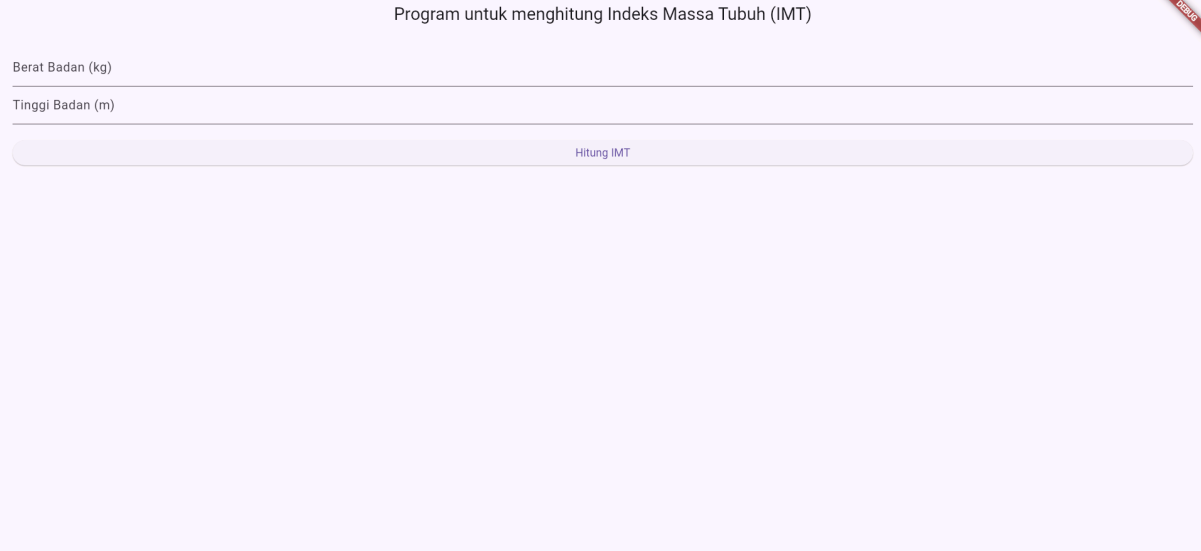


Nama : Yacobus Daeli
NIM : H1D022024
Shift : Awal (A) - Sekarang (E)

Program untuk menghitung Indeks Massa Tubuh (IMT)



Penjelasan code dibawah ini :

1. `import 'package:flutter/material.dart';`
 - Perintah diatas untuk mengimport flutter/material.dart ke project. Package ini berisi widget dan tema UI dasar berbasis Material Design yang digunakan dalam aplikasi Flutter.
2. `void main() { runApp(BMICalculator()); }`
 - Fungsi main() adalah titik masuk utama aplikasi Flutter. Fungsi ini memanggil runApp(), yang menjalankan aplikasi dengan menampilkan widget root (dalam hal ini, BMICalculator).
3. `class BMICalculator extends StatelessWidget { ... }`
 - BMICalculator adalah kelas utama aplikasi yang merupakan turunan dari StatelessWidget.
 - StatelessWidget adalah widget yang tidak berubah selama masa hidup aplikasi. Ini artinya, widget ini tidak memiliki state (data yang dapat berubah), dan hanya berfungsi untuk menampilkan UI statis.
4. `@override Widget build(BuildContext context) { ... }`

- Fungsi `build()` bertanggung jawab untuk membangun UI dari widget ini. Setiap widget dalam Flutter harus mendefinisikan metode `build()` untuk menampilkan komponen visualnya.
- Parameter `context` adalah objek yang menyimpan informasi tentang lokasi widget ini di dalam tree widget.

5. `return MaterialApp(...);`

- `MaterialApp` adalah widget utama dari aplikasi Flutter berbasis Material Design. Ini menyediakan berbagai fitur seperti tema, navigasi, dan banyak lagi.
- Di sini, `title` adalah judul aplikasi yang ditampilkan pada task manager atau saat pengguna melihat daftar aplikasi terbuka.
- `theme` menentukan tampilan aplikasi secara keseluruhan. Dalam hal ini, warna utama aplikasi adalah biru (`Colors.blue`).
- `home` adalah halaman utama yang ditampilkan oleh aplikasi saat pertama kali diluncurkan, dalam hal ini adalah `BMICalculatorHome`.

6. `class BMICalculatorHome extends StatefulWidget { ... }`

- `BMICalculatorHome` adalah kelas widget yang `stateful`. Berbeda dengan `StatelessWidget`, widget ini dapat berubah berdasarkan interaksi pengguna.
- `StatefulWidget` memiliki `state`, yang memungkinkan widget ini untuk berubah dan menampilkan UI yang berbeda saat data atau `state` berubah.

7. `class _BMICalculatorHomeState extends State<BMICalculatorHome> { ... }`

- `_BMICalculatorHomeState` adalah kelas yang mengelola `state` dari `BMICalculatorHome`. Tanda underscore (`_`) menunjukkan bahwa kelas ini bersifat privat dan hanya dapat diakses dalam file ini.
- `State<BMICalculatorHome>` berarti kelas ini memperluas `state` dari widget `BMICalculatorHome`.

8. `TextEditingController _weightController = TextEditingController();`

- `TextEditingController` digunakan untuk mengontrol dan melacak teks yang dimasukkan pengguna ke dalam `TextField`.
- Dalam hal ini, `_weightController` digunakan untuk menangani input berat badan.

9. `void _calculateBMI() { ... }`

- Fungsi `_calculateBMI()` digunakan untuk menghitung nilai Indeks Massa Tubuh (IMT) berdasarkan input pengguna (berat dan tinggi badan).

- Pertama, berat dan tinggi badan diambil dari TextField menggunakan TextEditingController dan diubah menjadi double.
- Rumus IMT diterapkan: $IMT = \text{berat} / (\text{tinggi} * \text{tinggi})$.
- Hasilnya akan menentukan kategori (berat badan kurang, normal, berlebih, obesitas) berdasarkan nilai IMT.
- setState() digunakan untuk memperbarui UI dengan hasil IMT yang baru.

10. @override Widget build(BuildContext context) { ... }

- Deskripsi: Fungsi build() dalam kelas _BMICalculatorHomeState digunakan untuk membangun UI dari halaman kalkulator IMT.

11. return Scaffold(...);

- Scaffold adalah widget dasar untuk membangun struktur tata letak Material Design. Ini menyediakan berbagai komponen UI seperti AppBar, Drawer, FloatingActionButton, dan lainnya.
- Dalam aplikasi ini, Scaffold digunakan untuk menampilkan AppBar di bagian atas dan elemen UI lainnya dalam body.

12. AppBar(title: Text('Program untuk menghitung Indeks Massa Tubuh (IMT)'), centerTitle: true,)

- AppBar adalah bilah alat yang berada di bagian atas halaman aplikasi.
- title adalah teks yang ditampilkan di AppBar, dan centerTitle: true memastikan bahwa teks tersebut berada di tengah.

13. body: Padding(...)

- Padding memberikan jarak (padding) di sekitar isi widget. Dalam hal ini, padding diterapkan dengan jarak 16 piksel di semua sisi (EdgeInsets.all(16.0)).

14. Column(crossAxisAlignment: CrossAxisAlignment.stretch, children: <Widget>[...])

- Column adalah widget yang menyusun anak-anaknya secara vertikal. Dalam hal ini, widget TextField, ElevatedButton, dan Text diatur secara vertikal.
- crossAxisAlignment: CrossAxisAlignment.stretch digunakan untuk membuat setiap elemen dalam kolom meregang sepanjang lebar kolom.

15. TextField(controller: _weightController, ...)

- TextField adalah widget input teks yang memungkinkan pengguna memasukkan data. Dalam hal ini, widget ini digunakan untuk mengambil input berat dan tinggi badan.

- controller menghubungkan input pengguna ke TextEditingController yang sudah dideklarasikan.

16. `ElevatedButton(onPressed: _calculateBMI, child: Text('Hitung IMT'),)`

- ElevatedButton adalah tombol yang digunakan untuk menjalankan fungsi. Ketika tombol ini ditekan (onPressed), fungsi `_calculateBMI()` dipanggil untuk menghitung IMT.

17. `Text(_result, style: TextStyle(fontSize: 24), textAlign: TextAlign.center,)`

- Text digunakan untuk menampilkan hasil perhitungan IMT. Nilai yang ditampilkan berasal dari variabel `_result`, yang diubah melalui `setState()` saat IMT dihitung.
- `textAlign: TextAlign.center` memastikan teks hasil ditampilkan di tengah layar.