

'logindata' merupakan variabel baru untuk menyimpan SharedPreference

'newUser' merupakan tipe data Boolean

'initState' artinya ini akan merupakan fungsi pertama kali yang dipanggil sebelum widget tree nya dibangun

```
lib > login_screen.dart > _LogInScreenState > checkLogin
1 import 'package:flutter/material.dart';
2 import 'package:flutter_projectsection17_18_19/main.dart';
3 import 'package:shared_preferences/shared_preferences.dart';
4 import 'package:email_validator/email_validator.dart';
5
6 class LogInScreen extends StatefulWidget {
7   const LogInScreen({super.key});
8
9   @override
10  State<LogInScreen> createState() => _LogInScreenState();
11 }
12
13 class _LogInScreenState extends State<LogInScreen> {
14
15   final GlobalKey<FormState> formKey = GlobalKey<FormState>();
16
17   final TextEditingController _nameController = TextEditingController();
18   final TextEditingController _emailController = TextEditingController();
19   final TextEditingController _passwordController = TextEditingController();
20
21   // 'logindata' merupakan variabel baru untuk menyimpan SharedPreference
22   // 'newUser' merupakan tipe data boolean
23   late SharedPreferences logindata;
24   late bool newUser;
25
26   // 'initState' artinya ini akan merupakan fungsi pertama kali yang dipanggil sebelum widget tree nya dibangun
27   @override
28   void initState() {
29     super.initState();
30     checkLogin();
31   }
32
33   void checkLogin() {
34     logindata = SharedPreferences.getInstance();
35     logindata.getBool('login') ?? true
36   }
37
38   @override
39   Widget build(BuildContext context) {
40     return Scaffold(
41       appBar: AppBar(
42         title: Text('Log In'),
43       ),
44       body: Form(
45         key: formKey,
46         child: Column(
47           children: [
48             TextFormField(
49               controller: _nameController,
50               decoration: InputDecoration(labelText: 'Name'),
51             ),
52             TextFormField(
53               controller: _emailController,
54               decoration: InputDecoration(labelText: 'Email'),
55             ),
56             TextFormField(
57               controller: _passwordController,
58               decoration: InputDecoration(labelText: 'Password'),
59             ),
60             ElevatedButton(
61               onPressed: () {
62                 if (formKey.currentState!.validate()) {
63                   Navigator.pushNamed(context, '/home');
64                 }
65               },
66               child: Text('Log In'),
67             ),
68           ],
69         ),
70       ),
71     );
72   }
73 }
```

'checkLogin' adalah fungsi asynchronous yang memeriksa apakah user telah login sebelumnya pada aplikasi menggunakan SharedPreference

'SharedPreferences' digunakan untuk menyimpan info yang perlu diakses kembali oleh aplikasi pada waktu yang beda seperti info login, data pengaturan, atau preferensi pengguna

'SharedPreferences.getInstance()' adalah untuk mendapatkan objek 'SharedPreferences' yang merepresentasikan file shared preferences default pada aplikasi

'await' digunakan untuk menunggu hasil pengembalian dari 'getInstance()' yang merupakan sebuah future sehingga kode selanjutnya akan tunggu sampai objek 'SharedPreferences' selesai

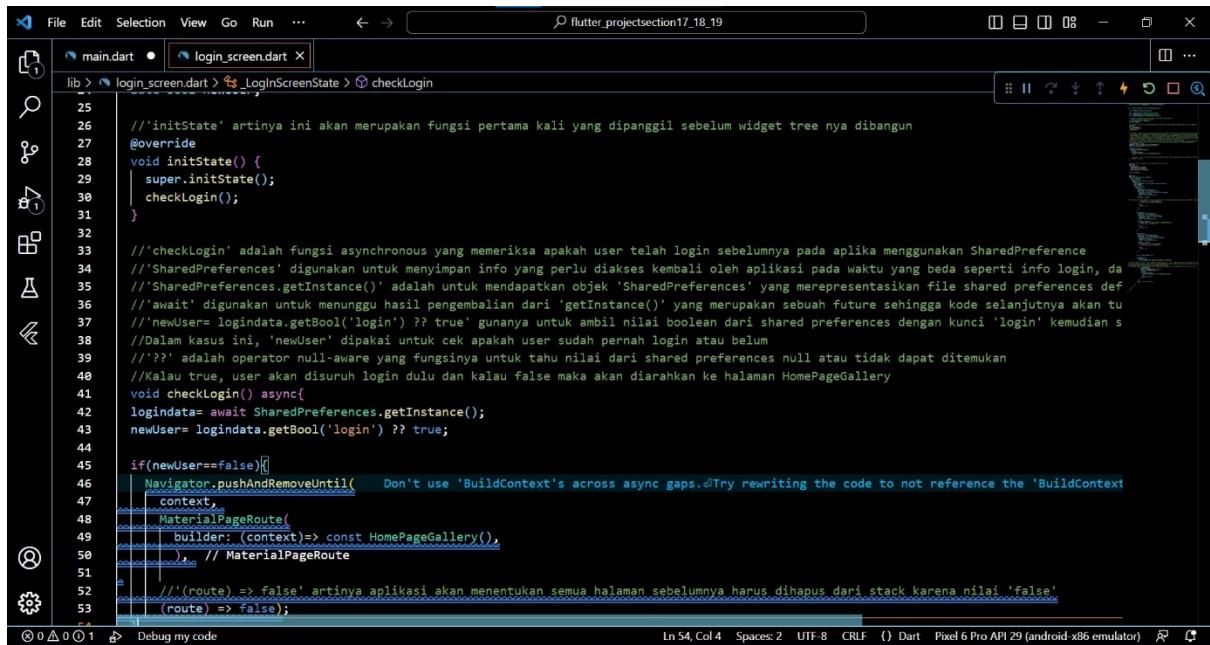
'newUser= logindata.getBool('login') ?? true' gunanya untuk ambil nilai boolean dari shared preferences dengan kunci 'login' kemudian simpan ke dalam varibel 'newUser'

Dalam kasus ini, 'newUser' dipakai untuk cek apakah user sudah pernah login atau belum

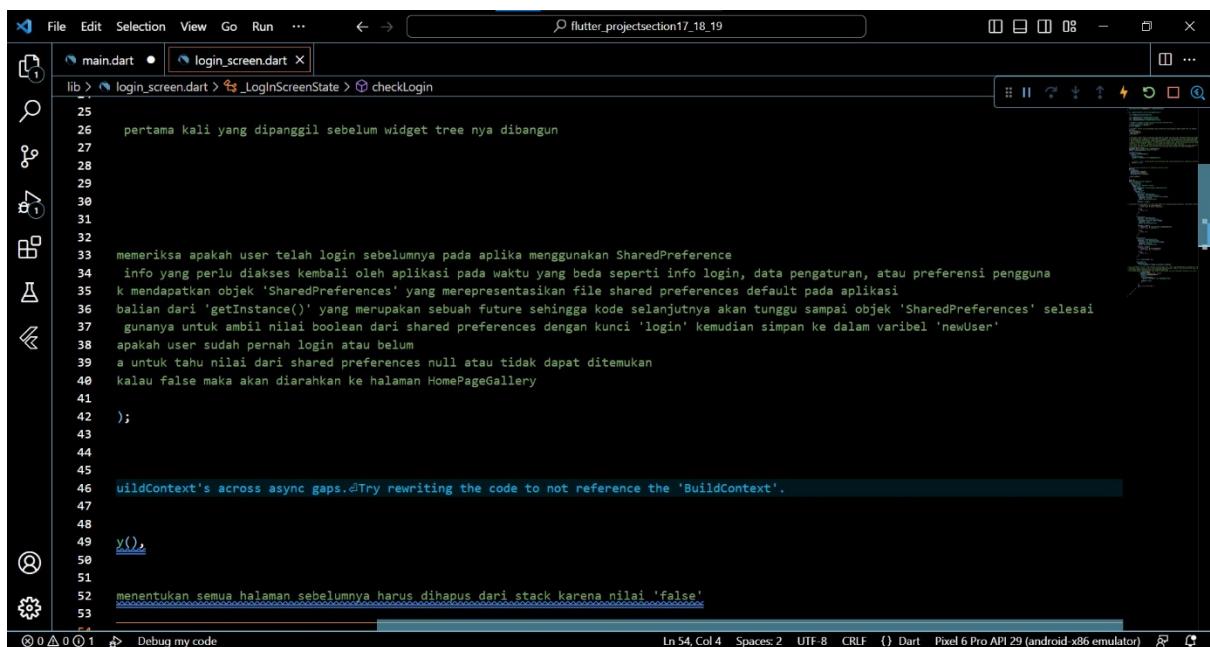
'???' adalah operator null-aware yang fungsinya untuk tahu nilai dari shared preferences null atau tidak dapat ditemukan

Kalau true, user akan disuruh login dulu dan kalau false maka akan diarahkan ke halaman HomePageGallery

'(route) => false' artinya aplikasi akan menentukan semua halaman sebelumnya harus dihapus dari stack karena nilai 'false'



```
lib > login_screen.dart > _LoginScreenState > checkLogin
25
26 // 'initState' artinya ini akan merupakan fungsi pertama kali yang dipanggil sebelum widget tree nya dibangun
27 @override
28 void initState() {
29   super.initState();
30   checkLogin();
31 }
32
33 // 'checkLogin' adalah fungsi asynchronous yang memeriksa apakah user telah login sebelumnya pada aplikasi menggunakan SharedPreference
34 // 'SharedPreferences.getInstance()' adalah untuk mendapatkan objek 'SharedPreferences' yang merepresentasikan file shared preferences default pada aplikasi
35 // 'await' digunakan untuk menunggu hasil pengembalian dari 'getInstance()' yang merupakan sebuah future sehingga kode selanjutnya akan terealisasi setelah 'getInstance()' berhasil
36 // 'newUser' merupakan variabel yang menyimpan hasil pengembalian dari 'getBool('login') ?? true' gunanya untuk ambil nilai boolean dari shared preferences dengan kunci 'login' kemudian simpan ke dalam variabel 'newUser'
37 // Dalam kasus ini, 'newUser' dipakai untuk cek apakah user sudah pernah login atau belum
38 // '???' adalah operator null-aware yang fungsinya untuk tahu nilai dari shared preferences null atau tidak dapat ditemukan
39 // Kalau true, user akan disuruh login dulu dan kalau false maka akan diarahkan ke halaman HomePageGallery
40 void checkLogin() async{
41   logindata= await SharedPreferences.getInstance();
42   newUser= logindata.getBool('login') ?? true;
43
44   if(newUser==false){
45     Navigator.pushAndRemoveUntil(  Don't use 'BuildContext's across async gaps.<-- Try rewriting the code to not reference the 'BuildContext'
46       context,
47       MaterialPageRoute(
48         builder: (context)> const HomePageGallery(),
49       ), // MaterialPageRoute
50     );
51   }
52   // '(route) => false' artinya aplikasi akan menentukan semua halaman sebelumnya harus dihapus dari stack karena nilai 'false'
53   // (route) => false);
54 }
```



```
lib > login_screen.dart > _LoginScreenState > checkLogin
25
26 pertama kali yang dipanggil sebelum widget tree nya dibangun
27
28
29
30
31
32
33 memeriksa apakah user telah login sebelumnya pada aplikasi menggunakan SharedPreference
34   info yang perlu diakses kembali oleh aplikasi pada waktu yang beda seperti info login, data pengaturan, atau preferensi pengguna
35   k mendapatkan objek 'SharedPreferences' yang merepresentasikan file shared preferences default pada aplikasi
36   bahan dari 'getInstance()' yang merupakan sebuah future sehingga kode selanjutnya akan tunggu sampai objek 'SharedPreferences' selesai
37   gunanya untuk ambil nilai boolean dari shared preferences dengan kunci 'login' kemudian simpan ke dalam variabel 'newUser'
38   apakah user sudah pernah login atau belum
39   a untuk tahu nilai dari shared preferences null atau tidak dapat ditemukan
40   kalau false maka akan diarahkan ke halaman HomePageGallery
41
42 );
43
44
45
46 buildContext's across async gaps.<-- Try rewriting the code to not reference the 'BuildContext'.
47
48
49 y());
50
51
52 menentukan semua halaman sebelumnya harus dihapus dari stack karena nilai 'false'
53
```

Method dispose() digunakan untuk menghindari kebocoran memori

```
lib > login_screen.dart > _LoginScreenState > checkLogin
52     // (route) == false artinya aplikasi akan menentukan semua halaman sebelumnya harus dihapus dari stack karena nilai
53     // (route) > false;
54   }
55 }
56
57 //Method dispose() digunakan untuk menghindari kebocoran memori
58 @override
59 void dispose(){
60   _nameController.dispose();
61   _emailController.dispose();
62   _passwordController.dispose();
63 }
64
65 super.dispose();
66
67 @override
68 Widget build(BuildContext context) {
69   return Scaffold(
70     body: Container(
71       padding: const EdgeInsets.all(16),
72       child: Form(
73         autovalidateMode: AutovalidateMode.onUserInteraction,
74         key: formKey,
75         child: ListView(
76           children: [
77             TextFormField(
78               controller: _nameController,
79               decoration: const InputDecoration(
80                 prefixIcon: Icon(Icons.account_circle_rounded),
81                 labelText: 'Username',
82               ),
83             ),
84           ],
85         ),
86       ),
87     ),
88   );
89 }
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
```

'(value!=null && value.length<4)' artinya jika inputan user tidak kosong dan kurang dari lima karakter maka akan ada pesan error

```
lib > login_screen.dart > _LoginScreenState > checkLogin
80     prefixIcon: Icon(Icons.account_circle_rounded),
81     labelText: 'Username',
82     border: OutlineInputBorder(),
83   ), // InputDecoration
84   validator: (value){
85
86   // '(value!=null && value.length<4)' artinya jika inputan user tidak kosong dan kurang dari lima karakter maka akan ada pesan error
87   if (value==null && value.length<4){
88     return 'Enter at least 4 characters';
89   }
90   else{
91     return null;
92   }
93 },
94 ), // TextFormField
95
96 TextFormField(
97   controller: _emailController,
98   decoration: const InputDecoration(
99     prefixIcon: Icon(Icons.email_rounded),
100     labelText: 'Email',
101     border: OutlineInputBorder(),
102   ), // InputDecoration
103   validator: (email){
104     if (email==null && !EmailValidator.validate(email)){
105       return 'Enter a valid email';
106     }
107     else{
108       return null;
109     }
110   }
111 ),
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
```

```
lib > login_screen.dart > LoginScreenState > build
104 |     if (_email!=null && !EmailValidator.validate(_email)){
105 |         | return 'Enter a valid email';
106 |     }
107 |     else{
108 |         | return null;
109 |     },
110 | ),
111 | ), // TextFormField
112 |
113 | TextFormField(
114 |     controller: _passwordController,
115 |     decoration: const InputDecoration(
116 |         prefixIcon: Icon(Icons.password_rounded),
117 |         labelText: 'Password',
118 |         border: OutlineInputBorder(),
119 |     ), // InputDecoration
120 |     validator: (value){
121 |         if (value==null && value.length<5){
122 |             | return 'Enter min. 5 Characters';
123 |         }
124 |         else{
125 |             | return null;
126 |         },
127 |     }, // TextFormField
128 |
129 |     const SizedBox(height: 20),
130 |
131 |     ElevatedButton(
132 |         ...
133 |     );
134 |
135 |     ...
136 | ),
137 | ),
138 | ),
139 | ),
140 | ),
141 | ),
142 | ),
143 | ),
144 | ),
145 | ),
146 | ),
147 | ),
148 | ),
149 | ),
150 | ),
151 | ),
152 | ),
153 | ),
154 | ),
155 | ),
156 | ),
157 | );
158 | );
159 | );
160 | );
```

Jika 'isValidForm' bernilai true, aplikasi akan menyimpan nilai 'false' pada SharedPreferences dengan key 'login'

'logindata.setBool('login', false)' artinya aplikasi memanggil metode 'setBool' untuk ubah nilai key 'login' menjadi 'false'

Arti 'false' adalah menandakan kalau user sudah melakukan login

'pushAndRemoveUntil' akan menghapus semua halaman yang ada pada stack dan menambahkan halaman baru ke stack

```
lib > login_screen.dart > LoginScreenState > build
131 |     ElevatedButton(
132 |         onPressed: (){
133 |             | final isValidForm= formKey.currentState!.validate();
134 |
135 |             //Jika 'isValidForm' bernilai true, aplikasi akan menyimpan nilai 'false' pada SharedPreferences dengan key 'login'
136 |             //logindata.setBool('login', false)' artinya aplikasi memanggil metode 'setBool' untuk ubah nilai key 'login' menjadi 'false'
137 |             //Arti 'false' adalah menandakan kalau user sudah melakukan login
138 |             //pushAndRemoveUntil' akan menghapus semua halaman yang ada pada stack dan menambahkan halaman baru ke stack
139 |
140 |             if (isValidForm){
141 |                 logindata.setBool('login', false);
142 |                 Navigator.pushAndRemoveUntil(
143 |                     context,
144 |                     MaterialPageRoute(
145 |                         builder: (context)=> const HomePageGallery(),
146 |                     ), // MaterialPageRoute
147 |                     (route) => false
148 |                 );
149 |             },
150 |             ],
151 |             child: const Text('Login'),
152 |         ), // ElevatedButton
153 |     ],
154 | ), // ListView
155 | ),
156 | ), // Container
157 | ); // Scaffold
158 | );
159 | );
160 | );
```

'late' fungsinya untuk menunda inisialisasi sebuah variabel sampai benar-benar dibutuhkan agar hemat memori

'initState' artinya merupakan sebuah metode yang pertama kali diinisialisasi

Dalam kasus ini 'initial()' akan pertama kali diinisialisasi

'initial' akan menjalankan metode 'getInstance' dari 'SharedPreferences' dan mengembalikan hasil sebagai 'Future'

Artinya akan ditunggu sampai hasil Future diterima sebelum lanjut ke kode selanjutnya

```
lib > main.dart > _HomePageGalleryState
55 // initial akan menjalankan metode getInstance dari SharedPreferences dan mengembalikan hasil sebagai Future
56 //Artinya akan ditunggu sampai hasil Future diterima sebelum lanjut ke kode selanjutnya
57 void initial() async{
58 | logindata= await SharedPreferences.getInstance();
59 }

60
61 @override
62 Widget build(BuildContext context) {
63 | return Scaffold(
64 | | appBar: AppBar(
65 | | | backgroundColor: Colors.purple,
66 | | | title: const Text('Gallery'),
67 | | ), // AppBar
68 | | drawer: Drawer(
69 | | | child: ListView(
70 | | | | children: [
71 | | | | | TextButton(
72 | | | | | onPressed: (){
73 | | | | | logindata.setBool('login', true);
74 | | | | | Navigator.pushReplacement(
75 | | | | | | context,
76 | | | | | | MaterialPageRoute(
77 | | | | | | | builder: (context)=> const LogInScreen(),
78 | | | | | | ), // MaterialPageRoute
79 | | | | | );
80 | | | | | ),
81 | | | | | child: const Text('Log Out', style: TextStyle(fontSize: 20),),
82 | | | | | ), // TextButton
83 | | | | ],
84 | | | ),
85 | | | ),
86 | | | ),
87 | | | body: Gallery(),
88 | | | ); // Scaffold
89 |
90 |

91 //Dibuat kelas GalleryItem dan memiliki 3 properti yaitu image, onYesPressed, dan onNoPressed
92 //image merupakan objek dari kelas Image dan properti ini akan simpan info tentang gambar yang ditampilkan pada 'GalleryItem'
93 //onYesPressed merupakan sebuah fungsi yang akan dipanggil saat tombol 'Yes' ditekan dan begitu juga untuk onNoPressed
94 class GalleryItem{
95 | final Image image;
96 | final Function onYesPressed;
97 | final Function onNoPressed;
98 |
99 | GalleryItem({required this.image, required this.onYesPressed, required this.onNoPressed});
100 }
101
102 }

103
104 class Gallery extends StatefulWidget {
105 | | Gallery({super.key});
106 |
107 |
108 //Dibuat objek baru yaitu 'myGallerItems' yang bertipe List yang isinya adalah objek-objek dari kelas 'GalleryItem'
109 //Objek dari kelas 'GalleryItem' memiliki 3 parameter yaitu item, onYesPressed, dan onNoPressed
110 final List<GalleryItem> myGallerItems=[
111 | | GalleryItem(
112 | | |
```

```
lib > main.dart > _HomePageGalleryState
83 | | | ],
84 | | | ],
85 | | | ),
86 | | | ),
87 | | | body: Gallery(),
88 | | | );
89 |
90 |

91 //Dibuat kelas GalleryItem dan memiliki 3 properti yaitu image, onYesPressed, dan onNoPressed
92 //image merupakan objek dari kelas Image dan properti ini akan simpan info tentang gambar yang ditampilkan pada 'GalleryItem'
93 //onYesPressed merupakan sebuah fungsi yang akan dipanggil saat tombol 'Yes' ditekan dan begitu juga untuk onNoPressed
94 class GalleryItem{
95 | final Image image;
96 | final Function onYesPressed;
97 | final Function onNoPressed;
98 |
99 | GalleryItem({required this.image, required this.onYesPressed, required this.onNoPressed});
100 }
101
102 }

103
104 class Gallery extends StatefulWidget {
105 | | Gallery({super.key});
106 |
107 |
108 //Dibuat objek baru yaitu 'myGallerItems' yang bertipe List yang isinya adalah objek-objek dari kelas 'GalleryItem'
109 //Objek dari kelas 'GalleryItem' memiliki 3 parameter yaitu item, onYesPressed, dan onNoPressed
110 final List<GalleryItem> myGallerItems=[
111 | | GalleryItem(
112 | | |
```

```
lib > main.dart > _HomePageGalleryState
111 | GalleryItem(
112 |   image: Image.asset('assets/images/image1.jpg', fit: BoxFit.cover),
113 |   onYesPressed: (BuildContext context, ImageProvider image){
114 |     Navigator.pushNamed(
115 |       context,
116 |       '/2',
117 |       arguments: image
118 |     );
119 |   },
120 |   onNoPressed: (BuildContext context){
121 |     Navigator.pop(context);
122 |   },
123 | }, // GalleryItem
124 | GalleryItem(
125 |   image: Image.asset('assets/images/image2.jpg', fit: BoxFit.cover),
126 |   onYesPressed: (BuildContext context, ImageProvider image){
127 |     Navigator.pushNamed(
128 |       context,
129 |       '/3',
130 |       arguments: image
131 |     );
132 |   },
133 |   onNoPressed: (BuildContext context){
134 |     Navigator.pop(context);
135 |   },
136 |   GalleryItem(
137 |     image: Image.asset('assets/images/image3.jpg', fit: BoxFit.cover),
138 |     onYesPressed: (BuildContext context, ImageProvider image){
139 |       Navigator.pushNamed(
140 |         context,
141 |         '/1',
142 |       );
143 |     },
144 |     onNoPressed: (BuildContext context){
145 |       Navigator.pop(context);
146 |     },
147 |     GalleryItem(
148 |       image: Image.asset('assets/images/image4.jpg', fit: BoxFit.cover),
149 |       onYesPressed: (BuildContext context, ImageProvider image){
150 |         Navigator.pushNamed(
151 |           context,
152 |           '/3',
153 |           arguments: image
154 |         );
155 |       },
156 |       onNoPressed: (BuildContext context){
157 |         Navigator.pop(context);
158 |       },
159 |       GalleryItem(
160 |         image: Image.asset('assets/images/image5.jpg', fit: BoxFit.cover),
161 |         onYesPressed: (BuildContext context, ImageProvider image){
162 |           Navigator.pushNamed(
163 |             context,
164 |             '/3',
165 |             arguments: image
166 |           );
167 |         },
168 |         onNoPressed: (BuildContext context){
169 |           Navigator.pop(context);
170 |         },
171 |         GalleryItem(
172 |           image: Image.asset('assets/images/image6.jpg', fit: BoxFit.cover),
173 |           onYesPressed: (BuildContext context, ImageProvider image){
174 |             Navigator.pushNamed(
175 |               context,
176 |               '/1',
177 |             );
178 |           },
179 |           onNoPressed: (BuildContext context){
180 |             Navigator.pop(context);
181 |           }
182 |         );
183 |       );
184 |     );
185 |   );
186 | }
```

```
lib > main.dart > _HomePageGalleryState
167 |   },
168 |   GalleryItem(
169 |     image: Image.asset('assets/images/image6.jpg', fit: BoxFit.cover),
170 |     onYesPressed: (BuildContext context, ImageProvider image){
171 |       Navigator.pushNamed(
172 |         context,
173 |         '/3',
174 |         arguments: image
175 |       );
176 |     },
177 |     onNoPressed: (BuildContext context){
178 |       Navigator.pop(context);
179 |     },
180 |   ),
181 |   GalleryItem(
182 |     image: Image.asset('assets/images/image7.jpg', fit: BoxFit.cover),
183 |     onYesPressed: (BuildContext context, ImageProvider image){
184 |       Navigator.pushNamed(
185 |         context,
186 |         '/3',
187 |         arguments: image
188 |       );
189 |     },
190 |     onNoPressed: (BuildContext context){
191 |       Navigator.pop(context);
192 |     },
193 |   ),
194 |   GalleryItem(
195 |     image: Image.asset('assets/images/image8.jpg', fit: BoxFit.cover),
196 |     onYesPressed: (BuildContext context, ImageProvider image){
197 |       Navigator.pushNamed(
198 |         context,
199 |         '/3',
200 |         arguments: image
201 |       );
202 |     },
203 |   );
204 |
205 |   @override
206 |   State<Gallery> createState() => _GalleryState();
207 }
208
209 class _GalleryState extends State<Gallery> {
210   @override
211   Widget build(BuildContext context) {
212     return GridView.count(
213       crossAxisCount: 3,
214
215       //Setiap item pada 'myGalleryItems' akan dirender menggunakan widget 'Inkwell'
216       //Inkwell' adalah widget yang mengimplementasikan sebuah tindakan ketika pengguna menyentuh widget tersebut
217       //Setiap 'Inkwell' akan menerima sebuah tindakan ketika diklik oleh pengguna menggunakan properti 'onTap'
218       //widget.'digunakan untuk mengakses properti 'myGalleryItems'.
219       //map()' adalah metode pada List untuk ubah tiap item di dalam List 'myGalleryItems' ke bentuk widget yang akan dirender di dalam Grid
220       children: widget.myGalleryItems.map((item) => InkWell(
221         onTap: () {
222           showModalBottomSheet(
223             shape: const RoundedRectangleBorder(
```

The image displays two screenshots of a Flutter development environment, likely Android Studio or VS Code, showing the code for a mobile application.

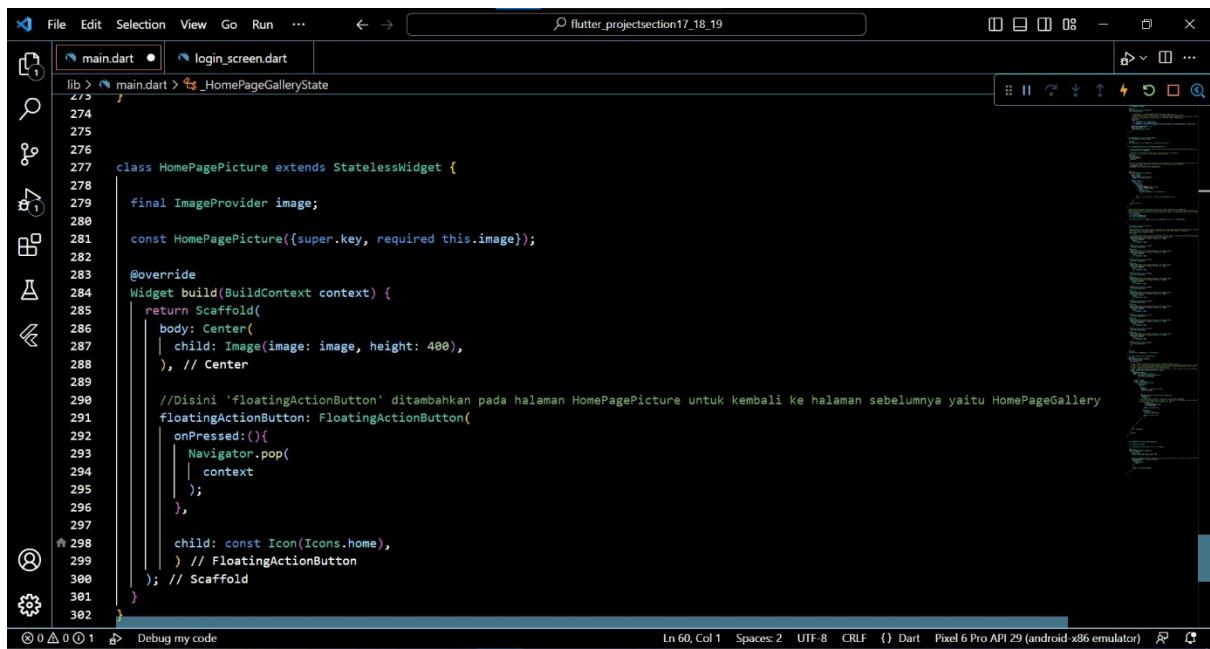
Left Editor (main.dart):

```
lib > main.dart > _HomePageGalleryState
218 | // widget.' digunakan untuk mengakses properti 'myGalleryItems'
219 | /// 'map()' adalah metode pada List untuk ubah tiap item di dalam List 'myGalleryItems' ke bentuk widget yang akan dirender di dalam Grid
220 | children: widget.myGalleryItems.map((item) => InkWell(
221 |   onTap: () {
222 |     showModalBottomSheet(
223 |       shape: const RoundedRectangleBorder(
224 |         borderRadius: BorderRadius.vertical(
225 |           top: Radius.circular(20)
226 |         ) // BorderRadius.vertical
227 |       ), // RoundedRectangleBorder
228 |       context: context,
229 |       builder: (context) {
230 |         return SingleChildScrollView(
231 |           child: Column(
232 |             children: [
233 |
234 |               Padding(
235 |                 padding: const EdgeInsets.all(50),
236 |                 child: item.image,
237 |               ), // Padding
238 |
239 |               Row(
240 |                 mainAxisAlignment: MainAxisAlignment.spaceAround,
241 |                 children: [
242 |                   TextButton(
243 |                     onPressed: () {
244 |                       // 'item.image' adalah objek dari 'Image.asset' yang bertipe 'AssetImage'
245 |                       // Kita perlu menggunakan properti 'image' dari 'AssetImage' untuk mendapatkan objek 'ImageProvider'
246 |                       // Karena itu ada '.image' lagi dibelakang 'item.image'
247 |                     onPressed: () {
248 |                       item.onYesPressed(context,
249 |                         item.image.image);
250 |                     },
251 |                     child: const Text('Yes'),
252 |                   ), // TextButton
253 |                   TextButton(
254 |                     onPressed: () {
255 |                       Navigator.pop(context);
256 |                     },
257 |                     child: const Text('No'),
258 |                   ), // TextButton
259 |                 ],
260 |               ), // Row
261 |             ],
262 |           ); // SingleChildScrollView
263 |
264 |         );
265 |       },
266 |       child: item.image,
267 |     ) // InkWell
268 |   );
269 | }
270 | .toList(),
271 | ); // GridView.count
272 |
273 | }
274 |
275 | };
```

Right Editor (login_screen.dart):

```
lib > main.dart > _HomePageGalleryState
246 |   onPressed: () {
247 |     item.onYesPressed(context,
248 |       item.image.image);
249 |   },
250 |   child: const Text('Yes'),
251 | ), // TextButton
252 | TextButton(
253 |   onPressed: () {
254 |     Navigator.pop(context);
255 |   },
256 |   child: const Text('No'),
257 | ), // TextButton
258 | ],
259 | ), // Row
260 | ],
261 | ), // Column
262 | ); // SingleChildScrollView
263 |
264 | );
265 | },
266 | child: item.image,
267 | ) // InkWell
268 | )
269 | .toList(),
270 | ); // GridView.count
271 |
272 |
273 | }
274 |
275 | };
```

In both editors, the status bar at the bottom indicates the current file is main.dart, and the code is being run on a Pixel 6 Pro API 29 (android-x86 emulator). The code snippets show parts of the `_HomePageGalleryState` class, specifically the implementation of a modal bottom sheet and a grid view.



```
File Edit Selection View Go Run ... flutter_projectsection17_18_19
main.dart lib > main.dart > _HomePageGalleryState
274
275
276
277 class HomePagePicture extends StatelessWidget {
278   final ImageProvider image;
279
280   const HomePagePicture({super.key, required this.image});
281
282   @override
283   Widget build(BuildContext context) {
284     return Scaffold(
285       body: Center(
286         child: Image(image: image, height: 400),
287       ), // Center
288
289       //Disini 'floatingActionButton' ditambahkan pada halaman HomePagePicture untuk kembali ke halaman sebelumnya yaitu HomePageGallery
290       floatingActionButton: FloatingActionButton(
291         onPressed:(){
292           Navigator.pop(
293             context
294           );
295         },
296
297         child: const Icon(Icons.home),
298       ); // FloatingActionButton
299     ); // Scaffold
300   }
301 }
```

Ln 60, Col 1 Spaces: 2 UTRF-8 CRLF {} Dart Pixel 6 Pro API 29 (android-x86 emulator)

```
main.dart lib/main.dart _HomePageGalleryState
273
274
275
276
277 class HomePagePicture extends StatelessWidget {
278
279   final ImageProvider image;
280
281   const HomePagePicture({super.key, required this.image});
282
283   @override
284   Widget build(BuildContext context) {
285     return Scaffold(
286       body: Center(
287         child: Image(image: image, height: 400),
288       ), // Center
289
290       //Disini 'floatingActionButton' ditambahkan pada halaman HomePagePicture untuk kembali ke halaman
291       floatingActionButton: FloatingActionButton(
292         onPressed:(){
293           Navigator.pop(
294             context
295           );
296         },
297
298         child: const Icon(Icons.home),
299       ) // FloatingActionButton
300     ); // Scaffold
301   }
302 }
```

Debug my code

File Edit Selection View Go Run ...

flutter_projectsection17_18_19

lib > main.dart > _HomePageGalleryState

273

274

275

276

277 class HomePagePicture extends StatelessWidget {

278

279 final ImageProvider image;

280

281 const HomePagePicture({super.key, required this.image});

282

283 @override

284 Widget build(BuildContext context) {

285 return Scaffold(

286 body: Center(

287 | child: Image(image: image, height: 400),

288), // Center

289

290 //Disini 'floatingActionButton' ditambahkan pada halaman HomePagePicture untuk kembali ke halaman

291 floatingActionButton: FloatingActionButton(

292 onPressed:(){

293 Navigator.pop(

294 | context

295);

296 },

297

298 child: const Icon(Icons.home),

299) // FloatingActionButton

300); // Scaffold

301 }

302 }

Debug my code

File Edit Selection View Go Run ...

flutter_projectsection17_18_19

lib > main.dart > _HomePageGalleryState

273

274

275

276

277 class HomePagePicture extends StatelessWidget {

278

279 final ImageProvider image;

280

281 const HomePagePicture({super.key, required this.image});

282

283 @override

284 Widget build(BuildContext context) {

285 return Scaffold(

286 body: Center(

287 | child: Image(image: image, height: 400),

288), // Center

289

290 //Disini 'floatingActionButton' ditambahkan pada halaman HomePagePicture untuk kembali ke halaman

291 floatingActionButton: FloatingActionButton(

292 onPressed:(){

293 Navigator.pop(

294 | context

295);

296 },

297

298 child: const Icon(Icons.home),

299) // FloatingActionButton

300); // Scaffold

301 }

302 }

The screenshot shows a development environment with a code editor and a running application. The code editor on the left displays the file `main.dart`, which includes imports for `main.dart` and `_HomePageGalleryState`. It defines a class `HomePagePicture` that extends `StatelessWidget`. The class contains a floating action button that, when pressed, performs a `Navigator.pop` operation to close the current screen. The floating action button's icon is `Icons.home`. The code editor has line numbers from 275 to 302.

The right side of the interface shows a mobile application running on an emulator. The app is a login screen with three text input fields: `Username`, `Email`, and `Password`. Each field has an error message above it: `Enter at least 4 characters`, `Enter a valid email`, and `Enter min. 5 Characters`. Below the fields is a blue `Login` button. At the bottom of the screen is a keyboard.