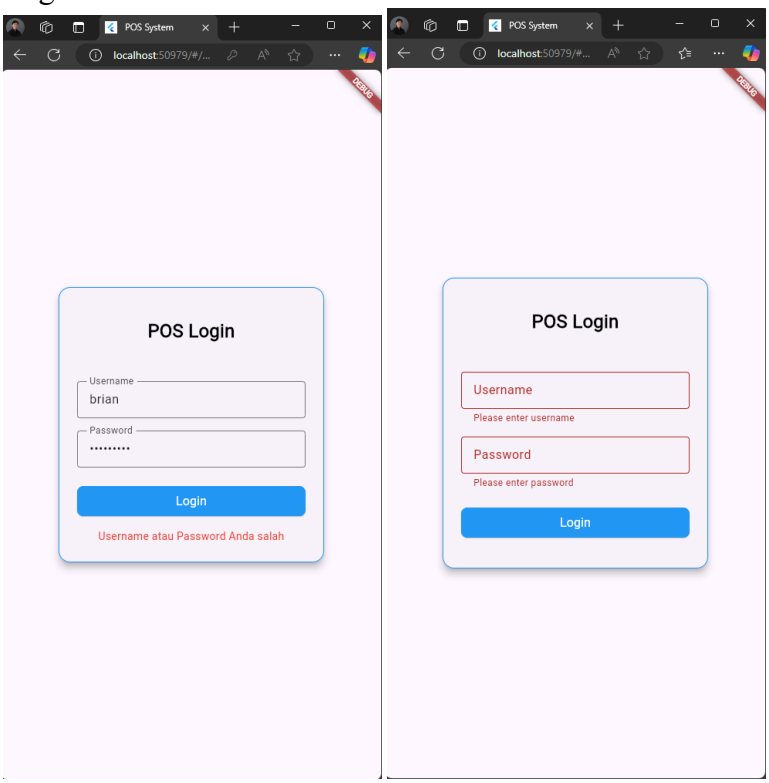
Nama : Brian Cahya Purnama
NIM : H1D022009
Mata Kuliah : Pemrograman Mobile B
Pengampu : Mohammad Irham Akbar, S.Kom., M.Cs.
Link GitHub : https://github.com/buriane/uas-mobile

Laporan UAS Pemrograman Mobile

1. Login



Potongan Kode:

```
//login_view.dart
        class LoginView extends StatelessWidget {
          final _formKey = GlobalKey<FormState>();
          final _usernameController = TextEditingController();
          final _passwordController = TextEditingController();
          final AuthController authController = Get.put(AuthController());

          @override
          Widget build(BuildContext context) {
            return Scaffold(
              body: Center(
                child: Container(
                  child: Form(
                    key: _formKey,
                    child: Column(
                      children: [
                        CustomInput(
                          label: 'Username',
                          controller: _usernameController,
                          validator: (value) {
                            if (value?.isEmpty ?? true) {
                              return 'Please enter username';
                            }
```

```
                            return null;
                          },
                        ),
                        CustomInput(
                          label: 'Password',
                          controller: _passwordController,
                          isPassword: true,
                          validator: (value) {
                            if (value?.isEmpty ?? true) {
                              return 'Please enter password';
                            }
                            return null;
                          },
                        ),
                        ElevatedButton(
                          onPressed: () async {
                            if (_formKey.currentState!.validate()) {
                              final success = await authController.login(
                                _usernameController.text,
                                _passwordController.text,
                              );
                              if (success) {
                                Get.offAllNamed('/dashboard');
                              }
                            }
                          },
                          child: Text('Login'),
                        ),
                      ],
                    ),
                  ),
                ),
              ),
            );
          }
}
```

Penjelasan:

Tampilan login dilengkapi dengan CustomInput widget untuk memasukkan username dan password. Desain card berwarna biru, judul "POS Login", dan tombol login yang terlihat jelas. Sistem memiliki validasi form untuk mencegah pengiriman data kosong. Lalu skrip PHP bertugas memproses request login. Ketika pengguna mengirimkan kredensial, sistem melakukan query ke database untuk memverifikasi kebenaran username dan password. Respon dari backend bersifat dinamis - berhasil login akan mengembalikan data pengguna dalam format JSON, sedangkan login gagal akan menghasilkan pesan error. Kemudian mengarahkan pengguna ke dashboard jika sukses atau menampilkan pesan kesalahan jika gagal.

2. Dashboard

Potongan Kode:

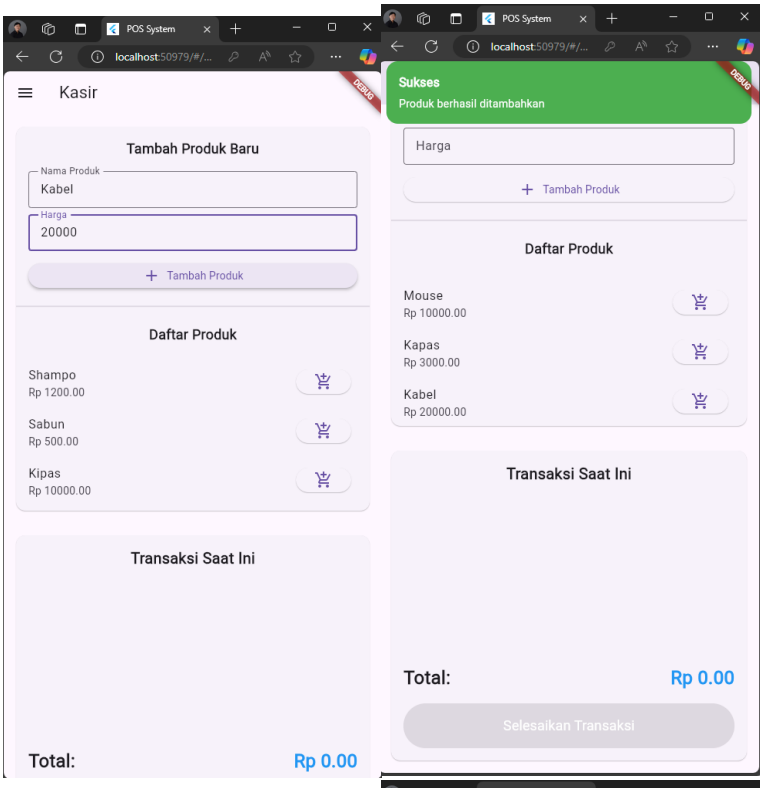
```
class DashboardView extends StatelessWidget {
  final DashboardController controller = Get.put(DashboardController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Dashboard POS')),
      body: RefreshIndicator(
        onRefresh: controller.fetchDashboardData,
        child: ListView(
          children: [
            // Kartu Informasi
            Row(
              children: [
                // Card Transaksi Harian
                Card(child: Column(
                  children: [
                    Icon(Icons.receipt_long),
                    Text('Transaksi Hari Ini'),
                    Obx(() => Text('${controller.transactionCount}'))
                  ]
                )),

                // Card Total Penjualan
                Card(child: Column(
                  children: [
                    Icon(Icons.payments_outlined),
                    Text('Total Penjualan'),
                    Obx(() => Text('Rp ${controller.todaySales}'))
                  ]
                )),
              ],
            ),

            // Grafik Penjualan Mingguan
            Obx(() => BarChart(
              data: controller.weeklySales.value,
              // Konfigurasi tambahan grafik
            ))
          ],
```

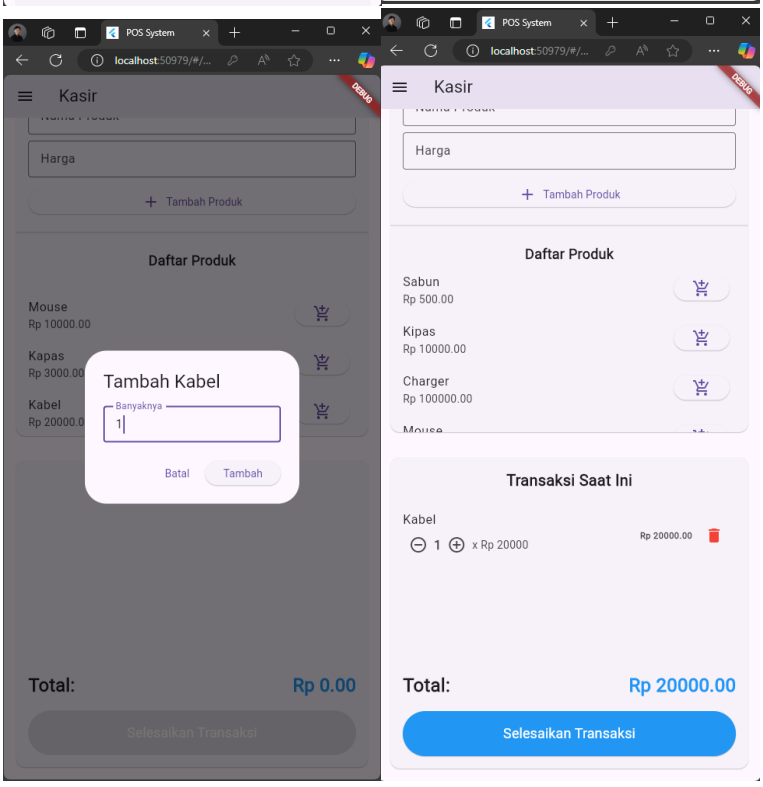
```
                ),
              ),
            );
          }
        }

        class DashboardController extends GetxController {
          final todaySales = 0.0.obs;
          final transactionCount = 0.obs;
          final weeklySales = <Map<String, dynamic>>[].obs;

          Future<void> fetchDashboardData() async {
            try {
              final response = await http.get('/api/dashboard.php');
              final data = json.decode(response.body);

              if (data['success']) {
                todaySales.value = data['total_sales'];
                transactionCount.value = data['transaction_count'];
                weeklySales.value = data['weekly_sales'];
              }
            } catch (e) {
              print('Gagal memuat dashboard: $e');
            }
          }
}
```

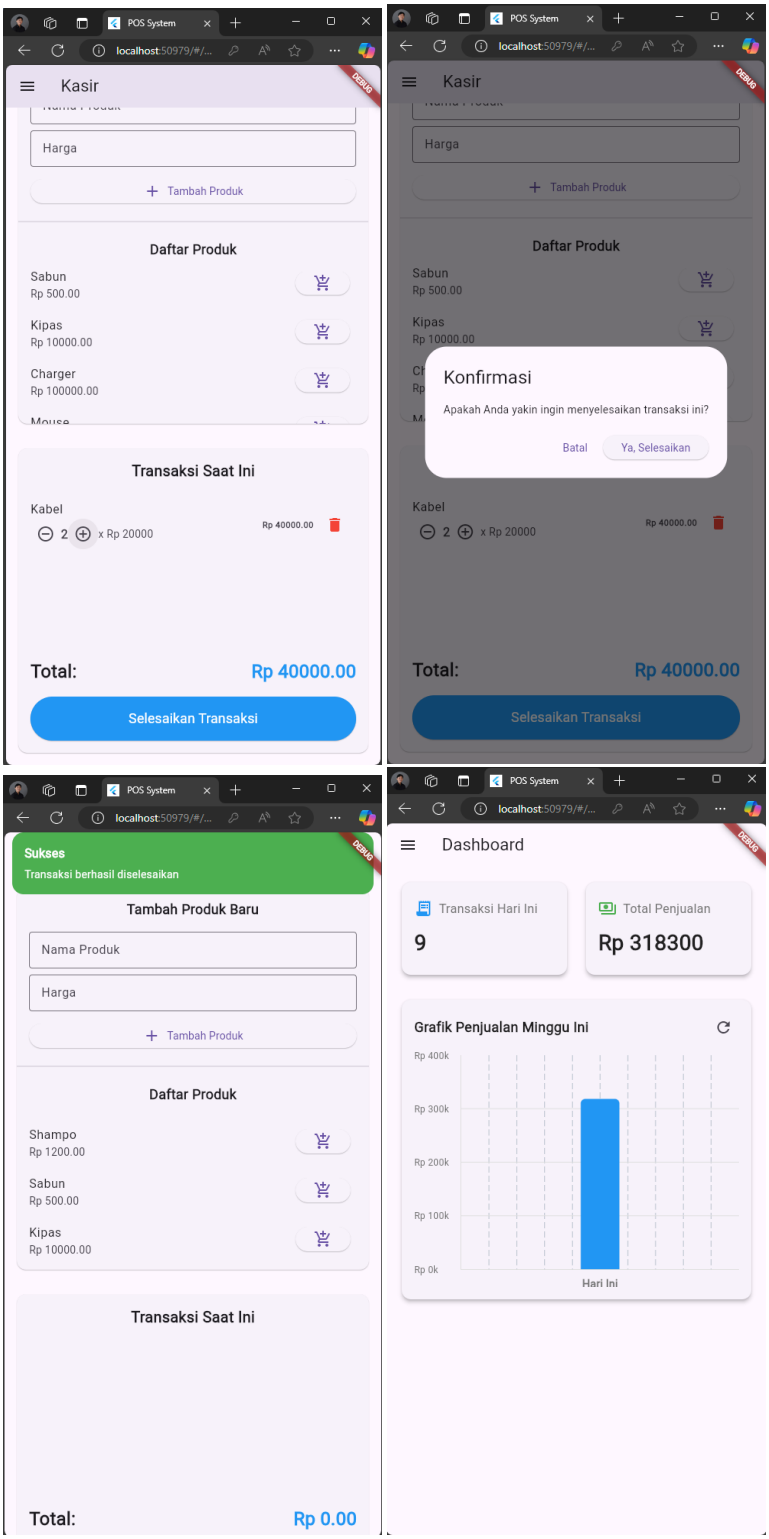
Penjelasan:

Dashboard aplikasi Point of Sale (POS) dirancang menggunakan Flutter dan GetX untuk memberikan gambaran komprehensif performa bisnis, menampilkan dua kartu informasi kritis (transaksi harian dan total penjualan) dan grafik penjualan mingguan. Dengan fitur pull-to-refresh dan mekanisme update real-time, dashboard memungkinkan pengguna melihat statistik bisnis secara cepat dan interaktif, menggunakan bar chart dengan tooltip untuk menampilkan perkembangan penjualan selama 1 minggu.

3. Cashier


POS System
localhost:50979/#/...
Kasir
Tambah Produk Baru
Nama Produk
Kabel
Harga
20000
Tambah Produk
Daftar Produk
Shampo
Rp 1200.00
Sabun
Rp 500.00
Kipas
Rp 10000.00
Transaksi Saat Ini
Total:
Rp 0.00
Sukses
Produk berhasil ditambahkan
Harga
Tambah Produk
Daftar Produk
Mouse
Rp 10000.00
Kapas
Rp 3000.00
Kabel
Rp 20000.00
Transaksi Saat Ini
Total:
Rp 0.00
Selesaikan Transaksi


POS System
localhost:50979/#/...
Kasir
Harga
Tambah Produk
Daftar Produk
Mouse
Rp 10000.00
Kapas
Rp 3000.00
Kabel
Rp 20000.00
Tambah Kabel
Banyaknya
1
Batal
Tambah
Total:
Rp 0.00
Selesaikan Transaksi
Kasir
Harga
Tambah Produk
Daftar Produk
Sabun
Rp 500.00
Kipas
Rp 10000.00
Charger
Rp 100000.00
Transaksi Saat Ini
Kabel
1
x Rp 20000
Rp 20000.00
Total:
Rp 20000.00
Selesaikan Transaksi



Potongan Kode:

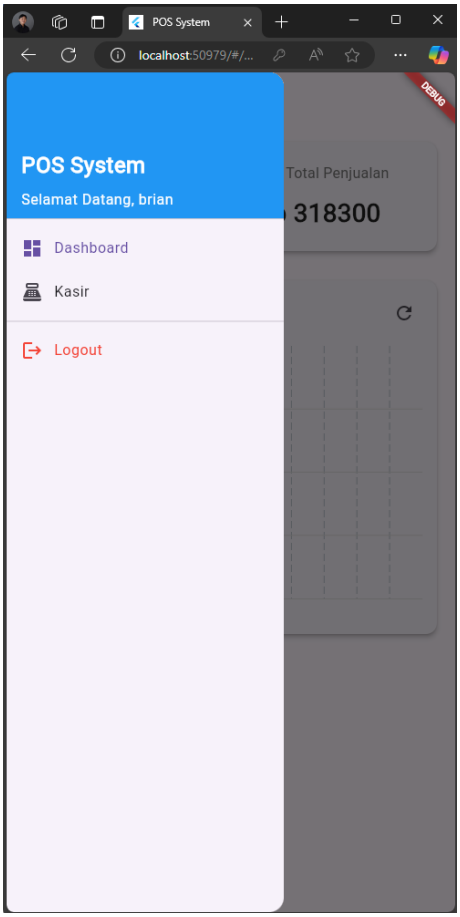
```
class CashierView extends StatelessWidget {
 final CashierController controller = Get.put(CashierController());

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     body: Row(
       children: [
         // Tambah Produk
         Form(child: Column(
```

```
                children: [
                  CustomInput(label: 'Nama Produk'),
                  CustomInput(label: 'Harga'),
                  ElevatedButton(
                    child: Text('Tambah Produk'),
                    onPressed: controller.addNewProduct
                  )
                ]
              )),

              // Daftar Produk
              Expanded(child: Obx(() => ListView.builder(
                itemCount: controller.products.length,
                itemBuilder: (context, index) {
                  var product = controller.products[index];
                  return ListTile(
                    title: Text(product.name),
                    subtitle: Text('Rp ${product.price}'),
                    trailing: IconButton(
                      icon: Icon(Icons.add_shopping_cart),
                      onPressed: () => controller.addToCart(product)
                    )
                  );
                }
              ))),

              // Keranjang Transaksi
              Container(
                width: 300,
                child: Obx(() => Column(
                  children: [
                    ListView.builder(
                      itemCount: controller.currentItems.length,
                      itemBuilder: (context, index) {
                        var item = controller.currentItems[index];
                        return ListTile(
                          title: Text(item.product.name),
                          subtitle: Text('Rp ${item.total}'),
                          trailing: Row(
                            mainAxisSize: MainAxisSize.min,
                            children: [
                              IconButton(icon: Icon(Icons.remove), onPressed: ()
        => controller.updateQuantity(index, false)),
                              Text('${item.quantity}'),
                              IconButton(icon: Icon(Icons.add), onPressed: () =>
        controller.updateQuantity(index, true))
                            ]
                          )
                        );
                      }
                    ),
                    Text('Total: Rp ${controller.total}'),
                    ElevatedButton(
                      child: Text('Proses Transaksi'),
                      onPressed: controller.completeTransaction
                    )
                  ]
                ))
              )
            ]
          )
        );
      }
      }

      class CashierController extends GetxController {
       final products = <Product>[].obs;
       final currentItems = <TransactionItem>[].obs;
       final total = 0.0.obs;

       Future<void> addNewProduct() async {
         // Logika tambah produk
       }

       void addToCart(Product product) {
         // Logika tambah ke keranjang
```

```
          }

          Future<void> completeTransaction() async {
            // Proses transaksi
          }
}
```

Penjelasan:
Fitur kasir dirancang sebagai antarmuka terintegrasi yang memungkinkan manajemen produk dan transaksi secara real-time. Terbagi menjadi tiga area utama: formulir tambah produk, daftar produk tersedia, dan keranjang transaksi (total harga). Menggunakan GetX untuk manajemen state, aplikasi menyediakan kontrol interaktif untuk menambah, mengubah kuantitas, menghapus produk dari keranjang, dan memproses penyelesaian transaksi dengan validasi.

4. Sidebar



Potongan Kode:

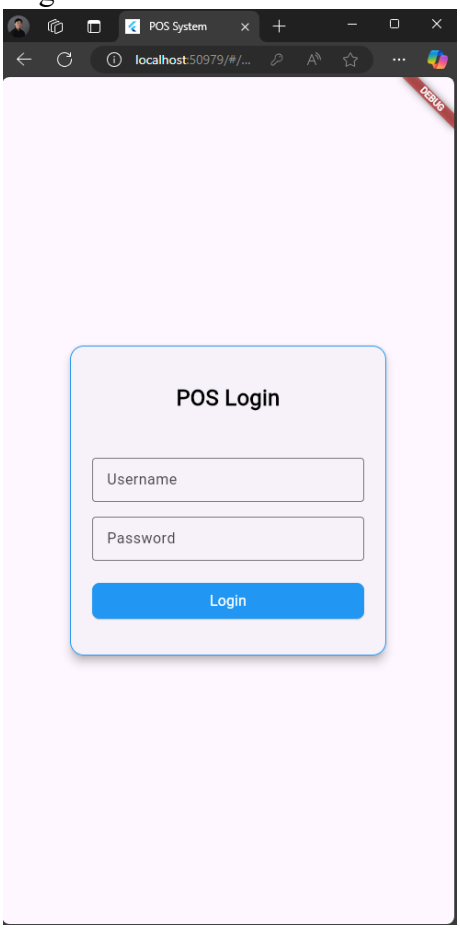
```
        class Sidebar extends StatelessWidget {
          final String currentRoute;
          final AuthController authController = Get.find();

          Sidebar({Key? key, required this.currentRoute}) : super(key: key);

          @override
          Widget build(BuildContext context) {
            return Drawer(
              child: ListView(
                children: [
                  // Header Drawer
```

```
                DrawerHeader(
                  decoration: BoxDecoration(color: Colors.blue),
                  child: Column(
                    children: [
                      Text('POS System', style: TextStyle(color: Colors.white)),
                      Obx(() => Text(
                        'Selamat Datang, ${authController.user.value?.username}',
                        style: TextStyle(color: Colors.white)
                      ))
                    ]
                  )
                ),

                // Menu Navigasi
                ListTile(
                  leading: Icon(Icons.dashboard),
                  title: Text('Dashboard'),
                  selected: currentRoute == '/dashboard',
                  onTap: () => Get.offNamed('/dashboard'),
                ),
                ListTile(
                  leading: Icon(Icons.point_of_sale),
                  title: Text('Kasir'),
                  selected: currentRoute == '/cashier',
                  onTap: () => Get.offNamed('/cashier'),
                ),

                // Logout Menu
                Divider(),
                ListTile(
                  leading: Icon(Icons.logout, color: Colors.red),
                  title: Text('Logout', style: TextStyle(color: Colors.red)),
                  onTap: () => authController.logout(),
                )
              ],
            ),
          );
        }
}
```

Penjelasan:

Sidebar pada aplikasi POS System saya ini dirancang sebagai komponen navigasi, menggunakan GetX untuk manajemen state dan navigasi. Sidebar menampilkan header dengan nama pengguna, menu utama untuk dashboard dan kasir dengan highlight menu aktif, serta tombol logout terpisah yang dapat memungkinkan pengguna dengan mudah berpindah antar fitur utama aplikasi sambil memberikan pengalaman pengguna yang efisien.

5. Logout



Potongan Kode:

```
        class AuthController extends GetxController {
         final user = Rx<User?>(null);

         void logout() {
           user.value = null;  // Hapus data user
           Get.offAllNamed('/login');  // Redirect ke login
         }
        }

        class LogoutButton extends StatelessWidget {
         final AuthController authController = Get.find();

         @override
         Widget build(BuildContext context) {
           return ListTile(
             leading: Icon(Icons.logout, color: Colors.red),
             title: Text('Logout', style: TextStyle(color: Colors.red)),
             onTap: () => authController.logout()
           );
         }
}
```

Penjelasan:

Implementasi logout dirancang untuk memberikan pengalaman pengguna yang aman. AuthController mengelola state autentikasi dengan metode logout yang membersihkan data pengguna dan mengarahkan kembali ke halaman login. Tombol logout didesain dengan warna merah untuk memberikan visual feedback yang jelas, sementara menggunakan GetX untuk navigasi dan manajemen state secara efisien.