

Nama : Dwita Isma Aprilia
NIM : H1D022034
Shift : B Shift Asli : B
Responsi

Hasil Modulo NIM : 2
3 digit terakhir : 034

Segment Satu

1. Paket: 2
2. Jenis Aplikasi: Aplikasi Manajemen Keuangan

Segment Dua

1. Digit Puluhan: 3
2. Nama Tabel: saldo
3. Kolom 1: id (int, PK, increment)
4. Kolom 2: account (String)
5. Kolom 3: balance (Integer)
6. Kolom 4: status (String)

Segment Tiga

1. Digit Satuan: 4
2. Kustomisasi Tampilan UI: Tema Gradien Biru-Ungu, Font Courier New

Source Code saldo_page.dart

```
import 'package:flutter/material.dart';
import 'package:keuangan/bloc/logout_bloc.dart';
import 'package:keuangan/bloc/saldo_bloc.dart';
import 'package:keuangan/model/saldo.dart';
import 'package:keuangan/ui/login_page.dart';
import 'package:keuangan/ui/saldo_detail.dart';
import 'package:keuangan/ui/saldo_form.dart';

class SaldoPage extends StatefulWidget {
  const SaldoPage({Key? key}) : super(key: key);
  @override
  _SaldoPageState createState() => _SaldoPageState();
}

class _SaldoPageState extends State<SaldoPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('List Saldo'),
```

```

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

```

```
}  
}
```

```
class ListSaldo extends StatelessWidget {  
  final List? list;  
  const ListSaldo({Key? key, this.list}) : super(key: key);  
  @override  
  Widget build(BuildContext context) {  
    return ListView.builder(  
      itemCount: list == null ? 0 : list!.length,  
      itemBuilder: (context, i) {  
        return ItemSaldo(  
          saldo: list![i],  
        );  
      });  
  }  
}
```

```
class ItemSaldo extends StatelessWidget {  
  final Saldo saldo;  
  const ItemSaldo({Key? key, required this.saldo}) : super(key: key);  
  @override  
  Widget build(BuildContext context) {  
    return GestureDetector(  
      onTap: () {  
        Navigator.push(  
          context,  
          MaterialPageRoute(  
            builder: (context) => SaldoDetail(  
              saldo: saldo,  
            )),  
        ),  
      child: Card(  
        child: ListTile(  
          title: Text(saldo.accountSaldo!),  
          subtitle: Text(saldo.statusSaldo.toString()),  
        ),  
      ),  
    );  
  }  
}
```

Source Code saldo_form.dart

```
import 'package:flutter/material.dart';
import 'package:keuangan/bloc/saldo_bloc.dart';
import 'package:keuangan/model/saldo.dart';
import 'package:keuangan/ui/saldo_page.dart';
import 'package:keuangan/widget/warning_dialog.dart';

// ignore: must_be_immutable
class SaldoForm extends StatefulWidget {
  Saldo? saldo;
  SaldoForm({Key? key, this.saldo}) : super(key: key);
  @override
  _SaldoFormState createState() => _SaldoFormState();
}

class _SaldoFormState extends State<SaldoForm> {
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  String judul = "TAMBAH SALDO";
  String tombolSubmit = "SIMPAN";
  final _accountSaldoTextboxController = TextEditingController();
  final _balanceSaldoTextboxController = TextEditingController();
  final _statusSaldoTextboxController = TextEditingController();

  @override
  void initState() {
    super.initState();
    isUpdate();
  }

  isUpdate() {
    if (widget.saldo != null) {
      setState(() {
        judul = "UBAH SALDO";
        tombolSubmit = "UBAH";
        _accountSaldoTextboxController.text = widget.saldo!.accountSaldo!;
        _balanceSaldoTextboxController.text = widget.saldo!.balanceSaldo.toString();
        _statusSaldoTextboxController.text = widget.saldo!.statusSaldo.toString();
      });
    } else {
      judul = "TAMBAH SALDO";
      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: [  
              _accountSaldoTextField(),  
              _balanceSaldoTextField(),  
              _statusSaldoTextField(),  
              _buttonSubmit()  
            ],  
          ),  
        ),  
      ),  
    ),  
  );  
}
```

```
//Membuat Textbox Kode Saldo
```

```
Widget _accountSaldoTextField() {  
  return TextFormField(  
    decoration: const InputDecoration(labelText: "Account Saldo"),  
    keyboardType: TextInputType.text,  
    controller: _accountSaldoTextboxController,  
    validator: (value) {  
      if (value!.isEmpty) {  
        return "Account Saldo harus diisi";  
      }  
      return null;  
    },  
  );  
}
```

```
//Membuat Textbox Balance Saldo
```

```
Widget _balanceSaldoTextField() {  
  return TextFormField(  

```

```

        decoration: const InputDecoration(labelText: "Balance Saldo"),
        keyboardType: TextInputType.text,
        controller: _balanceSaldoTextboxController,
        validator: (value) {
            if (value!.isEmpty) {
                return "Balance Saldo harus diisi";
            }
            return null;
        },
    );
}

```

//Membuat Textbox Status Saldo

```

Widget _statusSaldoTextField() {
    return TextFormField(
        decoration: const InputDecoration(labelText: "Status"),
        keyboardType: TextInputType.number,
        controller: _statusSaldoTextboxController,
        validator: (value) {
            if (value!.isEmpty) {
                return "Status harus diisi";
            }
            return null;
        },
    );
}

```

//Membuat Tombol Simpan/Ubah

```

Widget _buttonSubmit() {
    return OutlinedButton(
        child: Text(tombolSubmit),
        onPressed: () {
            var validate = _formKey.currentState!.validate();
            if (validate) {
                if (!_isLoading) {
                    if (widget.saldo != null) {
//kondisi update saldo
                        ubah();
                    } else {
//kondisi tambah saldo
                        simpan();
                    }
                }
            }
        },
    );
}

```

```

    }
  });
}

```

```

simpan() {
  setState(() {
    _isLoading = true;
  });
  Saldo createSaldo = Saldo(id: null);
  createSaldo.accountSaldo = _accountSaldoTextboxController.text;
  createSaldo.balanceSaldo = int.parse(_balanceSaldoTextboxController.text);
  createSaldo.statusSaldo = _statusSaldoTextboxController.text;
  SaldoBloc.addSaldo(saldo: createSaldo).then((value) {
    Navigator.of(context).push(MaterialPageRoute(
      builder: (BuildContext context) => const SaldoPage()));
  }, onError: (error) {
    showDialog(
      context: context,
      builder: (BuildContext context) => const WarningDialog(
        description: "Simpan gagal, silahkan coba lagi",
      ));
  });
  setState(() {
    _isLoading = false;
  });
}

```

```

ubah() {
  setState(() {
    _isLoading = true;
  });
  Saldo updateSaldo = Saldo(id: widget.saldo!.id!);
  updateSaldo.accountSaldo = _accountSaldoTextboxController.text;
  updateSaldo.balanceSaldo = int.tryParse(_balanceSaldoTextboxController.text) ?? 0;
  updateSaldo.statusSaldo = _statusSaldoTextboxController.text;
  SaldoBloc.updateSaldo(saldo: updateSaldo).then((value) {
    Navigator.of(context).push(MaterialPageRoute(
      builder: (BuildContext context) => const SaldoPage()));
  }, onError: (error) {
    showDialog(
      context: context,
      builder: (BuildContext context) => const WarningDialog(
        description: "Permintaan ubah data gagal, silahkan coba lagi",
      ));
  });
}

```

```

    ));
  });
  setState() {
    _isLoading = false;
  });
}
}

```

Source Code saldo_detail.dart

```

import 'package:flutter/material.dart';
import 'package:keuangan/bloc/saldo_bloc.dart';
import 'package:keuangan/model/saldo.dart';
import 'package:keuangan/ui/saldo_form.dart';
import 'package:keuangan/ui/saldo_page.dart';
import 'package:keuangan/widget/warning_dialog.dart';

```

```

// ignore: must_be_immutable
class SaldoDetail extends StatefulWidget {
  Saldo? saldo;

  SaldoDetail({Key? key, this.saldo}) : super(key: key);

  @override
  _SaldoDetailState createState() => _SaldoDetailState();
}

```

```

class _SaldoDetailState extends State<SaldoDetail> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Detail Saldo'),
      ),
      body: Center(
        child: Column(
          children: [
            Text(
              "Account : ${widget.saldo!.accountSaldo}",
              style: const TextStyle(fontSize: 20.0),
            ),
            Text(
              "Balance : ${widget.saldo!.balanceSaldo}",
              style: const TextStyle(fontSize: 18.0),
            ),
          ],
        ),
      ),
    );
  }
}

```



```

    ),
    Text(
      "Status : ${widget.saldo!.statusSaldo}",
      style: const TextStyle(fontSize: 18.0),
    ),
    _tombolHapusEdit()
  ],
),
),
);
}

```

```

Widget _tombolHapusEdit() {
  return Row(
    mainAxisAlignment: MainAxisAlignment.min,
    children: [
// Tombol Edit
      OutlinedButton(
        child: const Text("EDIT"),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => SaldoForm(
                saldo: widget.saldo!,
              ),
            ),
          );
        },
      ),
// Tombol Hapus
      OutlinedButton(
        child: const Text("DELETE"),
        onPressed: () => confirmHapus(),
      ),
    ],
  );
}

```

```

void confirmHapus() {
  AlertDialog alertDialog = AlertDialog(
    content: const Text("Yakin ingin menghapus data ini?"),
    actions: [

```

```

//tombol hapus
OutlinedButton(
  child: const Text("Ya"),
  onPressed: () {
    SaldoBloc.deleteSaldo(id: widget.saldo!.id!).then(
      (value) => {
        Navigator.off(context).push(MaterialPageRoute(
          builder: (context) => const SaldoPage()))
      }, onError: (error) {
        showDialog(
          context: context,
          builder: (BuildContext context) => const WarningDialog(
            description: "Hapus gagal, silahkan coba lagi",
          ));
      });
  },
),
//tombol batal
OutlinedButton(
  child: const Text("Batal"),
  onPressed: () => Navigator.pop(context),
)
],
);
showDialog(builder: (context) => alertDialog, context: context);
}
}

```

Source Code Registrasi

```

import 'package:flutter/material.dart';
import 'package:keuangan/bloc/registrasi_bloc.dart';
import 'package:keuangan/widget/success_dialog.dart';
import 'package:keuangan/widget/warning_dialog.dart';

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",
        style: TextStyle(
          fontFamily: 'Courier New',
          fontWeight: FontWeight.bold,
          color: Colors.white,
        ),
      ),
    ),
    flexibleSpace: Container(
      decoration: const BoxDecoration(
        gradient: LinearGradient(
          colors: [Colors.blue, Colors.purple],
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
        ),
      ),
    ),
    body: Container(
      decoration: const BoxDecoration(
        gradient: LinearGradient(
          colors: [Colors.blue, Colors.purple],
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
        ),
      ),
      child: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            children: [
              const SizedBox(height: 40),
              _logo(),
              const SizedBox(height: 40),
            ],
          ),
        ),
      ),
    ),
  );
}
```

```

        _registrationForm(),
        const SizedBox(height: 20),
        _buttonRegistrasi(),
        const SizedBox(height: 30),
        _menuLogin(),
      ],
    ),
  ),
),
);
}

```

// Logo atau ikon di atas form

```

Widget _logo() {
  return Center(
    child: Text(
      'Keuangan App',
      style: TextStyle(
        fontFamily: 'Courier New',
        fontSize: 30,
        fontWeight: FontWeight.bold,
        color: Colors.white,
      ),
    ),
  );
}

```

// Form Registrasi

```

Widget _registrationForm() {
  return Container(
    padding: const EdgeInsets.all(16.0),
    decoration: BoxDecoration(
      color: Colors.white.withOpacity(0.8),
      borderRadius: BorderRadius.circular(16),
    ),
    child: Form(
      key: _formKey,
      child: Column(
        children: [
          _namaTextField(),
          const SizedBox(height: 16),
          _emailTextField(),

```

```

        const SizedBox(height: 16),
        _passwordTextField(),
        const SizedBox(height: 16),
        _passwordKonfirmasiTextField(),
      ],
    ),
  ),
);
}

```

// Membuat Textbox Nama

```

Widget _namaTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Nama",
      labelStyle: TextStyle(fontFamily: 'Courier New', color: Colors.blue),
      enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.blue),
      ),
    ),
    keyboardType: TextInputType.text,
    controller: _namaTextboxController,
    style: const TextStyle(fontFamily: 'Courier New', color: Colors.black),
    validator: (value) {
      if (value!.length < 3) {
        return "Nama harus diisi minimal 3 karakter";
      }
      return null;
    },
  );
}

```

// Membuat Textbox Email

```

Widget _emailTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Email",
      labelStyle: TextStyle(fontFamily: 'Courier New', color: Colors.blue),
      enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.blue),
      ),
    ),
    keyboardType: TextInputType.emailAddress,

```

```

controller: _emailTextboxController,
style: const TextStyle(fontFamily: 'Courier New', color: Colors.black),
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\-0-9]+\.]|[a-zA-Z]{2,}))$';
  RegExp regex = RegExp(pattern.toString());
  if (!regex.hasMatch(value)) {
    return "Email tidak valid";
  }
  return null;
},
);
}

```

// Membuat Textbox Password

```

Widget _passwordTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Password",
      labelStyle: TextStyle(fontFamily: 'Courier New', color: Colors.blue),
      enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.blue),
      ),
    ),
    keyboardType: TextInputType.text,
    obscureText: true,
    controller: _passwordTextboxController,
    style: const TextStyle(fontFamily: 'Courier New', color: Colors.black),
    validator: (value) {
      if (value!.length < 6) {
        return "Password harus diisi minimal 6 karakter";
      }
      return null;
    },
  );
}

```

// Membuat Textbox Konfirmasi Password

```

Widget _passwordKonfirmasiTextField() {

```

```

return TextFormField(
  decoration: const InputDecoration(
    labelText: "Konfirmasi Password",
    labelStyle: TextStyle(fontFamily: 'Courier New', color: Colors.blue),
    enabledBorder: UnderlineInputBorder(
      borderSide: BorderSide(color: Colors.blue),
    ),
  ),
  keyboardType: TextInputType.text,
  obscureText: true,
  style: const TextStyle(fontFamily: 'Courier New', color: Colors.black),
  validator: (value) {
    if (value != _passwordTextboxController.text) {
      return "Konfirmasi Password tidak sama";
    }
    return null;
  },
);
}

```

// Tombol Registrasi

```

Widget _buttonRegistrasi() {
  return ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.blueAccent, // background color
      foregroundColor: Colors.white, // text (foreground) color
      textStyle: const TextStyle(fontFamily: 'Courier New'),
      padding: const EdgeInsets.symmetric(vertical: 14, horizontal: 40),
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30),
      ),
    ),
    child: const Text("Registrasi"),
    onPressed: () {
      var validate = _formKey.currentState!.validate();
      if (validate) {
        if (!_isLoading) _submit();
      }
    },
  );
}

```

// Fungsi untuk handle registrasi

```

void _submit() {
  _formKey.currentState!.save();
  setState(() {
    _isLoading = true;
  });
  RegistrasiBloc.registrasi(
    nama: _namaTextboxController.text,
    email: _emailTextboxController.text,
    password: _passwordTextboxController.text,
  ).then((value) {
    showDialog(
      context: context,
      barrierDismissible: false,
      builder: (BuildContext context) => SuccessDialog(
        description: "Registrasi berhasil, silahkan login",
        onClick: () {
          Navigator.pop(context);
          Navigator.pop(context); // Kembali ke halaman login
        },
      ),
    );
  }, onError: (error) {
    showDialog(
      context: context,
      barrierDismissible: false,
      builder: (BuildContext context) => const WarningDialog(
        description: "Registrasi gagal, silahkan coba lagi",
      ),
    );
  });
  setState(() {
    _isLoading = false;
  });
}

```

// Menu untuk berpindah ke halaman login

```

Widget _menuLogin() {
  return Center(
    child: InkWell(
      child: const Text(
        "Sudah punya akun? Login di sini",
        style: TextStyle(
          color: Colors.blueAccent,

```

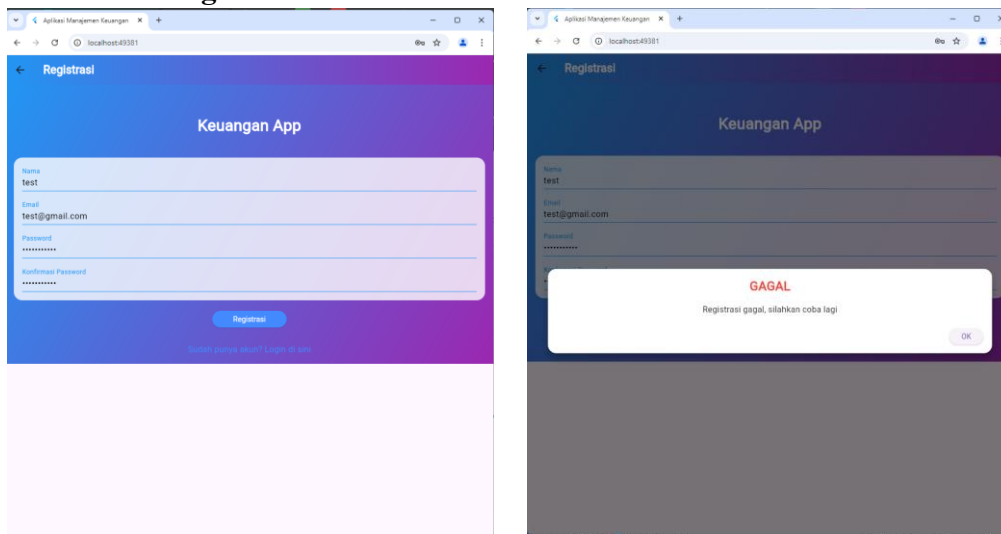


```

        fontFamily: 'Courier New',
        fontSize: 16,
      ),
    ),
    onTap: () {
      Navigator.pop(context); // Kembali ke halaman login
    },
  ),
);
}
}

```

Screenshot Registrasi



Source Code Login

```

import 'package:flutter/material.dart';
import 'package:keuangan/bloc/login_bloc.dart';
import 'package:keuangan/helpers/user_info.dart';
import 'package:keuangan/ui/saldo_page.dart';
import 'package:keuangan/ui/registrasi_page.dart';
import 'package:keuangan/widget/warning_dialog.dart';

```

```

class LoginPage extends StatefulWidget {
  const LoginPage({Key? key}) : super(key: key);
  @override
  _LoginPageState createState() => _LoginPageState();
}

```

```

class _LoginPageState extends State<LoginPage> {

```

```

final _formKey = GlobalKey<FormState>();
bool _isLoading = false;
final _emailTextboxController = TextEditingController();
final _passwordTextboxController = TextEditingController();

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text(
        'Login',
        style: TextStyle(fontFamily: 'Courier New', fontWeight: FontWeight.bold),
      ),
    ),
    flexibleSpace: Container(
      decoration: const BoxDecoration(
        gradient: LinearGradient(
          colors: [Colors.blue, Colors.purple],
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
        ),
      ),
    ),
    body: Container(
      decoration: const BoxDecoration(
        gradient: LinearGradient(
          colors: [Colors.blue, Colors.purple],
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
        ),
      ),
      child: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            children: [
              const SizedBox(height: 40),
              _logo(),
              const SizedBox(height: 40),
              _loginForm(),
              const SizedBox(height: 20),
              _buttonLogin(),
              const SizedBox(height: 30),
            ],
          ),
        ),
      ),
    ),
  );
}

```

```

        _menuRegistrasi(),
      ],
    ),
  ),
),
);
}

```

// Logo atau ikon di atas form

```

Widget _logo() {
  return Center(
    child: Text(
      'Keuangan App',
      style: TextStyle(
        fontFamily: 'Courier New',
        fontSize: 30,
        fontWeight: FontWeight.bold,
        color: Colors.white,
      ),
    ),
  );
}

```

// Membuat form login

```

Widget _loginForm() {
  return Container(
    padding: const EdgeInsets.all(16.0),
    decoration: BoxDecoration(
      color: Colors.white.withOpacity(0.8),
      borderRadius: BorderRadius.circular(16),
    ),
    child: Form(
      key: _formKey,
      child: Column(
        children: [
          _emailTextField(),
          const SizedBox(height: 16),
          _passwordTextField(),
        ],
      ),
    ),
  );
}

```

```
}
```

```
// Membuat Textbox email
```

```
Widget _emailTextField() {  
  return TextFormField(  
    decoration: const InputDecoration(  
      labelText: "Email",  
      labelStyle: TextStyle(fontFamily: 'Courier New', color: Colors.blue),  
      enabledBorder: UnderlineInputBorder(  
        borderSide: BorderSide(color: Colors.blue),  
      ),  
    ),  
    keyboardType: TextInputType.emailAddress,  
    controller: _emailTextboxController,  
    style: const TextStyle(fontFamily: 'Courier New', color: Colors.black),  
    validator: (value) {  
      if (value!.isEmpty) {  
        return 'Email harus diisi';  
      }  
      return null;  
    },  
  );  
}
```

```
// Membuat Textbox password
```

```
Widget _passwordTextField() {  
  return TextFormField(  
    decoration: const InputDecoration(  
      labelText: "Password",  
      labelStyle: TextStyle(fontFamily: 'Courier New', color: Colors.blue),  
      enabledBorder: UnderlineInputBorder(  
        borderSide: BorderSide(color: Colors.blue),  
      ),  
    ),  
    keyboardType: TextInputType.text,  
    obscureText: true,  
    controller: _passwordTextboxController,  
    style: const TextStyle(fontFamily: 'Courier New', color: Colors.black),  
    validator: (value) {  
      if (value!.isEmpty) {  
        return "Password harus diisi";  
      }  
      return null;  
    },  
  );  
}
```

```

    },
  );
}

// Membuat Tombol Login
Widget _buttonLogin() {
  return ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.blueAccent, // Warna latar belakang tombol
      foregroundColor: Colors.white, // Warna teks tombol
      textStyle: const TextStyle(fontFamily: 'Courier New', fontSize: 16),
      padding: const EdgeInsets.symmetric(vertical: 14, horizontal: 40),
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30),
      ),
    ),
    child: const Text("Login"),
    onPressed: () {
      var validate = _formKey.currentState!.validate();
      if (validate) {
        if (!_isLoading) _submit();
      }
    },
  );
}

void _submit() {
  _formKey.currentState!.save();
  setState(() {
    _isLoading = true;
  });
  LoginBloc.login(
    email: _emailTextboxController.text,
    password: _passwordTextboxController.text
  ).then((value) async {
    if (value.code == 200) {
      await UserInfo().setToken(value.token.toString());
      await UserInfo().setUserID(int.parse(value.userID.toString()));
      Navigator.pushReplacement(context,
        MaterialPageRoute(builder: (context) => const SaldoPage()));
    } else {
      showDialog(
        context: context,

```

```

        barrierDismissible: false,
        builder: (BuildContext context) => const WarningDialog(
          description: "Login gagal, silahkan coba lagi",
        ));
      }
    }, onError: (error) {
      print(error);
      showDialog(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) => const WarningDialog(
          description: "Login gagal, silahkan coba lagi",
        ));
    });
    setState(() {
      _isLoading = false;
    });
  }
}

```

// Membuat menu untuk membuka halaman registrasi

```

Widget _menuRegistrasi() {
  return Center(
    child: InkWell(
      child: const Text(
        "Registrasi",
        style: TextStyle(
          color: Colors.blueAccent,
          fontFamily: 'Courier New',
          fontSize: 16,
        ),
      ),
      onTap: () {
        Navigator.push(context,
          MaterialPageRoute(builder: (context) => const RegistrasiPage()));
      },
    ),
  );
}

```

Screenshot Login

