



# Tahun Ajar 2022/2023 PEMROGRAMAN PERANGKAT BERGERAK

## Praktikum 7:

Firebase Authentication (Google dan Phone)

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 Authentication 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.

#### Otentikasi Menggunakan Akun Google

- 1. Buka projek praktikum sebelumnya, praktikum kali ini melanjutkan praktikum sebelumnya.
- 2. Pada halaman login.dart tambahkan code berikut.

```
"Login",
          fontWeight: FontWeight.bold,
          fontSize: 24,
const SizedBox(
  height: 30.0,
),
Row (
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    GestureDetector(
      onTap: () {},
      child: const CircleAvatar(
        radius: 20.0,
        backgroundImage: NetworkImage(
             'https://img2.pngdownload.id/20190228/qby/kisspng-
google-logo-google-account-g-suite-google-images-g-icon-
archives-search-png-5c77ad39b77471.9286340315513470017515.jpg'),
      ),
    ),
    const SizedBox(
      width: 30.0,
    ),
    GestureDetector (
      onTap: () {},
      child: const CircleAvatar(
        radius: 20.0,
        backgroundImage: NetworkImage(
             'https://freepngimg.com/thumb/business/83615-blue-
icons-symbol-telephone-computer-logo.png'),
      ),
    )
  ],
),
```

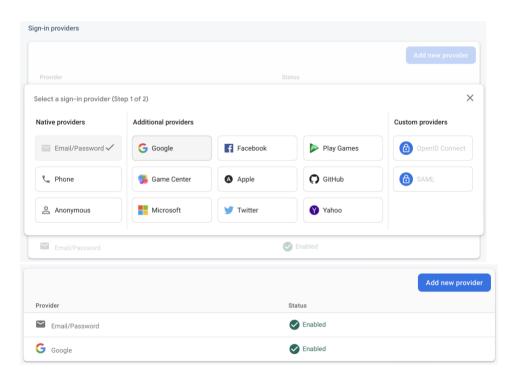




Maka ketika di run akan menjadi seperti berikut:



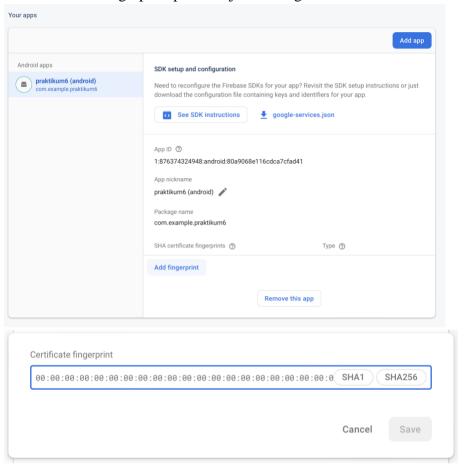
- 3. Tambahkan dependency google\_sign\_in dengan menjalankan command **flutter pub add google\_sign\_in** pada terminal.
- 4. Untuk mengaktifkan otentikasi menggunakan akun Google perlu menambahkan signin method Google pada web firebase project anda.







5. Untuk menggunakan otentikasi menggunakan akun Google pada Android perlu menambahkan fingerprint pada Project Settings web firebase.



Untuk mendapatkan SHA1 dan SHA256 menggunakan command berikut pada terminal:

Windows: keytool -list -v \
-alias androiddebugkey -keystore %USERPROFILE%\.android\debug.keystore

Mac: keytool -list -v \
-alias androiddebugkey -keystore ~/.android/debug.keystore

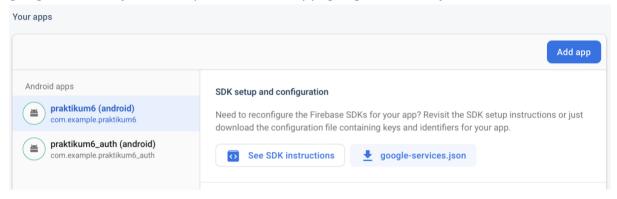
(base) alamsyah@Alamsyah android % keytool -list -v  $\$ 

Akan diminta untuk menginputkan keystore password, isikan android.





6. Download google-services.json pada project settings web firebase dan replace google-services.json lama pada android/app/google-services.json.



7. Masih pada login.dart tambahkan fungsi berikut untuk sign in menggunakan akun Google.

```
final emailEdc = TextEditingController();
final passEdc = TextEditingController();
bool passInvisible = false;
Future<UserCredential> signInWithGoogle() async {
  final GoogleSignInAccount? gUser = await GoogleSignIn().signIn();
  final GoogleSignInAuthentication gAuth = await
qUser!.authentication;
  final credential = GoogleAuthProvider.credential(
    accessToken: gAuth.accessToken,
    idToken: gAuth.idToken,
 );
  return await
FirebaseAuth.instance.signInWithCredential(credential).then(
      (value) async => await Navigator.pushAndRemoveUntil(
         MaterialPageRoute(builder: (context) => HomeScreen()),
          (route) => false));
}
```





8. Tambahkan fungsi signInWithGoogle pada 🧲.

9. Maka ketika <sup>G</sup> di tap akan muncul pilihan akun Google yang tersedia.







10. Pilih akun yang anda inginkan maka akan otomatis terdaftar dan masuk ke home screen aplikasi.





#### Sign Out

1. Buat tombol untuk sign out pada halaman Home Screen (home\_screen.dart).

```
body: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Text(
        "HOME SCREEN",
        style: TextStyle(
            fontSize: 40,
            fontWeight: FontWeight.bold,
            color: Color(0xff3D4DE0)),
    ),
    ElevatedButton(
        onPressed: () {},
        child: const Text('Keluar'),
    ),
    ],
    ),
    ),
    ),
}
```





Tampilan Home Screen



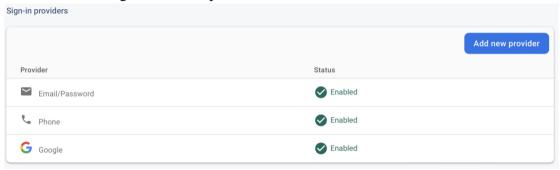
2. Tambahkan fungsi sign out pada tombol keluar.





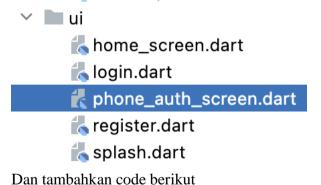
#### Otentikasi Menggunakan Nomor Handphone

1. Aktifkan Phone sign-in method pada firebase.



2. Pada login.dart tambahkan Navigator agar ketika ditekan aplikasi menampilkan halaman Phone Auth.

3. Buat file baru bernama phone\_auth\_screen.dart;







```
import 'package:firebase auth/firebase auth.dart';
import 'package:flutter/material.dart';
class PhoneAuthScreen extends StatefulWidget {
  @override
  PhoneAuthScreenState createState() => PhoneAuthScreenState();
class PhoneAuthScreenState extends State<PhoneAuthScreen> {
  FirebaseAuth auth = FirebaseAuth.instance;
  bool otpVisibility = false;
  User? user;
  String verificationID = "";
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        margin: EdgeInsets.all(10),
        child: Column (
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            TextField(
              decoration: InputDecoration(
                hintText: 'Masukkan nomor telepon anda',
                prefix: Padding(padding: EdgeInsets.all(4), child:
Text('+62')),
              keyboardType: TextInputType.phone,
            ),
            Visibility(
              child: TextField(
                decoration: InputDecoration (
                  hintText: 'OTP',
                  prefix: Padding(padding: EdgeInsets.all(4), child:
Text('')),
                ),
                keyboardType: TextInputType.number,
              ),
              visible: otpVisibility,
            ),
            SizedBox(
              height: 10,
            MaterialButton (
              color: const Color(0xff3D4DE0),
              onPressed: () {},
              child: Text(otpVisibility ? "Verify" : "Login",
                  style: TextStyle(color: Colors.white, fontSize: 20)),
            ),
          ],
        ),
      ),
    );
  }
}
```







4. Tambahkan TextEditingController.

```
User? user;
String verificationID = "";

TextEditingController _phoneController = TextEditingController();
TextEditingController _otpController = TextEditingController();
```

5. Dispose controller tersebut agar tidak terjadi memory leak.

```
TextEditingController _phoneController = TextEditingController();
TextEditingController _otpController = TextEditingController();

@override
void dispose() {
    _phoneController.dispose();
    _otpController.dispose();
    super.dispose();
}
```





Pasang controller pada TextField.

```
TextField(
   controller: _phoneController,
   decoration: InputDecoration(
      hintText: 'Masukkan nomor telepon anda',
      prefix: Padding(padding: EdgeInsets.all(4), child:

Text('+62')),
   ),
   keyboardType: TextInputType.phone,
),
Visibility(
   child: TextField(
      controller: _otpController,
      decoration: InputDecoration(
        hintText: 'OTP',
        prefix: Padding(padding: EdgeInsets.all(4), child:

Text('')),
   ),
   keyboardType: TextInputType.number,
   ),
   visible: otpVisibility,
),
```

7. Buat fungsi untuk otentikasi menggunakan nomor handphone.

```
void loginWithPhone() async {
  auth.verifyPhoneNumber(
    phoneNumber: "+62" + _phoneController.text,
    timeout: const Duration (seconds: 60),
    verificationCompleted: (PhoneAuthCredential credential) async
{
      await auth.signInWithCredential(credential).then((value) {
        print("You are logged in successfully");
      });
    },
    verificationFailed: (FirebaseAuthException e) {
      print(e.message);
    codeSent: (String verificationId, int? resendToken) {
      otpVisibility = true;
      verificationID = verificationId;
      setState(() {});
    },
    codeAutoRetrievalTimeout: (String verificationId) {},
  );
}
```





Otentikasi menggunakan nomor handphone menggunakan method verifyPhoneNumber yang memiliki 6 attribute wajib yaitu:

- phoneNumber: berisi nomor handphone yang akan diverifikasi dan dikirim code OTP.
- timeout: Durasi aktif kode OTP sebelum expired.
- verificationCompleted: Apa yang akan dilakukan apabila nomor handphone terverifikasi.
- verificationFailed: Apa yang akan dilakukan apabila nomor handphone gagal diverifikasi.
- codeSent: Apa yang dilakukan apabila code sudah terkirim.
- codeAutoRetrievalTimeout: Dapat digunakan untuk memverifikasi bahwa nomor telepon adalah milik pengguna.

#### 8. Buat fungsi untuk memverifikasi code OTP.

```
void verifyOTP() async {
  PhoneAuthCredential credential = PhoneAuthProvider.credential(
      verificationId: verificationID, smsCode: _otpController.text);
  await auth.signInWithCredential(credential).then(
    (value) {
      setState(() {
        user = FirebaseAuth.instance.currentUser;
      });
    },
  ).whenComplete(
    () {
      if (user != null) {
        ScaffoldMessenger.of(context)
          ..hideCurrentSnackBar()
          ..showSnackBar(const SnackBar(content: Text('Berhasil
Login')));
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            builder: (context) => HomeScreen(),
          ),
        );
      } else {
        ScaffoldMessenger.of(context)
          ..hideCurrentSnackBar()
          ..showSnackBar(const SnackBar(content: Text('Login Gagal')));
      }
    },
  );
}
```





9. Pada MaterialButton tambahkan code berikut.

```
MaterialButton(
  color: const Color(0xff3D4DE0),
  onPressed: () {
    if (_phoneController.text.isNotEmpty) {
        if (otpVisibility) {
            verifyOTP();
        } else {
            loginWithPhone();
        }
    }
    },
    child: Text(otpVisibility ? "Verify" : "Login",
        style: TextStyle(color: Colors.white, fontSize: 20)),
},
```

10. Ketika tombol Login ditekan akan muncul TextField untuk menginputkan code OTP yang telah dikirimkan ke nomor yang telah diinputkan sebelumnya. Inputkan code OTP yang benar dan aplikasi akan menuju Home Screen.







11. Untuk membuat user tidak perlu login setelah aplikasi ditutup, tambahkan code berikut pada main.dart.

```
home: StreamBuilder<User?>(
  stream: FirebaseAuth.instance.authStateChanges(),
 builder: (context, snapshot) {
    if (snapshot.connectionState ==
ConnectionState.waiting) {
      return const CircularProgressIndicator();
    } else if (snapshot.hasData) {
      return HomeScreen();
    } else if (snapshot.hasError) {
      return const Center (
        child: Text('Something went wrong'),
      );
    } else {
      return const LoginScreen();
 },
),
```