

Pemrograman Mobile

[JOBSITE 4]

FLUTTER 1



Veren Regina Tirsya

2341760127

D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI

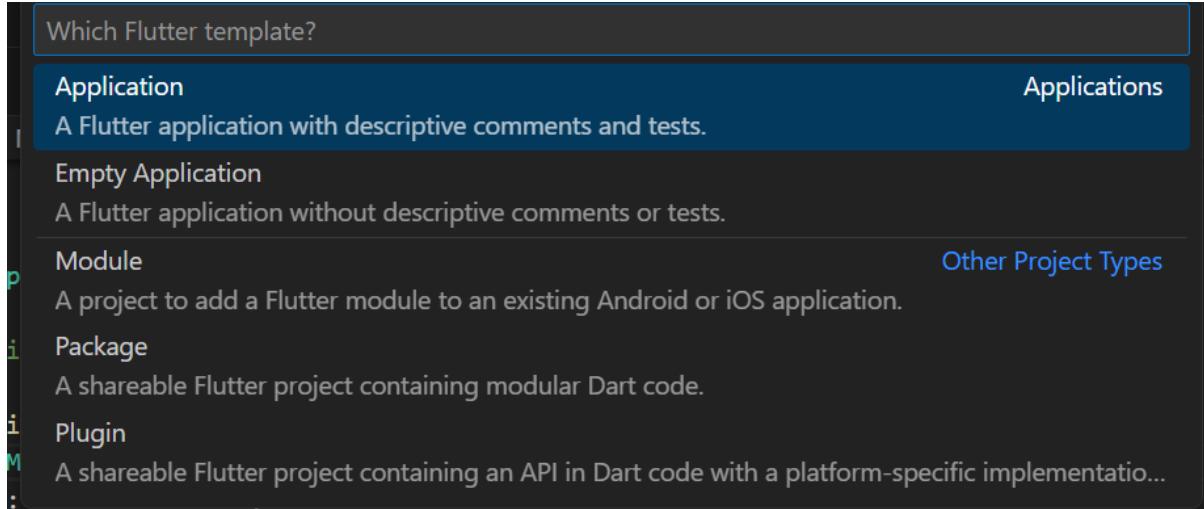
POLITEKNIK NEGERI MALANG

2025

PRAKTIKUM 1: Membuat Project Flutter Baru

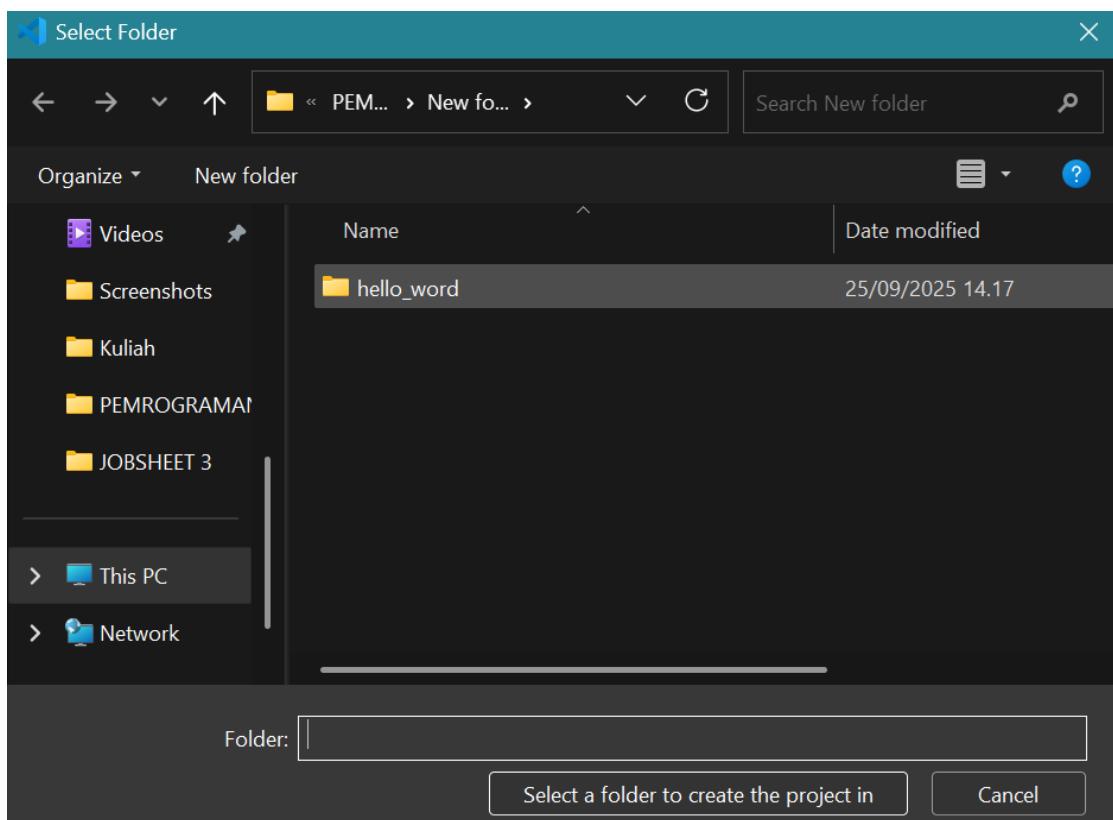
Langkah 1:

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



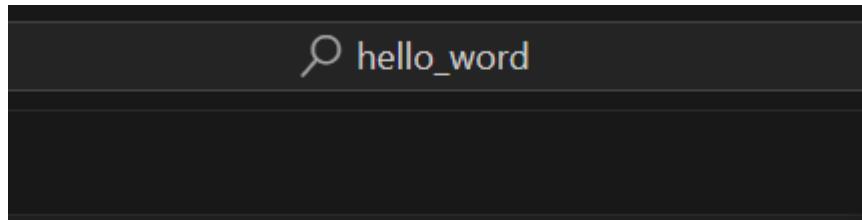
Langkah 2:

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.



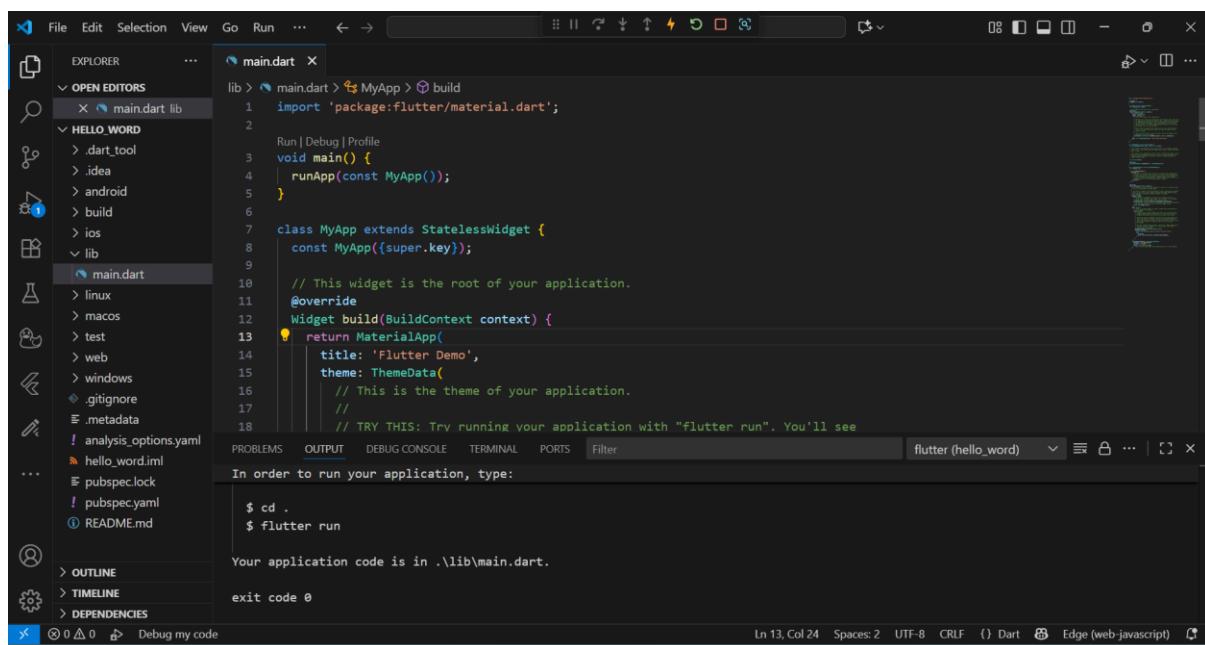
Langkah 3

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.



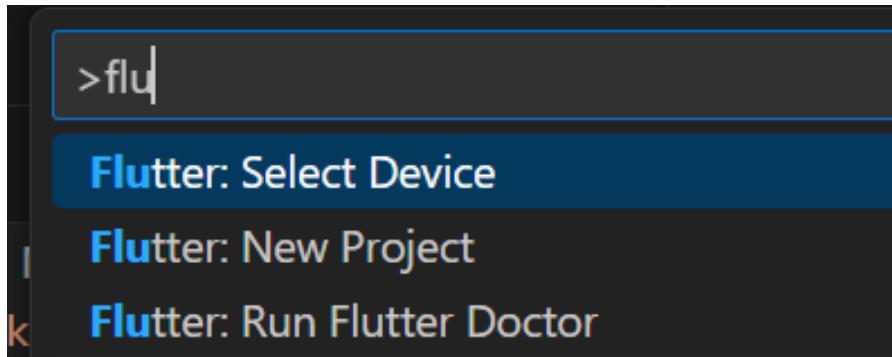
PRAKTIKUM 2: Menghubungkan Perangkat Android atau Emulator

Langkah 1

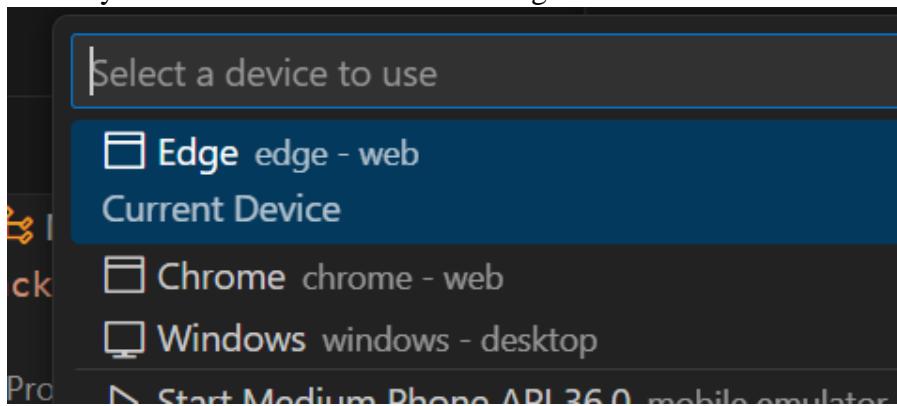
Melanjutkan dari praktikum 1, Anda diminta untuk menjalankan aplikasi ke perangkat fisik (device Android atau iOS). Silakan ikuti langkah-langkah pada codelab tautan berikut ini.

<https://developer.android.com/codelabs/basic-android-kotlin-compose-connect-device?hl=id#0>

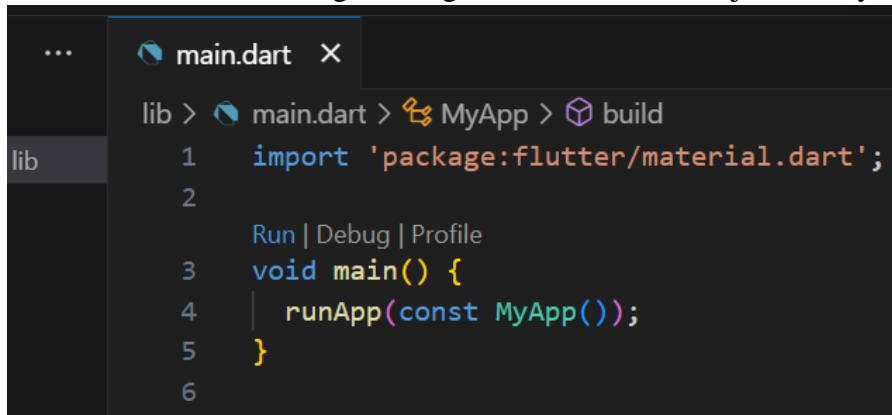
1. Menekan tombol search di VS Code, lalu muncul Command Palette, lalu ketik “>Flutter: Select Device”



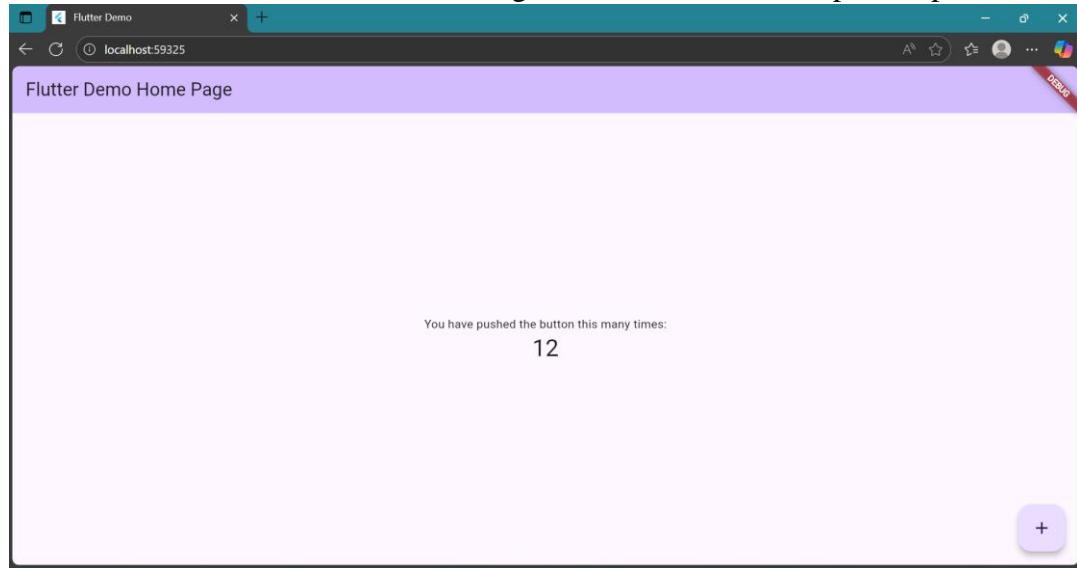
2. Lalu saya memilih untuk membuka di edge



3. Setelah memilih dan mengklik Edge, klik run untuk menjalankannya



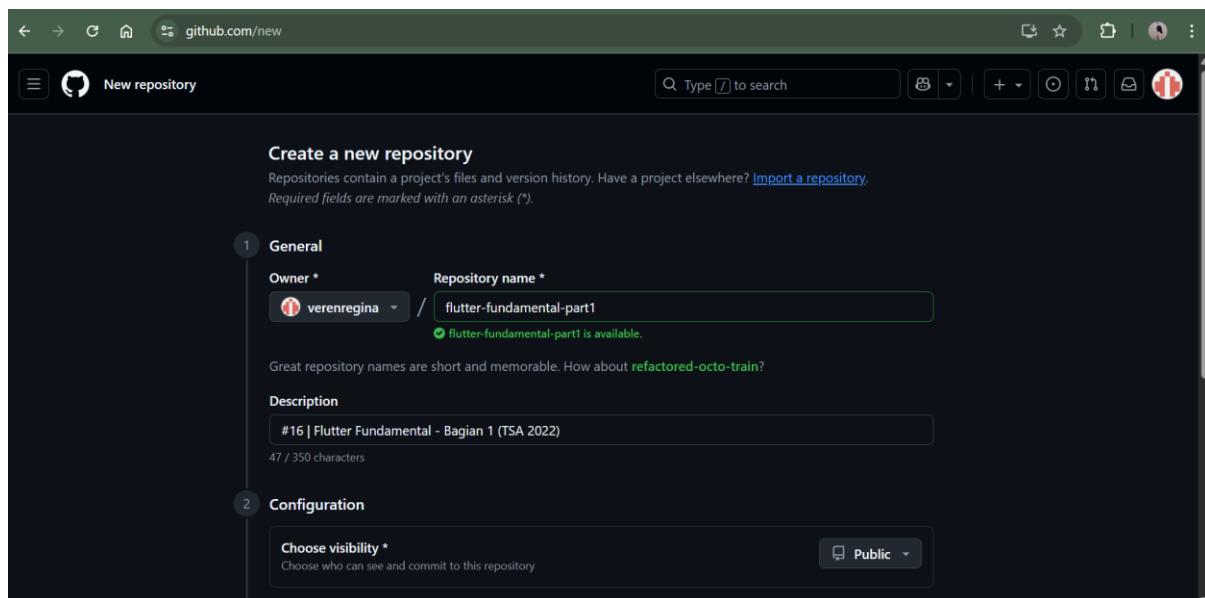
4. Setelah itu akan otomatis terbuka di Edge, dan akan muncul tampilan seperti ini



PRAKTIKUM 3: Membuat Repository GitHub dan Laporan Praktikum

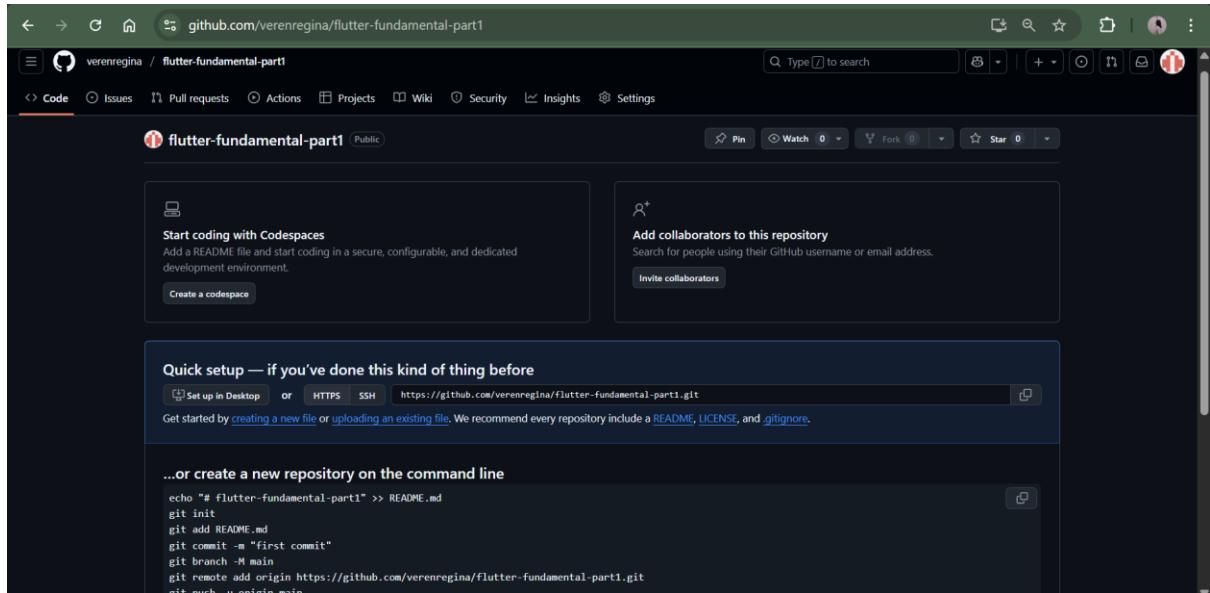
Langkah 1

Login ke akun GitHub Anda, lalu buat repository baru dengan nama "flutter-fundamental-part1"



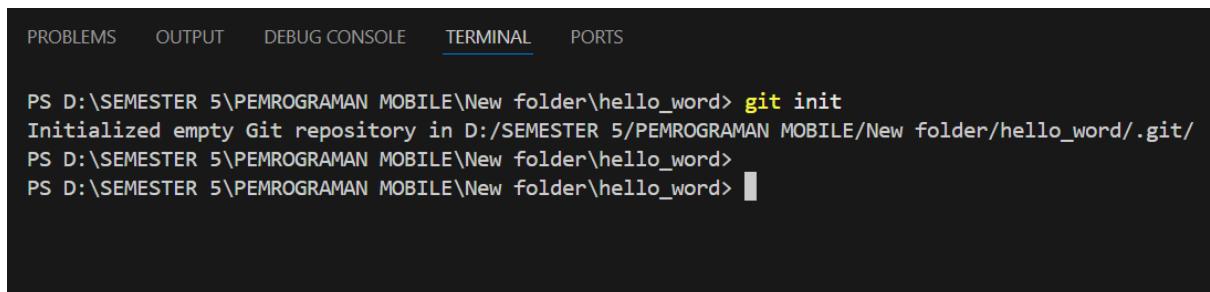
Langkah 2

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



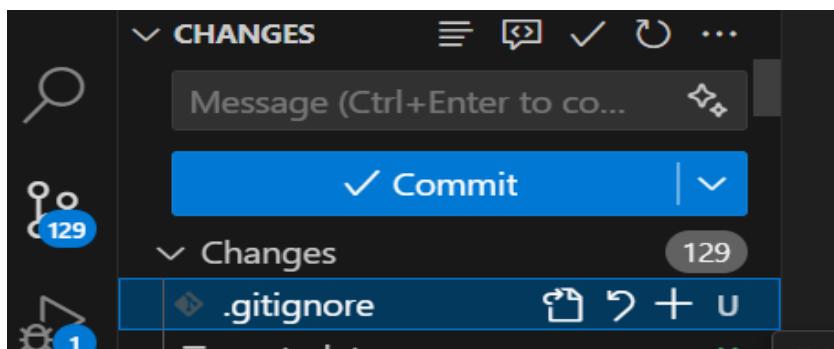
Langkah 3

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.



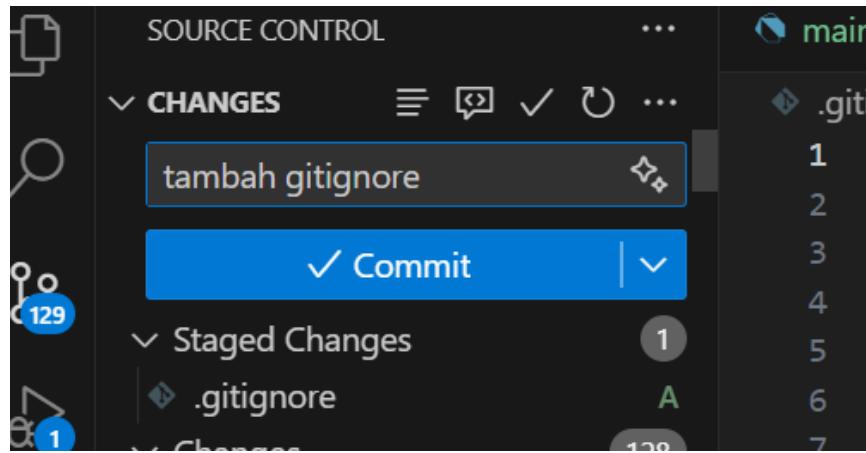
Langkah 4

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



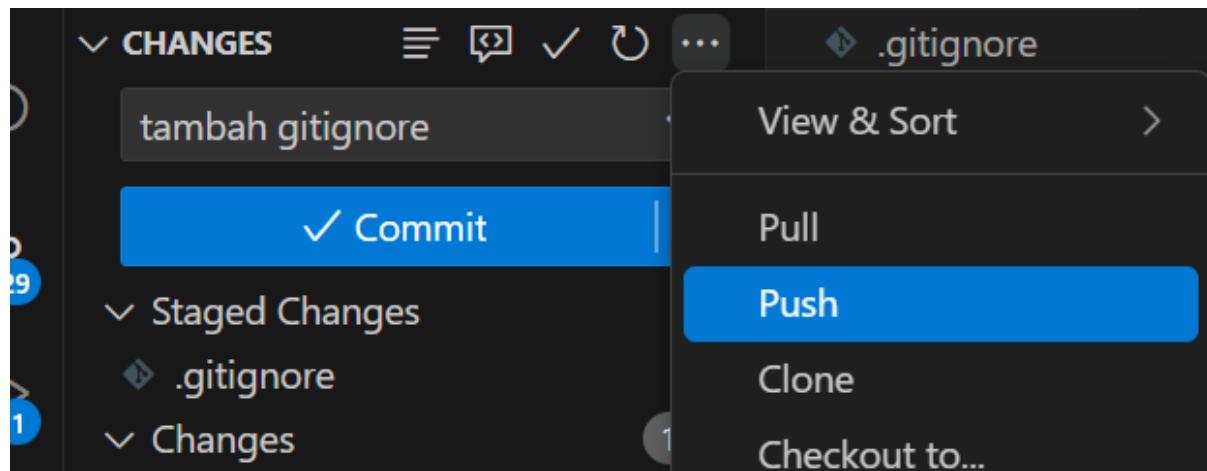
Langkah 5

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



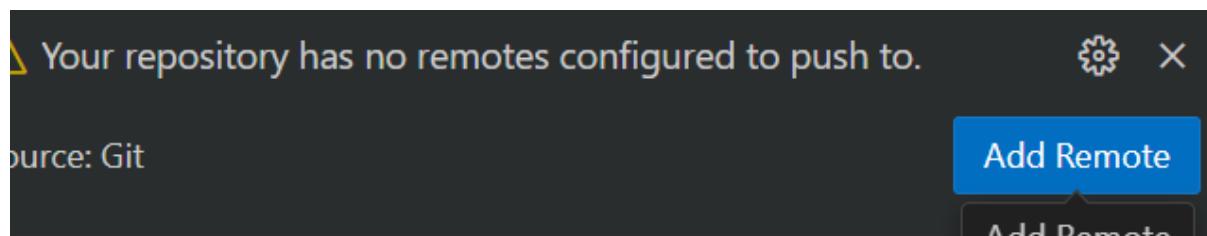
Langkah 6

Lakukan push dengan klik bagian menu titik tiga > Push



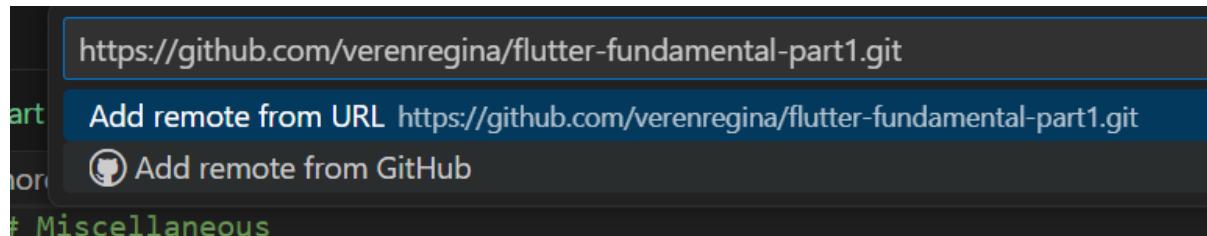
Langkah 7

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



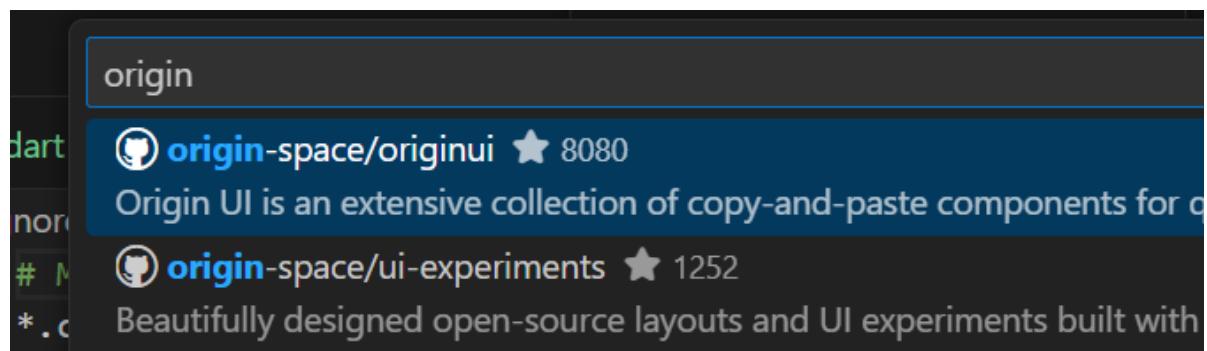
```
https://github.com/verenregina/flutter-fundamental-part1.git
```

Add remote from URL https://github.com/verenregina/flutter-fundamental-part1.git

Add remote from GitHub

Miscellaneous

Setelah berhasil, tulis remote name dengan "origin"



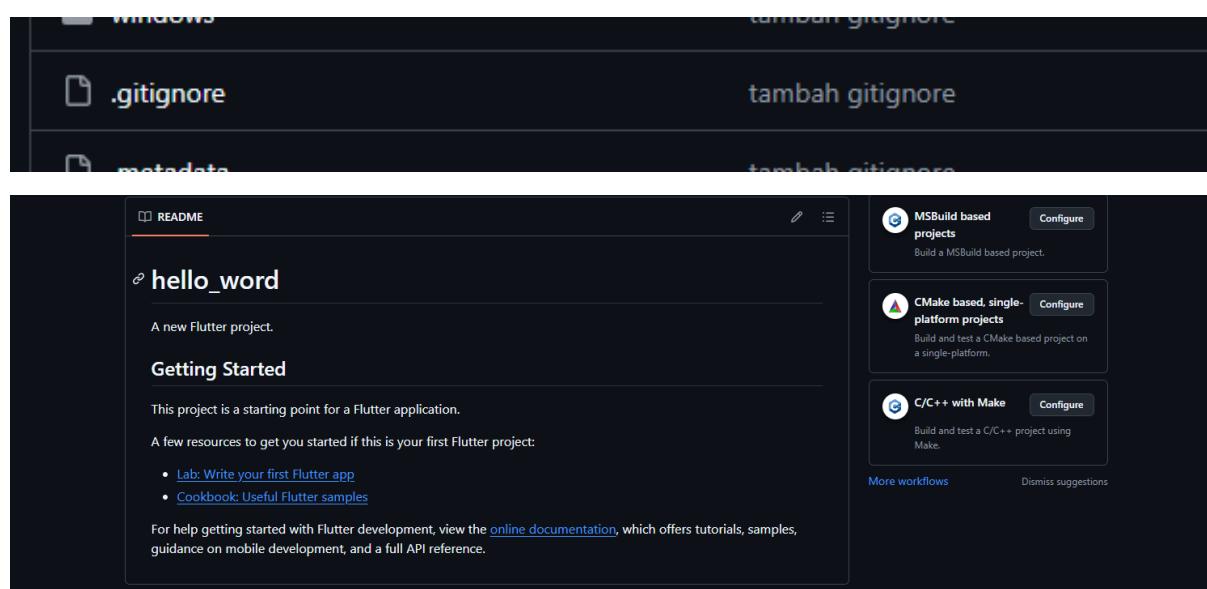
```
origin
```

origin-space/originui ★ 8080
Origin UI is an extensive collection of copy-and-paste components for c

origin-space/ui-experiments ★ 1252
Beautifully designed open-source layouts and UI experiments built with

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.



tambah gitignore

tambah gitignore

tambah gitignore

README

hello_world

A new Flutter project.

Getting Started

This project is a starting point for a Flutter application.

A few resources to get you started if this is your first Flutter project:

- [Lab: Write your first Flutter app](#)
- [Cookbook: Useful Flutter samples](#)

For help getting started with Flutter development, view the [online documentation](#), which offers tutorials, samples, guidance on mobile development, and a full API reference.

MSBuild based projects

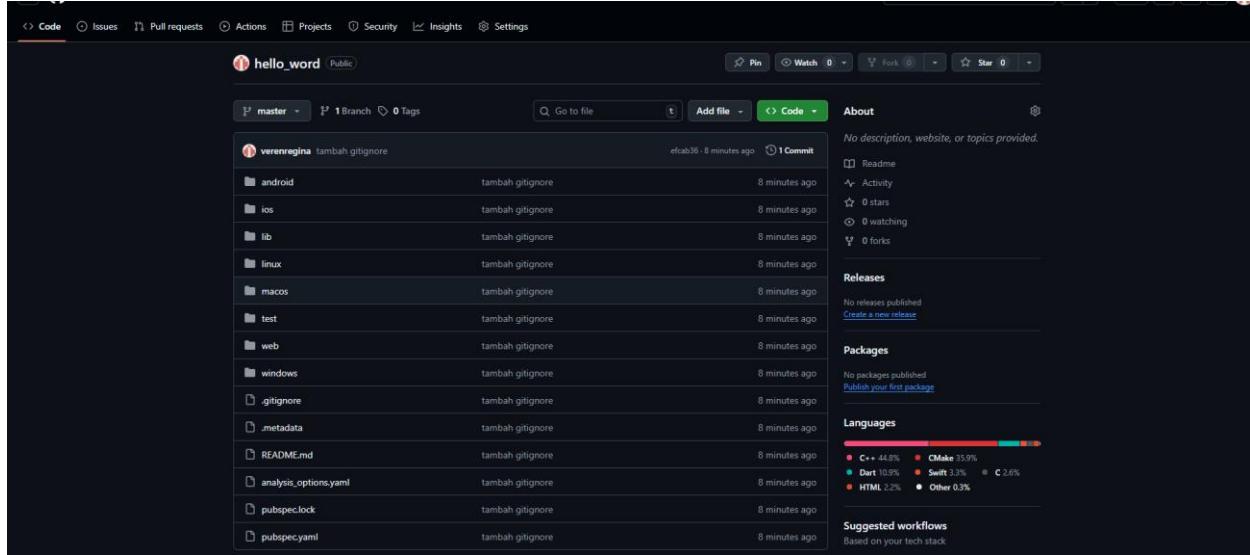
CMake based, single-platform projects

C/C++ with Make

More workflows Dismiss suggestions

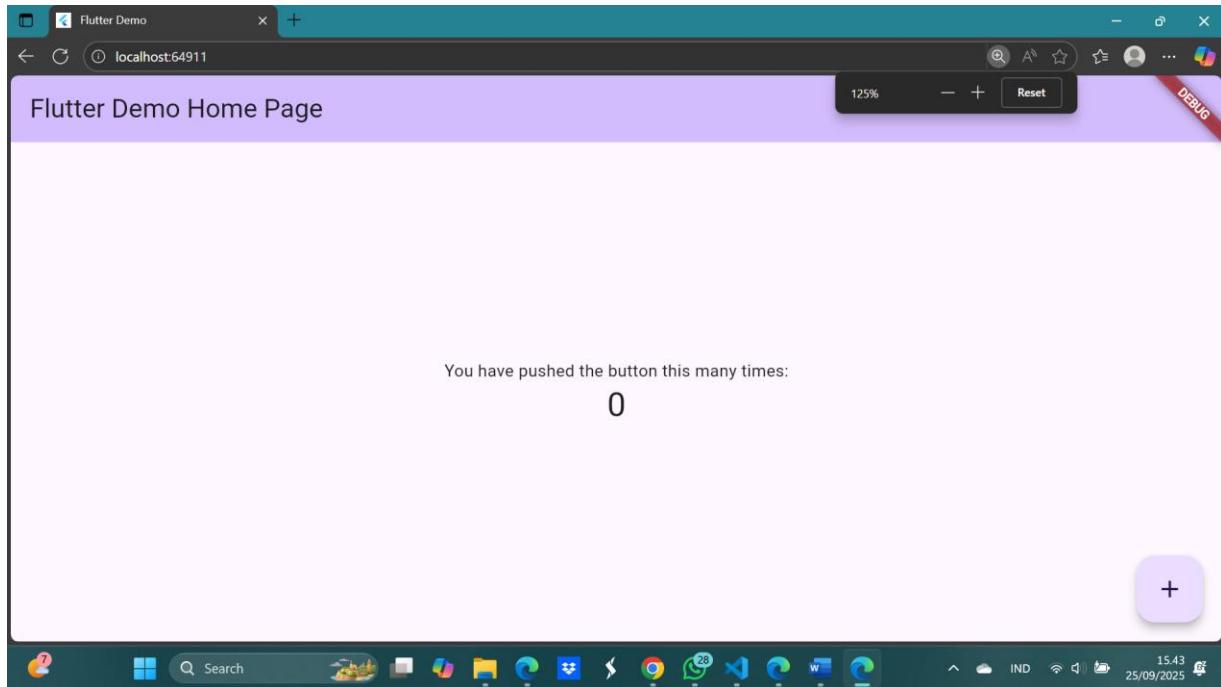
Langkah 10

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.



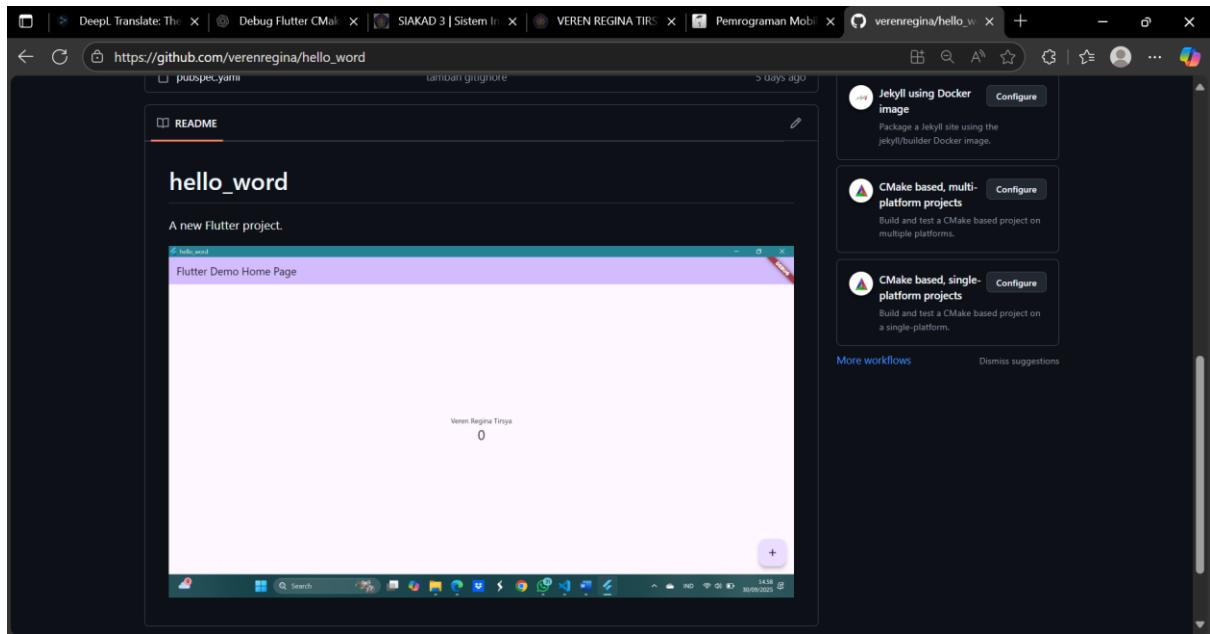
Langkah 11

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.



PRAKTIKUM 4: Menerapkan Widget Dasar

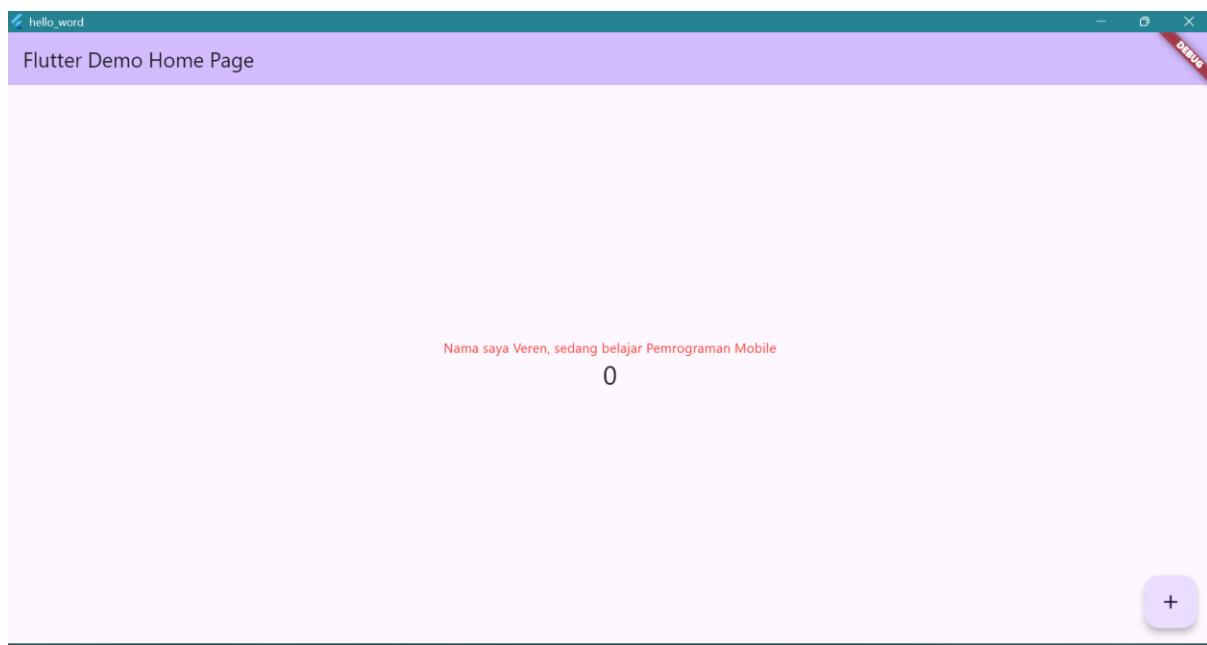
Langkah 1

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

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.

```
105     // wireframe for each widget.
106     mainAxisAlignment: MainAxisAlignment.center,
107     children: <Widget>[
108       const MyTextWidget(),
109       Text(
110         '${_counter}',
111         style: Theme.of(context).textTheme.headlineMedium,
112       ), // Text
113       ], // <Widget>[]
114     ), // Column
115   ), // Center
116   floatingActionButton: FloatingActionButton(
117     onPressed: _incrementCounter,
```



Langkah 2

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

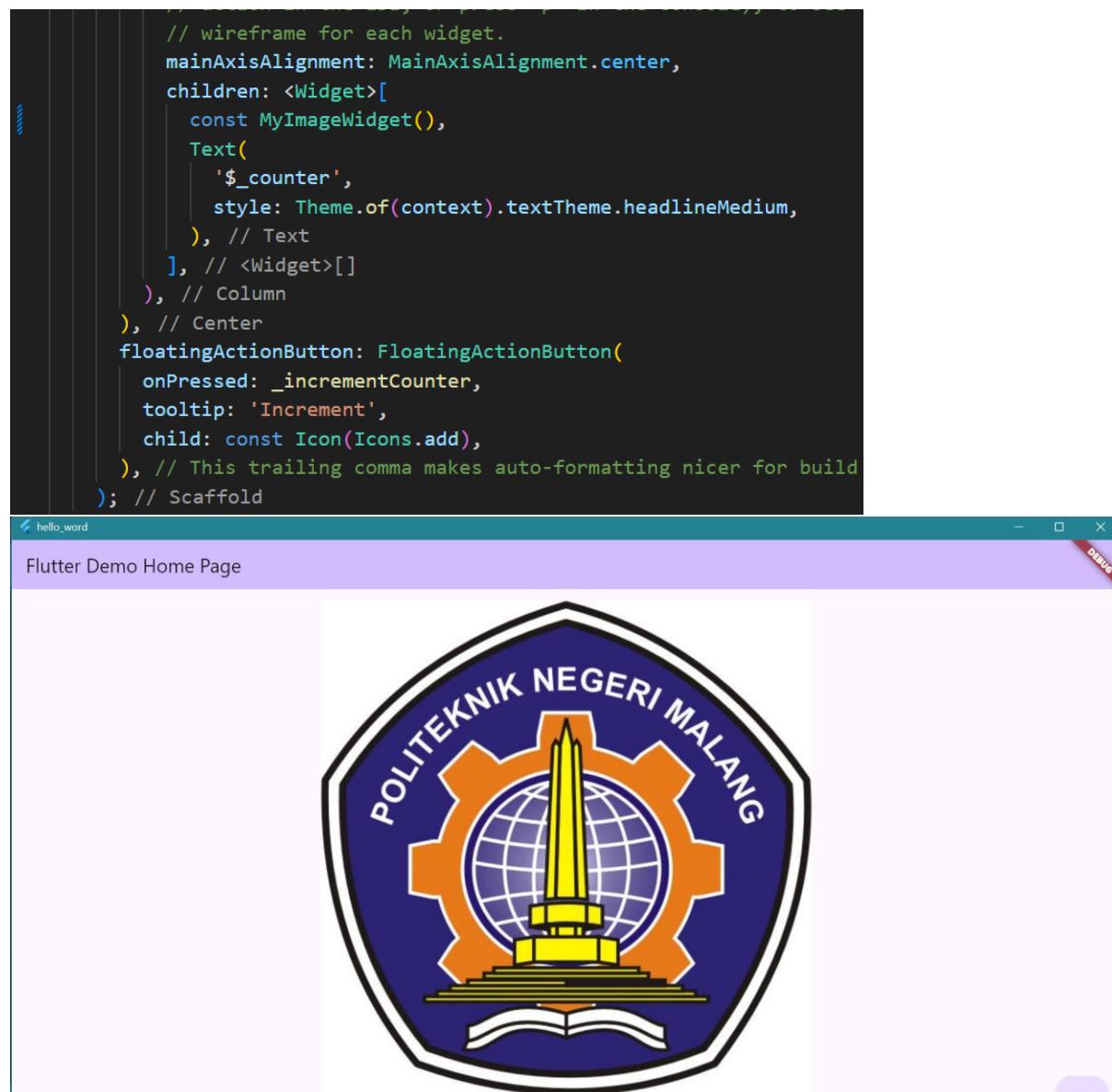
```
lib > basic_widgets > image_widget.dart > MyImageWidget
1  import 'package:flutter/material.dart';
2
3  class MyImageWidget extends StatelessWidget {
4    const MyImageWidget({Key? key}) : super(key: key);
5
6    @override
7    Widget build(BuildContext context) {
8      return const Image(
9   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

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

```
52  # The following section is specific to Flutter packages
53  flutter:
54
55      # The following line ensures that the Material Icons
56      # included with your application, so that you can
57      # use the material Icons class.
58  uses-material-design: true
59
60  assets:
61      - assets/logo_polinema.jpg
62
63  # To add assets to your application, add an assets

```

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



The screenshot shows an Android emulator window titled "Flutter Demo Home Page". The window displays the logo of Politeknik Negeri Malang, which is a blue octagonal emblem featuring a yellow gear, a globe, and a stylized building. The text "POLITEKNIK NEGERI MALANG" is written around the top of the gear. The emulator interface includes standard controls (minimize, maximize, close) and a "DEBUG" button in the top right corner.

```
// This trailing comma makes auto-formatting nicer for build
); // Scaffold
```

PRAKTIKUM 5: Menerapkan Widget Material Design dan iOS Cupertino

Langkah 1

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

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.

```
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 digunakan untuk mengatur tata letak sesuai dengan material design.

Ubah isi kode main.dart seperti berikut.

```
lib > basic_widgets > scaffold_widget.dart > _ScaffoldWidgetState > build
1   import 'package:flutter/material.dart';
2
3   class ScaffoldWidget extends StatefulWidget {
4     const ScaffoldWidget({Key? key}) : super(key: key);
5
6     @override
7     State<ScaffoldWidget> createState() => _ScaffoldWidgetState();
8   }
9
10  class _ScaffoldWidgetState extends State<ScaffoldWidget> {
11    int _counter = 0;
12
13    void _incrementCounter() {
14      setState(() {
15        _counter++;
16      });
17    }
18
19    @override
20    Widget build(BuildContext context) {
21      return MaterialApp(
22        title: 'Flutter Demo',
23        theme: ThemeData(primarySwatch: Colors.red),
24        home: Scaffold(
25          appBar: AppBar(
26            title: const Text('My Increment App'),
27          ), // AppBar
28          body: Center(
29            child: Column(
30              mainAxisAlignment: MainAxisAlignment.center,
31              children: <Widget>[
32                const Text('You have pushed the button this many times:'),
33                Text(
34                  '_$counter',
35                  style: Theme.of(context).textTheme.headlineMedium,
36                ), // Text
37                ], // <Widget>[]
38              ), // Column
39            ), // Center
40            bottomNavigationBar: BottomAppBar(
41              child: Container(height: 50.0),
42            ), // BottomAppBar
43            floatingActionButton: FloatingActionButton(
44              onPressed: _incrementCounter,
45              tooltip: 'Increment Counter',
46              child: const Icon(Icons.add),
47            ), // FloatingActionButton
48            floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
49          ), // Scaffold
50        ); // MaterialApp
51      }
52    }
```

Langkah 4

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

Ubah isi kode main.dart seperti berikut.

```
lib > basic_widgets > dialog_widget.dart > showAlertDialog
1 import 'package:flutter/material.dart';
2
3 class DialogWidget extends StatelessWidget {
4   const DialogWidget({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return const MaterialApp(
9       home: Scaffold(
10         body: MyLayout(),
11       ), // Scaffold
12     ); // MaterialApp
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   Widget okButton = TextButton(
35     child: const Text("OK"),
36     onPressed: () {
37       Navigator.pop(context);
38     },
39   ); // TextButton
40
41   AlertDialog alert = AlertDialog(
42     title: const Text("My title"),
43     content: const Text("This is my message."),
44     actions: [okButton],
45   ); // AlertDialog
46
47   showDialog(
48     context: context,
49     builder: (BuildContext context) {
50       return alert;
51     },
52   );
53 }
```

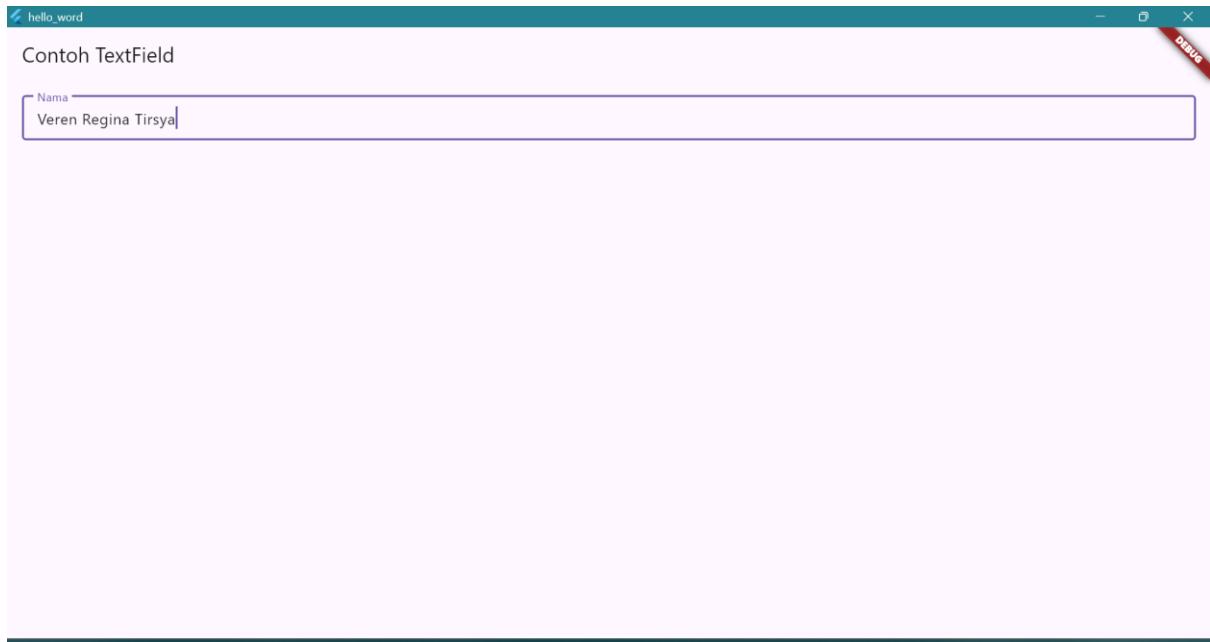
Langkah 5

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:

```
lib > basic_widgets > textfield_widget.dart > TextFieldWidget
1 import 'package:flutter/material.dart';
2
3 class TextFieldWidget extends StatelessWidget {
4   const TextFieldWidget({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return MaterialApp(
9       home: Scaffold(
10         appBar: AppBar(title: const Text("Contoh TextField")),
11         body: const Padding(
12           padding: EdgeInsets.all(16.0),
13           child: TextField(
14             obscureText: false,
15             decoration: InputDecoration(
16               border: OutlineInputBorder(),
17               labelText: 'Nama',
18             ), // InputDecoration
19             ), // TextField
20             ), // Padding
21           ), // Scaffold
22         ); // MaterialApp
23     }
24 }
```





Langkah 6

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';

class DatePickerWidget extends StatelessWidget {
  const DatePickerWidget({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> {
```

```
DateTime selectedDate = DateTime.now();

Future<void> _selectDate(BuildContext context) async {
    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),
                        child: const Text('Pilih Tanggal'),
                    ),
                ],
            ),
        );
}
}
```

Tugas Praktikum

1. Selesaikan Praktikum 1 sampai 5, lalu dokumentasikan dan push ke repository Anda berupa screenshot setiap hasil pekerjaan beserta penjelasannya di file README.md!
2. Selesaikan Praktikum 2 dan Anda wajib menjalankan aplikasi hello_world pada perangkat fisik (device Android/iOS) agar Anda mempunyai pengalaman untuk menghubungkan ke perangkat fisik. Capture hasil aplikasi di perangkat, lalu buatlah laporan praktikum pada file README.md.
3. Pada praktikum 5 mulai dari Langkah 3 sampai 6, buatlah file widget tersendiri di folder basic_widgets, kemudian pada file main.dart cukup melakukan import widget sesuai masing-masing langkah tersebut!
4. Selesaikan Codelabs: Your first Flutter app, lalu buatlah laporan praktikumnya dan push ke repository GitHub Anda!
5. README.md berisi: capture hasil akhir tiap praktikum (side-by-side, bisa juga berupa file GIF agar terlihat proses perubahan ketika ada aksi dari pengguna) dengan menampilkan NIM dan Nama Anda sebagai ciri pekerjaan Anda.
6. Kumpulkan berupa link repository/commit GitHub Anda kepada dosen yang telah disepakati!

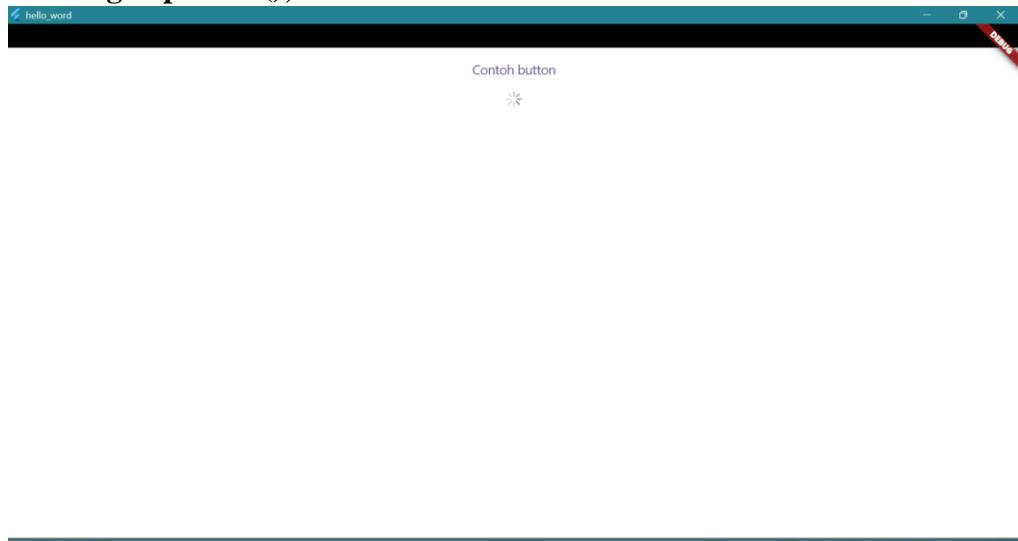
Jawab:

1. Melakukan import widget di main.dart pada praktikum 5

```
lib > main.dart > MyHomePage > MyHomePage
  1 import 'package:flutter/material.dart';
  2 import 'package:hello_word/basic_widgets/image_widget.dart';
  3 import 'package:hello_word/basic_widgets/text_widget.dart';
  4
  5 // mengimport semua widget yang sudah kamu buat
  6 //-----
  7 import 'package:hello_word/basic_widgets/loading_cupertino.dart';
  8 import 'package:hello_word/basic_widgets/fab_widget.dart';
  9 import 'package:hello_word/basic_widgets/scaffold_widget.dart';
 10 import 'package:hello_word/basic_widgets/dialog_widget.dart';
 11 import 'package:hello_word/basic_widgets/textfield_widget.dart';
 12 import 'package:hello_word/basic_widgets/datepicker_widget.dart';
 13
 14     // home: const LoadingCupertino()
 15     // home: const FabWidget(),
 16     // home: const ScaffoldWidget(),
 17     // home: const DialogWidget(),
 18     // home: const TextFieldWidget(),
 19     // home: const DatepickerWidget(),
 20     home: const DatepickerWidget(),
 21 ); // MaterialApp
```

2. Menjalankan widget sesuai langkah

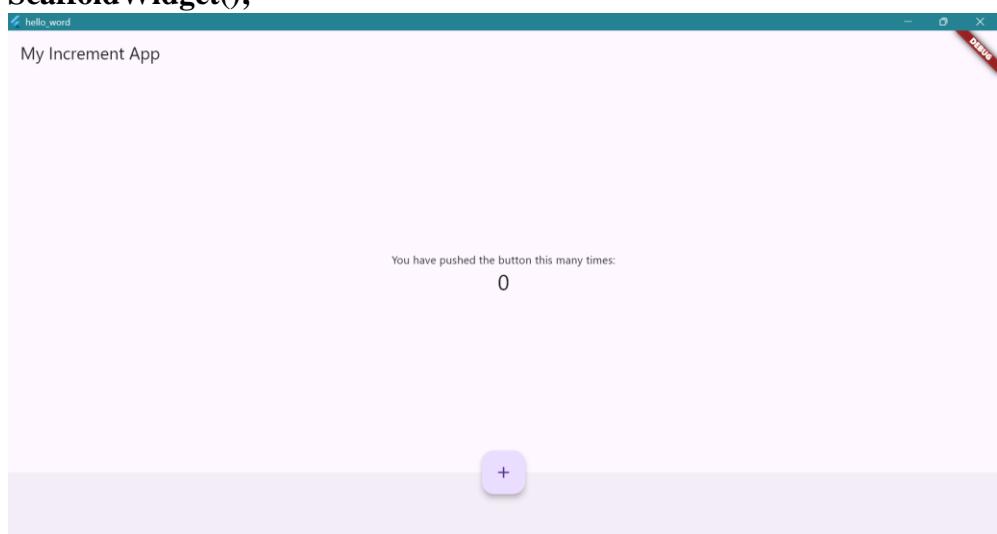
- **LoadingCupertino();**



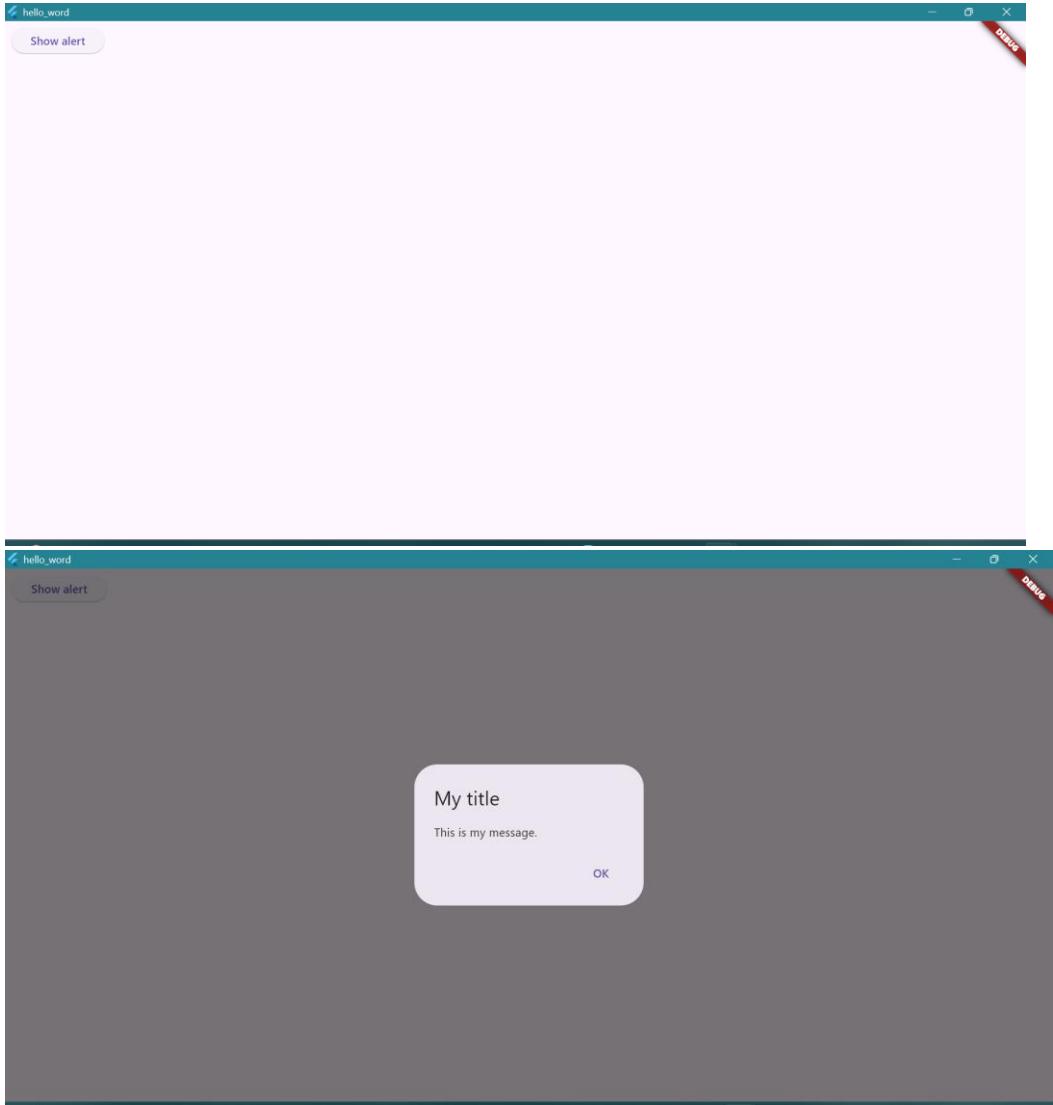
- **FabWidget();**



- **ScaffoldWidget();**

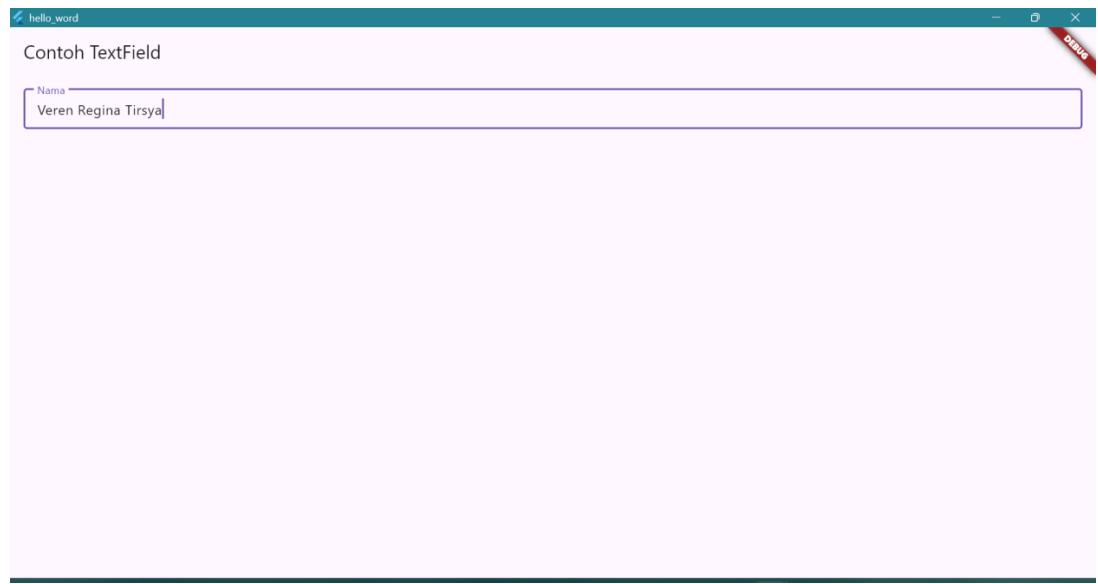


- **DialogWidget();**



- **TextFieldWidget();**





- **DatePickerWidget;**

