

LAPORAN PRAKTIKUM
PEMMROGRAMAN MOBILE JOBSHEET 5

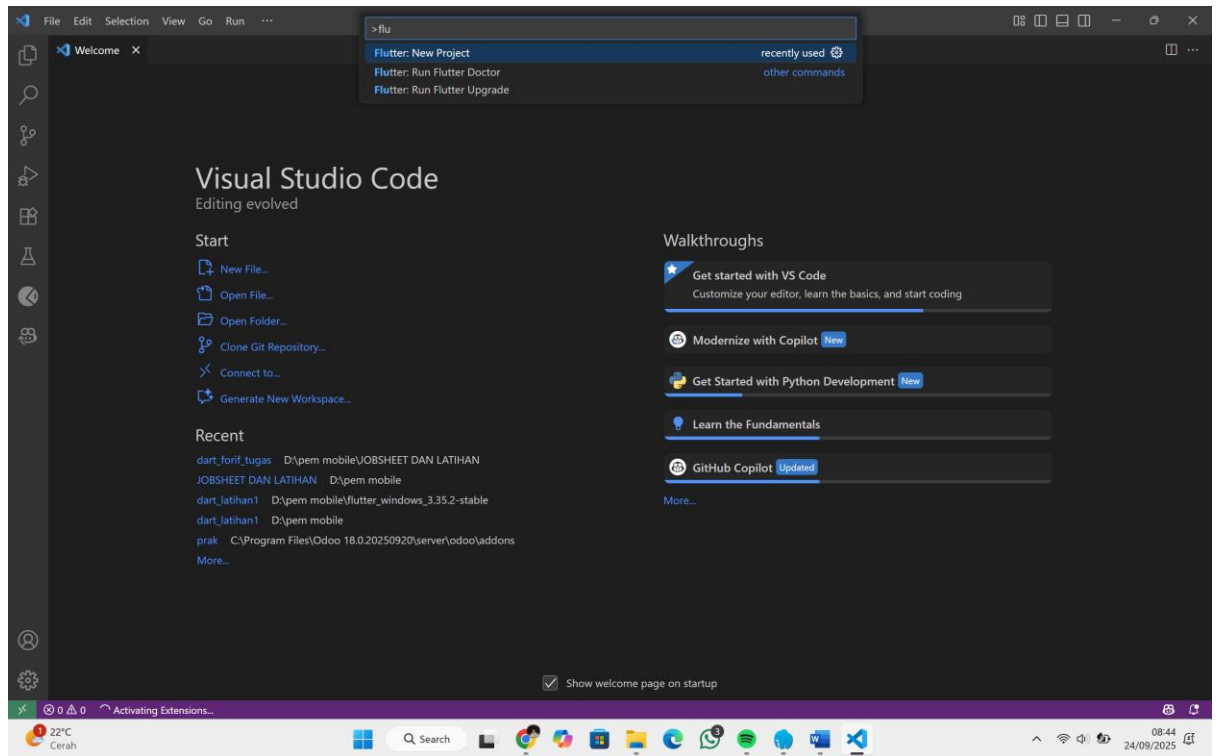


Disusun Oleh:
Mariska Putri
No. Absen: 23
Kelas: SIB 3C
NIM: 2341760051

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

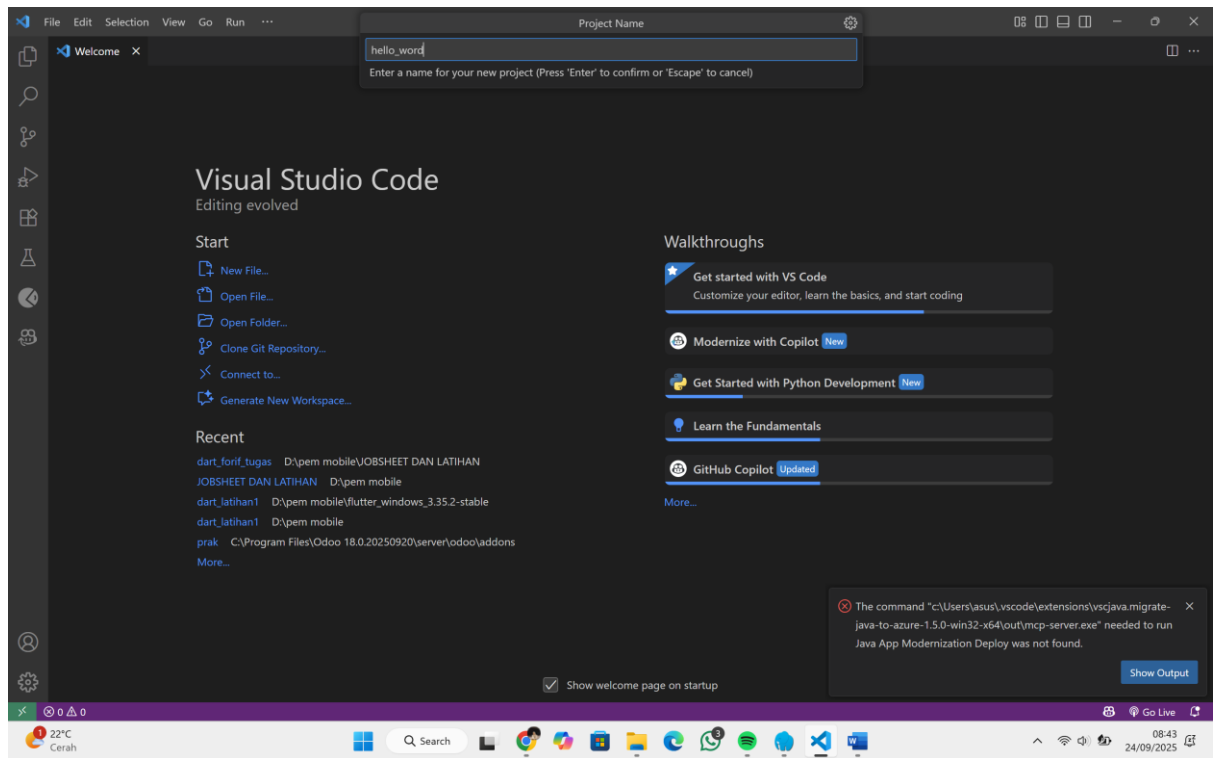
NO	Langkah-Langkah
	Praktikum 1

1. Langkah 1:
Buka VS Code, lalu tekan tombol Ctrl + Shift + P maka akan tampil *Command Palette*, lalu ketik Flutter. Pilih New Application Project.

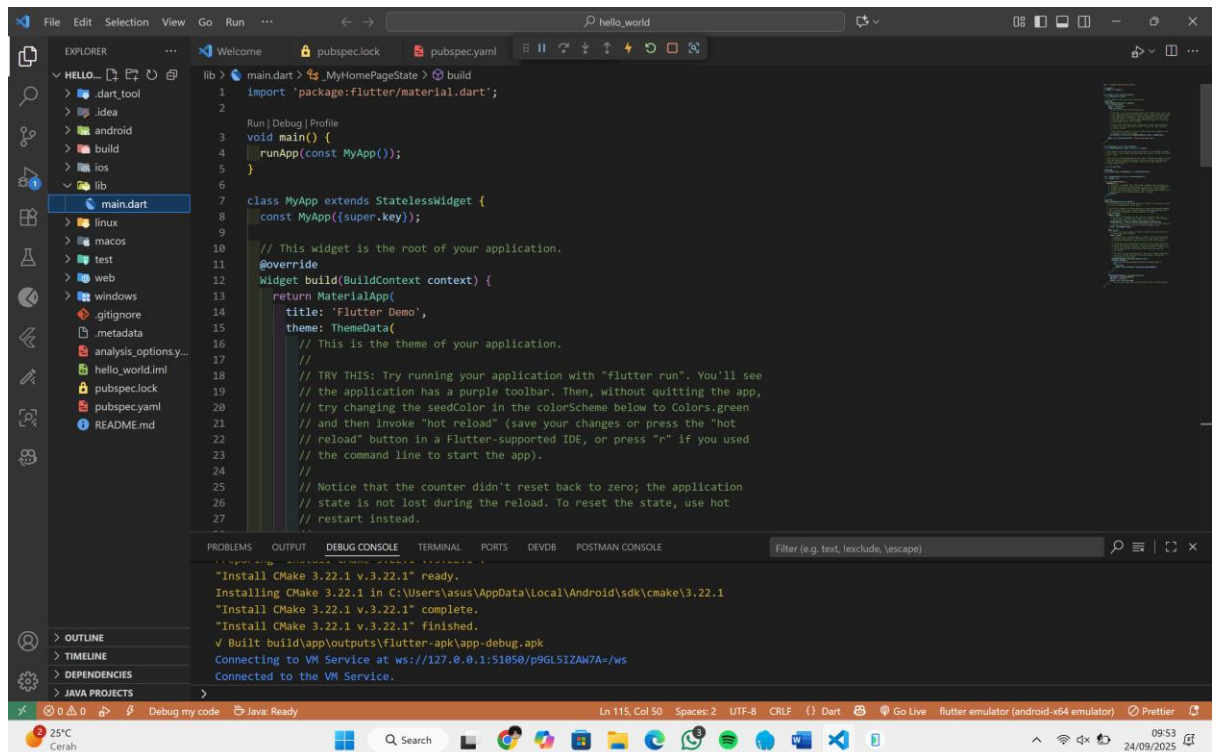


2. Langkah 2:
Kemudian buat folder sesuai style laporan praktikum yang Anda pilih. Disarankan pada folder dokumen atau desktop atau alamat folder lain yang tidak terlalu dalam atau panjang. Lalu pilih Select a folder to create the project in.

3. Langkah 3:
Buat nama project flutter hello_world seperti berikut, lalu tekan Enter. Tunggu hingga proses pembuatan project baru selesai.

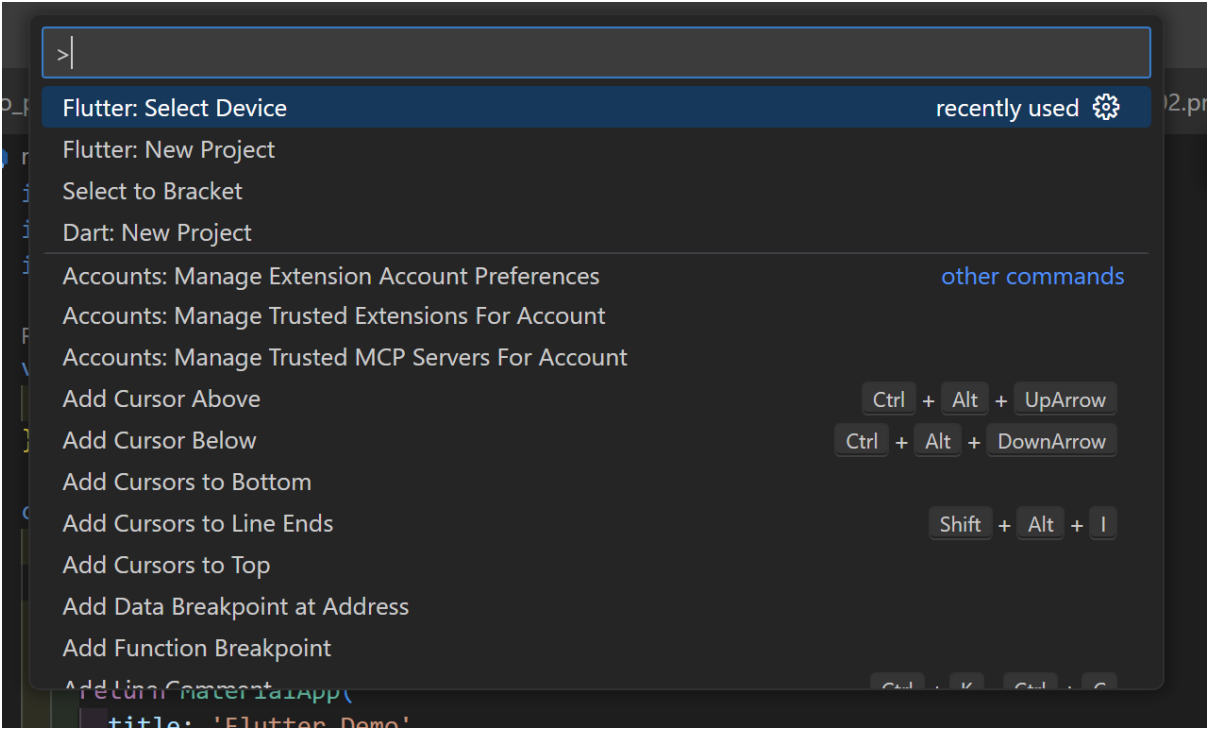
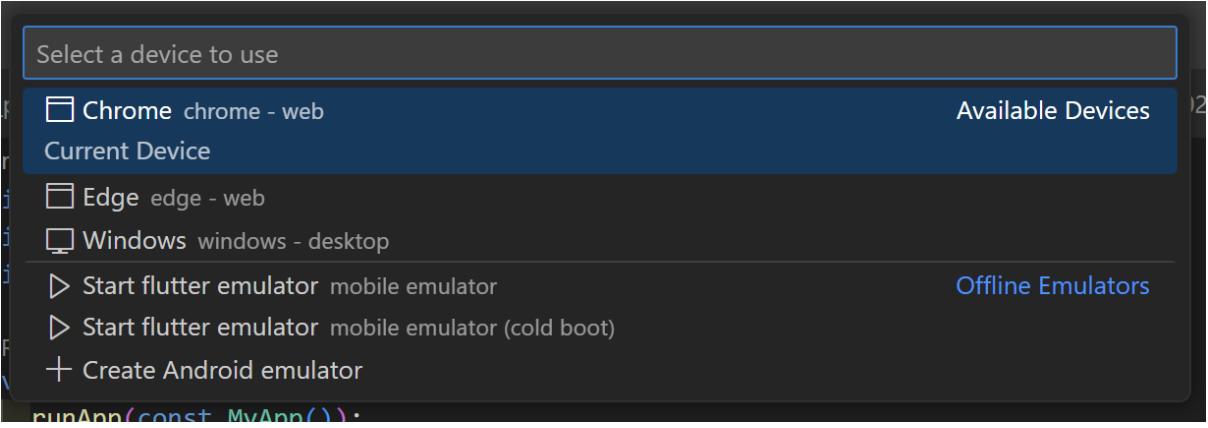


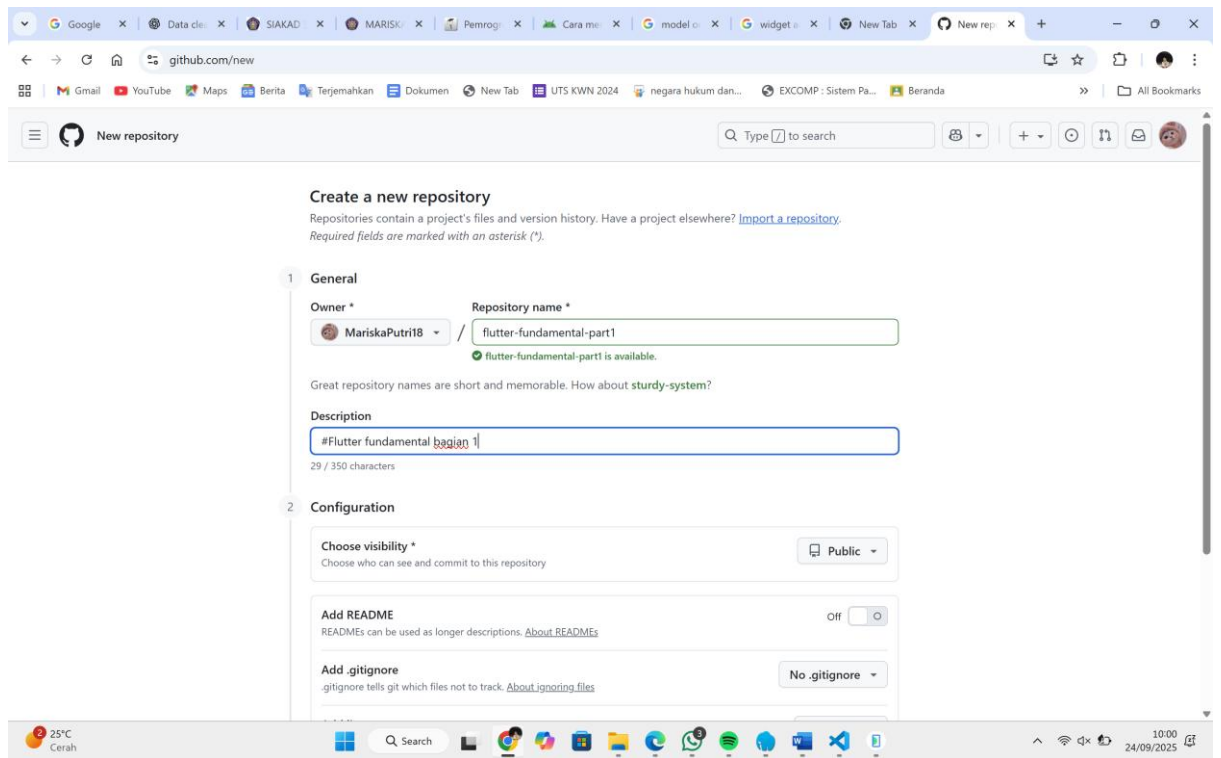
4. Langkah 4:
Jika sudah selesai proses pembuatan project baru, pastikan tampilan seperti berikut. Pesan akan tampil berupa "Your Flutter Project is ready!"



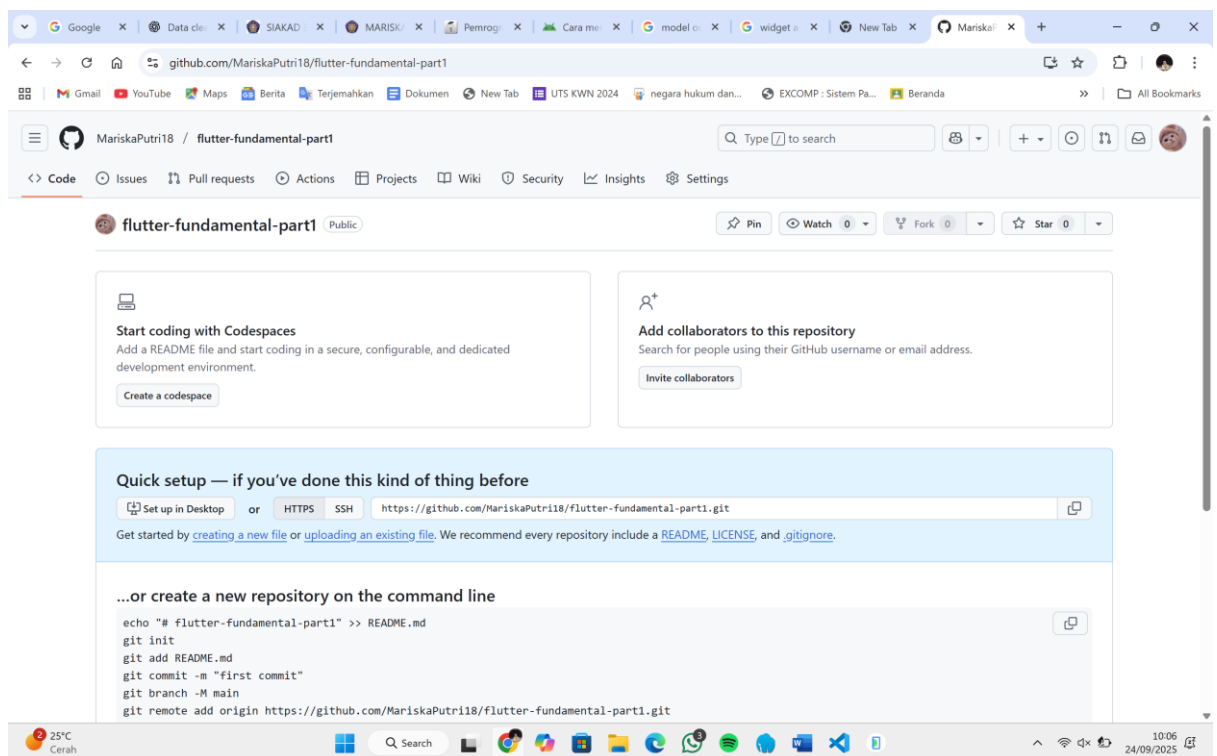
Praktikum 2

1. Cari: >flutter:select device

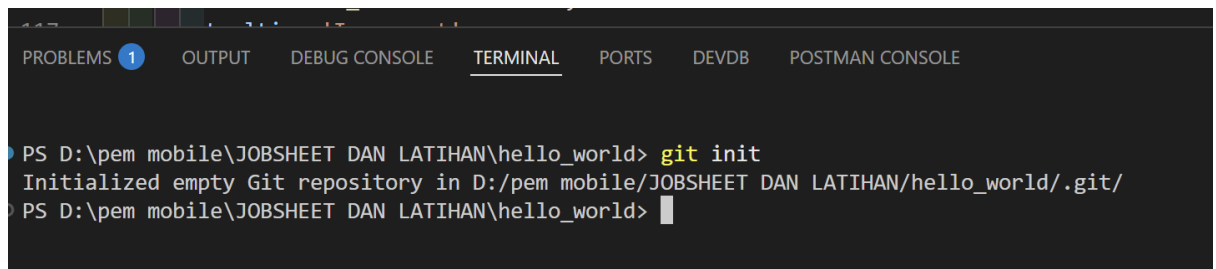
	
2.	<p>Pilih emulator yang akan dijalankan</p> 
Praktikum 3	
1	<p>Langkah 1: Login ke akun Github , lalu buat repository baru dengan nama "flutter-fundamental-part1"</p>



2. Langkah 2:
Lalu klik tombol "Create repository" lalu akan tampil seperti gambar berikut.

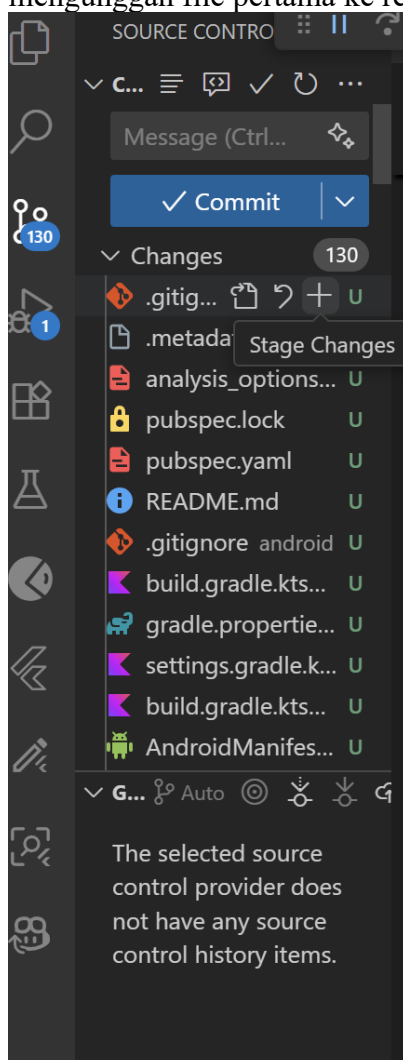


3. Langkah 3:
Kembali ke VS code, project flutter hello_world, buka terminal pada menu Terminal > New Terminal. Lalu ketik perintah berikut untuk inisialisasi git pada project Anda.

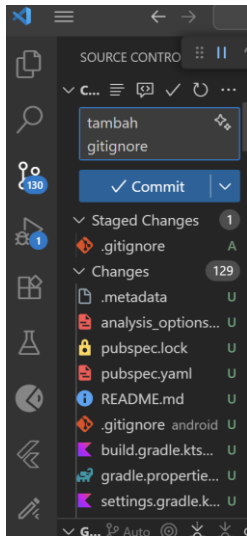


```
PS D:\pem mobile\JOBSHEET DAN LATIHAN\hello_world> git init
Initialized empty Git repository in D:/pem mobile/JOBSHEET DAN LATIHAN/hello_world/.git/
PS D:\pem mobile\JOBSHEET DAN LATIHAN\hello_world> 
```

4. Langkah 4:
Pilih menu Source Control di bagian kiri, lalu lakukan stages (+) pada file .gitignore untuk mengunggah file pertama ke repository GitHub.



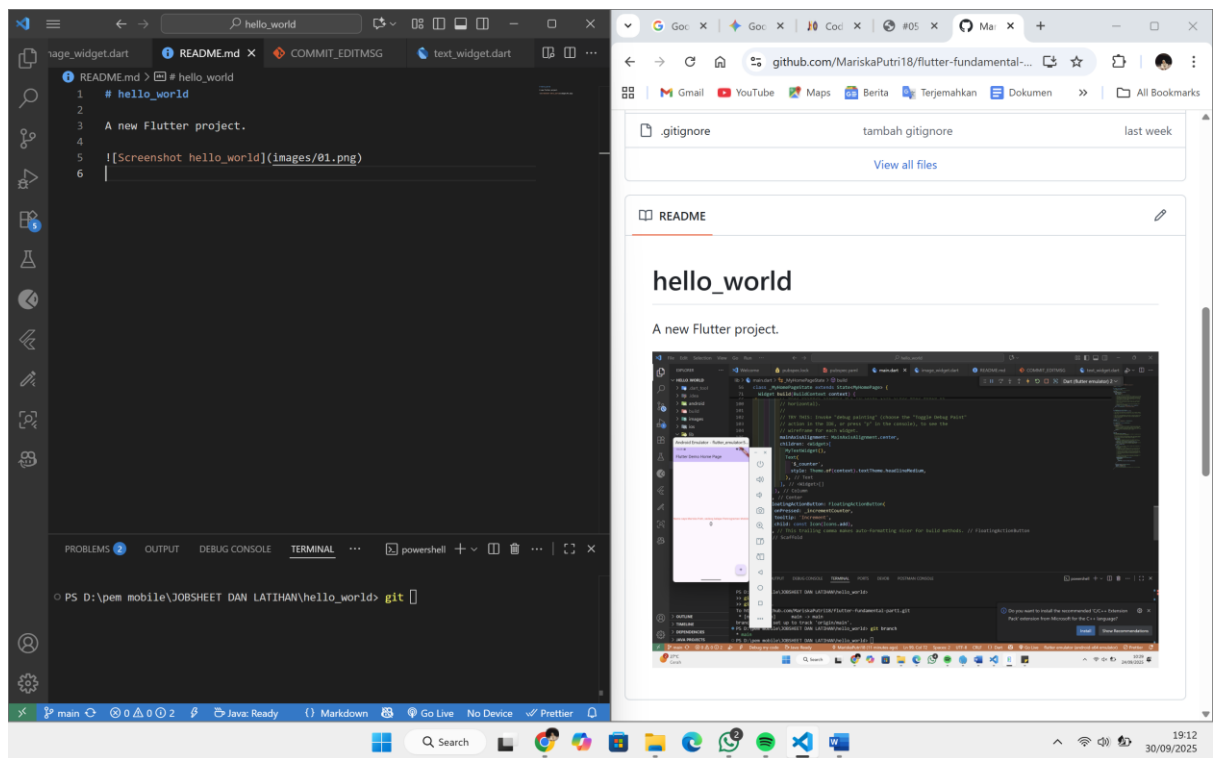
5. Langkah 5:
Beri pesan commit "tambah gitignore" lalu klik Commit (✓)



6. Langkah 6:
Lakukan commit dan push semua perubahan ke repository

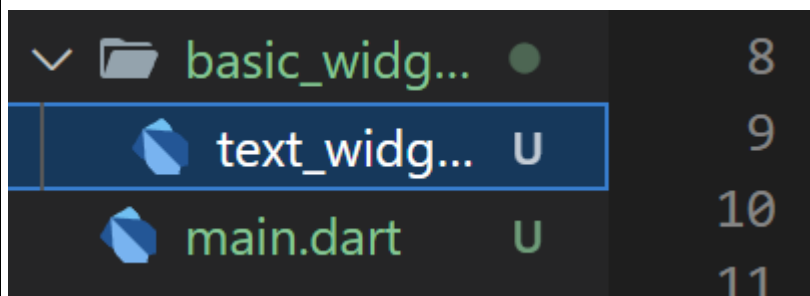


7. Langkah 7:
Lakukan screenshot pada hasil run dari project Simpan file screenshot dengan nama 01.png pada folder images (buat folder baru jika belum ada) di project hello_world Anda. Lalu ubah isi README.md seperti berikut, sehingga tampil hasil screenshot pada file README.md. Kemudian push ke repository.



Praktikum 4: Menerapkan Widget Dasar

- 1, Langkah 1: Text Widget
Buat folder baru `basic_widgets` di dalam folder `lib`. Kemudian buat file baru di dalam `basic_widgets` dengan nama `text_widget.dart`. Ketik atau salin kode program berikut ke project `hello_world` Anda pada file `text_widget.dart`.



```
lib > basic_widget > text_widget.dart > MyTextWidget
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 Mariska Putri, sedang belajar Pemrograman Mobile",
10       style: TextStyle(color: Colors.red, fontSize: 14),
11       textAlign: TextAlign.center); // Text
12    }
13  }
```

2. Lakukan import file `text_widget.dart` ke `main.dart`, lalu ganti bagian text widget dengan kode di atas. Maka hasilnya seperti gambar berikut. Screenshot hasil milik Anda, lalu dibuat laporan pada file `README.md`.

```
mainAxisAlignment: MainAxisAlignment.center,  
children: <Widget>[  
  MyTextWidget(),  
  Text(  
    '$_counter',  
    style: Theme.of(context).textTheme.headlineMedium,  
  ), // Text  
, // <Widget>[]  
, // Column  
, // Center  
floatingActionButton: FloatingActionButton(  
  onPressed: _incrementCounter,  
  tooltip: 'Increment',  
  child: const Icon(Icons.add),  
), // This trailing comma makes auto-formatting nicer for build methods. // FloatingActionButton  
); // Scaffold
```

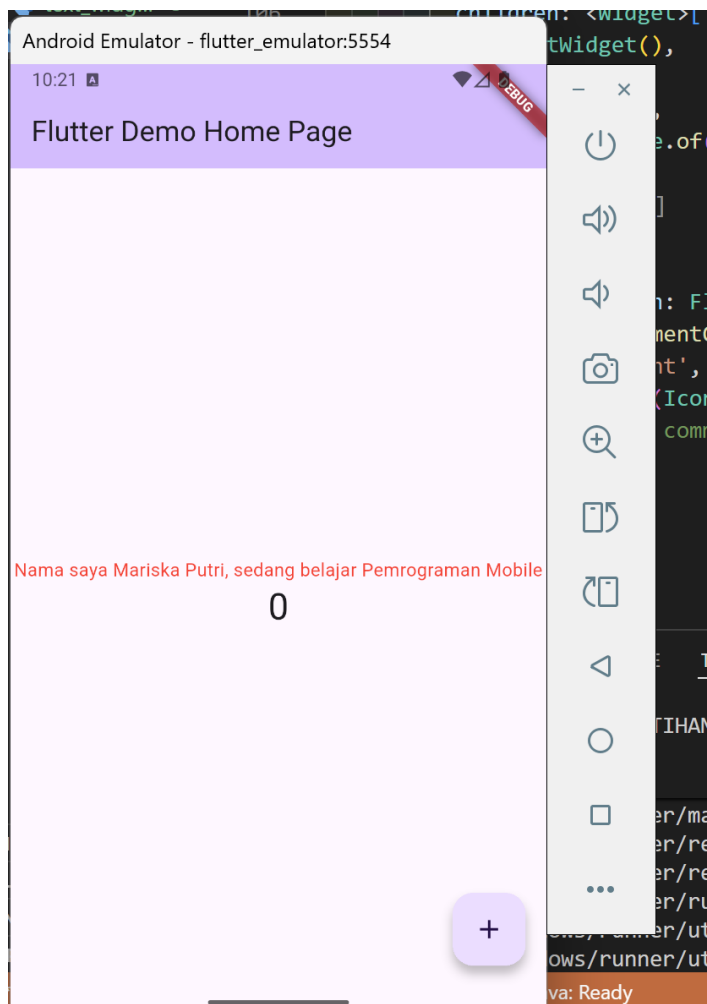
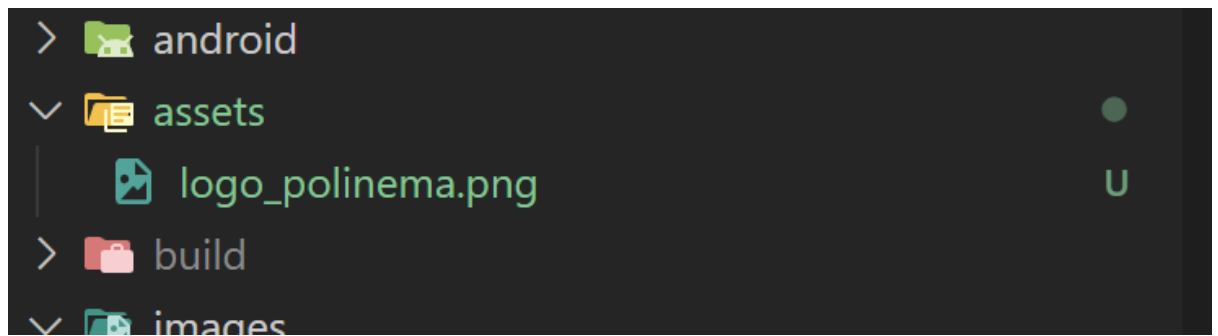


Image Widget

3. Buat sebuah file `image_widget.dart` di dalam folder `basic_widgets` dengan isi kode berikut.

```
lib > basic_widget > image_widget.dart > MyImageWidget
1  import 'package:flutter/material.dart';
2
3  class MyImageWidget extends StatelessWidget {
4    const MyImageWidget({Key? key}) : super(key: key);
5
6    @override
7    Widget build(BuildContext context) {
8      return const Image(
9        image: AssetImage("logo_polinema.jpg")
10     ); // Image
11   }
12 }
```

4. Buat folder `assets` dan masukkan file `logo_polinema.jpg` pada folder `assets`



5. Lakukan penyesuaian asset pada file `pubspec.yaml` dan tambahkan file logo Anda di folder `assets` project `hello_world`.

```
51
52 # The following section is specific to Flutter packages.
53 flutter:
54   assets:
55     - logo_polinema.png
56
```

6. Sesuaikan kode dan import di file main.dart

```
// Wireframe for each widget.
mainAxisAlignment: MainAxisAlignment.center,
children: <Widget>[
  const MyImageWidget(),
  Text(
    '$_counter',
    style: Theme.of(context).textTheme.headlineMedium,
  ), // Text
], // <Widget>[]
), // Column
), // Center
floatingActionButton: FloatingActionButton(
  onPressed: _incrementCounter,
  tooltip: 'Increment',
  child: const Icon(Icons.add),
), // This trailing comma makes auto-formatting nicer for build methods. // FloatingActionButton
); // Scaffold
}
```

7. Hasil Run



Praktikum 5: Menerapkan Widget Material Design dan iOS Cupertino

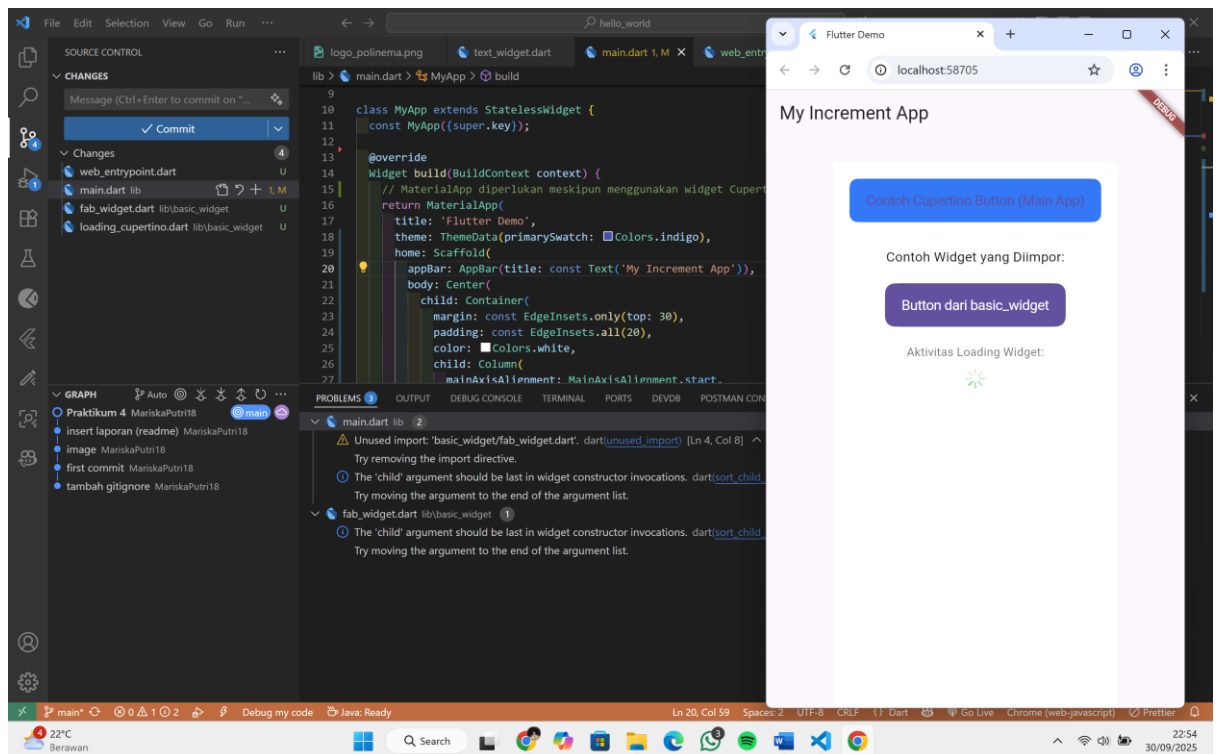
1. Langkah 1: Cupertino Button dan Loading Bar
Buat file di `basic_widgets > loading_cupertino.dart`. Import stateless widget dari material dan cupertino. Lalu isi kode di dalam method `Widget build` adalah sebagai berikut.

```

logo_polinema.png  text_widget.dart  main.dart  02.png  hello_world.iml  01.png
lib > basic_widget > loading_cupertino.dart > LoadingCupertino
1  import 'package:flutter/material.dart';
2  import 'package:flutter/cupertino.dart';
3
4  class LoadingCupertino extends StatelessWidget {
5    const LoadingCupertino({super.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

```

2. Main.dart dan hasil



The screenshot shows an IDE (VS Code) with the following components:

- SOURCE CONTROL:** Shows a commit message "Message (Ctrl+Enter to commit on '...'" and a "Commit" button.
- CHANGES:** Lists changes to files: web_entrypoint.dart, main.dart, fab_widget.dart, and loading_cupertino.dart.
- main.dart:** Contains the following code:


```

9
10 class MyApp extends StatelessWidget {
11   const MyApp({super.key});
12
13   @override
14   Widget build(BuildContext context) {
15     // MaterialApp diperlukan meskipun menggunakan widget Cupertino
16     return MaterialApp(
17       title: 'Flutter Demo',
18       theme: ThemeData(primarySwatch: Colors.indigo),
19       home: Scaffold(
20         appBar: AppBar(title: const Text('My Increment App')),
21         body: Center(
22           child: Container(
23             margin: const EdgeInsets.only(top: 30),
24             padding: const EdgeInsets.all(20),
25             color: Colors.white,
26             child: Column(
27               mainAxisAlignment: MainAxisAlignment.start,

```
- PROBLEMS:** Lists two errors:
 - Unused import: 'basic_widget/fab_widget.dart'. dart(unused_import) [Ln 4, Col 8]
 - The 'child' argument should be last in widget constructor invocations. dart(sort_child_argument) [Ln 22, Col 12]
- Preview:** A browser window titled "Flutter Demo" showing the app running at localhost:58705. The app displays:
 - A blue button labeled "Contoh Cupertino Button (Main App)".
 - A text label "Contoh Widget yang Diimpor:".
 - A purple button labeled "Button dari basic_widget".
 - A text label "Aktivitas Loading Widget:".
 - A green loading indicator (CupertinoActivityIndicator).

3. Langkah 2: Floating Action Button (FAB)

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_widget > fab_widget.dart > FabWidget
1  import 'package:flutter/material.dart';
2
3  class FabWidget extends StatelessWidget {
4    const FabWidget({super.key});
5
6    @override
7    Widget build(BuildContext context) {
8      return MaterialApp(
9        home: Scaffold(
10          floatingActionButton: FloatingActionButton
11            onPressed: () {
12              // aksi ketika tombol ditekan
13              debugPrint("FAB ditekan!");
14            },
15          child: const Icon(Icons.thumb_up),
16          backgroundColor: Colors.pink,
17        ), // FloatingActionButton
18      ), // Scaffold
19    ); // MaterialApp
20  }
21 }
```

3. Langkah 3: Scaffold Widget
Scaffold widget digunakan untuk mengatur tata letak sesuai dengan material design. Ubah isi kode `main.dart` seperti berikut.

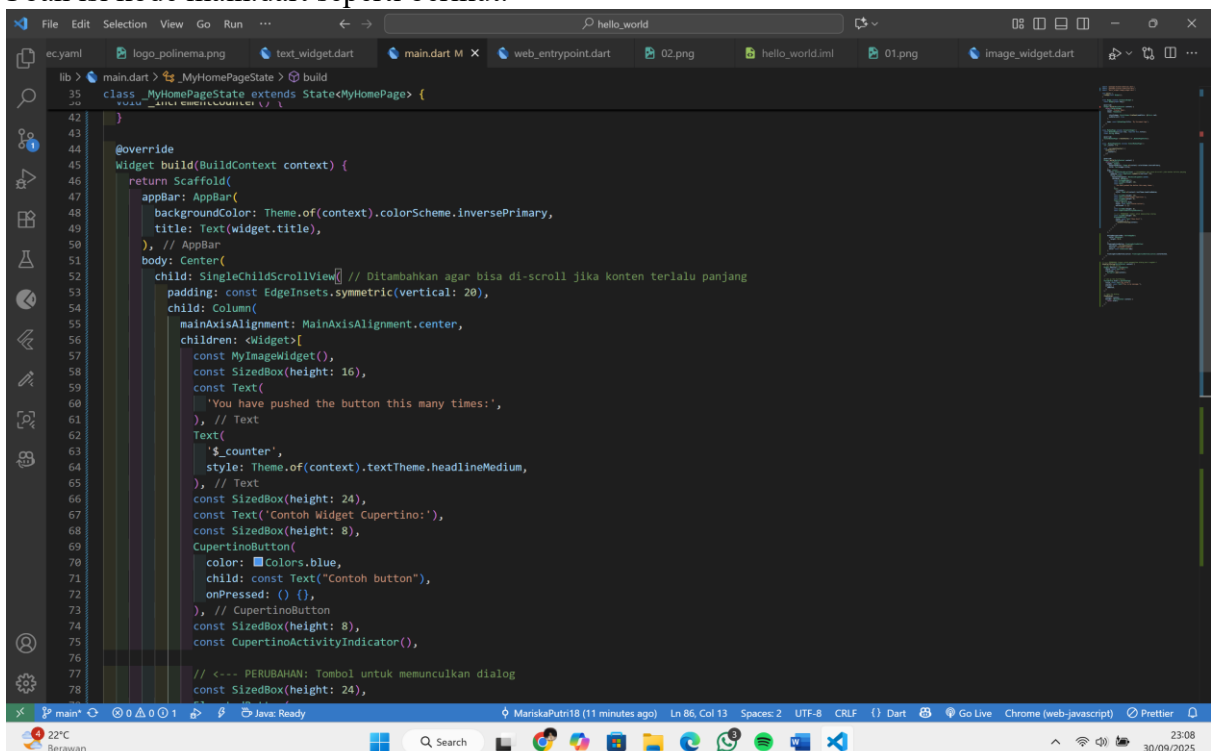
```

lib > main.dart > _MyHomePageState > build
1  import 'package:flutter/material.dart';
2
3  Run | Debug | Profile
4  void main() {
5    runApp(const MyApp());
6  }
7
8  class MyApp extends StatelessWidget {
9    const MyApp({Key? key}) : super(key: key);
10
11    // This widget is the root of your application.
12    @override
13    Widget build(BuildContext context) {
14      return MaterialApp(
15        title: 'Flutter Demo',
16        theme: ThemeData(
17          primarySwatch: Colors.red,
18        ), // ThemeData
19        home: const MyHomePage(title: 'My Increment App'),
20      ); // MaterialApp
21    }
22  }
23
24  class MyHomePage extends StatefulWidget {
25    const MyHomePage({Key? key, required this.title}) : super(key: key);
26
27    final String title;
28
29    @override
30    State<MyHomePage> createState() => _MyHomePageState();
31  }
32
33  class _MyHomePageState extends State<MyHomePage> {
34    int _counter = 0;
35
36    void _incrementCounter() {
37      setState(() {
38        _counter++;
39      });
40    }
41  }

```

4. Langkah 4: Dialog Widget

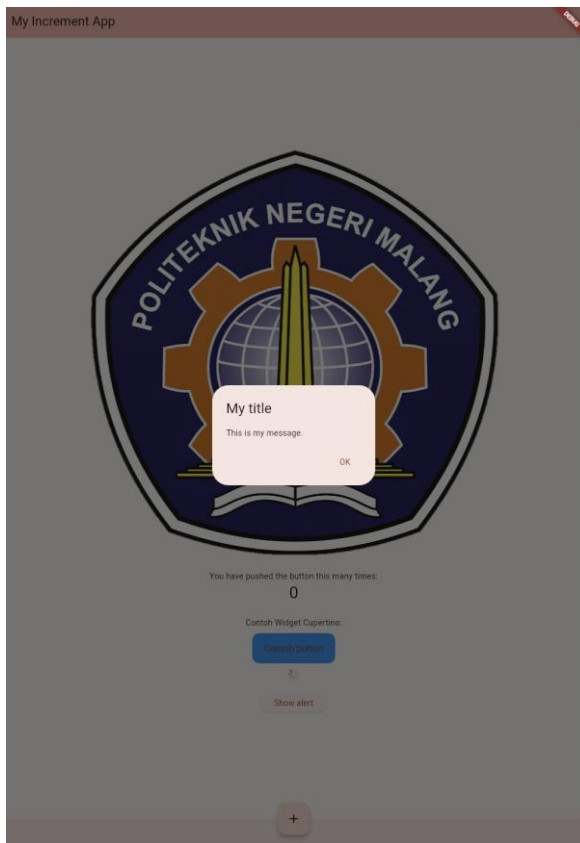
Dialog widget pada flutter memiliki dua jenis dialog yaitu AlertDialog dan SimpleDialog. Ubah isi kode main.dart seperti berikut.



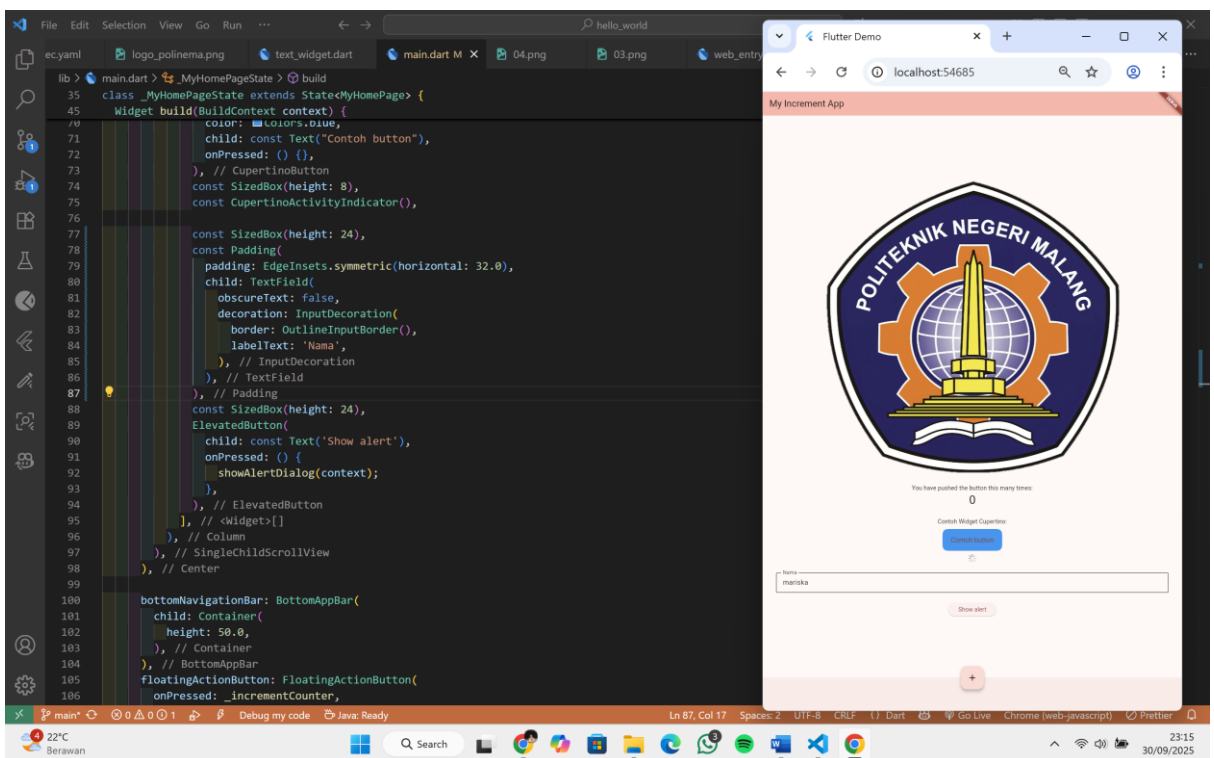
```

lib > main.dart > _MyHomePageState > build
35  class _MyHomePageState extends State<MyHomePage> {
36    void _incrementCounter() {}
37  }
38
39  @override
40  Widget build(BuildContext context) {
41    return Scaffold(
42      appBar: AppBar(
43        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
44        title: Text(widget.title),
45      ), // AppBar
46      body: Center(
47        child: SingleChildScrollView // Ditambahkan agar bisa di-scroll jika konten terlalu panjang
48        padding: const EdgeInsets.symmetric(vertical: 20),
49        child: Column(
50          mainAxisAlignment: MainAxisAlignment.center,
51          children: <Widget>[
52            const MyImageWidget(),
53            const SizedBox(height: 16),
54            const Text(
55              'You have pushed the button this many times:',
56            ), // Text
57            Text(
58              '$_counter',
59              style: Theme.of(context).textTheme.headlineMedium,
60            ), // Text
61            const SizedBox(height: 24),
62            const Text('Contoh Widget Cupertino:'),
63            const SizedBox(height: 8),
64            CupertinoButton(
65              color: Colors.blue,
66              child: const Text("Contoh button"),
67              onPressed: () {},
68            ), // CupertinoButton
69            const SizedBox(height: 8),
70            const CupertinoActivityIndicator(),
71          ],
72        ),
73      ),
74    );
75
76    // <--- PERUBAHAN: Tombol untuk memunculkan dialog
77    const SizedBox(height: 24),
78  }

```



5. Langkah 5: Input dan Selection Widget
Flutter menyediakan widget yang dapat menerima input dari pengguna aplikasi yaitu antara lain Checkbox, Date and Time Pickers, Radio Button, Slider, Switch, TextField.
Contoh penggunaan TextField widget adalah sebagai berikut:





6. 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.

