Nicholas Hasian

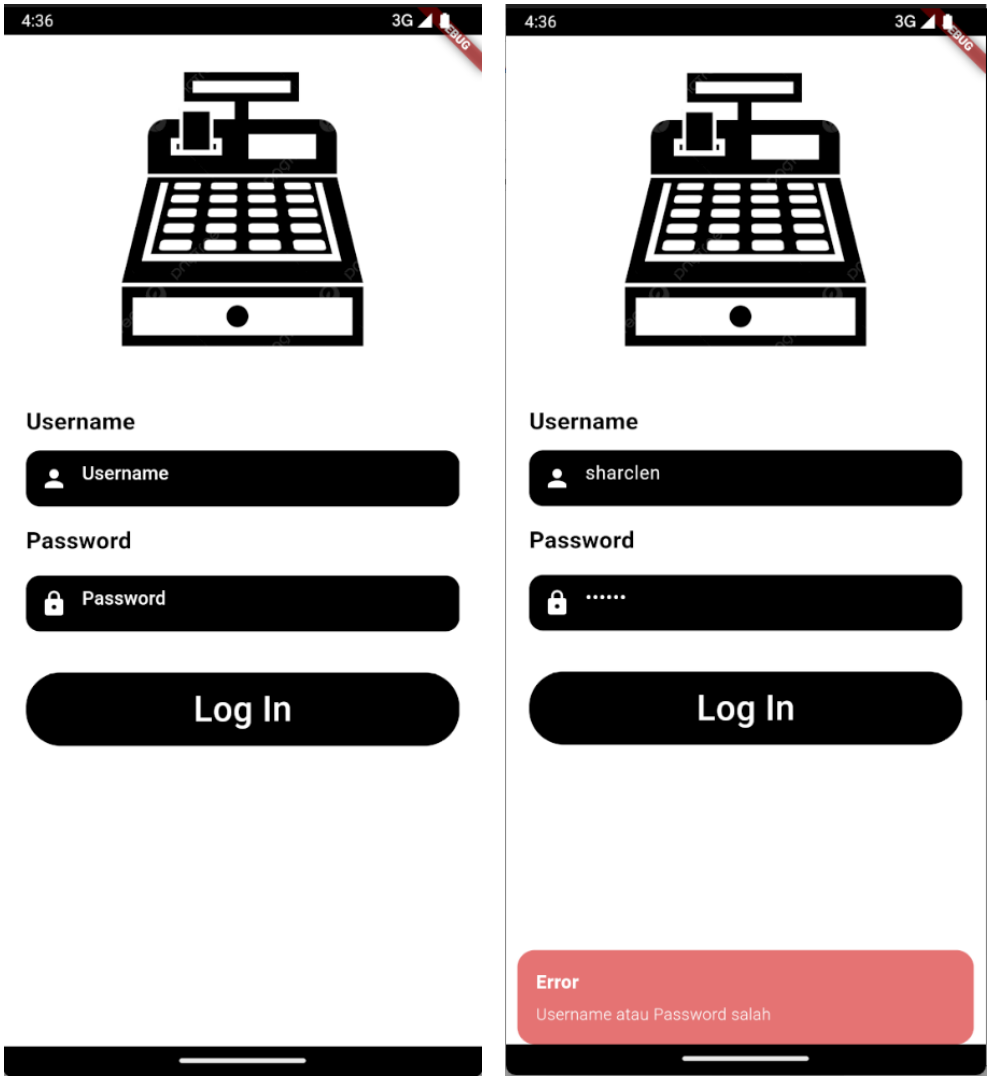
H1D022053

Pemrograman Mobile

Ujian Akhir Semester

Lampiran Beserta Penjelasan

1. Login



Kode :

login_view.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/login_controller.dart';

class LoginView extends StatelessWidget {
  final LoginController loginController = Get.put(LoginController());
```

```dart
final TextEditingController usernameController = TextEditingController();
final TextEditingController passwordController = TextEditingController();

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      backgroundColor: Colors.white,
      body: Column(
        children: [
          Container(
            width: double.infinity,
            height: MediaQuery.of(context).size.height / 3,
            child: Image.asset('lib/images/cashier.png'),
          ),
          Expanded(
            child: Container(
              width: double.infinity,
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: const BorderRadius.only(
                  topLeft: Radius.circular(30),
                  topRight: Radius.circular(30),
                ),
              ),
              child: Padding(
                padding: const EdgeInsets.all(20.0),
                child: SingleChildScrollView(
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      const Text(
                        'Username',
                        style: TextStyle(
                          color: Colors.black,
                          fontSize: 20,
                          fontWeight: FontWeight.bold,
                        ),
                      ),
                      const SizedBox(height: 10),
                      Container(
                        decoration: BoxDecoration(
                          borderRadius: BorderRadius.circular(12),
                          color: Colors.black,
                        ),
                        child: TextField(
                          controller: usernameController,
                          style: const TextStyle(color: Colors.white),
```

```
                    decoration: const InputDecoration(
                      border: InputBorder.none,
                      prefixIcon: Icon(
                        Icons.person,
                        color: Colors.white,
                      ),
                      hintText: 'Username',
                      hintStyle: TextStyle(color: Colors.white),
                    ),
                  ),
                ),
                const SizedBox(height: 15),
                const Text(
                  'Password',
                  style: TextStyle(
                    color: Colors.black,
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                const SizedBox(height: 15),
                Container(
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(12),
                    color: Colors.black,
                  ),
                  child: TextField(
                    controller: passwordController,
                    obscureText: true,
                    style: const TextStyle(color: Colors.white),
                    decoration: const InputDecoration(
                      border: InputBorder.none,
                      prefixIcon: Icon(
                        Icons.lock,
                        color: Colors.white,
                      ),
                      hintText: 'Password',
                      hintStyle: TextStyle(color: Colors.white),
                    ),
                  ),
                ),
                const SizedBox(height: 35),
                GestureDetector(
                  onTap: () {
                    loginController.login(
                      usernameController.text,
                      passwordController.text,
                    );
```

```
                        },
                        child: Container(
                          decoration: BoxDecoration(
                            borderRadius: BorderRadius.circular(30),
                            color: Colors.black,
                          ),
                          child: const Center(
                            child: Padding(
                              padding: EdgeInsets.all(10.0),
                              child: Text(
                                'Log In',
                                style: TextStyle(
                                  color: Colors.white,
                                  fontSize: 30,
                                  fontWeight: FontWeight.w500,
                                ),
                              ),
                            ),
                          ),
                        ),
                      ),
                    ),
                  ],
                ),
              ),
            ),
          );
        }
}
```

login_controller.dart

```
import 'dart:ui';
import 'package:get/get.dart';

class LoginController extends GetxController {
  var isLogged = false.obs;

  void login(String username, String password) {
    if (username == 'sharclen' && password == '12345') {
      isLogged.value = true;
      Get.offAllNamed('/dashboard');
    } else {
      Get.snackbar(
```

```
        'Error',
        'Username atau Password salah',
        snackPosition: SnackPosition.BOTTOM,
        backgroundColor: const Color.fromARGB(255, 19, 11, 11),
        colorText: const Color(0xFFFFFFFF),
      );
    }
  }

  void logout() {
    isLogged.value = false;
    Get.offAllNamed('/login');
  }
}
```

Kode ini berfungsi untuk membuat tampilan dan logic untuk login. LoginView menyediakan interfacec dengan dua kolom input untuk username dan password serta tombol login. Data yang dimasukkan pengguna dikendalikan oleh TextEditingController, dan ketika tombol login ditekan, metode login di LoginController akan dijalankan. Jika username adalah *"sharclen"* dan password adalah *"12345"*, pengguna diarahkan ke halaman dashboard menggunakan Get.offAllNamed. Sebaliknya, jika login gagal, pesan error akan muncul menggunakan Get.snackbar. Controller login ini juga mendukung fungsi logout untuk mengembalikan pengguna ke halaman login.

2. Dashboard (kosong)

dashboard_view.dart

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:fl_chart/fl_chart.dart';
import '../controllers/dashboard_controller.dart';
import '../widgets/sidebar.dart';
import 'history_view.dart';

class DashboardView extends StatelessWidget {
  final DashboardController dashboardController =
      Get.put(DashboardController());

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          title: Text('Dashboard'),
          leading: Builder(
```

```
            builder: (context) => IconButton(
              icon: Icon(Icons.menu),
              onPressed: () {
                Scaffold.of(context).openDrawer();
              },
            ),
          ),
        ),
        drawer: Sidebar(),
        body: SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Row(
                  mainAxisAlignment: MainAxisAlignment.end,
                  children: [
                    GestureDetector(
                      onTap: () {
                        Get.to(() => HistoryView());
                      },
                      child: Container(
                        padding: EdgeInsets.all(12),
                        decoration: BoxDecoration(
                          border: Border.all(color: Colors.grey),
                          borderRadius: BorderRadius.circular(8),
                        ),
                        child: Obx(() => Text(
                              'Total Transaksi:
${dashboardController.totalTransactions.value}',
                              style: TextStyle(fontSize: 16),
                            )),
                      ),
                    ),
                  ],
                ),
                SizedBox(height: 20),
                Text('Grafik Penjualan',
                    style:
                        TextStyle(fontSize: 18, fontWeight: FontWeight.bold)),
                SizedBox(height: 20),
                Obx(() {
                  return SizedBox(
                    height: 200,
                    child: BarChart(
                      BarChartData(
                        barGroups: dashboardController.salesData
```

```
                            .asMap()
                            .entries
                            .map(
                              (entry) => BarChartGroupData(
                                x: entry.key,
                                barRods: [
                                  BarChartRodData(
                                    toY: entry.value,
                                    color: Colors.primaries[
                                        entry.key % Colors.primaries.length],
                                    width: 20,
                                    borderRadius: BorderRadius.zero,
                                  )
                                ],
                              ),
                            )
                            .toList(),
                        titlesData: FlTitlesData(
                          leftTitles: AxisTitles(
                            sideTitles: SideTitles(
                              showTitles: true,
                              reservedSize: 40,
                              getTitlesWidget: (value, meta) {
                                if (value %
                                        calculateDynamicInterval(
                                            dashboardController.salesData) ==
                                    0) {
                                  return Text(
                                    '${value.toInt()}',
                                    style: TextStyle(fontSize: 12),
                                  );
                                }
                                return SizedBox.shrink();
                              },
                            ),
                          ),
                          bottomTitles: AxisTitles(
                            sideTitles: SideTitles(
                              showTitles: true,
                              getTitlesWidget: (value, meta) {
                                int index = value.toInt();
                                if (index >= 0 &&
                                    index <
                                        dashboardController.salesData.length)
{
                                  return Text(
                                    'Transaksi ${index + 1}',
                                    style: TextStyle(fontSize: 12),
```

```
                                );
                              }
                              return SizedBox.shrink();
                            },
                            reservedSize: 30,
                          ),
                        ),
                        rightTitles: AxisTitles(
                          sideTitles: SideTitles(showTitles: false),
                        ),
                        topTitles: AxisTitles(
                          sideTitles: SideTitles(showTitles: false),
                        ),
                      ),
                      gridData: FlGridData(
                        show: true,
                        horizontalInterval: calculateDynamicInterval(
                            dashboardController.salesData),
                      ),
                      borderData: FlBorderData(
                        show: true,
                        border: Border.all(color: Colors.grey, width: 1),
                      ),
                      maxY: calculateMaxY(dashboardController.salesData),
                    ),
                  ),
                );
              }),
              SizedBox(height: 20),
              Center(
                child: Container(
                  padding: EdgeInsets.all(16),
                  decoration: BoxDecoration(
                    color: Colors.blue,
                    borderRadius: BorderRadius.circular(8),
                  ),
                  child: Obx(() => Text(
                        'Total Penjualan: Rp
${dashboardController.totalSales.value.toStringAsFixed(2)}',
                        style: TextStyle(fontSize: 18, color: Colors.white),
                      )),
                ),
              ),
            ],
          ),
        ),
      ),
    ),
```

```
    );
  }

  double calculateMaxY(List<double> salesData) {
    if (salesData.isEmpty) return 1;
    double maxValue = salesData.reduce((a, b) => a > b ? a : b);
    return (maxValue * 1.2).ceilToDouble();
  }

  double calculateDynamicInterval(List<double> salesData) {
    if (salesData.isEmpty) return 1;
    double maxValue = calculateMaxY(salesData);
    return (maxValue / 5).ceilToDouble();
  }
}
```

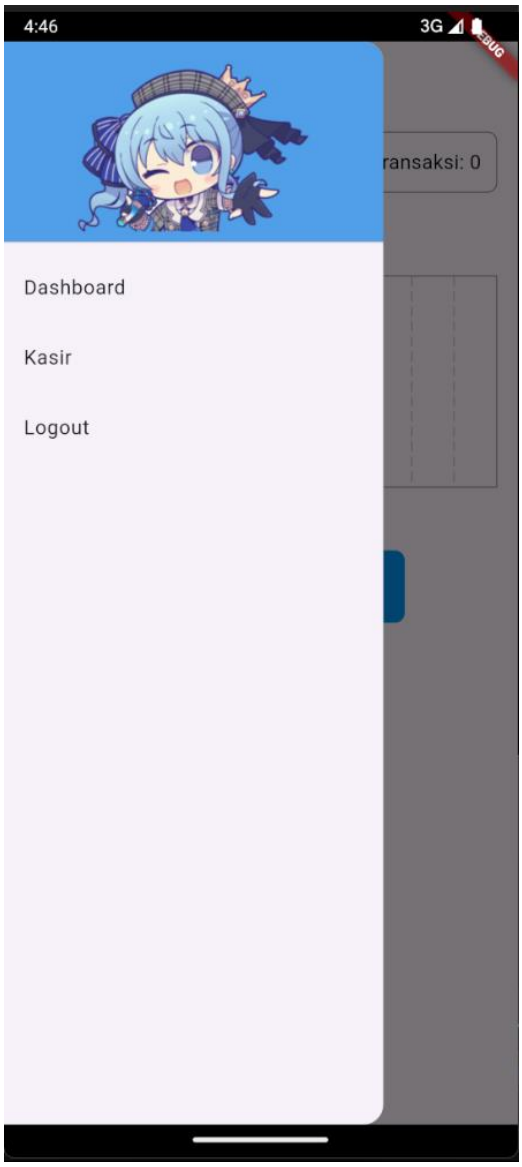
dashboard_controller.dart

```
import 'package:get/get.dart';
import '../controllers/cashier_controller.dart';

class DashboardController extends GetxController {
  var totalTransactions = 0.obs;
  var totalSales = 0.0.obs;
  var salesData = <double>[].obs;

  final CashierController cashierController = Get.find<CashierController>();

  @override
  void onInit() {
    super.onInit();

    cashierController.history.listen((history) {
      totalTransactions.value = history.length;

      totalSales.value = history.fold(
        0.0,
        (sum, transaction) =>
            sum +
            transaction.fold(0.0, (subSum, product) => subSum +
product.price),
      );

      salesData.value = history.map((transaction) {
        return transaction.fold(
          0.0,
          (sum, product) => sum + product.price,
        );
```

```
      }).toList();
    });
  }
}
```

Kode ini berfungsi untuk menampilkan Dashboard yang menampilkan ringkasan data penjualan menggunakan grafik batang dan UI yang interaktif. DashboardView memanfaatkan DashboardController untuk mengelola data seperti total transaksi, total penjualan, dan data penjualan. Keudian terdapat juga sidebar untuk navigasi, total transaksi yang dapat membuka halaman histori, grafik batang yang merepresentasikan penjualan per transaksi, serta total penjualan dalam rupiah. Grafik dirender menggunakan library FL Chart, dengan data yang diperbarui secara real-time melalui Rx observables dari GetX. Controller memanfaatkan data dari CashierController untuk menghitung dan menyinkronkan data yang ditampilkan pada dashboard.

3. Sidebar



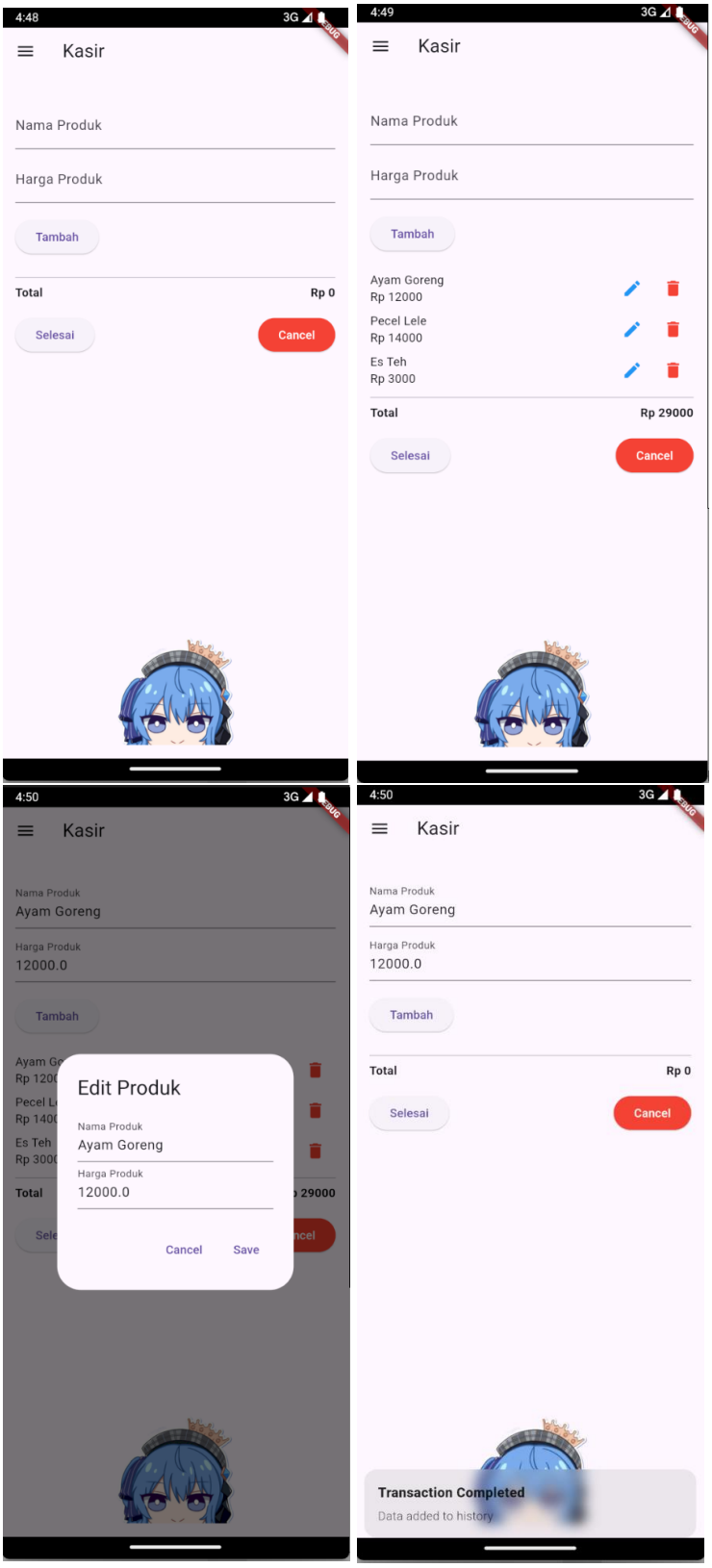
sidebar.dart

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/login_controller.dart';
import '../controllers/cashier_controller.dart';

class Sidebar extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    Get.lazyPut(() => CashierController());
    Get.lazyPut(() => LoginController());

    final LoginController loginController = Get.find<LoginController>();

    return Drawer(
```

```
      child: ListView(
        padding: EdgeInsets.zero,
        children: [
          DrawerHeader(
            decoration: BoxDecoration(color: Color.fromARGB(255, 80, 159,
232)),
            child: Image.asset('lib/images/sui.png'),
          ),
          ListTile(
            title: Text('Dashboard'),
            onTap: () {
              Get.toNamed('/dashboard');
            },
          ),
          ListTile(
            title: Text('Kasir'),
            onTap: () {
              if (!Get.isRegistered<CashierController>()) {
                Get.lazyPut(() => CashierController());
              }
              Get.toNamed('/cashier');
            },
          ),
          ListTile(
            title: Text('Logout'),
            onTap: () {
              loginController.logout();
            },
          ),
        ],
      ),
    );
  }
}
```

Kode ini berfungsi untuk membuat Sidebar menggunakan Drawer pada Flutter. Sidebar ini memungkinkan pengguna untuk mengakses halaman-halaman pada aplikasi, yaitu Dashboard dan Kasir, serta melakukan Logout. Pada menu Dashboard dan Kasir, navigasi dilakukan menggunakan GetX routing, di mana controller yang diperlukan, seperti CashierController, akan diinisialisasi secara otomatis jika belum tersedia. Pada menu Logout, metode logout dari LoginController dijalankan untuk keluar dari akun pengguna dan mengarahkan kembali ke halaman login.

4. Kasir



cashier_view.dart

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

```dart
import '../controllers/cashier_controller.dart';
import '../widgets/sidebar.dart';
import 'history_view.dart';

class CashierView extends StatelessWidget {
  final CashierController cashierController = Get.find<CashierController>();
  final TextEditingController nameController = TextEditingController();
  final TextEditingController priceController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          title: Text('Kasir'),
          leading: Builder(
            builder: (context) => IconButton(
              icon: Icon(Icons.menu),
              onPressed: () {
                Scaffold.of(context).openDrawer();
              },
            ),
          ),
        ),
        drawer: Sidebar(),
        body: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              SizedBox(height: 16),
              TextField(
                controller: nameController,
                decoration: InputDecoration(labelText: 'Nama Produk'),
              ),
              SizedBox(height: 8),
              TextField(
                controller: priceController,
                keyboardType: TextInputType.number,
                decoration: InputDecoration(labelText: 'Harga Produk'),
              ),
              SizedBox(height: 16),
              ElevatedButton(
                onPressed: () {
                  cashierController.addProduct(
                    nameController.text,
                    double.tryParse(priceController.text) ?? 0.0,
                  );
```

```
                  nameController.clear();
                  priceController.clear();
                },
                child: Text('Tambah'),
              ),
              SizedBox(height: 16),
              Obx(() {
                return Column(
                  children: [
                    ...cashierController.products.asMap().entries.map((entry)
{
                      int index = entry.key;
                      var product = entry.value;
                      return Row(
                        mainAxisAlignment: MainAxisAlignment.spaceBetween,
                        children: [
                          Column(
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: [
                              Text(product.name),
                              Text('Rp ${product.price.toStringAsFixed(0)}'),
                            ],
                          ),
                          Row(
                            children: [
                              IconButton(
                                icon: Icon(Icons.edit, color: Colors.blue),
                                onPressed: () {
                                  nameController.text = product.name;
                                  priceController.text =
                                      product.price.toString();
                                  showDialog(
                                    context: context,
                                    builder: (context) => AlertDialog(
                                      title: Text('Edit Produk'),
                                      content: Column(
                                        mainAxisSize: MainAxisSize.min,
                                        children: [
                                          TextField(
                                            controller: nameController,
                                            decoration: InputDecoration(
                                                labelText: 'Nama Produk'),
                                          ),
                                          TextField(
                                            controller: priceController,
                                            keyboardType:
TextInputType.number,
                                            decoration: InputDecoration(
```

```
                                    labelText: 'Harga Produk'),
                              ),
                            ],
                          ),
                          actions: [
                            TextButton(
                              onPressed: () {
                                Get.back();
                              },
                              child: Text('Cancel'),
                            ),
                            TextButton(
                              onPressed: () {
                                cashierController.editProduct(
                                  index,
                                  nameController.text,
                                  double.tryParse(
                                          priceController.text) ??
                                      0.0,
                                );
                                nameController.clear();
                                priceController.clear();
                                Get.back();
                              },
                              child: Text('Save'),
                            ),
                          ],
                        ),
                      );
                    },
                  ),
                  IconButton(
                    icon: Icon(Icons.delete, color: Colors.red),
                    onPressed: () {
                      cashierController.removeProduct(index);
                    },
                  ),
                ],
              ),
            ],
          );
        }).toList(),
        Divider(),
        Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            Text('Total',
                style: TextStyle(fontWeight: FontWeight.bold)),
```

```
                      Text(
                          'Rp
${cashierController.totalPrice.value.toStringAsFixed(0)}',
                          style: TextStyle(fontWeight: FontWeight.bold)),
                    ],
                  ),
                ],
              );
            }),
            SizedBox(height: 16),
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                ElevatedButton(
                  onPressed: () {
                    cashierController.completeTransaction();
                  },
                  child: Text('Selesai'),
                ),
                ElevatedButton(
                  onPressed: () {
                    cashierController.cancelTransaction();
                  },
                  child:
                      Text('Cancel', style: TextStyle(color: Colors.white)),
                  style:
                      ElevatedButton.styleFrom(backgroundColor: Colors.red),
                ),
              ],
            ),
            Spacer(),
            Center(
              child: GestureDetector(
                onTap: () {
                  Get.to(() => HistoryView());
                },
                child: Image.asset(
                  'lib/images/minisui.png',
                  width: 150,
                  height: 150,
                  fit: BoxFit.cover,
                ),
              ),
            ),
          ],
        ),
      ),
    ),
```

```
    );
  }
}
```

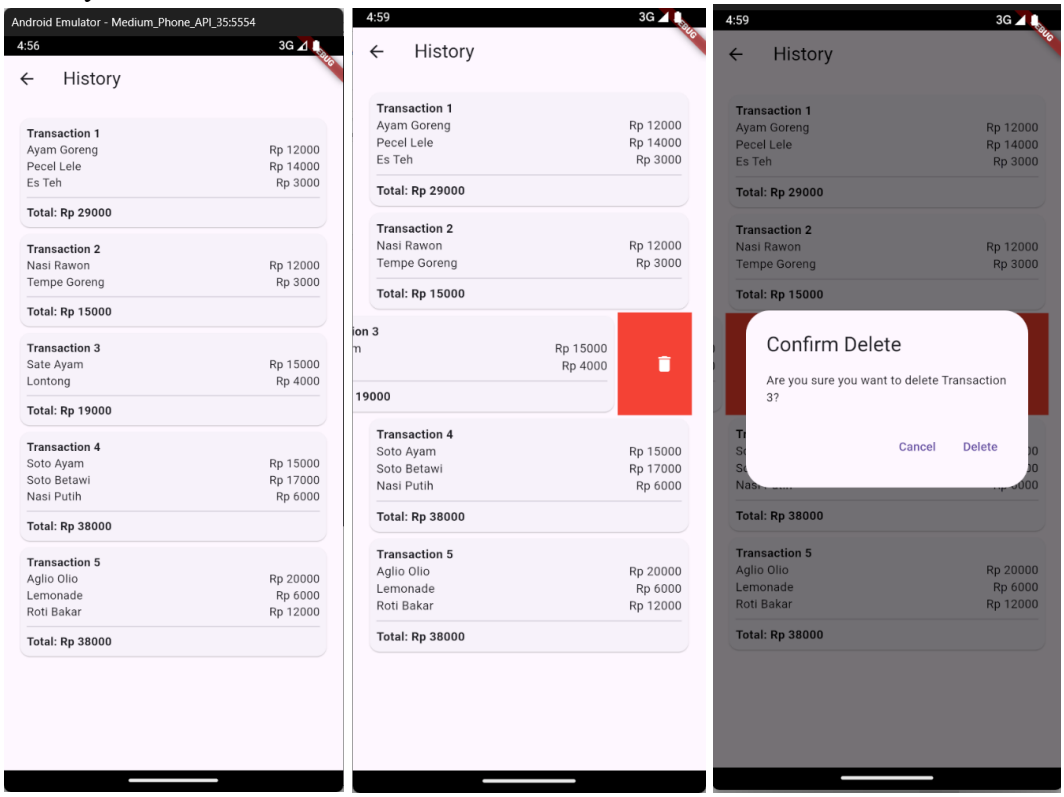
cashier_controller.dart

```
import 'package:get/get.dart';
import 'package:uas_getx/models/product_model.dart';

class CashierController extends GetxController {
  var products = <Product>[].obs;
  var totalPrice = 0.0.obs;
  var history = <List<Product>>[].obs;
  var transactions = <double>[].obs;

  void addProduct(String name, double price) {
    if (name.isNotEmpty && price > 0) {
      products.add(Product(name: name, price: price));
      totalPrice.value += price;
    }
  }

  void editProduct(int index, String newName, double newPrice) {
    if (newName.isNotEmpty && newPrice > 0) {
      var product = products[index];
      totalPrice.value -= product.price;
      products[index] = Product(name: newName, price: newPrice);
      totalPrice.value += newPrice;
    }
  }

  void removeProduct(int index) {
    var product = products[index];
    totalPrice.value -= product.price;
    products.removeAt(index);
  }

  void completeTransaction() {
    if (products.isNotEmpty) {
      history.add(List.from(products));
      products.clear();
      totalPrice.value = 0.0;
      Get.snackbar('Transaction Completed', 'Data added to history',
          snackPosition: SnackPosition.BOTTOM);
    } else {
      Get.snackbar('Error', 'No products to complete the transaction',
          snackPosition: SnackPosition.BOTTOM);
    }
```

```
  }

  void removeTransaction(int index) {
  history.removeAt(index);
}

  void cancelTransaction() {
    products.clear();
    totalPrice.value = 0.0;
    Get.snackbar('Transaction Canceled', 'All products have been removed',
        snackPosition: SnackPosition.BOTTOM);
  }
}
```

Kode ini berfungsi untuk menampilkan halaman Kasir yang memungkinkan pengguna mencatat transaksi penjualan produk. Dengan CashierView, pengguna dapat menambahkan produk dengan mengisi nama dan harga, mengedit produk yang sudah dimasukkan, menghapus produk tertentu, serta melihat total harga keseluruhan dari transaksi yang sedang berlangsung. Data produk kemudian disimpan secara dinamis menggunakan observables dari GetX yang dikelola oleh CashierController. Pengguna juga dapat menekan tombol “Selesai” untuk mengakhiri transaksi, yang secara otomatis menyimpan daftar produk ke dalam riwayat (halaman history), atau membatalkan transaksi untuk menghapus seluruh produk. Selain itu, tersedia navigasi menuju halaman history transaksi melalui ikon gambar pada bagian bawah halaman, serta fitur notifikasi untuk memberikan informasi saat transaksi berhasil diselesaikan atau dibatalkan.

5. History



history_view.dart

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/cashier_controller.dart';

class HistoryView extends StatelessWidget {
  final CashierController cashierController = Get.find<CashierController>();

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(title: Text('History')),
        body: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Obx(() {
            if (cashierController.history.isEmpty) {
              return Center(child: Text('No Transactions Yet'));
            }
            return ListView.builder(
              itemCount: cashierController.history.length,
              itemBuilder: (context, index) {
                final transaction = cashierController.history[index];
                return Dismissible(
                  key: Key(transaction.hashCode.toString()),
                  direction: DismissDirection.endToStart,
```

```
                confirmDismiss: (direction) async {
                  return await showDialog(
                    context: context,
                    builder: (context) {
                      return AlertDialog(
                        title: Text('Confirm Delete'),
                        content: Text(
                            'Are you sure you want to delete Transaction
${index + 1}?'),
                        actions: [
                          TextButton(
                            onPressed: () =>
Navigator.of(context).pop(false),
                            child: Text('Cancel'),
                          ),
                          TextButton(
                            onPressed: () =>
Navigator.of(context).pop(true),
                            child: Text('Delete'),
                          ),
                        ],
                      );
                    },
                  );
                },
                background: Container(
                  color: Colors.red,
                  alignment: Alignment.centerRight,
                  padding: EdgeInsets.symmetric(horizontal: 20),
                  child: Icon(Icons.delete, color: Colors.white),
                ),
                onDismissed: (direction) {
                  cashierController.removeTransaction(index);
                  Get.snackbar(
                    'Transaction Deleted',
                    'Transaction ${index + 1} has been removed',
                    snackPosition: SnackPosition.BOTTOM,
                  );
                },
                child: Card(
                  child: Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        Text('Transaction ${index + 1}',
                            style: TextStyle(fontWeight: FontWeight.bold)),
                        ...transaction.map((product) {
```

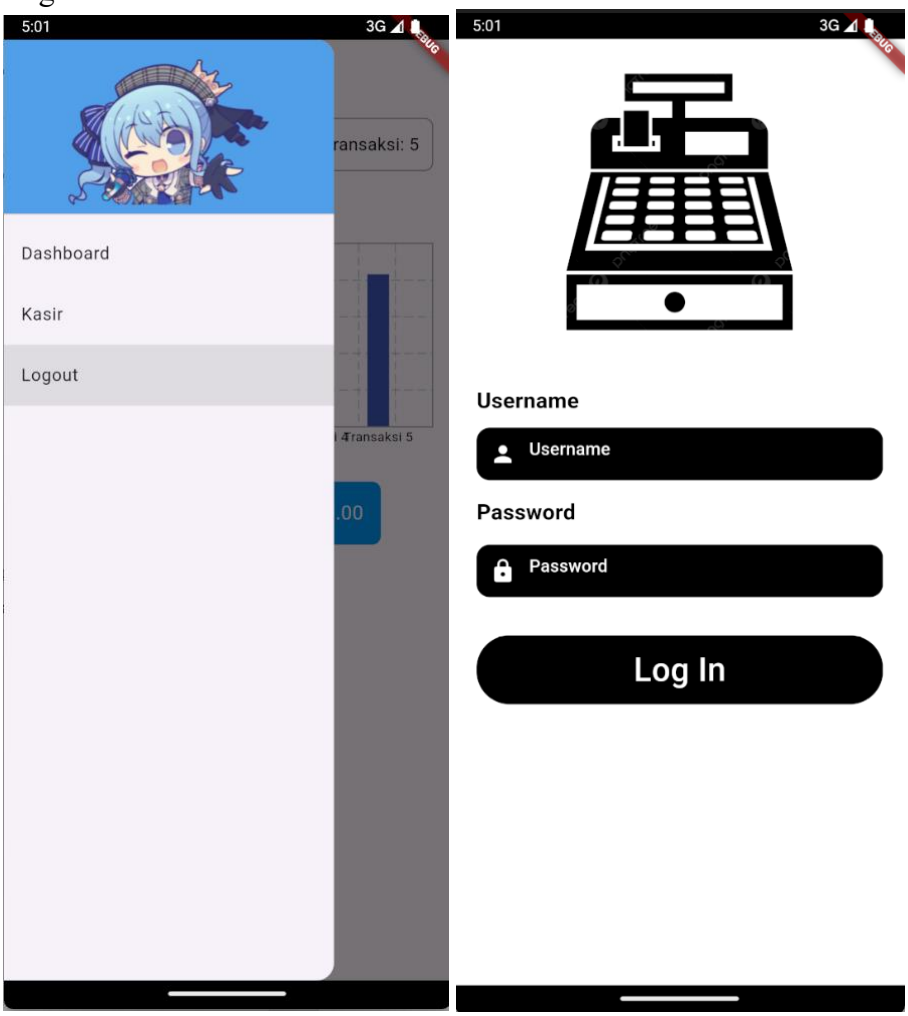
```
                            return Row(
                              mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                              children: [
                                Text(product.name),
                                Text('Rp
${product.price.toStringAsFixed(0)}'),
                              ],
                            );
                          }).toList(),
                          Divider(),
                          Text(
                            'Total: Rp ${transaction.fold<double>(0.0, (sum,
product) => sum + product.price).toStringAsFixed(0)}',
                            style: TextStyle(fontWeight: FontWeight.bold),
                          ),
                        ],
                      ),
                    ),
                  ),
                );
              },
            );
          }),
        ),
      ),
    );
  }
}
```

Kode ini berfungsi untuk membangun halaman Riwayat Transaksi yang menampilkan daftar transaksi yang telah selesai, halaman ini dapat diakses melalui halaman kasir maupun dashboard, dengan menekan gambar pada halaman kasir, dan menekan total transaksi pada dashboard. Pada tampilan HistoryView, setiap transaksi yang sebelumnya diselesaikan akan ditampilkan sebagai kartu terpisah, menampilkan nama produk, harga, dan total biaya transaksi tersebut. Pengguna juga dapat menghapus transaksi tertentu dengan menggunakan fitur Dismissible, di mana gesekan ke arah kanan akan memunculkan opsi konfirmasi sebelum transaksi dihapus dari daftar. Jika tidak ada transaksi yang tersimpan, halaman akan menampilkan pesan "No Transactions Yet" sebagai indikator.

6. Dashboard (Isi)

Dengan kode yang sama seperti sebelumnya, apabila telah terdapat data pada halaman history, aka total transaksi, grafik, dan total penjualan akan berubah menyesuaikan data yang terdapat pada catatan transaksi

7. Logout



Dengan memanfaatkan fungsi logout yang terdapat pada LoginController, maka sesuai dengan tujuannya, apabila tombol logout pada sidebar ditekan, maka user akan dikembalikan ke halaman login dan diperlukan login ulang untuk membuka kembali aplikasi ini.