Nama : Akmal Muhamad Firdaus

NIM : 1301204188

Jurnal Praktikum Modul 9

1. Pengenalan Widget

Widget di Flutter adalah istilah yang digunakan untuk mengacu pada elemen dasar yang digunakan untuk membangun antarmuka pengguna pada aplikasi Flutter. Widget dapat berupa objek grafis seperti tombol, teks, gambar, atau bahkan objek yang lebih kompleks seperti list atau form.

Dalam Flutter, setiap widget adalah objek yang dapat diatur, ditempatkan, dan diubah dengan kode. Widget dapat dikombinasikan untuk membentuk struktur antarmuka pengguna yang lebih kompleks.

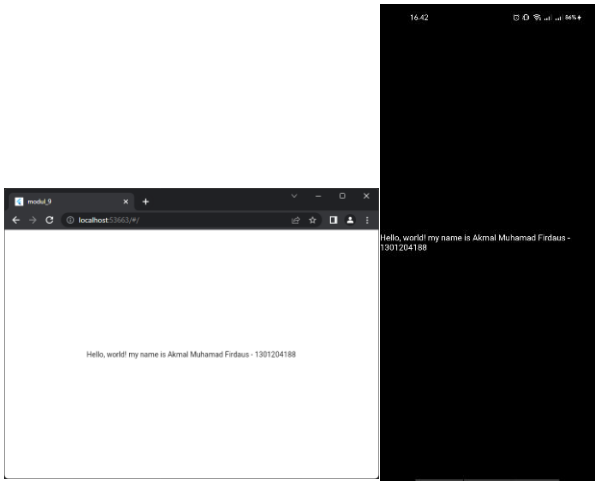
Widget dapat dibagi menjadi dua kategori: stateful dan stateless. Widget stateless adalah widget yang tidak memiliki data yang berubah selama waktu runtime, sedangkan widget stateful adalah widget yang memiliki data yang dapat berubah selama waktu runtime.

Flutter menyediakan banyak widget yang sudah dibuat siap pakai, dan juga memungkinkan pembuatan widget baru secara kustom dengan menurunkannya dari widget dasar yang sudah ada. Dengan menggunakan widget, membangun antarmuka pengguna yang menarik dan interaktif di Flutter menjadi lebih mudah dan cepat dilakukan.

Berikut contoh penerapan HelloWorld pada Flutter:

```dart
import 'package:flutter/material.dart';

void main() {
  runApp(
    const Center(
      child: Text(
        'Hello, world! my name is Akmal Muhamad Firdaus - 1301204188',
        textDirection: TextDirection.ltr,
      ), // Text
    ), // Center
  );
}
```



Untuk output, sebenarnya dapat di build kedalam platform mobile (Android / iOS), tetapi selanjutnya saya akan melakukan build dan menampilkan di browser saja, karena jika build menggunakan emulator akan memberatkan kinerja ram.

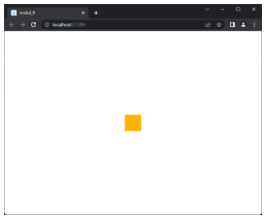
2. Container

Container adalah widget yang digunakan untuk menampung dan mengatur widget lainnya dalam antarmuka pengguna. Container biasanya digunakan untuk mengatur tata letak (layout) widget lainnya, memberikan margin dan padding, dan memberikan batas (border) di sekitar widget.

Dalam Flutter, Container memiliki properti seperti margin, padding, width, height, decoration, transform, dan lain-lain. Properti-properti ini memungkinkan kita untuk mengubah tampilan dan perilaku Container serta widget yang berada di dalamnya.

Berikut adalah contoh penggunaan Container:

```dart
import 'package:flutter/material.dart';
void main() {
  //Akmal Muhamad Firdaus - 1301204188
  runApp(
    Center(
      child: Container(
        margin: const EdgeInsets.all(10.0),
        color: Colors.amber[600],
        width: 48.0,
        height: 48.0,
      ),  // Container
    ),  // Center
  );
}
```

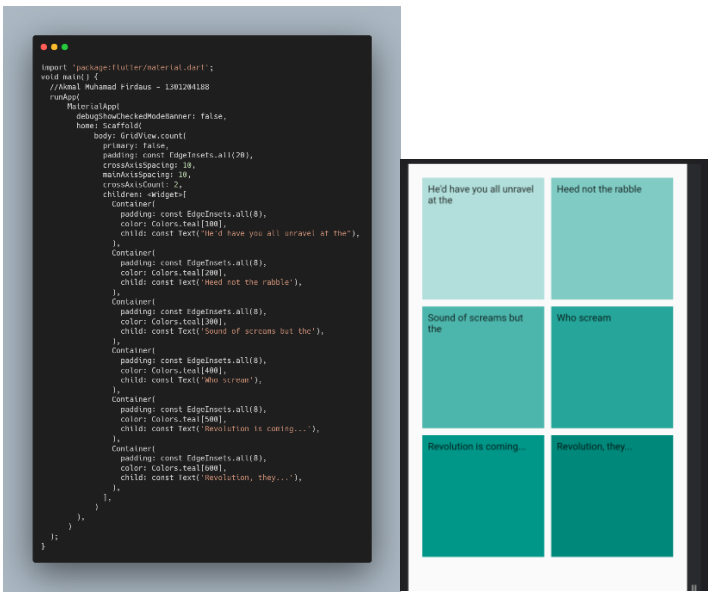


3. GridView

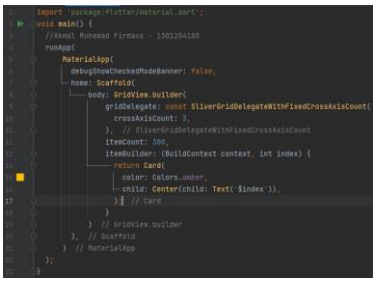
GridView adalah salah satu widget yang digunakan untuk menampilkan widget dalam bentuk grid atau tabel. GridView dapat menampilkan banyak widget dalam bentuk kotak-kotak dengan ukuran yang sama atau berbeda, dan kita dapat mengatur jumlah kolom yang ingin ditampilkan dalam GridView.

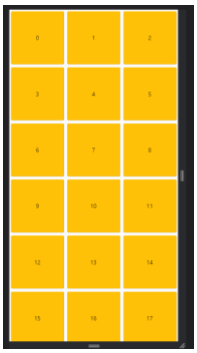
GridView dapat dibagi menjadi dua jenis, yaitu:

3.1. GridView.count: Menampilkan widget dalam bentuk grid dengan jumlah kolom yang sudah ditentukan.



3.2. GridView.builder: Menampilkan widget dalam bentuk grid dengan jumlah kolom yang bervariasi.

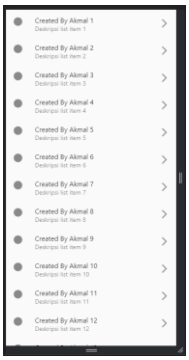




4. ListView

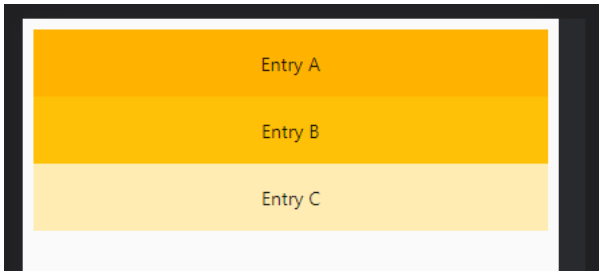
ListView adalah salah satu widget yang digunakan untuk menampilkan daftar data dalam antarmuka pengguna. ListView dapat menampilkan daftar item secara vertikal atau horizontal, dan dapat menampilkan daftar data dalam bentuk yang berbeda, seperti teks, gambar, atau widget kompleks.

Contoh penggunaan ListView:



4.1. ListView.builder

Widget ini cocok digunakan ketika memiliki data list yang lebih besar. ListView.builder membutuhkan itemBuilder dan itemCount. Parameter itemBuilder merupakan fungsi yang mengembalikan widget untuk ditampilkan. Sedangkan itemCount kita isi dengan jumlah seluruh item yang ingin ditampilkan.

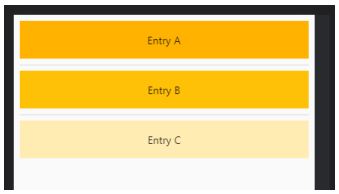


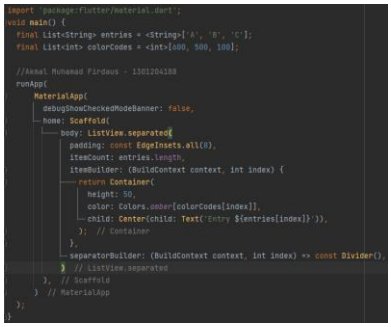

```dart
import 'package:flutter/material.dart';
void main() {
  final List<String> entries = <String>['A', 'B', 'C'];
  final List<int> colorCodes = <int>[600, 500, 100];

  //Akmal Muhamad Firdaus - 1301204188
  runApp(
      MaterialApp(
        debugShowCheckedModeBanner: false,
        home: Scaffold(
          body: ListView.builder(
              padding: const EdgeInsets.all(8),
              itemCount: entries.length,
              itemBuilder: (BuildContext context, int index) {
                return Container(
                  height: 50,
                  color: Colors.amber[colorCodes[index]],
                  child: Center(child: Text('Entry ${entries[index]}')),
                ); // Container
              }
          ), // ListView.builder
        ), // Scaffold
      ) // MaterialApp
  );
}
```

4.2. ListView.separated

ListView jenis ini akan menampilkan daftar item yang dipisahkan dengan separator. Penggunaan ListView.separated mirip dengan builder, yang membedakan adalah terdapat satu parameter tambahan wajib yaitu separatorBuilder yang mengembalikan Widget yang akan berperan sebagai separator.





5. Stack

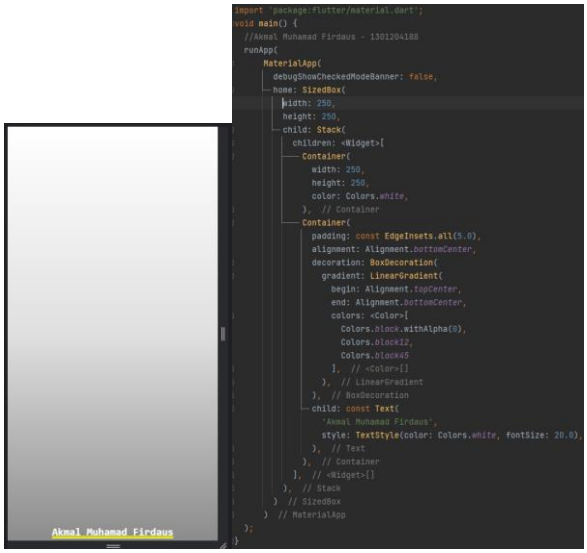
Widget ini merupakan widget yang saling tumpang tindih terhadap widget lain. Seperti image dan text yang saling bertumpuk, atau overlay yang terdapat button dan widget lainnya. Dengan menggunakan Stack dapat memposisikan widget satu sama lain dan bertumpukan antar widget.

```dart
import 'package:flutter/material.dart';
void main() {
  final List<String> entries = <String>['A', 'B', 'C'];
  final List<int> colorCodes = <int>[600, 500, 100];

  //Akmal Muhamad Firdaus - 1301204188
  runApp(
      MaterialApp(
        debugShowCheckedModeBanner: false,
        home: Scaffold(
          body: Stack(
            children: <Widget>[
              Container(
                width: 100,
                height: 100,
                color: Colors.red,
              ),  // Container
              Container(
                width: 90,
                height: 90,
                color: Colors.green,
              ),  // Container
              Container(
                width: 80,
                height: 80,
                color: Colors.blue,
              ),  // Container
            ],  // <Widget>[]
          )  // Stack

        ),  // Scaffold
      )  // MaterialApp
  );
}
```

Selanjutnya adalah penerapan Stack dengan text dan ditambahkan dengan background gradient di belakangnya. Berikut contoh penerapannya:



Source code: https://github.com/codernewbie04/CII3H4-ABP-Praktikum