

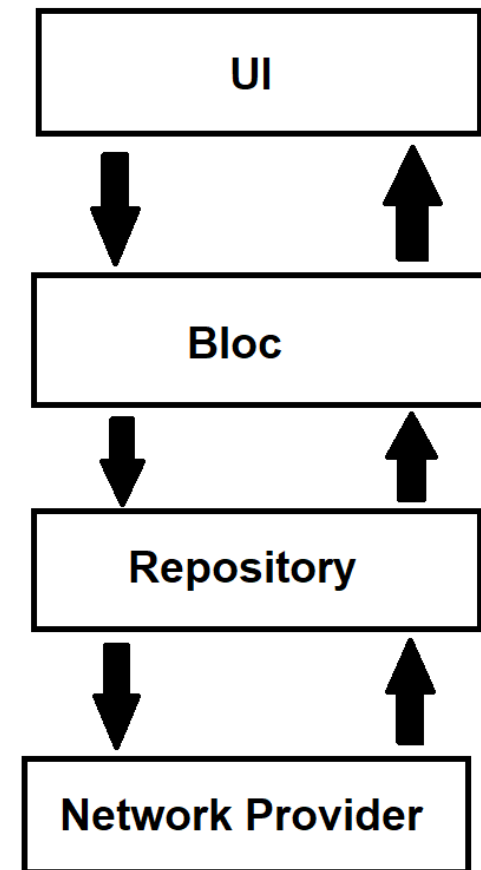


# PERTEMUAN 9

## PROJEK DENGAN FLUTTER

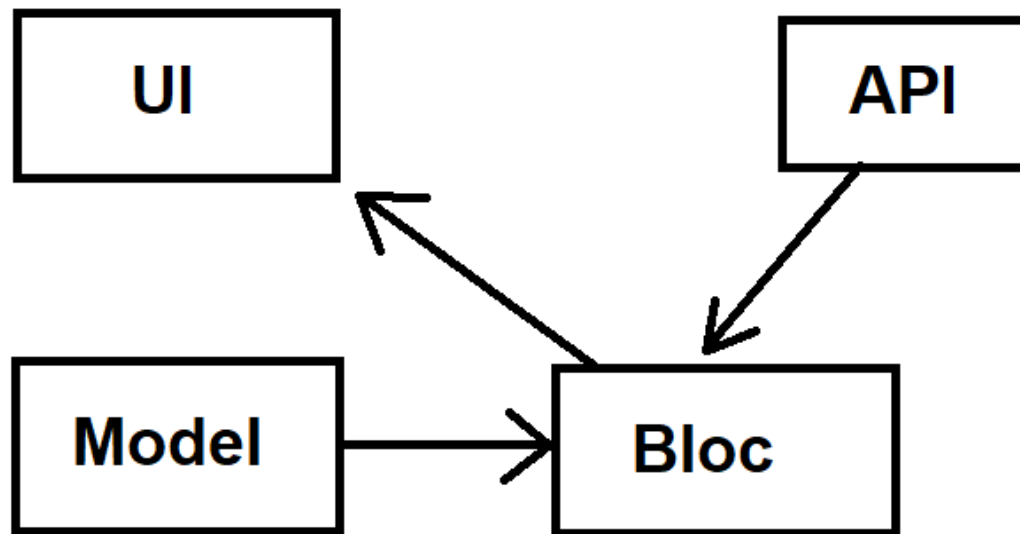
# 1. Membangun Proyek Dengan Flutter - 1

Dalam membangun proyek aplikasi mobile dengan flutter kali ini, kita menggunakan konsep kerja Design Pattern Bloc sederhana, yang ditampilkan pada gambar berikut ini.



# 1. Membangun Proyek Dengan Flutter - 2

Untuk proses mengambil data dapat digambarkan sebagai berikut.



Bloc akan mengambil data dari API kemudian menterjemahkan kedalam Objek Model yang kemudian ditampilkan ke UI

## 2. Pengembangan Proyek flutter tokokita - 1

Buat sebuah proyek flutter dengan nama **tokokita**

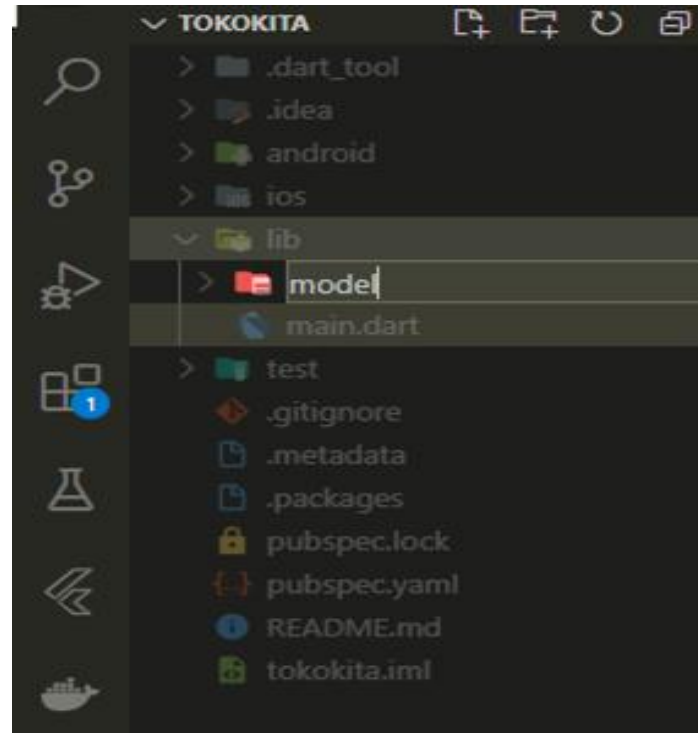
```
lib > main.dart > MyApp
1  import 'package:flutter/material.dart';
2
3  Run | Debug
4  void main() {
5    runApp(MyApp());
6  }
7
8  class MyApp extends StatelessWidget {
9    // This widget is the root of your application.
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(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

Ln 8, Col 50 Spaces: 2 UTF-8 CRLF Dart Flutter: 1.22.6 Nexus 5X API 29 (android-x86 emulator)

Your Flutter project is ready! Connect a device and press F5 to start running...

## a. Membuat Model

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



## b. Membuat Class Login Pada Model

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.  }
19. }
```



## c. Membuat Class Registrasi Pada Model

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.   }  
15. }
```

## d. Membuat Class Produk Pada Model

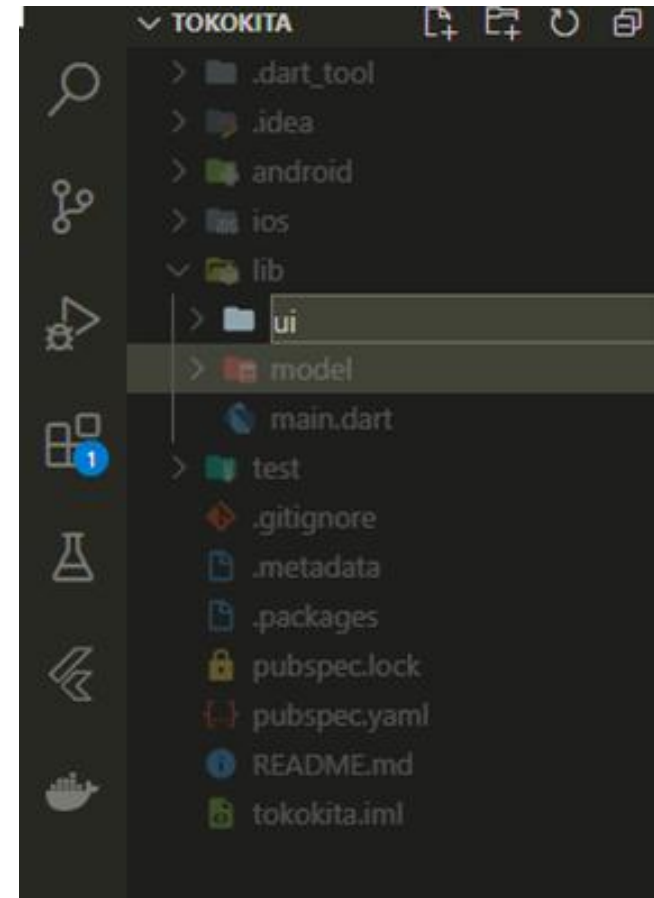
Buat sebuah file dengan nama **produk.dart** pada folder **model**.  
Kemudian ketikkan kode berikut

```
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.      id: obj['id'],
12.      kodeProduk: obj['kode_produk'],
13.      namaProduk: obj['nama_produk'],
14.      hargaProduk: obj['harga']
15.    );
16.  }
17. }
```



## 3. Membuat Halaman

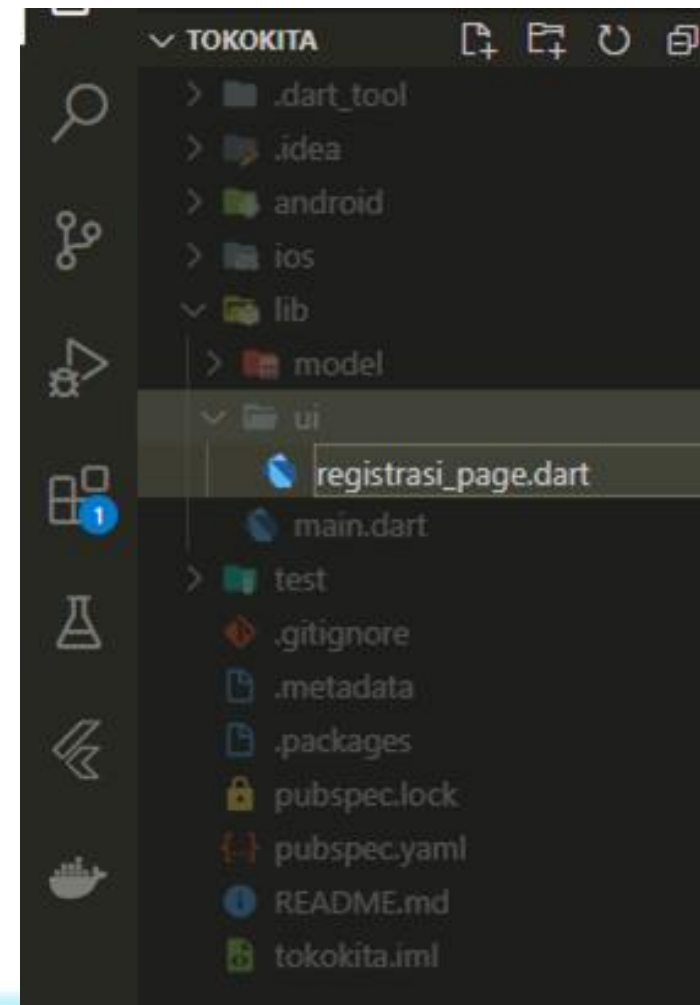
Pertama kita akan memecah bagian-bagian kode menjadi beberapa bagian, adapun untuk tampilan, dikelompokkan kedalam folder **ui**.





## a. Membuat Form Registrasi - 1

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



## a. Membuat Form Registrasi - 2

Pada file **registrasi\_page.dart** ketikkan kode berikut

```
1. import 'package:flutter/material.dart';
2.
3. class RegistrasiPage extends StatefulWidget {
4.   @override
5.   _RegistrasiPageState createState() => _RegistrasiPageState();
6. }
7.
8. class _RegistrasiPageState extends State<RegistrasiPage> {
9.   final _formKey = GlobalKey<FormState>();
10.  bool _isLoading = false;
11.
12.  final _namaTextboxController = TextEditingController();
13.  final _emailTextboxController = TextEditingController();
14.  final _passwordTextboxController = TextEditingController();
15.
```

## a. Membuat Form Registrasi - 3

```
---
16.  @override
17.  Widget build(BuildContext context) {
18.    return Scaffold(
19.      appBar: AppBar(title: Text("Registrasi"),),
20.      body: SingleChildScrollView(
21.        child: Container(
22.          child: Padding(
23.            padding: const EdgeInsets.all(8.0),
24.            child: Form(
25.              key: _formKey,
26.              child: Column(
27.                mainAxisAlignment: MainAxisAlignment.center,
28.                children: [
29.                  _namaTextField(),
30.                  _emailTextField(),
31.                  _passwordTextField(),
32.                  _passwordKonfirmasiTextField(),
33.                  _buttonRegistrasi()
34.                ],
35.              ),
36.            ),
37.          ),
38.        ),
39.      ),
40.    );
41.  }
42.
```

## a. Membuat Form Registrasi - 4

```
42.  
43.  //Membuat Textbox Nama  
44.  Widget _namaTextField() {  
45.    return TextFormField(  
46.      decoration: InputDecoration(labelText: "Nama"),  
47.      keyboardType: TextInputType.text,  
48.      controller: _namaTextboxController,  
49.      validator: (value){  
50.        if(value.length < 3){  
51.          return "Nama harus diisi minimal 3 karakter";  
52.        }  
53.        return null;  
54.      },  
55.    );  
56.  }  
57.
```

## a. Membuat Form Registrasi - 5

```
57.
58. //Membuat Textbox email
59. Widget _emailTextField() {
60.     return TextFormField(
61.         decoration: InputDecoration(labelText: "Email"),
62.         keyboardType: TextInputType.emailAddress,
63.         controller: _emailTextboxController,
64.         validator: (value){
65.             //validasi harus diisi
66.             if(value.isEmpty){
67.                 return 'Email harus diisi';
68.             }
69.             //validasi email
70.             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,}))$';
71.             RegExp regex = new RegExp(pattern);
72.             if (!regex.hasMatch(value)) {
73.                 return "Email tidak valid";
74.             }
75.             return null;
76.         },
77.     );
78. }
79.
```



## a. Membuat Form Registrasi - 6

```
78. }
79.
80. //Membuat Textbox password
81. Widget _passwordTextField() {
82.   return TextFormField(
83.     decoration: InputDecoration(labelText: "Password"),
84.     keyboardType: TextInputType.text,
85.     obscureText: true,
86.     controller: _passwordTextboxController,
87.     validator: (value){
88.       //jika karakter yang dimasukkan kurang dari 6 karakter
89.       if(value.length < 6){
90.         return "Password harus diisi minimal 6 karakter";
91.       }
92.       return null;
93.     },
94.   );
95. }
96.
```

## a. Membuat Form Registrasi - 7

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

## a. Membuat Form Registrasi - 8

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(MyApp());
6. }
7.
8. class MyApp extends StatefulWidget {
9.   @override
10.  _MyAppState createState() => _MyAppState();
11. }
12.
13. class _MyAppState extends State<MyApp> {
14.
15.   @override
16.   Widget build(BuildContext context) {
17.     return MaterialApp(
18.       title: 'Toko Kita',
19.       debugShowCheckedModeBanner: false,
20.       home: RegistrasiPage(),
21.     );
22.   }
23. }
```



## a. Membuat Form Registrasi - 9

Dan hasilnya seperti berikut

The screenshot shows a mobile application interface for registration. At the top, there is a blue header bar with the word "Registrasi" in white. Below the header, the form consists of four text input fields, each with a label above it: "Nama", "Email", "Password", and "Konfirmasi Password". The labels are in a light gray font. Below the "Konfirmasi Password" field, there is a gray button with the text "Registrasi" in black. The background of the app is white, and the status bar at the top shows the time as 2:18, LTE signal, and battery level.

## b. Membuat Form Login - 1

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.   @override
6.   _LoginPageState createState() => _LoginPageState();
7. }
8.
9. class _LoginPageState extends State<LoginPage> {
10.   final _formKey = GlobalKey<FormState>();
11.   bool _isLoading = false;
12.
13.   final _emailTextboxController = TextEditingController();
14.   final _passwordTextboxController = TextEditingController();
15.
16.   .. ..
```

## b. Membuat Form Login - 2

```
16.  @override
17.  Widget build(BuildContext context) {
18.    return Scaffold(
19.      appBar: AppBar(title: Text("Login")),
20.      body: SingleChildScrollView(
21.        child: Container(
22.          child: Padding(
23.            padding: const EdgeInsets.all(8.0),
24.            child: Form(
25.              key: _formKey,
26.              child: Column(
27.                children: [
28.                  _emailTextField(),
29.                  _passwordTextField(),
30.                  _buttonLogin(),
31.                  SizedBox(height: 30,),
32.                  _menuRegistrasi()
33.                ],
34.              ),
35.            ),
36.          ),
37.        ),
38.      ),
39.    );
40.  }
41.
```



## b. Membuat Form Login - 3

```
41.  
42.  //Membuat Textbox email  
43.  Widget _emailTextField() {  
44.    return TextFormField(  
45.      decoration: InputDecoration(labelText: "Email"),  
46.      keyboardType: TextInputType.emailAddress,  
47.      controller: _emailTextboxController,  
48.      validator: (value){  
49.        //validasi harus diisi  
50.        if(value.isEmpty){  
51.          return 'Email harus diisi';  
52.        }  
53.        return null;  
54.      },  
55.    );  
56.  }  
57.
```

## b. Membuat Form Login - 4

```
57.  
58.  //Membuat Textbox password  
59.  Widget _passwordTextField() {  
60.      return TextFormField(  
61.          decoration: InputDecoration(labelText: "Password"),  
62.          keyboardType: TextInputType.text,  
63.          obscureText: true,  
64.          controller: _passwordTextboxController,  
65.          validator: (value){  
66.              //validasi harus diisi  
67.              if(value.isEmpty){  
68.                  return "Password harus diisi";  
69.              }  
70.              return null;  
71.          },  
72.      );  
73.  }  
74.
```

## b. Membuat Form Login - 5

```
74.  
75.  //Membuat Tombol Login  
76.  Widget _buttonLogin() {  
77.    return RaisedButton(  
78.      child: Text("Login"),  
79.      onPressed: (){  
80.        var validate = _formKey.currentState.validate();  
81.      });  
82.  }  
83.  
84.  // Membuat menu untuk membuka halaman registrasi  
85.  Widget _menuRegistrasi() {  
86.    return Container(  
87.      child: Center(  
88.        child: InkWell(  
89.          child: Text("Registrasi",style: TextStyle(color: Colors.blue),),  
90.          onTap: () {  
91.            Navigator.push(context, new MaterialPageRoute(builder: (context)=> Regis  
trasiPage()));  
92.          },  
93.        ),  
94.      ),  
95.    );  
96.  }  
97. }
```

## b. Membuat Form Login - 6

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(MyApp());
6. }
7.
8. class MyApp extends StatefulWidget {
9.   @override
10.  _MyAppState createState() => _MyAppState();
11. }
12.
13. class _MyAppState extends State<MyApp> {
14.
15.   @override
16.   Widget build(BuildContext context) {
17.     return MaterialApp(
18.       title: 'Toko Kita',
19.       debugShowCheckedModeBanner: false,
20.       home: LoginPage(),
21.     );
22.   }
23. }
```



## b. Membuat Form Login - 7

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

2:40 LTE

Login

Email

Password

Login

[Registrasi](#)

## c. Membuat Form Produk - 1

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.   ProdukForm({this.produk});
7.   @override
8.   _ProdukFormState createState() => _ProdukFormState();
9. }
10.
11. class _ProdukFormState extends State<ProdukForm> {
12.   final _formKey = GlobalKey<FormState>();
13.   bool _isLoading = false;
14.   String judul = "TAMBAH PRODUK";
15.   String tombolSubmit = "SIMPAN";
16.
17.   final _kodeProdukTextboxController = TextEditingController();
18.   final _namaProdukTextboxController = TextEditingController();
19.   final _hargaProdukTextboxController = TextEditingController();
20. }
```



## c. Membuat Form Produk - 2

```
21. @override
22. void initState() {
23.     // TODO: implement initState
24.     super.initState();
25.     isUpdate();
26. }
27.
28. isUpdate(){
29.     if(widget.produk!=null){
30.         setState(() {
31.             judul = "UBAH PRODUK";
32.             tombolSubmit = "UBAH";
33.             _kodeProdukTextboxController.text = widget.produk.kodeProduk;
34.             _namaProdukTextboxController.text = widget.produk.namaProduk;
35.             _hargaProdukTextboxController.text = widget.produk.hargaProduk.toString();
36.         });
37.     }else{
38.         judul = "TAMBAH PRODUK";
39.         tombolSubmit = "SIMPAN";
40.     }
41. }
42.
```

## c. Membuat Form Produk - 3

```
42.  
43.  @override  
44.  Widget build(BuildContext context) {  
45.    return Scaffold(  
46.      appBar: AppBar(title: Text(judul)),  
47.      body: SingleChildScrollView(  
48.        child: Container(  
49.          child: Padding(  
50.            padding: const EdgeInsets.all(8.0),  
51.            child: Form(  
52.              key: _formKey,  
53.              child: Column(  
54.                children: [  
55.                  _kodeProdukTextField(),  
56.                  _namaProdukTextField(),  
57.                  _hargaProdukTextField(),  
58.                  _buttonSubmit()  
59.                ],  
60.              ),  
61.            ),  
62.          ),  
63.        ),  
64.      ),  
65.    );  
66.  }  
67.
```

## c. Membuat Form Produk - 4

```
67.  
68. //Membuat Textbox Kode Produk  
69. Widget _kodeProdukTextField() {  
70.   return TextFormField(  
71.     decoration: 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: 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.
```

## c. Membuat Form Produk - 5

```
97.  
98.  //Membuat Textbox Harga Produk  
99.  Widget _hargaProdukTextField() {  
100.      return TextFormField(  
101.          decoration: 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 RaisedButton(  
116.          child: Text(tombolSubmit),  
117.          onPressed: () {  
118.              var validate = _formKey.currentState.validate();  
119.          });  
120.  }  
121.  }
```

## d. Membuat Form Detail Produk - 1

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.   ProdukDetail({this.produk});
8.   @override
9.   _ProdukDetailState createState() => _ProdukDetailState();
10. }
11.
```

## d. Membuat Form Detail Produk - 2

```
11.
12. class _ProdukDetailState extends State<ProdukDetail> {
13.   @override
14.   Widget build(BuildContext context) {
15.     return Scaffold(
16.       appBar: AppBar(
17.         title: Text('Detail Produk'),
18.       ),
19.       body: Center(
20.         child: Column(
21.           children: [
22.             Text(
23.               "Kode : ${widget.produk.kodeProduk}",
24.               style: TextStyle(fontSize: 20.0),
25.             ),
26.             Text(
27.               "Nama : ${widget.produk.namaProduk}",
28.               style: TextStyle(fontSize: 18.0),
29.             ),
30.             Text(
31.               "Harga : Rp. ${widget.produk.hargaProduk.toString()}",
32.               style: TextStyle(fontSize: 18.0),
33.             ),
34.             _tombolHapusEdit()
35.           ],
36.         ),
37.       ),
38.     );
39.   }
40.
```



## d. Membuat Form Detail Produk - 3

```
40.
41. Widget _tombolHapusEdit() {
42.   return Row(
43.     mainAxisAlignment: MainAxisAlignment.min,
44.     children: [
45.       //Tombol Edit
46.       RaisedButton(
47.         child: Text("EDIT"), color: Colors.green, onPressed: () {
48.           Navigator.push(
49.             context,
50.             new MaterialPageRoute(
51.               builder: (context) => ProdukForm(produk: widget.produk,));
52.         }),
53.       //Tombol Hapus
54.       RaisedButton(
55.         child: Text("DELETE"), color: Colors.red, onPressed: ()=>confirmHapus())
56.     ],
57.   );
58. }
59.
```

## d. Membuat Form Detail Produk - 4

```
60. void confirmHapus() {  
61.     AlertDialog alertDialog = new AlertDialog(  
62.         content: Text("Yakin ingin menghapus data ini?"),  
63.         actions: [  
64.             //tombol hapus  
65.             RaisedButton(  
66.                 child: Text("Ya"),  
67.                 color: Colors.green,  
68.                 onPressed: (){},  
69.             ),  
70.             //tombol batal  
71.             RaisedButton(  
72.                 child: Text("Batal"),  
73.                 color: Colors.red,  
74.                 onPressed: ()=>Navigator.pop(context),  
75.             )  
76.         ],  
77.     );  
78.  
79.     showDialog(context: context, child: alertDialog);  
80. }  
81. }
```

## e. Membuat Tampilan List Produk - 1

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

```
1. import 'package:flutter/cupertino.dart';
2. import 'package:flutter/material.dart';
3. import 'package:tokokita/model/produk.dart';
4. import 'package:tokokita/ui/produk_detail.dart';
5. import 'package:tokokita/ui/produk_form.dart';
6.
7. class ProdukPage extends StatefulWidget {
8.   @override
9.   _ProdukPageState createState() => _ProdukPageState();
10. }
11.
```

## e. Membuat Tampilan List Produk - 2

```
11.  
12. class _ProdukPageState extends State<ProdukPage> {  
13.   @override  
14.   Widget build(BuildContext context) {  
15.     return Scaffold(  
16.       appBar: AppBar(  
17.         title: Text('List Produk'),  
18.         actions: [  
19.           Padding(  
20.             padding: EdgeInsets.only(right: 20.0),  
21.             child: GestureDetector(  
22.               child: Icon(Icons.add, size: 26.0),  
23.               onTap: () async {  
24.                 Navigator.push(  
25.                   context,  
26.                   new MaterialPageRoute(  
27.                     builder: (context) => ProdukForm()));  
28.               },  
29.             )),  
30.         ],  
31.       ),  
--
```

## e. Membuat Tampilan List Produk - 3

```
32.     drawer: Drawer(  
33.         child: ListView(  
34.             children: [  
35.                 ListTile(  
36.                     title: Text('Logout'),  
37.                     trailing: Icon(Icons.logout),  
38.                     onTap: () async {},  
39.                 )  
40.             ],  
41.         ),  
42.     ),  
43.     body: ListView(  
44.         children: [  
45.             ItemProduk(produk: Produk(id: 1, kodeProduk: 'A001', namaProduk: 'Kamera',  
             hargaProduk: 5000000)),  
46.             ItemProduk(produk: Produk(id: 2, kodeProduk: 'A002', namaProduk: 'Kulkas',  
             hargaProduk: 2500000)),  
47.             ItemProduk(produk: Produk(id: 3, kodeProduk: 'A003', namaProduk: 'Mesin Cu  
ci', hargaProduk: 2000000)),  
48.         ],  
49.     ),  
50. );  
51. }  
52. }
```

## e. Membuat Tampilan List Produk - 4

```
53.  
54. class ItemProduk extends StatelessWidget {  
55.   final Produk produk;  
56.  
57.   ItemProduk({this.produk});  
58.  
59.   @override  
60.   Widget build(BuildContext context) {  
61.     return Container(  
62.       child: GestureDetector(  
63.         onTap: () {  
64.           Navigator.push(  
65.             context,  
66.             new MaterialPageRoute(  
67.               builder: (context) => ProdukDetail(produk: produk,));  
68.           },  
69.           child: Card(  
70.             child: ListTile(  
71.               title: Text(produk.namaProduk),  
72.               subtitle: Text(produk.hargaProduk.toString()),  
73.             ),  
74.           ),  
75.         ),  
76.       );  
77.     }  
78. }
```

## e. Membuat Tampilan List Produk - 5

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

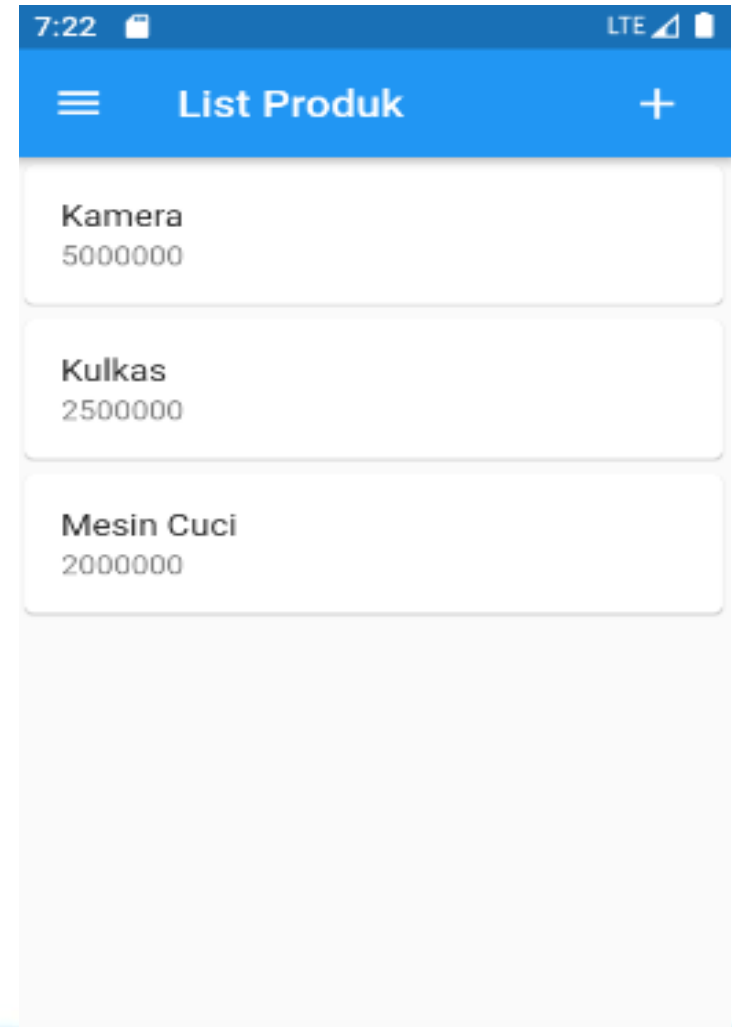
```
1. import 'package:flutter/material.dart';
2. import 'package:tokokita/ui/produk_page.dart';
3.
4. void main() {
5.   runApp(MyApp());
6. }
7.
8. class MyApp extends StatefulWidget {
9.   @override
10.  _MyAppState createState() => _MyAppState();
11. }
12.
13. class _MyAppState extends State<MyApp> {
14.
15.   @override
16.  Widget build(BuildContext context) {
17.    return MaterialApp(
18.      title: 'Toko Kita',
19.      debugShowCheckedModeBanner: false,
20.      home: ProdukPage(),
21.    );
22.  }
23. }
```





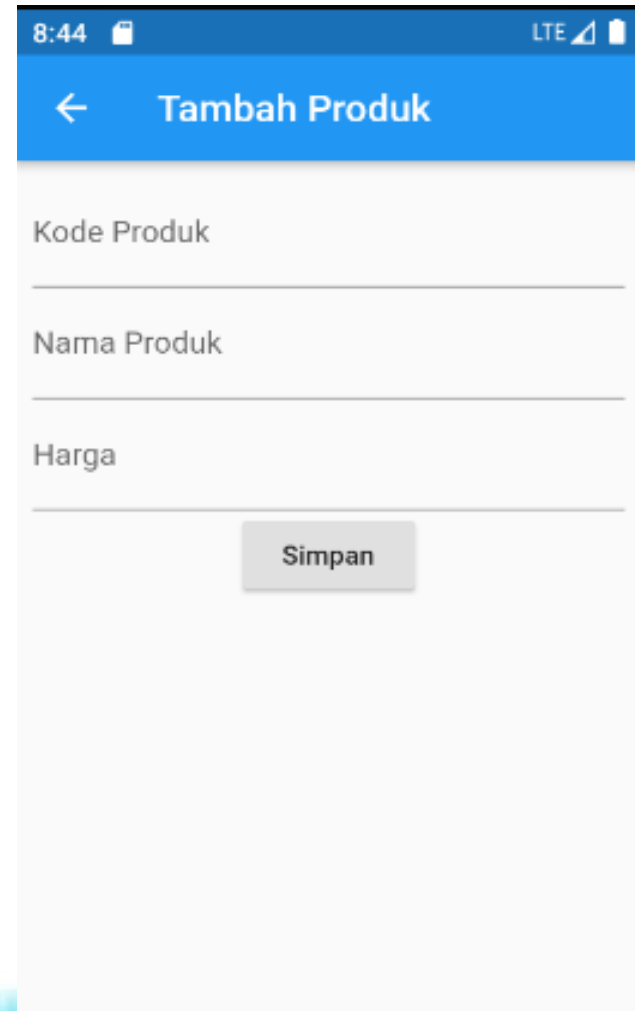
## e. Membuat Tampilan List Produk - 6

Maka tampilannya akan menjadi seperti berikut



## e. Membuat Tampilan List Produk - 7

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



8:44 LTE

← Tambah Produk

Kode Produk

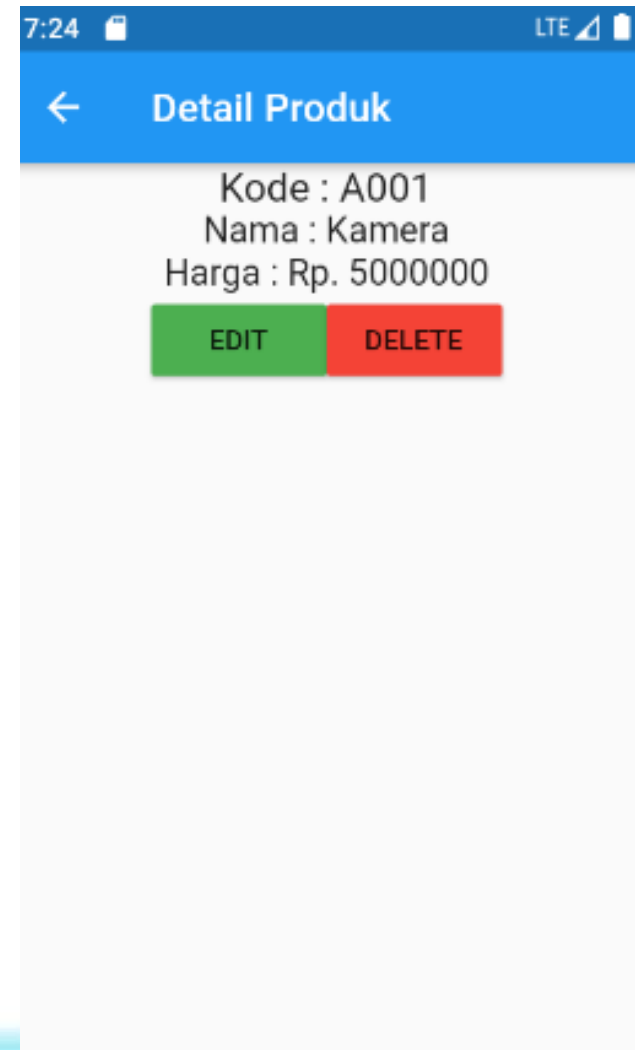
Nama Produk

Harga

Simpan

## e. Membuat Tampilan List Produk - 8

Jika **salah satu data produk** diklik  
maka akan muncul detail produk



## e. Membuat Tampilan List Produk - 9

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.



7:26 LTE

← UBAH PRODUK

Kode Produk  
A001

Nama Produk  
Kamera

Harga  
5000000

UBAH