

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT  
BERGERAK**

**MODUL 6 : INTERAKSI PENGGUNA**



**Disusun Oleh :**

**Ade Fatkhul Anam / 22111040451**

**SE-06-02**

**Asisten Praktikum :**

**Muhammad Faza Zulian Gesit Al Barru**

**Aisyah Hasna Aulia**

**Dosen Pengampu :**

**Yudha Islami Sulistya**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## MODUL 5

### ANTARMUKA PENGGUNA LANJUTAN

#### A. Tujuan Praktikum

1. Mahasiswa mampu memahami konsep layout pada Flutter
2. Mahasiswa dapat mengimplementasikan desain user interface pada Flutter

#### B. Package

Dart package terletak dalam direktori dengan file *pubspec.yaml*. Contohnya, dapat digunakan untuk melakukan *request* ke server dengan protokol HTTP atau untuk *custom navigation* menggunakan *Fluro*.

Source code:

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

#### C. User Interaction

##### - Stateful & Stateless

Widget stateless bersifat tetap dan tidak berubah, contohnya adalah *Icon*, *IconButton*, dan *Text*, dengan subkelas *StatelessWidget*. Widget stateful bersifat dinamis, dapat berubah sesuai interaksi pengguna atau data yang diterima, seperti *Checkbox*, *Radio*, *Slider*, *InkWell*, *Form*, dan *TextField*, dengan subkelas *StatefulWidget*.

Source code:

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

  @override
  State<MyPackage> createState() => _MyPackageState();
}

class _MyPackageState extends State<MyPackage> {
  String? selectedValue;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Belajar Package",
          style: GoogleFonts.sriracha(),
        ),
        backgroundColor: Colors.yellow,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            // Form (TextField & DropdownButton)
          ],
        ),
      ),
    );
  }
}
```

- **Form**

*Form* adalah kumpulan widget untuk mengumpulkan input pengguna, biasanya terdiri dari beberapa *field* seperti *TextFormField*, *Checkbox*, dan *Radio*. *Form widget* digunakan untuk mengelompokkan dan mengelola status semua *field* dalam *form*.

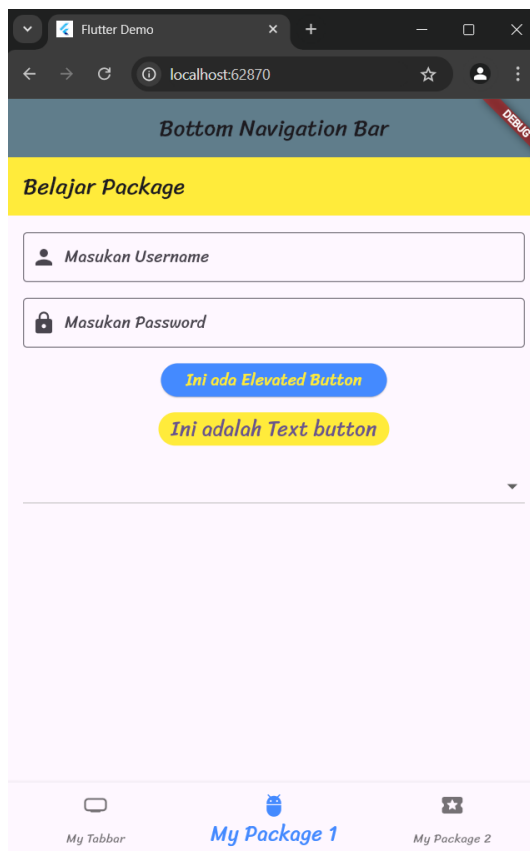
Source code:

```

TextField(
    obscureText: false,
    obscuringCharacter: "*",
    decoration: InputDecoration(
        prefixIcon: Icon(Icons.person),
        hintText: "Masukan Username",
        border: OutlineInputBorder(),
    ),
),
 SizedBox(height: 15),
TextField(
    obscureText: true,
    obscuringCharacter: "*",
    decoration: InputDecoration(
        prefixIcon: Icon(Icons.lock),
        hintText: "Masukan Password",
        border: OutlineInputBorder(),
    ),
),
 SizedBox(height: 15),
ElevatedButton(
    onPressed: () {},
    child: Text(
        "Ini ada Elevated Button",
        style: TextStyle(
            color: Colors.yellow,
            fontSize: 16,
        ),
    ),
),
 style: ElevatedButton.styleFrom(
     backgroundColor: Colors.blueAccent,
 ),
),
SizedBox(height: 15),
TextButton(
    style: TextButton.styleFrom(
        backgroundColor: Colors.yellow,
    ),
    onPressed: () {},
    child: Text(
        "Ini adalah Text button",
        style: TextStyle(fontSize: 20),
    ),
),
SizedBox(height: 15),
DropdownButton<String>(
    dropdownColor: Colors.yellowAccent,
    focusColor: Colors.greenAccent,
    isExpanded: true,
    value: selectedValue,
    items: <String>['Ops1 1', 'Ops1 2', 'Ops1 3']
        .map<DropDownMenuItem<String>>((String value) {
            return DropDownMenuItem<String>(
                value: value,
                child: Text(value),
            );
        }).toList(),
    onChanged: (String? newValue) {
        setState(() {
            selectedValue = newValue;
        });
    },
),

```

Output:



#### D. Menu

Salah satu elemen penting dalam pembuatan aplikasi adalah menu, yang berfungsi untuk memisahkan fitur atau halaman. Menampilkan semua fitur di satu halaman akan menyulitkan pengguna dalam navigasi. Oleh karena itu, menu halaman sangat membantu. Ada dua jenis widget menu yang umum digunakan: *bottom navigation bar* dan *tab bar*, yang didukung penuh oleh *Material Design* dari Google. Tab Bar Berikut adalah langkah-langkah membuat Tab Bar:

1. Membuat TabController
2. Membuat tab
3. Menyiapkan konten untuk setiap tab

Source code:

```
// membuat Tab Controller
return DefaultTabController(
  length: 3, // Terdapat 3 tab
  child: Scaffold(
    appBar: AppBar(
      title: Text("Menu Tab Bar"),
      bottom: TabBar(tabs: [ // TabBar berfungsi sebagai kontrol tab
        Tab(
          icon: Icon(Icons.food_bank),
          text: "Menu",
        ),
        Tab(
          icon: Icon(Icons.calendar_month),
          text: "Calender",
        ),
        Tab(
          icon: Icon(Icons.history),
          text: "History",
        )
      ]
    )
  )
)
```

```

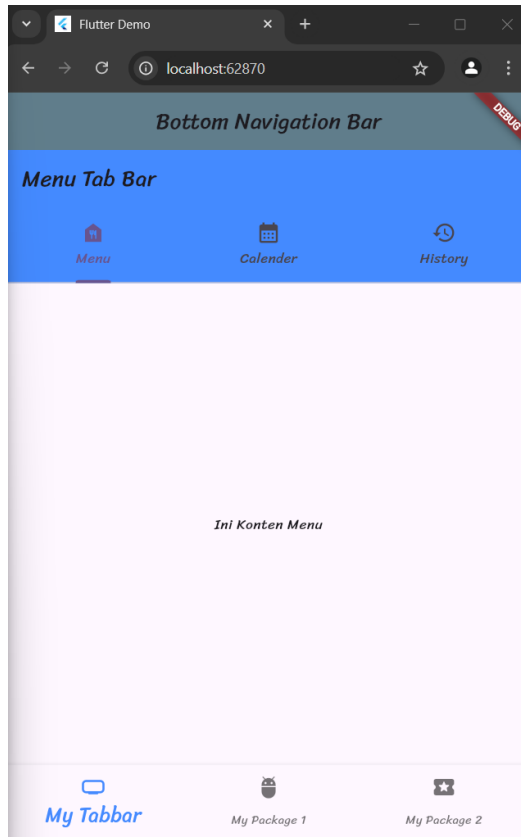
    ]),
    backgroundColor: Colors.blueAccent,
  ),
// Membuat Konten untuk Masing-Masing Tab
body: TabBarView(children: [
  Center(
    child: Text("Ini Konten Menu"),
  ),
  Center(
    child: Text("Ini Konten Calender"),
  ),
  Center(
    child: Text("Ini Konten History"),
  ),
]),
),

// Membuat Bottom Navigation Bar

bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.business),
      label: 'Business',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.school),
      label: 'School',
    ),
  ],
),

```

Output:



## E. Buttons

- **ElevatedButton**

ElevatedButton adalah tombol yang umum digunakan untuk tindakan penting seperti pendaftaran, pengiriman, atau masuk. Widget ini dirancang untuk menonjol dengan efek bayangan yang memberikan kesan kedalaman, ideal untuk menarik perhatian pengguna pada tindakan utama dalam aplikasi.

- **TextButton**

TextButton adalah widget untuk membuat tombol sederhana yang hanya menampilkan teks, cocok untuk tindakan yang tidak perlu terlalu menonjol. Tombol ini sering digunakan untuk opsi tambahan yang lebih kecil di dalam aplikasi.

- **DropDownButton**

DropDownButton adalah widget yang memungkinkan pengguna memilih satu nilai dari daftar opsi dalam menu dropdown. Ini sangat berguna ketika pilihan pengguna perlu dibatasi pada beberapa opsi yang telah ditentukan.

Source Code:

```

// ElevatedButton
ElevatedButton(
  onPressed: () {
    // Aksi yang akan dijalankan saat tombol ditekan
  },
  child: Text(
    "Elevated Button",
    style: TextStyle(
      color: Colors.white,
      fontSize: 16,
    ),
  ),
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.blue,
    padding: EdgeInsets.symmetric(horizontal: 24, vertical: 12),
  ),
),

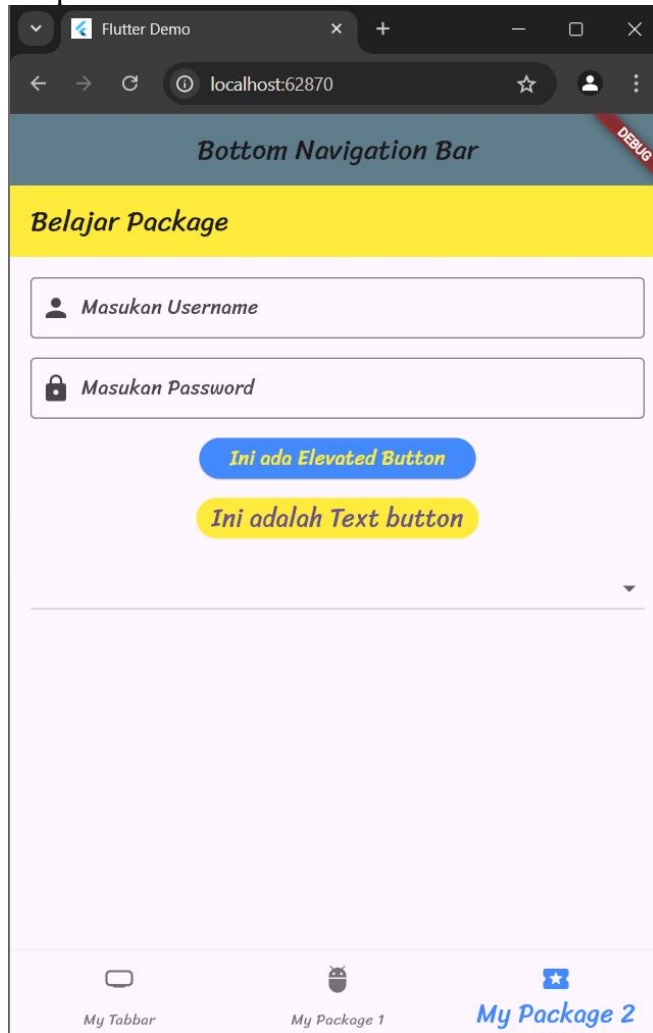
// TextButton
TextButton(
  onPressed: () {
    // Aksi yang akan dijalankan saat tombol ditekan
  },
  child: Text(
    "Text Button",
    style: TextStyle(
      fontSize: 16,
      color: Colors.black,
    ),
  ),
  style: TextButton.styleFrom(
    backgroundColor: Colors.yellow,
    padding: EdgeInsets.symmetric(horizontal: 24, vertical: 12),
  ),
),

// DropdownButton
String? selectedValue;

DropdownButton<String>(
  value: selectedValue,
  hint: Text("Select an option"),
  items: <String>['Option 1', 'Option 2', 'Option 3']
    .map<DropdownMenuItem<String>>((String value) {
      return DropdownMenuItem<String>(
        value: value,
        child: Text(value),
      );
    }).toList(),
  onChanged: (String? newValue) {
    setState(() {
      selectedValue = newValue;
    });
  },
  dropdownColor: Colors.white,
  iconEnabledColor: Colors.blue,
),

```

Output:





## F. Unguided

1. Modifikasi project TP 06 (Tugas Pendahuluan) Interaksi Pengguna, dengan mengimplementasikan penggunaan form dan button.

Source code:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Bottom Navigation Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MainPage(),
    );
  }
}

class MainPage extends StatefulWidget {
  @override
  _MainPageState createState() => _MainPageState();
}

class _MainPageState extends State<MainPage> {
  int _selectedIndex = 0;

  final TextEditingController _nameController = TextEditingController();
  final TextEditingController _emailController = TextEditingController();

  String _submittedName = "";
  String _submittedEmail = "";

  static const List<Widget> _pages = <Widget>[
    Center(child: Text('Ini adalah halaman Home', style: TextStyle(fontSize: 24))),
    Center(child: Text('Ini adalah halaman Wisata', style: TextStyle(fontSize: 24))),
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

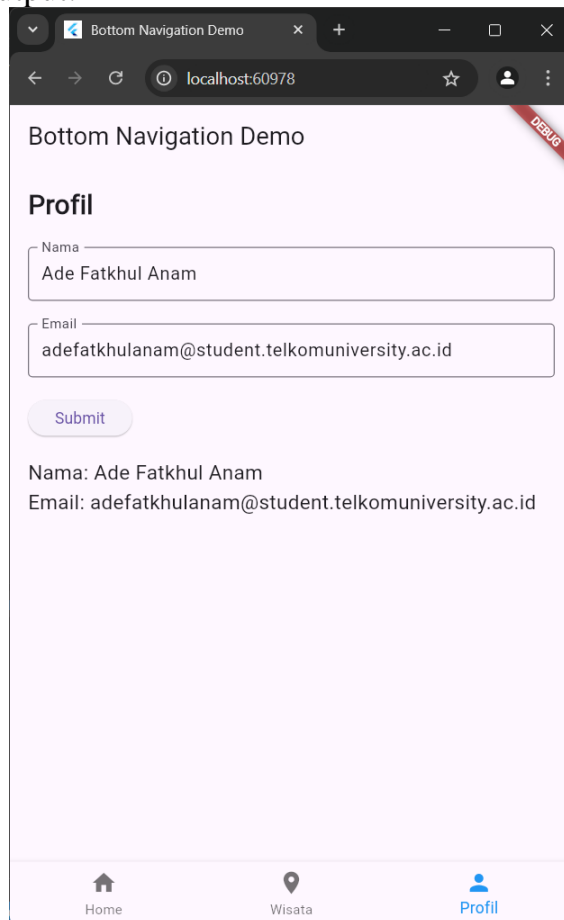
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Bottom Navigation Demo'),
      ),
      body: _selectedIndex == 2 ? _buildProfilePage() : _pages[_selectedIndex],
      bottomNavigationBar: BottomNavigationBar(
        items: const <BottomNavigationBarItem>[
          BottomNavigationBarItem(
            icon: Icon(Icons.home),
            label: 'Home',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.place),
            label: 'Wisata',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.person),
            label: 'Profil',
          ),
        ],
        currentIndex: _selectedIndex,
        selectedItemColor: Colors.blue,
        onTap: _onItemTapped,
      ),
    );
  }
}
```

```

Widget _buildProfilePage() {
  return Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text('Profil', style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
        SizedBox(height: 20),
        TextField(
          controller: _nameController,
          decoration: InputDecoration(
            labelText: 'Nama',
            border: OutlineInputBorder(),
          ),
        ),
        SizedBox(height: 20),
        TextField(
          controller: _emailController,
          decoration: InputDecoration(
            labelText: 'Email',
            border: OutlineInputBorder(),
          ),
        ),
        SizedBox(height: 20),
        ElevatedButton(
          onPressed: () {
            setState(() {
              _submittedName = _nameController.text;
              _submittedEmail = _emailController.text;
            });
          },
          child: Text('Submit'),
        ),
        SizedBox(height: 20),
        if (_submittedName.isNotEmpty || _submittedEmail.isNotEmpty)
          Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text('Nama: $_submittedName', style: TextStyle(fontSize: 18)),
              Text('Email: $_submittedEmail', style: TextStyle(fontSize: 18)),
            ],
          ),
      ],
    ),
  );
}

```

Output:



## G. Kesimpulan

Pada praktikum kali ini, kita telah berhasil memodifikasi sebuah aplikasi Flutter dengan mengimplementasikan penggunaan form dan tombol dalam konteks antarmuka pengguna. Praktikum ini memberikan pemahaman tentang cara menambahkan elemen interaktif ke dalam aplikasi, seperti `TextField` untuk input pengguna, `ElevatedButton` untuk aksi submit, dan cara mengelola state untuk menampilkan data yang dimasukkan. Selain itu, kita juga mempelajari bagaimana mengelola navigasi antar halaman menggunakan *Bottom Navigation Bar*, sehingga pengguna dapat beralih antara berbagai tampilan dengan mudah. Dengan adanya penambahan fitur ini, aplikasi menjadi lebih fungsional dan responsif terhadap interaksi pengguna. Praktikum ini memberikan wawasan yang lebih mendalam tentang pengembangan aplikasi mobile menggunakan Flutter dan bagaimana menciptakan pengalaman pengguna yang lebih baik melalui desain antarmuka yang intuitif.