

RESPONSI 1 PRAKTIKUM PEMROGRAMAN MOBILE

Syadina Zufarin Afifah

H1D021077

Shift F

A. Database

Ketentuan :

Nama Tabel : riwayat_alergi

Kolom 1 : id (int, PK, increment)

Kolom 2 : allergen (String)

Kolom 3 : reaction (String)

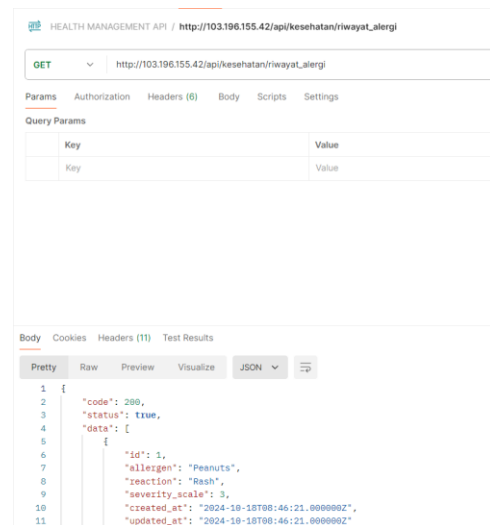
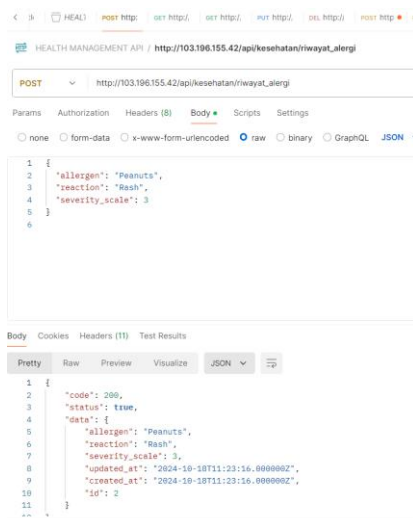
Kolom 4 : severity_scale (Integer)

Screenshoot Database :

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id	int(10)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 allergen	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3 reaction	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4 severity_scale	int(20)			Tidak	Tidak ada			Ubah Hapus Lainnya

B. POSTMAN

Untuk postman, saya membuat postman untuk login, registrasi dan riwayat alergi



HEALTH MANAGEMENT API / http://103.196.155.42/api/kesehatan/riwayat_alergi/1

GET ▼ http://103.196.155.42/api/kesehatan/riwayat_alergi/1

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value
Key	Value

HEALTH MANAGEMENT API / http://103.196.155.42/api/kesehatan/riwayat_alergi/1/update

PUT ▼ http://103.196.155.42/api/kesehatan/riwayat_alergi/1/update

Params Authorization Headers (8) Body ● Scripts Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 {
2   "code": 200,
3   "status": true,
4   "data": {
5     "id": 1,
6     "allergen": "Peanuts",
7     "reaction": "Rash",
8     "severity_scale": 3,
9     "created_at": "2024-10-18T08:46:21.000000Z",
10    "updated_at": "2024-10-18T08:46:21.000000Z"
11  }
12 }
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 {
2   "code": 200,
3   "status": true,
4   "data": {
5     "id": 1,
6     "allergen": "Peanuts",
7     "reaction": "Severe Rash",
8     "severity_scale": 4,
9     "created_at": "2024-10-18T08:46:21.000000Z",
10    "updated_at": "2024-10-18T11:34:29.000000Z"
11  }
12 }
```

HEALTH MANAGEMENT API / http://103.196.155.42/api/kesehatan/riwayat_alergi/

DELETE ▼ http://103.196.155.42/api/kesehatan/riwayat_alergi/

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 {
2   "code": 200,
3   "status": true,
4   "data": true
5 }
```

C. Source Code

Saya membuat beberapa folder di dalam folder lib yakni folder models, services, screens dan widgets.

Untuk penggunaan fonts calibri saya meletakkannya di folder assets.

Folder models yakni file **Riwayat_alergi.dart** :

```
riwayat_alergi.dart •
lib > models > riwayat_alergi.dart > RiwayatAlergi > RiwayatAlergi
1  class RiwayatAlergi {
2      final int? id; // ID bisa null saat menambah data baru
3      final String allergen;
4      final String reaction;
5      final int severityScale;
6
7      // Konstruktor untuk RiwayatAlergi
8      RiwayatAlergi({
9          this.id, |
10         required this.allergen,
11         required this.reaction,
12         required this.severityScale,
13     });
14
15     // Factory method untuk membuat objek dari JSON
16     factory RiwayatAlergi.fromJson(Map<String, dynamic> json) {
17         return RiwayatAlergi(
18             id: json['id'],
19             allergen: json['allergen'],
20             reaction: json['reaction'],
21             severityScale: json['severity_scale'],
22         );
23     }
24
25     // Method untuk mengonversi objek ke format JSON
26     Map<String, dynamic> toJson() {
27         return {
28             'id': id,
29             'allergen': allergen,
30             'reaction': reaction,
31             'severity_scale': severityScale,
32         };
33     }
34 }
35
```

Folder Screens

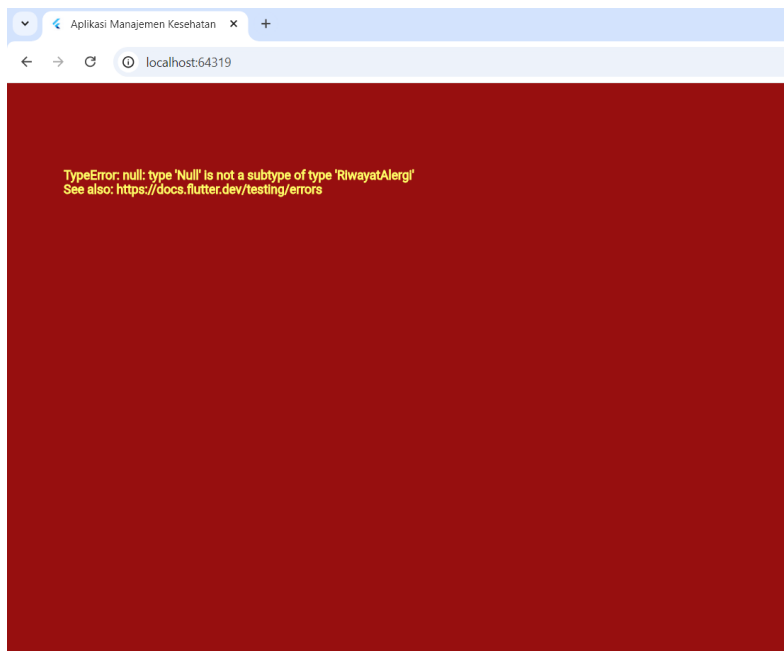
1. alergi_detail_screen.dart

```

riwayat_alergi.dart  alergi_detail_screen.dart X
lib > screens > alergi_detail_screen.dart > ...
1  import 'package:flutter/material.dart';
2  import '../models/riwayat_alergi.dart';
3
4  class AlergiDetailScreen extends StatelessWidget {
5    @override
6    Widget build(BuildContext context) {
7      final RiwayatAlergi alergi = ModalRoute.of(context)!.settings.arguments as RiwayatAlergi;
8
9      return Scaffold(
10        appBar: AppBar(
11          title: Text('Detail Alergi'),
12          backgroundColor: Colors.pink[100],
13        ), // AppBar
14        body: Padding(
15          padding: const EdgeInsets.all(16.0),
16          child: Column(
17            crossAxisAlignment: CrossAxisAlignment.start,
18            children: [
19              Text('Allergen'),
20              SizedBox(height: 10),
21              Text('Reaction'),
22              SizedBox(height: 10),
23              Text('Severity Scale'),
24              SizedBox(height: 20),
25              ElevatedButton(
26                onPressed: () {
27                  Navigator.pushNamed(context, '/edit', arguments: alergi);
28                },
29                child: Text('Ubah Data'),
30                style: ElevatedButton.styleFrom(backgroundColor: Colors.pink[100]),
31              ), // ElevatedButton
32            ],
33          ), // Column
34        ), // Padding
35      ); // Scaffold
36    }
37  }

```

Screenshoot hasil :



2. alergi_list_screen

```

screens > alergi_list_screen.dart > ...
import 'package:flutter/material.dart';
import '../services/alergi_service.dart';
import '../models/riwayat_alergi.dart';

class AlergiListScreen extends StatefulWidget {
  @override
  _AlergiListScreenState createState() => _AlergiListScreenState();
}

class _AlergiListScreenState extends State<AlergiListScreen> {
  late Future<List<RiwayatAlergi>> alergiList;

  @override
  void initState() {
    super.initState();
    alergiList = AlergiService().getRiwayatAlergi();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Riwayat Alergi'),
        backgroundColor: Colors.pink[100],
      ), // AppBar
      body: FutureBuilder<List<RiwayatAlergi>>(
        future: alergiList,
        builder: (context, snapshot) {
          if (snapshot.hasData) {
            return ListView.builder(
              itemCount: snapshot.data!.length,
              itemBuilder: (context, index) {
                RiwayatAlergi alergi = snapshot.data![index];
                return ListTile(
                  title: Text(alergi.allergen),
                  subtitle: Text('Reaksi: ${alergi.reaction}, Skala: ${alergi.severityScale}'),
                  onTap: () {
                    Navigator.pushNamed(context, '/detail', arguments: alergi);
                  },
                );
              },
            );
          } else if (snapshot.hasError) {
            return Center(child: Text('Error: ${snapshot.error}'));
          }
          return Center(child: CircularProgressIndicator());
        },
      ), // FutureBuilder
      floatingActionButton: FloatingActionButton(
        onPressed: () {
          Navigator.pushNamed(context, '/edit', arguments: null);
        },
        child: Icon(Icons.add),
        backgroundColor: Colors.pink[100],
      ), // FloatingActionButton
    ); // Scaffold
  }
}

```

Screenshoot hasil :



3. alergi_edit_screen.dart

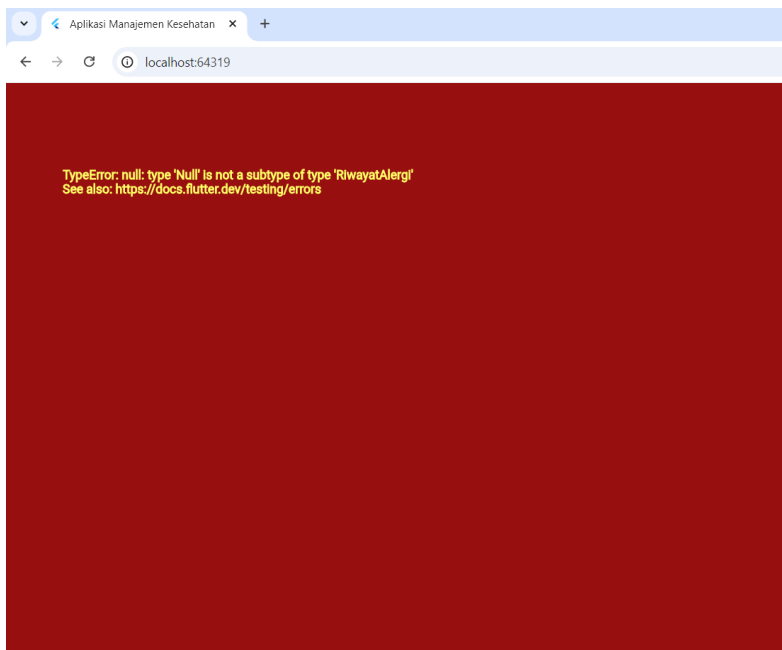
```
alergi_edit_screen.dart
> screens > alergi_edit_screen.dart > ...
1 import 'package:flutter/material.dart';
2 import '../models/riwayat_alergi.dart';
3 import '../services/alergi_service.dart';
4
5 class AlergiEditScreen extends StatefulWidget {
6   @override
7   _AlergiEditScreenState createState() => _AlergiEditScreenState();
8 }
9
10 class _AlergiEditScreenState extends State<AlergiEditScreen> {
11   final _allergenController = TextEditingController();
12   final _reactionController = TextEditingController();
13   final _severityScaleController = TextEditingController();
14
15   void _save(RiwayatAlergi? alergi) {
16     if (alergi == null) {
17       // Tambah data
18       AlergiService().createRiwayatAlergi(
19         RiwayatAlergi(
20           allergen: _allergenController.text,
21           reaction: _reactionController.text,
22           severityScale: int.parse(_severityScaleController.text),
23         ),
24       );
25     } else {
26       // Update data
27       AlergiService().updateRiwayatAlergi(
28         RiwayatAlergi(
29           id: alergi.id,
30           allergen: _allergenController.text,
31           reaction: _reactionController.text,
32           severityScale: int.parse(_severityScaleController.text),
33         ),
34       );
35     }
36
37     Navigator.pop(context);
38   }
39 }
```

String get text
Type: String
package:flutter/src/widgets/editable_text.dart
The current string the user is editing.

```
screens > alergi_edit_screen.dart > ...
class _AlergiEditScreenState extends State<AlergiEditScreen> {
  Widget build(BuildContext context) {
    if (alergi != null) {
      _allergenController.text = alergi.allergen;
      _reactionController.text = alergi.reaction;
      _severityScaleController.text = alergi.severityScale.toString();
    }

    return Scaffold(
      appBar: AppBar(
        title: Text(alergi == null ? 'Tambah Alergi' : 'Ubah Alergi'),
        backgroundColor: Colors.pink[100],
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            TextField(
              controller: _allergenController,
              decoration: InputDecoration(labelText: 'Allergen'),
            ), // TextField
            TextField(
              controller: _reactionController,
              decoration: InputDecoration(labelText: 'Reaction'),
            ), // TextField
            TextField(
              controller: _severityScaleController,
              decoration: InputDecoration(labelText: 'Severity Scale'),
              keyboardType: TextInputType.number,
            ), // TextField
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () => _save(alergi),
              child: Text(alergi == null ? 'Tambah' : 'Ubah'),
              style: ElevatedButton.styleFrom(backgroundColor: Colors.pink[100]),
            ), // ElevatedButton
          ],
        ), // Column
      ),
    );
  }
}
```

Screenshoot hasil :



4. login_screen.dart

```
import 'package:flutter/material.dart';

class LoginScreen extends StatefulWidget {
  @override
  _LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();

  void _login() {
    // Proses Login
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Login'),
        backgroundColor: Colors.pink[100],
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            TextField(
              controller: _emailController,
              decoration: InputDecoration(labelText: 'Email'),
            ), // TextField
            TextField(
              controller: _passwordController,
              decoration: InputDecoration(labelText: 'Password'),
              obscureText: true,
            ), // TextField
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: _login,
            ),
          ],
        ), // Column
      ), // Padding
    ); // Scaffold
  }
}
```

```
lib > screens > login_screen.dart > ...
8   class _LoginScreenState extends State<LoginScreen> {
17   widget build(BuildContext context) {
33       decoration: InputDecoration(labelText: 'Password'),
34       obscureText: true,
35     ), // TextField
36     SizedBox(height: 20),
37     ElevatedButton(
38       onPressed: _login,
39       child: Text('Login'),
40       style: ElevatedButton.styleFrom(
41         backgroundColor: Colors.pink[100],
42       ),
43     ), // ElevatedButton
44   ],
45 ), // Column
46 ), // Padding
47 ); // Scaffold
48 }
49 }
50
```


Screenshoot hasil :

Aplikasi Manajemen Kesehatan

localhost:64319

Login

Email
SYADINA

Password

Login

DEBUG

5. register_screen.dart

```
lib > screens > register_screen.dart > ...
1  import 'package:flutter/material.dart';
2
3  class RegisterScreen extends StatefulWidget {
4    @override
5    _RegisterScreenState createState() => _RegisterScreenState();
6  }
7
8  class _RegisterScreenState extends State<RegisterScreen> {
9    final _emailController = TextEditingController();
10   final _passwordController = TextEditingController();
11
12   void _register() {
13     // Implementasi registrasi
14     Navigator.pushNamed(context, '/login');
15   }
16
17   @override
18   Widget build(BuildContext context) {
19     return Scaffold(
20       appBar: AppBar(
21         title: Text('Registrasi'),
22         backgroundColor: Colors.pink[100],
23       ), // AppBar
24       body: Padding(
25         padding: const EdgeInsets.all(16.0),
26         child: Column(
27           children: [
28             TextField(
29               controller: _emailController,
30               decoration: InputDecoration(labelText: 'Email'),
31             ), // TextField
32             TextField(
33               controller: _passwordController,
34               decoration: InputDecoration(labelText: 'Password'),
35               obscureText: true,
36             ), // TextField
37             SizedBox(height: 20),
38             ElevatedButton(
```

```
riwayat_alergi.dart • register_screen.dart X
lib > screens > register_screen.dart > ...
8 class _RegisterScreenState extends State<RegisterScreen> {
18   Widget build(BuildContext context) {
29     controller: _emailController,
30     decoration: InputDecoration(labelText: 'Email'),
31   ), // TextField
32   TextField(
33     controller: _passwordController,
34     decoration: InputDecoration(labelText: 'Password'),
35     obscureText: true,
36   ), // TextField
37   SizedBox(height: 20),
38   ElevatedButton(
39     onPressed: _register,
40     child: Text('Registrasi'),
41     style: ElevatedButton.styleFrom(
42       backgroundColor: Colors.pink[100],
43     ),
44   ), // ElevatedButton
45 ],
46 ), // Column
47 ), // Padding
48 ); // Scaffold
49 }
50 }
51
```

Folder services

1. alergi_service.dart

```
terminal Help health_management_app
riwayat_alergi.dart • alergi_service.dart •
lib > services > alergi_service.dart > AlergiService
1 import 'dart:convert';
2 import 'package:http/http.dart' as http;
3 import '../models/riwayat_alergi.dart';
4
5 class AlergiService {
6   final String baseUrl = 'http://103.196.155.42/api';
7
8   Future<void> registerUser(String username, String password) async {
9     final response = await http.post(
10       Uri.parse('$baseUrl/register'),
11       headers: <String, String>{
12         'Content-Type': 'application/json; charset=UTF-8',
13       },
14       body: jsonEncode(<String, String>{
15         'username': username,
16         'password': password,
17       })),
18     );
19
20     if (response.statusCode != 201) {
21       throw Exception('Failed to register user');
22     }
23   }
24
25
26   Future<void> loginUser(String username, String password) async {
27     final response = await http.post(
28       Uri.parse('$baseUrl/login'), // Endpoint Login
29       headers: <String, String>{
30         'Content-Type': 'application/json; charset=UTF-8',
31       },
32       body: jsonEncode(<String, String>{
33         'username': username,
34         'password': password,
35       })),
36     );
37
38     if (response.statusCode != 200) {
```

```

> services > alergi_service.dart > AlergiService
5 class AlergiService {
26   Future<void> loginUser(String username, String password) async {
40     }
41   }
42
43   // Endpoint untuk mendapatkan daftar riwayat alergi
44   Future<List<RiwayatAlergi>> getRiwayatAlergi() async {
45     final response = await http.get(Uri.parse('$baseUrl/kesehatan/riwayat_alergi'));
46     if (response.statusCode == 200) {
47       final List<dynamic> jsonResponse = json.decode(response.body);
48       return jsonResponse.map((data) => RiwayatAlergi.fromJson(data)).toList();
49     } else {
50       throw Exception('Failed to load riwayat alergi');
51     }
52   }
53
54   // Endpoint untuk menambah riwayat alergi
55   Future<void> createRiwayatAlergi(RiwayatAlergi alergi) async {
56     final response = await http.post(
57       Uri.parse('$baseUrl/kesehatan/riwayat_alergi'),
58       headers: <String, String>{
59         'Content-Type': 'application/json; charset=UTF-8',
60       },
61       body: jsonEncode(alergi.toJson()),
62     );
63
64     if (response.statusCode != 201) {
65       throw Exception('Failed to add riwayat alergi');
66     }
67   }
68
69   // Endpoint untuk memperbarui riwayat alergi
70   Future<void> updateRiwayatAlergi(RiwayatAlergi alergi) async {
71     final response = await http.put(
72       Uri.parse('$baseUrl/kesehatan/riwayat_alergi/${alergi.id}'),
73       headers: <String, String>{
74         'Content-Type': 'application/json; charset=UTF-8',
75       },

```

```
Terminal Help ← → health_management_app
riwayat_alergi.dart ● alergi_service.dart ●
lib > services > alergi_service.dart > AlergiService
5 class AlergiService {
55   Future<void> createRiwayatAlergi(RiwayatAlergi alergi) async {
65     | throw Exception('Failed to add riwayat alergi');
66   }
67 }
68
69 // Endpoint untuk memperbarui riwayat alergi
70 Future<void> updateRiwayatAlergi(RiwayatAlergi alergi) async {
71   final response = await http.put(
72     Uri.parse('$baseUrl/kesehatan/riwayat_alergi/${alergi.id}'),
73     headers: <String, String>{
74       'Content-Type': 'application/json; charset=UTF-8',
75     },
76     body: jsonEncode(alergi.toJson()),
77   );
78
79   if (response.statusCode != 200) {
80     | throw Exception('Failed to update riwayat alergi');
81   }
82 }
83
84 // Endpoint untuk menghapus riwayat alergi
85 Future<void> deleteRiwayatAlergi(int id) async {
86   final response = await http.delete(
87     Uri.parse('$baseUrl/kesehatan/riwayat_alergi/$id'),
88   );
89
90   if (response.statusCode != 204) {
91     | throw Exception('Failed to delete riwayat alergi');
92   }
93 }
94 }
95 }
```

Folder widgets

1. custom_app_bar.dart

```
Terminal Help ← → health_management_app
riwayat_alergi.dart ● alergi_service.dart ● custom_app_bar.dart ✕
lib > widgets > custom_app_bar.dart > ...
1 import 'package:flutter/material.dart';
2
3 class CustomAppBar extends StatelessWidget implements PreferredSizeWidget {
4   final String title;
5   final List<Widget>? actions;
6
7   const CustomAppBar({
8     Key? key,
9     required this.title,
10    this.actions,
11  }) : super(key: key);
12
13   @override
14   Widget build(BuildContext context) {
15     return AppBar(
16       title: Text(
17         title,
18         style: TextStyle(
19           fontFamily: 'calibri', // Font Calibri
20           fontSize: 20, // Ukuran font judul
21         ), // TextStyle
22       ), // Text
23       backgroundColor: Colors.pink[100], // warna pastel
24       actions: actions,
25     ); // AppBar
26   }
27
28   @override
29   Size get preferredSize => Size.fromHeight(kToolbarHeight);
30 }
31 }
```

2. custom_elevated_button.dart

```
lib > widgets > custom_elevated_button.dart > ...
1 import 'package:flutter/material.dart';
2
3 class CustomElevatedButton extends StatelessWidget {
4   final String text;
5   final VoidCallback onPressed;
6
7   const CustomElevatedButton({
8     Key? key,
9     required this.text,
10    required this.onPressed,
11  }) : super(key: key);
12
13  @override
14  Widget build(BuildContext context) {
15    return ElevatedButton(
16      onPressed: onPressed,
17      child: Text(
18        text,
19        style: TextStyle(
20          fontFamily: 'Calibri', // Font Calibri
21          fontSize: 16, // Ukuran font
22        ), // TextStyle
23      ), // Text
24      style: ElevatedButton.styleFrom(
25        backgroundColor: Colors.pink[100], // Warna pastel
26        padding: EdgeInsets.symmetric(vertical: 12, horizontal: 24),
27        shape: RoundedRectangleBorder(
28          borderRadius: BorderRadius.circular(8.0),
29        ), // RoundedRectangleBorder
30      ),
31    ); // ElevatedButton
32  }
33 }
34
```

Main.dart

```
Terminal Help < -> health_management_app
riwayat_alergi.dart alergi_service.dart main.dart 8
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2 import 'widgets/custom_elevated_button.dart';
3 import 'widgets/custom_app_bar.dart';
4 import 'screens/login_screen.dart';
5 import 'screens/register_screen.dart';
6 import 'screens/alergi_list_screen.dart';
7 import 'screens/alergi_detail_screen.dart';
8 import 'screens/alergi_edit_screen.dart';
9
10 void main() {
11   runApp(MyApp());
12 }
13
14 class MyApp extends StatelessWidget {
15   @override
16   Widget build(BuildContext context) {
17     return MaterialApp(
18       title: 'Aplikasi Manajemen Kesehatan',
19       theme: ThemeData(
20         primarySwatch: Colors.pink,
21       ), // ThemeData
22       home: LoginScreen(),
23     ); // MaterialApp
24   }
25 }
26
```