Nama : Kintan Kinasih Mahaputri

NIM : H1D022019

Shift Lama : D

Shift Baru : C

Code Main:

```
import 'package:flutter/material.dart';
import 'package:pertemuan1/home_page.dart';

void main() {
  runApp(const MyApp());
}

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Calculator',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const HomePage(),
    );
  }
```

}

Penjelasan Code Main :

- Import flutter/material.dart: Mengimpor pustaka material Flutter, yang menyediakan komponen dasar antarmuka pengguna berbasis desain Material (seperti Scaffold, AppBar, Text, dll.).
- Import pertemuan1/home_page.dart: Mengimpor file eksternal bernama home_page.dart dari package pertemuan1. Ini mengindikasikan bahwa halaman utama (HomePage) didefinisikan di file lain.
- main() function: Fungsi ini adalah entry point dari aplikasi Flutter. Saat aplikasi dijalankan, Flutter memanggil fungsi ini.
- runApp(): Fungsi ini digunakan untuk meluncurkan aplikasi. Dalam hal ini, aplikasi dimulai dengan widget MyApp.
- const MyApp(): Membuat instance dari widget MyApp secara const (konstan), yang berarti nilai dari widget ini tidak akan berubah.
- class MyApp extends StatelessWidget: Definisi kelas MyApp yang merupakan subclass dari StatelessWidget. StatelessWidget adalah widget yang tidak dapat berubah setelah dibuat (tidak memiliki state).
- const MyApp({Key? key}): Konstruktor kelas MyApp dengan parameter opsional key. super(key: key) digunakan untuk memanggil konstruktor superclass (StatelessWidget).
- @override: Mengindikasikan bahwa fungsi build() di-override dari superclass StatelessWidget.
- Widget build(BuildContext context): Fungsi build() ini mendefinisikan UI dari widget MyApp. Fungsi ini akan dipanggil ketika widget dibangun. Parameter BuildContext context menyimpan informasi mengenai lokasi widget di dalam tree widget.
- return MaterialApp(: MaterialApp adalah widget inti yang mengimplementasikan desain Material dalam aplikasi. Ini mengatur tema, struktur navigasi, dan halaman utama aplikasi.
- debugShowCheckedModeBanner: false: Menghilangkan banner debug kecil (label "DEBUG") yang biasanya muncul di pojok kanan atas aplikasi ketika sedang dalam mode debug.
- itle: 'Calculator': Menetapkan judul aplikasi. Judul ini bisa muncul di task manager perangkat atau saat multitasking.
- theme: ThemeData(: Mengatur tema aplikasi. Dalam hal ini, tema menggunakan warna utama (primarySwatch) biru.
- ThemeData: Ini adalah struktur yang menyimpan informasi mengenai tema aplikasi, seperti warna, font, dan gaya visual lainnya.
- home: const HomePage(): home mendefinisikan halaman utama aplikasi, yaitu widget HomePage. Ini adalah halaman pertama yang akan ditampilkan ketika aplikasi diluncurkan. HomePage diimpor dari file home_page.dart dan diinstansiasi sebagai const, yang berarti tidak akan berubah.
- enutup fungsi dan kelas: Menutup fungsi build() dan kelas MyApp.

Code :

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

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  double result = 0;
  final TextEditingController firstController = TextEditingController();
  final TextEditingController secondController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Calculator App'),
      ),
      body: Container(
        padding: const EdgeInsets.all(20),
        child: Column(
          children: [
            const SizedBox(
              height: 10,
```

```
),
TextField(
  controller: firstController,
  decoration: const InputDecoration(
    border: OutlineInputBorder(),
    hintText: 'First Number',
  ),
  keyboardType: TextInputType.number,
),
const SizedBox(
  height: 10,
),
TextField(
  controller: secondController,
  decoration: const InputDecoration(
    border: OutlineInputBorder(),
    hintText: 'Second Number',
  ),
  keyboardType: TextInputType.number,
),
const SizedBox(
  height: 10,
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    ElevatedButton(
      onPressed: () {
```

```
    double a = double.parse(firstController.text);
    double b = double.parse(secondController.text);

    setState(() {
      result = a + b;
    });
  },
  child: const Text('Add'),
),
ElevatedButton(
  onPressed: () {
    double a = double.parse(firstController.text);
    double b = double.parse(secondController.text);

    setState(() {
      result = a - b;
    });
  },
  child: const Text('Subtract'),
),
ElevatedButton(
  onPressed: () {
    double a = double.parse(firstController.text);
    double b = double.parse(secondController.text);

    setState(() {
      result = a * b;
    });
```

```
            },
            child: const Text('Multiply'),
          ),
          ElevatedButton(
            onPressed: () {
              double a = double.parse(firstController.text);
              double b = double.parse(secondController.text);

              setState(() {
                result = a / b;
              });
            },
            child: const Text('Divide'),
          ),
        ],
      ),
      const SizedBox(
        height: 20,
      ),
      Text(
        'Result: $result',
        style: const TextStyle(fontSize: 20),
      ),
    ],
  ),
 ),
);
}
```

}

Penjelasan Code :

- Struktur Utama
  Aplikasi ini menggunakan dua kelas:
- HomePage: Ini adalah `StatefulWidget`, yang berarti aplikasi ini memiliki state yang dapat berubah selama siklus hidup widget. `HomePage` adalah halaman utama dari aplikasi kalkulator, dan di dalamnya terdapat logika serta antarmuka pengguna.
  HomePageState: Ini adalah kelas yang berhubungan dengan `HomePage` dan mengelola state yang ada. Di sini, perhitungan kalkulator dan tampilan UI diimplementasikan.
- Deklarasi Variabel
  - double result = 0`: Variabel ini digunakan untuk menyimpan hasil dari operasi kalkulator.
  - TextEditingController firstController` dan `secondController: Kedua kontroler ini digunakan untuk mengontrol input dari pengguna dalam `TextField`. `firstController` untuk angka pertama, dan `secondController` untuk angka kedua.
- Fungsi `build` dan Struktur Antarmuka
  Pada fungsi `build()`, UI aplikasi didefinisikan. Flutter menggunakan fungsi ini untuk merender tampilan ketika aplikasi dibuka atau ada perubahan state. Berikut penjelasan elemen-elemennya:
  - Scaffold: Ini adalah widget dasar untuk halaman dengan struktur Material Design. `Scaffold` menyediakan komponen seperti `AppBar` dan `body` yang merupakan area tampilan utama.
  - AppBar: Merupakan bagian atas aplikasi yang menampilkan judul "Calculator App". Ini memberi pengguna konteks tentang aplikasi yang mereka gunakan.
  - Container: Ini adalah pembungkus untuk semua widget dalam aplikasi, dan di sini diberikan padding sebesar 20 pixel agar ada ruang di sekitar elemen-elemen UI.
  - Column: Semua elemen di dalam body ditempatkan dalam sebuah kolom vertikal. Anak-anak dari kolom ini termasuk `SizedBox`, `TextField`, `Row`, dan `Text`.
- Input Angka
  - TextField: Ada dua `TextField` yang digunakan untuk memasukkan angka pertama dan angka kedua. Setiap `TextField` memiliki `TextEditingController` (yaitu `firstController` dan `secondController`) untuk menangkap input dari pengguna. Selain itu, `keyboardType: TextInputType.number` memastikan bahwa keyboard numerik muncul saat pengguna berinteraksi dengan input.
- Operasi Kalkulator
  - Row: Baris ini mengatur empat tombol (Add, Subtract, Multiply, Divide) secara horizontal dan memberikan jarak yang merata antara tombol-tombol tersebut.
  - Setiap tombol menggunakan `ElevatedButton` dan memiliki teks yang menunjukkan operasi yang akan dilakukan.

- Ketika salah satu tombol ditekan, angka dari `TextField` pertama dan kedua diambil, dan operasi matematika (penambahan, pengurangan, perkalian, atau pembagian) dilakukan.
- setState(): Fungsi ini digunakan untuk memberitahu Flutter bahwa state aplikasi telah berubah, dan tampilan perlu diperbarui. Dalam hal ini, variabel `result` diperbarui dengan hasil dari operasi aritmatika, sehingga aplikasi dapat menampilkan hasil terbaru.

- Menampilkan Hasil

  Text: Pada bagian akhir UI, hasil dari operasi aritmatika ditampilkan dalam widget `Text` dengan teks `Result: $result`. Setiap kali operasi dijalankan dan hasilnya diubah, teks ini akan diperbarui untuk mencerminkan hasil baru.

- Kesimpulan

  Secara keseluruhan, aplikasi ini adalah kalkulator dasar yang menampilkan dua input angka dan empat operasi matematika dasar: penambahan, pengurangan, perkalian, dan pembagian. UI dibuat sederhana dengan dua inputan angka dan beberapa tombol operasi, serta hasil operasi ditampilkan di bagian bawah layar. Flutter secara efisien menangani pembaruan UI dengan menggunakan `setState()` setiap kali ada perubahan state, seperti hasil kalkulasi yang baru.

Screenshoot hasil: