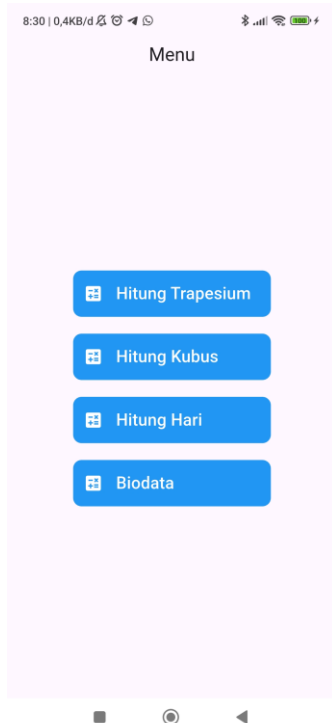


NAMA : MADE VIDYATMA ADHI KRISNA

NIM : 123220038

LAPORAN QUIZ TPM

1. Menu



```
class MenuPage extends StatelessWidget {  
  const MenuPage({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: const Text('Menu'), centerTitle: true),  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            Container(  
              width: MediaQuery.of(context).size.width * 0.65,  
              height: 75,  
              padding: const EdgeInsets.all(10),  
              child: TextButton(  
                style: TextButton.styleFrom(  
                  shape: RoundedRectangleBorder(  
                    borderRadius: BorderRadius.circular(10),
```

```

    ),
    foregroundColor: Colors.white,
    backgroundColor: Colors.blue,
  ),
  child: Row(
    spacing: 15,
    children: [
      Icon(Icons.calculate, size: 24, color: Colors.white),
      const Text(
        'Hitung Trapesium',
        style: TextStyle(fontSize: 20),
      ),
    ],
  ),
  onPressed: () {
    Navigator.pushNamed(context, RouteName.trapesium);
  },
),
Container(
  width: MediaQuery.of(context).size.width * 0.65,
  height: 75,
  padding: const EdgeInsets.all(10),
  child: TextButton(
    style: TextButton.styleFrom(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
      ),
      foregroundColor: Colors.white,
      backgroundColor: Colors.blue,
    ),
    child: Row(
      spacing: 15,
      children: [
        Icon(Icons.calculate, size: 24, color: Colors.white),
        const Text('Hitung Kubus', style: TextStyle(fontSize: 20)),
      ],
    ),
    onPressed: () {
      Navigator.pushNamed(context, RouteName.kubus);
    },
  ),
),

```

```

Container(
  width: MediaQuery.of(context).size.width * 0.65,
  height: 75,
  padding: const EdgeInsets.all(10),
  child: TextButton(
    style: TextButton.styleFrom(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
      ),
      foregroundColor: Colors.white,
      backgroundColor: Colors.blue,
    ),
    child: Row(
      spacing: 15,
      children: [
        Icon(Icons.calculate, size: 24, color: Colors.white),
        const Text('Hitung Hari', style: TextStyle(fontSize: 20)),
      ],
    ),
    onPressed: () {
      Navigator.pushNamed(context, RouteName.hari);
    },
  ),
),
Container(
  width: MediaQuery.of(context).size.width * 0.65,
  height: 75,
  padding: const EdgeInsets.all(10),
  child: TextButton(
    style: TextButton.styleFrom(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
      ),
      foregroundColor: Colors.white,
      backgroundColor: Colors.blue,
    ),
    child: Row(
      spacing: 15,
      children: [
        Icon(Icons.calculate, size: 24, color: Colors.white),
        const Text('Biodata', style: TextStyle(fontSize: 20)),
      ],
    ),
  ),
),

```

```

        onPressed: () {
          Navigator.pushNamed(context, RouteName.biodata);
        },
      ),
    ],
  ),
);
}
}

```

PENJELASAN:

Kode di atas, menampilkan halaman menu dengan beberapa tombol yang masing-masing berfungsi untuk menavigasi ke halaman tertentu. Saat tombol "Hitung Trapesium" ditekan, aplikasi akan berpindah ke halaman perhitungan trapesium. Tombol "Hitung Kubus" akan membawa pengguna ke halaman perhitungan kubus, sedangkan tombol "Hitung Hari" akan menuju halaman perhitungan hari. Terakhir, tombol "Biodata" akan mengarahkan pengguna ke halaman biodata. Semua tombol memiliki desain seragam dengan warna biru, teks putih, dan ikon di sampingnya. Namun, ada kesalahan kecil pada properti spacing yang seharusnya diganti dengan `SizedBox(width: 15)`.

Terdapat class **RouteName** untuk menyimpan nama routing nya, dan **RoutePages** untuk menyimpan class pages nya sesuai route named nya

```

class RouteName {
  static const String menu = '/menu';
  static const String trapesium = '/trapesium';
  static const String kubus = '/kubus';
  static const String hari = '/hari';
  static const String biodata = '/biodata';
}

```

```

class RoutePages {
  static Route<dynamic>? generateRoute(RouteSettings settings) {
    switch (settings.name) {
      case RouteName.menu:
        return MaterialPageRoute(builder: (_) => const MenuPage());
      case RouteName.trapesium:
        return MaterialPageRoute(builder: (_) => const TrapesiumPage());
      case RouteName.kubus:
        return MaterialPageRoute(builder: (_) => const KubusPage());
      case RouteName.hari:
        return MaterialPageRoute(builder: (_) => const HitungHari());
      case RouteName.biodata:
        return MaterialPageRoute(builder: (_) => const BiodataPage());
      default:
        return MaterialPageRoute(
          builder: (_) => Scaffold(
            body: Center(child: Text('404 - Page Not Found')),
          ), // Scaffold
        ); // MaterialPageRoute
    }
  }
}

```

2. Textfield

Textfielddya menggunakan widget buatan sendiri, jadi menghasilkan text field yang standar untuk semua input di setiap halaman nantinya

```

TextfieldWidget(
  controller: controller.tinggiController,
  hintText: 'Masukkan tinggi',
), // TextfieldWidget

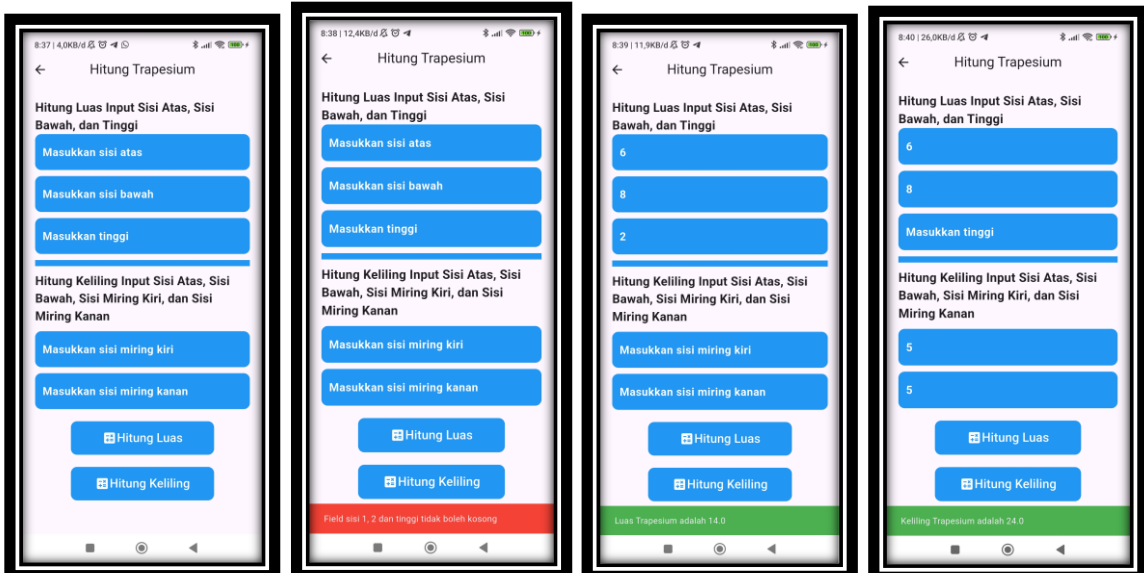
```

```

class TextFieldWidget extends StatelessWidget {
  final TextEditingController controller;
  final String hintText;
  const TextFieldWidget({super.key, required this.controller, required this.hintText});
  @override
  Widget build(BuildContext context) {
    return TextField(
      keyboardType: TextInputType.number,
      controller: controller,
      style: TextStyle(
        fontSize: 18.0,
        color: Colors.white,
        fontWeight: FontWeight.bold,
      ), // TextStyle
      decoration: InputDecoration(
        filled: true,
        fillColor: Colors.blue,
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10.0),
          borderSide: BorderSide.none,
        ), // OutlineInputBorder
        hintText: hintText,
        hintStyle: TextStyle(
          fontSize: 18.0,
          color: Colors.white,
          fontWeight: FontWeight.bold,
        ), // TextStyle
      ), // InputDecoration
    ); // TextField
  }
}

```

3. Hitung Trapesium



//code UI

```
class _TrapesiumPageState extends State<TrapesiumPage> {
  build(BuildContext context) {
    body: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.all(20),
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                'Hitung Luas Input Sisi Atas, Sisi Bawah, dan Tinggi',
                style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
              ), // Text
              TextFieldWidget(
                controller: controller.sisi1Controller,
                hintText: 'Masukkan sisi atas',
              ), // TextFieldWidget
              const SizedBox(height: 10),
              TextFieldWidget(
                controller: controller.sisi2Controller,
                hintText: 'Masukkan sisi bawah',
              ), // TextFieldWidget
              const SizedBox(height: 10),
              TextFieldWidget(
                controller: controller.tinggiController,
                hintText: 'Masukkan tinggi',
              ), // TextFieldWidget
              const SizedBox(height: 10),
              Container(height: 10, color: Colors.blue),
              const SizedBox(height: 10),
              Text(
                'Hitung Keliling Input Sisi Atas, Sisi Bawah, Sisi Miring Kiri, dan Sisi Miring Kanan',
                style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
              ), // Text
              const SizedBox(height: 10),
            ],
          ),
        ),
      ),
    ),
  ),
}
```

//code button diklik, maka hitung keliling

```
Text('Hitung Keliling', style: TextStyle(fontSize: 20)),
],
), // Row
onPressed: () {
  controller.hitungKeliling(
    context,
    double.tryParse(controller.sisi1Controller.text) ?? 0,
    double.tryParse(controller.sisi2Controller.text) ?? 0,
    double.tryParse(controller.sisi3Controller.text) ?? 0,
    double.tryParse(controller.sisi4Controller.text) ?? 0,
  );
},
// TextButton
```

//code button diklik, maka hitung luas

```
        Text('Hitung Luas', style: TextStyle(fontSize: 20)),
      ],
    ), // ROW
    onPressed: () {
      controller.hitungLuas(
        context,
        double.tryParse(controller.sisi1Controller.text) ?? 0,
        double.tryParse(controller.sisi2Controller.text) ?? 0,
        double.tryParse(controller.tinggiController.text) ?? 0,
      );
    },
  ), // TextButton
```

//controller untuk hitung trapesium

```
class TrapesiumController {
  TextEditingController sisi1Controller = TextEditingController();
  TextEditingController sisi2Controller = TextEditingController();
  TextEditingController sisi3Controller = TextEditingController();
  TextEditingController sisi4Controller = TextEditingController();
  TextEditingController tinggiController = TextEditingController();

  void hitungLuas(context, double sisi1, double sisi2, double tinggi) {
    if (sisi1Controller.text.isEmpty ||
        sisi2Controller.text.isEmpty ||
        tinggiController.text.isEmpty) {
      showSnackBar(context, 'Field sisi 1, 2 dan tinggi tidak boleh kosong', Colors.red);
      return;
    }
    showSnackBar(context, 'Luas Trapesium adalah  $\{0.5 * (sisi1 + sisi2) * tinggi\}$ ', Colors.green);
  }

  void hitungKeliling(context,
    double sisi1, double sisi2, double sisi3, double sisi4) {
    if (sisi1Controller.text.isEmpty ||
        sisi2Controller.text.isEmpty ||
        sisi3Controller.text.isEmpty ||
        sisi4Controller.text.isEmpty) {
      showSnackBar(context, 'Field sisi 1, 2, 3 dan 4 tidak boleh kosong', Colors.red);
      return;
    }
    showSnackBar(context, 'Keliling Trapesium adalah  $\{sisi1 + sisi2 + sisi3 + sisi4\}$ ', Colors.green);
  }

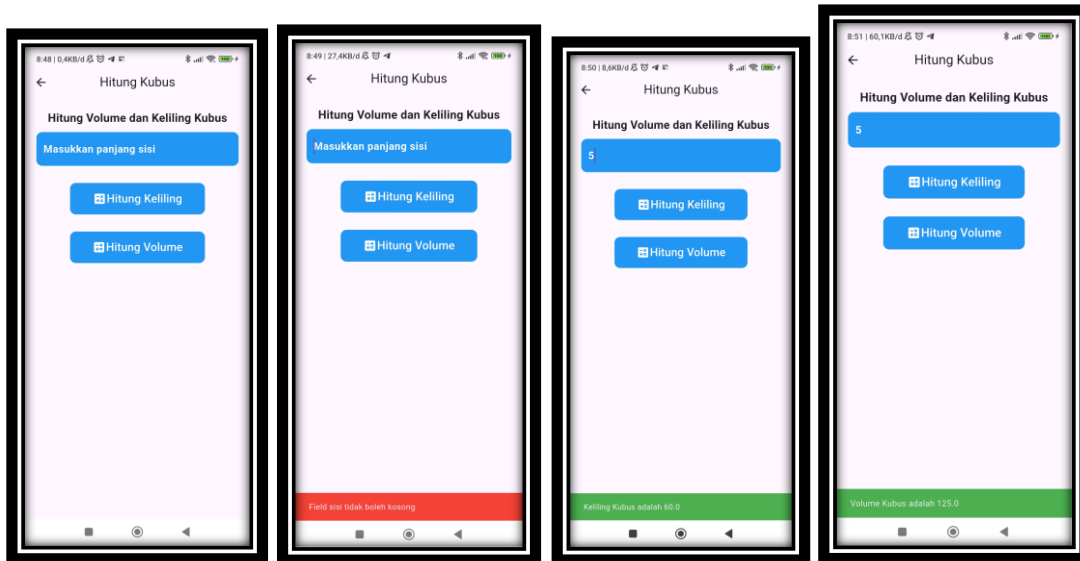
  void showSnackBar(BuildContext context, String message, Color color) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text(message),
        backgroundColor: color,
      ), // SnackBar
    );
  }
}
```

TextEditingController untuk textfield selalu berada di controllernya

PENJELASAN:

Kode di atas adalah halaman perhitungan luas dan keliling trapesium di Flutter. Pengguna menginput nilai sisi atas, sisi bawah, tinggi **untuk menghitung volume**, serta nilai sisi atas, sisi bawah, sisi miring kiri dan kanan **untuk menghitung keliling** menggunakan TextFieldWidget. Ada dua tombol, "Hitung Luas" dan "Hitung Keliling", yang saat ditekan akan memanggil fungsi hitungLuas dan hitungKeliling dari TrapesiumController. Tampilan dirancang dengan Column di dalam SingleChildScrollView agar tidak terjadi overflow saat keyboard muncul.

4. Hitung Kubus



//code UI

```
children: [
  const Text(
    'Hitung Volume dan Keliling Kubus',
    style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
  ), // Text
  const SizedBox(height: 10),
  TextFieldWidget(
    controller: controller.sisiController,
    hintText: 'Masukkan panjang sisi',
  ), // TextFieldWidget
  const SizedBox(height: 20),
  Container(
    width: MediaQuery.of(context).size.width * 0.65,
    height: 75,
    padding: const EdgeInsets.all(10),
    child: TextButton(
      style: TextButton.styleFrom([
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(10),
        ), // RoundedRectangleBorder
        foregroundColor: Colors.white,
        backgroundColor: Colors.blue,
      ]),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: const [
          Icon(Icons.calculate, size: 24, color: Colors.white),
          Text('Hitung Keliling', style: TextStyle(fontSize: 20)),
        ],
      ), // Row
    ),
  ),
],
```

//code button diklik, maka hitung keliling dan volume

```
onPressed: () {  
  controller.hitungKelilingKubus(  
    context,  
    double.tryParse(controller.sisiController.text) ?? 0,  
  );  
},  
  
onPressed: () {  
  controller.hitungVolumeKubus(  
    context,  
    double.parse(controller.sisiController.text),  
  );  
},
```

//controller hitung kubus

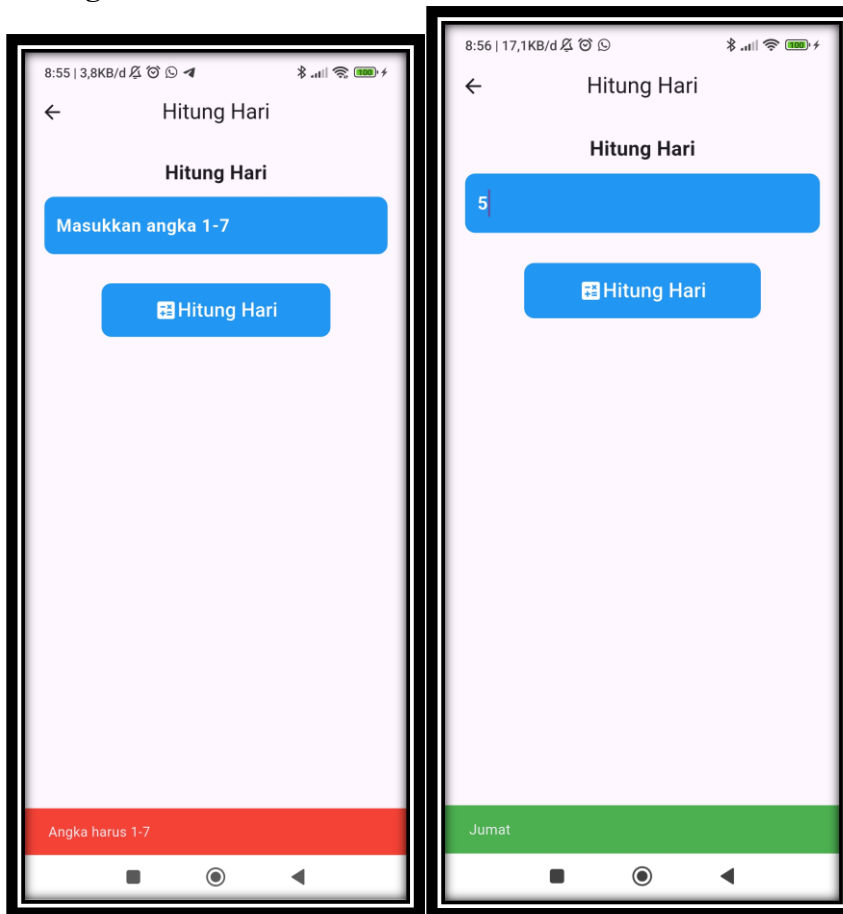
```
class KubusController {  
  TextEditingController sisiController = TextEditingController();  
  void hitungVolumeKubus(context, double sisi) {  
    if (sisiController.text.isEmpty) {  
      showSnackBar(context, 'Field sisi tidak boleh kosong', Colors.red);  
      return;  
    }  
    showSnackBar(  
      context,  
      'Volume Kubus adalah ${sisi * sisi * sisi}',  
      Colors.green,  
    );  
  }  
  
  void hitungKelilingKubus(context, double sisi) {  
    if (sisiController.text.isEmpty) {  
      showSnackBar(context, 'Field sisi tidak boleh kosong', Colors.red);  
      return;  
    }  
    showSnackBar(context, 'Keliling Kubus adalah ${12 * sisi}', Colors.green);  
  }  
  
  void showSnackBar(BuildContext context, String message, Color color) {  
    ScaffoldMessenger.of(  
      context,  
    ).showSnackBar(SnackBar(content: Text(message), backgroundColor: color));  
  }  
}
```

TextEditingController untuk textfield selalu berada di controllernya

PENJELASAN:

Kode ini memungkinkan pengguna untuk menghitung luas dan keliling trapesium dengan memasukkan panjang sisi atas, sisi bawah, tinggi, dan sisi miring. **Sisi tidak boleh KOSONG**

5. Hitung Hari



//code UI

```
class _HitungHariState extends State<HitungHari> {  
  widget build(BuildContext context) {  
    child: Padding(  
      padding: const EdgeInsets.all(20),  
      child: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            const Text(  
              'Hitung Hari',  
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),  
            ), // Text  
            const SizedBox(height: 10),  
            TextFieldWidget(  
              controller: controller.hariController,  
              hintText: 'Masukkan angka 1-7',  
            ), // TextFieldWidget  
            const SizedBox(height: 20),  
            Container(  
              width: MediaQuery.of(context).size.width * 0.65,  
              height: 75,  
              padding: const EdgeInsets.all(10),  
              child: TextButton(  
                style: TextButton.styleFrom(  
                  shape: RoundedRectangleBorder(  
                    borderRadius: BorderRadius.circular(10),  
                  ), // RoundedRectangleBorder  
                  foregroundColor: Colors.white,  
                  backgroundColor: Colors.blue,  
                ),  
                child: Row(  
                  mainAxisAlignment: MainAxisAlignment.center,  
                  children: const [  
                    Icon(Icons.calculate, size: 24, color: Colors.white),  
                    Text('Hitung Hari', style: TextStyle(fontSize: 20)),  
                  ],  
                ),  
              ),  
            ),  
          ],  
        ),  
      ),  
    ),  
  ),  
}
```

//code button hitung hari

```
child: Row(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: const [  
    Icon(Icons.calculate, size: 24, color: Colors.white),  
    Text('Hitung Hari', style: TextStyle(fontSize: 20)),  
  ],  
) , // Row  
onPressed: () {  
  final result = controller.hitungHari(  
    int.tryParse(controller.hariController.text) ?? 0,  
  );  
  if (result != 'Tidak ada hari ke-0') {  
    showSnackBar(context, result, Colors.green);  
  } else {  
    showSnackBar(context, 'Angka harus 1-7', Colors.red);  
  }  
},
```

//code hitung hari controller

```
class HariController {  
  TextEditingController hariController = TextEditingController();  
  static List<String> hari = [  
    'Senin',  
    'Selasa',  
    'Rabu',  
    'Kamis',  
    'Jumat',  
    'Sabtu',  
    'Minggu'  
  ];  
  
  String hitungHari(int index) {  
    if(index < 1 || index > 7) {  
      return 'Tidak ada hari ke-$index';  
    }  
    return hari[index-1];  
  }  
}
```

TextEditingController untuk textfield selalu berada di controllernya

PENJELASAN:

Kode di atas memungkinkan pengguna memasukkan angka 1-7 untuk mendapatkan nama hari dalam seminggu (Senin-Minggu). Jika angka di luar rentang tersebut dimasukkan, aplikasi akan menampilkan pesan kesalahan. Perhitungan hari didasarkan pada indeks:

- 1 → Senin
- 2 → Selasa
- 3 → Rabu
- 4 → Kamis
- 5 → Jumat

6 → Sabtu

7 → Minggu

Jika angka yang dimasukkan bukan 1-7, maka akan muncul pesan bahwa angka harus dalam rentang tersebut.

6. Biodata



//code UI

```
class BiodataPage extends StatelessWidget {  
  const BiodataPage({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: const Text('Biodata'), centerTitle: true),  
      body: Center(  
        child: Column(  
          children: [  
            Text("Nama : Made Vidyatma Adhi Krisna"),  
            Text("NIM : 123220038"),  
            Text("Kelas : IF-H"),  
            Text("Hobi : Mendaki Gunung"),  
            Image.asset('assets/krisna.jpg'),  
          ],  
        ),  
      ),  
    );  
  }  
}
```

PENJELASAN:

Kode ini adalah halaman biodata sederhana saya untuk menampilkan nama, NIM, kelas, hobi, dan gambar menggunakan StatelessWidget. Scaffold digunakan sebagai struktur utama dengan AppBar sebagai judul. Column menyusun elemen secara vertikal, termasuk beberapa Text untuk informasi biodata dan Image.asset untuk menampilkan gambar dari folder assets. Halaman ini **tidak ada controllernya** karena menggunakan data halaman statis