

Coder - Mobile Div

# Flutter Basic & Fundamentals

Sabtu, 12 Oktober 2024



# CREATING NEW FLUTTER PROJECT

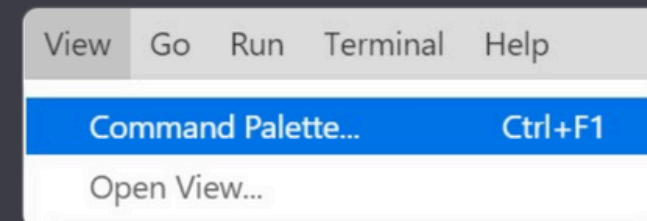
Command Line

```
flutter create [project]  
cd [project]
```

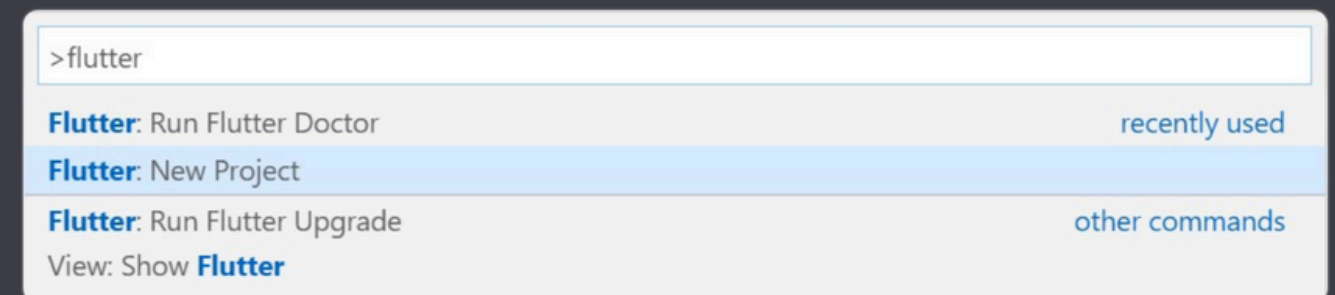
VS Code

1. Buka Visual Studio Code.
2. Aktifkan Command Line dengan cara klik **View>Command**

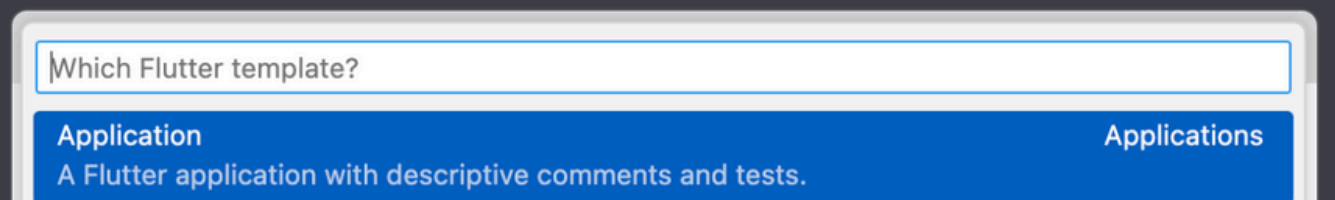
**Pallete...**



3. Ketik **"Flutter"** lalu pilih **Flutter: New Project**.



4. Pilihlah **Application** sebagai template project baru.



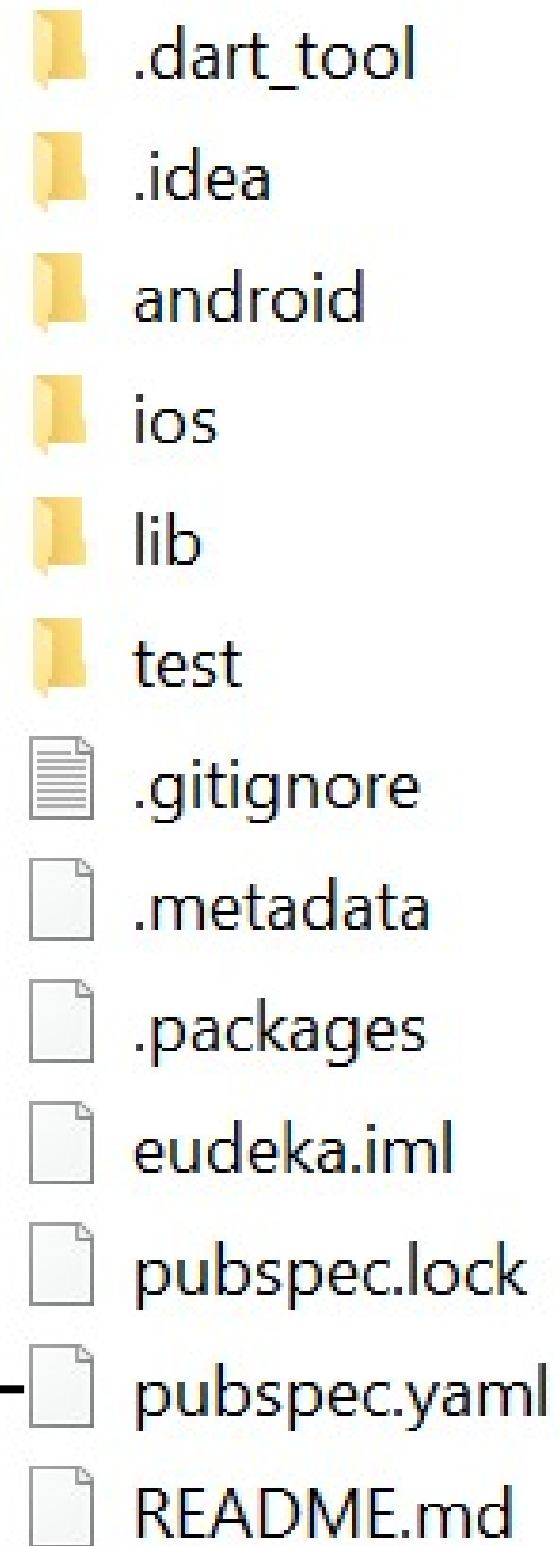
# Struktur Project Flutter



Ketika pertama kali membuat project Flutter, kita akan diberikan aplikasi starter bertema counter. Kode starter ini memiliki contoh dan komentar yang menjelaskan bagaimana sebuah aplikasi Flutter disusun.

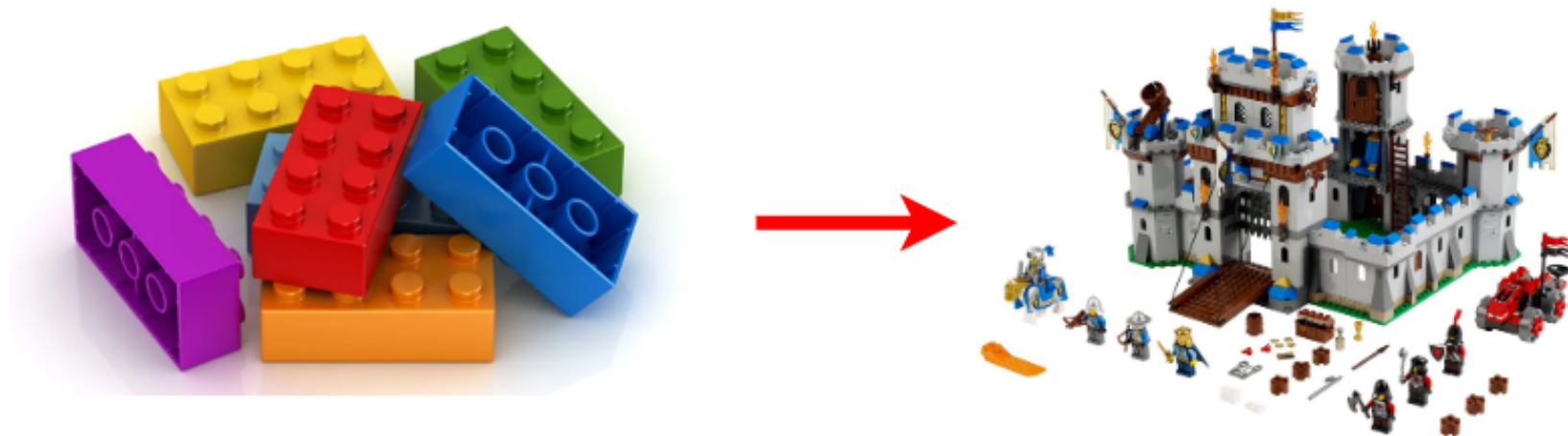
# Struktur Project Flutter

Flutter memiliki package manager bernama pub. Dengan package manager, kita dapat menambahkan package/library yang dibuat oleh developer lain ke dalam project kita.



# Flutter Widget

*“Flutter is all about widget.”*

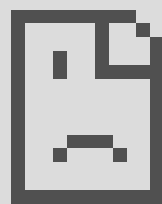


Widget pada Flutter itu terinspirasi oleh salah satu library JavaScript yang digunakan untuk membangun sebuah website yaitu ReactJS. ReactJS memiliki konsep Component.

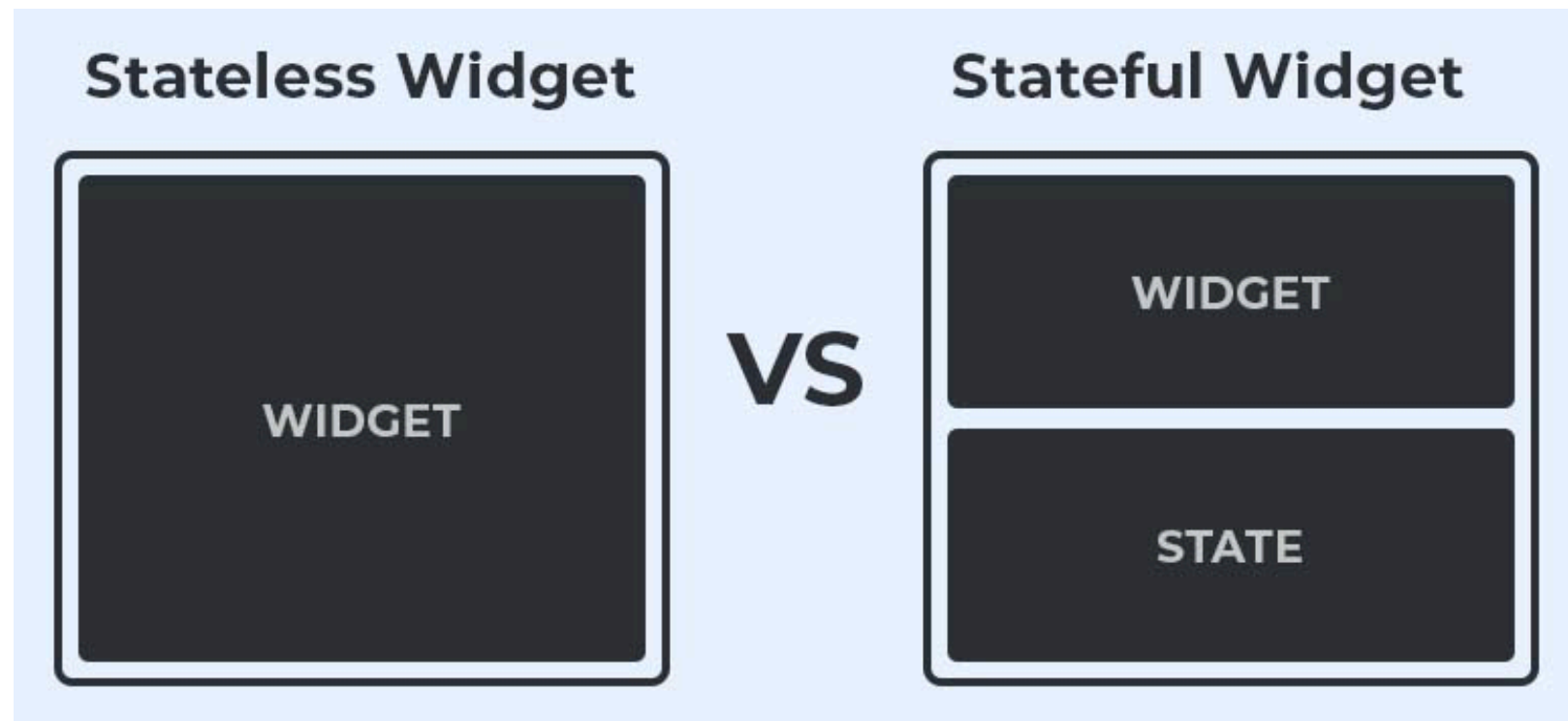
```
class MyApp extends StatelessWidget { Widget
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp( Widget
      title: 'Flutter Demo',
      theme: ThemeData( Widget
        primarySwatch: Colors.blue,
      ),
      home: Scaffold( Widget
        appBar: AppBar( Widget
          title: const Text('Hello, world!'), Widget
        ),
        body: const Center( Widget
          child: Text('Hello, world!'), Widget
        ),
      ),
    );
  }
}
```

**Widget**



# StatelessWidget dan StatefulWidget



## Apa Itu State?

State adalah data yang ada pada suatu widget. Widget menyimpan data yang nantinya dapat berubah sesuai interaksi pengguna.

## Stateless Widget

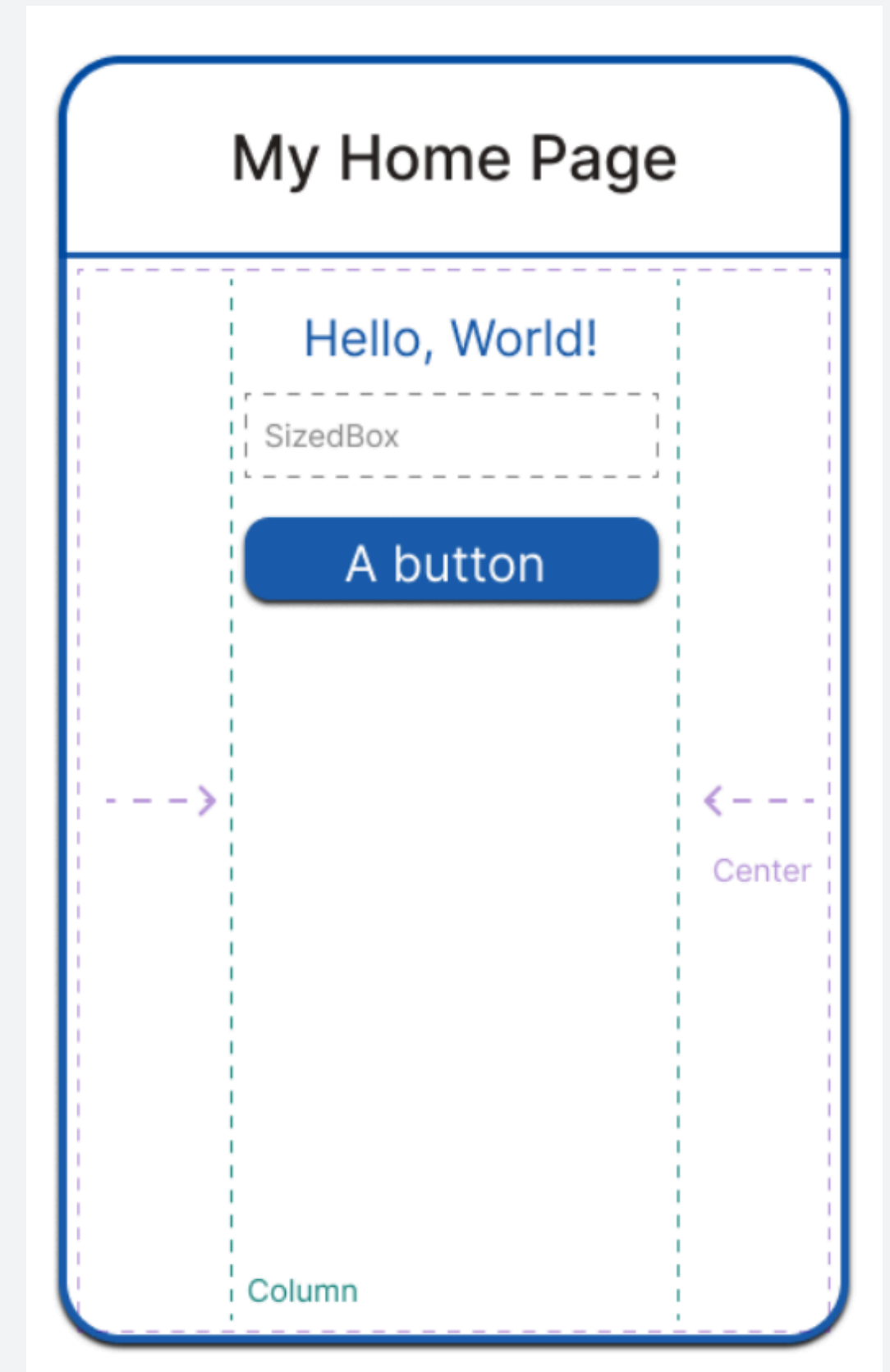
StatelessWidget adalah widget yang nilai state-nya tidak dapat berubah (immutable) maka widget tersebut lebih bersifat statis dan memiliki interaksi yang terbatas.

## Stateful Widget

Kebalikan dari StatelessWidget, StatefulWidget ialah widget yang state-nya dapat berubah-ubah nilainya, yang berarti StatefulWidget bersifat dinamis dan memiliki interaksi yang tak terbatas.

# Flutter's Important Widgets

- Container
- Text
- Scaffold
- AppBar
- Row and Column
- ElevatedButton
- Image
- Icon



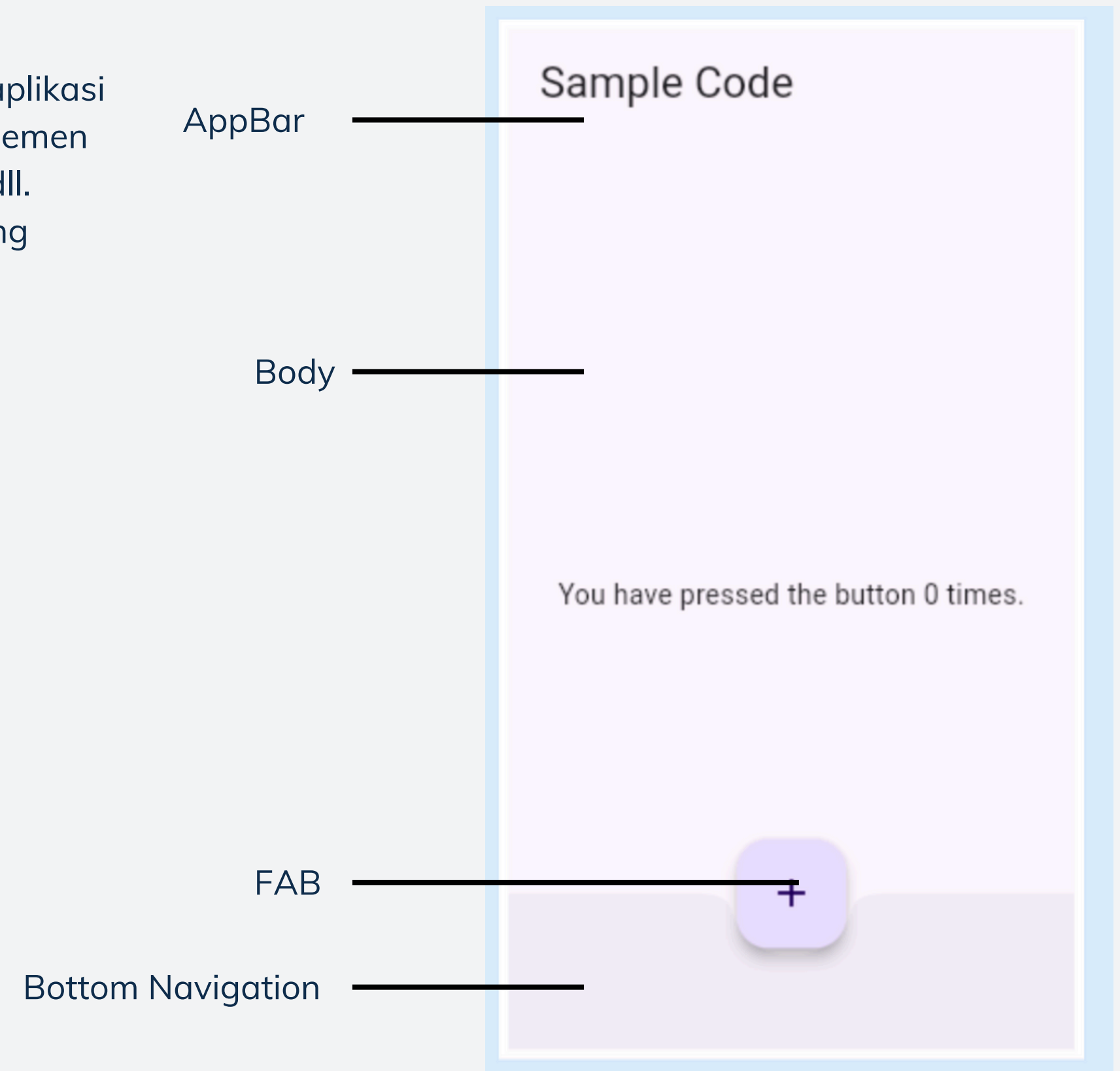


# Scaffold

Scaffold adalah struktur dasar yang menyediakan framework untuk aplikasi yang menggunakan *Material Design*. Biasanya sudah mengikuti elemen seperti AppBar, Body, Floating action button (FAB) , Navigation Bar, dll. Widget Scaffold menyederhanakan pembuatan struktur aplikasi yang umum

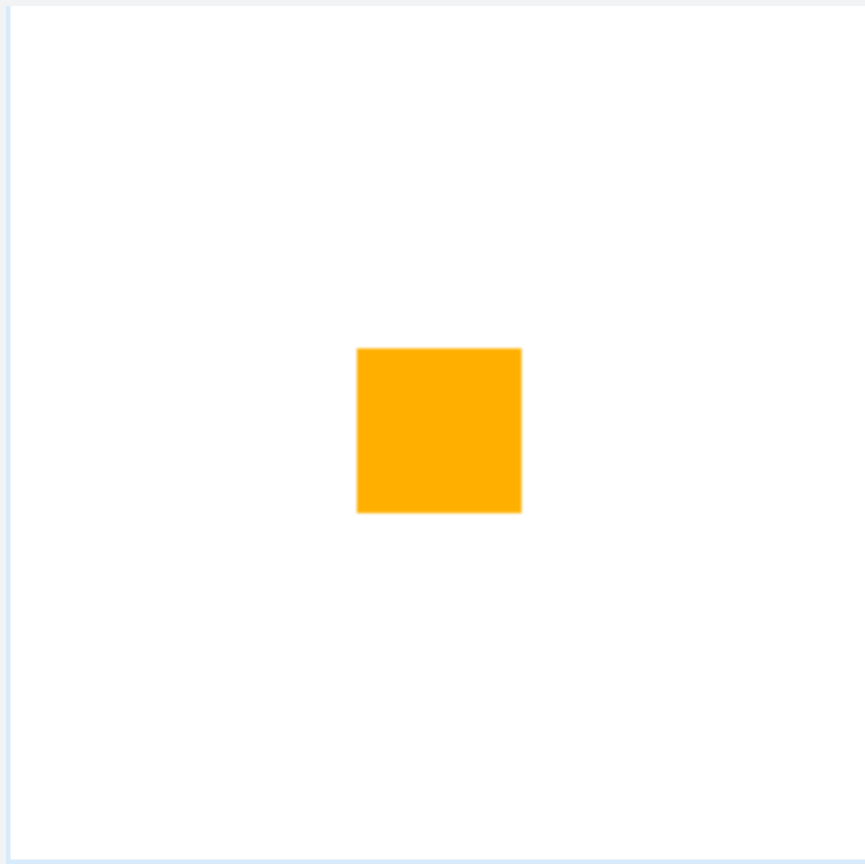
@override

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: const Text('Sample Code'),  
    ),  
    body: Center(  
      child: Text('You have pressed the button $_count times.'),  
    ),  
    bottomNavigationBar: BottomAppBar(  
      shape: const CircularNotchedRectangle(),  
      child: Container(height: 50.0),  
    ),  
    floatingActionButton: FloatingActionButton(  
      onPressed: () => setState(() {  
        _count++;  
      }),  
      tooltip: 'Increment Counter',  
      child: const Icon(Icons.add),  
    ),  
    floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,  
  );  
}
```



# Container

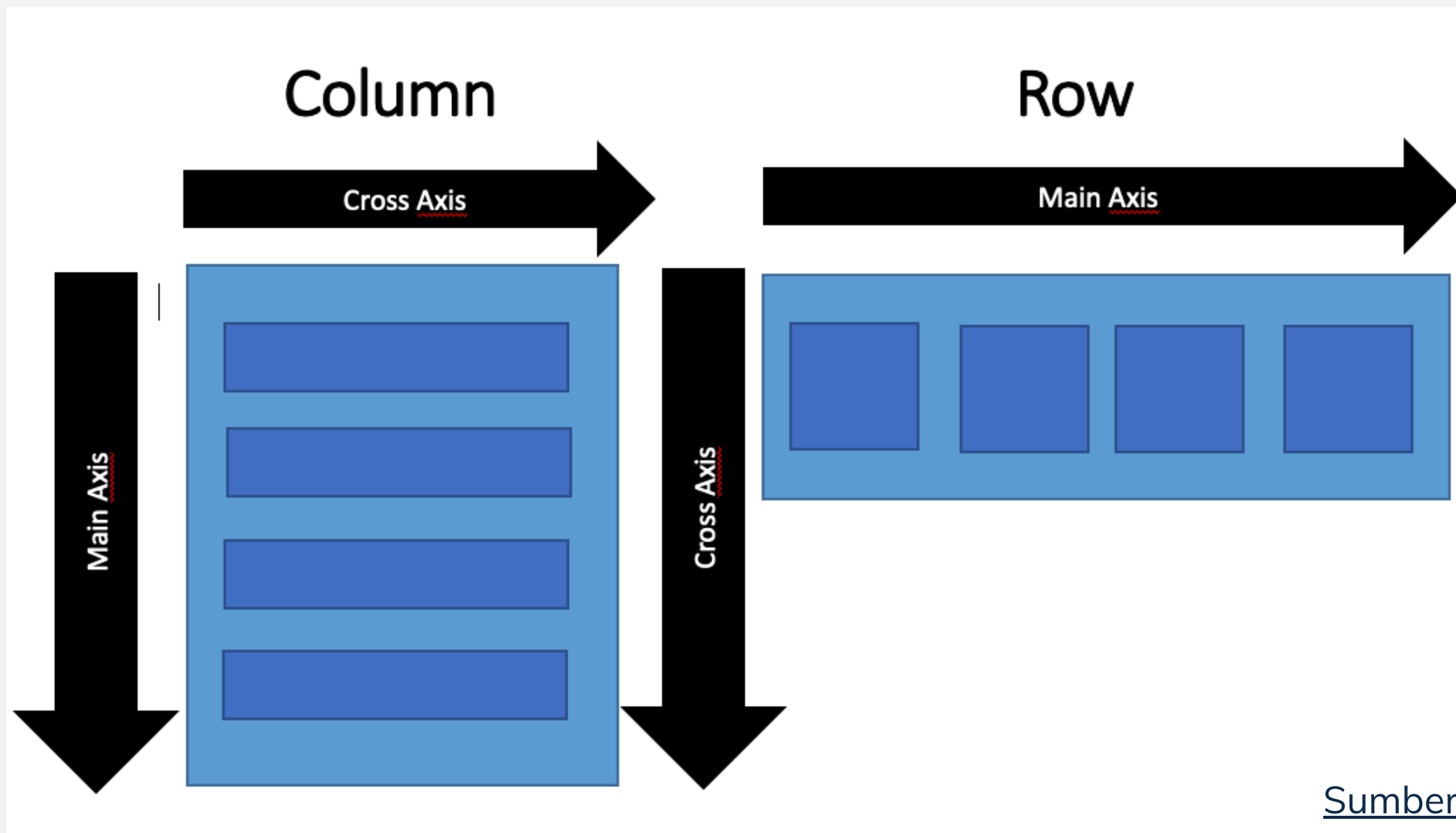
Container adalah widget yang secara praktis menggabungkan penampilan, pengaturan posisi, dan pengaturan ukuran dari widget. Container bisa digunakan untuk menyimpan satu atau banyak widget dan dengan mudah memosisikannya sesuka kita. Secara sederhana, container berperan seperti sebuah kotak untuk menaruh konten yang sudah memiliki padding & margin yang akan memisahkan elemen satu dengan yang lain.



```
Center(  
  child: Container(  
    margin: const EdgeInsets.all(10.0),  
    color: Colors.amber[600],  
    width: 48.0,  
    height: 48.0,  
  ),  
)
```

# Row & Column

Row & Column adalah dua widget paling penting dan paling *powerful* di flutter. Widget-widget ini memungkinkan kita untuk menata *children* secara horizontal maupun vertical sesuai dengan kebutuhan. Hal ini menjadi dasar dalam pembuatan UI di flutter, yang dimana kita perlu menata konten dari aplikasi baik dalam baris ataupun kolom, sehingga widget ini sangatlah dibutuhkan untuk membuat tampilan aplikasi yang menarik.



# Image

Untuk menampilkan gambar yang bersumber dari internet, kita akan menggunakan method `Image.network`. Cara penulisan method ini sebagai berikut:

```
Image.network(url)
```

## Image.asset

Selain melalui internet, kita juga dapat menampilkan gambar yang bersumber dari asset project. Asset di sini berupa gambar-gambar yang nantinya didaftarkan pada project. Untuk mendaftarkan asset gambar pada project kita harus menambahkannya pada berkas `pubspec.yaml`.

Di dalam berkas `pubspec.yaml`, kita bisa mendaftarkan aset gambar pada bagian flutter seperti di bawah ini:

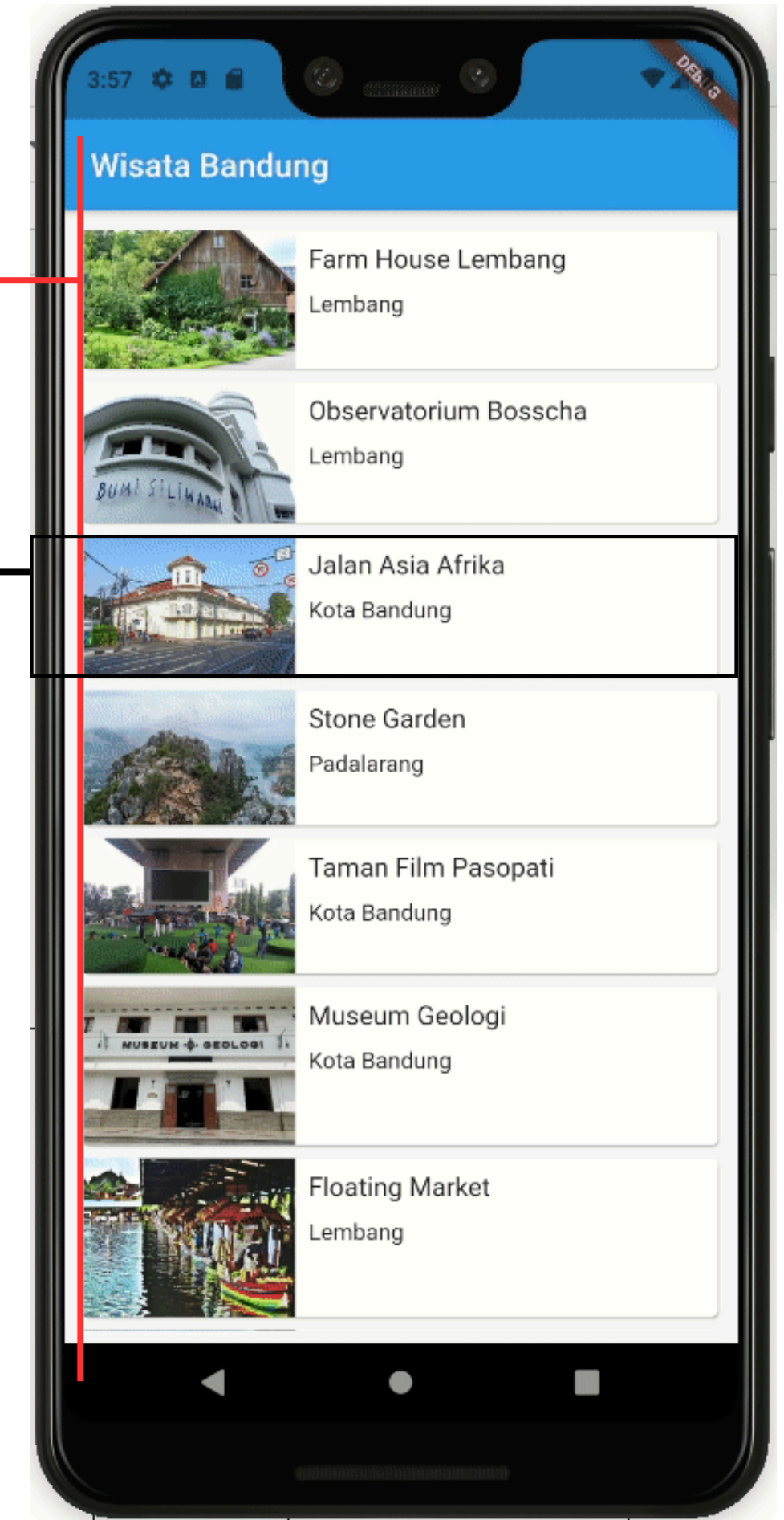
```
2. flutter:  
3.   
4.   uses-material-design: true  
5.   
6.   assets:  
7.     - images/
```

Kita telah mendaftarkan suatu asset. Sekarang kita akan panggil asset tersebut pada kode kita dengan method `Image.asset`. Cara penulisannya seperti berikut:

```
Image.asset(lokasi_asset)
```

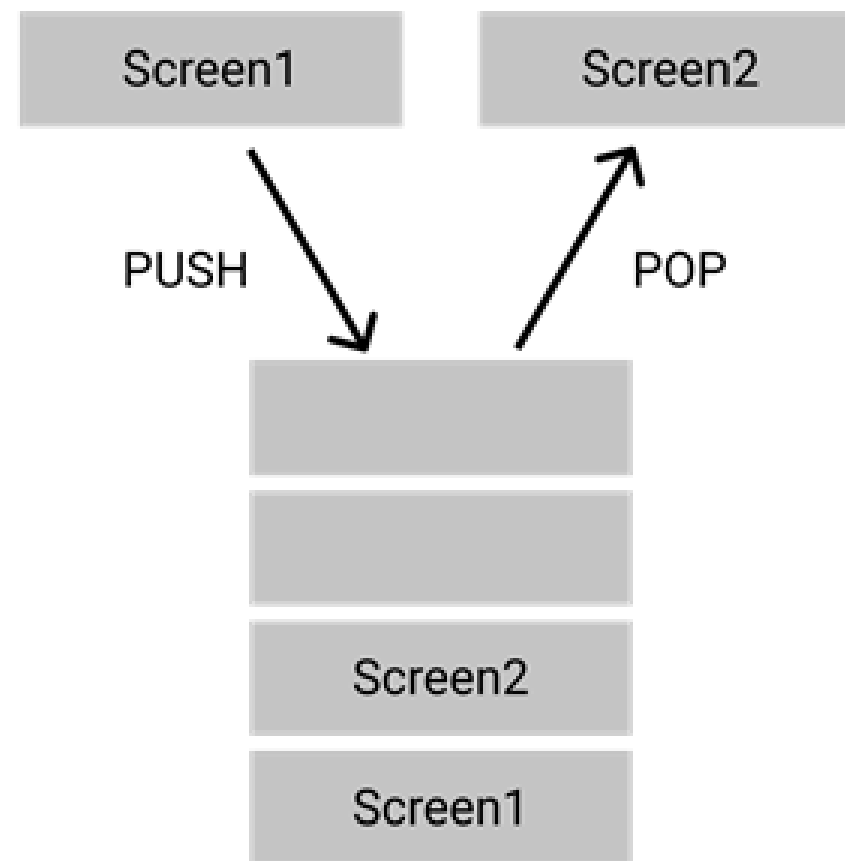
# List View Builder

```
ListView.builder(  
    itemCount: items.length,  
    itemBuilder: (BuildContext context, int index) {  
        return ItemListLayout();  
    },  
)
```



# Flutter Navigation

Konsep navigasi pada Flutter mirip sekali dengan pemrograman Android, yakni bahwa ketika berpindah screen/activity akan menjadi tumpukan (stack). Jadi ketika berpindah dari satu screen ke screen lain (push), maka screen pertama akan ditumpuk oleh screen kedua. Kemudian apabila kembali dari screen kedua ke pertama, maka screen kedua akan dihapus (pop).

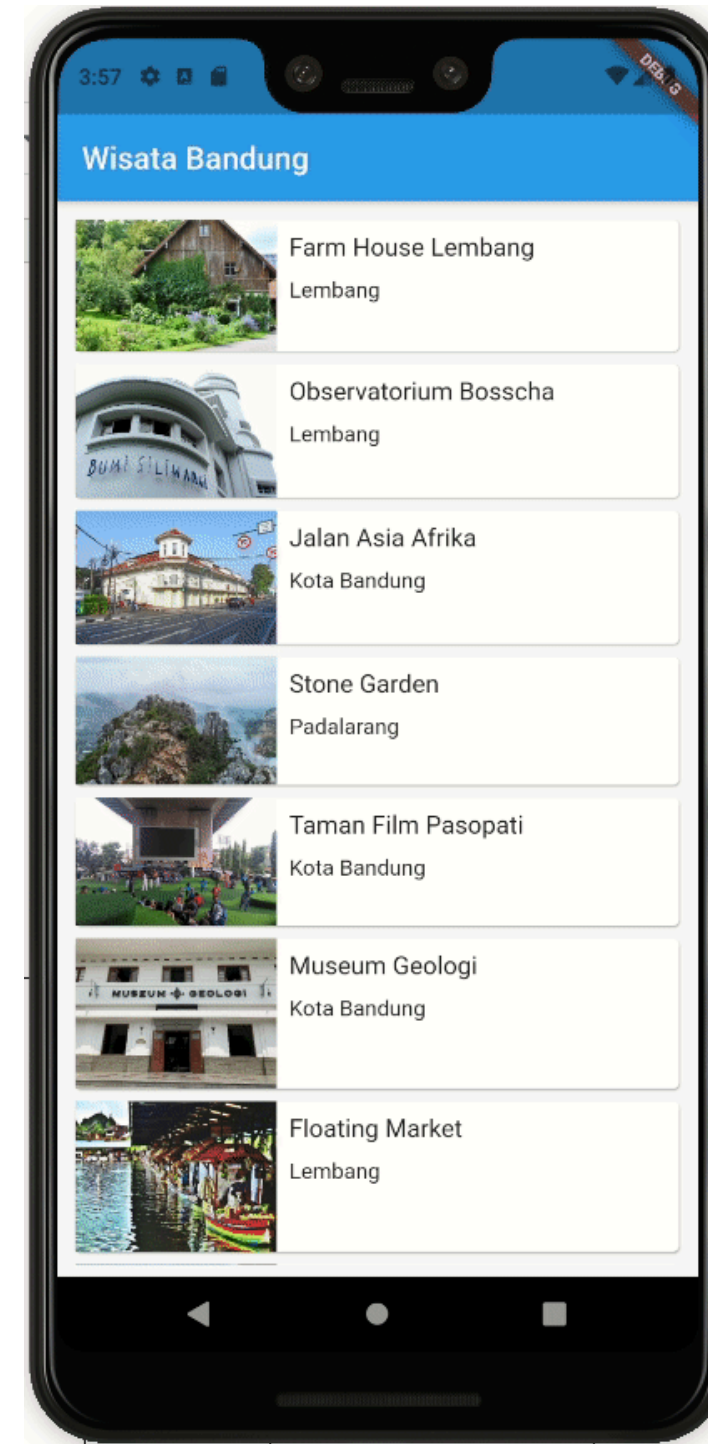


```
Navigator.push(  
  context,  
  MaterialPageRoute(  
    builder: (context) {  
      return WidgetScreen();  
    }  
  )  
);
```

`Navigator.pop(context)`



# Codelab 1



# Referensi

- <https://www.dicoding.com/academies/159-belajar-membuat-aplikasi-flutter-untuk-pemula>
- <https://docs.flutter.dev/ui/widgets>
- <https://docs.flutter.dev/get-started/fundamentals/widgets>
- <https://api.flutter.dev/flutter/material/Scaffold-class.html>
- <https://www.geeksforgeeks.org/container-class-in-flutter/>
- <https://www.geeksforgeeks.org/row-and-column-widgets-in-flutter-with-example/>