



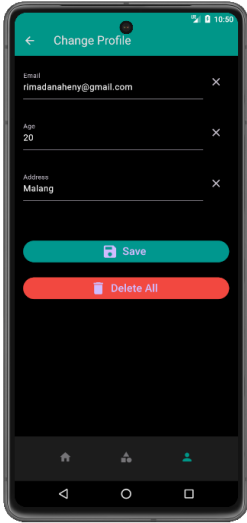
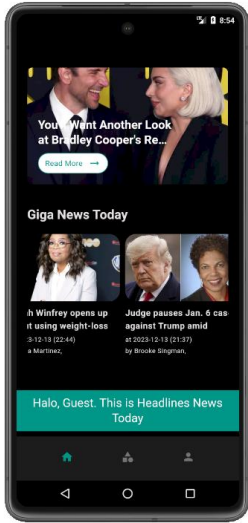
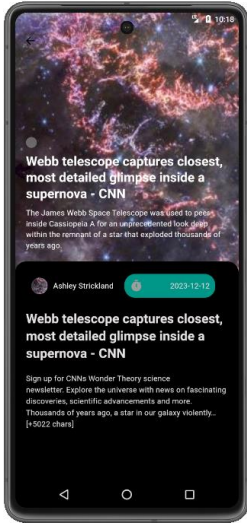
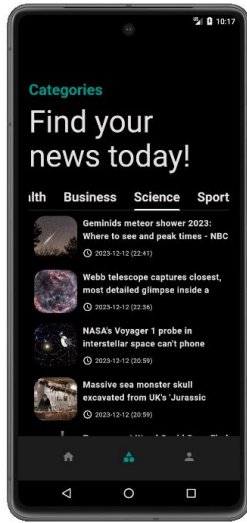
**LAPORAN UJIAN AKHIR SEMESTER PEMROGRAMAN MOBILE**  
**SEMESTER GANJIL 2023/2024**  
**APLIKASI BERITA: GIGA NEWS**



Disusun oleh:  
Nama: Heny Rimadana  
NIM: 210605110009

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM**  
**MALANG**  
**2023**

## Desain UI Aplikasi Berita : Giga News

Login Page	Profile Page	Change Profile
		
Home Page	Detail Page	Categories Page
		

### A. Model

Pada bagian ini akan dibahas tentang model data yang digunakan dalam aplikasi.

Data yang diperlukan dalam aplikasi ini meliputi:

1. Top Headlines
2. Kategori Berita

Masing-masing data tersebut mempunyai bentuk class data object Dart sebagai berikut:

## 1. Top Headlines

1	import 'dart:convert';
2	
3	import 'package:http/http.dart';
4	
5	class ApiDatas {
6	Map<String, String> headers = {
7	'Content-Type': 'application/json;charset=UTF-8',
8	'Charset': 'utf-8'
9	};
10	Future<Map> getData() async {
11	Response response = await get(Uri.parse(
12	" https://newsapi.org/v2/top-headlines?country=us&apiKey=1c862e10c5e74882ab88901cf1aeb027
13	),headers: headers);
14	Map data = jsonDecode(response.body);
15	if (data.isNotEmpty) {
16	print(data);
17	return data;
18	} else {
19	print(response.statusCode.toString());
20	return {};
21	}
22	}
23	import 'package:equatable/equatable.dart';
24	class Article extends Equatable {
25	final String id;
26	final String title;
27	final String subtitle;
28	final String body;
29	final String author;
30	final String authorImageUrl;
31	final String category;
32	final String imageUrl;
33	final int views;
34	final DateTime createdAt;
35	
36	static List<Article> articles = [];
37	
38	const Article(
39	{required this.id,
40	required this.title,
41	required this.subtitle,
42	required this.body,
43	required this.author,
44	required this.authorImageUrl,
45	required this.category,
46	required this.imageUrl,
47	required this.views,
48	required this.createdAt});
49	
50	List<Object> get props => [];
51	}

Model data Top Headline yang ditampilkan dalam kode tersebut mencakup dua bagian utama:

### 1. ApiDatas Class (API Service):

- Merepresentasikan pengambilan data dari NewsAPI menggunakan metode **getData**.
- Memanfaatkan paket **http** untuk membuat permintaan HTTP ke NewsAPI dengan mengirimkan permintaan untuk top headlines dengan kriteria tertentu (dalam hal ini, berita dari Amerika Serikat).
- Data yang diterima dari API di-decode dari JSON ke dalam **Map** dan kemudian dikembalikan untuk digunakan di aplikasi.

## 2. Article Class (Model Data):

- Menggunakan paket **equatable** untuk membuat model data yang menyimpan informasi tentang artikel.
- Setiap **Article** memiliki properti seperti ID, judul, subjudul, isi, penulis, URL gambar, kategori, jumlah views, dan tanggal dibuat.
- Menyediakan fungsi **props** yang kosong karena **Equatable** membutuhkan perubahan pada properti untuk mempengaruhi kesamaan objek.

Kedua bagian ini menyajikan struktur dasar untuk mengambil data dari NewsAPI dan merepresentasikan artikel dalam aplikasi. Kelas **ApiDatas** bertanggung jawab atas pengambilan data dari sumber eksternal, sementara kelas **Article** berperan sebagai wadah untuk menyimpan informasi artikel. Dengan bantuan kelas ini, aplikasi dapat mengonsumsi data dari NewsAPI dengan cara yang terstruktur dan dapat digunakan di seluruh aplikasi.

## 2. Kategori Berita

1	import 'dart:convert';
2	
3	import 'package:http/http.dart';
4	
5	class ApiDatas {
6	Future<Map> getDataByCategory(String category) async {
7	Response response = await get(Uri.parse(
8	"https://newsapi.org/v2/top-headlines?apiKey=1c862e10c5e74882ab88901cf1aeb027&country=us&category=\$category")
9	);
10	Map data = jsonDecode(response.body);
11	if (data.isNotEmpty) {
12	print(data);
13	return data;
14	} else {
15	print(response.statusCode.toString());
16	return {};

16	}
17	}
18	Future<Map> getDataByCategory(String category) async {
19	Response response = await get(Uri.parse(
20	"https://newsapi.org/v2/top-headlines?apiKey=2142e09042164d01be5d63e97dcd0dc6&country=us&category=\$category");
21	Map data = jsonDecode(response.body);
22	import 'package:equatable/equatable.dart';
23	
24	class Article extends Equatable {
25	final String id;
26	final String title;
27	final String subtitle;
28	final String body;
29	final String author;
30	final String authorImageUrl;
31	final String category;
32	final String imageUrl;
33	final int views;
34	final DateTime createdAt;
35	
36	static List<Article> articles = [];
37	
38	const Article(
39	{required this.id,
40	required this.title,
41	required this.subtitle,
42	required this.body,
43	required this.author,
44	required this.authorImageUrl,
45	required this.category,
46	required this.imageUrl,
47	required this.views,
48	required this.createdAt});
49	
50	List<Object> get props => [];
51	}

Bagian utama yang terkait dengan model data kategori berita yakni:

#### 1. **Kelas ApiDatas:**

- **Deskripsi:** Kelas ini bertanggung jawab untuk mengambil data dari API NewsAPI.
- **Fungsi getDataByCategory(String category):** Method ini digunakan untuk mengambil data berita berdasarkan kategori yang ditentukan dari NewsAPI. Metode ini memanfaatkan HTTP request **get** untuk mendapatkan respons dari URL yang ditentukan dengan kategori yang disediakan. Data yang diterima diubah dari format JSON ke dalam bentuk **Map**. Jika respons tidak kosong, data ditampilkan dalam bentuk **Map**, jika tidak, akan dikembalikan sebuah map kosong.

- **Headers:** Kelas ini juga memiliki konfigurasi **headers** yang digunakan dalam setiap permintaan ke API, menentukan tipe konten dan pengkodean karakter.

## 2. **Kelas Article:**

- **Deskripsi:** Kelas ini merupakan model data yang merepresentasikan setiap artikel berita.
- **Properti Artikel:**
  - **id:** ID unik untuk setiap artikel.
  - **title:** Judul artikel.
  - **subtitle:** Subjudul artikel.
  - **body:** Isi atau konten artikel.
  - **author:** Nama penulis artikel.
  - **authorImageUrl:** URL gambar penulis.
  - **category:** Kategori artikel.
  - **imageUrl:** URL gambar artikel.
  - **views:** Jumlah views atau tampilan artikel.
  - **createdAt:** Tanggal dan waktu pembuatan artikel.
- **Static List<Article> articles:** Menyimpan daftar artikel yang dibuat dari objek **Article**. List ini digunakan untuk menyimpan koleksi artikel yang telah dimuat atau diambil dari API.

## **B. View**

Pada bagian ini akan dibahas kode program yang memanfaatkan widget flutter untuk menampilkan data hasil request dari server dengan API. Dimana pada aplikasi ini view dibagi menjadi beberapa bagian ialah:

1. Login
2. Profile
3. Change Profile
4. Home
5. Kategori
6. Detail

Adapun masing-masing view tersebut mempunyai kode program sebagai berikut:

## 1. Login

1.	import 'package:flutter/material.dart';
2.	import 'package:shared_preferences/shared_preferences.dart';
3.	import 'package:tds_pemrograman_mobile/screens/home_screen.dart';
4.	import 'package:tds_pemrograman_mobile/screens/profile.dart';
5.	import 'package:tds_pemrograman_mobile/services/fetchapi.dart';
6.	import './widgets/custom_tag.dart';
7.	import 'package:google_fonts/google_fonts.dart';
8.	
9.	class LoginScreen extends StatelessWidget {
10.	LoginScreen({super.key});
11.	
12.	static const routeName = '/login';
13.	
14.	final TextEditingController usernameController = TextEditingController();
15.	final TextEditingController passwordController = TextEditingController();
16.	
17.	final Map<String, String> users = {
18.	"heny": "heny123",
19.	
20.	// Tambahkan username dan password tambahan di sini
21.	};
22.	
23.	@override
24.	Widget build(BuildContext context) {
25.	return Scaffold(
26.	body: SingleChildScrollView(
27.	child: Column(
28.	children: [
29.	Padding(padding: EdgeInsets.only(top: 50)),
30.	
31.	Image.asset('assets/images/giganews-login.png'),
32.	
33.	// Image.asset(""),
34.	Text('SIGN IN',
35.	style: GoogleFonts.inter(
36.	fontSize: 20,
37.	color: Colors.orangeAccent,
38.	fontWeight: FontWeight.bold),
39.	SizedBox(height: 30),
40.	Text(
41.	'Username',
42.	style: TextStyle(fontSize: 15),
43.	),
44.	SizedBox(height: 5),
45.	Padding(
46.	padding: EdgeInsets.symmetric(horizontal: 40),
47.	child: TextField(
48.	maxLength: 7,
49.	controller: usernameController,
50.	style: TextStyle(color: Color.fromARGB(255, 255, 255, 255)),
51.	decoration: InputDecoration(
52.	labelStyle:
53.	TextStyle(color: Color.fromARGB(255, 255, 255, 255)),
54.	enabledBorder: OutlineInputBorder(
55.	borderSide: BorderSide(color: Colors.grey),
56.	borderRadius: BorderRadius.circular(10),
57.	),
58.	focusedBorder: OutlineInputBorder(
59.	borderSide: BorderSide(color: Colors.tealAccent),
60.	borderRadius: BorderRadius.circular(10),

61.	),
62.	),
63.	),
64.	),
65.	SizedBox(height: 20),
66.	Text(
67.	'Password',
68.	style: TextStyle(fontSize: 15),
69.	),
70.	SizedBox(height: 5),
71.	Padding(
72.	padding: EdgeInsets.symmetric(horizontal: 40),
73.	child: TextField(
74.	obscureText: true,
75.	controller: _passwordController,
76.	style: TextStyle(color: Color.fromARGB(255, 255, 255, 255)),
77.	decoration: InputDecoration(
78.	labelStyle: TextStyle(color: Colors.white),
79.	enabledBorder: OutlineInputBorder(
80.	borderSide: BorderSide(color: Colors.grey),
81.	borderRadius: BorderRadius.circular(10),
82.	),
83.	focusedBorder: OutlineInputBorder(
84.	borderSide: BorderSide(color: Colors.teal),
85.	borderRadius: BorderRadius.circular(10),
86.	),
87.	),
88.	),
89.	),
90.	SizedBox(height: 30),
91.	ElevatedButton(
92.	style: ElevatedButton.styleFrom(
93.	padding: EdgeInsets.all(16.0),
94.	primary: Color(0xFF0093A9),
95.	shape: RoundedRectangleBorder(
96.	borderRadius: BorderRadius.circular(10),
97.	),
98.	),
99.	onPressed: () async {
100.	String enteredUsername = _usernameController.text;
101.	String enteredPassword = _passwordController.text;
102.	
103.	if (users.containsKey(enteredUsername) &&
104.	users[enteredUsername] == enteredPassword) {
105.	final prefs = await SharedPreferences.getInstance();
106.	await prefs.setString('username', enteredUsername);
107.	await prefs.setString('password', enteredPassword);
108.	
109.	Navigator.pushReplacementNamed(
110.	context,
111.	Profile.routeName,
112.	);
113.	} else {
114.	ScaffoldMessenger.of(context).showSnackBar(
115.	SnackBar(
116.	backgroundColor: Color.fromARGB(255, 255, 15, 15),
117.	content: Text(
118.	'Incorrect Username or Password',
119.	style: TextStyle(
120.	color: Colors
121.	.white), // Mengubah warna teks menjadi putih
122.	),



123.	),
124.	);
125.	}
126.	},
127.	child: Text(
128.	'LOGIN',
129.	style: TextStyle(
130.	color: const Color.fromARGB(255, 0, 0, 0),
131.	letterSpacing: 2,
132.	fontSize: 15,
133.	fontWeight: FontWeight.bold,
134.	),
135.	),
136.	),
137.	],
138.	),
139.	),
140.	);
141.	}

## 2. Profile

1.	import 'package:flutter/material.dart';
2.	import 'package:shared_preferences/shared_preferences.dart';
3.	import 'package:tgs_pemrograman_mobile/screens/change_profile.dart';
4.	import 'package:tgs_pemrograman_mobile/widgets/bottom_nav_bar.dart';
5.	
6.	class Profile extends StatefulWidget {
7.	static const routeName = '/profile';
8.	
9.	@override
10.	_ProfileState createState() => _ProfileState();
11.	}
12.	
13.	class _ProfileState extends State<Profile> {
14.	late SharedPreferences _prefs;
15.	late String _email = '';
16.	late String _age = '';
17.	late String _address = '';
18.	late String _username = '';
19.	late String _password = '';
20.	
21.	@override
22.	void initState() {
23.	super.initState();
24.	_loadProfileData();
25.	// _clearProfileData();
26.	}
27.	
28.	@override
29.	void didChangeDependencies() {
30.	super.didChangeDependencies();
31.	_loadProfileData(); // Memuat data setiap kali ada perubahan dependencies (widget ini)
32.	}
33.	
34.	Future<void> _loadProfileData() async {
35.	_prefs = await SharedPreferences.getInstance();
36.	setState(() {
37.	_email = _prefs.getString('email') ?? '';
38.	_age = _prefs.getString('age') ?? '';
39.	_address = _prefs.getString('address') ?? '';
40.	_username = _prefs.getString('username') ?? '';
41.	_password = _prefs.getString('password') ?? '';

42.	});
43.	}
44.	
45.	Widget _buildProfileCard(String title, String value) {
46.	return Card(
47.	margin: EdgeInsets.symmetric(vertical: 9, horizontal: 16),
48.	child: Padding(
49.	padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 16),
50.	child: Row(
51.	mainAxisAlignment: MainAxisAlignment.spaceBetween,
52.	children: [
53.	Text(
54.	title,
55.	style: TextStyle(
56.	fontWeight: FontWeight.bold,
57.	fontSize: 20,
58.	),
59.	),
60.	Flexible(
61.	child: title == 'Password'
62.	? Text(
63.	'*' * value.length,
64.	style: TextStyle(fontSize: 16),
65.	)
66.	: Text(
67.	value,
68.	style: TextStyle(fontSize: 16),
69.	),
70.	),
71.	],
72.	),
73.	),
74.	);
75.	}
76.	
77.	@override
78.	Widget build(BuildContext context) {
79.	return Scaffold(
80.	appBar: AppBar(
81.	title: Text('Profile'),
82.	),
83.	body: ListView(
84.	padding: EdgeInsets.all(15.0),
85.	children: [
86.	Center(
87.	child: CircleAvatar(
88.	radius: 70,
89.	backgroundImage: AssetImage('assets/images/developer.jpeg'),
90.	),
91.	),
92.	SizedBox(height: 30),
93.	_buildProfileCard('Username', _username),
94.	_buildProfileCard('Password', _password),
95.	_buildProfileCard('Email', _email),
96.	_buildProfileCard('Umur', _age),
97.	_buildProfileCard('Alamat', _address),
98.	SizedBox(height: 20),
99.	Center(
100.	child: ElevatedButton(
101.	onPressed: () {
102.	Navigator.push(
103.	context,

104	MaterialPageRoute(builder: (context) => ChangeProfile()),
105	);
106	},
107	child: Text(
108	'Ubah Data',
109	style: TextStyle(
110	color: Colors.orange, // Mengatur warna teks menjadi oranye
111	),
112	),
113	),
114	),
115	],
116	),
117	bottomNavigationBar: BottomNavBar(index: 2),
118	);
119	}
120	}
121	import 'package:flutter/material.dart';
122	import 'package:shared_preferences/shared_preferences.dart';
123	import 'package:tgs_pemrograman_mobile/screens/change_profile.dart';
124	import 'package:tgs_pemrograman_mobile/widgets/bottom_nav_bar.dart';
125	
126	class Profile extends StatefulWidget {
127	static const routeName = '/profile';
128	
129	@override
130	_ProfileState createState() => _ProfileState();

### 3. Change Profile

1.	import 'package:flutter/material.dart';
2.	import 'package:shared_preferences/shared_preferences.dart';
3.	import 'package:tgs_pemrograman_mobile/screens/change_profile.dart';
4.	import 'package:tgs_pemrograman_mobile/widgets/bottom_nav_bar.dart';
5.	
6.	class Profile extends StatefulWidget {
7.	static const routeName = '/profile';
8.	
9.	@override
10.	_ProfileState createState() => _ProfileState();
11.	}
12.	
13.	class _ProfileState extends State<Profile> {
14.	late SharedPreferences _prefs;
15.	late String _email = '';
16.	late String _age = '';
17.	late String _address = '';
18.	late String _username = '';
19.	late String _password = '';
20.	
21.	@override
22.	void initState() {
23.	super.initState();
24.	_loadProfileData();
25.	// _clearProfileData();
26.	}
27.	
28.	@override
29.	void didChangeDependencies() {
30.	super.didChangeDependencies();
31.	_loadProfileData(); // Memuat data setiap kali ada perubahan dependencies (widget ini)

32.	}
33.	
34.	Future<void> _loadProfileData() async {
35.	_prefs = await SharedPreferences.getInstance();
36.	setState(() {
37.	_email = _prefs.getString('email') ?? '';
38.	_age = _prefs.getString('age') ?? '';
39.	_address = _prefs.getString('address') ?? '';
40.	_username = _prefs.getString('username') ?? '';
41.	_password = _prefs.getString('password') ?? '';
42.	});
43.	}
44.	
45.	Widget _buildProfileCard(String title, String value) {
46.	return Card(
47.	margin: EdgeInsets.symmetric(vertical: 9, horizontal: 16),
48.	child: Padding(
49.	padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 16),
50.	child: Row(
51.	mainAxisAlignment: MainAxisAlignment.spaceBetween,
52.	children: [
53.	Text(
54.	title,
55.	style: TextStyle(
56.	fontWeight: FontWeight.bold,
57.	fontSize: 20,
58.	),
59.	),
60.	Flexible(
61.	child: title == 'Password'
62.	? Text(
63.	'*' * value.length,
64.	style: TextStyle(fontSize: 16),
65.	)
66.	: Text(
67.	value,
68.	style: TextStyle(fontSize: 16),
69.	),
70.	),
71.	],
72.	),
73.	),
74.	);
75.	}
76.	
77.	@override
78.	Widget build(BuildContext context) {
79.	return Scaffold(
80.	appBar: AppBar(
81.	title: Text('Profile'),
82.	),
83.	body: ListView(
84.	padding: EdgeInsets.all(15.0),
85.	children: [
86.	Center(
87.	child: CircleAvatar(
88.	radius: 70,
89.	backgroundImage: AssetImage('assets/images/developer.jpeg'),
90.	),
91.	),
92.	SizedBox(height: 30),
93.	_buildProfileCard('Username', _username),

94.	_buildProfileCard('Password', _password),
95.	_buildProfileCard('Email', _email),
96.	_buildProfileCard('Umur', _age),
97.	_buildProfileCard('Alamat', _address),
98.	SizedBox(height: 20),
99.	Center(
100.	child: ElevatedButton(
101.	onPressed: () {
102.	Navigator.push(
103.	context,
104.	MaterialPageRoute(builder: (context) => ChangeProfile()),
105.	);
106.	),
107.	child: Text(
108.	'Ubah Data',
109.	style: TextStyle(
110.	color: Colors.orange, // Mengatur warna teks menjadi oranye
111.	),
112.	),
113.	),
114.	),
115.	],
116.	),
117.	bottomNavigationBar: BottomNavBar(index: 2),
118.	);
119.	}
120.	}
121.	import 'package:flutter/material.dart';
122.	import 'package:shared_preferences/shared_preferences.dart';
123.	import 'package:tgs_pemrograman_mobile/screens/change_profile.dart';
124.	import 'package:tgs_pemrograman_mobile/widgets/bottom_nav_bar.dart';
125.	
126.	class Profile extends StatefulWidget {
127.	static const routeName = '/profile';
128.	
129.	@override
130.	_ProfileState createState() => _ProfileState();

#### 4. Home

1.	import 'package:flutter/material.dart';
2.	import 'package:tgs_pemrograman_mobile/models/article_model.dart';
3.	import 'package:tgs_pemrograman_mobile/screens/article_screen.dart';
4.	import 'package:tgs_pemrograman_mobile/screens/discover_screen.dart';
5.	import 'package:tgs_pemrograman_mobile/screens/profile.dart';
6.	import 'package:tgs_pemrograman_mobile/screens/settings.dart';
7.	import 'package:tgs_pemrograman_mobile/services/fetchapi.dart';
8.	import 'package:tgs_pemrograman_mobile/widgets/custom_tag.dart';
9.	import 'package:tgs_pemrograman_mobile/screens/login_screen.dart';
10.	import 'package:google_fonts/google_fonts.dart';
11.	import '../widgets/bottom_nav_bar.dart';
12.	import '../widgets/image_container.dart';
13.	
14.	class HomeScreen extends StatefulWidget {
15.	const HomeScreen({super.key});
16.	
17.	static const routeName = '/';
18.	

19.	@override
20.	State<HomeScreen> createState() => _HomeScreenState();
21.	}
22.	
23.	class _HomeScreenState extends State<HomeScreen> {
24.	ApiDatas fetchapi = new ApiDatas();
25.	
26.	final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
27.	
28.	@override
29.	Widget build(BuildContext context) {
30.	String? username = ModalRoute.of(context)?.settings.arguments as String?;
31.	Article article = Article.articles[0];
32.	
33.	if (username == null) {
34.	username = 'Guest';
35.	}
36.	
37.	return Scaffold(
38.	key: _scaffoldKey,
39.	appBar: AppBar(
40.	backgroundColor: Colors.transparent,
41.	elevation: 0,
42.	),
43.	bottomNavigationBar: BottomNavBar(index: 0),
44.	extendBodyBehindAppBar: true,
45.	body: FutureBuilder(
46.	future: fetchapi.getData(),
47.	builder: (context, snapshot) {
48.	if (snapshot.connectionState == ConnectionState.waiting) {
49.	return Center(
50.	child: CircularProgressIndicator(),
51.	);
52.	} else if (snapshot.hasError) {
53.	return Center(
54.	child: Text("Error: \${snapshot.error}"),
55.	);
56.	} else {
57.	return ListView(padding: EdgeInsets.zero, children: [
58.	_NewsOfTheDay(article: snapshot.data!["articles"]),
59.	_BreakingNews(
60.	articles: snapshot.data!["articles"],
61.	username: username!,
62.	),
63.	
64.	SizedBox(height: 20), // Spasi sebelum teks tambahan
65.	Container(
66.	color: Colors.teal, // Background warna teal
67.	padding:
68.	const EdgeInsets.symmetric(horizontal: 20, vertical: 10),
69.	child: Center(
70.	child: Text(
71.	'Halo, \$username. This is Headlines News Today',
72.	textAlign: TextAlign.center, // Rata tengah
73.	style: Theme.of(context).textTheme.headline6!.copyWith(
74.	color: Colors.white), // Teks besar dan warna putih
75.	),
76.	),

77.	),
78.	]);
79.	}
80.	},
81.	),
82.	);
83.	}
84.	}
85.	
86.	class BreakingNews extends StatelessWidget {
87.	const _BreakingNews({
88.	required this.articles,
89.	required this.username,
90.	super.key,
91.	});
92.	
93.	final List articles;
94.	final String username;
95.	
96.	String ubahdate(String date) {
97.	String dateTimeString = date;
98.	
99.	// Ambil tahun, bulan, dan tanggal menggunakan slicing character
100.	String year = dateTimeString.substring(0, 4);
101.	String month = dateTimeString.substring(5, 7);
102.	String day = dateTimeString.substring(8, 10);
103.	String time = dateTimeString.substring(11, 16); // Ambil jam (HH:mm)
104.	
105.	return "\$year-\$month-\$day (\$time)";
106.	}
107.	
108.	String ubahAuthor(String author) {
109.	String originalString = author;
110.	
111.	// Membagi string menjadi kata-kata
112.	List<String> words = originalString.split(' ');
113.	
114.	// Mengambil dua kata pertama
115.	String firstTwoWords =
116.	words.length >= 2 ? '\${words[0]} \${words[1]}' : originalString;
117.	
118.	// Tampilkan hasil
119.	return firstTwoWords;
120.	}
121.	
122.	@override
123.	Widget build(BuildContext context) {
124.	return Padding(
125.	padding: const EdgeInsets.all(20.0),
126.	child: Column(
127.	children: [
128.	Row(
129.	mainAxisAlignment: MainAxisAlignment.spaceBetween,
130.	children: [
131.	Text(
132.	'Giga News Today',
133.	style: Theme.of(context)
134.	.textTheme

135.	.headlineSmall!
136.	.copyWith(fontWeight: FontWeight.bold),
137.	),
138.	// Text('More', style: Theme.of(context).textTheme.bodyLarge),
139.	],
140.	),
141.	const SizedBox(height: 20),
142.	SizedBox(
143.	height: 250,
144.	child: ListView.builder(
145.	scrollDirection: Axis.horizontal,
146.	itemCount: articles.length,
147.	itemBuilder: (context, index) {
148.	if (articles[index]["urlToImage"] == null
149.	articles[index]["content"] == null) {
150.	return Container();
151.	} else {
152.	return Container(
153.	width: MediaQuery.of(context).size.width * 0.5,
154.	margin: const EdgeInsets.only(right: 10),
155.	child: InkWell(
156.	onTap: () {
157.	Navigator.pushNamed(
158.	context,
159.	ArticleScreen.routeName,
160.	arguments: articles[index],
161.	);
162.	},
163.	child: Column(
164.	crossAxisAlignment: CrossAxisAlignment.start,
165.	children: [
166.	ImageContainer(
167.	width: MediaQuery.of(context).size.width * 0.5,
168.	imageUrl: articles[index]["urlToImage"] ?? "",
169.	),
170.	const SizedBox(height: 10),
171.	Text(
172.	articles[index]["title"] ?? "",
173.	maxLines: 2,
174.	style: Theme.of(context)
175.	.textTheme
176.	.bodyLarge!
177.	.copyWith(
178.	fontWeight: FontWeight.bold, height: 1.5),
179.	),
180.	const SizedBox(height: 5),
181.	Text(
182.	"at \${ubahdate(articles[index]["publishedAt"] ?? "")}",
183.	style: Theme.of(context).textTheme.bodySmall!),
184.	const SizedBox(height: 5),
185.	Text(
186.	"by \${ubahAuthor(articles[index]["author"] ?? "")}",
187.	style: Theme.of(context).textTheme.bodySmall!),
188.	],
189.	),
190.	),
191.	);
192.	}



193.	},
194.	),
195.	),
196.	],
197.	),
198.	);
199.	}
200.	}
201.	
202.	class NewsOfTheDay extends StatelessWidget {
203.	const _NewsOfTheDay({
204.	Key? key,
205.	required this.article,
206.	}) : super(key: key);
207.	
208.	final List article;
209.	
210.	get index => null;
211.	
212.	@override
213.	Widget build(BuildContext context) {
214.	String? imageUrl = "";
215.	
216.	// Cari artikel pertama yang memiliki gambar
217.	for (var article in article) {
218.	if (article["urlToImage"] != null && article["title"] != null) {
219.	imageUrl = article["urlToImage"];
220.	break;
221.	}
222.	}
223.	return Column(
224.	crossAxisAlignment: CrossAxisAlignment.start,
225.	children: [
226.	Padding(
227.	padding: const EdgeInsets.all(20.0),
228.	child: Column(
229.	crossAxisAlignment: CrossAxisAlignment.start,
230.	children: [
231.	const SizedBox(height: 80),
232.	Container(
233.	height: MediaQuery.of(context).size.height * 0.25,
234.	width: double.infinity,
235.	decoration: BoxDecoration(
236.	image: DecorationImage(
237.	image: NetworkImage(article[0]["urlToImage"] ?? ""),
238.	fit: BoxFit.cover,
239.	),
240.	borderRadius: BorderRadius.circular(10),
241.	),
242.	child: Padding(
243.	padding: const EdgeInsets.all(20.0),
244.	child: Column(
245.	crossAxisAlignment: CrossAxisAlignment.start,
246.	mainAxisAlignment: MainAxisAlignment.end,
247.	children: [
248.	SizedBox(
249.	width: MediaQuery.of(context).size.width * 0.6,
250.	child: Text(

251.	article[0]["title"] ?? "",
252.	maxLines: 2,
253.	overflow: TextOverflow.ellipsis,
254.	style:
255.	Theme.of(context).textTheme.headline6!.copyWith(
256.	fontWeight: FontWeight.bold,
257.	color: Colors.white,
258.	),
259.	),
260.	),
261.	const SizedBox(height: 10),
262.	InkWell(
263.	onTap: () {
264.	Navigator.pushNamed(
265.	context,
266.	ArticleScreen.routeName,
267.	arguments: article[0],
268.	);
269.	},
270.	child: Container(
271.	decoration: BoxDecoration(
272.	color: Colors.white,
273.	borderRadius: BorderRadius.circular(20),
274.	border: Border.all(color: Colors.teal),
275.	),
276.	padding: const EdgeInsets.symmetric(
277.	horizontal: 12, vertical: 6),
278.	child: Wrap(
279.	crossAxisAlignment: WrapCrossAlignment.center,
280.	children: [
281.	Text(
282.	'Read More',
283.	style: Theme.of(context)
284.	.textTheme
285.	.bodyText2!
286.	.copyWith(
287.	color: Colors.teal,
288.	),
289.	),
290.	const SizedBox(width: 10),
291.	const Icon(
292.	Icons.arrow_right_alt,
293.	color: Colors.teal,
294.	),
295.	],
296.	),
297.	),
298.	),
299.	],
300.	),
301.	),
302.	),
303.	],
304.	),
305.	),
306.	],
307.	);
308.	}

309.	}
------	---

## 5. Kategori

1.	import 'package:flutter/material.dart';
2.	import 'package:tgs_pemrograman_mobile/screens/article_screen.dart';
3.	import 'package:tgs_pemrograman_mobile/screens/home_screen.dart';
4.	import 'package:tgs_pemrograman_mobile/screens/login_screen.dart';
5.	import 'package:tgs_pemrograman_mobile/screens/settings.dart';
6.	import 'package:tgs_pemrograman_mobile/services/fetchapi.dart';
7.	import 'package:tgs_pemrograman_mobile/widgets/image_container.dart';
8.	
9.	// import '../models/article_model.dart';
10.	import '../widgets/bottom_nav_bar.dart';
11.	
12.	class DiscoverScreen extends StatelessWidget {
13.	const DiscoverScreen({super.key});
14.	
15.	static const routeName = '/discover';
16.	@override
17.	Widget build(BuildContext context) {
18.	List<String> tabs = [
19.	"Entertainment",
20.	"General",
21.	"Health",
22.	"Business",
23.	"Science",
24.	"Sport",
25.	"Technology"
26.	];
27.	
28.	int selectedIndex = 0;
29.	
30.	return DefaultTabController(
31.	initialIndex: 0,
32.	length: tabs.length,
33.	child: Scaffold(
34.	appBar: AppBar(
35.	backgroundColor: Colors.transparent,
36.	elevation: 0,
37.	leading: IconButton(
38.	onPressed: () {},
39.	icon: const Icon(
40.	Icons.menu,
41.	color: Colors.black,
42.	),
43.	),
44.	),
45.	bottomNavigationBar: BottomNavBar(index: 1),
46.	body: ListView(
47.	padding: const EdgeInsets.all(20.0),
48.	children: [const _DiscoverNews(), _CategoryNews(tabs: tabs)],
49.	),
50.	);
51.	}
52.	}
53.	
54.	class _CategoryNews extends StatefulWidget {
55.	const _CategoryNews({
56.	Key? key,
57.	required this.tabs,
58.	}) : super(key: key);
59.	

60.	final List<String> tabs;
61.	
62.	@override
63.	State<_CategoryNews> createState() => _CategoryNewsState();
64.	}
65.	
66.	class _CategoryNewsState extends State<_CategoryNews> {
67.	//---API Fetch Section---//
68.	
69.	//---API Fetch Section---//
70.	Map data = {};
71.	ApiDatas fetchapi = new ApiDatas();
72.	
73.	void setupData(String category) async {
74.	Map response = await fetchapi.getDataByCategory(category);
75.	if (response.isEmpty) {
76.	print("Something error");
77.	} else {
78.	print("Data Fetch Success");
79.	setState(() {
80.	data = response;
81.	});
82.	}
83.	}
84.	
85.	@override
86.	void initState() {
87.	// Memanggil dengan kategori pertama sebagai nilai default
88.	setupData(widget.tabs[0].toLowerCase());
89.	super.initState();
90.	}
91.	
92.	String ubahdate(String date) {
93.	String dateTimeString = date;
94.	
95.	// Ambil tahun, bulan, dan tanggal menggunakan slicing character
96.	String year = dateTimeString.substring(0, 4);
97.	String month = dateTimeString.substring(5, 7);
98.	String day = dateTimeString.substring(8, 10);
99.	String time = dateTimeString.substring(11, 16); // Ambil jam (HH:mm)
100.	
101.	return "\$year-\$month-\$day (\$time)";
102.	}
103.	
104.	@override
105.	Widget build(BuildContext context) {
106.	return Column(
107.	children: [
108.	TabBar(
109.	isScrollable: true,
110.	indicatorColor: Colors.black,
111.	tabs: widget.tabs
112.	.map(
113.	(tab) => Tab(
114.	icon: Text(
115.	tab,
116.	style: Theme.of(context)
117.	.textTheme
118.	.headlineSmall!
119.	.copyWith(fontWeight: FontWeight.bold),
120.	),
121.	),

122.	)
123.	.toList()),
124.	SizedBox(
125.	height: MediaQuery.of(context).size.height,
126.	child: TabBarView(
127.	children: widget.tabs
128.	.map(
129.	(tab => FutureBuilder(
130.	future: fetchapi.getDataByCategory(tab.toLowerCase()),
131.	builder: (context, snapshot) {
132.	if (snapshot.connectionState ==
133.	ConnectionState.waiting) {
134.	return Center(
135.	child: CircularProgressIndicator(),
136.	);
137.	} else if (snapshot.hasError) {
138.	return Center(
139.	child: Text("Error: \${snapshot.error}"),
140.	);
141.	} else {
142.	final articles = snapshot.data!["articles"];
143.	return ListView.builder(
144.	shrinkWrap: true,
145.	itemCount: articles.length,
146.	itemBuilder: ((context, index) {
147.	return InkWell(
148.	onTap: () {
149.	setupData(tab);
150.	Navigator.pushNamed(
151.	context,
152.	ArticleScreen.routeName,
153.	arguments: articles[index],
154.	);
155.	),
156.	child: Row(
157.	children: [
158.	ImageContainer(
159.	width: 80,
160.	height: 80,
161.	margin: const EdgeInsets.all(10.0),
162.	borderRadius: 5,
163.	imageUrl:
164.	articles[index]["urlToImage"] ?? "",
165.	),
166.	Expanded(
167.	child: Column(
168.	crossAxisAlignment:
169.	CrossAxisAlignment.start,
170.	mainAxisAlignment:
171.	MainAxisAlignment.center,
172.	children: [
173.	Text(
174.	articles[index]["title"] ?? "",
175.	maxLines: 2,
176.	overflow: TextOverflow.clip,
177.	style: Theme.of(context)
178.	.textTheme
179.	.bodyLarge!
180.	.copyWith(
181.	fontWeight: FontWeight.bold,
182.	),
183.	),

184.	const SizedBox(
185.	height: 10,
186.	),
187.	Row(
188.	children: [
189.	const Icon(
190.	Icons.schedule,
191.	size: 18,
192.	),
193.	const SizedBox(width: 5),
194.	Text(
195.	ubahdate(articles[index]
196.	["publishedAt"] ??
197.	""),
198.	style: const TextStyle(
199.	fontSize: 12),
200.	),
201.	const SizedBox(width: 20),
202.	),
203.	)
204.	],
205.	),
206.	),
207.	],
208.	),
209.	);
210.	}},
211.	);
212.	}
213.	},
214.	),
215.	)
216.	.toList()),
217.	)
218.	],
219.	);
220.	}
221.	}
222.	
223.	class _DiscoverNews extends StatelessWidget {
224.	const _DiscoverNews({
225.	super.key,
226.	});
227.	
228.	@override
229.	Widget build(BuildContext context) {
230.	return SizedBox(
231.	height: MediaQuery.of(context).size.height * 0.25,
232.	child: Column(
233.	crossAxisAlignment: CrossAxisAlignment.start,
234.	mainAxisAlignment: MainAxisAlignment.center,
235.	children: [
236.	Text(
237.	'Categories',
238.	style: Theme.of(context)
239.	.textTheme
240.	.headline4!
241.	.copyWith(color: Colors.teal, fontWeight: FontWeight.w700),
242.	),
243.	const SizedBox(height: 10),
244.	Text('Find your news today!',
245.	style: Theme.of(context).textTheme.displayLarge),

246.	],
247.	),
248.	);
249.	}
250.	}
251.	import 'package:flutter/material.dart';
252.	import 'package:tgs_pemrograman_mobile/screens/article_screen.dart';
253.	import 'package:tgs_pemrograman_mobile/screens/home_screen.dart';
254.	import 'package:tgs_pemrograman_mobile/screens/login_screen.dart';
255.	import 'package:tgs_pemrograman_mobile/screens/settings.dart';
256.	import 'package:tgs_pemrograman_mobile/services/fetchapi.dart';
257.	import 'package:tgs_pemrograman_mobile/widgets/image_container.dart';
258.	
259.	// import '../models/article_model.dart';
260.	import '../widgets/bottom_nav_bar.dart';
261.	
262.	class DiscoverScreen extends StatelessWidget {
263.	const DiscoverScreen({super.key});
264.	
265.	static const routeName = '/discover';
266.	@override
267.	Widget build(BuildContext context) {
268.	List<String> tabs = [
269.	"Entertainment",
270.	"General",
271.	"Health",
272.	"Business",
273.	"Science",
274.	"Sport",
275.	"Technology"
276.	];
277.	
278.	int selectedIndex = 0;
279.	
280.	return DefaultTabController(
281.	initialIndex: 0,
282.	length: tabs.length,
283.	child: Scaffold(
284.	appBar: AppBar(
285.	backgroundColor: Colors.transparent,
286.	elevation: 0,
287.	leading: IconButton(
288.	onPressed: () {},
289.	icon: const Icon(
290.	Icons.menu,
291.	color: Colors.black,
292.	),
293.	),
294.	),
295.	bottomNavigationBar: BottomNavBar(index: 1),
296.	body: ListView(
297.	padding: const EdgeInsets.all(20.0),
298.	children: [const _DiscoverNews(), _CategoryNews(tabs: tabs)],
299.	),
300.	);
301.	}
302.	}
303.	
304.	class _CategoryNews extends StatefulWidget {
305.	const _CategoryNews({
306.	Key? key,
307.	required this.tabs,

308.	}) : super(key: key);
309.	
310.	final List<String> tabs;
311.	
312.	@override
313.	State<_CategoryNews> createState() => _CategoryNewsState();
314.	}
315.	
316.	class _CategoryNewsState extends State<_CategoryNews> {
317.	///---API Fetch Section----//
318.	
319.	///---API Fetch Section----//
320.	Map data = {};
321.	ApiDatas fetchapi = new ApiDatas();
322.	
323.	void setupData(String category) async {
324.	Map response = await fetchapi.getDataByCategory(category);
325.	if (response.isEmpty) {
326.	print("Something error");
327.	} else {
328.	print("Data Fetch Success");
329.	setState(() {
330.	data = response;
331.	});
332.	}
333.	}
334.	
335.	@override
336.	void initState() {
337.	/// Memanggil dengan kategori pertama sebagai nilai default
338.	setupData(widget.tabs[0].toLowerCase());
339.	super.initState();
340.	}
341.	
342.	String ubahdate(String date) {
343.	String dateTimeString = date;
344.	
345.	/// Ambil tahun, bulan, dan tanggal menggunakan slicing character
346.	String year = dateTimeString.substring(0, 4);
347.	String month = dateTimeString.substring(5, 7);
348.	String day = dateTimeString.substring(8, 10);
349.	String time = dateTimeString.substring(11, 16); // Ambil jam (HH:mm)
350.	
351.	return "\$year-\$month-\$day (\$time)";
352.	}
353.	
354.	@override
355.	Widget build(BuildContext context) {
356.	return Column(
357.	children: [
358.	TabBar(
359.	isScrollable: true,
360.	indicatorColor: Colors.black,
361.	tabs: widget.tabs
362.	.map(
363.	(tab) => Tab(
364.	icon: Text(
365.	tab,
366.	style: Theme.of(context)
367.	.textTheme
368.	.headlineSmall!
369.	.copyWith(fontWeight: FontWeight.bold),



370.	),
371.	),
372.	)
373.	.toList()),
374.	SizedBox(
375.	height: MediaQuery.of(context).size.height,
376.	child: TabBarView(
377.	children: widget.tabs
378.	.map(
379.	(tab) => FutureBuilder(
380.	future: fetchapi.getDataByCategory(tab.toLowerCase()),
381.	builder: (context, snapshot) {
382.	if (snapshot.connectionState ==
383.	ConnectionState.waiting) {
384.	return Center(
385.	child: CircularProgressIndicator(),
386.	);
387.	} else if (snapshot.hasError) {
388.	return Center(
389.	child: Text("Error: \${snapshot.error}"),
390.	);
391.	} else {
392.	final articles = snapshot.data!["articles"];
393.	return ListView.builder(
394.	shrinkWrap: true,
395.	itemCount: articles.length,
396.	itemBuilder: ((context, index) {
397.	return InkWell(
398.	onTap: () {
399.	setupData(tab);
400.	Navigator.pushNamed(
401.	context,
402.	ArticleScreen.routeName,
403.	arguments: articles[index],
404.	);
405.	),
406.	child: Row(
407.	children: [
408.	ImageContainer(
409.	width: 80,
410.	height: 80,
411.	margin: const EdgeInsets.all(10.0),
412.	borderRadius: 5,
413.	imageUrl:
414.	articles[index]["urlToImage"] ?? "",
415.	),
416.	Expanded(
417.	child: Column(
418.	crossAxisAlignment:
419.	CrossAxisAlignment.start,
420.	mainAxisAlignment:
421.	MainAxisAlignment.center,
422.	children: [
423.	Text(
424.	articles[index]["title"] ?? "",
425.	maxLines: 2,
426.	overflow: TextOverflow.clip,
427.	style: Theme.of(context)
428.	.textTheme
429.	.bodyLarge!
430.	.copyWith(
431.	fontWeight: FontWeight.bold,

432.	),
433.	),
434.	const SizedBox(
435.	height: 10,
436.	),
437.	Row(
438.	children: [
439.	const Icon(
440.	Icons.schedule,
441.	size: 18,
442.	),
443.	const SizedBox(width: 5),
444.	Text(
445.	ubahdate(articles[index]
446.	["publishedAt"] ??
447.	""),
448.	style: const TextStyle(
449.	fontSize: 12),
450.	),
451.	const SizedBox(width: 20),
452.	),
453.	)
454.	],
455.	),
456.	),
457.	],
458.	),
459.	);
460.	}},
461.	);
462.	}
463.	},
464.	),
465.	)
466.	.toList()),
467.	)
468.	],
469.	);
470.	}
471.	}
472.	
473.	class _DiscoverNews extends StatelessWidget {
474.	const _DiscoverNews({
475.	super.key,
476.	});
477.	
478.	@override
479.	Widget build(BuildContext context) {
480.	return SizedBox(
481.	height: MediaQuery.of(context).size.height * 0.25,
482.	child: Column(
483.	crossAxisAlignment: CrossAxisAlignment.start,
484.	mainAxisAlignment: MainAxisAlignment.center,
485.	children: [
486.	Text(
487.	'Categories',
488.	style: Theme.of(context)
489.	.textTheme
490.	.headline4!
491.	.copyWith(color: Colors.teal, fontWeight: FontWeight.w700),
492.	),
493.	const SizedBox(height: 10),

494.	Text('Find your news today!',
495.	style: Theme.of(context).textTheme.displayLarge),
496.	],
497.	),
498.	);
499.	}
500.	}

## 6. Detail

1.	import 'package:flutter/material.dart';
2.	import 'package:tgspemrograman_mobile/widgets/image_container.dart';
3.	
4.	// import '../models/article_model.dart';
5.	import '../widgets/custom_tag.dart';
6.	
7.	class ArticleScreen extends StatelessWidget {
8.	const ArticleScreen({super.key});
9.	
10.	static const routeName = '/article';
11.	@override
12.	Widget build(BuildContext context) {
13.	final article = ModalRoute.of(context)!.settings.arguments as Map;
14.	return ImageContainer(
15.	width: double.minPositive,
16.	imageUrl: article["urlToImage"],
17.	child: Scaffold(
18.	backgroundColor: Colors.transparent,
19.	appBar: AppBar(
20.	iconTheme: const IconThemeData(color: Color.fromARGB(255, 0, 0, 0)),
21.	backgroundColor: Colors.transparent,
22.	elevation: 0,
23.	),
24.	extendBodyBehindAppBar: true,
25.	body: ListView(
26.	children: [
27.	_NewsHeadline(article: article),
28.	_NewsBody(article: article)
29.	],
30.	),
31.	),
32.	);
33.	}
34.	}
35.	
36.	class _NewsBody extends StatelessWidget {
37.	const _NewsBody({
38.	super.key,
39.	required this.article,
40.	});
41.	
42.	final Map article;
43.	
44.	@override
45.	Widget build(BuildContext context) {
46.	String ubahdate(String date) {
47.	String dateTimeString = date;
48.	// Ambil tahun, bulan, dan tanggal menggunakan slicing character
49.	String year = dateTimeString.substring(0, 4);
50.	String month = dateTimeString.substring(5, 7);
51.	String day = dateTimeString.substring(8, 10);

52.	
53.	return "\$year-\$month-\$day";
54.	}
55.	
56.	String ubahAuthor(String author) {
57.	String originalString = author;
58.	
59.	// Membagi string menjadi kata-kata
60.	List<String> words = originalString.split(' ');
61.	
62.	// Mengambil dua kata pertama
63.	String firstTwoWords =
64.	words.length >= 2 ? '\${words[0]} \${words[1]}' : originalString;
65.	
66.	// Tampilkan hasil
67.	return firstTwoWords;
68.	}
69.	
70.	return Container(
71.	padding: const EdgeInsets.all(20.0),
72.	decoration: BoxDecoration(
73.	borderRadius: BorderRadius.only(
74.	topLeft: Radius.circular(20), topRight: Radius.circular(20)),
75.	color: Color.fromARGB(255, 0, 0, 0),
76.	),
77.	child: Column(
78.	children: [
79.	Row(
80.	children: [
81.	CustomTag(
82.	backgroundColor: Colors.black,
83.	children: [
84.	CircleAvatar(
85.	radius: 15,
86.	backgroundImage: NetworkImage(article["urlToImage"] ?? ""),
87.	),
88.	const SizedBox(width: 10),
89.	Text(
90.	ubahAuthor(article["author"] ?? ""),
91.	style: Theme.of(context).textTheme.bodyMedium!.copyWith(
92.	color: Colors.white,
93.	),
94.	),
95.	],
96.	),
97.	const SizedBox(width: 10),
98.	CustomTag(
99.	backgroundColor: Colors.teal,
100.	children: [
101.	const Icon(Icons.timer, color: Colors.grey),
102.	const SizedBox(width: 50),
103.	Text(
104.	ubahdate(article["publishedAt"] ?? ""),
105.	style: Theme.of(context).textTheme.bodyMedium,
106.	),
107.	],
108.	),
109.	const SizedBox(width: 10),
110.	],
111.	),
112.	const SizedBox(height: 20),
113.	Text(

114.	article["title"] ?? "",
115.	style: Theme.of(context)
116.	.textTheme
117.	.headlineSmall!
118.	.copyWith(fontWeight: FontWeight.bold),
119.	),
120.	const SizedBox(height: 20),
121.	Text(
122.	article["content"] ?? "",
123.	style:
124.	Theme.of(context).textTheme.bodyMedium!.copyWith(height: 1.5),
125.	),
126.	const SizedBox(height: 200),
127.	],
128.	),
129.	);
130.	}
131.	}
132.	
133.	class _NewsHeadline extends StatelessWidget {
134.	const _NewsHeadline({
135.	super.key,
136.	required this.article,
137.	});
138.	
139.	final Map article;
140.	
141.	@override
142.	Widget build(BuildContext context) {
143.	return Padding(
144.	padding: const EdgeInsets.all(20.0),
145.	child: Column(
146.	mainAxisAlignment: MainAxisAlignment.end,
147.	crossAxisAlignment: CrossAxisAlignment.start,
148.	children: [
149.	SizedBox(
150.	height: MediaQuery.of(context).size.height * 0.15,
151.	),
152.	CustomTag(
153.	backgroundColor: Colors.grey.withAlpha(150),
154.	children: [],
155.	),
156.	const SizedBox(height: 10),
157.	Text(
158.	article["title"] ?? "",
159.	style: Theme.of(context).textTheme.headlineSmall!.copyWith(
160.	fontWeight: FontWeight.bold,
161.	color: Colors.white,
162.	height: 1.25,
163.	),
164.	),
165.	const SizedBox(height: 10),
166.	Text(
167.	article["description"] ?? "",
168.	style: Theme.of(context).textTheme.bodyMedium!.copyWith(
169.	color: Colors.white,
170.	),
171.	),
172.	],
173.	),
174.	);
175.	}

### C. View Model

Pada bagian ini akan dibahas tentang kode program yang berisi fungsi-fungsi untuk request data ke server dengan format API yang telah ditentukan. Adapun fungsi-fungsi tersebut ialah:

1. `getData ()` untuk request data headline berita.
2. `getDataByCategory ()` untuk request data menyesuaikan kategori.

Adapun masing-masing kode program fungsi tersebut adalah sebagai berikut:

#### 1. `getData()`

1	body: FutureBuilder(
2	future: fetchapi.getData(),
3	builder: (context, snapshot) {
4	if (snapshot.connectionState == ConnectionState.waiting) {
5	return Center(
6	child: CircularProgressIndicator(),
7	);
8	} else if (snapshot.hasError) {
9	return Center(
10	child: Text("Error: \${snapshot.error}"),
11	);
12	} else {
13	return ListView(padding: EdgeInsets.zero, children: [
14	_NewsOfTheDay(article: snapshot.data!["articles"]),
15	_BreakingNews(
16	articles: snapshot.data!["articles"],
17	username: username!,
18	),
19	]);
20	}
21	},
22	),

Fungsi **getData()** pada view model ini bertujuan untuk membuat permintaan (request) ke server untuk mengambil data headline berita. Melalui **fetchapi**, fungsi ini menginisiasi permintaan ke sumber data yang relevan, kemudian menggunakan **Future** untuk menunggu respons dari server. Saat status koneksi menunjukkan sedang menunggu (waiting), tampilan menampilkan indikator loading. Jika terjadi kesalahan dalam proses pengambilan data, tampilan akan menampilkan pesan error. Setelah data diterima dengan sukses, tampilan akan menampilkan dua bagian dari artikel: **\_NewsOfTheDay** untuk artikel utama dan **\_BreakingNews** untuk berita terkini dengan informasi pengguna yang diberikan.

## 2. getDataByCategory ()

1.	(tab) => FutureBuilder(
2.	future: fetchapi.getDataByCategory(tab.toLowerCase()),
3.	builder: (context, snapshot) {
4.	if (snapshot.connectionState ==
5.	ConnectionState.waiting) {
6.	return Center(
7.	child: CircularProgressIndicator(),
8.	);
9.	} else if (snapshot.hasError) {
10.	return Center(
11.	child: Text("Error: \${snapshot.error}"),
12.	);
13.	} else {
14.	final articles = snapshot.data!["articles"];
15.	return ListView.builder(
16.	shrinkWrap: true,
17.	itemCount: articles.length,
18.	itemBuilder: ((context, index) {
19.	return InkWell(
20.	onTap: () {
21.	setupData(tab);
22.	Navigator.pushNamed(
23.	context,
24.	ArticleScreen.routeName,
25.	arguments: articles[index],
26.	);
27.	},
28.	child: Row(
29.	children: [
30.	ImageContainer(
31.	width: 80,
32.	height: 80,
33.	margin: const EdgeInsets.all(10.0),
34.	borderRadius: 5,
35.	imageUrl:
36.	articles[index]["urlToImage"] ?? "",
37.	),
38.	Expanded(
39.	child: Column(
40.	crossAxisAlignment:
41.	CrossAxisAlignment.start,
42.	mainAxisAlignment:

43.	MainAxisAlignment.center,
44.	children: [
45.	Text(
46.	articles[index]["title"] ?? "",
47.	maxLines: 2,
48.	overflow: TextOverflow.clip,
49.	style: Theme.of(context)
50.	.textTheme
51.	.bodyLarge!
52.	.copyWith(
53.	fontWeight: FontWeight.bold,
54.	),
55.	),
56.	const SizedBox(
57.	height: 10,
58.	),
59.	Row(
60.	children: [
61.	const Icon(
62.	Icons.schedule,
63.	size: 18,
64.	),
65.	const SizedBox(width: 5),
66.	Text(
67.	ubahdate(articles[index]
68.	["publishedAt"] ??
69.	""),
70.	style: const TextStyle(
71.	fontSize: 12),
72.	),
73.	const SizedBox(width: 20),
74.	],
75.	)
76.	],
77.	),
78.	),
79.	],
80.	),
81.	);
82.	}),
83.	);
84.	}
85.	},
86.	),



View model di atas menggunakan **FutureBuilder** untuk mengambil data berita berdasarkan kategori yang dipilih (dari tab). Melalui **fetchapi.getDataByCategory()**, permintaan dilakukan dengan kategori tertentu. Saat menunggu respons dari server, tampilan menampilkan indikator loading. Jika terjadi kesalahan saat pengambilan data, pesan error ditampilkan. Ketika data diterima, ListView dibangun dengan setiap elemen menampilkan gambar, judul, dan waktu terbit artikel. Ketika pengguna mengetuk salah satu artikel, fungsi **setupData()** dipanggil untuk menyiapkan data terkait kategori yang dipilih, kemudian pengguna diarahkan ke layar Artikel dengan informasi artikel yang dipilih sebagai argumen.

## D. Route dan Navigasi Antar Halaman

### 1. Pada Halaman Login

1.	onPressed: () async {
2.	String enteredUsername = _usernameController.text;
3.	String enteredPassword = _passwordController.text;
4.	
5.	if (users.containsKey(enteredUsername) &&
6.	users[enteredUsername] == enteredPassword) {
7.	final prefs = await SharedPreferences.getInstance();
8.	await prefs.setString('username', enteredUsername);
9.	await prefs.setString('password', enteredPassword);
10.	
11.	Navigator.pushReplacementNamed(
12.	context,
13.	Profile.routeName,
14.	);
15.	} else {
16.	ScaffoldMessenger.of(context).showSnackBar(
17.	SnackBar(
18.	backgroundColor: Color.fromARGB(255, 255, 15, 15),
19.	content: Text(
20.	'Incorrect Username or Password',
21.	style: TextStyle(
22.	color: Colors
23.	.white), // Mengubah warna teks menjadi putih
24.	),
25.	),
26.	);

27.	}
28.	},

Di halaman **LoginScreen** terdapat penggunaan route dan navigasi.

Pada bagian tombol **LOGIN**, ketika pengguna menekannya, terdapat perintah `Navigator.pushReplacementNamed(context, Profile.routeName)` untuk melakukan navigasi ke layar **Profile** menggunakan named route setelah pengguna berhasil login dengan username dan password yang sesuai.

Selain itu, setelah pengguna berhasil login, terdapat penyimpanan data username dan password menggunakan **SharedPreferences**, yang nantinya dapat digunakan untuk autentikasi atau identifikasi pengguna ketika membuka kembali aplikasi.

## 2. Pada halaman profile

1.	child: ElevatedButton(
2.	onPressed: () {
3.	Navigator.push(
4.	context,
5.	MaterialPageRoute(builder: (context) => ChangeProfile()),
6.	);
7.	},
8.	child: Text(
9.	'Ubah Data',
10.	style: TextStyle(
11.	color: Colors.orange, // Mengatur warna teks menjadi oranye
12.	),
13.	),
14.	),

Di halaman **Profile** terdapat penggunaan route dan navigasi. Saat tombol 'Ubah Data' ditekan, terdapat perintah navigasi yang menggunakan **Navigator.push** untuk beralih ke halaman **ChangeProfile** menggunakan **MaterialPageRoute**.

## 3. Pada halaman utama dan halaman kategori

1.	onTap: () {
2.	Navigator.pushNamed(
3.	context,

4.	<code>ArticleScreen.routeName,</code>
5.	<code>arguments: articles[index],</code>
6.	<code>);</code>
7.	<code>},</code>

Kode ini menggunakan **Navigator.pushNamed** untuk beralih ke halaman **ArticleScreen** ketika pengguna mengetuk salah satu dari berita yang ditampilkan. Dalam proses navigasi, **ArticleScreen.routeName** digunakan sebagai rujukan ke rute halaman yang akan ditampilkan, dan **arguments** digunakan untuk mengirimkan data artikel terkait yang akan ditampilkan di halaman **ArticleScreen**. Dengan demikian, pengguna akan melihat detail artikel yang sesuai dengan berita yang mereka pilih ketika mengetuk elemen berita pada halaman **HomeScreen**.

#### 4. Button Navigator

Pada aplikasi ini saya implementasikan button navbar untuk navigasi ke berbagai halaman.

- BottomNavigationBar: BottomNavBar(index: 0)  
Index: 0 mendeskripsikan bahwa navigator akan menuju home page.
- BottomNavigationBar: BottomNavBar(index: 1)  
Index: 1 mendeskripsikan bahwa navigator akan menuju categories page.
- BottomNavigationBar: BottomNavBar(index: 2)  
Index: 2 mendeskripsikan bahwa navigator akan menuju profile page.

Link Github:

<https://github.com/henryrimadana/UASMOBILE>