

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

## 1. Proses Registrasi

### a. Mengisi Form Registrasi

Pengguna diminta untuk mengisi nama, email, password, dan konfirmasi password.

Kode terkait:

```
class RegistrasiPage extends StatefulWidget {
  const RegistrasiPage({Key? key}) : super(key: key);
  @override
  _RegistrasiPageState createState() => _RegistrasiPageState();
}

class _RegistrasiPageState extends State<RegistrasiPage> {
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  final _namaTextboxController = TextEditingController();
  final _emailTextboxController = TextEditingController();
  final _passwordTextboxController = TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Registrasi"),
      ),
      body: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(8.0),
          child: Form(
            key: _formKey,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                _namaTextField(),
                _emailTextField(),
                _passwordTextField(),
                _passwordKonfirmasiTextField(),
                _buttonRegistrasi()
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
    ),  
    ),  
);  
}  
  
Widget _namaTextField() {  
  return TextFormField(  
    decoration: const InputDecoration(labelText: "Nama"),  
    keyboardType: TextInputType.text,  
    controller: _namaTextboxController,  
    validator: (value) {  
      if (value!.length < 3) {  
        return "Nama harus diisi minimal 3 karakter";  
      }  
      return null;  
    },  
  );  
}  
  
Widget _emailTextField() {  
  return TextFormField(  
    decoration: const InputDecoration(labelText: "Email"),  
    keyboardType: TextInputType.emailAddress,  
    controller: _emailTextboxController,  
    validator: (value) {  
      if (value!.isEmpty) {  
        return 'Email harus diisi';  
      }  
      Pattern pattern =  
r'^(([^<>() []\.\.,;:\s@\"']+(\.[^<>() []\.\.\.,;:\s@\"']+)*|(\".+\"))@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\)|([a-zA-Z\d-0-9]+\.)+[a-zA-Z]{2,}))$';  
      RegExp regex = RegExp(pattern.toString());  
      if (!regex.hasMatch(value)) {  
        return "Email tidak valid";  
      }  
      return null;  
    },  
  );  
}  
  
Widget _passwordTextField() {
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
return TextFormField(  
  decoration: const InputDecoration(labelText: "Password"),  
  keyboardType: TextInputType.text,  
  obscureText: true,  
  controller: _passwordTextboxController,  
  validator: (value) {  
    if (value!.length < 3) {  
      return "Password harus diisi minimal 3 karakter";  
    }  
    return null;  
  },  
);  
}
```

#### b. Proses Pengiriman Data Registrasi

Setelah menekan tombol registrasi, aplikasi akan mengirim data ke API.

```
void _submit() {  
  _formKey.currentState!.save();  
  setState(() {  
    _isLoading = true;  
  });  
  RegistrasiBloc.registrasi(  
    nama: _namaTextboxController.text,  
    email: _emailTextboxController.text,  
    password: _passwordTextboxController.text,  
  ).then((value) {  
    if (value['status']) {  
      showDialog(  
        context: context,  
        barrierDismissible: false,  
        builder: (BuildContext context) => SuccessDialog(  
          description: "Registrasi berhasil, silahkan login",  
          okClick: () {  
            Navigator.pop(context);  
          },  
        ),  
      );  
    } else {  
      showDialog(  
        context: context,  
        barrierDismissible: false,  

```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
builder: (BuildContext context) => WarningDialog(  
  description: value['message'],  
),  
);  
}  
)).catchError((error) {  
  showDialog(  
    context: context,  
    barrierDismissible: false,  
    builder: (BuildContext context) => const WarningDialog(  
      description: "Registrasi gagal, silahkan coba lagi",  
    ),  
  );  
}).whenComplete(() {  
  setState(() {  
    _isLoading = false;  
  });  
});  
}  
}
```

c. Hasil Registrasi:

Pengguna akan melihat popup yang menginformasikan hasil registrasi.

## 2. Proses Login

a. Mengisi Form Login:

Pengguna diminta untuk memasukkan email dan password pada form login.

Kode terkait:

```
Widget _emailTextField() {  
  return TextFormField(  
    decoration: const InputDecoration(labelText: "Email"),  
    keyboardType: TextInputType.emailAddress,  
    controller: _emailTextboxController,  
    validator: (value) {  
      if (value!.isEmpty) {  
        return 'Email harus diisi';  
      }  
      return null;  
    },  
  );  
}
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
Widget _passwordTextField() {  
  return TextFormField(  
    decoration: const InputDecoration(labelText: "Password"),  
    keyboardType: TextInputType.text,  
    obscureText: true,  
    controller: _passwordTextboxController,  
    validator: (value) {  
      if (value!.isEmpty) {  
        return "Password harus diisi";  
      }  
      return null;  
    },  
  );  
}
```

#### b. Proses Autentikasi

Setelah menekan tombol login, aplikasi akan mengirim permintaan ke API untuk melakukan autentikasi (kode sama dengan diatas.

#### c. Hasil Login

Setelah proses autentikasi, pengguna akan melihat popup yang menginformasikan hasil login.

```
if (value['status']) {
```

```
  showDialog(  
    context: context,  
    barrierDismissible: false,  
    builder: (BuildContext context) => SuccessDialog(  
      description: "Registrasi berhasil, silahkan  
login",  
      okClick: () {  
        Navigator.pop(context);  
      },  
    ),  
  );  
} else {  
  showDialog(  
    context: context,  
    barrierDismissible: false,  
    builder: (BuildContext context) => ErrorDialog(  
      description: "Registrasi gagal, silahkan  
login",  
      okClick: () {  
        Navigator.pop(context);  
      },  
    ),  
  );  
}
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
        context: context,  
  
        barrierDismissible: false,  
  
        builder: (BuildContext context) => WarningDialog(  
  
            description: value['message'],  
  
        ),  
  
    );  
  
}
```

### 3. Menampilkan Daftar Account

#### a. Halaman Account

Setelah login berhasil, pengguna akan diarahkan ke halaman utama yang menampilkan daftar Account beserta saldo dan status account.

Kode terkait:

```
class ProdukPage extends StatefulWidget {  
  const ProdukPage({Key? key}) : super(key: key);  
  @override  
  _ProdukPageState createState() => _ProdukPageState();  
}  
  
class _ProdukPageState extends State<ProdukPage> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('List Account'),  
        actions: [  
          Padding(  
            padding: const EdgeInsets.only(right: 20.0),  
            child: GestureDetector(  
              child: const Icon(Icons.add, size: 26.0),  
              onTap: () async {  
                Navigator.push(context,  
                  MaterialPageRoute(builder: (context) =>  
ProdukForm())));  
              }  
            ),  
        ],  
      ),  
    );  
  }  
}
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
        },
      ),
    ],
  ),
  drawer: Drawer(
    child: ListView(
      children: [
        ListTile(
          title: const Text('Logout', style: TextStyle(color:
Colors.white)),
          trailing: const Icon(Icons.logout, color: Colors.white),
          onTap: () async {
            await LogoutBloc.logout().then((value) => {
              Navigator.of(context).pushAndRemoveUntil(
                MaterialPageRoute(builder: (context) =>
LoginPage()),
                (route) => false);
            });
          },
        ),
      ],
    ),
  ),
  body: FutureBuilder<List>(
    future: ProdukBloc.getProduk(),
    builder: (context, snapshot) {
      if (snapshot.hasError) print(snapshot.error);
      return snapshot.hasData
        ? ListProduk(
            list: snapshot.data,
          )
        : const Center(
            child: CircularProgressIndicator(),
          );
    },
  ),
);
}

class ListProduk extends StatelessWidget {
  final List? list;
  const ListProduk({Key? key, this.list}) : super(key: key);
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
@override
Widget build(BuildContext context) {
  return ListView.builder(
    itemCount: list == null ? 0 : list!.length,
    itemBuilder: (context, i) {
      return ItemProduk(
        produk: list![i],
      );
    });
}

class ItemProduk extends StatelessWidget {
  final Produk produk;
  const ItemProduk({Key? key, required this.produk}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => ProdukDetail(
              produk: produk,
            )),
        ),
      child: Card(
        child: ListTile(
          title: Text(produk.account!, style: const TextStyle(color:
Colors.white)),
          subtitle: Text(produk.status.toString(), style: const
TextStyle(color: Colors.white)),
        ),
      ),
    );
  }
}
```

#### b. Proses Pengambilan Data Account

Data Account diambil dari API menggunakan `ProdukBloc.getProduks()`.

Kode terkait:



NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
class ProdukBloc {  
  static Future<List<Produk>> getProduk() async {  
    String apiUrl = ApiUrl.listProduk;  
    var response = await Api().get(apiUrl);  
    var jsonObj = json.decode(response.body);  
    List<dynamic> listProduk = (jsonObj as Map<String,  
dynamic>)['data'];  
    List<Produk> produks = [];  
    for (int i = 0; i < listProduk.length; i++) {  
      produks.add(Produk.fromJson(listProduk[i]));  
    }  
    return produks;  
  }  
}
```

#### 4. Melihat Detail Account

##### a. Memilih Account dari Daftar Account

Pengguna dapat melihat detail Account dengan menekan item Account di halaman utama.

##### b. Halaman Detail Account

Halaman ini menampilkan informasi lengkap tentang Account yang dipilih.

Kode terkait:

```
import 'package:flutter/material.dart';  
import 'package:responsil/bloc/produk_bloc.dart';  
import 'package:responsil/model/produk.dart';  
import 'package:responsil/ui/produk_form.dart';  
import 'package:responsil/ui/produk_page.dart';  
import 'package:responsil/widget/success_dialog.dart';  
import 'package:responsil/widget/warning_dialog.dart';  
  
// ignore: must_be_immutable  
class ProdukDetail extends StatefulWidget {  
  Produk? produk;  
  ProdukDetail({Key? key, this.produk}) : super(key: key);  
  @override  
  _ProdukDetailState createState() => _ProdukDetailState();  
}  
  
class _ProdukDetailState extends State<ProdukDetail> {
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Detail Account'),
    ),
    body: Center(
      child: Column(
        children: [
          Text(
            "Account : ${widget.produk!.account}",
            style: const TextStyle(fontSize: 20.0, color:
Colors.white),
          ),
          Text(
            "Balance : Rp. ${widget.produk!.balance.toString()}",
            style: const TextStyle(fontSize: 18.0, color:
Colors.white),
          ),
          Text(
            "Status : ${widget.produk!.status}",
            style: const TextStyle(fontSize: 18.0, color:
Colors.white),
          ),
          _tombolHapusEdit()
        ],
      ),
    ),
  );
}

Widget _tombolHapusEdit() {
  return Row(
    mainAxisAlignment: MainAxisAlignment.min,
    children: [
      OutlinedButton(
        child: const Text(
          "EDIT",
          style: TextStyle(color: Colors.white),
        ),
        onPressed: () {
          Navigator.push(
            context,
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
        MaterialPageRoute(
          builder: (context) => ProdukForm(
            produk: widget.produk!,
          ),
        ),
      ),
    );
  },
),
OutlinedButton(
  child: const Text(
    "DELETE",
    style: TextStyle(color: Colors.white),
  ),
  onPressed: () => confirmHapus(),
),
],
);
}

void confirmHapus() {
  if (widget.produk?.id == null) {
    showDialog(
      context: context,
      builder: (BuildContext context) => const WarningDialog(
        description: "ID Account tidak ditemukan, tidak bisa
menghapus.",
      ),
    );
    return;
  }

  AlertDialog alertDialog = AlertDialog(
    content: const Text("Yakin ingin menghapus data ini?"),
    actions: [
      OutlinedButton(
        child: const Text("Ya"),
        onPressed: () async {
          bool success = await ProdukBloc.deleteProduk(
            id: widget.produk!.id!);
          if (success) {
            showDialog(
              context: context,
              barrierDismissible: false,
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
        builder: (BuildContext context) => SuccessDialog(  
            description: "Account berhasil dihapus",  
            onClick: () {  
                Navigator.of(context).pushReplacement(  
                    MaterialPageRoute(  
                        builder: (context) => const ProdukPage(),  
                    ),  
                );  
            },  
        ),  
    );  
} else {  
    // Jika penghapusan gagal  
    showDialog(  
        context: context,  
        builder: (BuildContext context) => const WarningDialog(  
            description: "Hapus gagal, silahkan coba lagi",  
        ),  
    );  
}  
},  
,  
),  
OutlinedButton(  
    child: const Text("Batal"),  
    onPressed: () => Navigator.pop(context),  
,  
],  
);  
showDialog(builder: (context) => alertDialog, context: context);  
}  
}
```

## 5. Menambah Account Baru

### a. Membuka Form Tambah Account

Pengguna dapat menambah Account baru dengan menekan ikon "+" di halaman utama.

### b. Mengisi Data Account Baru

Pengguna diminta untuk mengisi Bank, Saldo, dan Status.

Kode terkait:

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
class _ProdukFormState extends State<ProdukForm> {  
  final _formKey = GlobalKey<FormState>();  
  bool _isLoading = false;  
  String judul = "TAMBAH ACCOUNT";  
  String tombolSubmit = "SIMPAN";  
  final _accountTextboxController = TextEditingController();  
  final _balanceTextboxController = TextEditingController();  
  final _statusTextboxController = TextEditingController();  
  @override  
  void initState() {  
    super.initState();  
    isUpdate();  
  }  
  
  isUpdate() {  
    if (widget.produk != null) {  
      setState(() {  
        judul = "UBAH ACCOUNT";  
        tombolSubmit = "UBAH";  
        _accountTextboxController.text = widget.produk!.account!;  
        _balanceTextboxController.text =  
widget.produk!.balance.toString();  
        _statusTextboxController.text = widget.produk!.status!;  
      });  
    } else {  
      judul = "TAMBAH ACCOUNT";  
      tombolSubmit = "SIMPAN";  
    }  
  }  
}  
  
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(title: Text(judul)),  
    body: SingleChildScrollView(  
      child: Padding(  
        padding: const EdgeInsets.all(8.0),  
        child: Form(  
          key: _formKey,  
          child: Column(  
            children: [  
              _accountTextField(),  
              _balanceTextField(),  
            ],  
          ),  
        ),  
      ),  
    ),  
  );  
}
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
        _statusTextField(),
        _buttonSubmit()
    ],
    ),
),
),
),
);
}

Widget _accountTextField() {
  return TextFormField(
    decoration: const InputDecoration(labelText: "Account"),
    keyboardType: TextInputType.text,
    controller: _accountTextboxController,
    style: const TextStyle(color: Colors.white),
    validator: (value) {
      if (value!.isEmpty) {
        return "Account harus diisi";
      }
      return null;
    },
  );
}

Widget _balanceTextField() {
  return TextFormField(
    decoration: const InputDecoration(labelText: "Balance"),
    keyboardType: TextInputType.number,
    controller: _balanceTextboxController,
    style: const TextStyle(color: Colors.white),
    validator: (value) {
      if (value!.isEmpty) {
        return "Balance harus diisi";
      }
      return null;
    },
  );
}

Widget _statusTextField() {
  return TextFormField(
    decoration: const InputDecoration(labelText: "Status"),
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
keyboardType: TextInputType.text,  
controller: _statusTextboxController,  
style: const TextStyle(color: Colors.white),  
validator: (value) {  
  if (value!.isEmpty) {  
    return "Status harus diisi";  
  }  
  return null;  
},  
);  
}  
  
Widget _buttonSubmit() {  
  return OutlinedButton(  
    child: Text(tombolSubmit),  
    onPressed: () {  
      var validate = _formKey.currentState!.validate();  
      if (validate) {  
        if (!_isLoading) {  
          if (widget.produk != null) {  
            ubah();  
          } else {  
            simpan();  
          }  
        }  
      }  
    });  
}  
  
simpan() {  
  setState(() {  
    _isLoading = true;  
  });  
  Produk createProduk = Produk(id: null);  
  createProduk.account = _accountTextboxController.text;  
  createProduk.balance = int.parse(_balanceTextboxController.text);  
  createProduk.status = _statusTextboxController.text;  
  ProdukBloc.addProduk(produk: createProduk).then((value) {  
    if (value['status']) {  
      showDialog(  
        context: context,  
        barrierDismissible: false,  
        builder: (BuildContext context) => SuccessDialog(  

```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
        description: "Account berhasil ditambahkan",  
        onClick: () {  
            Navigator.of(context).pushReplacement(  
                MaterialPageRoute(  
                    builder: (BuildContext context) => const  
ProdukPage(),  
                ),  
            );  
        },  
    ),  
);  
} else {  
    showDialog(  
        context: context,  
        builder: (BuildContext context) => WarningDialog(  
            description: value['message'],  
        ),  
    );  
}  
)).catchError((error) {  
    showDialog(  
        context: context,  
        builder: (BuildContext context) => const WarningDialog(  
            description: "Simpan gagal, silahkan coba lagi",  
        ),  
    );  
}).whenComplete(() {  
    setState(() {  
        _isLoading = false;  
    });  
});  
}
```

### c. Proses Penyimpanan Account baru

Setelah menekan tombol simpan, aplikasi akan mengirim data ke API.

Kode terkait:

```
simpan() {  
    setState(() {  
        _isLoading = true;  
    });  
};
```



NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
Produk createProduk = Produk(id: null);
createProduk.account = _accountTextboxController.text;
createProduk.balance = int.parse(_balanceTextboxController.text);
createProduk.status = _statusTextboxController.text;
ProdukBloc.addProduk(produk: createProduk).then((value) {
  if (value['status']) {
    showDialog(
      context: context,
      barrierDismissible: false,
      builder: (BuildContext context) => SuccessDialog(
        description: "Account berhasil ditambahkan",
        onClick: () {
          Navigator.of(context).pushReplacement(
            MaterialPageRoute(
              builder: (BuildContext context) => const
ProdukPage(),
            ),
          );
        },
      ),
    );
  } else {
    showDialog(
      context: context,
      builder: (BuildContext context) => WarningDialog(
        description: value['message'],
      ),
    );
  }
}).catchError((error) {
  showDialog(
    context: context,
    builder: (BuildContext context) => const WarningDialog(
      description: "Simpan gagal, silahkan coba lagi",
    ),
  );
}).whenComplete(() {
  setState(() {
    _isLoading = false;
  });
});
}
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

d. Hasil Penambahan Account

Pengguna akan melihat popup yang menginformasikan hasil penambahan Account..

6. Mengubah Account

a. Membuka Form Ubah Account

Dari halaman detail Account, pengguna dapat menekan tombol "EDIT" untuk membuka form ubah Account.

b. Mengisi Perubahan Data Account

Form ubah Account akan terisi dengan data Account yang ada, dan pengguna dapat mengubahnya.

c. Proses Penyimpanan Perubahan Account

Setelah menekan tombol ubah, aplikasi akan mengirim data perubahan ke API.

Kode terkait:

```
ubah() {  
  setState(() {  
    _isLoading = true;  
  });  
  Produk updateProduk = Produk(id: widget.produk!.id!);  
  updateProduk.account = _accountTextboxController.text;  
  updateProduk.balance = int.parse(_balanceTextboxController.text);  
  updateProduk.status = _statusTextboxController.text;  
  ProdukBloc.updateProduk(produk: updateProduk).then((value) {  
    showDialog(  
      context: context,  
      barrierDismissible: false,  
      builder: (BuildContext context) => SuccessDialog(  
        description: "Account berhasil diubah",  
        onClick: () {  
          Navigator.of(context).push(  
            MaterialPageRoute(  
              builder: (BuildContext context) => const ProdukPage(),  
            ),  
          );  
        },  
      ),  
    );  
  }  
},
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
    ),  
    );  
  }, onError: (error) {  
    showDialog(  
      context: context,  
      builder: (BuildContext context) => const WarningDialog(  
        description: "Permintaan ubah data gagal, silahkan coba  
lagi",  
      ));  
    });  
  setState(() {  
    _isLoading = false;  
  });  
}  
}
```

#### d. Hasil Pengubahan Account

Pengguna akan melihat popup yang menginformasikan hasil perubahan account.

### 7. Menghapus Accoiunt

#### a. Memilih Account Untuk Dihapus

Dari halaman detail Account, pengguna dapat menekan tombol "DELETE" untuk menghapus Account.

#### b. Konfirmasi Penghapusan

Sebelum menghapus, aplikasi akan menampilkan dialog konfirmasi.

Kode terkait:

```
Widget _tombolHapusEdit() {  
  return Row(  
    mainAxisAlignment: MainAxisAlignment.min,  
    children: [  
      OutlinedButton(  
        child: const Text(  
          "EDIT",  
          style: TextStyle(color: Colors.white),  
        ),  
        onPressed: () {  
          Navigator.push(  

```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
        context,
        MaterialPageRoute(
          builder: (context) => ProdukForm(
            produk: widget.produk!,
          ),
        ),
      );
    },
  ),
  OutlinedButton(
    child: const Text(
      "DELETE",
      style: TextStyle(color: Colors.white),
    ),
    onPressed: () => confirmHapus(),
  ),
],
);
}

void confirmHapus() {
  if (widget.produk?.id == null) {
    showDialog(
      context: context,
      builder: (BuildContext context) => const WarningDialog(
        description: "ID Account tidak ditemukan, tidak bisa
menghapus.",
      ),
    );
    return;
  }

  AlertDialog alertDialog = AlertDialog(
    content: const Text("Yakin ingin menghapus data ini?"),
    actions: [
      OutlinedButton(
        child: const Text("Ya"),
        onPressed: () async {
          bool success = await ProdukBloc.deleteProduk(
            id: widget.produk!.id!);
          if (success) {
            showDialog(
              context: context,
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
        barrierDismissible: false,  
        builder: (BuildContext context) => SuccessDialog(  
            description: "Account berhasil dihapus",  
            onClick: () {  
                Navigator.of(context).pushReplacement(  
                    MaterialPageRoute(  
                        builder: (context) => const ProdukPage(),  
                    ),  
                );  
            },  
        ),  
    );  
} else {  
    // Jika penghapusan gagal  
    showDialog(  
        context: context,  
        builder: (BuildContext context) => const WarningDialog(  
            description: "Hapus gagal, silahkan coba lagi",  
        ),  
    );  
}  
},  
),  
OutlinedButton(  
    child: const Text("Batal"),  
    onPressed: () => Navigator.pop(context),  
),  
],  
);  
showDialog(builder: (context) => alertDialog, context: context);  
}  
}
```

### c. Proses Penghapusan

Jika pengguna mengkonfirmasi, aplikasi akan mengirim permintaan hapus ke API.

Kode terkait:

```
AlertDialog alertDialog = AlertDialog(  
    content: const Text("Yakin ingin menghapus data ini?"),  
    actions: [  
        TextButton(  
            onPressed: () => Navigator.pop(context),  
            child: const Text("Ya"),  
        ),  
        TextButton(  
            onPressed: () => Navigator.pop(context),  
            child: const Text("Tidak"),  
        ),  
    ],  
);
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
OutlinedButton(  
  child: const Text("Ya"),  
  onPressed: () async {  
    bool success = await ProdukBloc.deleteProduk(  
      id: widget.produk!.id!);  
    if (success) {  
      showDialog(  
        context: context,  
        barrierDismissible: false,  
        builder: (BuildContext context) => SuccessDialog(  
          description: "Account berhasil dihapus",  
          onClick: () {  
            Navigator.of(context).pushReplacement(  
              MaterialPageRoute(  
                builder: (context) => const ProdukPage(),  
              ),  
            );  
          },  
        ),  
      );  
    } else {  
      // Jika penghapusan gagal  
      showDialog(  
        context: context,  
        builder: (BuildContext context) => const WarningDialog(  
          description: "Hapus gagal, silahkan coba lagi",  
        ),  
      );  
    }  
  },  
)  
OutlinedButton(  
  child: const Text("Batal"),  
  onPressed: () => Navigator.pop(context),  
)  
],  
);  
showDialog(builder: (context) => alertDialog, context: context);  
}
```

d. Hasil Penghapusan

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

Pengguna akan melihat popup yang menginformasikan hasil penghapusan account.

## 8. Proses Logout

### a. Memilih Menu Logout

Pengguna dapat mengakses menu logout dari drawer aplikasi.

Kode terkait:

```
drawer: Drawer(  
  child: ListView(  
    children: [  
      ListTile(  
        title: const Text('Logout', style: TextStyle(color: Colors.white)),  
        trailing: const Icon(Icons.logout, color: Colors.white),  
        onTap: () async {  
          await LogoutBloc.logout().then((value) => {  
            Navigator.of(context).pushAndRemoveUntil(  
              MaterialPageRoute(builder: (context) =>  
LoginPage()),  
              (route) => false)  
            });  
          },  
        ),  
      ],  
    ),  
  ),  
),
```

### b. Proses Logout

Ketika pengguna memilih logout, aplikasi akan menghapus token dan informasi pengguna dari penyimpanan lokal.

Kode terkait:

```
drawer: Drawer(  
  child: ListView(  
    children: [  
      ListTile(  
        title: const Text('Logout', style: TextStyle(color: Colors.white)),  
        trailing: const Icon(Icons.logout, color: Colors.white),
```

NIM : H1D022030  
Nama : Bagus Wijoyoseno  
Shift Baru : D

```
onTap: () async {  
  await LogoutBloc.logout().then((value) => {  
    Navigator.of(context).pushAndRemoveUntil(  
      MaterialPageRoute(builder: (context) =>  
LoginPage()),  
      (route) => false)  
    });  
  },  
)  
],  
),  
),
```

c. Kembali ke Halaman Login

Setelah proses logout selesai, pengguna akan diarahkan kembali ke halaman login.

- NB: kata 'Produk' dapat dijadikan Saldo