MINGGU KE 5-9

ViGeNesia – VISI GENERASI INDONESIA

BERABASIS WEB SERVICE

Deskripsi

Untuk memberikan pengalaman belajar mahasiswa implementasi web service, materi minggu ke 5-7 merupakan pembelajaran yang berkelanjutan dan didukung dengan case study pengembangan aplikasi ViGeNesia, serta dikerjakan secara mandiri oleh mahasiswa. ViGeNesia didukung struktur data yang dikembangkan menggunakan software database management system (**mariadb**). Untuk layanan web servis menggunakan restfull API (json). Front end dikembangkan menggunakan bahasa pemrogramman dart, framework flutter dan proses coding menggunakan editor *visual studi code*. Luaran dari pembelajaran adalah aplikasi ViGeNesia dapat dikembangkan secara berkelompok dan dipresentasikan pada pertemuan 10-15

Tujuan Pembelajaran

Mahasiswa memiliki pemahaman dan kemampuan dalam membangun sistem informasi berbasis web service. Kemampuan dalam mendisain struktur data, web service, serta pembuatan front end berbasis mobile. Mahasiswa secara berkelompok dapat mengembangkan aplikasi ViGeNesia yang siap dimanfaatkan masyarakat.

Seputar Aplikasi ViGeNesia

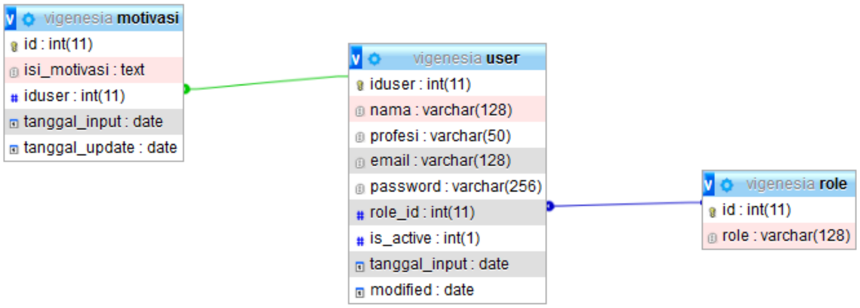
Generasi muda Indonesia menjadi tumpuan masa depan bangsa Indonesia dalam mempertahankan persatuan dan kesatuan melalui 3S (saling kenal, saling sapa, saling berbagi), memberikan motivasi dan menginspirasi, mengedukasi serta berbagi cerita dan cita-cita membangun negeri Indonesia. Kemajuan teknologi informasi dan komunikasi sangat memungkinkan **ViGeNesia** menjadi media secara virtual untuk menyatukan generasi muda melalui satu gengaman (*smartphone*). **ViGeNesia** akan dikembangkan secara bertahap mulai menjadi *case study* di mata kuliah Teknologi Web Service, dikembangkan oleh mahasiswa sebagai projek (*projct base*), dan dalam jangka panjang akan menjadi icon nasional melalui google playstore dan mendapatkan support dari masyarakat umum.

Kebutuhan awal User **ViGeNesia**

USER	KETERANGAN
1. All User	✓ Registrasi ✓ Login ✓ Logout
2. All User	**Create/Post ->** tabel motivasi Untuk menambahkan/create/insert data baru ke tabel motivasi. Catatan: Dibutuhkan **iduser** dari tabel user berdasarkan **emai user**, data **iduser** tersebut akan tersimpan di tabel motivasi.
	Read/Get -> tabel motivasi Untuk membaca/melihat/view data dari tabel motivasi. Pada aplikasi ViGeNesia **Read** dibagi menjadi dua pilihan 1. View data tabel motivasi keseluruhan 2. View data tabel motivasi berdasarkan **iduser** Catatan: Setelah user login, masuk ke fitur entry motivasi, maka record motivasi yang ditampilkan berdasarkan **iduser**, dilengkapi dengan fitur U**pdate/Put** dan **Delete**

Struktur database **ViGeNesia**

1. Disain logic (Normalisasi)



2. Disain fisik
   - Software DBMS : mariadb
3. Kebutuhan Software
   - Bahasa pemrograman: Dart
   - Framework : Flutter Versi 2.0.2
   - Editor : Visual Studio Code

1. **Create Project**

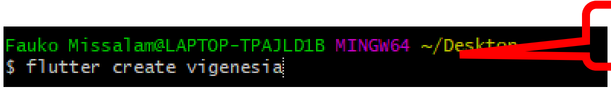
   Catatan Dalam Penamaan Projek. Dalam penamaan projek Flutter ada beberapa hal yang perlu diketahui, jika tidak maka akan terjadi error.

   - Penamaa projek hanya satu kata.
   - Jika lebih dari satu kata gunakan pemisah underscore (_), contoh : projek_flutter.
   - Jangan gunakan spasi atau dash (-) untuk pemisah kata di nama projek Flutter.

   Dalam studi kasus SIBERI persiapan projek seperti pada tabel

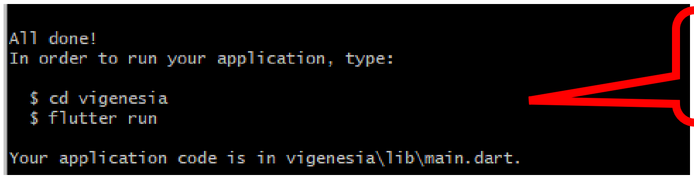
Path/Tempat penyimpanan projek	C:\Projek_TWS
Nama Projek	vigenesia

2. Langgkah pembuatan projek **Vigensia**



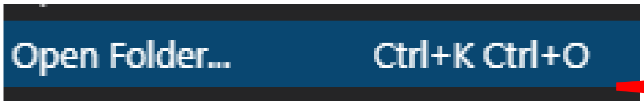
```
Fauko Missalam@LAPTOP-TPAJLD1B MINGW64 ~/Desktop
$ flutter create vigenesia
```

1. Melaui GitBash



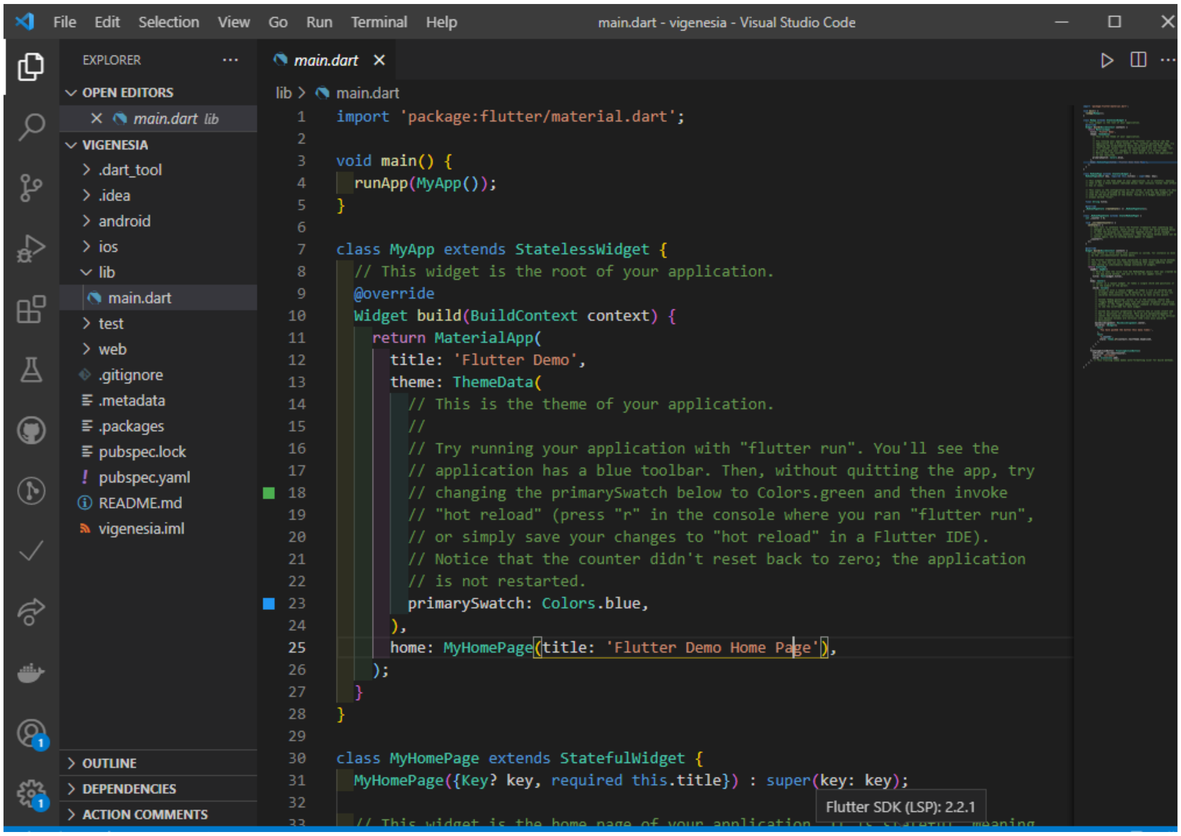
```
All done!
In order to run your application, type:

  $ cd vigenesia
  $ flutter run

Your application code is in vigenesia\lib\main.dart.
```

All done membuat project flutter

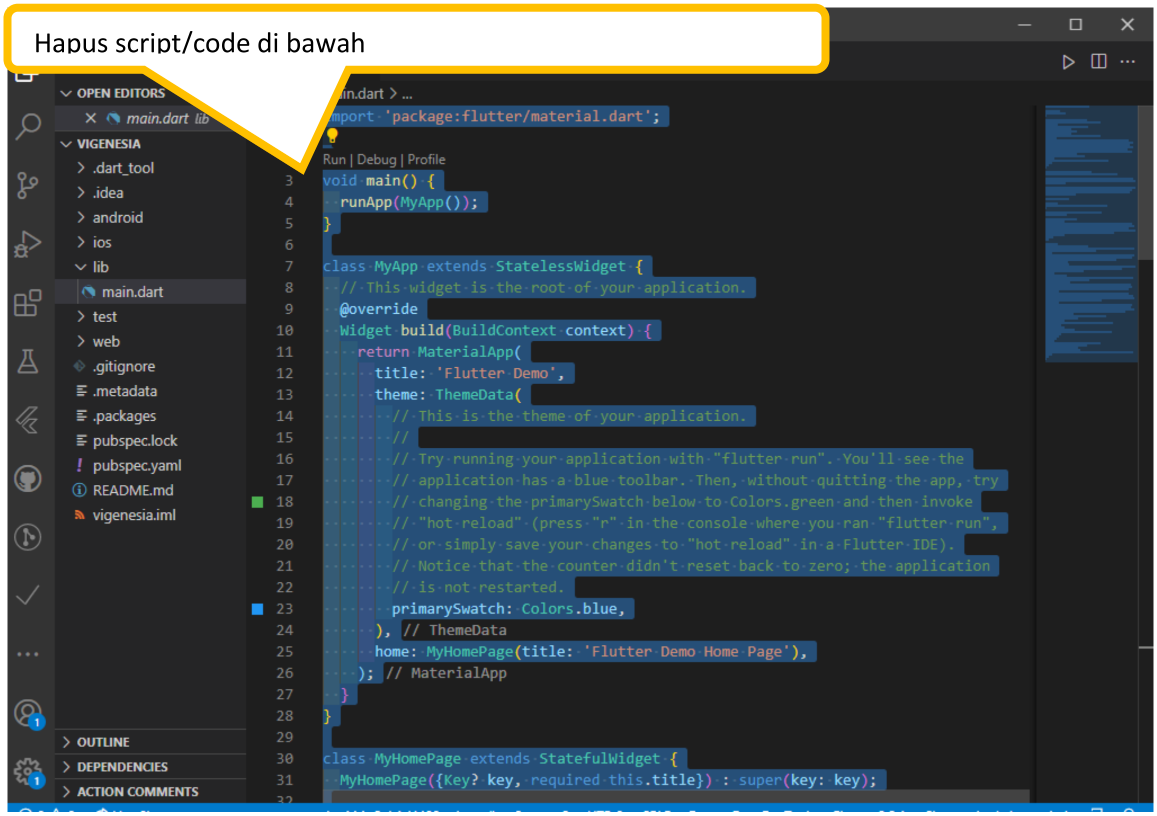


2. Open Folder Vigenesia yang sudah di buat

Hasil awal pembuatan projek **Vigenesia**

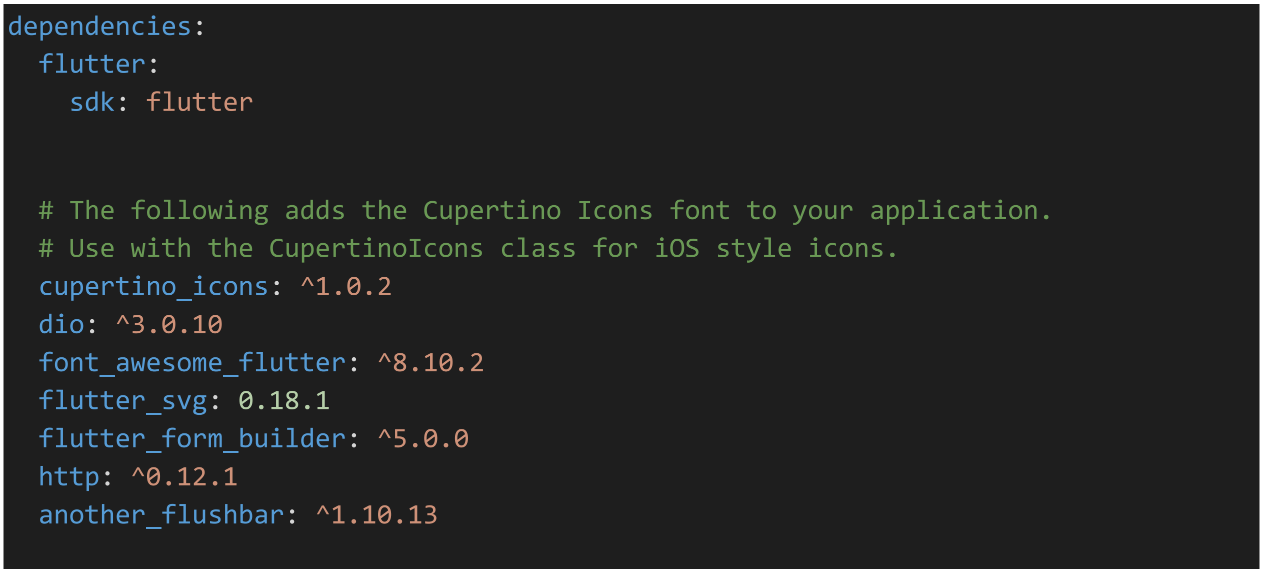


```dart
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  // 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 running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ),
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  MyHomePage({Key? key, required this.title}) : super(key: key);

  // This widget is the home page of your application. It is stateful, meaning
```


Hapus script/code di bawah

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

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

class MyApp extends StatelessWidget {
  // 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 running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ), // ThemeData
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    ); // MaterialApp
  }
}

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

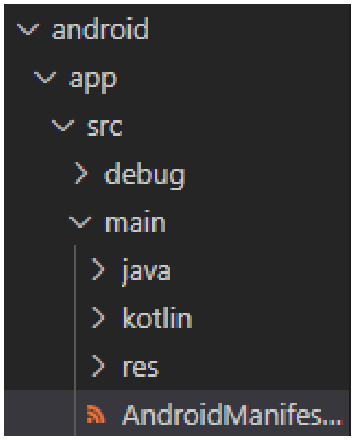
3. Setelah itu buka file **pubspec.yaml** lalu ketikan di bagian ini :



```yaml
dependencies:
  flutter:
    sdk: flutter


  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  dio: ^3.0.10
  font_awesome_flutter: ^8.10.2
  flutter_svg: 0.18.1
  flutter_form_builder: ^5.0.0
  http: ^0.12.1
  another_flushbar: ^1.10.13
```

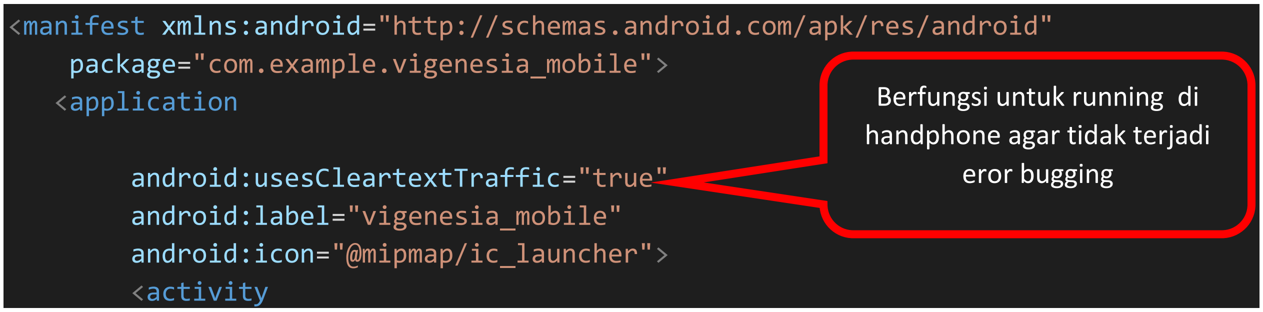
Code tersebut akan berpengaruh dengan package flutter , bisa di buka pada laman pub.dev

4. Setelah kita memasukan package flutter di file **pubspec.yaml** kita akan memodifikasi pada folder **Android** di dalam folder **main** terdapat file Bernama **AndroidManifest.xml** di modifikasi menjadi .



```
⌄ android
  ⌄ app
    ⌄ src
      > debug
      ⌄ main
        > java
        > kotlin
        > res
        AndroidManifes...
```

Code **AndroidManifest.xml :**



```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.vigenesia_mobile">
   <application

        android:usesCleartextTraffic="true"
        android:label="vigenesia_mobile"
        android:icon="@mipmap/ic_launcher">
        <activity
```

Berfungsi untuk running di handphone agar tidak terjadi eror bugging

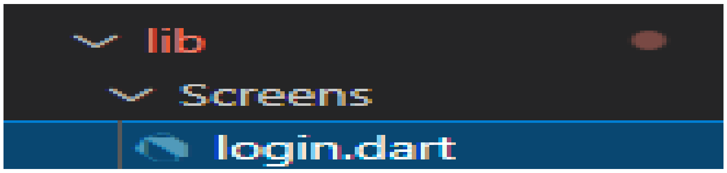
```
            android:name=".MainActivity"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize
|smallestScreenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">

            <meta-data
              android:name="io.flutter.embedding.android.NormalTheme"
              android:resource="@style/NormalTheme"
              />

            <meta-data
              android:name="io.flutter.embedding.android.SplashScreenDrawable"
              android:resource="@drawable/launch_background"
              />
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>
</manifest>
```

5. Coding awal **Main.dart** :

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

import 'Screens/Login.dart';

void main() => runApp(MaterialApp(
      debugShowCheckedModeBanner: false,
      home:
          Login(), // <--
 Buat Class Baru yg bernama MyScreen di dalam lib bikin folder baru screens isiny
a MyScreen.dart
    ));
```

Eror di karenakan kita harus buat folder screens

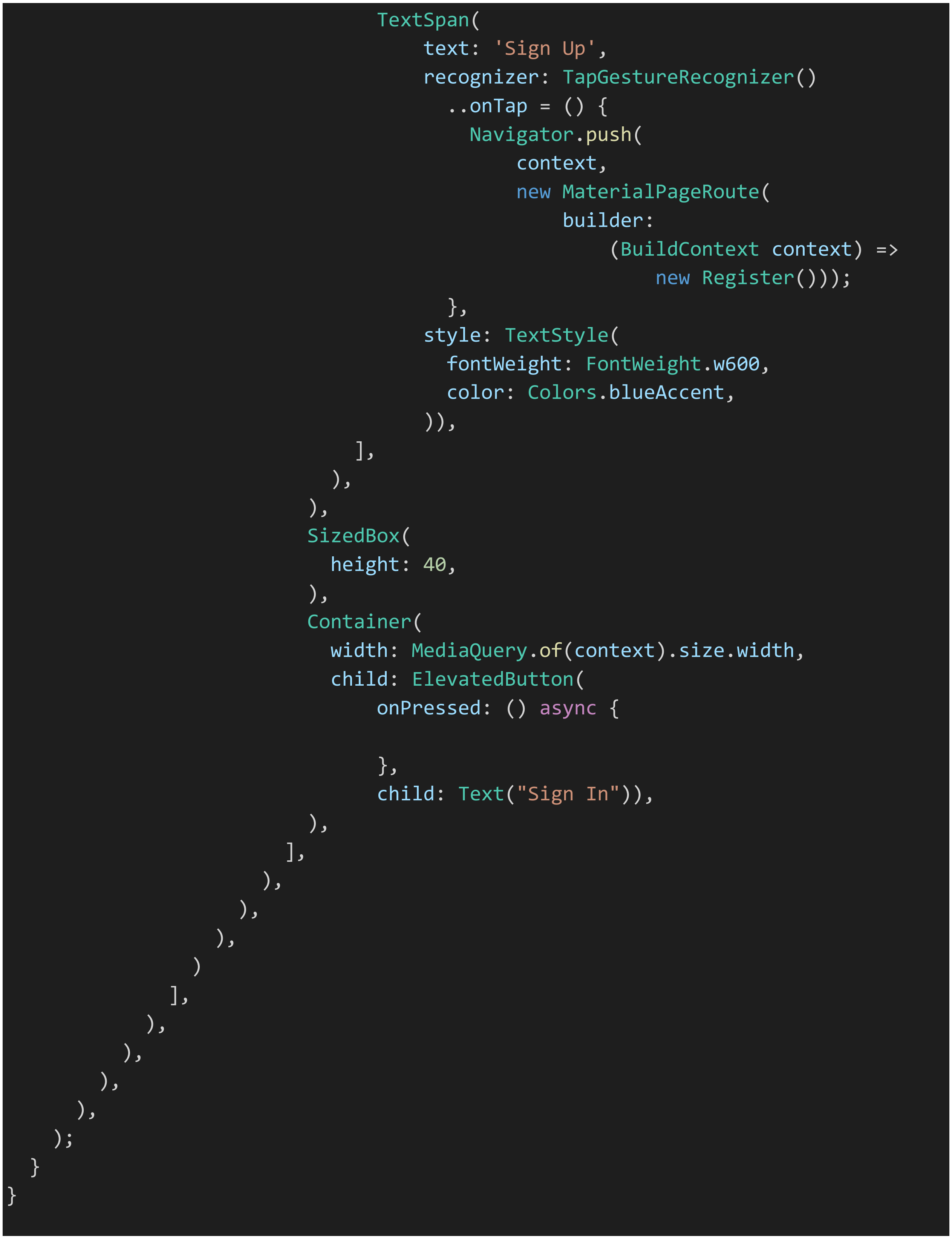
6. Buatlah Folder baru di dalam Lib dengan nama **Screens** dan buat file Bernama **login.dart**:



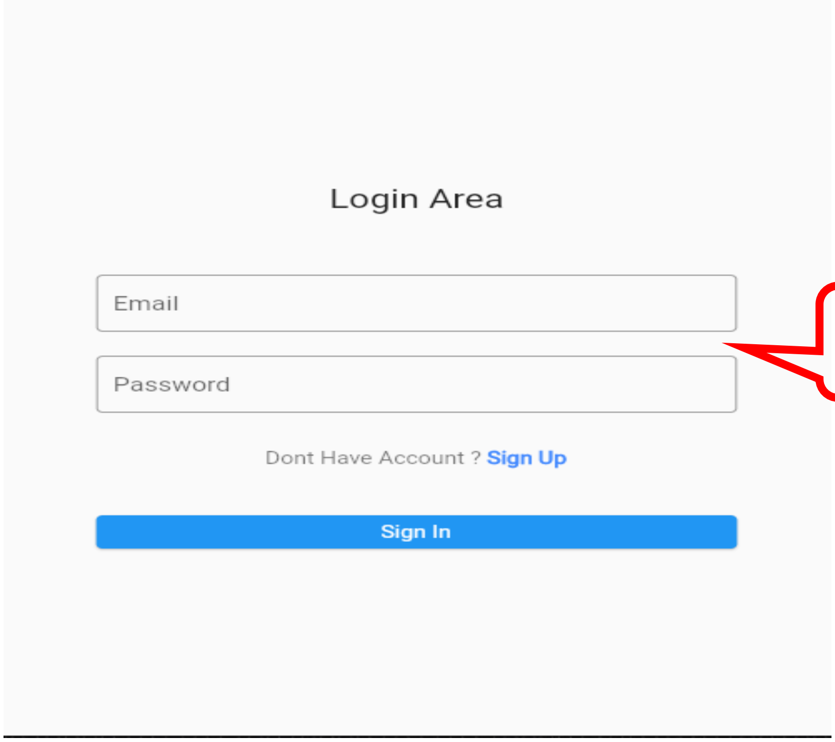
```
lib
  Screens
    login.dart
```

Login.dart :

```
import 'package:another_flushbar/flushbar.dart';
import 'package:flutter/material.dart';
import 'package:flutter_form_builder/flutter_form_builder.dart';
import 'Register.dart';
import 'package:flutter/gestures.dart';
import 'dart:convert';

class Login extends StatefulWidget {
  const Login({Key key}) : super(key: key);

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

class _LoginState extends State<Login> {
  String nama;

  final GlobalKey<FormBuilderState> _fbKey = GlobalKey<FormBuilderState>();

  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        // <-- Berfungsi Untuk  Bisa Scroll
        child: SafeArea(
          // < -- Biar Gak Keluar Area Screen HP
          child: Container(
            height: MediaQuery.of(context).size.height,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Text(
                  "Login Area",
                  style: TextStyle(fontSize: 22, fontWeight: FontWeight.w500),
                ),
                SizedBox(height: 50), // <-- Kasih Jarak Tinggi : 50px
                Center(
                  child: Form(
                    key: _fbKey,
                    child: Container(
                      width: MediaQuery.of(context).size.width / 1.3,
                      child: Column(
                        children: [
                          FormBuilderTextField(
                            name: "email",
                            controller: emailController,
                            decoration: InputDecoration(
                                contentPadding: EdgeInsets.only(left: 10),
                                border: OutlineInputBorder(),
                                labelText: "Email"),
                          ),
                          SizedBox(
                            height: 20,
                          ),
                          FormBuilderTextField(
                            obscureText:
                                true, // <--
Buat bikin setiap inputan jadi bintang " * "
                            name: "password",
                            controller: passwordController,

                            decoration: InputDecoration(
                                contentPadding: EdgeInsets.only(left: 10),
                                border: OutlineInputBorder(),
                                labelText: "Password"),
                          ),
                          SizedBox(
                            height: 30,
                          ),
                          Text.rich(
                            TextSpan(
                              children: [
                                TextSpan(
                                  text: 'Dont Have Account ? ',
                                  style: TextStyle(color: Colors.black54),
                                ),
```

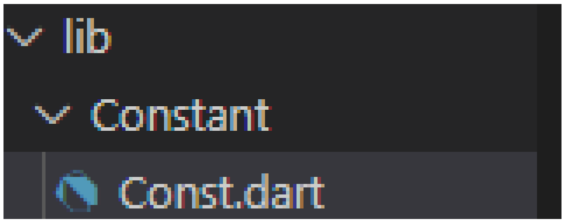


```
                                TextSpan(
                                    text: 'Sign Up',
                                    recognizer: TapGestureRecognizer()
                                      ..onTap = () {
                                        Navigator.push(
                                            context,
                                            new MaterialPageRoute(
                                                builder:
                                                    (BuildContext context) =>
                                                        new Register()));
                                      },
                                    style: TextStyle(
                                      fontWeight: FontWeight.w600,
                                      color: Colors.blueAccent,
                                    )),
                              ],
                            ),
                          ),
                          SizedBox(
                            height: 40,
                          ),
                          Container(
                            width: MediaQuery.of(context).size.width,
                            child: ElevatedButton(
                                onPressed: () async {

                                },
                                child: Text("Sign In")),
                          ),
                        ],
                      ),
                    ),
                  ),
                )
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```



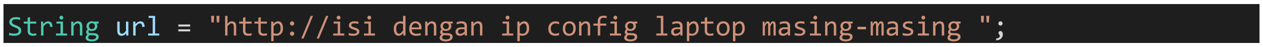
Hasil dari running file login.dart

7. Setelah selesai membuat Login . Buatlah Folder ***Constant*** di dalam folder Lib dan file nya ***Const.dart*** :



Folder **Constant** dan file **const** dimana berfungsi menampung string URL IP addres perangkat

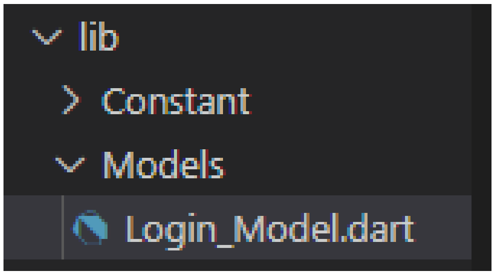
Const.dart :



```
String url = "http://isi dengan ip config laptop masing-masing ";
```

Langkah mengetahui ip laptop:

8. Setelah itu buatlah folder ***Models*** di dalam Lib dan buat file dengan nama ***Login_Model.dart*** :

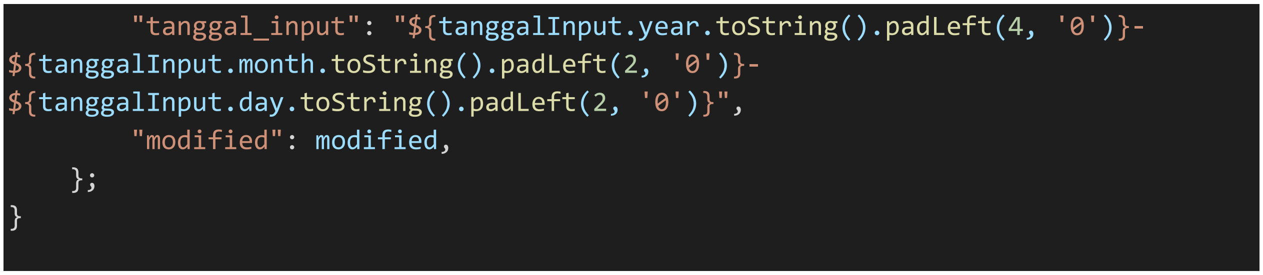


Folder **Models** dan file **Login_Model.dart** ini berfungsi menampung database dan table *user* dari API ke flutter

Login_Model.dart :

```
import 'dart:convert';

LoginModels loginModelsFromJson(String str) => LoginModels.fromJson(json.decode(str));

String loginModelsToJson(LoginModels data) => json.encode(data.toJson());

class LoginModels {
    LoginModels({
        this.isActive,
        this.message,
        this.data,
    });

    bool isActive;
    String message;
    Data data;

    factory LoginModels.fromJson(Map<String, dynamic> json) => LoginModels(
        isActive: json["is_active"],
        message: json["message"],
        data: Data.fromJson(json["data"]),
    );

    Map<String, dynamic> toJson() => {
        "is_active": isActive,
        "message": message,
        "data": data.toJson(),
    };
}
```

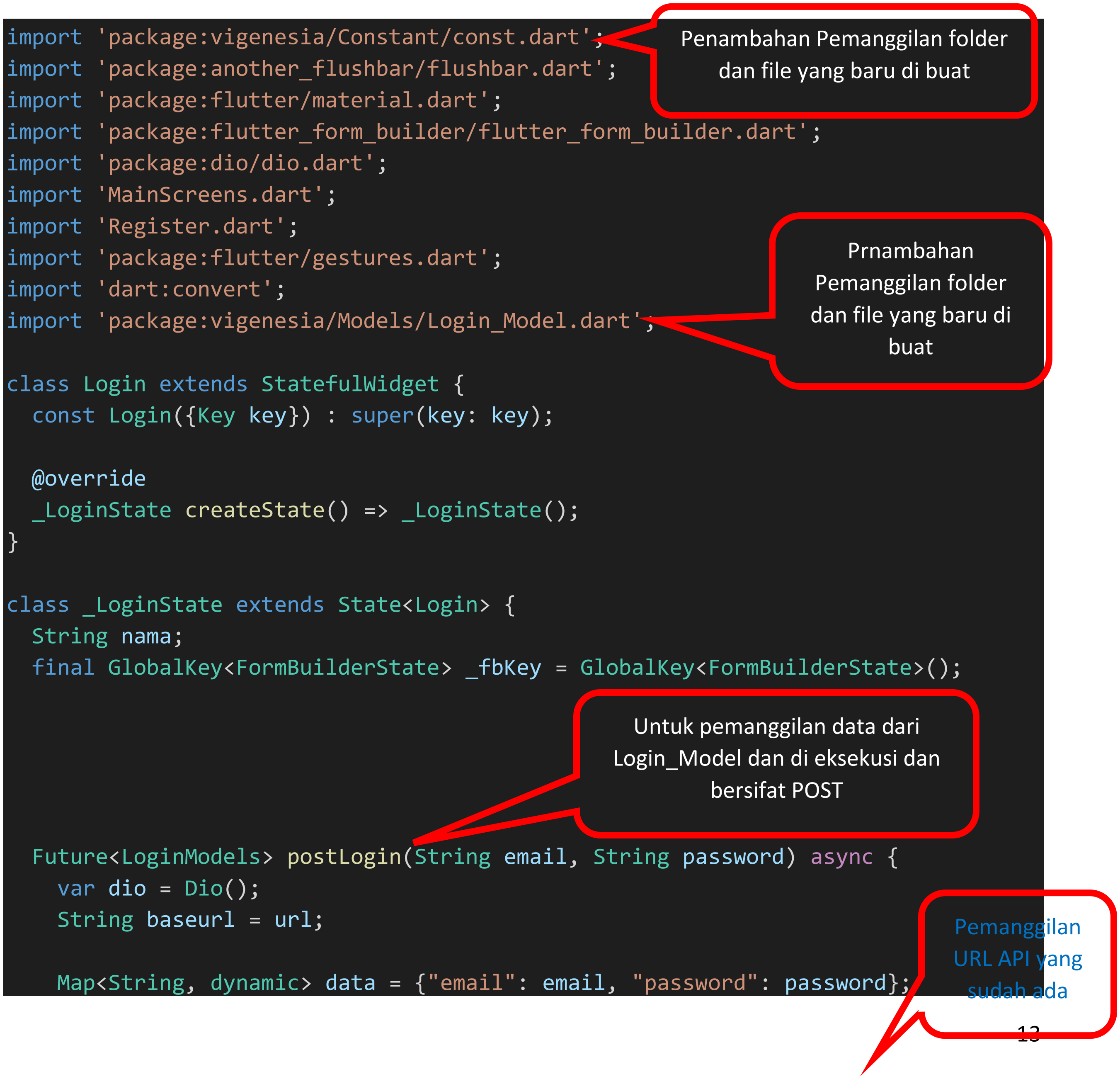
```
class Data {
    Data({
        this.iduser,
        this.nama,
        this.profesi,
        this.email,
        this.password,
        this.roleId,
        this.isActive,
        this.tanggalInput,
        this.modified,
    });

    String iduser;
    String nama;
    String profesi;
    String email;
    String password;
    String roleId;
    String isActive;
    DateTime tanggalInput;
    String modified;

    factory Data.fromJson(Map<String, dynamic> json) => Data(
        iduser: json["iduser"],
        nama: json["nama"],
        profesi: json["profesi"],
        email: json["email"],
        password: json["password"],
        roleId: json["role_id"],
        isActive: json["is_active"],
        tanggalInput: DateTime.parse(json["tanggal_input"]),
        modified: json["modified"],
    );

    Map<String, dynamic> toJson() => {
        "iduser": iduser,
        "nama": nama,
        "profesi": profesi,
        "email": email,
        "password": password,
        "role_id": roleId,
        "is_active": isActive,
```



```
        "tanggal_input": "${tanggalInput.year.toString().padLeft(4, '0')}-
${tanggalInput.month.toString().padLeft(2, '0')}-
${tanggalInput.day.toString().padLeft(2, '0')}",
        "modified": modified,
    };
}
```

9. Setelah file **Cons.dart** dan **Login_Model.dart** di buat mari kita Kembali ke folder **screens** dan ke file **Login** agar bisa memasukan parsing data API ke Flutter .

Perbarui file **Login.dart** :



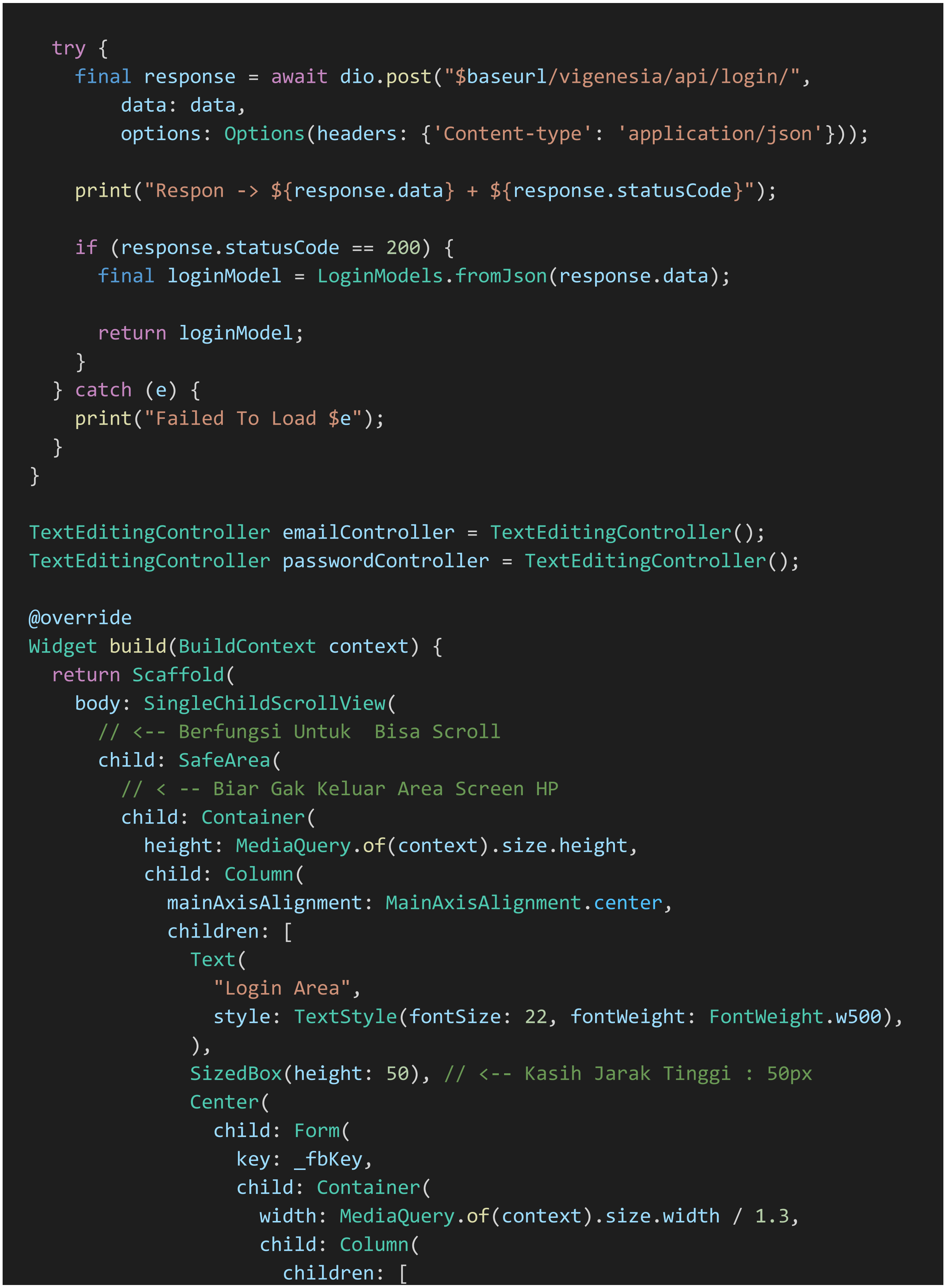
```
import 'package:vigenesia/Constant/const.dart';
import 'package:another_flushbar/flushbar.dart';
import 'package:flutter/material.dart';
import 'package:flutter_form_builder/flutter_form_builder.dart';
import 'package:dio/dio.dart';
import 'MainScreens.dart';
import 'Register.dart';
import 'package:flutter/gestures.dart';
import 'dart:convert';
import 'package:vigenesia/Models/Login_Model.dart';

class Login extends StatefulWidget {
  const Login({Key key}) : super(key: key);

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

class _LoginState extends State<Login> {
  String nama;
  final GlobalKey<FormBuilderState> _fbKey = GlobalKey<FormBuilderState>();




  Future<LoginModels> postLogin(String email, String password) async {
    var dio = Dio();
    String baseurl = url;

    Map<String, dynamic> data = {"email": email, "password": password};
```

Penambahan Pemanggilan folder dan file yang baru di buat

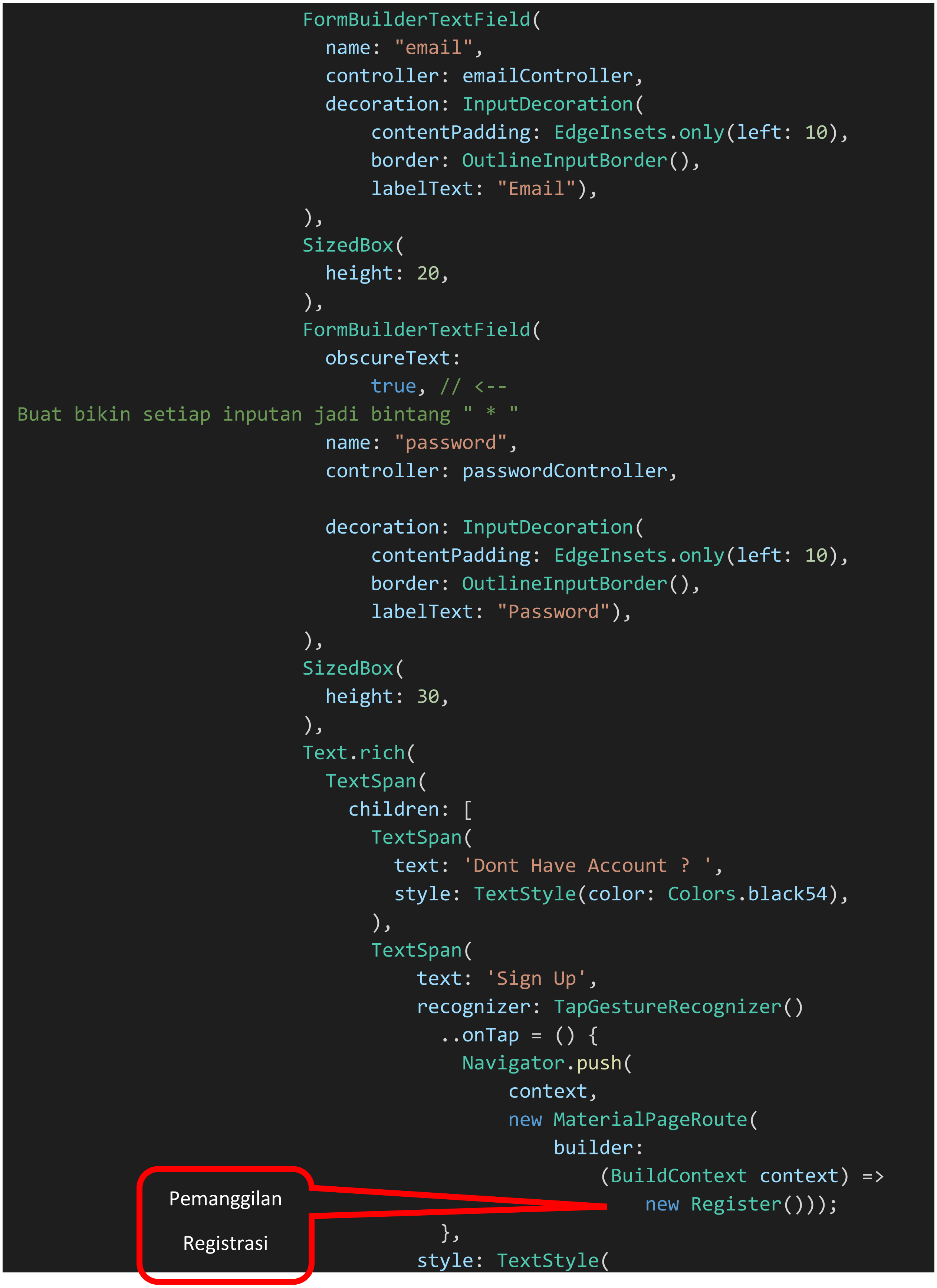
Prnambahan Pemanggilan folder dan file yang baru di buat

Untuk pemanggilan data dari Login_Model dan di eksekusi dan bersifat POST

Pemanggilan URL API yang sudah ada



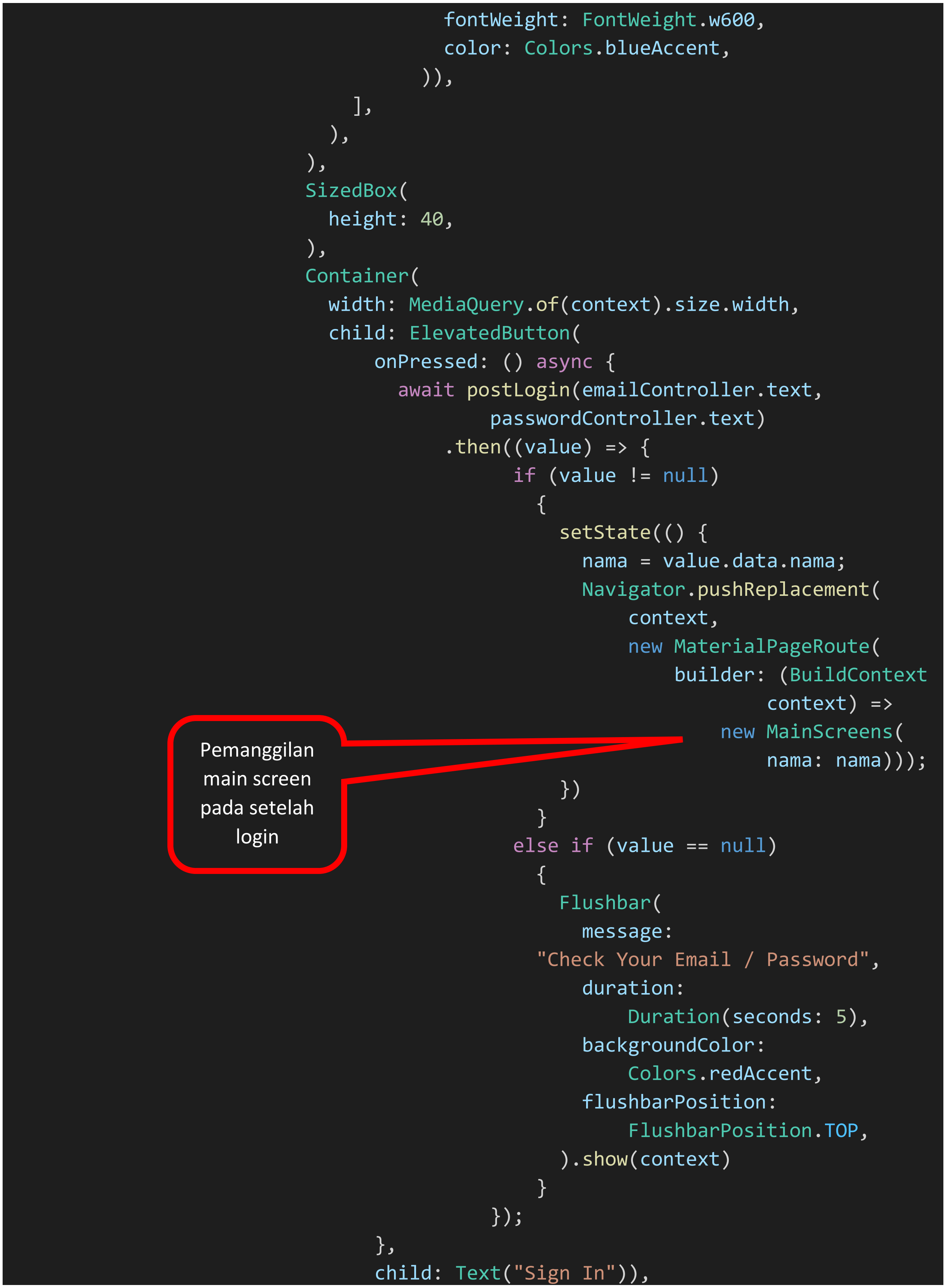
```
  try {
    final response = await dio.post("$baseurl/vigenesia/api/login/",
        data: data,
        options: Options(headers: {'Content-type': 'application/json'}));

    print("Respon -> ${response.data} + ${response.statusCode}");

    if (response.statusCode == 200) {
      final loginModel = LoginModels.fromJson(response.data);

      return loginModel;
    }
  } catch (e) {
    print("Failed To Load $e");
  }
}

TextEditingController emailController = TextEditingController();
TextEditingController passwordController = TextEditingController();

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      // <-- Berfungsi Untuk  Bisa Scroll
      child: SafeArea(
        // < -- Biar Gak Keluar Area Screen HP
        child: Container(
          height: MediaQuery.of(context).size.height,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                "Login Area",
                style: TextStyle(fontSize: 22, fontWeight: FontWeight.w500),
              ),
              SizedBox(height: 50), // <-- Kasih Jarak Tinggi : 50px
              Center(
                child: Form(
                  key: _fbKey,
                  child: Container(
                    width: MediaQuery.of(context).size.width / 1.3,
                    child: Column(
                      children: [
```



```
                         FormBuilderTextField(
                           name: "email",
                           controller: emailController,
                           decoration: InputDecoration(
                               contentPadding: EdgeInsets.only(left: 10),
                               border: OutlineInputBorder(),
                               labelText: "Email"),
                         ),
                         SizedBox(
                           height: 20,
                         ),
                         FormBuilderTextField(
                           obscureText:
                               true, // <--
Buat bikin setiap inputan jadi bintang " * "
                           name: "password",
                           controller: passwordController,

                           decoration: InputDecoration(
                               contentPadding: EdgeInsets.only(left: 10),
                               border: OutlineInputBorder(),
                               labelText: "Password"),
                         ),
                         SizedBox(
                           height: 30,
                         ),
                         Text.rich(
                           TextSpan(
                             children: [
                               TextSpan(
                                 text: 'Dont Have Account ? ',
                                 style: TextStyle(color: Colors.black54),
                               ),
                               TextSpan(
                                   text: 'Sign Up',
                                   recognizer: TapGestureRecognizer()
                                     ..onTap = () {
                                       Navigator.push(
                                           context,
                                           new MaterialPageRoute(
                                               builder:
                                                   (BuildContext context) =>
                                                       new Register()));
                                     },
                                   style: TextStyle(
```

Pemanggilan

Registrasi

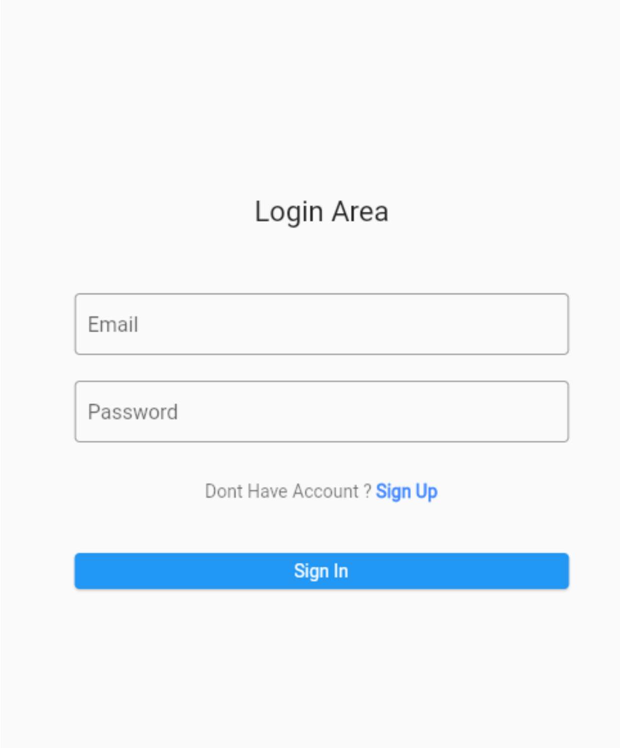


```
            fontWeight: FontWeight.w600,
            color: Colors.blueAccent,
          )),
    ],
  ),
),
SizedBox(
  height: 40,
),
Container(
  width: MediaQuery.of(context).size.width,
  child: ElevatedButton(
      onPressed: () async {
        await postLogin(emailController.text,
                passwordController.text)
            .then((value) => {
                  if (value != null)
                    {
                      setState(() {
                        nama = value.data.nama;
                        Navigator.pushReplacement(
                            context,
                            new MaterialPageRoute(
                                builder: (BuildContext
                                        context) =>
                                    new MainScreens(
                                        nama: nama)));
                      })
                    }
                  else if (value == null)
                    {
                      Flushbar(
                        message:
                    "Check Your Email / Password",
                        duration:
                            Duration(seconds: 5),
                        backgroundColor:
                            Colors.redAccent,
                        flushbarPosition:
                            FlushbarPosition.TOP,
                      ).show(context)
                    }
                });
      },
      child: Text("Sign In")),
```

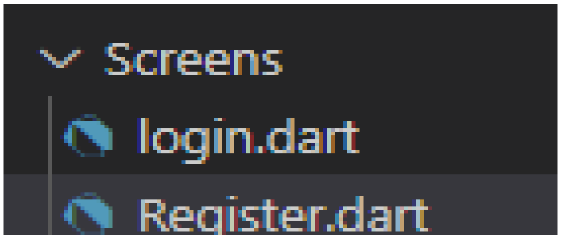
Pemanggilan main screen pada setelah login

```
                          ),
                        ],
                      ),
                    ),
                  ),
                )
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

Tampilan akan sama seperti di awal kita run hanya membedakan sudah memasukan data URL API yang sudah tersedia



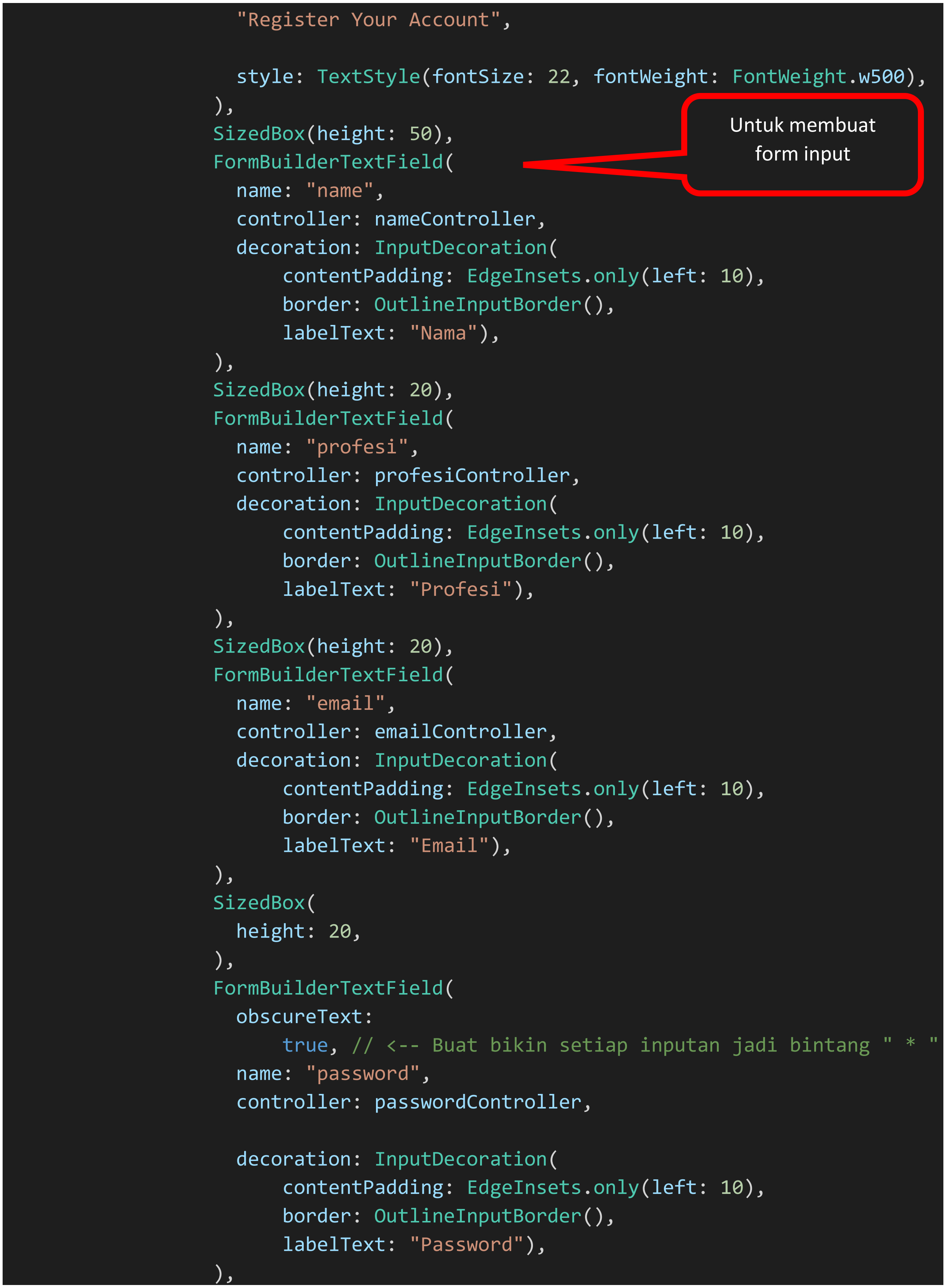
10. Setelah itu mari kita buat file **Register.dart** di dalam folder **screens** :



Disini kita akan membuat **UI** file **Register.dart** Sebelum memasukan URL API :

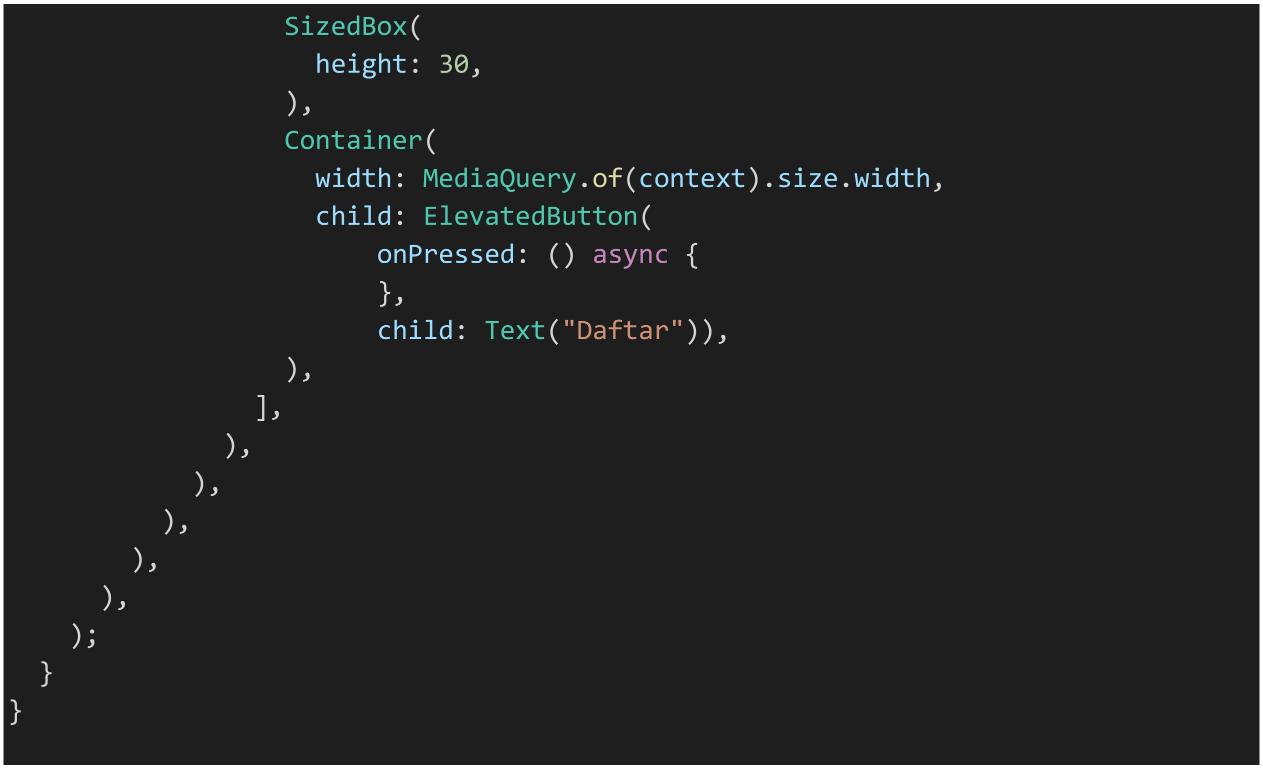
Register.dart :

```
import 'package:flutter/material.dart';
import 'package:another_flushbar/flushbar.dart';
import 'package:flutter_form_builder/flutter_form_builder.dart';
import 'package:dio/dio.dart';

class Register extends StatefulWidget {
  const Register({Key key}) : super(key: key);

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

class _RegisterState extends State<Register> {


  TextEditingController nameController = TextEditingController();
  TextEditingController profesiController = TextEditingController();
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: SafeArea(
          child: Center(
            child: Container(
              width: MediaQuery.of(context).size.width / 1.3,
              height: MediaQuery.of(context).size.height,
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Text(
```

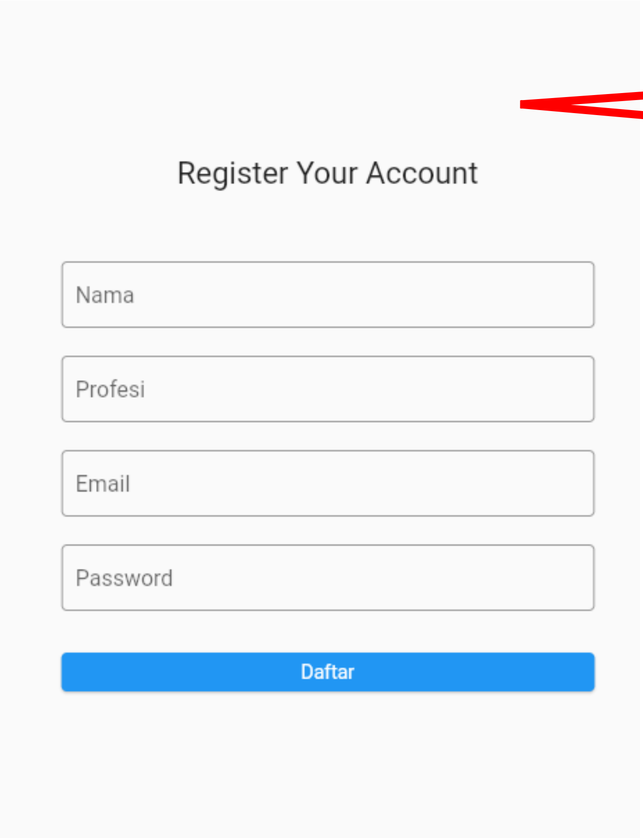


```
  "Register Your Account",

  style: TextStyle(fontSize: 22, fontWeight: FontWeight.w500),
),
SizedBox(height: 50),
FormBuilderTextField(
  name: "name",
  controller: nameController,
  decoration: InputDecoration(
      contentPadding: EdgeInsets.only(left: 10),
      border: OutlineInputBorder(),
      labelText: "Nama"),
),
SizedBox(height: 20),
FormBuilderTextField(
  name: "profesi",
  controller: profesiController,
  decoration: InputDecoration(
      contentPadding: EdgeInsets.only(left: 10),
      border: OutlineInputBorder(),
      labelText: "Profesi"),
),
SizedBox(height: 20),
FormBuilderTextField(
  name: "email",
  controller: emailController,
  decoration: InputDecoration(
      contentPadding: EdgeInsets.only(left: 10),
      border: OutlineInputBorder(),
      labelText: "Email"),
),
SizedBox(
  height: 20,
),
FormBuilderTextField(
  obscureText:
      true, // <-- Buat bikin setiap inputan jadi bintang " * "
  name: "password",
  controller: passwordController,

  decoration: InputDecoration(
      contentPadding: EdgeInsets.only(left: 10),
      border: OutlineInputBorder(),
      labelText: "Password"),
),
```

Untuk membuat form input



```
                  SizedBox(
                    height: 30,
                  ),
                  Container(
                    width: MediaQuery.of(context).size.width,
                    child: ElevatedButton(
                        onPressed: () async {
                        },
                        child: Text("Daftar")),
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```



Register Your Account

Nama

Profesi

Email

Password

Hasil dari running file
Register.dart

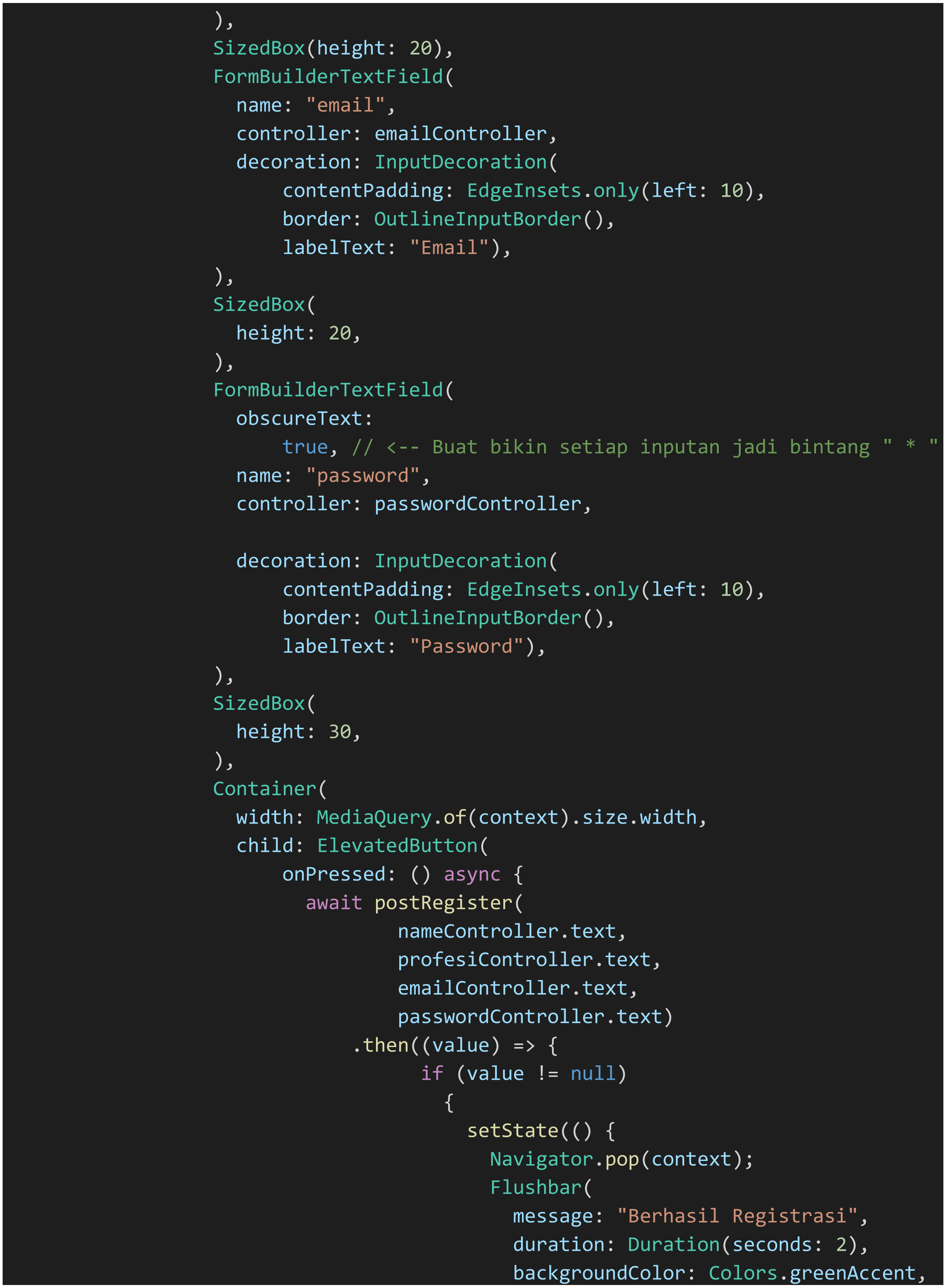
11. Hasil tersebut masi dalam UI belum memasukan data URL dari API yang tersedia dan masi kita Kembali ke file **Register.dart** supaya bisa memasukan data URL API yang telah di sediakan :

Perbarui Register.dart :

```
import 'package:vigenesia/Constant/const.dart';
import 'package:flutter/material.dart';
import 'package:another_flushbar/flushbar.dart';
import 'package:flutter_form_builder/flutter_form_builder.dart';
import 'package:dio/dio.dart';

class Register extends StatefulWidget {
  const Register({Key key}) : super(key: key);

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

class _RegisterState extends State<Register> {
  // Ganti Base URL




  String baseurl = url;

  Future postRegister(
      String nama, String profesi, String email, String password) async {
    var dio = Dio();

    dynamic data = {
      "nama": nama,
      "profesi": profesi,
      "email": email,
      "password": password
    };

    try {
      final response = await dio.post("$baseurl/vigenesia/api/registrasi/",
          data: data,
          options: Options(headers: {'Content-type': 'application/json'}));

      print("Respon -> ${response.data} + ${response.statusCode}");
```

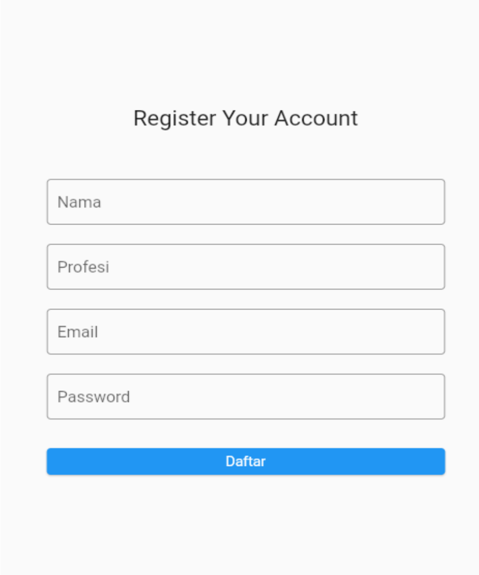
Untuk pemanggilan data dari Login_Modeldan di eksekusi dan bersifat POST

```
    if (response.statusCode == 200) {
      return response.data;
    }
  } catch (e) {
    print("Failed To Load $e");
  }
}

TextEditingController nameController = TextEditingController();
TextEditingController profesiController = TextEditingController();
TextEditingController emailController = TextEditingController();
TextEditingController passwordController = TextEditingController();
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: SafeArea(
        child: Center(
          child: Container(
            width: MediaQuery.of(context).size.width / 1.3,
            height: MediaQuery.of(context).size.height,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Text(
                  "Register Your Account",
                  style: TextStyle(fontSize: 22, fontWeight: FontWeight.w500),
                ),
                SizedBox(height: 50),
                FormBuilderTextField(
                  name: "name",
                  controller: nameController,
                  decoration: InputDecoration(
                      contentPadding: EdgeInsets.only(left: 10),
                      border: OutlineInputBorder(),
                      labelText: "Nama"),
                ),
                SizedBox(height: 20),
                FormBuilderTextField(
                  name: "profesi",
                  controller: profesiController,
                  decoration: InputDecoration(
                      contentPadding: EdgeInsets.only(left: 10),
                      border: OutlineInputBorder(),
                      labelText: "Profesi"),
```

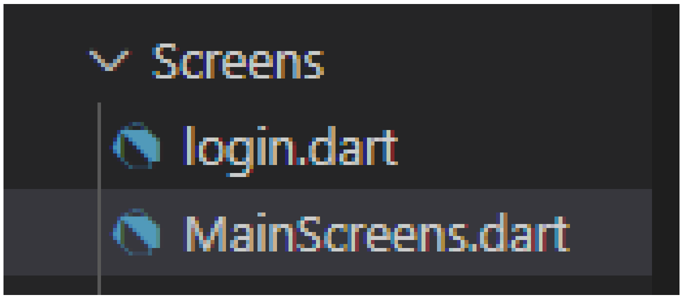


```
),
SizedBox(height: 20),
FormBuilderTextField(
  name: "email",
  controller: emailController,
  decoration: InputDecoration(
      contentPadding: EdgeInsets.only(left: 10),
      border: OutlineInputBorder(),
      labelText: "Email"),
),
SizedBox(
  height: 20,
),
FormBuilderTextField(
  obscureText:
      true, // <-- Buat bikin setiap inputan jadi bintang " * "
  name: "password",
  controller: passwordController,

  decoration: InputDecoration(
      contentPadding: EdgeInsets.only(left: 10),
      border: OutlineInputBorder(),
      labelText: "Password"),
),
SizedBox(
  height: 30,
),
Container(
  width: MediaQuery.of(context).size.width,
  child: ElevatedButton(
      onPressed: () async {
        await postRegister(
                nameController.text,
                profesiController.text,
                emailController.text,
                passwordController.text)
            .then((value) => {
                  if (value != null)
                    {
                      setState(() {
                        Navigator.pop(context);
                        Flushbar(
                          message: "Berhasil Registrasi",
                          duration: Duration(seconds: 2),
                          backgroundColor: Colors.greenAccent,
                                            flushbarPosition:
                                                FlushbarPosition.TOP,
                                          ).show(context);
                                        })
                                      }
                                    else if (value == null)
                                      {
                                        Flushbar(
                                          message:
                                              "Check Your Field Before Register",
                                          duration: Duration(seconds: 5),
                                          backgroundColor: Colors.redAccent,
                                          flushbarPosition:
                                              FlushbarPosition.TOP,
                                        ).show(context)
                                      }
                                  });
                        },
                        child: Text("Daftar")),
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

Hasil akan sama dan hanya membedakan terdapat Data URL dari API

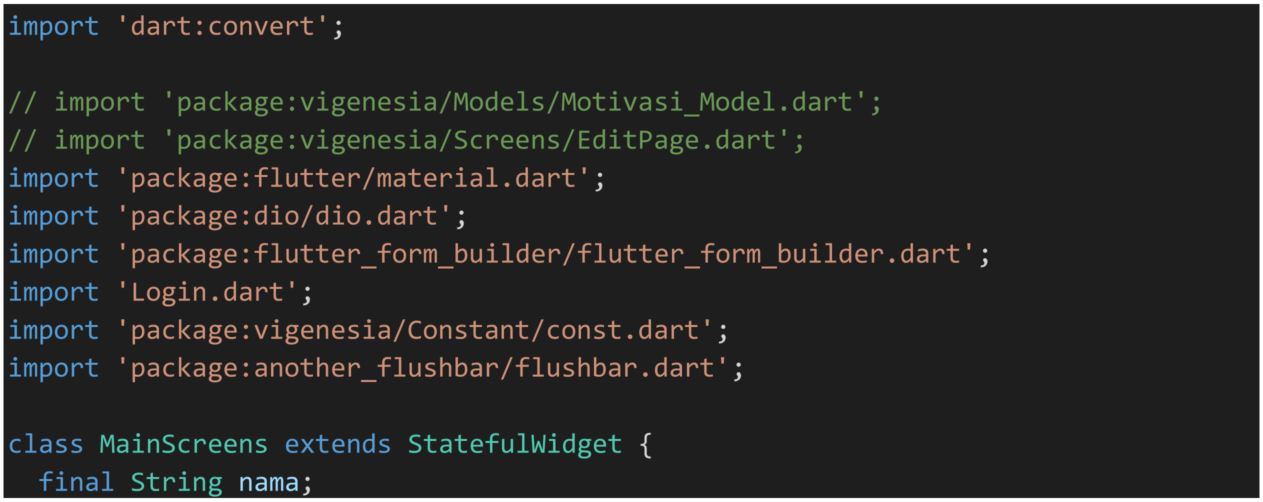


12. Setelah itu kita akan membuat file **MainScreens.dart** di dalam folder **Screens** :

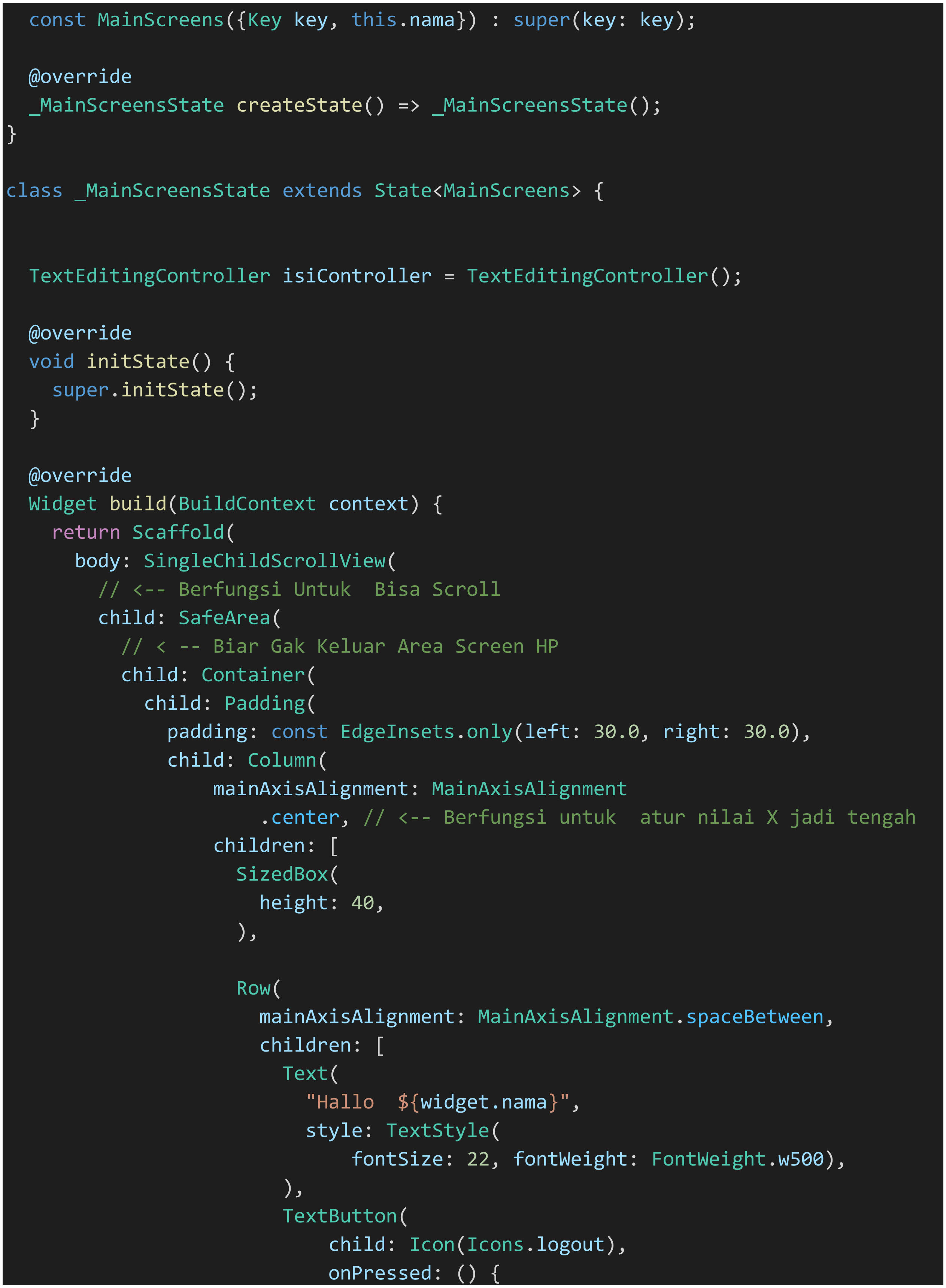


File **MainScreens.dart** berfungsi untuk menampilkan tampilan setelah melakukan authentifikasi

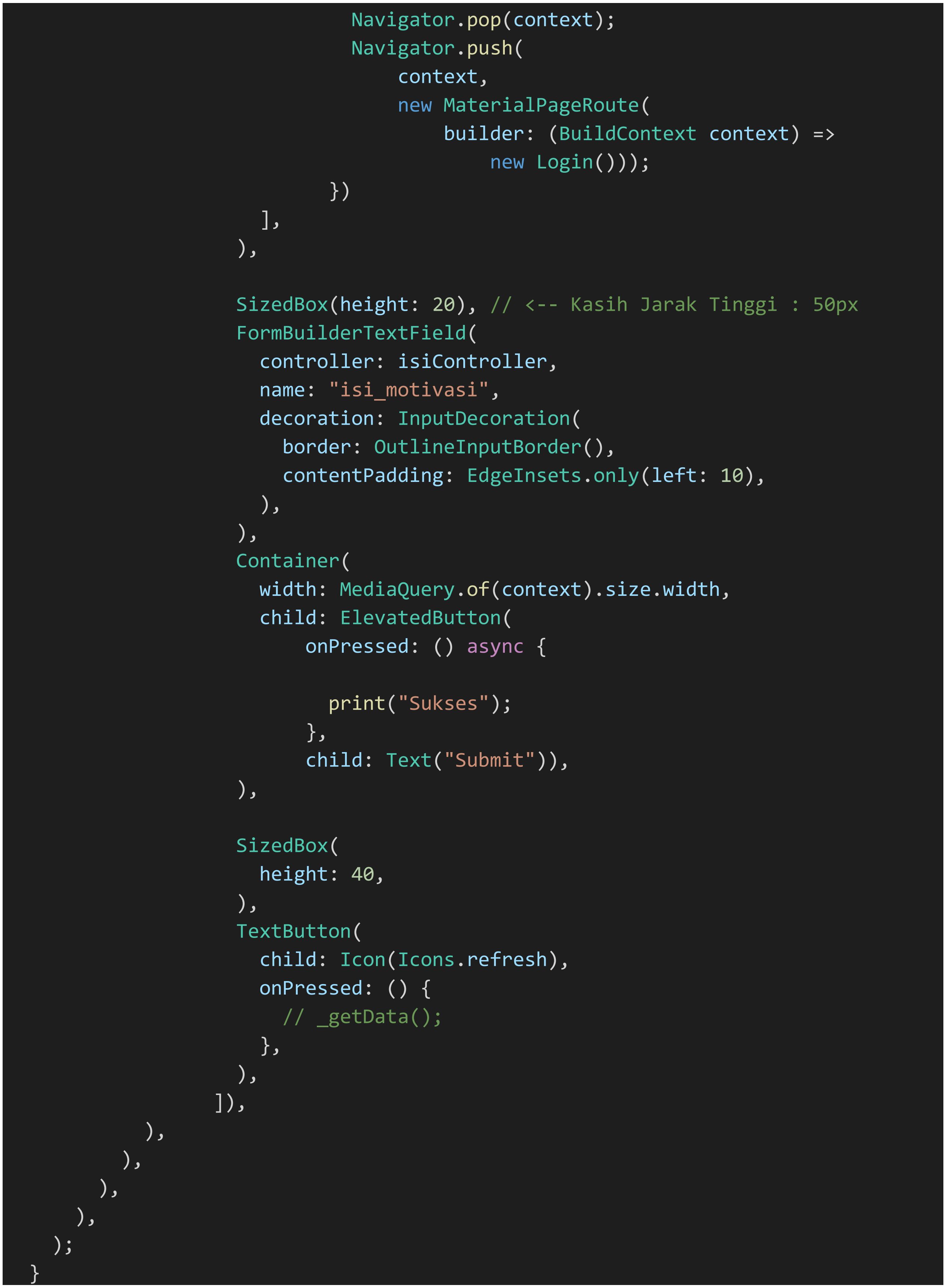
UI MainScreens.dart :



```
import 'dart:convert';

// import 'package:vigenesia/Models/Motivasi_Model.dart';
// import 'package:vigenesia/Screens/EditPage.dart';
import 'package:flutter/material.dart';
import 'package:dio/dio.dart';
import 'package:flutter_form_builder/flutter_form_builder.dart';
import 'Login.dart';
import 'package:vigenesia/Constant/const.dart';
import 'package:another_flushbar/flushbar.dart';

class MainScreens extends StatefulWidget {
  final String nama;
```



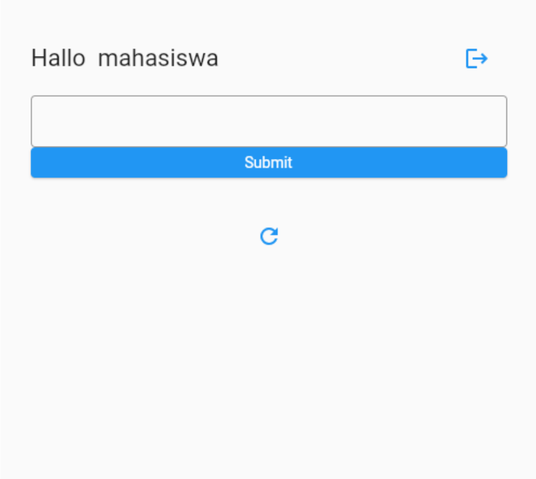
```
  const MainScreens({Key key, this.nama}) : super(key: key);

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

class _MainScreensState extends State<MainScreens> {


  TextEditingController isiController = TextEditingController();

  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        // <-- Berfungsi Untuk  Bisa Scroll
        child: SafeArea(
          // < -- Biar Gak Keluar Area Screen HP
          child: Container(
            child: Padding(
              padding: const EdgeInsets.only(left: 30.0, right: 30.0),
              child: Column(
                  mainAxisAlignment: MainAxisAlignment
                      .center, // <-- Berfungsi untuk  atur nilai X jadi tengah
                  children: [
                    SizedBox(
                      height: 40,
                    ),

                    Row(
                      mainAxisAlignment: MainAxisAlignment.spaceBetween,
                      children: [
                        Text(
                          "Hallo  ${widget.nama}",
                          style: TextStyle(
                              fontSize: 22, fontWeight: FontWeight.w500),
                        ),
                        TextButton(
                            child: Icon(Icons.logout),
                            onPressed: () {
```



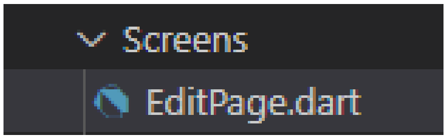
```
                            Navigator.pop(context);
                            Navigator.push(
                                context,
                                new MaterialPageRoute(
                                    builder: (BuildContext context) =>
                                        new Login()));
                          })
                    ],
                  ),

                  SizedBox(height: 20), // <-- Kasih Jarak Tinggi : 50px
                  FormBuilderTextField(
                    controller: isiController,
                    name: "isi_motivasi",
                    decoration: InputDecoration(
                      border: OutlineInputBorder(),
                      contentPadding: EdgeInsets.only(left: 10),
                    ),
                  ),
                  Container(
                    width: MediaQuery.of(context).size.width,
                    child: ElevatedButton(
                        onPressed: () async {

                          print("Sukses");
                        },
                        child: Text("Submit")),
                  ),

                  SizedBox(
                    height: 40,
                  ),
                  TextButton(
                    child: Icon(Icons.refresh),
                    onPressed: () {
                      // _getData();
                    },
                  ),
                ]),
          ),
        ),
      ),
    ),
  );
}
```

```
}
```

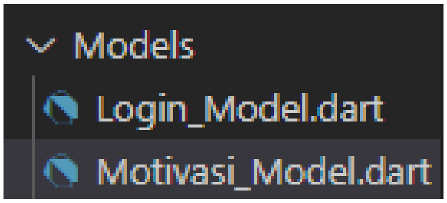
Hasil dari code tersebut menghasilkan tampilan seperti di bawah ini :



13. Disini tampilan belum ada data dari URL API jadi kita akan membuat data agar bisa masuk dengan memperbarui **MainScreens.dart** supaya bisa memasukan data URL API. Buat terlebih dahulu **EditPage.dart** di dalam Folder Screens dan membuat File **Motivasi_Model.dart** di dalam Folder **Models**.

1. 

Screens
EditPage.dart

2. 

Models
Login_Model.dart
Motivasi_Model.dart

Motivasi_Model.dart:

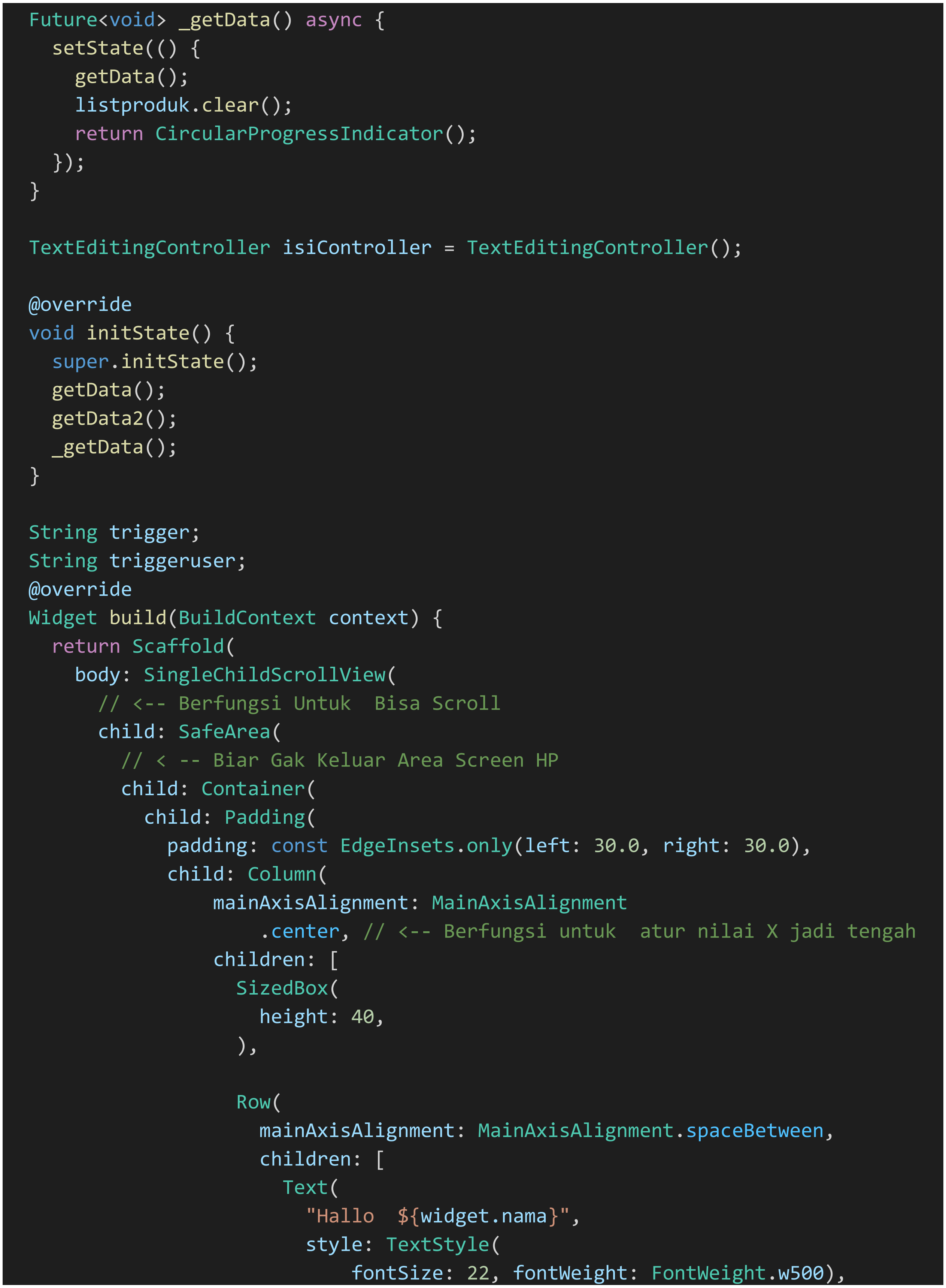
```
import 'dart:convert';

List<MotivasiModel> motivasiModelFromJson(String str) =>
    List<MotivasiModel>.from(
        json.decode(str).map((x) => MotivasiModel.fromJson(x)));

String motivasiModelToJson(List<MotivasiModel> data) =>
```

```
    json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class MotivasiModel {
  MotivasiModel({
    this.id,
    this.isiMotivasi,
    this.idKategori,
    this.tanggalInput,
    this.tanggalUpdate,
  });

  String id;
  String isiMotivasi;
  String idKategori;
  DateTime tanggalInput;
  String tanggalUpdate;

  factory MotivasiModel.fromJson(Map<String, dynamic> json) => MotivasiModel(
        id: json["id"],
        isiMotivasi: json["isi_motivasi"],
        idKategori: json["id_kategori"],
        tanggalInput: DateTime.parse(json["tanggal_input"]),
        tanggalUpdate: json["tanggal_update"],
      );

  Map<String, dynamic> toJson() => {
        "id": id,
        "isi_motivasi": isiMotivasi,
        "id_kategori": idKategori,
        "tanggal_input":
            "${tanggalInput.year.toString().padLeft(4, '0')}-
${tanggalInput.month.toString().padLeft(2, '0')}-
${tanggalInput.day.toString().padLeft(2, '0')}",
        "tanggal_update": tanggalUpdate,
      };
}
```

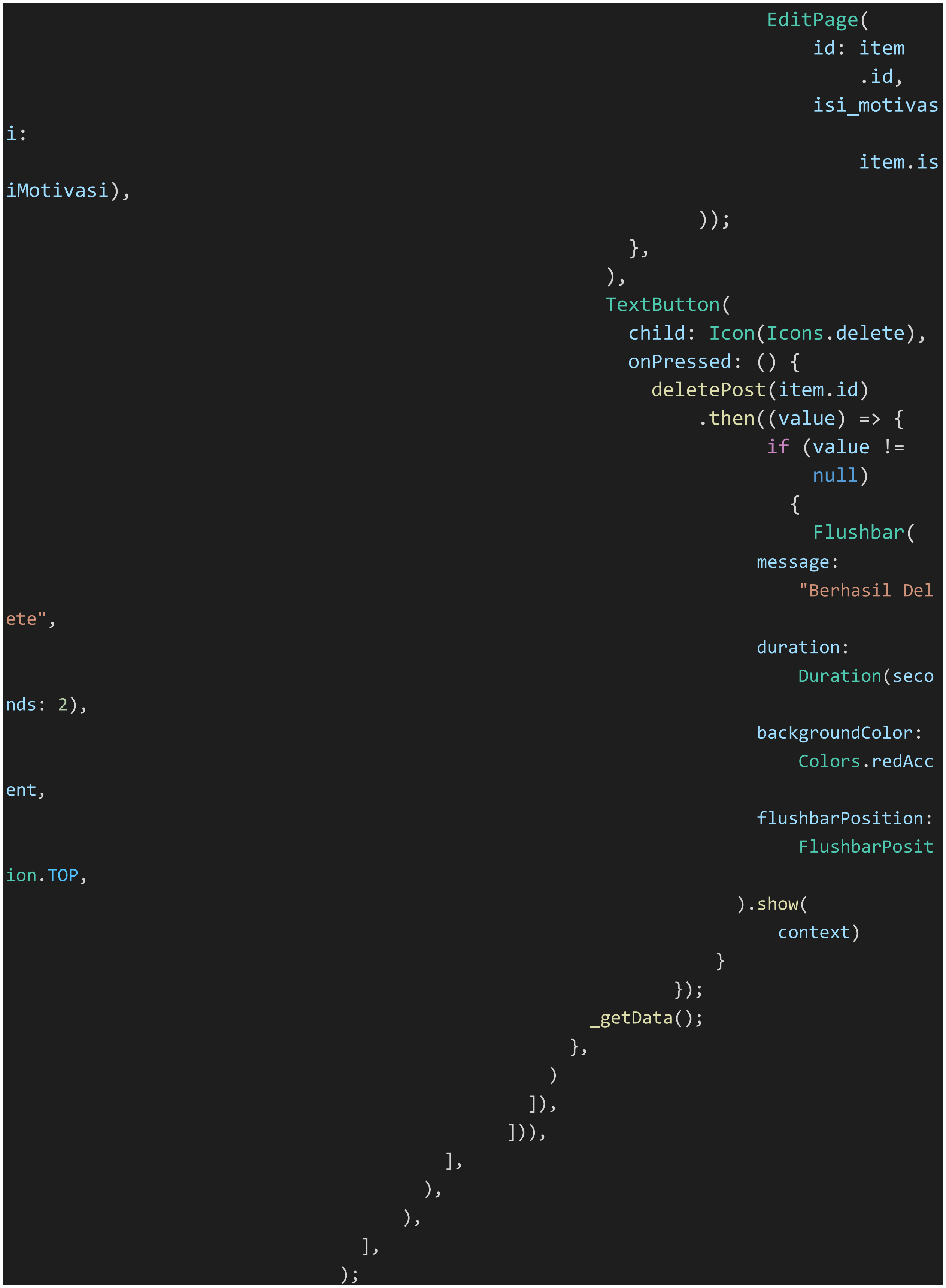
Perbarui **MainScreens.dart :**

```
import 'dart:convert';
import 'dart:convert';
import 'package:vigenesia/Models/Motivasi_Model.dart';
import 'package:vigenesia/Screens/EditPage.dart';
import 'package:flutter/material.dart';
import 'package:dio/dio.dart';
import 'package:flutter_form_builder/flutter_form_builder.dart';
import 'Login.dart';
import 'package:vigenesia/Constant/const.dart';
import 'package:another_flushbar/flushbar.dart';

class MainScreens extends StatefulWidget {
  final String idUser;
  final String nama;
  const MainScreens({Key key, this.nama, this.idUser}) : super(key: key);

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

class _MainScreensState extends State<MainScreens> {
  String baseurl = url;
  String id;
  var dio = Dio();
  List<MotivasiModel> ass = [];
  TextEditingController titleController = TextEditingController();

  Future<dynamic> sendMotivasi(String isi) async {
    Map<String, dynamic> body = {"isi_motivasi": isi, "iduser": widget.idUser}; /
/ [Tambah IDUSER -> Widget.iduser]

    try {
      Response response = await dio.post(
          "$baseurl/vigenesia/api/dev/POSTmotivasi/",
          data: body,
          options: Options(contentType: Headers.formUrlEncodedContentType)); // F
ormatnya Harus Form Data

      print("Respon -> ${response.data} + ${response.statusCode}");

      return response;
    } catch (e) {
      print("Error di -> $e");
    }
  }

  List<MotivasiModel> listproduk = [];

  Future<List<MotivasiModel>> getData() async {
    var response = await dio
```

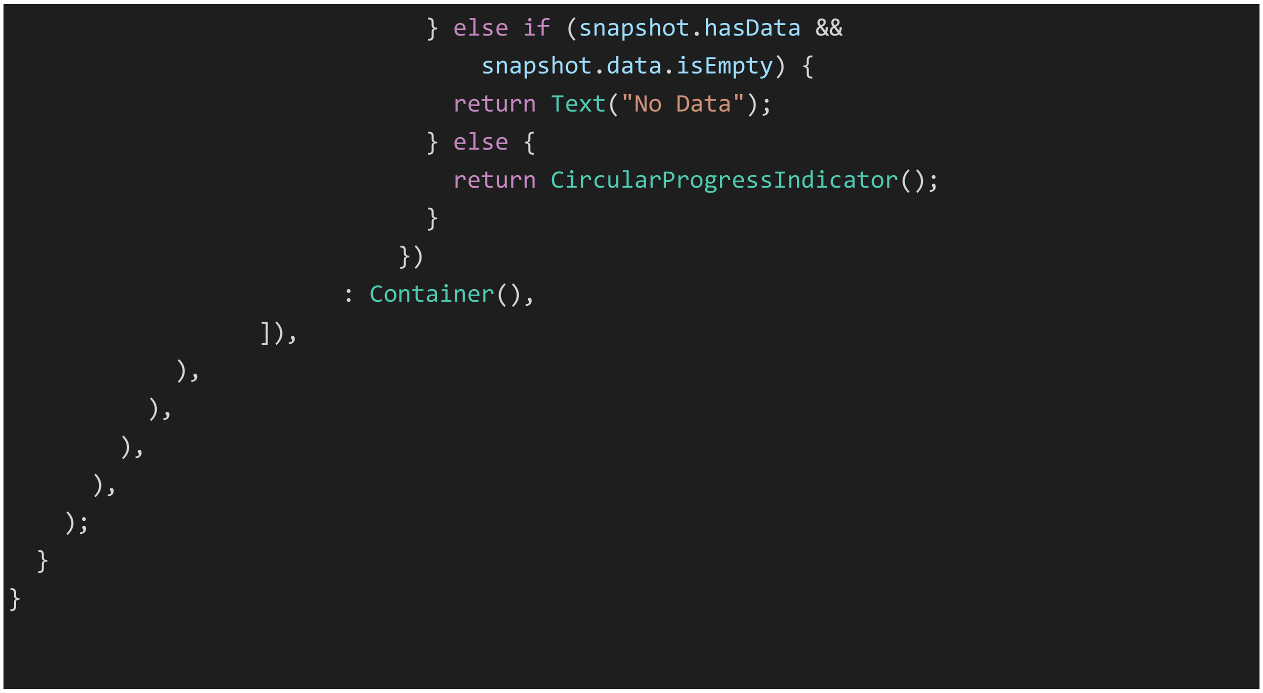
```
        .get('$baseurl/vigenesia/api/Get_motivasi?iduser=${widget.idUser}'); // N
Gambil by data

    print(" ${response.data}");
    if (response.statusCode == 200) {
      var getUsersData = response.data as List;
      var listUsers =
          getUsersData.map((i) => MotivasiModel.fromJson(i)).toList();
      return listUsers;
    } else {
      throw Exception('Failed to load');
    }
  }

  Future<dynamic> deletePost(String id) async {
    dynamic data = {
      "id": id,
    };
    var response = await dio.delete('$baseurl/vigenesia/api/dev/DELETEmotivasi',
        data: data,
        options: Options(
            contentType: Headers.formUrlEncodedContentType,
            headers: {"Content-type": "application/json"}));

    print(" ${response.data}");

    var resbody = jsonDecode(response.data);
    return resbody;
  }

  Future<List<MotivasiModel>> getData2() async {
    var response = await dio.get('$baseurl/vigenesia/api/Get_motivasi'); // Ngamb
il by ALL USER

    print(" ${response.data}");
    if (response.statusCode == 200) {
      var getUsersData = response.data as List;
      var listUsers =
          getUsersData.map((i) => MotivasiModel.fromJson(i)).toList();
      return listUsers;
    } else {
      throw Exception('Failed to load');
    }
  }
```



```
  Future<void> _getData() async {
    setState(() {
      getData();
      listproduk.clear();
      return CircularProgressIndicator();
    });
  }

  TextEditingController isiController = TextEditingController();

  @override
  void initState() {
    super.initState();
    getData();
    getData2();
    _getData();
  }

  String trigger;
  String triggeruser;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        // <-- Berfungsi Untuk  Bisa Scroll
        child: SafeArea(
          // < -- Biar Gak Keluar Area Screen HP
          child: Container(
            child: Padding(
              padding: const EdgeInsets.only(left: 30.0, right: 30.0),
              child: Column(
                  mainAxisAlignment: MainAxisAlignment
                      .center, // <-- Berfungsi untuk  atur nilai X jadi tengah
                  children: [
                    SizedBox(
                      height: 40,
                    ),

                    Row(
                      mainAxisAlignment: MainAxisAlignment.spaceBetween,
                      children: [
                        Text(
                          "Hallo  ${widget.nama}",
                          style: TextStyle(
                              fontSize: 22, fontWeight: FontWeight.w500),
                        ),
                        TextButton(
                            child: Icon(Icons.logout),
                            onPressed: () {
                              Navigator.pop(context);
                              Navigator.push(
                                  context,
                                  new MaterialPageRoute(
                                      builder: (BuildContext context) =>
                                          new Login()));
                            })
                      ],
                    ),

                    SizedBox(height: 20), // <-- Kasih Jarak Tinggi : 50px
                    FormBuilderTextField(
                      controller: isiController,
                      name: "isi_motivasi",
                      decoration: InputDecoration(
                        border: OutlineInputBorder(),
                        contentPadding: EdgeInsets.only(left: 10),
                      ),
                    ),
                    Container(
                      width: MediaQuery.of(context).size.width,
                      child: ElevatedButton(
                          onPressed: () async {
                            await sendMotivasi(
                              isiController.text.toString(),
                            ).then((value) => {
                                  if (value != null)
                                    {
                                      Flushbar(
                                        message: "Berhasil Submit",
                                        duration: Duration(seconds: 2),
                                        backgroundColor: Colors.greenAccent,
                                        flushbarPosition: FlushbarPosition.TOP,
                                      ).show(context)
                                    }
                                });

                            _getData();
                            print("Sukses");
                          },
                          child: Text("Submit")),
                    ),

                    TextButton(
                      child: Icon(Icons.refresh),
                      onPressed: () {
                        _getData();
                      },
                    ),
                    FormBuilderRadioGroup(
                        onChanged: (value) {
                          setState(() {
                            trigger = value;
                            print(" HASILNYA --
> ${trigger}"); // hasil ganti value
                          });
                        },
                        name: "_",
                       options: ["Motivasi By All", "Motivasi By User"]
                            .map((e) => FormBuilderFieldOption(
                                value: e, child: Text("${e}")))
                            .toList()),

                    trigger == "Motivasi By All"
                        ? FutureBuilder(
                            future: getData2(),
                            builder: (BuildContext context,
                                AsyncSnapshot<List<MotivasiModel>> snapshot) {
                              if (snapshot.hasData) {
                                return Container(
                                  child: Column(
                                    children: [
                                      for (var item in snapshot.data)
                                        Container(
                                          width:
                                              MediaQuery.of(context).size.width,
                                          child: ListView(
                                            shrinkWrap: true,
                                            children: [
                                              Container(
                                                  child:
                                                      Text(item.isiMotivasi)),
                                            ],
                                          ),
                                        ),
                                    ],
                                  ),
                                );
                              } else if (snapshot.hasData &&
                                  snapshot.data.isEmpty) {
                                return Text("No Data");
                              } else {
                                return CircularProgressIndicator();
                              }
                            })
                        : Container(),
                    trigger == "Motivasi By User"
                        ? FutureBuilder(
                            future: getData(),
                            builder: (BuildContext context,
                                AsyncSnapshot<List<MotivasiModel>> snapshot) {
                              if (snapshot.hasData) {
                                return Column(
                                  children: [
                                    for (var item in snapshot.data)
                                      Container(
                                        width:
                                            MediaQuery.of(context).size.width,
                                        child: ListView(
                                          shrinkWrap: true,
                                          children: [
                                            Expanded(
                                                child: Row(
                                                    mainAxisAlignment:
                                                        MainAxisAlignment
                                                            .spaceBetween,
                                                    children: [
                                                  Text(item.isiMotivasi),
                                                  Row(children: [
                                                    TextButton(
                                                      child:
                                                          Icon(Icons.settings),
                                                      onPressed: () {
                                                        String id;
                                                        String isi_motivasi;
                                                        Navigator.push(
                                                            context,
                                                            MaterialPageRoute(
                                                              builder: (BuildCont
ext
                                                                      context) =>
```

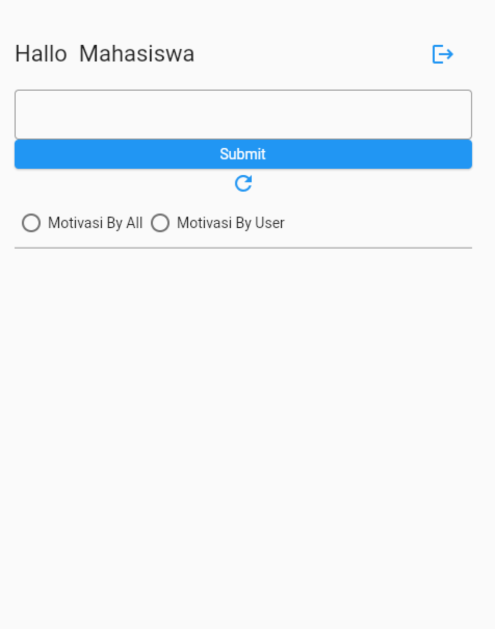


```
                                                                  EditPage(
                                                                      id: item
                                                                          .id,
                                                                      isi_motivas
i:
                                                                          item.is
iMotivasi),
                                                            ));
                                                      },
                                                    ),
                                                    TextButton(
                                                      child: Icon(Icons.delete),
                                                      onPressed: () {
                                                        deletePost(item.id)
                                                            .then((value) => {
                                                                  if (value !=
                                                                      null)
                                                                    {
                                                                      Flushbar(
                                                                        message:
                                                                            "Berhasil Del
ete",
                                                                        duration:
                                                                            Duration(seco
nds: 2),
                                                                        backgroundColor:
                                                                            Colors.redAcc
ent,
                                                                        flushbarPosition:
                                                                            FlushbarPosit
ion.TOP,
                                                                ).show(
                                                                   context)
                                                              }
                                                          });
                                                   _getData();
                                                 },
                                               )
                                             ]),
                                            ])),
                                      ],
                                    ),
                                  ),
                               ],
                             );
```



```
                              } else if (snapshot.hasData &&
                                  snapshot.data.isEmpty) {
                                return Text("No Data");
                              } else {
                                return CircularProgressIndicator();
                              }
                            })
                        : Container(),
                  ]),
            ),
          ),
        ),
      ),
    );
  }
}
```

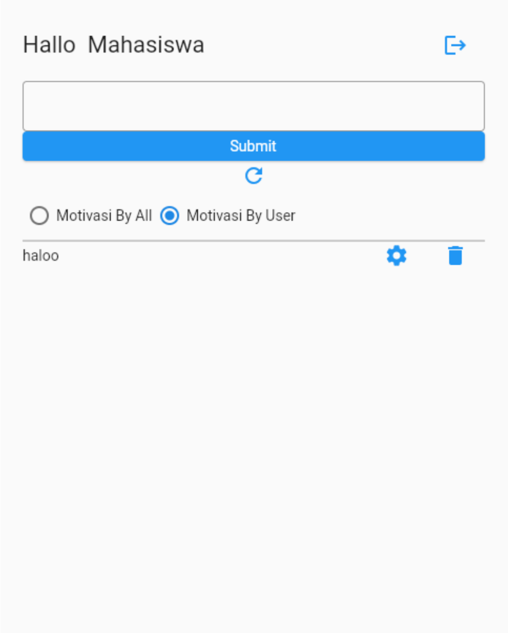
Berikut hasil pembaharuan dari **MainScreens.dart** dengan memasukan Data URL API



Hallo Mahasiswa

Submit

○ Motivasi By All ○ Motivasi By User



Hallo Mahasiswa

Submit

○ Motivasi By All ● Motivasi By User

haloo

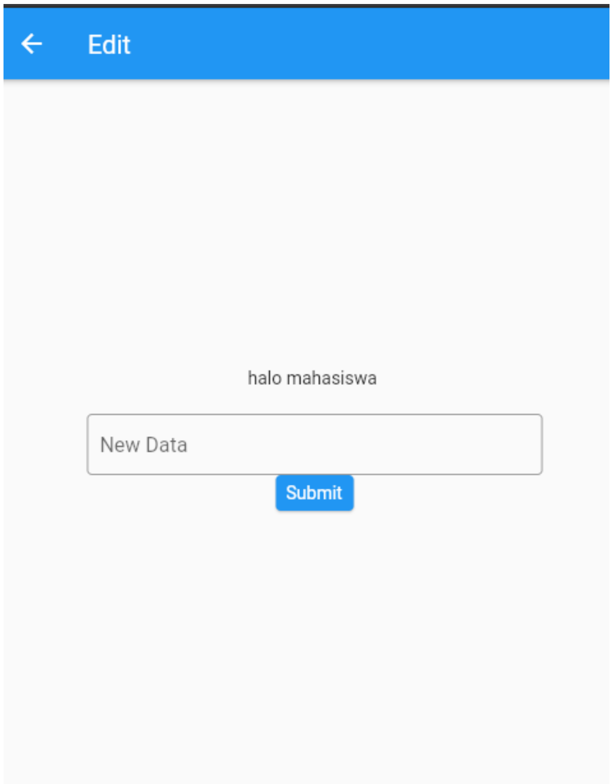
14. Langkah selanjut kita membuat fungsi edit motivasi di file **EditPage.dart** .

EditPage.dart:

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

import 'package:another_flushbar/flushbar.dart';
import 'package:dio/dio.dart';

import 'package:vigenesia/Constant/const.dart';

import 'package:vigenesia/Models/Motivasi_Model.dart';

class EditPage extends StatefulWidget {
  final String id;
  final String isi_motivasi;
  const EditPage({Key key, this.id, this.isi_motivasi}) : super(key: key);

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

class _EditPageState extends State<EditPage> {
  String baseurl = url;

  var dio = Dio();
  Future<dynamic> putPost(String isi_motivasi, String ids) async {
    Map<String, dynamic> data = {"isi_motivasi": isi_motivasi, "id": ids};
    var response = await dio.put('$baseurl/vigenesia/api/dev/PUTmotivasi',
        data: data,
        options: Options(
          contentType: Headers.formUrlEncodedContentType,
        ));

    print("---> ${response.data} + ${response.statusCode}");

    return response.data;
  }

  TextEditingController isiMotivasiC = TextEditingController();
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Edit"),
    ),
    body: SafeArea(
        child: Container(
      child: Container(
        width: MediaQuery.of(context).size.width,
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text("${widget.isi_motivasi}"),
              SizedBox(
                height: 20,
              ),
              Container(
                width: MediaQuery.of(context).size.width / 1.4,
                child: FormBuilderTextField(
                  name: "isi_motivasi",
                  controller: isiMotivasiC,
                  decoration: InputDecoration(
                    labelText: "New Data",
                    border: OutlineInputBorder(),
                    contentPadding: EdgeInsets.only(left: 10),
                  ),
                ),
              ),
              ElevatedButton(
                  onPressed: () {
                    putPost(isiMotivasiC.text, widget.id).then((value) => {
                          if (value != null)
                            {
                              Navigator.pop(context),
                              Flushbar(
                                message: "Berhasil Update & Refresh dlu",
                                duration: Duration(seconds: 5),
                                backgroundColor: Colors.green,
                                flushbarPosition: FlushbarPosition.TOP,
                              ).show(context)
                            }
                        });
                  },
                    child: Text("Submit"))
              ],
            ),
          ),
        ),
      )),
    );
  }
}
```

Hasil dari **EditPage.dart** :



15.Langkah akhir membuat file **generated_plugin_registrant.dart** dengan sejajar **main.dart** membuat file tersebut agar dimana user memstabilkan keyboard saat menginput ketikan data

```
import 'package:flutter_keyboard_visibility_web/flutter_keyboard_visibility_web.dart';

import 'package:flutter_web_plugins/flutter_web_plugins.dart';
```

```
// ignore: public_member_api_docs
void registerPlugins(Registrar registrar) {
  FlutterKeyboardVisibilityPlugin.registerWith(registrar);
  registrar.registerMessageHandler();
}
```