counter'​,

style: Theme.of(context).textTheme.display1,

),

],

),

),

floatingActionButton: FloatingActionButton(

onPressed: \_incrementCounter,

tooltip: ​'Increment'​,

child: Icon(Icons.add),

), ​*// This trailing comma makes auto-formatting nicer for*

*buildmethods.*

);

}

}

Pada class MyApp kita mengembalikan/return MaterialApp(), MaterialApp merupakan pondasi dari aplikasi kita berjalan nantinya. Lalu didalamnya terdapat property seperti title, theme, dan home, pada title kita isikan dengan nama aplikasi kita, pada theme kita bisa mengatur tema dari aplikasi kita, lalu pada home inilah kita mengisikan halaman awal pada saat aplikasi dijalankan, disini kita memanggil MyHomePage sebagai halaman awal kita. Pada MyHomePage kita memanggil \_MyHomePageState dan mererturn sebuah Scaffold, scaffold adalah Widget yang bisa kita isi dengan tampilan-tampilan umum pada aplikasi android seperti dukungan AppBar, bagian body, ataupun floating action button.

**RANGKUMAN**

**[Diisi dengan rangkuman materi yang telah diuraikan]**

**TES FORMATIF**

**[Diisi dengan tes formatif sesuai dengan materi yang telah diuraikan]**

**DAFTAR PUSTAKA**

**[Diisi dengan daftar pustaka yang digunakan dalam menyusun modul pembelajaran]**