

1. Login Page

```
// ignore_for_file: use_build_context_synchronously

import 'dart:convert';
import 'package:aplikasi_manajemen_buku/Page/home_page.dart'; // Ensure this
import points to your actual home page
import 'package:aplikasi_manajemen_buku/Page/registrasi_page.dart';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

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

  @override
  // ignore: library_private_types_in_public_api
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;

  final _emailTextboxController = TextEditingController();
  final _passwordTextboxController = TextEditingController();

  @override
  void dispose() {
    _emailTextboxController.dispose();
    _passwordTextboxController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: PreferredSize(
        preferredSize:
          const Size.fromHeight(60.0), // Adjust the height as needed
        child: Container(
          decoration: const BoxDecoration(
            gradient: LinearGradient(
              colors: [
                Color(0xFF6A11CB), // Start color (Blue-Purple)
                Color(0xFF2575FC), // End color (Blue)
              ],
              begin: Alignment.topLeft,
              end: Alignment.bottomRight,
            ),
          ),
        ),
      ),
    );
  }
}
```

```
    ),
    child: AppBar(
      centerTitle: true,
      elevation: 0,
      title: const Text(
        'Login',
        style: TextStyle(color: Colors.white, fontFamily: 'Courier
New'),
      ),
      backgroundColor:
        Colors.transparent, // Make AppBar background transparent
      iconTheme: const IconThemeData(color: Colors.white),
    ),
  ),
),
extendBodyBehindAppBar: true,
body: Center(
  child: SingleChildScrollView(
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Form(
        key: _formKey,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            _emailTextField(),
            const SizedBox(height: 16),
            _passwordTextField(),
            const SizedBox(height: 32),
            _buttonLogin(),
            const SizedBox(height: 30),
            _menuRegistrasi(),
          ],
        ),
      ),
    ),
  ),
),
);
}

Widget _emailTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Email",
      labelStyle: TextStyle(color: Colors.black, fontFamily: 'Courier New'),
      border: OutlineInputBorder(
```

```
        borderSide: BorderSide(color: Colors.black),
      ),
    ),
    style: const TextStyle(color: Colors.black, fontFamily: 'Courier New'),
    keyboardType: TextInputType.emailAddress,
    controller: _emailTextboxController,
    validator: (value) {
      if (value!.isEmpty) {
        return 'Email harus diisi';
      }
      return null;
    },
  );
}

Widget _passwordTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Password",
      labelStyle: TextStyle(color: Colors.black, fontFamily: 'Courier New'),
      border: OutlineInputBorder(
        borderSide: BorderSide(color: Colors.black),
      ),
    ),
    style: const TextStyle(color: Colors.black, fontFamily: 'Courier New'),
    keyboardType: TextInputType.text,
    obscureText: true,
    controller: _passwordTextboxController,
    validator: (value) {
      if (value!.isEmpty) {
        return "Password harus diisi";
      }
      return null;
    },
  );
}

Widget _buttonLogin() {
  return ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: const Color(0xFF6A11CB), // Blue-Purple Button
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30.0),
      ),
      textStyle: const TextStyle(
        fontFamily: 'Courier New',
      ),
    ),
  );
}
```

```
onPressed: _isLoading
? null
: () async {
    if (_formKey.currentState!.validate()) {
        setState(() {
            _isLoading = true;
        });

        String? userName = await _login();

        if (userName != null) {
            Navigator.pushReplacement(
                context,
                MaterialPageRoute(
                    builder: (context) => HomePage(userName: userName),
                ),
            );
        } else {
            ScaffoldMessenger.of(context).showSnackBar(
                const SnackBar(
                    content: Text("Login gagal! Periksa email dan
password."),
                ),
            );
        }

        setState(() {
            _isLoading = false;
        });
    },
    child: _isLoading
    ? const CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation<Color>(Colors.white),
    )
    : const Text(
        "Login",
        style: TextStyle(color: Colors.white),
    ),
);
}

Future<String?> _login() async {
    const url = 'http://103.196.155.42/api/login';
    final response = await http.post(
        Uri.parse(url),
        headers: {"Content-Type": "application/json"},
        body: jsonEncode({
```

```
        "email": _emailTextboxController.text,  
        "password": _passwordTextboxController.text,  
    })),  
    );  
  
    if (response.statusCode == 200) {  
        final data = jsonDecode(response.body);  
        if (data['status'] == true) {  
            // Return the user's name on successful login  
            return data['data']['user']['email'];  
        }  
    }  
  
    return null;  
}  
  
Widget _menuRegistrasi() {  
    return ElevatedButton(  
        style: ElevatedButton.styleFrom(  
            backgroundColor: const Color(0xFF6A11CB), // Matching Blue-Purple  
            Color  
            shape: RoundedRectangleBorder(  
                borderRadius: BorderRadius.circular(30.0),  
            ),  
            textStyle: const TextStyle(  
                fontFamily: 'Courier New',  
            ),  
        ),  
        onPressed: () {  
            Navigator.push(  
                context,  
                MaterialPageRoute(builder: (context) => const RegistrasiPage()),  
            );  
        },  
        child: const Text(  
            "Registrasi",  
            style: TextStyle(color: Colors.white),  
        ),  
    );  
}
```

Kode di atas adalah implementasi halaman login di aplikasi Flutter yang memungkinkan pengguna untuk masuk dengan email dan password. Berikut adalah fungsionalitas utama dari kode tersebut:

1. **Form Input dan Validasi:**

- Terdapat dua kolom input: satu untuk email dan satu untuk password. Keduanya memiliki validasi untuk memastikan bahwa tidak ada kolom yang dibiarkan kosong.
 - Jika ada kesalahan dalam pengisian, pengguna akan diberi tahu.
2. **Proses Login:**
- Ketika pengguna menekan tombol "Login", aplikasi akan memvalidasi input.
 - Jika input valid, aplikasi akan mengirimkan permintaan POST ke API untuk memverifikasi kredensial login pengguna.
 - Jika login berhasil (respon status dari API adalah true), aplikasi akan mengarahkan pengguna ke halaman HomePage dengan menampilkan nama pengguna.
 - Jika gagal, akan muncul pesan kesalahan yang memberitahu pengguna bahwa login tidak berhasil.
3. **State Management:**
- Menggunakan state management untuk menampilkan indikator pemuatan (CircularProgressIndicator) saat login sedang diproses agar pengguna tahu bahwa proses sedang berlangsung.
4. **Navigasi ke Halaman Registrasi:**
- Terdapat tombol "Registrasi" yang memungkinkan pengguna untuk berpindah ke halaman registrasi (RegistrasiPage) jika mereka belum memiliki akun.
5. **Keamanan Data:**
- Kata sandi yang dimasukkan oleh pengguna disembunyikan saat diketik untuk menjaga kerahasiaan.
6. **Penggunaan Kontroler:**
- Menggunakan TextEditingController untuk mengelola input teks dari pengguna dan membersihkannya saat halaman dihancurkan (dispose()).
- Secara keseluruhan, kode ini menangani fungsionalitas login dasar, validasi input, komunikasi dengan API untuk memverifikasi kredensial, dan navigasi ke halaman lain di aplikasi Flutter.

2. Register

```
// ignore_for_file: use_build_context_synchronously

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

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

  @override
  // ignore: library_private_types_in_public_api
  _RegistrasiPageState createState() => _RegistrasiPageState();
}
```

```
class _RegistrasiPageState extends State<RegistrasiPage> {
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;

  final _namaTextboxController = TextEditingController();
  final _emailTextboxController = TextEditingController();
  final _passwordTextboxController = TextEditingController();

  @override
  void dispose() {
    _namaTextboxController.dispose();
    _emailTextboxController.dispose();
    _passwordTextboxController.dispose();
    super.dispose();
  }

  Future<void> _register() async {
    if (_formKey.currentState!.validate()) {
      setState(() {
        _isLoading = true;
      });

      final String nama = _namaTextboxController.text;
      final String email = _emailTextboxController.text;
      final String password = _passwordTextboxController.text;

      final url = Uri.parse('http://103.196.155.42/api/registrasi');
      final response = await http.post(
        url,
        headers: {
          'Content-Type': 'application/json',
        },
        body: jsonEncode({
          'nama': nama,
          'email': email,
          'password': password,
        }),
      );

      setState(() {
        _isLoading = false;
      });

      if (response.statusCode == 200) {
        final responseData = jsonDecode(response.body);
        if (responseData['status']) {
          ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text("Registrasi berhasil!")),
          );
        }
      }
    }
  }
}
```

```
    );  
    Navigator.pop(context); // Kembali ke halaman login  
  } else {  
    ScaffoldMessenger.of(context).showSnackBar(  
      SnackBar(  
        content: Text("Registrasi gagal:  
${responseData['message']}")),  
    );  
  }  
  } else {  
    ScaffoldMessenger.of(context).showSnackBar(  
      const SnackBar(content: Text("Terjadi kesalahan, coba lagi  
nanti.")),  
    );  
  }  
}  
}  
}  
  
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: PreferredSize(  
      preferredSize:  
        const Size.fromHeight(60.0), // Adjust the height as needed  
      child: Container(  
        decoration: const BoxDecoration(  
          gradient: LinearGradient(  
            colors: [  
              Color(0xFF6A11CB), // Start color (Blue-Purple)  
              Color(0xFF2575FC), // End color (Blue)  
            ],  
            begin: Alignment.topLeft,  
            end: Alignment.bottomRight,  
          ),  
        ),  
      ),  
      child: AppBar(  
        centerTitle: true,  
        elevation: 0,  
        title: const Text(  
          'Registrasi',  
          style: TextStyle(color: Colors.white, fontFamily: 'Courier  
New'),  
        ),  
        backgroundColor:  
          Colors.transparent, // Make AppBar background transparent  
        iconTheme: const IconThemeData(color: Colors.white),  
      ),  
    ),  
  ),  
);
```



```
),
body: Center(
  child: SingleChildScrollView(
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Form(
        key: _formKey,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            _namaTextField(),
            const SizedBox(height: 16),
            _emailTextField(),
            const SizedBox(height: 16),
            _passwordTextField(),
            const SizedBox(height: 16),
            _passwordKonfirmasiTextField(),
            const SizedBox(height: 32),
            _buttonRegistrasi(),
          ],
        ),
      ),
    ),
  ),
),
);
}

Widget _namaTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Nama",
      border: OutlineInputBorder(),
    ),
    keyboardType: TextInputType.text,
    controller: _namaTextboxController,
    validator: (value) {
      if (value!.isEmpty || value.length < 3) {
        return "Nama harus diisi minimal 3 karakter";
      }
      return null;
    },
  );
}

Widget _emailTextField() {
  return TextFormField(
```

```
decoration: const InputDecoration(
  labelText: "Email",
  border: OutlineInputBorder(),
),
keyboardType: TextInputType.emailAddress,
controller: _emailTextboxController,
validator: (value) {
  if (value!.isEmpty) {
    return 'Email harus diisi';
  }

  // Validasi format email
  const pattern = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$';
  final regex = RegExp(pattern);

  if (!regex.hasMatch(value)) {
    return "Email tidak valid";
  }

  return null;
},
);
}

Widget _passwordTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Password",
      border: OutlineInputBorder(),
    ),
    keyboardType: TextInputType.text,
    obscureText: true,
    controller: _passwordTextboxController,
    validator: (value) {
      if (value!.isEmpty || value.length < 3) {
        return "Password harus diisi minimal 3 karakter";
      }
      return null;
    },
  );
}

Widget _passwordKonfirmasiTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Konfirmasi Password",
      border: OutlineInputBorder(),
    ),
  );
}
```

```
keyboardType: TextInputType.text,
obscureText: true,
validator: (value) {
  if (value != _passwordTextboxController.text) {
    return "Konfirmasi Password tidak sama";
  }
  return null;
},
);
}

Widget _buttonRegistrasi() {
  return ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: const Color(0xFF6A11CB), // Matching Blue-Purple
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30.0),
      ),
      textStyle: const TextStyle(
        fontFamily: 'Courier New',
      ),
    ),
    onPressed: _isLoading ? null : _register,
    child: _isLoading
      ? const CircularProgressIndicator(
          valueColor: AlwaysStoppedAnimation<Color>(Colors.white),
        )
      : const Text(
          "Registrasi",
          style: TextStyle(color: Colors.white),
        ),
  );
}
```

Kode di atas adalah implementasi halaman registrasi di aplikasi Flutter yang memungkinkan pengguna untuk membuat akun baru. Berikut adalah fungsionalitas utama dari kode tersebut:

1. **Form Input dan Validasi:**

- Terdapat empat kolom input: Nama, Email, Password, dan Konfirmasi Password.
- Setiap kolom memiliki validasi untuk memastikan input memenuhi syarat:
 - Nama: minimal 3 karakter.
 - Email: format email yang valid.
 - Password: minimal 3 karakter.
 - Konfirmasi Password: harus cocok dengan password yang diinputkan.

2. Proses Registrasi:

- Ketika pengguna menekan tombol "Registrasi", aplikasi akan memvalidasi semua input.
- Jika validasi berhasil, aplikasi akan mengirimkan permintaan POST ke API untuk melakukan pendaftaran akun baru.
- Jika pendaftaran berhasil (status dari API adalah true), pengguna akan melihat notifikasi keberhasilan dan diarahkan kembali ke halaman login.
- Jika gagal, pengguna akan mendapatkan pesan kesalahan yang sesuai.

3. State Management:

- Menggunakan state untuk mengontrol indikator pemuatan (CircularProgressIndicator) saat proses registrasi sedang berlangsung. Ini memberikan feedback visual kepada pengguna bahwa aplikasi sedang memproses permintaan.

4. Penggunaan Kontroler:

- Menggunakan TextEditingController untuk mengelola input dari pengguna dan membersihkannya saat halaman dihancurkan (dispose()).

5. Navigasi:

- Jika registrasi berhasil, aplikasi akan mengarahkan pengguna kembali ke halaman login menggunakan Navigator.pop(context);.

Secara keseluruhan, kode ini menangani fungsionalitas pendaftaran akun baru di aplikasi Flutter, termasuk validasi input, komunikasi dengan API untuk registrasi, dan navigasi di dalam aplikasi.

3. Home Page

```
import 'package:aplikasi_manajemen_buku/Page/ISBN_page.dart';
import 'package:aplikasi_manajemen_buku/Page/bahasa_page.dart';
import 'package:aplikasi_manajemen_buku/Page/genre_page.dart';
import 'package:aplikasi_manajemen_buku/Page/jumlahhalaman_page.dart';
import 'package:aplikasi_manajemen_buku/Page/lokasirak_page.dart';
import 'package:aplikasi_manajemen_buku/Page/penerbit_page.dart';
import 'package:aplikasi_manajemen_buku/Page/penulis_page.dart';
import 'package:aplikasi_manajemen_buku/Page/rating_page.dart';
import 'package:aplikasi_manajemen_buku/Page/status_page.dart';
import 'package:aplikasi_manajemen_buku/Page/tahunterbit_page.dart';
import 'package:flutter/material.dart';

class HomePage extends StatelessWidget {
  final String userName;

  // Constructor to receive the user's name
  const HomePage({super.key, required this.userName});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: PreferredSize(
        preferredSize:
```

```
        const Size.fromHeight(60.0), // Adjust the height as needed
      child: Container(
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Color(0xFF6A11CB), // Start color (Blue-Purple)
              Color(0xFF2575FC), // End color (Blue)
            ],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
      ),
      child: AppBar(
        centerTitle: true,
        elevation: 0,
        title: const Text(
          'Home',
          style: TextStyle(color: Colors.white, fontFamily: 'Courier
New'),
        ),
        backgroundColor:
          Colors.transparent, // Make AppBar background transparent
        iconTheme: const IconThemeData(color: Colors.white),
      ),
    ),
  ),
  drawer: Drawer(
    child: ListView(
      padding: EdgeInsets.zero,
      children: [
        DrawerHeader(
          decoration: const BoxDecoration(
            color: Color.fromARGB(255, 53, 0, 122),
          ),
          child: Text(
            'Welcome, $userName!',
            style: const TextStyle(
              color: Colors.white,
              fontSize: 20,
              fontFamily: 'Courier New',
            ),
          ),
        ),
        // Menu Items
        _drawerItem(context, 'Genre Page', Icons.pages, const
GenrePage()),
        _drawerItem(
          context, 'Penulis Page', Icons.pages, const PenulisPage()),
```

```

        _drawerItem(
            context, 'Penerbit Page', Icons.pages, const PenerbitPage()),
        _drawerItem(context, 'Tahun Terbit Page', Icons.pages,
            const TahunterbitPage()),
        _drawerItem(context, 'ISBN Page', Icons.pages, const IsbnPage()),
        _drawerItem(context, 'Jumlah Halaman Page', Icons.pages,
            const JumlahhalamanPage()),
        _drawerItem(
            context, 'Bahasa Page', Icons.pages, const BahasaPage()),
        _drawerItem(
            context, 'Lokasi Rak Page', Icons.pages, const
LokasirakPage()),
        _drawerItem(
            context, 'Status Page', Icons.pages, const StatusPage()),
        _drawerItem(
            context, 'Rating Page', Icons.pages, const RatingPage()),
    ],
),
),
body: Container(
    width: MediaQuery.of(context).size.width, // Full width
    height: MediaQuery.of(context).size.height, // Full heigh
    color: const Color.fromARGB(255, 53, 0, 122),
    child: Center(
        child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                // Circular Image Widget
                const CircleAvatar(
                    radius: 50, // Adjust the size as needed
                    backgroundImage: AssetImage(
                        'profil_image.png'), // Change to your image asset path
                ),
                const SizedBox(height: 20),
                // Welcome Text
                Text(
                    'Welcome, $userName!',
                    style: const TextStyle(
                        fontSize: 24,
                        fontWeight: FontWeight.bold,
                        fontFamily: 'Courier New',
                        color: Colors.white,
                    ),
                ),
            ],
        ),
    ),
),
),
),
),
),

```

```
);  
}  
}  
  
Widget _drawerItem(  
  BuildContext context, String title, IconData icon, Widget destination) {  
  return ListTile(  
    leading: Icon(icon),  
    title: Text(title),  
    onTap: () {  
      Navigator.push(  
        context,  
        MaterialPageRoute(builder: (context) => destination),  
      );  
    },  
  );  
}
```

Kode di atas merupakan implementasi dari halaman beranda (HomePage) untuk aplikasi Flutter yang mengelola navigasi ke berbagai halaman lainnya. Berikut adalah penjelasan detail dari kode tersebut:

1. Struktur Halaman Utama

- **AppBar:** Menggunakan AppBar dengan desain gradasi biru-ungu ke biru dan teks yang terpusat di tengah.
- **Drawer:** Menyediakan navigasi ke berbagai halaman melalui Drawer yang memuat daftar menu.
 - **Drawer Header:** Menampilkan pesan "Welcome, [nama pengguna]!" menggunakan nama pengguna yang diterima dari konstruktor.
 - **Menu Items:** Berbagai halaman seperti Genre, Penulis, Penerbit, Tahun Terbit, ISBN, dan lainnya dapat diakses dengan mengklik item yang ada di Drawer.
- **Body:** Bagian utama halaman menampilkan gambar profil melingkar dan teks sambutan dengan nama pengguna yang dicetak tebal dan besar.

2. Drawer dan Navigasi

- Fungsi `_drawerItem` digunakan untuk membuat item di Drawer. Saat pengguna mengetuk salah satu item, mereka akan dinavigasi ke halaman tujuan yang sesuai menggunakan `Navigator.push`.
- Setiap item di drawer memiliki ikon, teks, dan rute ke halaman lain, memungkinkan pengguna untuk berpindah halaman dengan mudah.

3. Penerimaan Nama Pengguna

- Nama pengguna (`userName`) diterima sebagai parameter dari konstruktor `HomePage` dan digunakan untuk menampilkan pesan selamat datang di Drawer Header dan di bagian body.

4. Desain dan Tema

- Menggunakan warna yang konsisten dengan tema biru dan ungu gelap.

- Teks menggunakan font khusus (Courier New), memberikan tampilan yang lebih unik dan bergaya.
- Gambar profil disesuaikan dalam bentuk lingkaran (CircleAvatar) dengan radius yang diatur dan memuat gambar dari aset aplikasi.

5. Fungsionalitas Utama

- **Drawer Navigation:** Membuat navigasi lebih mudah dengan menyediakan Drawer yang berisi item menu ke berbagai halaman manajemen buku.
- **Penerimaan Nama Pengguna:** Halaman ini menyapa pengguna secara personal menggunakan nama yang diterima dari LoginPage atau halaman registrasi.
- **UI Responsif:** Menggunakan MediaQuery untuk menyesuaikan ukuran widget berdasarkan lebar dan tinggi layar.