**LAPORAN PEMROGRAMAN MOBILE**

**MEMBUAT KARTU NAMA**

**Dosen Pengampu:**

I Gde Agung Sidhimantra, S.Kom., M. Kom.

**Disusun Oleh:**

Kelompok 7

1. Feis Aulia Fatchuriani (22091397053)
2. Berlin Marsyah Yustina (22091397062)
3. Tsamarah Mu’adzah Lubis (22091397074)

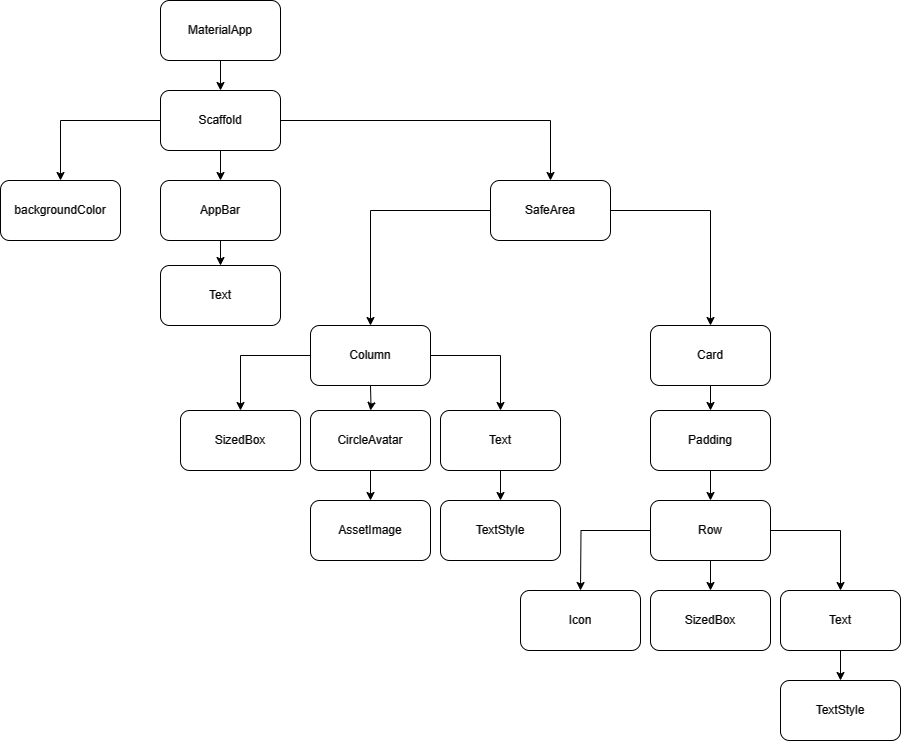
**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2024**

* **Diagram Aplikasi Kartu Nama**



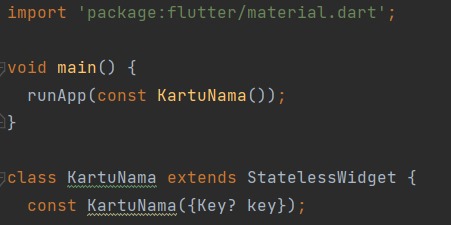
* **Penjelasan Diagram**

Pada tingkat tertinggi, aplikasi dimulai dengan widget root yang disebut "MaterialApp". Ini bertanggung jawab untuk mengatur tema dan skema warna yang akan diterapkan ke seluruh aplikasi. Di dalam "MaterialApp", kita memiliki "Scaffold" yang berperan sebagai layout dasar untuk aplikasi. "Scaffold" terdiri dari beberapa elemen penting seperti "BackgroundColor", "AppBar", dan "SafeArea". AppBar adalah komponen yang umumnya terletak di bagian atas layar dan berisi judul aplikasi, tombol navigasi, dan ikon aksi. Sedangkan "SafeArea" memastikan bahwa konten aplikasi tidak terhalangi oleh status bar atau notch pada perangkat tertentu.

Di dalam AppBar, kita biasanya menemukan widget "Text" yang digunakan untuk menampilkan teks dengan properti-properti seperti ukuran, warna, dan gaya huruf yang dapat diatur. Di sisi lain, di dalam "SafeArea", terdapat "Column" dan "Card". Column digunakan untuk mengatur layout secara vertikal, di mana widget anak ditumpuk ke bawah. Di dalam Column, seringkali terdapat widget seperti "SizedBox" yang menambahkan jarak atau ukuran spesifik pada layout, "CircleAvatar" yang menampilkan gambar profil berbentuk lingkaran, dan "Text".

Sementara itu, di dalam "Card", kita menemukan "Row" yang digunakan untuk mengatur layout secara horizontal, di mana widget anak ditumpuk berdampingan. Di dalam "Row" ini, mungkin terdapat widget "Icon" yang menampilkan ikon dari Material Design atau ikon kustom, serta "SizedBox" dan "Text". Card juga sering menggunakan Padding untuk menambahkan jarak di dalamnya, yang biasanya digunakan untuk menampilkan konten dengan sedikit efek visual tambahan. Dengan struktur hierarki widget yang terorganisir seperti ini, pengembang dapat dengan mudah membangun dan mengatur tata letak serta elemen-elemen UI dalam aplikasi Flutter mereka.

* **Source Code Pada Folder ‘main.dart’**



Kode tersebut adalah kode Flutter yang digunakan untuk membuat sebuah aplikasi Flutter sederhana yang menampilkan sebuah "card" atau kartu nama. Aplikasi ini menggunakan Flutter framework, yang merupakan framework yang digunakan untuk membuat aplikasi mobile lintas platform. Dengan penjelasan kode sebagai berikut:

* Import Flutter Material Package

**(import 'package:flutter/material.dart';`)**

Ini adalah baris yang mengimpor pustaka Material Design dari framework Flutter. Pustaka ini menyediakan widget dan class untuk membangun antarmuka pengguna aplikasi Flutter.

* Fungsi Main

**(void main() { ... }`)**

Fungsi utama yang dijalankan ketika aplikasi dimulai. Di dalamnya, kita menjalankan fungsi `runApp` dengan menyediakan widget `KartuNama` sebagai root widget.

* Class KartuNama

**(class KartuNama extends StatelessWidget { ... }`)**

Kelas ini adalah widget stateless yang mewarisi `StatelessWidget`. Kelas ini tidak memiliki status internal dan membangun tampilan berdasarkan data yang diberikan padanya.

* Konstruktor KartuNama

**(const KartuNama({Key? key});`)**

Konstruktor kelas `KartuNama` yang tidak melakukan apa pun selain menerima kunci opsional.

* Metode Build

**(@override Widget build(BuildContext context) { ... }`)**

Metode ini mengembalikan widget yang akan ditampilkan di layar. Di sini, kita mengembalikan widget `MaterialApp` yang berisi `Scaffold` dengan `AppBar` dan `Center` sebagai badan konten.

* RunApp

**(runApp(MaterialApp(...));`)**

Fungsi `runApp` menerima sebuah widget yang akan menjadi root dari pohon widget aplikasi. Di sini, kita memberikan `MaterialApp` sebagai root widget, yang menyediakan struktur dasar untuk aplikasi Flutter.



Tujuan dari kode tersebut adalah untuk membuat tampilan awal (home) dari sebuah aplikasi Flutter yang menggunakan Material Design. Dengan penjelasan kode sebagai berikut:

* Metode Build

**(`@override Widget build(BuildContext context) { ... }`)**

Metode ini mengembalikan widget yang akan ditampilkan di layar. Di sini, kita mengembalikan widget `MaterialApp` yang berisi `Scaffold`.

* MaterialApp

**(`MaterialApp( ... )`)**

Widget ini menyediakan struktur dasar untuk aplikasi Flutter. Dalam kasus ini, `home` dari `MaterialApp` adalah widget `Scaffold`.

* Scaffold

**(`Scaffold( ... )`)**

Widget ini menyediakan kerangka dasar untuk menampilkan berbagai elemen antarmuka pengguna, seperti appBar, body, dan lain-lain. Dalam kasus ini, kita mengatur warna latar belakang (`backgroundColor`) menjadi teal[300].

* AppBar

**(`AppBar( ... )`)**

Widget ini adalah bagian dari `Scaffold` yang menampilkan bar navigasi di bagian atas aplikasi. Di sini, kita mengatur judul (`title`) dan properti lain seperti warna teks (`style`) dan warna latar belakang (`backgroundColor`) dari `AppBar`.

* TextStyle

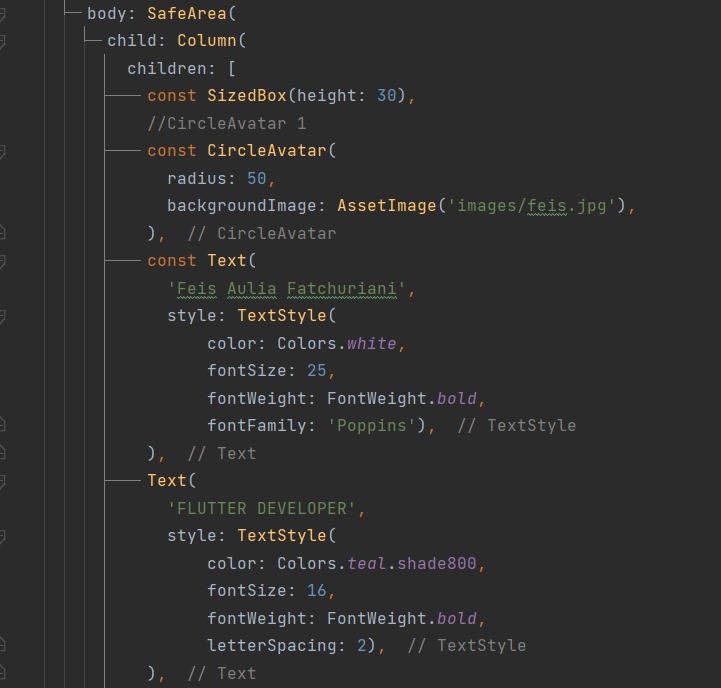
**(`TextStyle( ... )`)**

Kelas ini digunakan untuk menentukan gaya teks seperti warna, tebal, dan jenis font. Di sini, kita mengatur warna teks judul menjadi putih (`Colors.white`), tebal (`FontWeight.bold`), dan menggunakan font 'Poppins' (`fontFamily: 'Poppins'`).

* Colors

**(`Colors.teal[300]`, `Colors.black38`, `Colors.white`)**

Kelas ini menyediakan berbagai warna yang dapat digunakan dalam aplikasi Flutter. Dalam kasus ini, kita menggunakan warna teal[300] untuk latar belakang, black38 untuk latar belakang `AppBar`, dan putih untuk warna teks.



Kode tersebut digunakan untuk membuat tampilan kartu nama dengan informasi pengembang Flutter. Pada tampilan tersebut terdapat gambar profil (CircleAvatar) dengan nama pengembang, jabatan, dan gaya teks yang berbeda untuk setiap elemen. Tampilan ini menggunakan warna teal sebagai latar belakang dan menampilkan informasi dengan gaya yang menarik dan mudah dibaca. Dengan penjelasan kode sebagai berikut:

* SafeArea

**(`SafeArea( ... )`)**

Widget ini digunakan untuk menghindari tumpang tindih dengan area yang tidak aman seperti sensor kamera atau notch pada perangkat. Pada contoh ini, `SafeArea` digunakan sebagai parent dari widget `Column`.

* Column

**(`Column( ... )`)**

Widget ini digunakan untuk menata widget-child secara vertikal. Di dalam `Column`, kita memiliki beberapa widget sebagai children, yaitu `SizedBox`, `CircleAvatar`, dan dua buah `Text`.

* SizedBox

**(`const SizedBox(height: 30)`)**

Widget ini digunakan untuk memberikan spasi vertikal sebesar 30 piksel antara widget sebelumnya (jika ada) dan `CircleAvatar`.

* CircleAvatar

**(`const CircleAvatar( ... )`)**

Widget ini digunakan untuk menampilkan gambar bulat. Di sini, kita menggunakan gambar dari `AssetImage('images/feis.jpg')` sebagai background image.

* Text Widget

**(`const Text(...), Text(...)`)**

Widget ini digunakan untuk menampilkan teks. Pertama, kita memiliki teks 'Feis Aulia Fatchuriani' dengan gaya teks berwarna putih, ukuran font 25, tebal, dan menggunakan font 'Poppins'. Kemudian, kita memiliki teks 'FLUTTER DEVELOPER' dengan gaya teks berwarna teal.shade800, ukuran font 16, tebal, dan spasi antar huruf sebesar 2.

Kode tersebut digunakan untuk membuat kartu kontak sederhana dalam aplikasi Flutter. Kartu tersebut menampilkan nomor telepon dan menggunakan desain Material yang khas, dengan latar belakang putih dan ikon telepon diikuti oleh nomor telepon dalam gaya teks yang menarik. Penggunaan widget seperti Card, Row, SizedBox, Icon, dan Text memungkinkan kita untuk dengan mudah membuat tampilan yang terstruktur dan informatif dalam aplikasi Flutter. Berikut penjelasan bagian-bagian kode:

* Card

**(`const Card( ... )`)**

Widget `Card` digunakan untuk membuat kartu dengan bayangan (elevation) yang memberikan tampilan yang lebih menarik. Di dalam `Card`, kita memiliki properti `color` untuk menentukan warna latar belakang kartu, `margin` untuk memberikan jarak dari kartu ke elemen lain, dan `child` untuk menampilkan konten dalam kartu.

* Padding

**(`Padding( ... )`)**

Widget `Padding` digunakan untuk menambahkan padding (jarak) di sekeliling child widget. Di sini, kita menggunakan padding sebesar 1.0.

* Row

**(Row( ... )`)**

Widget `Row` digunakan untuk menata widget-child secara horizontal. Di dalam `Row`, kita memiliki beberapa widget sebagai children, yaitu `Icon` dan `Text`.

* Icon

**(`Icon( ... )`)**

Widget `Icon` digunakan untuk menampilkan ikon. Pada contoh ini, kita menggunakan ikon telepon (Icons.phone) dengan warna teal, ukuran 30.0.

* SizedBox

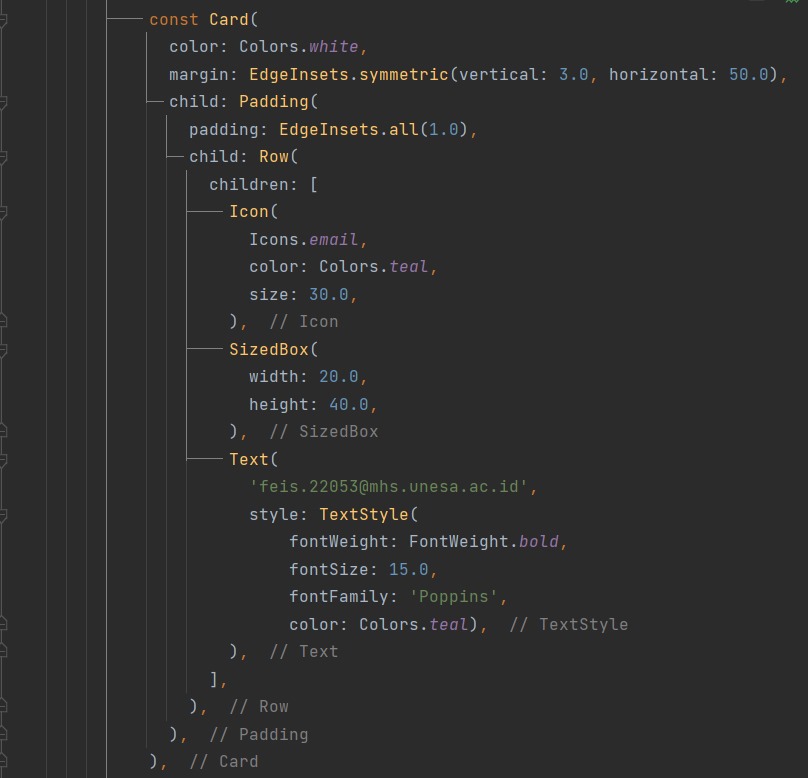
**(`SizedBox( ... )`)**

Widget `SizedBox` digunakan untuk menambahkan spasi berukuran tertentu. Pada contoh ini, kita menggunakan `SizedBox` untuk memberikan spasi horizontal sebesar 20.0 dan spasi vertikal sebesar 40.0 antara ikon dan teks.

* Text

**(`Text( ... )`)**

Widget `Text` digunakan untuk menampilkan teks. Di sini, kita menampilkan nomor telepon '085785702231' dengan gaya teks berwarna teal, tebal, ukuran font 17.0, dan menggunakan font 'Poppins'.



Kode tersebut digunakan untuk membuat kartu kontak sederhana dalam aplikasi Flutter. Kartu tersebut menampilkan email dan menggunakan desain Material yang khas, dengan latar belakang putih dan ikon telepon diikuti oleh email dalam gaya teks yang menarik. Penggunaan widget seperti Card, Row, SizedBox, Icon, dan Text memungkinkan kita untuk dengan mudah membuat tampilan yang terstruktur dan informatif dalam aplikasi Flutter. Berikut penjelasan bagian-bagian kode:

* Card

**(`const Card( ... )`)**

Widget `Card` digunakan untuk membuat kartu dengan bayangan (elevation) yang memberikan tampilan yang lebih menarik. Di dalam `Card`, kita memiliki properti `color` untuk menentukan warna latar belakang kartu, `margin` untuk memberikan jarak dari kartu ke elemen lain, dan `child` untuk menampilkan konten dalam kartu.

* Padding

**(`Padding( ... )`)**

Widget `Padding` digunakan untuk menambahkan padding (jarak) di sekeliling child widget. Di sini, kita menggunakan padding sebesar 1.0.

* Row

**(`Row( ... )`)**

Widget `Row` digunakan untuk menata widget-child secara horizontal. Di dalam `Row`, kita memiliki beberapa widget sebagai children, yaitu `Icon` dan `Text`.

* Icon

**(`Icon( ... )`)**

Widget `Icon` digunakan untuk menampilkan ikon. Pada contoh ini, kita menggunakan ikon email (Icons.email) dengan warna teal, ukuran 30.0.

* SizedBox

**(`SizedBox( ... )`)**

Widget `SizedBox` digunakan untuk menambahkan spasi berukuran tertentu. Pada contoh ini, kita menggunakan `SizedBox` untuk memberikan spasi horizontal sebesar 20.0 dan spasi vertikal sebesar 40.0 antara ikon dan teks.

* Text

**(`Text( ... )`)**

Widget `Text` digunakan untuk menampilkan teks. Di sini, kita menampilkan alamat email 'feis.22053@mhs.unesa.ac.id' dengan gaya teks berwarna teal, tebal, ukuran font 15.0, dan menggunakan font 'Poppins'.



Kode tersebut digunakan untuk membuat kartu (Card) kedua dalam aplikasi Flutter yang menampilkan informasi tentang seorang pengembang Flutter. Kartu tersebut berisi gambar profil (CircleAvatar) dengan gambar 'berlin.jpg', nama pengembang, dan jabatan pengembang. Berikut penjelasan bagian-bagian kode:

* CircleAvatar

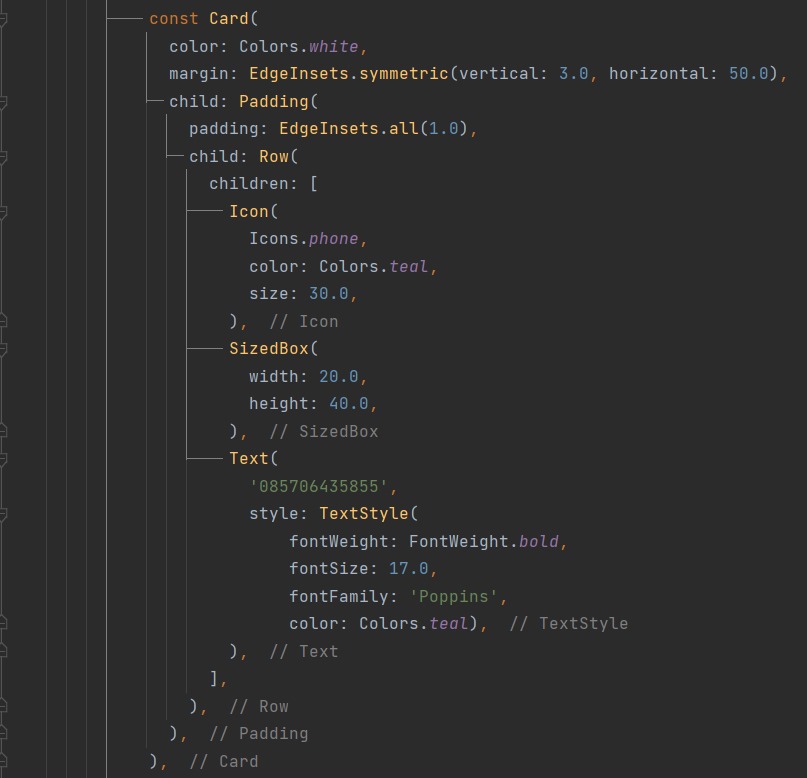
**(`const CircleAvatar( ... )`)**

Widget `CircleAvatar` digunakan untuk menampilkan gambar bulat. Pada contoh ini, kita menggunakan gambar dengan path 'images/berlin.jpg' sebagai background image dengan radius 50.

* Text

**(`const Text( ... )` dan `Text( ... )`)**

Widget `Text` digunakan untuk menampilkan teks. Dua blok `Text` digunakan untuk menampilkan nama 'Berlin Marsyah Yustina' dengan gaya teks berwarna putih, ukuran font 25, tebal, dan menggunakan font 'Poppins'. Kemudian, kita memiliki teks 'FLUTTER DEVELOPER' dengan gaya teks berwarna teal.shade800, ukuran font 16, tebal, dan spasi antar huruf sebesar 2.



Kode tersebut digunakan untuk membuat kartu kontak sederhana dalam aplikasi Flutter. Kartu tersebut menampilkan nomor telepon dan menggunakan desain Material yang khas, dengan latar belakang putih dan ikon telepon diikuti oleh nomor telepon dalam gaya teks yang menarik. Penggunaan widget seperti Card, Row, SizedBox, Icon, dan Text memungkinkan kita untuk dengan mudah membuat tampilan yang terstruktur dan informatif dalam aplikasi Flutter. Berikut penjelasan bagian-bagian kode:

* Card

**(const Card( ... )`)**

Widget `Card` digunakan untuk membuat kartu dengan bayangan (elevation) yang memberikan tampilan yang lebih menarik. Di dalam `Card`, kita memiliki properti `color` untuk menentukan warna latar belakang kartu, `margin` untuk memberikan jarak dari kartu ke elemen lain, dan `child` untuk menampilkan konten dalam kartu.

* Padding

**(Padding( ... )`)**

Widget `Padding` digunakan untuk menambahkan padding (jarak) di sekeliling child widget. Di sini, kita menggunakan padding sebesar 1.0.

* Row

**(Row( ... )`)**

Widget `Row` digunakan untuk menata widget-child secara horizontal. Di dalam `Row`, kita memiliki beberapa widget sebagai children, yaitu `Icon` dan `Text`.

* Icon

**(Icon( ... )`)**

Widget `Icon` digunakan untuk menampilkan ikon. Pada contoh ini, kita menggunakan ikon telepon (Icons.phone) dengan warna teal, ukuran 30.0.

* SizedBox

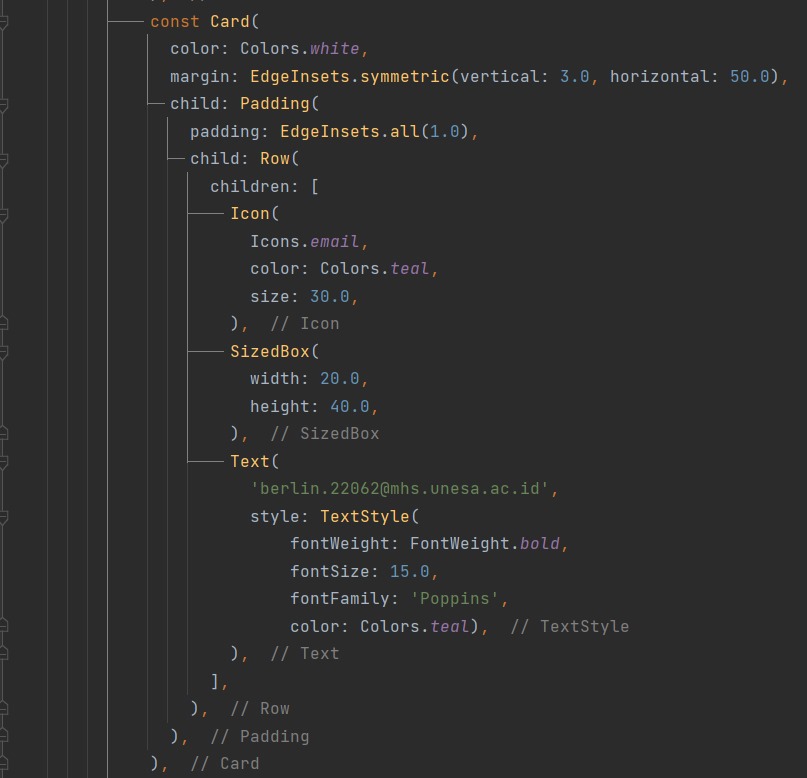
**(SizedBox( ... )`)**

Widget `SizedBox` digunakan untuk menambahkan spasi berukuran tertentu. Pada contoh ini, kita menggunakan `SizedBox` untuk memberikan spasi horizontal sebesar 20.0 dan spasi vertikal sebesar 40.0 antara ikon dan teks.

* Text

**(Text( ... )`)**

Widget `Text` digunakan untuk menampilkan teks. Di sini, kita menampilkan nomor telepon '085706435855' dengan gaya teks berwarna teal, tebal, ukuran font 17.0, dan menggunakan font 'Poppins'.



Kode tersebut digunakan untuk membuat kartu kontak sederhana dalam aplikasi Flutter. Kartu tersebut menampilkan email dan menggunakan desain Material yang khas, dengan latar belakang putih dan ikon telepon diikuti oleh email dalam gaya teks yang menarik. Penggunaan widget seperti Card, Row, SizedBox, Icon, dan Text memungkinkan kita untuk dengan mudah membuat tampilan yang terstruktur dan informatif dalam aplikasi Flutter. Berikut penjelasan bagian-bagian kode:

* Card

**(‘const Card( ... )`)**

Widget `Card` digunakan untuk membuat kartu dengan bayangan (elevation) yang memberikan tampilan yang lebih menarik. Di dalam `Card`, kita memiliki properti `color` untuk menentukan warna latar belakang kartu, `margin` untuk memberikan jarak dari kartu ke elemen lain, dan `child` untuk menampilkan konten dalam kartu.

* Padding

**(`Padding( ... )`)**

Widget `Padding` digunakan untuk menambahkan padding (jarak) di sekeliling child widget. Di sini, kita menggunakan padding sebesar 1.0.

* Row

**(`Row( ... )`)**

Widget `Row` digunakan untuk menata widget-child secara horizontal. Di dalam `Row`, kita memiliki beberapa widget sebagai children, yaitu `Icon` dan `Text`.

* Icon

**(`Icon( ... )`)**

Widget `Icon` digunakan untuk menampilkan ikon. Pada contoh ini, kita menggunakan ikon email (Icons.email) dengan warna teal, ukuran 30.0.

* SizedBox

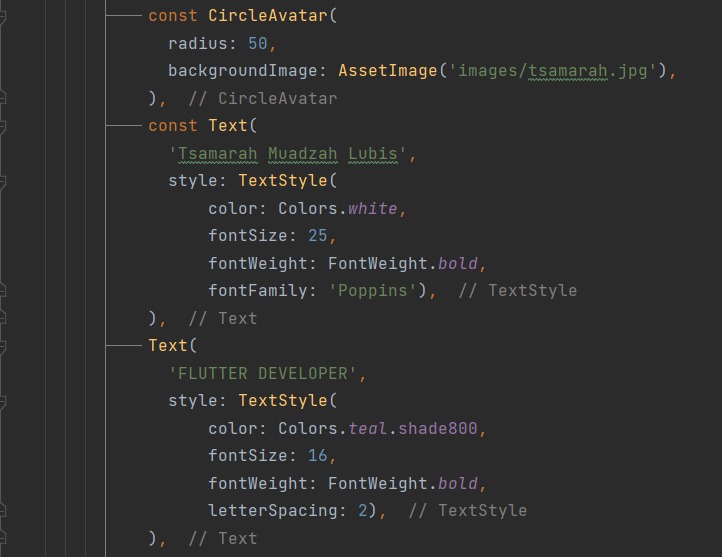
**(`SizedBox( ... )`)**

Widget `SizedBox` digunakan untuk menambahkan spasi berukuran tertentu. Pada contoh ini, kita menggunakan `SizedBox` untuk memberikan spasi horizontal sebesar 20.0 dan spasi vertikal sebesar 40.0 antara ikon dan teks.

* Text

**(`Text( ... )`)**

Widget `Text` digunakan untuk menampilkan teks. Di sini, kita menampilkan alamat email ‘berlin.22062@mhs.unesa.ac.id' dengan gaya teks berwarna teal, tebal, ukuran font 15.0, dan menggunakan font 'Poppins'.



Kode tersebut digunakan untuk membuat kartu (Card) ketiga dalam aplikasi Flutter yang menampilkan informasi tentang seorang pengembang Flutter. Kartu tersebut berisi gambar profil (CircleAvatar) dengan gambar 'tsamarah.jpg', nama pengembang, dan jabatan pengembang. Berikut penjelasan bagian-bagian kode:

* CircleAvatar

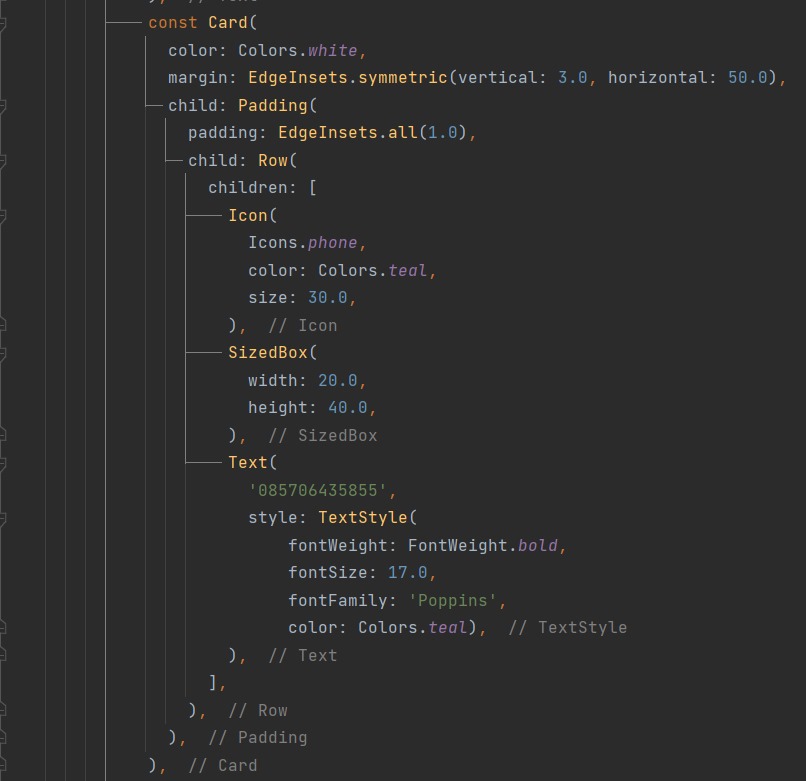
**(`const CircleAvatar( ... )`)**

Widget `CircleAvatar` digunakan untuk menampilkan gambar bulat. Pada contoh ini, kita menggunakan gambar dengan path 'images/tsamarah.jpg' sebagai background image dengan radius 50.

* Text

**(`const Text( ... )` dan `Text( ... )`)**

Widget `Text` digunakan untuk menampilkan teks. Dua blok `Text` digunakan untuk menampilkan nama 'Tsamarah Muadzah Lubis' dengan gaya teks berwarna putih, ukuran font 25, tebal, dan menggunakan font 'Poppins'. Kemudian, kita memiliki teks 'FLUTTER DEVELOPER' dengan gaya teks berwarna teal.shade800, ukuran font 16, tebal, dan spasi antar huruf sebesar 2.



Kode tersebut digunakan untuk membuat kartu kontak sederhana dalam aplikasi Flutter. Kartu tersebut menampilkan nomor telepon dan menggunakan desain Material yang khas, dengan latar belakang putih dan ikon telepon diikuti oleh nomor telepon dalam gaya teks yang menarik. Penggunaan widget seperti Card, Row, SizedBox, Icon, dan Text memungkinkan kita untuk dengan mudah membuat tampilan yang terstruktur dan informatif dalam aplikasi Flutter. Berikut penjelasan bagian-bagian kode:

* Card

**(const Card( ... )`)**

Widget `Card` digunakan untuk membuat kartu dengan bayangan (elevation) yang memberikan tampilan yang lebih menarik. Di dalam `Card`, kita memiliki properti `color` untuk menentukan warna latar belakang kartu, `margin` untuk memberikan jarak dari kartu ke elemen lain, dan `child` untuk menampilkan konten dalam kartu.

* Padding

**(Padding( ... )`)**

Widget `Padding` digunakan untuk menambahkan padding (jarak) di sekeliling child widget. Di sini, kita menggunakan padding sebesar 1.0.

* Row

**(Row( ... )`)**

Widget `Row` digunakan untuk menata widget-child secara horizontal. Di dalam `Row`, kita memiliki beberapa widget sebagai children, yaitu `Icon` dan `Text`.

* Icon

**(Icon( ... )`)**

Widget `Icon` digunakan untuk menampilkan ikon. Pada contoh ini, kita menggunakan ikon telepon (Icons.phone) dengan warna teal, ukuran 30.0.

* SizedBox

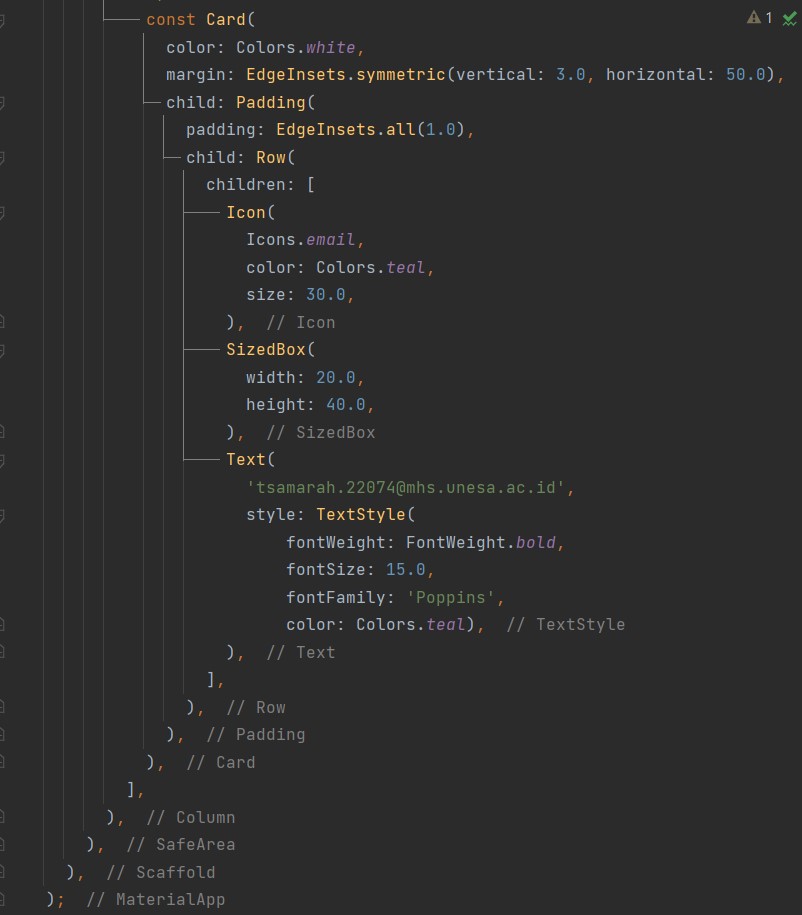
**(SizedBox( ... )`)**

Widget `SizedBox` digunakan untuk menambahkan spasi berukuran tertentu. Pada contoh ini, kita menggunakan `SizedBox` untuk memberikan spasi horizontal sebesar 20.0 dan spasi vertikal sebesar 40.0 antara ikon dan teks.

* Text

**(Text( ... )`)**

Widget `Text` digunakan untuk menampilkan teks. Di sini, kita menampilkan nomor telepon '085706435855' dengan gaya teks berwarna teal, tebal, ukuran font 17.0, dan menggunakan font 'Poppins'.



Kode tersebut digunakan untuk membuat kartu kontak sederhana dalam aplikasi Flutter. Kartu tersebut menampilkan email dan menggunakan desain Material yang khas, dengan latar belakang putih dan ikon telepon diikuti oleh email dalam gaya teks yang menarik. Penggunaan widget seperti Card, Row, SizedBox, Icon, dan Text memungkinkan kita untuk dengan mudah membuat tampilan yang terstruktur dan informatif dalam aplikasi Flutter. Berikut penjelasan bagian-bagian kode:

* Card

**(‘const Card( ... )`)**

Widget `Card` digunakan untuk membuat kartu dengan bayangan (elevation) yang memberikan tampilan yang lebih menarik. Di dalam `Card`, kita memiliki properti `color` untuk menentukan warna latar belakang kartu, `margin` untuk memberikan jarak dari kartu ke elemen lain, dan `child` untuk menampilkan konten dalam kartu.

* Padding

**(`Padding( ... )`)**

Widget `Padding` digunakan untuk menambahkan padding (jarak) di sekeliling child widget. Di sini, kita menggunakan padding sebesar 1.0.

* Row

**(`Row( ... )`)**

Widget `Row` digunakan untuk menata widget-child secara horizontal. Di dalam `Row`, kita memiliki beberapa widget sebagai children, yaitu `Icon` dan `Text`.

* Icon

**(`Icon( ... )`)**

Widget `Icon` digunakan untuk menampilkan ikon. Pada contoh ini, kita menggunakan ikon email (Icons.email) dengan warna teal, ukuran 30.0.

* SizedBox

**(`SizedBox( ... )`)**

Widget `SizedBox` digunakan untuk menambahkan spasi berukuran tertentu. Pada contoh ini, kita menggunakan `SizedBox` untuk memberikan spasi horizontal sebesar 20.0 dan spasi vertikal sebesar 40.0 antara ikon dan teks.

* Text

**(`Text( ... )`)**

Widget `Text` digunakan untuk menampilkan teks. Di sini, kita menampilkan alamat email ‘tsamarah.22074@mhs.unesa.ac.id' dengan gaya teks berwarna teal, tebal, ukuran font 15.0, dan menggunakan font 'Poppins'.

* **Source Code Pada Folder pubspec.yaml’**



Kode tersebut merupakan bagian dari file `pubspec.yaml` dalam proyek Flutter. File `pubspec.yaml` digunakan untuk mengelola dependensi, aset, dan konfigurasi proyek Flutter.

* **(uses-material-design)**

Baris ini menunjukkan bahwa proyek menggunakan Material Design, yang berarti kita dapat menggunakan ikon-ikon dari kelas `Icons` yang disediakan oleh Flutter.

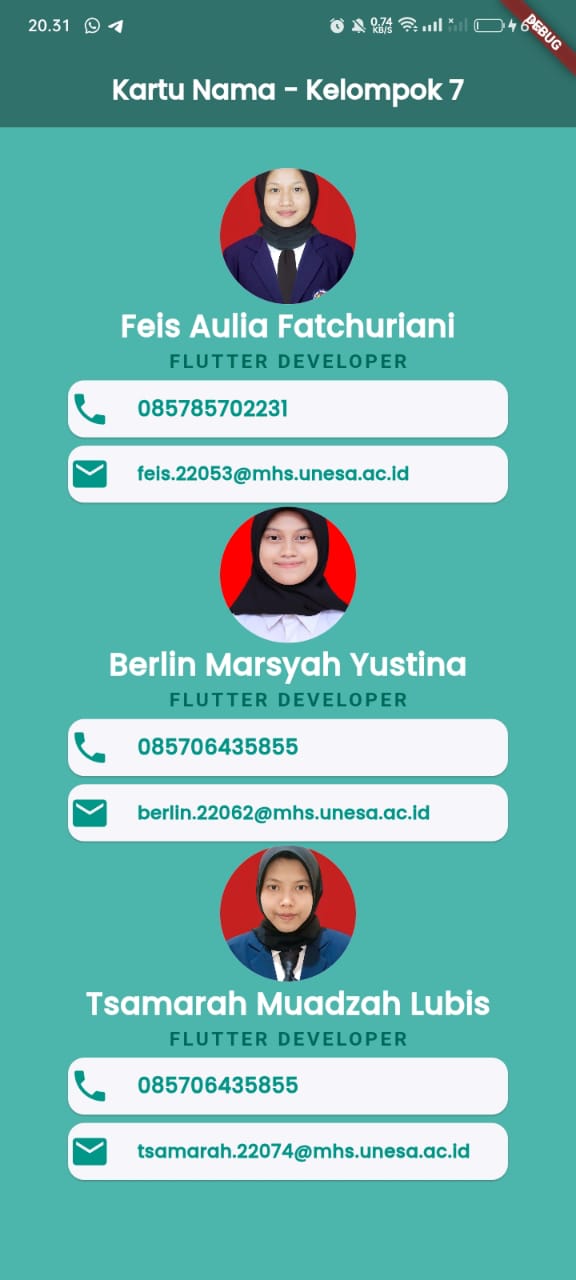
* **(assets)**

Bagian ini digunakan untuk menambahkan aset ke proyek. Dalam contoh ini, semua file di dalam folder `images/` akan dianggap sebagai aset.

* **(fonts)**

Bagian ini digunakan untuk menambahkan font kustom ke proyek. Dalam contoh ini, font kustom dengan nama "Poppins" ditambahkan. Font ini memiliki satu varian yang diambil dari file `Poppins-Medium.ttf` yang berada di dalam folder `fonts/`.

Dengan konfigurasi ini, proyek dapat menggunakan ikon Material Icons, mengakses aset dari folder `images/`, dan menggunakan font kustom "Poppins" dalam aplikasi Flutter.

* **Tampilan Aplikasi Kartu Nama**
* **Link Github**

**https://github.com/feisauu/aplikasi\_kartu\_nama**