



# PERTEMUAN 2

## EDITOR VSCODE

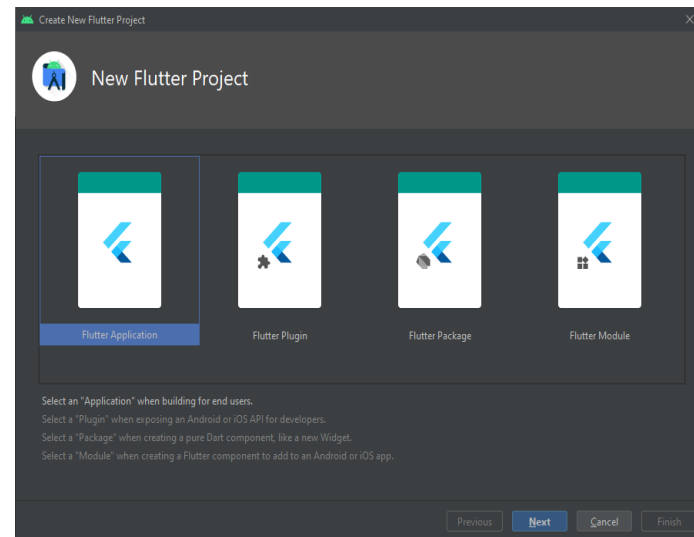
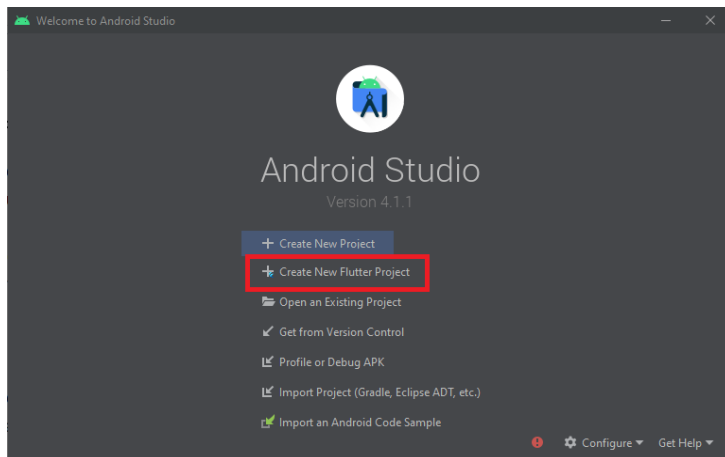


# Pokok Bahasan

- Membuat dan menjalankan Proyek
- Membuat Dan Menjalankan Proyek Dengan VC Code dan Handphone Android
- Struktur Folder Flutter

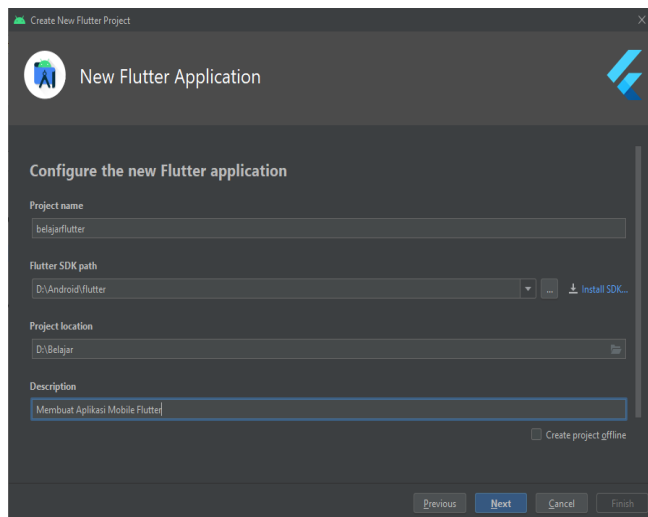
# Membuat dan Menjalankan Proyek

- Jalankan Android Studio kemudian pilih “Create New Flutter Project”
- Kemudian pilih “Flutter Application” dan klik “Next”

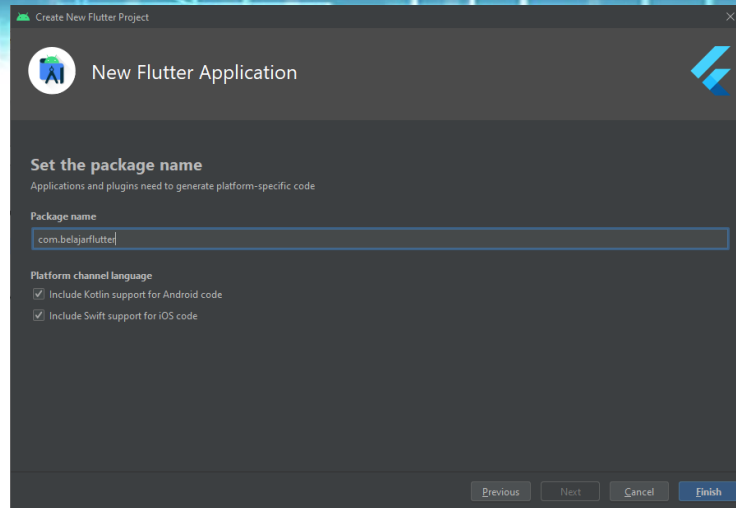




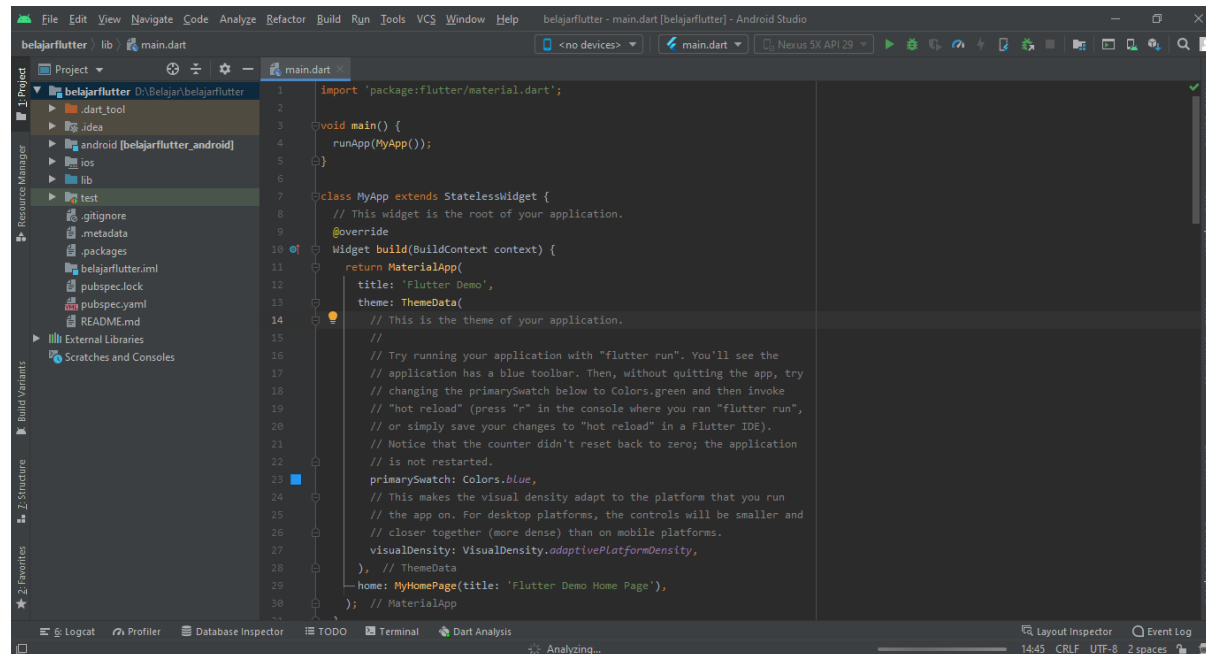
- Kemudian tentukan :
- **Project Name** : belajarflutter
- **Flutter SDK path** : D:\Android\flutter (masukkan sesuai alamat folder flutter diletakkan)
- **Project location** : D:\Belajar (bebas memasukkan dimanapun)
- **Description** : Membuat Aplikasi Mobile Flutter (bebas)
- Kemudian klik tombol “Next”



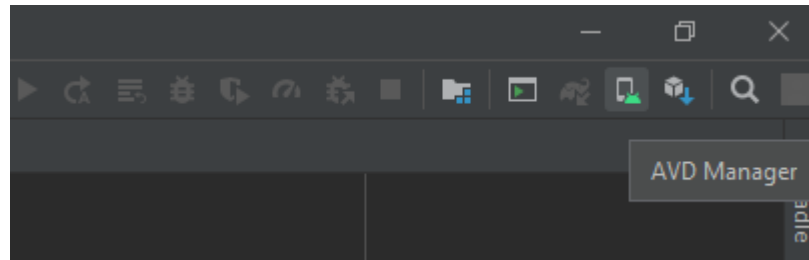
- Kemudian klik tombol “Finish”. Pastikan perangkat terhubung ke internet, karena diperlukan untuk mengunduh proyek yang dibuat



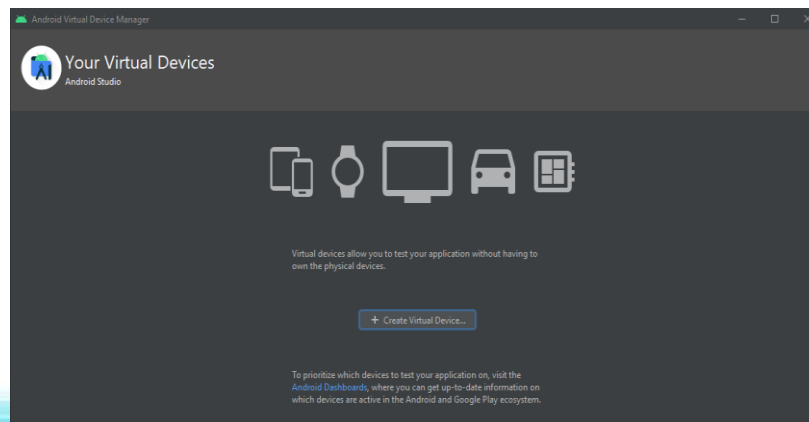
- Setelah selesai akan muncul proyek yang telah dibuat



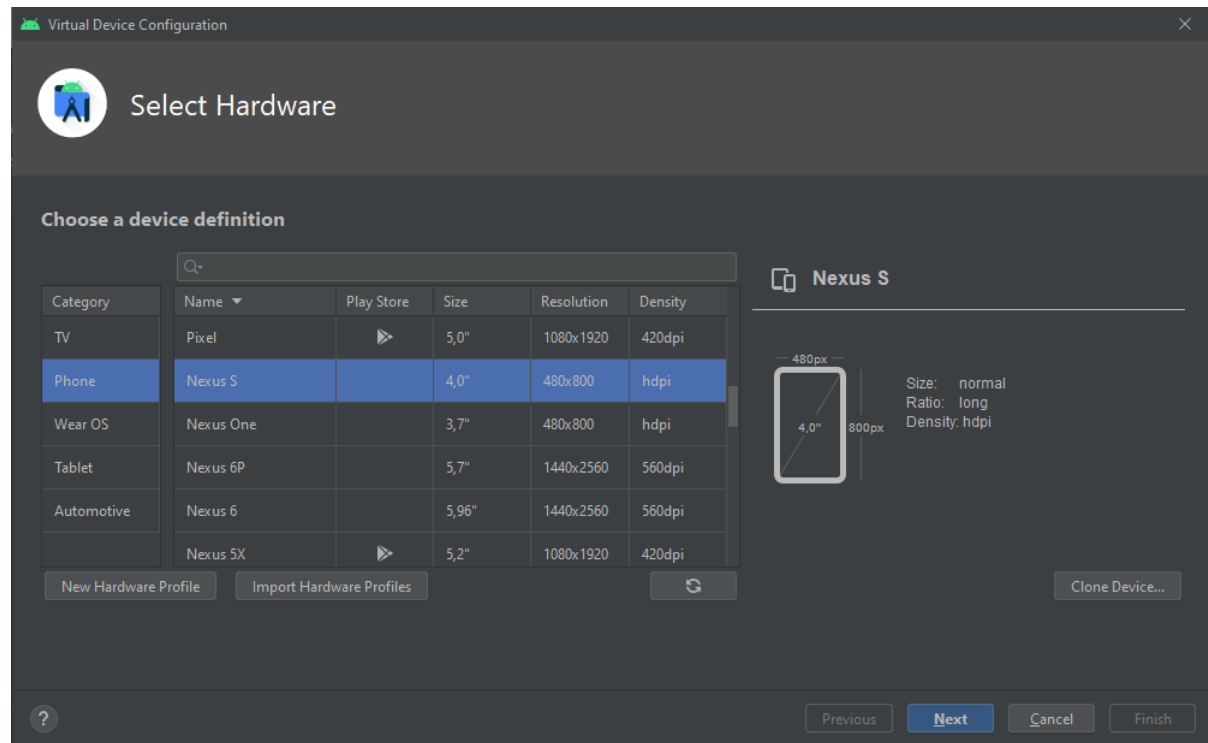
- Untuk menjalankan proyek, kita memerlukan emulator android. Pertama kita akan membuat emulator android dengan cara klik “AVD Manager” pada pojok kiri atas



- Kemudian klik tombol “Create Virtual Device”

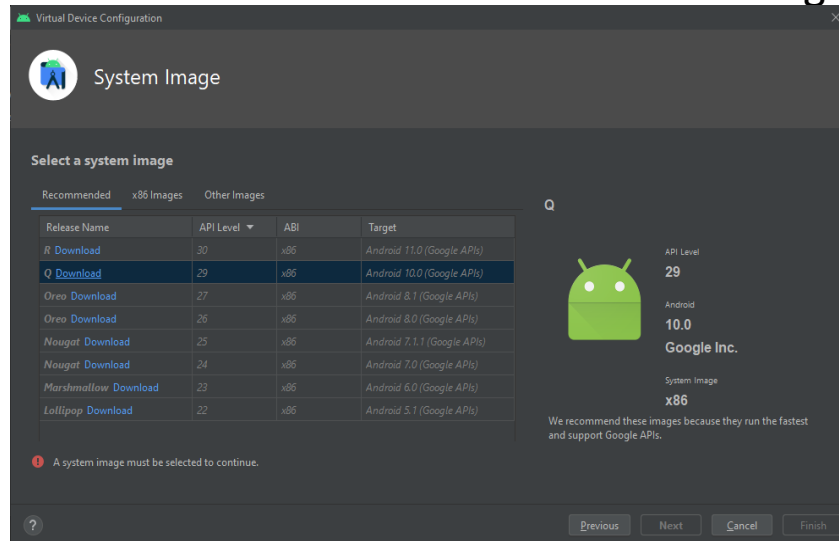


- Pilih salah satu device yang diinginkan, misalnya “Nexus S” kemudian klik tombol “Next”

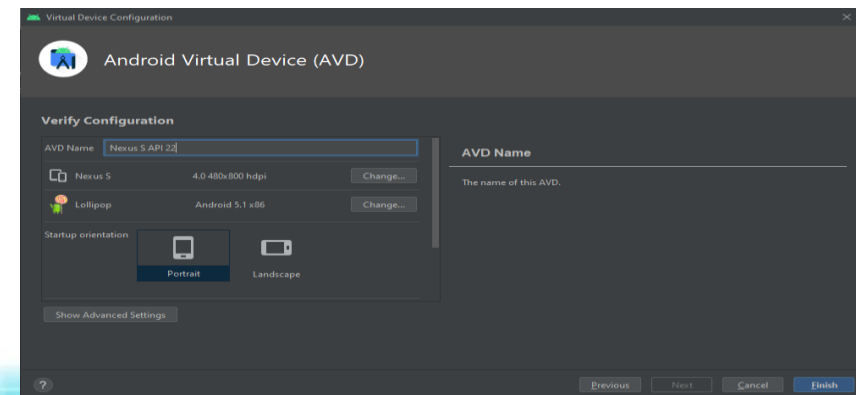




- Kemudian kita akan memilih Versi Sistem Operasi Android, dalam hal ini misalnya “Q”, jika belum ada maka kita akan diminta untuk mengunduh terlebih dahulu.

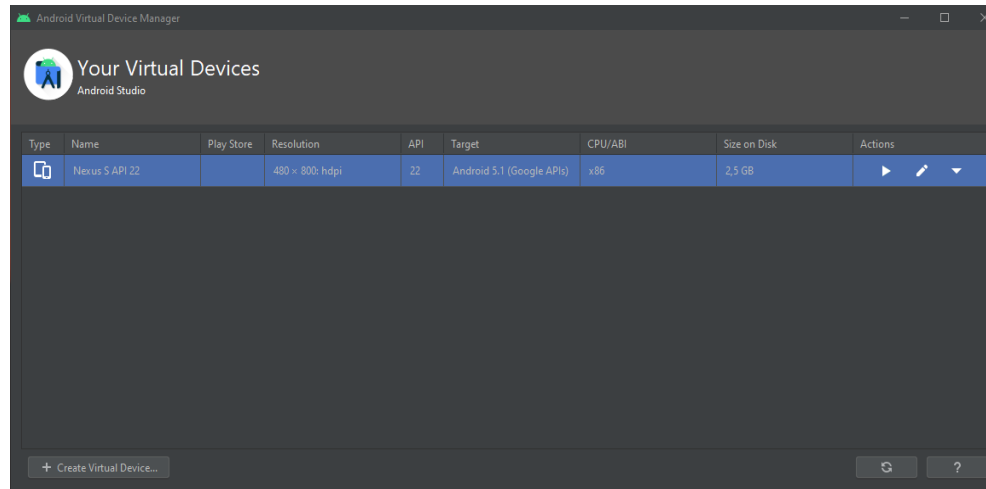


- Setelah itu klik Finish





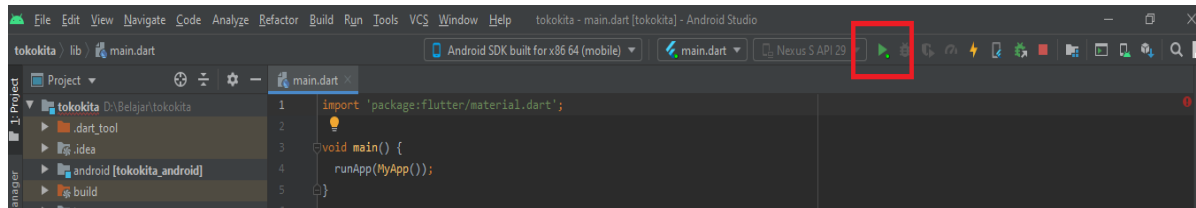
- Untuk Menjalankan Emulator, klik logo play



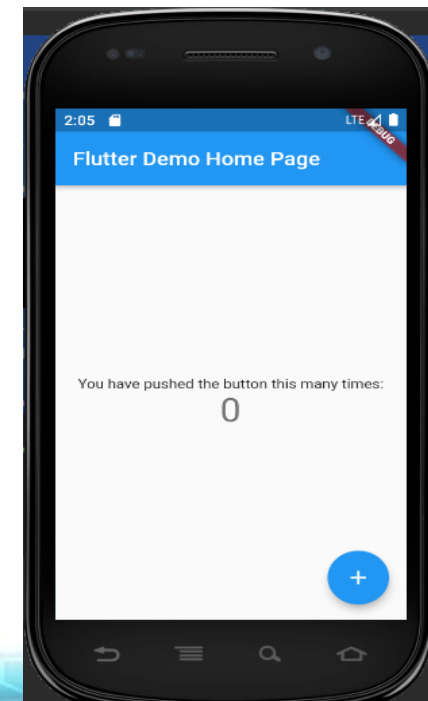
- Kemudian akan muncul emulator android seperti berikut:



- Jika emulator sudah berjalan, kita dapat mengeksekusi proyek dengan cara klik tombol play (berwarna hijau) pada Android Studio



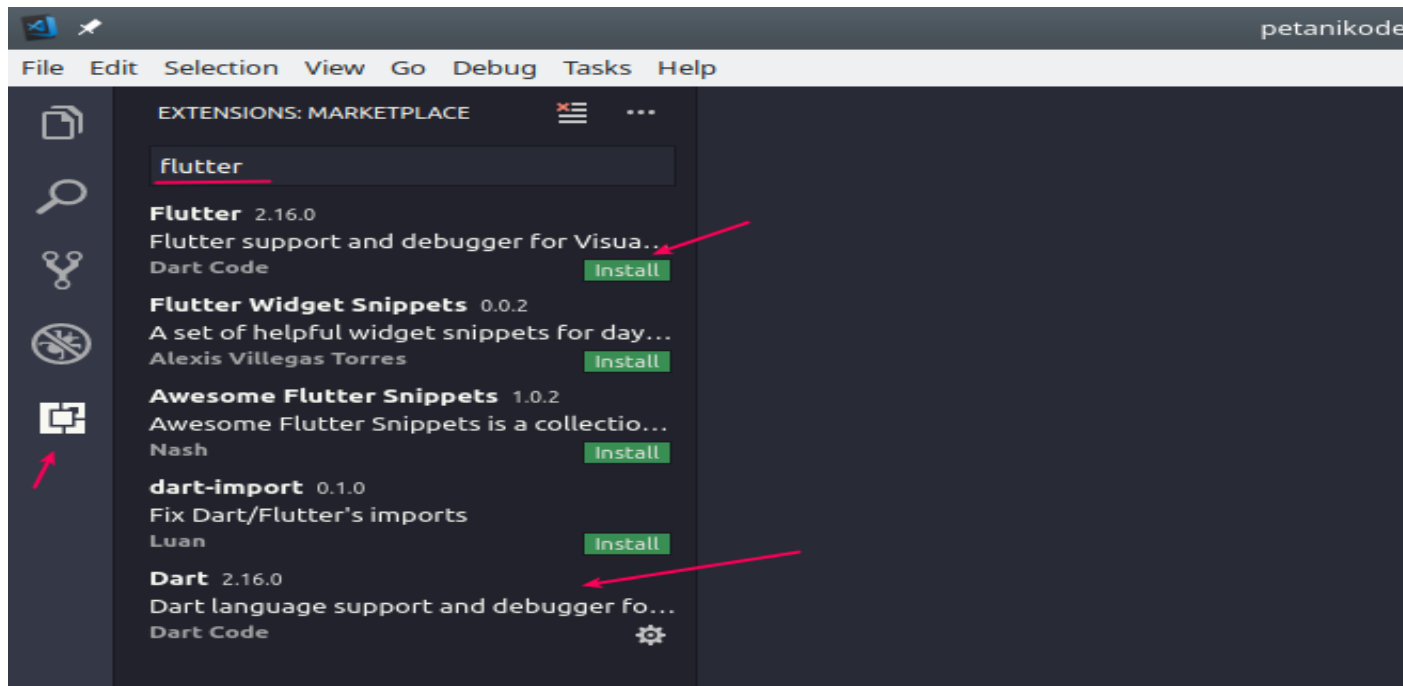
- Pada emulator akan tampil hasil proyek seperti berikut





# Membuat dan menjalankan project dengan VSCode dan Handphone Android

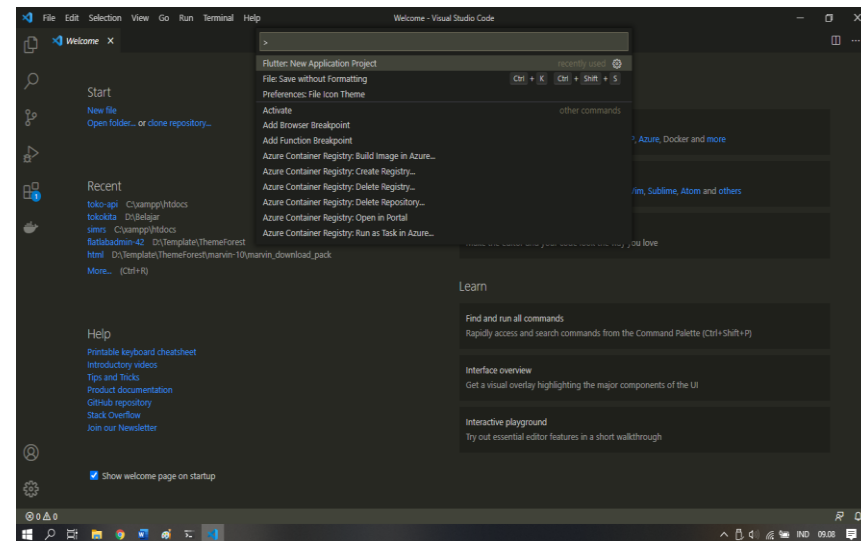
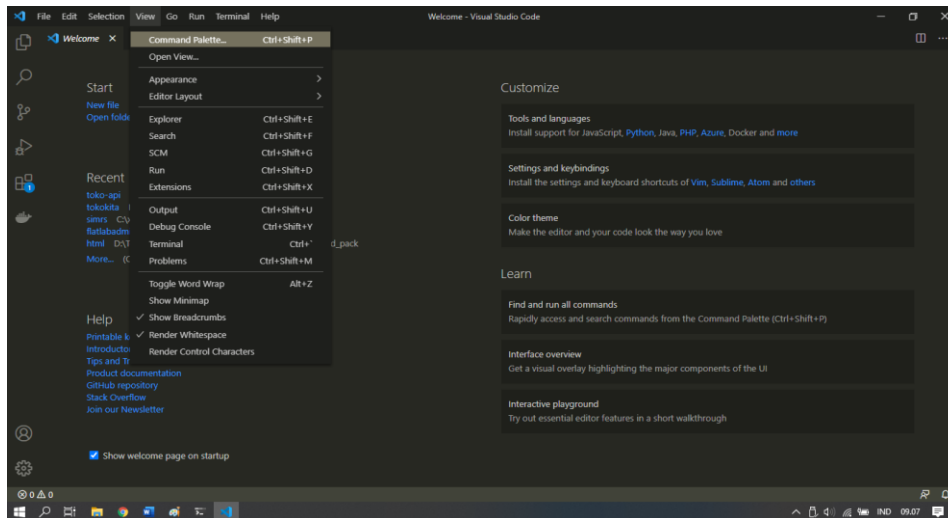
- Instalasi VSCode sebagai alternatif editor
  - Unduh VSCode pada laman <https://code.visualstudio.com/download> kemudian install. Agar flutter dapat digunakan pada VSCode, perlu diinstall beberapa extension flutter yang dibutuhkan.



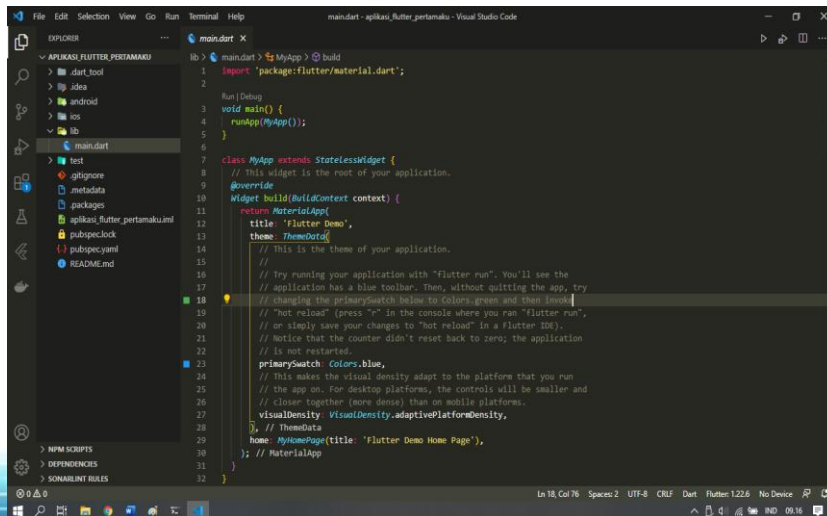
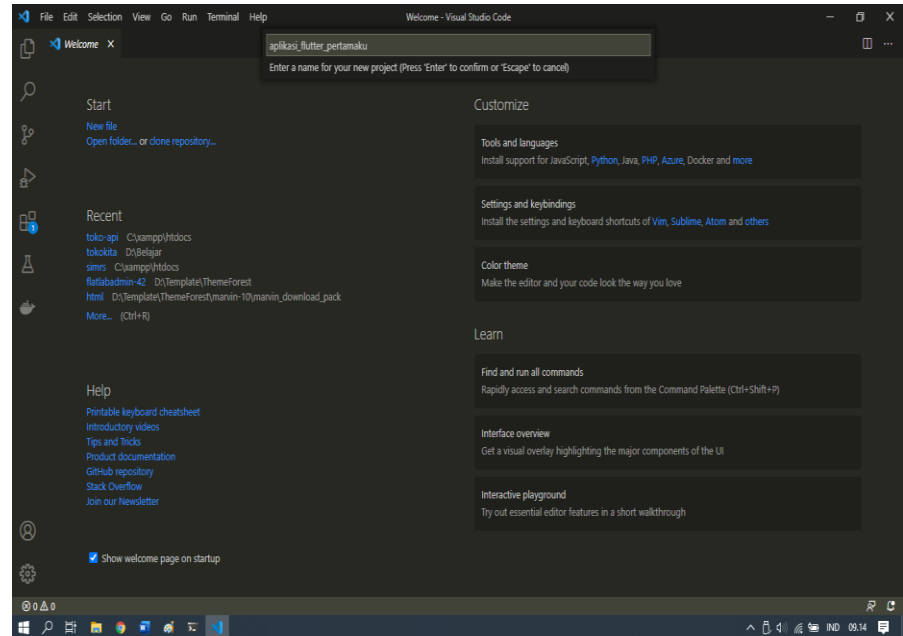
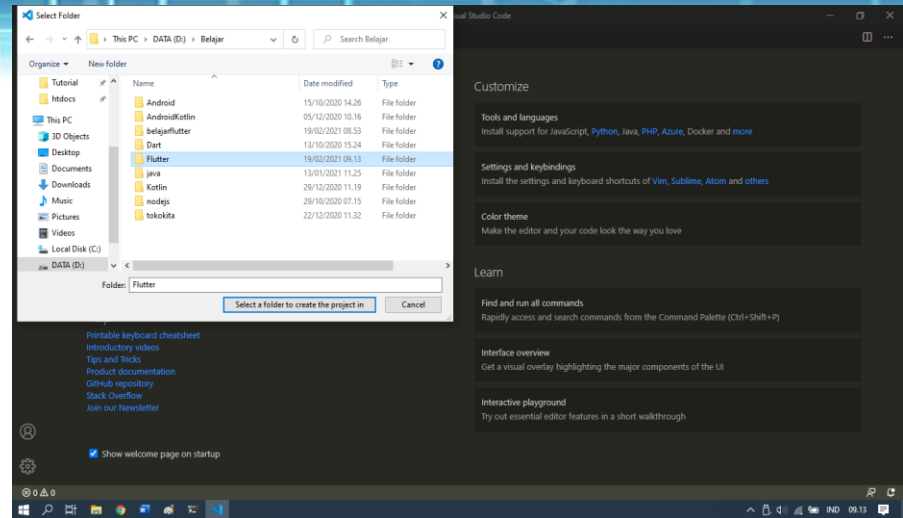
- Setelah itu restart/tutup VSCode

# • Membuat Projek 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 Application Project**

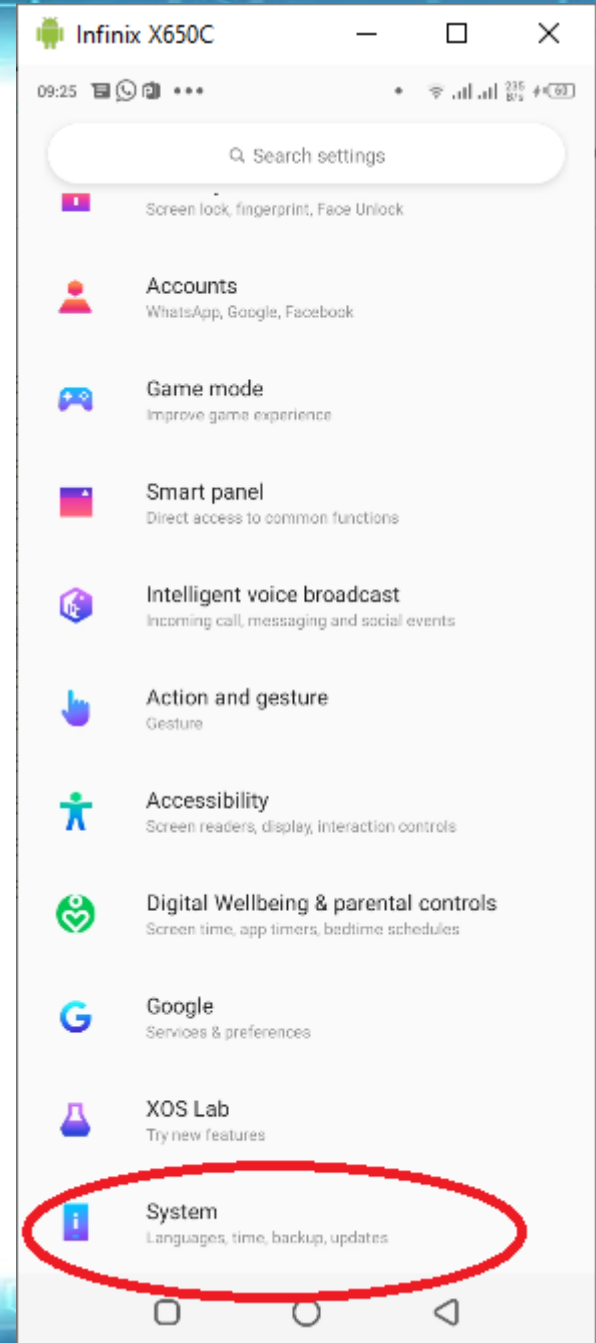


- Pilih folder tempat proyek tersebut
- Kemudian tentukan nama proyek flutter yang ingin dibuat misalnya **aplikasi\_flutter\_pertamaku**
- Kemudian tekan **Enter** dan tunggu hingga proses unduhan selesai

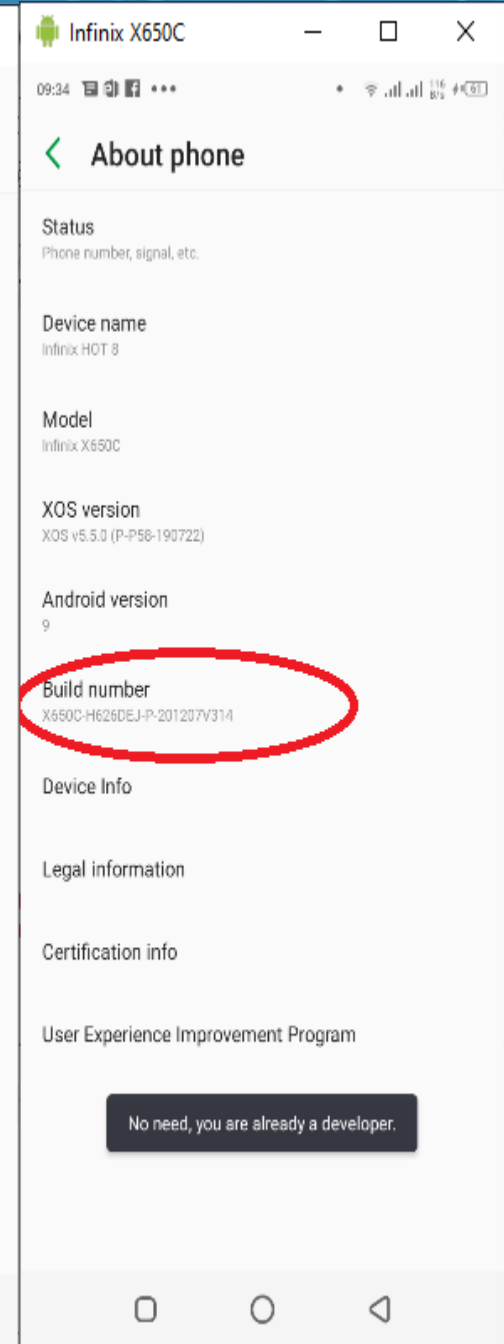
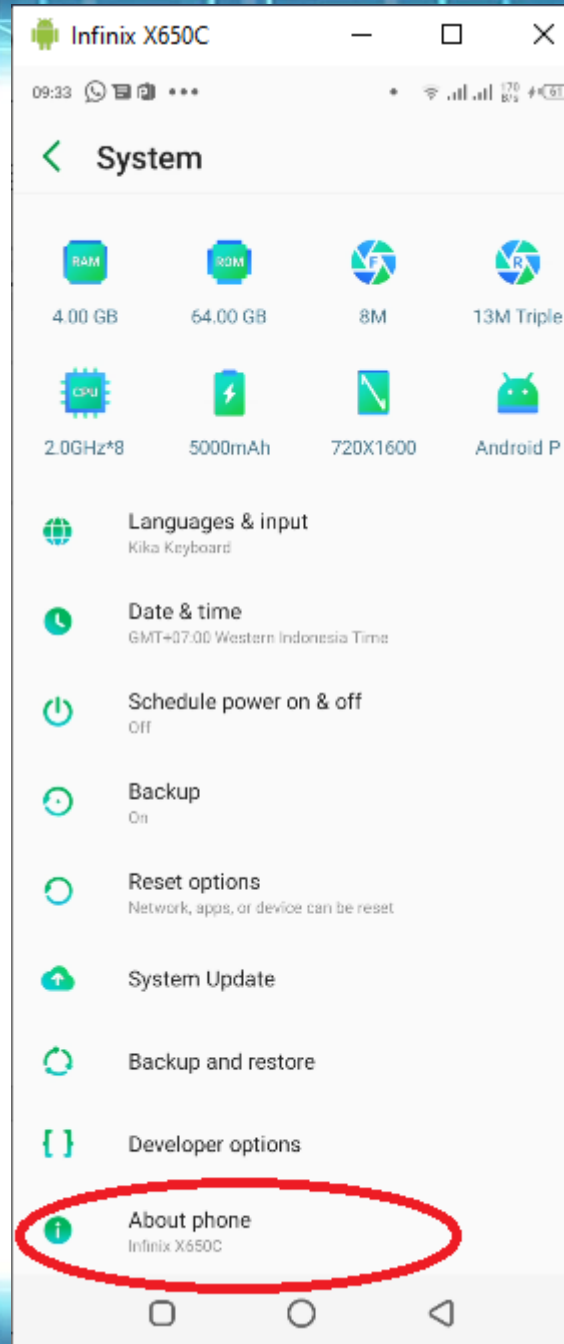




- Menjalankan Aplikasi Flutter Dengan Handphone Android
- Untuk menjalankan projek flutter dari VSCode dapat menggunakan Emulator AVD yang telah dibuat sebelumnya menggunakan Android Studio ataupun menggunakan Device Handphone Android langsung
- Untuk menggunakan android device secara langsung, pertama 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**

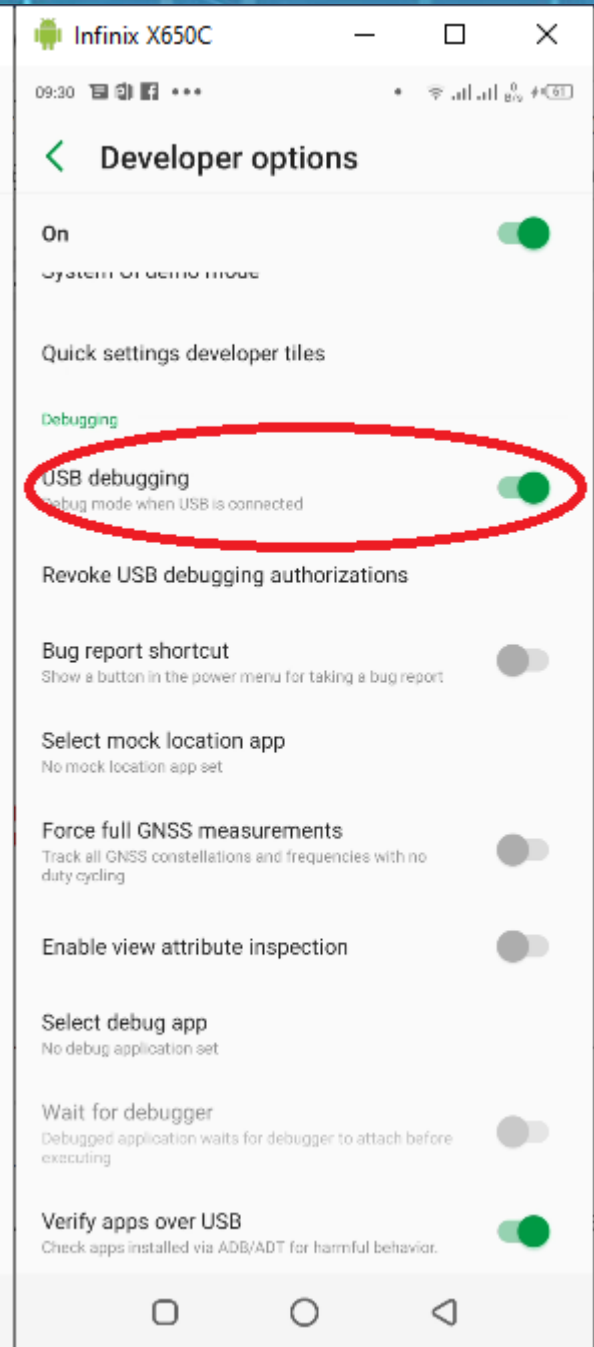
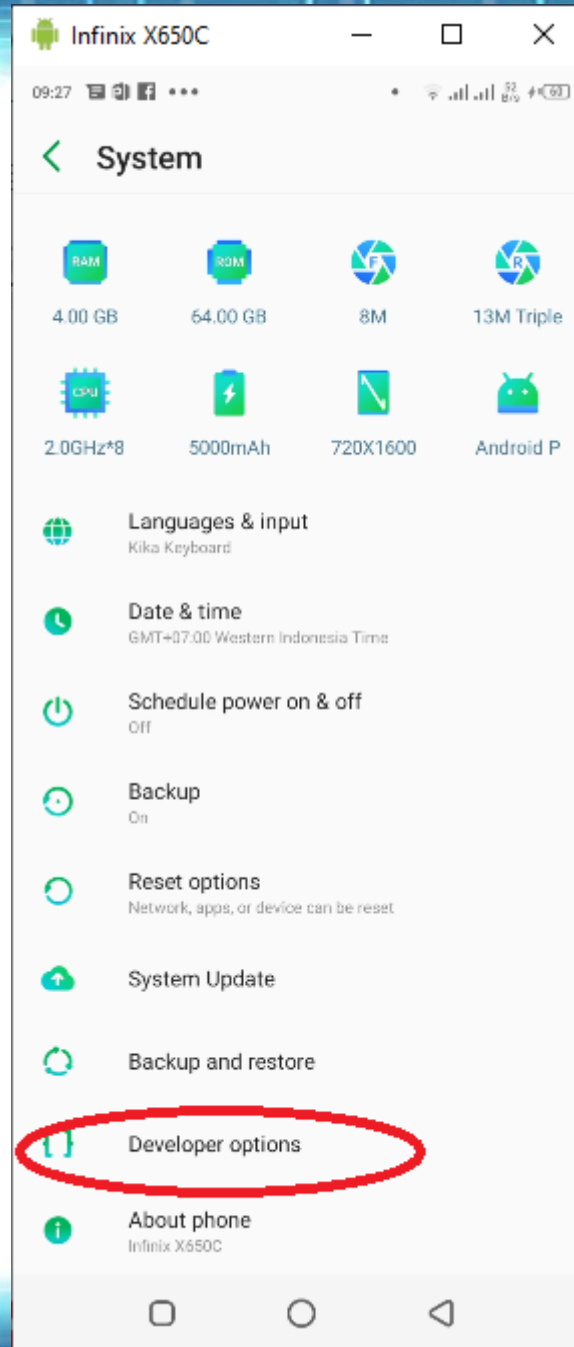


- Kemudian pilih **About Phone**
- Kemudian ketuk **Build number** beberapa kali, namun ini juga berbeda untuk beberapa versi misalnya untuk Xiaomi dengan mengetuk **MIUI Version** beberapa kali





- 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

```

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         // Try running your application with "flutter run". You'll see the
16         // application has a blue toolbar. Then, without quitting the app, try
17         // changing the primarySwatch below to Colors.green and then invoke
18         // "hot reload" (press "r" in the console where you run "flutter run",
19         // or simply save your changes to "hot reload" in a Flutter IDE).
20         // Notice that the counter didn't reset back to zero; the application
21         // is not restarted.
22         primarySwatch: Colors.blue,
23       ),
24       // This makes the visual density adapt to the platform that you run
25       // the app on. For desktop platforms, the controls will be smaller and
26       // closer together (more dense) than on mobile platforms.
27       visualDensity: VisualDensity.adaptivePlatformDensity,
28     );
29   }
30 }

```

- Atau jika pada Android Studio terletak pada toolbar bagian atas tengah

```

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         // Try running your application with "flutter run". You'll see the
16         // application has a blue toolbar. Then, without quitting the app, try
17         // changing the primarySwatch below to Colors.green and then invoke
18         // "hot reload" (press "r" in the console where you run "flutter run",
19         // or simply save your changes to "hot reload" in a Flutter IDE).
20         // Notice that the counter didn't reset back to zero; the application
21         // is not restarted.
22         primarySwatch: Colors.blue,
23       ),
24       // This makes the visual density adapt to the platform that you run
25       // the app on. For desktop platforms, the controls will be smaller and
26       // closer together (more dense) than on mobile platforms.
27       visualDensity: VisualDensity.adaptivePlatformDensity,
28     );
29   }
30 }

```

- 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

```

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
22         // is not restarted.
23         primarySwatch: Colors.blue,
24       ),
25     );
26   }
27 }
  
```

```

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
  
```

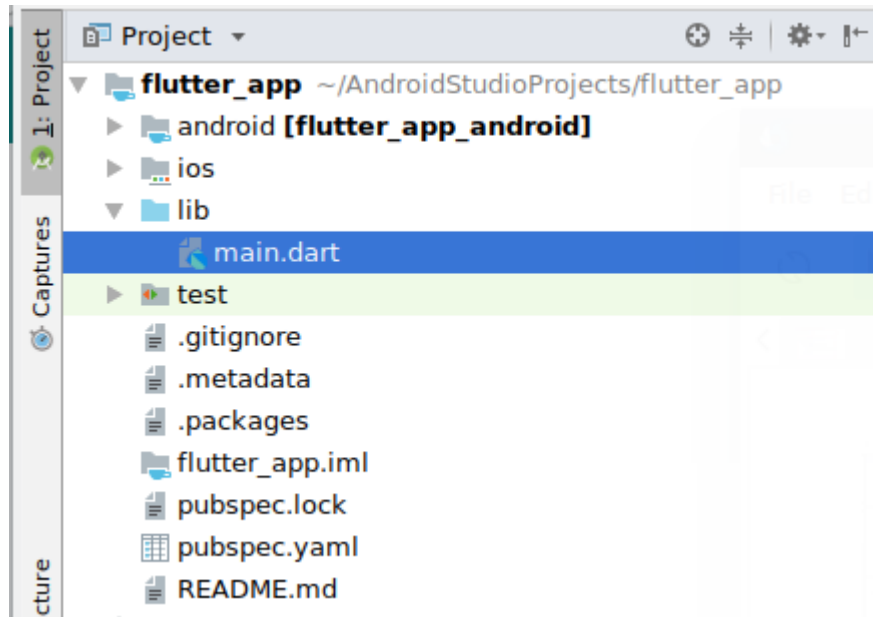
Launching lib/main.dart on Infinix X650C in debug mode...

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

Connecting to VM Service at ws://127.0.0.1:59054/EzY0X0v9j4/ws

# Struktur Folder Flutter

- Adapun struktur folder Proyek flutter adalah sebagai berikut:



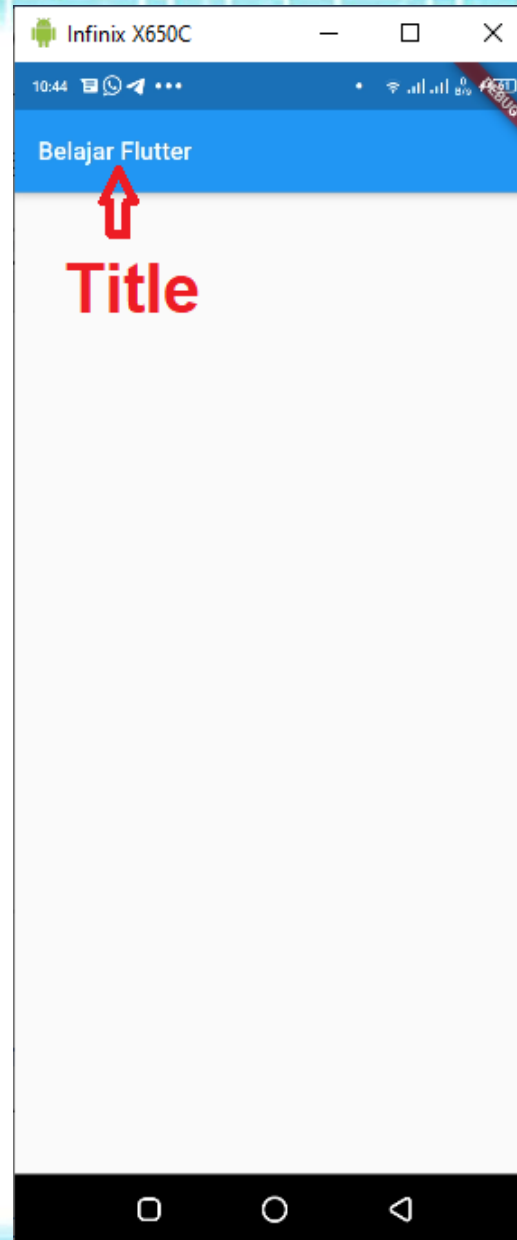
- ✎ **android** berisi source code untuk aplikasi android;
- ✎ **ios** berisi source code untuk aplikasi iOS;
- ✎ **lib** berisi source code Dart, di sini kita akan menulis kode aplikasi;
- ✎ **test** berisi source code Dart untuk testing aplikasi;
- ✎ **.gitignore** adalah file [Git](#);
- ✎ **.metadata** merupakan file yang berisi metadata project yang di-generate otomatis;
- ✎ **.packages** merupakan file yang berisi alamat path package yang dibuat oleh pub;
- ✎ **flutter\_app.iml** merupakan file XML yang berisi keterangan project;
- ✎ **pubspec.lock** merupakan file yang berisi versi-versi library atau package. File ini dibuat oleh pub. Fungsinya untuk mengunci versi package.
- ✎ **pubspec.yaml** merupakan file yang berisi informasi tentang project dan library yang dibutuhkan;
- ✎ **README.md** merupakan file markdown yang berisi penjelasan tentang source code.

- Membuat “Hello World”
- Buka proyek **aplikasi\_flutter\_pertamaku** menggunakan VSCode agar lebih ringan. Buka file **main.dart** yang terletak pada folder **lib** kemudian ubah menjadi

```
1. import 'package:flutter/material.dart';
2.
3. void main() {
4.   runApp(MyApp());
5. }
6.
7. class MyApp extends StatelessWidget {
8.   @override
9.   Widget build(BuildContext context) {
10.    return MaterialApp(
11.      title: 'Aplikasi Flutter Pertama',
12.      home: Scaffold(
13.        appBar: AppBar(
14.          title: Text('Belajar Flutter'),
15.        ),
16.      ),
17.    );
18.  }
19. }
```



- Ketika dijalankan akan menghasilkan



← AppBar

- Kemudian untuk menambahkan tampilan dibagian dalam (body), pada fungsi Scaffold terdapat parameter **body**. Silahkan modifikasi **main.dart** menjadi seperti berikut

```
1. import 'package:flutter/material.dart';
2.
3. void main() {
4.   runApp(MyApp());
5. }
6.
7. class MyApp extends StatelessWidget {
8.   @override
9.   Widget build(BuildContext context) {
10.    return MaterialApp(
11.      title: 'Aplikasi Flutter Pertama',
12.      home: Scaffold(
13.        appBar: AppBar(
14.          title: Text('Belajar Flutter'),
15.        ),
16.        body: Center(
17.          child: Text('Hello World'),
18.        ),
19.      ),
20.    );
21.  }
22. }
```



- Hasilnya



Pada aplikasi di atas, kita membuat ***StatelessWidget*** yang berisi widget **MaterialApp()**. Kemudian di dalam **MaterialApp()** berisi widget lagi: **Scaffold**, **AppBar**, **Center**, dan **Text**.

Ini adalah widget dasar.

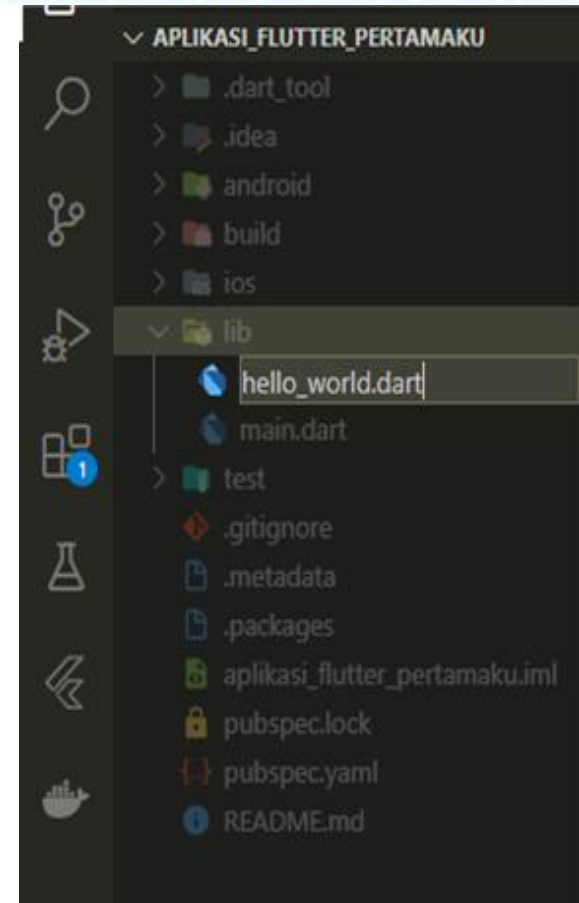
Penjelasan:

- MyApp adalah StatelessWidget, merupakan widget induk;
- MaterialApp adalah widget yang membungkus beberapa widget yang menggunakan tema material design
- Scaffold adalah widget untuk struktur dasar material design;
- AppBar adalah widget untuk membuat AppBar;
- Center adalah Widget layout untuk membuat widget ke tengah;
- Text adalah widget untuk membuat teks.

Untuk mempermudah dalam pembacaan kode dan maintenance dapat dilakukan dengan memisahkan **MyApp** dengan halaman yang ingin ditampilkan.

- Silahkan buat sebuah file dengan nama **hello\_world.dart** di dalam folder **lib**
- Kemudian bagian **Scaffold** pada **main.dart** yang telah dibuat tadi akan kita masukkan ke dalam **hello\_world.dart**, sehingga pada **hello\_world.dart** akan menjadi

```
1. import 'package:flutter/material.dart';
2.
3. class HelloWorld extends StatelessWidget {
4.   @override
5.   Widget build(BuildContext context) {
6.     return Scaffold(
7.       appBar: AppBar(
8.         title: Text('Belajar Flutter'),
9.       ),
10.      body: Center(
11.        child: Text('Hello World'),
12.      ),
13.    );
14.  }
15. }
```



- Pada file **main.dart** kita modifikasi kembali pada bagian **home** menjadi

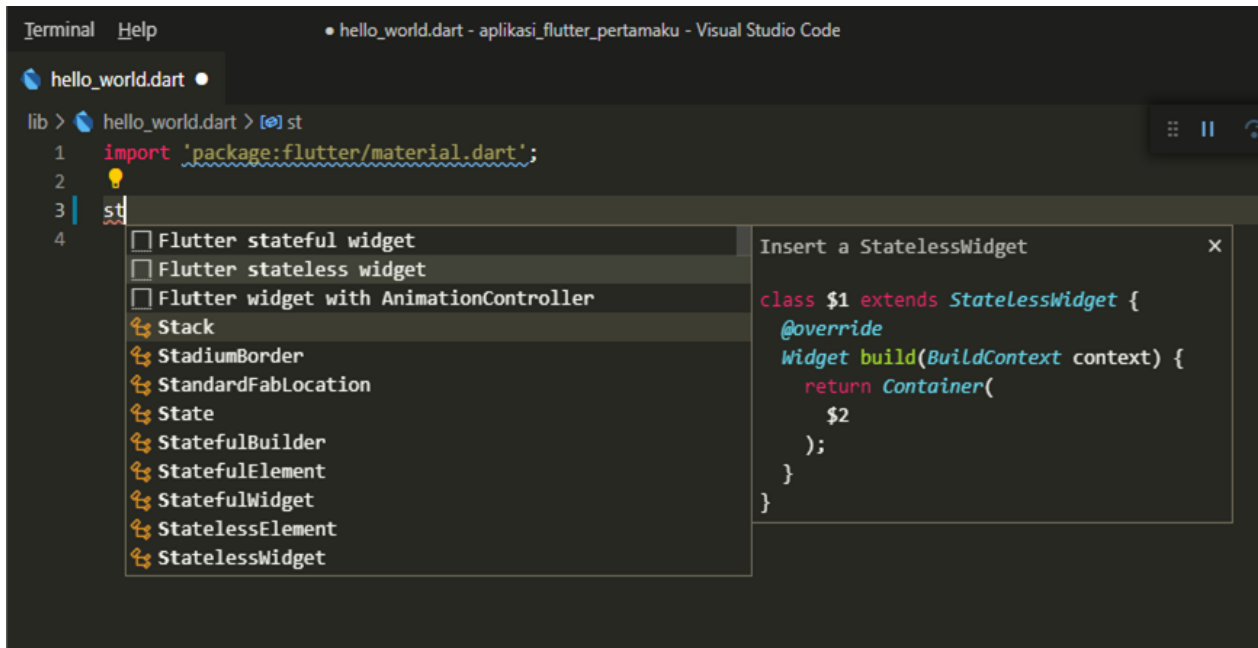
```
1. import 'package:aplikasi_flutter_pertamaku/hello_world.dart';
2. import 'package:flutter/material.dart';
3.
4. void main() {
5.   runApp(MyApp());
6. }
7.
8. class MyApp extends StatelessWidget {
9.   @override
10.  Widget build(BuildContext context) {
11.    return MaterialApp(
12.      title: 'Aplikasi Flutter Pertama',
13.      home: HelloWorld(),
14.    );
15.  }
16. }
```

- Pada bagian home, kita memanggil class **HelloWorld** yang telah kita buat sebelumnya pada file **hello\_world.dart**
- Jika kita perhatikan pada bagian body, terdapat Widget Center kemudian didalam Widget Center tersebut terdapat parameter child untuk meletakkan Widget lain didalam widget tersebut, dalam hal ini adalah Widget Text

```
Center(  
  child: Text('Hello World'),  
),
```

- Catatan : dalam Widget selain **child**, terdapat pula **children** dengan type data array yang dimana kita dapat menempatkan beberapa Widget didalamnya contohnya pada Widget Column dan Row

- Untuk mempercepat dalam pembuatan class pada VSCode dapat dilakukan dengan mengetik **st** kemudian memilih **stateless widget** ataupun **stateless widget** kemudian ketikkan nama class yang diinginkan



```
lib > hello_world.dart > st
1 import 'package:flutter/material.dart';
2
3 st
4
Flutter stateful widget
Flutter stateless widget
Flutter widget with AnimationController
Stack
StadiumBorder
StandardFabLocation
State
StatefulBuilder
StatefulElement
StatefulWidget
StatelessElement
StatelessWidget
```

```
class $1 extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container(
      $2
    );
  }
}
```

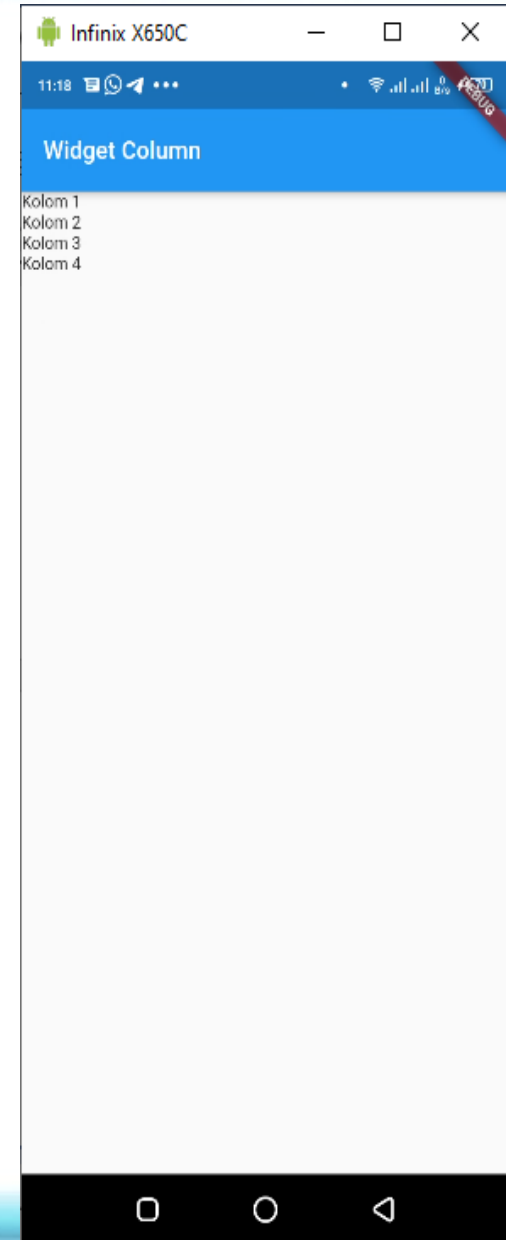


## • Membuat Widget Column

- Buat sebuah file dengan nama **column\_widget.dart** didalam folder lib, kemudian ketikkan kode berikut

```
1. import 'package:flutter/material.dart';
2.
3. class ColumnWidget extends StatelessWidget {
4.   @override
5.   Widget build(BuildContext context) {
6.     return Scaffold(
7.       appBar: AppBar(
8.         title: Text('Widget Column'),
9.       ),
10.      body: Column(
11.        children: [
12.          Text('Kolom 1'),
13.          Text('Kolom 2'),
14.          Text('Kolom 3'),
15.          Text('Kolom 4'),
16.        ],
17.      ),
18.    );
19.  }
20. }
```

- Column biasanya digunakan untuk membuat Form

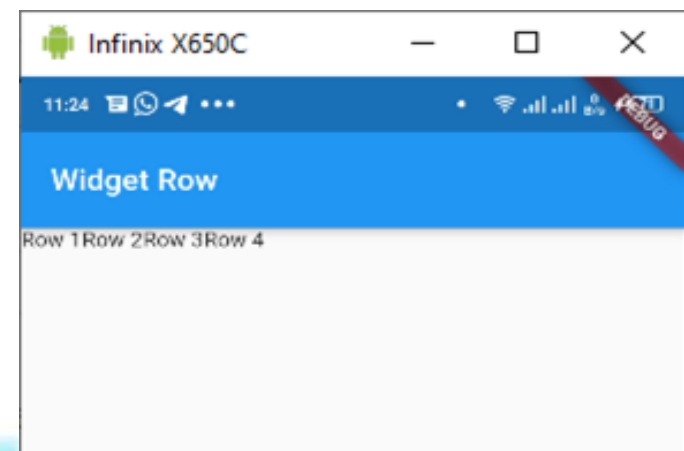




## • Membuat widget Row

- Untuk menampilkan Widget dalam posisi horizontal dapat menggunakan Widget Row. Buat sebuah file didalam folder **lib** dengan nama **row\_widget.dart**, kemudian ketikkan kode berikut
- Kemudian seperti sebelumnya masukkan class **RowWidget** tersebut kedalam home pada **main.dart**, dan hasilnya akan menjadi

```
1. import 'package:flutter/material.dart';
2.
3. class RowWidget extends StatelessWidget {
4.   @override
5.   Widget build(BuildContext context) {
6.     return Scaffold(
7.       appBar: AppBar(
8.         title: Text('Widget Row'),
9.       ),
10.      body: Row(
11.        children: [
12.          Text('Row 1'),
13.          Text('Row 2'),
14.          Text('Row 3'),
15.          Text('Row 4'),
16.        ],
17.      ),
18.    );
19.  }
20. }
```



- **Mengenal StatelessWidget dan StatefulWidget**

- **StatelessWidget** adalah class widget yang propertinya *immutable*, artinya nilainya tidak bisa diubah, sedangkan **StatefulWidget** nilainya dapat berubah-ubah.

#### Contoh StatelessWidget :

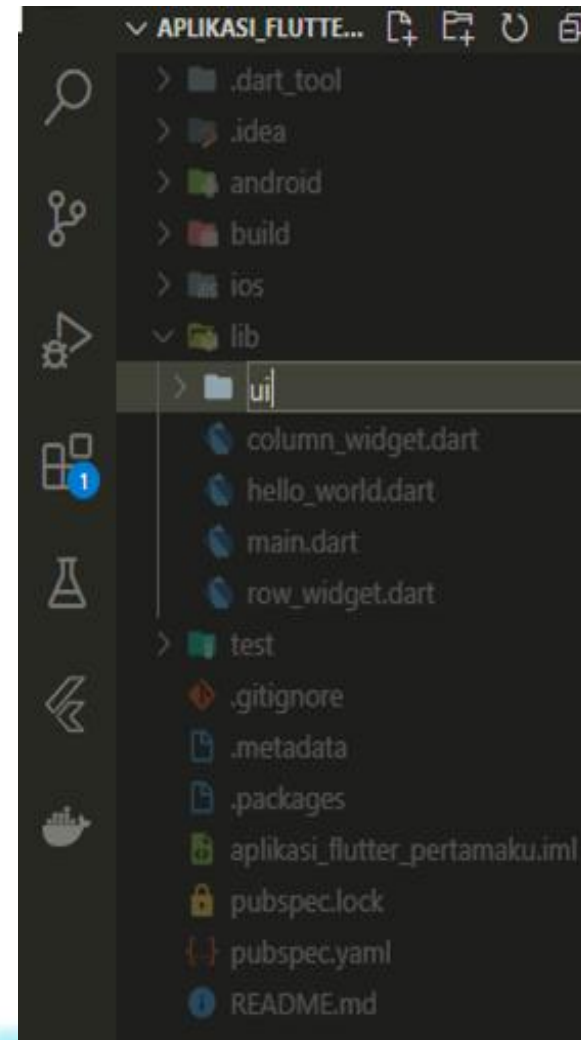
```
1. class HelloWorld extends StatelessWidget {  
2.   @override  
3.   Widget build(BuildContext context) {  
4.     return Container(  
5.  
6.   );  
7. }  
8. }
```

#### Contoh StatefulWidget

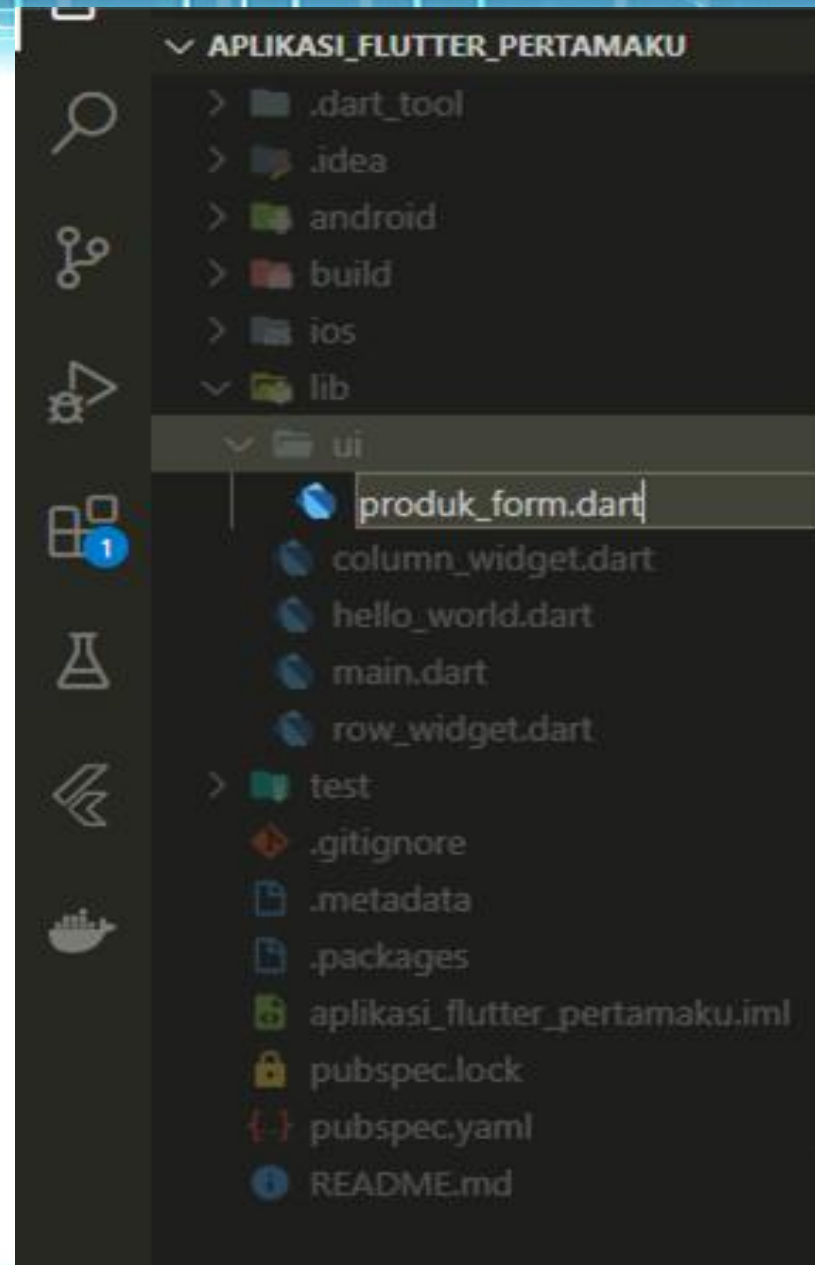
```
1. class HelloWorld extends StatefulWidget {  
2.   @override  
3.   _HelloWorldState createState() => _HelloWorldState();  
4. }  
5.  
6. class _HelloWorldState extends State<HelloWorld> {  
7.   @override  
8.   Widget build(BuildContext context) {  
9.     return Container(  
10.  
11.   );  
12. }  
13. }
```

# TUGAS PERTEMUAN 2

1. Membuat Form dengan flutter dengan mengikuti Langkah-langkah seperti berikut ini:
  - a. Untuk membuat form dengan flutter, agar lebih rapi untuk tampilan halaman akan kita kelompokkan dalam sebuah folder tersendiri, dalam hal ini kita membuat folder dengan nama **ui** didalam folder **lib**.



- Kemudian didalam folder **ui** tersebut kita buat sebuah file dengan nama **produk\_form.dart**



- Kemudian Ketikkan kode berikut

```
1. import 'package:flutter/material.dart';
2.
3. class ProdukForm extends StatefulWidget {
4.   @override
5.   _ProdukFormState createState() => _ProdukFormState();
6. }
7.
8. class _ProdukFormState extends State<ProdukForm> {
9.   @override
10.  Widget build(BuildContext context) {
11.    return Scaffold(
12.      appBar: AppBar(
13.        title: Text('Form Produk'),
14.      ),
15.      body: SingleChildScrollView(
16.        child: Column(
17.          children: [
18.            TextField(
19.              decoration: InputDecoration(labelText: "Kode Produk"),
20.            ),
21.            TextField(
22.              decoration: InputDecoration(labelText: "Nama Produk"),
23.            ),
24.            TextField(
25.              decoration: InputDecoration(labelText: "Harga"),
26.            ),
27.            RaisedButton(
28.              child: Text('Simpan'),
29.              onPressed: () {},
30.            )
31.          ],
32.        ),
33.      ),
34.    );
35.  }
36. }
```



b. Ubah pada **main.dart** dengan memanggil class **ProdukForm**, sehingga hasilnya akan menjadi



The screenshot shows a mobile application interface on an Infinix X650C. The status bar at the top displays the time 13:02, battery level at 93%, and various connectivity icons. The app's title bar is blue and labeled 'Form Produk'. Below the title bar, there are three text input fields labeled 'Kode Produk', 'Nama Produk', and 'Harga'. At the bottom of the form, there is a grey button labeled 'Simpan'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.