

https://github.com/bintangyosua/UAS_FLUTTER_GETX | [Google Drive](#)

```
lib
├── controllers
│   ├── cashier_controller.dart
│   ├── dashboard_controller.dart
│   ├── login_controller.dart
│   └── product_controller.dart
├── models
│   ├── mulan.dart
│   ├── products.dart
│   ├── transaction_item.dart
│   └── transactions.dart
├── routes
│   └── routes.dart
├── views
│   ├── cashier_view.dart
│   ├── dashboard_view.dart
│   ├── login_view.dart
│   └── product_view.dart
└── widgets
    └── main.dart
```

```
Future<void> main() async {  
    WidgetsFlutterBinding.ensureInitialized();  
  
    await Supabase.initialize(  
        url: 'https://slagbwlovzbkzelytgxz.supabase.co',  
        anonKey: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzd  
runApp(const MyApp());  
}
```

Routing menggunakan GetX

```
You, 4 hours ago | 1 author (You)
1 import 'package:flutter_getx/views/dashboard_view.dart';
2 import 'package:flutter_getx/views/login_view.dart';
3 import 'package:flutter_getx/views/cashier_view.dart';
4 import 'package:flutter_getx/views/product_view.dart';
5 import 'package:get/get.dart';
6 You, 8 hours ago • first commit
You, 4 hours ago | 1 author (You) | Codeium: Refactor | Explain
7 class AppRoutes {
8   static final routes = [
9     GetPage(name: '/', page: () => LoginView()), Use 'const' with th
10    GetPage(name: '/dashboard', page: () => DashboardView()),
11    GetPage(name: '/cashier', page: () => CashierView()),
12    GetPage(name: '/products', page: () => ProductView()),
13  ];
14 }
15
```

a. Login View

Di halaman ini, pengguna dapat melakukan login. Terdapat validasi seperti field yang perlu diisi dan data yang tidak cocok yang terdapat pada database.

Data yang benar adalah **username: admin** dan **password: admin**.

```
You, 1 second ago | 1 author (You) | Codeium: Refactor | Explain
class LoginController extends GetxController { You, 8 hours ago • first commit
  var isLoggedIn = false.obs;
  final SupabaseClient supabase = Supabase.instance.client;

  Codeium: Refactor | Explain | Generate Function Comment | X
  Future<void> login(String username, String password) async {
    try {
      final response = await supabase
        .from('users')
        .select('id, username, password')
        .eq('username', username);

      print('res: $response'); Don't invoke 'print' in production code. Try using

      final userData = response;
      if (userData[0]['password'] == password) {
        SharedPreferences prefs = await SharedPreferences.getInstance(); "prefs":
        await prefs.setBool('isLoggedIn', true); "prefs": Unknown word.
        await prefs.setString('username', username); "prefs": Unknown word.
        isLoggedIn.value = true;
        Get.offAllNamed('/dashboard');
      } else {
        Get.snackbar('Error', 'Username or password is incorrect.');
```

Proses login akan dilakukan pengecekan apakah username dan password yang ada pada database cocok atau tidak. Jika cocok, maka set state secara local username dan state login. Setelah itu diarahkan ke halaman **/dashboard**.

```
ElevatedButton( You, 4 hours ago • styling done
  onPressed: () {
    setState(() {
      _isSubmitted = true; // Set the flag to true when button is pressed
    });

    // Only proceed with login if validation is successful
    if (_validateUsername() == null && _validatePassword() == null) {
      loginController.login(
        usernameController.text,
        passwordController.text,
      );
    } else {
      // Show snackbar if validation fails
      Get.snackbar(
        'Error',
        'Username and password cannot be empty.',
        snackPosition: SnackPosition.TOP,
      );
    }
  },
),
```

Jika gagal, maka memberikan snackbar atau notifikasi dalam aplikasi bahwa login gagal dilakukan beserta pesan yang dibalikkan, seperti username dan password yang salah.


Selain itu, terdapat juga validasi dari form jika username atau password salah, Hal tersebut juga akan mengembalikan notifikasi dalam bentuk snackbar bahwa username dan password tidak boleh kosong.

Error


Username and password cannot be empty.


Error

Username or password is incorrect.




Point of Sale (POS)


 Username


 Password

Login




Point of Sale (POS)


 Username


 Password

Login



Point of Sale (POS)

 Username

 Password

Login

b. Dashboard View

Di halaman ini, user dapat melihat gambaran secara umum bagaimana transaksi dari bisnis yang dijalankan.

Terdapat info-info dasar seperti Total, Total Hari Ini, Total Transaksi Keseluruhan, dan Total Transaksi Hari ini, serta grafik garis yang menggambarkan penjualan tiap harinya.

Di dashboard, state pada DashboardController yang dimiliki ada beberapa, yaitu **dailySales**, **saleDates**, **totalSales**, **totalTodaySales**, **totalTransactions**, dll. Setelah itu, data yang dikirim untuk grafik line chart digrouping berdasarkan hari tertentu

≡ Dashboard

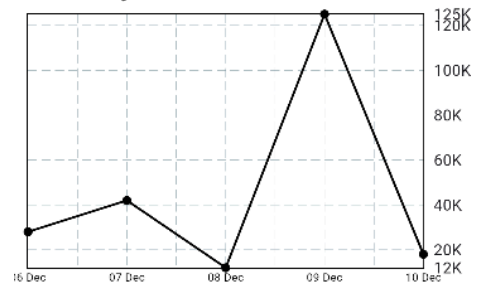
\$ Total
Rp225.000,00

\$ Total Hari Ini
Rp125.000,00

Keseluruhan
11

Hari Ini
5

Grafik Penjualan



widget Scaffold untuk halaman dashboard. `controller.loadSalesData()` dipanggil untuk mendapatkan data transaksi. Obx digunakan untuk membangun tampilan data ringkasan penjualan yang terbagi dalam dua kolom, masing-masing menampilkan total penjualan dan jumlah transaksi keseluruhan serta harian. Setelah itu, terdapat "Grafik Penjualan" dan komponen LineChart untuk menampilkan grafik penjualan harian.

```

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

  DashboardView({super.key});

  @override
  Widget build(BuildContext context) {
    controller.loadSalesData();

    return Scaffold(
      appBar: AppBar(
        title: const Text('Dashboard'),
        backgroundColor: Colors.white,
      ), // AppBar
      drawer: const Sidemenu(),
      backgroundColor: Colors.white,
      body: Padding(
        padding: const EdgeInsets.all(16),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Obx(() {
              return Row(
                children: [
                  Expanded(
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        _buildSummaryList('Total', '${Transaction.formatRupiah(controller.totalSales.value as d
                        _buildSummaryList('Keseluruhan', '${controller.totalTransactions.value}', Icons.numbers
                      ],
                    ), // Column
                  ), // Expanded
                  Expanded(
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        _buildSummaryList('Total Hari ini', '${Transaction.formatRupiah(controller.totalTodaySa
                        _buildSummaryList('Hari Ini', '${controller.totalTodayTransactions}', Icons.numbers),
                      ],
                    ), // Column
                  ), // Expanded
                ],
              ); // Row
            }), // Obx
          ],
        ),
      ),
    );
  }
}

```

```

    const SizedBox(height: 32),

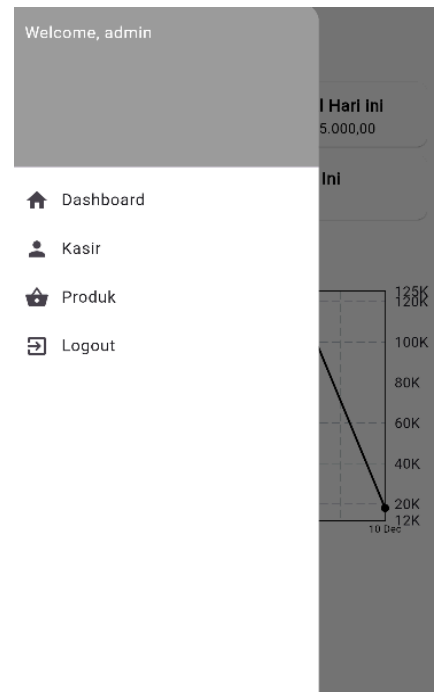
    const Text(
      'Grafik Penjualan',
      style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color: Colors.black),
    ), // Text
    const SizedBox(height: 8),
    Obx(() {
      return controller.dailySales.isEmpty
        ? const Center(child: CircularProgressIndicator())
        : SizedBox( // SizedBox ...
      ), // Obx
    const SizedBox(height: 32),
  ],
), // Column
), // Padding
); // Scaffold
}

```

c. Sidebar

Di sidebar, terdapat menu utama seperti Dashboard, Kasir, Produk dan Logout dengan Icon nya masing-masing

```
@override
Codeium: Refactor | Explain | Generate Function Comment | X
Widget build(BuildContext context) {
  return Drawer(
    backgroundColor: Colors.white,
    child: ListView(children: [
      // Using FutureBuilder to display the user
      FutureBuilder<String>(>
        future: getUsername(),
        builder: (context, snapshot) { ...
      ), // FutureBuilder
      ListTile( // ListTile ...
      ListTile( // ListTile ...
      ListTile( // ListTile ...
      ListTile( // ListTile ...
    ]), // ListView
  ); // Drawer
}
```



d. Kasir

Halaman Kasir digunakan untuk mengelola data transaksi. Untuk menambah produk, klik Button pada pojok kanan atas untuk membuka drawer kanan. Setelah itu klik produk yang akan ditambahkan di transaksi. User juga dapat menambah kuantitas produk dengan klik + ataupun mengurangi kuantitas dengan klik -. Setelah itu klik Selesaikan Transaksi untuk menyelesaikan proses penambahan transaksi

Ketika diinisiasi akan langsung fetch data dari database pada onInit(). Fungsi findTransactions() digunakan untuk mendapatkan seluruh data transaksi dari database. Sedangkan addTransaction() digunakan untuk menambah transaksi.

widget Scaffold untuk halaman kasir. endDrawer, digunakan untuk daftar produk dari productController. Pada body, ditampilkan daftar produk yang dipilih untuk transaksi. Jika belum ada produk, ditampilkan pesan cara menambahkan produk. Jika ada produk, ditampilkan daftar produk dengan opsi menambah atau mengurangi kuantitas. Total harga

ditampilkan di bagian bawah, serta tombol untuk menyelesaikan transaksi, yang hanya aktif jika ada produk dalam daftar.

```
@override
Codeium: Refactor | Explain | Generate Function Comment | X
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    appBar: AppBar(
      title: const Text('Halaman Kasir'),
      backgroundColor: Colors.white,
    ), // AppBar
    drawer: const Sidemenu(),
    endDrawer: Obx(() {
      if (productController.isLoading.value) {
        return const Center(child: CircularProgressIndicator());
      } else {
        return Drawer(
          backgroundColor: Colors.white,
          child: ListView.builder(
            itemCount: productController.products.length,
            itemBuilder: (context, index) {
              final product = productController.products[index];
              return ListTile(
                title: Text(product.name),
                subtitle: Text('Rp${product.price}'),
                trailing: const Icon(Icons.add), // Add icon here
                onTap: () {
                  addProduct(product);
                },
              ); // ListTile
            },
          ), // ListView.builder
        ); // Drawer
      }
    }), // Obx
    body: Obx(() {
      return Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          if (products.isEmpty)
            const Expanded(
              child: Center(
                child: Text(
                  'Belum ada produk di dalam daftar transaksi. Klik button kanan atas untuk men',
                  textAlign: TextAlign.center,
                  style: TextStyle(
                    color: Colors.black,

```

```

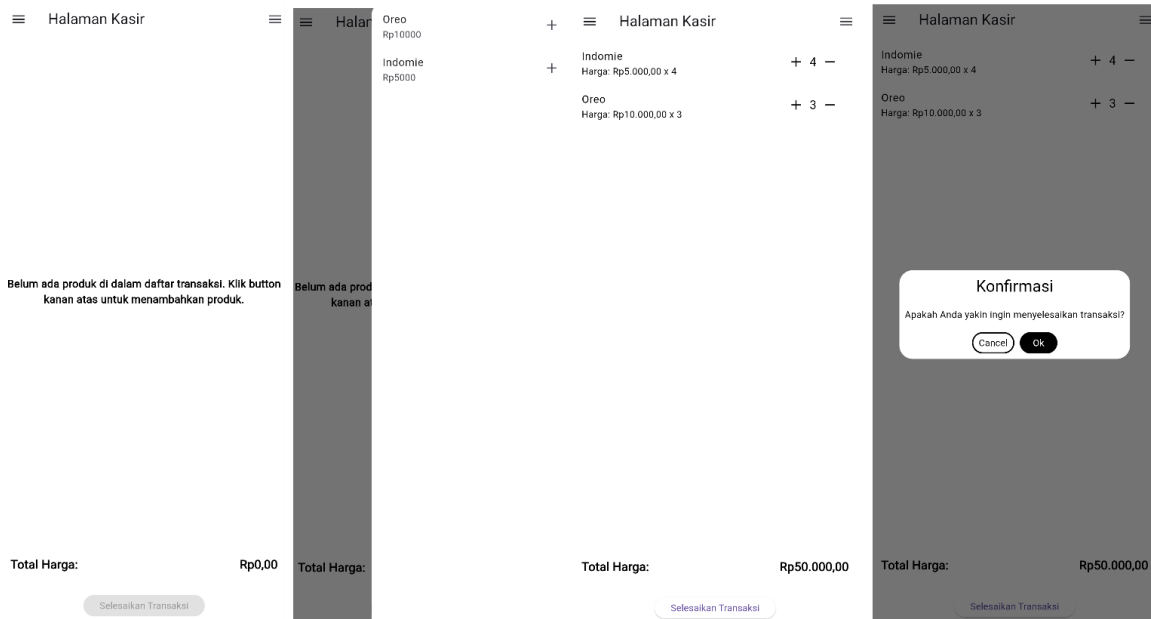
        fontSize: 16,
        fontWeight: FontWeight.bold,
      ), // TextStyle
    ), // Text
  ), // Center
), // Expanded
if (products.isNotEmpty)
Expanded(
  child: ListView.builder(
    itemCount: products.length,
    itemBuilder: (context, index) {
      final item = products[index];
      return ListTile(
        title: Text(item.name, style: const TextStyle(color: Colors.black)),
        subtitle: Text('Harga: ${Transaction.formatRupiah(item.price)} x ${item.quantity}', style: const
        trailing: Row(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            // Tombol tambah kuantitas "Tombol": Unknown word.
            IconButton(
              icon: const Icon(Icons.add, color: Colors.black),
              onPressed: () {
                _increaseQuantity(item); // Increase quantity
              },
            ), // IconButton
            Text(item.quantity.toString(), style: const TextStyle(color: Colors.black, fontSize: 18)),
            // Tombol kurangi kuantitas "Tombol": Unknown word.
            IconButton(
              icon: const Icon(Icons.remove, color: Colors.black),
              onPressed: () {
                _decreaseQuantity(item); // Decrease quantity
              },
            ), // IconButton
          ],
        ), // Row
      ); // ListTile
    }, // ListView.builder
  ), // Expanded
  Padding(
    padding: const EdgeInsets.all(16),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        You, 3 hours ago • features already fixed
        const Text(

```

```

        'Total Harga:',
        style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color: Colors.black),
      ), // Text
      Text(
        '${Transaction.formatRupiah(totalPrice as double)}', // Unnecessary use of string interpolation
        style: const TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color: Colors.black),
      ), // Text
    ],
  ), // Row
), // Padding
Padding(
  padding: const EdgeInsets.all(16),
  child: ElevatedButton(
    onPressed: products.isEmpty ? null : completeTransaction,
    child: const Text('Selesaikan Transaksi'), // The 'child' argument should be last in widget constructor
    style: ElevatedButton.styleFrom(backgroundColor: Colors.white),
  ), // ElevatedButton
), // Padding
],
); // Column
}), // Obx
); // Scaffold
}
}

```

e. Produk View

Halaman ini digunakan untuk mengelola daftar produk yang dapat ditambahkan ke transaksi

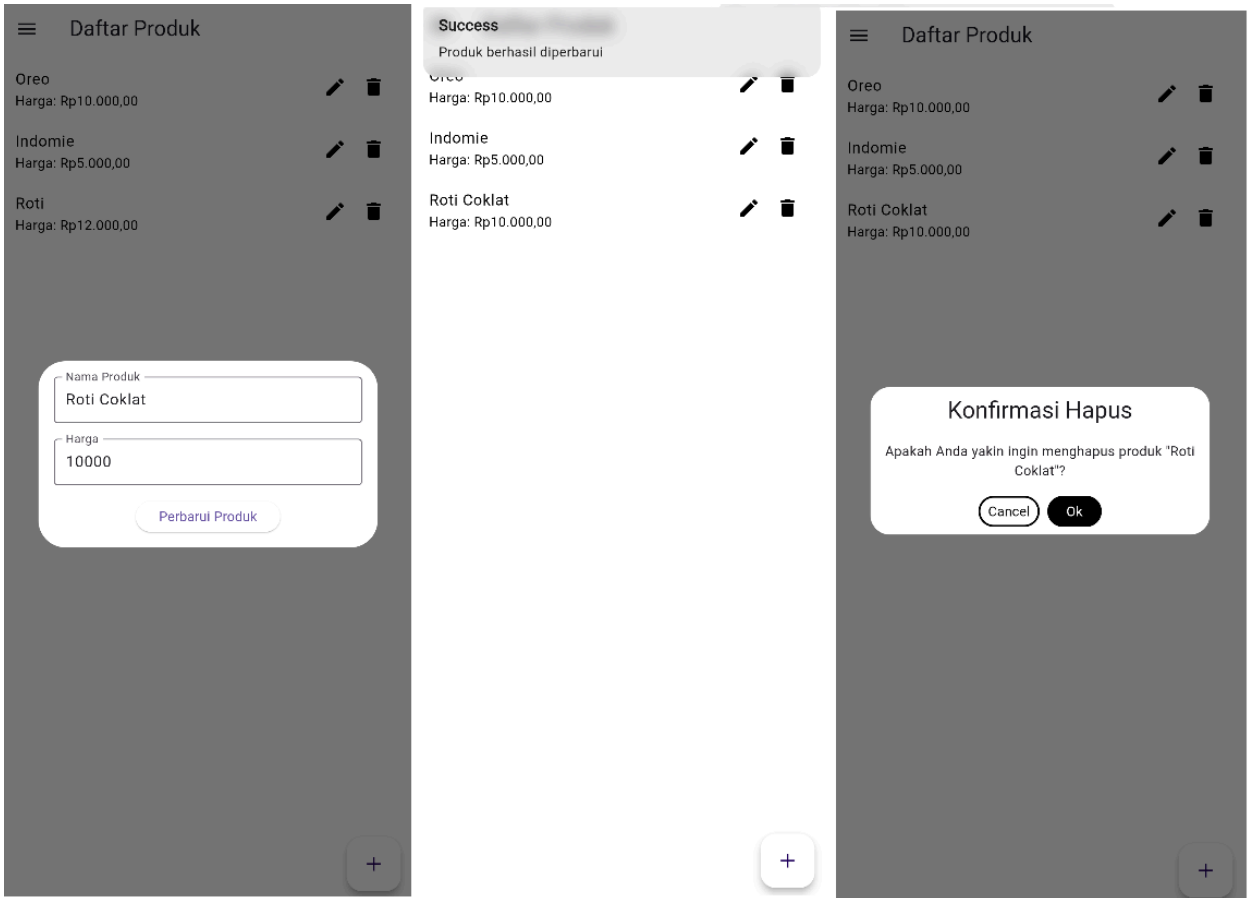
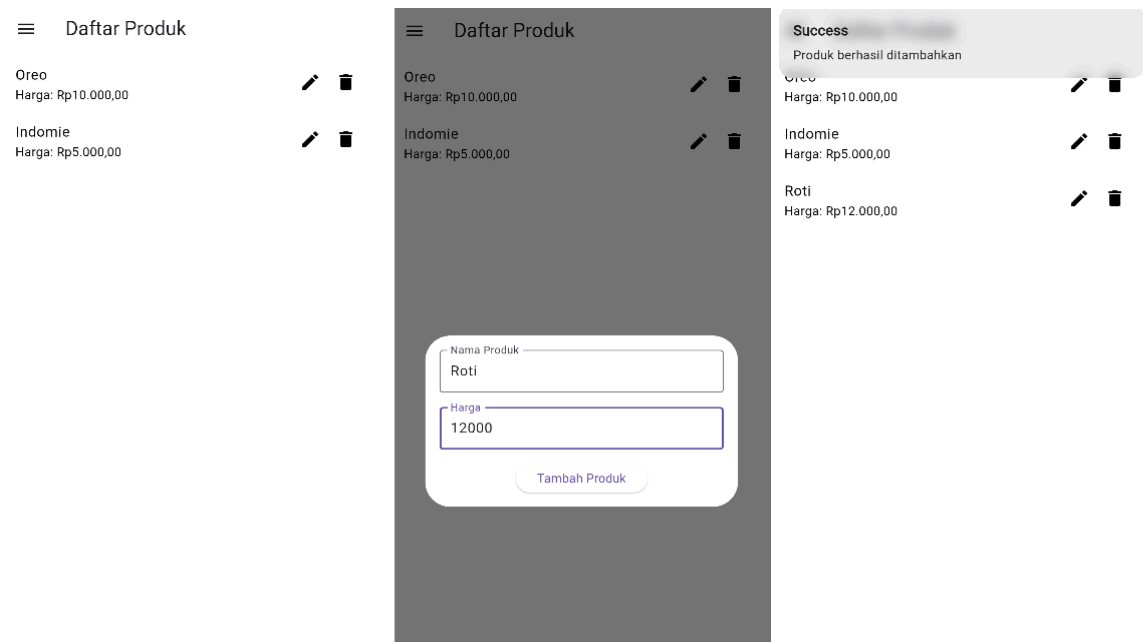
Widget `ProductView` menggunakan `Scaffold`. `AppBar` dengan judul "Daftar Produk" dan menu samping (Drawer) menggunakan widget `Sidemenu`. Dalam body, jika `controller.isLoading` bernilai `true`, maka ditampilkan `CircularProgressIndicator`. Jika tidak, ditampilkan daftar produk menggunakan `ListView.builder`. Setiap item dalam daftar memiliki nama produk, harga produk yang diformat dalam Rupiah, serta tombol untuk mengedit dan menghapus produk. Tombol `FloatingActionButton` digunakan untuk menambahkan produk baru dengan memanggil `_showProductDialog`.

```
class ProductView extends StatelessWidget {
```

```
@override You, 4 hours ago • product feature added
```

```
Codeium: Refactor | Explain | Generate Function Comment | X
```

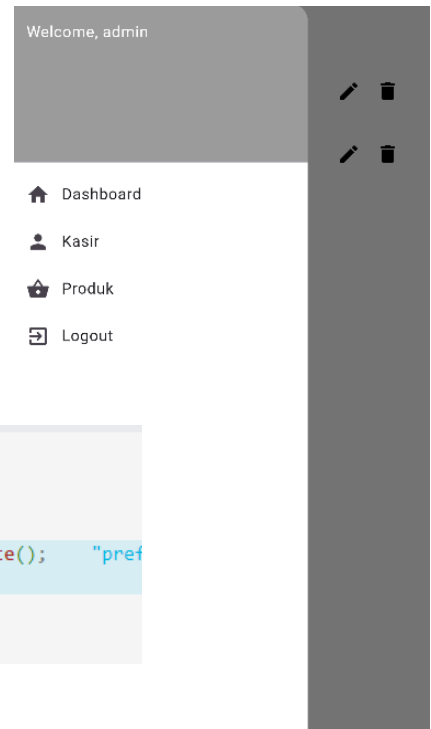
```
Widget build(BuildContext context) {  
  return Scaffold(  
    backgroundColor: Colors.white,  
    appBar: AppBar(  
      title: const Text('Daftar Produk'),  
      backgroundColor: Colors.white,  
    ), // AppBar  
    drawer: const Sidemenu(),  
    body: Obx(() {  
      if (controller.isLoading.value) {  
        return const Center(child: CircularProgressIndicator());  
      }  
  
      return ListView.builder(  
        itemCount: controller.products.length,  
        itemBuilder: (context, index) {  
          final product = controller.products[index];  
          return ListTile(  
            title: Text(product.name, style: const TextStyle(color: Colors.black)),  
            subtitle: Text('Harga: ${Transaction.formatRupiah(product.price)}', style: const TextStyle(color: Colors.black)),  
            trailing: Row(  
              mainAxisAlignment: MainAxisAlignment.min,  
              children: [  
                IconButton(  
                  icon: const Icon(Icons.edit, color: Colors.black),  
                  onPressed: () => _showProductDialog(context, product),  
                ), // IconButton  
                IconButton(  
                  icon: const Icon(Icons.delete, color: Colors.black),  
                  onPressed: () => _showDeleteDialog(product),  
                ), // IconButton  
              ],  
            ), // Row  
          ); // ListTile  
        },  
      ); // ListView.builder  
    }), // Obx  
    floatingActionButton: FloatingActionButton(  
      onPressed: () => _showProductDialog(context),  
      backgroundColor: Colors.white,  
      child: const Icon(Icons.add),  
    ), // FloatingActionButton  
  ); // Scaffold  
}
```



f. Logout

Logout dilakukan dengan klik sidebar, kemudian pilih menu **Logout**.

ListTile sebagai tombol logout. Ketika diklik, maka akan menghapus data username yang tersimpan di SharedPreferences menggunakan `prefs.remove('username')`. Setelah data dihapus, pengguna diarahkan ke halaman awal ('/') menggunakan `Get.toNamed('/')`.



```
ListTile(  
  leading: const Icon(Icons.exit_to_app_rounded),  
  title: const Text('Logout'),  
  onTap: () async {  
    SharedPreferences prefs = await SharedPreferences.getInstance();  
    await prefs.remove('username');  
    Get.toNamed('/');  
  },  
) // ListTile
```

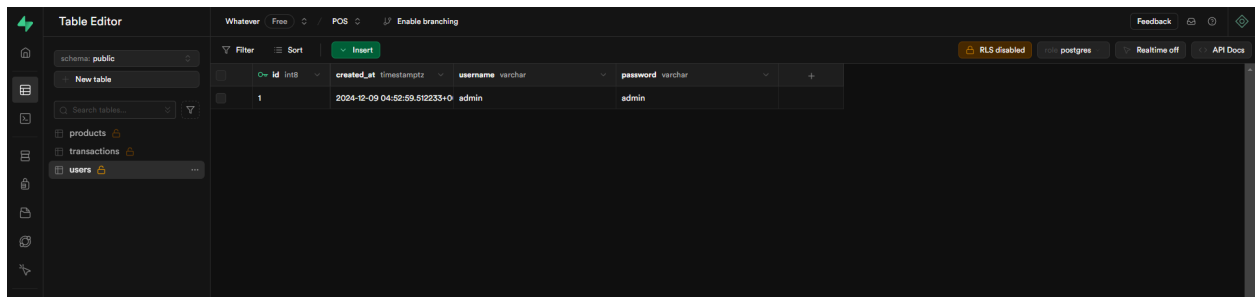
5. Supabase

Server yang digunakan adalah Supabase

Daftar Table:

- users
- transactions
- products

a. Table users



b. Table transactions

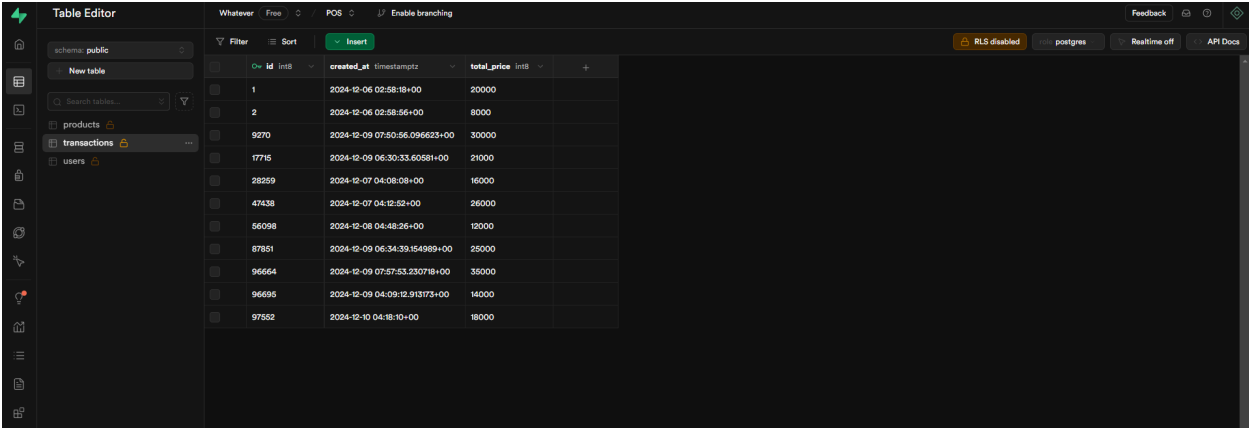


Table Editor interface showing the 'transactions' table. The table has columns: id (int8), created_at (timestamp), and total_price (int8). The table contains 10 rows of data.

id	created_at	total_price
1	2024-12-06 02:58:18+00	20000
2	2024-12-06 02:58:56+00	8000
9270	2024-12-09 07:50:56.096623+00	30000
17715	2024-12-09 06:30:33.60581+00	21000
28259	2024-12-07 04:08:08+00	16000
47438	2024-12-07 04:12:52+00	26000
56098	2024-12-08 04:48:26+00	12000
87851	2024-12-09 06:34:39.154989+00	25000
96664	2024-12-09 07:57:53.230718+00	35000
96695	2024-12-09 04:09:12.913173+00	14000
97552	2024-12-10 04:18:10+00	18000

c. Table products

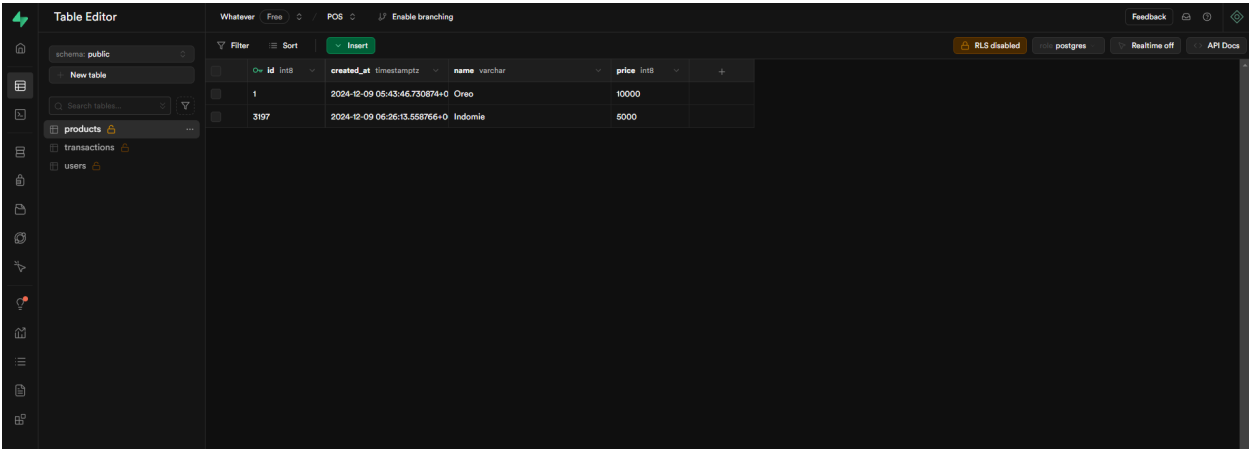


Table Editor interface showing the 'products' table. The table has columns: created_at (timestamp), name (varchar), and price (int8). The table contains 2 rows of data.

created_at	name	price
2024-12-09 05:43:46.730874+0	Oreo	10000
2024-12-09 06:26:13.558766+0	Indomie	5000