

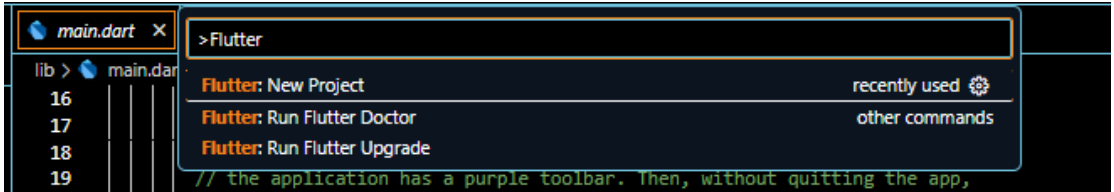
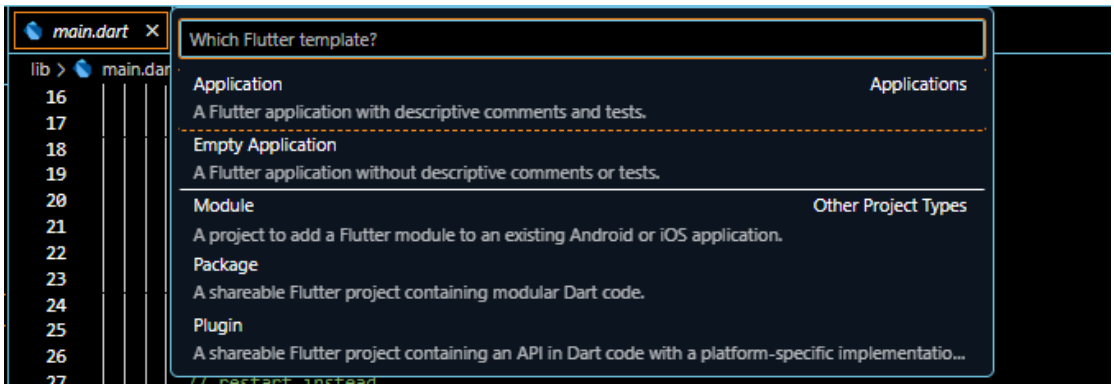


Mata Kuliah : Pemrograman Mobile  
Program Studi : D4 – Sistem Informasi Bisnis  
Semester : 5

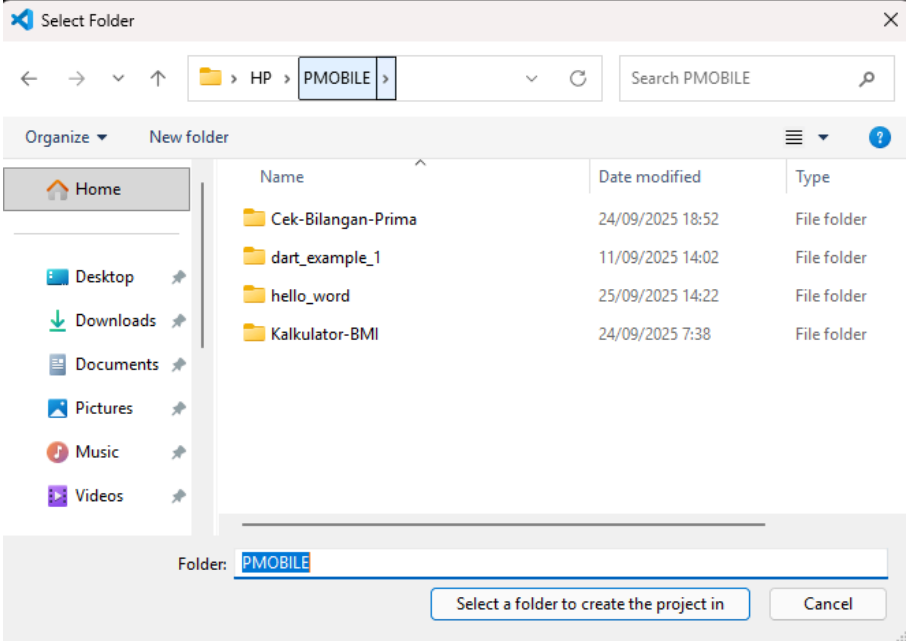
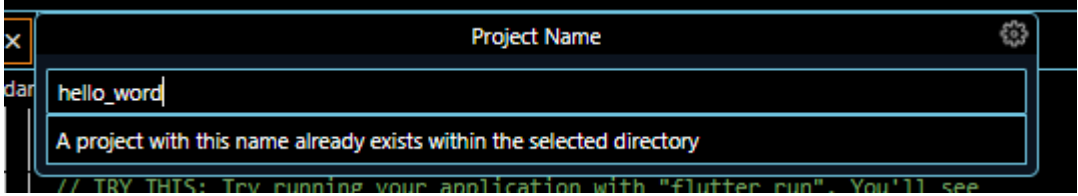
Kelas : SIB - 3D  
NIM : 2341760178  
Nama : Saria Fauzani  
Jobsheet Ke- : 4  
Link Github : <https://github.com/Sariafauzani/flutter-fundamental-part1>

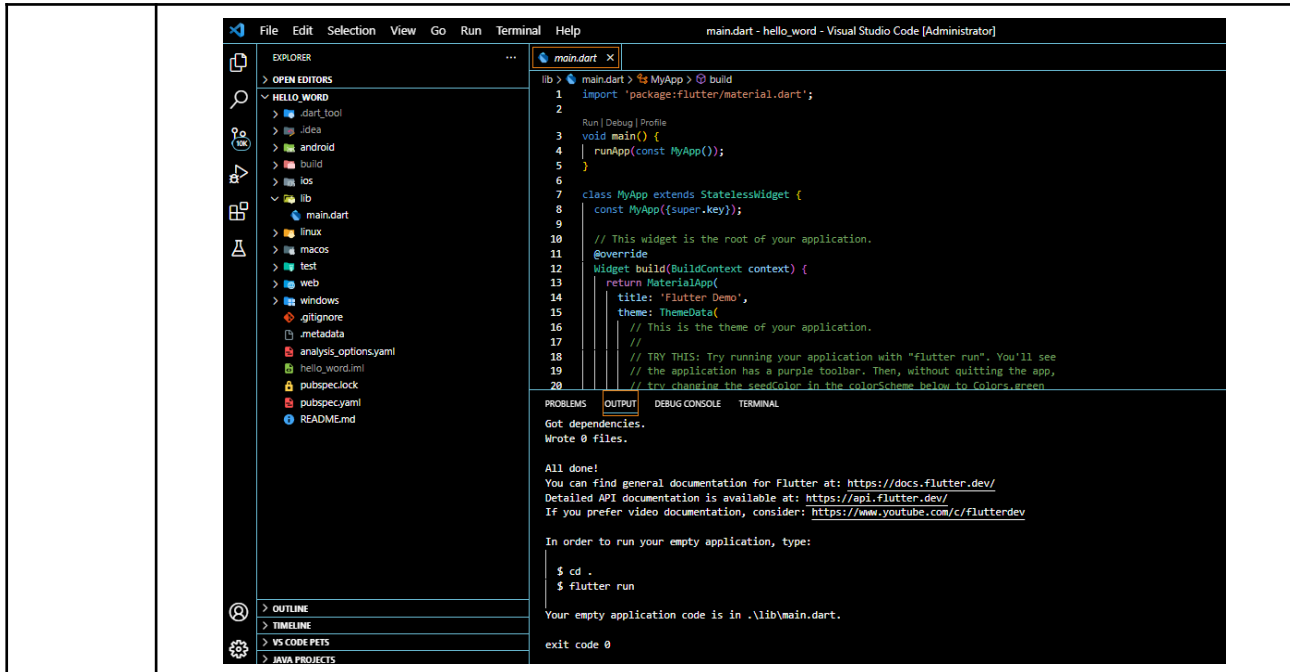
## Laporan Jobsheet

### Praktikum Ke-1 : Membuat Project Flutter Baru

Langkah	Jawaban/Deskripsi
1	<p>Buka VS Code, lalu tekan tombol Ctrl + Shift + P maka akan tampil Command Palette, lalu ketik Flutter. Pilih New Application Project.</p>  
2	<p>Kemudian buat folder sesuai style laporan praktikum yang Anda pilih. Disarankan pada folder dokumen atau desktop atau alamat folder lain yang tidak terlalu dalam atau panjang. Lalu pilih Select a folder to create the project in.</p>



	
3	<p>Buat nama project flutter hello_world seperti berikut, lalu tekan Enter. Tunggu hingga proses pembuatan project baru selesai.</p> 
4	<p>Jika sudah selesai proses pembuatan project baru, pastikan tampilan seperti berikut. Pesan akan tampil berupa "Your Flutter Project is ready!" artinya Anda telah berhasil membuat project Flutter baru.</p>

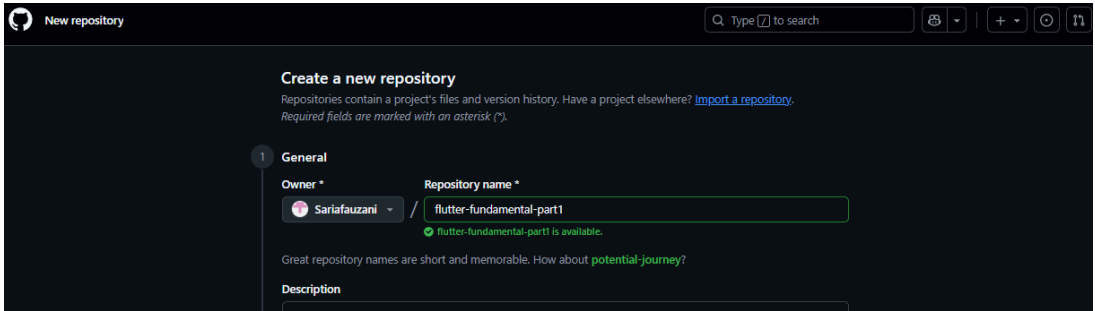
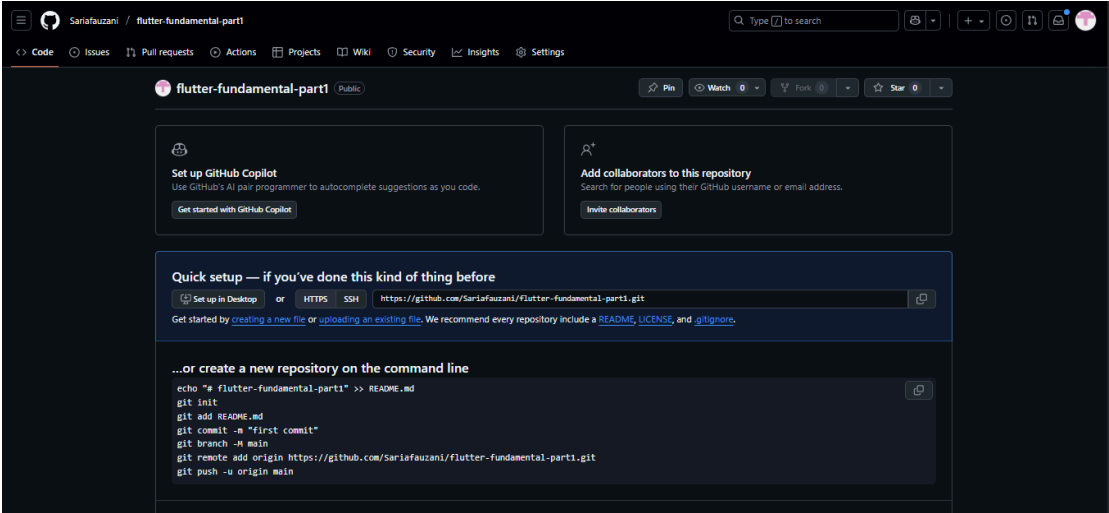
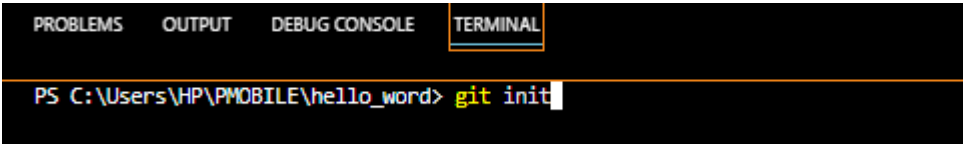


## Praktikum Ke-2 : Menghubungkan Perangkat Android atau Emulator

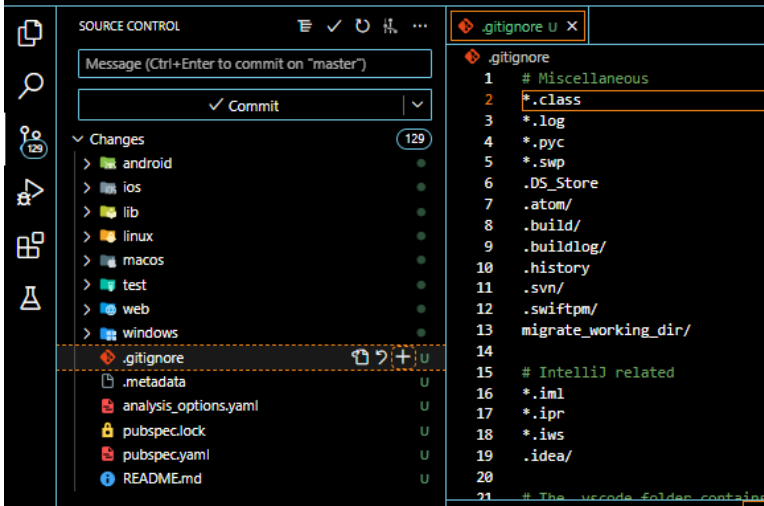
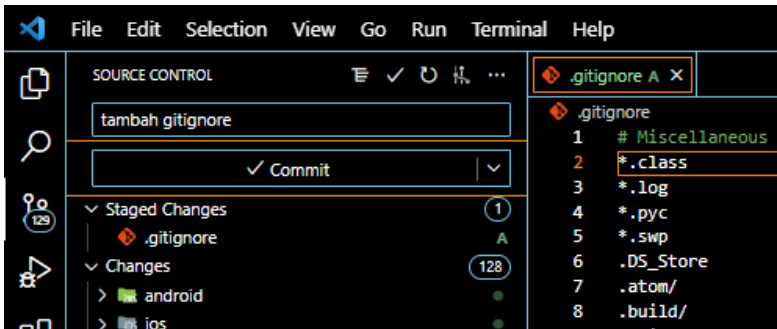
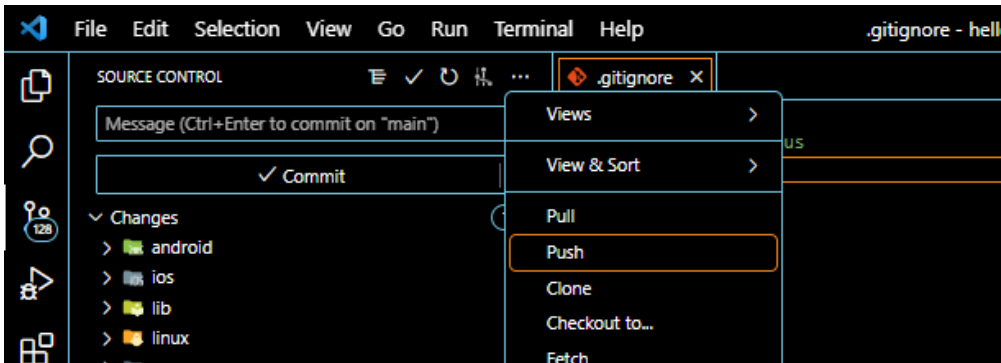
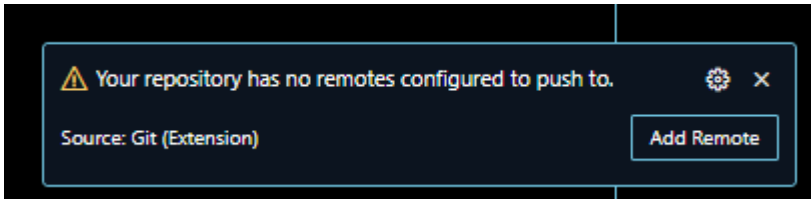
Langkah	Jawaban/Deskripsi
1	



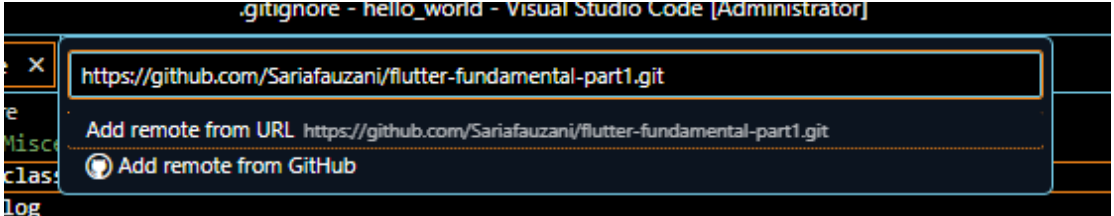
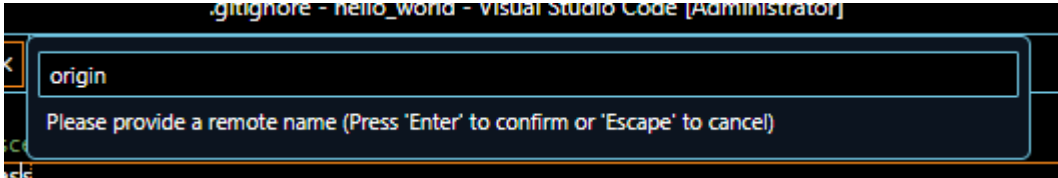
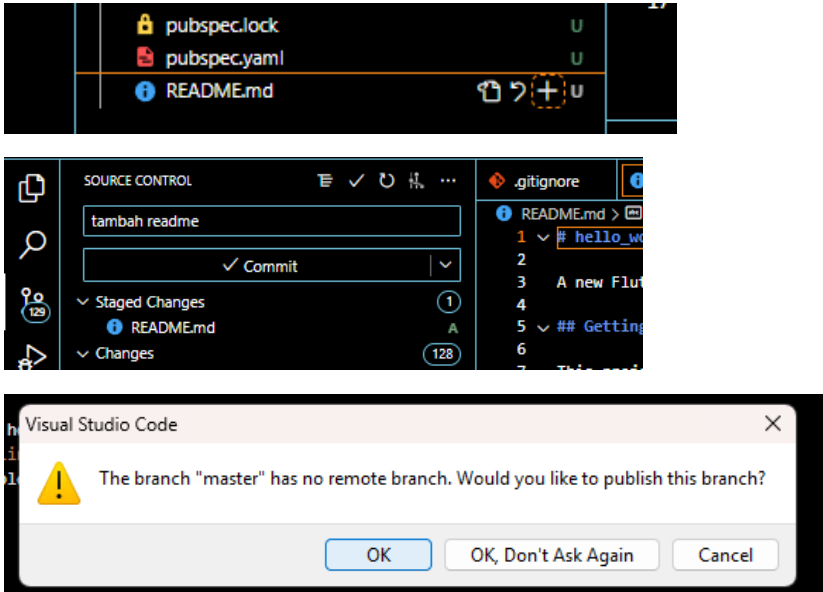
### Praktikum Ke-3 : Membuat Repository GitHub dan Laporan Praktikum

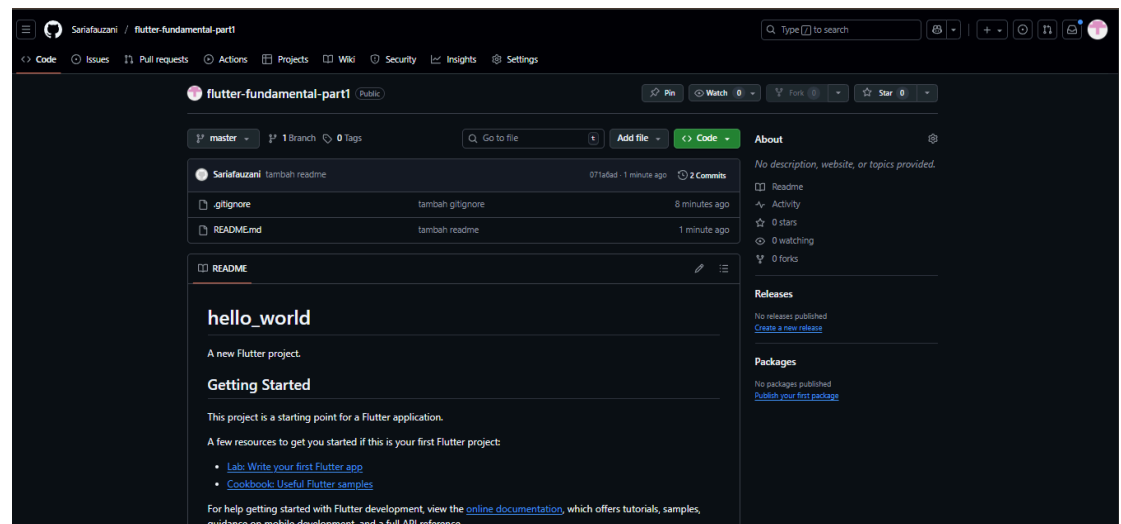
Langkah	Jawaban/Deskripsi
1	<p>Login ke akun GitHub Anda, lalu buat repository baru dengan nama "flutter-fundamental-part1"</p> 
2	<p>Lalu klik tombol "Create repository" lalu akan tampil seperti gambar berikut.</p> 
3	<p>Kembali ke VS code, project flutter hello_world, buka terminal pada menu Terminal &gt; New Terminal. Lalu ketik perintah berikut untuk inisialisasi git pada project Anda.</p> 
4	<p>Pilih menu Source Control di bagian kiri, lalu lakukan stages (+) pada file .gitignore untuk mengunggah file pertama ke repository GitHub.</p>



	
5	<p>Beri pesan commit "tambah gitignore" lalu klik Commit (✓)</p> 
6	<p>Lakukan push dengan klik bagian menu titik tiga &gt; Push</p> 
7	<p>Di pojok kanan bawah akan tampil seperti gambar berikut. Klik "Add Remote"</p> 

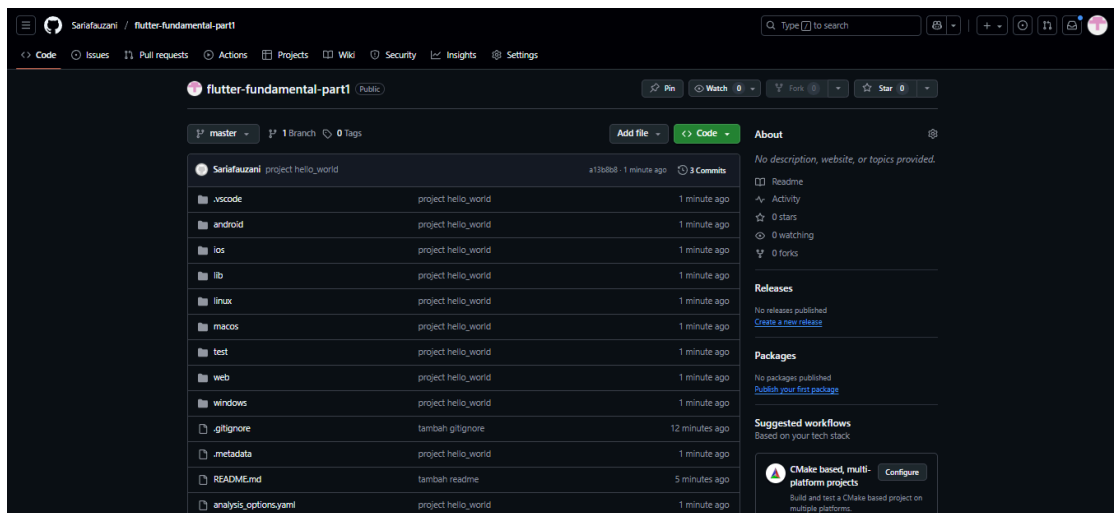
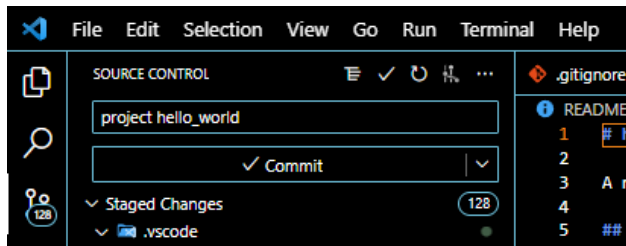


8	<p>Salin tautan repository Anda dari browser ke bagian ini, lalu klik Add remote</p> 
	<p>Setelah berhasil, tulis remote name dengan "origin"</p> 
9	<p>Lakukan hal yang sama pada file README.md mulai dari Langkah 4. Setelah berhasil melakukan push, masukkan username GitHub Anda dan password berupa token yang telah dibuat (pengganti password konvensional ketika Anda login di browser GitHub). Reload halaman repository GitHub Anda, maka akan tampil hasil push kedua file tersebut seperti gambar berikut.</p>
	



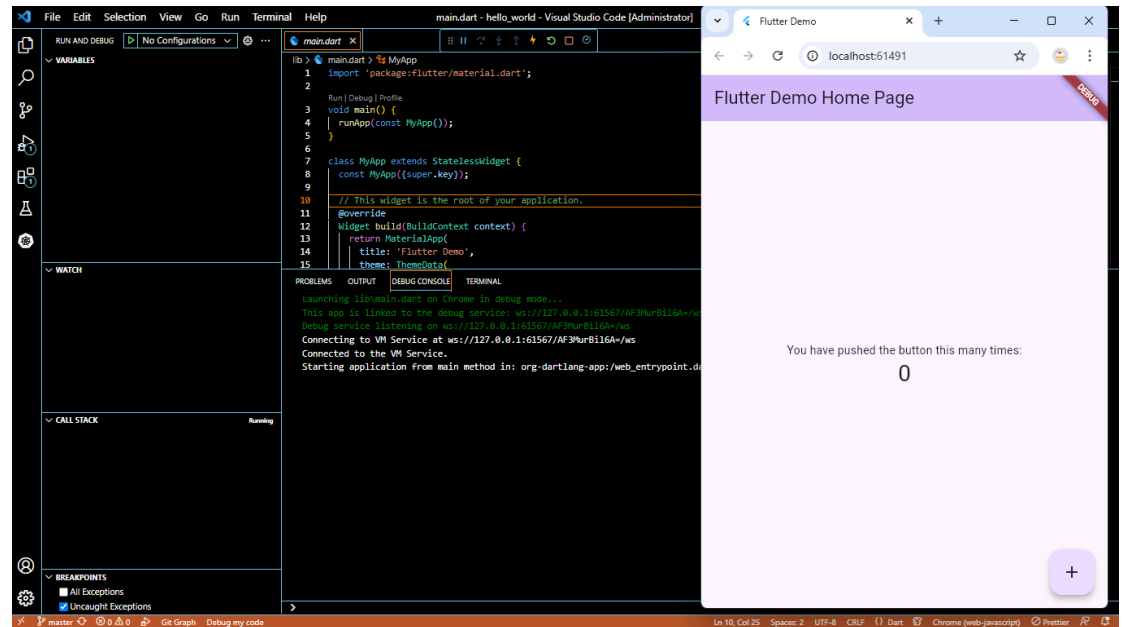
10

Lakukan push juga untuk semua file lainnya dengan pilih Stage All Changes. Beri pesan commit "project hello\_world". Maka akan tampil di repository GitHub Anda seperti berikut.



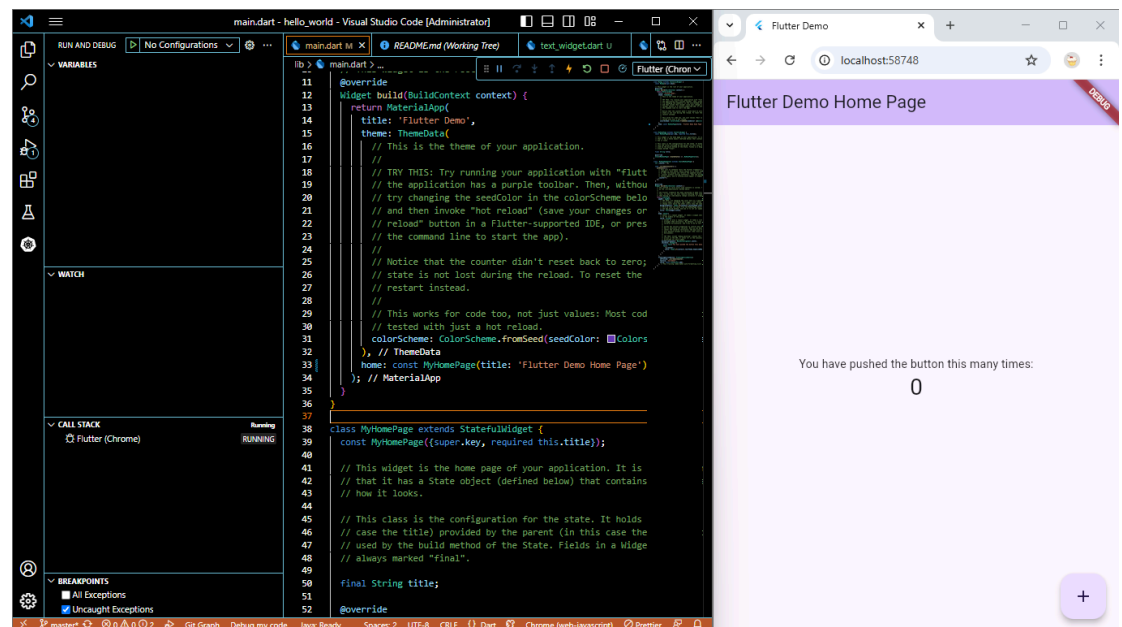
11

Kembali ke VS Code, ubah platform di pojok kanan bawah ke emulator atau device atau bisa juga menggunakan browser Chrome. Lalu coba running project hello\_world dengan tekan F5 atau Run > Start Debugging. Tunggu proses kompilasi hingga selesai, maka aplikasi flutter pertama Anda akan tampil seperti berikut.



12

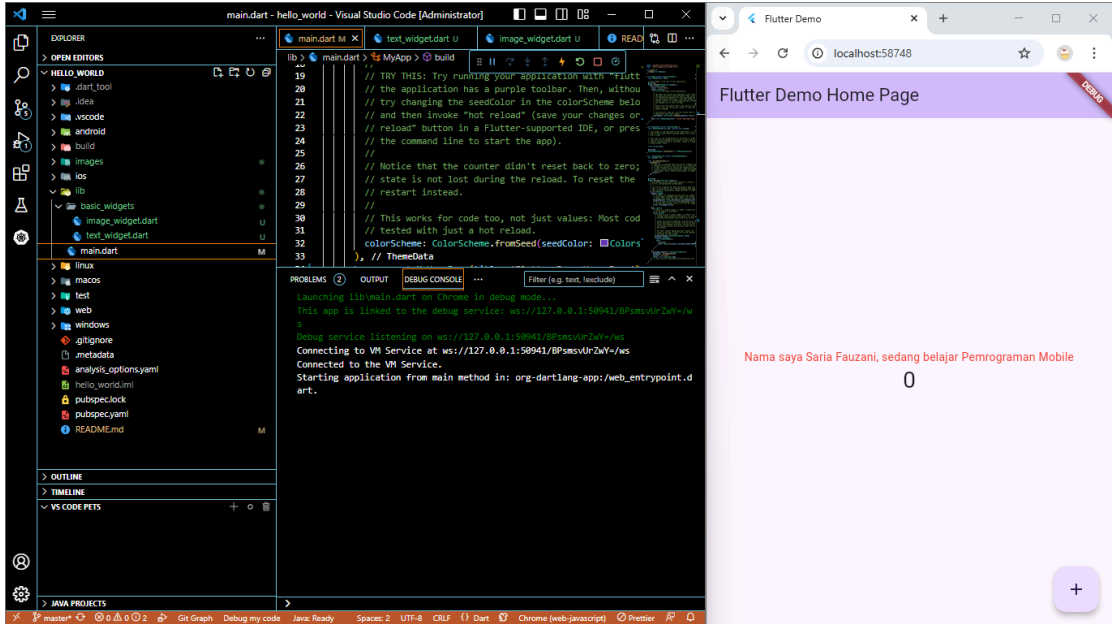
Silakan screenshot seperti pada Langkah 11, namun teks yang ditampilkan dalam aplikasi berupa nama lengkap Anda. Simpan file screenshot dengan nama 01.png pada folder images (buat folder baru jika belum ada) di project hello\_world Anda. Lalu ubah isi README.md seperti berikut, sehingga tampil hasil screenshot pada file README.md. Kemudian push ke repository Anda.



## Praktikum Ke-4 : Menerapkan Widget Dasar





Langkah	Jawaban/Deskripsi
	<b>Langkah 1: Text Widget</b>
	<p>Buat folder baru <code>basic_widgets</code> di dalam folder <code>lib</code>. Kemudian buat file baru di dalam <code>basic_widgets</code> dengan nama <code>text_widget.dart</code>. Ketik atau salin kode program berikut ke project <code>hello_world</code> Anda pada file <code>text_widget.dart</code>.</p> <pre>lib &gt; basic_widgets &gt; text_widget.dart &gt; ... 1  import 'package:flutter/material.dart'; 2 3  class MyTextWidget extends StatelessWidget { 4    const MyTextWidget({Key? key}) : super(key: key); 5 6    @override 7    Widget build(BuildContext context) { 8      return const Text( 9        "Nama saya Saria Fauzani, sedang belajar Pemrograman Mobile", 10       style: TextStyle(color: Colors.red, fontSize: 14), 11       textAlign: TextAlign.center); // Text 12    } 13  } 14</pre>
	<p>Lakukan import file <code>text_widget.dart</code> ke <code>main.dart</code>, lalu ganti bagian text widget dengan kode di atas. Maka hasilnya seperti gambar berikut. Screenshot hasil milik Anda, lalu dibuat laporan pada file <a href="#">README.md</a>.</p> 
	<b>Langkah 2: Image Widget</b>
	<p>Buat sebuah file <code>image_widget.dart</code> di dalam folder <code>basic_widgets</code> dengan isi kode berikut.</p>



	<p>Lakukan penyesuaian asset pada file pubspec.yaml dan tambahkan file logo Anda di folder assets project hello_world.</p> <pre>53 flutter: 54   assets: 55     - assets/logo-polinema.png</pre>
	<p>Jangan lupa sesuaikan kode dan import di file main.dart kemudian akan tampil gambar seperti berikut.</p>

## Praktikum Ke-5 :Menerapkan Widget Material Design dan iOS Cupertino

Langkah	Jawaban/Deskripsi
	<b>Langkah 1: Cupertino Button dan Loading Bar</b>



Buat file di `basic_widgets > loading_cupertino.dart`. Import stateless widget dari material dan cupertino. Lalu isi kode di dalam method Widget build adalah sebagai berikut.

```
lib > basic_widgets > loading_cupertino.dart > ...
1  import 'package:flutter/material.dart';
2  import 'package:flutter/cupertino.dart';
3
4  class LoadingCupertino extends StatelessWidget {
5    const LoadingCupertino({Key? key}) : super(key: key);
6
7    @override
8    Widget build(BuildContext context) {
9      return MaterialApp(
10        home: Container(
11          margin: const EdgeInsets.only(top: 30),
12          color: Colors.white,
13          child: Column(
14            children: <Widget>[
15              CupertinoButton(
16                child: const Text("Contoh button"),
17                onPressed: () {},
18              ), // CupertinoButton
19              const CupertinoActivityIndicator(),
20            ], // <Widget>[]
21          ), // Column
22        ), // Container
23      ); // MaterialApp
24    }
25  }
```

## Langkah 2: Floating Action Button (FAB)

Button widget terdapat beberapa macam pada flutter yaitu `AppBar`, `DropDownButton`, `TextButton`, `FloatingActionButton`, `IconButton`, `OutlineButton`, `PopupMenuButton`, dan `ElevatedButton`.

Buat file di `basic_widgets > fab_widget.dart`. Import stateless widget dari material. Lalu isi kode di dalam method Widget build adalah sebagai berikut.

```
lib > basic_widgets > fab_widget.dart > FabWidget
1  import 'package:flutter/material.dart';
2
3  class FabWidget extends StatelessWidget {
4    const FabWidget({Key? key}) : super(key: key);
5
6    @override
7    Widget build(BuildContext context) {
8      return MaterialApp(
9        home: Scaffold(
10          floatingActionButton: FloatingActionButton(
11            onPressed: () {
12              // Add your onPressed code here!
13            },
14            child: const Icon(Icons.thumb_up),
15            backgroundColor: Colors.pink,
16          ), // FloatingActionButton
17        ), // Scaffold
18      ); // MaterialApp
19    }
20  }
```

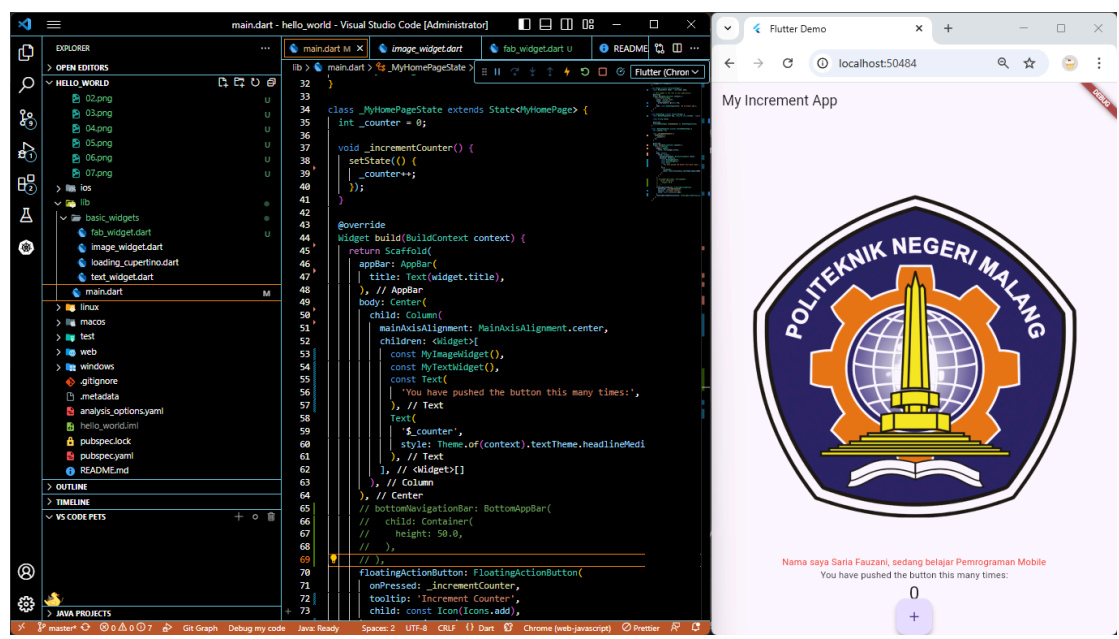


### Langkah 3: Scaffold Widget

Ubah isi kode main.dart seperti berikut.

```
lib > main.dart > MyHomePage
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({Key? key}) : super(key: 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        primarySwatch: Colors.red,
17      ), // ThemeData
18      home: const MyHomePage(title: 'My Increment App'),
19    ); // MaterialApp
20  }
21 }
22
23 class MyHomePage extends StatefulWidget {
24   const MyHomePage({Key? key, required this.title}) : super(key: key);
25
26   final String title;
27
28   @override
29   State<MyHomePage> createState() => _MyHomePageState();
30 }
31
32 class _MyHomePageState extends State<MyHomePage> {
33   int _counter = 0;
34
35   void _incrementCounter() {
36     setState(() {
37       _counter++;
38     });
39   }
40 }
```

### Output



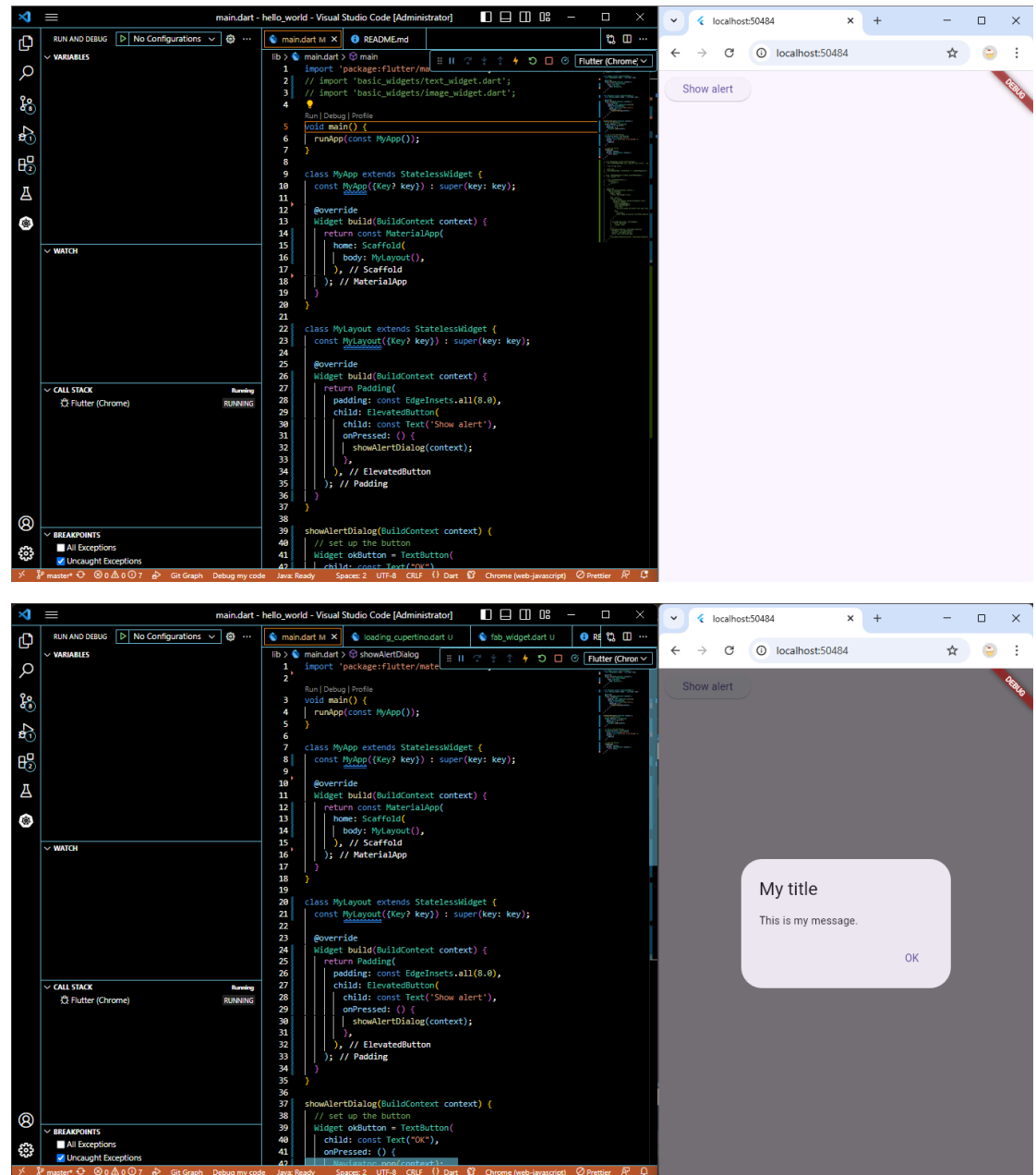


#### Langkah 4: Dialog Widget

Ubah isi kode main.dart seperti berikut.

```
lib > main.dart > showAlertDialog
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({Key? key}) : super(key: key);
9
10  @override
11  Widget build(BuildContext context) {
12    return const MaterialApp(home: Scaffold(body: MyLayout()));
13  }
14 }
15
16 class MyLayout extends StatelessWidget {
17   const MyLayout({Key? key}) : super(key: key);
18
19   @override
20   Widget build(BuildContext context) {
21     return Padding(
22       padding: const EdgeInsets.all(8.0),
23       child: ElevatedButton(
24         child: const Text('Show alert'),
25         onPressed: () {
26           showAlertDialog(context);
27         },
28       ), // ElevatedButton
29     ); // Padding
30   }
31 }
32
33 showAlertDialog(BuildContext context) {
34   // set up the button
35   Widget okButton = TextButton(
36     child: const Text("OK"),
37     onPressed: () {
38       Navigator.pop(context);
39     },
40   ); // TextButton
41 }
```

Output



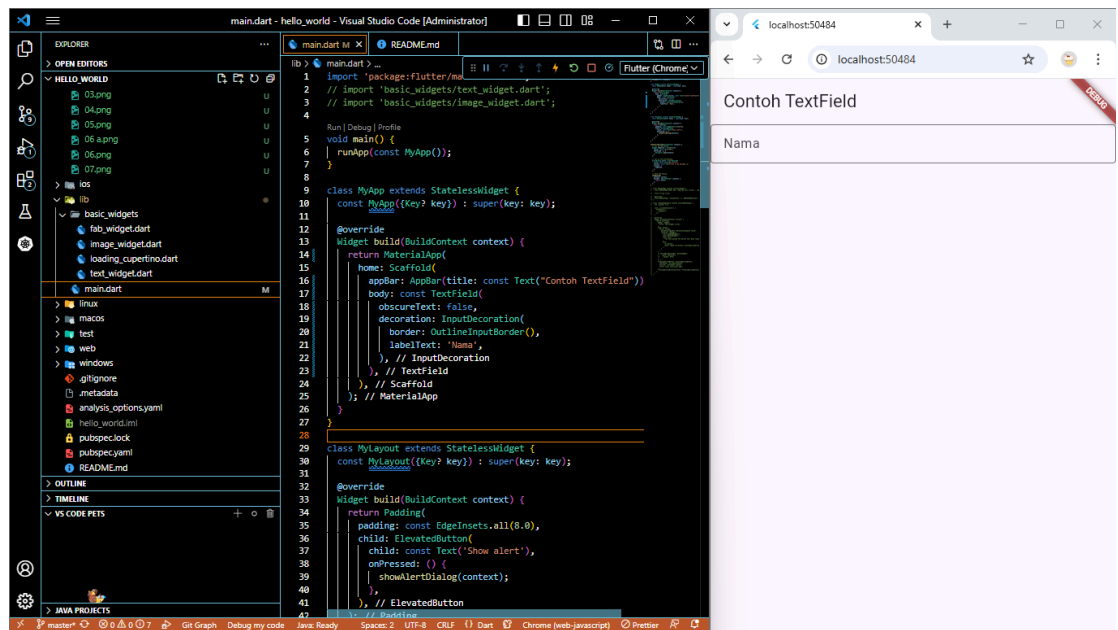
### Langkah 5: Input dan Selection Widget

Contoh penggunaan TextField widget adalah sebagai berikut:

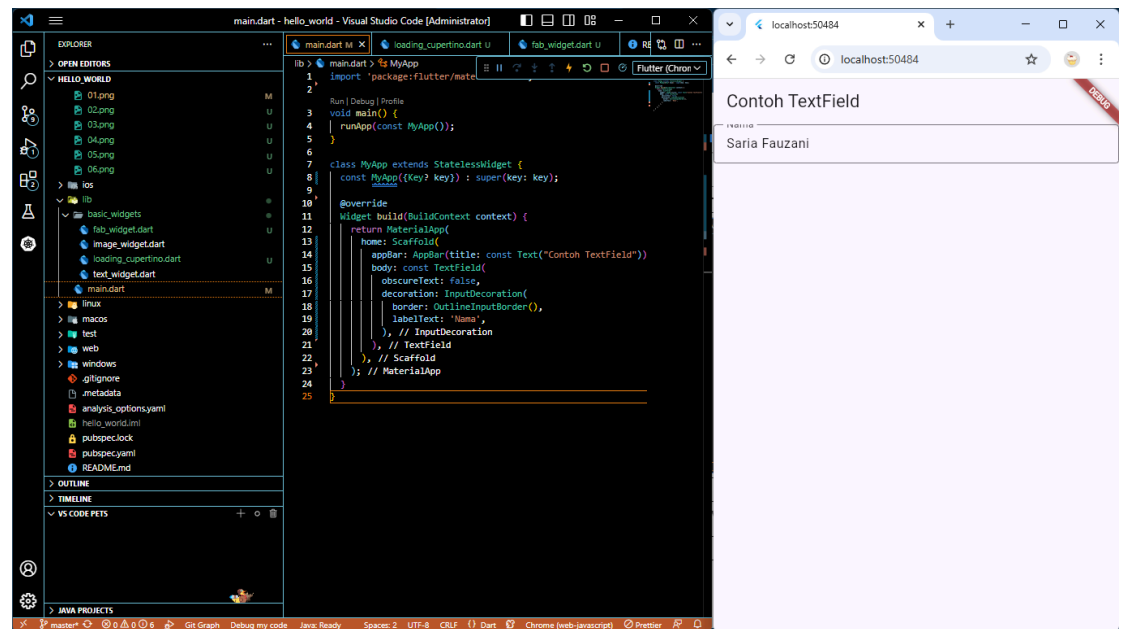


```
9 class MyApp extends StatelessWidget {  
0   const MyApp({Key? key}) : super(key: key);  
1  
2   @override  
3   Widget build(BuildContext context) {  
4     return MaterialApp(  
5       home: Scaffold(  
6         appBar: AppBar(title: const Text("Contoh TextField")),  
7         body: const TextField(  
8           obscureText: false,  
9           decoration: InputDecoration(  
0             border: OutlineInputBorder(),  
1             labelText: 'Nama',  
2           ), // InputDecoration  
3         ), // TextField  
4       ), // Scaffold  
5     ); // MaterialApp  
6   }  
7 }
```

### Output

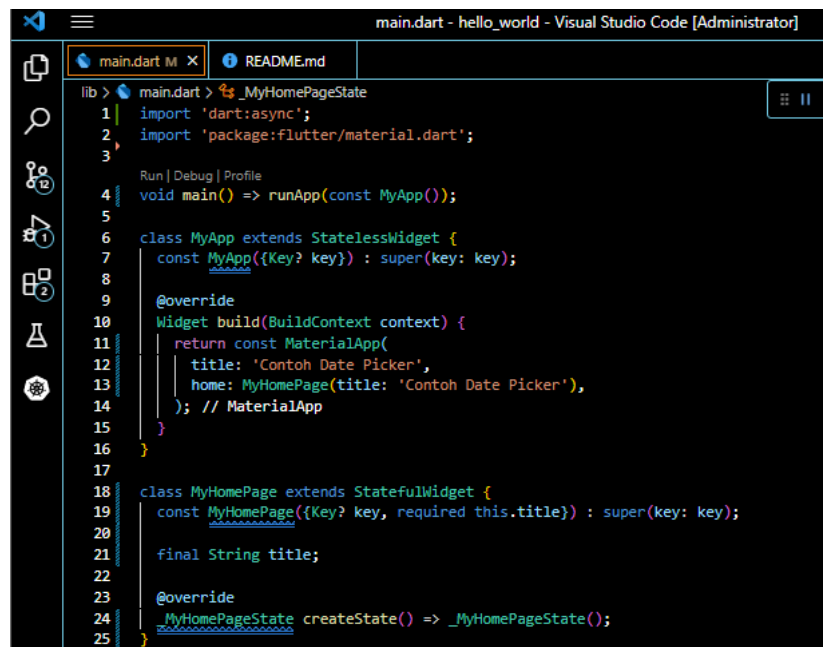






## Langkah 6: Date and Time Pickers

Date and Time Pickers termasuk pada kategori input dan selection widget, berikut adalah contoh penggunaan Date and Time Pickers.







## Output

