

LAPORAN TUGAS PRAKTIKUM
MOBILE & WEB SERVICE PRAKTIK

Pertemuan ke-4

Semester Ganjil TA. 2024/2025

Tugas Membuat Halaman Login dan Halaman Dashbord

Dosen Pengampu: Suyud Widiono, S.Pd., M.Kom.



Disusun oleh :

Regita Cahya Arrahma (5220411359)

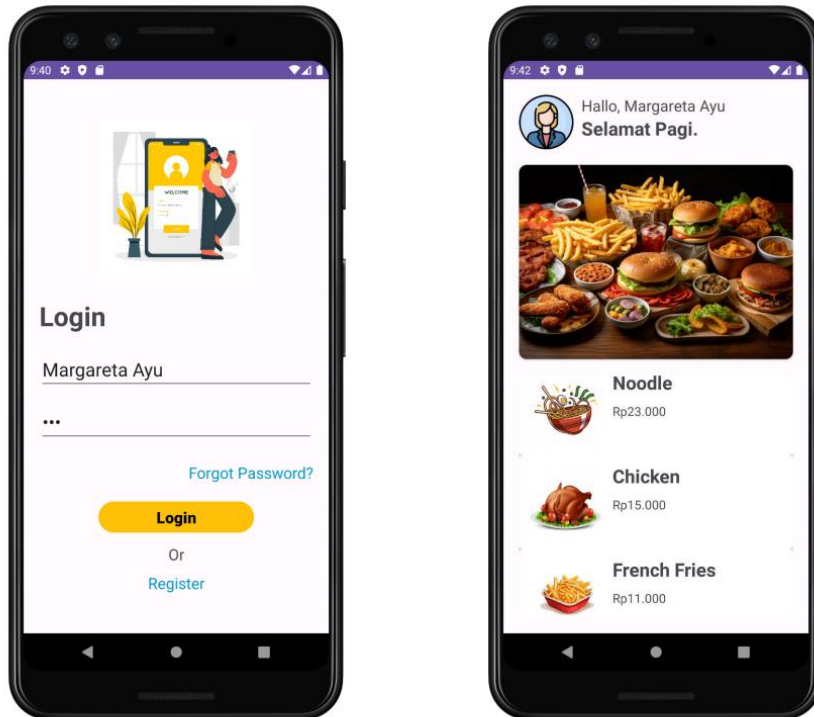
**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024**

DAFTAR ISI

DAFTAR ISI.....	i
1. Soal	1
2. Praktikum	1
2.1. Kode Program.....	1
2.2. Penjelasan.....	9
2.3. Hasil	14
REFERENSI.....	16

1. Soal

Buatlah program widget flutter seperti pada gambar dibawah ini!



2. Praktikum

2.1. Kode Program

Berikut adalah kode programnya:

```
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',
      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 == "Regita" && _passwordController.text ==
    "123") {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => DashboardPage(username:
        _usernameController.text, onLogout: _logout)),
      );
    } else {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Username atau password yang kamu
        masukan salah, silahkan coba lagi!')),
      );
    }
  }

  void _logout() {
    // Reset nama pengguna dan kata sandi saat Logout
    _usernameController.clear();
    _passwordController.clear();

    // Kembali ke Login page
    Navigator.pop(context);
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(toolbarHeight: 10),
      body: Padding(
        padding: const EdgeInsets.all(30.0),

```

```

        child: SingleChildScrollView( // Menggunakan SingleChildScrollView
        child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                SizedBox(
                    height: 300,
                    child: Image.asset('assets/images/logo.png', fit:
BoxFit.contain),
                ),
                // Teks Login
                const Align(
                    alignment: Alignment.centerLeft,
                    child: Text(
                        'Login',
                        style: TextStyle(fontSize: 40, fontWeight:
FontWeight.bold),
                    ),
                ),
                const SizedBox(height: 10),
                // Input Username
                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), // Memperbesar teks
                    ),
                ),
                const SizedBox(height: 20),

                // TextField Password
                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), // Memperbesar teks

```

```

    ),
  ),
  // "Forgot Password" di sebelah kanan dengan warna biru
  const SizedBox(height: 5),
  Align(
    alignment: Alignment.centerRight,
    child: TextButton(
      onPressed: () {
        // Forgot password action
      },
      child: const Text(
        'Forgot Password?',
        style: TextStyle(color: Colors.blue, fontSize: 20), //
Mengubah warna menjadi biru
      ),
    ),
  ),
  // Tombol Login dengan warna kuning, teks italic hitam dan
panjang sesuai keinginan
  const SizedBox(height: 15),
  SizedBox(
    width: 250, // Panjang tombol sesuai keinginan
    height: 50, // Tinggi tombol tetap 50
    child: ElevatedButton(
      onPressed: _login,
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.red, // Mengubah warna tombol
menjadi kuning
      ),
      child: const Text(
        'Login',
        style: TextStyle(
          color: Colors.white, // Warna teks menjadi hitam
          fontWeight: FontWeight.bold,
          fontSize: 20,
        ),
      ),
    ),
  ),
  // Spasi di atas dan bawah teks "Or"
  const SizedBox(height: 17), // Spasi di atas
  const Text('Or', style: TextStyle(fontSize: 20)), // Teks
"Or"

  // Tombol Register dengan warna biru
  TextButton(
    onPressed: () {
      // Register action
    },

```

```

        child: const Text(
          'Register',
          style: TextStyle(color: Colors.blue, fontSize: 20), //
Mengubah warna menjadi biru
        ),
      ),
    ],
  ),
),
);
}
}

```

```

class DashboardPage extends StatelessWidget {
  final String username;
  final VoidCallback onLogout; // Callback for Logout

  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( // Menambahkan 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(

```

```

        'Halo, $username',
        style: const TextStyle(fontSize: 18),
      ),
      const Text(
        'Itadakimasu.',
        style: TextStyle(fontSize: 23, fontWeight:
FontWeight.bold),
      ),
    ],
  ),
],
),
ElevatedButton(
  onPressed: onLogout, // Perform Logout
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.red,
  ),
  child: const Text(
    'Log out',
    style: TextStyle(
      color: Colors.white,
      fontWeight: FontWeight.bold,
      fontSize: 20,
    ),
  ),
),
],
),
const SizedBox(height: 20),

// Centering kedai.png
Center(
  child: Image.asset('assets/images/kedai.png', height: 260,
fit: BoxFit.fill),
),

const SizedBox(height: 20),

// Menambahkan text "Our Menu" ditengah
const Center(
  child: Text(
    '-- Our Menu --',
    style: TextStyle(
      fontSize: 24,
      fontWeight: FontWeight.bold,
    ),
  ),
),
),
),

```



```

const SizedBox(height: 10),

// Ramen Udon Row
Row(
  children: [
    SizedBox(
      width: 130,
      height: 130,
      child: Image.asset('assets/images/udon.png', fit:
BoxFit.cover),
    ),
    const SizedBox(width: 30),
    const Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text('Ramen Udon', style: TextStyle(fontSize: 20,
fontWeight: FontWeight.bold)),
        Text('Rp34.000', style: TextStyle(fontSize: 20)),
      ],
    ),
  ],
),
const SizedBox(height: 10),

// Kurume Ramen Row
Row(
  children: [
    SizedBox(
      width: 130,
      height: 130,
      child: Image.asset('assets/images/kurume.png', fit:
BoxFit.cover),
    ),
    const SizedBox(width: 30),
    const Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text('Kurume Ramen', style: TextStyle(fontSize: 20,
fontWeight: FontWeight.bold)),
        Text('Rp45.000', style: TextStyle(fontSize: 20)),
      ],
    ),
  ],
),
const SizedBox(height: 10),

// Shoyu Ramen Row
Row(

```

```

        children: [
          SizedBox(
            width: 130,
            height: 130,
            child: Image.asset('assets/images/shoyu.png', fit:
BoxFit.cover),
          ),
          const SizedBox(width: 30),
          const Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text('Shoyu Ramen', style: TextStyle(fontSize: 20,
fontWeight: FontWeight.bold)),
              Text('Rp48.500', style: TextStyle(fontSize: 20)),
            ],
          ),
        ],
      ),
    ],
  ),

  // Spicy Ramen Row
  Row(
    children: [
      SizedBox(
        width: 130,
        height: 130,
        child: Image.asset('assets/images/spicy.png', fit:
BoxFit.cover),
      ),
      const SizedBox(width: 30),
      const Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text('Spicy Ramen', style: TextStyle(fontSize: 20,
fontWeight: FontWeight.bold)),
          Text('Rp50.000', style: TextStyle(fontSize: 20)),
        ],
      ),
    ],
  ),
],
),
],
),
),
);
}
}

```

2.2. Penjelasan

Berikut ini adalah penjelasan dari kode program diatas:

1) `runApp(const MyApp())`:

`runApp` adalah fungsi yang pertama kali dipanggil untuk menjalankan aplikasi Flutter. Fungsi ini menerima sebuah widget dan menampilkannya sebagai root widget dari aplikasi. Dalam hal ini, widget yang diberikan adalah `MyApp`.

2) `MyApp (StatelessWidget)`:

- **StatelessWidget**: Widget ini digunakan karena `MyApp` tidak memerlukan perubahan data secara dinamis. Aplikasi ini hanya mendefinisikan tampilan statis seperti tema dan halaman pertama yang ditampilkan.
- **MaterialApp**: Widget utama yang membungkus aplikasi dan menyediakan berbagai properti seperti tema, navigasi, dan konfigurasi lainnya.
 - **title**: Menentukan judul aplikasi, yang biasanya muncul di taskbar atau recent apps.
 - **theme**: Mengatur tema global aplikasi. Dalam kode ini, tema menggunakan `primarySwatch: Colors.purple`, yang akan memberi warna ungu ke berbagai komponen Material (seperti `AppBar`).
 - **debugShowCheckedModeBanner: false**: Menyembunyikan label debug default di pojok kanan atas aplikasi.
 - **home: LoginPage()**: Menentukan halaman awal yang ditampilkan saat aplikasi diluncurkan, yaitu halaman login (`LoginPage`)

3) `LoginPage (StatefulWidget)`:

- **StatefulWidget**: `LoginPage` adalah widget yang dapat berubah (dinamis), karena ada input pengguna (username dan password) yang perlu diproses, serta fungsi login yang bisa memengaruhi tampilan.
- **TextEditingController**: Kontroler ini digunakan untuk mengambil dan memanipulasi teks yang dimasukkan pengguna ke dalam `TextField`. Di sini ada dua kontroler:
 - **_usernameController**: Mengontrol input username.
 - **_passwordController**: Mengontrol input password.
- **Fungsi `_login()`**:

- Fungsi ini memeriksa apakah username yang dimasukkan adalah "Regita" dan password "123". Jika benar, aplikasi akan beralih ke halaman dashboard (`DashboardPage`).
- Jika salah, `SnackBar` akan ditampilkan, yang menunjukkan pesan kesalahan kepada pengguna.
- `Navigator.push`: Digunakan untuk berpindah halaman dari login ke dashboard. `MaterialPageRoute` dipakai untuk mendefinisikan transisi halaman baru (`DashboardPage`).
- Fungsi `_logout()`:
 - Fungsi ini membersihkan (reset) input username dan password saat pengguna logout.
 - `Navigator.pop`: Digunakan untuk kembali ke halaman sebelumnya (halaman login) setelah logout.

4) Widget pada `LoginPage` (build method):

- Scaffold:
 - Widget dasar untuk halaman yang menyediakan struktur layout Material Design seperti `AppBar`, `Body`, dan `FloatingActionButton`.
 - `AppBar`: Sebuah toolbar di bagian atas layar yang diatur dengan tinggi 10. Ini mungkin digunakan hanya untuk memberi sedikit ruang di atas halaman.
 - `Padding`: Memberi jarak di sekitar konten sebanyak 30 piksel.
 - `SingleChildScrollView`: Membungkus seluruh isi halaman agar bisa di-scroll, terutama jika konten lebih tinggi dari layar.
 - `Column`: Widget yang menampilkan anak-anaknya secara vertikal (ke bawah). Di sini, digunakan untuk mengatur seluruh elemen login dalam satu kolom.
 - `SizedBox` untuk gambar logo: Menampilkan logo aplikasi yang diambil dari aset lokal (`assets/images/logo.png`) dengan tinggi 300 piksel, dan menyesuaikan lebar sesuai gambar.
 - `Align` untuk teks "Login": Mengatur teks "Login" agar rata kiri dengan ukuran font besar (40) dan tebal (bold).
 - `TextField` untuk username:
 - ❖ Field input yang dikontrol oleh `_usernameController`.

- ❖ `InputDecoration`: Menentukan tampilan input, termasuk label teks, batas luar (border), dan padding dalam konten.
- `TextField` untuk password: Sama seperti input username, tapi menggunakan properti `obscureText: true` untuk menyembunyikan karakter saat mengetik password.
- `TextButton` untuk "Forgot Password?": Tombol di kanan bawah input password yang berfungsi jika pengguna lupa password. Pada contoh ini, tombol belum memiliki aksi apapun, hanya mengubah warnanya menjadi biru.
- `ElevatedButton` untuk login:
 - ❖ Tombol untuk login yang akan memanggil fungsi `_login()` saat ditekan.
 - ❖ `style`: Mengatur tampilan tombol dengan warna merah.
- `Text` untuk pemisah "Or": Menampilkan teks "Or" sebagai pemisah antara tombol login dan tombol register.
- `TextButton` untuk register: Tombol yang memungkinkan pengguna untuk menuju ke halaman register. Saat ini, fungsinya belum diimplementasikan.

5) `DashboardPage` (`StatelessWidget`):

- `StatelessWidget`: Halaman dashboard ini bersifat statis karena hanya menampilkan informasi yang tidak berubah (misalnya, sapaan kepada pengguna dan menu).
- Parameter `username` dan `onLogout`:
 - `username`: Menerima username dari halaman login dan menampilkannya di halaman dashboard.
 - `onLogout`: Fungsi callback yang dipanggil saat pengguna logout, untuk kembali ke halaman login.

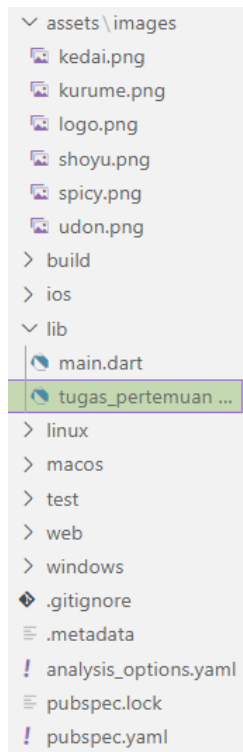
6) Widget pada `DashboardPage` (`build method`):

- `Scaffold`:
 - Struktur dasar halaman dashboard, sama seperti di halaman login.
 - `AppBar`: Toolbar di bagian atas halaman dengan tinggi lebih kecil, yaitu 15 piksel. Toolbar ini tidak memiliki tombol kembali karena logout akan ditangani dengan tombol di halaman.

- **Padding:** Memberi jarak sebesar 16 piksel di seluruh sisi halaman.
- **SingleChildScrollView:** Membungkus konten dashboard agar bisa di-scroll jika konten melebihi layar.
- **Column:** Menyusun elemen-elemen dalam satu kolom.
 - **Row** untuk informasi pengguna:
 - ❖ **CircleAvatar:** Menampilkan avatar berbentuk lingkaran dengan ikon orang di dalamnya (menggunakan `Icon(Icons.person)`).
 - ❖ **Text** untuk sapaan pengguna: Menampilkan teks sapaan "Halo, username" dengan font 18 dan "Itadakimasu" dengan font lebih besar dan tebal (23).
 - ❖ **ElevatedButton** untuk logout: Tombol berwarna merah yang menampilkan "Log out" dengan fungsi `onLogout` untuk kembali ke halaman login.
 - **Image.asset** untuk gambar kedai: Menampilkan gambar restoran `kedai.png` yang diambil dari aset lokal. Gambar ini diatur berada di tengah halaman menggunakan `Center`.
 - **Text** untuk "Our Menu": Teks "-- Our Menu --" diletakkan di tengah halaman sebagai penanda bagian menu.
 - **Row** untuk setiap menu:
 - ❖ **Image.asset:** Menampilkan gambar menu seperti Ramen Udon, Kurume Ramen, dan Shoyu Ramen dengan ukuran 130x130 piksel dan `BoxFit.cover` untuk menutupi area gambar.
 - ❖ **Column:** Di sebelah gambar, terdapat kolom yang menampilkan nama menu dan harganya dengan font ukuran 20 dan tebal.

7) Langkah untuk menambahkan gambar assets:

- a. Masukkan gambar ke dalam folder baru `assets/images` posisi folder sejajar dengan `pubspec.yaml`.



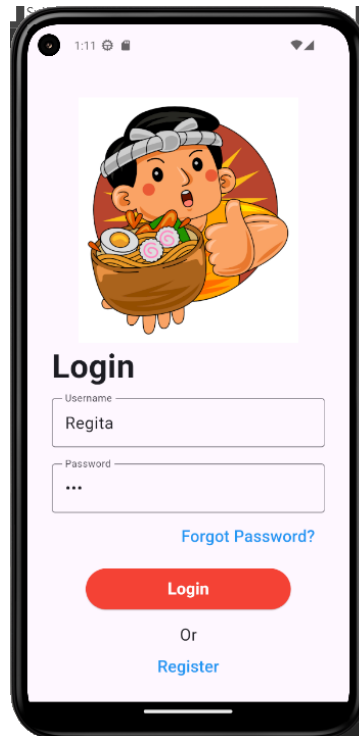
- b. Registrasikan gambar yang tadi dimasukkan dengan cara edit pubspec.yaml

```
assets:  
  - assets/images/
```

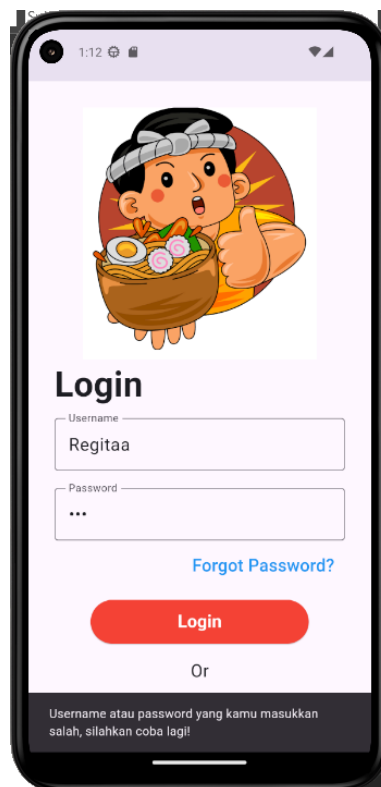
2.3. Hasil

Pada kode program di atas dapat menampilkan hasil yaitu dua halaman utama:

1) Halaman Login:

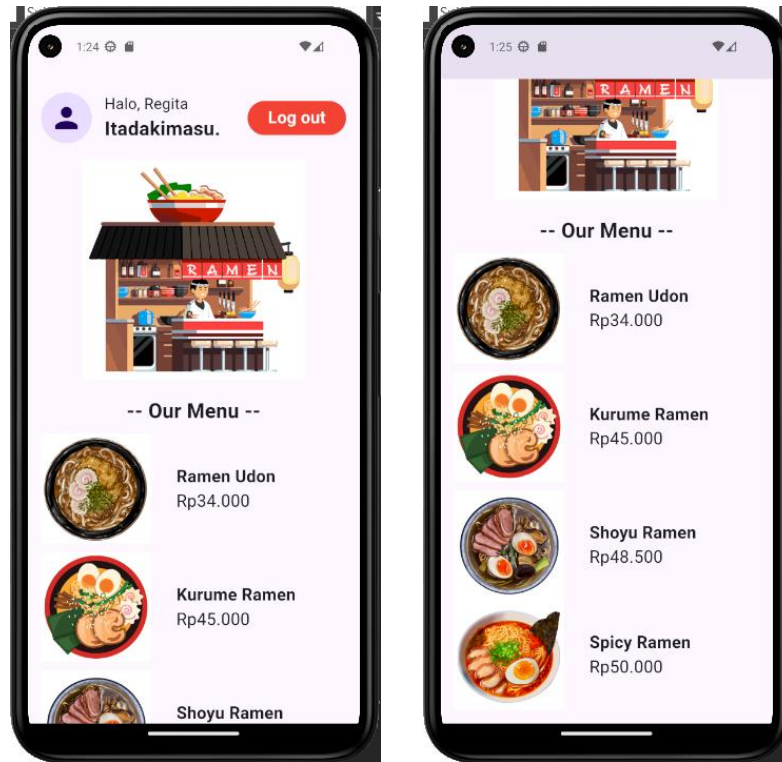


Halaman ini berisi form login untuk memasukkan username dan password. Jika berhasil, pengguna akan diarahkan ke halaman dashboard.



Jika pengguna memasukkan username dan password yang salah. Maka, pengguna akan SnackBar message akan ditampilkan, yang menunjukkan pesan kesalahan “Username atau password yang kamu masukkan salah, silahkan coba lagi!” kepada pengguna.

2) Halaman Dashboard:



Setelah pengguna berhasil melakukan login, mereka akan diarahkan ke halaman dashboard. Di halaman ini, terdapat sambutan hangat yang ditujukan kepada pengguna, serta daftar menu restoran yang disajikan dengan gambar dan harga dari berbagai pilihan makanan. Halaman ini dirancang untuk memberikan informasi visual yang menarik.

REFERENSI

- Blup. (2024). How to Display a SnackBar in Flutter? Diakses dari <https://medium.com/@blup-tool/how-to-display-a-snackbar-in-flutter-ee5e846d4eb>
- Display a snackbar. (n.d.). Diakses dari <https://docs.flutter.dev/cookbook/design/snackbars>
- Napoli, M. L. (2020). *Beginning Flutter A Hands On Guide To App Development*. Canada: John Wiley & Sons, Inc.
- SingleChildScrollView class. (n.d.). Diakses dari <https://api.flutter.dev/flutter/widgets/SingleChildScrollView-class.html>
- Tahir, M. (2022). ListView and SingleChildScrollView Widgets in Flutter. Diakses dari <https://medium.com/@mustafatahirhussein/listview-and-singlechildscrollview-widgets-in-flutter-b2d794ed93a8>