

**PEMROGRAMAN MOBILE  
PENGANTAR BAHASA PEMROGRAMAN DART -  
BAGIAN 6**



OLEH:

**Nama : Agung Rizky S**

**NIM : 2241720187**

**Kelas : TI – 3C**

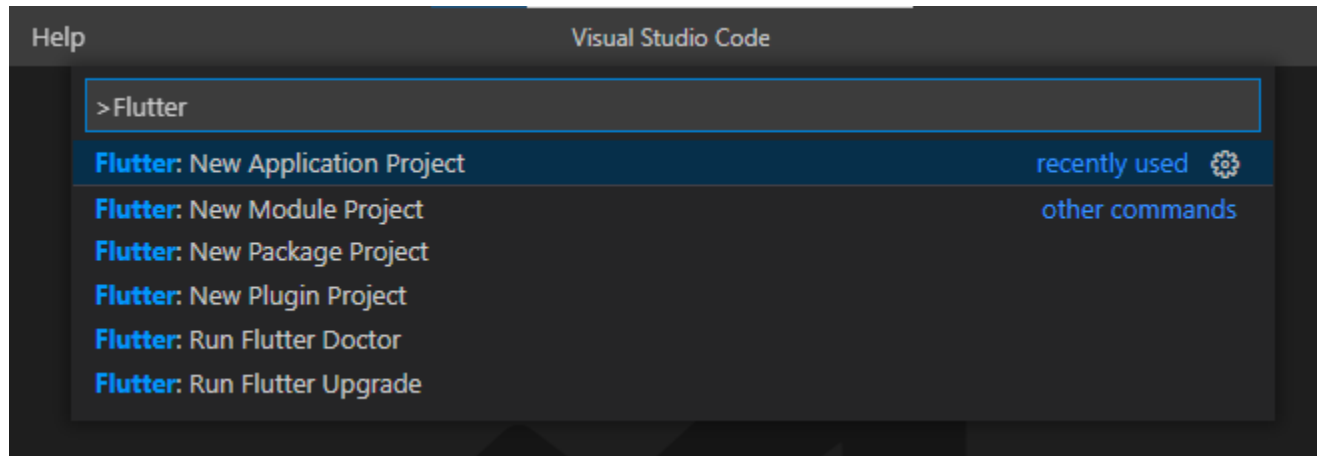
**PROGRAM STUDI D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**2024**

# Praktikum 1: Membuat Project Flutter Baru

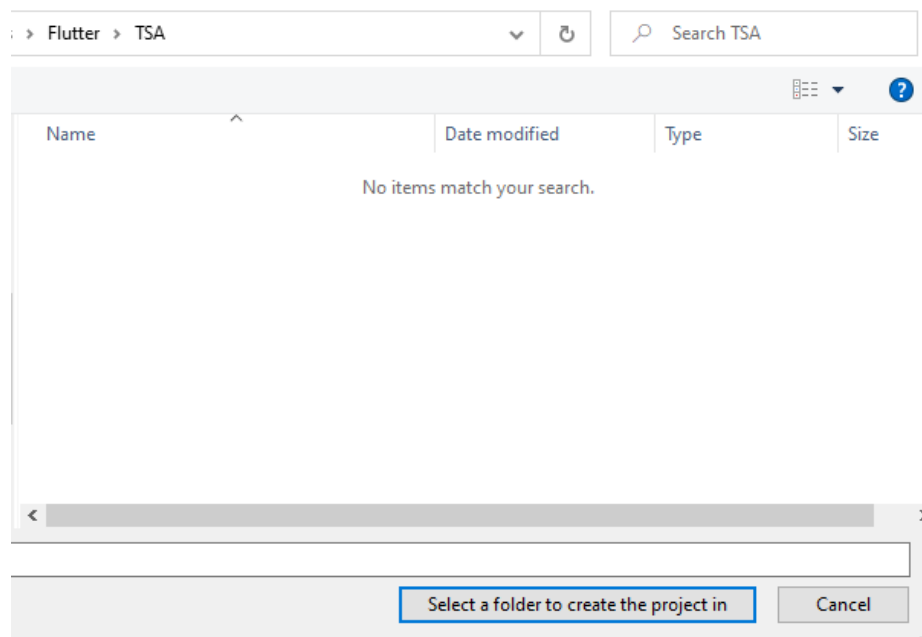
## Langkah 1:

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



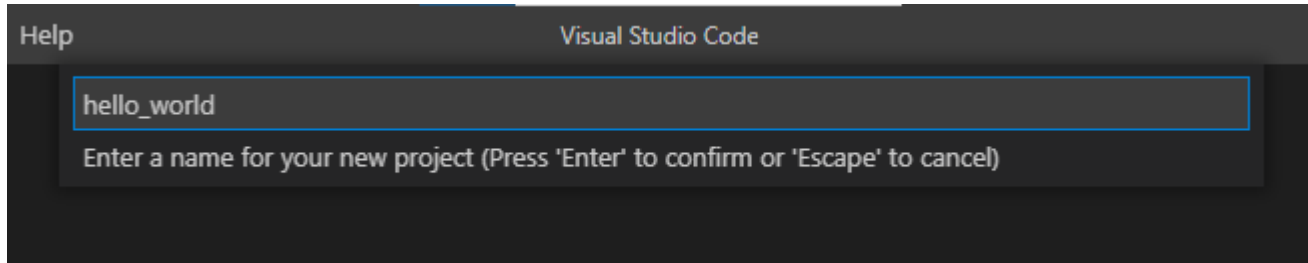
## Langkah 2:

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**.



### Langkah 3:

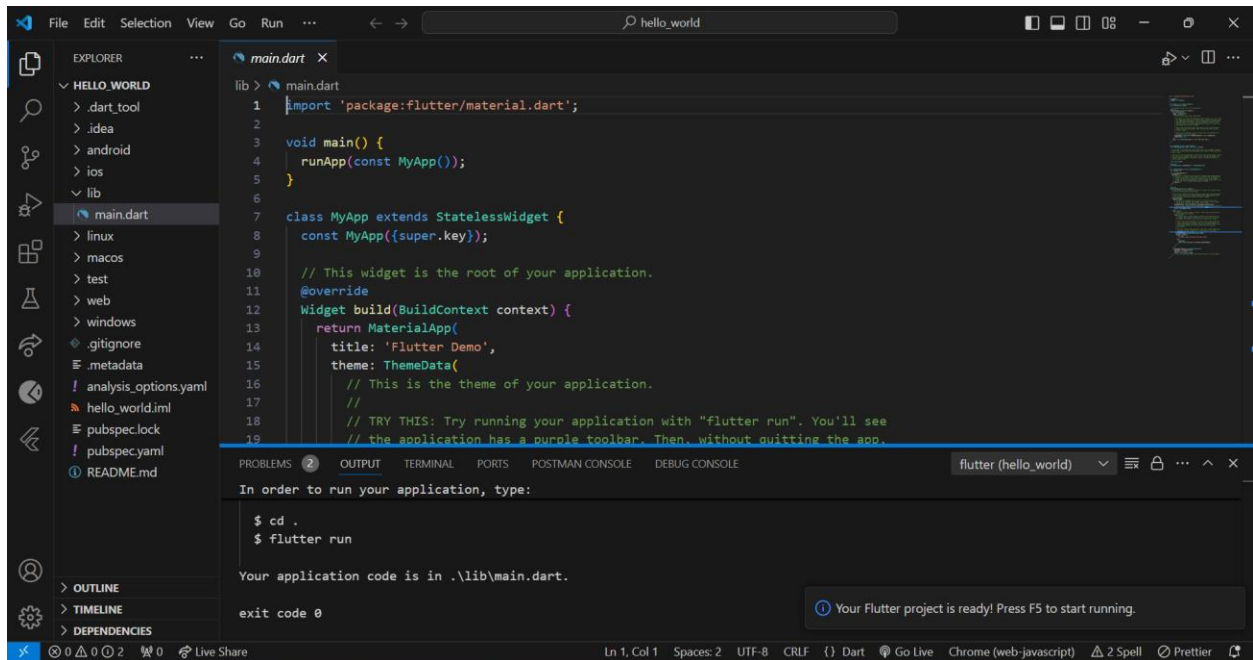
Buat nama project flutter **hello\_world** seperti berikut, lalu tekan **Enter**. Tunggu hingga proses pembuatan project baru selesai.



### Langkah 4:

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.

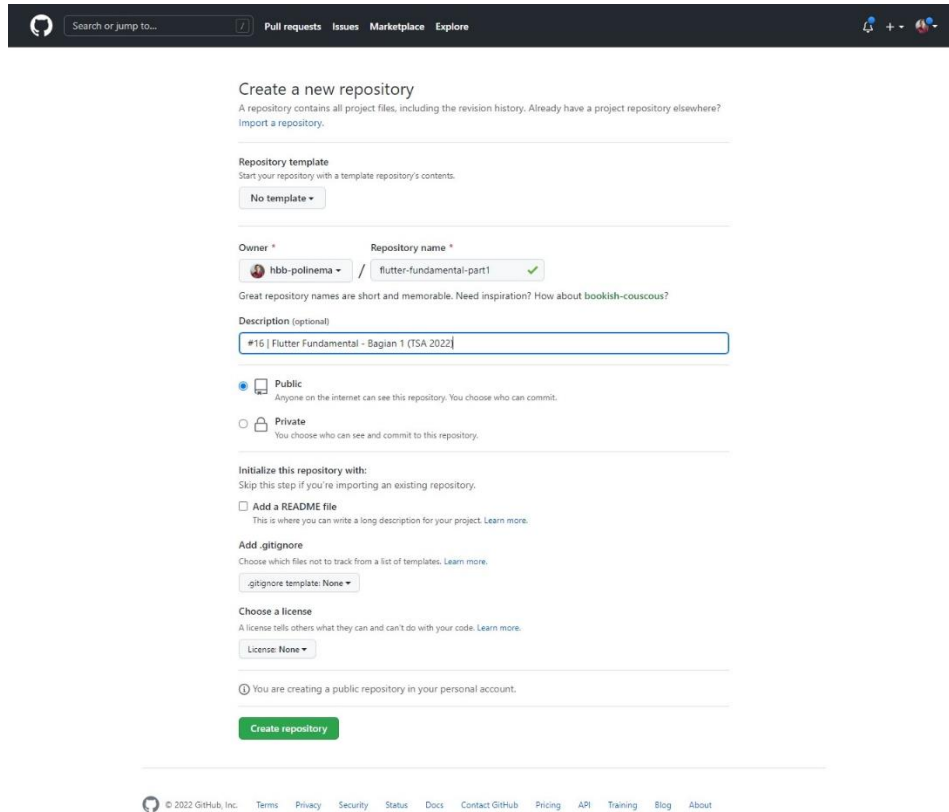
Jawaban :



# Praktikum 2: Membuat Repository GitHub dan Laporan Praktikum

## Langkah 1:

Login ke akun [GitHub](#) Anda, lalu buat repository baru dengan nama "**flutter-fundamental-part1**"



Search or jump to... Pull requests Issues Marketplace Explore

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

**Repository template**  
Start your repository with a template repository's contents.

No template ▾

Owner <sup>\*</sup> Repository name <sup>\*</sup>

hbb-polinema / flutter-fundamental-part1 ✓

Great repository names are short and memorable. Need inspiration? How about [bookish-couscous](#)?

**Description (optional)**

#16 | Flutter Fundamental - Bagian 1 (TSA 2022)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

🔔 You are creating a public repository in your personal account.

**Create repository**

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Jawaban :

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

Owner \* agungrizkysetiawan / Repository name \* flutter-fundamental-part1  
✓ flutter-fundamental-part1 is available.

Great repository names are short and memorable. Need inspiration? How about [verbose-robot](#) ?

Description (optional)  
Flutter Fundamental - Bagian 1

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**  
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**  
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

① You are creating a public repository in your personal account.

[Create repository](#)

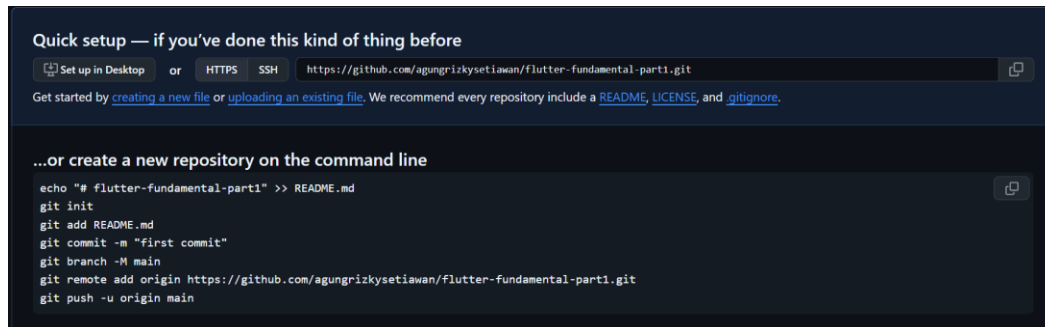
## Langkah 2:

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

The screenshot shows the GitHub interface for a newly created repository named 'flutter-fundamental-part1' under the user 'hbb-polinema'. The repository is public. The page displays 'Quick setup' instructions for cloning the repository using either a desktop client or the command line. The command line instructions are as follows:

```
echo "# flutter-fundamental-part1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/hbb-polinema/flutter-fundamental-part1.git
git push -u origin main
```

**Jawaban :**

A screenshot of the GitHub 'Quick setup' page. It shows two options: 'Set up in Desktop' and 'HTTPS SSH'. The SSH URL is 'https://github.com/agungrizkysetiawan/flutter-fundamental-part1.git'. Below this, it says 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.' The second section, '...or create a new repository on the command line', contains a terminal snippet:

```
echo "# flutter-fundamental-part1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/agungrizkysetiawan/flutter-fundamental-part1.git
git push -u origin main
```

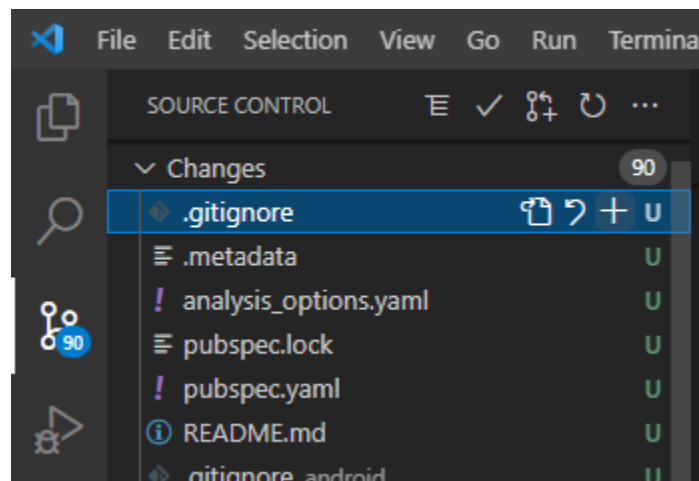
### Langkah 3:

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.

A screenshot of a terminal window with a dark background. The text 'git init' is entered at the prompt.

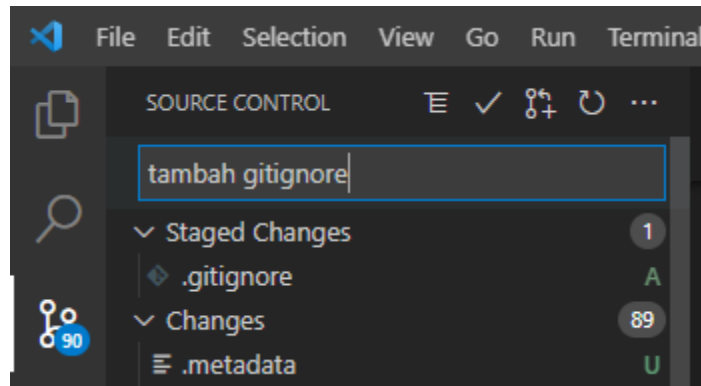
### Langkah 4:

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



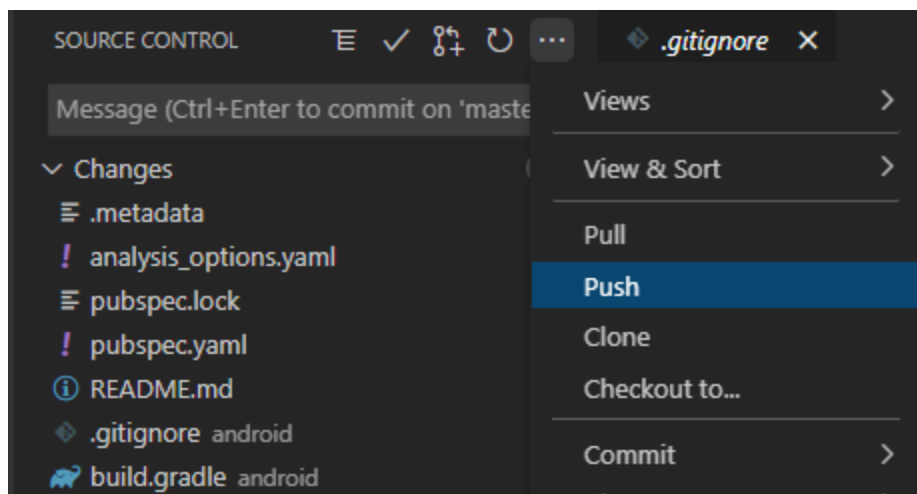
### Langkah 5:

Beri pesan commit "**tambah gitignore**" lalu klik **Commit (✓)**



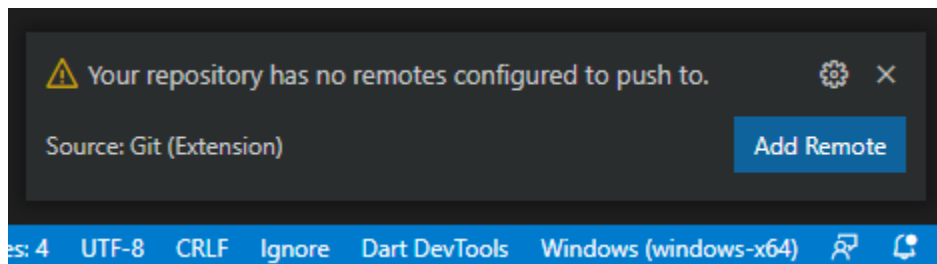
## Langkah 6:

Lakukan push dengan klik bagian menu titik tiga > **Push**



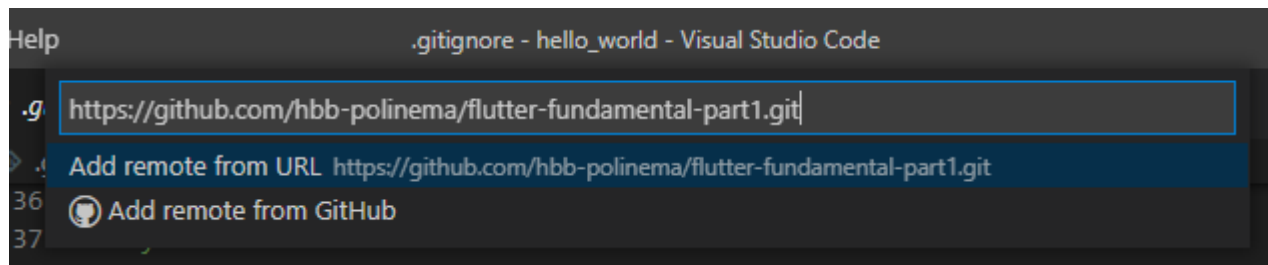
## Langkah 7:

Di pojok kanan bawah akan tampil seperti gambar berikut. Klik "**Add Remote**"

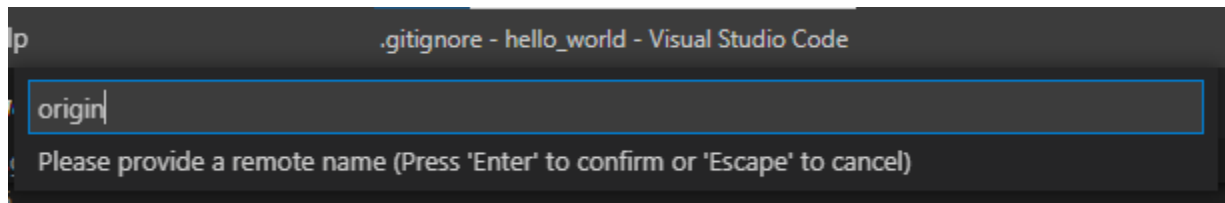


## Langkah 8:

Salin tautan repository Anda dari browser ke bagian ini, lalu klik **Add remote**

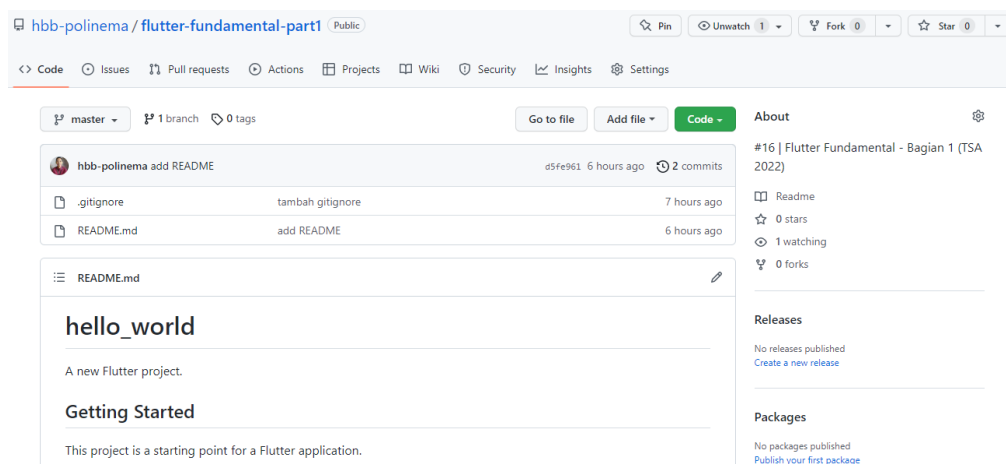


Setelah berhasil, tulis remote name dengan "**origin**"



## Langkah 9:

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.



## Langkah 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.



master

1 branch

0 tags

Go to file

Add file

Code

hbb-polinema project hello\_world

ac7995b 1 minute ago

3 commits

|                       |                     |              |
|-----------------------|---------------------|--------------|
| android               | project hello_world | 1 minute ago |
| ios                   | project hello_world | 1 minute ago |
| lib                   | project hello_world | 1 minute ago |
| test                  | project hello_world | 1 minute ago |
| web                   | project hello_world | 1 minute ago |
| windows               | project hello_world | 1 minute ago |
| .gitignore            | tambah gitignore    | 7 hours ago  |
| .metadata             | project hello_world | 1 minute ago |
| README.md             | add README          | 7 hours ago  |
| analysis_options.yaml | project hello_world | 1 minute ago |
| pubspec.lock          | project hello_world | 1 minute ago |
| pubspec.yaml          | project hello_world | 1 minute ago |

#16 | Flutter Fundamental - Bagian 1 (TSA 2022)

Readme

0 stars

1 watching

0 forks

Releases

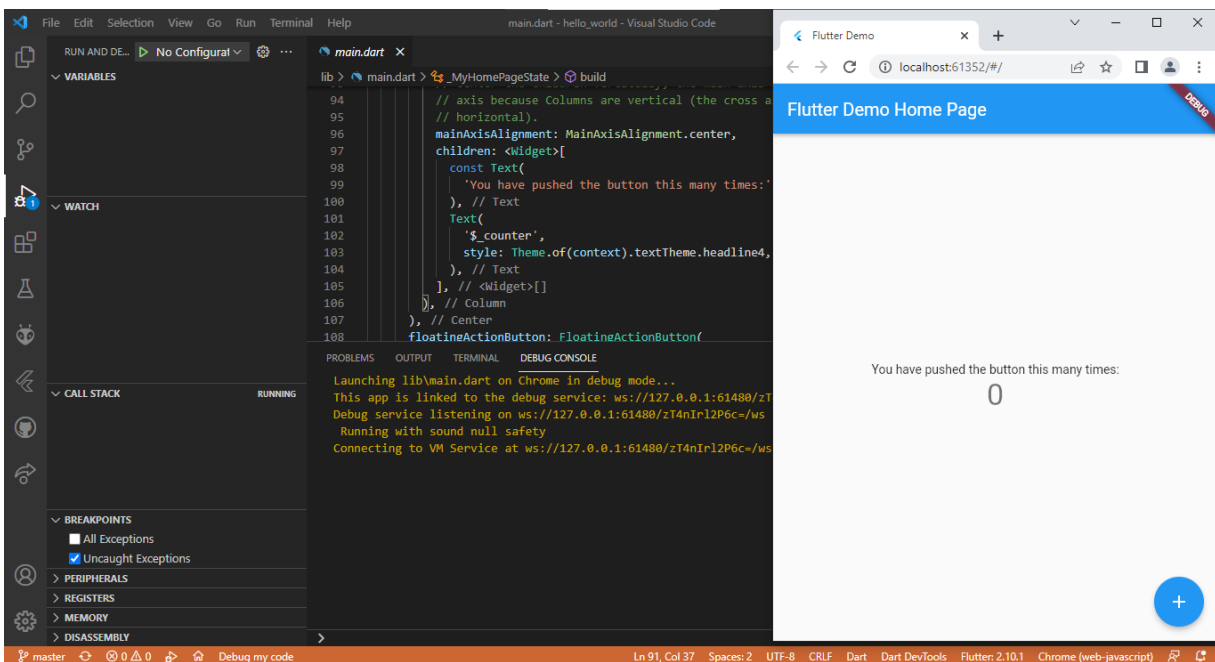
No releases published  
[Create a new release](#)

Packages

No packages published  
[Publish your first package](#)

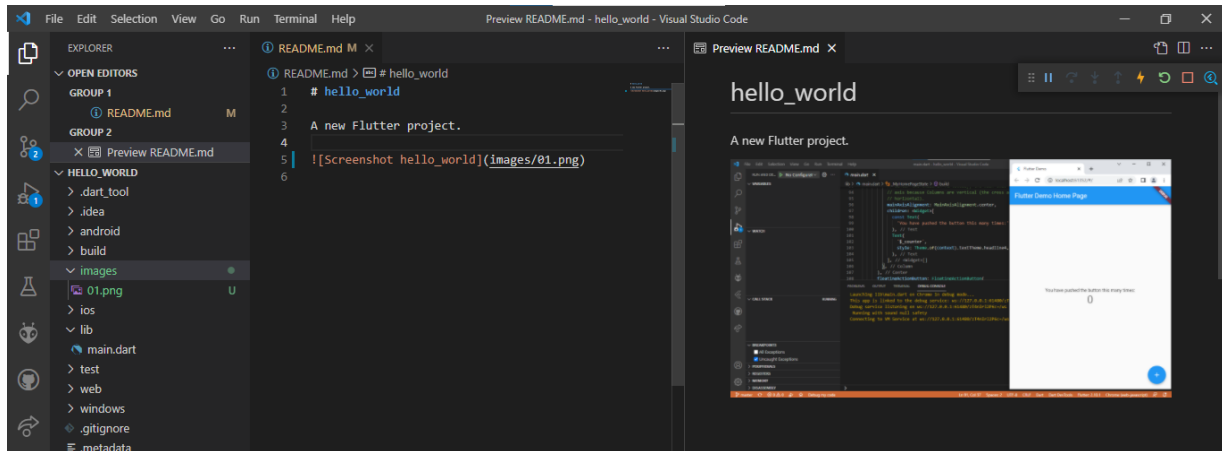
### Langkah 11:

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.

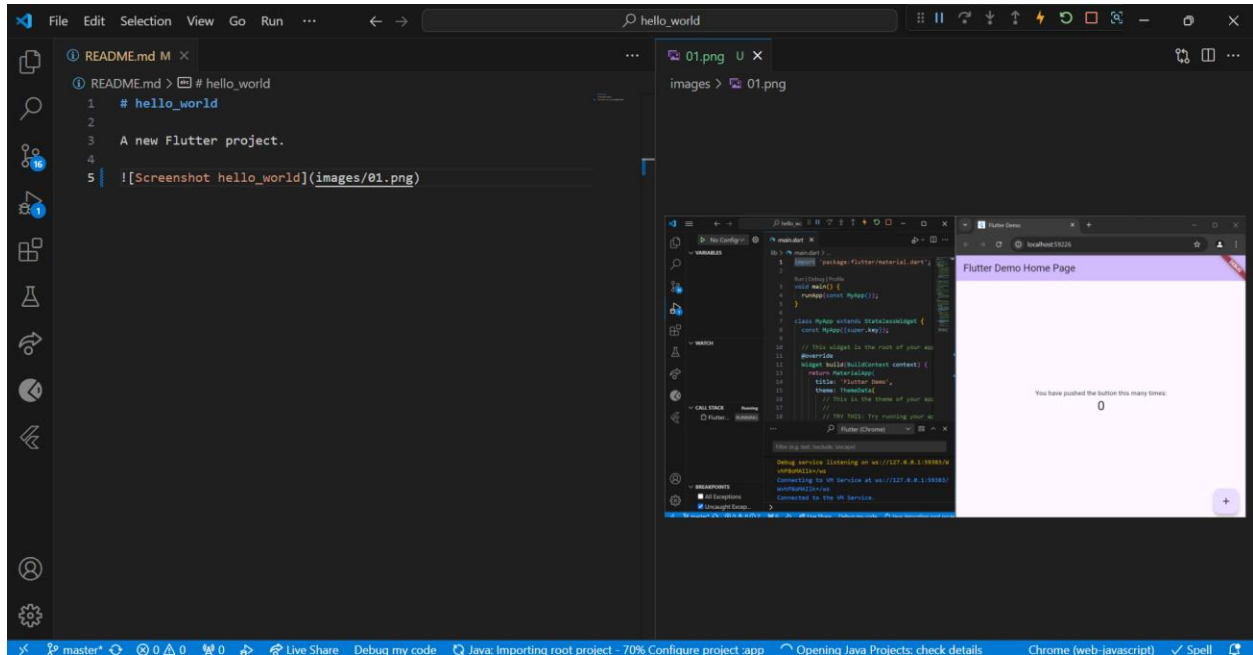


## Langkah 12:

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.



Jawaban :



## Praktikum 3: Menerapkan Widget Dasar

### 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**.

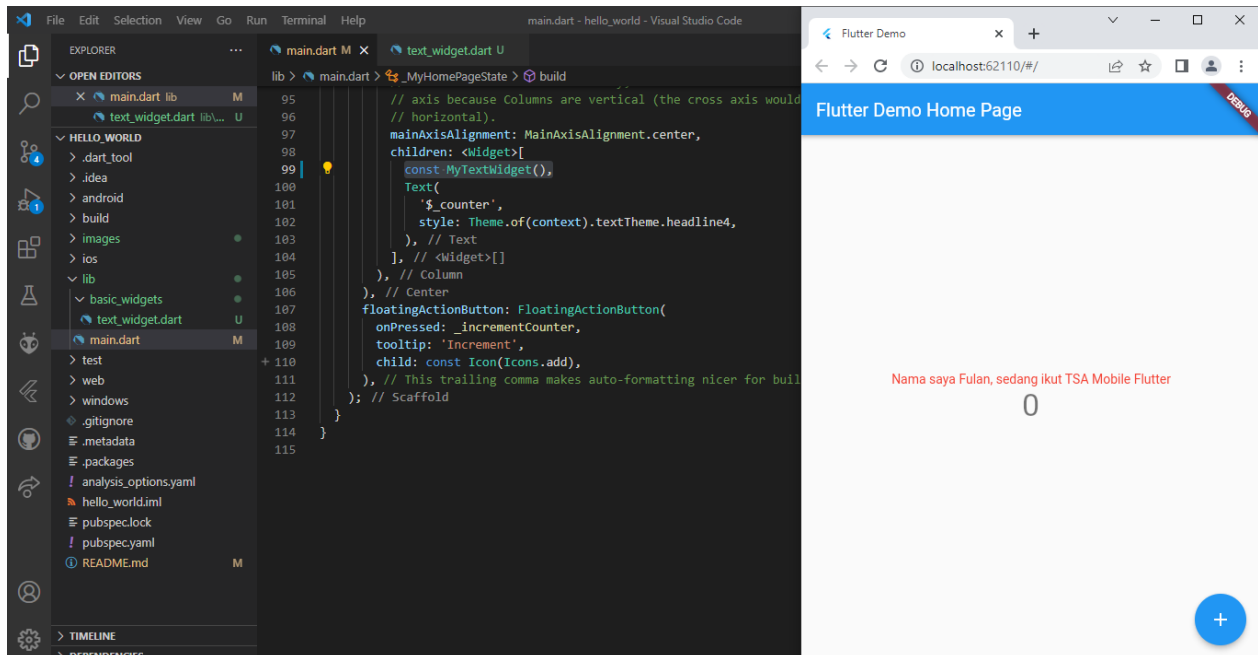
```
import 'package:flutter/material.dart';

class MyTextWidget extends StatelessWidget {
  const MyTextWidget({Key? key}) : super(key: key);

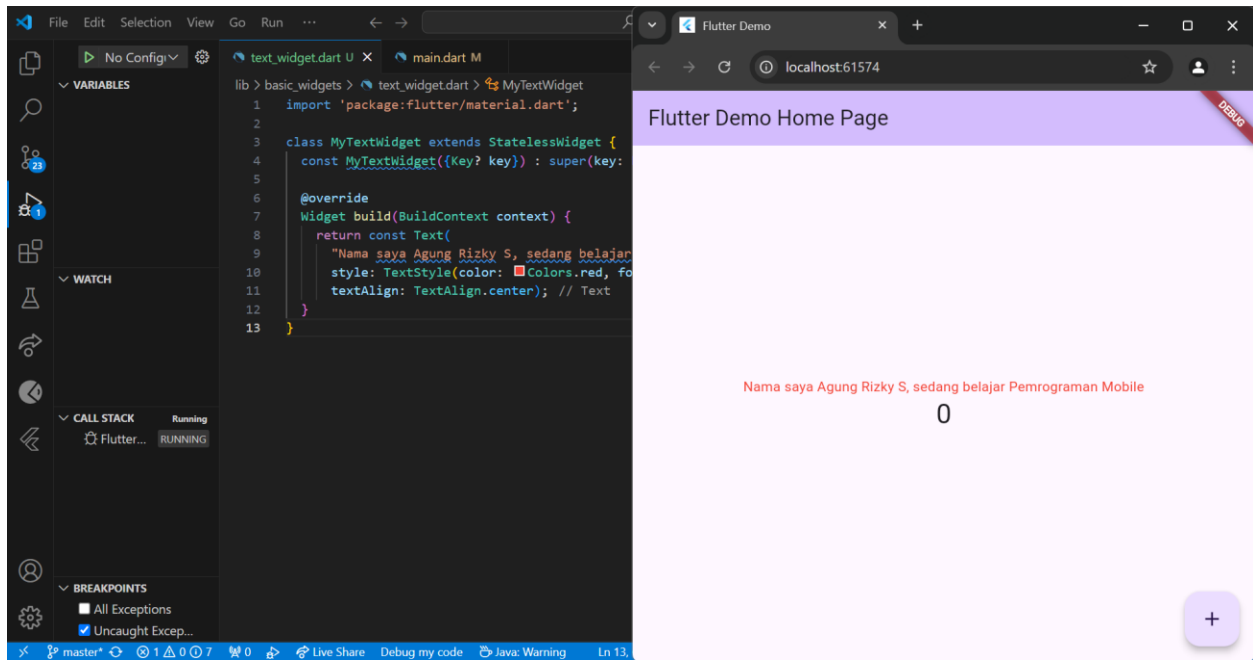
  @override
  Widget build(BuildContext context) {
    return const Text(
      "Nama saya Fulan, sedang belajar Pemrograman Mobile",
      style: TextStyle(color: Colors.red, fontSize: 14),
      textAlign: TextAlign.center;
    );
  }
}
```

**Perhatian:** Gantilah teks **Fulan** dengan nama lengkap Anda.

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**.



Jawaban :



## Langkah 2: Image Widget

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

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

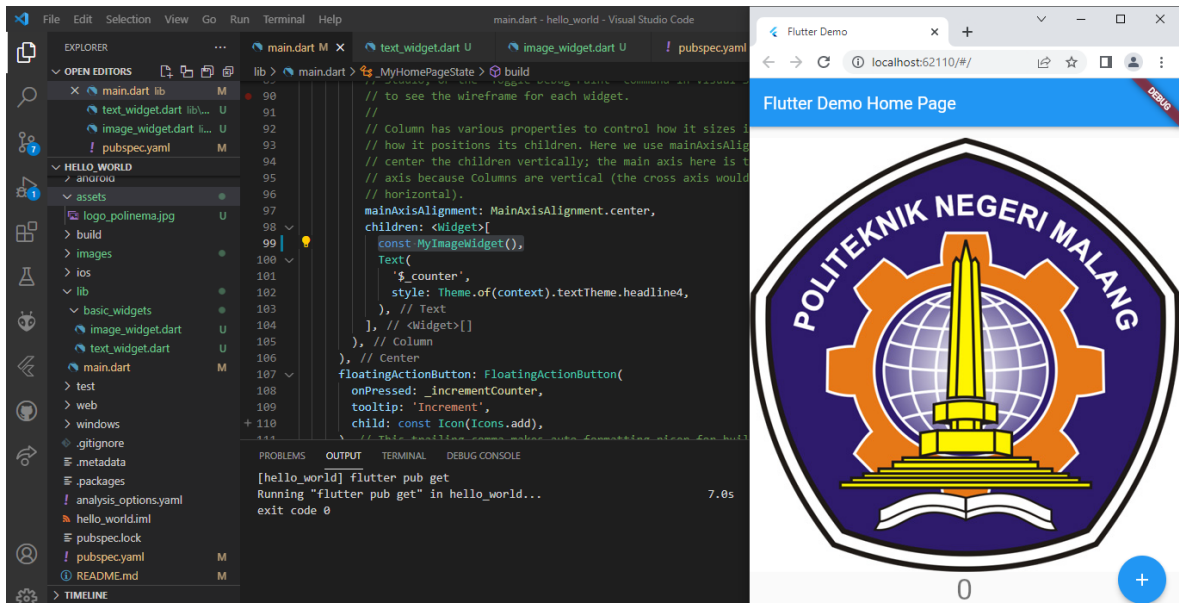
class MyImageWidget extends StatelessWidget {
  const MyImageWidget({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const Image(
      image: AssetImage("logo_polinema.jpg")
    );
  }
}
```

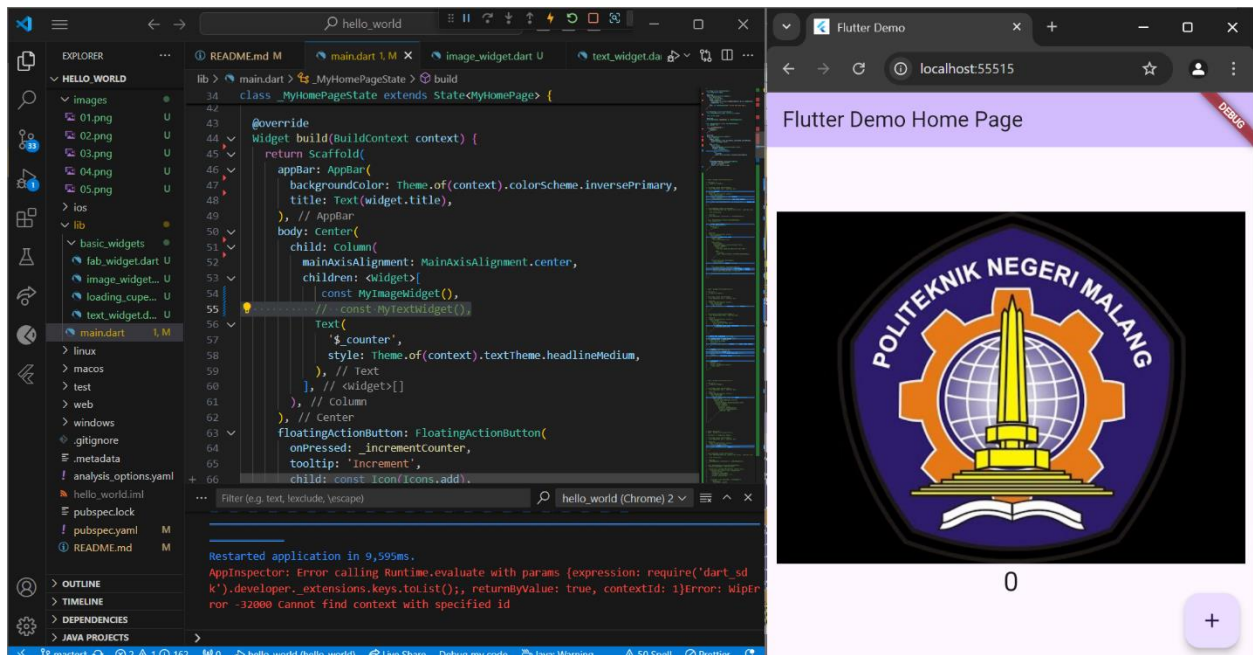
Lakukan penyesuaian asset pada file `pubspec.yaml` dan tambahkan file logo Anda di folder `assets` project `hello_world`.

```
flutter:
  assets:
    - logo_polinema.jpg
```

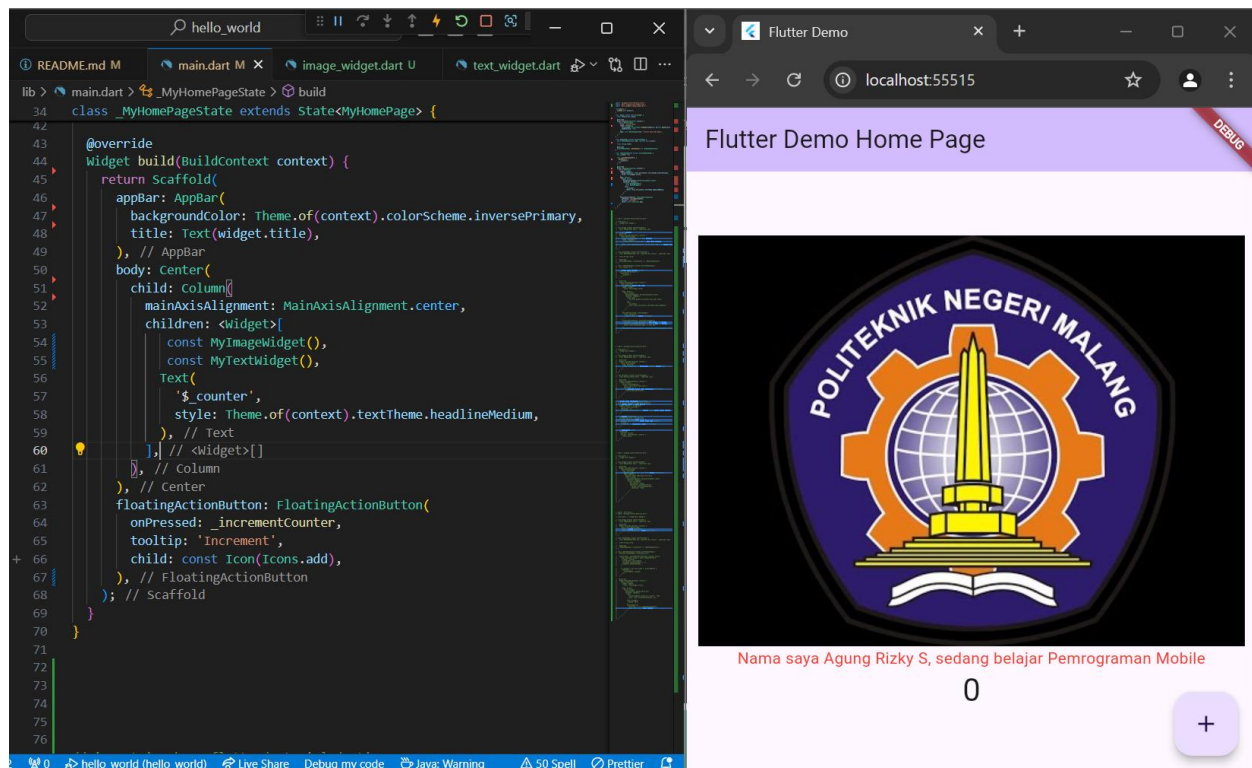
Jangan lupa sesuaikan kode dan import di file `main.dart` kemudian akan tampil gambar seperti berikut.



**Jawaban :**



Tampilan hasil penggabungan antara langkah 1 dan langkah 2 :



## Praktikum 4: Menerapkan Widget Material Design dan iOS Cupertino

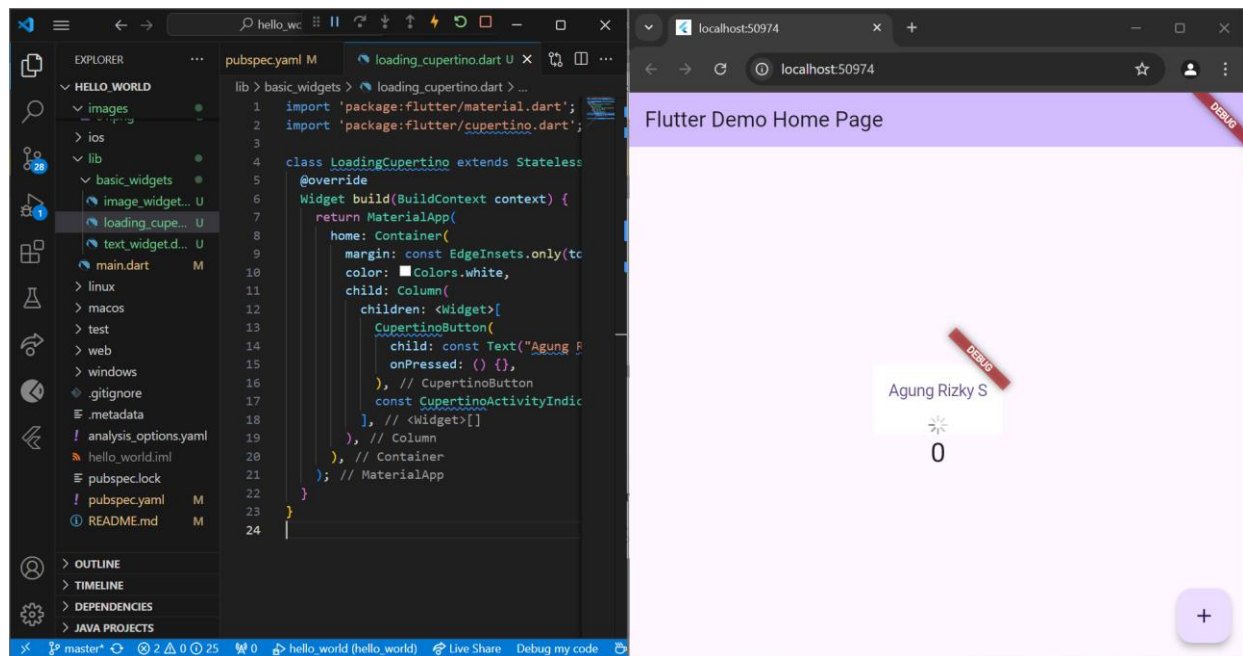
### Langkah 1: Cupertino Button dan Loading Bar

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.

```
return MaterialApp(
  home: Container(
    margin: const EdgeInsets.only(top: 30),
    color: Colors.white,
    child: Column(
      children: <Widget>[
        CupertinoButton(
          child: const Text("Contoh button"),
          onPressed: () {},
        ),
        const CupertinoActivityIndicator(),
      ],
    ),
  ),
);
```

);

Jawaban :



## Langkah 2: Floating Action Button (FAB)

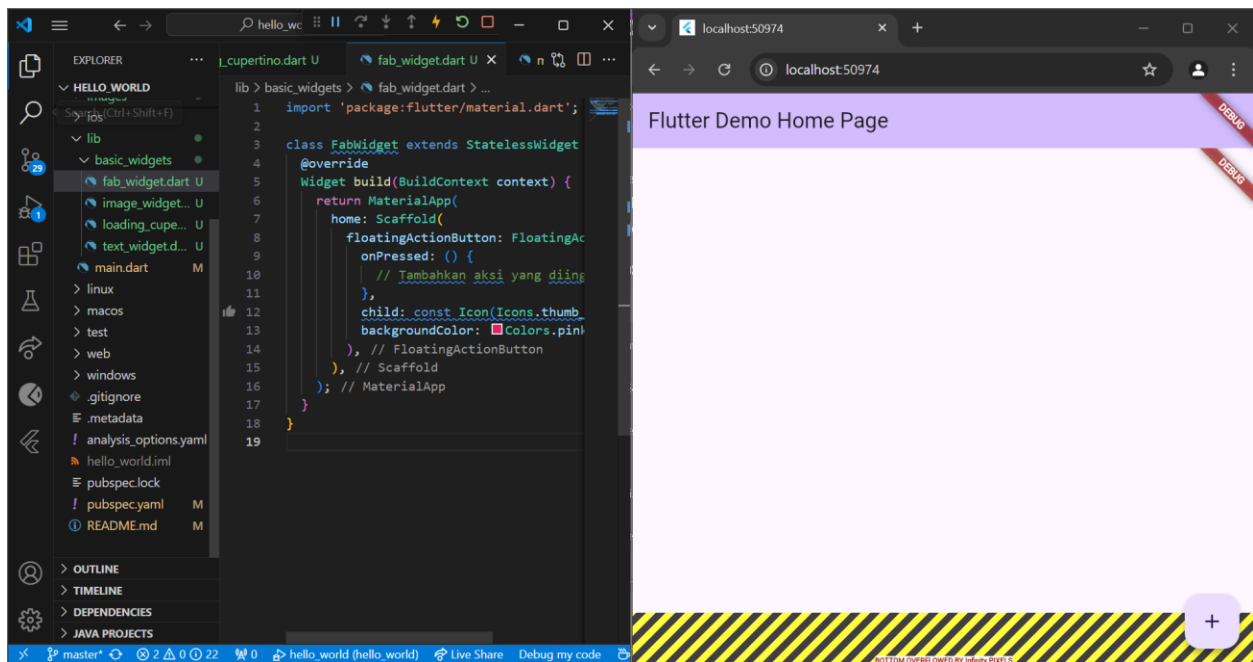
Button widget terdapat beberapa macam pada flutter yaitu ButtonBar, DropdownButton, TextButton, FloatingActionButton, IconButton, OutlineButton, PopupMenuButton, dan ElevatedButton.

Buat file di basic\_widgets > fab\_widget.dart. Import stateless widget dari material. Lalu isi kode di dalam method Widget build adalah sebagai berikut.

```
return MaterialApp(
  home: Scaffold(
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        // Add your onPressed code here!
      },
      child: const Icon(Icons.thumb_up),
      backgroundColor: Colors.pink,
    ),
  ),
);
```

Jawaban :





### Langkah 3: Scaffold Widget

Scaffold widget digunakan untuk mengatur tata letak sesuai dengan material design.

Ubah isi kode `main.dart` seperti berikut.

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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.red,
      ),
    );
  }
}
```



```

    home: const MyHomePage(title: 'My Increment App'),
  );
}
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key, required this.title}) : super(key: key);

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

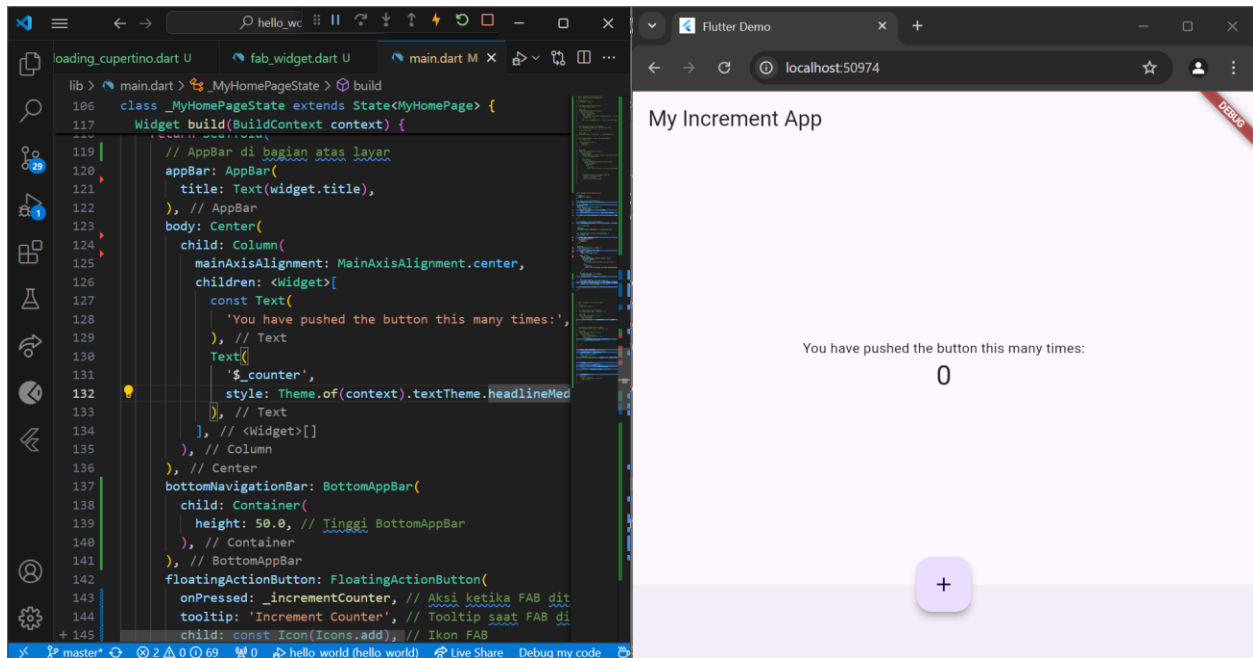
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  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>[
            const Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.headline4,
            ),
          ],
        ),
      ),
      bottomNavigationBar: BottomAppBar(
        child: Container(
          height: 50.0,
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment Counter',
        child: const Icon(Icons.add),
      ),
      floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
    );
  }
}

```

Jawaban :



## Langkah 4: Dialog Widget

Dialog widget pada flutter memiliki dua jenis dialog yaitu `AlertDialog` dan `SimpleDialog`.

Ubah isi kode `main.dart` seperti berikut.

```
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return const MaterialApp(  
      home: Scaffold(  
        body: MyLayout(),  
      ),  
    );  
  }  
}  
  
class MyLayout extends StatelessWidget {  
  const MyLayout({Key? key}) : super(key: key);
```

```

@override
Widget build(BuildContext context) {
  return Padding(
    padding: const EdgeInsets.all(8.0),
    child: ElevatedButton(
      child: const Text('Show alert'),
      onPressed: () {
        showAlertDialog(context);
      },
    ),
  );
}

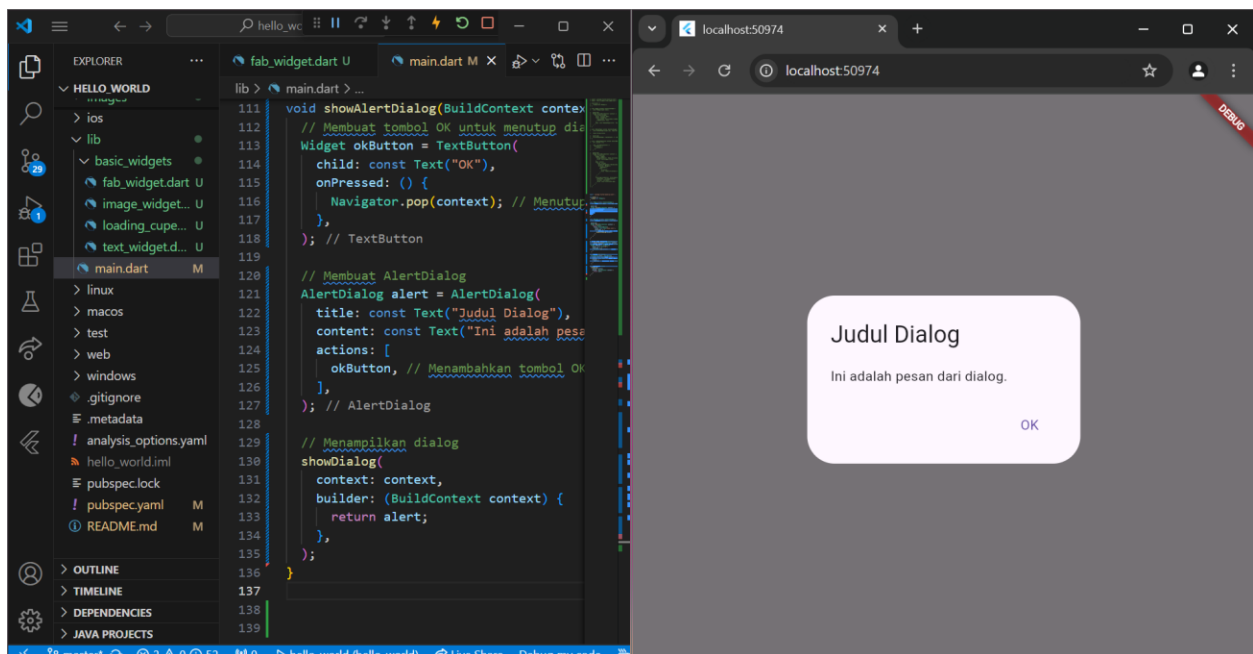
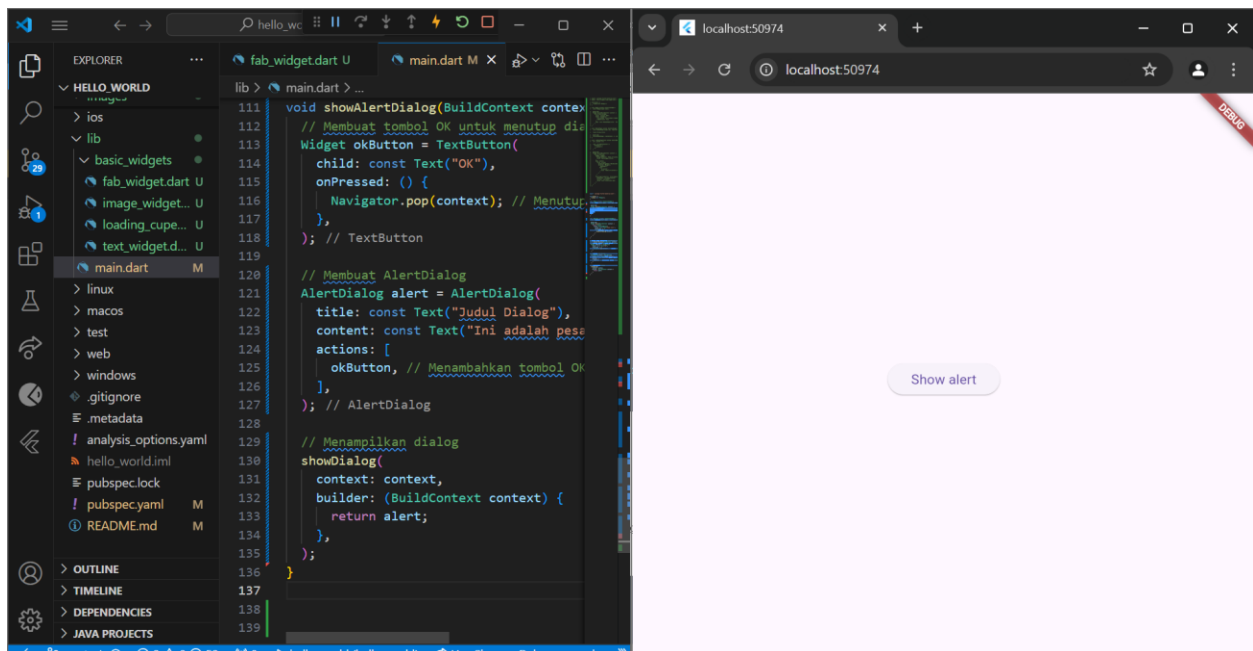
showAlertDialog(BuildContext context) {
  // set up the button
  Widget okButton = TextButton(
    child: const Text("OK"),
    onPressed: () {
      Navigator.pop(context);
    },
  );

  // set up the AlertDialog
  AlertDialog alert = AlertDialog(
    title: const Text("My title"),
    content: const Text("This is my message."),
    actions: [
      okButton,
    ],
  );

  // show the dialog
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return alert;
    },
  );
}

```

Jawaban :



## 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:

```

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

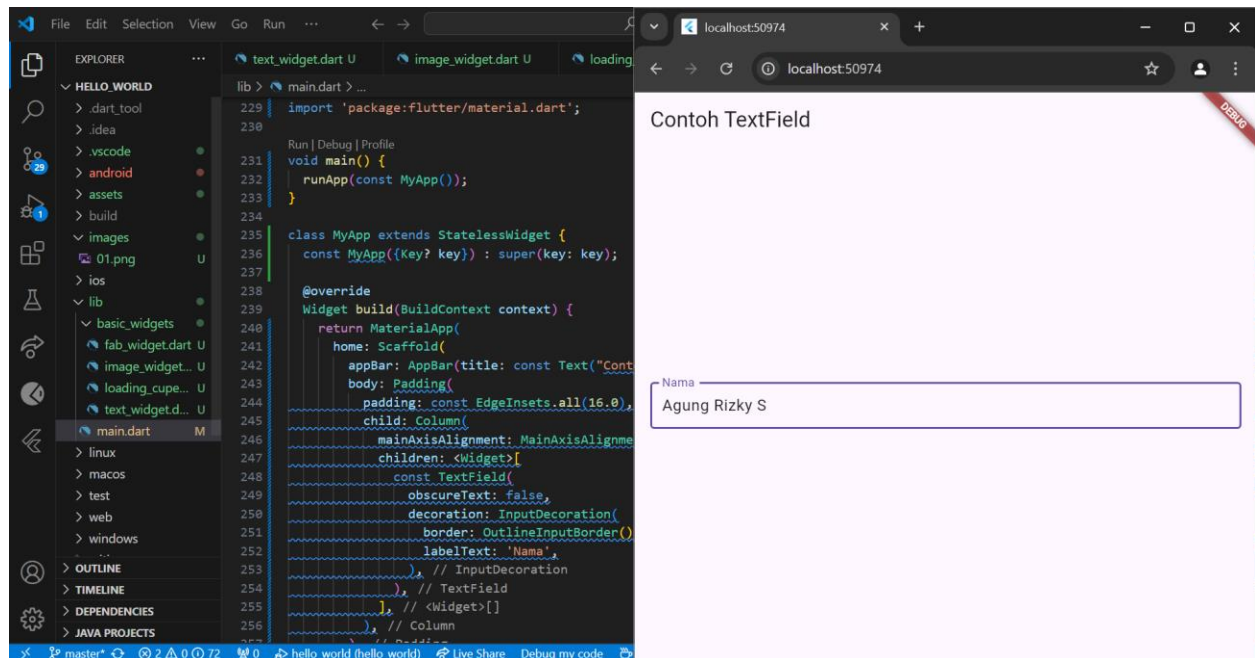
```

```

@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(title: const Text("Contoh TextField")),
      body: const TextField(
        obscureText: false,
        decoration: InputDecoration(
          border: OutlineInputBorder(),
          labelText: 'Nama',
        ),
      ),
    ),
  );
}

```

Jawaban :



## 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.

```

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

void main() => runApp(const MyApp());

```

```

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      title: 'Contoh Date Picker',
      home: MyHomePage(title: 'Contoh Date Picker'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key, required this.title}) : super(key: key);

  final String title;

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  // Variable/State untuk mengambil tanggal
  DateTime selectedDate = DateTime.now();

  // Initial SelectDate FLutter
  Future<void> _selectDate(BuildContext context) async {
    // Initial DateTime Ffinal Picked
    final DateTime? picked = await showDatePicker(
      context: context,
      initialDate: selectedDate,
      firstDate: DateTime(2015, 8),
      lastDate: DateTime(2101));
    if (picked != null && picked != selectedDate) {
      setState() {
        selectedDate = picked;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: <Widget>[
            Text("${selectedDate.toLocal()}".split(' ')[0]),
            const SizedBox(
              height: 20.0,
            ),
            ElevatedButton(
              onPressed: () => {

```

```

        _selectDate(context),
        // ignore: avoid_print
        print(selectedDate.day + selectedDate.month + selectedDate.year)
      },
      child: const Text('Pilih Tanggal'),
    ),
  ],
),
),
);
}
}

```

Jawaban :

