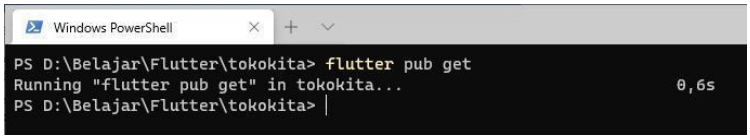
Membuat Helper Modul

Menambahkan depedencies

Dalam pembuatan aplikasi ini dibutuhkan depedensi untuk menerima dan mengirim request ke Rest API (http) dan depedensi untuk menyimpan token (shared_preferences). Pastikan komputer atau laptop terhubung ke internet, buka file pubspec.yaml kemudian pada bagian dependencies tambahkan kode berikut

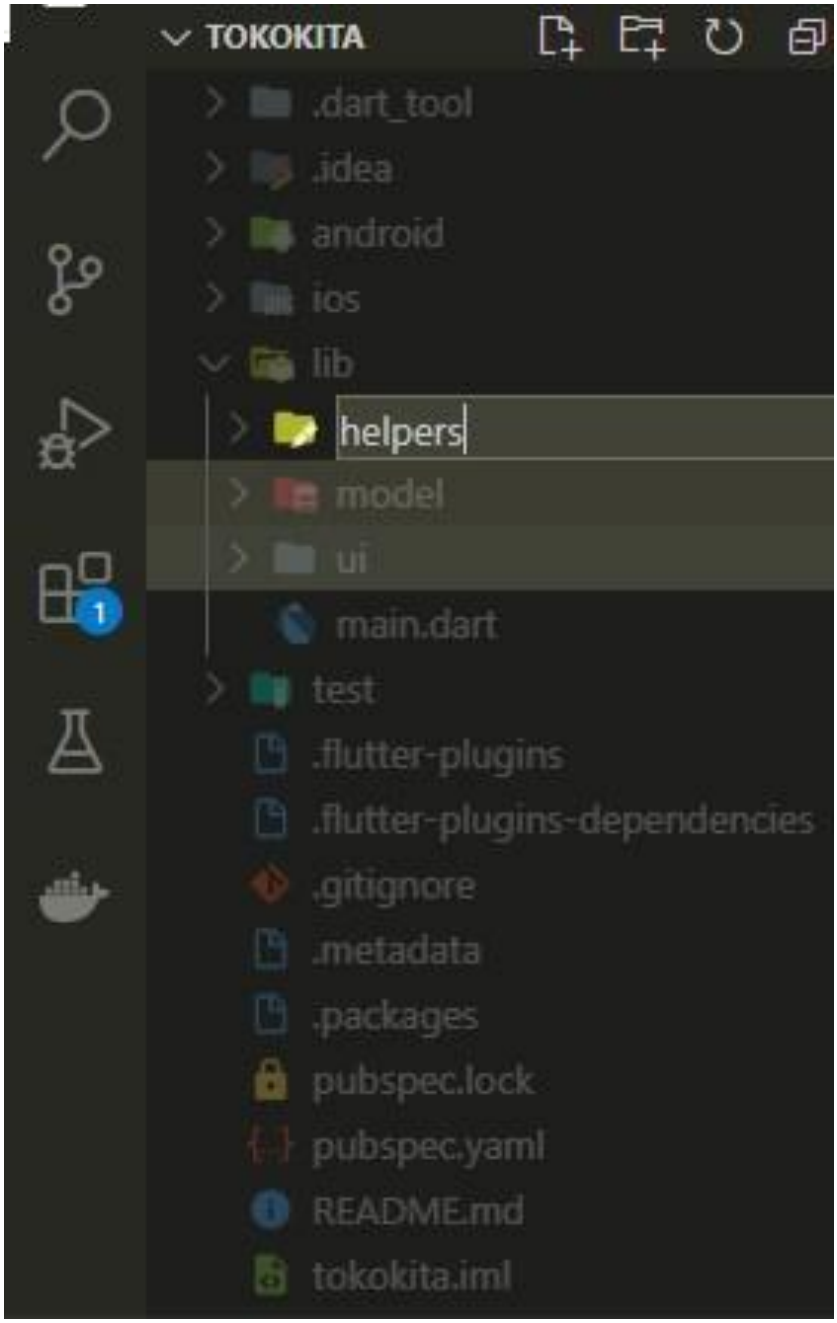
```
29. dependencies:
30.   flutter:
31.     sdk: flutter
32.
33.
34.   # The following adds the Cupertino Icons font to your application.
35.   # Use with the CupertinoIcons class for iOS style icons.
36.   cupertino_icons: ^1.0.2
37.   http: ^0.13.4
38.   shared_preferences: ^2.0.11
```

Untuk mengunduh depedencies atau package yang telah ditambahkan, buka CommandPrompt kemudian masuk ke folder projek dan ketikkan flutter pub get kemudian tekan Enter



Membuat Class Token

Buat sebuah folder dengan nama helpers



Kemudian pada folder helpers buat sebuah file dengan nama user_info.dart dan masukkan kode berikut

```
import 'package:shared_preferences/shared_preferences.dart';

class UserInfo {
  Future setToken(String value) async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
    return pref.setString("token", value);
  }

  Future<String?> getToken() async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
```

```
    return pref.getString("token");
  }

  Future setUserID(int value) async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
    return pref.setInt("userID", value);
  }

  Future<int?> getUserID() async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
    return pref.getInt("userID");
  }

  Future logout() async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
    pref.clear();
  }
}
```

Http request

Membuat Modul Error Handling

Buat sebuah file pada folder helpers dengan nama app_exception.dart kemudian ketikkan kode berikut

```
class AppException implements Exception {
  final _message;
  final _prefix;
  AppException([this._message, this._prefix]);
  @override
  String toString() {
    return "$_prefix$_message";
  }
}

class FetchDataException extends AppException {
  FetchDataException([String? message])
      : super(message, "Error During Communication: ");
}

class BadRequestException extends AppException {
```

```
  BadRequestException([message]) : super(message, "Invalid Request: ");
}

class UnauthorisedException extends AppException {
  UnauthorisedException([message]) : super(message, "Unauthorised: ");
}

class UnprocessableEntityException extends AppException {
  UnprocessableEntityException([message])
      : super(message, "Unprocessable Entity: ");
}

class InvalidInputException extends AppException {
  InvalidInputException([String? message]) : super(message, "Invalid Input: ");
}
```

Pada file ini berfungsi sebagai penanganan jika terjadi error saat melakukan permintaan atau pengiriman ke Rest API

Membuat Modul Http Request

Agar dapat mengirim atau menerima permintaan ke Rest API, dibuat sebuah fungsi baik itu method POST, GET dan DELETE.

Buat sebuah file di dalam folder helpers dengan nama api.dart dan masukkan kode berikut

```
import 'dart:io';
import 'package:http/http.dart' as http;
import 'package:tokokita/helpers/user_info.dart';
import 'app_exception.dart';

class Api {
  Future<dynamic> post(dynamic url, dynamic data) async {
    var token = await UserInfo().getToken();
    var responseJson;
    try {
      final response = await http.post(Uri.parse(url),
          body: data,
          headers: {HttpHeaders.authorizationHeader: "Bearer $token"});
      responseJson = _returnResponse(response);
    } on SocketException {
      throw FetchDataException('No Internet connection');
    }
    return responseJson;
```

```
}

Future<dynamic> get(dynamic url) async {
  var token = await UserInfo().getToken();
  var responseJson;
  try {
    final response = await http.get(Uri.parse(url),
        headers: {HttpHeaders.authorizationHeader: "Bearer $token"});
    responseJson = _returnResponse(response);
  } on SocketException {
    throw FetchDataException('No Internet connection');
  }
  return responseJson;
}

Future<dynamic> put(dynamic url, dynamic data) async {
  var token = await UserInfo().getToken();
  var responseJson;
  try {
    final response = await http.put(Uri.parse(url), body: data, headers: {
      HttpHeaders.authorizationHeader: "Bearer $token",
      HttpHeaders.contentTypeHeader: "application/json"
    });
    responseJson = _returnResponse(response);
  } on SocketException {
    throw FetchDataException('No Internet connection');
  }
  return responseJson;
}

Future<dynamic> delete(dynamic url) async {
  var token = await UserInfo().getToken();
  var responseJson;
  try {
    final response = await http.delete(Uri.parse(url),
        headers: {HttpHeaders.authorizationHeader: "Bearer $token"});
    responseJson = _returnResponse(response);
  } on SocketException {
    throw FetchDataException('No Internet connection');
  }
  return responseJson;
}
```

```
  dynamic _returnResponse(http.Response response) {
    switch (response.statusCode) {
      case 200:
        return response;
      case 400:
        throw BadRequestException(response.body.toString());
      case 401:
      case 403:
        throw UnauthorisedException(response.body.toString());
      case 422:
        throw InvalidInputException(response.body.toString());
      case 500:
      default:
        throw FetchDataException(
            'Error occured while Communication with Server with StatusCode :
${response.statusCode}');
    }
  }
}
```

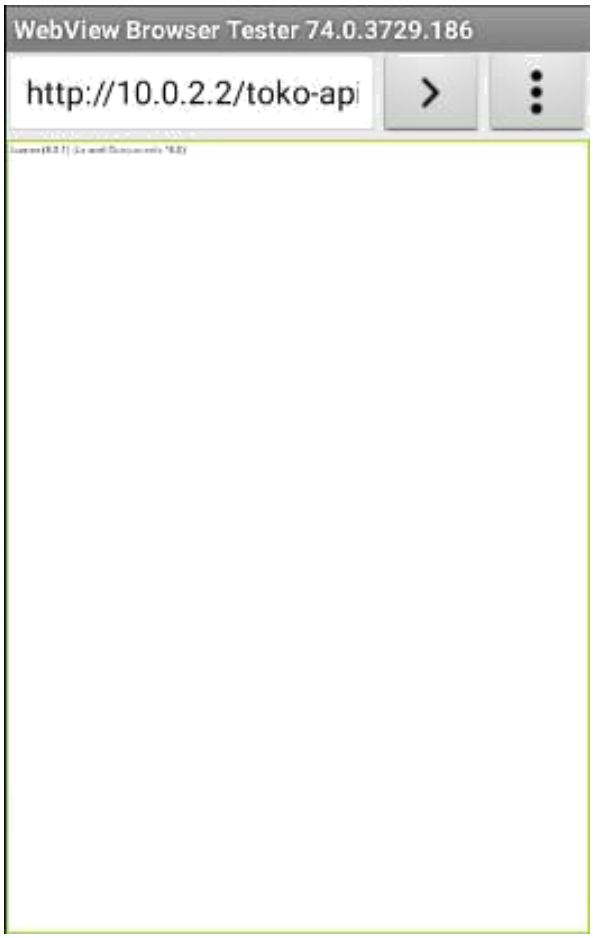
Membuat List API url

Buat sebuah file di dalam folder helpers dengan nama api_url.dart, dimana fungsi dari file ini adalah menyimpan alamat-alamat API yang telah dibuat sebelumnya (Endpoint dari API Spec)

```
class ApiUrl {
  static const String baseUrl = 'http://10.99.4.182/tokokita/public';

  static const String registrasi = baseUrl + '/registrasi';
  static const String login = baseUrl + '/login';
  static const String listProduk = baseUrl + '/produk';
  static const String createProduk = baseUrl + '/produk';

  static String updateProduk(int id) {
    return baseUrl + '/produk/' + id.toString();
  }

  static String showProduk(int id) {
    return baseUrl + '/produk/' + id.toString();
```

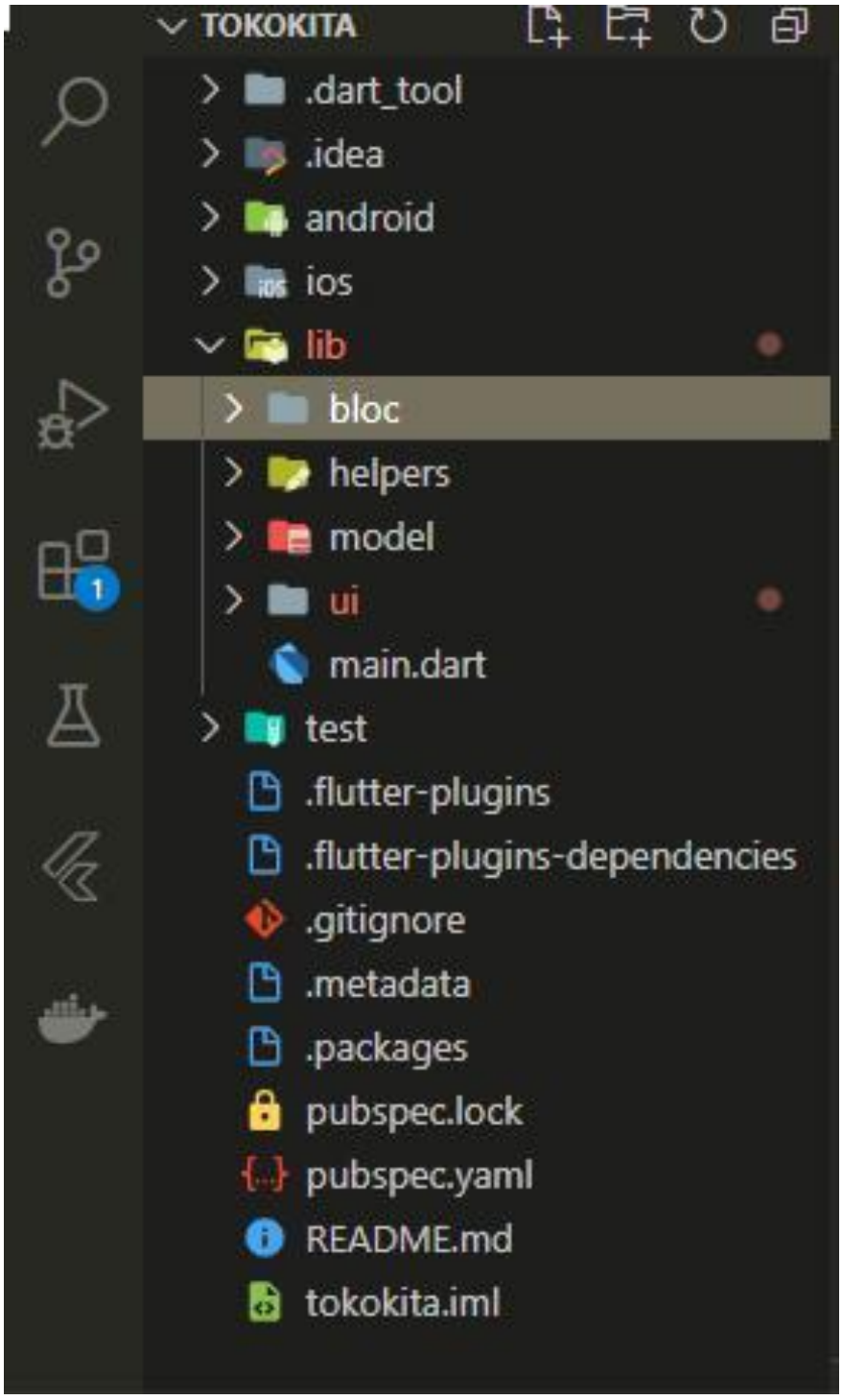
```
  }

  static String deleteProduk(int id) {
    return baseUrl + '/produk/' + id.toString();
  }
}
```

Pada baris kedua (baseUrl) merupakan alamat IP dari Rest API, untuk memeriksa apakah terhubung dengan emulator atau tidak, dapat dilakukan dengan memasukkan alamat tersebut pada browser yang ada pada emulator atau handphone android yang terkoneksi dengan laptop



Membuat Bloc

Buat sebuah folder bernama bloc. Di dalam folder ini berisis file-file yang berfungsi sebagai controller baik itu untuk melakukan proses login, registrasi dan lain-lain.



Registrasi

Buat sebuah file dengan nama registrasi_bloc.dart pada folder bloc. Kemudian masukkan kode berikut

```
import 'dart:convert';
import 'package:tokokita/helpers/api.dart';
import 'package:tokokita/helpers/api_url.dart';
import 'package:tokokita/model/registrasi.dart';

class RegistrasiBloc {
  static Future<Registrasi> registrasi(
```

```
      {String? nama, String? email, String? password}) async {
    String apiUrl = ApiUrl.registrasi;

    var body = {"nama": nama, "email": email, "password": password};

    var response = await Api().post(apiUrl, body);
    var jsonObj = json.decode(response.body);
    return Registrasi.fromJson(jsonObj);
  }
}
```

Login

Buat sebuah file dengan nama login_bloc.dart pada folder bloc. Kemudian masukkan kode berikut

```
import 'dart:convert';
import 'package:tokokita/helpers/api.dart';
import 'package:tokokita/helpers/api_url.dart';
import 'package:tokokita/model/login.dart';

class LoginBloc {
  static Future<Login> login({String? email, String? password}) async {
    String apiUrl = ApiUrl.login;
    var body = {"email": email, "password": password};
    var response = await Api().post(apiUrl, body);
    var jsonObj = json.decode(response.body);
    return Login.fromJson(jsonObj);
  }
}
```

Logout

Buat sebuah file dengan nama logout_bloc.dart pada folder bloc. Kemudian masukkan kode berikut

```
import 'package:tokokita/helpers/user_info.dart';

class LogoutBloc {
  static Future logout() async {
    await UserInfo().logout();
  }
}
```

Produk

Pada bagian ini akan dibuat beberapa fungsi untuk mengambil, mengubah dan menghapus data produk dari Rest API. Buat sebuah file dengan nama produk_bloc.dart pada folder bloc kemudian masukkan kode berikut

```
import 'dart:convert';
import 'package:tokokita/helpers/api.dart';
import 'package:tokokita/helpers/api_url.dart';
import 'package:tokokita/model/produk.dart';
```

```dart
class ProdukBloc {
  static Future<List<Produk>> getProduks() 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;
  }

  static Future addProduk({Produk? produk}) async {
    String apiUrl = ApiUrl.createProduk;

    var body = {
      "kode_produk": produk!.kodeProduk,
      "nama_produk": produk.namaProduk,
      "harga": produk.hargaProduk.toString()
    };

    var response = await Api().post(apiUrl, body);
    var jsonObj = json.decode(response.body);
    return jsonObj['status'];
  }

  static Future updateProduk({required Produk produk}) async {
    String apiUrl = ApiUrl.updateProduk(int.parse(produk.id!));
    print(apiUrl);

    var body = {
      "kode_produk": produk.kodeProduk,
      "nama_produk": produk.namaProduk,
      "harga": produk.hargaProduk.toString()
    };
    print("Body : $body");
    var response = await Api().put(apiUrl, jsonEncode(body));
    var jsonObj = json.decode(response.body);
    return jsonObj['status'];
  }
```

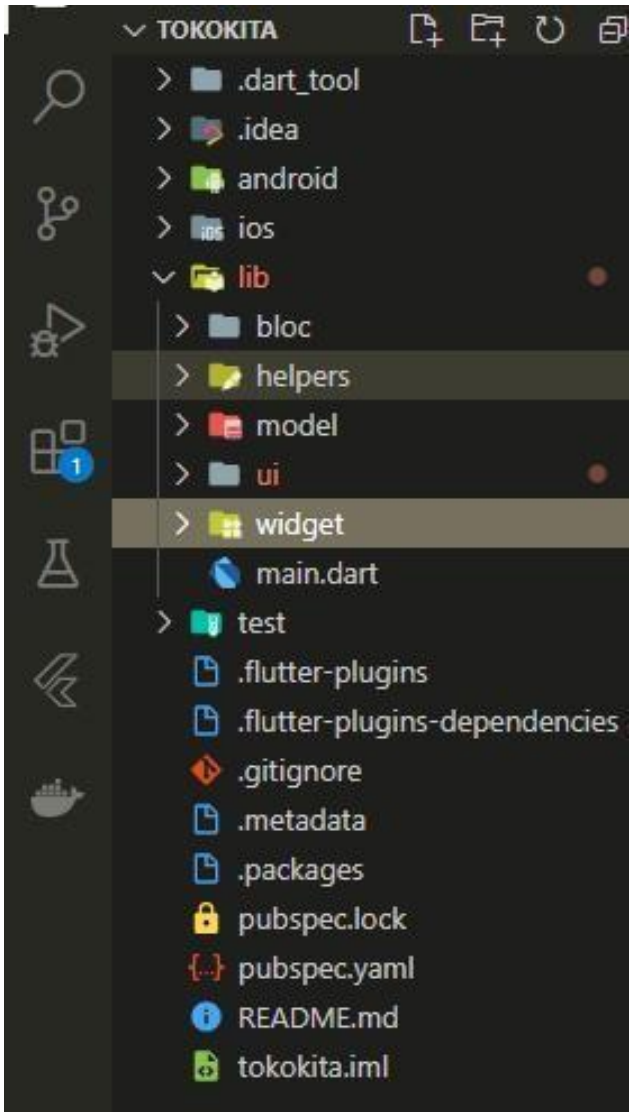
```
  static Future<bool> deleteProduk({int? id}) async {
    String apiUrl = ApiUrl.deleteProduk(id!);

    var response = await Api().delete(apiUrl);
    var jsonObj = json.decode(response.body);
    return (jsonObj as Map<String, dynamic>)['data'];
  }
}
```

Menyatukan Fungsionalitas

Membuat Common Dialog Widget

Pada bagian ini akan dibuat dua buah dialog yang nantinya akan digunakan pada tampilan.

Buat sebuah folder dengan nama widget



Kemudian buat sebuah file dengan nama success_dialog.dart dengan kode

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

class Consts {
  Consts._();
  static const double padding = 16.0;
  static const double avatarRadius = 66.0;
}

class SuccessDialog extends StatelessWidget {
  final String? description;
  final VoidCallback? okClick;

  const SuccessDialog({Key? key, this.description, this.okClick})
```

```
    : super(key: key);

@override
Widget build(BuildContext context) {
  return Dialog(
    shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(Consts.padding)),
    elevation: 0.0,
    backgroundColor: Colors.transparent,
    child: dialogContent(context),
  );
}

dialogContent(BuildContext context) {
  return Container(
    padding: const EdgeInsets.only(
      top: Consts.padding,
      bottom: Consts.padding,
      left: Consts.padding,
      right: Consts.padding,
    ),
    margin: const EdgeInsets.only(top: Consts.avatarRadius),
    decoration: BoxDecoration(
      color: Colors.white,
      shape: BoxShape.rectangle,
      borderRadius: BorderRadius.circular(Consts.padding),
      boxShadow: const [
        BoxShadow(
          color: Colors.black26,
          blurRadius: 10.0,
          offset: Offset(0.0, 10.0),
        ),
      ],
    ),
    child: Column(
      mainAxisSize: MainAxisSize.min,
      children: [
        const Text(
          "SUKSES",
          style: TextStyle(
              fontSize: 24.0,
              fontWeight: FontWeight.w700,
```

```
                color: Colors.green),
          ),
          const SizedBox(height: 16.0),
          Text(
            description!,
            textAlign: TextAlign.center,
            style: const TextStyle(
              fontSize: 16.0,
            ),
          ),
          const SizedBox(height: 24.0),
          Align(
            alignment: Alignment.bottomRight,
            child: OutlinedButton(
              onPressed: () {
                Navigator.of(context).pop(); // To close the dialog
                okClick!();
              },
              child: const Text("OK"),
            ),
          )
        ],
      ),
    );
  }
}
```

Kemudian buat file dengan nama warning_dialog.dart dengan kode

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

class Consts {
  Consts._();
  static const double padding = 16.0;
  static const double avatarRadius = 66.0;
}

class WarningDialog extends StatelessWidget {
  final String? description;
  final VoidCallback? okClick;
```

```
const WarningDialog({Key? key, this.description, this.okClick})
    : super(key: key);

@override
Widget build(BuildContext context) {
  return Dialog(
    shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(Consts.padding)),
    elevation: 0.0,
    backgroundColor: Colors.transparent,
    child: dialogContent(context),
  );
}

dialogContent(BuildContext context) {
  return Container(
    padding: const EdgeInsets.only(
      top: Consts.padding,
      bottom: Consts.padding,
      left: Consts.padding,
      right: Consts.padding,
    ),
    margin: const EdgeInsets.only(top: Consts.avatarRadius),
    decoration: BoxDecoration(
      color: Colors.white,
      shape: BoxShape.rectangle,
      borderRadius: BorderRadius.circular(Consts.padding),
      boxShadow: const [
        BoxShadow(
          color: Colors.black26,
          blurRadius: 10.0,
          offset: Offset(0.0, 10.0),
        ),
      ],
    ),
    child: Column(
      mainAxisSize: MainAxisSize.min,
      children: [
        const Text(
          "GAGAL",
          style: TextStyle(
              fontSize: 24.0, fontWeight: FontWeight.w700, color: Colors.red),
```

```
          ),
          const SizedBox(height: 16.0),
          Text(
            description!,
            textAlign: TextAlign.center,
            style: const TextStyle(
              fontSize: 16.0,
            ),
          ),
          const SizedBox(height: 24.0),
          Align(
            alignment: Alignment.bottomRight,
            child: ElevatedButton(
              onPressed: () {
                Navigator.of(context).pop(); // To close the dialog
              },
              child: const Text("OK"),
            ),
          )
        ],
      ),
    );
  }
}
```

Modifikasi main.dart

Buka kembali file main.dart kita akan memodifikasi file tersebut dengan kondisi jika belum login maka akan membuka halaman login, namun jika sudah login maka akan membuka halaman list produk

```dart
import 'package:flutter/material.dart';
import 'package:tokokita/helpers/user_info.dart';
import 'package:tokokita/ui/login_page.dart';
import 'package:tokokita/ui/produk_page.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);

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

class _MyAppState extends State<MyApp> {
  Widget page = const CircularProgressIndicator();

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

  void isLogin() async {
    var token = await UserInfo().getToken();
    if (token != null) {
      setState(() {
        page = const ProdukPage();
      });
    } else {
      setState(() {
        page = const LoginPage();
      });
    }
  }
```

```
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Toko Kita',
      debugShowCheckedModeBanner: false,
      home: page,
    );
  }
}
```

Modifikasi registrasi_page.dart
Buka file registrasi_page.dart pada folder ui kemudian modifikasi fungsi _buttonRegistrasi dan tambahkan fungsi dengan nama _submit seperti dibawah

```
//Membuat Tombol Registrasi
  Widget _buttonRegistrasi() {
    return ElevatedButton(
        child: const Text("Registrasi"),
        onPressed: () {
          var validate = _formKey.currentState!.validate();
          if (validate) {
            if (!_isLoading) _submit();
          }
        });
  }

  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",
              okClick: () {
                Navigator.pop(context);
              },
            ));
  }, onError: (error) {
    showDialog(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) => const WarningDialog(
              description: "Registrasi gagal, silahkan coba lagi",
            ));
  });
```

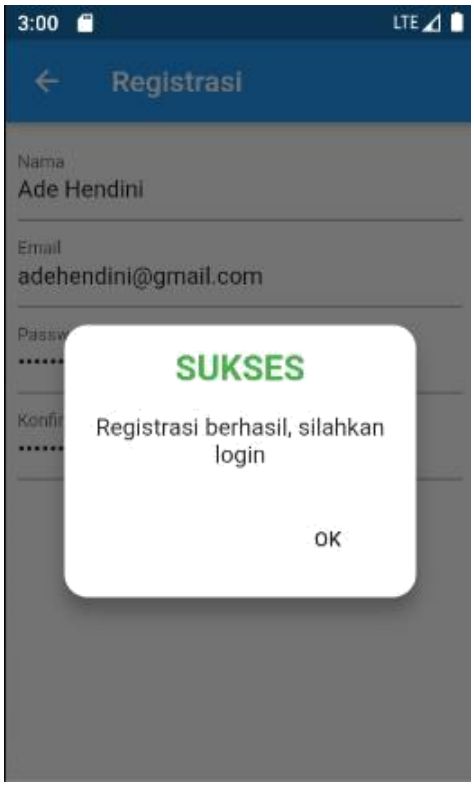
Sehingga keseluruhan kode pada registrasi_page.dart menjadi seperti berikut

```
import 'package:flutter/material.dart';
import 'package:tokokita/bloc/registrasi_bloc.dart';
import 'package:tokokita/widget/success_dialog.dart';
import 'package:tokokita/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"),
      ),
      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()
              ],
```

```
            ),
          ),
        ),
      ),
    );
  }

//Membuat Textbox Nama
  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;
      },
    );
  }

  //Membuat Textbox email
  Widget _emailTextField() {
    return TextFormField(
      decoration: const InputDecoration(labelText: "Email"),
      keyboardType: TextInputType.emailAddress,
      controller: _emailTextboxController,
      validator: (value) {
        //validasi harus diisi
        if (value!.isEmpty) {
          return 'Email harus diisi';
        }
        //validasi email
        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"),
      keyboardType: TextInputType.text,
      obscureText: true,
      controller: _passwordTextboxController,
      validator: (value) {
        //jika karakter yang dimasukkan kurang dari 6 karakter
        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"),
      keyboardType: TextInputType.text,
      obscureText: true,
      validator: (value) {
        //jika inputan tidak sama dengan password
        if (value != _passwordTextboxController.text) {
          return "Konfirmasi Password tidak sama";
        }
        return null;
      },
    );
  }

//Membuat Tombol Registrasi
  Widget _buttonRegistrasi() {
    return ElevatedButton(
        child: const Text("Registrasi"),
        onPressed: () {
```

```
          var validate = _formKey.currentState!.validate();
          if (validate) {
            if (!_isLoading) _submit();
          }
        });
  }

  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",
                okClick: () {
                  Navigator.pop(context);
                },
              ));
    }, onError: (error) {
      showDialog(
          context: context,
          barrierDismissible: false,
          builder: (BuildContext context) => const WarningDialog(
                description: "Registrasi gagal, silahkan coba lagi",
              ));
    });
    setState(() {
      _isLoading = false;
    });
  }
}
```



Modifikasi login_page.dart (fungsi login)
Buka file login_page.dart pada folder ui kemudian modifikasi fungsi _buttonLogin dan tambahkan fungsi dengan nama _submit seperti dibawah

```
//Membuat Tombol Login
Widget _buttonLogin() {
  return ElevatedButton(
      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 ProdukPage()));
      } 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",
              ));
    });
```

Adapun kode secara keseluruhan menjadi seperti berikut

```
import 'package:flutter/material.dart';
import 'package:tokokita/bloc/login_bloc.dart';
import 'package:tokokita/helpers/user_info.dart';
import 'package:tokokita/ui/produk_page.dart';
import 'package:tokokita/ui/registrasi_page.dart';
import 'package:tokokita/widget/warning_dialog.dart';

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

class _LoginPageState extends State<LoginPage> {
```

```dart
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'),
    ),
    body: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Form(
          key: _formKey,
          child: Column(
            children: [
              _emailTextField(),
              _passwordTextField(),
              _buttonLogin(),
              const SizedBox(
                height: 30,
              ),
              _menuRegistrasi()
            ],
          ),
        ),
      ),
    ),
  );
}

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

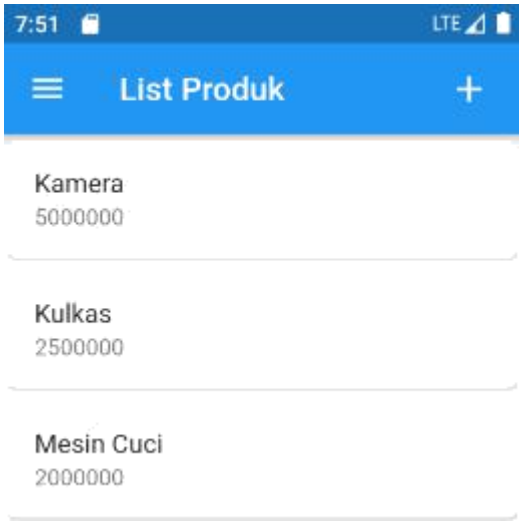
```
        }
        return null;
      },
    );
  }

//Membuat Textbox password
  Widget _passwordTextField() {
    return TextFormField(
      decoration: const InputDecoration(labelText: "Password"),
      keyboardType: TextInputType.text,
      obscureText: true,
      controller: _passwordTextboxController,
      validator: (value) {
        //jika karakter yang dimasukkan kurang dari 6 karakter
        if (value!.isEmpty) {
          return "Password harus diisi";
        }
        return null;
      },
    );
  }

  //Membuat Tombol Login
  //Membuat Tombol Login
  Widget _buttonLogin() {
    return ElevatedButton(
        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 ProdukPage()));
    } 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.blue),
      ),
      onTap: () {
        Navigator.push(context,
            MaterialPageRoute(builder: (context) => const RegistrasiPage()));
      },
```

```
      ),
    );
  }
}
```

Jika Gagal akan muncul pesan seperti berikut

Jika berhasil akan menuju ke halaman produk_page.dart



Modifikasi produk_page.dart

Menambahkan fungsi logout pada drawer

Agar link logout dapat berfungsi, akan ditambahkan kode pada drawer logout, seperti berikut

```
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>(
```

Secara keseluruhan kode pada produk_page.dart menjadi seperti berikut

```
import 'package:flutter/material.dart';
import 'package:tokokita/bloc/logout_bloc.dart';
import 'package:tokokita/bloc/produk_bloc.dart';
import 'package:tokokita/model/produk.dart';
import 'package:tokokita/ui/login_page.dart';
import 'package:tokokita/ui/produk_detail.dart';
import 'package:tokokita/ui/produk_form.dart';

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 Produk'),
        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()));
                },
              ))
        ],
      ),
      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: ProdukBloc.getProduks(),
        builder: (context, snapshot) {
          if (snapshot.hasError) print(snapshot.error);
          return snapshot.hasData
              ? ListProduk(
                  list: snapshot.data,
                )
              : const Center(
                  child: CircularProgressIndicator(),
                );
        },
      ),
    );
  }
}
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.namaProduk!),
          subtitle: Text(produk.hargaProduk.toString()),
        ),
      ),
    );
  }
}
```

Menampilkan Data Produk dari Rest API

Pada bagian ini akan dimodifikasi file produk_page.dart sehingga dapat menampilkan data dari Rest API. Berikut kode keseluruhan

```
import 'package:flutter/material.dart';
import 'package:tokokita/bloc/logout_bloc.dart';
import 'package:tokokita/bloc/produk_bloc.dart';
import 'package:tokokita/model/produk.dart';
import 'package:tokokita/ui/login_page.dart';
import 'package:tokokita/ui/produk_detail.dart';
import 'package:tokokita/ui/produk_form.dart';

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 Produk'),
        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()));
                },
              ))
        ],
      ),
      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: ProdukBloc.getProduks(),
        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);

  @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.namaProduk!),
          subtitle: Text(produk.hargaProduk.toString()),
        ),
      ),
    );
  }
}
```

Adapun perubahan yang dilakukan adalah penambahan sebuah class bernama ListProduk dengan kode

```
class ListProduk extends StatelessWidget {
  final List? list;

  const ListProduk({Key? key, this.list}) : super(key: key);

  @override
```

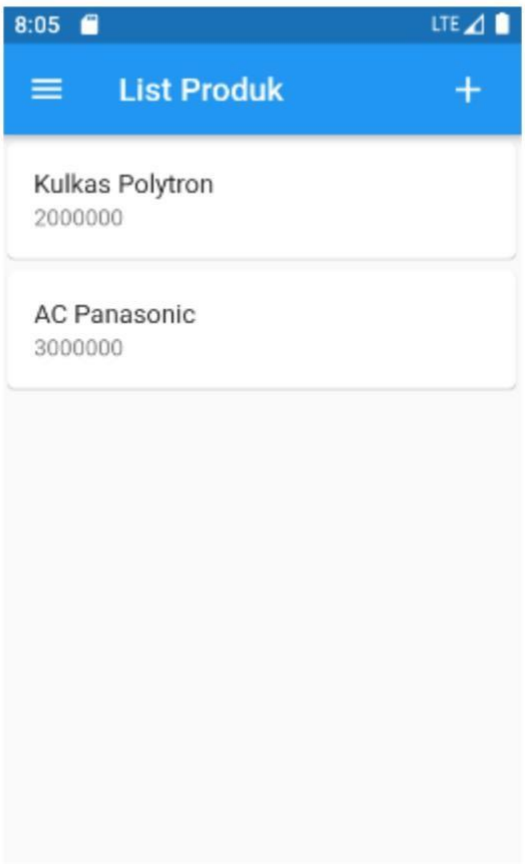
```
  Widget build(BuildContext context) {
    return ListView.builder(
        itemCount: list == null ? 0 : list!.length,
        itemBuilder: (context, i) {
          return ItemProduk(
            produk: list![i],
          );
        });
  }
}
```

Kemudian perubahan pada bagian body menjadi

```
body: FutureBuilder<List>(
        future: ProdukBloc.getProduks(),
        builder: (context, snapshot) {
          if (snapshot.hasError) print(snapshot.error);
          return snapshot.hasData
              ? ListProduk(
                  list: snapshot.data,
                )
              : const Center(
                  child: CircularProgressIndicator(),
                );
        },
      ),
```

Serta memasukkan beberapa package yang diperlukan

```
1. import 'package:flutter/material.dart';
2. import 'package:tokokita/bloc/logout_bloc.dart';
3. import 'package:tokokita/bloc/produk_bloc.dart';
4. import 'package:tokokita/model/produk.dart';
5. import 'package:tokokita/ui/login_page.dart';
6. import 'package:tokokita/ui/produk_detail.dart';
7. import 'package:tokokita/ui/produk_form.dart';
```



Memodifikasi Form Produk (produk_form.dart)

Membuat fungsi simpan

Agar tombol simpan dapat berfungsi diperlukan kode fungsi untuk menyimpan data dengan memanggil bloc produk_bloc yang telah dibuat sebelumnya, kita akan menambahkan sebuah fungsi dengan nama simpan dan memodifikasi fungsi _buttonSubmit

```
//Membuat Tombol Simpan/Ubah
 Widget _buttonSubmit() {
   return OutlinedButton(
       child: Text(tombolSubmit),
       onPressed: () {
         var validate = _formKey.currentState!.validate();
         if (validate) {
           if (!_isLoading) {
             if (widget.produk != null) {
               //kondisi update produk
```

```
              } else {
                //kondisi tambah produk
                simpan();
              }
            }
          }
        });
  }

  simpan() {
    setState(() {
      _isLoading = true;
    });
    Produk createProduk = Produk(id: null);
    createProduk.kodeProduk = _kodeProdukTextboxController.text;
    createProduk.namaProduk = _namaProdukTextboxController.text;
    createProduk.hargaProduk = int.parse(_hargaProdukTextboxController.text);
    ProdukBloc.addProduk(produk: createProduk).then((value) {
      Navigator.of(context).push(MaterialPageRoute(
          builder: (BuildContext context) => const ProdukPage()));
    }, onError: (error) {
      showDialog(
          context: context,
          builder: (BuildContext context) => const WarningDialog(
                description: "Simpan gagal, silahkan coba lagi",
              ));
    });
    setState(() {
      _isLoading = false;
    });
  }
```

Dengan keseluruhan kode menjadi

```
import 'package:flutter/material.dart';
import 'package:tokokita/bloc/produk_bloc.dart';
import 'package:tokokita/model/produk.dart';
import 'package:tokokita/ui/produk_page.dart';
import 'package:tokokita/widget/warning_dialog.dart';

// ignore: must_be_immutable
class ProdukForm extends StatefulWidget {
```

```
  Produk? produk;
  ProdukForm({Key? key, this.produk}) : super(key: key);
  @override
  _ProdukFormState createState() => _ProdukFormState();
}

class _ProdukFormState extends State<ProdukForm> {
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  String judul = "TAMBAH PRODUK";
  String tombolSubmit = "SIMPAN";
  final _kodeProdukTextboxController = TextEditingController();
  final _namaProdukTextboxController = TextEditingController();
  final _hargaProdukTextboxController = TextEditingController();
  @override
  void initState() {
    super.initState();
    isUpdate();
  }

  isUpdate() {
    if (widget.produk != null) {
      setState(() {
        judul = "UBAH PRODUK";
        tombolSubmit = "UBAH";
        _kodeProdukTextboxController.text = widget.produk!.kodeProduk!;
        _namaProdukTextboxController.text = widget.produk!.namaProduk!;
        _hargaProdukTextboxController.text =
            widget.produk!.hargaProduk.toString();
      });
    } else {
      judul = "TAMBAH PRODUK";
      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: [
              _kodeProdukTextField(),
              _namaProdukTextField(),
              _hargaProdukTextField(),
              _buttonSubmit()
            ],
          ),
        ),
      ),
    ),
  );
}

//Membuat Textbox Kode Produk
Widget _kodeProdukTextField() {
  return TextFormField(
    decoration: const InputDecoration(labelText: "Kode Produk"),
    keyboardType: TextInputType.text,
    controller: _kodeProdukTextboxController,
    validator: (value) {
      if (value!.isEmpty) {
        return "Kode Produk harus diisi";
      }
      return null;
    },
  );
}

//Membuat Textbox Nama Produk
Widget _namaProdukTextField() {
  return TextFormField(
    decoration: const InputDecoration(labelText: "Nama Produk"),
    keyboardType: TextInputType.text,
    controller: _namaProdukTextboxController,
    validator: (value) {
      if (value!.isEmpty) {
        return "Nama Produk harus diisi";
      }
```

```
        return null;
      },
    );
  }

//Membuat Textbox Harga Produk
  Widget _hargaProdukTextField() {
    return TextFormField(
      decoration: const InputDecoration(labelText: "Harga"),
      keyboardType: TextInputType.number,
      controller: _hargaProdukTextboxController,
      validator: (value) {
        if (value!.isEmpty) {
          return "Harga 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.produk != null) {
                //kondisi update produk
              } else {
                //kondisi tambah produk
                simpan();
              }
            }
          }
        });
  }

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

```
    });
    Produk createProduk = Produk(id: null);
    createProduk.kodeProduk = _kodeProdukTextboxController.text;
    createProduk.namaProduk = _namaProdukTextboxController.text;
    createProduk.hargaProduk = int.parse(_hargaProdukTextboxController.text);
    ProdukBloc.addProduk(produk: createProduk).then((value) {
      Navigator.of(context).push(MaterialPageRoute(
          builder: (BuildContext context) => const ProdukPage()));
    }, onError: (error) {
      showDialog(
          context: context,
          builder: (BuildContext context) => const WarningDialog(
                description: "Simpan gagal, silahkan coba lagi",
              ));
    });
    setState(() {
      _isLoading = false;
    });
  }
```

Membuat fungsi ubah

Sama halnya dengan simpan, kita buat sebuah fungsi ubah kemudian kita sertakan pada fungsi _buttonSubmit

Dengan fungsi ubah sebagai berikut

```
ubah() {
    setState(() {
      _isLoading = true;
    });
    Produk updateProduk = Produk(id: widget.produk!.id!);
    updateProduk.kodeProduk = _kodeProdukTextboxController.text;
    updateProduk.namaProduk = _namaProdukTextboxController.text;
    updateProduk.hargaProduk = int.parse(_hargaProdukTextboxController.text);
    ProdukBloc.updateProduk(produk: updateProduk).then((value) {
      Navigator.of(context).push(MaterialPageRoute(
          builder: (BuildContext context) => const ProdukPage()));
    }, onError: (error) {
      showDialog(
          context: context,
          builder: (BuildContext context) => const WarningDialog(
                description: "Permintaan ubah data gagal, silahkan coba lagi",
              ));
```

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

Kemudian kita tambahkan pada fungsi _buttonSubmit

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

Dengan kode keseluruhan menjadi seperti berikut

```
import 'package:flutter/material.dart';
import 'package:tokokita/bloc/produk_bloc.dart';
import 'package:tokokita/model/produk.dart';
import 'package:tokokita/ui/produk_page.dart';
import 'package:tokokita/widget/warning_dialog.dart';

// ignore: must_be_immutable
class ProdukForm extends StatefulWidget {
  Produk? produk;
  ProdukForm({Key? key, this.produk}) : super(key: key);
  @override
  _ProdukFormState createState() => _ProdukFormState();
}

class _ProdukFormState extends State<ProdukForm> {
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  String judul = "TAMBAH PRODUK";
  String tombolSubmit = "SIMPAN";
  final _kodeProdukTextboxController = TextEditingController();
  final _namaProdukTextboxController = TextEditingController();
  final _hargaProdukTextboxController = TextEditingController();
  @override
  void initState() {
    super.initState();
    isUpdate();
  }

  isUpdate() {
    if (widget.produk != null) {
      setState(() {
        judul = "UBAH PRODUK";
        tombolSubmit = "UBAH";
        _kodeProdukTextboxController.text = widget.produk!.kodeProduk!;
        _namaProdukTextboxController.text = widget.produk!.namaProduk!;
        _hargaProdukTextboxController.text =
            widget.produk!.hargaProduk.toString();
      });
    } else {
      judul = "TAMBAH PRODUK";
```

```
    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: [
              _kodeProdukTextField(),
              _namaProdukTextField(),
              _hargaProdukTextField(),
              _buttonSubmit()
            ],
          ),
        ),
      ),
    ),
  );
}

//Membuat Textbox Kode Produk
Widget _kodeProdukTextField() {
  return TextFormField(
    decoration: const InputDecoration(labelText: "Kode Produk"),
    keyboardType: TextInputType.text,
    controller: _kodeProdukTextboxController,
    validator: (value) {
      if (value!.isEmpty) {
        return "Kode Produk harus diisi";
      }
      return null;
    },
  );
}
```

```
  //Membuat Textbox Nama Produk
  Widget _namaProdukTextField() {
    return TextFormField(
      decoration: const InputDecoration(labelText: "Nama Produk"),
      keyboardType: TextInputType.text,
      controller: _namaProdukTextboxController,
      validator: (value) {
        if (value!.isEmpty) {
          return "Nama Produk harus diisi";
        }
        return null;
      },
    );
  }

//Membuat Textbox Harga Produk
  Widget _hargaProdukTextField() {
    return TextFormField(
      decoration: const InputDecoration(labelText: "Harga"),
      keyboardType: TextInputType.number,
      controller: _hargaProdukTextboxController,
      validator: (value) {
        if (value!.isEmpty) {
          return "Harga 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.produk != null) {
                //kondisi update produk
                ubah();
              } else {
```

```
                //kondisi tambah produk
                simpan();
              }
            }
          }
        });
}

simpan() {
  setState(() {
    _isLoading = true;
  });
  Produk createProduk = Produk(id: null);
  createProduk.kodeProduk = _kodeProdukTextboxController.text;
  createProduk.namaProduk = _namaProdukTextboxController.text;
  createProduk.hargaProduk = int.parse(_hargaProdukTextboxController.text);
  ProdukBloc.addProduk(produk: createProduk).then((value) {
    Navigator.of(context).push(MaterialPageRoute(
        builder: (BuildContext context) => const ProdukPage()));
  }, onError: (error) {
    showDialog(
        context: context,
        builder: (BuildContext context) => const WarningDialog(
              description: "Simpan gagal, silahkan coba lagi",
            ));
  });
  setState(() {
    _isLoading = false;
  });
}

ubah() {
  setState(() {
    _isLoading = true;
  });
  Produk updateProduk = Produk(id: widget.produk!.id!);
  updateProduk.kodeProduk = _kodeProdukTextboxController.text;
  updateProduk.namaProduk = _namaProdukTextboxController.text;
  updateProduk.hargaProduk = int.parse(_hargaProdukTextboxController.text);
  ProdukBloc.updateProduk(produk: updateProduk).then((value) {
    Navigator.of(context).push(MaterialPageRoute(
        builder: (BuildContext context) => const ProdukPage()));
```

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

Menambahkan fungsi hapus pada Detail Produk (produk_detail.dart)

Buka file produk_detail.dart pada folder ui, kemudian kita modifikasi pada fungsi confirmHapus menjadi seperti berikut

```
  void confirmHapus() {
    AlertDialog alertDialog = AlertDialog(
      content: const Text("Yakin ingin menghapus data ini?"),
      actions: [
        //tombol hapus
        OutlinedButton(
          child: const Text("Ya"),
          onPressed: () {
            ProdukBloc.deleteProduk(id: int.parse(widget.produk!.id!)).then(
                (value) => {
                      Navigator.of(context).push(MaterialPageRoute(
                          builder: (context) => const ProdukPage()))
                    }, 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);
  }
```

Dengan kode keseluruhan menjadi

```
import 'package:flutter/material.dart';
import 'package:tokokita/bloc/produk_bloc.dart';
import 'package:tokokita/model/produk.dart';
import 'package:tokokita/ui/produk_form.dart';
import 'package:tokokita/ui/produk_page.dart';
import 'package:tokokita/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> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Detail Produk'),
      ),
      body: Center(
        child: Column(
          children: [
            Text(
              "Kode : ${widget.produk!.kodeProduk}",
              style: const TextStyle(fontSize: 20.0),
            ),
            Text(
```

```
            "Nama : ${widget.produk!.namaProduk}",
            style: const TextStyle(fontSize: 18.0),
          ),
          Text(
            "Harga : Rp. ${widget.produk!.hargaProduk.toString()}",
            style: const TextStyle(fontSize: 18.0),
          ),
          _tombolHapusEdit()
        ],
      ),
    ),
  );
}

Widget _tombolHapusEdit() {
  return Row(
    mainAxisSize: MainAxisSize.min,
    children: [
      // Tombol Edit
      OutlinedButton(
        child: const Text("EDIT"),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => ProdukForm(
                produk: widget.produk!,
              ),
            ),
          );
        },
      ),
      // 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: () {
            ProdukBloc.deleteProduk(id: int.parse(widget.produk!.id!)).then(
                (value) => {
                      Navigator.of(context).push(MaterialPageRoute(
                          builder: (context) => const ProdukPage()))
                    }, 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);
  }
}
```

Untuk menjalankan berikut instruksinya:

untuk run di chrome

- flutter run -d chrome --web-browser-flag "--disable-web-security"

untuk run di emulator/device fisik pake ip address local

- *10.99.4.182 ganti pake ip lokal kalian masing2

- di project ci4

  php spark serve --host 10.99.4.182

- di project flutter helpers/api_url.dart

  ganti base url jadi http://10.99.4.182:8080/

Daftar Pustaka

[1]. A. Sasongko, M.S. Maulana & W. Nugraha. “Web Programming; Membangun Aplikasi Mobile Kolaborasi Antara Flutter dan Codeigniter”. Graha Ilmu. 2019.

[2]. https://flutter.dev/multi-platform/mobile

[3]. https://developer.android.com/studio

[4]. A. Sasongko, M.S. Maulana, & Latifah. “Presensi Karyawan Berbasis Aplikasi Mobile Dengan Filter Jaringan Intranet Dan Imei”. Jurnal Sistemasi, vol. 9, no. 1, pp. 92–102, 2020.