**LAPORAN PRAKTIKUM PEMROGRAMAN BERGERAK (MOBILE)**

Sebuah gambar berisi logo, teks, simbol, Font

Deskripsi dibuat secara otomatis**DISARANKAN MEMBUKA MENGGUNAKAN LAPTOP AGAR BISA KLIK HYPERLINK!!**

G

**Dikerjakan Oleh:**

Evi Fitriya (1201222005)

**Fakultas Informatika (FIF)**

**Rekayasa Perangkat Lunak (S1)**

**SURABAYA**

**2024**

# TUGAS 5 LMS

## Pembahasan:

SOAL:

Sebuah gambar berisi teks, cuplikan layar, software, Laman internet

Deskripsi dibuat secara otomatis

Komponen lainnya yang digunakan seperti yang sudah dijelaskan pada pertemuan – pertemuan sebelumnya, yang berbeda adalah sebagai berikut:

* **Penggunaan if else pada flutter dalam dart**

Hal yang baru disini adalah ada 2 kondisi dimana dihalaman home tulisannya berada dikiri atas dan pada halaman lain tulisannya berada pada tengah halaman, itu menggunakan if else yang dimana “?” sebagai if dan “:” sebagai else

Contoh :

child: \_selectIndex == 0

                ? Column(children: [

                    pags.elementAt(\_selectIndex),

                  ])

                : Center(

                    child: pags.elementAt(\_selectIndex),

                  )),

Apabila if terpenuhi maka kode akan seperti berikut:

child: Column(

children: [

pags.elementAt(\_selectIndex),

             ])

Sebaliknya apabila else terpenuhi kode akan seperti berikut:

child: Center(

child:

pags.elementAt(\_selectIndex),

             )

* **Penggunaan saveArea**

Save Area digunakan agar content didalam body berada dibawahnya top notification disemua device

Kodenya seperti berikut:

body: SafeArea(

....

),

## Tampilan:

Sebuah gambar berisi teks, cuplikan layar, software, Sistem operasi

Deskripsi dibuat secara otomatisSebuah gambar berisi teks, cuplikan layar, software, Sistem operasi

Deskripsi dibuat secara otomatisSebuah gambar berisi teks, cuplikan layar, software, Sistem operasi

Deskripsi dibuat secara otomatis  
*Tampilan Tab Home Tampilan Tab Email Tampilan Tab Profile*

## Code:

**Tinggal diklik saja untuk menuju ke kode (Semua Kode ada di bagian “Lampiran”)**

* [Main.dart](#_Main.dart)
* [Tugas5lms.dart](#_tugas5lms.dart)

# LAMPIRAN

### Main.dart

import 'package:flutter/material.dart';

// import 'package:praktikum\_1/layout.dart';

import 'package:praktikum\_1/meet1/latihan1.dart';

import 'package:praktikum\_1/meet2/latihan1.dart';

import 'package:praktikum\_1/meet2/tabbar.dart';

import 'package:praktikum\_1/meet2/pageView.dart';

import 'package:praktikum\_1/meet3n4/button.dart';

import 'package:praktikum\_1/meet4/bot\_navbar.dart';

import 'package:praktikum\_1/tugas/tugas4lms.dart';

import 'package:praktikum\_1/tugas/tugas5lms.dart';

import 'package:praktikum\_1/meet5/arguments\_screen.dart';

import 'package:praktikum\_1/tugas/tugas6lms.dart';

import 'package:praktikum\_1/meet6/media\_camera.dart';

import 'package:praktikum\_1/meet5/arguments\_screen.dart';

import 'package:praktikum\_1/meet5/first\_screen.dart';

import 'package:praktikum\_1/meet5/second\_screen.dart';

import 'package:praktikum\_1/meet5/return\_data\_screen.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: 'Praktikum',

      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,

      ),

      initialRoute: '/tugas4',

      routes: {

        '/botnav': (context) => const Botnavbar(),

        '/return-data': (context) => const ReturnDataScreen(),

        '/button': (context) => const ButtonScreenState(),

        '/tugas6': (context) => const tugas\_6\_lms(),

        '/tugas4': (context) => const Evi\_1201222005s(),

        '/tugas5': (context) => const AdvanceLayout(),

        '/M1/l1': (context) => const latihan\_2(),

        '/M1/pageView': (context) => const Page\_View(),

        '/M1/tabbar': (context) => const tabbar(),

        '/media-camera': (context) => const mediaCamera(),

        '/': (context) => const FirstScreen(),

        '/second': (context) => const SecondScreen(),

        '/arguments': (context) =>

            const ArgumentsScreen(message: "Hi Everyone"),

      },

    );

  }

}

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(

              'tes:',

            ),

            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.

    );

  }

}

### tugas5lms.dart

import 'package:flutter/material.dart';

class AdvanceLayout extends StatefulWidget {

  const AdvanceLayout({super.key});

  @override

  State<AdvanceLayout> createState() => \_AdvanceLayout();

}

class \_AdvanceLayout extends State<AdvanceLayout> {

  int \_selectIndex = 0;

  static const List<Widget> pags = <Widget>[

    Text(

      'Home Page',

    ),

    Text(

      'Email Page',

    ),

    Text(

      'Profile Page',

    ),

  ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: SafeArea(

        child: DefaultTextStyle(

            style: TextStyle(

              color: Colors.black,

              fontSize: 40,

            ),

            child: \_selectIndex == 0

                ? Column(children: [

                    pags.elementAt(\_selectIndex),

                  ])

                : Center(

                    child: pags.elementAt(\_selectIndex),

                  )),

      ),

      bottomNavigationBar: BottomNavigationBar(

        items: const [

          BottomNavigationBarItem(

            label: 'Home',

            icon: Icon(Icons.home),

          ),

          BottomNavigationBarItem(

            label: 'Email',

            icon: Icon(Icons.email),

          ),

          BottomNavigationBarItem(

            label: 'Profile',

            icon: Icon(Icons.person),

          ),

        ],

        backgroundColor: Colors.orange,

        unselectedItemColor: Colors.green,

        selectedItemColor: Colors.blue,

        currentIndex: \_selectIndex,

        onTap: (indexSelected) {

          setState(() {

            \_selectIndex = indexSelected;

          });

        },

      ),

    );

  }

}