

Tahun Ajar 2022/2023

PEMROGRAMAN

PERANGKAT BERGERAK

Praktikum 6 :
Firebase Authentication

Pengembang Modul :

Novian Adi Prasety, S.Kom., M.Kom.

Muhammad Lulu Latif Usman, S.Pd., M.Han.

Muhamad Azrino Gustalika, S.Kom., M. Tr. T

Akhmad Nur Alamsyah (Asisten Praktikum)

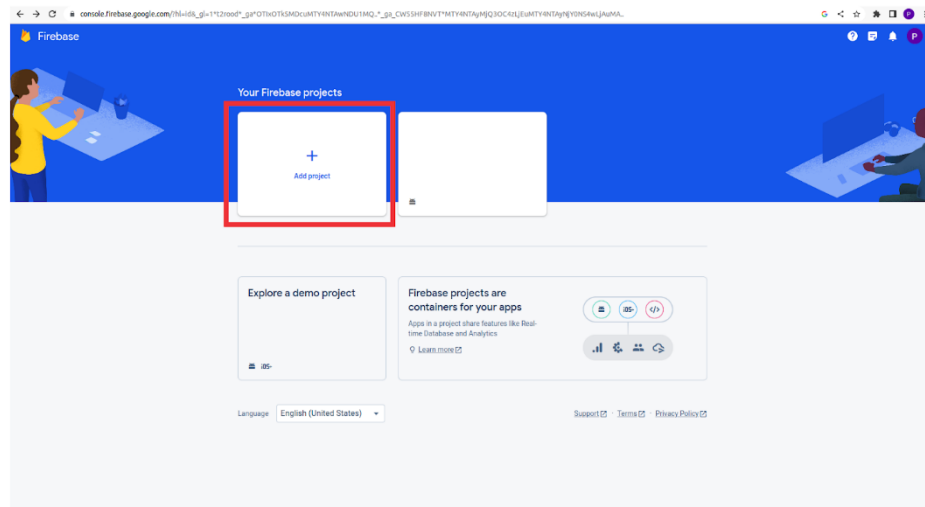
Puspita Kartika Sari (Asisten Praktikum)

Dalam modul ini, kita akan membahas langkah-langkah untuk mengintegrasikan Firebase ke dalam aplikasi Flutter Anda. Firebase adalah platform pengembangan aplikasi yang kuat yang dapat membantu Anda membangun, meningkatkan, dan mengelola aplikasi mobile dengan mudah.

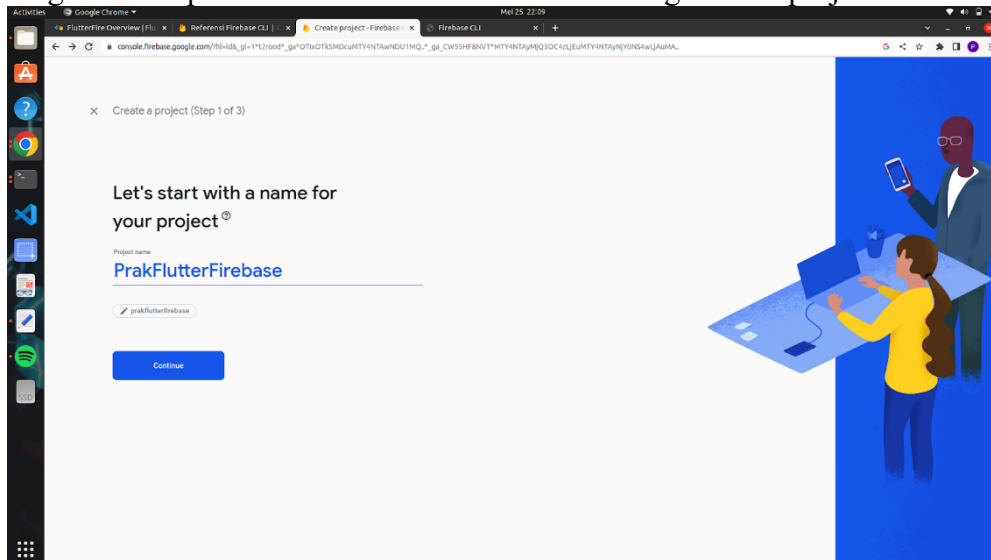
Dengan menggunakan Firebase, Anda dapat memanfaatkan berbagai fitur yang ditawarkannya, seperti penyimpanan data real-time, otentikasi pengguna, analitik, notifikasi, dan masih banyak lagi. Integrasi Firebase ke dalam aplikasi Flutter akan membuka peluang baru dalam pengembangan aplikasi yang lebih interaktif dan canggih.

Cara menambahkan firebase ke project aplikasi kita :

1. Langkah pertama login ke firebase, apabila belum memiliki akun maka perlu membuat akun terlebih dahulu melalui [link berikut](#).
2. Langkah kedua install firebase CLI dengan panduan pada [link berikut](#).
3. Langkah ketiga kita akan membuat project pada firebase, dengan klik “add project” yang ada di web firebase.



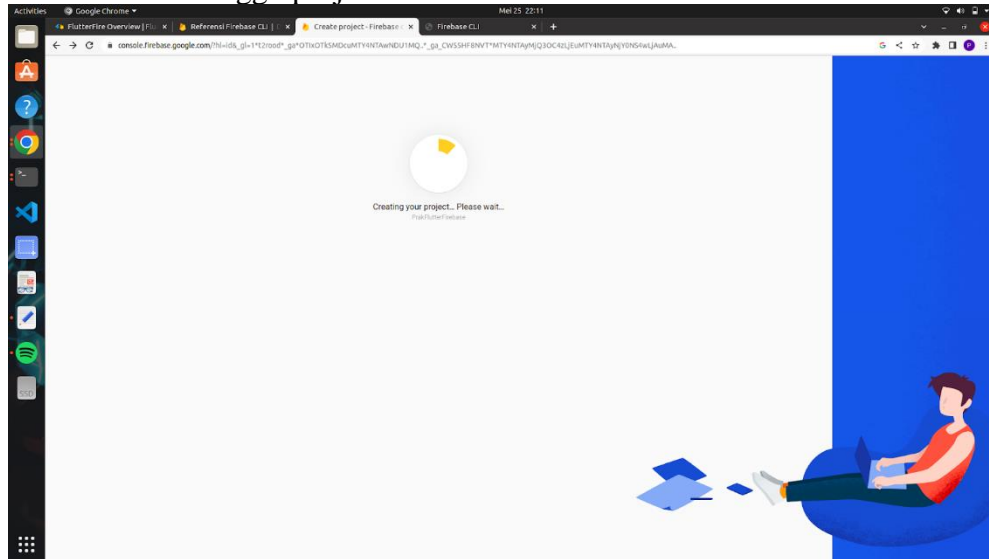
4. Langkah keempat isikan “PrakFlutterFirebase” sebagai nama project



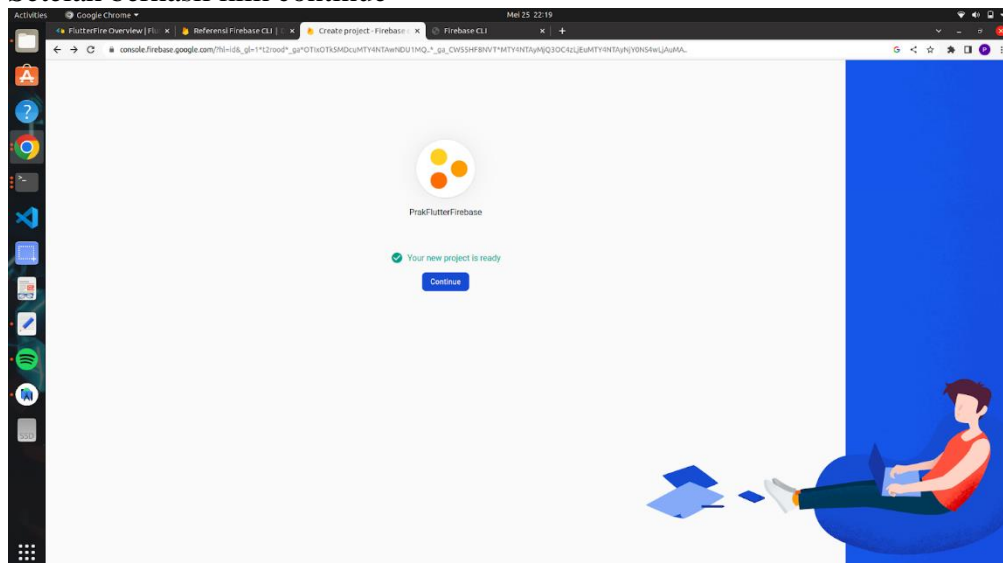
-

- [illegible]

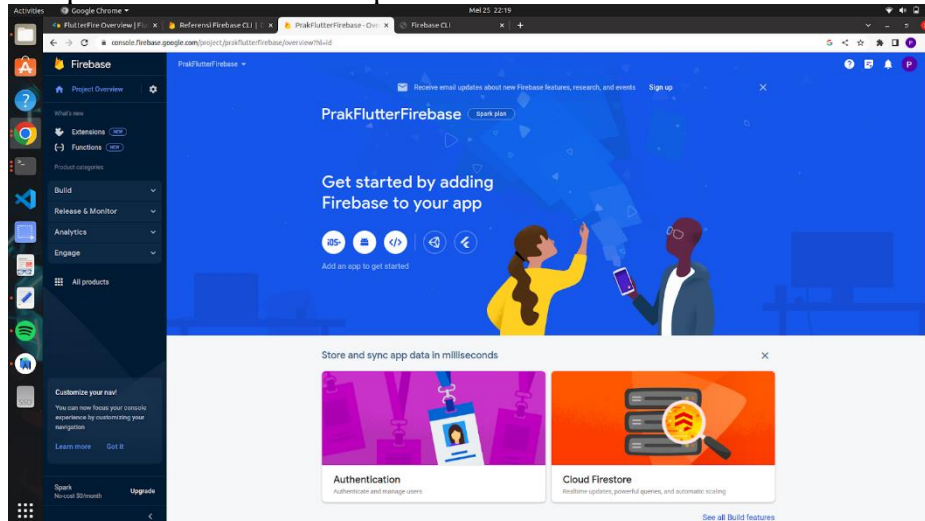
7. Kemudian menunggu project dibuat



8. Setelah berhasil klik continue

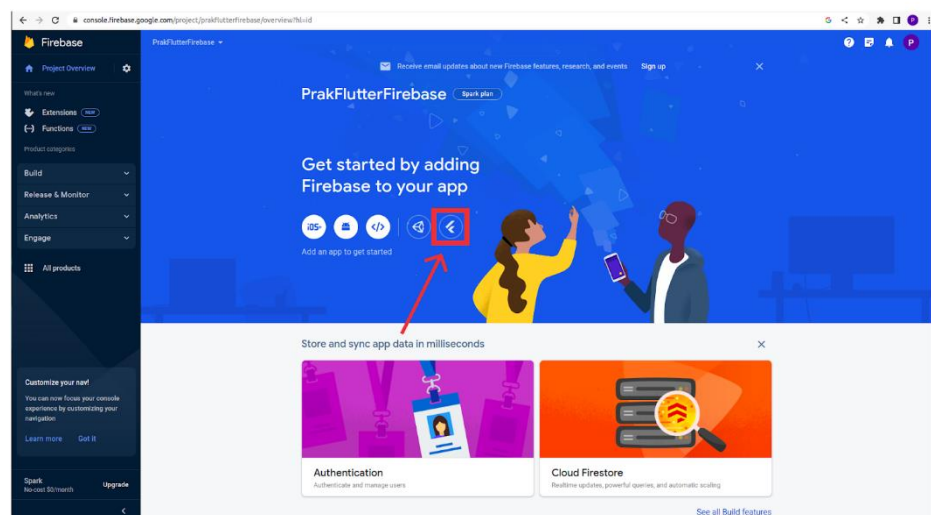


9. Tampilan dashboard akan seperti berikut:

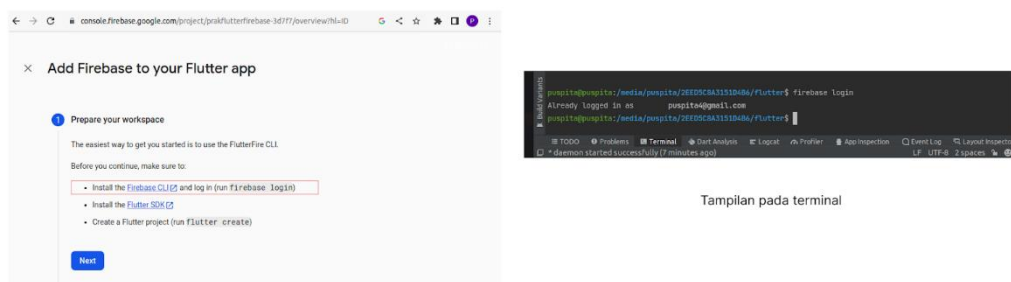


10. Menyambungkan project firebase yang sudah kita buat dengan project flutter

- Buat flutter project dengan nama “praktikum6”
- Klik ikon flutter yang ada pada dashboard

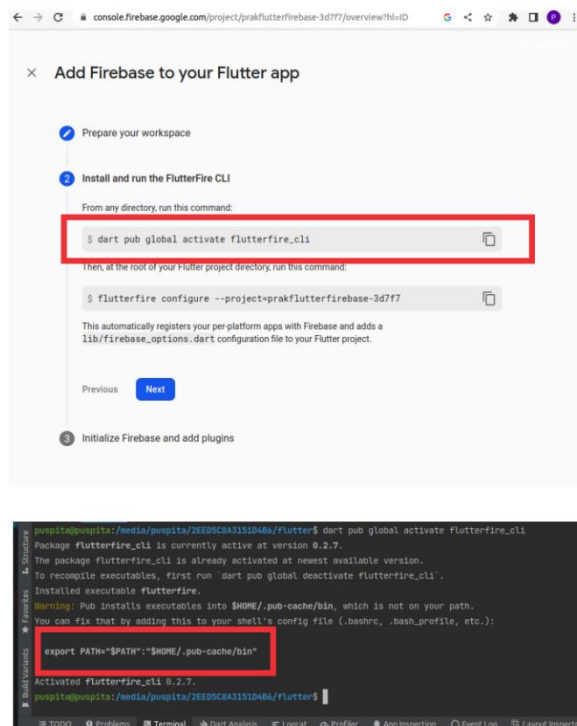


- Ketik “firebase login” pada terminal/cmd.

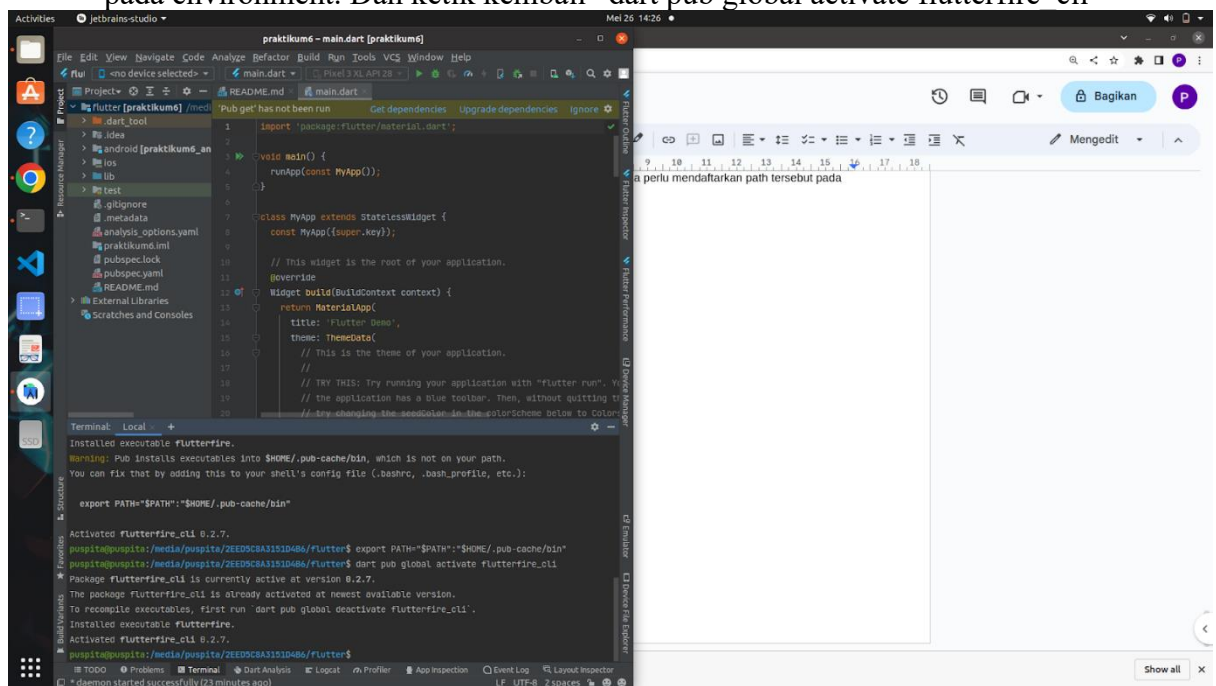


Tampilan pada terminal

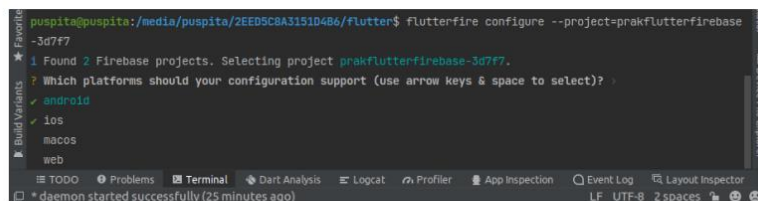
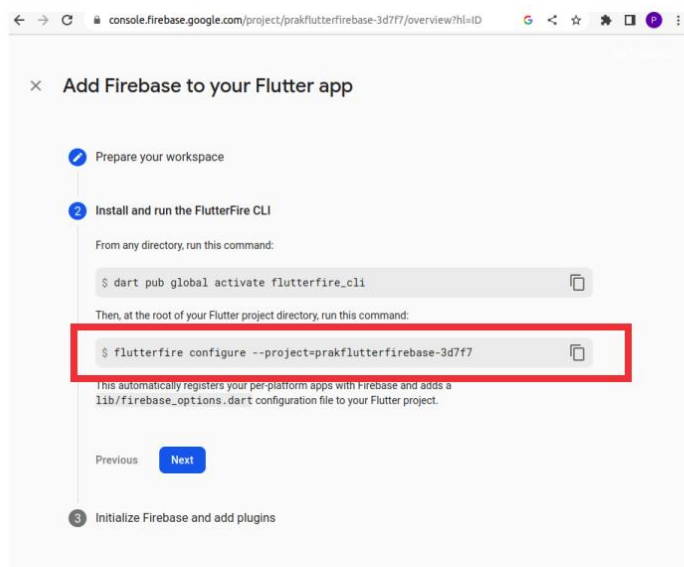
- Ketik “dart pub global activate flutterfire_cli” pada terminal.



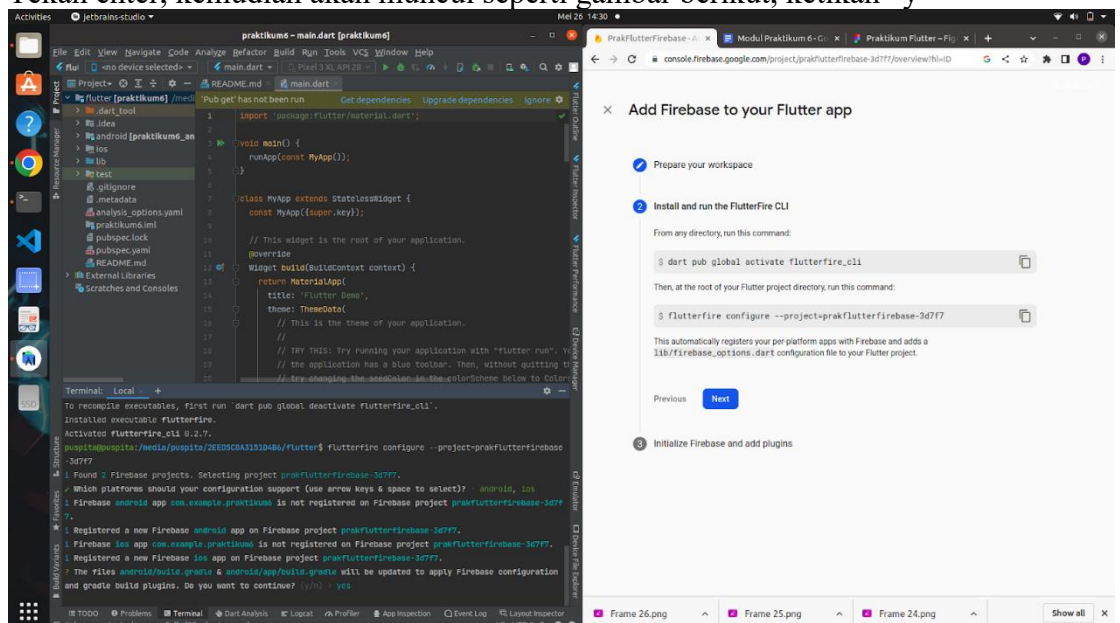
- Terdapat error pada kotak merah, sehingga perlu mendaftarkan path tersebut pada environment. Dan ketik kembali “dart pub global activate flutterfire cli”



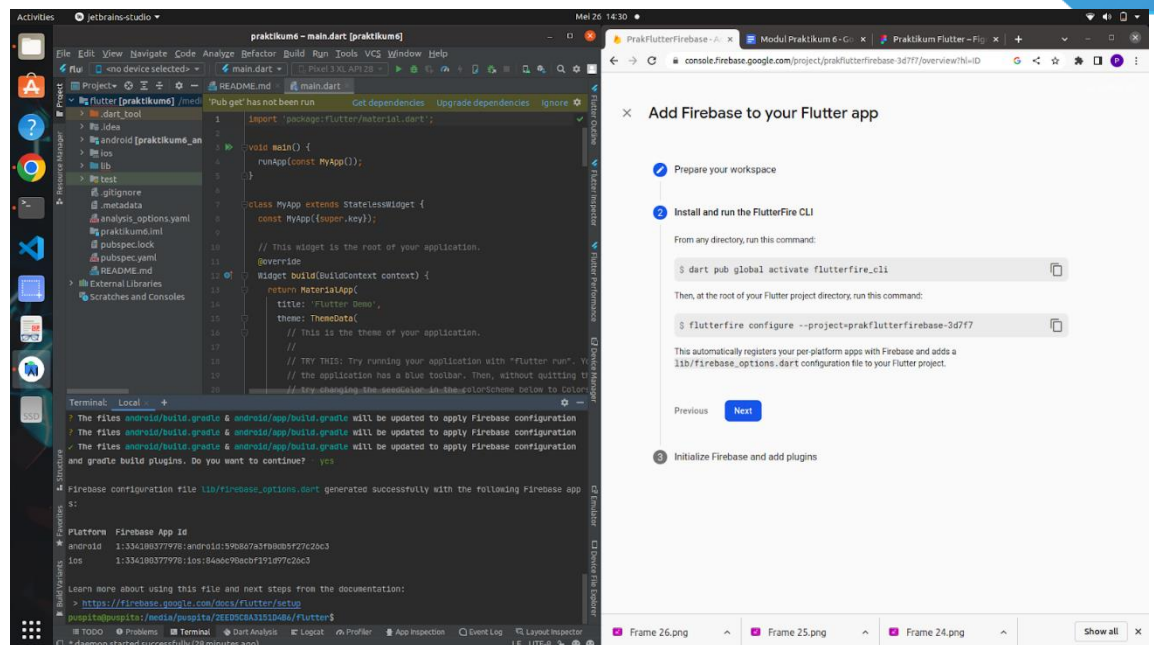
- Klik “flutterfire configure --project=prakflutterfirebase-3d7f7” pada terminal



- Tekan enter, kemudian akan muncul seperti gambar berikut, ketikan “y”



- Maka kita akan mendapatkan app id



11. Langkah kedelapan kita perlu menginstal dependensi

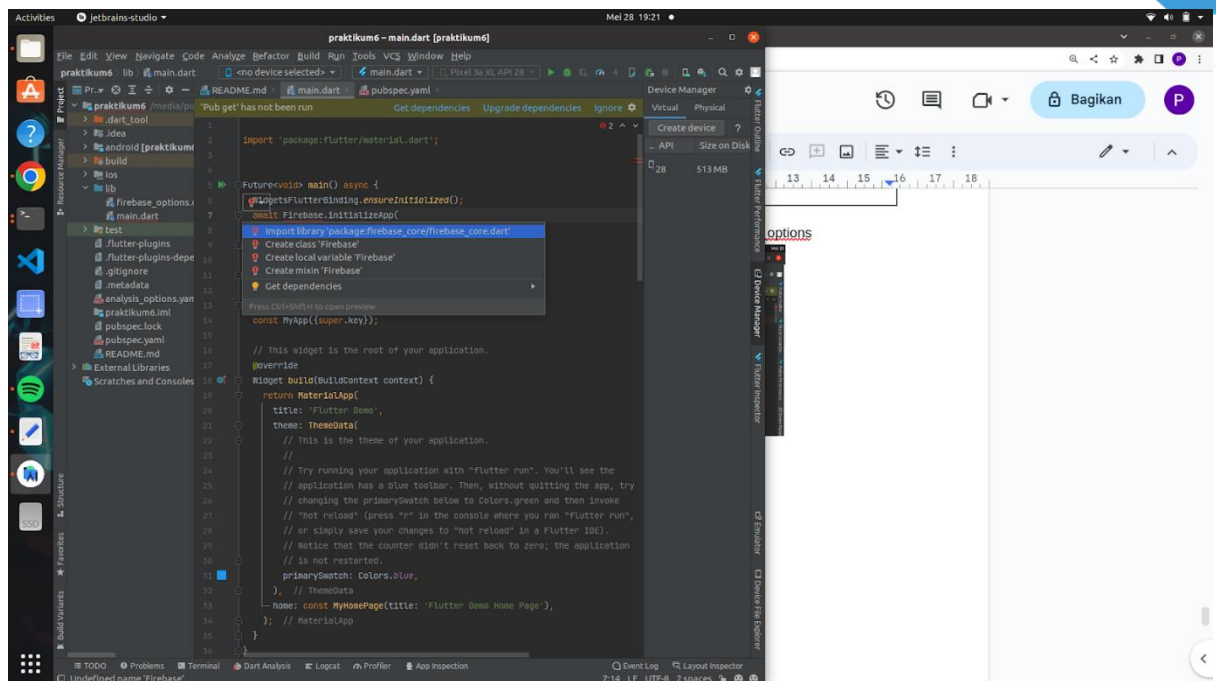
- Ketik “flutter pub add firebase_core” pada terminal

```
puspita@puspita:/media/puspita/2EED5C8A3151D4B6/flutter$ flutter pub add firebase_core
Resolving dependencies...
  collection 1.17.1 (1.17.2 available)
+ firebase_core 2.13.0
+ firebase_core_platform_interface 4.8.0
+ firebase_core_web 2.5.0
+ flutter_web_plugins 0.0.0 from sdk flutter
  matcher 0.12.15 (0.12.16 available)
  material_color_utilities 0.2.0 (0.5.0 available)
+ plugin_platform_interface 2.1.4
  source_span 1.9.1 (1.10.0 available)
  test_api 0.5.1 (0.6.0 available)
Changed 5 dependencies!
puspita@puspita:/media/puspita/2EED5C8A3151D4B6/flutter$
```

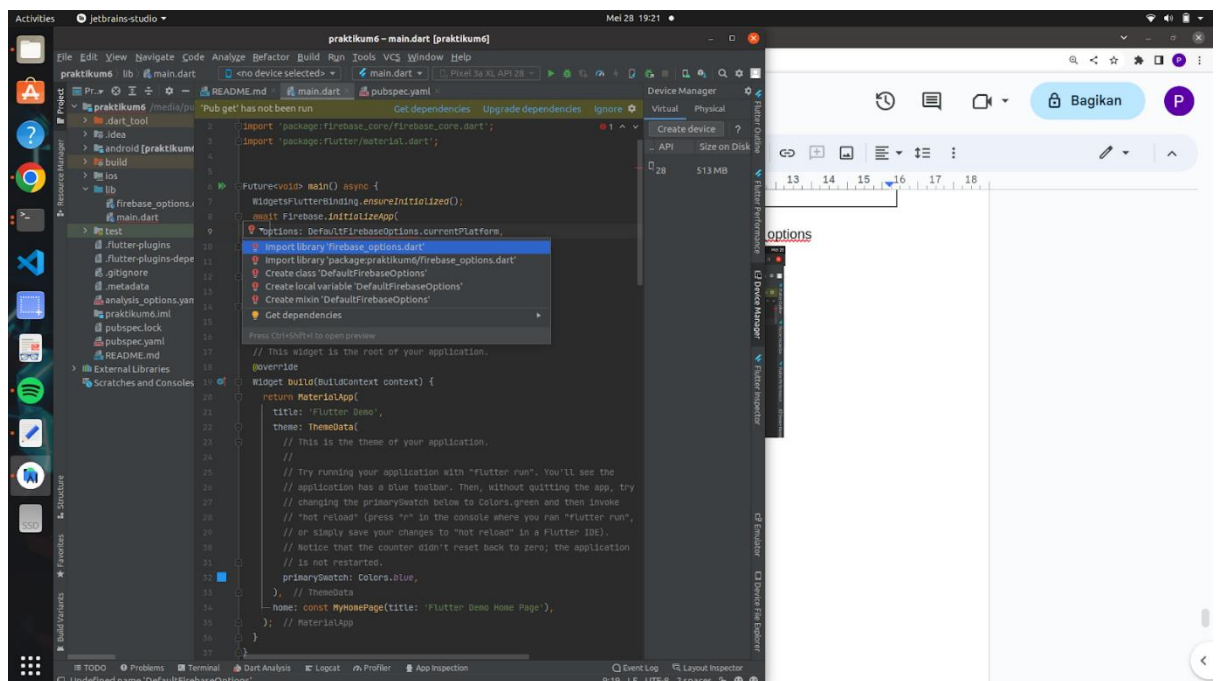
- Ubah kode program pada file main.dart seperti berikut

Sebelum	Sesudah
<pre>void main() { runApp(const MyApp()); }</pre>	<pre>Future<void> main() async { WidgetsFlutterBinding.ensureInitialized(); await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform,); runApp(const MyApp()); }</pre>

- Selanjutnya kita perlu import library firebase_core dan firebase_options



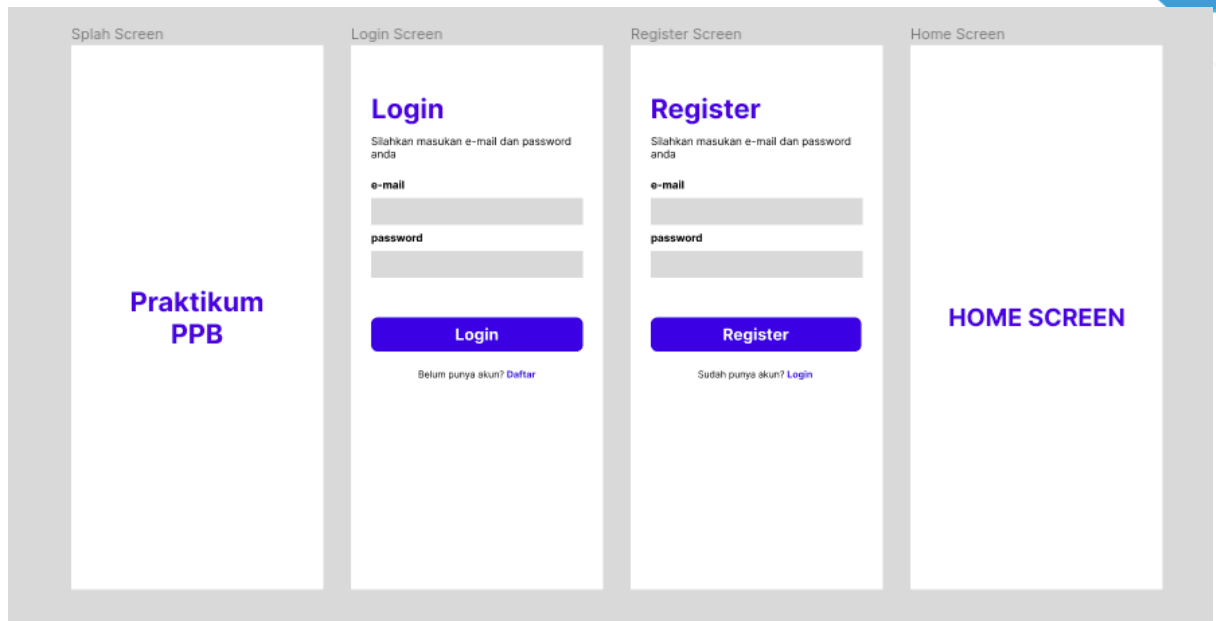
Library firebase_core



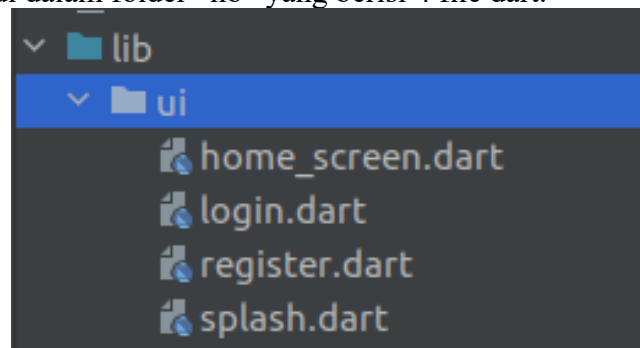
Library firebase_options

Membuat Login & Register dengan firebase authentication :

1. Slicing UI berikut :



2. Buat folder "ui" di dalam folder "lib" yang berisi 4 file dart.



splash.dart

```
import 'dart:async';

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

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

  @override
  State<SplashScreen> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
    Timer(Duration(seconds: 3),
      () => Navigator.pushNamed(context, '/login')
    );
    super.initState();
  }
}
```

```
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          Text("Praktikum",
            style: TextStyle(
              fontSize: 40,
              fontWeight: FontWeight.bold,
              color: Color(0xff3D4DE0)
            ),
          ),
          Text("PPB",
            style: TextStyle(
              fontSize: 40,
              fontWeight: FontWeight.bold,
              color: Color(0xff3D4DE0)
            ),
          ),
        ],
      ),
    ),
  );
}
```

login.dart

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

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

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final emailEdc = TextEditingController();
  final passEdc = TextEditingController();
  bool passInvisible = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        margin: EdgeInsets.symmetric(horizontal: 30, vertical: 70),
        child: ListView(
          children: [
            Text("Login", style: TextStyle(
              fontSize: 40, fontWeight: FontWeight.bold, color:
Color(0xff3D4DE0)
            ),),
            SizedBox(height: 15,),
            Text("Silahkan masukan e-mail dan password anda", style:
TextStyle(
              fontSize: 16,
            ),),
          ],
        ),
      ),
    );
  }
}
```

```
        SizedBox(height: 25,),
        Text("e-mail", style: TextStyle(
          fontSize: 16, fontWeight: FontWeight.bold
        )),
        TextFormField(
          controller: emailEdc,
        ),
        SizedBox(height: 10,),
        Text("password", style: TextStyle(
          fontSize: 16, fontWeight: FontWeight.bold
        )),
        TextFormField(
          controller: passEdc,
          decoration: InputDecoration(

            suffixIcon: IconButton(
              icon: Icon(passInvisible ? Icons.visibility :
Icons.visibility_off),
              onPressed: () {
                setState(() {
                  passInvisible = !passInvisible;
                });
              },
            ),
            obscureText: !passInvisible,
          ),
        ),
        SizedBox(height: 50,),
        ElevatedButton(onPressed: (){
          Navigator.pushNamed(context, '/home');
        },
          style: ElevatedButton.styleFrom(
            backgroundColor: Color(0xff3D4DE0),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(10)
            )
          ),
          child: Text("Login", style: TextStyle(
            fontWeight: FontWeight.bold, fontSize: 24, color:
Colors.white
          )),
        ),
        SizedBox(height: 25,),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text("Belum punya akun ?"),
            TextButton(onPressed: (){
              Navigator.pushNamed(context, '/register');
            },
              child: Text("Daftar", style: TextStyle(
                fontWeight: FontWeight.bold,
                color: Color(0xff3D4DE0)
              )),
            ],
        ),
      ],
    ),
  ),
),
);
```

```
}  
}
```

register.dart

```
import 'package:flutter/material.dart';  
  
class RegisterScreen extends StatefulWidget {  
  const RegisterScreen({Key? key}) : super(key: key);  
  
  @override  
  State<RegisterScreen> createState() => _RegisterScreenState();  
}  
  
class _RegisterScreenState extends State<RegisterScreen> {  
  final emailEdc = TextEditingController();  
  final passEdc = TextEditingController();  
  bool passInvisible = false;  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Container(  
        margin: EdgeInsets.symmetric(horizontal: 30, vertical: 70),  
        child: ListView(  
          children: [  
            Text("Register", style: TextStyle(  
              fontSize: 40, fontWeight: FontWeight.bold, color:  
Color(0xff3D4DE0)  
            ),),  
            SizedBox(height: 15,),  
            Text("Silahkan masukan e-mail dan password anda", style:  
TextStyle(  
              fontSize: 16,  
            ),),  
            SizedBox(height: 25,),  
            Text("e-mail", style: TextStyle(  
              fontSize: 16, fontWeight: FontWeight.bold  
            ),),  
            TextFormField(  
              controller: emailEdc,  
            ),  
            SizedBox(height: 10,),  
            Text("password", style: TextStyle(  
              fontSize: 16, fontWeight: FontWeight.bold  
            ),),  
            TextFormField(  
              controller: passEdc,  
              decoration: InputDecoration(  
                suffixIcon: IconButton(  
                  icon: Icon(passInvisible ? Icons.visibility :  
Icons.visibility_off),  
                  onPressed: () {  
                    setState(() {  
                      passInvisible = !passInvisible; // Toggle  
_isPasswordVisible ketika ikon mata ditekan  
                    });  
                  },  
                ),  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}
```

```
        obscureText: !passInvisible, // Atur obscureText
berdasarkan _isPasswordVisible
    ),
    SizedBox(height: 50,),
    ElevatedButton(onPressed: (){},
        style: ElevatedButton.styleFrom(
            backgroundColor: Color(0xff3D4DE0),
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10)
            )
        ),
        child: Text("Register", style: TextStyle(
            fontWeight: FontWeight.bold, fontSize: 24, color:
Colors.white
        )),),
    SizedBox(height: 25,),
    Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            Text("Sudah punya akun ?"),
            TextButton(onPressed: (){
                Navigator.pushNamed(context, '/login');
            },
                child: Text("Login", style: TextStyle(
                    fontWeight: FontWeight.bold,
                    color: Color(0xff3D4DE0)
                )),)
        ],
    ),
),
),
),
);
}
```

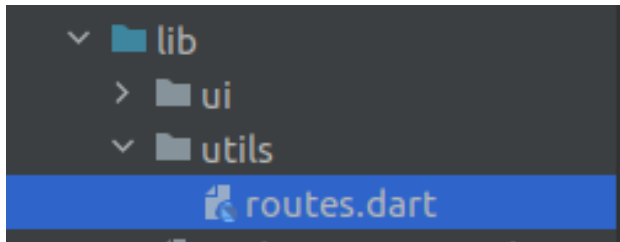
home screen.dart

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Text("HOME SCREEN",
            style: TextStyle(
              fontSize: 40,
              fontWeight: FontWeight.bold,
              color: Color(0xff3D4DE0)
            )),
      ),
    ),
  );
}
```


3. Selanjutnya kita perlu mendaftarkan state yang sudah kita buat dalam route dengan cara buat folder “utils” di dalam folder “lib”, dan buat file “routes.dart” di dalam folder “utils”.



routes.dart

```
import 'package:flutter/material.dart';
import 'package:praktikum6/ui/home_screen.dart';
import 'package:praktikum6/ui/login.dart';
import 'package:praktikum6/ui/register.dart';

MaterialPageRoute _pageRoute(
  {required Widget body, required RouteSettings settings}) =>
  MaterialPageRoute(builder: (_) => body, settings: settings);

Route? generateRoute(RouteSettings settings) {
  Route? _route;
  final _args = settings.arguments;

  switch (settings.name) {
    case rLogin:
      _route = _pageRoute(body: LoginScreen(), settings: settings);
      break;
    case rRegister:
      _route = _pageRoute(body: RegisterScreen(), settings: settings);
      break;
    case rHome:
      _route = _pageRoute(body: HomeScreen(), settings: settings);
      break;
  }
  return _route;
}

final NAV_KEY = GlobalKey<NavigatorState>();
const String rLogin = '/login';
const String rRegister = '/register';
const String rHome = '/home';
```

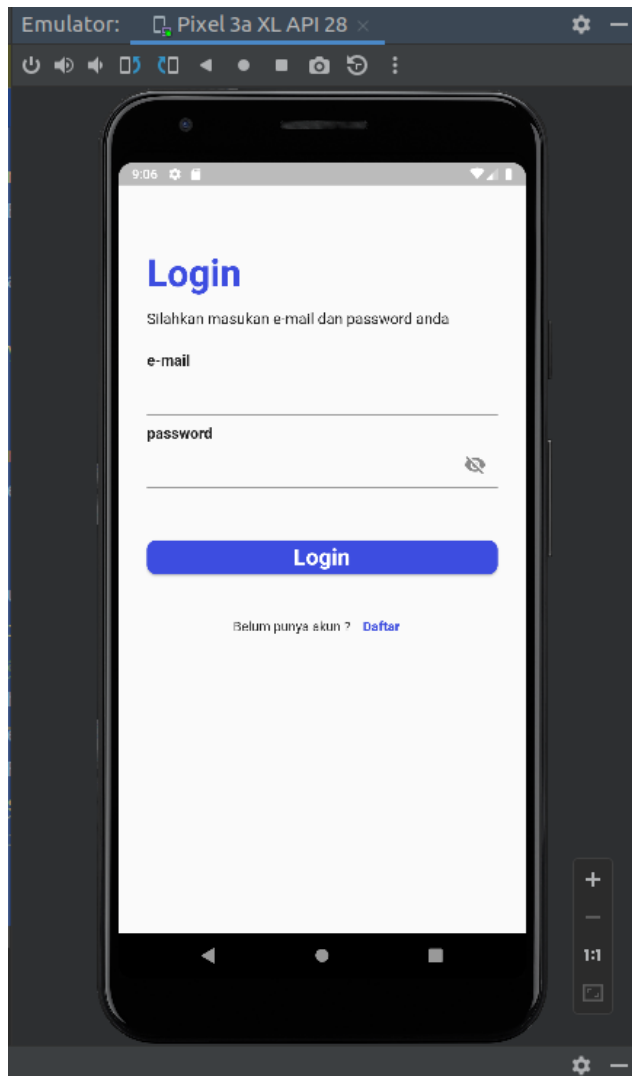
4. Selanjutnya mendeklarasikan main.dart

main.dart

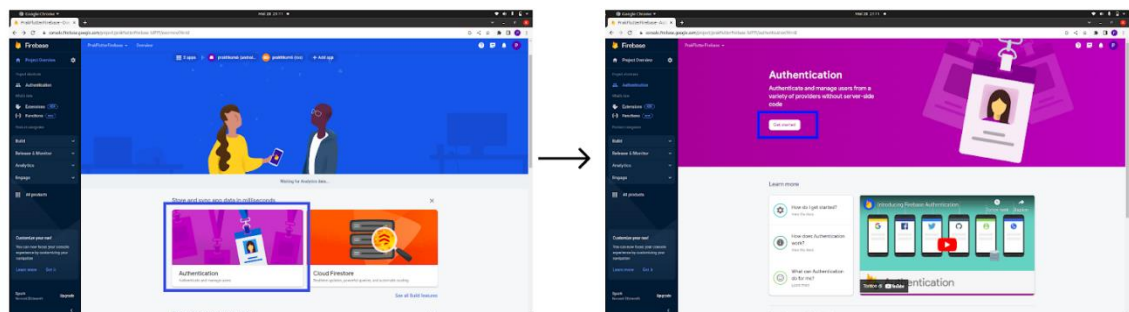
```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:praktikum6/ui/splash.dart';
import 'package:praktikum6/utils/routes.dart';
import 'package:firebase_options.dart';
```

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp(  
    options: DefaultFirebaseOptions.currentPlatform,  
  );  
  runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: "Praktikum 6",  
      debugShowCheckedModeBanner: false,  
      navigatorKey: NAV_KEY,  
      onGenerateRoute: generateRoute,  
      home: SplashScreen(),  
    );  
  }  
}
```

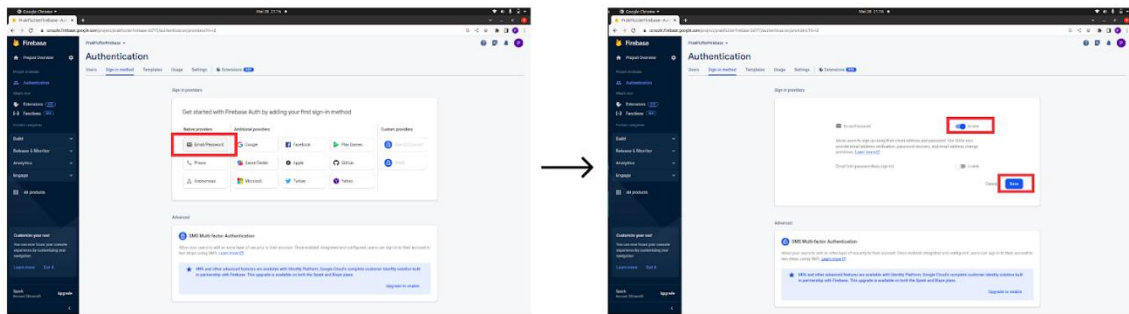
5. Run project pada emulator



6. Masuk ke dashboard yang sudah kita buat dan aktifkan autentifikasinya



7. Selanjutnya pilih email/password



8. Untuk menggunakan firebase Authentication perlu menambahkan dependensi

“`firebase_auth: ^4.3.0`” pada `pubspec.yaml`

9. Untuk memproses logika bisnis pada aplikasi kita akan menggunakan flutter bloc, sehingga perlu menambahkan 2 dependensi yaitu :

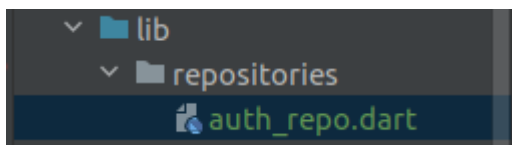
`bloc: ^8.1.2`

`flutter_bloc: ^8.1.3`

jangan lupa untuk menjalankan perintah “flutter pub upgrade”

```
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  
  firebase_core: ^2.8.0  
  firebase_auth: ^4.3.0  
  bloc: ^8.1.2  
  flutter_bloc: ^8.1.3
```

10. Buat folder “repositories” pada folder “lib”, kemudian buat file “auth_repo.dart”. folder repositories ini akan kita gunakan untuk berkomunikasi dengan firebase/API dari back end.



11. Kita akan membuat 2 buah Future<void> dalam class AuthRepo untuk login dan register pada file auth_repo.dart. Untuk penjelasan lebih lanjut tentang penggunaan firebase_auth bisa dilihat pada dokumentasi berikut :

[dokumentasi 1](#)

[dokumentasi 2](#)

auth_repo.dart

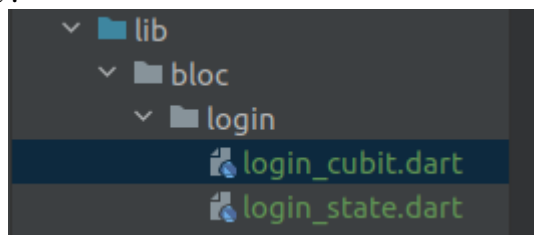
```
import 'package:firebase_auth/firebase_auth.dart';
```

```
class AuthRepo {
  final _auth = FirebaseAuth.instance;

  Future<void> login({required String email, required String password})
  async {
    try {
      final user = await _auth.signInWithEmailAndPassword(
        email: email, password: password);
    } on FirebaseException catch (e) {
      throw e.message ?? 'Something wrong!';
    } catch (e) {
      throw e;
    }
  }

  Future<void> register(
    {required String email, required String password}) async {
    try {
      final user = await _auth.createUserWithEmailAndPassword(
        email: email, password: password);
    } on FirebaseException catch (e) {
      throw e.message ?? 'Something wrong!';
    } catch (e) {
      throw e;
    }
  }
}
```

12. Kemudian kita akan mengatur bloc untuk login. Buat folder dengan struktur seperti berikut :



login_state.dart

```
part of 'login_cubit.dart';

@immutable
abstract class LoginState {}

class LoginInitial extends LoginState {}

class LoginLoading extends LoginState {}

class LoginSuccess extends LoginState {
  final String msg;

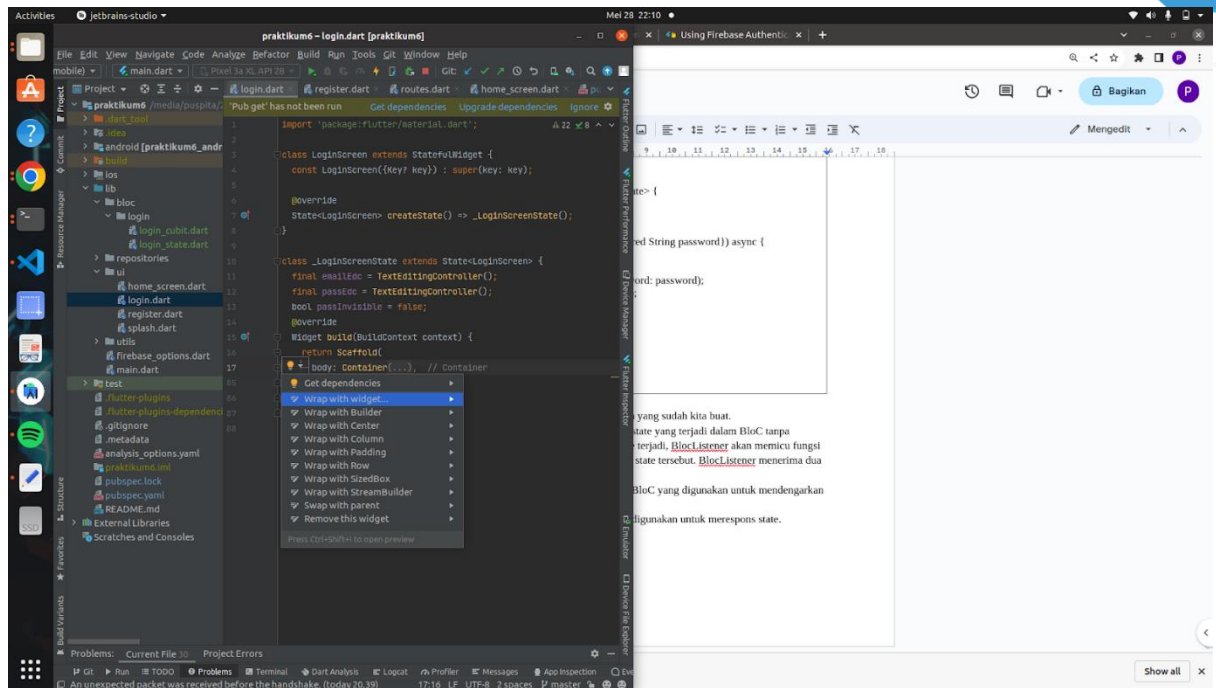
  LoginSuccess(this.msg);
}
```

```
}  
  
class LoginFailure extends LoginState {  
  final String msg;  
  
  LoginFailure(this.msg);  
}
```

login_cubit.dart

```
import 'package:bloc/bloc.dart';  
  
import 'package:meta/meta.dart';  
  
import '../repositories/auth_repo.dart';  
  
part 'login_state.dart';  
  
class LoginCubit extends Cubit<LoginState> {  
  LoginCubit() : super(LoginInitial());  
  final _repo = AuthRepo();  
  
  void login({required String email, required String password}) async {  
    emit(LoginLoading());  
    try {  
      await _repo.login(email: email, password: password);  
      emit(LoginSuccess('Login berhasil!'));  
    } catch (e) {  
      print(e);  
      emit(LoginFailure(e.toString()));  
    }  
  }  
}
```

13. Kita akan menggunakan bloc pada ui login yang sudah kita buat. BlocListener digunakan untuk merespons state yang terjadi dalam BloC tanpa membangun ulang seluruh UI. Ketika state terjadi, BlocListener akan memicu fungsi yang ditentukan dan memberikan akses ke state tersebut. BlocListener menerima dua parameter utama:
- bloc: Parameter ini mengacu pada instans BloC yang digunakan untuk mendengarkan state.
 - listener: Parameter ini adalah fungsi yang digunakan untuk merespons state.



Ket: Kita akan membungkus Container dengan Bloc Listener
kode program :

login.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:praktikum6/bloc/login/login_cubit.dart';

import '../utils/routes.dart';

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

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final emailEdc = TextEditingController();
  final passEdc = TextEditingController();
  bool passInvisible = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: BlocListener<LoginCubit, LoginState>(
        listener: (context, state) {
          if (state is LoginLoading) {
            ScaffoldMessenger.of(context)
              ..hideCurrentSnackBar()
              ..showSnackBar(SnackBar(content: Text('Loading..')));
          }
          if (state is LoginFailure) {
            ScaffoldMessenger.of(context)
              ..hideCurrentSnackBar()
          }
        },
      ),
    );
  }
}
```

```
        ..showSnackBar(SnackBar(  
          content: Text(state.msg),  
          backgroundColor: Colors.red,  
        ));  
      }  
      if (state is LoginSuccess) {  
        // context.read<AuthCubit>().loggedIn();  
        ScaffoldMessenger.of(context)  
          ..hideCurrentSnackBar()  
          ..showSnackBar(SnackBar(  
            content: Text(state.msg),  
            backgroundColor: Colors.green,  
          ));  
        Navigator.pushNamedAndRemoveUntil(  
          context, rHome, (route) => false);  
      }  
    },  
    child: Container(  
      margin: EdgeInsets.symmetric(horizontal: 30, vertical: 70),  
      child: ListView(  
        children: [  
          Text("Login", style: TextStyle(  
            fontSize: 40, fontWeight: FontWeight.bold, color:  
Color(0xff3D4DE0)  
            ),),  
          SizedBox(height: 15,),  
          Text("Silahkan masukan e-mail dan password anda", style:  
TextStyle(  
            fontSize: 16,  
            ),),  
          SizedBox(height: 25,),  
          Text("e-mail", style: TextStyle(  
            fontSize: 16, fontWeight: FontWeight.bold  
            ),),  
          TextFormField(  
            controller: emailEdc,  
          ),  
          SizedBox(height: 10,),  
          Text("password", style: TextStyle(  
            fontSize: 16, fontWeight: FontWeight.bold  
            ),),  
          TextFormField(  
            controller: passEdc,  
            decoration: InputDecoration(  
  
              suffixIcon: IconButton(  
                icon: Icon(passInvisible ? Icons.visibility :  
Icons.visibility_off),  
                onPressed: () {  
                  setState(() {  
                    passInvisible = !passInvisible;  
                  });  
                },  
              ),  
            ),  
            obscureText: !passInvisible,  
          ),  
          SizedBox(height: 50,),  
          ElevatedButton(onPressed: () {  
            Navigator.pushNamed(context, '/home');
```

```

        },
        style: ElevatedButton.styleFrom(
          backgroundColor: Color(0xff3D4DE0),
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10)
          )
        ),
        child: Text("Login", style: TextStyle(
          fontWeight: FontWeight.bold, fontSize: 24, color:
Colors.white
        )),
      ),
      SizedBox(height: 25,),
      Row(
        mainAxisAlignment: MainAxisAlignment.center, //
Menengahkan elemen horizontal
        children: [
          Text("Belum punya akun ?"),
          TextButton(onPressed: (){
            Navigator.pushNamed(context, '/register');
          },
            child: Text("Daftar", style: TextStyle(
              fontWeight: FontWeight.bold,
              color: Color(0xff3D4DE0)
            )),
        ],
      ),
    ),
  ],
),
),
),
),
);
}
}

```

Untuk mengirimkan data email dan password kita perlu mengubah kode program pada button login

Sebelum
<pre> ElevatedButton(onPressed: (){ Navigator.pushNamed(context, '/home'); }, style: ElevatedButton.styleFrom(backgroundColor: Color(0xff3D4DE0), shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10))), child: Text("Login", style: TextStyle(fontWeight: FontWeight.bold, fontSize: 24, color: Colors.white)),) </pre>
Sesudah
<pre> ElevatedButton(onPressed: (){ context .read<LoginCubit>() </pre>

Sebelum
<pre>import 'package:firebase_core/firebase_core.dart'; import 'package:flutter/material.dart'; import 'package:praktikum6/ui/splash.dart'; import 'package:praktikum6/utils/routes.dart'; import 'firebase_options.dart'; void main() async { WidgetsFlutterBinding.ensureInitialized(); await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform,); runApp(const MyApp()); } class MyApp extends StatelessWidget { const MyApp({Key? key}) : super(key: key); @override Widget build(BuildContext context) { return MaterialApp(title: "Praktikum 6", debugShowCheckedModeBanner: false, navigatorKey: NAV_KEY, onGenerateRoute: generateRoute, home: SplashScreen(),); } }</pre>
Sesudah
<pre>import 'package:firebase_core/firebase_core.dart'; import 'package:flutter/material.dart'; import 'package:flutter_bloc/flutter_bloc.dart'; import 'package:praktikum6/bloc/login/login_cubit.dart'; import 'package:praktikum6/ui/splash.dart';</pre>

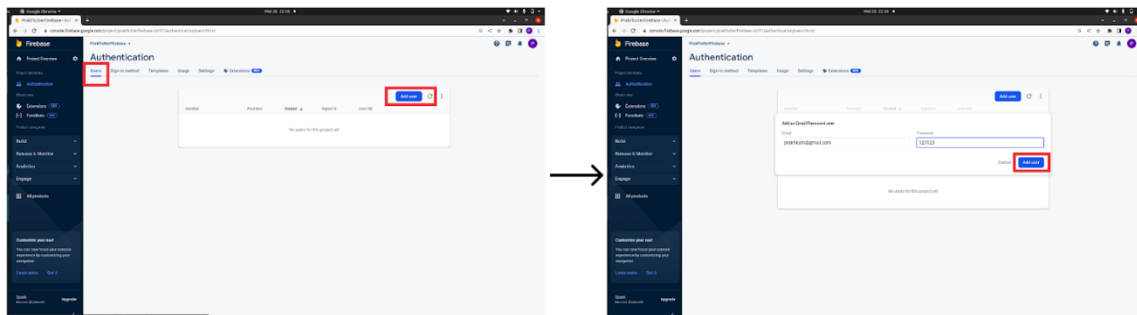
```
import 'package:praktikum6/utils/routes.dart';
import 'firebase_options.dart';

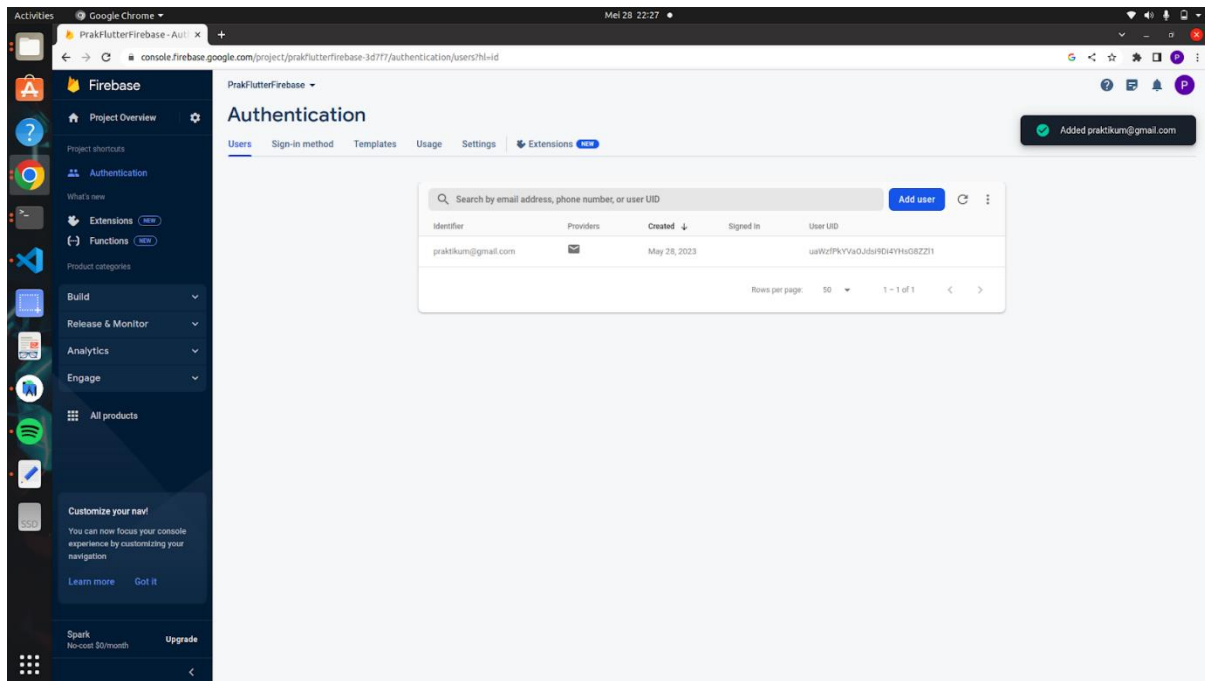
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MultiBlocProvider(
      providers: [
        BlocProvider(create: (context) => LoginCubit()),
      ],
      child: MaterialApp(
        title: "Praktikum 6",
        debugShowCheckedModeBanner: false,
        navigatorKey: NAV_KEY,
        onGenerateRoute: generateRoute,
        home: SplashScreen(),
      ),
    );
  }
}
```

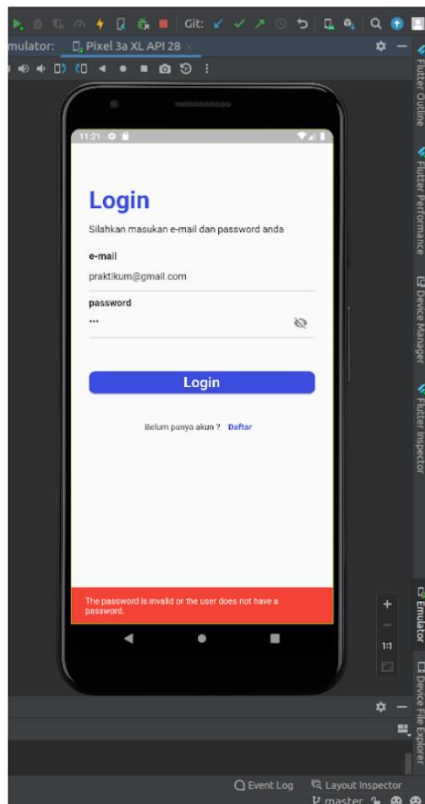
15. Untuk mengetahui apakah program kita sudah berjalan maka perlu menambahkan user secara manual pada firebase console kita dengan cara sebagai berikut :



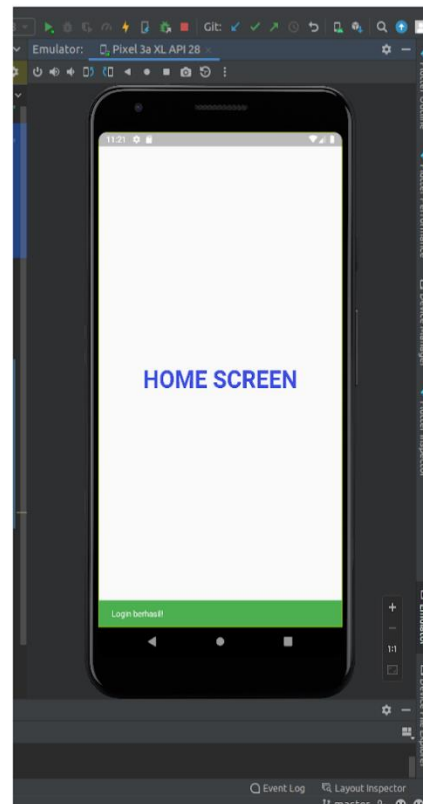


Berhasil menambahkan user

16. Login pada aplikasi dengan email dan password yang sudah dibuat

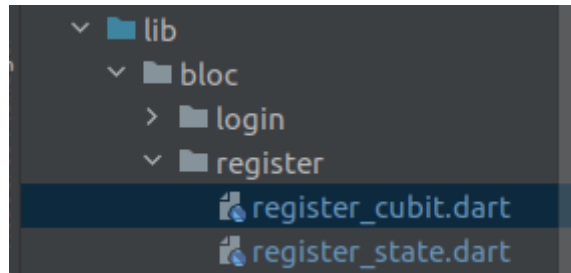


Login gagal



Login berhasil

17. Login sudah berhasil kita buat, maka selanjutnya kita akan membuat bagian register. Karena kita sudah membuat Future<void> register. Selanjutnya kita atur untuk blocnya. Buat folder “register” di dalam folder bloc.



register state.dart

```
part of 'register_cubit.dart';

@immutable
abstract class RegisterState {}

class RegisterInitial extends RegisterState {}

class RegisterLoading extends RegisterState {}

class RegisterSuccess extends RegisterState {
  final String msg;

  RegisterSuccess(this.msg);
}

class RegisterFailure extends RegisterState {
  final String msg;

  RegisterFailure(this.msg);
}
```

register cubit.dart

```
import 'package:bloc/bloc.dart';
import 'package:meta/meta.dart';
import '../repositories/auth_repo.dart';

part 'register_state.dart';

class RegisterCubit extends Cubit<RegisterState> {
  RegisterCubit() : super(RegisterInitial());
  final _repo = AuthRepo();

  Future<void> register(
    {required String email, required String password}) async {
    emit(RegisterLoading());
    try {
      await _repo.register(email: email, password: password);
      emit(RegisterSuccess('Berhasil!'));
    } catch (e) {
      emit(RegisterFailure(e.toString()));
    }
  }
}
```

```
    } catch (e) {  
      print(e);  
      emit(RegisterFailure(e.toString()));  
    }  
  }  
}
```

18. Menggunakan bloc pada register.dart, sehingga kode program menjadi seperti berikut :

register.dart

```
import 'package:flutter/material.dart';  
import 'package:flutter_bloc/flutter_bloc.dart';  
import 'package:praktikum6/bloc/register/register_cubit.dart';  
  
import '../utils/routes.dart';  
  
class RegisterScreen extends StatefulWidget {  
  const RegisterScreen({Key? key}) : super(key: key);  
  
  @override  
  State<RegisterScreen> createState() => _RegisterScreenState();  
}  
  
class _RegisterScreenState extends State<RegisterScreen> {  
  final emailEdc = TextEditingController();  
  final passEdc = TextEditingController();  
  bool passInvisible = false;  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: BlocListener<RegisterCubit, RegisterState>(  
        listener: (context, state) {  
          if (state is RegisterLoading) {  
            ScaffoldMessenger.of(context)  
              ..hideCurrentSnackBar()  
              ..showSnackBar(SnackBar(content: Text('Loading..')));  
          }  
          if (state is RegisterFailure) {  
            ScaffoldMessenger.of(context)  
              ..hideCurrentSnackBar()  
              ..showSnackBar(SnackBar(  
                content: Text(state.msg),  
                backgroundColor: Colors.red,  
              ));  
          }  
          if (state is RegisterSuccess) {  
            // context.read<AuthCubit>().loggedIn();  
            ScaffoldMessenger.of(context)  
              ..hideCurrentSnackBar()  
              ..showSnackBar(SnackBar(  
                content: Text(state.msg),  
                backgroundColor: Colors.green,  
              ));  
            Navigator.pushNamedAndRemoveUntil(  

```

```
        context, rLogin, (route) => false);
    },
    child: Container(
      margin: EdgeInsets.symmetric(horizontal: 30, vertical: 70),
      child: ListView(
        children: [
          Text("Register", style: TextStyle(
            fontSize: 40, fontWeight: FontWeight.bold, color:
Color(0xff3D4DE0)
          )),
          SizedBox(height: 15,),
          Text("Silahkan masukan e-mail dan password anda", style:
TextStyle(
            fontSize: 16,
          )),
          SizedBox(height: 25,),
          Text("e-mail", style: TextStyle(
            fontSize: 16, fontWeight: FontWeight.bold
          )),
          TextFormField(
            controller: emailEdc,
          ),
          SizedBox(height: 10,),
          Text("password", style: TextStyle(
            fontSize: 16, fontWeight: FontWeight.bold
          )),
          TextFormField(
            controller: passEdc,
            decoration: InputDecoration(
              suffixIcon: IconButton(
                icon: Icon(passInvisible ? Icons.visibility :
Icons.visibility_off),
                onPressed: () {
                  setState(() {
                    passInvisible = !passInvisible; // Toggle
_isPasswordVisible ketika ikon mata ditekan
                  });
                },
              ),
            obscureText: !passInvisible, // Atur obscureText
berdasarkan _isPasswordVisible
          ),
          SizedBox(height: 50,),
          ElevatedButton(onPressed: () {
            context
              .read<RegisterCubit>()
              .register(email: emailEdc.text, password:
passEdc.text);
          },
            style: ElevatedButton.styleFrom(
              backgroundColor: Color(0xff3D4DE0),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10)
              )
            ),
            child: Text("Register", style: TextStyle(
              fontWeight: FontWeight.bold, fontSize: 24, color:
Colors.white
```

```
        ),),),
        SizedBox(height: 25,),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text("Sudah punya akun ?"),
            TextButton(onPressed: (){
              Navigator.pushNamed(context, '/login');
            },
              child: Text("Login", style: TextStyle(
                fontWeight: FontWeight.bold,
                color: Color(0xff3D4DE0)
              ),))
          ],
        ),
      ],
    ),
  ),
),
);
}
```

19. Menambahkan RegisterCubit pada MultiBlocProvider yang ada pada main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:praktikum6/bloc/login/login_cubit.dart';
import 'package:praktikum6/ui/splash.dart';
import 'package:praktikum6/utils/routes.dart';
import 'bloc/register/register_cubit.dart';
import 'firebase_options.dart';

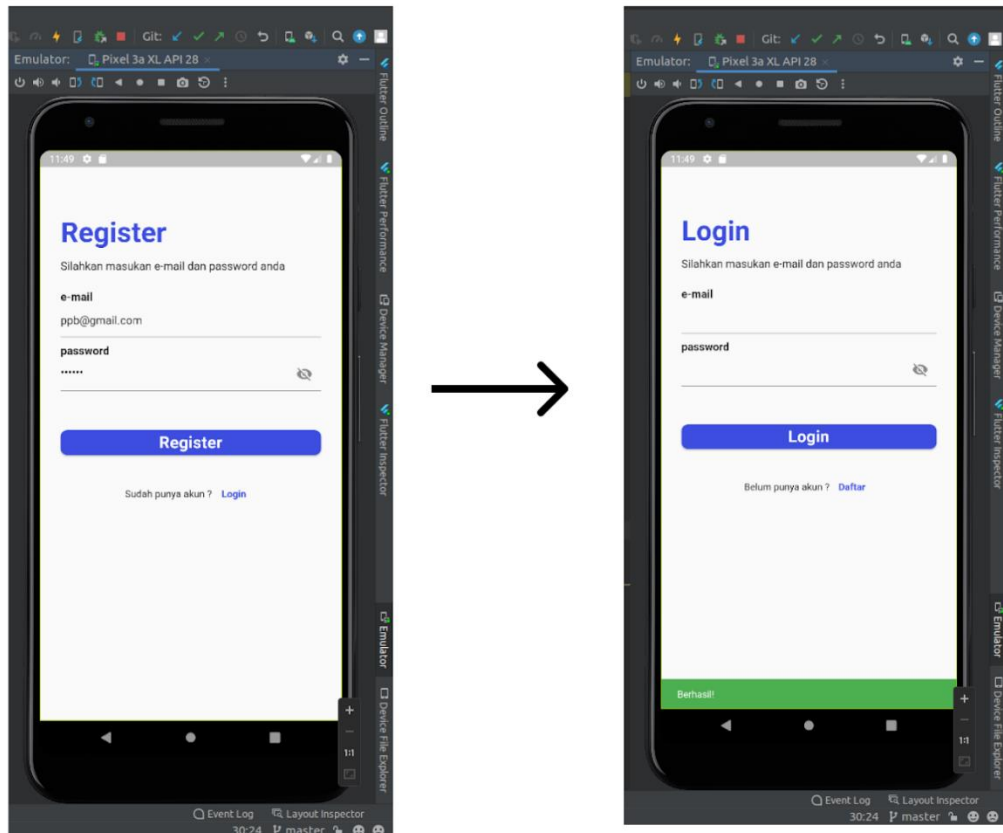
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MultiBlocProvider(
      providers: [
        BlocProvider(create: (context) => LoginCubit()),
        BlocProvider(create: (context) => RegisterCubit()),
      ],
      child: MaterialApp(
        title: "Praktikum 6",
        debugShowCheckedModeBanner: false,
        navigatorKey: NAV_KEY,
        onGenerateRoute: generateRoute,
        home: SplashScreen(),
      ),
    );
  }
}
```

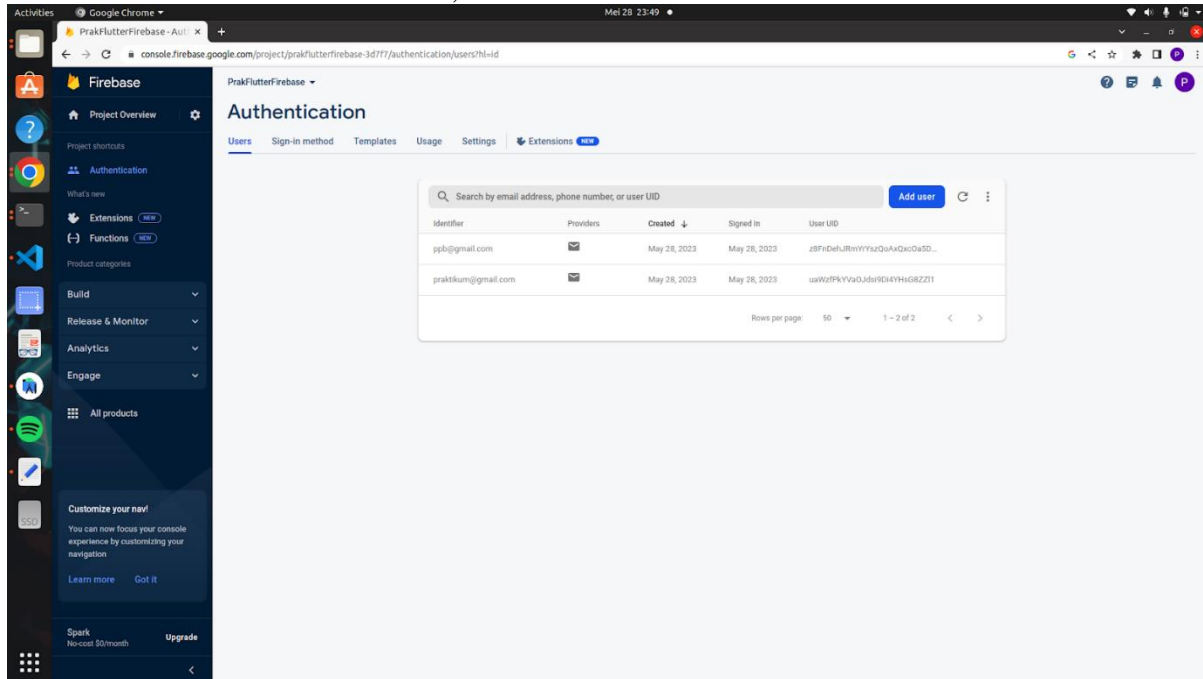
```
}  
    },  
};
```

20. Melakukan register melalui aplikasi



Register berhasil

21. Refresh dashboard firebase, maka user akan bertambah



The screenshot shows the Firebase Authentication console in a web browser. The left sidebar contains the Firebase navigation menu with options like Project Overview, Authentication, Extensions, Functions, Build, Release & Monitor, Analytics, Engage, and All products. The main content area is titled 'Authentication' and has tabs for Users, Sign-in method, Templates, Usage, Settings, and Extensions. The 'Users' tab is active, displaying a table of users. The table has columns for Identifier, Providers, Created, Signed in, and User UID. There are two users listed: 'ppb@gmail.com' and 'praktikum@gmail.com', both created on May 28, 2023. At the bottom of the table, it indicates 'Rows per page: 50' and '1 - 2 of 2'.

Identifier	Providers	Created	Signed in	User UID
ppb@gmail.com		May 28, 2023	May 28, 2023	z8Fid6uJRWYtYsQuAaQcD4sD...
praktikum@gmail.com		May 28, 2023	May 28, 2023	uaWtPKYVaQJdeRd4YHsG8ZZ1