TEKNOLOGI PEMROGRAMAN MOBILE

SEMESTER GENAP TAHUN AJARAN 2022/2023

LAPORAN TUGAS AKHIR

Oleh

Nama : Rhyo Argasiwi

NIM : 123200059

Kelas : IF - H

PROGRAM STUDI INFORMATIKA

JURUSAN INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" YOGYAKARTA

2023

LANGKAH PENGERJAAN

1. Pastikan sudah menginstal Flutter di komputer yang digunakan.
2. Buat proyek baru melalui IDE seperti Android Studio atau Visual Studio Code. Pada kali ini, saya menggunakan IDE Visual Studio Code.
3. Menetukan tema dari project yang dibuat. Pada project kali ini, saya membuat aplikasi yang bertemakan “Valorant Weapon Skins List”.
4. Membuat halaman login dan register yang berisi widget TextFormField untuk username dan password, serta ElevatedButton dan TextButton untuk tombol login dan tombol register. Kemudian, memberikan error handling untuk memeriksa apakah username dan password sudah benar atau belum.
5. Membuat halaman Homepage yang berisi halaman list Weapon Skin Valorant, serta halaman untuk konversi.
6. Di dalam halaman list Weapon Skin Valorant, ditampilkan list dari skin senjata yang ada di dalam game Valorant, serta detail varian dari beberapa senjata.
7. Di dalam halaman konversi, terdapat halaman Konversi Uang yang bisa mengubah nilai mata uang ke beberapa nilai mata uang yang sudah ditentukan, serta terdapat halaman Konversi Waktu yang bisa mengubah waktu ke beberapa zona waktu yang sudah ditentukan.
8. Membuat halaman Kesan Pesan yang berisikan kesan dan pesan selama mengikuti mata kuliah Teknologi & Pemrograman Mobile.
9. Membuat halaman profile yang terkoneksi dengan data dari user. Di dalam halaman profile, ditampilkan foto dan username.
10. Membuat tombol logout yang digunakan untuk keluar dari akun yang sedang digunakan.

SOURCE CODE

1. main.dart

```
import 'package:flutter/material.dart';
import 'package:tpm_final_project/views/loginpage.dart';

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

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Valorant Weapon Skin List',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LoginPage(),
      debugShowCheckedModeBanner: false,
    );
  }
}
```

main.dart merupakan awal dari program kita berjalan, atau tempat kita menjalankan program yang sudah kita buat. Di main.dart, kita menentukan halaman mana yang akan muncul pertama kali ketika user membuka aplikasi.

2. homepage.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:tpm_final_project/helpers/shared_pref.dart';
import 'package:tpm_final_project/views/kesanpesan.dart';
import 'package:tpm_final_project/views/konversipage.dart';
import 'package:tpm_final_project/views/loginpage.dart';
import 'package:tpm_final_project/views/weaponskinlist.dart';
import 'package:tpm_final_project/views/profilepage.dart';

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

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int _selectedIndex = 0;

  static List<Widget> _widgetOptions = <Widget>[
    Home_Page(),
    KesanPesanPage(),
```

```
    ProfilePage(),
    const Log_Out(),
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
      if (index == 3) {
        logout(context);
      }
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: _widgetOptions[_selectedIndex],
      ),
      bottomNavigationBar: BottomNavigationBar(
        items: const [
          BottomNavigationBarItem(
            icon: Icon(Icons.home),
            label: 'Home Page',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.book),
            label: 'Kesan Pesan',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.account_circle),
            label: 'Profile',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.logout),
            label: 'Log Out',
          ),
        ],
        currentIndex: _selectedIndex,
        selectedItemColor: Color.fromRGBO(29, 66, 137, 1),
        unselectedItemColor: Color.fromRGBO(200, 16, 46, 0.6),
        showUnselectedLabels: true,
        onTap: _onItemTapped,
      ),
    );
  }
}

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

  @override
  State<Home_Page> createState() => _Home_PageState();
}
```

```
class _Home_PageState extends State<Home_Page> {
  @override
  Widget build(BuildContext context) {
    return SafeArea(
        child: Scaffold(
          appBar: AppBar(
            title: Text("Home Page"),
            backgroundColor: Color.fromRGBO(29, 66, 137, 1),
          ),
          body: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                IconButton(
                  icon: Icon(Icons.list_alt),
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) =>
WeaponSkinsList()),
                    );
                  },
                ),
                Text('Valorant Weapon Skins List'),
                SizedBox(height: 20),
                IconButton(
                  icon: Icon(Icons.compare_arrows),
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) =>
KonversiPage()),
                    );
                  },
                ),
                Text('Konversi'),
              ],
            ),
          ),
        )
    );
  }
}

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

  @override
  State<Log_Out> createState() => _Log_OutState();
}

class _Log_OutState extends State<Log_Out> {
  @override
```

```
  Widget build(BuildContext context) {
    return Scaffold();
  }
}

void logout(context) async {
    await UserPreferences.setLoggedIn(false);
    await UserPreferences.setUsername('');
    Navigator.pushReplacement(
      context, MaterialPageRoute(builder: (context) =>
LoginPage()));
}
```

homepage.dart adalah halaman yang akan muncul ketika user sudah berhasil login. Di dalam homepage berisi widget untuk membuat bottom navbar. Di dalam bottom navbar, digunakan icon untuk mempermudah user untuk mengetahui halaman yang ada di bottom navbar.

3. kesanpesan.dart

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

class KesanPesanPage extends StatelessWidget {
  const KesanPesanPage({ Key? key }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          title: Text("Kesan Pesan"),
          backgroundColor: Color.fromRGBO(29, 66, 137, 1),
        ),
        body: ListView(
        children: [
          Container(
            padding: EdgeInsets.all(15),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text("Kesan :",
                  style: TextStyle(fontSize: 24, fontWeight:
FontWeight.bold),),
                  Text('Mata Kuliah Teknologi Pemrograman Mobile
ini sangat membantu saya dalam mempelajari pembuatan program
Mobile.',
                  style: TextStyle(fontSize: 18),),
                  Text('Dengan adanya mata kuliah ini, saya
dapat mempelajari program Mobile dengan lebih baik.',
                  style: TextStyle(fontSize: 18),),
                SizedBox(height: 15,),
                Text("Pesan :",
                  style: TextStyle(fontSize: 24, fontWeight:
FontWeight.bold),),
```

```
                    Text('Semoga tugas yang diberikan tidak
terlalu berat ya pak :)',
                    style: TextStyle(fontSize: 18),),
                    Text('Dan semoga saya mendapatkan nilai A ya
pak :)',
                    style: TextStyle(fontSize: 18),),
                ],
              ),

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

kesanpesan.dart berisikan kesan dan pesan selama mengikuti pembelajaran di matakuliah Teknologi & Pemrograman Mobile. File ini berisikan beberapa text yang membentuk kalimat untuk ditampilkan kepada user. Text tersebut diberikan beberapa styling seperti styling ukuran dan ketebalan huruf.

4. profile.dart

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

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

  @override
  State<ProfilePage> createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  String username = "";

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

  Future<void> getUsername() async {
    SharedPreferences prefs = await
SharedPreferences.getInstance();
    String? storedUsername = prefs.getString("username");
    setState(() {
      username = storedUsername ?? "";
    });
  }

  @override
 Widget build(BuildContext context) {
```

```
  String imagePath = (username == 'Rhyo')
        ? 'assets/rhyoargasiwi.jpg'
        : 'assets/avatar.png';
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          backgroundColor: Color.fromRGBO(29, 66, 137, 1),
          title: Text("Profile"),
        ),
        body: Container(
        padding: EdgeInsets.all(20),
        alignment: Alignment.center,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            CircleAvatar(
              radius: 60,
              backgroundImage: AssetImage(imagePath),
            ),
            SizedBox(height: 20),
            Text(
              '$username',
              style: TextStyle(fontSize: 24),
              maxLines: 1,
            ),
          ],
        ),
      ),
      ),
    );
  }
}
```

profile.dart berisikan data dari user, dimana datauser yang ditampilkan adalah foto dan username. Username pada halaman ini diambil dari shared preferences dari akun yang sedang digunakan atau akun yang digunakan untuk login. Oleh karena itu, setiap akun akan menampilkan nama yang berbeda ketika kita melihat halaman Profile.

5. weaponskinlist.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:tpm_final_project/models/weapon_skin.dart';
import 'package:tpm_final_project/views/weaponskin_detail.dart';

class WeaponSkinsList extends StatefulWidget {
  @override
  _WeaponSkinsListState createState() =>
_WeaponSkinsListState();
}

class _WeaponSkinsListState extends State<WeaponSkinsList> {
  List<dynamic> weaponSkins = [];
```

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

  Future<void> fetchWeaponSkins() async {
    var url = Uri.parse('https://valorant-
api.com/v1/weapons/skins');
    var response = await http.get(url);

    if (response.statusCode == 200) {
      var jsonData = json.decode(response.body);
      setState(() {
        weaponSkins = jsonData['data'];
      });
    } else {
      print('Error: ${response.statusCode}');
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Valorant Weapon Skins'),
        backgroundColor: Color.fromRGBO(29, 66, 137, 1),
      ),
      body: GridView.builder(
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
          crossAxisCount: 2,
          childAspectRatio: 0.8,
        ),
        itemCount: weaponSkins.length,
        itemBuilder: (context, index) {
          if (weaponSkins[index]['displayIcon'] != null) {
            return InkWell(
              onTap: () {
                Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) => WeaponSkinDetail(
                          weaponSkin: WeaponSkin(
                        displayName:
weaponSkins[index]['displayName'],
                            displayIcon:
weaponSkins[index]['displayIcon'],
                        chromas: weaponSkins[index]['chromas'],
                      )),
                    ));
              },
              child: Column(
                children: [
                  Container(
```

```
                    width: double.infinity,
                    height: 150,
                    child: Image.network(
                      weaponSkins[index]['displayIcon'],
                      fit: BoxFit.contain,
                    ),
                  ),
                  Padding(
                    padding: EdgeInsets.all(8.0),
                    child: Text(
                      weaponSkins[index]['displayName'],
                      style: TextStyle(
                        fontSize: 16,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ),
                ],
              ),
            );
          } else {
            return SizedBox.shrink();
          }
        },
      ),
    );
  }
}
```

weaponskinlist.dart menampilkan list dari skin senjata yang ada di Valorant. Hal pertama yang dilakukan di file ini adalah fetching API untuk mendapatkan data dari API yang digunakan. Kemudian, untuk di halaman ini, data yang ditampilkan hanya ‘displayIcon’ dan ‘displayName’ dari setiap skin senjata. Dengan menggunakan gridView serta membuat sebauh variabel untuk menghitung total data yang ada, seluruh data dapat ditampilkan tanpa harus membuatnya satu per satu.

6. weaponskin_detail.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:tpm_final_project/models/weapon_skin.dart';

class WeaponSkinDetail extends StatefulWidget {
  WeaponSkin weaponSkin;
  WeaponSkinDetail({Key? key, required this.weaponSkin}) :
super(key: key);

  @override
  State<WeaponSkinDetail> createState() =>
_WeaponSkinDetailState();
}

class _WeaponSkinDetailState extends State<WeaponSkinDetail> {
  @override
  Widget build(BuildContext context) {
```

```
    print(widget.weaponSkin.chromas);
    return SafeArea(
        child: Scaffold(
      appBar: AppBar(
        title: Text(widget.weaponSkin.displayName!),
        backgroundColor: Color.fromRGBO(29, 66, 137, 1),
      ),
      body: ListView.builder(
        shrinkWrap: true,
        itemCount: widget.weaponSkin.chromas!.length,
        itemBuilder: (context, index) {
          return Container(
              width: MediaQuery.of(context).size.width,
              height: MediaQuery.of(context).size.height / 3,
              decoration: BoxDecoration(
                image: DecorationImage(
                  image: NetworkImage(

widget.weaponSkin.chromas![index]['fullRender']),
                  fit: BoxFit.contain,
                ),
              ));
        },
      ),
    ));
  }
}
```

weaponskin_detail.dart merupakan lanjutan dari halaman weaponskinlist.dart, dimana ketika menekan salah satu skin senjata, maka akan masuk ke halaman ini. Dengan memnggunakan itemCount untuk menghitung total data yang ada, serta menerapkannya menggunakan itemBuilder, maka bisa ditampilkan detail dari skin senjata yang dipilih.

7. loginpage.dart

```
// import 'package:flutter/cupertino.dart';
// import 'package:flutter/material.dart';
// import 'package:tpm_final_project/views/homepage.dart';
// import 'package:tpm_final_project/views/registerpage.dart';

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

//   @override
//   State<LoginPage> createState() => _LoginPageState();
// }

// class _LoginPageState extends State<LoginPage> {
//   String username = "";
//   String password = "";
//   bool isLoginSuccess = true;
//   bool isUsername = true;
//   bool isPassword = true;
//   @override
//   Widget build(BuildContext context) {
```

```
//     return SafeArea(
//         child: Scaffold(
//       appBar: AppBar(
//         backgroundColor: (isLoginSuccess) ?
Color.fromRGBO(29, 66, 137, 1) : Color.fromRGBO(200, 16, 46, 1),
//         title: Text("Login Page"),
//       ),
//       body: Column(
//         mainAxisAlignment: MainAxisAlignment.center,
//         crossAxisAlignment: CrossAxisAlignment.center,
//         children: [
//           _usernameField(),
//           _passwordField(),
//           _loginButton(),
//           _register()
//         ],
//       ),
//     ));
//   }

//   Widget _usernameField() {
//     return Container(
//       padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
//       child: TextFormField(
//         onChanged: (value) {
//           username = value;
//         },
//         decoration: InputDecoration(
//           labelText: 'Username',
//           enabledBorder: OutlineInputBorder(
//               borderRadius: BorderRadius.circular(10),
//               borderSide: BorderSide(
//                 // color: Colors.blue,
//                 color:
//                     (isLoginSuccess || isUsername) ?
Color.fromRGBO(29, 66, 137, 1) : Color.fromRGBO(200, 16, 46, 1),
//               )),
//           focusedBorder: OutlineInputBorder(
//               borderRadius: BorderRadius.circular(10),
//               borderSide: BorderSide(
//                 width: 3,
//                 color: Colors.blue,
//               )),
//         ),
//       ),
//     );
//   }

//   Widget _passwordField() {
//     return Container(
//       padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
//       child: TextFormField(
```

```
//          onChanged: (value) {
//            password = value;
//          },
//          obscureText: true,
//          decoration: InputDecoration(
//            labelText: 'Password',
//            enabledBorder: OutlineInputBorder(
//                borderRadius: BorderRadius.circular(10),
//                borderSide: BorderSide(
//                  // color: Colors.blue,
//                  color:
//                      (isLoginSuccess || isPassword) ?
Color.fromRGBO(29, 66, 137, 1) : Color.fromRGBO(200, 16, 46, 1),
//                )),
//            focusedBorder: OutlineInputBorder(
//                borderRadius: BorderRadius.circular(10),
//                borderSide: BorderSide(
//                  width: 3,
//                  color: Colors.blue,
//                )),
//          ),
//        ),
//      );
//    }

//    Widget _loginButton() {
//      return Container(
//        padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
//        width: MediaQuery.of(context).size.width,
//        child: ElevatedButton(
//          style: ElevatedButton.styleFrom(
//              primary: (isLoginSuccess) ? Color.fromRGBO(29,
66, 137, 1) : Color.fromRGBO(200, 16, 46, 1)),
//          onPressed: () {
//            String text = "";
//            if (username == "rhyo" && password == "0059") {
//              setState(() {
//                isLoginSuccess = true;
//                isUsername = true;
//                isPassword = true;
//                text = "Login Berhasil";
//              });
//              Navigator.of(context)
//                  .pushReplacement(MaterialPageRoute(builder:
(context) {
//                return HomePage();
//              }));
//            } else if (username != "rhyo" && password ==
"0059") {
//              setState(() {
//                isLoginSuccess = false;
//                isUsername = false;
//                isPassword = true;
```

```
//                text = "Login Gagal, Username Salah";
//              });
//            } else if (username == "rhyo" && password !=
"0059") {
//              setState(() {
//                isLoginSuccess = false;
//                isUsername = true;
//                isPassword = false;
//                text = "Login Gagal, Password Salah";
//              });
//            } else {
//              setState(() {
//                isLoginSuccess = false;
//                isUsername = false;
//                isPassword = false;
//                text = "Login Gagal";
//              });
//            }

//            SnackBar snackBar = SnackBar(
//              content: Text(text),
//            );
//
ScaffoldMessenger.of(context).showSnackBar(snackBar);
//          },
//          child: Text(
//            "Login",
//            style: TextStyle(fontSize: 20),
//          ),
//        ),
//      );
//    }

//    Widget _register(){
//      return Container(
//        padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
//        width: MediaQuery.of(context).size.width,
//        child: TextButton(
//          style: TextButton.styleFrom(
//              primary: (isLoginSuccess) ? Color.fromRGBO(29,
66, 137, 1) : Color.fromRGBO(200, 16, 46, 1)),
//          onPressed: () {
//            Navigator.of(context)
//            .pushReplacement(MaterialPageRoute(builder:
(context) {
//              return RegisterPage();
//              }));
//          },
//          child: Text(
//            "Register",
//            style: TextStyle(fontSize: 20),
//          ),
//        ),
```

```
//     );
//   }
// }


import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:tpm_final_project/utils/auth_services.dart';
import 'package:tpm_final_project/views/homepage.dart';
import 'package:tpm_final_project/views/registerpage.dart';

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

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  String username = "";
  String password = "";
  bool isLoginSuccess = true;
  bool isUsername = true;
  bool isPassword = true;

  AuthService authService = AuthService();

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          backgroundColor: (isLoginSuccess) ? Color.fromRGBO(29,
66, 137, 1) : Color.fromRGBO(200, 16, 46, 1),
          title: Text("Login Page"),
        ),
        body: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            _usernameField(),
            _passwordField(),
            _loginButton(),
            _register(),
          ],
        ),
      ),
    );
  }

  Widget _usernameField() {
    return Container(
      padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
```

```
      child: TextFormField(
        onChanged: (value) {
          setState(() {
            username = value;
          });
        },
        decoration: InputDecoration(
          labelText: 'Username',
          enabledBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(10),
            borderSide: BorderSide(
              color: (isLoginSuccess || isUsername) ?
Color.fromRGBO(29, 66, 137, 1) : Color.fromRGBO(200, 16, 46, 1),
            ),
          ),
          focusedBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(10),
            borderSide: BorderSide(
              width: 3,
              color: Colors.blue,
            ),
          ),
        ),
      ),
    );
  }

  Widget _passwordField() {
    return Container(
      padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
      child: TextFormField(
        onChanged: (value) {
          setState(() {
            password = value;
          });
        },
        obscureText: true,
        decoration: InputDecoration(
          labelText: 'Password',
          enabledBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(10),
            borderSide: BorderSide(
              color: (isLoginSuccess || isPassword) ?
Color.fromRGBO(29, 66, 137, 1) : Color.fromRGBO(200, 16, 46, 1),
            ),
          ),
          focusedBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(10),
            borderSide: BorderSide(
              width: 3,
              color: Colors.blue,
            ),
          ),
```

```
        ),
      ),
    );
  }

  Widget _loginButton() {
    return Container(
      padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
      width: MediaQuery.of(context).size.width,
      child: ElevatedButton(
        style: ElevatedButton.styleFrom(
          primary: (isLoginSuccess) ? Color.fromRGBO(29, 66,
137, 1) : Color.fromRGBO(200, 16, 46, 1),
        ),
        onPressed: () async {
          String text = "";
          bool loginSuccess = await authService.login(username,
password);

          if (loginSuccess) {
            setState(() {
              isLoginSuccess = true;
              isUsername = true;
              isPassword = true;
              text = "Login Berhasil";
            });

Navigator.of(context).pushReplacement(MaterialPageRoute(builder:
(context) {
              return HomePage();
            }));
          } else {
            setState(() {
              isLoginSuccess = false;
              isUsername = false;
              isPassword = false;
              text = "Login Gagal";
            });
          }

          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text(text),
            ),
          );
        },
        child: Text(
          "Login",
          style: TextStyle(fontSize: 20),
        ),
      ),
    );
  }
```

```
  Widget _register() {
    return Container(
      padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
      width: MediaQuery.of(context).size.width,
      child: TextButton(
        style: TextButton.styleFrom(
          primary: (isLoginSuccess) ? Color.fromRGBO(29, 66,
137, 1) : Color.fromRGBO(200, 16, 46, 1),
        ),
        onPressed: () {

Navigator.of(context).pushReplacement(MaterialPageRoute(builder:
(context) {
            return RegisterPage();
          }));
        },
        child: Text(
          "Register",
          style: TextStyle(fontSize: 20),
        ),
      ),
    );
  }
}
```

loginpage.dart akan tampil sebagai halaman pertama ketika kita membuka aplikasi. Sesuai namanya, halaman ini digunakan oleh user untuk melakukan login. Ketika mengisikan username dan password, maka username dan password akan diambil dan dicek oleh auth_service, apakah data tersebut ada di database atau tidak. Jika ada, maka akan di set shared preference nya, serta dapat lanjut ke halaman berikutnya. Jika tidak, maka user salah memasukkan username atau password, atau user dapat register dengan memasukkan username dan password baru.

8. registerpage.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:tpm_final_project/utils/auth_services.dart';
import 'package:tpm_final_project/views/loginpage.dart';

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

  @override
  State<RegisterPage> createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  String username = "";
  String password = "";
  AuthService authService = AuthService();
```

```
  @override
  Widget build(BuildContext context) {
    return SafeArea(
        child: Scaffold(
      appBar: AppBar(
        backgroundColor: Color.fromRGBO(29, 66, 137, 1),
        title: Text("Register Page"),
      ),
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          _usernameField(),
          _passwordField(),
          _registerButton(),
          _login()
        ],
      ),
    ));
  }

  Widget _usernameField() {
    return Container(
      padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
      child: TextFormField(
        onChanged: (value) {
          setState(() {
            username = value;
          });
        },
        decoration: InputDecoration(
            labelText: 'Username',
            enabledBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(10),
                borderSide: BorderSide(
                  color: Color.fromRGBO(29, 66, 137, 1),
                )),
            focusedBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(10),
                borderSide: BorderSide(
                  width: 3,
                  color: Colors.blue,
                ))),
      ),
    );
  }

  Widget _passwordField() {
    return Container(
      padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
      child: TextFormField(
        onChanged: (value) {
```

```
          setState(() {
            password = value;
          });
        },
        obscureText: true,
        decoration: InputDecoration(
            labelText: 'Password',
            enabledBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(10),
                borderSide: BorderSide(
                  color: Color.fromRGBO(29, 66, 137, 1),
                )),
            focusedBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(10),
                borderSide: BorderSide(
                  width: 3,
                  color: Colors.blue,
                ))),
      ),
    );
  }

  Widget _registerButton() {
    return Container(
      width: double.infinity,
      padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
      child: ElevatedButton(
        onPressed: () async {
          String text = "";
          bool registerSuccess = await
authService.register(username, password);

          if (registerSuccess) {
            text = 'Registration successful';

Navigator.of(context).pushReplacement(MaterialPageRoute(
                builder: (context) => LoginPage(),
            ));
          } else {
            text = 'Registration failed';
          }

          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text(text),
            ),
          );
        },
        child: Text("Register"),
        style: ElevatedButton.styleFrom(
            primary: Color.fromRGBO(29, 66, 137, 1),
            padding: EdgeInsets.symmetric(vertical: 20,
horizontal: 40),
```

```
            textStyle: TextStyle(fontSize: 20, fontWeight:
FontWeight.bold)),
      ),
    );
  }

  Widget _login() {
    return Container(
      padding: EdgeInsets.symmetric(vertical: 10, horizontal:
20),
      width: MediaQuery.of(context).size.width,
      child: TextButton(
        style: TextButton.styleFrom(primary: Color.fromRGBO(29,
66, 137, 1)),
        onPressed: () {
          Navigator.of(context)
              .pushReplacement(MaterialPageRoute(builder:
(context) {
            return LoginPage();
          }));
        },
        child: Text(
          "Login",
          style: TextStyle(fontSize: 20),
        ),
      ),
    );
  }
}
```

register.dart digunakan ketika user belum mempunyai akun dan ingin membuat akun baru agar bisa melakukan login. User akan memasukkan username dan password sesuai keinginan user. Ketika menekan tombol register, maka username dan password yang tadi sudah dimasukkan oleh user akan diambil dan dienkripsi. Setelah itu, data tersebut akan dimasukkan ke dalam database, sehingga ketika user ingin masuk ke aplikasi, tinggal melakukan login saja, tanpa harus register.

9. konversipage.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:tpm_final_project/views/konversi_uang.dart';
import 'package:tpm_final_project/views/konversi_waktu.dart';

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

  @override
  State<KonversiPage> createState() => _KonversiPageState();
}

class _KonversiPageState extends State<KonversiPage> {
  @override
  Widget build(BuildContext context) {
    return SafeArea(
```

```
      child: Scaffold(
        appBar: AppBar(
          title: Text("Konversi"),
          backgroundColor: Color.fromRGBO(29, 66, 137, 1),
        ),
        body: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                IconButton(
                  icon: Icon(Icons.money),
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) =>
KonversiUang()),
                    );
                  },
                ),
                Text('Konversi Uang'),
                SizedBox(height: 20),
                IconButton(
                  icon: Icon(Icons.timer),
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) =>
KonversiWaktu()),
                    );
                  },
                ),
                Text('Konversi Waktu'),
              ],
            ),
          ),
      )
    );
  }
}
```

konversi.dart merupakan halaman yang digunakan untuk menampilkan menu-menu konversi, yaitu menu konversi_uang dan konversi_waktu.

10.konversi_uang.dart

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

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

  @override
  _KonversiUangState createState() => _KonversiUangState();
}
```

```
class _KonversiUangState extends State<KonversiUang> {
  late double _input;
  late double _output;
  // currency IDR, USD, EUR, JPY
  late String _currencyInput;
  late String _currencyOutput;
  late String _result;

  final TextEditingController _inputController =
TextEditingController();

  @override
  void initState() {
    super.initState();
    _input = 0;
    _currencyInput = 'IDR';
    _currencyOutput = 'IDR';
    _result = '';
  }

  void _onInputChanged(String value) {
    setState(() {
      _input = double.tryParse(value) ?? 0;
    });
  }

  void _onCurrencyInputChanged(String? value) {
    setState(() {
      _currencyInput = value ?? 'IDR';
    });
  }

  void _onCurrencyOutputChanged(String? value) {
    setState(() {
      _currencyOutput = value ?? 'IDR';
    });
  }

  void _convert() {
    setState(() {
      switch (_currencyInput) {
        case 'IDR':
          switch (_currencyOutput) {
            case 'IDR':
              _output = _input;
              break;
            case 'USD':
              _output = _input / 14200;
              break;
            case 'EUR':
              _output = _input / 17000;
              break;
            case 'JPY':
              _output = _input / 130;
```

```
            break;
        }
        break;
      case 'USD':
        switch (_currencyOutput) {
          case 'IDR':
            _output = _input * 14200;
            break;
          case 'USD':
            _output = _input;
            break;
          case 'EUR':
            _output = _input * 0.85;
            break;
          case 'JPY':
            _output = _input * 110;
            break;
        }
        break;
      case 'EUR':
        switch (_currencyOutput) {
          case 'IDR':
            _output = _input * 17000;
            break;
          case 'USD':
            _output = _input * 1.17;
            break;
          case 'EUR':
            _output = _input;
            break;
          case 'JPY':
            _output = _input * 130;
            break;
        }
        break;
      case 'JPY':
        switch (_currencyOutput) {
          case 'IDR':
            _output = _input * 130;
            break;
          case 'USD':
            _output = _input * 0.0091;
            break;
          case 'EUR':
            _output = _input * 0.0077;
            break;
          case 'JPY':
            _output = _input;
            break;
        }
        break;
    }
    _result = _output.toStringAsFixed(2);
  });
```

```
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Konversi Uang'),
        backgroundColor: Color.fromRGBO(29, 66, 137, 1),
      ),
      body: Container(
        padding: const EdgeInsets.all(20),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            const SizedBox(
              height: 20,
            ),
            TextField(
              onChanged: _onInputChanged,
              controller: _inputController,
              keyboardType: TextInputType.number,
              decoration: const InputDecoration(
                border: OutlineInputBorder(
                  borderRadius:
BorderRadius.all(Radius.circular(30)),
                ),
                labelText: 'Input',
              ),
            ),
            const SizedBox(
              height: 20,
            ),
            DropdownButton<String>(
              value: _currencyInput,
              onChanged: _onCurrencyInputChanged,
              items: const <String>['IDR', 'USD', 'EUR', 'JPY']
                  .map<DropdownMenuItem<String>>((String value)
{
                return DropdownMenuItem<String>(
                  value: value,
                  child: Text(value, style: TextStyle(fontSize:
20)),
                );
              }).toList(),
            ),
            const SizedBox(
              height: 20,
            ),
            DropdownButton<String>(
              value: _currencyOutput,
              onChanged: _onCurrencyOutputChanged,
              items: const <String>['IDR', 'USD', 'EUR', 'JPY']
                  .map<DropdownMenuItem<String>>((String value)
{
```

```
                return DropdownMenuItem<String>(
                  value: value,
                  child: Text(value, style: TextStyle(fontSize:
20)),
                );
              }).toList(),
            ),
            const SizedBox(
              height: 20,
            ),
            ElevatedButton(
              style: ElevatedButton.styleFrom(
                padding:
                    const EdgeInsets.symmetric(horizontal: 100,
vertical: 20),
                primary: Color.fromRGBO(29, 66, 137, 1),
                textStyle: const TextStyle(
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                ),
              ),
              onPressed: _convert,
              child: Text('Convert'),
            ),
            const SizedBox(
              height: 20,
            ),
            Text(
              _result,
              style: const TextStyle(fontSize: 20),
            ),
          ],
        ),
      ),
    );
  }
}
```

konversi_uang.dart adalah halaman untuk melakukan konversi mata uang yang satu ke mata uang yang lain. Halaman ini memiliki input field yang digunakan oleh user untuk memasukkan jumlah yang akan dikonversi, kemudian akan ada dropdown untuk memilih mata uang awal serta mata uang yang akan dijadikan hasil konversi. Setelah menekan tombol “convert”, maka akan muncul hasil konversi dari mata uang awal ke mata uang hasil.

11.konversi_waktu.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'dart:async';

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

```
  @override
  State<KonversiWaktu> createState() => _KonversiWaktuState();
}

class _KonversiWaktuState extends State<KonversiWaktu> {
  DateTime _currentTime = DateTime.now();
  String _selectedLocation = 'WIB';

  @override
  void initState() {
    super.initState();
    Timer.periodic(Duration(seconds: 1), (Timer timer) {
      setState(() {
        _currentTime = DateTime.now();
      });
    });
  }

  String _formatTime(DateTime time) {
    return DateFormat('EEEE, dd MMMM yyyy -
HH:mm:ss').format(time);
  }

  List<String> locations = ['WIB', 'WITA', 'WIT', 'London'];

  Map<String, Duration> timeOffsets = {
    'WIB': Duration(hours: 7),
    'WITA': Duration(hours: 8),
    'WIT': Duration(hours: 9),
    'London': Duration(hours: 1),
  };

  @override
  Widget build(BuildContext context) {
    return SafeArea(
        child: Scaffold(
          appBar: AppBar(
            title: Text("Konversi Waktu"),
            backgroundColor: Color.fromRGBO(29, 66, 137, 1),
          ),
          body: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Text(
                  '${_selectedLocation}:',
                  style: TextStyle(fontSize: 18),
                ),
                Text(

_formatTime(_currentTime.toUtc().add(timeOffsets[_selectedLocati
on]!)),
                  style: TextStyle(fontSize: 24, fontWeight:
FontWeight.bold),
```

```
                ),
                SizedBox(height: 20),
                DropdownButton(
                  value: _selectedLocation,
                  onChanged: (String? newValue) {
                    setState(() {
                      _selectedLocation = newValue!;
                    });
                  },
                  isDense: true,
                  items:
locations.map<DropdownMenuItem<String>>((String value) {
                    return DropdownMenuItem<String>(
                      value: value,
                      child: Text(value),
                    );
                  }).toList(),
                ),
              ],
            ),
          ),
        )
    );
  }
}
```

konversi_waktu.dart adalah halaman yang digunakan untuk konversi dari salah satu zona waktu ke zona waktu yang lain. Pada halaman ini, terdapat empat zona waktu yang bisa di konversi, yaitu WIB, WITA, WIT, dan London. Patokan waktu yang digunakan adalah UTC, sehingga untuk WIB maka +7UTC, WITA +8UTC, WIT +9UTC, dan London +1UTC.

12.auth_services.dart

```
import 'package:tpm_final_project/helpers/database_helper.dart';
import 'package:tpm_final_project/helpers/shared_pref.dart';
import 'package:tpm_final_project/models/users.dart';

class AuthService {
  final DatabaseHelper databaseHelper = DatabaseHelper.instance;

  Future<bool> register(String username, String password) async
{
    final encryptedPassword = encryptPassword(password);

    User newUser = User(username: username, password:
encryptedPassword);

    int result = await databaseHelper.registerUser(newUser);
    return result > 0;
  }

  Future<bool> login(String username, String password) async {
    final encryptedPassword = encryptPassword(password);
```

```
    User? user = await databaseHelper.loginUser(username,
encryptedPassword);
    if (user != null) {
      await UserPreferences.setLoggedIn(true);
      await UserPreferences.setUsername(user.username!);
      return true;
    } else {
      return false;
    }
  }

  void logout() async {
    await UserPreferences.setLoggedIn(false);
    await UserPreferences.setUsername('');
  }

  String encryptPassword(String password) {
    return password.split('').reversed.join();
  }
}
```

auth_services.dart merupakan halaman untuk melakukan autentikasi terhadap user. Didalam halama ini, dilakukan interaksi dengan database dari SQlite, seperti mengambil data dan menyimpan data ke database. Ada juga beberapa method yang digunakan untuk melakukan login dan register, dimana login dan register tersebut sudah terenkripsi.

13.database_helper.dart

```
import 'package:sqflite/sqflite.dart';
import 'package:path/path.dart';
import 'package:tpm_final_project/models/users.dart';

class DatabaseHelper {
  static final DatabaseHelper instance =
DatabaseHelper._getInstance();
  static Database? _database;

  DatabaseHelper._getInstance();

  Future<Database> get database async {
    if (_database != null) return _database!;

    _database = await _initDatabase();
    return _database!;
  }

  Future<Database> _initDatabase() async {
    String path = await getDatabasesPath();
    path = join(path, 'app.db');

    return await openDatabase(
      path,
      version: 1,
      onCreate: (db, version) async {
        await db.execute(
```

```
          '''
          CREATE TABLE users (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            username TEXT,
            password TEXT
          )
          ''',
        );
      },
    );
  }

  Future<int> registerUser(User user) async {
    final db = await database;
    return await db.insert('users', user.toMap());
  }

  Future<User?> loginUser(String username, String password)
async {
    final db = await database;
    final result = await db.query(
      'users',
      where: 'username = ? AND password = ?',
      whereArgs: [username, password],
    );

    if (result.isNotEmpty) {
      return User.fromMap(result.first);
    } else {
      return null;
    }
  }
}
```

database_helper.dart adalah helper yang bertanggung jawab untuk berinteraksi dengan database. Dari helper ini, banyak sekali al yang dilakukan, seperti mengelola koneksi dan operasi database, menyimpan dan membuka database, serta membuat table untuk menyimpan data.

14.shared_pref.dart

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

class UserPreferences {
  static const String kLoggedInKey = 'isLoggedIn';
  static const String kUsernameKey = 'username';

  static Future<bool> get isLoggedIn async {
    final prefs = await SharedPreferences.getInstance();
    return prefs.getBool(kLoggedInKey) ?? false;
  }

  static Future<String?> get username async {
    final prefs = await SharedPreferences.getInstance();
    return prefs.getString(kUsernameKey);
```

```
  }

  static Future<void> setLoggedIn(bool isLoggedIn) async {
    final prefs = await SharedPreferences.getInstance();
    await prefs.setBool(kLoggedInKey, isLoggedIn);
  }

  static Future<void> setUsername(String username) async {
    final prefs = await SharedPreferences.getInstance();
    await prefs.setString(kUsernameKey, username);
  }

  static Future<String> getUsername() async {
  SharedPreferences prefs = await
SharedPreferences.getInstance();
  return prefs.getString(kUsernameKey) ?? '';
}

  static Future<void> clear() async {
    final prefs = await SharedPreferences.getInstance();
    await prefs.clear();
  }
}
```

shared_pref.dart digunakan untuk menyimpan dan mengambil data user, mengelola data user dengan shared preferences, serta menetukan session dari user.

15.weapon_skin.dart

```
class WeaponSkin {
  final String? uuid;
  final String? displayName;
  final String? category;
  final String? themeUuid;
  final String? contentTierUuid;
  final String? displayIcon;
  final String? fullRender;
  final String? killStreamIcon;
  final String? streamedVideo;
  final List<dynamic>? chromas;
  final List<dynamic>? levels;

  WeaponSkin({
    this.uuid,
    this.displayName,
    this.category,
    this.themeUuid,
    this.contentTierUuid,
    this.displayIcon,
    this.fullRender,
    this.killStreamIcon,
    this.streamedVideo,
    this.chromas,
    this.levels,
  });
```

```
  factory WeaponSkin.fromJson(Map<String, dynamic> json) {
    return WeaponSkin(
      uuid: json['uuid'],
      displayName: json['displayName'],
      category: json['category'],
      themeUuid: json['themeUuid'],
      contentTierUuid: json['contentTierUuid'],
      displayIcon: json['displayIcon'],
      fullRender: json['fullRender'],
      killStreamIcon: json['killStreamIcon'],
      streamedVideo: json['streamedVideo'],
      chromas: json['chromas'],
      levels: json['levels'],
    );
  }
}
```

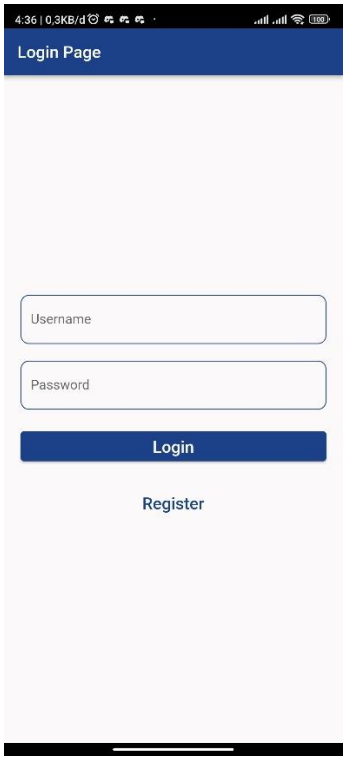
weapon_skin.dart merupakan sebuah model yang dibuat berdasarkan response dari API yang digunakan.

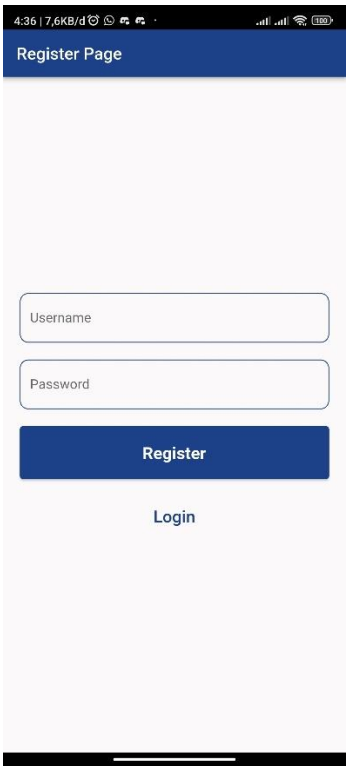
16.users.dart

```
class User {
  int? Id;
  String? Username;
  String? Password;

  User({this.id, this.username, this.password});

  Map<String, dynamic> toMap() {
    return {
      'id': id,
      'username': username,
      'password': password,
    };
  }

  factory User.fromMap(Map<String, dynamic> map) {
    return User(
      id: map['id'],
      username: map['username'],
      password: map['password'],
    );
  }
}
```

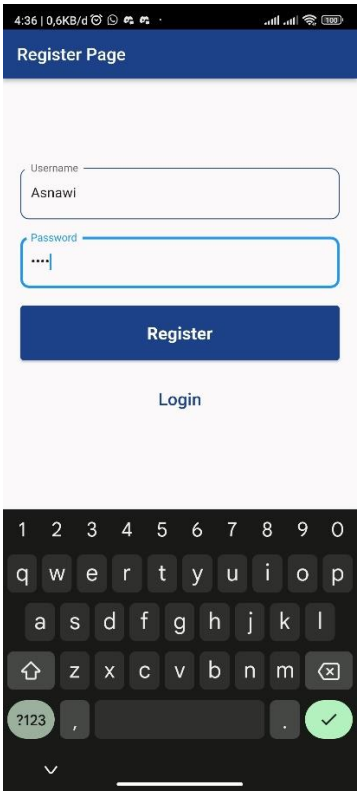
users.dart digunakan untuk mempresentasikan user dalam aplikasi.

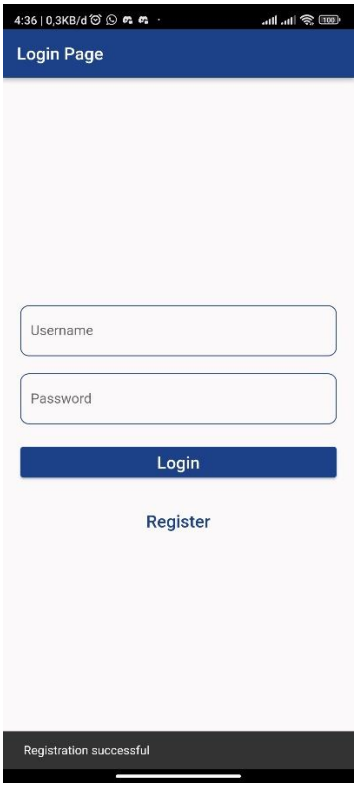
SCREENSHOT PROGRAM



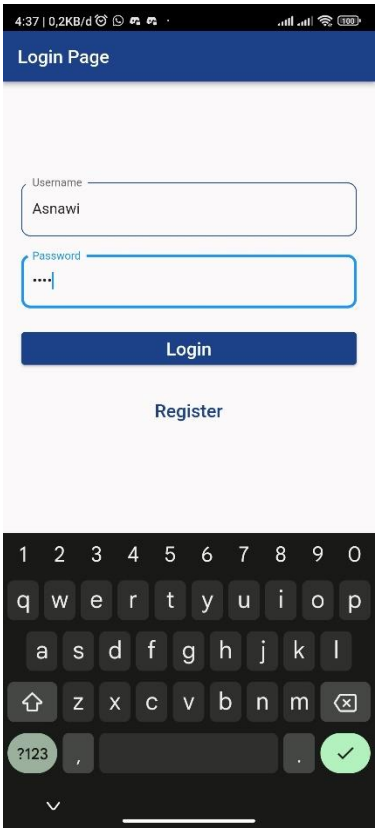


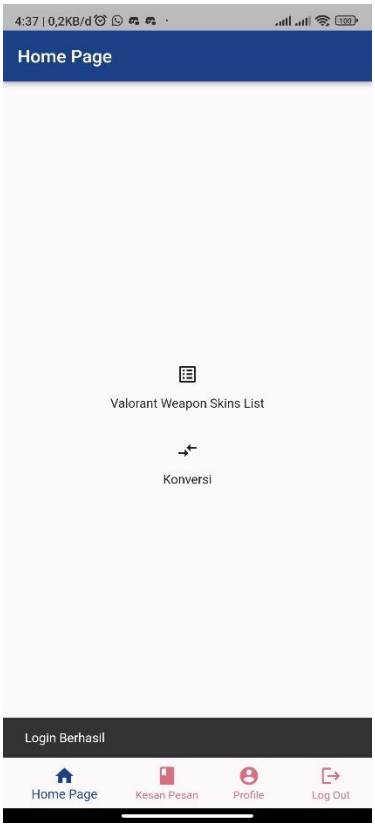
Berikut merupakan tampilan dari halaman Login dan Register





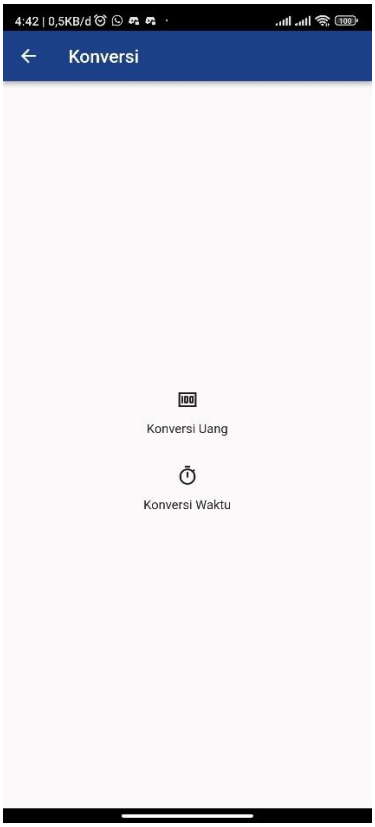
Ketika register, user dapat memasukkan username dan password sesuai keingina. Ketika register berhasil, maka akan muncul snackbar yang bertuliskan “Registration succesfull”.



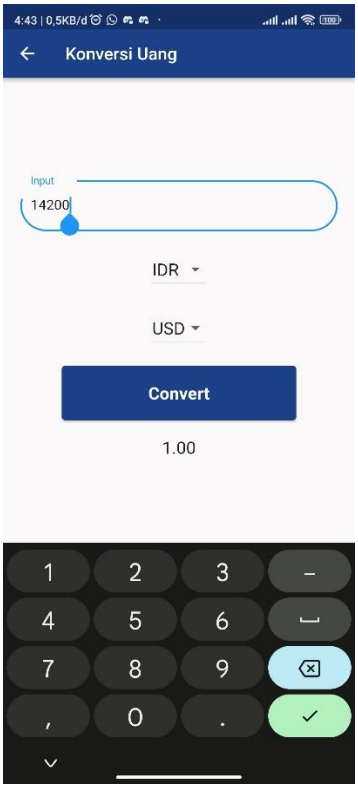


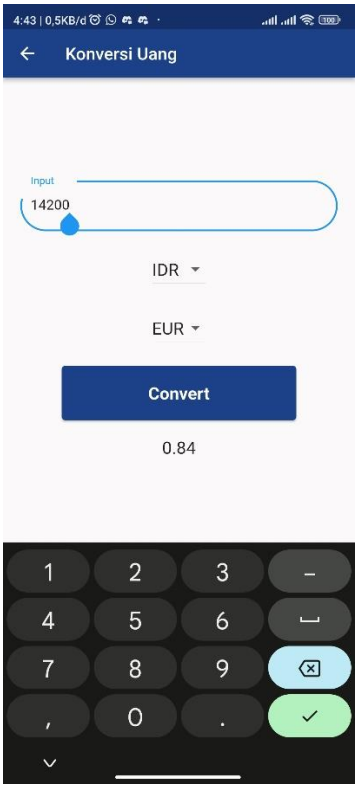
Ketika sudah memiliki akun, maka user bisa login menggunakan username dan password yang sudah dibuat sebelumnya. Ketika login berhasil, maka akan masuk ke halaman Homepage dan akan muncul snackbar yang bertuliskan “Login berhasil”. Di halaman Homepage, kita bisa memilih menu Valorant Weapon Skin List atau halaman Konversi.

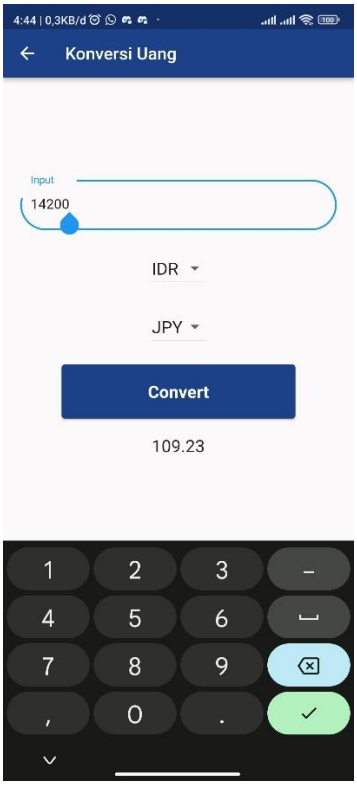
Keika kita memilih halaman Valorant Weapon Skin List, maka user akan diberikan list dari semua senjata beserta skin yang ada di dalam game Valorant. Kemudian, ketika user menekan salah satu skin, maka akan ditampilkan detail dari skin tersebut, dimana detail yang ditampilkan adalah beberapa varian dari skin senjata tersebut (jika skin nya memang memiliki beberapa varian. Jika tidak, maka akan menampilkan gambar seperti pada halaman list).



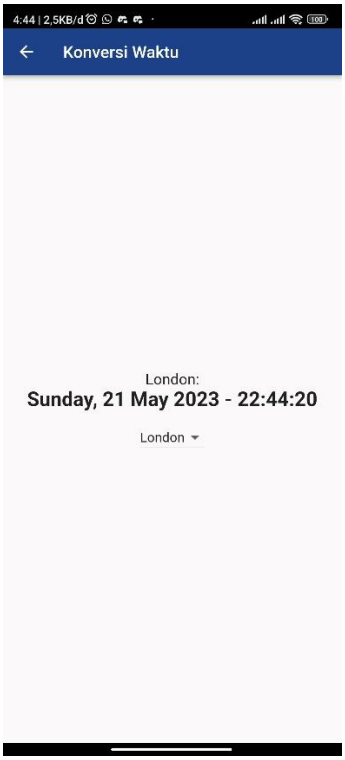
Ketika memilih halaman konversi, maka user dapat memilih ingin melakukan konversi apa. Di halaman ini, disediakan dua jenis konversi, yaitu konversi uang dan konversi waktu.



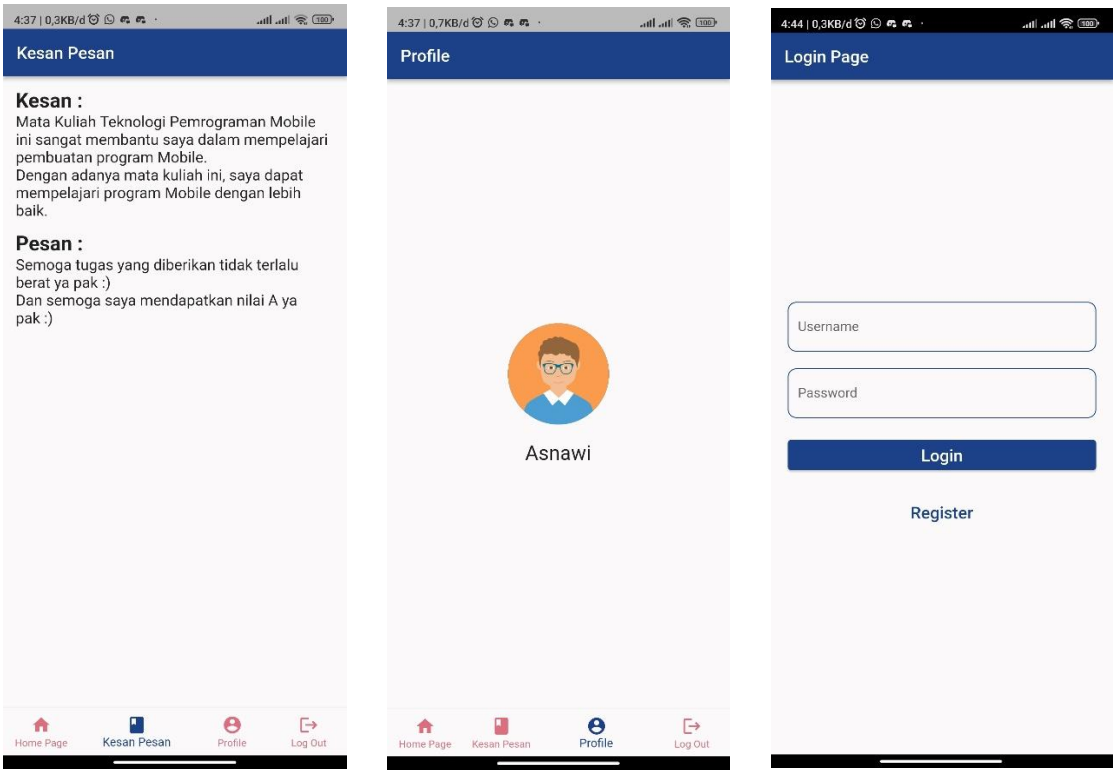




Ketika memilih konversi uang, maka user dapat melakukan konversi salah satu mata uang yang sudah ada, ke mata uang yang lainnya.



Ketika memilih konversi waktu, maka user dapat melakukan konversi zona waktu di sesuai dengan zona waktu yang sudah ditentukan.



User dapat memilih beberapa menu yang ada di bottom navbar, seperti halaman Kesan Pesan yang menapilkan kesan dan pesan selama mengikuti matakuliah Teknologi & Pemrograman Mobile, serta halaman Profile yang menampilkan data dari user, berupa foto dan username. Jika ingin keluar dari akun, user dapat menekan tombol Logout yang berada di bottom navbar, dan akan diarahkan ke halaman Login kembali.