PERTEMUAN 12

MODIFIKASI APLIKASI

1. Modifikasi produk_page.dart

a. Menambahkan fungsi logout pada drawer

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

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


b. Menampilkan Data Produk dari Rest API - 1

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

Adapun perubahan yang dilakukan adalah penambahan sebuah class bernama **ListProduk**, bagian **body** dan memasukkan beberapa package.

b. Menampilkan Data Produk dari Rest API - 2

Pyang dilakukan adalah penambahan sebuah class bernama **ListProduk** dengan kode

```
61. class ListProduk extends StatelessWidget {
62.   final List list;
63.   ListProduk({this.list});
64.
65.   @override
66.   Widget build(BuildContext context) {
67.     return ListView.builder(
68.         itemCount: list==null ? 0:list.length,
69.         itemBuilder: (context, i){
70.           return ItemProduk(produk: list[i],);
71.         });
72.   }
73. }
```


b. Menampilkan Data Produk dari Rest API - 3

Kemudian perubahan pada bagian **body** menjadi

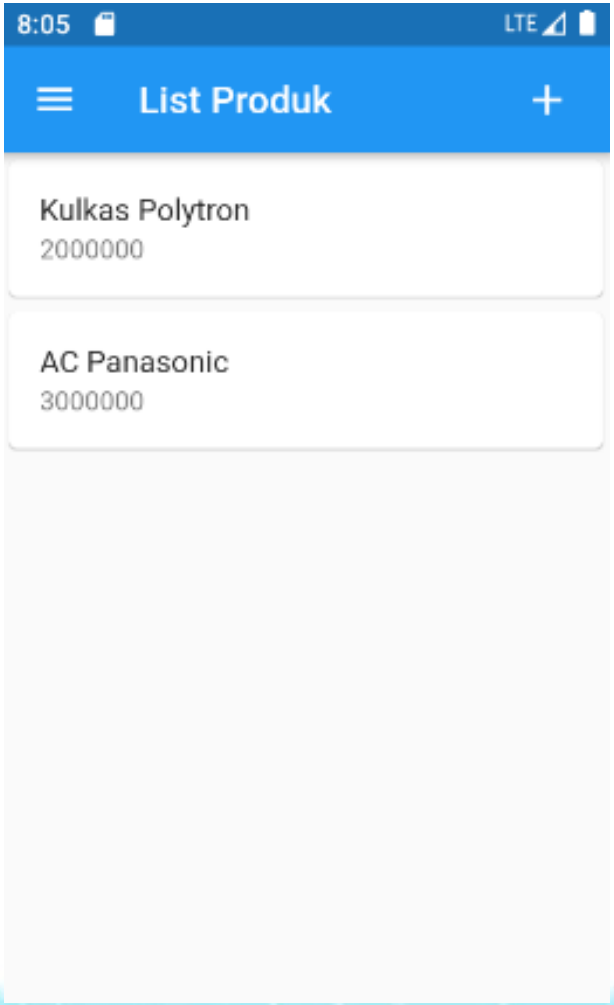
```
50. body: FutureBuilder<List>(
51.         future: ProdukBloc.getProduks(),
52.         builder: (context, snapshot){
53.           if(snapshot.hasError) print(snapshot.error);
54.           return snapshot.hasData ? ListProduk(list: snapshot.data,) : Center(child:
    CircularProgressIndicator(),);
55.         },
56.       ),
```

b. Menampilkan Data Produk dari Rest API - 4

Serta memasukkan beberapa package yang diperlukan

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

b. Menampilkan Data Produk dari Rest API - 4



2. Memodifikasi Form Produk (produk_form.dart)

a. Membuat fungsi simpan - 1

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**

a. Membuat fungsi simpan - 2

```
116.  //Membuat Tombol Simpan/Ubah
117.  Widget _buttonSubmit() {
118.    return RaisedButton(
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.
```

a. Membuat fungsi simpan - 3

```
136.  simpan() {
137.    setState(() {
138.      _isLoading = true;
139.    });
140.    Produk createProduk = new Produk();
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(new MaterialPageRoute(builder: (BuildContext context)
    => ProdukPage()));
146.    },onError: (error){
147.      showDialog(
148.          context: context,
149.          builder: (BuildContext context) => WarningDialog(
150.            description: "Simpan gagal, silahkan coba lagi",
151.          )
152.      );
153.    });
154.    setState(() {
155.      _isLoading = false;
156.    });
157.  }
```

b. Membuat fungsi ubah - 1

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

b. Membuat fungsi ubah - 2

Dengan fungsi ubah sebagai berikut

```
159. ubah() {
160.     setState(() {
161.       _isLoading = true;
162.     });
163.     Produk updateProduk = new Produk();
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(new MaterialPageRoute(builder: (BuildContext context)
    => ProdukPage()));
170.     },onError: (error){
171.       showDialog(
172.           context: context,
173.           builder: (BuildContext context) => WarningDialog(
174.             description: "Permintaan ubah data gagal, silahkan coba lagi",
175.           )
176.       );
177.     });
178.     setState(() {
179.       _isLoading = false;
180.     });
181.   }
```


b. Membuat fungsi ubah - 3

Kemudian kita tambahkan pada fungsi **_buttonSubmit**

```
116.   //Membuat Tombol Simpan/Ubah
117.     Widget _buttonSubmit() {
118.       return RaisedButton(
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.     }
```


c. Menambahkan fungsi hapus pada Detail Produk (produk_detail.dart) - 1

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

```
63.   void confirmHapus() {
64.     AlertDialog alertDialog = new AlertDialog(
65.       content: Text("Yakin ingin menghapus data ini?"),
66.       actions: [
67.         //tombol hapus
68.         RaisedButton(
69.           child: Text("Ya"),
70.           color: Colors.green,
71.           onPressed: (){
72.             ProdukBloc.deleteProduk(id: widget.produk.id).then((value){
73.               Navigator.of(context).push(new MaterialPageRoute(builder: (BuildContex
    t context) => ProdukPage()));
74.             },onError: (error){
```


c. Menambahkan fungsi hapus pada Detail Produk (produk_detail.dart) - 2

```
75.                 showDialog(
76.                     context: context,
77.                     builder: (BuildContext context) => WarningDialog(
78.                       description: "Hapus data gagal, silahkan coba lagi",
79.                     )
80.                 );
81.               });
82.           },
83.         ),
84.         //tombol batal
85.         RaisedButton(
86.           child: Text("Batal"),
87.           color: Colors.red,
88.           onPressed: ()=>Navigator.pop(context),
89.         )
90.       ],
91.     );
92.
93.     showDialog(context: context, child: alertDialog);
94.   }
```