

**LAPORAN PROGRESS
APLIKASI BERBASIS PLATFORM
DIGIBRARY VERSI FLUTTER**



**Telkom
University**

**KELAS: IF-43-10
KELOMPOK: 6**

ANGGOTA KELOMPOK:

Bima Mahardika W.	(1301194304)
Fiyona Annilia Syamsir	(1301194201)
M. Aldi Haryojudanto	(1301194025)
Moch. Hanafi Sony S.	(1301194139)

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY**

TAHUN AKADEMIK 2021/2022

Latar Belakang

Perpustakaan memiliki peran yang sangat penting dalam pendidikan. Seiring dengan perkembangan teknologi, perpustakaan harus mudah untuk diakses oleh siapapun kapanpun dan dimanapun. Maka untuk mengelola dan mempermudah semua aset bacaan yang berkepentingan dalam suatu kampus atau lembaga pendidikan, aplikasi berbasis web Digibrary dibuat untuk memenuhi kebutuhan tersebut.

Digibrary adalah aplikasi peminjaman dan pengembalian koleksi Open Library Telkom University secara *online*. Fitur yang ada dalam aplikasi tersebut menyediakan koleksi berupa buku, jurnal, dokumen, *e-journal*, *e-book* dengan audio, dan koleksi Open Library Telkom University lainnya yang tentunya dapat dipinjam. Aplikasi tersebut digunakan untuk mempermudah dalam mengelola dari koleksi yang ada perpustakaan serta untuk meningkatkan sistem yang sudah ada sebelumnya (Open Library).

Tujuan

Perangkat lunak Digibrary yang dikembangkan akan digunakan untuk mengelola peminjaman dan pengembalian buku dengan beberapa fitur tambahan pada perpustakaan Open Library di Telkom University. Perangkat lunak ini harus mampu melayani autentikasi pengguna, peminjaman buku, pembayaran denda, dan pengembalian buku.

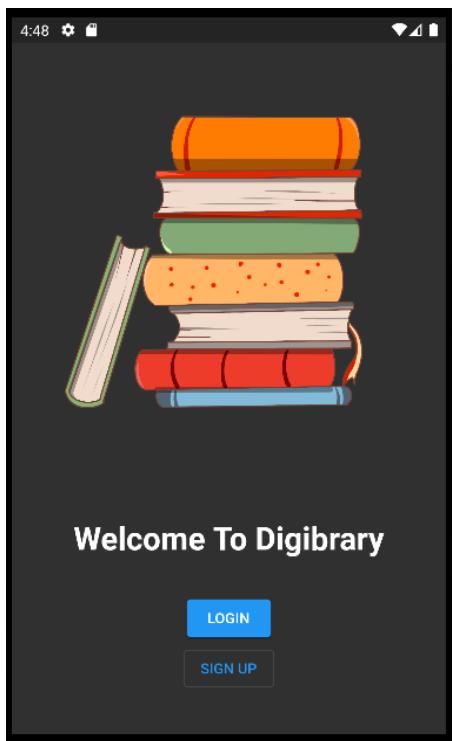
Implementasi

- Bab 7: Checked Widget

Untuk progress checked Widget pada aplikasi kami digibrary versi mobile terdapat beberapa widget yang bisa diakses diantaranya : Home, Halaman Login, Halaman SIGN UP, Halaman Home awal aplikasi , Halaman Dashboard, Halaman Profile, Halaman logout.

- Welcome (Halaman awal sebelum login)

Tampilan Halaman Welcome



Source code halaman Welcome

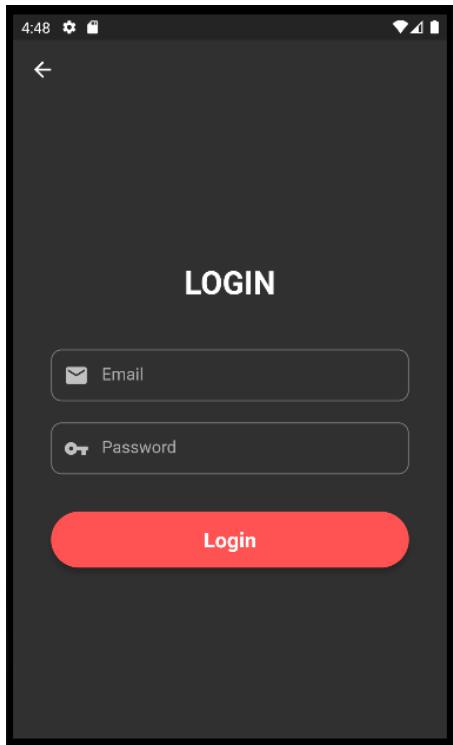
```
lms > lib > screens > Welcome > welcome_screen.dart > WelcomeScreen > build
1 import 'package:flutter/material.dart';
2 import 'package:lms/utils/constants.dart';
3
4 class WelcomeScreen extends StatelessWidget {
5   const WelcomeScreen({Key? key}) : super(key: key);
6
7   @override
8   Widget build(BuildContext context) {
9     final size = MediaQuery.of(context).size;
10    return Scaffold(
11      body: SingleChildScrollView(
12        child: Column(
13          mainAxisAlignment: MainAxisAlignment.center,
14          crossAxisAlignment: CrossAxisAlignment.center,
15          children: <Widget>[
16            SizedBox(height: size.height * 0.05),
17            Image.asset('assets/images/books_image.png'),
18            SizedBox(height: size.height * 0.05),
19            const Text(
20              "Welcome To Digibrary",
21              style: TextStyle(fontWeight: FontWeight.bold, fontSize: 30),
22              textAlign: TextAlign.center,
23            ), // Text
24            SizedBox(height: size.height * 0.05),
25            ElevatedButton(
26              onPressed: () {
27                Navigator.pushNamed(context, loginRoute);
28              },
29              child: const Text(" LOGIN ")), // ElevatedButton
30            OutlinedButton(

```

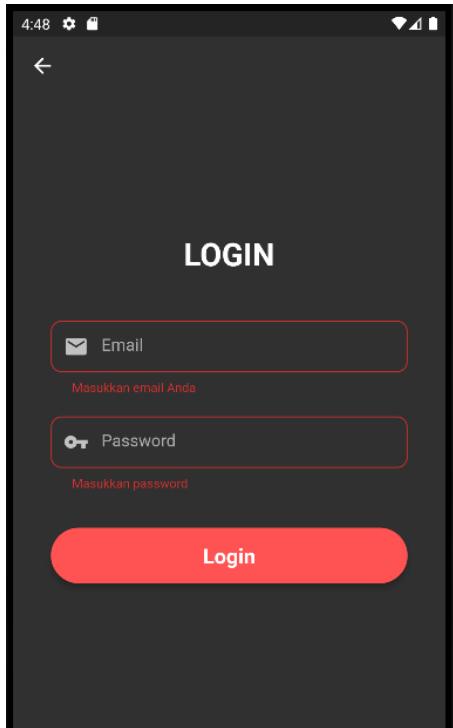
```
lms > lib > screens > Welcome > welcome_screen.dart > WelcomeScreen > build
31          onPressed: () {
32            Navigator.pushNamed(context, signUpRoute);
33          },
34          child: const Text("SIGN UP")) // OutlinedButton
35        ], // <Widget>[] // Column
36      ), // SingleChildScrollView
37    ); // Scaffold
38  }
39 }
40
```

- Login

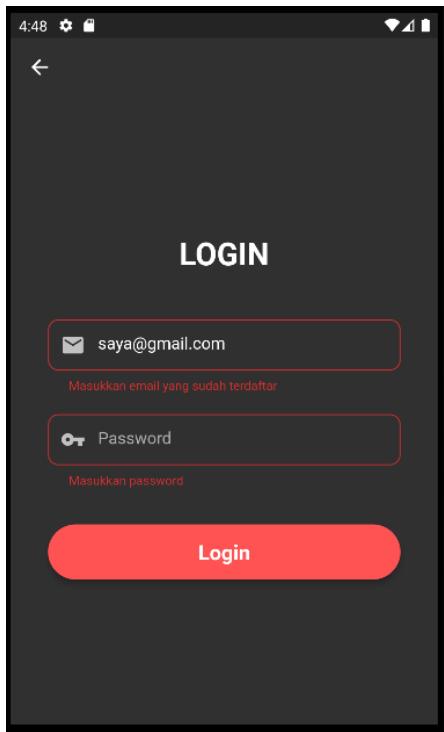
Tampilan halaman Login



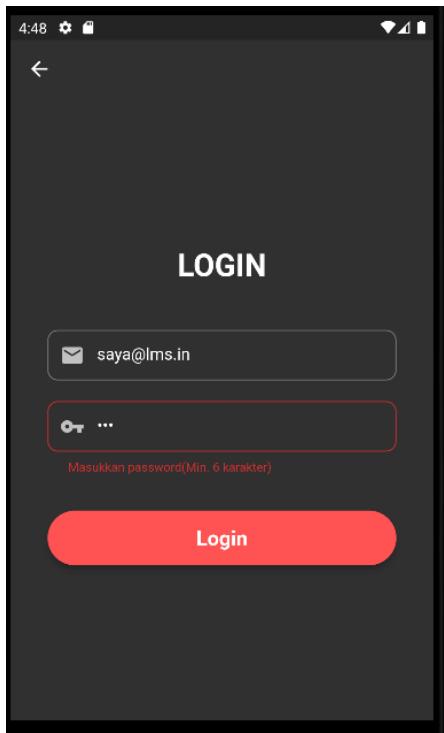
Tampilan halaman Login (Saat tekan login dengan input kosong)



Tampilan halaman Login (Saat email tidak terdaftar)



Tampilan halaman Login (Saat password kurang dari 6 karakter)



Source code halaman Login

```
login_screen.dart M ...
lms > lib > screens > Login > login_screen.dart > _LoginScreenState > login
1   import 'package:firebase_auth/firebase_auth.dart';
2   import 'package:flutter/material.dart';
3   import 'package:lms/main.dart';
4   import 'package:lms/utils/constants.dart';
5   import 'package:lms/services/authentication_service.dart';
6   import 'package:lms/utils/helper.dart';
7   import 'package:lms/utils/user_preferences.dart';
8   // ignore: implementation_imports
9   import 'package:provider/src/provider.dart';
10
11  class LoginScreen extends StatefulWidget {
12    const LoginScreen({Key? key}) : super(key: key);
13
14    @override
15    State<LoginScreen> createState() => _LoginScreenState();
16  }
17
18  class _LoginScreenState extends State<LoginScreen> {
19    //form key
20    final _formKey = GlobalKey<FormState>();
21
22    //progress bar state
23    bool _isLoading = false;
24
25    //editing controller
26    final TextEditingController emailController = TextEditingController();
27    final passwordController = TextEditingController();
28
29    @override
30    Widget build(BuildContext context) {
```

```
login_screen.dart M ...
lms > lib > screens > Login > login_screen.dart > _LoginScreenState > login
31    //email field
32    final emailField = TextFormField(
33      autofocus: false,
34      keyboardType: TextInputType.emailAddress,
35      controller: emailController,
36      onSaved: (value) {
37        emailController.text = value!;
38      },
39     textInputAction: TextInputAction.next,
40      decoration: InputDecoration(
41        prefixIcon: const Icon(Icons.mail),
42        contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
43        hintText: 'Email',
44        border: OutlineInputBorder(
45          borderRadius: BorderRadius.circular(10),
46        ), // OutlineInputBorder // InputDecoration
47        validator: (value) {
48          if (value!.isEmpty) {
49            return "Masukkan email Anda";
50          }
51          if (!RegExp("^[a-zA-Z0-9.!#%&*+/=?^`{|}~-]+@[lms.in]"
52            .hasMatch(value)) {
53            return "Masukkan email yang sudah terdaftar";
54          }
55          return null;
56        },
57      ); // TextFormField
58
59      //password field
60      final passwordField = TextFormField(
```

```
login_screen.dart M X
lms > lib > screens > Login > login_screen.dart > _LoginScreenState > login
61   autofocus: false,
62   obscureText: true,
63   controller: passwordController,
64   onSaved: (value) {
65     passwordController.text = value!;
66   },
67   validator: (value) {
68     if (value!.isEmpty) {
69       return "Masukkan password";
70     }
71     RegExp regex = RegExp(r"^(?=.*[a-zA-Z])(?=.*[0-9]).{6,}$");
72     if (!regex.hasMatch(value)) {
73       return "Masukkan password(Min. 6 karakter)";
74     }
75     return null;
76   },
77   textInputAction: TextInputAction.done,
78   decoration: InputDecoration(
79     prefixIcon: const Icon(Icons.vpn_key),
80     contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
81     hintText: 'Password',
82     border: OutlineInputBorder(
83       borderRadius: BorderRadius.circular(10),
84     )), // OutlineInputBorder // InputDecoration
85 ); // TextFormField
86
87   return (_isLoading)
88     ? const Center(
89       child: CircularProgressIndicator(),
90     ) // Center

```

```
login_screen.dart M X
lms > lib > screens > Login > login_screen.dart > _LoginScreenState > login
91   : Scaffold(
92     appBar: AppBar(
93       backgroundColor: Colors.transparent,
94       elevation: 0,
95       leading: IconButton(
96         onPressed: () {
97           Navigator.of(context).pop();
98         },
99         icon: const Icon(Icons.arrow_back), // IconButton
100     ), // AppBar
101     body: Center(
102       child: SingleChildScrollView(
103         child: Padding(
104           padding: const EdgeInsets.all(36.0),
105           child: Form(
106             key: _formKey,
107             child: Column(
108               mainAxisAlignment: MainAxisAlignment.center,
109               crossAxisAlignment: CrossAxisAlignment.center,
110               children: <Widget>[
111                 sizedBoxMargin(10),
112                 const Text(
113                   "LOGIN",
114                   textAlign: TextAlign.center,
115                   style: TextStyle(
116                     fontWeight: FontWeight.bold,
117                     fontSize: 30,
118                   ), // TextStyle
119                 ), // Text
120                 sizedBoxMargin(45),

```

```
login_screen.dart M X
lms > lib > screens > Login > login_screen.dart > _LoginScreenState > logIn
121     emailField,
122     sizedBoxMargin(20),
123     passwordField,
124     sizedBoxMargin(35),
125     customButton(() {
126         logIn(emailController.text, passwordController.text);
127     }, "Login", context, redButtonColor),
128     sizedBoxMargin(15),
129     ], // <Widget>[]]
130     ), // Column // Form
131     ), // Padding
132     ), // SingleChildScrollView // Center
133 ); // Scaffold
134 }
135
136 logIn(String email, String password) async {
137     if (_formKey.currentState!.validate()) {
138         setState(() {
139             _isLoading = true;
140         });
141         User? user;
142         await context
143             .read<AuthenticationService>()
144             .signIn(
145                 email: email.trim(),
146                 password: password.trim(),
147             )
148             .then((value) async {
149                 user = FirebaseAuth.instance.currentUser;
150                 if (user != null) {

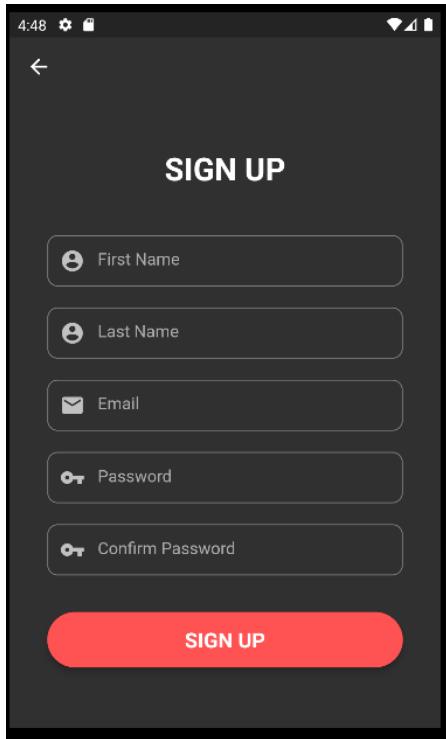
```

```
login_screen.dart M X
lms > lib > screens > Login > login_screen.dart > _LoginScreenState > logIn
151         await UserPreferences.saveUserPreferences(user!);
152         admin = isAdmin();
153         Navigator.pushNamedAndRemoveUntil(
154             context, bottomNavPanelRoute, (route) => false);
155     }
156     showSnackbar(value, context);
157 };
158     setState(() {
159         _isLoading = false;
160     });
161 }
162 }
163 }
164 }
```

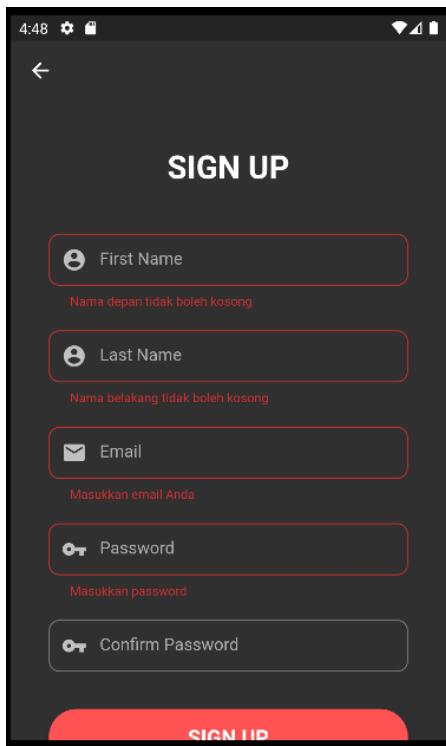
Berdasarkan implementasi di atas, halaman login memiliki beberapa widget di antaranya, TextFormField yang berisi untuk TextInputAction email dan password yang diberi icon serta sebuah button Login untuk submit input user agar dapat masuk ke dalam akun.

- Sign up

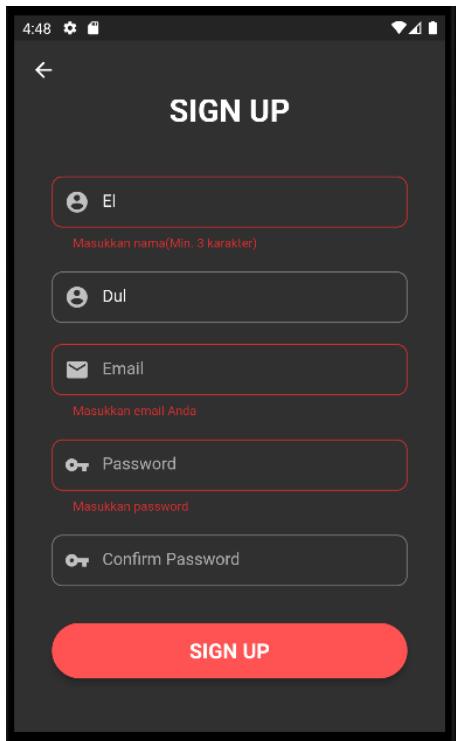
Tampilan halaman Sign up



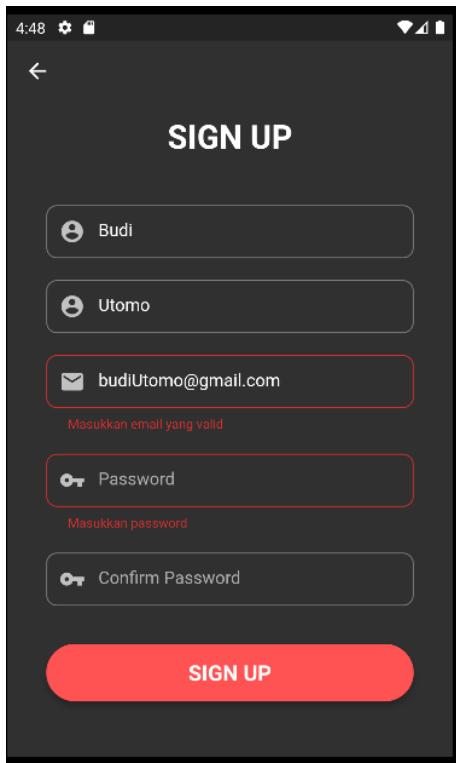
Tampilan halaman Sign up (Saat tekan submit dengan input kosong)



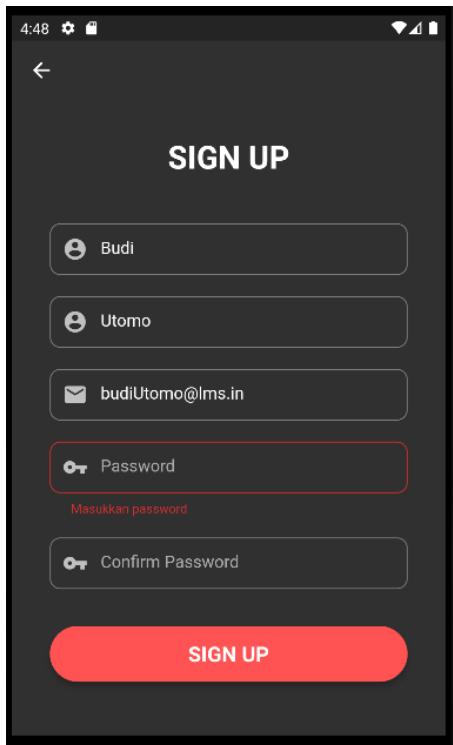
Tampilan halaman Sign up (Saat nama kurang dari 3 karakter)



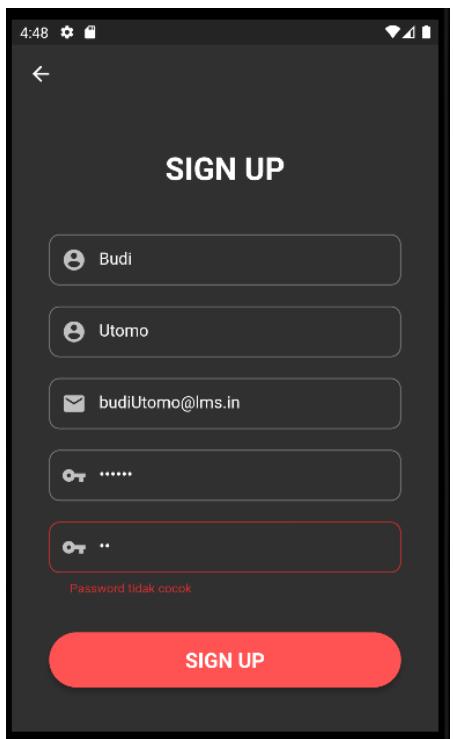
Tampilan halaman Sign up (Saat email tidak valid)



Tampilan halaman Sign up (Saat password kurang dari 6 karakter)



Tampilan halaman Sign up (saat password konfirmasi tidak sama)



Source code halaman Sign up

```
❶ signup_page.dart M
Ims > lib > screens > SignUp > ❷ signup_page.dart > _SignUpScreenState > ❸ build
  1 import 'package:cloud_firestore/cloud_firestore.dart';
  2 import 'package:firebase_auth/firebase_auth.dart';
  3 import 'package:flutter/material.dart';
  4 import 'package:lms/utils/constants.dart';
  5 import 'package:lms/model/user_model.dart';
  6 import 'package:lms/services/authentication_service.dart';
  7 import 'package:lms/utils/helper.dart';
  8 // ignore: implementation_imports
  9 import 'package:provider/src/provider.dart';
 10
 11 import '../../../../../main.dart';
 12 import '../../../../../utils/user_preferences.dart';
 13
 14 class SignUpScreen extends StatefulWidget {
 15   const SignUpScreen({Key? key}) : super(key: key);
 16
 17   @override
 18   State<SignUpScreen> createState() => _SignUpScreenState();
 19 }
 20
 21 class _SignUpScreenState extends State<SignUpScreen> {
 22   var _isLoading = false;
 23   final _auth = FirebaseAuth.instance;
 24
 25   final _formKey = GlobalKey<FormState>();
 26
 27   final firstNameController = TextEditingController();
 28
 29   final lastNameController = TextEditingController();
 30 }
```

```
❶ signup_page.dart M
Ims > lib > screens > SignUp > ❷ signup_page.dart > _SignUpScreenState > ❸ build
  31   final emailController = TextEditingController();
  32
  33   final passwordController = TextEditingController();
  34
  35   final confirmPasswordController = TextEditingController();
  36
  37   @override
  38   Widget build(BuildContext context) {
  39     //first name field
  40     final firstNameField = TextFormField(
  41       autofocus: false,
  42       controller: firstNameController,
  43       keyboardType: TextInputType.name,
  44       onSaved: (value) {
  45         firstNameController.text = value!;
  46       },
  47       validator: (value) {
  48         RegExp regex = RegExp(r"^.{3,}$");
  49         if (value!.isEmpty) {
  50           return "Nama depan tidak boleh kosong";
  51         }
  52
  53         if (!regex.hasMatch(value)) {
  54           return "Masukkan nama(Min. 3 karakter)";
  55         }
  56         return null;
  57       },
  58      textInputAction: TextInputAction.next,
  59       decoration: InputDecoration(
  60         prefixIcon: const Icon(Icons.account_circle),
  61       ),
  62     );
  63
  64     final passwordField = TextFormField(
  65       controller: passwordController,
  66       obscureText: true,
  67       onSaved: (value) {
  68         passwordController.text = value!;
  69       },
  70       validator: (value) {
  71         if (value!.isEmpty) {
  72           return "Masukkan kata sandi";
  73         }
  74         if (value.length < 6) {
  75           return "Kata sandi minimal 6 karakter";
  76         }
  77         return null;
  78       },
  79      textInputAction: TextInputAction.next,
  80       decoration: InputDecoration(
  81         prefixIcon: const Icon(Icons.vpn_key),
  82       ),
  83     );
  84
  85     final confirmPasswordField = TextFormField(
  86       controller: confirmPasswordController,
  87       obscureText: true,
  88       onSaved: (value) {
  89         confirmPasswordController.text = value!;
  90       },
  91       validator: (value) {
  92         if (value!.isEmpty) {
  93           return "Masukkan ulang kata sandi";
  94         }
  95         if (value != passwordController.text) {
  96           return "Kata sandi tidak sama";
  97         }
  98         return null;
  99       },
 100      textInputAction: TextInputAction.done,
 101      decoration: InputDecoration(
 102        prefixIcon: const Icon(Icons.vpn_key),
 103      ),
 104    );
 105
 106    return Scaffold(
 107      appBar: AppBar(
 108        title: Text("Sign Up"),
 109      ),
 110      body: Padding(
 111        padding: EdgeInsets.all(16),
 112        child: Form(
 113          key: _formKey,
 114          child: Column(
 115            children: [
 116              firstNameField,
 117              SizedBox(height: 16),
 118              passwordField,
 119              SizedBox(height: 16),
 120              confirmPasswordField,
 121              SizedBox(height: 16),
 122              Row(
 123                mainAxisAlignment: MainAxisAlignment.end,
 124                children: [
 125                  ElevatedButton(
 126                    onPressed: () {
 127                      if (_formKey.currentState!.validate()) {
 128                        _auth.createUserWithEmailAndPassword(
 129                          email: emailController.text,
 130                          password: passwordController.text
 131                        ).then((value) {
 132                          Navigator.pushNamed(context, '/home');
 133                        });
 134                      setState(() {
 135                        _isLoading = true;
 136                      });
 137                    }
 138                  ),
 139                  SizedBox(width: 16),
 140                  Text("Cancel")
 141                ],
 142              ),
 143            ],
 144          ),
 145        ),
 146      ),
 147    );
 148  }
```

```
❶ signup_page.dart M X
Ims > lib > screens > SignUp > ❷ _signUpScreenState > ⚭ build
  61   contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
  62   hintText: 'First Name',
  63   border: OutlineInputBorder(
  64     borderRadius: BorderRadius.circular(10),
  65   )), // OutlineInputBorder // InputDecoration
  66 ); // TextFormField
  67
  68 //last name field
  69 final lastNameField = TextFormField(
  70   autofocus: false,
  71   controller: lastNameController,
  72   keyboardType: TextInputType.name,
  73   onSaved: (value) {
  74     lastNameController.text = value!;
  75   },
  76   validator: (value) {
  77     if (value!.isEmpty) {
  78       return "Nama belakang tidak boleh kosong";
  79     }
  80     return null;
  81   },
  82  textInputAction: TextInputAction.next,
  83   decoration: InputDecoration(
  84     prefixIcon: const Icon(Icons.account_circle),
  85     contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
  86     hintText: 'Last Name',
  87     border: OutlineInputBorder(
  88       borderRadius: BorderRadius.circular(10),
  89     )), // OutlineInputBorder // InputDecoration
  90 ); // TextFormField
```

```
❶ signup_page.dart M X
Ims > lib > screens > SignUp > ❷ _signUpScreenState > ⚭ build
  91 //email field
  92 final emailField = TextFormField(
  93   autofocus: false,
  94   controller: emailController,
  95   keyboardType: TextInputType.emailAddress,
  96   onSaved: (value) {
  97     emailController.text = value!;
  98   },
  99   validator: (value) {
 100     if (value!.isEmpty) {
 101       return "Masukkan email Anda";
 102     }
 103     if (!RegExp("^[a-zA-Z0-9.!#%&*+=?^`{|}~-]+@[lms.in]")
 104         .hasMatch(value)) {
 105       return "Masukkan email yang valid";
 106     }
 107     return null;
 108   },
 109  textInputAction: TextInputAction.next,
 110   decoration: InputDecoration(
 111     prefixIcon: const Icon(Icons.mail),
 112     contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
 113     hintText: 'Email',
 114     border: OutlineInputBorder(
 115       borderRadius: BorderRadius.circular(10),
 116     )), // OutlineInputBorder // InputDecoration
 117 ); // TextFormField
 118
 119 //password field
```

```
❶ signup_page.dart M
Ims > lib > screens > SignUp > signup_page.dart > _SignUpScreenState > build
121   final passwordField = TextFormField(
122     controller: passwordController,
123     autofocus: false,
124     keyboardType: TextInputType.visiblePassword,
125     obscureText: true,
126     onSaved: (value) {
127       passwordController.text = value!;
128     },
129     validator: (value) {
130       if (value!.isEmpty) {
131         return "Masukkan password";
132       }
133       RegExp regex = RegExp(r"^(?=.*[a-zA-Z])(?=.*[0-9]).{6,}$");
134       if (!regex.hasMatch(value)) {
135         return "Masukkan password(Min. 6 karakter)";
136       }
137       return null;
138     },
139    textInputAction: TextInputAction.next,
140     decoration: InputDecoration(
141       prefixIcon: const Icon(Icons.vpn_key),
142       contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
143       hintText: 'Password',
144       border: OutlineInputBorder(
145         borderRadius: BorderRadius.circular(10),
146       ), // OutlineInputBorder // InputDecoration
147     ); // TextFormField
148
149   //confirm password field
150   final confirmPasswordField = TextFormField(
```

```
❷ signup_page.dart M
Ims > lib > screens > SignUp > signup_page.dart > _SignUpScreenState > build
151   controller: confirmPasswordController,
152   autofocus: false,
153   obscureText: true,
154   validator: (value) {
155     if (value!.length < 6 && passwordController.text != value) {
156       return "Password tidak cocok";
157     }
158     return null;
159   },
160   onSaved: (value) {
161     confirmPasswordController.text = value!;
162   },
163   textInputAction: TextInputAction.done,
164   decoration: InputDecoration(
165     prefixIcon: const Icon(Icons.vpn_key),
166     contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
167     hintText: 'Confirm Password',
168     border: OutlineInputBorder(
169       borderRadius: BorderRadius.circular(10),
170     ), // OutlineInputBorder // InputDecoration
171   ); // TextFormField
172
173   return (_isLoading)
174     ? const Center(
175       child: CircularProgressIndicator(),
176     ) // Center
177     : Scaffold(
178       appBar: AppBar(
179         backgroundColor: Colors.transparent,
180         elevation: 0,
```

```
lms > lib > screens > SignUp > signup_page.dart > _SignUpScreenState > build
181         leading: IconButton(
182             onPressed: () {
183                 Navigator.of(context).pop();
184             },
185             icon: const Icon(Icons.arrow_back), // IconButton
186         ), // AppBar
187         body: Center(
188             child: SingleChildScrollView(
189                 child: Padding(
190                     padding: const EdgeInsets.all(36.0),
191                     child: Form(
192                         key: _formKey,
193                         child: Column(
194                             mainAxisAlignment: MainAxisAlignment.center,
195                             crossAxisAlignment: CrossAxisAlignment.center,
196                             children: <Widget>[
197                                 sizedBoxMargin(10),
198                                 const Text(
199                                     "SIGN UP",
200                                     textAlign: TextAlign.center,
201                                     style: TextStyle(
202                                         fontWeight: FontWeight.bold,
203                                         fontSize: 30,
204                                     ), // TextStyle
205                                 ), // Text
206                                 sizedBoxMargin(45),
207                                 firstNameField,
208                                 sizedBoxMargin(20),
209                                 lastNameField,
210                                 sizedBoxMargin(20).
```

The screenshot shows a code editor with the file `signup_page.dart` open. The current view is the `build` method of the `_SignUpScreenState` class. The code uses the `Form`, `SingleChildScrollView`, and `Scaffold` widgets from the `flutter` framework. It also imports `Authenticationservice` and `User` from `model`. The `signUp` function is defined as an asynchronous method that takes `String email` and `String password` as parameters. It performs validation, sets a loading state, and then calls the `read<AuthenticationService>()` method to perform the sign-up operation.

```
211     emailField,
212     sizedBoxMargin(20),
213     passwordField,
214     sizedBoxMargin(20),
215     confirmPasswordField,
216     sizedBoxMargin(35),
217     customButton(() {
218         | signUp(emailController.text, passwordController.text);
219     }, "SIGN UP", context, redButtonColor),
220     sizedBoxMargin(15),
221 ], // <Widget>[]
222 ), // Column
223 ), // Form
224 ), // Padding
225 ), // SingleChildScrollView
226 ), // Center
227 ); // Scaffold
228 }
229
230 signUp(String email, String password) async {
231     if (_formKey.currentState!.validate()) {
232         setState(() {
233             _isLoading = true;
234         });
235         User? user;
236         await context
237             .read<AuthenticationService>()
238             .signUp(
239                 email: email,
240                 password:
```

```
signup_page.dart M X
lms > lib > screens > SignUp > signup_page.dart > _SignUpScreenState > build
241     )
242     .then((value) async {
243         await postUserDetailsToFirestore(context);
244         user = FirebaseAuth.instance.currentUser;
245         if (user != null) {
246             await UserPreferences.saveUserPreferences(user!);
247             admin = isAdmin();
248             Navigator.pushNamedAndRemoveUntil(
249                 context, homeRoute, (route) => false);
250         }
251         showSnackbar(value, context);
252     });
253     setState(() {
254         _isLoading = false;
255     });
256 }
257 }

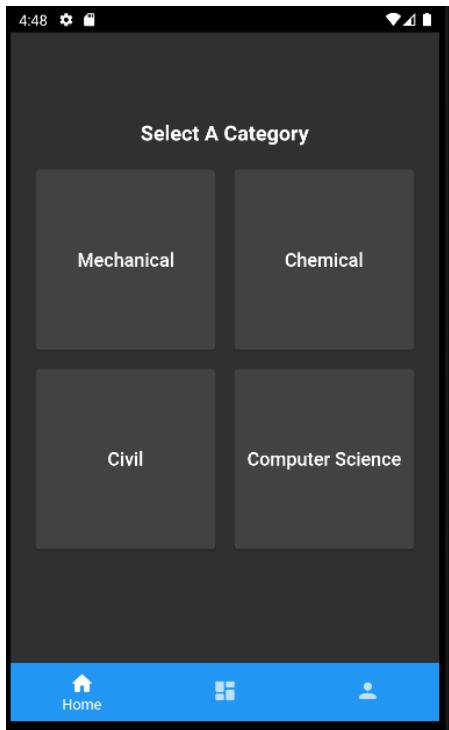
259 postUserDetailsToFirestore(BuildContext context) async {
260     FirebaseFirestore firebaseFirestore = FirebaseFirestore.instance;
261     User? user = _auth.currentUser;
262
263     UserModel userModel = UserModel(
264         email: user!.email,
265         uid: user.uid,
266         firstName: firstNameController.text,
267         lastName: lastNameController.text,
268         userRole: "user");
269
270     await firebaseFirestore
```

```
signup_page.dart M X
lms > lib > screens > SignUp > signup_page.dart > _SignUpScreenState > build
271     .collection("users")
272     .doc(user.uid)
273     .set(userModel.toMap());
274
275     Navigator.pushNamedAndRemoveUntil(
276         context, bottomNavPanelRoute, (route) => false);
277     }
278 }
279 }
```

Berdasarkan implementasi di atas, halaman SIGN UP memiliki beberapa widget di antaranya, Text Form Field yang berisi untuk TextInput Action First Name, Last Name, Email, Password, dan Confirm password yang diberi icon serta sebuah button Sign Up untuk submit input user agar dapat registrasi kedalam aplikasi serta dapat menyimpan data pribadi kedalam database.

- Home

Tampilan Halaman Home



Source code Halaman Home

```
category_screen.dart X
Ims > lib > screens > Home > category_screen.dart > CategoryScreen
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:flutter/material.dart';
3 import 'package:lms/model/category_model.dart';
4 import 'package:lms/repository/data_repository.dart';
5 import 'package:lms/utils/constants.dart';
6
7 class CategoryScreen extends StatefulWidget {
8   const CategoryScreen({Key? key}) : super(key: key);
9
10  @override
11  State<CategoryScreen> createState() => _CategoryScreenState();
12}
13
14 class _CategoryScreenState extends State {
15  @override
16  void initState() {
17    super.initState();
18  }
19
20  @override
21  Widget build(BuildContext context) {
22    Category? category;
23    return Column(
24      mainAxisAlignment: MainAxisAlignment.center,
25      mainAxisSize: MainAxisSize.center,
26      children: [
27        SizedBox(
28          child: const Center(
29            child: Text(
30              "Select A Category",
31            style: TextStyle(
32              fontSize: 20,
33              fontWeight: FontWeight.bold,
34            ), // TextStyle
35            ), // Text
36            ), // Center
37            width: MediaQuery.of(context).size.width,
38        ), // SizedBox
39        StreamBuilder(
40          stream: DataRepository().getCategoryStream(),
41          builder: (context, AsyncSnapshot<QuerySnapshot> snapshot) {
42            if (snapshot.hasError) {
43              return const Text('Something went wrong');
44            }
45            if (!snapshot.hasData) {
46              return const Center(
47                child: Text("No Books Available"),
48              ); // Center
49            }
50
51            if (snapshot.connectionState == ConnectionState.waiting) {
52              return const Center(
53                child: CircularProgressIndicator(),
54              ); // Center
55            }
56            final data = snapshot.requireData;
57
58            return GridView.builder(
59              padding: const EdgeInsets.all(20),
60              scrollDirection: Axis.vertical,
```

```
category_screen.dart X
Ims > lib > screens > Home > category_screen.dart > CategoryScreen
31           style: TextStyle(
32             fontSize: 20,
33             fontWeight: FontWeight.bold,
34           ), // TextStyle
35           ), // Text
36           ), // Center
37           width: MediaQuery.of(context).size.width,
38       ), // SizedBox
39       StreamBuilder(
40         stream: DataRepository().getCategoryStream(),
41         builder: (context, AsyncSnapshot<QuerySnapshot> snapshot) {
42           if (snapshot.hasError) {
43             return const Text('Something went wrong');
44           }
45           if (!snapshot.hasData) {
46             return const Center(
47               child: Text("No Books Available"),
48             ); // Center
49           }
50
51           if (snapshot.connectionState == ConnectionState.waiting) {
52             return const Center(
53               child: CircularProgressIndicator(),
54             ); // Center
55           }
56           final data = snapshot.requireData;
57
58           return GridView.builder(
59             padding: const EdgeInsets.all(20),
60             scrollDirection: Axis.vertical,
```

```
category_screen.dart ×
Ims > lib > screens > Home > category_screen.dart > CategoryScreen
61     shrinkWrap: true,
62     gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
63         crossAxisCount: 2,
64         crossAxisSpacing: 10,
65         mainAxisSpacing: 10,
66     ), // SliverGridDelegateWithFixedCrossAxisCount
67     itemCount: data.size,
68     itemBuilder: (context, index) {
69         category = Category.fromMap(data.docs[index]);
70         return CategoryCard(category!);
71     },
72 ); // GridView.builder
73 ),
74 ), // StreamBuilder
75 ],
76 ); // Column
77 }
78 }
79
80 class CategoryCard extends StatefulWidget {
81     final Category category;
82     const CategoryCard(this.category, {Key? key}) : super(key: key);
83
84     @override
85     State<CategoryCard> createState() => _CategoryCardState();
86 }
87
88 class _CategoryCardState extends State<CategoryCard> {
89     @override
90     Widget build(BuildContext context) {

```

```
category_screen.dart ×
Ims > lib > screens > Home > category_screen.dart > CategoryScreen
91     return GestureDetector(
92         onTap: () {
93             Navigator.pushNamed(
94                 context,
95                 booksListRoute,
96                 arguments: widget.category,
97             );
98         },
99         child: Card(
100             child: Padding(
101                 padding: const EdgeInsets.all(8),
102                 child: Center(
103                     child: Text(
104                         widget.category.title!,
105                         style: const TextStyle(
106                             fontWeight: FontWeight.w600,
107                             fontSize: 18,
108                         ), // TextStyle
109                     ), // Text
110                 ), // Center
111             ), // Padding
112         ), // Card
113     ); // GestureDetector
114 }
115 }
```

Berdasarkan implementasi di atas, halaman Home memiliki beberapa widget di antaranya, Button Select A Category yang berisi beberapa button tentang jenis buku dimana apabila di klik oleh user akan langsung muncul list buku yang sesuai dengan jenis buku yang di klik oleh user.

- Dashboard

Tampilan Halaman Dashboard



Untuk halaman dashboard masih ada beberapa code yang error sehingga untuk progress kali ini halaman dashboard masih belum bisa untuk ditampilkan isi nya.

Source code Halaman Dashboard

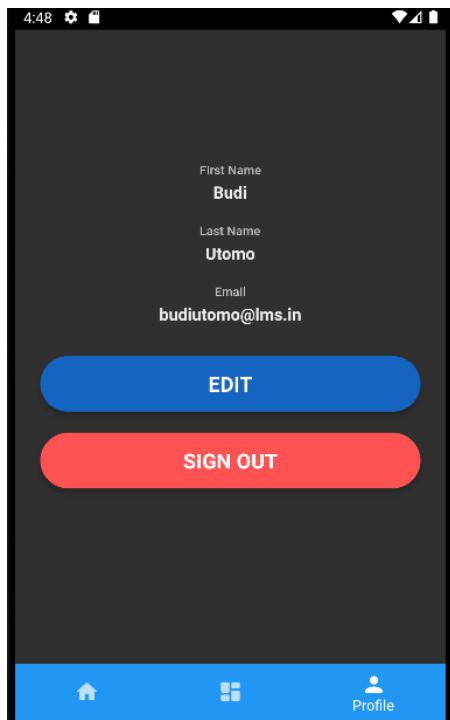
```
dashboard_screen.dart X
lms > lib > screens > Dashboard > dashboard_screen.dart > DashboardScreen > tabs
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:flutter/material.dart';
3 import 'package:lms/main.dart';
4 import 'package:lms/utils/user_preferences.dart';
5 import '../../../../../model/book_model.dart';
6 import '../../../../../repository/data_repository.dart';
7 import '../../../../../utils/constants.dart';
8 import '../Home/components/book_card.dart';
9 import 'components/request_card.dart';
10
11 class DashboardScreen extends StatelessWidget {
12   DashboardScreen({Key? key}) : super(key: key);
13
14   final tabs = [
15     const Tab(text: 'Sent Requests', icon: Icon(Icons.send_rounded)),
16     const Tab(text: 'Inventory', icon: Icon(Icons.inventory_2_rounded)),
17   ];
18
19   @override
20   Widget build(BuildContext context) {
21     sentRequestTab(bool isInventory) => StreamBuilder(
22       stream: DataRepository().getBooksStream(),
23       builder: ((context, AsyncSnapshot<QuerySnapshot> snapshot) {
24         if (snapshot.hasError) {
25           return const Text('Something went wrong');
26         }
27         if (snapshot.connectionState == ConnectionState.waiting) {
28           return const CircularProgressIndicator();
29         }
30         final data = snapshot.requireData;
31       ),
32     );
33   }
34
35   @override
36   void dispose() {
37     DataRepository().cancelStreams();
38     super.dispose();
39   }
40
41   @override
42   void initState() {
43     super.initState();
44     DataRepository().init();
45   }
46
47   @override
48   Widget build(BuildContext context) {
49     return Scaffold(
50       appBar: AppBar(
51         title: const Text('Dashboard'),
52         centerTitle: true,
53       ),
54       body: Padding(
55         padding: const EdgeInsets.all(16),
56         child: Column(
57           children: [
58             Container(
59               padding: const EdgeInsets.all(16),
60               decoration: BoxDecoration(
61                 color: Colors.white,
62                 border: Border.all(
63                   color: Colors.grey.shade300,
64                   width: 1,
65                 ),
66               ),
67               child: Row(
68                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
69                 children: [
70                   Text('Sent Requests'),
71                   Text('Inventory')
72                 ],
73               ),
74             ),
75             Container(
76               padding: const EdgeInsets.all(16),
77               decoration: BoxDecoration(
78                 color: Colors.white,
79                 border: Border.all(
80                   color: Colors.grey.shade300,
81                   width: 1,
82                 ),
83               ),
84               child: TabBar(
85                 tabs: tabs,
86               ),
87             ),
88             Container(
89               padding: const EdgeInsets.all(16),
90               decoration: BoxDecoration(
91                 color: Colors.white,
92                 border: Border.all(
93                   color: Colors.grey.shade300,
94                   width: 1,
95                 ),
96               ),
97               child: TabBarView(
98                 children: [
99                   sentRequestTab(true),
100                  sentRequestTab(false)
101                ],
102              ),
103            )
104          ],
105        ),
106      ),
107    );
108  }
109}
```

```
dashboard_screen.dart X
lms > lib > screens > Dashboard > dashboard_screen.dart > DashboardScreen > tabs
31
32   return ListView.builder(
33     physics: const BouncingScrollPhysics(
34       parent: AlwaysScrollableScrollPhysics(),
35     ), // BouncingScrollPhysics
36     padding: const EdgeInsets.all(8),
37     itemCount: data.size,
38     itemBuilder: ((context, index) {
39       final book = Book.fromMap(data.docs[index].data());
40       return GestureDetector(
41         onTap: () {
42           Navigator.pushNamed(
43             context,
44             bookDetailRoute,
45             arguments: book.uid,
46           );
47         },
48       );
49     },
50     child: (admin)
51       ? (book.requestedBy?.uid != null)
52         ? RequestedCard(book)
53         : Container()
54       : (isInventory)
55         ? (book.issuedTo?.uid == UserPreferences.getUserUid())
56           ? BookCard(book)
57           : Container()
58         : (book.requestedBy?.uid ==
59           UserPreferences.getUserUid())
60           ? BookCard(book)
61           : Container(),
62     );
63   ); // GestureDetector
64 }
```

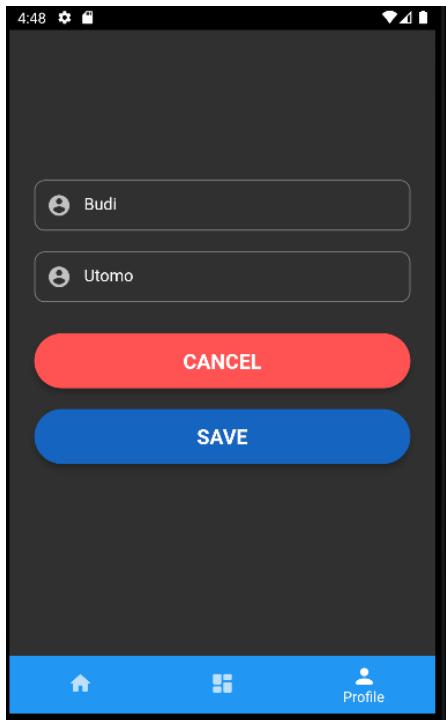
```
dashboard_screen.dart
lms > lib > screens > Dashboard > dashboard_screen.dart > DashboardScreen > tabs
61     },
62     );
63   ),
64   );
65 }
66 return DefaultTabController(
67   length: tabs.length,
68   child: Scaffold(
69     appBar: (admin)
70       ? null
71       : AppBar(
72         toolbarHeight: 0,
73         elevation: 0,
74         backgroundColor: Colors.blue,
75         bottom: TabBar(
76           tabs: tabs,
77         ), // TabBar
78       ), // AppBar
79     body: (admin)
80       ? sentRequestTab(false)
81       : TabBarView(
82         children: [sentRequestTab(false), sentRequestTab(true)],
83       ), // TabBarView
84   ); // Scaffold // DefaultTabController
85 }
86 }
87 }
```

○ Profile

Tampilan Halaman Profile



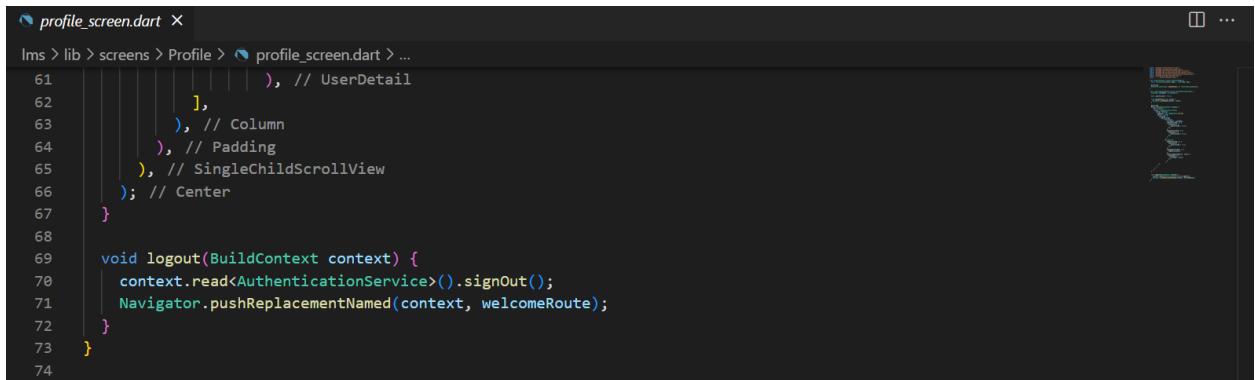
Tampilan Halaman Edit Profile



Source code halaman Profile

```
profile_screen.dart X
lms > lib > screens > Profile > profile_screen.dart > ...
1 import 'package:flutter/material.dart';
2 import 'package:lms/model/user_model.dart';
3 import 'package:lms/screens/Profile/edit_form.dart';
4 import 'package:lms/screens/Profile/user_details.dart';
5 import 'package:lms/services/authentication_service.dart';
6 import 'package:provider/provider.dart';
7 import '.../../utils/constants.dart';
8
9 class ProfileScreen extends StatefulWidget {
10   const ProfileScreen({Key? key}) : super(key: key);
11
12   @override
13   State<ProfileScreen> createState() => _ProfileScreenState();
14 }
15
16 class _ProfileScreenState extends State<ProfileScreen> {
17   UserModel userModel = UserModel();
18
19   bool _editClicked = false;
20
21   void navigateTo(String route) {
22     Navigator.pushNamed(context, route);
23   }
24
25   @override
26   Widget build(BuildContext context) {
27     return Center(
28       child: SingleChildScrollView(
29         child: Padding(
30           padding: const EdgeInsets.all(24),
31
32           child: Column(
33             children: [
34               _editClicked
35                 ? EditForm(
36                   userModel: userModel,
37                   onSaveClicked: () {
38                     setState(() {
39                       _editClicked = false;
40                     });
41                   },
42                   onCancelClicked: () {
43                     setState(() {
44                       _editClicked = false;
45                     });
46                 },
47                 ) // EditForm
48                 : UserDetail(
49                   onEditClicked: () {
50                     setState(() {
51                       _editClicked = true;
52                     });
53                   },
54                   onSignOutClicked: () {
55                     logout(context);
56                   },
57                   onGettingUserDetails: (user) {
58                     setState(() {
59                       userModel = user;
60                     });
61
62           ],
63         ),
64       ),
65     );
66   }
67 }
```

```
profile_screen.dart X
lms > lib > screens > Profile > profile_screen.dart > ...
31   child: Column(
32     children: [
33       _editClicked
34         ? EditForm(
35           userModel: userModel,
36           onSaveClicked: () {
37             setState(() {
38               _editClicked = false;
39             });
40           },
41           onCancelClicked: () {
42             setState(() {
43               _editClicked = false;
44             });
45           },
46         ),
47         ) // EditForm
48         : UserDetail(
49           onEditClicked: () {
50             setState(() {
51               _editClicked = true;
52             });
53           },
54           onSignOutClicked: () {
55             logout(context);
56           },
57           onGettingUserDetails: (user) {
58             setState(() {
59               userModel = user;
60             });
61
62           ],
63         ),
64       ),
65     ),
66   );
67 }
```



A screenshot of a code editor showing the file `profile_screen.dart`. The code is written in Dart and defines a class `ProfileScreen` with a `Widget build(BuildContext context)` method. The method contains logic for displaying user details and handling logout. The code is part of a project structure under `lms > lib > screens > Profile`.

```
profile_screen.dart ×
lms > lib > screens > Profile > profile_screen.dart > ...
61     ],
62   ],
63   ],
64   ],
65   ],
66   );
67 }
68
69 void logout(BuildContext context) {
70   context.read<AuthenticationService>().signOut();
71   Navigator.pushReplacementNamed(context, welcomeRoute);
72 }
73
74 }
```

Source code Halaman Edit Profile

```
edit_form.dart X
Ims > lib > screens > Profile > edit_form.dart > EditForm
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:flutter/material.dart';
3 import 'package:lms/model/user_model.dart';
4 import 'package:lms/utils/constants.dart';
5 import 'package:lms/utils/helper.dart';
6
7 class EditForm extends StatefulWidget {
8   UserModel userModel = UserModel();
9   final VoidCallback onSaveClicked;
10  final VoidCallback onCancelClicked;
11
12  EditForm({
13    Key? key,
14    required this.userModel,
15    required this.onSaveClicked,
16    required this.onCancelClicked,
17  }) : super(key: key);
18
19  @override
20  State<EditForm> createState() => _EditFormState();
21}
22
23 class _EditFormState extends State<EditForm> {
24   final _formKey = GlobalKey<FormState>();
25
26   var firstNameController = TextEditingController();
27
28   var lastNameController = TextEditingController();
29
30   @override
```

```
edit_form.dart X
Ims > lib > screens > Profile > edit_form.dart > EditForm
31 void initState() {
32   firstNameController =
33     TextEditingController(text: widget.userModel.firstName);
34   lastNameController = TextEditingController(text: widget.userModel.lastName);
35   super.initState();
36 }
37
38 @override
39 Widget build(BuildContext context) {
40   updateUserData() async {
41     await FirebaseFirestore.instance
42       .collection('users')
43       .doc(widget.userModel.uid)
44       .update(
45         {
46           'first_name': firstNameController.text,
47           'last_name': lastNameController.text
48         },
49       ).then((value) {
50         showSnackbar("User details updated successfully", context);
51         widget.onSaveClicked();
52       }).catchError((error) {
53         showSnackbar(error.toString(), context);
54       });
55   }
56
57   final firstNameField = TextFormField(
58     autofocus: false,
59     controller: firstNameController,
60     keyboardType: TextInputType.name,
```

```
edit_form.dart X
lms > lib > screens > Profile > edit_form.dart > EditForm
61     onSaved: (value) {
62         firstNameController.text = value!;
63     },
64     validator: (value) {
65         if (value!.isEmpty) {
66             return "First name cannot be empty";
67         }
68         RegExp regex = RegExp(r"^.{3,}$");
69         if (!regex.hasMatch(value)) {
70             return "Enter valid name(Min. 3 characters)";
71         }
72         return null;
73     },
74     textInputAction: TextInputAction.next,
75     decoration: InputDecoration(
76         prefixIcon: const Icon(Icons.account_circle),
77         contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
78         hintText: 'First Name',
79         border: OutlineInputBorder(
80             borderRadius: BorderRadius.circular(10),
81         )), // OutlineInputBorder // InputDecoration
82     ); // TextFormField
83
84 //last name field
85 final lastNameField = TextFormField(
86     autofocus: false,
87     controller: lastNameController,
88     keyboardType: TextInputType.name,
89     onSaved: (value) {
90         lastNameController.text = value!;
```

```
edit_form.dart X
lms > lib > screens > Profile > edit_form.dart > EditForm
91     },
92     validator: (value) {
93         if (value!.isEmpty) {
94             return "Last name cannot be empty";
95         }
96         return null;
97     },
98     textInputAction: TextInputAction.next,
99     decoration: InputDecoration(
100         prefixIcon: const Icon(Icons.account_circle),
101         contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
102         hintText: 'Last Name',
103         border: OutlineInputBorder(
104             borderRadius: BorderRadius.circular(10),
105         )), // OutlineInputBorder // InputDecoration
106     ); // TextFormField
107
108 //Save Button
109 final saveButton = Material(
110     elevation: 5,
111     color: Colors.blue.shade800,
112     borderRadius: BorderRadius.circular(30),
113     child: MaterialButton(
114         padding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
115         minWidth: MediaQuery.of(context).size.width,
116         onPressed: () {
117             updateUserData();
118         },
119         child: const Text(
120             "SAVE",
```

```
edit_form.dart X
Ims > lib > screens > Profile > edit_form.dart > EditForm
121     textAlign: TextAlign.center,
122     style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
123   ), // Text
124   ), // MaterialButton
125 ); // Material
126
127 final cancelButton = Material(
128   elevation: 5,
129   color: Colors.redAccent,
130   borderRadius: BorderRadius.circular(30),
131   child: MaterialButton(
132     padding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
133     minWidth: MediaQuery.of(context).size.width,
134     onPressed: () {
135       widget.onCancelClicked();
136     },
137     child: const Text(
138       "CANCEL",
139       textAlign: TextAlign.center,
140       style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
141     ), // Text
142   ), // MaterialButton
143 ); // Material
144
145 Widget sizedBoxMargin(double value) {
146   return SizedBox(height: value);
147 }
148
149 return Form(
150   key: _formKey,
```

```
edit_form.dart X
Ims > lib > screens > Profile > edit_form.dart > EditForm
151   child: Column(
152     mainAxisAlignment: MainAxisAlignment.center,
153     crossAxisAlignment: CrossAxisAlignment.center,
154     children: [
155       firstNameField,
156       sizedBoxMargin(20),
157       lastNameField,
158       sizedBoxMargin(30),
159       customButton(
160         widget.onCancelClicked, "CANCEL", context, redButtonColor),
161         sizedBoxMargin(20),
162         customButton(updateUserData, "SAVE", context, blueButtonColor),
163         sizedBoxMargin(40),
164       ],
165     ), // Column
166   ); // Form
167 }
168 }
```

○

Berdasarkan implementasi di atas, widget button edit, button sign out, dan tab bar terdapat pada halaman profile untuk menampilkan info dari akun user yang sedang login saat itu. Selanjutnya, halaman edit profile terdapat widget form, button cancel, dan button save untuk dapat mengubah info akun user yang sedang login saat itu.

- Bab 8: Navigation, Form, dan Form Validation
 - Implementasi Navigation pada Halaman Profile

```
profile_screen.dart X
Ims > lib > screens > Profile > profile_screen.dart > _ProfileScreenState > navigateTo
1 import 'package:flutter/material.dart';
2 import 'package:lms/model/user_model.dart';
3 import 'package:lms/screens/Profile/edit_form.dart';
4 import 'package:lms/screens/Profile/user_details.dart';
5 import 'package:lms/services/authentication_service.dart';
6 import 'package:provider/provider.dart';
7 import '.../../utils/constants.dart';
8
9 class ProfileScreen extends StatefulWidget {
10   const ProfileScreen({Key? key}) : super(key: key);
11
12   @override
13   State<ProfileScreen> createState() => _ProfileScreenState();
14 }
15
16 class _ProfileScreenState extends State<ProfileScreen> {
17   UserModel userModel = UserModel();
18
19   bool _editClicked = false;
20
21   void navigateTo(String route) {
22     Navigator.pushNamed(context, route);
23   }
24
25   @override
26   Widget build(BuildContext context) {
27     return Center(
28       child: SingleChildScrollView(
29         child: Padding(
30           padding: const EdgeInsets.all(24),

```

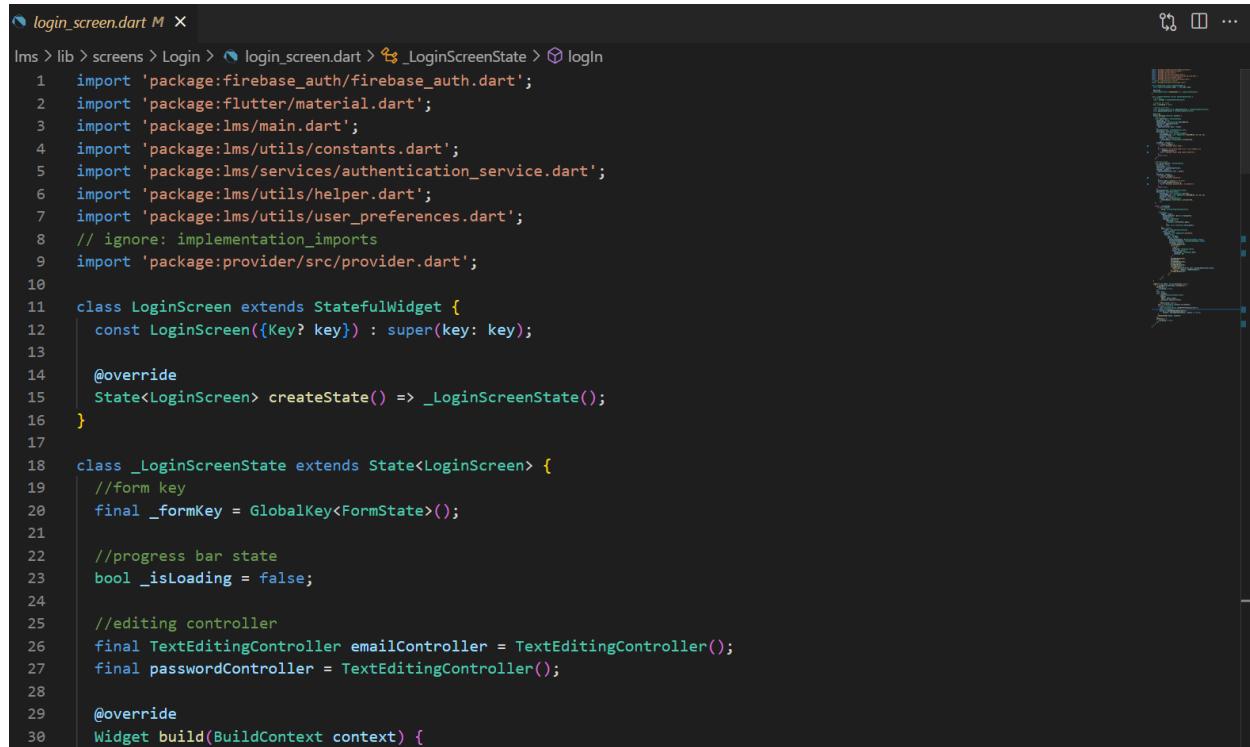
```
profile_screen.dart X
Ims > lib > screens > Profile > profile_screen.dart > _ProfileScreenState > navigateTo
31   child: Column(
32     children: [
33       _editClicked
34         ? EditForm(
35           userModel: userModel,
36           onSaveClicked: () {
37             setState(() {
38               _editClicked = false;
39             });
40           },
41           onCancelClicked: () {
42             setState(() {
43               _editClicked = false;
44             });
45           },
46         ) // EditForm
47         : UserDetail(
48           onEditClicked: () {
49             setState(() {
50               _editClicked = true;
51             });
52           },
53           onSignOutClicked: () {
54             logout(context);
55           },
56           onGettingUserDetails: (user) {
57             setState(() {
58               userModel = user;
59             });
60           },

```

A screenshot of a code editor interface, likely Dart code in a Flutter application. The main area shows the code for `profile_screen.dart`. The code includes a `UserDetail` class definition and a `logout` function. The code editor has a dark theme with syntax highlighting. A navigation bar at the top shows the file path: `lms > lib > screens > Profile > profile_screen.dart`. To the right of the code area is a sidebar containing a tree view of the project's file structure.

```
profile_screen.dart
lms > lib > screens > Profile > profile_screen.dart > _ProfileScreenState > navigateTo
61     ],
62     ],
63     ],
64     ],
65     ],
66     );
67   }
68
69   void logout(BuildContext context) {
70     context.read<AuthenticationService>().signOut();
71     Navigator.pushReplacementNamed(context, welcomeRoute);
72   }
73 }
74
```

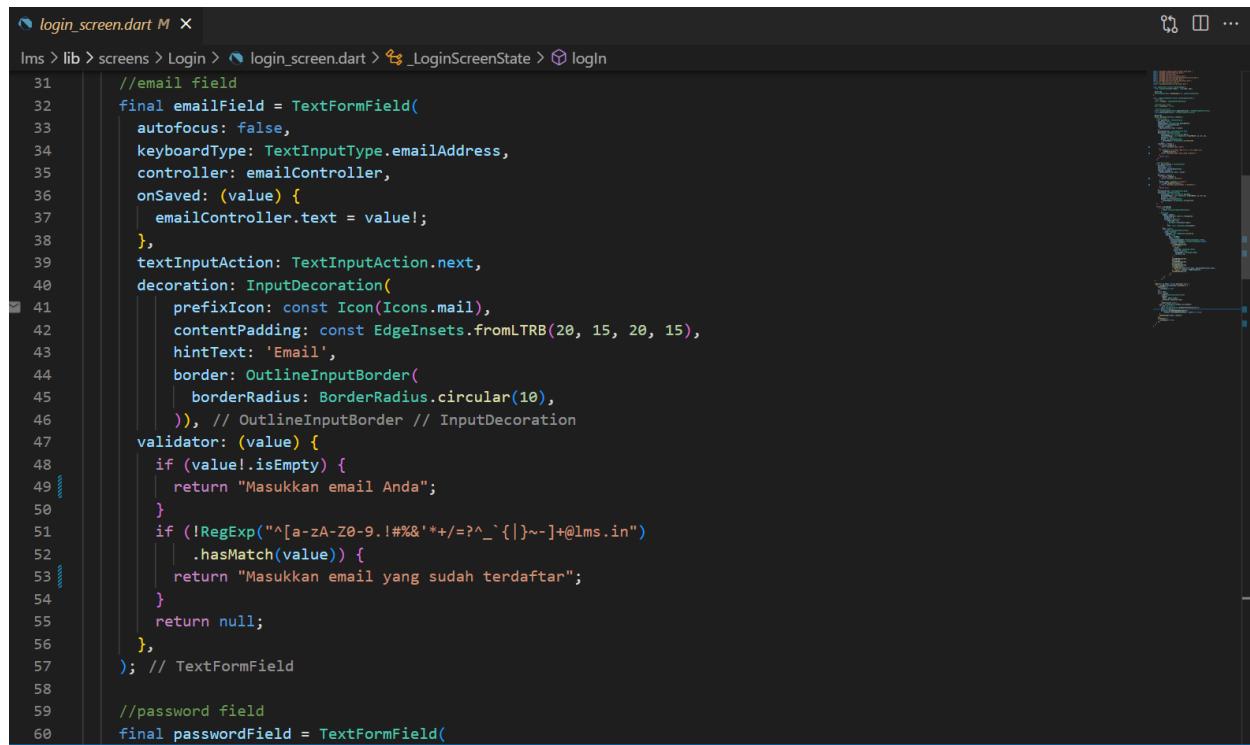
- Implementasi Form dan Form Validation pada Halaman Login



```

lms > lib > screens > Login > login_screen.dart > _LoginScreenState > login
1 import 'package:firebase_auth/firebase_auth.dart';
2 import 'package:flutter/material.dart';
3 import 'package:lms/main.dart';
4 import 'package:lms/utils/constants.dart';
5 import 'package:lms/services/authentication_service.dart';
6 import 'package:lms/utils/helper.dart';
7 import 'package:lms/utils/user_preferences.dart';
8 // ignore: implementation_imports
9 import 'package:provider/src/provider.dart';
10
11 class LoginScreen extends StatefulWidget {
12   const LoginScreen({Key? key}) : super(key: key);
13
14   @override
15   State<LoginScreen> createState() => _LoginScreenState();
16 }
17
18 class _LoginScreenState extends State<LoginScreen> {
19   //form key
20   final _formKey = GlobalKey<FormState>();
21
22   //progress bar state
23   bool _isLoading = false;
24
25   //editing controller
26   final TextEditingController emailController = TextEditingController();
27   final passwordController = TextEditingController();
28
29   @override
30   Widget build(BuildContext context) {

```



```

lms > lib > screens > Login > login_screen.dart > _LoginScreenState > login
31   //email field
32   final emailField = TextFormField(
33     autofocus: false,
34     keyboardType: TextInputType.emailAddress,
35     controller: emailController,
36     onSaved: (value) {
37       emailController.text = value!;
38     },
39    textInputAction: TextInputAction.next,
40     decoration: InputDecoration(
41       prefixIcon: const Icon(Icons.mail),
42       contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
43       hintText: 'Email',
44       border: OutlineInputBorder(
45         borderRadius: BorderRadius.circular(10),
46       ), // OutlineInputBorder // InputDecoration
47     validator: (value) {
48       if (value!.isEmpty) {
49         return "Masukkan email Anda";
50       }
51       if (!RegExp("^[a-zA-Z0-9.!#%&'*/=?^_`{|}~-]+@[lms.in]")
52           .hasMatch(value)) {
53         return "Masukkan email yang sudah terdaftar";
54       }
55       return null;
56     },
57   ); // TextFormField
58
59   //password field
60   final passwordField = TextFormField(

```

```
login_screen.dart M X
lms > lib > screens > Login > login_screen.dart > _LoginScreenState > login
61   autofocus: false,
62   obscureText: true,
63   controller: passwordController,
64   onSaved: (value) {
65     passwordController.text = value!;
66   },
67   validator: (value) {
68     if (value!.isEmpty) {
69       return "Masukkan password";
70     }
71     RegExp regex = RegExp(r"^.{6,$}");
72     if (!regex.hasMatch(value)) {
73       return "Masukkan password(Min. 6 karakter)";
74     }
75     return null;
76   },
77   textInputAction: TextInputAction.done,
78   decoration: InputDecoration(
79     prefixIcon: const Icon(Icons.vpn_key),
80     contentPadding: const EdgeInsets.fromLTRB(20, 15, 20, 15),
81     hintText: 'Password',
82     border: OutlineInputBorder(
83       borderRadius: BorderRadius.circular(10),
84     ), // OutlineInputBorder // InputDecoration
85   ); // TextFormField
86
87   return (_isLoading)
88     ? const Center(
89       child: CircularProgressIndicator(),
90     ) // Center

```

```
login_screen.dart M X
lms > lib > screens > Login > login_screen.dart > _LoginScreenState > login
91   : Scaffold(
92     appBar: AppBar(
93       backgroundColor: Colors.transparent,
94       elevation: 0,
95       leading: IconButton(
96         onPressed: () {
97           Navigator.of(context).pop();
98         },
99         icon: const Icon(Icons.arrow_back), // IconButton
100      ), // AppBar
101      body: Center(
102        child: SingleChildScrollView(
103          child: Padding(
104            padding: const EdgeInsets.all(36.0),
105            child: Form(
106              key: _formKey,
107              child: Column(
108                mainAxisAlignment: MainAxisAlignment.center,
109                crossAxisAlignment: CrossAxisAlignment.center,
110                children: <Widget>[
111                  sizedBoxMargin(10),
112                  const Text(
113                    "LOGIN",
114                    textAlign: TextAlign.center,
115                    style: TextStyle(
116                      fontWeight: FontWeight.bold,
117                      fontSize: 30,
118                    ), // TextStyle
119                  sizedBoxMargin(45),

```

```
login_screen.dart M X
Ims > lib > screens > Login > login_screen.dart > _LoginScreenState > login
121           emailField,
122           sizedBoxMargin(20),
123           passwordField,
124           sizedBoxMargin(35),
125           customButton(() {
126             | logIn(emailController.text, passwordController.text);
127             | }, "Login", context, redButtonColor),
128             sizedBoxMargin(15),
129           ], // <Widget>[]
130         ), // Column // Form
131       ), // Padding
132     ), // SingleChildScrollView // Center
133   ); // Scaffold
134 }
135
136 logIn(String email, String password) async {
137   if (_formKey.currentState!.validate()) {
138     setState(() {
139       _isLoading = true;
140     });
141     User? user;
142     await context
143       .read<AuthenticationService>()
144       .signIn(
145         | email: email.trim(),
146         | password: password.trim(),
147       )
148       .then((value) async {
149         user = FirebaseAuth.instance.currentUser;
150         if (user != null) {
```

```
login_screen.dart M X
Ims > lib > screens > Login > login_screen.dart > _LoginScreenState > login
151           await UserPreferences.saveUserPreferences(user!);
152           admin = isAdmin();
153           Navigator.pushNamedAndRemoveUntil(
154             | context, bottomNavPanelRoute, (route) => false);
155           }
156           showSnackbar(value, context);
157         );
158       setState(() {
159         _isLoading = false;
160       });
161     }
162   }
163 }
```

○ Navigation

```
constants.dart X
Ims > lib > utils > constants.dart > ...
1 // ignore_for_file: unnecessary_string_escapes
2
3 import 'package:flutter/material.dart';
4
5 void showSnackbar(String message, BuildContext context) {
6 | ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(message)));
7 }
8
9 // Routes Constants
10 const welcomeRoute = "\welcome";
11 const loginRoute = "\login";
12 const signUpRoute = "\signup";
13 const bottomNavPanelRoute = "\bottom_nav_panel";
14 const homeRoute = "\home";
15 const notificationRoute = "\notification";
16 const profileRoute = "\profile";
17 const addBookRoute = "\add_book";
18 const bookDetailRoute = "\book_detail";
19 const booksListRoute = "\books_list_screen";
20
21 // shared preferences key
22 const userFNameKey = 'first_name';
23 const userLNameKey = 'last_name';
24 const userEmailKey = 'email';
25 const userRoleKey = 'role';
26 const userUidKey = 'uid';
27
28 //role
29 const adminR = "admin";
30 const userR = "user";
```

```
constants.dart X
Ims > lib > utils > constants.dart > ...
31
32 //one signal id
33 const appId = "ce001597-0449-43d4-9fe5-e1afda694702";
34
35 //btn colors
36 const blueButtonColor = Color(0xFF1564C0);
37 const redButtonColor = Color(0xFFFF5252);
38
```

- Bab 9: Networking, JSON, dan Serialization
 - Implementasi Networking

The screenshot shows a code editor window with the following details:

- Title Bar:** notification_services.dart
- File Path:** lms > lib > services > notification_services.dart
- Function Name:** sendNotification
- Code Content:**

```
1 import 'dart:convert';
2
3 import 'package:http/http.dart';
4 import 'package:lms/utils/constants.dart';
5
6 class NotificationServices {
7     static Future<Response> sendNotification(
8         String heading, String content, List<String> tokenIds) async {
9     return await post(
10         Uri.parse('https://onesignal.com/api/v1/notifications'),
11         headers: <String, String>{
12             'Content-Type': 'application/json; charset=UTF-8',
13         },
14         body: jsonEncode(<String, dynamic>{
15             "app_id": appId,
16             "include_player_ids": tokenIds,
17             "android_accent_color": "FF9976D2",
18             "headings": {"en": heading},
19             "contents": {"en": content},
20         })),
21     );
22 }
23
24 }
```

The code implements a `sendNotification` method that sends a notification to a list of token IDs using the OneSignal API. It uses `http` package for making the POST request and `dart:convert` for JSON encoding.