

Nama: Muhammad Rafi Attariq

NIM: H1D022056

UAS: Pemrograman Mobile A

Github: https://github.com/KucingSantuy/H1D022056_PEMMOB_UAS_POS.git

Laporan Penjelasan Kode Flutter

1. Struktur Aplikasi Flutter

Pada aplikasi ini, kita menggunakan **Flutter** untuk membangun aplikasi berbasis mobile dengan menggunakan **GetX** untuk manajemen state dan routing. Aplikasi ini memiliki beberapa tampilan utama, yaitu halaman **Login**, **Dashboard**, dan **Cashier** yang semuanya dikelola dengan **GetMaterialApp** dan dikendalikan oleh **GetPage**.

2. Main.dart

```
Unset
void main() {
  runApp(MyApp());
}
```

Kode ini memulai aplikasi Flutter dengan menjalankan **MyApp()**, yang berfungsi sebagai root widget. Widget ini mengonfigurasi aplikasi dan mendefinisikan **routing** menggunakan **GetMaterialApp**. Routing dilakukan dengan **GetPage**, yang menghubungkan route URL dengan widget terkait, yaitu **/login**, **/dashboard**, dan **/cashier**.

3. Login View (**login_view.dart**)

```
Unset
class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  @override
  void initState() {
```

```

    super.initState();
    // Menambahkan controller
    Get.put(LoginController());
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Login")),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: <Widget>[
            TextField(decoration: InputDecoration(labelText:
'Username'))),
            TextField(decoration: InputDecoration(labelText:
'Password'), obscureText: true),
            SizedBox(height: 20),
            ElevatedButton(onPressed: () {}, child:
Text('Login')),
          ],
        ),
      ),
    );
  }
}

```

- **LoginPage** adalah halaman untuk melakukan login.
- Menggunakan **Get.put()** untuk memastikan controller (**LoginController**) hanya diinisialisasi sekali.
- **Scaffold** digunakan untuk menyediakan struktur dasar halaman, dengan **AppBar** di bagian atas dan **Column** untuk menampilkan input teks (username dan password) serta tombol login.

4. Cashier View (**cashier_view.dart**)

Unset

```
class CashierPage extends StatefulWidget {
  @override
  _CashierPageState createState() => _CashierPageState();
}

class _CashierPageState extends State<CashierPage> {
  @override
  void initState() {
    super.initState();
    Get.put(CashierController());
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Cashier")),
      drawer: Drawer(
        child: ListView(
          children: <Widget>[
            UserAccountsDrawerHeader(
              accountName: Text("User Name"),
              accountEmail: Text("user@example.com"),
              currentAccountPicture: CircleAvatar(child:
Icon(Icons.person)),
            ),
            ListTile(title: Text("Dashboard"), onTap: () {}),
            ListTile(title: Text("Kasir"), onTap: () {}),
            ListTile(title: Text("Logout"), onTap: () {}),
          ],
        ),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: <Widget>[
```

```

        TextField(decoration: InputDecoration(labelText:
'Nama Produk')),
        TextField(decoration: InputDecoration(labelText:
'Harga')),
        SizedBox(height: 20),
        ElevatedButton(onPressed: () {}, child:
Text('Tambah')),
        SizedBox(height: 20),
        Expanded(
          child: ListView.builder(
            itemCount: 10,
            itemBuilder: (context, index) {
              return ListTile(
                title: Text("Produk ${index + 1}"),
                subtitle: Text("Harga: 10000"),
              );
            },
          ),
        ),
        ElevatedButton(onPressed: () {}, child:
Text('Selesaikan Transaksi')),
      ],
    ),
  ),
);
}
}

```

- **CashierPage** adalah halaman untuk menangani transaksi kasir.
- Terdapat **Drawer** untuk navigasi antar halaman.
- **Column** berisi elemen-elemen seperti **TextField** untuk memasukkan produk dan harga, serta **ListView** untuk menampilkan daftar produk.
- **Expanded** digunakan untuk memastikan **ListView** mendapatkan ruang yang cukup dalam tata letak kolom.

5. Dashboard View (**dashboard_view.dart**)

```
class DashboardPage extends StatefulWidget {
  @override
  _DashboardPageState createState() => _DashboardPageState();
}
```

```
class _DashboardPageState extends State<DashboardPage> {
  @override
  void initState() {
    super.initState();
    Get.put(DashboardController());
  }
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text("Dashboard")),
    drawer: Drawer(
      child: ListView(
        children: <Widget>[
          ListTile(title: Text("Dashboard"), onTap: () {}),
          ListTile(title: Text("Kasir"), onTap: () {}),
          ListTile(title: Text("Logout"), onTap: () {}),
        ],
      ),
    ),
    body: Center(
      child: Column(
        children: <Widget>[
          Text("Ringkasan Penjualan"),
          SizedBox(height: 20),
          Expanded(
            child: LineChart(
              LineChartData(
                gridData: FlGridData(show: true),
                titlesData: FlTitlesData(show: true),
                borderData: FlBorderData(show: true),
```

```

        minX: 0,
        maxX: 7,
        minY: 0,
        maxY: 6,
        lineBarsData: [
            LineChartBarData(spots: [
                FlSpot(0, 1),
                FlSpot(1, 2),
                FlSpot(2, 1),
            ])
        ],
    ),
),
),
],
),
),
);
}
}

```

- **DashboardPage** adalah halaman utama yang menampilkan ringkasan penjualan.
- Menggunakan **LineChart** dari `fl_chart` untuk menampilkan grafik penjualan dalam bentuk garis.
- **Drawer** digunakan untuk navigasi ke halaman lain seperti Dashboard dan Kasir.

6. Konsep GetX

Pada setiap view, kita menggunakan **Get.put()** untuk mendaftarkan controller yang akan diakses oleh widget. Ini memungkinkan aplikasi untuk mengelola state dengan mudah dan melakukan navigasi antar halaman dengan **Get.to()** atau **Get.off()**. **GetMaterialApp** juga menyediakan pengelolaan routing secara deklaratif dan sederhana.

7. Kesimpulan

Aplikasi Flutter ini menggunakan beberapa fitur dasar seperti **StatefulWidget**, **Scaffold**, **TextField**, **Drawer**, dan **ListView** untuk membangun antarmuka pengguna. Dengan bantuan **GetX**, aplikasi ini mengelola routing dan state dengan cara yang efisien. Struktur kode ini

memungkinkan pengembangan aplikasi yang mudah diperluas dengan menambah halaman atau fitur baru.

Tampilan

← Login

Username

admin

Password

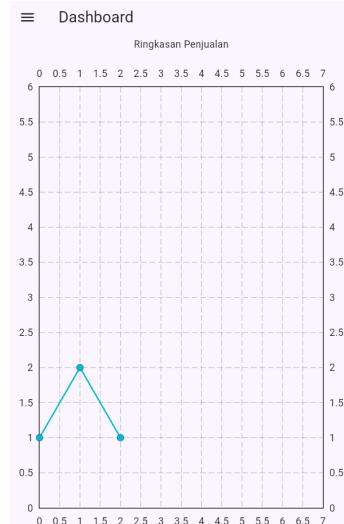
Login

User Name
user@example.com

Dashboard

Kasir

Logout



≡ Cashier

Nama Produk

Harga

Tambah

Produk 1

Harga: 10000

Produk 2

Harga: 10000

Produk 3

Harga: 10000

Produk 4

Harga: 10000

Produk 5

Harga: 10000

Produk 6

Harga: 10000

Produk 7

Harga: 10000

Selesaikan Transaksi