

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL VII
NAVIGASI DAN NOTIFIKASI**



Disusun Oleh :

SalmanAlfaRizzi

2211104056

SE-06-02

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

TUGAS PENDAHULUAN

A. SOAL NOMOR 1

- a) Buatlah satu project baru, yang mana di dalamnya terdapat *navigasi* untuk pindah dari satu halaman ke halaman lainnya.

Source Code:

```
Home.dart
import 'package:flutter/material.dart';
import 'profile.dart';
import 'login.dart';

class HomePage extends StatefulWidget {
  final String username;
  const HomePage({super.key, required this.username});

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int _selectedIndex = 0;

  void _onItemTapped(int index) {
    if (index == 2) {
      // Logout and go back to LoginPage
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => const LoginPage()),
      );
    } else {
      setState() {
        _selectedIndex = index;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    List<Widget> _pages = [
      Center(child: Text('Selamat datang, ${widget.username}', style: const
TextStyle(fontSize: 24))),
      ProfilePage(username: widget.username),
      Container(), // Placeholder for logout
    ];

    return Scaffold(
      appBar: AppBar(
        title: Text(_selectedIndex == 0 ? 'Home' : 'Profile'),
```

```

    ),
    body: _pages[_selectedIndex],
    bottomNavigationBar: BottomNavigationBar(
      items: const <BottomNavigationBarItem>[
        BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
        BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profile'),
        BottomNavigationBarItem(icon: Icon(Icons.logout), label: 'Logout'),
      ],
      currentIndex: _selectedIndex,
      selectedItemColor: Colors.blue,
      onTap: _onItemTapped,
    ),
  );
}
}

```

Login.dart

```

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

```

```

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

  @override
  _LoginPageState createState() => _LoginPageState();
}

```

```

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

  void _login() {
    if (_formKey.currentState!.validate()) {
      String username = _usernameController.text;

      Navigator.pushReplacement(
        context,
        MaterialPageRoute(
          builder: (context) => HomePage(username: username),
        ),
      );
    }
  }
}

```

```

@override
void dispose() {
  _usernameController.dispose();
  _passwordController.dispose();
  super.dispose();
}

```

```

@override
Widget build(BuildContext context) {

```

```

return Scaffold(
  appBar: AppBar(
    title: const Text('Login'),
  ),
  body: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Form(
      key: _formKey,
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          TextFormField(
            controller: _usernameController,
            decoration: InputDecoration(
              labelText: 'Username',
              prefixIcon: const Icon(Icons.person),
              border: OutlineInputBorder(),
            ),
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Username tidak boleh kosong';
              }
              return null;
            },
          ),
          const SizedBox(height: 16),
          TextFormField(
            controller: _passwordController,
            decoration: InputDecoration(
              labelText: 'Password',
              prefixIcon: const Icon(Icons.lock),
              border: OutlineInputBorder(),
            ),
            obscureText: true,
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Password tidak boleh kosong';
              }
              return null;
            },
          ),
          const SizedBox(height: 20),
          ElevatedButton(
            onPressed: _login,
            child: const Text('Login'),
          ),
        ],
      ),
    ),
  );
}

```

Profile.dart

import 'package:flutter/material.dart';

class ProfilePage extends StatelessWidget {
 final String username;
 const ProfilePage({super.key, required this.username});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

centerTitle: true,

),

body: SingleChildScrollView(

child: Padding(

padding: const EdgeInsets.all(3.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const SizedBox(height: 20),

Text(

username,

style: const TextStyle(

fontSize: 28,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 8),

const SizedBox(height: 20),

// Card for User Information

Card(

elevation: 5,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(15),

),

child: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

ListTile(

leading: const Icon(Icons.email, color: Colors.blue),

title: const Text('Email'),

subtitle: Text('\$username@gmail.com'), // contoh email

),

const Divider(),

ListTile(

leading: const Icon(Icons.phone, color: Colors.green),

title: const Text('Nomor Telepon'),

subtitle: const Text('+62 83866033060'), // contoh nomor telepon

),

const Divider(),

```

        ListTile(
          leading: const Icon(Icons.location_on, color: Colors.red),
          title: const Text('Lokasi'),
          subtitle: const Text('Banyumas, Indonesia'), // contoh lokasi
        ),
      ],
    ),
  ),
],
),
);
}
}

```

Main.dart

```

import 'package:flutter/material.dart';
import 'package:praktikum_07/TP/home.dart';
import 'package:praktikum_07/TP/login.dart';

```

```

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

```

```

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

```

// This widget is the root of your application.

@override

```

Widget build(BuildContext context) {

```

```
  return MaterialApp(
    title: 'Flutter Demo',

```

```
    theme: ThemeData(

```

// This is the theme of your application.

//

*// TRY THIS: Try running your application with "flutter run". You'll see
 // the application has a purple toolbar. Then, without quitting the app,
 // try changing the seedColor in the colorScheme below to Colors.green
 // and then invoke "hot reload" (save your changes or press the "hot
 // reload" button in a Flutter-supported IDE, or press "r" if you used
 // the command line to start the app).*

//

*// Notice that the counter didn't reset back to zero; the application
 // state is not lost during the reload. To reset the state, use hot
 // restart instead.*

//

*// This works for code too, not just values: Most code changes can be
 // tested with just a hot reload.*

```

    colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
    useMaterial3: true,

```

```
  ),

```

```
  home: LoginPage()

```

```
);
}
}
```

```
class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});
```

```
  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.
```

```
  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".
```

```
  final String title;
```

```
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}
```

```
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;
```

```
  void _incrementCounter() {
    setState() {
      // This call to setState tells the Flutter framework that something has
      // changed in this State, which causes it to rerun the build method below
      // so that the display can reflect the updated values. If we changed
      // _counter without calling setState(), then the build method would not be
      // called again, and so nothing would appear to happen.
      _counter++;
    });
  }
}
```

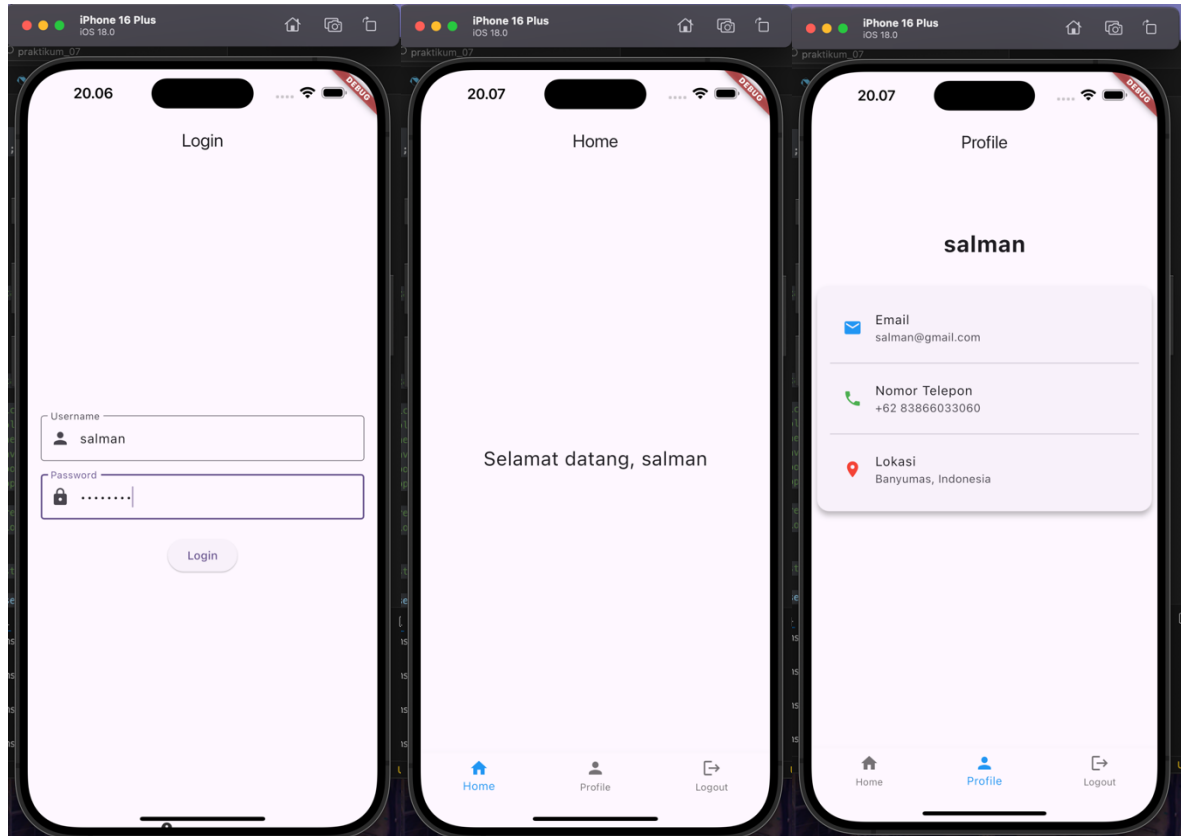
```
  @override
  Widget build(BuildContext context) {
    // This method is rerun every time setState is called, for instance as done
    // by the _incrementCounter method above.
    //
    // The Flutter framework has been optimized to make rerunning build methods
    // fast, so that you can just rebuild anything that needs updating rather
    // than having to individually change instances of widgets.
    return Scaffold(
      appBar: AppBar(
        // TRY THIS: Try changing the color here to a specific color (to
        // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar
        // change color while the other colors stay the same.
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        // Here we take the value from the MyHomePage object that was created by
        // the App.build method, and use it to set our appBar title.
        title: Text(widget.title),
      ),
```

```

body: Center(
  // Center is a layout widget. It takes a single child and positions it
  // in the middle of the parent.
  child: Column(
    // Column is also a layout widget. It takes a list of children and
    // arranges them vertically. By default, it sizes itself to fit its
    // children horizontally, and tries to be as tall as its parent.
    //
    // Column has various properties to control how it sizes itself and
    // how it positions its children. Here we use mainAxisAlignment to
    // center the children vertically; the main axis here is the vertical
    // axis because Columns are vertical (the cross axis would be
    // horizontal).
    //
    // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
    // action in the IDE, or press "p" in the console), to see the
    // wireframe for each widget.
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      const Text(
        'You have pushed the button this many times:',
      ),
      Text(
        '$_counter',
        style: Theme.of(context).textTheme.headlineMedium,
      ),
    ],
  ),
),
floatingActionButton: FloatingActionButton(
  onPressed: _incrementCounter,
  tooltip: 'Increment',
  child: const Icon(Icons.add),
), // This trailing comma makes auto-formatting nicer for build methods.
);
}
}

```

Screenshot Output



Deskripsi Program

Pengguna masuk dengan memasukkan nama dan kata sandi, lalu diarahkan ke halaman utama jika berhasil. Di halaman utama, pengguna bisa melihat profil atau logout lewat menu navigasi bawah. Halaman profil menampilkan informasi pengguna seperti email, nomor telepon, dan lokasi. Saat memilih logout, pengguna akan kembali ke halaman login.