

Dokumentasi

Uas

Mobile Programming



Nama : Muhamad Rafid Fajar

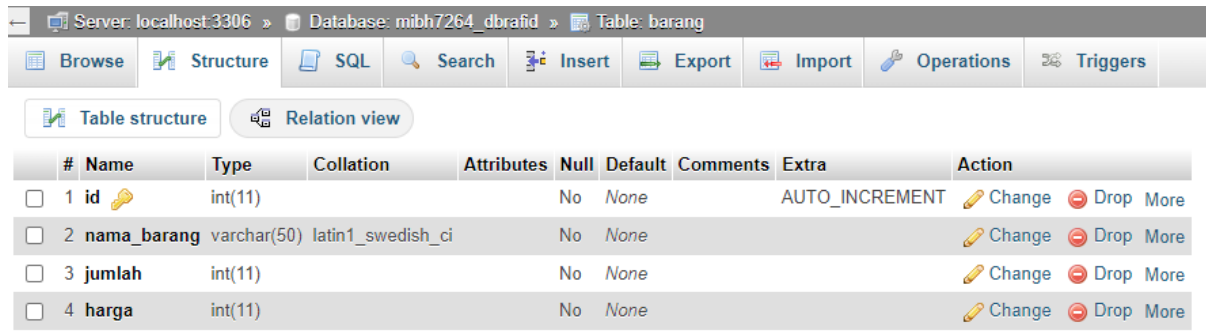
Kelas : Mi05421

NIM : 210444050047

- Aplikasi yang saya buat adalah aplikasi pengadaan barang pada suatu perusahaan

Dan untuk membaangun aplikasi ini kita membutuhkan 2 table yaitu:

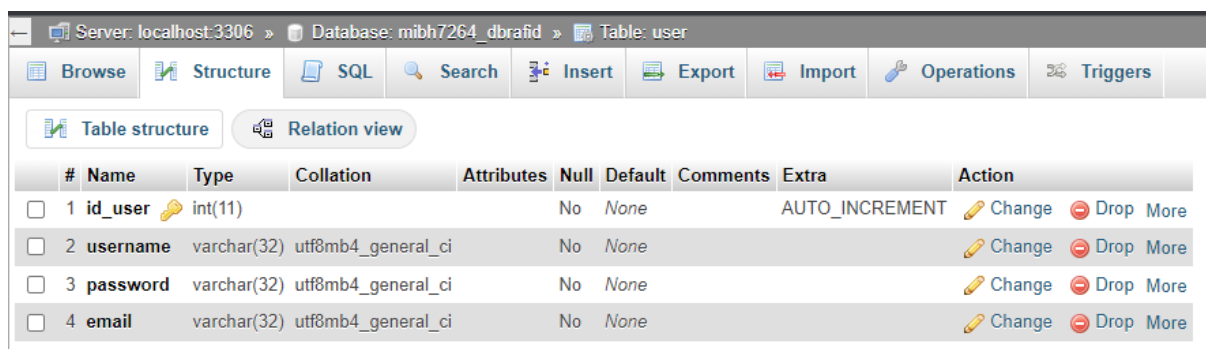
Table **barang**



The screenshot shows the MySQL Table structure for the 'barang' table. The table has four columns: 'id' (int(11), primary key, auto-increment), 'nama_barang' (varchar(50), latin1_swedish_ci), 'jumlah' (int(11)), and 'harga' (int(11)). Each column has a 'Change', 'Drop', and 'More' action link.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama_barang	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
3	jumlah	int(11)			No	None			Change Drop More
4	harga	int(11)			No	None			Change Drop More

Dan Table **user**



The screenshot shows the MySQL Table structure for the 'user' table. The table has four columns: 'id_user' (int(11), primary key, auto-increment), 'username' (varchar(32), utf8mb4_general_ci), 'password' (varchar(32), utf8mb4_general_ci), and 'email' (varchar(32), utf8mb4_general_ci). Each column has a 'Change', 'Drop', and 'More' action link.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_user	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More
3	password	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More
4	email	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More

Selanjutnya kita buat beberapa API untuk menghubungkan ke table yang kita telah buat yang disimpan pada folder hosting **public_html/api_rafid/**

Di aplikasi saya ada halaman login jadi kita buat dulu API untuk halaman Login dan registrasinya yang disimpan pada folder hosting **public_html/api_rafid/login**

login.php

```
<?php
$db = mysqli_connect('localhost','mibh7264_rafid','Wadadbwdwa%wada','mibh7264_db_rafid');

$username = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT * FROM user WHERE username = '". $username. "' AND password = '". $password. "'";

$result = mysqli_query($db,$sql);
$count = mysqli_num_rows($result);
if($count == 1){
    echo json_encode("Success");
}
```

```

}
else{
    echo json_encode("Error");
}
?>

```

Register.php

```

<?php
$db = mysqli_connect('localhost','mibh7264_rafid','Wadadbwdwa%wada&','mibh7264_dbr
afid');
if(!$db)
{
    echo "Database connection failed";
}
$username = $_POST['username'];
$password = $_POST['password'];
$email = $_POST['email'];

$sql = "SELECT username FROM user WHERE username = '". $username. "'";
$result = mysqli_query($db,$sql);
$count = mysqli_num_rows($result);
if($count == 1){
    echo json_encode("Error");
}else{
    $insert = "INSERT INTO user(username,password,email) VALUES ('". $username. "','
". $password. "','". $email. "')";
    $query = mysqli_query($db,$insert);
    if($query){
        echo json_encode("Success");
    }
}
?>

```

- Selanjutnya kita akan buat API CRUD untuk terhubung pada data barang dan simpan pada folder **public_html/api_rafid/barang**

list.php

```
<?php

$connection = new mysqli("localhost","mibh7264_rafid","Wadadbwdwa%wada&","mibh7264_dbrafid");

$data = mysqli_query($connection, "select * from barang");
$data = mysqli_fetch_all($data, MYSQLI_ASSOC);

echo json_encode($data);
```

detail.php

```
<?php

$connection = new mysqli("localhost","mibh7264_rafid","Wadadbwdwa%wada&","mibh7264_dbrafid");

$data = mysqli_query($connection, "select * from barang");
$data = mysqli_fetch_all($data, MYSQLI_ASSOC);

echo json_encode($data);
```

create.php

```
<?php

$connection = new mysqli("localhost","mibh7264_rafid","Wadadbwdwa%wada&","mibh7264_dbrafid");

$nama = $_POST['nama_barang'];
$jumlah = $_POST['jumlah'];
$harga = $_POST['harga'];
```

```

$result = mysqli_query($connection, "insert into barang set nama_barang='$nama',jumlah='$jumlah',harga='$harga'");
if ($result){
    echo json_encode(['message'=>'data input successfully']);
}else{
    echo json_encode(['message'=>'data failed to input']);
}

```

update.php

```

<?php

$connection = new mysqli("localhost","mibh7264_rafid","Wadadbwdwa%wada%", "mibh7264_dbrafid");

$nama = $_POST['nama_barang'];
$jumlah = $_POST['jumlah'];
$harga = $_POST['harga'];
$id = $_POST['id'];

$result = mysqli_query($connection,"update barang set nama_barang='$nama',jumlah='$jumlah',harga='$harga' where id='$id'");
if ($result){
    echo json_encode(['message'=>'data edit successfully']);
}else{
    echo json_encode(['message'=>'data failed to update']);
}
?>

```

delete.php

```

<?php

$connection = new mysqli("localhost","mibh7264_rafid","Wadadbwdwa%wada%", "mibh7264_dbrafid");

$id = $_POST['id'];

```

```

$result = mysqli_query($connection, "delete from barang where id=".$id);
if ($result){
    echo json_encode(['message'=>'data delete successfully']);
}else{
    echo json_encode(['message'=>'data failed to delete']);
}
?>

```

Selanjutnya kita buat API CRUD untuk menghubungkan ke table user yang disimpan pada folder hosting public_html/api_rafid/user

list_user.php

```

<?php

$connection = new mysqli('localhost','mibh7264_rafid','Wadadbwdwa%wada&','mibh7264_dbrafid');

$data = mysqli_query($connection, "select * from user");
$data = mysqli_fetch_all($data, MYSQLI_ASSOC);

echo json_encode($data);

```

detail_user

```

<?php

$connection = new mysqli('localhost','mibh7264_rafid','Wadadbwdwa%wada&','mibh7264_dbrafid');

$data = mysqli_query($connection, "SELECT * from user where id_user=".$_GET['id_user']);
$data = mysqli_fetch_array($data, MYSQLI_ASSOC);

echo json_encode($data);

```

create_user.php

```

<?php

$connection = new mysqli('localhost','mibh7264_rafid','Wadadbwdwa%wada&','mibh7264_dbrafid');

$username = $_POST['username'];
$password = $_POST['password'];

```

```

$email = $_POST['email'];

$result = mysqli_query($connection, "insert into user set username='$username',password='$password',email='$email'");
if ($result){
    echo json_encode(['message'=>'data input successfully']);
}else{
    echo json_encode(['message'=>'data failed to input']);
}

```

update_user.php

```

<?php

$connection = new mysqli('localhost','mibh7264_rafid','Wadadbwdwa%wada&','mibh7264_dbrafid');
$username = $_POST['username'];
$password = $_POST['password'];
$email = $_POST['email'];
$id_user = $_POST['id_user'];

$result = mysqli_query($connection, "update user set username='$username',password='$password',email='$email' where id_user='$id_user'");
if ($result){
    echo json_encode(['message'=>'data edit successfully']);
}else{
    echo json_encode(['message'=>'data failed to update']);
}
?>

```

delete.php

```

<?php

$connection = new mysqli('localhost','mibh7264_rafid','Wadadbwdwa%wada&','mibh7264_dbrafid');

$id = $_POST['id_user'];

```

```

$result = mysqli_query($connection, "delete from user where id_user=".$id);
if ($result){
    echo json_encode(['message'=>'data delete successfully']);
}else{
    echo json_encode(['message'=>'data failed to delete']);
}
?>

```

1. Buatlah project fultter dengan nama bebas
2. Buka folder **pubspec.yaml** lalu tambahkan liblary di bawah ini untuk mendownload liblary yang dibutuhkan kemudian **pub get** :

flutter_staggered_grid_view: ^0.6.1

http: ^0.13.4

fluttermtoast: ^8.0.9

letakkan seperti ini:

```

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
  flutter_staggered_grid_view: ^0.6.1
  http: ^0.13.4 #to interact with api
  fluttermtoast: ^8.0.9 #to display toast message

```

Lanjut kita menuju folder lib pada project dan ganti main.dart kalian jadi seperti ini untuk halaman login, pada halaman login kita menggunakan username dan password yang ada pada **table data user**

main.dart

apada main.dart ini kita akan membuat untuk halaman login pada aplikasi

```

import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:fluttermtoast/fluttermtoast.dart';
/*fluttermtoast ke dalam proyek Flutter. Paket
fluttermtoast menyediakan fungsionalitas untuk
menampilkan pesan toast (balon informasi singkat) di aplikasi Flutter.

```



```

*/
import 'register.dart';
import 'package:http/http.dart' as http;
import 'dashboard.dart';
import '/barang/baranglist.dart';
import '/user/userlist.dart';
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter',
      theme: ThemeData(
        /*theme: ThemeData() adalah properti yang digunakan untuk mengatur
        tema keseluruhan aplikasi Flutter.
        Dalam properti ini, Anda dapat mengatur berbagai aspek
        tampilan visual aplikasi seperti warna, tipografi, ikon, efek
        bayangan, dan sebagainya.
        */
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
        /* visualDensity: VisualDensity.adaptivePlatformDensity adalah
        properti yang
        digunakan untuk mengatur kepadatan
        visual (visual density) dari widget pada platform yang berbeda
        secara adaptif.*/
      ),
      home: const MyHomePage(),
      routes:<String,WidgetBuilder>{
        /*routes: <String, WidgetBuilder>{} adalah properti yang digunakan
        dalam widget MaterialApp untuk
        mendefinisikan rute (routes) dalam aplikasi Flutter. Properti ini
        memungkinkan Anda untuk menetapkan
        nama rute ke widget builder yang sesuai untuk setiap rute yang
        didefinisikan.
        */
        '/MyHomePage':(BuildContext context)=> new MyHomePage(),
        '/Dashboard':(BuildContext context)=> new Dashboard(),
        '/Baranghome':(BuildContext context)=> new Baranghome(),
        '/userhome':(BuildContext context)=> new userhome(),
      },
    );
  }
}

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

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

class _MyHomePageState extends State<MyHomePage> {
  TextEditingController user = TextEditingController();
  TextEditingController pass = TextEditingController();

```

```

Future login() async {
  /*Future login() async {} adalah deklarasi fungsi login yang
mengembalikan objek Future dan diikuti
dengan kata kunci async. Ini menandakan bahwa
fungsi login adalah sebuah fungsi asinkron yang berpotensi
mengembalikan nilai di masa mendatang.
*/
  var url = Uri.http("mibelajarngoding.online",
'api_rafid/login/login.php', {'q': '{http}'));
  //untuk memanggil api yang telah dibuat
  var response = await http.post(url, body: {
    //menggunakan metode post dari paket http untuk melakukan permintaan
    HTTP POST ke URL yang ditentukan.
    "username": user.text,
    "password": pass.text,
  });
  var data = json.decode(response.body);
  //Anda menggunakan fungsi json.decode() untuk menguraikan (decode) data
  JSON yang diterima dari respons HTTP.
  if (data.toString() == "Success") {
    Fluttertoast.showToast(
      // fluttertoast untuk menampilkan pesan toast (popup sementara) di
      aplikasi Flutter.
      msg: 'Login Successful',
      backgroundColor: Colors.green,
      textColor: Colors.white,
      toastLength: Toast.LENGTH_SHORT,
    );
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => Dashboard(),
      ),
    );
  } else {
    Fluttertoast.showToast(
      backgroundColor: Colors.red,
      textColor: Colors.white,
      msg: 'Username and password invalid',
      toastLength: Toast.LENGTH_SHORT,
    );
  }
}

bool isPasswordVisible = false;
/*deklarasi variabel isPasswordVisible dengan tipe data bool (boolean)
yang digunakan untuk
mengontrol visibilitas password dalam suatu fitur yang melibatkan input
password, seperti
pengaturan password pada formulir.
*/

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: Container(
        height: double.infinity,
        width: double.infinity,
        alignment: Alignment.center,
        decoration: BoxDecoration(
          gradient: LinearGradient(

```

```

        colors: [Colors.teal.shade200, Colors.purple.shade900])),
child: SingleChildScrollView(
  child: Column(
    children: [
      Align(
        alignment: Alignment.topRight,
        child: Container(
          height: 100,
          width: 300,
          decoration: const BoxDecoration(
            gradient:
Colors.teal]],
                boxShadow: [
                  BoxShadow(
                    //BoxShadow adalah kelas yang digunakan untuk
membuat efek bayangan (shadow) pada suatu widget di Flutter.
                    blurRadius: 4,
                    spreadRadius: 3,
                    color: Colors.black12)
                ],
                borderRadius: BorderRadius.only(
                  topLeft: Radius.circular(200),
                  bottomRight: Radius.circular(200))),
            child: Padding(
              padding: const EdgeInsets.only(bottom: 35, left:
100),
                child: Row(
                  crossAxisAlignment: CrossAxisAlignment.end,
                  children: [
                    const Text(
                      'Login',
                      style: TextStyle(
                        fontSize: 30,
                        fontWeight: FontWeight.bold,
                        color: Colors.white,
                        shadows: [
                          Shadow(
                            color: Colors.black45,
                            offset: Offset(1, 1),
                            blurRadius: 5)
                        ],
                      ),
                    ),
                  ],
                ),
              ),
            ),
          ),
        ),
      const SizedBox(
        height: 40,
      ),
      Padding(
        padding: const EdgeInsets.symmetric(horizontal: 30)
          .copyWith(bottom: 10),
        child: TextField(
          controller: user,
          style: const TextStyle(color: Colors.white, fontSize:
14.5),
          decoration: InputDecoration(
            prefixIconConstraints:

```

```

        const BoxConstraints(minWidth: 45),
        prefixIcon: const Icon(
          Icons.person,
          color: Colors.white70,
          size: 22,
        ),
        border: InputBorder.none,
        hintText: 'Enter Username',
        hintStyle: const TextStyle(
          color: Colors.white60, fontSize: 14.5),
        enabledBorder: OutlineInputBorder(
          borderRadius:
BorderRadius.circular(100).copyWith(
            bottomRight: const Radius.circular(0)),
            borderSide:
            const BorderSide(color: Colors.white38)),
        focusedBorder: OutlineInputBorder(
          borderRadius:
BorderRadius.circular(100).copyWith(
            bottomRight: const Radius.circular(0)),
            borderSide:
            const BorderSide(color: Colors.white70))),
      ),
    ),
    const SizedBox(
      height: 20,
    ),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 30)
        .copyWith(bottom: 10),
      child: TextField(
        controller: pass,
        style: const TextStyle(color: Colors.white, fontSize:
14.5),
        obscureText: isPasswordVisible ? false : true,
        decoration: InputDecoration(
          prefixIconConstraints:
            const BoxConstraints(minWidth: 45),
          prefixIcon: const Icon(
            Icons.lock,
            color: Colors.white70,
            size: 22,
          ),
          suffixIconConstraints:
            const BoxConstraints(minWidth: 45, maxWidth: 46),
          suffixIcon: GestureDetector(
            onTap: () {
              setState(() {
                isPasswordVisible = !isPasswordVisible;
              });
            },
            child: Icon(
              isPasswordVisible
                ? Icons.visibility
                : Icons.visibility_off,
              color: Colors.white70,
              size: 22,
            ),
          ),
        ),
        border: InputBorder.none,
        hintText: 'Enter Password',

```

```

        hintStyle: const TextStyle(
          color: Colors.white60, fontSize: 14.5),
        enabledBorder: OutlineInputBorder(
          borderRadius:
BorderRadius.circular(100).copyWith(
            bottomRight: const Radius.circular(0)),
          borderSide:
            const BorderSide(color: Colors.white38)),
        focusedBorder: OutlineInputBorder(
          borderRadius:
BorderRadius.circular(100).copyWith(
            bottomRight: const Radius.circular(0)),
            borderSide:
              const BorderSide(color: Colors.white70))),
    ),
  ),
  const SizedBox(
    height: 50,
  ),
  GestureDetector(
    onTap: () {
      login();
    },
    child: Container(
      height: 53,
      width: double.infinity,
      margin: const EdgeInsets.symmetric(horizontal: 30),
      alignment: Alignment.center,
      decoration: BoxDecoration(
        boxShadow: [
          BoxShadow(
            blurRadius: 4,
            color: Colors.black12.withOpacity(.2),
            offset: const Offset(2, 2))
        ],
        borderRadius: BorderRadius.circular(100)
          .copyWith(bottomRight: const
Radius.circular(0)),
        gradient: LinearGradient(colors: [
          Colors.purple.shade600,
          Colors.teal.shade400
        ])),
      child: Text('Login',
        style: TextStyle(
          color: Colors.white.withOpacity(.8),
          fontSize: 15,
          fontWeight: FontWeight.bold)),
    ),
  ),
  const SizedBox(
    height: 50,
  ),
  const Text('Don\'t have an account?',
    style: TextStyle(color: Colors.white70, fontSize: 13)),
  const SizedBox(
    height: 20,
  ),
  GestureDetector(
    onTap: () {
      Navigator.push(
        context,

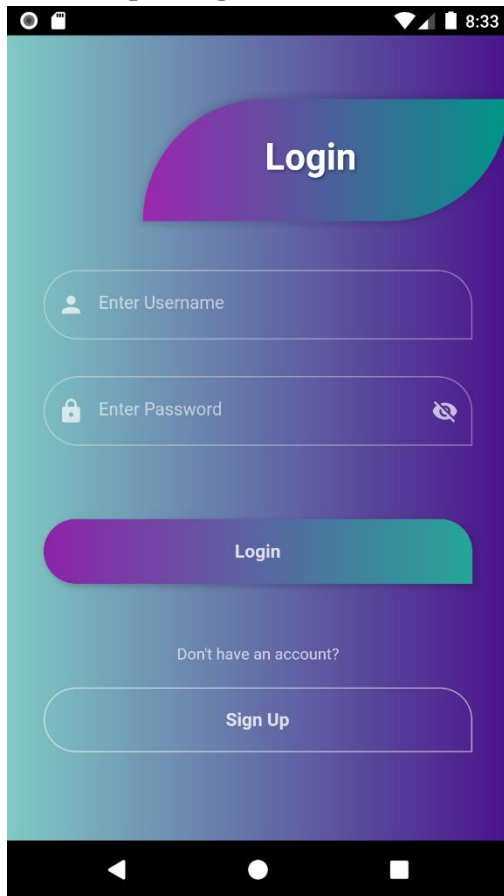
```

```

        MaterialPageRoute(builder: (context) => const
Register()),
    );
  },
  child: Container(
    height: 53,
    width: double.infinity,
    margin: const EdgeInsets.symmetric(horizontal: 30),
    alignment: Alignment.center,
    decoration: BoxDecoration(
      border: Border.all(color: Colors.white60),
      borderRadius: BorderRadius.circular(100)
        .copyWith(bottomRight: const Radius.circular(0)),
    ),
    child: Text('Sign Up',
      style: TextStyle(
        color: Colors.white.withOpacity(.8),
        fontSize: 15,
        fontWeight: FontWeight.bold)),
  ),
),
const SizedBox(
  height: 20,
),
],
),
),
),
),
),
);
}
}

```

ini output login



lanjut kalian buat file register yang berfungsi untuk membuat form registrasi jika belum punya akun

register.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:http/http.dart' as http;
import 'dashboard.dart';
import 'main.dart';

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

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

class _RegisterState extends State<Register> {
  TextEditingController user = TextEditingController();
  TextEditingController pass = TextEditingController();
  TextEditingController email = TextEditingController();
```

```

Future register() async {
  var url = Uri.http("mibelajarngoding.online",
    'api_rafid/login/register.php', {'q': '{http}'});
  var response = await http.post(url, body: {
    "username": user.text.toString(),
    "password": pass.text.toString(),
    "email": email.text.toString(),
  });
  var data = json.decode(response.body);
  if (data == "Error") {
    Fluttertoast.showToast(
      backgroundColor: Colors.orange,
      textColor: Colors.white,
      msg: 'User already exit!',
      toastLength: Toast.LENGTH_SHORT,
    );
  } else {
    Fluttertoast.showToast(
      backgroundColor: Colors.green,
      textColor: Colors.white,
      msg: 'Registration Successful',
      toastLength: Toast.LENGTH_SHORT,
    );
    Navigator.push(context,
      MaterialPageRoute(
        builder: (context) => Dashboard(),
      ),
    );
  }
}

bool isPasswordVisible = false;
@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: Container(
        height: double.infinity,
        width: double.infinity,
        alignment: Alignment.center,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.teal.shade200, Colors.purple.shade900]),
        ),
        child: SingleChildScrollView(
          child: Column(
            children: [
              Align(
                alignment: Alignment.topRight,
                child: Container(
                  height: 100,
                  width: 300,
                  decoration: const BoxDecoration(
                    gradient:
                      LinearGradient(colors: [Colors.purple,
Colors.teal]),
                    boxShadow: [
                      BoxShadow(
                        blurRadius: 4,
                        spreadRadius: 3,
                        color: Colors.black12)
                    ],
                    borderRadius: BorderRadius.only(

```



```

        topLeft: Radius.circular(200),
        bottomRight: Radius.circular(200))),
child: Padding(
  padding: const EdgeInsets.only(bottom: 35, left:
100),

  child: Row(
    crossAxisAlignment: CrossAxisAlignment.end,
    children: [
      const Text(
        'Register',
        style: TextStyle(
          fontSize: 30,
          fontWeight: FontWeight.bold,
          color: Colors.white,
          shadows: [
            Shadow(
              color: Colors.black45,
              offset: Offset(1, 1),
              blurRadius: 5)
          ],
        ),
      ),
    ],
  ),
),
const SizedBox(
  height: 40,
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 30)
    .copyWith(bottom: 10),
  child: TextField(
    controller: user,
    style: const TextStyle(color: Colors.white, fontSize:
14.5),

    decoration: InputDecoration(
      prefixIconConstraints:
        const BoxConstraints(minWidth: 45),
      prefixIcon: const Icon(
        Icons.person,
        color: Colors.white70,
        size: 22,
      ),
      border: InputBorder.none,
      hintText: 'Enter Username',
      hintStyle: const TextStyle(
        color: Colors.white60, fontSize: 14.5),
      enabledBorder: OutlineInputBorder(
        borderRadius:
BorderRadius.circular(100).copyWith(
          bottomRight: const Radius.circular(0)),
        borderSide:
          const BorderSide(color: Colors.white38)),
      focusedBorder: OutlineInputBorder(
        borderRadius:
BorderRadius.circular(100).copyWith(
          bottomRight: const Radius.circular(0)),
        borderSide:
          const BorderSide(color: Colors.white70))),
    ),

```

```

    ),
    const SizedBox(
      height: 20,
    ),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 30)
        .copyWith(bottom: 10),
      child: TextField(
        controller: pass,
        style: const TextStyle(color: Colors.white, fontSize:
14.5),
        obscureText: isPasswordVisible ? false : true,
        decoration: InputDecoration(
          prefixIconConstraints:
            const BoxConstraints(minWidth: 45),
          prefixIcon: const Icon(
            Icons.lock,
            color: Colors.white70,
            size: 22,
          ),
          suffixIconConstraints:
            const BoxConstraints(minWidth: 45, maxWidth: 46),
          suffixIcon: GestureDetector(
            onTap: () {
              setState(() {
                isPasswordVisible = !isPasswordVisible;
              });
            },
            child: Icon(
              isPasswordVisible
                ? Icons.visibility
                : Icons.visibility_off,
              color: Colors.white70,
              size: 22,
            ),
          ),
          border: InputBorder.none,
          hintText: 'Enter Password',
          hintStyle: const TextStyle(
            color: Colors.white60, fontSize: 14.5),
          enabledBorder: OutlineInputBorder(
            borderRadius:
BorderRadius.circular(100).copyWith(
              bottomRight: const Radius.circular(0)),
            borderSide:
              const BorderSide(color: Colors.white38),
          focusedBorder: OutlineInputBorder(
            borderRadius:
BorderRadius.circular(100).copyWith(
              bottomRight: const Radius.circular(0),
              borderSide:
                const BorderSide(color: Colors.white70))),
        ),
      ),
    const SizedBox(
      height: 20,
    ),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 30)
        .copyWith(bottom: 10),
      child: TextField(

```

```

        controller: email,
        style: const TextStyle(color: Colors.white, fontSize:
14.5),

        decoration: InputDecoration(
          prefixIconConstraints:
            const BoxConstraints(minWidth: 45),
          prefixIcon: const Icon(
            Icons.mail_outline_outlined,
            color: Colors.white70,
            size: 22,
          ),
          border: InputBorder.none,
          hintText: 'Enter Email',
          hintStyle: const TextStyle(
            color: Colors.white60, fontSize: 14.5),
          enabledBorder: OutlineInputBorder(
            borderRadius:
BorderRadius.circular(100).copyWith(
              bottomRight: const Radius.circular(0)),
            borderSide:
              const BorderSide(color: Colors.white38)),
          focusedBorder: OutlineInputBorder(
            borderRadius:
BorderRadius.circular(100).copyWith(
              bottomRight: const Radius.circular(0)),
            borderSide:
              const BorderSide(color: Colors.white70))),
      ),
    ),
    const SizedBox(
      height: 50,
    ),
    GestureDetector(
      onTap: () {
        register();
      },
      child: Container(
        height: 53,
        width: double.infinity,
        margin: const EdgeInsets.symmetric(horizontal: 30),
        alignment: Alignment.center,
        decoration: BoxDecoration(
          boxShadow: [
            BoxShadow(
              blurRadius: 4,
              color: Colors.black12.withOpacity(.2),
              offset: const Offset(2, 2))
          ],
          borderRadius: BorderRadius.circular(100)
            .copyWith(bottomRight: const
Radius.circular(0)),
          gradient: LinearGradient(colors: [
            Colors.purple.shade600,
            Colors.teal.shade400
          ])),
        child: Text('Signup',
          style: TextStyle(
            color: Colors.white.withOpacity(.8),
            fontSize: 15,
            fontWeight: FontWeight.bold)),
      ),
    ),

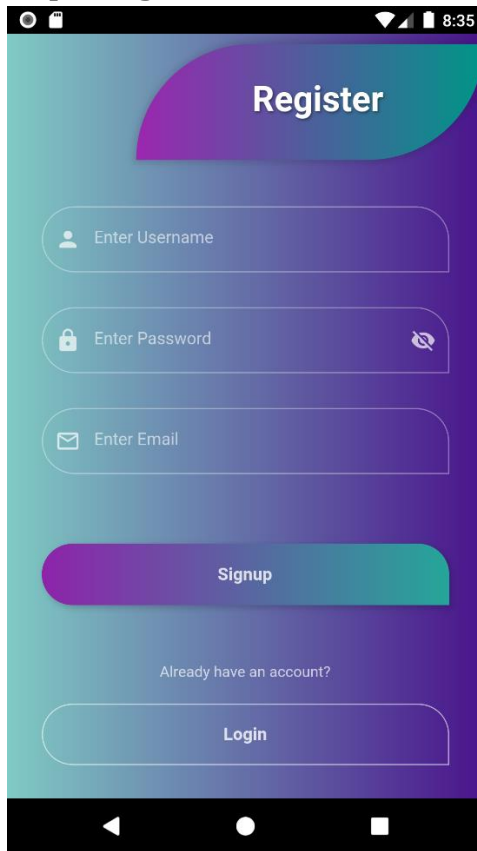
```

```

    ),
    const SizedBox(
      height: 50,
    ),
    const Text('Already have an account?',
      style: TextStyle(color: Colors.white70, fontSize: 13)),
    const SizedBox(
      height: 20,
    ),
    GestureDetector(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => const
MyHomePage()),
        );
      },
    ),
    child: Container(
      height: 53,
      width: double.infinity,
      margin: const EdgeInsets.symmetric(horizontal: 30),
      alignment: Alignment.center,
      decoration: BoxDecoration(
        border: Border.all(color: Colors.white60),
        borderRadius: BorderRadius.circular(100)
          .copyWith(bottomRight: const Radius.circular(0)),
      ),
    ),
    child: Text('Login',
      style: TextStyle(
        color: Colors.white.withOpacity(.8),
        fontSize: 15,
        fontWeight: FontWeight.bold)),
    ),
  ),
  const SizedBox(
    height: 20,
  ),
],
),
),
),
),
),
),
);
}
}

```

Ouput register



Lanjut kita buat file dashboard.dart pada folder lib untuk menjadikannya halaman utama pada aplikasi ini

Dashboard.dart

```
import 'package:flutter/material.dart';
import '/barang/baranglist.dart';
import '/user/userlist.dart';

void main() {
  runApp(MaterialApp(
    title: "Home",
    //menghilangkan debug banner
    debugShowCheckedModeBanner: false,
    home: Dashboard(),
  ));
}

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

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

class _HomePageState extends State<Dashboard> {
```

```

final GlobalKey<ScaffoldState> _key = GlobalKey();
// Create a key
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      centerTitle: true,
      title: const Text(
        'Home',
      ),
      backgroundColor: const Color(0xff764abc),
    ),
    drawer: Drawer(
      child: ListView(
        // Important: Remove any padding from the ListView.
        padding: EdgeInsets.zero,
        children: [
          DrawerHeader(
            decoration: BoxDecoration(
              color: Colors.deepPurple,
              image: DecorationImage(
                fit: BoxFit.fill,
                image:
NetworkImage('https://st2.depositphotos.com/12678588/42606/v/450/depositpho
tos_426065770-stock-illustration-messy-home-office-business-man.jpg'),
              ),
            ),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.end,
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                'Welcome',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 25,
                ),
              ),
            ],
          ),
        ],
      ),
    ),
    ListTile(
      leading: Icon(
        Icons.home,
      ),
      title: const Text('Home'),
      onTap: () {
        Navigator.pop(context);
      },
    ),
    ListTile(
      leading: Icon(
        Icons.file_copy,
      ),
      title: const Text('Data Barang'),
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => Baranghome()),
        );
      },
    ),
  );
}

```

```

    },
  ),
  ListTile(
    leading: Icon(
      Icons.file_present,
    ),
    title: const Text('Data user'),
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => userhome()),
      );
    },
  ),
  ListTile(
    leading: Icon(
      Icons.logout,
    ),
    title: const Text('Logout'),
    onTap: () {
      Navigator.pushReplacementNamed(context, '/MyHomePage');
    },
  ),
],
),
),
body: Column(
  children: [
    Card(
      // Set the shape of the card using a rounded rectangle border
      // with a 8 pixel radius
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(8),
      ),
      // Set the clip behavior of the card
      clipBehavior: Clip.antiAliasWithSaveLayer,
      // Define the child widgets of the card
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          // Display an image at the top of the card that fills the
          // width of the card and has a height of 160 pixels
          Image.network(
            'https://blog.payrollbozz.com/wp-
            content/uploads/sites/28/2021/09/cara-buat-company-profile-570x320.jpg',
            height: 160,
            width: double.infinity,
            fit: BoxFit.cover,
          ),
          // Add a container with padding that contains the card's
          // title, text, and buttons
          Container(
            padding: const EdgeInsets.fromLTRB(15, 15, 15, 0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: <Widget>[
                // Display the card's title using a font size of 24
                // and a dark grey color
                Text(
                  "Rapid Office",
                  style: TextStyle(

```

[illegible]


```

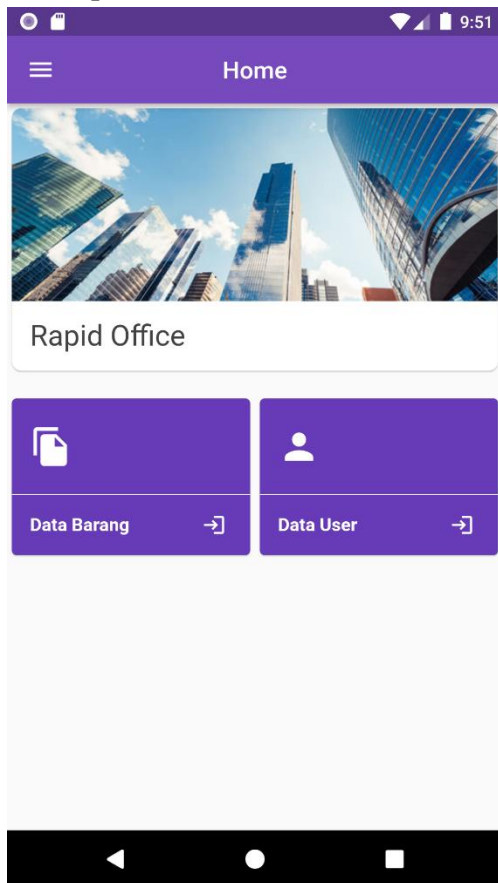
        width: 15,
        height: 50,
      ),
      const Text(
        "Data Barang",
        style: TextStyle(
          color: Colors.white,
          fontSize: 15.0,
          fontWeight: FontWeight.bold),
      ),
      const Spacer(),
      Icon(
        Icons.login,
        color: Colors.white,
        size: 20,
      ),
      SizedBox(
        width: 20,
      ),
    ],
  ),
),
),
),
),
Expanded(
  child: GestureDetector(
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => userhome(),
        ),
      );
    },
  ),
  child: Card(
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(4),
    ),
    //color: Colors.teal[800],
    color: Colors.deepPurple,
    clipBehavior: Clip.antiAliasWithSaveLayer,

    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: <Widget>[
        Container(
          height: 80,
          padding: const EdgeInsets.all(15),
          child: const Icon(Icons.person,
            size: 35, color: Colors.white),
        ),
        const Divider(
          color: Colors.white,
          thickness: 0.5,
          height: 0,
        ),
        Row(
          children: <Widget>[

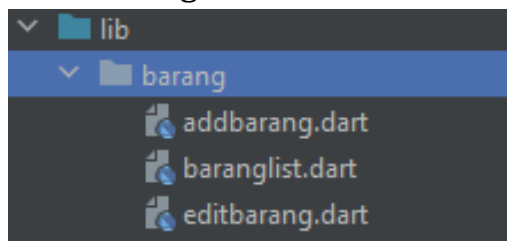
```

```
        Container(  
          width: 15,  
          height: 50,  
        ),  
        const Text(  
          "Data User",  
          style: TextStyle(  
            color: Colors.white,  
            fontSize: 15.0,  
            fontWeight: FontWeight.bold),  
        ),  
  
        const Spacer(),  
        Icon(  
          Icons.login,  
          color: Colors.white,  
          size: 20,  
        ),  
        SizedBox(  
          width: 20,  
        ),  
      ],  
    ),  
  ],  
),  
);  
}
```

ini output dashboard



3. Kemudian kalian buat folder **barang** pada lib untuk menyimpan file data barang



Pertama kita buat file baranglist.dart yang berfungsi untuk menampilkan list barang yang ada

baranglist.dart

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import
'package:flutter_staggered_grid_view/flutter_staggered_grid_view.dart';
import 'package:flutter/material.dart';
import '/barang/editbarang.dart';
import '/barang/addbarang.dart';
import '/user/userlist.dart';
import '/dashboard.dart';

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

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

class _HomePageState extends State<Baranghome> {
  final GlobalKey<ScaffoldState> _key = GlobalKey();
  // Create a key
  List _get = [];
  final _lightColors = [
    Colors.teal.shade100,
    Colors.blueGrey.shade100,
    Colors.redAccent.shade100,
    Colors.teal.shade100,
    Colors.lime.shade100,
    Colors.orange.shade300,
    Colors.pinkAccent.shade100,
    Colors.tealAccent.shade100,
  ];
  @override
  void initState() {
    super.initState();
    _getData();
  }
  Future<void> _getData() async {
    try {
      final response = await http.get(Uri.parse(
        "http://mibelajarngoding.online/api_rafid/barang/list.php",
      ));
      if (response.statusCode == 200) {
        final data = jsonDecode(response.body);
        setState(() {
          _get = data;
        });
      }
    } catch (e) {
      print(e);
    }
  }
  Future<void> _refreshData() async {
    await _getData();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
```

```

        centerTitle: true,
        title: const Text(
          'Data Barang',
        ),
        backgroundColor: const Color(0xff764abc),
      ),
      drawer: Drawer(
        child: ListView(
          // Important: Remove any padding from the ListView.
          padding: EdgeInsets.zero,
          children: [
            DrawerHeader(
              decoration: BoxDecoration(
                color: Colors.deepPurple,
                image: DecorationImage(
                  fit: BoxFit.fill,
                  image:
NetworkImage('https://st2.depositphotos.com/12678588/42606/v/450/depositpho
tos_426065770-stock-illustration-messy-home-office-business-man.jpg'),
                ),
              ),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.end,
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  'Welcome',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 25,
                  ),
                ),
              ],
            ),
          ],
        ),
      ),

      ListTile(
        leading: Icon(
          Icons.home,
        ),
        title: const Text('Home'),
        onTap: () {
          Navigator.push(context,
            MaterialPageRoute(builder: (context) => Dashboard()),);
        },
      ),
      ListTile(
        leading: Icon(
          Icons.file_copy,
        ),
        title: const Text('Data Barang'),
        onTap: () {
          Navigator.pop(context);
        },
      ),
      ListTile(
        leading: Icon(
          Icons.file_present,
        ),
        title: const Text('Data user'),
        onTap: () {

```

```

        Navigator.push(context,
            MaterialPageRoute(builder: (context) => userhome()),);
    },
),
ListTile(
    leading: Icon(
        Icons.logout,
    ),
    title: const Text('Logout'),
    onTap: () {
        Navigator.pushReplacementNamed(context, '/MyHomePage');
    },
),
],
),
),
body: RefreshIndicator(
    onRefresh: _refreshData,
    child: _get.length != 0
        ? MasonryGridView.count(
            crossAxisCount: 1,
            itemCount: _get.length,
            itemBuilder: (context, index) {
                final colorIndex = index % _lightColors.length;
                final color = _lightColors[colorIndex];
                return GestureDetector(
                    onTap: () {
                        Navigator.push(
                            context,
                            MaterialPageRoute(
                                builder: (context) => Edit(id: _get[index]['id'],),
                            ),
                        );
                    },
                ),
                child: Card(
                    child: Container(
                        height: 100.0,
                        margin: const EdgeInsets.all(10.0),
                        decoration: BoxDecoration(
                            borderRadius: BorderRadius.circular(10.0),
                            gradient: LinearGradient( colors: [color,
color.withOpacity(0.8)],
                                begin: Alignment.topLeft,
                                end: Alignment.bottomRight,
                                tileMode: TileMode.clamp,
                            ),
                        ),
                    child: Row(
                        crossAxisAlignment: CrossAxisAlignment.center,
                        children: <Widget>[
                            const Padding(padding: EdgeInsets.only(left: 10.0, right:
10.0),
                                ),
                            Expanded(child: Column(
                                mainAxisAlignment: MainAxisAlignment.center,
                                crossAxisAlignment: CrossAxisAlignment.start,
                                children: [
                                    SizedBox(height: 10),
                                    Text(

```

```

        "Nama      : " + '${_get[index]['nama_barang']}',
        style: TextStyle(
          color: Colors.black,
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ),
      ),
      Text(
        "jumlah    : " +
          '${_get[index]['jumlah']}',
        style: TextStyle(
          color: Colors.black,
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ),
      ),
      Text(
        "harga      : " +
          '${_get[index]['harga']}',
        style: TextStyle(
          color: Colors.black,
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ),
      ),
    ],
  ),
),
);
},
)
: Center(
  child: Text(
    "Data Kosong",
    style: TextStyle(
      color: Colors.white,
      fontSize: 20,
      fontWeight: FontWeight.bold,
    ),
  ),
),
),
),
floatingActionButton: FloatingActionButton(
  backgroundColor: Colors.black,
  child: Icon(Icons.add),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => Add()),
    );
  },
),
);
}
}

```

ini ouput baranglist.dart



- lanjut buat file addbarang.dart pada folder barang yang berfungsi untuk menambahkan barang pada list

addbarang.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
class Add extends StatefulWidget {
  const Add({Key? key}) : super(key: key);
  @override
  State<Add> createState() => _AddState();
}
class _AddState extends State<Add> {
  final _formKey = GlobalKey<FormState>();
  //inisialize field
  var nama = TextEditingController();
  var jumlah = TextEditingController();
  var harga = TextEditingController();
  @override
  Future _onSubmit() async {
    try {
      return await http.post(
        Uri.parse("http://mibelajarngoding.online/api_rafid/barang/create.php"),
        body: {
```



```

        fontWeight: FontWeight.bold,
        fontSize: 16,
      ),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Nama Barang Belum DiInput!';
        }
        return null;
      },
    ),
    SizedBox(height: 20),
    Text(
      'Jumlah',
      style: TextStyle(
        color: Colors.black,
        fontSize: 16,
        fontWeight: FontWeight.bold,
      ),
    ),
    SizedBox(height: 5),
    TextFormField(
      controller: jumlah,
      decoration: InputDecoration(
        hintText: 'Input Jumlah Barang',
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15.0),
        ),
        fillColor: Colors.white,
        filled: true,
        prefixIcon: Icon(
          Icons.summarize_outlined,
          color: Colors.black,
          size: 22,)),
      style: const TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 16,
      ),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Jumlah Barang Belum DiInput!';
        }
        return null;
      },
    ),
    SizedBox(height: 20),
    Text(
      'Harga',
      style: TextStyle(
        color: Colors.black,
        fontSize: 16,
        fontWeight: FontWeight.bold,
      ),
    ),
    SizedBox(height: 5),
    TextFormField(
      controller: harga,
      decoration: InputDecoration(
        hintText: 'Input Harga Barang',
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15.0),
        ),
      ),
    ),
  ),
),

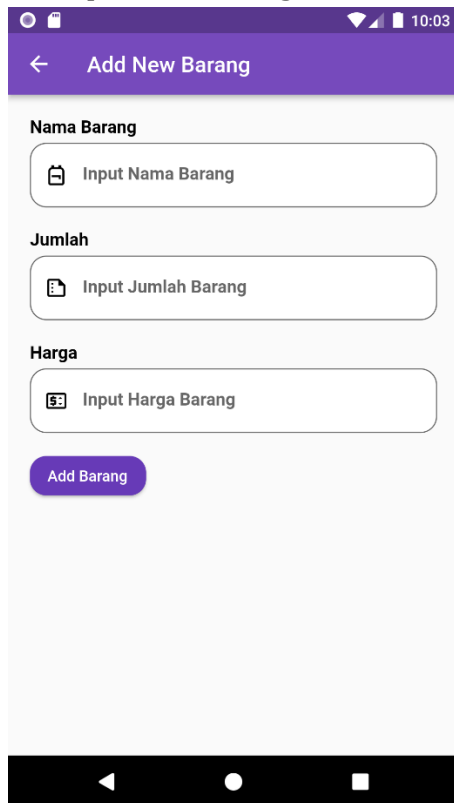
```

```

        fillColor: Colors.white,
        filled: true,
        prefixIcon: Icon(
            Icons.price_change_outlined,
            color: Colors.black,
            size: 22,)),
        style: const TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 16,
        ),
        validator: (value) {
            if (value!.isEmpty) {
                return 'Harga Barang Belum DiInput!';
            }
            return null;
        },
    ),
    SizedBox(height: 15),
    ElevatedButton(
        style: ElevatedButton.styleFrom(
            backgroundColor: Colors.deepPurple, // Ganti dengan
warna yang diinginkan
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(15),
            ),
        ),
        child: Text(
            "Add Barang",
            style: TextStyle(color: Colors.white),
        ),
        onPressed: () {
            //validate
            if (_formKey.currentState!.validate()) {
                //send data to database with this method
                _onSubmit();
            }
        },
    ),
),
),
),
),
),
),
);
}
}

```

ini output add barang



- lanjut kalian buat file editbarang.dart pada folder barang ,pada file ini kita bisa mengedit data barang dan juga menghapus data barang

editbarang.dart

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

class Edit extends StatefulWidget {
  Edit({required this.id});
  String id;

  @override
  State<Edit> createState() => _EditState();
}

class _EditState extends State<Edit> {
  final _formKey = GlobalKey<FormState>();
  //inisialize field
  var nama = TextEditingController();
  var jumlah = TextEditingController();
  var harga = TextEditingController();
  @override
  void initState() {
    super.initState();
    //in first time, this method will be executed
    _getData();
  }
  //Http to get detail data
```

```

Future _getData() async {
  try {
    final response = await http.get(Uri.parse(
      //you have to take the ip address of your computer.
      //because using localhost will cause an error
      //get detail data with id

"http://mibelajarngoding.online/api_rafid/barang/detail.php?id='${widget.id}'"));
    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      setState(() {
        nama = TextEditingController(text: data['nama_barang']);
        jumlah = TextEditingController(text: data['jumlah']);
        harga = TextEditingController(text: data['harga']);
      });
    }
  } catch (e) {
    print(e);
  }
}

Future _onUpdate(context) async {
  try {
    return await http.post(

Uri.parse("http://mibelajarngoding.online/api_rafid/barang/update.php")

    ,
    body: {
      "id": widget.id,
      "nama_barang": nama.text,
      "jumlah": jumlah.text,
      "harga": harga.text,
    },
  ).then((value) {
    //print message after insert to database
    //you can improve this message with alert dialog
    var data = jsonDecode(value.body);
    print(data["message"]);
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text("Data updated successfully"),
      ),
    );
    Navigator.of(context)
      .pushNamedAndRemoveUntil('/Baranghome', (Route<dynamic> route)
=>
      false);
  });
} catch (e) {
  print(e);
}
}

Future _onDelete(context) async {
  try {
    return await http.post(

Uri.parse("http://mibelajarngoding.online/api_rafid/barang/delete.php")

```

```

        body: {
          "id": widget.id,
        },
      ).then((value) {
        //print message after insert to database
        //you can improve this message with alert dialog
        var data = jsonDecode(value.body);
        print(data["message"]);
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text("Data Delete successfully"),
          ),
        );
      });
// Remove all existing routes until the home.dart, then rebuildHome.
Navigator.of(context)
  .pushNamedAndRemoveUntil('/Baranghome', (Route<dynamic> route)
=>
    false);
  });
} catch (e) {
  print(e);
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Edit Data Barang"),
      backgroundColor: Color(0xff764abc),
      // ignore: prefer_const_literals_to_create_immutables
      actions: [
        Container(
          padding: EdgeInsets.only(right: 20),
          child: IconButton(
            onPressed: () {
              showDialog(
                context: context,
                builder: (BuildContext context) {
                  //show dialog to confirm delete data
                  return AlertDialog(
                    content: Text('Yakin ini mau dihapus...?'),
                    actions: <Widget>[
                      ElevatedButton(
                        style: ElevatedButton.styleFrom(
                          backgroundColor: Colors.deepPurple, //
Ganti dengan warna yang diinginkan
                        ),
                        child: Icon(Icons.cancel),
                        onPressed: () => Navigator.of(context).pop(),
                      ),
                      ElevatedButton(
                        style: ElevatedButton.styleFrom(
                          backgroundColor: Colors.red, // Ganti
dengan warna yang diinginkan
                        ),
                        child: Icon(Icons.check_circle),
                        onPressed: () => _onDelete(context),
                      ),
                    ],
                  ),
                ),
              );
            }
          ],
        ),
      ],
    ),
  );
}

```

```

        },
    ),
    icon: Icon(Icons.delete)),
  ),
],
),
body: SingleChildScrollView(
  child: Form(
    key: _formKey,
    child: Container(
      padding: EdgeInsets.all(20.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            'Nama Barang',
            style: TextStyle(
              color: Colors.black,
              fontSize: 16,
              fontWeight: FontWeight.bold,
            ),
          ),
          SizedBox(height: 5),
          TextFormField(
            controller: nama,
            decoration: InputDecoration(
              hintText: "Input Nama Barang",
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(15.0),
              ),
              fillColor: Colors.white,
              filled: true,
              prefixIcon: Icon(
                Icons.backpack_outlined,
                color: Colors.black,
                size: 22,)),
            style: const TextStyle(
              fontWeight: FontWeight.bold,
              fontSize: 16,
            ),
            validator: (value) {
              if (value!.isEmpty) {
                return 'Nama Barang Belum Di Input!';
              }
              return null;
            },
          ),
          SizedBox(height: 20),
          Text(
            'Jumlah',
            style: TextStyle(
              color: Colors.black,
              fontSize: 16,
              fontWeight: FontWeight.bold,
            ),
          ),
          SizedBox(height: 5),
          TextFormField(
            controller: jumlah,
            decoration: InputDecoration(

```

```

hintText: 'Input Jumlah Barang',
border: OutlineInputBorder(
borderRadius: BorderRadius.circular(15.0),
),
fillColor: Colors.white,
filled: true,
prefixIcon: Icon(
Icons.summarize_outlined,
color: Colors.black,
size: 22,)),
style: const TextStyle(
fontWeight: FontWeight.bold,
fontSize: 16,
),
validator: (value) {
if (value!.isEmpty) {
return 'Jumlah Barang Belum Di Input!';
}
return null;
},
),
 SizedBox(height: 20),
Text(
'Harga',
style: TextStyle(
color: Colors.black,
fontSize: 16,
fontWeight: FontWeight.bold,
),
),
),
SizedBox(height: 5),
TextFormField(
controller: harga,
decoration: InputDecoration(
hintText: 'Input Harga Barang',
border: OutlineInputBorder(
borderRadius: BorderRadius.circular(15.0),
),
fillColor: Colors.white,
filled: true,
prefixIcon: Icon(
Icons.price_change_outlined,
color: Colors.black,
size: 22,)),
style: const TextStyle(
fontWeight: FontWeight.bold,
fontSize: 16,
),
validator: (value) {
if (value!.isEmpty) {
return 'Harga Barang Belum Di Input!';
}
return null;
},
),
),
SizedBox(height: 15),
ElevatedButton(
style: ElevatedButton.styleFrom(
primary: Colors.deepPurple,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(15),

```



```

    ),
  ),
  child: Text(
    "Edit Data",
    style: TextStyle(color: Colors.white),
  ),
  onPressed: () {
    //validate
    if (_formKey.currentState!.validate()) {
      //send data to database with this method
      _onUpdate(context);
    }
  },
),
],
),
),
),
);
}
}

```

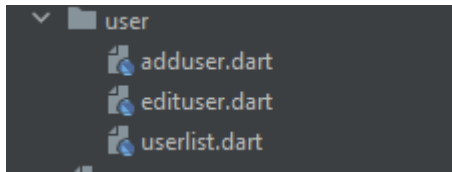
ouput dari editbarang.dart

The screenshot shows a mobile application interface with a purple header bar. The header contains a back arrow, the text "Edit Data Barang", and a trash icon. Below the header, there are three input fields, each with a label and a corresponding icon:

- Nama Barang**: A text input field with a document icon, containing the text "motor".
- Jumlah**: A text input field with a document icon, containing the text "2".
- Harga**: A text input field with a currency icon, containing the text "1000".

Below the input fields, there is a purple button labeled "Edit Data". At the bottom of the screen, there is a black navigation bar with three white icons: a back arrow, a circle, and a square.

4. Kemudian kalian buat folder baru pada lib yaitu buat folder **user** untuk menyimpan file data user



Pertama kita buat file userlist.dart pada folder user yang berfungsi untuk menampilkan list data user pada aplikasi

userlist.dart

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import
'package:flutter_staggered_grid_view/flutter_staggered_grid_view.dart';
import 'package:flutter/material.dart';
import '/user/edituser.dart';
import '/user/adduser.dart';
import '/barang/baranglist.dart';
import '/dashboard.dart';

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

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

class _HomePageState extends State<userhome> {
  final GlobalKey<ScaffoldState> _key = GlobalKey();
  // Create a key
  List _get = [];
  final _lightColors = [
    Colors.teal.shade100,
    Colors.blueGrey.shade100,
    Colors.teal.shade100,
    Colors.lime.shade100,
    Colors.tealAccent.shade100,
  ];
  @override
  void initState() {
    super.initState();
    _getData();
  }
  Future<void> _getData() async {
    try {
      final response = await http.get(Uri.parse(
        "http://mibelajarngoding.online/api_rafid/user/list_user.php",
      ));
      if (response.statusCode == 200) {
        final data = jsonDecode(response.body);
        setState(() {
          _get = data;
        });
      }
    }
  }
}
```

```

    } catch (e) {
      print(e);
    }
  }
  Future<void> _refreshData() async {
    await _getData();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        title: const Text(
          'Data User',
        ),
        backgroundColor: const Color(0xff764abc),
      ),
      drawer: Drawer(
        child: ListView(
          // Important: Remove any padding from the ListView.
          padding: EdgeInsets.zero,
          children: [
            DrawerHeader(
              decoration: BoxDecoration(
                color: Colors.deepPurple,
                image: DecorationImage(
                  fit: BoxFit.fill,
                  image:
NetworkImage('https://st2.depositphotos.com/12678588/42606/v/450/depositpho
tos_426065770-stock-illustration-messy-home-office-business-man.jpg'),
                ),
            ),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.end,
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  'Welcome',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 25,
                  ),
                ),
              ],
            ),
          ],
        ),
      ),

      ListTile(
        leading: Icon(
          Icons.home,
        ),
        title: const Text('Home'),
        onTap: () {
          Navigator.push(context,
            MaterialPageRoute(builder: (context) => Dashboard()),);
        },
      ),
      ListTile(
        leading: Icon(
          Icons.file_copy,

```

```

        ),
        title: const Text('Data Barang'),
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => Baranghome()),
          );
        },
      ),
      ListTile(
        leading: Icon(
          Icons.file_present,
        ),
        title: const Text('Data user'),
        onTap: () {
          Navigator.pop(context);
        },
      ),
      ListTile(
        leading: Icon(
          Icons.logout,
        ),
        title: const Text('Logout'),
        onTap: () {
          Navigator.pushReplacementNamed(context, '/MyHomePage');
        },
      ),
    ],
  ),
),
body: RefreshIndicator(
  onRefresh: _refreshData,
  child: _get.length != 0
    ? MasonryGridView.count(
        crossAxisCount: 1,
        itemCount: _get.length,
        itemBuilder: (context, index) {
          final colorIndex = index % _lightColors.length;
          final color = _lightColors[colorIndex];
          return GestureDetector(
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => Edituser(id:
_get[index]['id_user'],
                ),
              ),
            );
          },
        ),
      ),
    child: Card(
      color: _lightColors[index % _lightColors.length],
      shadowColor: Colors.deepPurple,
      elevation: 3,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
      ),
      child: ListTile(
        leading: CircleAvatar(child: Icon(Icons.person, color:
Colors.white), backgroundColor: Colors.deepPurple, ),

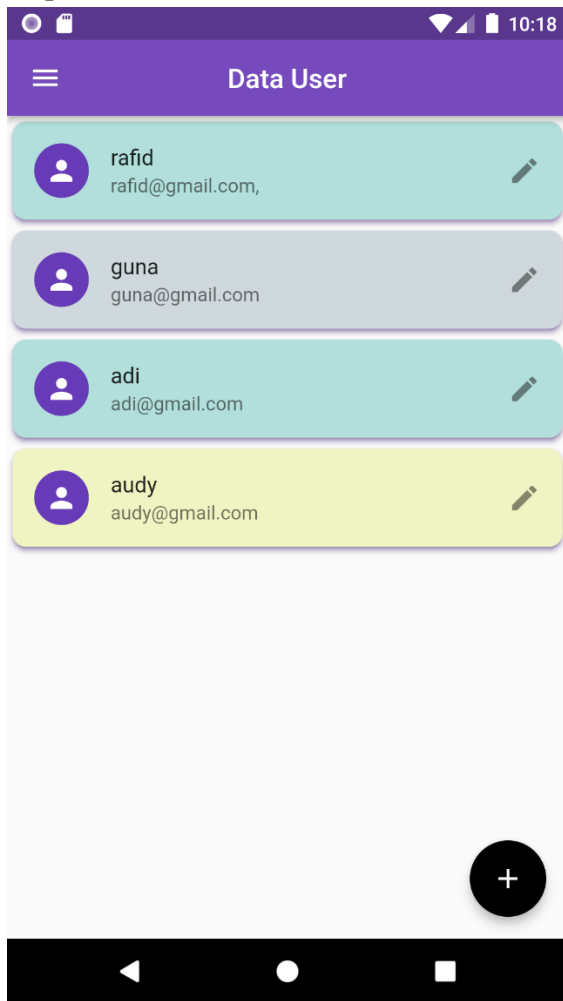
```

```

        title: Text('${_get[index]['username']}'),
        subtitle: Text('${_get[index]['email']}'),
        trailing: Icon(Icons.edit),
      ),
    ),
  );
},
)
: Center(
  child: Text(
    "Data Kosong",
    style: TextStyle(
      color: Colors.white,
      fontSize: 20,
      fontWeight: FontWeight.bold,
    ),
  ),
),
),
),
floatingActionButton: FloatingActionButton(
  backgroundColor: Colors.black,
  child: Icon(Icons.add),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => Add()),
    );
  },
),
);
}
}

```

ouput userlist



- lanjut kita buat file adduser.dart pada folder user, file ini berfungsi untuk menambahkan user baru pada data user yang bisa di gunakan saat login

adduser.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
class Add extends StatefulWidget {
  const Add({Key? key}) : super(key: key);
  @override
  State<Add> createState() => _AddState();
}
class _AddState extends State<Add> {
  final _formKey = GlobalKey<FormState>();
  //inisialize field
  var username = TextEditingController();
  var password = TextEditingController();
  var email = TextEditingController();

  Future _onSubmit() async {
    try {
      return await http.post(
```

```

Uri.parse("http://mibelajarngoding.online/api_rafid/user/create_user.php")
    ,
    body: {
      "username": username.text,
      "password": password.text,
      "email": email.text,
    },
  ).then((value) {
    //print message after insert to database
    //you can improve this message with alert dialog
    var data = jsonDecode(value.body);
    print(data["message"]);
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text("Data Add successfully"),
      ),
    );
    Navigator.of(context)
      .pushNamedAndRemoveUntil('/userhome', (Route<dynamic> route) =>
        false);
  });
} catch (e) {
  print(e);
}
}

bool isPasswordVisible = false;
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Add New User"),
      backgroundColor: Color(0xff764abc),
    ),
    body: SingleChildScrollView(
      child: Form(
        key: _formKey,
        child: Container(
          padding: EdgeInsets.all(20.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                'Username',
                style: TextStyle(
                  color: Colors.black,
                  fontSize: 16,
                  fontWeight: FontWeight.bold,
                ),
              ),
              SizedBox(height: 5),
              TextFormField(
                controller: username,
                decoration: InputDecoration(
                  hintText: "Input Username Anda",
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(15.0),
                  ),
                  fillColor: Colors.white,
                  filled: true,
                  prefixIcon: Icon(

```

```

        Icons.person,
        color: Colors.black,
        size: 22,)),
    style: const TextStyle(
      fontWeight: FontWeight.bold,
      fontSize: 16,
    ),
    validator: (value) {
      if (value!.isEmpty) {
        return 'Username Belum DiInput!';
      }
      return null;
    },
  ),
  SizedBox(height: 20),
  Text(
    'Password',
    style: TextStyle(
      color: Colors.black,
      fontSize: 16,
      fontWeight: FontWeight.bold,
    ),
  ),
  SizedBox(height: 5),
  TextFormField(
    controller: password,
    keyboardType: TextInputType.multiline,
    obscureText: isPasswordVisible ? false : true,
    decoration: InputDecoration(
      hintText: 'Input Password Anda',
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(15.0),
      ),
      fillColor: Colors.white,
      filled: true,
      prefixIcon: Icon(
        Icons.lock,
        color: Colors.black,
        size: 22,
      ),
      suffixIcon: GestureDetector(
        onTap: () {
          setState(() {
            isPasswordVisible = !isPasswordVisible;
          });
        },
        child: Icon(
          isPasswordVisible
            ? Icons.visibility
            : Icons.visibility_off,
          color: Colors.black,
          size: 22,
        ),
      ),
    ),
  ),
  style: const TextStyle(
    fontWeight: FontWeight.bold,
    fontSize: 16,
  ),
  validator: (value) {
    if (value!.isEmpty) {
      return 'Password Belum DiInput!';
    }
  }
),

```



```

    }
    return null;
  },
), SizedBox(height: 20),
Text(
  'Email',
  style: TextStyle(
    color: Colors.black,
    fontSize: 16,
    fontWeight: FontWeight.bold,
  ),
),
),
SizedBox(height: 5),
TextFormField(
  controller: email,
  decoration: InputDecoration(
    hintText: 'Input Email Anda',
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(15.0),
    ),
    fillColor: Colors.white,
    filled: true,
    prefixIcon: Icon(
      Icons.email,
      color: Colors.black,
      size: 22,
    ),
  ),
  style: const TextStyle(
    fontWeight: FontWeight.bold,
    fontSize: 16,
  ),
  validator: (value) {
    if (value!.isEmpty) {
      return 'Email Belum DiInput!';
    }
    return null;
  },
),
),
SizedBox(height: 15),
ElevatedButton(
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.deepPurple,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(15),
    ),
  ),
  child: Text(
    "Add User",
    style: TextStyle(color: Colors.white),
  ),
  onPressed: () {
    //validate
    if (_formKey.currentState!.validate()) {
      //send data to database with this method
      _onSubmit();
    }
  },
),
),
),
),
),

```




ouput add user



10:24

← Add New User


Username

 Input Username Anda

Password

 Input Password Anda 

Email

 Input Email Anda

Add User

- kemudian kita buat edituser.dart pada folder user, di dalam file ini kita bisa mengedit data user dan juga bisa menghapus data user

edituser.dart

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

class Edituser extends StatefulWidget {
  Edituser({required this.id});
  String id;
  @override
  State<Edituser> createState() => _EditState();
}

class _EditState extends State<Edituser> {
  final _formKey = GlobalKey<FormState>();
  //inisialize field
  var username = TextEditingController();
  var password = TextEditingController();
  var email = TextEditingController();
  @override
  void initState() {
    super.initState();
    //in first time, this method will be executed
    _getData();
  }
  //Http to get detail data
  Future _getData() async {
    try {
      final response = await http.get(Uri.parse(
        "http://mibelajarngoding.online/api_rafid/user/detail_user.php?id_user='${w
idget.id}'"));
      // if response successful
      if (response.statusCode == 200) {
        final data = jsonDecode(response.body);
        setState(() {
          username = TextEditingController(text: data['username']);
          password = TextEditingController(text: data['password']);
          email = TextEditingController(text: data['email']);
        });
      }
    } catch (e) {
      print(e);
    }
  }
  Future _onUpdate(context) async {
    try {
      return await http.post(
        Uri.parse("http://mibelajarngoding.online/api_rafid/user/update_user.php"),
        body: {
          "id_user": widget.id,
          "username": username.text,
          "password": password.text,
```

```

        "email": email.text,
      },
    ).then((value) {
      //print message after insert to database
      //you can improve this message with alert dialog
      var data = jsonDecode(value.body);
      print(data["message"]);
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text("Data updated successfully"),
        ),
      );
      Navigator.of(context)
        .pushNamedAndRemoveUntil('/userhome', (Route<dynamic> route) =>
          false);
    });
  } catch (e) {
    print(e);
  }
}

Future _onDelete(context) async {
  try {
    return await http.post(
      Uri.parse("http://mibelajarnngoding.online/api_rafid/user/delete_user.php")
        ,
        body: {
          "id_user": widget.id,
        },
      ).then((value) {
        //print message after insert to database
        //you can improve this message with alert dialog
        var data = jsonDecode(value.body);
        print(data["message"]);
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text("Data Delete successfully"),
          ),
        );
      });
  } catch (e) {
    print(e);
  }
}

// Remove all existing routes until the home.dart, then rebuildHome.
Navigator.of(context)
  .pushNamedAndRemoveUntil('/userhome', (Route<dynamic> route) =>
    false);

bool isPasswordVisible = false;
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Edit Data User"),
      backgroundColor: Color(0xff764abc),
      // ignore: prefer_const_literals_to_create_immutables
      actions: [
        Container(
          padding: EdgeInsets.only(right: 20),
          child: IconButton(

```



```

        style: const TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 16,
        ),
        validator: (value) {
          if (value!.isEmpty) {
            return 'Username Belum DiInput!';
          }
          return null;
        },
      ),
      SizedBox(height: 20),
      Text(
        'Password',
        style: TextStyle(
          color: Colors.black,
          fontSize: 16,
          fontWeight: FontWeight.bold,
        ),
      ),
      SizedBox(height: 5),
      TextFormField(
        controller: password,
        obscureText: isPasswordVisible ? false : true,
        decoration: InputDecoration(
          hintText: 'Input Password Baru',
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(15.0),
          ),
          fillColor: Colors.white,
          filled: true,
          prefixIcon: Icon(
            Icons.lock,
            color: Colors.black,
            size: 22,
          ),
          suffixIcon: GestureDetector(
            onTap: () {
              setState(() {
                isPasswordVisible = !isPasswordVisible;
              });
            },
            child: Icon(
              isPasswordVisible
                ? Icons.visibility
                : Icons.visibility_off,
              color: Colors.black,
              size: 22,
            ),
          ),
        ),
      ),

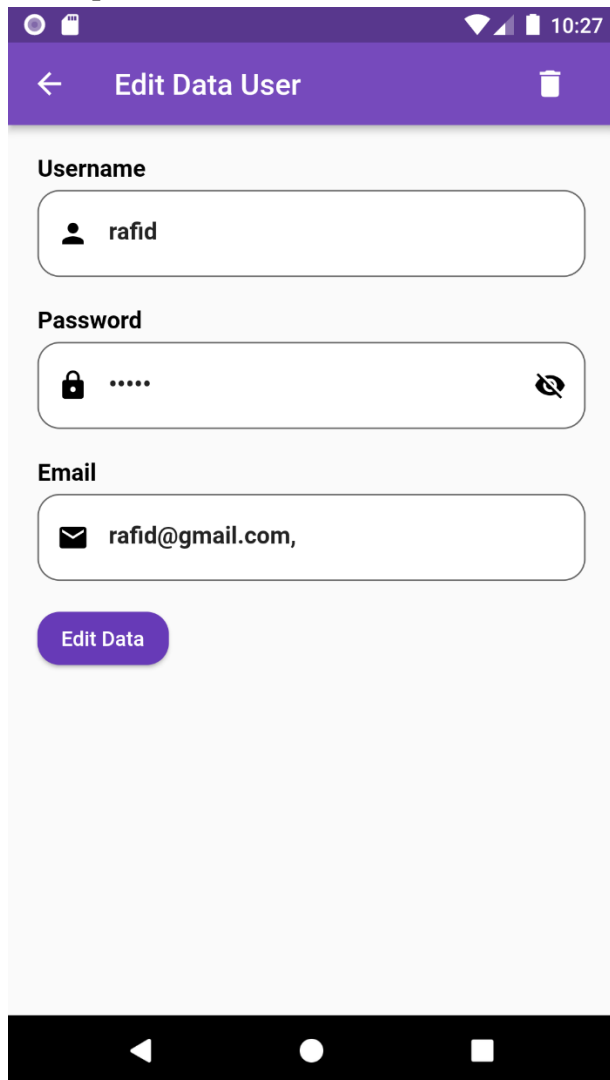
      style: const TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 16,
      ),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Password Belum DiInput!';
        }
        return null;
      },
    ),
  ),
)

```



```
} ;  
}  
}
```

ini ouput edit user



The screenshot shows a mobile application interface with a purple header bar. The header contains a back arrow, the text "Edit Data User", and a trash icon. Below the header, there are three input fields: "Username" with a person icon and the text "rafid", "Password" with a lock icon, dots, and an eye icon, and "Email" with an envelope icon and the text "rafid@gmail.com,". Below the input fields is a purple button labeled "Edit Data". The bottom of the screen shows the Android navigation bar with back, home, and recent apps icons.


Itu adalah penjelasan dari aplikasi yang saya buat yang berupa kodingan dan ouputnya.



Saya akan memberitahu alur dari aplikasi saya

Pertama login

8:33

Login

 Enter Username

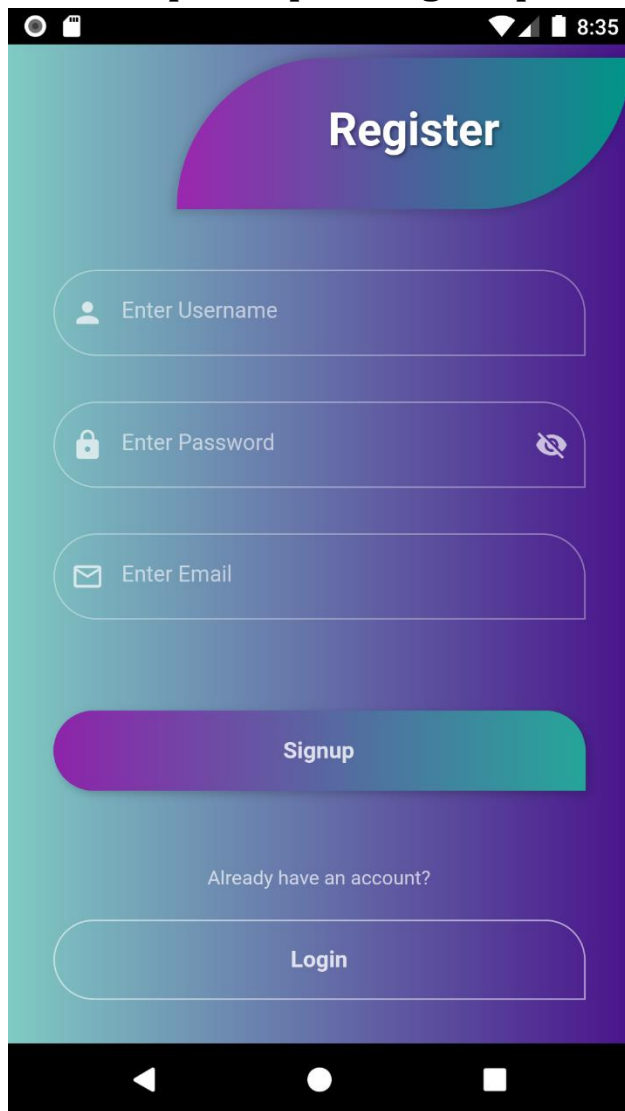
 Enter Password 

Login

Don't have an account?

Sign Up

Jika ingin belum punya akun klik tombol sign up pada halaman login, akan tampil tampilan regis seperti ini



The image shows a mobile application interface for user registration. At the top, there is a status bar with icons for signal, Wi-Fi, and battery, and the time 8:35. Below this, the word "Register" is displayed in a large, white, sans-serif font on a dark purple background. The registration form consists of three input fields: "Enter Username" with a person icon, "Enter Password" with a lock icon and a toggle for password visibility, and "Enter Email" with an envelope icon. Below these fields is a prominent "Signup" button with a purple-to-teal gradient. Underneath the button, the text "Already have an account?" is displayed in a small, light gray font. At the bottom of the form is a "Login" button with a white-to-purple gradient. The entire screen has a dark purple background with a subtle teal-to-purple gradient at the top. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

Register

Enter Username

Enter Password

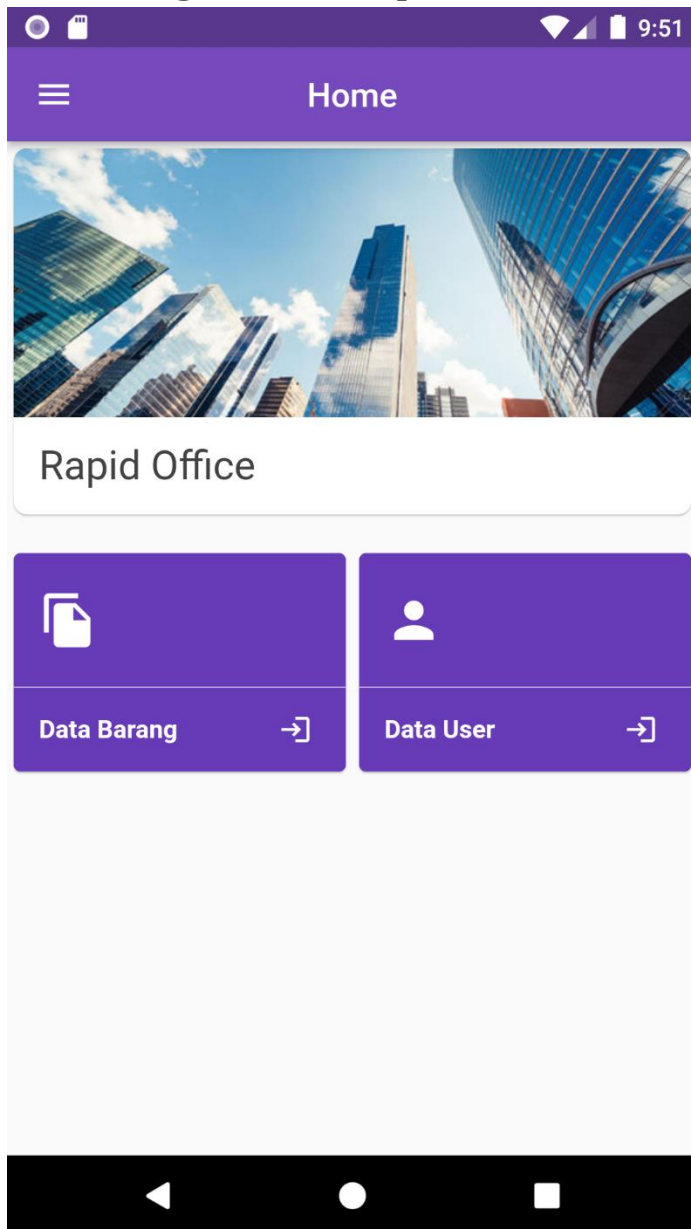
Enter Email

Signup

Already have an account?

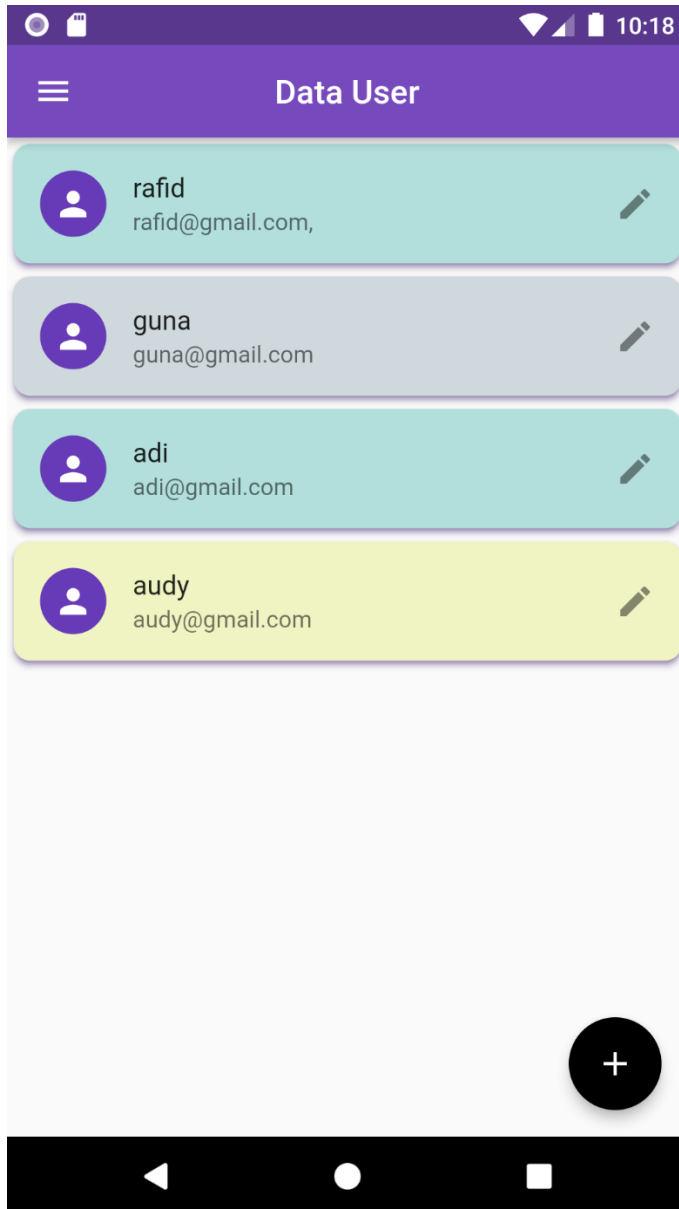
Login

Setelah login akan tampil menu dashboard seperti ini



Kemudia klik data user untuk menuju ke halaman user dan klik data barang untuk menuju ke data barang

Saya beri contoh tampilan data user



Jika ingin menambahkan data kalian klik tombol tambah pada pojok kanan bawah halaman data user dan akan tampil seperti ini

10:24

← Add New User

Username

Input Username Anda

Password

Input Password Anda

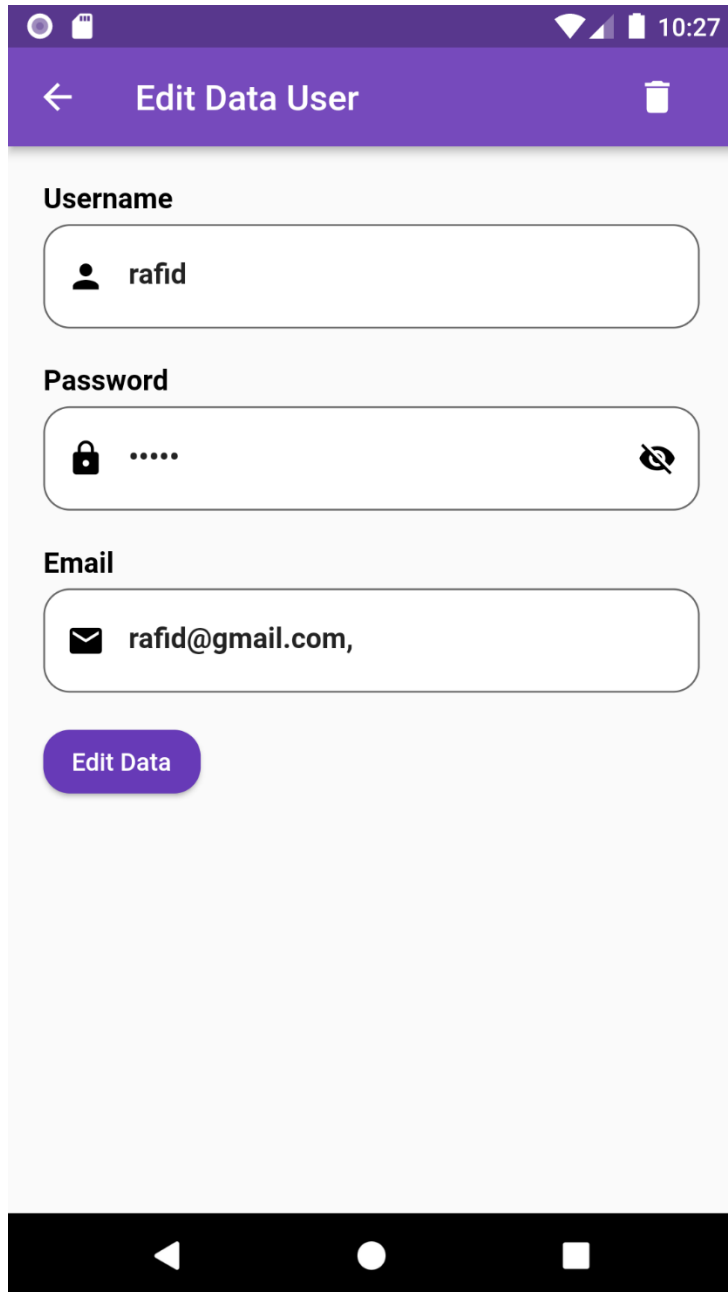
Email

Input Email Anda

Add User

Input datanya lalu klik add user kemudian akan terinput dan kembali ke halaman user

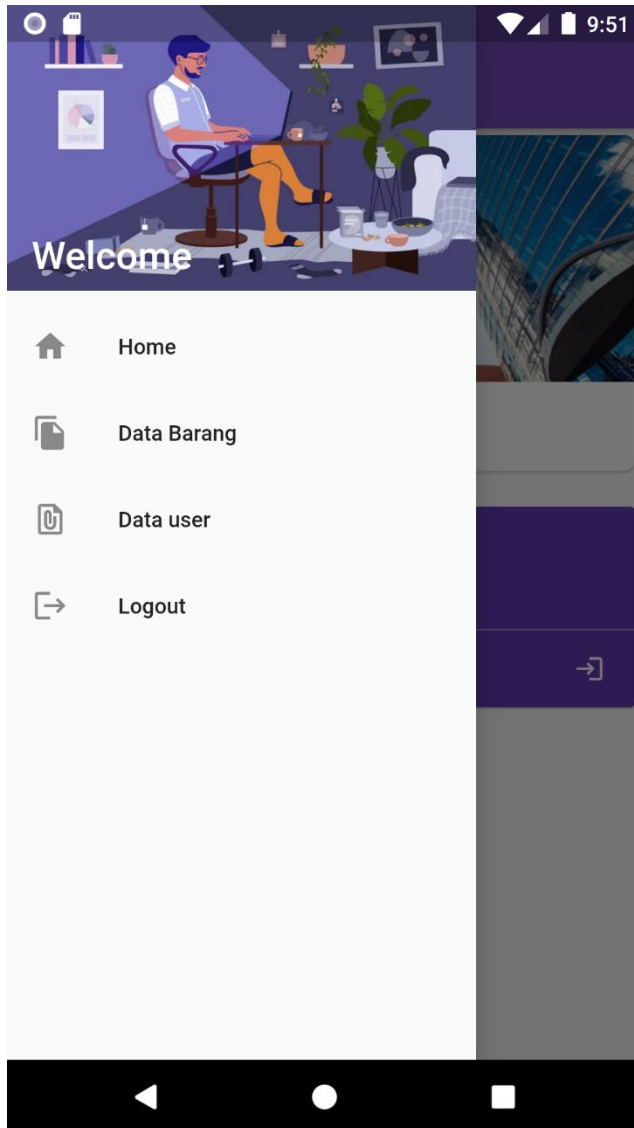
Jika ingin mengedit data kalian tinggal klik saja data yang mau di edit pada halaman data user lalu akan muncul tampilan seperti ini:



The screenshot shows a mobile application interface for editing user data. At the top, there is a purple header bar with a back arrow on the left, the text "Edit Data User" in the center, and a trash can icon on the right. Below the header, the form is divided into three sections: "Username" with a text input field containing "rafid" and a person icon; "Password" with a text input field containing five dots and a toggle icon; and "Email" with a text input field containing "rafid@gmail.com," and an envelope icon. At the bottom of the form, there is a purple button labeled "Edit Data". The entire form is set against a light gray background. At the very bottom of the screen, there is a black navigation bar with three white icons: a back arrow, a circle, and a square.

Jika telah diinput lalu klik edit user lalu data akan teredit dan kembali kehalaman user, jika ingin menghapus data kalian klik icon sampah pada pojok kanan atas pada halaman edit.

Setelah selesai kalian tinggal logout pada menu yang ada di dashboard atau data user/barang, klik garis 3 pada pojok kiri atas lalu muncul seperti ini:



Disini juga ada beberapa menu, jika ingin logout klik saja menu logout lalu akan menuju ke halaman login.

Terima kasih