

LAPORAN

PRAKTIKUM PEMROGRAMAN MOBILE

Dosen Pengampu:
Bapak Ade Ismail S. Kom, M.Ti



Disusun oleh:
Siti Fa'iqoh (2241760026)
SIB 3D / 25

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

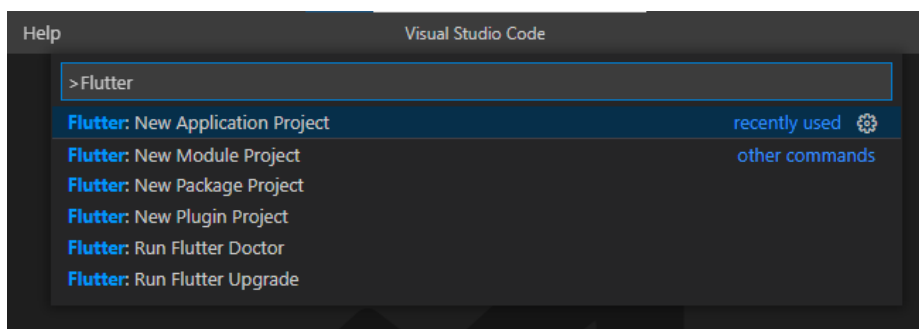
6. Praktikum 1: Membuat Project Flutter Baru

Selesaikan langkah-langkah praktikum berikut ini menggunakan editor Visual Studio Code (VS Code) atau Android Studio atau code editor lain kesukaan Anda.

Perhatian: Diasumsikan Anda telah berhasil melakukan setup environment Flutter SDK, VS Code, Flutter Plugin, dan Android/iOS SDK pada [Codelabs#1](#).

Langkah 1:

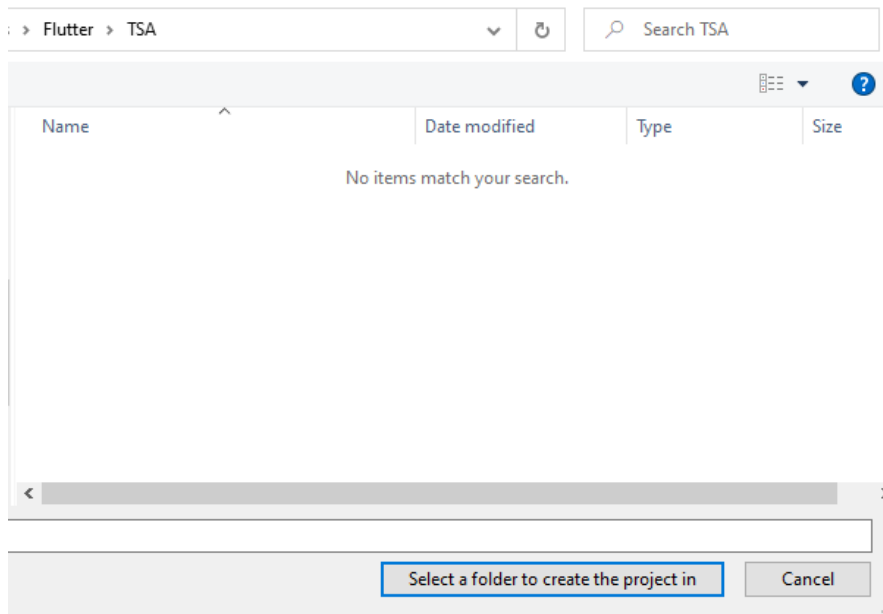
Buka VS Code, lalu tekan tombol **Ctrl + Shift + P** maka akan tampil *Command Palette*, lalu ketik **Flutter**. Pilih **New Application Project**.



Catatan: Anda dapat mengakses **Command Palette** dengan cara lain, yaitu pilih menu **View** > **Command Palette**. Tombol shortcut **Ctrl + Shift + P** mungkin hanya berlaku di Windows.

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**.

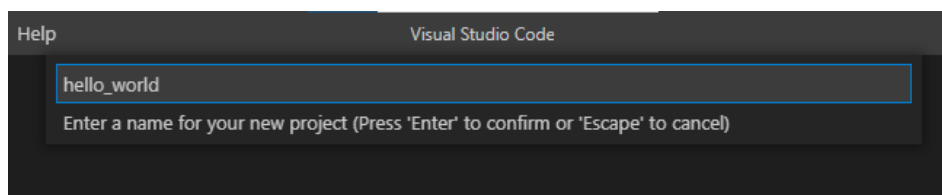


Setelah memilih maka tampilan di vscode

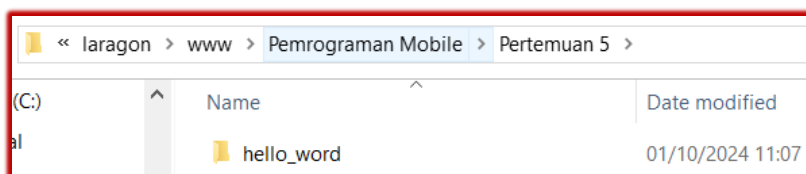
> Pertemuan 5

Langkah 3:

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

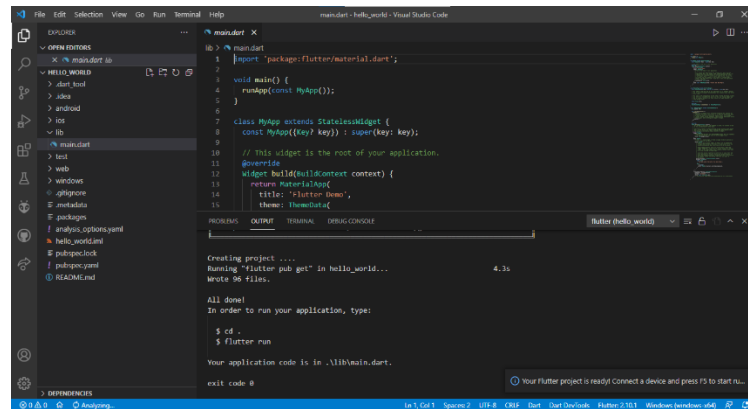


Perhatian: Nama project ini harus *lowercase* (huruf kecil semua) tanpa menggunakan spasi. Untuk memisahkan kata, bisa menggunakan *underline* (garis bawah). Nama project tidak dapat diawali dengan angka atau karakter khusus lain. Nama project ini bukan nama aplikasi yang akan tampil di Play Store atau App Store. Untuk nama aplikasi, nanti dapat diatur ketika melakukan *deployment*.



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.



```
lib > main.dart > ...
1  import 'package:flutter/material.dart';
2
3  void main() {
4    runApp(const MyApp());
5  }
6
7  class MyApp extends StatelessWidget {
8    const MyApp({super.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         // This is the theme of your application.
17       ),
18     );
19   }
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS flutter (helli

```
[hello_word] flutter create --template app --overwrite .
Waiting for another flutter command to release the startup lock...
Creating project ....
Resolving dependencies...
Downloading packages...
Got dependencies.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev

In order to run your application, type:

$ cd .
$ flutter run

Your application code is in .\lib\main.dart.

exit code 0
```

7. Praktikum 2: Menghubungkan Perangkat Android atau Emulator

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>

saya menggunakan flutter chrome

8. Praktikum 3: Membuat Repository GitHub dan Laporan Praktikum

Melanjutkan dari praktikum 2, silakan selesaikan langkah-langkah berikut ini.

Perhatian: Diasumsikan Anda telah mempunyai akun GitHub dan Anda telah memahami konsep dasar dalam bekerja menggunakan Git pada pertemuan pertama.

Praktikum ini dapat Anda **lewati langsung ke langkah 11** jika sudah paham cara membuat laporan praktikum sesuai *style* yang Anda pilih.

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:

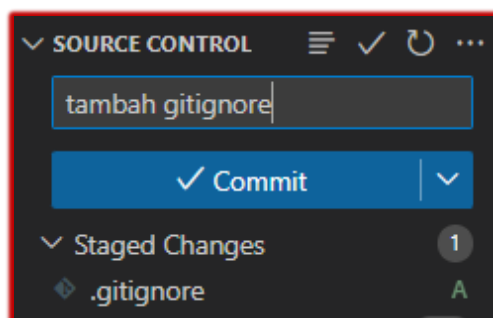
Kembali ke VS code, project flutter hello_world, buka terminal pada menu **Terminal > New Terminal**. Lalu ketik perintah berikut untuk inisialisasi git pada project Anda.

```
git init
```

Langkah 4:

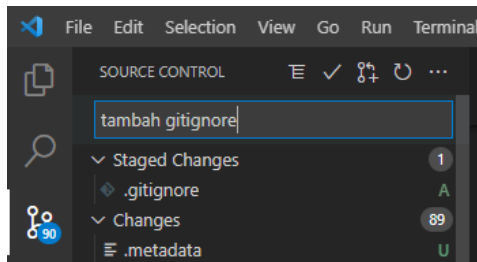
Pilih menu **Source Control** di bagian kiri, lalu lakukan **stages (+)** pada file **.gitignore** untuk mengunggah file pertama ke repository GitHub.

Saya sudah mempunyai repository sebelumnya, jadi saya tinggal commit push



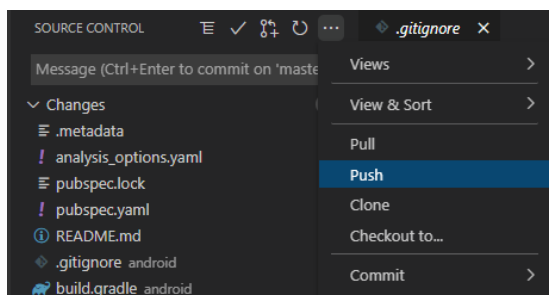
Langkah 5:

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

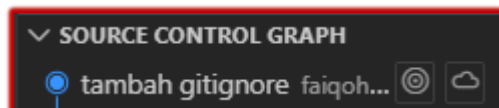


Langkah 6:

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

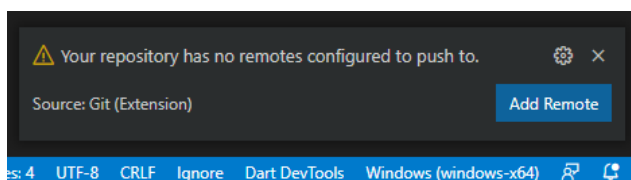


Setelah dipush



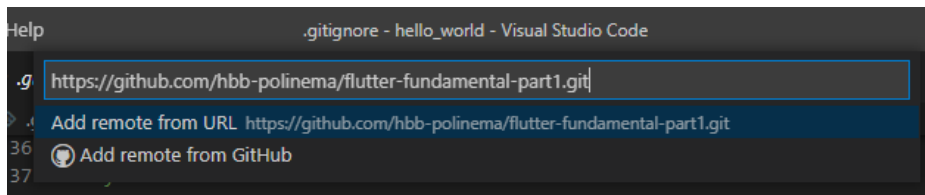
Langkah 7:

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

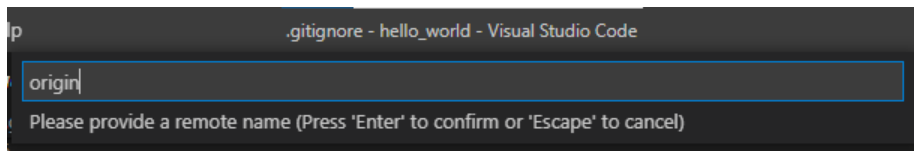


Langkah 8:

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

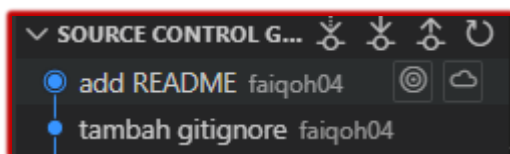
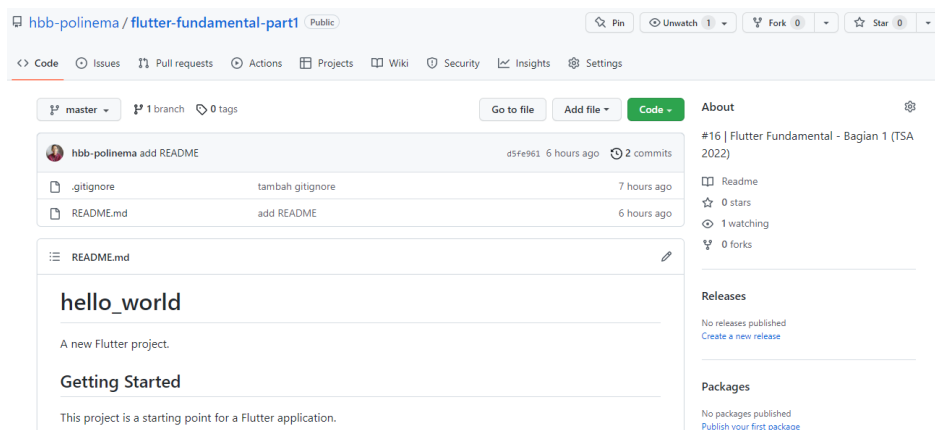


Setelah berhasil, tulis remote name dengan "origin"



Langkah 9:

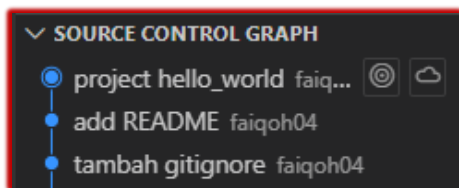
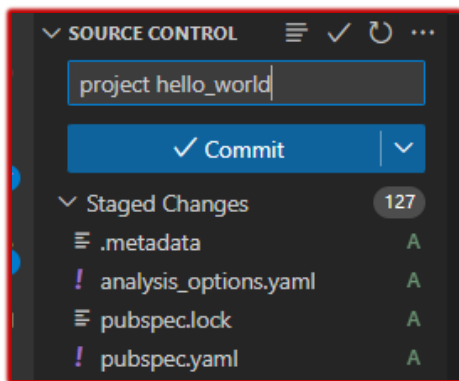
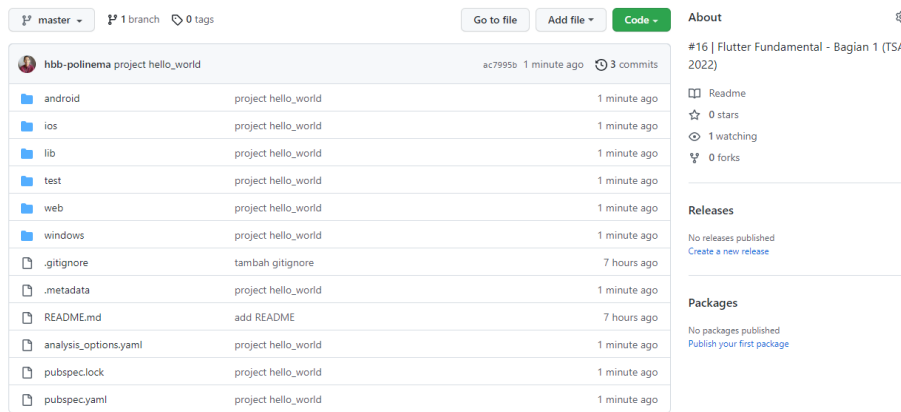
Lakukan hal yang sama pada file **README.md** mulai dari Langkah 4. Setelah berhasil melakukan push, masukkan username GitHub Anda dan password berupa token yang telah dibuat (pengganti password konvensional ketika Anda login di browser GitHub). Reload halaman repository GitHub Anda, maka akan tampil hasil push kedua file tersebut seperti gambar berikut.



Perhatian: Personal access token GitHub dapat Anda buat melalui <https://github.com/settings/tokens/new> atau dapat membaca dokumentasi [GitHub Token](#).

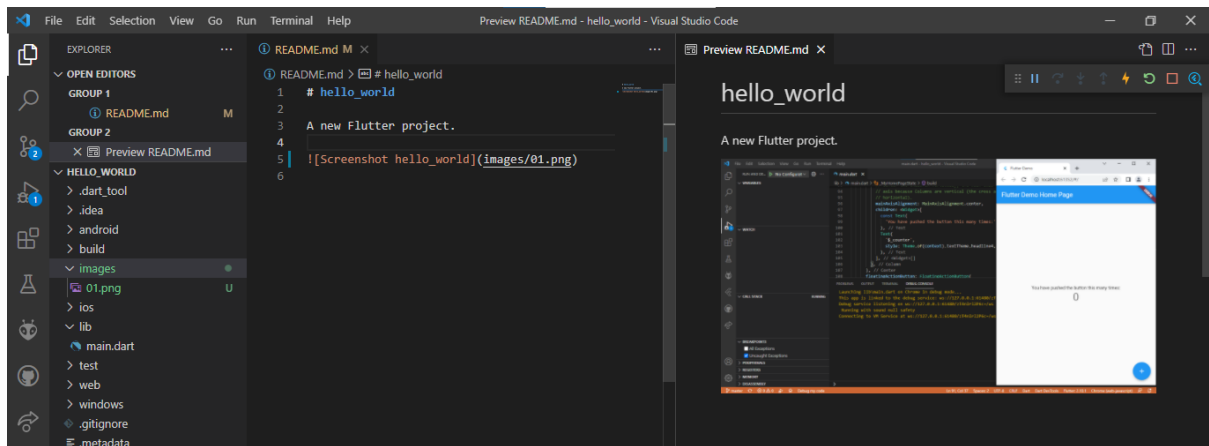
Langkah 10:

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.



Perhatian: Lakukan proses screenshot seperti pada Langkah 12 untuk setiap Laporan Praktikum yang Anda akan buat pada praktikum selanjutnya hingga pertemuan project final.

9. Praktikum 4: Menerapkan Widget Dasar

Selesaikan langkah-langkah praktikum berikut ini dengan melanjutkan dari praktikum sebelumnya.

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';

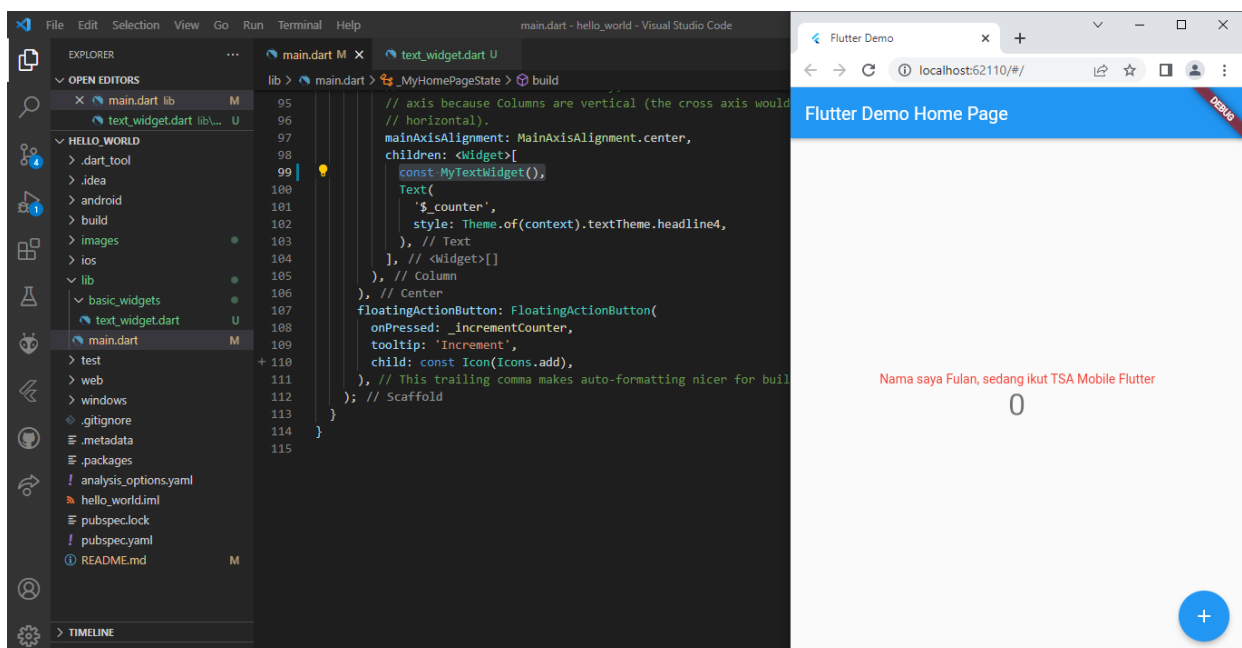
class MyTextWidget extends StatelessWidget {
  const MyTextWidget({Key? key}) : super(key: key);

  @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.

```
Pertemuan 5 > hello_word > lib > text_widget.dart > ...
1 import 'package:flutter/material.dart';
2
3 class MyTextWidget extends StatelessWidget {
4   const MyTextWidget({Key? key}) : super(key: key); Convert 'key' to a
5
6   @override
7   Widget build(BuildContext context) {
8     return const Text(
9       "Nama saya Siti Faiqoh, sedang belajar Pemrograman Mobile",
10      style: TextStyle(color: Colors.red, fontSize: 14),
11      textAlign: TextAlign.center); // Text
12   }
13 }
```

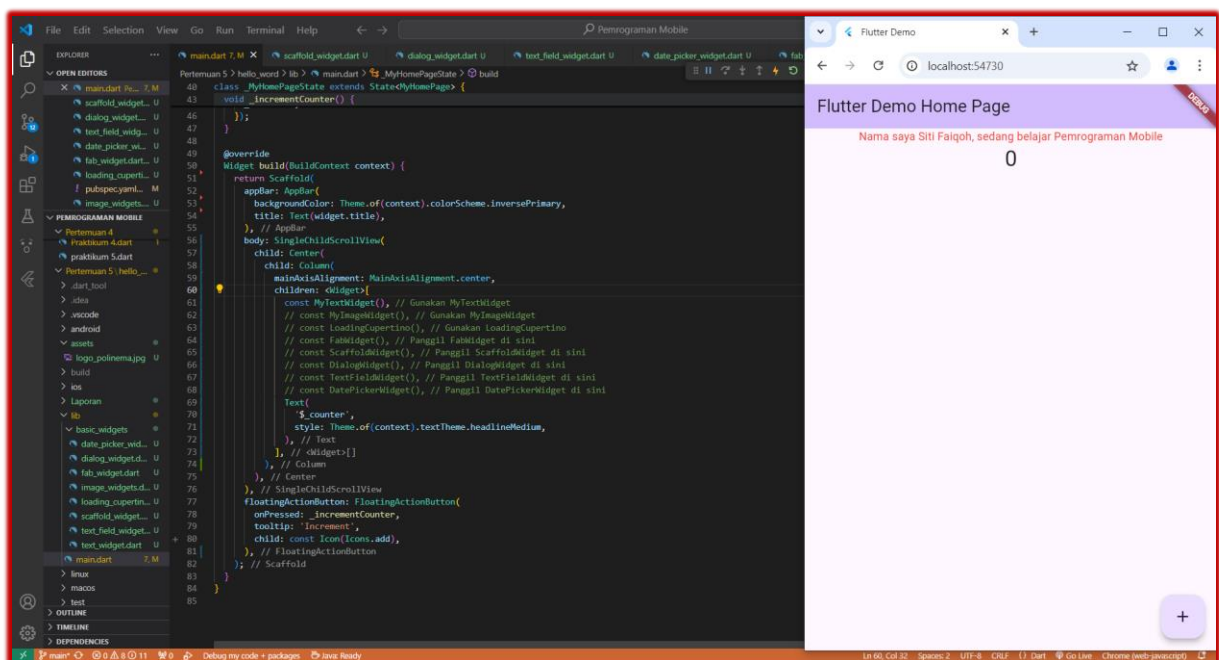
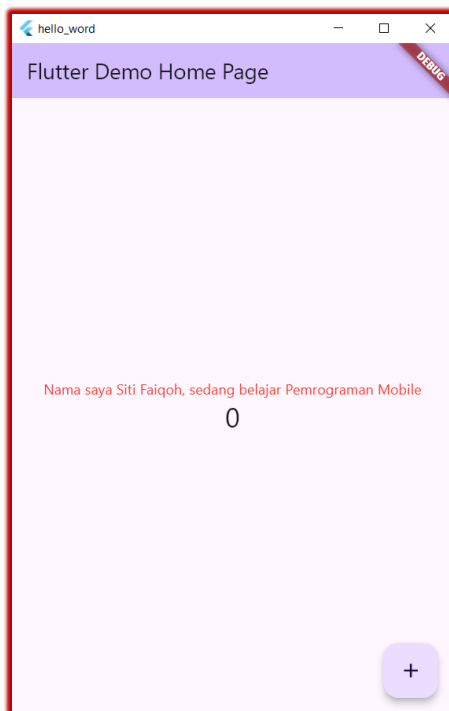
Lakukan import file `text_widget.dart` ke `main.dart`, lalu ganti bagian text widget dengan kode di atas. Maka hasilnya seperti gambar berikut. Screenshot hasil milik Anda, lalu dibuat laporan pada file `README.md`.



Pertemuan 5 > hello_word > lib > main.dart > MyApp > build

```
1 import 'package:flutter/material.dart';
2 import 'text_widget.dart'; // Mengimpor file text_widget.dart
3
4 Run | Debug | Profile
5 void main() {
6   runApp(const MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   const MyApp({super.key});
11
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: 'Flutter Demo',
16       theme: ThemeData(
17         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
18         useMaterial3: true,
19       ), // ThemeData
20       home: const MyHomePage(title: 'Flutter Demo Home Page'),
21     ); // MaterialApp
22   }
23
24   class MyHomePage extends StatefulWidget {
25     const MyHomePage({super.key, required this.title});
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
42     @override
43     Widget build(BuildContext context) {
44       return Scaffold(
45         appBar: AppBar(
46           backgroundColor: Theme.of(context).colorScheme.inversePrimary,
47           title: Text(widget.title),
48         ), // AppBar
49         body: Center(
50           child: Column(
51             mainAxisAlignment: MainAxisAlignment.center,
52             children: <Widget>[
53               const MyTextWidget(), // Mengganti Text widget dengan MyTextWidget
54               Text(
55                 '$_counter',
56                 style: Theme.of(context).textTheme.headlineMedium,
57               ), // Text
58             ], // <Widget>[]
59           ), // Column
60         ), // Center
61         floatingActionButton: FloatingActionButton(
62           onPressed: _incrementCounter,
63           tooltip: 'Increment',
64           child: const Icon(Icons.add),
65         ), // FloatingActionButton
66       ); // Scaffold
67     }
68   }
```



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")
  );
}
}

```

```

Pertemuan 5 > hello_world > lib > basic_widgets > image_widgets.dart > ...
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(image: AssetImage("logo_polinema.jpg"));
9    }
10 }

```

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

```

flutter:
  assets:
    - logo_polinema.jpg

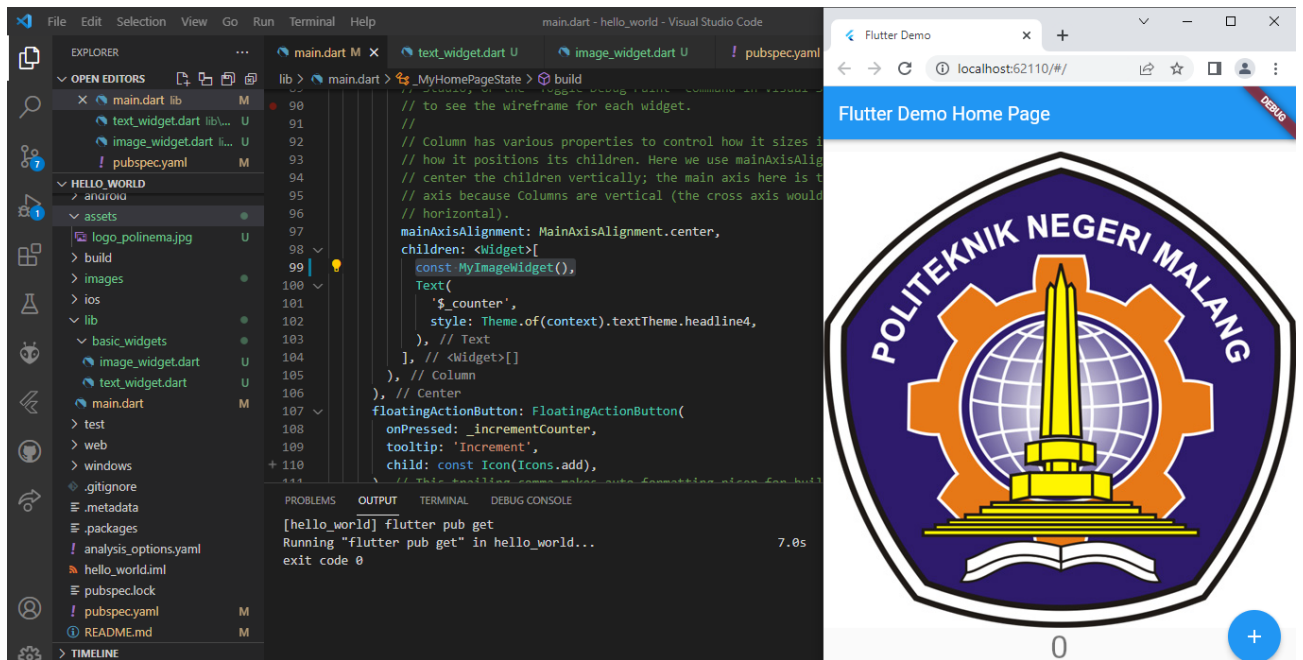
```

```

Pertemuan 5 > hello_world > ! pubspec.yaml
54  flutter:
59    uses-material-design: true
60    assets:
61      - assets/logo_polinema.jpg
62    # To add assets to your application, add an assets section, like this:
63    # assets:

```

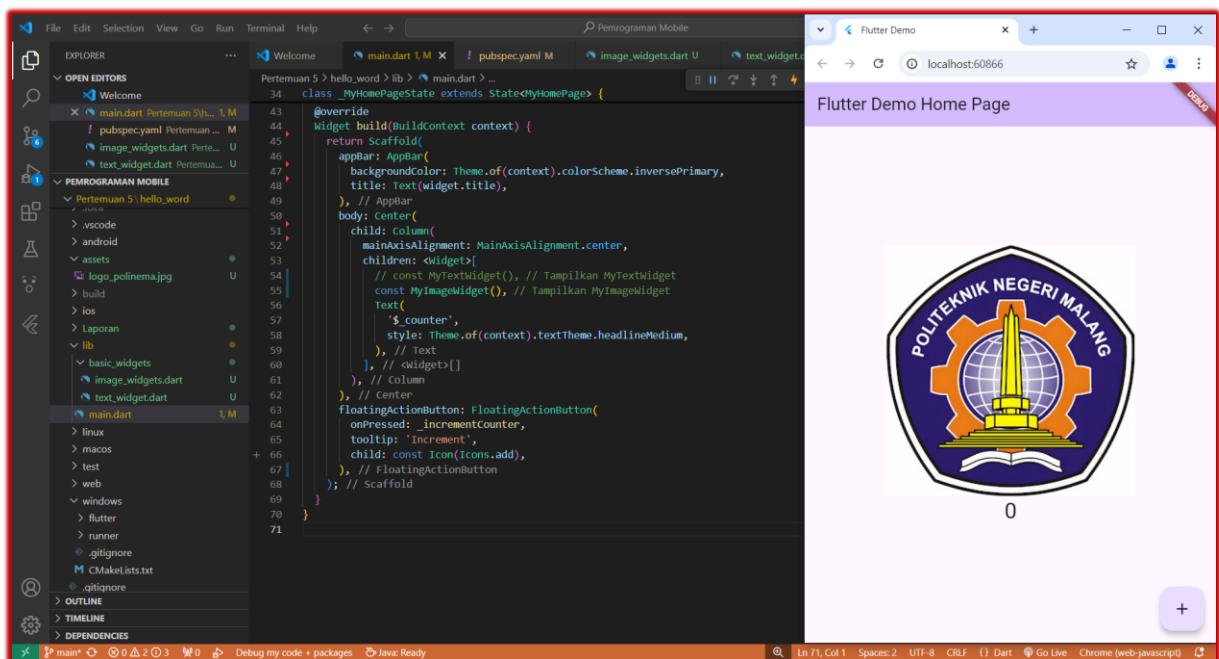
Jangan lupa sesuaikan kode dan import di file `main.dart` kemudian akan tampil gambar seperti berikut.



```

Pertemuan 5 > hello_word > lib > main.dart > ...
1 import 'package:flutter/material.dart';
2 import 'basic_widgets/text_widget.dart'; // Mengimpor MyTextWidget
3 import 'basic_widgets/image_widgets.dart'; // Mengimpor MyImageWidget
4

```



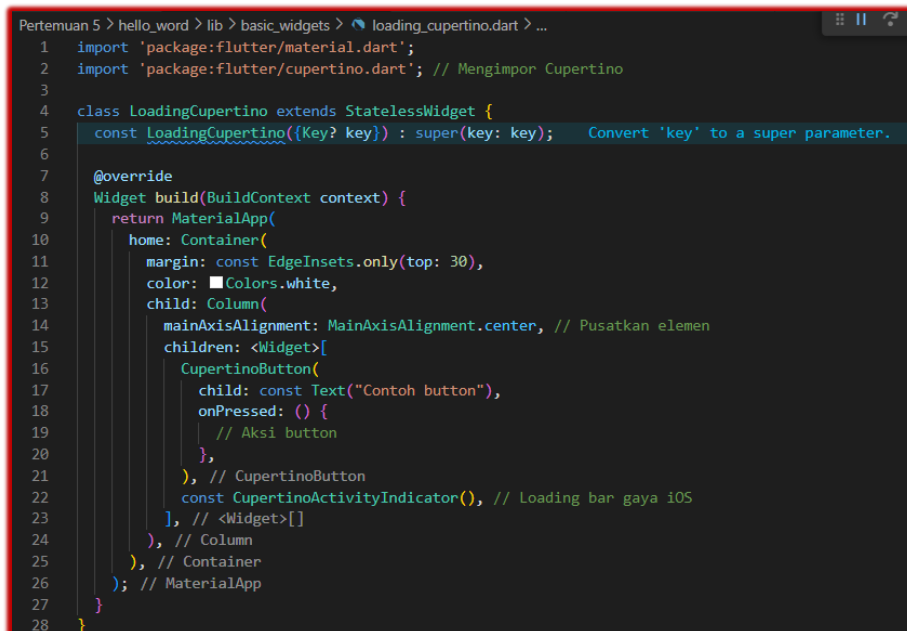
10. Praktikum 5: Menerapkan Widget Material Design dan iOS Cupertino

Selesaikan langkah-langkah praktikum berikut ini dengan melanjutkan project `hello_world` Anda. Lakukan langkah yang sama seperti pada Praktikum 3, yaitu setiap widget dibuat file sendiri lalu import ke `main.dart` dan screenshot hasilnya.

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(),  
      ],  
    ),  
  ),  
);
```

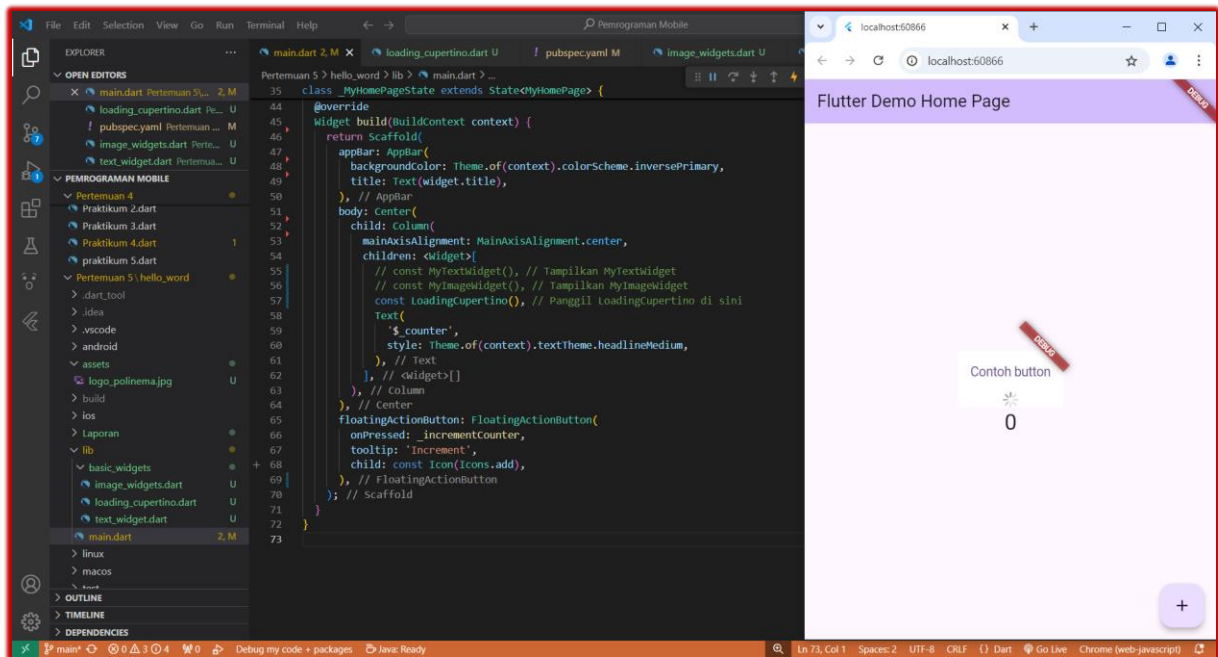


The screenshot shows a code editor with the following Dart code:

```
Pertemuan 5 > hello_world > lib > basic_widgets > loading_cupertino.dart > ...  
1 import 'package:flutter/material.dart';  
2 import 'package:flutter/cupertino.dart'; // Mengimpor Cupertino  
3  
4 class LoadingCupertino extends StatelessWidget {  
5   const LoadingCupertino({Key? key}) : super(key: key); // Convert 'key' to a super parameter.  
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          mainAxisAlignment: MainAxisAlignment.center, // Pusatkan elemen  
15          children: <Widget>[  
16            CupertinoButton(  
17              child: const Text("Contoh button"),  
18              onPressed: () {  
19                // Aksi button  
20              },  
21            ), // CupertinoButton  
22            const CupertinoActivityIndicator(), // Loading bar gaya iOS  
23          ], // <Widget>[]  
24        ), // Column  
25      ), // Container  
26    ); // MaterialApp  
27  }  
28 }
```



```
Pertemuan 5 > hello_word > lib > main.dart > _MyHomePageState > build
1 import 'package:flutter/material.dart';
2 import 'basic_widgets/text_widget.dart'; // Mengimpor MyTextWidget Unused
3 import 'basic_widgets/image_widgets.dart'; // Mengimpor MyImageWidget Unu
4 import 'basic_widgets/loading_cupertino.dart'; // Impor LoadingCupertino
5
```



Langkah 2: Floating Action Button (FAB)

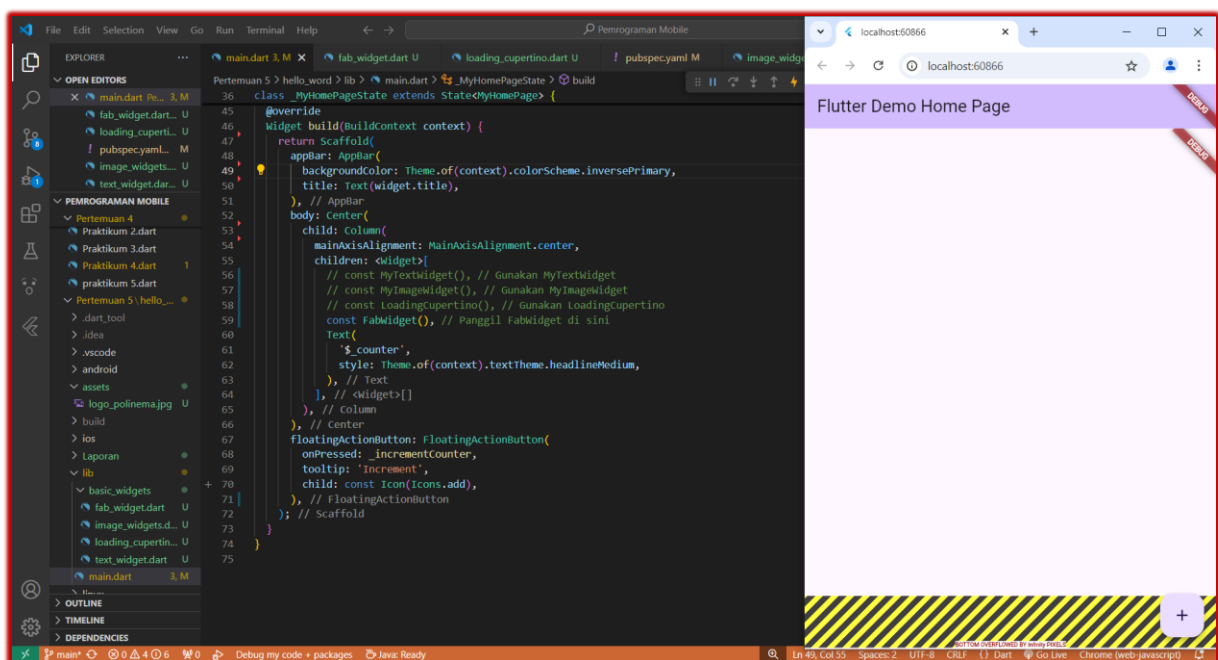
Button widget terdapat beberapa macam pada flutter yaitu ButtonBar, DropdownButton, TextButton, FloatingActionButton, IconButton, OutlineButton, PopupMenuButton, dan ElevatedButton.

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,
    ),
  ),
);
```

```
Pertemuan 5 > hello_word > lib > basic_widgets > fab_widget.dart > ...
1 import 'package:flutter/material.dart';
2
3 class FabWidget extends StatelessWidget {
4   const FabWidget({Key? key}) : super(key: key); Convert 'key' to a super
5
6   @override
7   Widget build(BuildContext context) {
8     return MaterialApp(
9       home: Scaffold(
10        floatingActionButton: FloatingActionButton(
11          onPressed: () {
12            // Tambahkan aksi saat tombol ditekan di sini
13          },
14          child: const Icon(Icons.thumb_up), The 'child' argument should
15          backgroundColor: Colors.pink, // Warna tombol
16        ), // FloatingActionButton
17      ), // Scaffold
18    ); // MaterialApp
19  }
20 }
```

```
Pertemuan 5 > hello_word > lib > main.dart > MyApp > MyApp
1 import 'package:flutter/material.dart';
2 import 'basic_widgets/text_widget.dart'; // Mengimpor MyTextWidget Unused im
3 import 'basic_widgets/image_widgets.dart'; // Mengimpor MyImageWidget Unused
4 import 'basic_widgets/loading_cupertino.dart'; // Impor LoadingCupertino Unu
5 import 'basic_widgets/fab_widget.dart'; // Impor FabWidget
6
```



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++;
    });
  }

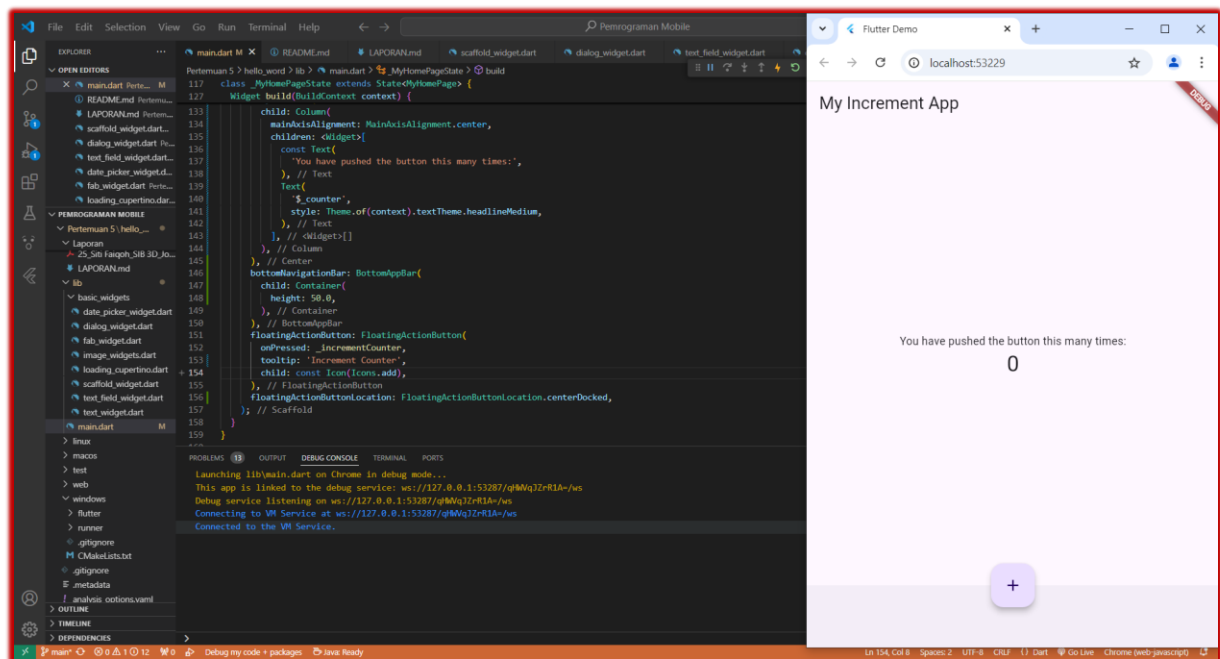
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[

```

```

        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,
);
}
}

```



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);
    },
  );

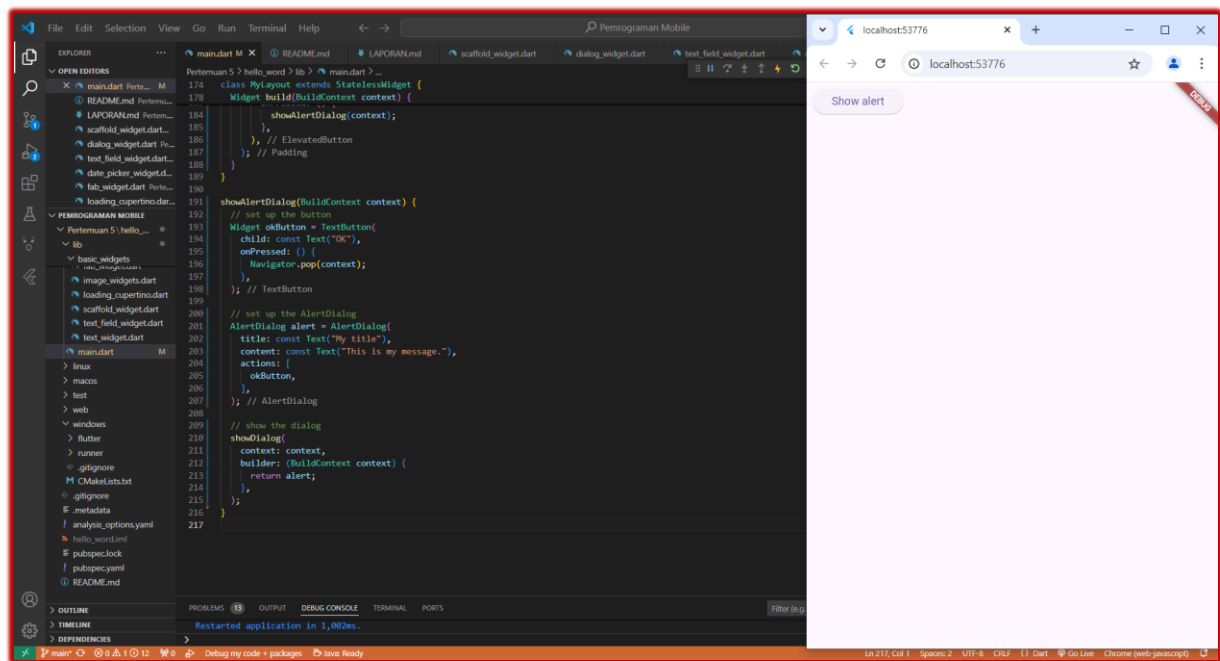
  // set up the AlertDialog
  AlertDialog alert = AlertDialog(
    title: const Text("My title"),
    content: const Text("This is my message."),
    actions: [
```

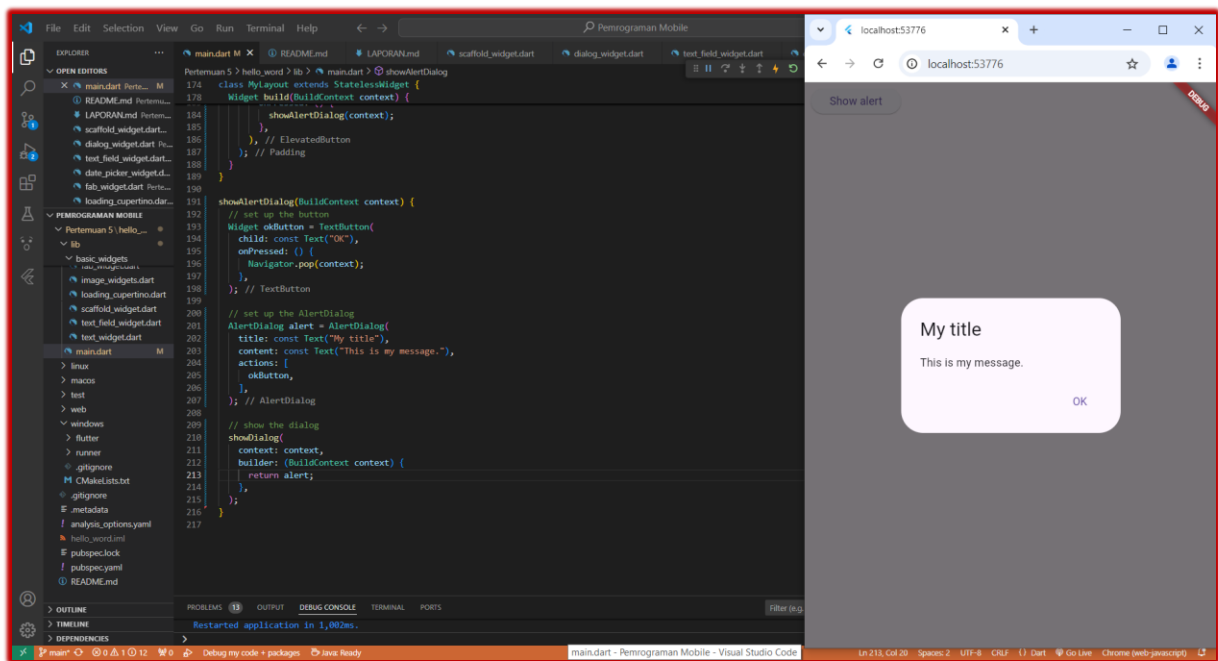
```

        okButton,
      ],
    );

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

```





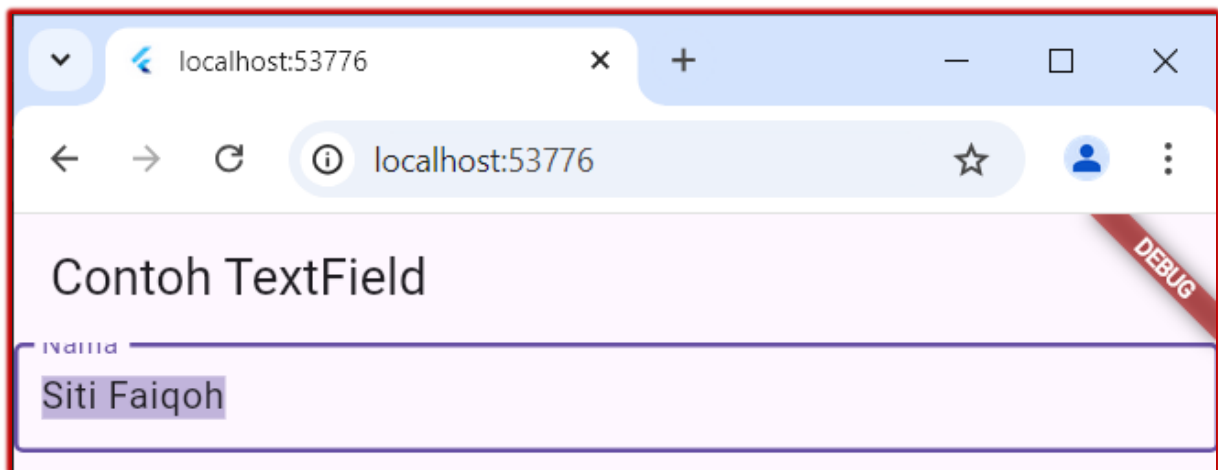
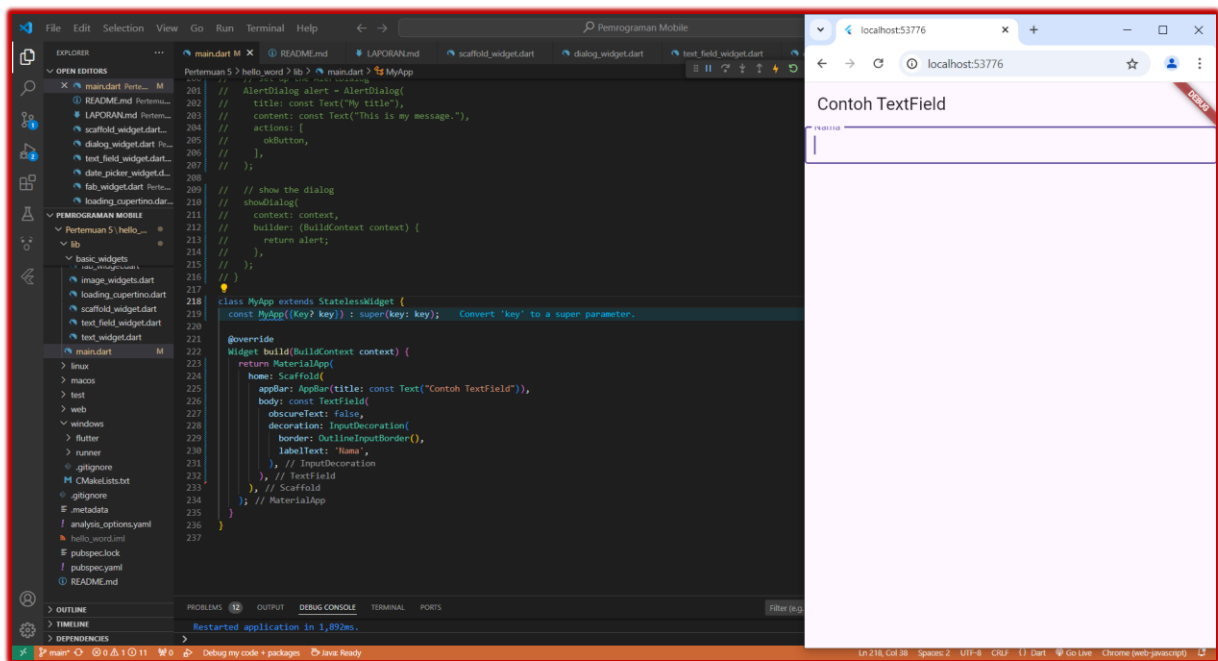
Langkah 5: Input dan Selection Widget

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',
          ),
        ),
      ),
    );
  }
}
```



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 Final 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;
      });
    }
  }

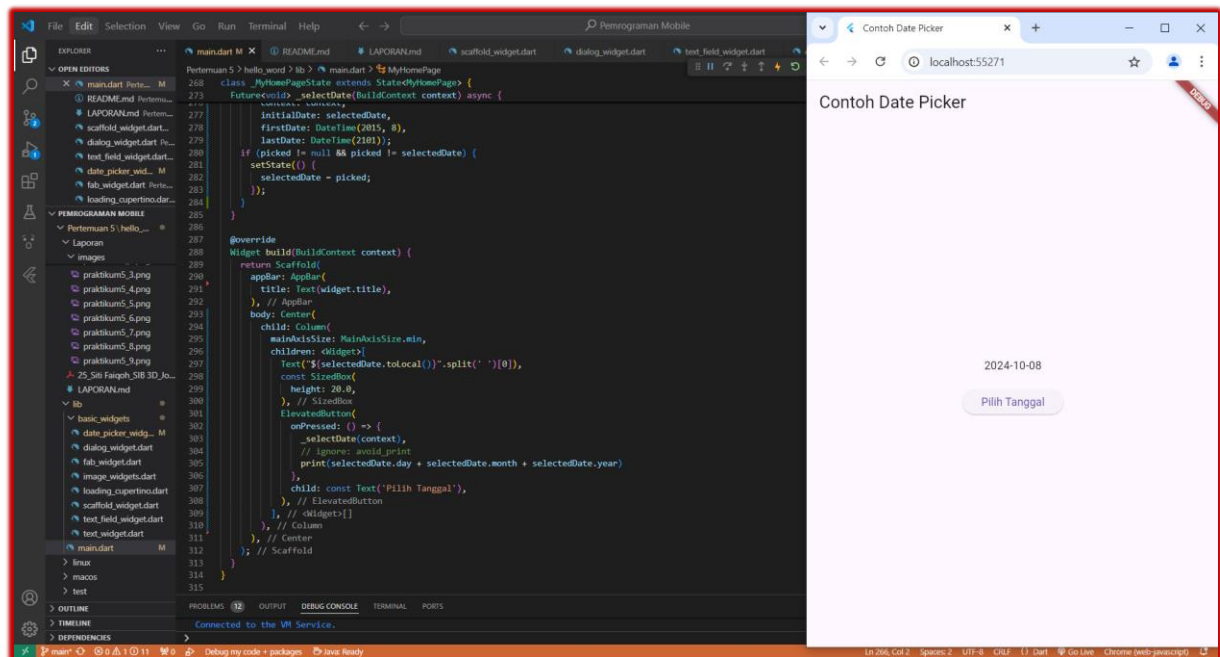
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: <Widget>[
            Text("${selectedDate.toLocal()}".split('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

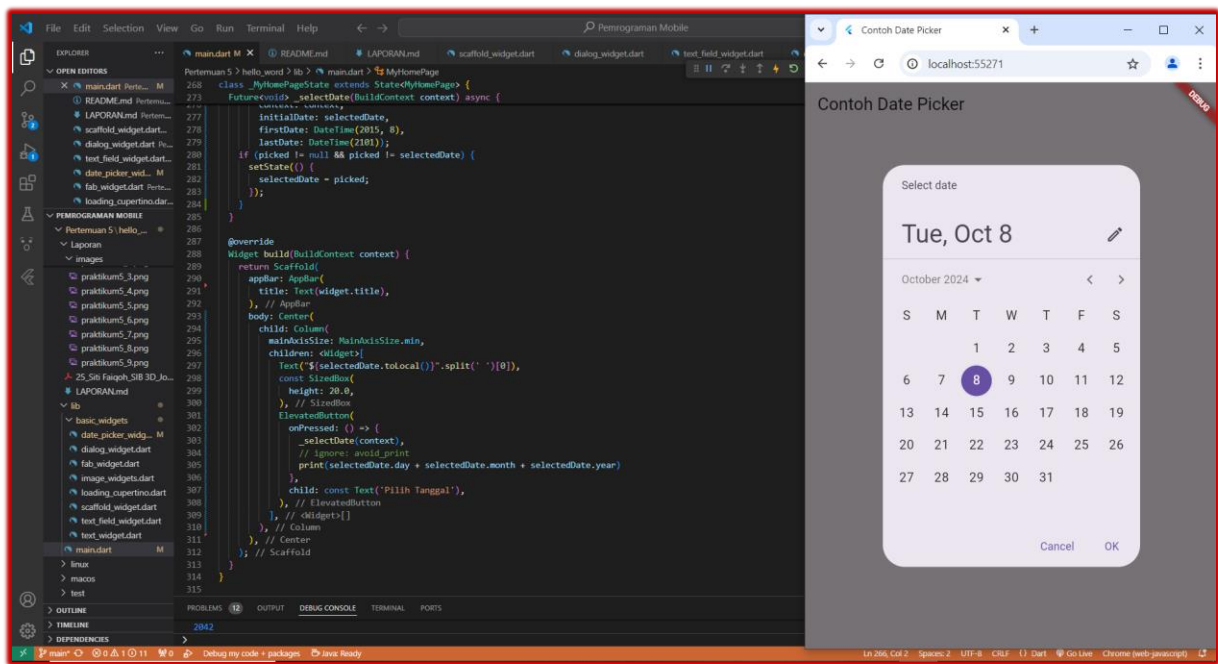
```

```

        height: 20.0,
      ),
      ElevatedButton(
        onPressed: () => {
          _selectDate(context),
          // ignore: avoid_print
          print(selectedDate.day + selectedDate.month +
selectedDate.year)
        },
        child: const Text('Pilih Tanggal'),
      ),
    ],
  ),
),
);
}
}

```





11. 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!

12. Selamat!

Selamat Anda telah menyelesaikan Codelab ini. Anda telah mempelajari terkait Flutter Fundamental untuk dasar-dasar widget.

Pada codelab berikutnya, Anda akan mempelajari tentang Flutter Fundamental Bagian 2 terkait layout, position, navigasi dan route.