

## Jobsheet-5: Aplikasi Pertama dan Widget Dasar Flutter



NIM : 2341760017  
Nama : Marga Reta Novia Putri  
Kelas : SIB-3D / 10

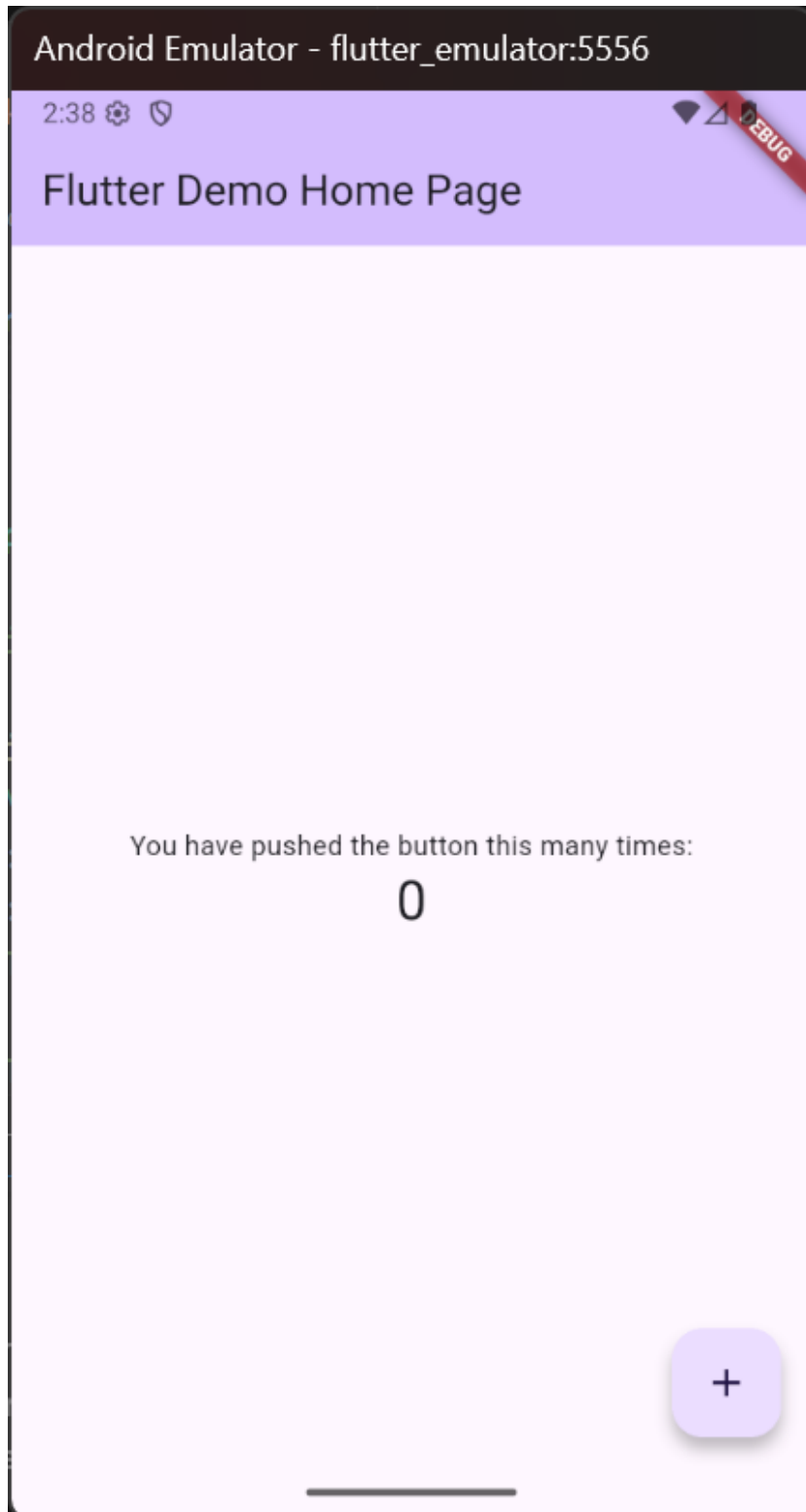
### Praktikum 1 : Membuat Project Flutter Baru

- **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

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.

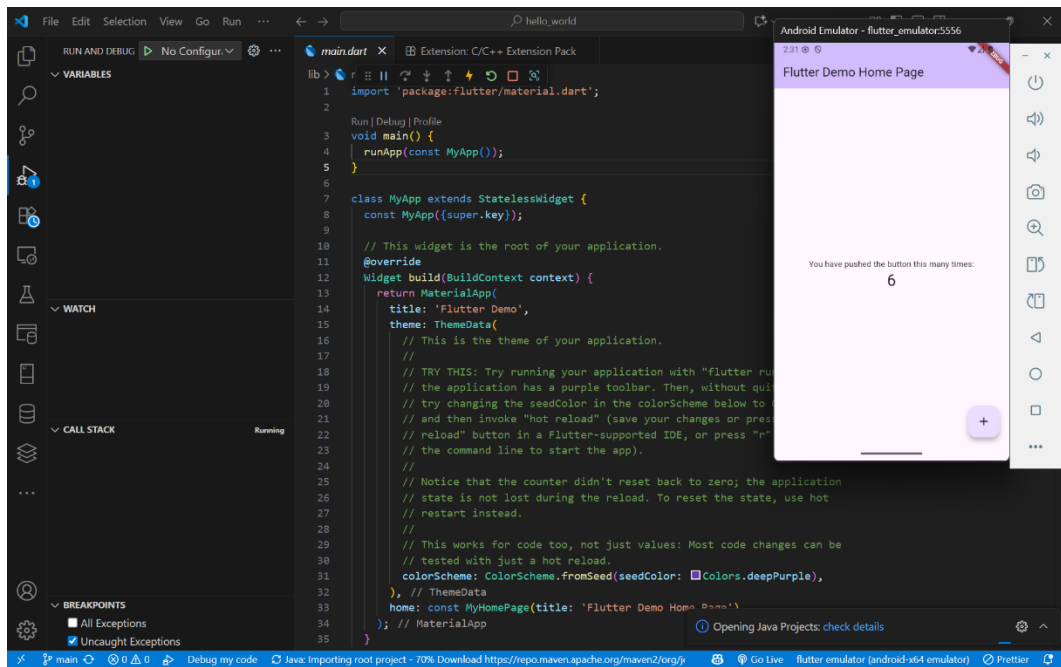
<https://developer.android.com/codelabs/basic-android-kotlin-compose-connect-device?hl=id#0>



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

#### - 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.



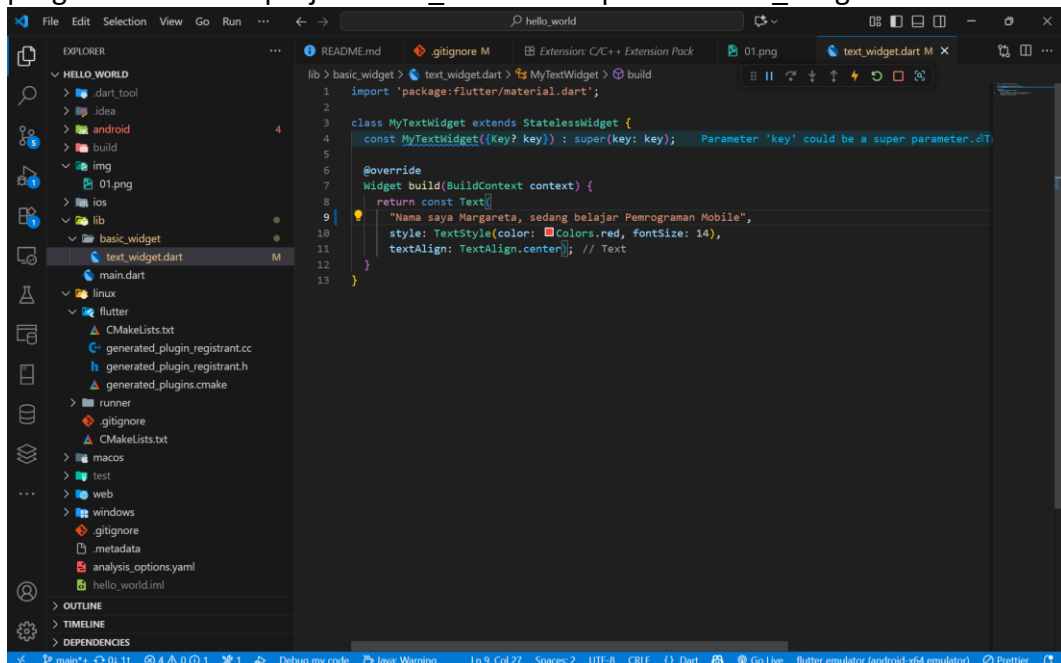
#### - Langkah 12:

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.

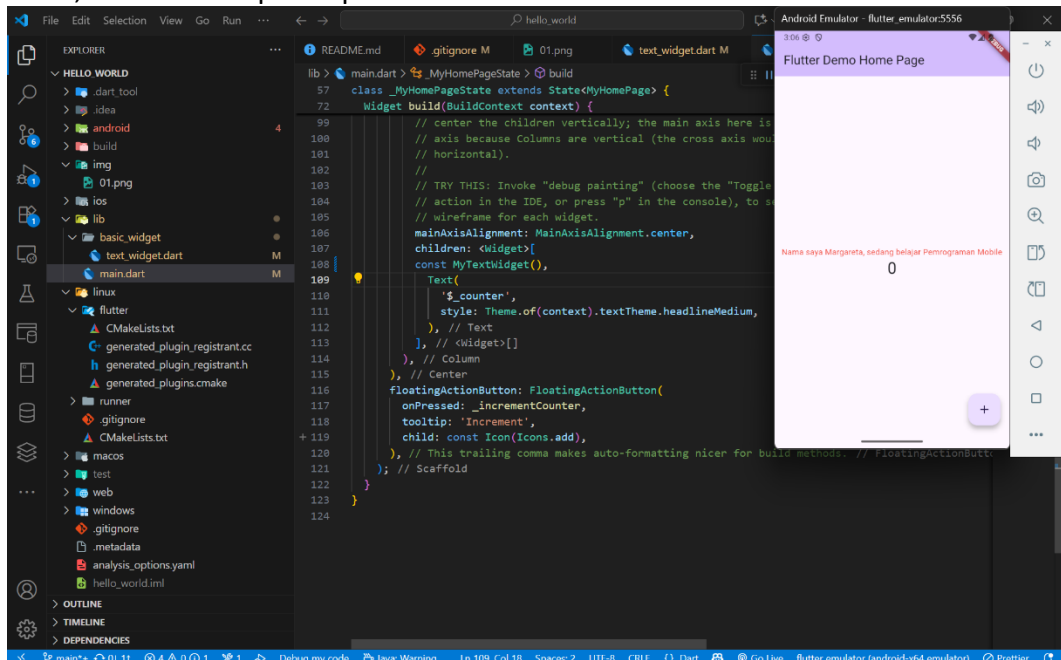
## Praktikum 4: Menerapkan Widget Dasar

### - Langkah 1: 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**.



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**.



### - Langkah 2: Image Widget

The screenshot shows an IDE with a dark theme. On the left, the 'EXPLORER' panel displays a project structure for 'hello\_world'. The 'lib' directory is expanded, showing 'basic\_widget.dart' selected. Other files include '01.png', '02.png', 'ios', 'main.dart', 'text\_widget.dart', and 'README.md'. The main editor area shows the content of 'basic\_widget.dart', which defines a 'MyImageWidget' class extending 'StatelessWidget'. The class has a constructor 'MyImageWidget((Key? key))' and an 'override' method 'build(BuildContext context)' that returns an 'Image' widget with an 'AssetImage' of 'logo\_polinema.jpg'. A tooltip is visible over the 'key' parameter in the constructor, stating 'Parameter 'key' could be a super parameter.'. The bottom status bar shows 'main.dart', '01.11', '4', '0', '2', '1', 'Debug my code', 'Java: Warning', 'Ln 12, Col 2', 'Spaces: 2', 'UTF-8', 'CRLF', '{}', 'Dart', 'Go Live', 'flutter emulator (android-x64 emulator)', and 'Prettier'.

```
flutter:  
  assets:  
    - logo_polinema.png
```

The screenshot shows an IDE with the following components:

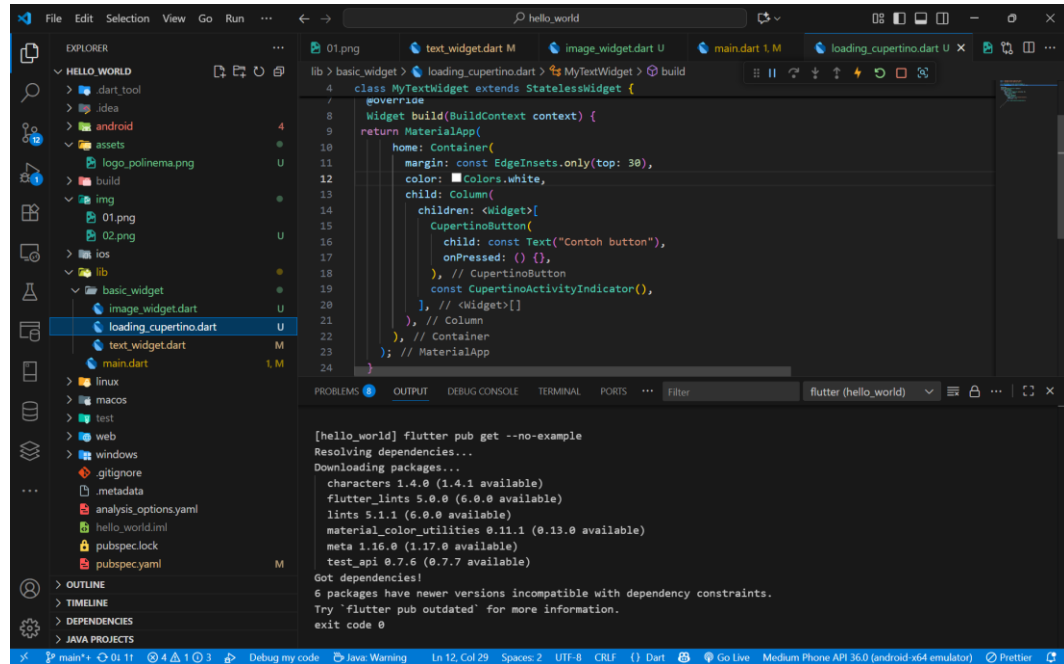
- Explorer Panel (Left):** Displays the project structure. The 'lib' directory is expanded, showing 'main.dart' (1 M) and 'MyHomePageState.dart' (1 M). Other files include 'dart\_tool', 'idea', 'android', 'assets', 'build', 'img', '01.png', '02.png', 'ios', 'lib', 'basic\_widget', 'image\_widget.dart', 'text\_widget.dart', 'linux', 'macos', 'test', 'web', 'windows', 'gitignore', 'metadata', 'analysis\_options.yaml', 'hello\_world.iml', 'pubspec.lock', 'pubspec.yaml', and 'README.md'.
- Main Editor:** Shows the code for 'main.dart'. The code defines a 'MyHomePageState' class that extends 'State<MyHomePage>' and implements 'Widget build(BuildContext context)' to return a horizontal widget list. The list contains a 'Text' widget with the text '0' and an 'Image' widget with the path 'assets/logo.png'.
- Bottom Panel:** The 'OUTPUT' tab is active, showing the command 'flutter pub get --no-example' and its output. The output indicates that dependencies are resolved and packages are downloaded, including 'characters 1.4.0 (1.4.1 available)', 'flutter\_lints 5.0.0 (6.0.0 available)', 'lints 5.1.1 (6.0.0 available)', 'material\_color\_utilities 0.11.1 (0.13.0 available)', 'meta 1.16.0 (1.17.0 available)', and 'test\_api 0.7.6 (0.7.7 available)'. It also states 'Got dependencies!' and '6 packages have newer versions incompatible with dependency constraints. Try 'flutter pub outdated' for more information. exit code 0'.
- Preview Window (Right):** A small window titled 'Flutter Demo Home Page' showing a purple header with the text 'Flutter Demo Home Page' and a large image of the Flutter logo.

## 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

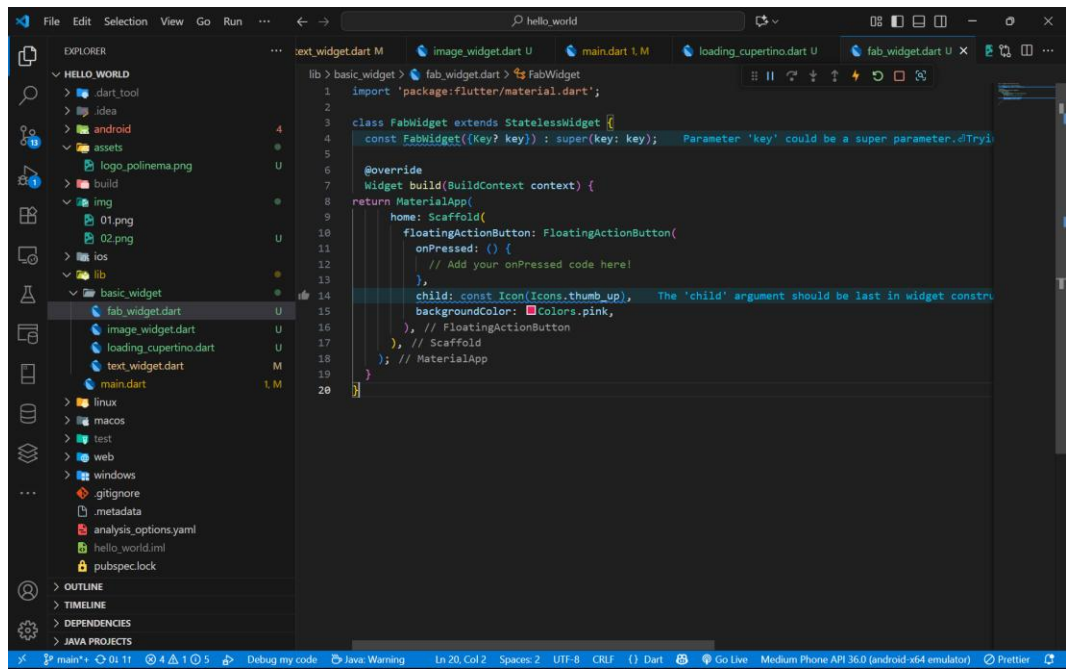
Buat file di `basic_widgets > loading_cupertino.dart`. Import stateless widget dari material dan cupertino. Lalu isi kode di dalam method `Widget build` adalah sebagai berikut.



### - Langkah 2: Floating Action Button (FAB)

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

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



### - 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.headlineMedium,
            ),
          ],
        ),
      ),
      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'),
            ),
          ],
        ),
      ),
    );
  }
}

```