

```

51.
52.     public function hapus($id)
53.     {
54.         $model = new MProduk();
55.         $produk = $model->delete($id);
56.
57.         return $this->responseHasil(200, true, $produk);
58.     }
59. }

```

### Menambahkan route produk

Agar ProdukController dapat diakses, selanjutnya tambah route untuk mengakses produk pada file `app/Config/Routes.php`

```

$routes->group('produk', function ($routes) {
    $routes->post('/', 'ProdukController::create');
    $routes->get('/', 'ProdukController::list');
    $routes->get('/:segment', 'ProdukController::detail/$1');
    $routes->put('/:segment', 'ProdukController::ubah/$1');
    $routes->delete('/:segment', 'ProdukController::hapus/$1');
});

```

Adapun keseluruhan kode pada `app/Config/Routes.php` adalah sebagai berikut

```

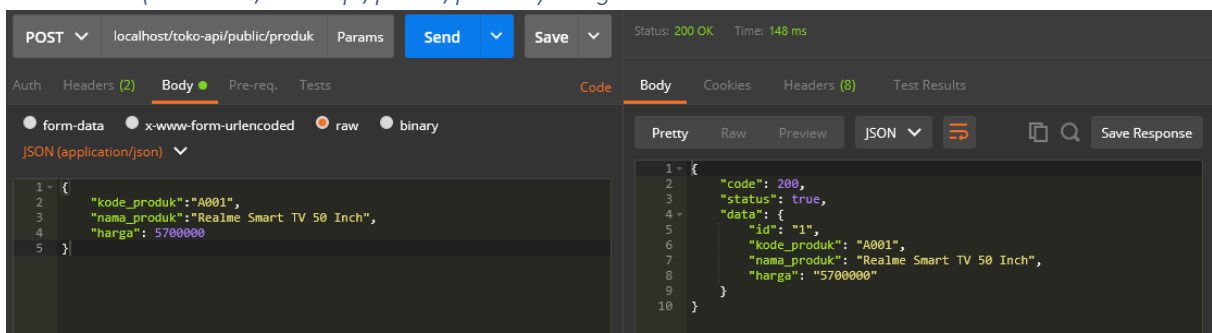
$routes->get('/', 'Home::index');
$routes->post('/registrasi', 'RegistrasiController::registrasi');
$routes->post('/login', 'LoginController::login');

$routes->group('produk', function ($routes) {
    $routes->post('/', 'ProdukController::create');
    $routes->get('/', 'ProdukController::list');
    $routes->get('/:segment', 'ProdukController::detail/$1');
    $routes->put('/:segment', 'ProdukController::ubah/$1');
    $routes->delete('/:segment', 'ProdukController::hapus/$1');
});

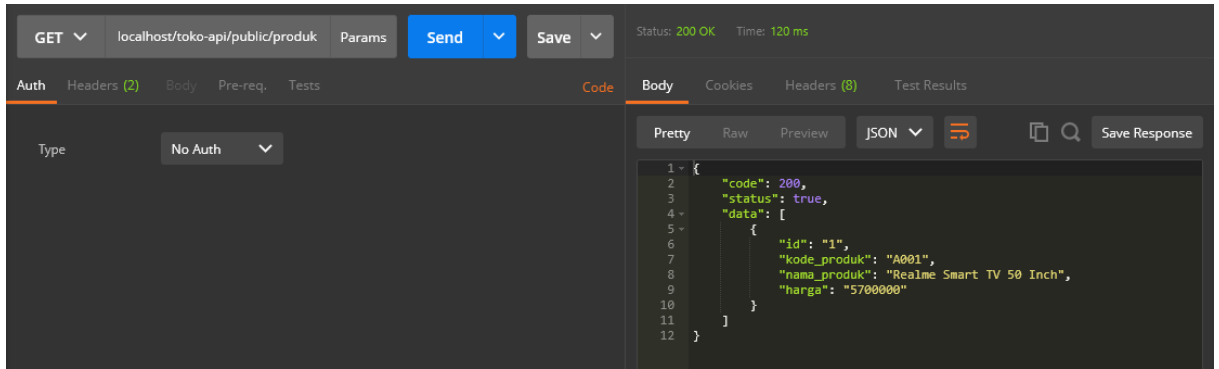
```

### Mencoba Rest

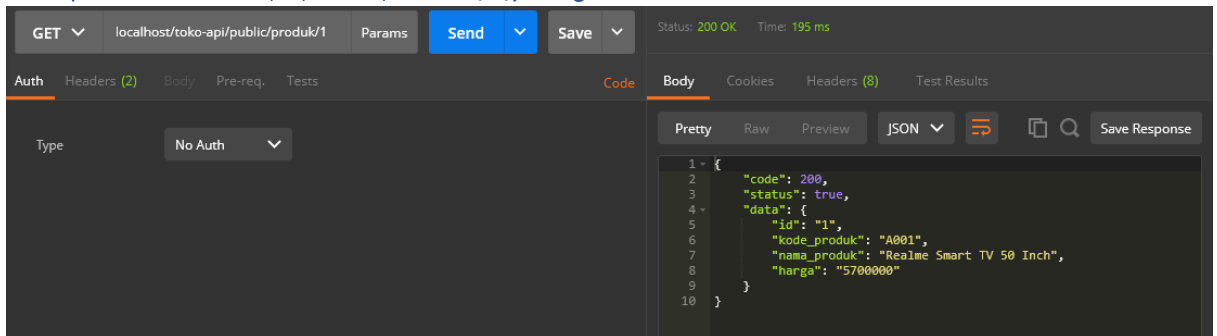
Create Produk (`localhost/toko-api/public/produk`) dengan method POST



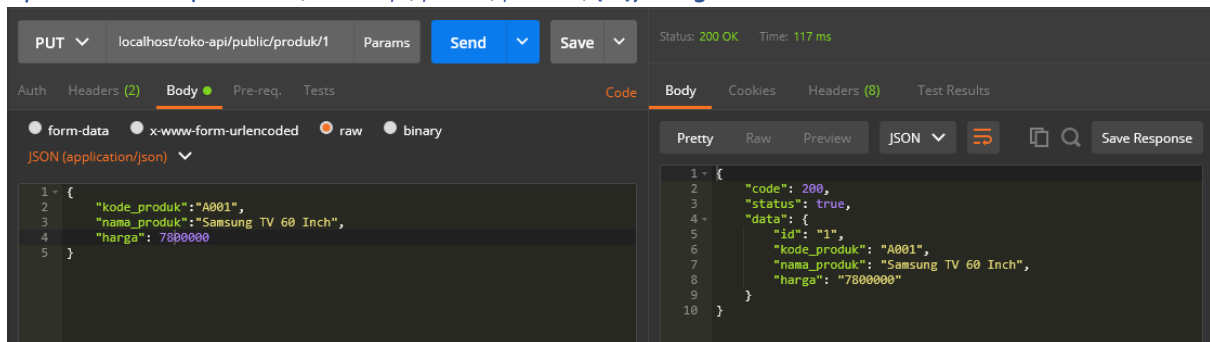
### List Produk (<localhost/toko-api/public/produk>) dengan method GET



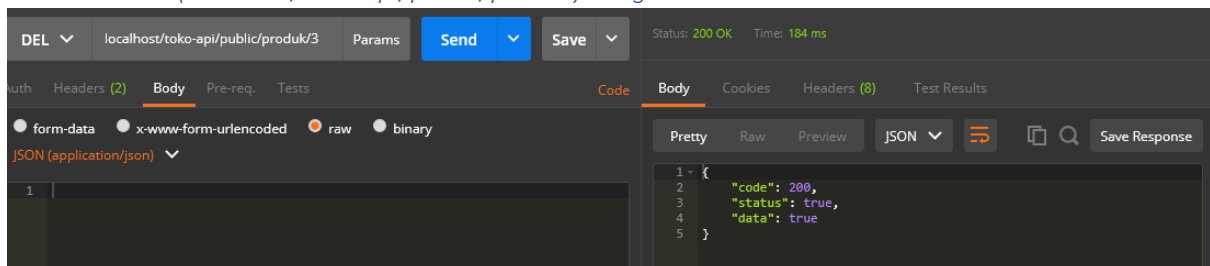
### Show Produk (<localhost/toko-api/public/produk/{id}>) dengan method GET



### Update Produk (<localhost/toko-api/public/produk/{id}>) dengan method PUT

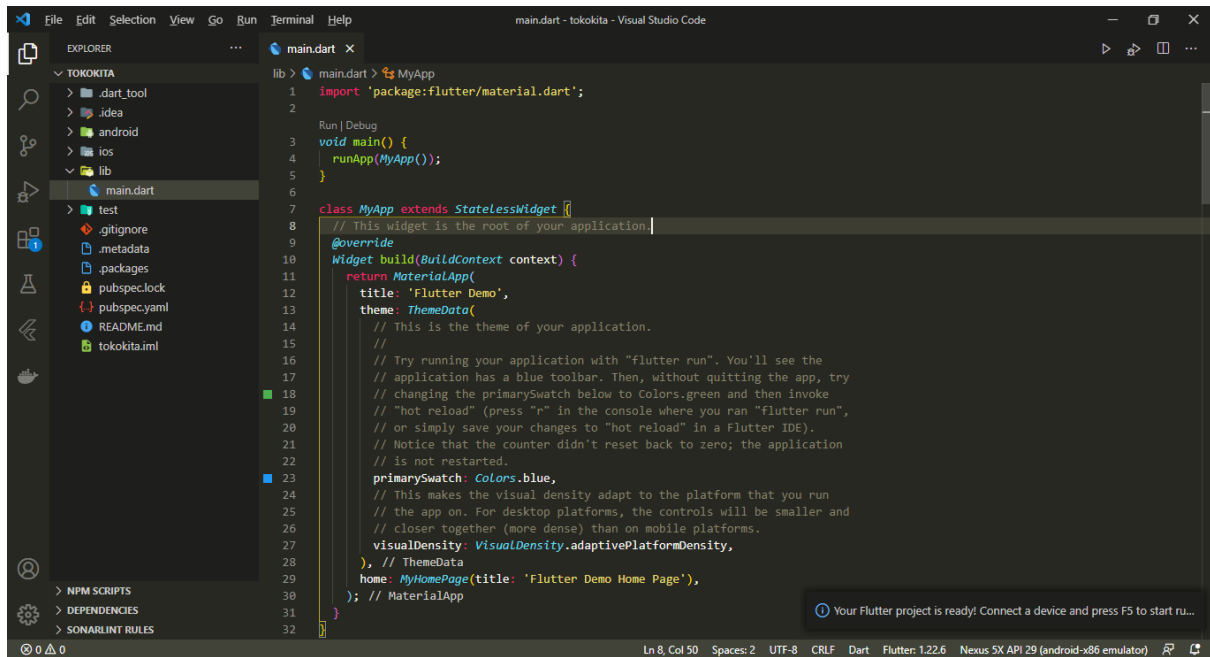


### Delete Produk (<localhost/toko-api/public/produk>) dengan method DELETE



## Membuat projek flutter tokokita

Buat sebuah projek flutter dengan nama **tokokita**



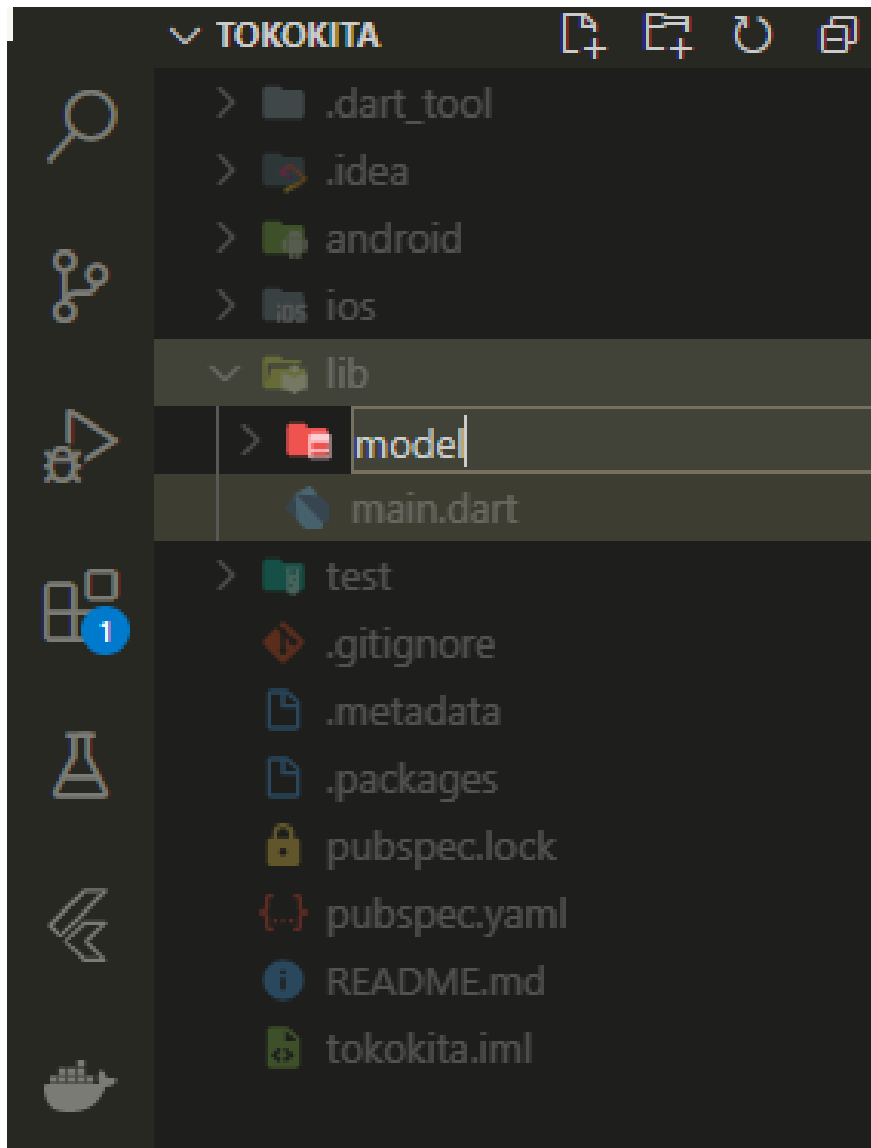
The screenshot shows the Visual Studio Code interface for a Flutter project named 'tokokita'. The Explorer view on the left shows the project structure with folders for .dart\_tool, .idea, android, ios, lib, and test. The main.dart file is open in the editor, showing the following code:

```
lib > main.dart > MyApp
1  import 'package:flutter/material.dart';
2
3  Run | Debug
4  void main() {
5    runApp(MyApp());
6  }
7
8  class MyApp extends StatelessWidget {
9    // This widget is the root of your application.
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       title: 'Flutter Demo',
14       theme: ThemeData(
15         // This is the theme of your application.
16         //
17         // Try running your application with "flutter run". You'll see the
18         // application has a blue toolbar. Then, without quitting the app, try
19         // changing the primarySwatch below to Colors.green and then invoke
20         // "hot reload" (press "r" in the console where you ran "flutter run",
21         // or simply save your changes to "hot reload" in a Flutter IDE).
22         // Notice that the counter didn't reset back to zero; the application
23         // is not restarted.
24         primarySwatch: Colors.blue,
25         // This makes the visual density adapt to the platform that you run
26         // the app on. For desktop platforms, the controls will be smaller and
27         // closer together (more dense) than on mobile platforms.
28         visualDensity: VisualDensity.adaptivePlatformDensity,
29       ), // ThemeData
30       home: MyHomePage(title: 'Flutter Demo Home Page'),
31     ); // MaterialApp
32   }
}
```

At the bottom of the editor, a status bar indicates: "Your Flutter project is ready! Connect a device and press F5 to start ru..."

## Membuat Model

Buat folder dengan nama **model** pada folder **lib**



## Login

Buat sebuah file dengan nama **login.dart** pada folder **model**. Kemudian masukkan kode berikut

```
1. class Login {
2.   int? code;
3.   bool? status;
4.   String? token;
5.   int? userID;
6.   String? userEmail;
7.
8.   Login({this.code, this.status, this.token, this.userID, this.userEmail});
9.
10.  factory Login.fromJson(Map<String, dynamic> obj) {
11.    return Login(
12.      code: obj['code'],
13.      status: obj['status'],
14.      token: obj['data']['token'],
15.      userID: obj['data']['user']['id'],
```

```
16.         userEmail: obj['data']['user']['email']);
17.     }
18. }
```

## Registrasi

Buat sebuah file dengan nama **registrasi.dart** pada folder **model**. Kemudian masukkan kode berikut

```
1. class Registrasi {
2.     int? code;
3.     bool? status;
4.     String? data;
5.
6.     Registrasi({this.code, this.status, this.data});
7.
8.     factory Registrasi.fromJson(Map<String, dynamic> obj) {
9.         return Registrasi(
10.            code: obj['code'],
11.            status: obj['status'],
12.            data: obj['data']);
13.     }
14. }
```

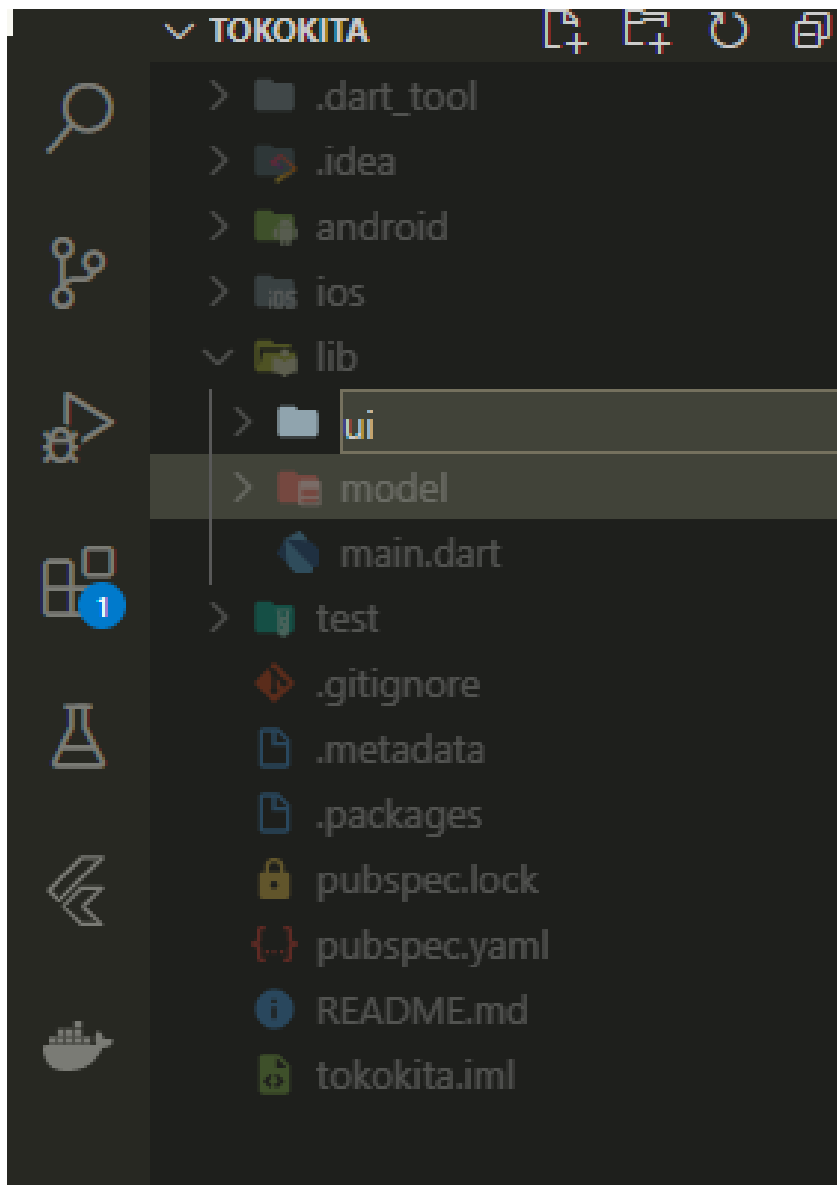
## Produk

Buat sebuah file dengan nama **produk.dart** pada folder **model**. Kemudian ketikkan kode berikut

```
1. class Produk {
2.     int? id;
3.     String? kodeProduk;
4.     String? namaProduk;
5.     int? hargaProduk;
6.
7.     Produk({this.id, this.kodeProduk, this.namaProduk, this.hargaProduk});
8.
9.     factory Produk.fromJson(Map<String, dynamic> obj) {
10.        return Produk(
11.            code: obj['id'],
12.            kodeProduk: obj['kode_produk'],
13.            namaProduk: obj['nama_produk'],
14.            hargaProduk: obj['harga']
15.        );
16.    }
17. }
```

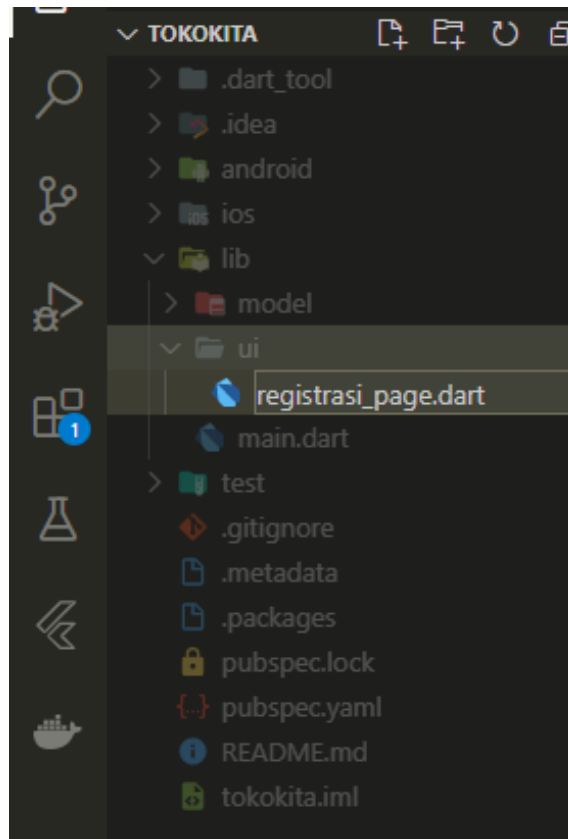
## Membuat Halaman

Pertama kita akan memecah bagian-bagian kode menjadi beberapa bagian, adapun untuk tampilan, dikelompokkan kedalam folder **ui**.



Registrasi

Buat sebuah file dengan nama **registrasi\_page.dart** pada folder **ui**.



Pada file `registrasi_page.dart` ketikkan kode berikut

```
1. import 'package:flutter/material.dart';
2.
3. class RegistrasiPage extends StatefulWidget {
4.   const RegistrasiPage({Key? key}) : super(key: key);
5.
6.   @override
7.   _RegistrasiPageState createState() => _RegistrasiPageState();
8. }
9.
10. class _RegistrasiPageState extends State<RegistrasiPage> {
11.   final _formKey = GlobalKey<FormState>();
12.   bool _isLoading = false;
13.
14.   final _namaTextboxController = TextEditingController();
15.   final _emailTextboxController = TextEditingController();
16.   final _passwordTextboxController = TextEditingController();
17.
18.   @override
19.   Widget build(BuildContext context) {
20.     return Scaffold(
21.       appBar: AppBar(
22.         title: const Text("Registrasi"),
23.       ),
24.       body: SingleChildScrollView(
25.         child: Padding(
26.           padding: const EdgeInsets.all(8.0),
27.           child: Form(
28.             key: _formKey,
29.             child: Column(
```

```

30.         mainAxisAlignment: MainAxisAlignment.center,
31.         children: [
32.             _namaTextField(),
33.             _emailTextField(),
34.             _passwordTextField(),
35.             _passwordKonfirmasiTextField(),
36.             _buttonRegistrasi()
37.         ],
38.     ),
39. ),
40. ),
41. ),
42. );
43. }
44.
45. //Membuat Textbox Nama
46. Widget _namaTextField() {
47.     return TextFormField(
48.         decoration: const InputDecoration(labelText: "Nama"),
49.         keyboardType: TextInputType.text,
50.         controller: _namaTextboxController,
51.         validator: (value) {
52.             if (value!.length < 3) {
53.                 return "Nama harus diisi minimal 3 karakter";
54.             }
55.             return null;
56.         },
57.     );
58. }
59.
60. //Membuat Textbox email
61. Widget _emailTextField() {
62.     return TextFormField(
63.         decoration: const InputDecoration(labelText: "Email"),
64.         keyboardType: TextInputType.emailAddress,
65.         controller: _emailTextboxController,
66.         validator: (value) {
67.             //validasi harus diisi
68.             if (value!.isEmpty) {
69.                 return 'Email harus diisi';
70.             }
71.             //validasi email
72.             Pattern pattern =
73.                 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,}))$';
74.             RegExp regex = RegExp(pattern.toString());
75.             if (!regex.hasMatch(value)) {
76.                 return "Email tidak valid";
77.             }
78.             return null;
79.         },
80.     );
81. }
82.
83. //Membuat Textbox password
84. Widget _passwordTextField() {
85.     return TextFormField(
86.         decoration: const InputDecoration(labelText: "Password"),
87.         keyboardType: TextInputType.text,
88.         obscureText: true,
89.         controller: _passwordTextboxController,
90.         validator: (value) {
91.             //jika karakter yang dimasukkan kurang dari 6 karakter
92.             if (value!.length < 6) {

```

```

93.         return "Password harus diisi minimal 6 karakter";
94.     }
95.     return null;
96. },
97. );
98. }
99.
100. //membuat textbox Konfirmasi Password
101. Widget _passwordKonfirmasiTextField() {
102.     return TextFormField(
103.         decoration: const InputDecoration(labelText: "Konfirmasi Password"),
104.         keyboardType: TextInputType.text,
105.         obscureText: true,
106.         validator: (value) {
107.             //jika inputan tidak sama dengan password
108.             if (value != _passwordTextboxController.text) {
109.                 return "Konfirmasi Password tidak sama";
110.             }
111.             return null;
112.         },
113.     );
114. }
115.
116. //Membuat Tombol Registrasi
117. Widget _buttonRegistrasi() {
118.     return ElevatedButton(
119.         child: const Text("Registrasi"),
120.         onPressed: () {
121.             var validate = _formKey.currentState!.validate();
122.         });
123. }
124. }

```

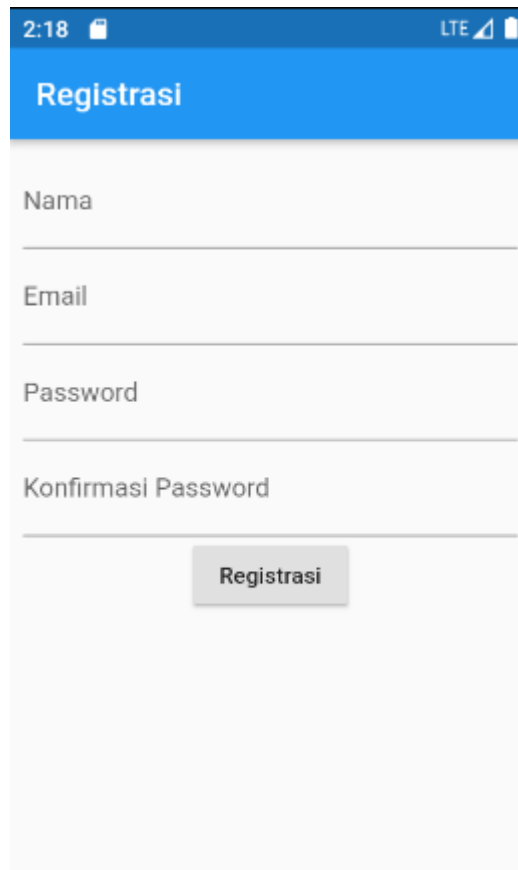
Untuk mencoba halaman `registrasi_page`, buka file `main.dart` kemudian ubah kode menjadi seperti berikut ini

```

1. import 'package:flutter/material.dart';
2. import 'package:tokokita/ui/registrasi_page.dart';
3.
4. void main() {
5.     runApp(const MyApp());
6. }
7.
8. class MyApp extends StatelessWidget {
9.     const MyApp({Key? key}) : super(key: key);
10.
11.     @override
12.     Widget build(BuildContext context) {
13.         return const MaterialApp(
14.             title: 'Toko Kita',
15.             debugShowCheckedModeBanner: false,
16.             home: RegistrasiPage(),
17.         );
18.     }
19. }

```

Dan hasilnya seperti berikut



## Login

Buat sebuah file dengan nama `login_page.dart` pada folder `ui` dengan kode berikut

```
1. import 'package:flutter/material.dart';
2. import 'package:tokokita/ui/registrasi_page.dart';
3.
4. class LoginPage extends StatefulWidget {
5.   const LoginPage({Key? key}) : super(key: key);
6.
7.   @override
8.   _LoginPageState createState() => _LoginPageState();
9. }
10.
11. class _LoginPageState extends State<LoginPage> {
12.   final _formKey = GlobalKey<FormState>();
13.   bool _isLoading = false;
14.
15.   final _emailTextboxController = TextEditingController();
16.   final _passwordTextboxController = TextEditingController();
17.
18.   @override
19.   Widget build(BuildContext context) {
20.     return Scaffold(
21.       appBar: AppBar(
22.         title: const Text('Login'),
23.       ),
24.       body: SingleChildScrollView(
25.         child: Padding(
26.           padding: const EdgeInsets.all(8.0),
27.           child: Form(
28.             key: _formKey,
```

```

29.         child: Column(
30.           children: [
31.             _emailTextField(),
32.             _passwordTextField(),
33.             _buttonLogin(),
34.             const SizedBox(
35.               height: 30,
36.             ),
37.             _menuRegistrasi()
38.           ],
39.         ),
40.       ),
41.     ),
42.   ),
43. );
44. }
45.
46. //Membuat Textbox email
47. Widget _emailTextField() {
48.   return TextFormField(
49.     decoration: const InputDecoration(labelText: "Email"),
50.     keyboardType: TextInputType.emailAddress,
51.     controller: _emailTextboxController,
52.     validator: (value) {
53.       //validasi harus diisi
54.       if (value!.isEmpty) {
55.         return 'Email harus diisi';
56.       }
57.       return null;
58.     },
59.   );
60. }
61.
62. //Membuat Textbox password
63. Widget _passwordTextField() {
64.   return TextFormField(
65.     decoration: const InputDecoration(labelText: "Password"),
66.     keyboardType: TextInputType.text,
67.     obscureText: true,
68.     controller: _passwordTextboxController,
69.     validator: (value) {
70.       //jika karakter yang dimasukkan kurang dari 6 karakter
71.       if (value!.isEmpty) {
72.         return "Password harus diisi";
73.       }
74.       return null;
75.     },
76.   );
77. }
78.
79. //Membuat Tombol Login
80. Widget _buttonLogin() {
81.   return ElevatedButton(
82.     child: const Text("Login"),
83.     onPressed: () {
84.       var validate = _formKey.currentState!.validate();
85.     });
86. }
87.
88. // Membuat menu untuk membuka halaman registrasi
89. Widget _menuRegistrasi() {
90.   return Center(
91.     child: InkWell(
92.       child: const Text(
93.         "Registrasi",

```

```

94.         style: TextStyle(color: Colors.blue),
95.     ),
96.     onTap: () {
97.         Navigator.push(context,
98.             MaterialPageRoute(builder: (context) => const RegistrasiPage()));
99.     },
100.    ),
101.    );
102.    }
103.    }

```

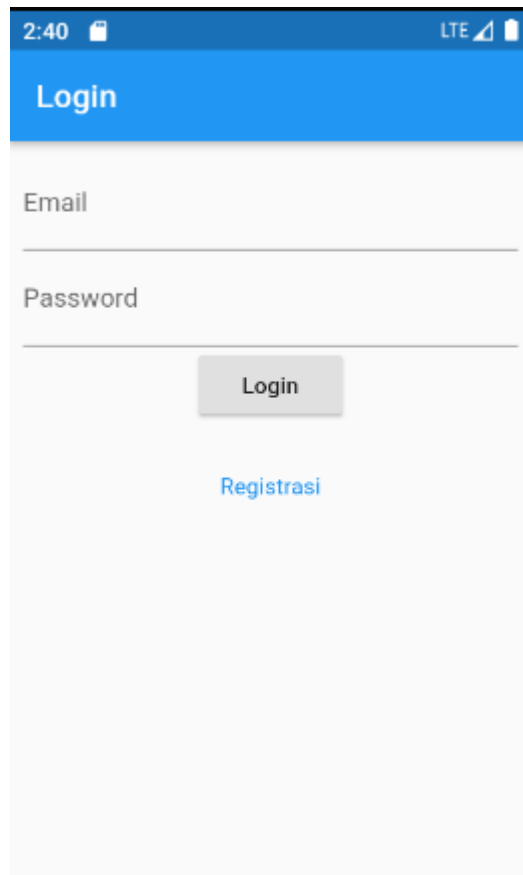
Untuk mencobanya modifikasi file **main.dart** dimana pada bagian home akan memanggil **LoginPage()**

```

1. import 'package:flutter/material.dart';
2. import 'package:tokokita/ui/login_page.dart';
3.
4. void main() {
5.     runApp(const MyApp());
6. }
7.
8. class MyApp extends StatelessWidget {
9.     const MyApp({Key? key}) : super(key: key);
10.
11.     @override
12.     Widget build(BuildContext context) {
13.         return const MaterialApp(
14.             title: 'Toko Kita',
15.             debugShowCheckedModeBanner: false,
16.             home: LoginPage(),
17.         );
18.     }
19. }

```

Pada saat dijalankan akan terdapat link untuk membuka halaman registrasi pada bagian bawah form



### Form Produk

Form produk yang akan kita buat berikut ini memiliki 2 fungsi yaitu untuk menambah data produk dan mengubah data produk

Buat sebuah file dengan nama **produk\_form.dart** pada folder **ui** dengan kode berikut

```
1. import 'package:flutter/material.dart';
2. import 'package:tokokita/model/produk.dart';
3.
4. class ProdukForm extends StatefulWidget {
5.   Produk? produk;
6.
7.   ProdukForm({Key? key, this.produk}) : super(key: key);
8.
9.   @override
10.  _ProdukFormState createState() => _ProdukFormState();
11. }
12.
13. class _ProdukFormState extends State<ProdukForm> {
14.   final _formKey = GlobalKey<FormState>();
15.   bool _isLoading = false;
16.   String judul = "TAMBAH PRODUK";
17.   String tombolSubmit = "SIMPAN";
18.
19.   final _kodeProdukTextboxController = TextEditingController();
20.   final _namaProdukTextboxController = TextEditingController();
21.   final _hargaProdukTextboxController = TextEditingController();
22.
23.   @override
24.   void initState() {
```

```

25.     super.initState();
26.     isUpdate();
27.   }
28.
29.   isUpdate() {
30.     if (widget.produk != null) {
31.       setState(() {
32.         judul = "UBAH PRODUK";
33.         tombolSubmit = "UBAH";
34.         _kodeProdukTextboxController.text = widget.produk!.kodeProduk!;
35.         _namaProdukTextboxController.text = widget.produk!.namaProduk!;
36.         _hargaProdukTextboxController.text =
37.           widget.produk!.hargaProduk.toString();
38.       });
39.     } else {
40.       judul = "TAMBAH PRODUK";
41.       tombolSubmit = "SIMPAN";
42.     }
43.   }
44.
45.   @override
46.   Widget build(BuildContext context) {
47.     return Scaffold(
48.       appBar: AppBar(title: Text(judul)),
49.       body: SingleChildScrollView(
50.         child: Padding(
51.           padding: const EdgeInsets.all(8.0),
52.           child: Form(
53.             key: _formKey,
54.             child: Column(
55.               children: [
56.                 _kodeProdukTextField(),
57.                 _namaProdukTextField(),
58.                 _hargaProdukTextField(),
59.                 _buttonSubmit()
60.               ],
61.             ),
62.           ),
63.         ),
64.       );
65.   }
66. }
67.
68. //Membuat Textbox Kode Produk
69. Widget _kodeProdukTextField() {
70.   return TextFormField(
71.     decoration: const InputDecoration(labelText: "Kode Produk"),
72.     keyboardType: TextInputType.text,
73.     controller: _kodeProdukTextboxController,
74.     validator: (value) {
75.       if (value!.isEmpty) {
76.         return "Kode Produk harus diisi";
77.       }
78.       return null;
79.     },
80.   );
81. }
82.
83. //Membuat Textbox Nama Produk
84. Widget _namaProdukTextField() {
85.   return TextFormField(
86.     decoration: const InputDecoration(labelText: "Nama Produk"),
87.     keyboardType: TextInputType.text,
88.     controller: _namaProdukTextboxController,
89.     validator: (value) {

```

```

90.     if (value!.isEmpty) {
91.         return "Nama Produk harus diisi";
92.     }
93.     return null;
94.   },
95. );
96. }
97.
98. //Membuat Textbox Harga Produk
99. Widget _hargaProdukTextField() {
100.    return TextFormField(
101.        decoration: const InputDecoration(labelText: "Harga"),
102.        keyboardType: TextInputType.number,
103.        controller: _hargaProdukTextboxController,
104.        validator: (value) {
105.            if (value!.isEmpty) {
106.                return "Harga harus diisi";
107.            }
108.            return null;
109.        },
110.    );
111. }
112.
113. //Membuat Tombol Simpan/Ubah
114. Widget _buttonSubmit() {
115.    return OutlinedButton(
116.        child: Text(tombolSubmit),
117.        onPressed: () {
118.            var validate = _formKey.currentState!.validate();
119.        });
120. }
121. }

```

## Detail Produk

Buat sebuah file dengan nama **produk\_detail.dart** pada folder **ui** dengan kode berikut

```

1. import 'package:flutter/material.dart';
2. import 'package:tokokita/model/produk.dart';
3. import 'package:tokokita/ui/produk_form.dart';
4.
5. class ProdukDetail extends StatefulWidget {
6.     Produk? produk;
7.
8.     ProdukDetail({Key? key, this.produk}) : super(key: key);
9.
10.    @override
11.    _ProdukDetailState createState() => _ProdukDetailState();
12. }
13.
14. class _ProdukDetailState extends State<ProdukDetail> {
15.    @override
16.    Widget build(BuildContext context) {
17.        return Scaffold(
18.            appBar: AppBar(
19.                title: const Text('Detail Produk'),
20.            ),
21.            body: Center(
22.                child: Column(
23.                    children: [
24.                        Text(

```

```

25.         "Kode : ${widget.produk!.kodeProduk}",
26.         style: const TextStyle(fontSize: 20.0),
27.     ),
28.     Text(
29.         "Nama : ${widget.produk!.namaProduk}",
30.         style: const TextStyle(fontSize: 18.0),
31.     ),
32.     Text(
33.         "Harga : Rp. ${widget.produk!.hargaProduk.toString()}",
34.         style: const TextStyle(fontSize: 18.0),
35.     ),
36.     _tombolHapusEdit()
37. ],
38. ),
39. ),
40. );
41. }
42.
43. Widget _tombolHapusEdit() {
44.     return Row(
45.         mainAxisAlignment: MainAxisAlignment.min,
46.         children: [
47.             //Tombol Edit
48.             OutlinedButton(
49.                 child: const Text("EDIT"),
50.                 onPressed: () {
51.                     Navigator.push(
52.                         context,
53.                         MaterialPageRoute(
54.                             builder: (context) => ProdukForm(
55.                                 produk: widget.produk!,
56.                             )),
57.                 ),
58.             //Tombol Hapus
59.             OutlinedButton(
60.                 child: const Text("DELETE"), onPressed: () => confirmHapus(),
61.             ),
62.         ],
63.     );
64. }
65. void confirmHapus() {
66.     AlertDialog alertDialog = AlertDialog(
67.         content: const Text("Yakin ingin menghapus data ini?"),
68.         actions: [
69.             //tombol hapus
70.             OutlinedButton(
71.                 child: const Text("Ya"),
72.                 onPressed: () {},
73.             ),
74.             //tombol batal
75.             OutlinedButton(
76.                 child: const Text("Batal"),
77.                 onPressed: () => Navigator.pop(context),
78.             )
79.         ],
80.     );
81.
82.     showDialog(builder: (context) => alertDialog, context: context);
83. }
84. }

```

## Tampil List Produk

Buat sebuah file dengan nama **produk\_page.dart** pada folder **ui** dengan kode berikut

```
1. import 'package:flutter/material.dart';
2. import 'package:tokokita/model/produk.dart';
3. import 'package:tokokita/ui/produk_detail.dart';
4. import 'package:tokokita/ui/produk_form.dart';
5.
6. class ProdukPage extends StatefulWidget {
7.   const ProdukPage({Key? key}) : super(key: key);
8.
9.   @override
10.  _ProdukPageState createState() => _ProdukPageState();
11. }
12.
13. class _ProdukPageState extends State<ProdukPage> {
14.   @override
15.  Widget build(BuildContext context) {
16.    return Scaffold(
17.      appBar: AppBar(
18.        title: const Text('List Produk'),
19.        actions: [
20.          Padding(
21.            padding: const EdgeInsets.only(right: 20.0),
22.            child: GestureDetector(
23.              child: const Icon(Icons.add, size: 26.0),
24.              onTap: () async {
25.                Navigator.push(context,
26.                  MaterialPageRoute(builder: (context) => ProdukForm()));
27.              },
28.            ))
29.        ],
30.      ),
31.      drawer: Drawer(
32.        child: ListView(
33.          children: [
34.            ListTile(
35.              title: const Text('Logout'),
36.              trailing: const Icon(Icons.logout),
37.              onTap: () async {},
38.            )
39.          ],
40.        ),
41.      ),
42.      body: ListView(
43.        children: [
44.          ItemProduk(
45.            produk: Produk(
46.              id: 1,
47.              kodeProduk: 'A001',
48.              namaProduk: 'Kamera',
49.              hargaProduk: 5000000)),
50.          ItemProduk(
51.            produk: Produk(
52.              id: 2,
53.              kodeProduk: 'A002',
54.              namaProduk: 'Kulkas',
55.              hargaProduk: 2500000)),
56.          ItemProduk(
57.            produk: Produk(
58.              id: 3,
59.              kodeProduk: 'A003',
60.              namaProduk: 'Mesin Cuci',
```

```

61.             hargaProduk: 2000000)),
62.         ],
63.     ));
64. }
65. }
66.
67. class ItemProduk extends StatelessWidget {
68.   final Produk produk;
69.
70.   const ItemProduk({Key? key, required this.produk}) : super(key: key);
71.
72.   @override
73.   Widget build(BuildContext context) {
74.     return GestureDetector(
75.       onTap: () {
76.         Navigator.push(
77.           context,
78.           MaterialPageRoute(
79.             builder: (context) => ProdukDetail(
80.               produk: produk,
81.             ));
82.       },
83.       child: Card(
84.         child: ListTile(
85.           title: Text(produk.namaProduk!),
86.           subtitle: Text(produk.hargaProduk.toString()),
87.         ),
88.       ),
89.     );
90.   }
91. }

```

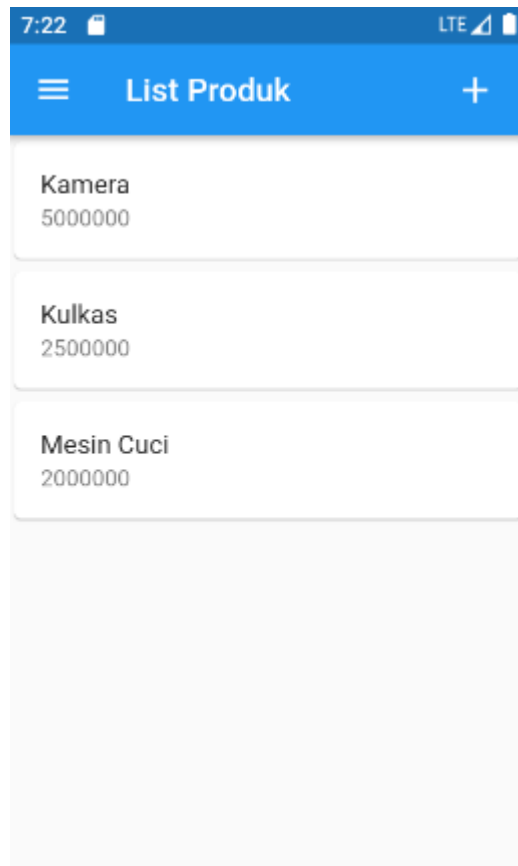
Untuk mencobanya modifikasi file **main.dart** menjadi seperti berikut

```

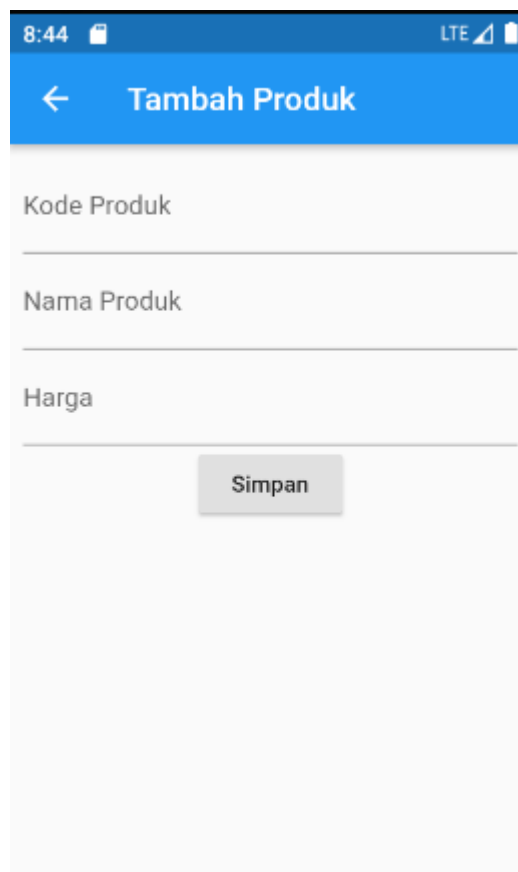
1. import 'package:flutter/material.dart';
2. import 'package:tokokita/ui/produk_page.dart';
3.
4. void main() {
5.   runApp(const MyApp());
6. }
7.
8. class MyApp extends StatelessWidget {
9.   const MyApp({Key? key}) : super(key: key);
10.
11.   @override
12.   Widget build(BuildContext context) {
13.     return const MaterialApp(
14.       title: 'Toko Kita',
15.       debugShowCheckedModeBanner: false,
16.       home: ProdukPage(),
17.     );
18.   }
19. }

```

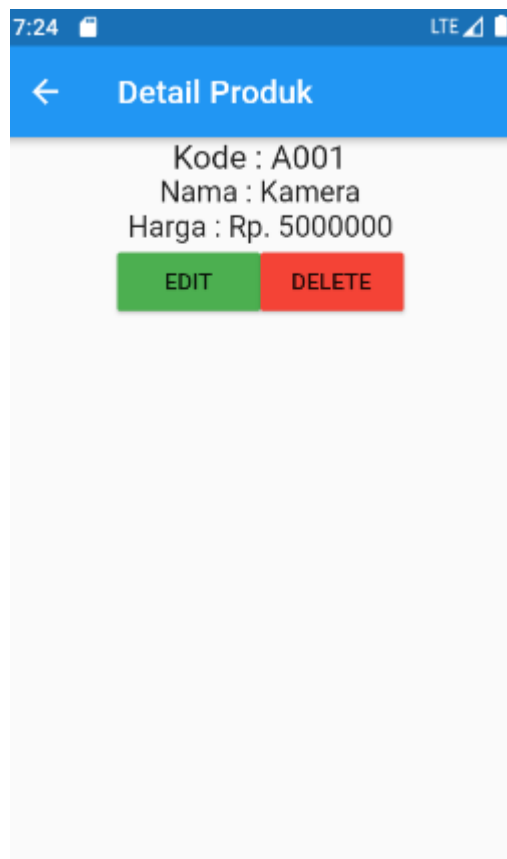
Maka tampilannya akan menjadi seperti berikut



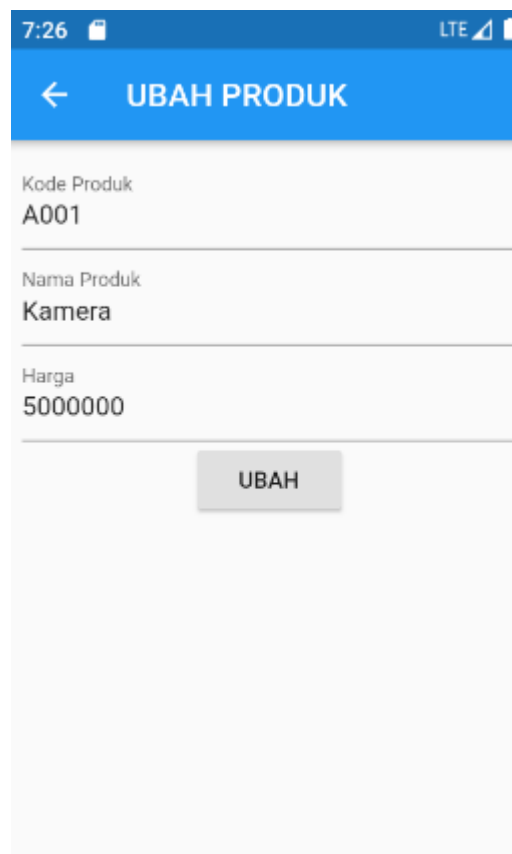
Pada saat tombol tambah diklik maka akan muncul form produk seperti berikut



Jika salah satu data produk diklik maka akan muncul detail produk



Ketika tombol **EDIT** diklik maka akan muncul form produk untuk mengubah data produk



Pada materi selanjutnya akan ada modifikasi pada tampil produk agar dapat menampilkan data dari Rest API serta modifikasi pada Form Produk sehingga dapat berfungsi untuk menyimpan ataupun mengubah data pada Rest API

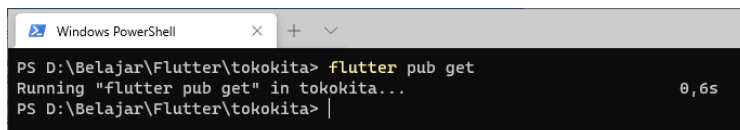
## Membuat Helper Modul

### Menambahkan dependencies

Dalam pembuatan aplikasi ini dibutuhkan dependency untuk menerima dan mengirim request ke Rest API (**http**) dan dependency 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
```

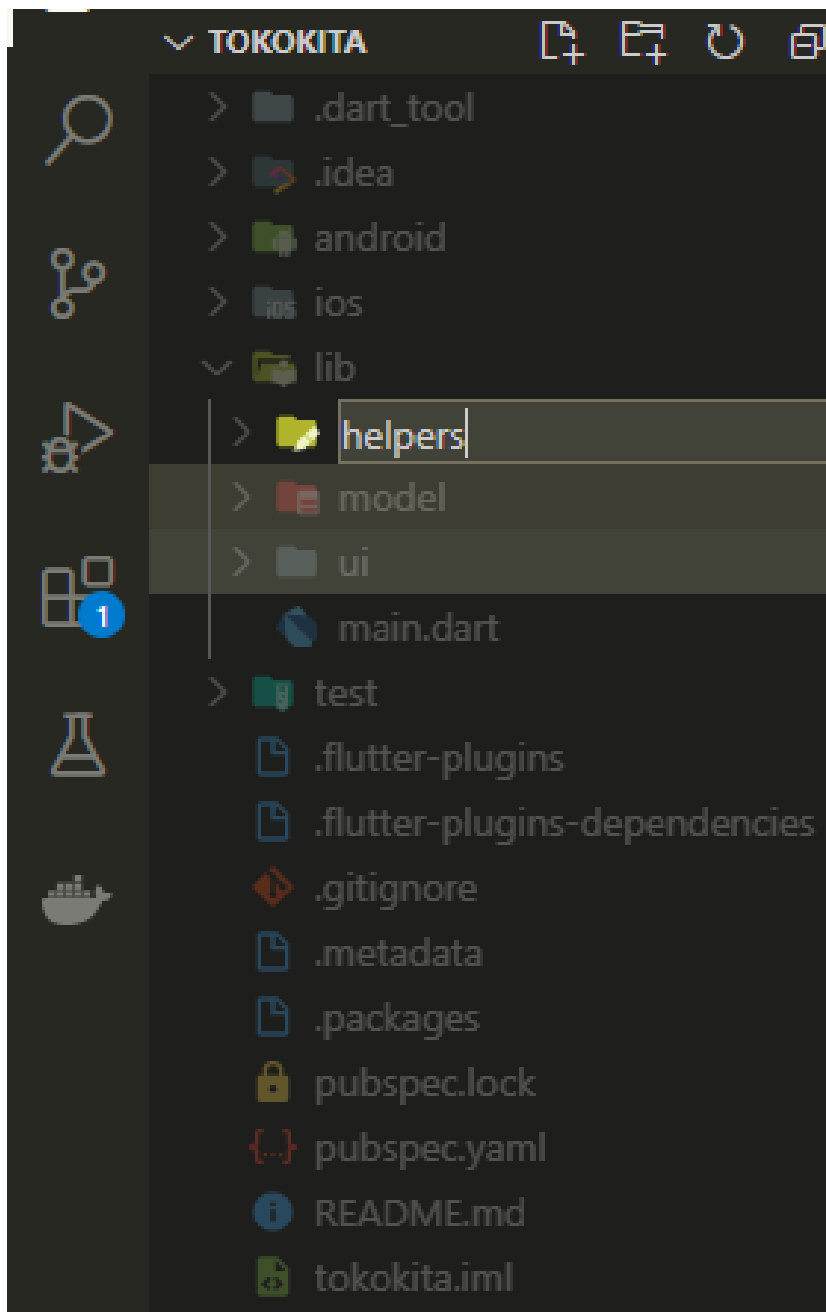
Untuk mengunduh dependencies atau package yang telah ditambahkan, buka **CommandPrompt** kemudian masuk ke folder proyek dan ketikkan **flutter pub get** kemudian tekan Enter



```
Windows PowerShell
PS D:\Belajar\Flutter\tokokita> flutter pub get
Running "flutter pub get" in tokokita... 0,6s
PS D:\Belajar\Flutter\tokokita> |
```

### 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

```
1. import 'package:shared_preferences/shared_preferences.dart';
2.
3. class UserInfo {
4.   Future setToken(String value) async {
5.     final SharedPreferences pref = await SharedPreferences.getInstance();
6.     return pref.setString("token", value);
7.   }
8.
9.   Future<String?> getToken() async {
10.    final SharedPreferences pref = await SharedPreferences.getInstance();
```

```

11.     return pref.getString("token");
12. }
13.
14. Future setUserID(int value) async {
15.     final SharedPreferences pref = await SharedPreferences.getInstance();
16.     return pref.setInt("userID", value);
17. }
18.
19. Future<int?> getUserID() async {
20.     final SharedPreferences pref = await SharedPreferences.getInstance();
21.     return pref.getInt("userID");
22. }
23.
24. Future logout() async {
25.     final SharedPreferences pref = await SharedPreferences.getInstance();
26.     pref.clear();
27. }
28. }

```

## Http request

### *Membuat Modul Error Handling*

Buat sebuah file pada folder **helpers** dengan nama **app\_exception.dart** kemudian ketikkan kode berikut

```

1. class AppException implements Exception {
2.     final _message;
3.     final _prefix;
4.
5.     AppException([this._message, this._prefix]);
6.
7.     @override
8.     String toString() {
9.         return "$_prefix$_message";
10.    }
11. }
12.
13. class FetchDataException extends AppException {
14.     FetchDataException([String? message])
15.         : super(message, "Error During Communication: ");
16. }
17.
18. class BadRequestException extends AppException {
19.     BadRequestException([message]) : super(message, "Invalid Request: ");
20. }
21.
22. class UnauthorisedException extends AppException {
23.     UnauthorisedException([message]) : super(message, "Unauthorised: ");
24. }
25.
26. class UnprocessableEntityException extends AppException {
27.     UnprocessableEntityException([message])
28.         : super(message, "Unprocessable Entity: ");
29. }
30.
31. class InvalidInputException extends AppException {
32.     InvalidInputException([String? message]) : super(message, "Invalid Input: ");
33. }

```

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

```
1. import 'dart:io';
2.
3. import 'package:http/http.dart' as http;
4. import 'package:tokokita/helpers/user_info.dart';
5. import 'app_exception.dart';
6.
7. class Api {
8.   Future<dynamic> post(dynamic url, dynamic data) async {
9.     var token = await UserInfo().getToken();
10.    var responseJson;
11.    try {
12.      final response = await http.post(Uri.parse(url),
13.        body: data,
14.        headers: {HttpHeaders.authorizationHeader: "Bearer $token"});
15.      responseJson = _returnResponse(response);
16.    } on SocketException {
17.      throw FetchDataException('No Internet connection');
18.    }
19.    return responseJson;
20.  }
21.
22. Future<dynamic> get(dynamic url) async {
23.   var token = await UserInfo().getToken();
24.   var responseJson;
25.   try {
26.     final response = await http.get(url,
27.       headers: {HttpHeaders.authorizationHeader: "Bearer $token"});
28.     responseJson = _returnResponse(response);
29.   } on SocketException {
30.     throw FetchDataException('No Internet connection');
31.   }
32.   return responseJson;
33. }
34.
35. Future<dynamic> delete(dynamic url) async {
36.   var token = await UserInfo().getToken();
37.   var responseJson;
38.   try {
39.     final response = await http.delete(url,
40.       headers: {HttpHeaders.authorizationHeader: "Bearer $token"});
41.     responseJson = _returnResponse(response);
42.   } on SocketException {
43.     throw FetchDataException('No Internet connection');
44.   }
45.   return responseJson;
46. }
47.
48. dynamic _returnResponse(http.Response response) {
49.   switch (response.statusCode) {
```

```

50.     case 200:
51.         return response;
52.     case 400:
53.         throw BadRequestException(response.body.toString());
54.     case 401:
55.     case 403:
56.         throw UnauthorizedException(response.body.toString());
57.     case 422:
58.         throw InvalidInputException(response.body.toString());
59.     case 500:
60.     default:
61.         throw FetchDataException(
62.             'Error occured while Communication with Server with StatusCode :
63.             ${response.statusCode}');
64.     }
65. }

```

### 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)

```

1. class ApiUrl {
2.     static const String baseUrl = 'http://10.0.2.2/toko-api/public';
3.
4.     static const String registrasi = baseUrl + '/registrasi';
5.     static const String login = baseUrl + '/login';
6.     static const String listProduk = baseUrl + '/produk';
7.     static const String createProduk = baseUrl + '/produk';
8.
9.     static String updateProduk(int id) {
10.         return baseUrl + '/produk/' + id.toString() + '/update';
11.     }
12.
13.     static String showProduk(int id) {
14.         return baseUrl + '/produk/' + id.toString();
15.     }
16.
17.     static String deleteProduk(int id) {
18.         return baseUrl + '/produk/' + id.toString();
19.     }
20. }

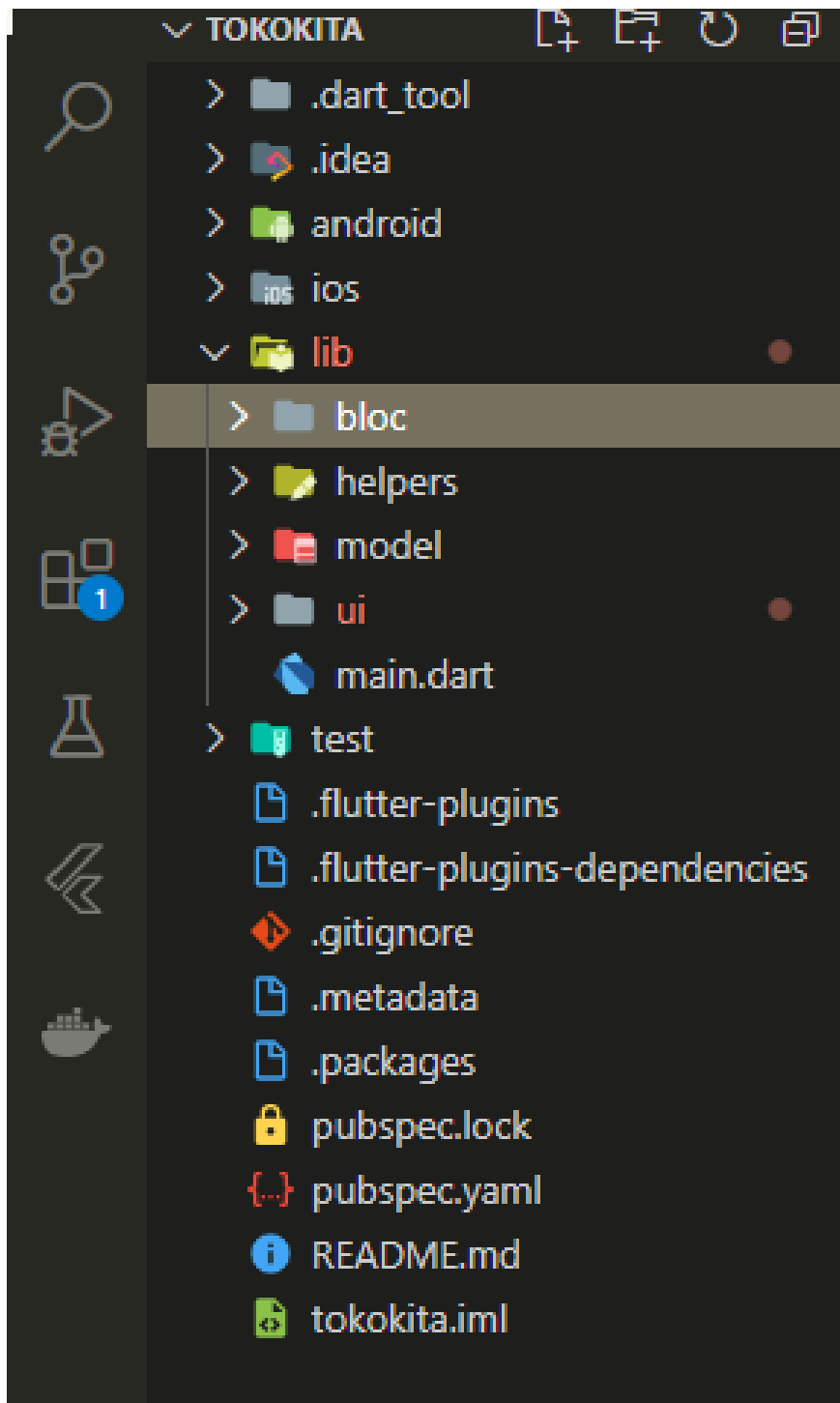
```

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

```
1. import 'dart:convert';  
2.  
3. import 'package:tokokita/helpers/api.dart';  
4. import 'package:tokokita/helpers/api_url.dart';  
5. import 'package:tokokita/model/registrasi.dart';
```

```

6.
7. class RegistrasiBloc {
8.   static Future<Registrasi> registrasi(
9.     {String? nama, String? email, String? password}) async {
10.    String apiUrl = ApiUrl.registrasi;
11.
12.    var body = {"nama": nama, "email": email, "password": password};
13.
14.    var response = await Api().post(apiUrl, body);
15.    var jsonObj = json.decode(response.body);
16.    return Registrasi.fromJson(jsonObj);
17.  }
18. }

```

### Login

Buat sebuah file dengan nama **login\_bloc.dart** pada folder **bloc**. Kemudian masukkan kode berikut

```

1. import 'dart:convert';
2.
3. import 'package:tokokita/helpers/api.dart';
4. import 'package:tokokita/helpers/api_url.dart';
5. import 'package:tokokita/model/login.dart';
6.
7. class LoginBloc {
8.   static Future<Login> login({String? email, String? password}) async {
9.    String apiUrl = ApiUrl.login;
10.    var body = {"email": email, "password": password};
11.    var response = await Api().post(apiUrl, body);
12.    var jsonObj = json.decode(response.body);
13.    return Login.fromJson(jsonObj);
14.  }
15. }

```

### Logout

Buat sebuah file dengan nama **logout\_bloc.dart** pada folder **bloc**. Kemudian masukkan kode berikut

```

1. import 'package:tokokita/helpers/user_info.dart';
2.
3. class LogoutBloc{
4.   static Future logout() async {
5.     await UserInfo().logout();
6.   }
7. }

```

### 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

```

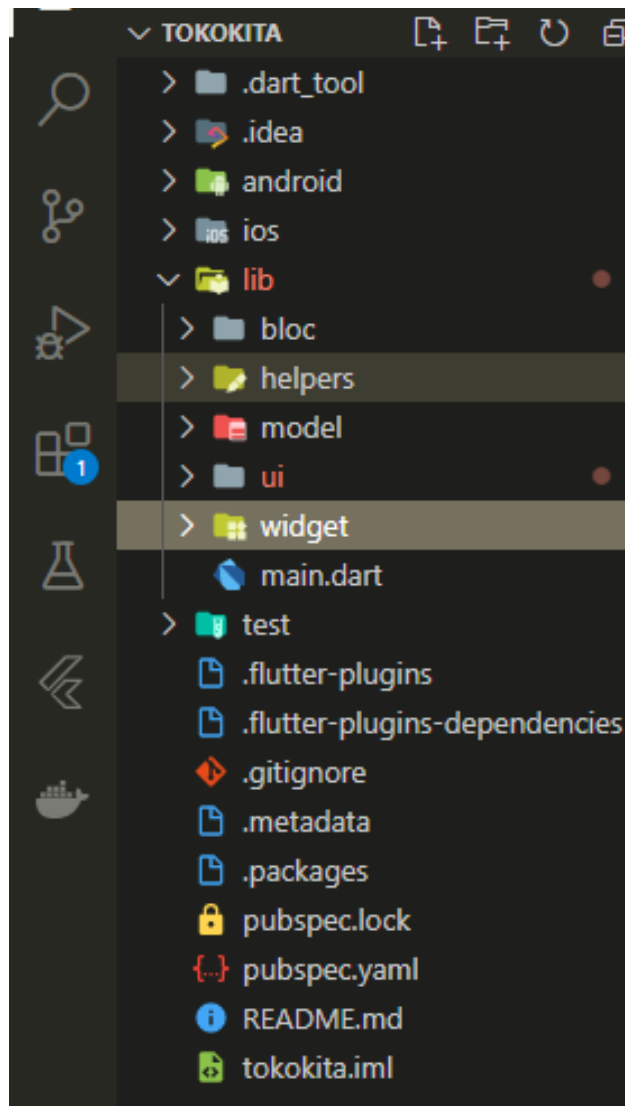
1. import 'dart:convert';
2.
3. import 'package:tokokita/helpers/api.dart';
4. import 'package:tokokita/helpers/api_url.dart';
5. import 'package:tokokita/model/produk.dart';
6.
7. class ProdukBloc {
8.   static Future<List<Produk>> getProduks() async {
9.     String apiUrl = ApiUrl.listProduk;
10.    var response = await Api().get(apiUrl);
11.    var jsonObj = json.decode(response.body);
12.    List<dynamic> listProduk = (jsonObj as Map<String, dynamic>)['data'];
13.    List<Produk> produks = [];
14.    for (int i = 0; i < listProduk.length; i++) {
15.      produks.add(Produk.fromJson(listProduk[i]));
16.    }
17.    return produks;
18.  }
19.
20.  static Future addProduk({Produk? produk}) async {
21.    String apiUrl = ApiUrl.createProduk;
22.
23.    var body = {
24.      "kode_produk": produk!.kodeProduk,
25.      "nama_produk": produk.namaProduk,
26.      "harga": produk.hargaProduk.toString()
27.    };
28.
29.    var response = await Api().post(apiUrl, body);
30.    var jsonObj = json.decode(response.body);
31.    return jsonObj['status'];
32.  }
33.
34.  static Future<bool> updateProduk({required Produk produk}) async {
35.    String apiUrl = ApiUrl.updateProduk(produk.id!);
36.
37.    var body = {
38.      "kode_produk": produk.kodeProduk,
39.      "nama_produk": produk.namaProduk,
40.      "harga": produk.hargaProduk.toString()
41.    };
42.    print("Body : $body");
43.    var response = await Api().post(apiUrl, body);
44.    var jsonObj = json.decode(response.body);
45.    return jsonObj['data'];
46.  }
47.
48.  static Future<bool> deleteProduk({int? id}) async {
49.    String apiUrl = ApiUrl.deleteProduk(id!);
50.
51.    var response = await Api().delete(apiUrl);
52.    var jsonObj = json.decode(response.body);
53.    return (jsonObj as Map<String, dynamic>)['data'];
54.  }
55. }

```

## 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

```
1. import 'package:flutter/material.dart';
2.
3. class Consts {
4.   Consts._();
5.
6.   static const double padding = 16.0;
7.   static const double avatarRadius = 66.0;
8. }
9.
10. class SuccessDialog extends StatelessWidget {
11.   final String? description;
12.   final VoidCallback? okClick;
13.
14.   const SuccessDialog({Key? key, this.description, this.okClick})
```

```

15.     : super(key: key);
16.
17.   @override
18.   Widget build(BuildContext context) {
19.     return Dialog(
20.       shape: RoundedRectangleBorder(
21.         borderRadius: BorderRadius.circular(Consts.padding)),
22.       elevation: 0.0,
23.       backgroundColor: Colors.transparent,
24.       child: dialogContent(context),
25.     );
26.   }
27.
28.   dialogContent(BuildContext context) {
29.     return Container(
30.       padding: const EdgeInsets.only(
31.         top: Consts.padding,
32.         bottom: Consts.padding,
33.         left: Consts.padding,
34.         right: Consts.padding,
35.       ),
36.       margin: const EdgeInsets.only(top: Consts.avatarRadius),
37.       decoration: BoxDecoration(
38.         color: Colors.white,
39.         shape: BoxShape.rectangle,
40.         borderRadius: BorderRadius.circular(Consts.padding),
41.         boxShadow: const [
42.           BoxShadow(
43.             color: Colors.black26,
44.             blurRadius: 10.0,
45.             offset: Offset(0.0, 10.0),
46.           ),
47.         ],
48.       ),
49.       child: Column(
50.         mainAxisAlignment: MainAxisAlignment.min,
51.         children: [
52.           const Text(
53.             "SUKSES",
54.             style: TextStyle(
55.               fontSize: 24.0,
56.               fontWeight: FontWeight.w700,
57.               color: Colors.green),
58.           ),
59.           const SizedBox(height: 16.0),
60.           Text(
61.             description!,
62.             textAlign: TextAlign.center,
63.             style: const TextStyle(
64.               fontSize: 16.0,
65.             ),
66.           ),
67.           const SizedBox(height: 24.0),
68.           Align(
69.             alignment: Alignment.bottomRight,
70.             child: OutlinedButton(
71.               onPressed: () {
72.                 Navigator.of(context).pop(); // To close the dialog
73.                 onClick!();
74.               },
75.               child: const Text("OK"),
76.             ),
77.         ],
78.       ),
79.     ),

```

```

80.   );
81.   }
82. }

```

Kemudian buat file dengan nama **warning\_dialog.dart** dengan kode

```

1. import 'package:flutter/material.dart';
2.
3. class Consts {
4.   Consts._();
5.
6.   static const double padding = 16.0;
7.   static const double avatarRadius = 66.0;
8. }
9.
10. class WarningDialog extends StatelessWidget {
11.   final String? description;
12.   final VoidCallback? onClick;
13.
14.   const WarningDialog({Key? key, this.description, this.onClick})
15.     : super(key: key);
16.
17.   @override
18.   Widget build(BuildContext context) {
19.     return Dialog(
20.       shape: RoundedRectangleBorder(
21.         borderRadius: BorderRadius.circular(Consts.padding)),
22.       elevation: 0.0,
23.       backgroundColor: Colors.transparent,
24.       child: dialogContent(context),
25.     );
26.   }
27.
28.   dialogContent(BuildContext context) {
29.     return Container(
30.       padding: const EdgeInsets.only(
31.         top: Consts.padding,
32.         bottom: Consts.padding,
33.         left: Consts.padding,
34.         right: Consts.padding,
35.       ),
36.       margin: const EdgeInsets.only(top: Consts.avatarRadius),
37.       decoration: BoxDecoration(
38.         color: Colors.white,
39.         shape: BoxShape.rectangle,
40.         borderRadius: BorderRadius.circular(Consts.padding),
41.         boxShadow: const [
42.           BoxShadow(
43.             color: Colors.black26,
44.             blurRadius: 10.0,
45.             offset: Offset(0.0, 10.0),
46.           ),
47.         ],
48.       ),
49.       child: Column(
50.         mainAxisAlignment: MainAxisAlignment.min,
51.         children: [
52.           const Text(
53.             "GAGAL",
54.             style: TextStyle(
55.               fontSize: 24.0, fontWeight: FontWeight.w700, color: Colors.red),
56.           ),
57.           const SizedBox(height: 16.0),
58.           Text(

```

```

59.         description!,
60.         textAlign: TextAlign.center,
61.         style: const TextStyle(
62.           fontSize: 16.0,
63.         ),
64.       ),
65.     const SizedBox(height: 24.0),
66.     Align(
67.       alignment: Alignment.bottomRight,
68.       child: ElevatedButton(
69.         onPressed: () {
70.           Navigator.of(context).pop(); // To close the dialog
71.           onClick!();
72.         },
73.         child: const Text("OK"),
74.       ),
75.     ),
76.   ],
77. ),
78. );
79. }
80. }

```

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

```

1. import 'package:flutter/material.dart';
2. import 'package:tokokita/helpers/user_info.dart';
3. import 'package:tokokita/ui/login_page.dart';
4. import 'package:tokokita/ui/produk_page.dart';
5.
6. void main() {
7.   runApp(const MyApp());
8. }
9.
10. class MyApp extends StatefulWidget {
11.   const MyApp({Key? key}) : super(key: key);
12.
13.   @override
14.   _MyAppState createState() => _MyAppState();
15. }
16.
17. class _MyAppState extends State<MyApp> {
18.   Widget page = const CircularProgressIndicator();
19.
20.   @override
21.   void initState() {
22.     super.initState();
23.     isLogin();
24.   }
25.
26.   void isLogin() async {
27.     var token = await UserInfo().getToken();
28.     if (token != null) {
29.       setState(() {
30.         page = const ProdukPage();
31.       });
32.     } else {
33.       setState(() {

```

```

34.     page = const LoginPage();
35.   });
36. }
37. }
38.
39. @override
40. Widget build(BuildContext context) {
41.   return MaterialApp(
42.     title: 'Toko Kita',
43.     debugShowCheckedModeBanner: false,
44.     home: page,
45.   );
46. }
47. }

```

Modifikasi registrasi\_page.dart

Buka file **registrasi\_page.dart** pada folder **ui** kemudian modifikasi fungsi **\_buttonRegistrasi** dan tambahkan fungsi dengan nama **\_submit** seperti dibawah

```

119.     //Membuat Tombol Registrasi
120.     Widget _buttonRegistrasi() {
121.       return ElevatedButton(
122.         child: const Text("Registrasi"),
123.         onPressed: () {
124.           var validate = _formKey.currentState!.validate();
125.           if (validate) {
126.             if (!_isLoading) _submit();
127.           }
128.         });
129.     }
130.
131.     void _submit() {
132.       _formKey.currentState!.save();
133.       setState(() {
134.         _isLoading = true;
135.       });
136.       RegistrasiBloc.registrasi(
137.         nama: _namaTextboxController.text,
138.         email: _emailTextboxController.text,
139.         password: _passwordTextboxController.text)
140.         .then((value) {
141.           showDialog(
142.             context: context,
143.             barrierDismissible: false,
144.             builder: (BuildContext context) => SuccessDialog(
145.               description: "Registrasi berhasil, silahkan login",
146.               onClick: () {
147.                 Navigator.pop(context);
148.               },
149.             ));
150.         }, onError: (error) {
151.           showDialog(
152.             context: context,
153.             barrierDismissible: false,
154.             builder: (BuildContext context) => const WarningDialog(
155.               description: "Registrasi gagal, silahkan coba lagi",
156.             ));
157.         });
158.       setState(() {
159.         _isLoading = false;
160.       });
161.     }

```

Sehingga keseluruhan kode pada `registrasi_page.dart` menjadi seperti berikut

```
1. import 'package:flutter/material.dart';
2. import 'package:tokokita/bloc/registrasi_bloc.dart';
3. import 'package:tokokita/widget/success_dialog.dart';
4. import 'package:tokokita/widget/warning_dialog.dart';
5.
6. class RegistrasiPage extends StatefulWidget {
7.   const RegistrasiPage({Key? key}) : super(key: key);
8.
9.   @override
10.  _RegistrasiPageState createState() => _RegistrasiPageState();
11. }
12.
13. class _RegistrasiPageState extends State<RegistrasiPage> {
14.   final _formKey = GlobalKey<FormState>();
15.   bool _isLoading = false;
16.
17.   final _namaTextboxController = TextEditingController();
18.   final _emailTextboxController = TextEditingController();
19.   final _passwordTextboxController = TextEditingController();
20.
21.   @override
22.  Widget build(BuildContext context) {
23.    return Scaffold(
24.      appBar: AppBar(
25.        title: const Text("Registrasi"),
26.      ),
27.      body: SingleChildScrollView(
28.        child: Padding(
29.          padding: const EdgeInsets.all(8.0),
30.          child: Form(
31.            key: _formKey,
32.            child: Column(
33.              mainAxisAlignment: MainAxisAlignment.center,
34.              children: [
35.                _namaTextField(),
36.                _emailTextField(),
37.                _passwordTextField(),
38.                _passwordKonfirmasiTextField(),
39.                _buttonRegistrasi()
40.              ],
41.            ),
42.          ),
43.        ),
44.      );
45.   }
46. }
47.
48. //Membuat Textbox Nama
49. Widget _namaTextField() {
50.   return TextFormField(
51.     decoration: const InputDecoration(labelText: "Nama"),
52.     keyboardType: TextInputType.text,
53.     controller: _namaTextboxController,
54.     validator: (value) {
55.       if (value!.length < 3) {
56.         return "Nama harus diisi minimal 3 karakter";
57.       }
58.       return null;
59.     },
60.   );
```

```

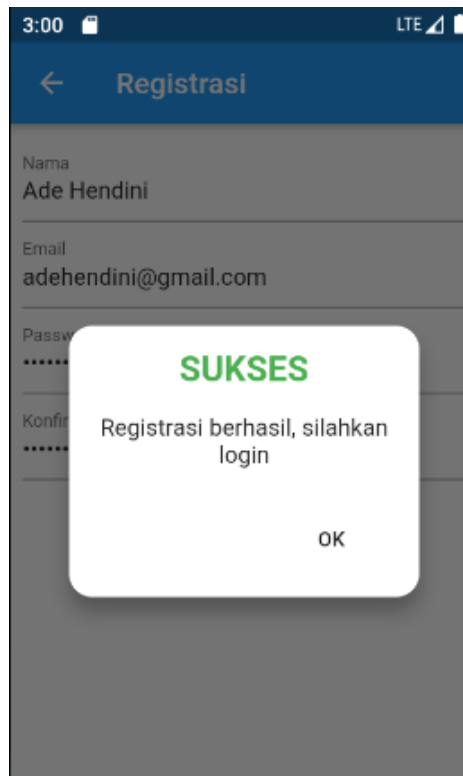
61. }
62.
63. //Membuat Textbox email
64. Widget _emailTextField() {
65.     return TextFormField(
66.         decoration: const InputDecoration(labelText: "Email"),
67.         keyboardType: TextInputType.emailAddress,
68.         controller: _emailTextboxController,
69.         validator: (value) {
70.             //validasi harus diisi
71.             if (value!.isEmpty) {
72.                 return 'Email harus diisi';
73.             }
74.             //validasi email
75.             Pattern pattern =
76.                 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,}))$';
77.             RegExp regex = RegExp(pattern.toString());
78.             if (!regex.hasMatch(value)) {
79.                 return "Email tidak valid";
80.             }
81.             return null;
82.         },
83.     );
84. }
85.
86. //Membuat Textbox password
87. Widget _passwordTextField() {
88.     return TextFormField(
89.         decoration: const InputDecoration(labelText: "Password"),
90.         keyboardType: TextInputType.text,
91.         obscureText: true,
92.         controller: _passwordTextboxController,
93.         validator: (value) {
94.             //jika karakter yang dimasukkan kurang dari 6 karakter
95.             if (value!.length < 6) {
96.                 return "Password harus diisi minimal 6 karakter";
97.             }
98.             return null;
99.         },
100.     );
101. }
102.
103. //membuat textbox Konfirmasi Password
104. Widget _passwordKonfirmasiTextField() {
105.     return TextFormField(
106.         decoration: const InputDecoration(labelText: "Konfirmasi Password"),
107.         keyboardType: TextInputType.text,
108.         obscureText: true,
109.         validator: (value) {
110.             //jika inputan tidak sama dengan password
111.             if (value != _passwordTextboxController.text) {
112.                 return "Konfirmasi Password tidak sama";
113.             }
114.             return null;
115.         },
116.     );
117. }
118.
119. //Membuat Tombol Registrasi
120. Widget _buttonRegistrasi() {
121.     return ElevatedButton(
122.         child: const Text("Registrasi"),
123.         onPressed: () {

```

```

124.         var validate = _formKey.currentState!.validate();
125.         if (validate) {
126.             if (!_isLoading) _submit();
127.         }
128.     });
129. }
130.
131. void _submit() {
132.     _formKey.currentState!.save();
133.     setState(() {
134.         _isLoading = true;
135.     });
136.     RegistrasiBloc.registrasi(
137.         nama: _namaTextboxController.text,
138.         email: _emailTextboxController.text,
139.         password: _passwordTextboxController.text)
140.     .then((value) {
141.         showDialog(
142.             context: context,
143.             barrierDismissible: false,
144.             builder: (BuildContext context) => SuccessDialog(
145.                 description: "Registrasi berhasil, silahkan login",
146.                 onClick: () {
147.                     Navigator.pop(context);
148.                 },
149.             ));
150.     }, onError: (error) {
151.         showDialog(
152.             context: context,
153.             barrierDismissible: false,
154.             builder: (BuildContext context) => const WarningDialog(
155.                 description: "Registrasi gagal, silahkan coba lagi",
156.             ));
157.     });
158.     setState(() {
159.         _isLoading = false;
160.     });
161. }
162. }

```



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

```

83. //Membuat Tombol Login
84. Widget _buttonLogin() {
85.   return ElevatedButton(
86.     child: const Text("Login"),
87.     onPressed: () {
88.       var validate = _formKey.currentState!.validate();
89.       if (validate) {
90.         if (!_isLoading) _submit();
91.       }
92.     });
93. }
94.
95. void _submit() {
96.   _formKey.currentState!.save();
97.   setState(() {
98.     _isLoading = true;
99.   });
100.   LoginBloc.login(
101.     email: _emailTextboxController.text,
102.     password: _passwordTextboxController.text)
103.     .then((value) async {
104.       await UserInfo().setToken(value.token.toString());
105.       await UserInfo().setUserID(int.parse(value.userID.toString()));
106.       Navigator.pushReplacement(
107.         context, MaterialPageRoute(builder: (context) => const
108.           ProdukPage()));
109.     }, onError: (error) {
110.       print(error);
111.       showDialog(

```

```

111.         context: context,
112.         barrierDismissible: false,
113.         builder: (BuildContext context) => const WarningDialog(
114.             description: "Login gagal, silahkan coba lagi",
115.         ));
116.     });
117.     setState(() {
118.         _isLoading = false;
119.     });
120. }

```

Adapun kode secara keseluruhan menjadi seperti berikut

```

1. import 'package:flutter/material.dart';
2. import 'package:tokokita/bloc/login_bloc.dart';
3. import 'package:tokokita/helpers/user_info.dart';
4. import 'package:tokokita/ui/produk_page.dart';
5. import 'package:tokokita/ui/registrasi_page.dart';
6. import 'package:tokokita/widget/warning_dialog.dart';
7.
8. class LoginPage extends StatefulWidget {
9.     const LoginPage({Key? key}) : super(key: key);
10.
11.     @override
12.     _LoginPageState createState() => _LoginPageState();
13. }
14.
15. class _LoginPageState extends State<LoginPage> {
16.     final _formKey = GlobalKey<FormState>();
17.     bool _isLoading = false;
18.
19.     final _emailTextboxController = TextEditingController();
20.     final _passwordTextboxController = TextEditingController();
21.
22.     @override
23.     Widget build(BuildContext context) {
24.         return Scaffold(
25.             appBar: AppBar(
26.                 title: const Text('Login'),
27.             ),
28.             body: SingleChildScrollView(
29.                 child: Padding(
30.                     padding: const EdgeInsets.all(8.0),
31.                     child: Form(
32.                         key: _formKey,
33.                         child: Column(
34.                             children: [
35.                                 _emailTextField(),
36.                                 _passwordTextField(),
37.                                 _buttonLogin(),
38.                                 const SizedBox(
39.                                     height: 30,
40.                                 ),
41.                                 _menuRegistrasi()
42.                             ],
43.                         ),
44.                     ),
45.                 ),
46.             ),
47.         );
48.     }
49.
50.     //Membuat Textbox email
51.     Widget _emailTextField() {

```

```

52.     return TextFormField(
53.       decoration: const InputDecoration(labelText: "Email"),
54.       keyboardType: TextInputType.emailAddress,
55.       controller: _emailTextboxController,
56.       validator: (value) {
57.         //validasi harus diisi
58.         if (value!.isEmpty) {
59.           return 'Email harus diisi';
60.         }
61.         return null;
62.       },
63.     );
64.   }
65.
66. //Membuat Textbox password
67. Widget _passwordTextField() {
68.   return TextFormField(
69.     decoration: const InputDecoration(labelText: "Password"),
70.     keyboardType: TextInputType.text,
71.     obscureText: true,
72.     controller: _passwordTextboxController,
73.     validator: (value) {
74.       //jika karakter yang dimasukkan kurang dari 6 karakter
75.       if (value!.isEmpty) {
76.         return "Password harus diisi";
77.       }
78.       return null;
79.     },
80.   );
81. }
82.
83. //Membuat Tombol Login
84. Widget _buttonLogin() {
85.   return ElevatedButton(
86.     child: const Text("Login"),
87.     onPressed: () {
88.       var validate = _formKey.currentState!.validate();
89.       if (validate) {
90.         if (!_isLoading) _submit();
91.       }
92.     });
93. }
94.
95. void _submit() {
96.   _formKey.currentState!.save();
97.   setState(() {
98.     _isLoading = true;
99.   });
100.   LoginBloc.login(
101.     email: _emailTextboxController.text,
102.     password: _passwordTextboxController.text)
103.     .then((value) async {
104.       await UserInfo().setToken(value.token.toString());
105.       await UserInfo().setUserID(int.parse(value.userID.toString()));
106.       Navigator.pushReplacement(
107.         context, MaterialPageRoute(builder: (context) => const
108.         ProdukPage()));
109.     }, onError: (error) {
110.       print(error);
111.       showDialog(
112.         context: context,
113.         barrierDismissible: false,
114.         builder: (BuildContext context) => const WarningDialog(
115.           description: "Login gagal, silahkan coba lagi",

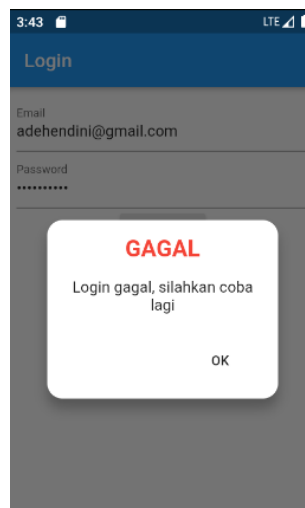
```

```

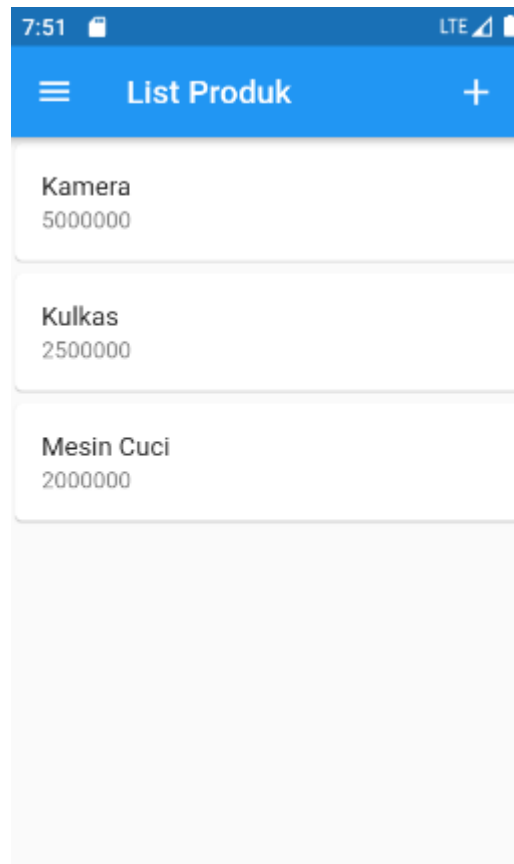
116.     });
117.     setState(() {
118.       _isLoading = false;
119.     });
120.   }
121.
122.   // Membuat menu untuk membuka halaman registrasi
123.   Widget _menuRegistrasi() {
124.     return Center(
125.       child: InkWell(
126.         child: const Text(
127.           "Registrasi",
128.           style: TextStyle(color: Colors.blue),
129.         ),
130.         onTap: () {
131.           Navigator.push(context,
132.             MaterialPageRoute(builder: (context) => const
RegistrasiPage()));
133.         },
134.       ),
135.     );
136.   }
137. }

```

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

```

33. drawer: Drawer(
34.   child: ListView(
35.     children: [
36.       ListTile(
37.         title: const Text('Logout'),
38.         trailing: const Icon(Icons.logout),
39.         onTap: () async {
40.           await LogoutBloc.logout().then((value) => {
41.             Navigator.pushReplacement(
42.               context,
43.               MaterialPageRoute(
44.                 builder: (context) => LoginPage())
45.             ));
46.           },
47.       )
48.     ],
49.   ),
50. ),
51. body: ListView(

```

Secara keseluruhan kode pada **produk\_page.dart** menjadi seperti berikut

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

```

2. import 'package:tokokita/bloc/logout_bloc.dart';
3. import 'package:tokokita/model/produk.dart';
4. import 'package:tokokita/ui/login_page.dart';
5. import 'package:tokokita/ui/produk_detail.dart';
6. import 'package:tokokita/ui/produk_form.dart';
7.
8. class ProdukPage extends StatefulWidget {
9.   const ProdukPage({Key? key}) : super(key: key);
10.
11.   @override
12.   _ProdukPageState createState() => _ProdukPageState();
13. }
14.
15. class _ProdukPageState extends State<ProdukPage> {
16.   @override
17.   Widget build(BuildContext context) {
18.     return Scaffold(
19.       appBar: AppBar(
20.         title: const Text('List Produk'),
21.         actions: [
22.           Padding(
23.             padding: const EdgeInsets.only(right: 20.0),
24.             child: GestureDetector(
25.               child: const Icon(Icons.add, size: 26.0),
26.               onTap: () async {
27.                 Navigator.push(context,
28.                   MaterialPageRoute(builder: (context) => ProdukForm()));
29.               },
30.             ),
31.           ],
32.         ),
33.       drawer: Drawer(
34.         child: ListView(
35.           children: [
36.             ListTile(
37.               title: const Text('Logout'),
38.               trailing: const Icon(Icons.logout),
39.               onTap: () async {
40.                 await LogoutBloc.logout().then((value) => {
41.                   Navigator.pushReplacement(
42.                     context,
43.                     MaterialPageRoute(
44.                       builder: (context) => LoginPage())
45.                   );
46.                 },
47.               ),
48.             ],
49.           ),
50.         ),
51.       body: ListView(
52.         children: [
53.           ItemProduk(
54.             produk: Produk(
55.               id: 1,
56.               kodeProduk: 'A001',
57.               namaProduk: 'Kamera',
58.               hargaProduk: 5000000)),
59.           ItemProduk(
60.             produk: Produk(
61.               id: 2,
62.               kodeProduk: 'A002',
63.               namaProduk: 'Kulkas',
64.               hargaProduk: 2500000)),
65.           ItemProduk(
66.             produk: Produk(

```

```

67.         id: 3,
68.         kodeProduk: 'A003',
69.         namaProduk: 'Mesin Cuci',
70.         hargaProduk: 2000000)),
71.     ],
72.   ));
73. }
74. }
75.
76. class ItemProduk extends StatelessWidget {
77.   final Produk produk;
78.
79.   const ItemProduk({Key? key, required this.produk}) : super(key: key);
80.
81.   @override
82.   Widget build(BuildContext context) {
83.     return GestureDetector(
84.       onTap: () {
85.         Navigator.push(
86.           context,
87.           MaterialPageRoute(
88.             builder: (context) => ProdukDetail(
89.               produk: produk,
90.             ));
91.       },
92.       child: Card(
93.         child: ListTile(
94.           title: Text(produk.namaProduk!),
95.           subtitle: Text(produk.hargaProduk.toString()),
96.         ),
97.       ),
98.     );
99.   }
100. }

```

### 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

```

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';
8.
9. class ProdukPage extends StatefulWidget {
10.   const ProdukPage({Key? key}) : super(key: key);
11.
12.   @override
13.   _ProdukPageState createState() => _ProdukPageState();
14. }
15.
16. class _ProdukPageState extends State<ProdukPage> {
17.   @override
18.   Widget build(BuildContext context) {
19.     return Scaffold(
20.       appBar: AppBar(

```

```

21.     title: const Text('List Produk'),
22.     actions: [
23.       Padding(
24.         padding: const EdgeInsets.only(right: 20.0),
25.         child: GestureDetector(
26.           child: const Icon(Icons.add, size: 26.0),
27.           onTap: () async {
28.             Navigator.push(context,
29.               MaterialPageRoute(builder: (context) => ProdukForm()));
30.           },
31.         ))
32.     ],
33.   ),
34.   drawer: Drawer(
35.     child: ListView(
36.       children: [
37.         ListTile(
38.           title: const Text('Logout'),
39.           trailing: const Icon(Icons.logout),
40.           onTap: () async {
41.             await LogoutBloc.logout().then((value) => {
42.               Navigator.pushReplacement(context,
43.                 MaterialPageRoute(builder: (context) => LoginPage()))
44.             });
45.           },
46.         )
47.       ],
48.     ),
49.   ),
50.   body: FutureBuilder<List>(
51.     future: ProdukBloc.getProduks(),
52.     builder: (context, snapshot) {
53.       if (snapshot.hasError) print(snapshot.error);
54.       return snapshot.hasData
55.         ? ListProduk(
56.           list: snapshot.data,
57.         )
58.         : const Center(
59.           child: CircularProgressIndicator(),
60.         );
61.     },
62.   ),
63. );
64. }
65. }
66.
67. class ListProduk extends StatelessWidget {
68.   final List? list;
69.
70.   const ListProduk({Key? key, this.list}) : super(key: key);
71.
72.   @override
73.   Widget build(BuildContext context) {
74.     return ListView.builder(
75.       itemCount: list == null ? 0 : list!.length,
76.       itemBuilder: (context, i) {
77.         return ItemProduk(
78.           produk: list![i],
79.         );
80.       });
81.   }
82. }
83.
84. class ItemProduk extends StatelessWidget {
85.   final Produk produk;

```

```

86.
87.  const ItemProduk({Key? key, required this.produk}) : super(key: key);
88.
89.  @override
90.  Widget build(BuildContext context) {
91.    return GestureDetector(
92.      onTap: () {
93.        Navigator.push(
94.          context,
95.          MaterialPageRoute(
96.            builder: (context) => ProdukDetail(
97.              produk: produk,
98.            ));
99.      },
100.     child: Card(
101.       child: ListTile(
102.         title: Text(produk.namaProduk!),
103.         subtitle: Text(produk.hargaProduk.toString()),
104.       ),
105.     ),
106.   );
107. }
108. }

```

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

```

67. class ListProduk extends StatelessWidget {
68.   final List? list;
69.
70.   const ListProduk({Key? key, this.list}) : super(key: key);
71.
72.   @override
73.   Widget build(BuildContext context) {
74.     return ListView.builder(
75.       itemCount: list == null ? 0 : list!.length,
76.       itemBuilder: (context, i) {
77.         return ItemProduk(
78.           produk: list![i],
79.         );
80.       });
81.   }
82. }

```

Kemudian perubahan pada bagian **body** menjadi

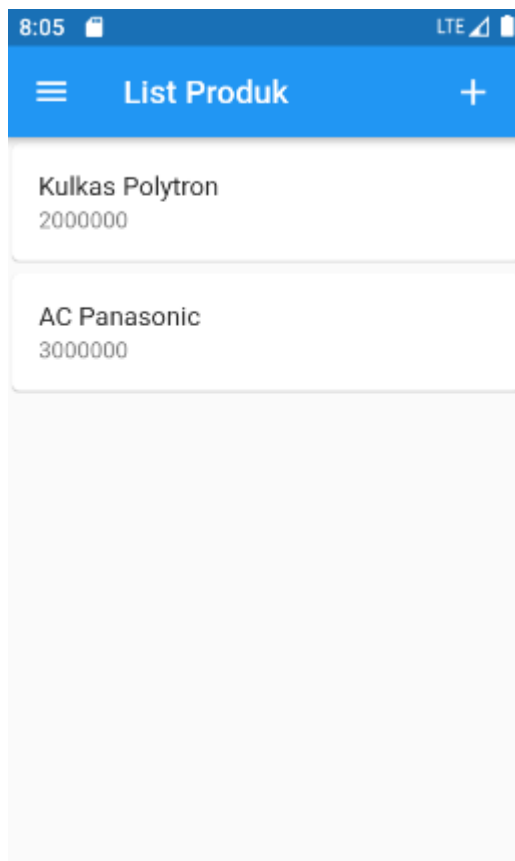
```

50. body: FutureBuilder<List>(
51.   future: ProdukBloc.getProduk(),
52.   builder: (context, snapshot) {
53.     if (snapshot.hasError) print(snapshot.error);
54.     return snapshot.hasData
55.       ? ListProduk(
56.         list: snapshot.data,
57.       )
58.       : const Center(
59.         child: CircularProgressIndicator(),
60.       );
61.   },
62. ),

```

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**

```
116. //Membuat Tombol Simpan/Ubah
117. Widget _buttonSubmit() {
118.   return OutlinedButton(
119.     child: Text(tombolSubmit),
120.     onPressed: () {
121.       var validate = _formKey.currentState!.validate();
122.       if (validate) {
123.         if (!_isLoading) {
124.           if (widget.produk != null) {
```

```

125.             //kondisi update produk
126.
127.             } else {
128.                 //kondisi tambah produk
129.                 simpan();
130.             }
131.         }
132.     }
133. });
134. }
135.
136.     simpan() {
137.         setState(() {
138.             _isLoading = true;
139.         });
140.         Produk createProduk = Produk(id: null);
141.         createProduk.kodeProduk = _kodeProdukTextboxController.text;
142.         createProduk.namaProduk = _namaProdukTextboxController.text;
143.         createProduk.hargaProduk =
144.             int.parse(_hargaProdukTextboxController.text);
145.         ProdukBloc.addProduk(produk: createProduk).then((value) {
146.             Navigator.of(context).push(MaterialPageRoute(
147.                 builder: (BuildContext context) => const ProdukPage()));
148.             showDialog(
149.                 context: context,
150.                 builder: (BuildContext context) => const WarningDialog(
151.                     description: "Simpan gagal, silahkan coba lagi",
152.                 ));
153.         });
154.         setState(() {
155.             _isLoading = false;
156.         });
157.     }

```

Dengan keseluruhan kode menjadi

```

1. import 'package:flutter/material.dart';
2. import 'package:tokokita/bloc/produk_bloc.dart';
3. import 'package:tokokita/model/produk.dart';
4. import 'package:tokokita/ui/produk_page.dart';
5. import 'package:tokokita/widget/warning_dialog.dart';
6.
7. class ProdukForm extends StatefulWidget {
8.     Produk? produk;
9.
10.     ProdukForm({Key? key, this.produk}) : super(key: key);
11.
12.     @override
13.     _ProdukFormState createState() => _ProdukFormState();
14. }
15.
16. class _ProdukFormState extends State<ProdukForm> {
17.     final _formKey = GlobalKey<FormState>();
18.     bool _isLoading = false;
19.     String judul = "TAMBAH PRODUK";
20.     String tombolSubmit = "SIMPAN";
21.
22.     final _kodeProdukTextboxController = TextEditingController();
23.     final _namaProdukTextboxController = TextEditingController();
24.     final _hargaProdukTextboxController = TextEditingController();
25.
26.     @override
27.     void initState() {

```

```

28.     super.initState();
29.     isUpdate();
30. }
31.
32. isUpdate() {
33.     if (widget.produk != null) {
34.         setState(() {
35.             judul = "UBAH PRODUK";
36.             tombolSubmit = "UBAH";
37.             _kodeProdukTextboxController.text = widget.produk!.kodeProduk!;
38.             _namaProdukTextboxController.text = widget.produk!.namaProduk!;
39.             _hargaProdukTextboxController.text =
40.                 widget.produk!.hargaProduk.toString();
41.         });
42.     } else {
43.         judul = "TAMBAH PRODUK";
44.         tombolSubmit = "SIMPAN";
45.     }
46. }
47.
48. @override
49. Widget build(BuildContext context) {
50.     return Scaffold(
51.         appBar: AppBar(title: Text(judul)),
52.         body: SingleChildScrollView(
53.             child: Padding(
54.                 padding: const EdgeInsets.all(8.0),
55.                 child: Form(
56.                     key: _formKey,
57.                     child: Column(
58.                         children: [
59.                             _kodeProdukTextField(),
60.                             _namaProdukTextField(),
61.                             _hargaProdukTextField(),
62.                             _buttonSubmit()
63.                         ],
64.                     ),
65.                 ),
66.             ),
67.         );
68.     };
69. }
70.
71. //Membuat Textbox Kode Produk
72. Widget _kodeProdukTextField() {
73.     return TextFormField(
74.         decoration: const InputDecoration(labelText: "Kode Produk"),
75.         keyboardType: TextInputType.text,
76.         controller: _kodeProdukTextboxController,
77.         validator: (value) {
78.             if (value!.isEmpty) {
79.                 return "Kode Produk harus diisi";
80.             }
81.             return null;
82.         },
83.     );
84. }
85.
86. //Membuat Textbox Nama Produk
87. Widget _namaProdukTextField() {
88.     return TextFormField(
89.         decoration: const InputDecoration(labelText: "Nama Produk"),
90.         keyboardType: TextInputType.text,
91.         controller: _namaProdukTextboxController,
92.         validator: (value) {

```

```

93.         if (value!.isEmpty) {
94.             return "Nama Produk harus diisi";
95.         }
96.         return null;
97.     },
98. );
99. }
100.
101.     //Membuat Textbox Harga Produk
102.     Widget _hargaProdukTextField() {
103.         return TextFormField(
104.             decoration: const InputDecoration(labelText: "Harga"),
105.             keyboardType: TextInputType.number,
106.             controller: _hargaProdukTextboxController,
107.             validator: (value) {
108.                 if (value!.isEmpty) {
109.                     return "Harga harus diisi";
110.                 }
111.                 return null;
112.             },
113. );
114. }
115.
116.     //Membuat Tombol Simpan/Ubah
117.     Widget _buttonSubmit() {
118.         return OutlinedButton(
119.             child: Text(tombolSubmit),
120.             onPressed: () {
121.                 var validate = _formKey.currentState!.validate();
122.                 if (validate) {
123.                     if (!_isLoading) {
124.                         if (widget.produk != null) {
125.                             //kondisi update produk
126.
127.                         } else {
128.                             //kondisi tambah produk
129.                             simpan();
130.                         }
131.                     }
132.                 }
133.             });
134.     }
135.
136.     simpan() {
137.         setState(() {
138.             _isLoading = true;
139.         });
140.         Produk createProduk = Produk(id: null);
141.         createProduk.kodeProduk = _kodeProdukTextboxController.text;
142.         createProduk.namaProduk = _namaProdukTextboxController.text;
143.         createProduk.hargaProduk = int.parse(_hargaProdukTextboxController.text);
144.         ProdukBloc.addProduk(produk: createProduk).then((value) {
145.             Navigator.of(context).push(MaterialPageRoute(
146.                 builder: (BuildContext context) => const ProdukPage()));
147.         }, onError: (error) {
148.             showDialog(
149.                 context: context,
150.                 builder: (BuildContext context) => const WarningDialog(
151.                     description: "Simpan gagal, silahkan coba lagi",
152.                 ));
153.         });
154.         setState(() {
155.             _isLoading = false;
156.         });
157.     }

```

```
158.     }
```

### Membuat fungsi ubah

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

Dengan fungsi ubah sebagai berikut

```
159.     ubah() {
160.         setState(() {
161.             _isLoading = true;
162.         });
163.         Produk updateProduk = Produk(id: null);
164.         updateProduk.id = widget.produk!.id;
165.         updateProduk.kodeProduk = _kodeProdukTextboxController.text;
166.         updateProduk.namaProduk = _namaProdukTextboxController.text;
167.         updateProduk.hargaProduk =
168.             int.parse(_hargaProdukTextboxController.text);
169.         ProdukBloc.updateProduk(produk: updateProduk).then((value) {
170.             Navigator.of(context).push(MaterialPageRoute(
171.                 builder: (BuildContext context) => const ProdukPage()));
172.             showDialog(
173.                 context: context,
174.                 builder: (BuildContext context) => const WarningDialog(
175.                     description: "Permintaan ubah data gagal, silahkan coba
176.                     lagi",
177.                     ));
178.         });
179.         setState(() {
180.             _isLoading = false;
181.         });
181.     }
```

Kemudian kita tambahkan pada fungsi `_buttonSubmit`

```
116.     //Membuat Tombol Simpan/Ubah
117.     Widget _buttonSubmit() {
118.         return OutlinedButton(
119.             child: Text(tombolSubmit),
120.             onPressed: () {
121.                 var validate = _formKey.currentState!.validate();
122.                 if (validate) {
123.                     if (!_isLoading) {
124.                         if (widget.produk != null) {
125.                             //kondisi update produk
126.                             ubah();
127.                         } else {
128.                             //kondisi tambah produk
129.                             simpan();
130.                         }
131.                     }
132.                 }
133.             });
134.     }
```

Dengan kode keseluruhan menjadi seperti berikut

```
1. import 'package:flutter/material.dart';
2. import 'package:tokokita/bloc/produk_bloc.dart';
3. import 'package:tokokita/model/produk.dart';
4. import 'package:tokokita/ui/produk_page.dart';
5. import 'package:tokokita/widget/warning_dialog.dart';
6.
7. class ProdukForm extends StatefulWidget {
8.   Produk? produk;
9.
10.  ProdukForm({Key? key, this.produk}) : super(key: key);
11.
12.  @override
13.  _ProdukFormState createState() => _ProdukFormState();
14. }
15.
16. class _ProdukFormState extends State<ProdukForm> {
17.   final _formKey = GlobalKey<FormState>();
18.   bool _isLoading = false;
19.   String judul = "TAMBAH PRODUK";
20.   String tombolSubmit = "SIMPAN";
21.
22.   final _kodeProdukTextboxController = TextEditingController();
23.   final _namaProdukTextboxController = TextEditingController();
24.   final _hargaProdukTextboxController = TextEditingController();
25.
26.   @override
27.   void initState() {
28.     super.initState();
29.     isUpdate();
30.   }
31.
32.   isUpdate() {
33.     if (widget.produk != null) {
34.       setState(() {
35.         judul = "UBAH PRODUK";
36.         tombolSubmit = "UBAH";
37.         _kodeProdukTextboxController.text = widget.produk!.kodeProduk!;
38.         _namaProdukTextboxController.text = widget.produk!.namaProduk!;
39.         _hargaProdukTextboxController.text =
40.           widget.produk!.hargaProduk.toString();
41.       });
42.     } else {
43.       judul = "TAMBAH PRODUK";
44.       tombolSubmit = "SIMPAN";
45.     }
46.   }
47.
48.   @override
49.   Widget build(BuildContext context) {
50.     return Scaffold(
51.       appBar: AppBar(title: Text(judul)),
52.       body: SingleChildScrollView(
53.         child: Padding(
54.           padding: const EdgeInsets.all(8.0),
55.           child: Form(
56.             key: _formKey,
57.             child: Column(
58.               children: [
59.                 _kodeProdukTextField(),
60.                 _namaProdukTextField(),
61.                 _hargaProdukTextField(),
62.                 _buttonSubmit()
```

```

63.         ],
64.     ),
65. ),
66. ),
67. ),
68. );
69. }
70.
71. //Membuat Textbox Kode Produk
72. Widget _kodeProdukTextField() {
73.     return TextFormField(
74.         decoration: const InputDecoration(labelText: "Kode Produk"),
75.         keyboardType: TextInputType.text,
76.         controller: _kodeProdukTextboxController,
77.         validator: (value) {
78.             if (value!.isEmpty) {
79.                 return "Kode Produk harus diisi";
80.             }
81.             return null;
82.         },
83.     );
84. }
85.
86. //Membuat Textbox Nama Produk
87. Widget _namaProdukTextField() {
88.     return TextFormField(
89.         decoration: const InputDecoration(labelText: "Nama Produk"),
90.         keyboardType: TextInputType.text,
91.         controller: _namaProdukTextboxController,
92.         validator: (value) {
93.             if (value!.isEmpty) {
94.                 return "Nama Produk harus diisi";
95.             }
96.             return null;
97.         },
98.     );
99. }
100.
101. //Membuat Textbox Harga Produk
102. Widget _hargaProdukTextField() {
103.     return TextFormField(
104.         decoration: const InputDecoration(labelText: "Harga"),
105.         keyboardType: TextInputType.number,
106.         controller: _hargaProdukTextboxController,
107.         validator: (value) {
108.             if (value!.isEmpty) {
109.                 return "Harga harus diisi";
110.             }
111.             return null;
112.         },
113.     );
114. }
115.
116. //Membuat Tombol Simpan/Ubah
117. Widget _buttonSubmit() {
118.     return OutlinedButton(
119.         child: Text(tombolSubmit),
120.         onPressed: () {
121.             var validate = _formKey.currentState!.validate();
122.             if (validate) {
123.                 if (!isLoading) {
124.                     if (widget.produk != null) {
125.                         //kondisi update produk
126.                         ubah();
127.                     } else {

```

```

128.             //kondisi tambah produk
129.             simpan();
130.         }
131.     }
132. }
133. });
134. }
135.
136.     simpan() {
137.         setState(() {
138.             _isLoading = true;
139.         });
140.         Produk createProduk = Produk(id: null);
141.         createProduk.kodeProduk = _kodeProdukTextboxController.text;
142.         createProduk.namaProduk = _namaProdukTextboxController.text;
143.         createProduk.hargaProduk = int.parse(_hargaProdukTextboxController.text);
144.         ProdukBloc.addProduk(produk: createProduk).then((value) {
145.             Navigator.of(context).push(MaterialPageRoute(
146.                 builder: (BuildContext context) => const ProdukPage()));
147.         }, onError: (error) {
148.             showDialog(
149.                 context: context,
150.                 builder: (BuildContext context) => const WarningDialog(
151.                     description: "Simpan gagal, silahkan coba lagi",
152.                 ));
153.         });
154.         setState(() {
155.             _isLoading = false;
156.         });
157.     }
158.
159.     ubah() {
160.         setState(() {
161.             _isLoading = true;
162.         });
163.         Produk updateProduk = Produk(id: null);
164.         updateProduk.id = widget.produk!.id;
165.         updateProduk.kodeProduk = _kodeProdukTextboxController.text;
166.         updateProduk.namaProduk = _namaProdukTextboxController.text;
167.         updateProduk.hargaProduk = int.parse(_hargaProdukTextboxController.text);
168.         ProdukBloc.updateProduk(produk: updateProduk).then((value) {
169.             Navigator.of(context).push(MaterialPageRoute(
170.                 builder: (BuildContext context) => const ProdukPage()));
171.         }, onError: (error) {
172.             showDialog(
173.                 context: context,
174.                 builder: (BuildContext context) => const WarningDialog(
175.                     description: "Permintaan ubah data gagal, silahkan coba lagi",
176.                 ));
177.         });
178.         setState(() {
179.             _isLoading = false;
180.         });
181.     }
182. }

```

*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

```

65. void confirmHapus() {
66.     AlertDialog alertDialog = AlertDialog(

```

```

67.     content: const Text("Yakin ingin menghapus data ini?"),
68.     actions: [
69.       //tombol hapus
70.       OutlinedButton(
71.         child: const Text("Ya"),
72.         onPressed: () {
73.           Navigator.push(
74.             context,
75.             MaterialPageRoute(
76.               builder: (context) => ProdukForm(
77.                 produk: widget.produk!,
78.               ));
79.         },
80.       ),
81.       //tombol batal
82.       OutlinedButton(
83.         child: const Text("Batal"),
84.         onPressed: () => Navigator.pop(context),
85.       )
86.     ],
87.   );
88.
89.   showDialog(builder: (context) => alertDialog, context: context);
90. }

```

Dengan kode keseluruhan menjadi

```

1. import 'package:flutter/material.dart';
2. import 'package:tokokita/model/produk.dart';
3. import 'package:tokokita/ui/produk_form.dart';
4.
5. class ProdukDetail extends StatefulWidget {
6.   Produk? produk;
7.
8.   ProdukDetail({Key? key, this.produk}) : super(key: key);
9.
10.  @override
11.  _ProdukDetailState createState() => _ProdukDetailState();
12. }
13.
14. class _ProdukDetailState extends State<ProdukDetail> {
15.  @override
16.  Widget build(BuildContext context) {
17.    return Scaffold(
18.      appBar: AppBar(
19.        title: const Text('Detail Produk'),
20.      ),
21.      body: Center(
22.        child: Column(
23.          children: [
24.            Text(
25.              "Kode : ${widget.produk!.kodeProduk}",
26.              style: const TextStyle(fontSize: 20.0),
27.            ),
28.            Text(
29.              "Nama : ${widget.produk!.namaProduk}",
30.              style: const TextStyle(fontSize: 18.0),
31.            ),
32.            Text(
33.              "Harga : Rp. ${widget.produk!.hargaProduk.toString()}",
34.              style: const TextStyle(fontSize: 18.0),
35.            ),
36.            _tombolHapusEdit()
37.          ],

```

```

38.     ),
39.   ),
40. );
41. }
42.
43. Widget _tombolHapusEdit() {
44.   return Row(
45.     mainAxisAlignment: MainAxisAlignment.min,
46.     children: [
47.       //Tombol Edit
48.       OutlinedButton(
49.         child: const Text("EDIT"),
50.         onPressed: () {
51.           Navigator.push(
52.             context,
53.             MaterialPageRoute(
54.               builder: (context) => ProdukForm(
55.                 produk: widget.produk!,
56.               )),
57.           ),
58.       //Tombol Hapus
59.       OutlinedButton(
60.         child: const Text("DELETE"), onPressed: () => confirmHapus(),
61.       ],
62.     );
63. }
64.
65. void confirmHapus() {
66.   AlertDialog alertDialog = AlertDialog(
67.     content: const Text("Yakin ingin menghapus data ini?"),
68.     actions: [
69.       //tombol hapus
70.       OutlinedButton(
71.         child: const Text("Ya"),
72.         onPressed: () {
73.           Navigator.push(
74.             context,
75.             MaterialPageRoute(
76.               builder: (context) => ProdukForm(
77.                 produk: widget.produk!,
78.               )),
79.           ),
80.       ),
81.       //tombol batal
82.       OutlinedButton(
83.         child: const Text("Batal"),
84.         onPressed: () => Navigator.pop(context),
85.       )
86.     ],
87.   );
88.
89.   showDialog(builder: (context) => alertDialog, context: context);
90. }
91. }

```