

Pemrograman Berbasis Mobile

LAPORAN APLIKASI FLUTTER NEWS PORTAL

Disusun untuk memenuhi tugas

mata kuliah Pemrograman Berbasis Mobile

Source Code : <https://github.com/almhfzrhmn/news-portal-pbm>

Oleh :

Al-Mahfuzh Fadhlur Rohman

(2208107010016)



**JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH**

2025

DAFTAR ISI

| | |
|--|----|
| DAFTAR ISI | 2 |
| BAB I PENDAHULUAN | 3 |
| BAB II KONSEP DASAR FLUTTER | 4 |
| 2.1. Widget dan Property..... | 4 |
| 2.2. Stateless vs Stateful Widget..... | 4 |
| 2.3 Layout Widget | 4 |
| BAB III PERSIAPAN PROJECT | 5 |
| 3.1 Pembuatan File-File Dart | 5 |
| 3.2 Instalasi Package..... | 5 |
| BAB IV IMPLEMENTASI KODE | 6 |
| 4.1. main.dart..... | 6 |
| 4.2. detail.dart..... | 7 |
| 4.3. kategori.dart | 8 |
| 4.4. home.dart | 11 |
| 4.5. profil.dart | 13 |
| 4.6. formlogin.dart | 15 |
| 4.7. formregist.dart..... | 17 |
| BAB V HASIL INTERFACE | 20 |
| 5.1. Home..... | 20 |
| | 20 |
| 5.2. Kategori..... | 20 |
| 5.3. Profile | 22 |
| 5.4. Register | 23 |
| 5.5. Login..... | 24 |

BAB I

PENDAHULUAN

Aplikasi *PBM News Portal* merupakan sebuah aplikasi mobile bertema berita yang dikembangkan menggunakan framework **Flutter**. Aplikasi ini dirancang sebagai bagian dari proses pembelajaran pada mata kuliah **Pemrograman Berbasis Mobile** di Semester 6. Tujuan utama pengembangan aplikasi ini adalah untuk mengaplikasikan konsep dasar Flutter dalam membangun aplikasi berita yang mampu menampilkan artikel-artikel dari sumber API berita secara dinamis.

Repositori proyek ini dapat diakses melalui GitHub di alamat <https://github.com/almhfzrhmn/news-portal-pbm>, yang memuat implementasi lengkap aplikasi. Laporan ini akan menguraikan secara rinci tentang implementasi, struktur kode, komponen-komponen utama, serta tampilan antarmuka dari aplikasi News Portal yang telah dikembangkan.

BAB II

KONSEP DASAR FLUTTER

2.1. Widget dan Property

Flutter adalah framework aplikasi mobile sumber terbuka yang diciptakan oleh Google. Dalam pengembangan aplikasi Flutter, semua komponen disebut sebagai Widget. Setiap Widget terdiri dari beberapa property. Nilai yang digunakan dalam property dapat berupa angka, string, atau widget lainnya.

Contoh penggunaan widget dan property:

```
Container(  
  child: Text('Ini text widget'),  
  height: 30,  
)
```

Pada contoh di atas:

- Container dan text adalah Widget
- child dan height adalah Property dari Widget Container

2.2. Stateless vs Stateful Widget

1. Stateless Widget:
 - Widget yang statis atau tidak dapat berubah-ubah
 - Tidak memiliki state internal yang dapat dimodifikasi
 - Digunakan untuk UI yang tidak perlu diperbarui secara dinamis
2. Stateful Widget:
 - Widget yang dinamis atau dapat berubah-ubah
 - Memiliki state internal yang dapat dimodifikasi
 - Digunakan untuk UI yang perlu diperbarui berdasarkan interaksi pengguna atau perubahan data

2.3 Layout Widget

Layout Widget yang sering digunakan di Flutter adalah:

1. Row Widget:
 - Menyusun widget-child secara horizontal (baris)
2. Column Widget:
 - Menyusun widget-child secara vertikal (kolom)

BAB III

PERSIAPAN PROJECT

3.1 Pembuatan File-File Dart

Project ini terdiri dari beberapa file Dart yang masing-masing memiliki fungsi khusus:

- main.dart - Entry point aplikasi dan navigasi
- detail.dart - Tampilan detail artikel berita
- formlogin.dart - Form untuk login pengguna
- formregist.dart - Form untuk registrasi pengguna
- home.dart - Halaman utama yang menampilkan daftar berita
- kategori.dart - Halaman kategori berita
- profil.dart - Halaman profil pengguna

3.2 Instalasi Package

Beberapa package Flutter yang digunakan dalam project ini:

1. curved_navigation_bar:
 - Package untuk membuat navigation bar dengan tampilan melengkung
 - URL: https://pub.dev/packages/curved_navigation_bar
2. http:
 - Package untuk melakukan request HTTP ke API
 - URL: <https://pub.dev/packages/http>
3. shared_preferences:

Package untuk menyimpan data sederhana secara persisten

 - URL: https://pub.dev/packages/shared_preferences
4. fluttertoast:
 - Package untuk menampilkan pesan toast
 - Digunakan untuk notifikasi singkat

BAB IV

IMPLEMENTASI KODE

4.1. main.dart

File main.dart berfungsi sebagai entry point aplikasi dan mengatur navigasi antar halaman melalui bottom navigation bar.

```
1 import 'package:flutter/material.dart';
2 import 'package:curved_navigation_bar/curved_navigation_bar.dart';
3 import 'home.dart';
4 import 'kategori.dart';
5 import 'profil.dart';
6
7 void main() {
8   runApp(MyApp(initialPage: 0));
9 }
10
11 class MyApp extends StatefulWidget {
12   const MyApp({super.key, required this.initialPage});
13   final int initialPage;
14   @override
15   _MyAppState createState() => _MyAppState();
16 }
17
18 class _MyAppState extends State<MyApp> {
19   int _page = 0;
20   final _pageOption = [KategoriPage(), HomePage(), ProfilPage()];
21   @override
22   void initState() {
23     super.initState();
24     setState(() {
25       _page = widget.initialPage;
26     });
27   }
28
29   @override
30   Widget build(BuildContext context) {
31     return MaterialApp(
32       title: 'Portal News',
33       debugShowCheckedModeBanner: false,
34       theme: ThemeData(
35         primarySwatch: Colors.blue,
36         scaffoldBackgroundColor: Colors.white,
37       ),
38       home: SafeArea(
39         child: Scaffold(
40           body: _pageOption[_page],
41           bottomNavigationBar: CurvedNavigationBar(
42             height: 50.0,
43             color: Colors.blue,
44             backgroundColor: Colors.white,
45             onTap: (index) {
46               setState(() {
47                 _page = index;
48               });
49             },
50             items: <Widget>[
51               Icon(Icons.category, color: Colors.white),
52               Icon(Icons.home, color: Colors.white),
53               Icon(Icons.account_circle, color: Colors.white),
54             ],
55           ),
56         ),
57       );
58   }
59 }
60
61
```

Key points dari file main.dart:

- Menggunakan StatefulWidget karena perlu menyimpan state navigasi
- Menggunakan array _pageOption untuk menyimpan halaman-halaman utama
- Mengimplementasikan CurvedNavigationBar dengan 3 item: kategori, home, dan profil
- State _page digunakan untuk menentukan halaman mana yang sedang aktif

4.2. detail.dart

Berfungsi untuk menampilkan detail dari artikel berita yang dipilih.

```
1 import 'package:flutter/material.dart';
2
3 class DetailPage extends StatefulWidget {
4   const DetailPage({
5     super.key,
6     this.judul,
7     this.content,
8     this.datetime,
9     this.image,
10  });
11   final String? judul; // Nullable parameter
12   final String? content; // Nullable parameter
13   final String? datetime; // Nullable parameter
14   final String? image; // Nullable parameter
15   @override
16   _DetailPageState createState() => _DetailPageState();
17 }
18
19 class _DetailPageState extends State<DetailPage> {
20   @override
21   Widget build(BuildContext context) {
22     return SafeArea(
23       child: Scaffold(
24         floatingActionButton: FloatingActionButton(
25           child: Icon(Icons.home),
26           onPressed: () {
27             Navigator.pop(context);
28           },
29         ),
30         body: Container(
31           padding: EdgeInsets.fromLTRB(20, 20, 20, 80),
32           child: ListView(
33             children: [
34               Text(
35                 widget.judul ??
36                 'No Title', // Menangani null dengan memberi default value
37                 style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
38                 textAlign: TextAlign.center,
39               ),
40               SizedBox(height: 20),
41               Text(
42                 widget.datetime ?? 'No Date Provided', // Menangani null
43                 style: TextStyle(fontWeight: FontWeight.w300),
44               ),
45               SizedBox(height: 20),
46               widget.image != null
47                 ? Image.network(
48                     widget.image!,
49                     height: 200,
50                   ) // Menangani null untuk image
51                 : Container(), // Menampilkan kosong jika image null
52               SizedBox(height: 20),
53               Text(
54                 widget.content ?? 'No Content Available',
55                 textAlign: TextAlign.justify,
56               ), // Menangani null
57             ],
58           ),
59         ),
60       );
61   }
62 }
63
64
```

Pada file ini:

- Parameter yang diterima semuanya nullable (dengan tanda ?)
- Menggunakan null safety untuk menangani data yang mungkin kosong
- Terdapat floating action button untuk kembali ke halaman sebelumnya
- Menampilkan gambar artikel, judul, tanggal, dan konten artikel

4.3. kategori.dart

File kategori.dart menampilkan daftar kategori berita dalam bentuk icon grid.

```
1 import 'package:flutter/material.dart';
2 import 'package:fluttertoast/fluttertoast.dart';
3
4 // Constants for styling
5 const _kSectionPadding = EdgeInsets.all(8.0);
6 const _kSectionBorder = BorderSide(color: Colors.blue, width: 3.0);
7 const _kSectionBorderRadius = BorderRadius.all(Radius.circular(10.0));
8 const _kDividerHeight = 1.5;
9 const _kSectionSpacing = 16.0;
10 const _kToastDuration = Duration(seconds: 2);
11 const _kToastMargin = EdgeInsets.only(bottom: 20.0);
12 const _kToastPadding = EdgeInsets.all(10.0);
13 const _kToastBorderRadius = BorderRadius.all(Radius.circular(25.0));
14
15 class KategoriPage extends StatefulWidget {
16   const KategoriPage({super.key});
17
18   @override
19   _KategoriPageState createState() => _KategoriPageState();
20 }
21
22 class _KategoriPageState extends State<KategoriPage> {
23   late FToast fToast;
24
25   @override
26   void initState() {
27     super.initState();
28     fToast = FToast();
29     // Initialize FToast with context in didChangeDependencies or build
30   }
31
32   @override
33   void didChangeDependencies() {
34     super.didChangeDependencies();
35     fToast.init(context); // Bind FToast to the current context
36   }
37
38   // Reusable toast widget
39   Widget _buildToast(String tag) {
40     return Container(
41       margin: _kToastMargin,
42       padding: _kToastPadding,
43       decoration: const BoxDecoration(
44         color: Colors.grey,
45         borderRadius: _kToastBorderRadius,
46       ),
47       child: Text('Anda memilih kategori $tag'),
48     );
49   }
50
51   // Reusable icon button
52   Widget _iconTag({required Icon icon, required String tag, required Key key}) {
53     return IconButton(
54       key: key,
55       icon: icon,
56       tooltip: 'Kategori $tag', // Accessibility
57       onPressed: () {
58         fToast.removeQueuedCustomToasts(); // Cancel previous toasts
59         fToast.showToast(
60           toastDuration: _kToastDuration,
61           child: _buildToast(tag),
62         );
63       },
64     );
65   }
```



```

1 // Reusable category section
2 Widget _categorySection({
3   required String title,
4   required List<Widget> children,
5   Key? key,
6 }) {
7   return Container(
8     key: key,
9     padding: _kSectionPadding,
10    decoration: BoxDecoration(
11      border: Border.all(
12        color: _kSectionBorder.color,
13        width: _kSectionBorder.width,
14      ),
15      borderRadius: _kSectionBorderRadius,
16    ),
17    child: Column(
18      crossAxisAlignment: CrossAxisAlignment.start,
19      children: [
20        Text(title, style: Theme.of(context).textTheme.titleMedium),
21        const SizedBox(height: 4.0),
22        Container(height: _kDividerHeight, color: Colors.grey),
23        const SizedBox(height: 8.0),
24        ...children,
25      ],
26    ),
27  );
28 }
29
30 @override
31 Widget build(BuildContext context) {
32   return Scaffold(
33     appBar: AppBar(
34       title: const Text('Kategori'),
35       key: const Key('kategoriAppBar'),
36     ),
37     body: Padding(
38       padding: const EdgeInsets.all(16.0),
39       child: SingleChildScrollView(
40         child: Column(
41           children: [
42             _categorySection(
43               key: const Key('generalSection'),
44               title: 'General',
45               children: [
46                 Row(
47                   mainAxisAlignment: MainAxisAlignment.spaceAround,
48                   children: [
49                     _iconTag(
50                       icon: const Icon(Icons.attach_money),
51                       tag: 'General',
52                       key: const Key('generalMoneyIcon'),
53                     ),
54                     _iconTag(
55                       icon: const Icon(Icons.card_travel),
56                       tag: 'General',
57                       key: const Key('generalTravelIcon'),
58                     ),
59                     _iconTag(
60                       icon: const Icon(Icons.local_hospital),
61                       tag: 'General',
62                       key: const Key('generalHospitalIcon'),
63                     ),
64                   ],
65                 ),
66               ],
67             ),

```

```

1  const SizedBox(height: _kSectionSpacing),
2  _categorySection(
3    key: const Key('entertainmentSection'),
4    title: 'Entertainment',
5    children: [
6      Row(
7        mainAxisAlignment: MainAxisAlignment.spaceAround,
8        children: [
9          _iconTag(
10           icon: const Icon(Icons.fastfood),
11           tag: 'Entertainment',
12           key: const Key('entertainmentFoodIcon'),
13         ),
14          _iconTag(
15           icon: const Icon(Icons.hotel),
16           tag: 'Entertainment',
17           key: const Key('entertainmentHotelIcon'),
18         ),
19          _iconTag(
20           icon: const Icon(Icons.local_grocery_store),
21           tag: 'Entertainment',
22           key: const Key('entertainmentGroceryIcon'),
23         ),
24          _iconTag(
25           icon: const Icon(Icons.local_movies),
26           tag: 'Entertainment',
27           key: const Key('entertainmentMoviesIcon'),
28         ),
29        ],
30      ),
31    ],
32  ),
33  const SizedBox(height: _kSectionSpacing),
34  _categorySection(
35    key: const Key('transportasiSection'),
36    title: 'Transportasi',
37    children: [
38      Row(
39        mainAxisAlignment: MainAxisAlignment.spaceAround,
40        children: [
41          _iconTag(
42           icon: const Icon(Icons.directions_bike),
43           tag: 'Transportasi',
44           key: const Key('transportasiBikeIcon'),
45         ),
46          _iconTag(
47           icon: const Icon(Icons.motorcycle),
48           tag: 'Transportasi',
49           key: const Key('transportasiMotorcycleIcon'),
50         ),
51          _iconTag(
52           icon: const Icon(Icons.directions_car),
53           tag: 'Transportasi',
54           key: const Key('transportasiCarIcon'),
55         ),
56          _iconTag(
57           icon: const Icon(Icons.local_shipping),
58           tag: 'Transportasi',
59           key: const Key('transportasiShippingIcon'),
60         ),
61        ],
62      ),
63      const SizedBox(height: 8.0),
64      Row(
65        mainAxisAlignment: MainAxisAlignment.spaceAround,
66        children: [
67          _iconTag(
68           icon: const Icon(Icons.directions_bus),
69           tag: 'Transportasi',
70           key: const Key('transportasiBusIcon'),
71         ),
72          _iconTag(
73           icon: const Icon(Icons.directions_boat),
74           tag: 'Transportasi',
75           key: const Key('transportasiBoatIcon'),
76         ),
77          _iconTag(
78           icon: const Icon(Icons.train),
79           tag: 'Transportasi',
80           key: const Key('transportasiTrainIcon'),
81         ),
82          _iconTag(
83           icon: const Icon(Icons.airplanemode_active),
84           tag: 'Transportasi',
85           key: const Key('transportasiAirplaneIcon'),
86         ),
87        ],
88      ),
89    ],
90  ),
91  ],
92  ),
93  ),
94  ),
95  );
96  }
97  }
98

```

Halaman Kategori:

- Menampilkan 3 kategori (General, Entertainment, Transportasi)
- Setiap kategori memiliki beberapa icon yang mewakili sub-kategori
- Saat user menekan icon, akan muncul toast message yang menunjukkan kategori yang dipilih
- Menggunakan container dan border untuk membuat tampilan card

4.4. home.dart

File home.dart menampilkan daftar berita terbaru yang diambil dari News API.

```
1 import 'package:flutter/material.dart';
2 import 'package:http/http.dart' as http;
3 import 'dart:convert';
4 import 'detail.dart';
5
6 // Constants
7 const _newsApiUrl = 'https://newsapi.org/v2/top-headlines';
8 const _apiKey = 'b8b23b65c6443d994ca378bdc09393'; // TODO: Move to .env
9 const _defaultImageUrl =
10   'https://img11-3-excelnews.com/img/06/0623/90ed8b2e4f44916f506af3a220363176598.jpg';
11 const _cardBorder = BorderSide(color: Colors.blue, width: 1.0);
12 const _cardBorderRadius = BorderRadius.all(Radius.circular(10.0));
13 const _cardPadding = EdgeInsets.all(8.0);
14 const _cardMargin = EdgeInsets.only(bottom: 16.0);
15
16 class HomePage extends StatelessWidget {
17   const HomePage({super.key});
18
19   @override
20   Widget build(BuildContext) => _HomePage();
21 }
22
23 class _HomePage extends StatelessWidget {
24   Future<List<dynamic>> getData() async {
25     try {
26       final response = await http.get(
27         Uri.parse('$newsApiUrl?sources=techcrunch&key=$_apiKey'),
28       );
29       print('response hit api : $response');
30       if (response.statusCode == 200) {
31         return json.decode(response.body)['articles'] ?? [];
32       } else {
33         throw Exception('Failed to load news: ${response.statusCode}');
34       }
35     } catch (e) {
36       throw Exception('Network error: $e');
37     }
38   }
39
40   Widget newCard({
41     required String datetime,
42     required String title,
43     required String? description,
44     required String? imageUrl,
45     required Key key,
46   }) {
47     return GestureDetector(
48       key: key,
49       onTap: () {
50         Navigator.push(
51           context,
52           MaterialPageRoute(
53             builder:
54               (context) => DetailPage(
55                 title: title,
56                 content: description ?? 'No description available',
57                 datetime: datetime,
58                 image: imageUrl ?? _defaultImageUrl,
59               ),
60             ),
61       ),
62     );
63   }
64
65   child Card(
66     elevation: 2,
67     shape: RoundedRectangleBorder(borderRadius: _cardBorderRadius),
68     margin: _cardMargin,
69     child: Padding(
70       padding: _cardPadding,
71       child: Row(
72         crossAxisAlignment: CrossAxisAlignment.start,
73         children: [
74           // Flexible
75           flex: 1,
76           child: ClipRect(
77             borderRadius: BorderRadius.circular(8.0),
78             child: Image.network(
79               imageUrl ?? _defaultImageUrl,
80               height: 100,
81               width: double.infinity,
82               fit: BoxFit.cover,
83               errorBuilder:
84                 (context, error, stackTrace) => Image.network(
85                   _defaultImageUrl,
86                   height: 100,
87                   width: double.infinity,
88                   fit: BoxFit.cover,
89                 ),
90             ),
91           const SizedBox(width: 8.0),
92           Flexible(
93             flex: 2,
94             child: Column(
95               crossAxisAlignment: CrossAxisAlignment.start,
96               children: [
97                 Text(
98                   title,
99                   style: Theme.of(context).textTheme.titleMedium?.copyWith(
100                     fontWeight: FontWeight.bold,
101                   ),
102                 ),
103                 Text(
104                   datetime,
105                   style: Theme.of(context).textTheme.bodySmall?.copyWith(
106                     fontWeight: FontWeight.w200,
107                   ),
108                 ),
109                 const SizedBox(height: 4.0),
110                 Text(
111                   description ?? 'No description available',
112                   style: Theme.of(context).textTheme.bodySmall,
113                   maxLines: 3,
114                   overflow: TextOverflow.ellipsis,
115                 ),
116               ],
117             ),
118           ),
119         ],
120       ),
121     ),
122   );
123 }
124
125 );
126
127 );
```

```

1  @override
2  Widget build(BuildContext context) {
3    return Scaffold(
4      appBar: AppBar(
5        title: const Text('Portal Berita'),
6        key: const Key('homeAppBar'),
7      ),
8      body: FutureBuilder<List<dynamic>>(
9        future: getData(),
10       builder: (context, snapshot) {
11         if (snapshot.connectionState == ConnectionState.waiting) {
12           return const Center(child: CircularProgressIndicator());
13         }
14         if (snapshot.hasError) {
15           return Center(
16             child: Column(
17               mainAxisAlignment: MainAxisAlignment.center,
18               children: [
19                 Text('Error: ${snapshot.error}'),
20                 const SizedBox(height: 16.0),
21                 ElevatedButton(
22                   onPressed: () => setState(() {}), // Retry
23                   child: const Text('Retry'),
24                 ),
25               ],
26             ),
27           );
28         }
29         if (!snapshot.hasData || snapshot.data!.isEmpty) {
30           return const Center(child: Text('No news available'));
31         }
32         return ListView.builder(
33           key: const Key('newsList'),
34           itemCount: snapshot.data!.length,
35           itemBuilder: (context, i) {
36             final article = snapshot.data![i];
37             return _newsCard(
38               key: Key('newsCard_$i'),
39               datetime: article['publishedAt'] ?? 'Unknown date',
40               title: article['title'] ?? 'No title',
41               description: article['description'],
42               imageUrl: article['urlToImage'],
43             );
44           },
45         );
46       },
47     );
48   }
49 }
50 }
51

```

Halaman Home:

- Menggunakan FutureBuilder untuk menangani operasi asynchronous fetch data dari API
- Mengimplementasikan fungsi getData() untuk mengambil artikel berita dari News API
- Menampilkan loading indicator saat data sedang diambil
- Menampilkan daftar artikel dalam bentuk ListView dengan gambar, judul, tanggal, dan deskripsi
- Setiap artikel dapat diklik untuk melihat detail di halaman DetailPage

4.5. profil.dart

File profil.dart menampilkan profil pengguna jika sudah login atau tampilan login jika belum.

```
1 import 'package:flutter/material.dart';
2 import 'package:fluttertoast/fluttertoast.dart';
3 import 'package:shared_preferences/shared_preferences.dart';
4 import 'formlogin.dart';
5 import 'formregist.dart';
6 import 'main.dart';
7
8 // Constants
9 const _kButtonPadding = EdgeInsets.symmetric(horizontal: 16.0, vertical: 12.0);
10 const _kBorderRadius = BorderRadius.all(Radius.circular(10.0));
11 const _kBorder = BorderSide(color: Colors.blue, width: 3.0);
12 const _kToastDuration = Duration(seconds: 2);
13
14 class ProfilePage extends StatefulWidget {
15   const ProfilePage({super.key});
16
17   @override
18   _ProfilePageState createState() => _ProfilePageState();
19 }
20
21 class _ProfilePageState extends State<ProfilePage> {
22   late Future<String> _loginFuture;
23   late FToast fToast;
24
25   @override
26   void initState() {
27     super.initState();
28     _loginFuture = getUsername();
29     fToast = FToast();
30   }
31
32   @override
33   void didChangeDependencies() {
34     super.didChangeDependencies();
35     fToast.init(context); // Initialize FToast with context
36   }
37
38   Future<String> getUsername() async {
39     try {
40       final prefs = await SharedPreferences.getInstance();
41       return prefs.getString('username');
42     } catch (e) {
43       fToast.showToast(
44         toastDuration: _kToastDuration,
45         child: const Text('Error accessing preferences'),
46       );
47       return null;
48     }
49   }
50
51   Future<void> logout() async {
52     try {
53       final prefs = await SharedPreferences.getInstance();
54       await prefs.remove('username');
55       await prefs.remove('token');
56       fToast.showToast(
57         toastDuration: _kToastDuration,
58         child: const Text('Berhasil logout'),
59       );
60       // Reset navigation stack to ProfilePage
61       Navigator.pushAndRemoveUntil(
62         context,
63         MaterialPageRoute(builder: (_) => const MyApp(initialPage: 0)),
64         (route) => false,
65       );
66     } catch (e) {
67       fToast.showToast(
68         toastDuration: _kToastDuration,
69         child: const Text('Error during logout'),
70       );
71     }
72   }
73 }
```

```

1  Widget _belumlogin() {
2    return Column(
3      mainAxisAlignment: MainAxisAlignment.center,
4      children: [
5        const Icon(
6          Icons.account_circle,
7          size: 150,
8          color: Colors.blue,
9          key: Key('profileIcon'),
10       ),
11       const SizedBox(height: 16.0),
12       ElevatedButton(
13         key: const Key('loginButton'),
14         style: ElevatedButton.styleFrom(
15           backgroundColor: Colors.blue,
16           padding: _kButtonPadding,
17         ),
18         onPressed: () {
19           Navigator.push(
20             context,
21             MaterialPageRoute(builder: (_) => LoginPage()),
22           );
23         },
24         child: const Text('Masuk', style: TextStyle(color: Colors.white)),
25       ),
26       const SizedBox(height: 8.0),
27       Row(
28         mainAxisAlignment: MainAxisAlignment.center,
29         children: [
30           const Text('Belum memiliki akun? '),
31           TextButton(
32             key: const Key('registerButton'),
33             onPressed: () {
34               Navigator.push(
35                 context,
36                 MaterialPageRoute(builder: (_) => RegisterPage()),
37               );
38             },
39           child: const Text(
40             'Create Account',
41             style: TextStyle(
42               decoration: TextDecoration.underline,
43               color: Colors.blue,
44             ),
45           ),
46         ],
47       ),
48     ],
49   );
50 }
51 }

```

```

1  Widget _sudahLogin(String username) {
2    return Column(
3      mainAxisAlignment: MainAxisAlignment.center,
4      children: [
5        const Icon(
6          Icons.account_circle,
7          size: 150,
8          color: Colors.blue,
9          key: Key('profileIconLoggedIn'),
10       ),
11       const SizedBox(height: 16.0),
12       Container(
13         key: const Key('usernameContainer'),
14         padding: const EdgeInsets.all(10),
15         decoration: BoxDecoration(
16           border: Border.all(color: _kBorder.color, width: _kBorder.width),
17           borderRadius: _kBorderRadius,
18         ),
19         child: Text(username, style: Theme.of(context).textTheme.titleLarge),
20       ),
21       const SizedBox(height: 16.0),
22       IconButton(
23         key: const Key('logoutButton'),
24         icon: const Icon(Icons.exit_to_app),
25         color: Colors.blue,
26         tooltip: 'Logout',
27         onPressed: logOut,
28       ),
29     ],
30   );
31 }
32
33 @override
34 Widget build(BuildContext context) {
35   return Scaffold(
36     body: FutureBuilder<String?>(
37       future: _loginFuture,
38       builder: (context, snapshot) {
39         if (snapshot.connectionState == ConnectionState.waiting) {
40           return const Center(child: CircularProgressIndicator());
41         }
42         if (snapshot.hasError) {
43           return const Center(child: Text('Error loading profile'));
44         }
45         final username = snapshot.data;
46         return Center(
47           child: username != null ? _sudahLogin(username) : _belumLogin(),
48         );
49       ),
50     ),
51   );
52 }
53 }
54

```

Halaman Profil:

- Menggunakan SharedPreferences untuk menyimpan dan mengambil data login
- Memiliki dua tampilan berbeda: tampilan untuk user yang belum login dan yang sudah login
- Jika belum login, menampilkan tombol untuk login dan link untuk registrasi
- Jika sudah login, menampilkan username dan tombol logout
- Menggunakan FutureBuilder untuk menangani operasi asynchronous pengambilan data login

4.6. formlogin.dart

File formlogin.dart menampilkan formulir login untuk pengguna.

```

1 import 'package:flutter/material.dart';
2 import 'package:shared_preferences/shared_preferences.dart';
3 import 'package:fluttertoast/fluttertoast.dart';
4 import 'package:http/http.dart' as http;
5 import 'dart:convert';
6 import 'main.dart';
7
8 class LoginPage extends StatefulWidget {
9   const LoginPage({super.key});
10
11   @override
12   _LoginPageState createState() => _LoginPageState();
13 }
14
15 class _LoginPageState extends State<LoginPage> {
16   final TextEditingController userController = TextEditingController();
17   final TextEditingController passController = TextEditingController();
18   late FToast ftoast;
19   bool isLoading = false;
20
21   @override
22   void initState() {
23     super.initState();
24     ftoast = FToast()..init(context);
25   }
26
27   Future<void> loginUser(String username, String password) async {
28     setState(() => isLoading = true);
29     final url = Uri.parse("https://e-commerce-store.glitch.me/login");
30     try {
31       final response = await http.post(
32         url,
33         headers: {"Content-Type": "application/json"},
34         body: jsonEncode({"username": username, "password": password}),
35       );
36       final statusCode = response.statusCode;
37       final body = jsonDecode(response.body);
38       if (statusCode == 200 && body['status'] == true) {
39         final user = body['data']['user'];
40         final token = body['data']['token'];
41         SharedPreferences prefs = await SharedPreferences.getInstance();
42         await prefs.setString('username', user['username']);
43         await prefs.setString('token', token);
44         showToast("Login berhasil", isSuccess: true);
45         Navigator.pushReplacement(
46           context,
47           MaterialPageRoute(builder: (_) => MyApp(initialPage: 0)),
48         );
49       } else {
50         showToast(
51           "Login gagal: ${body['message'] ?? 'Username atau password salah'}",
52         );
53       }
54     } catch (e) {
55       showToast("Kesalahan jaringan: $e");
56     } finally {
57       setState(() => isLoading = false);
58     }
59   }
60
61   void showToast(String message, {bool isSuccess = false}) {
62     ftoast.showToast(
63       gravity: ToastGravity.BOTTOM,
64       toastDuration: Duration(seconds: 2),
65       child: Container(
66         padding: EdgeInsets.symmetric(horizontal: 24.0, vertical: 12.0),
67         decoration: BoxDecoration(
68           borderRadius: BorderRadius.circular(25.0),
69           color: isSuccess ? Colors.green : Colors.redAccent,
70         ),
71         child: Text(message, style: TextStyle(color: Colors.white)),
72       ),
73     );
74   }
75 }

```



```

1 @override
2 Widget build(BuildContext context) {
3   return Scaffold(
4     body: Center(
5       child: SingleChildScrollView(
6         padding: EdgeInsets.all(20),
7         child: Column(
8           mainAxisAlignment: MainAxisAlignment.center,
9           children: [
10            Text(
11              'login',
12              style: TextStyle(fontSize: 30, fontWeight: FontWeight.bold),
13            ),
14            SizedBox(height: 20),
15            TextField(
16              controller: userController,
17              decoration: InputDecoration(labelText: "Username"),
18            ),
19            SizedBox(height: 10),
20            TextField(
21              controller: passController,
22              obscureText: true,
23              decoration: InputDecoration(labelText: "Password"),
24            ),
25            SizedBox(height: 20),
26            isLoading
27              ? CircularProgressIndicator()
28              : Row(
29                mainAxisAlignment: MainAxisAlignment.center,
30                children: [
31                  ElevatedButton(
32                    onPressed: () => Navigator.pop(context),
33                    style: ElevatedButton.styleFrom(
34                      backgroundColor: Colors.grey,
35                      padding: EdgeInsets.symmetric(
36                        horizontal: 20,
37                        vertical: 12,
38                      ),
39                    ),
40                  child: Text("Back"),
41                ],
42              ),
43            SizedBox(width: 20),
44            ElevatedButton(
45              onPressed: () {
46                final username = userController.text.trim();
47                final password = passController.text.trim();
48                if (username.isEmpty || password.isEmpty) {
49                  showToast("Harap isi semua kolom");
50                } else {
51                  loginUser(username, password);
52                }
53              },
54              style: ElevatedButton.styleFrom(
55                backgroundColor: Colors.blue,
56                padding: EdgeInsets.symmetric(
57                  horizontal: 20,
58                  vertical: 12,
59                ),
60            ),
61            child: Text("Login"),
62          ],
63        ),
64      ),
65    ),
66  ),
67  ),
68  ),
69  ),
70  ),
71 }
72 }
73 }

```

Halaman Login:

- Menggunakan TextEditingController untuk menangani input pengguna
- Implementasi fungsi loginUser() untuk proses autentikasi ke API
- Menampilkan indikator loading selama proses login
- Menggunakan SharedPreferences untuk menyimpan data login jika berhasil
- Menampilkan toast message untuk notifikasi keberhasilan atau kegagalan login

4.7. formregist.dart

File formregist.dart menampilkan formulir registrasi untuk pengguna baru.

```

1 import 'package:flutter/material.dart';
2 import 'package:http/http.dart' as http;
3 // import 'package:testing_flutter/home.dart';
4
5 class RegisterPage extends StatefulWidget {
6   const RegisterPage({super.key});
7
8   @override
9   _RegisterPageState createState() => _RegisterPageState();
10 }
11
12 class _RegisterPageState extends State<RegisterPage> {
13   TextEditingController userController = TextEditingController();
14   TextEditingController passController = TextEditingController();
15   TextEditingController pass2Controller = TextEditingController();
16   // Fungsi untuk menambahkan data pengguna
17   addDataUser(
18     TextEditingController userController,
19     TextEditingController passController,
20     TextEditingController pass2Controller,
21   ) async {
22     // Validasi input pengguna
23     if (userController.text != '' &&
24         passController.text != '' &&
25         pass2Controller.text != '' &&
26         passController.text == pass2Controller.text) {
27       setState(() {
28         var link = Uri.parse(
29           'https://e-commerce-store.glitch.me/signup',
30         ); // Menggunakan Uri.parse()
31         // Mengirim data menggunakan http.post
32         http
33           .post(
34             link,
35             body: {
36               "username": userController.text,
37               "password": passController.text,
38             },
39           )
40           .then((response) {
41             // Cek apakah registrasi berhasil berdasarkan status code
42             if (response.statusCode == 200) {
43               // Tindakan jika berhasil
44               _showErrorToasts("Registrasi berhasil.");
45               return Navigator.pop(context);
46             } else {
47               // Tindakan jika gagal
48               _showErrorToasts("Registrasi gagal. Coba lagi.");
49             }
50           })
51           .catchError((e) {
52             // Menangani kesalahan jaringan
53             _showErrorToasts(
54               "Terjadi kesalahan. Pastikan Anda terhubung ke internet.",
55             );
56           });
57       });
58     } else {
59       // Kosongkan form jika validasi gagal
60       userController.text = '';
61       passController.text = '';
62       pass2Controller.text = '';
63       _showErrorToasts("Pastikan semua kolom terisi dengan benar.");
64     }
65   }
66 }

```

```

1 // Menampilkan pesan toast jika ada error
2 void _showErrorToast(String message) {
3   ScaffoldMessenger.of(context).showSnackBar(
4     SnackBar(content: Text(message), duration: Duration(seconds: 3)),
5   );
6 }
7
8 void _showSuccessToast(String message) {
9   ScaffoldMessenger.of(context).showSnackBar(
10    SnackBar(content: Text(message), duration: Duration(seconds: 3)),
11  );
12 }
13
14 @override
15 Widget build(BuildContext context) {
16   return SafeArea(
17     child: Scaffold(
18       body: Center(
19         child: Container(
20           margin: EdgeInsets.all(20),
21           child: Column(
22             mainAxisAlignment: MainAxisAlignment.center,
23             children: [
24               Text(
25                 "Registrasi",
26                 style: TextStyle(fontWeight: FontWeight.bold, fontSize: 30),
27               ),
28               SizedBox(height: 20),
29               TextField(
30                 controller: userController,
31                 decoration: InputDecoration(hintText: "Username"),
32               ),
33               TextField(
34                 controller: passController,
35                 decoration: InputDecoration(hintText: "Password"),
36                 obscureText: true,
37               ),
38               TextField(
39                 controller: pass2Controller,
40                 decoration: InputDecoration(hintText: "Konfirmasi Password"),
41                 obscureText: true,
42               ),
43               SizedBox(height: 20),
44               Row(
45                 mainAxisAlignment: MainAxisAlignment.center,
46                 children: [
47                   TextButton(
48                     style: TextButton.styleFrom(
49                       backgroundColor: Colors.blue,
50                       shape: RoundedRectangleBorder(
51                         borderRadius: BorderRadius.circular(10.0),
52                       ),
53                     ),
54                     padding: EdgeInsets.symmetric(
55                       horizontal: 16.0,
56                       vertical: 12.0,
57                     ),
58                     onPressed: () {
59                       Navigator.pop(context);
60                     },
61                     child: Text(
62                       "Back",
63                       style: TextStyle(color: Colors.white),
64                     ),
65                   ),
66                   SizedBox(width: 20),
67                   TextButton(
68                     style: TextButton.styleFrom(
69                       backgroundColor: Colors.blue,
70                       shape: RoundedRectangleBorder(
71                         borderRadius: BorderRadius.circular(10.0),
72                       ),
73                     ),
74                     padding: EdgeInsets.symmetric(
75                       horizontal: 16.0,
76                       vertical: 12.0,
77                     ),
78                     onPressed: () {
79                       addUser(
80                         userController,
81                         passController,
82                         pass2Controller,
83                       );
84                     },
85                     child: Text(
86                       "Sign Up",
87                       style: TextStyle(color: Colors.white),
88                     ),
89                   ),
90                 ],
91               ),
92             ],
93           ),
94         ),
95       ),
96     );
97 }
98
99 }

```

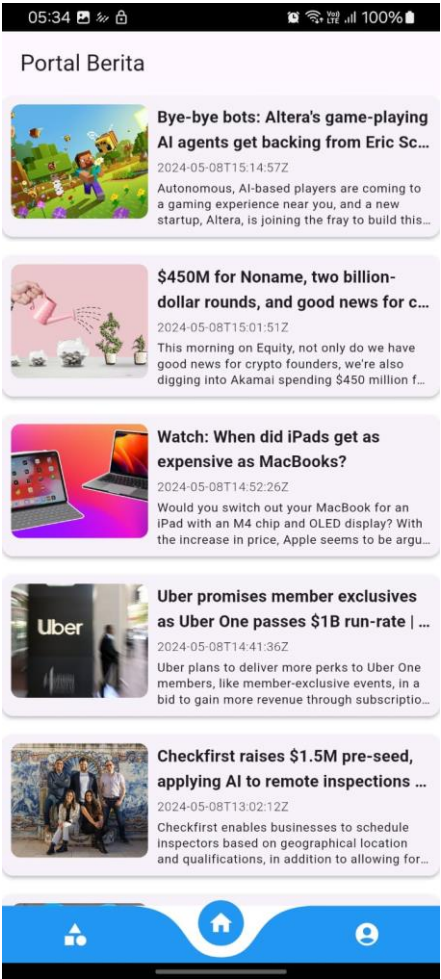
Halaman Register:

- Input: Menggunakan TextEditingController untuk username, password, dan konfirmasi password (dengan obscureText).
- Validasi: Memastikan semua field terisi dan password cocok sebelum kirim ke API.
- API Request: POST data ke endpoint signup menggunakan http.post, cek statusCode untuk respons berhasil/gagal.
- Feedback: Tampilkan SnackBar (toast) untuk notifikasi sukses/gagal registrasi atau error jaringan.
- Navigasi: Kembali ke halaman sebelumnya setelah registrasi berhasil (Navigator.pop).
- UI: Desain sederhana dengan tombol biru dan teks putih.

BAB V

HASIL INTERFACE

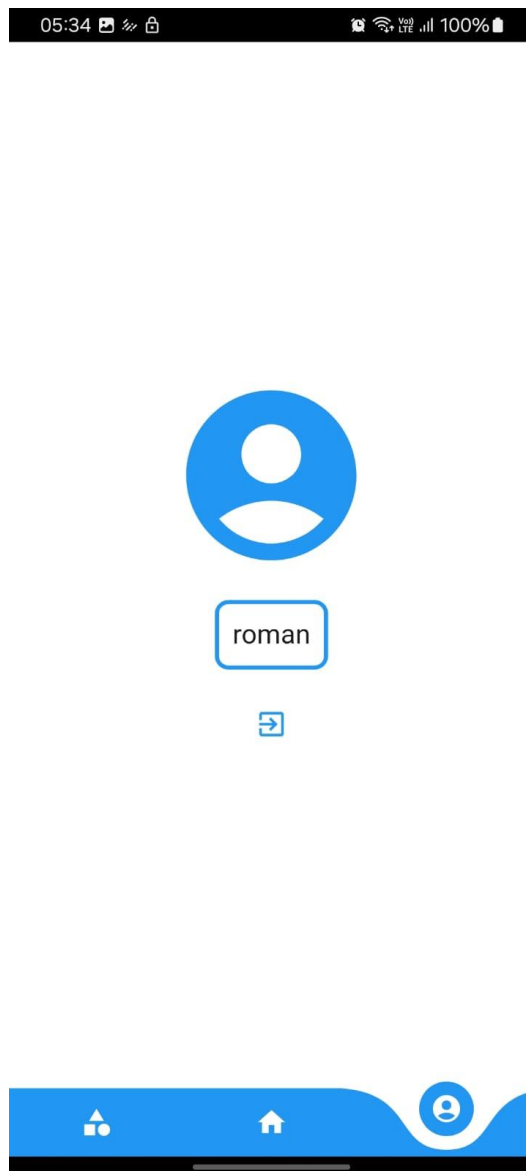
5.1. Home



5.2. Kategori



5.3. Profile



5.4. Register

05:34 100%

Registrasi

roman




.....

.....

Back Sign Up

5.5. Login

05:34

 100%

Login

Username

roman

Password

.....|

Back

Login