

LAPORAN PRAKTIKUM

TUGAS PEMBUATAN HALAMAN LOGIN BERNAVIGASI MENUJU HALAMAN DASHBOARD

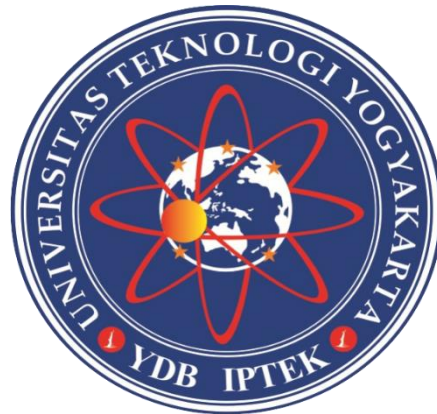
Disusun untuk Memenuhi Tugas Mata Kuliah Mobile & Web Service Praktik

Dosen Pengampu:

Suyud Widiono, S.Pd., M.Kom.

Asisten Dosen:

Margareta Dyah Ayu Christiasih



Disusun oleh:

Alfian Setya Dwi Saputra (5220411164)

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

2024

DAFTAR ISI

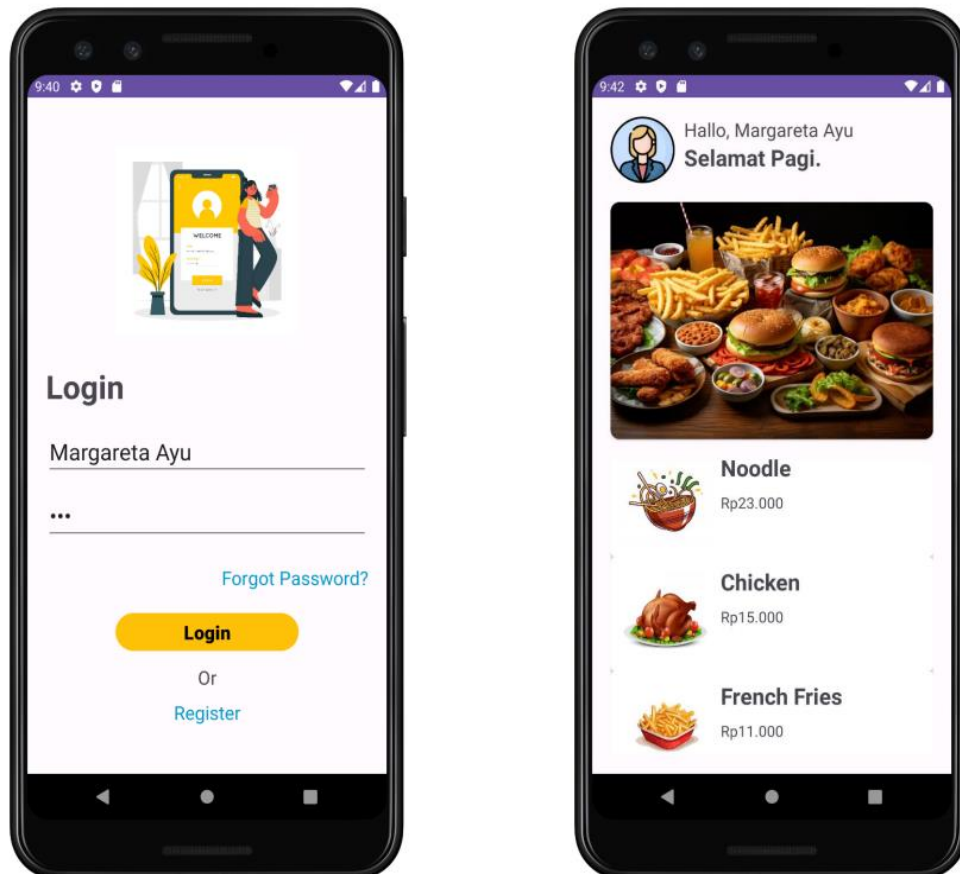
DAFTAR ISI.....	i
BAB I PEMBUATAN HALAMAN LOGIN BERNAVIGASI MENUJU DASHBOARD	1
1.1. Praktikum.....	1
1.1.1. Soal Tugas	1
1.1.2. Kode Pemrograman	1
1.1.3. Penjelasan Kode.....	7
1.1.4. Hasil.....	11
REFERENSI.....	15

BAB I

PEMBUATAN HALAMAN LOGIN BERNAVIGASI MENUJU DASHBOARD

1.1. Praktikum

1.1.1. Soal Tugas



Studi kasus ini melibatkan pembuatan aplikasi pemesanan makanan menggunakan Flutter, di mana pengguna harus login dengan nama pengguna untuk dapat mengakses halaman dashboard. Halaman login terdiri dari input teks untuk memasukkan nama pengguna, dan setelah pengguna menekan tombol "Login", mereka akan diarahkan ke halaman dashboard. Pada halaman dashboard, aplikasi menampilkan sapaan personal dengan nama pengguna yang diinput serta daftar menu makanan beserta harganya.

1.1.2. Kode Pemrograman

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

void main() {
  runApp(const MyApp());
}
```

```

}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Login Navigation Example',
      theme: ThemeData(
        primarySwatch: Colors.purple,
      ),
      debugShowCheckedModeBanner: false,
      home: const LoginPage(),
    );
  }
}

class LoginPage extends StatefulWidget {
  const LoginPage ({super.key});

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final TextEditingController _usernameController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();

  void _login() {
    if (_usernameController.text == "Alvinz" && _passwordController.text ==
"12345") {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => DashboardPage(username:
_usernameController.text, onLogout: _logout)),
      );
    } else {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Username atau password salah!')),
      );
    }
  }

  void _logout() {
    _usernameController.clear();
    _passwordController.clear();
  }
}

```

```

Navigator.pop(context);
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(toolbarHeight: 10),
    body: Padding(
      padding: const EdgeInsets.all(30.0),
      child: SingleChildScrollView(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            SizedBox(
              height: 300,
              child: Image.asset('assets/images/Login.jpeg', fit: BoxFit.contain),
            ),
            const Align(
              alignment: Alignment.centerLeft,
              child: Text(
                'Login',
                style: TextStyle(fontSize: 40, fontWeight: FontWeight.bold),
              ),
            ),
            const SizedBox(height: 10),
            SizedBox(
              width: double.infinity,
              child: TextField(
                controller: _usernameController,
                decoration: const InputDecoration(
                  labelText: 'Username',
                  border: OutlineInputBorder(),
                  contentPadding: EdgeInsets.symmetric(vertical: 15, horizontal: 16),
                ),
                style: const TextStyle(fontSize: 20),
              ),
            ),
            const SizedBox(height: 20),
            SizedBox(
              width: double.infinity,
              child: TextField(
                controller: _passwordController,
                obscureText: true,
                decoration: const InputDecoration(
                  labelText: 'Password',
                  border: OutlineInputBorder(),
                  contentPadding: EdgeInsets.symmetric(vertical: 15, horizontal: 16),
                ),
              ),
            ),
          ],
        ),
      ),
    ),
  );
}

```

```

        style: const TextStyle(fontSize: 20),
      ),
    ),
    const SizedBox(height: 5),
    Align(
      alignment: Alignment.centerRight,
      child: TextButton(
        onPressed: () {
        },
        child: const Text(
          'Forgot Password?',
          style: TextStyle(color: Colors.blue, fontSize: 20),
        ),
      ),
    ),
    const SizedBox(height: 15),
    SizedBox(
      width: 250,
      height: 50,
      child: ElevatedButton(
        onPressed: _login,
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.orange,
        ),
        child: const Text(
          'Login',
          style: TextStyle(
            color: Colors.black,
            fontWeight: FontWeight.bold,
            fontSize: 20,
          ),
        ),
      ),
    ),
    const SizedBox(height: 17),
    const Text('Or', style: TextStyle(fontSize: 20)),
    TextButton(
      onPressed: () {
      },
      child: const Text(
        'Register',
        style: TextStyle(color: Colors.blue, fontSize: 20),
      ),
    ),
  ],
),
),
),
),

```

```

    );
  }
}

class DashboardPage extends StatelessWidget {
  final String username;
  final VoidCallback onLogout;

  const DashboardPage({required this.username, required this.onLogout,
    super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        toolbarHeight: 15,
        automaticallyImplyLeading: false,
      ),
      body: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Row(
                    children: [
                      const CircleAvatar(
                        radius: 30,
                        child: Icon(Icons.person, size: 40),
                      ),
                      const SizedBox(width: 15),
                      Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          Text(
                            'Hallo, $username',
                            style: const TextStyle(fontSize: 20),
                          ),
                          const Text(
                            'Selamat Makan.',
                            style: TextStyle(fontSize: 23, fontWeight: FontWeight.bold),
                          ),
                        ],
                      ),
                    ],
                  ),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}

```

```

    ),
    ElevatedButton(
      onPressed: onLogout,
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.red,
      ),
      child: const Text(
        'Log out',
        style: TextStyle(
          color: Colors.white,
          fontWeight: FontWeight.bold,
          fontSize: 20,
        ),
      ),
    ),
  ],
),
Image.asset('assets/images/Dashboard.jpeg', height: 385),
const SizedBox(height: 10),
Row(
  children: [
    SizedBox(
      width: 150,
      height: 150,
      child: Image.asset('assets/images/Mie Ayam Bakso.jpeg', fit:
BoxFit.cover),
    ),
    const SizedBox(width: 30),
    const Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text('Mie Ayam Bakso', style: TextStyle(fontSize: 30, fontWeight:
FontWeight.bold)),
        Text('Rp20.000', style: TextStyle(fontSize: 30)),
      ],
    ),
  ],
),
const SizedBox(height: 10),
Row(
  children: [
    SizedBox(
      width: 150,
      height: 150,
      child: Image.asset('assets/images/Mie Ayam.jpeg', fit: BoxFit.cover),
    ),
    const SizedBox(width: 30),
    const Column(

```



```

        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text('Mie Ayam', style: TextStyle(fontSize: 30, fontWeight:
FontWeight.bold)),
          Text('Rp15.000', style: TextStyle(fontSize: 30)),
        ],
      ),
    ],
  ),
  const SizedBox(height: 20),
  Row(
    children: [
      SizedBox(
        width: 150,
        height: 120,
        child: Image.asset('assets/images/Bakso.jpeg', fit: BoxFit.cover),
      ),
      const SizedBox(width: 30),
      const Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text('Bakso', style: TextStyle(fontSize: 30, fontWeight:
FontWeight.bold)),
          Text('Rp10.000', style: TextStyle(fontSize: 30)),
        ],
      ),
    ],
  ),
],
),
],
),
),
);
}
}

```

1.1.3. Penjelasan Kode

1. Blok MyApp (Root Widget)

Blok ini adalah root atau titik awal aplikasi, dan karena aplikasi Flutter dibangun dengan sistem hirarki widget, maka MyApp adalah widget paling atas dalam hirarki ini. Ini adalah widget Stateless, yang berarti widget ini tidak memiliki state yang berubah selama siklus hidupnya. Penjelasan mendetail dari setiap elemen dalam MyApp:

- **runApp(const MyApp()):** Fungsi ini digunakan untuk memulai aplikasi Flutter. Ini mengambil widget yang disediakan (dalam hal ini MyApp) dan menampilkannya sebagai widget root aplikasi.
- **MyApp class:** Merupakan class yang mewarisi dari StatelessWidget, yang berarti tidak ada state yang dapat berubah selama masa hidup widget ini. Pada MyApp, properti build mengembalikan MaterialApp, yang merupakan widget pembungkus dasar dari aplikasi Flutter untuk menyediakan fitur material design.
- **MaterialApp:**
 - **title:** Ini menentukan judul aplikasi yang mungkin ditampilkan pada beberapa platform.
 - **theme:** Aplikasi menggunakan tema material design dengan palet warna utama ungu (purple). Properti primarySwatch menentukan warna utama untuk aplikasi, di mana berbagai variasi dari warna ini (seperti ungu muda atau tua) akan digunakan di seluruh elemen aplikasi seperti AppBar, tombol, dsb.
 - **debugShowCheckedModeBanner: false:** Menghilangkan label debug di kanan atas layar saat aplikasi dijalankan dalam mode debug.
 - **home: const LoginPage():** Halaman pertama atau beranda aplikasi ditentukan di sini, yaitu LoginPage. Karena LoginPage adalah stateful widget, const digunakan untuk membuatnya lebih efisien (hanya jika widget tidak memiliki state yang perlu diperbarui saat runtime).

2. Blok LoginPage (Stateful Widget)

LoginPage adalah halaman login yang merupakan stateful widget. Ini berarti halaman ini dapat berubah (misalnya, ketika pengguna memasukkan input atau ketika status login berubah).

- **StatefulWidget:** LoginPage mewarisi StatefulWidget, yang memungkinkan halaman ini memiliki state yang dapat diperbarui. Saat halaman login dimodifikasi (misalnya, ketika pengguna memasukkan nama pengguna atau kata sandi), state akan di-update dan widget akan di-render ulang sesuai dengan perubahan tersebut.

- **_LoginPageState:** Ini adalah class yang bertanggung jawab mengelola state dari LoginPage. Class ini mengontrol tampilan dan perilaku halaman login, serta menangani logika validasi login. Penjelasan elemen di dalamnya:
- **TextEditingController:** Dua controller (`_usernameController` dan `_passwordController`) digunakan untuk menangkap input pengguna pada field teks username dan password. Controller ini juga berguna untuk mengatur, mendapatkan, atau membersihkan teks dari TextField.
- **_login() method:** Metode ini adalah logika yang di-trigger ketika tombol "Login" ditekan. Fungsinya:
 - Cek apakah kombinasi username dan password benar (dalam contoh ini, username harus "Alvinz" dan password "12345").
 - Jika benar, aplikasi melakukan navigasi ke halaman DashboardPage dengan membawa parameter username dan callback onLogout. Navigasi ini dilakukan menggunakan `Navigator.push`, yang menambahkan halaman baru di atas stack navigasi.
 - Jika salah, aplikasi menampilkan pesan kesalahan menggunakan `ScaffoldMessenger` dengan `SnackBar` yang memberikan feedback kepada pengguna bahwa username atau password salah.
- **_logout() method:** Ini adalah callback yang akan dipanggil saat pengguna menekan tombol logout di DashboardPage. Fungsinya membersihkan field input di halaman login (menggunakan `clear()` pada controller), dan kemudian kembali ke halaman login menggunakan `Navigator.pop`, yang menghapus halaman DashboardPage dari stack navigasi.

3. Build Method dari LoginPage

Bagian `build()` method dari LoginPage berisi logika untuk membangun tampilan halaman login. Hal ini mencakup beberapa elemen dan layout penting:

- **Scaffold:** Scaffold adalah widget kerangka yang menyediakan struktur dasar halaman, termasuk elemen seperti AppBar, Body, FloatingActionButton, dll.
 - **AppBar:** Bagian ini hanya diatur dengan tinggi toolbar yang sangat minimal, tanpa judul atau elemen interaktif lain.
- **Padding dan SingleChildScrollView:**
 - Padding memberikan ruang di sekeliling konten halaman untuk memastikan komponen tidak menempel pada tepi layar.

- `SingleChildScrollView` memungkinkan konten di-scroll jika layar tidak cukup besar untuk menampilkan seluruh komponen, memastikan aplikasi responsif di berbagai ukuran layar.
- **Column:** Mengatur tata letak komponen secara vertikal, dimulai dari gambar ilustrasi login, judul "Login", dan input teks untuk username dan password.
- **TextField untuk Username dan Password:** Masing-masing input teks disediakan oleh widget `TextField`, di mana:
 - Controller mengikat nilai teks yang dimasukkan pengguna.
 - `obscureText: true` pada password digunakan untuk menyembunyikan input pengguna, sehingga tampak seperti bintang (*).
- **Tombol Login dan Register:**
 - Tombol "Login" adalah widget `ElevatedButton` yang, saat ditekan, memicu metode `_login()`.
 - Tombol "Register" dan "Forgot Password" menggunakan `TextButton` yang disetel untuk memberikan interaksi lebih lanjut, meski saat ini tidak diimplementasikan.

4. Blok `DashboardPage` (Stateless Widget)

`DashboardPage` adalah halaman yang dituju setelah login berhasil. Ini adalah `StatelessWidget`, artinya halaman ini tidak memiliki state yang berubah, tetapi menerima data dinamis melalui parameter konstruktor.

- **Parameter username dan onLogout:** Halaman ini menerima username (nama pengguna) yang ditampilkan pada UI dan callback `onLogout` yang dipanggil ketika pengguna logout.
- **AppBar dan Row:** Halaman ini menggunakan `AppBar` dengan toolbar minimalis. Pada bagian body, elemen-elemen ditata menggunakan `Column` dan `Row`:
 - **Sapaan Personal:** Menggunakan `Row` yang berisi `CircleAvatar` sebagai ikon pengguna dan teks yang menyapa pengguna dengan nama yang dimasukkan saat login. Ini dibuat lebih personal dengan menampilkan "Halo, (username)".
 - **Tombol Logout:** Terdapat tombol "Log out" yang diatur dengan `ElevatedButton`. Saat ditekan, ini akan memanggil callback `onLogout`, yang men-trigger logout dan membawa pengguna kembali ke halaman login.

- **Menu Makanan:** Bagian selanjutnya dari DashboardPage adalah daftar makanan. Masing-masing menu ditampilkan menggunakan kombinasi Row dan Column yang menampilkan gambar makanan di sebelah kiri dan informasi nama serta harga makanan di sebelah kanan.

5. Navigasi Antar Halaman

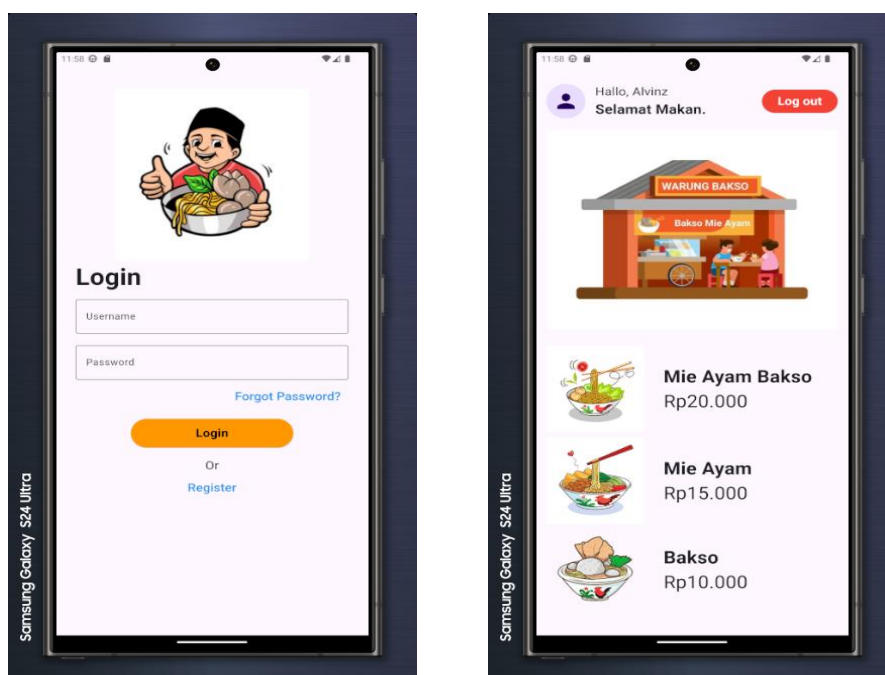
Navigasi di aplikasi ini dikelola dengan Navigator. Ketika login berhasil:

- **Navigator.push:** Menambahkan halaman DashboardPage ke stack navigasi, menggantikan halaman login. Ini dilakukan ketika login sukses melalui metode `_login`.
- **Navigator.pop:** Menghapus halaman DashboardPage dari stack dan mengembalikan ke halaman login ketika pengguna logout.

Kesimpulan Utama:

- **Modularitas:** Aplikasi dibagi menjadi dua halaman utama dengan pemisahan yang jelas antara halaman login dan dashboard.
- **State Management:** Input pengguna dikelola dengan baik menggunakan `TextEditingController`, dan navigasi diatur dengan Navigator.
- **Responsif:** Layout menggunakan `SingleChildScrollView` dan `Column/Row` untuk memastikan elemen bisa menyesuaikan diri dengan ukuran layar yang berbeda.

1.1.4. Hasil



❖ Halaman Login (LoginPage)

- Pada halaman login, pengguna diminta untuk memasukkan nama pengguna (username) dan kata sandi (password). Struktur dari halaman ini terdiri dari beberapa widget utama yang ditata dalam Column untuk mengatur tata letak elemen-elemen tersebut secara vertikal. Elemen-elemen ini meliputi input teks, tombol login, serta tautan untuk memulihkan kata sandi dan melakukan registrasi.
- **Input Username dan Password:** Dua TextField digunakan untuk menangkap input dari pengguna. Masing-masing TextField dikontrol oleh TextEditingController, yang memungkinkan aplikasi mengambil nilai dari input dan menggunakannya dalam proses validasi. Field untuk password menggunakan properti obscureText: true untuk menyembunyikan karakter saat pengguna mengetik, menjaga keamanan kata sandi. Selain itu, widget ini juga dibungkus dalam SizedBox untuk memastikan lebar dan padding yang konsisten pada kedua input field.
- **Fungsi Login:** Tombol login diimplementasikan menggunakan widget ElevatedButton. Ketika ditekan, tombol ini memanggil metode _login(), yang berfungsi untuk memvalidasi input pengguna. Jika nama pengguna dan kata sandi sesuai dengan data yang dihardcode dalam aplikasi (Alvinz sebagai username dan 12345 sebagai password), pengguna akan diarahkan ke halaman dashboard menggunakan Navigator.push(). Jika input tidak sesuai, aplikasi akan menampilkan pesan kesalahan menggunakan ScaffoldMessenger yang muncul di bagian bawah layar sebagai **SnackBar**.
- **Navigasi Tautan:** Elemen lain yang ditemukan di halaman login adalah teks tautan **Forgot Password?** dan **Register**, yang ditampilkan sebagai TextButton. Walaupun tidak diimplementasikan dalam kode ini, tombol tersebut biasanya digunakan untuk navigasi ke halaman pemulihan kata sandi atau pendaftaran akun baru. Teks ini diformat dengan ukuran font yang lebih kecil dan berwarna biru, memberikan isyarat visual yang lazim digunakan untuk menunjukkan elemen yang dapat diklik.
- **Desain dan Gambar:** Di bagian atas halaman login terdapat gambar logo yang ditampilkan menggunakan Image.asset(), mencerminkan identitas aplikasi. Ini membantu memberikan pengalaman pengguna yang lebih familiar dan menarik secara visual.

❖ Halaman Dashboard (DashboardPage)

- Setelah pengguna berhasil login, mereka akan diarahkan ke halaman dashboard, yang menampilkan informasi selamat datang, daftar menu makanan, serta fitur untuk logout. Halaman ini dirancang untuk memberikan tampilan informasi makanan dengan harga yang relevan, sambil tetap menjaga navigasi yang mudah dan fungsional.
- **Tampilan Pengguna dan Logout:** Di bagian atas halaman, pengguna disambut dengan sapaan pribadi berdasarkan input username dari halaman login. Sapaan ini ditampilkan dalam dua baris teks: baris pertama menyebutkan nama pengguna, sedangkan baris kedua memberikan pesan "Selamat Makan." Di samping sapaan, terdapat gambar avatar berbentuk lingkaran, yang biasanya digunakan untuk menunjukkan profil pengguna, serta tombol **Log out** berwarna merah yang digunakan untuk keluar dari aplikasi. Tombol logout ini memanggil fungsi onLogout, yang membersihkan input login pengguna dan mengarahkan mereka kembali ke halaman login.
- **Gambar Utama Warung:** Tepat di bawah sapaan pengguna, ada gambar besar yang menggambarkan warung bakso (kemungkinan salah satu fokus utama dari aplikasi), memberikan nuansa visual yang sesuai dengan tema makanan. Gambar ini juga diambil dari aset lokal menggunakan Image.asset(), dan ukurannya cukup besar untuk memberikan kesan visual yang dominan pada halaman.
- **Daftar Menu Makanan:** Daftar menu makanan diimplementasikan dalam beberapa Row, di mana setiap menu makanan memiliki gambar yang menunjukkan jenis makanannya, serta teks yang menunjukkan nama makanan dan harganya. Misalnya, ada tiga jenis makanan yang ditampilkan: **Mie Ayam Bakso**, **Mie Ayam**, dan **Bakso**, masing-masing dengan harga yang berbeda. Gambar makanan ini diambil dari aset aplikasi dan ditampilkan dengan ukuran yang cukup besar di sebelah kiri teks, memberikan representasi visual dari setiap item menu.
- **Tata Letak:** Tata letak halaman dashboard dirancang untuk memberikan pengalaman yang mudah bagi pengguna. Gambar makanan diatur dalam Row, dengan jarak antar elemen yang cukup, memberikan kesan bersih dan teratur. Teks nama dan harga makanan ditampilkan dalam ukuran font yang besar dan

tebal, memudahkan pengguna untuk membaca dan membandingkan harga makanan yang ditawarkan.

REFERENSI

<https://www.freecodecamp.org/news/how-to-build-a-simple-login-app-with-flutter/>
<https://www.loginradius.com/blog/engineering/guest-post/authenticating-flutter-apps/>
<https://medium.com/flutter-developer-indonesia/membuat-aplikasi-form-login-dan-validation-sederhana-pada-flutter-b695a4d5f9d5>
<https://medium.com/@anyeon/flutter-navigasi-8749face63cb>
<https://nextgen.co.id/flutter-navigator/>
<https://medium.com/@rijalprabesh145/a-complete-guide-to-flutters-singlechildscrollview-widget-7715a1058853>