

Coder - Mobile Dev

# Flutter Widget

Senin, 11 November 2024



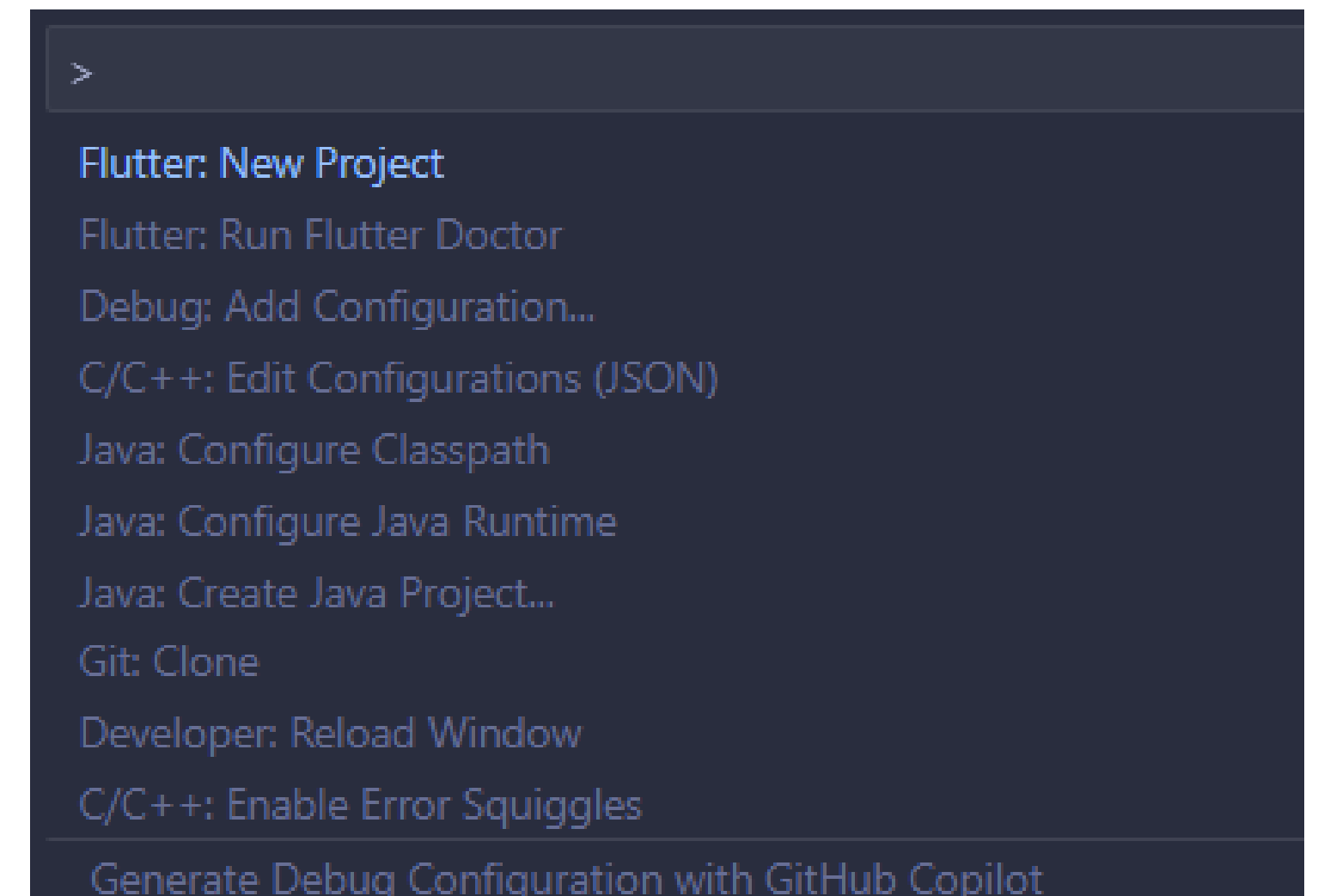
# CREATING NEW FLUTTER PROJECT

CTRL + SHIFT + P

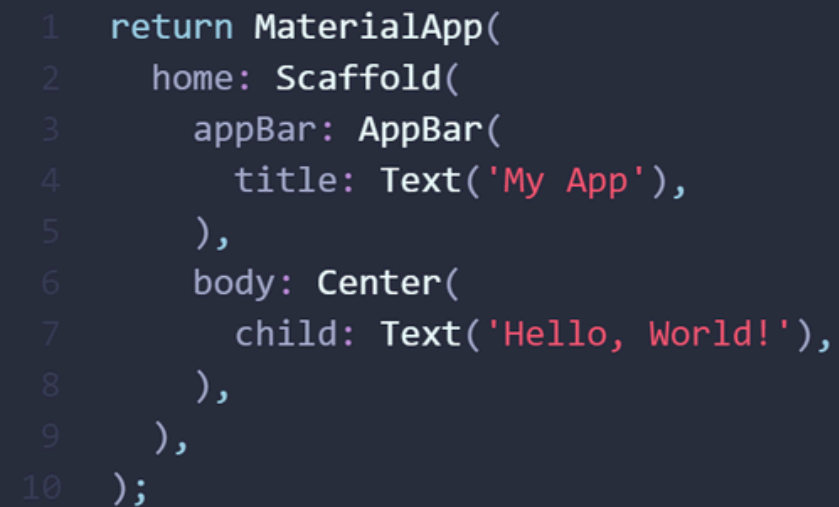
Command Line

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

VS Code



# Widget tree

A screenshot of a code editor with a dark background and light-colored text. The code is a Flutter widget tree definition. It starts with 'return MaterialApp(' followed by 'home: Scaffold(' and 'appBar: AppBar(' with a title 'My App'. The body is a 'Center' widget containing a 'Text' widget with the text 'Hello, World!'. The code is numbered 1 to 10.

```
1 return MaterialApp(  
2   home: Scaffold(  
3     appBar: AppBar(  
4       title: Text('My App'),  
5     ),  
6     body: Center(  
7       child: Text('Hello, World!'),  
8     ),  
9   ),  
10 );
```

```
MaterialApp  
├── Scaffold  
│   ├── AppBar  
│   └── Center  
│       └── Text
```

Dalam Flutter, widget tree adalah hierarki widget yang mewakili antarmuka pengguna aplikasi Flutter. Setiap widget mewakili bagian dari UI, seperti tombol, teks, gambar, atau kontainer, dan dapat berisi widget lain, sehingga membentuk struktur seperti pohon.

# Widget with a Single Child



```
1  body: Center(  
2    child: Text('Hello, World!'),  
3  ),
```

child: Digunakan ketika widget hanya bisa memiliki satu elemen anak. Misalnya, widget seperti Container atau Center menggunakan child untuk menampung satu widget di dalamnya.

# Widget with Multiple Children

```
1  return Scaffold(  
2    body: Row(  
3      children: [  
4        Container(  
5          color: Colors.red,  
6          width: 100,  
7          height: 100,  
8        ),  
9        Container(  
10         color: Colors.green,  
11         width: 100,  
12         height: 100,  
13       ),  
14       Container(  
15         color: Colors.blue,  
16         width: 100,  
17         height: 100,  
18       ),  
19     ],  
20   ),  
21 );
```

children: Digunakan ketika widget dapat memiliki beberapa elemen anak sekaligus, yang didefinisikan dalam bentuk array. Contohnya adalah widget seperti Column, Row, dan Stack, yang menggunakan children untuk menampung banyak widget.

# SafeArea



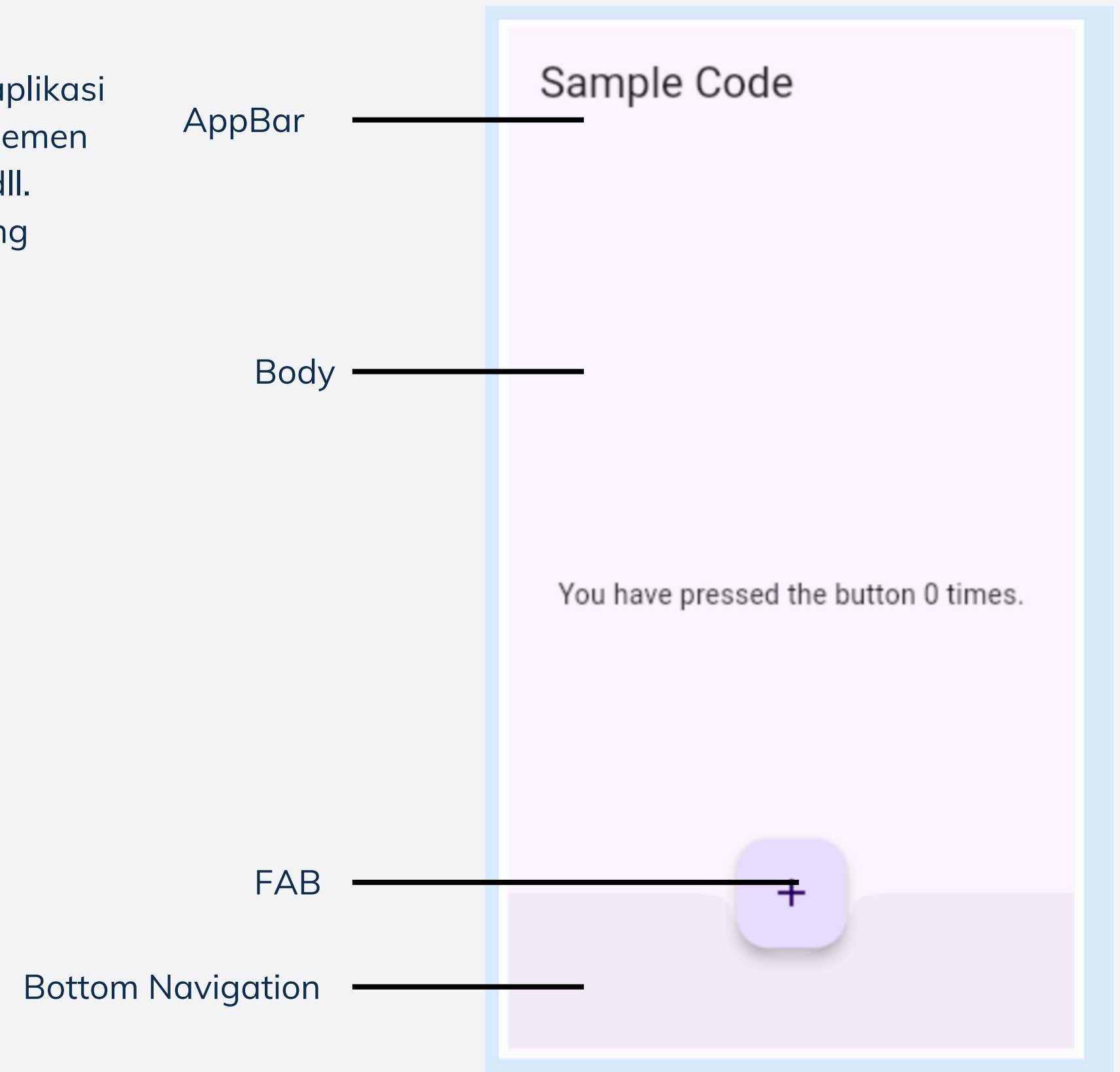
```
1 return SafeArea(  
2     child: Scaffold(  
3         body: Center(  
4             child: Text('Hello World'),  
5         ),  
6     ),  
7 );
```

SafeArea adalah widget di Flutter yang digunakan untuk menghindari area yang tidak aman di layar, seperti takik (notch), tepi layar, atau status bar. Dengan SafeArea, konten secara otomatis diberi jarak agar tidak tertutup oleh elemen-elemen tersebut, membuat tampilan lebih rapi dan nyaman dilihat di berbagai jenis perangkat.

# Scaffold

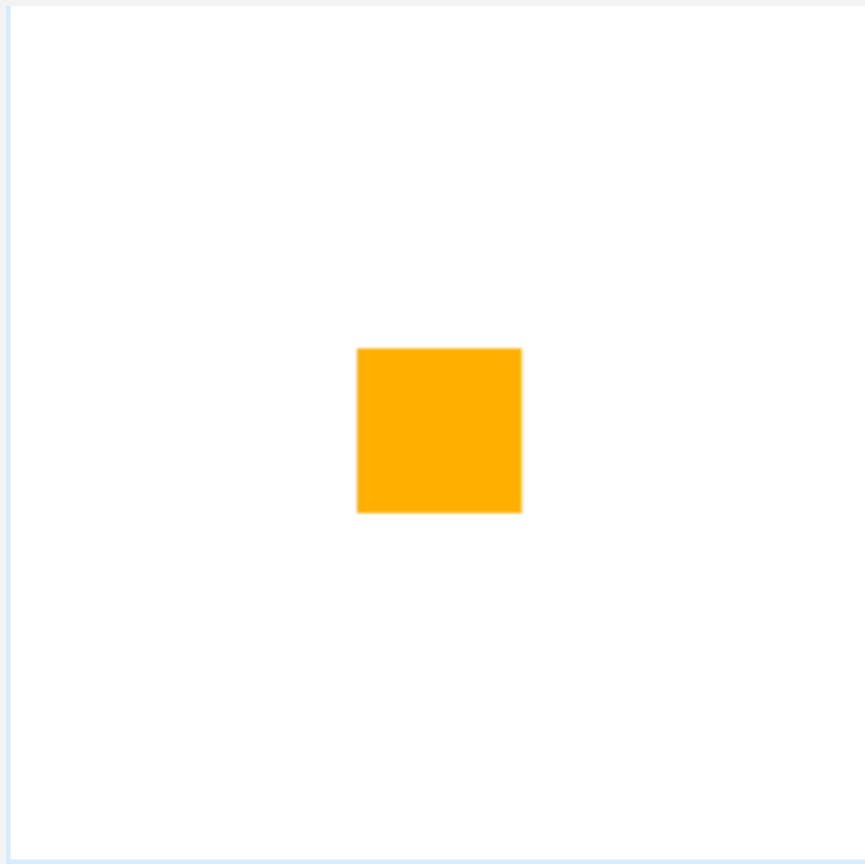
Scaffold adalah struktur dasar yang menyediakan framework untuk aplikasi yang menggunakan *Material Design*. Biasanya sudah mengikuti elemen seperti AppBar, Body, Floating action button (FAB) , Navigation Bar, dll. Widget Scaffold menyederhanakan pembuatan struktur aplikasi yang umum

```
1 return Scaffold(  
2   appBar: AppBar(  
3     title: Text('Widgets'),  
4   ),  
5   body: Center(  
6     child: Text('Hello World'),  
7   ),  
8   floatingActionButton: FloatingActionButton(  
9     onPressed: () {},  
10    child: Icon(Icons.add),  
11  ),  
12  bottomNavigationBar: BottomAppBar(  
13    child: Padding(  
14      padding: const EdgeInsets.all(16.0),  
15      child: Text('Bottom Navigation'),  
16    ),  
17  ),  
18 );
```



# 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 memposisikannya 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.

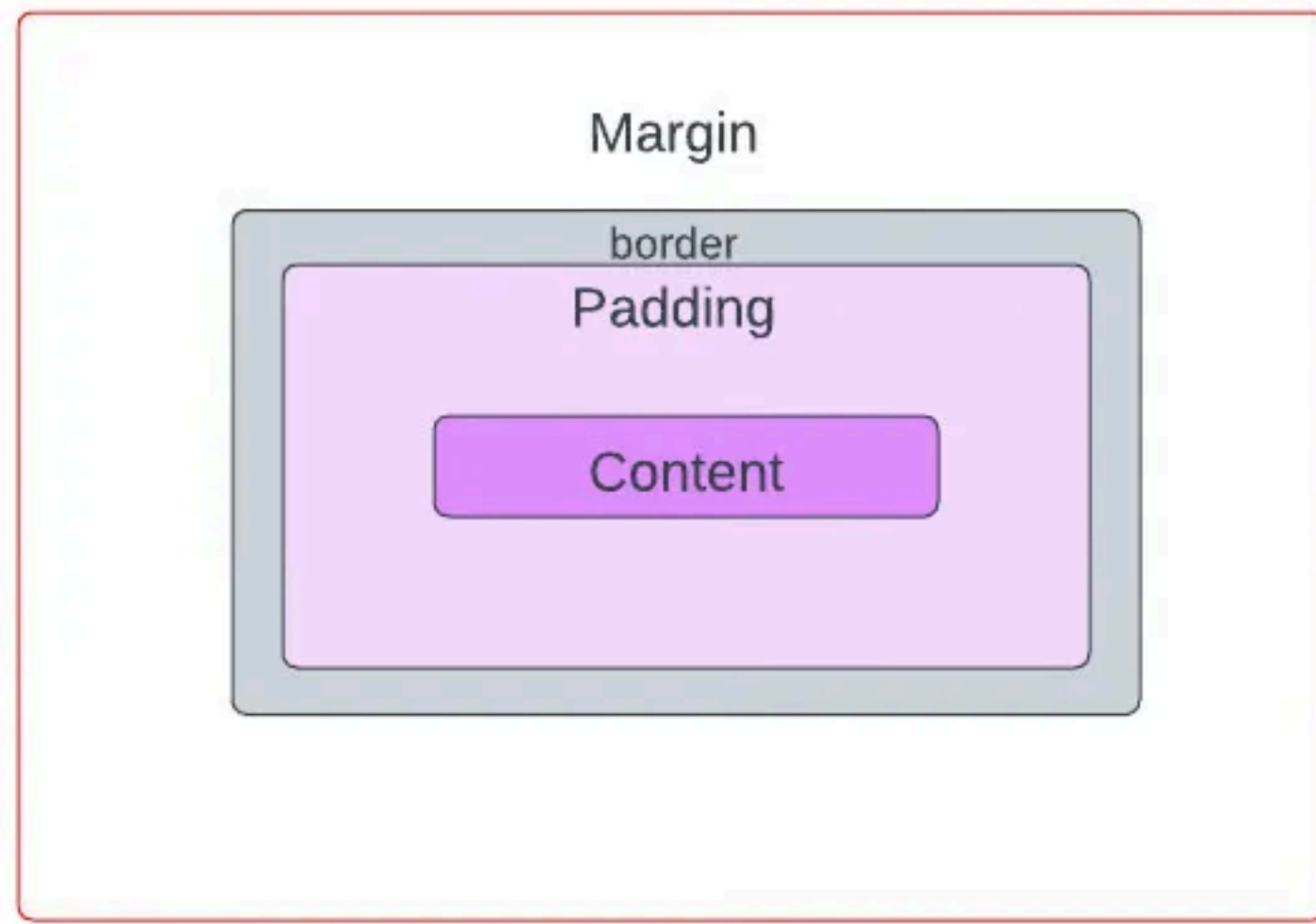


```
Center(  
  child: Container(  
    margin: const EdgeInsets.all(10.0),  
    color: Colors.amber[600],  
    width: 48.0,  
    height: 48.0,  
  ),  
)
```



# Margin and Padding

Container memiliki beberapa properti untuk mempermudah kita menata elemen UI kita. Diantaranya adalah Margin & Padding. Margin adalah ruang yang ada di sekitar widget. Sedangkan Padding adalah ruang yang ada di dalam widget dan biasanya mengelilingi konten dari widget itu. Dengan menggunakan container kita bisa mengatur padding dan margin dari elemen UI kita



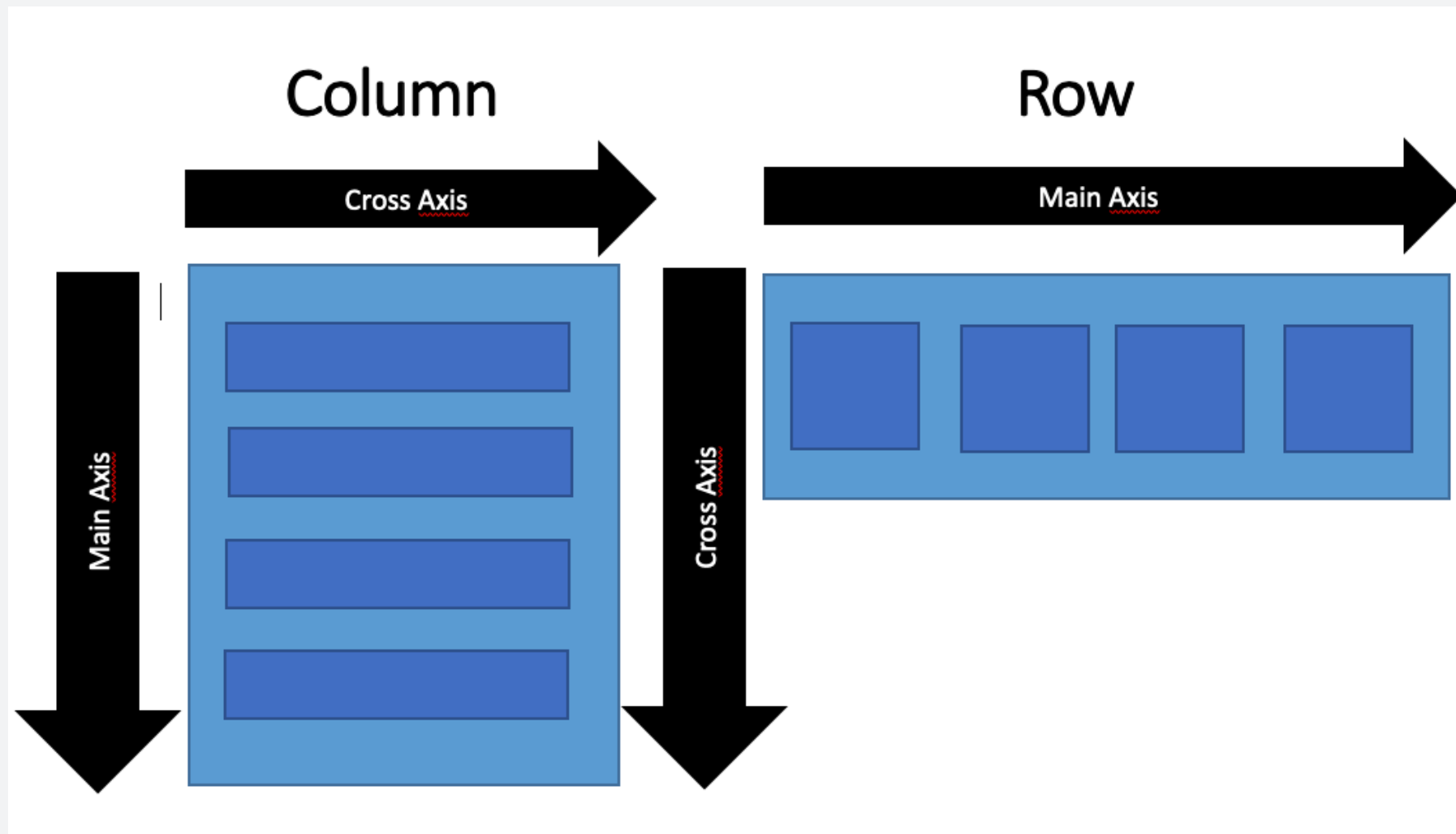
```
1 Container(  
2   margin: EdgeInsets.all(16.0),  
3   child: Text('Hello, World!'),  
4 )
```



```
1 Container(  
2   padding: EdgeInsets.all(16.0),  
3   child: Text('Hello, World!'),  
4 ),
```

# 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.



```
1 body: Row(  
2   mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
3   crossAxisAlignment: CrossAxisAlignment.center,  
4   children: [  
5     Text('Widget 1'),  
6     Text('Widget 2'),  
7     Text('Widget 3'),  
8   ],  
9 ),
```

```
1 body: Column(  
2   mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
3   crossAxisAlignment: CrossAxisAlignment.center,  
4   children: [  
5     Text('Widget 1'),  
6     Text('Widget 2'),  
7     Text('Widget 3'),  
8   ],  
9 ),
```

# 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:

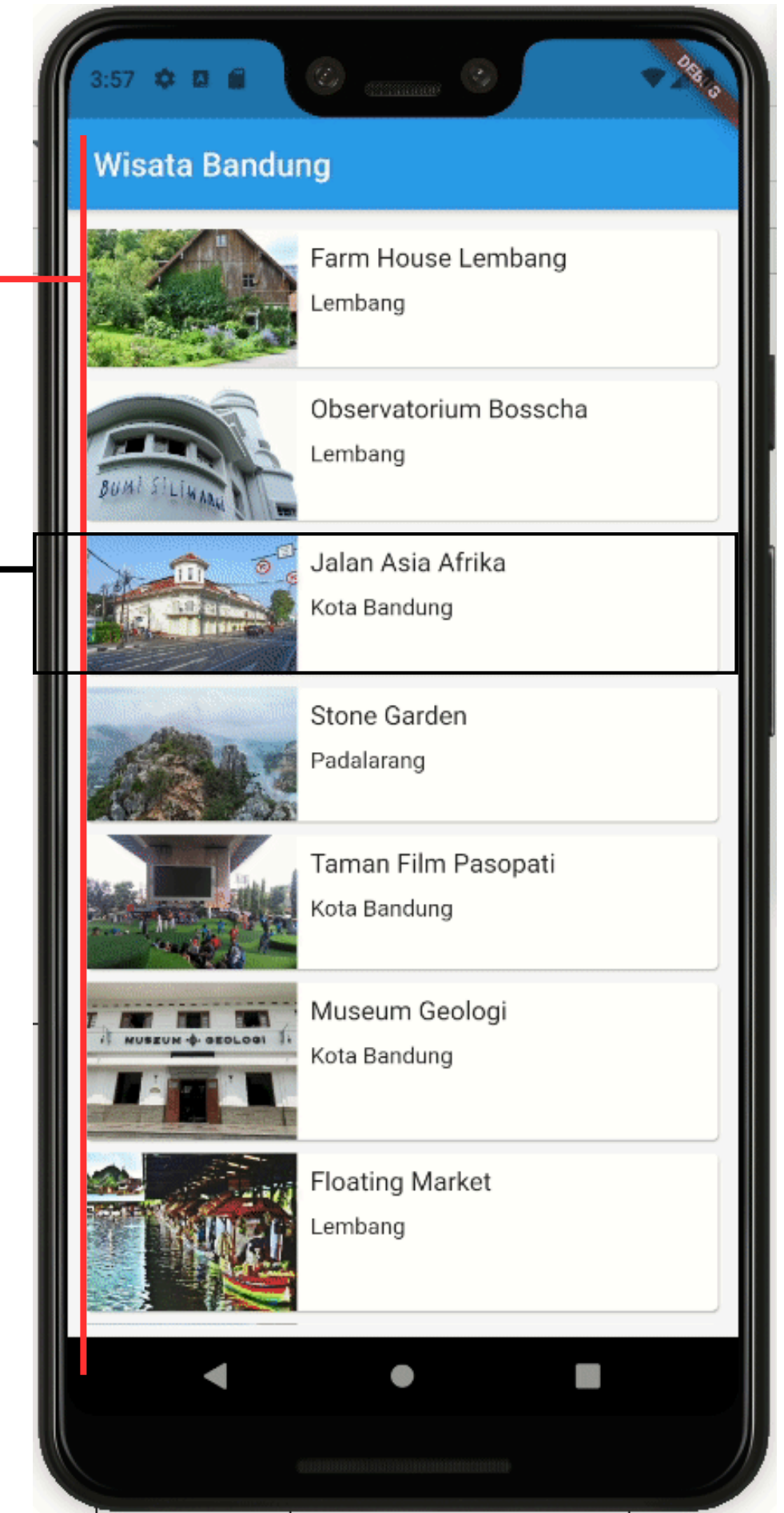
```
2. flutter:  
3.   
4.   uses-material-design: true  
5.   
6.   assets:  
7.     - images/
```

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();  
    },  
)
```



# 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/>