

BAB 8. FIREBASE

8.1 Teori

Firebase adalah suatu layanan dari Google untuk memberikan kemudahan bahkan mempermudah para developer aplikasi dalam mengembangkan aplikasinya. Firebase sebagai BaaS (Backend as a Service) merupakan solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan developer. Dengan menggunakan Firebase, apps developer bisa fokus dalam mengembangkan aplikasi tanpa memberikan effort yang besar untuk urusan backend.

Fitur yang ditawarkan firebase yaitu

1. Firebase Analytics
2. Firebase Cloud Messaging and Notifications
3. Firebase Authentication
4. Firebase Cloud Firestore
5. Firebase Realtime Database
6. Firebase Hosting
7. Dan lain sebagainya

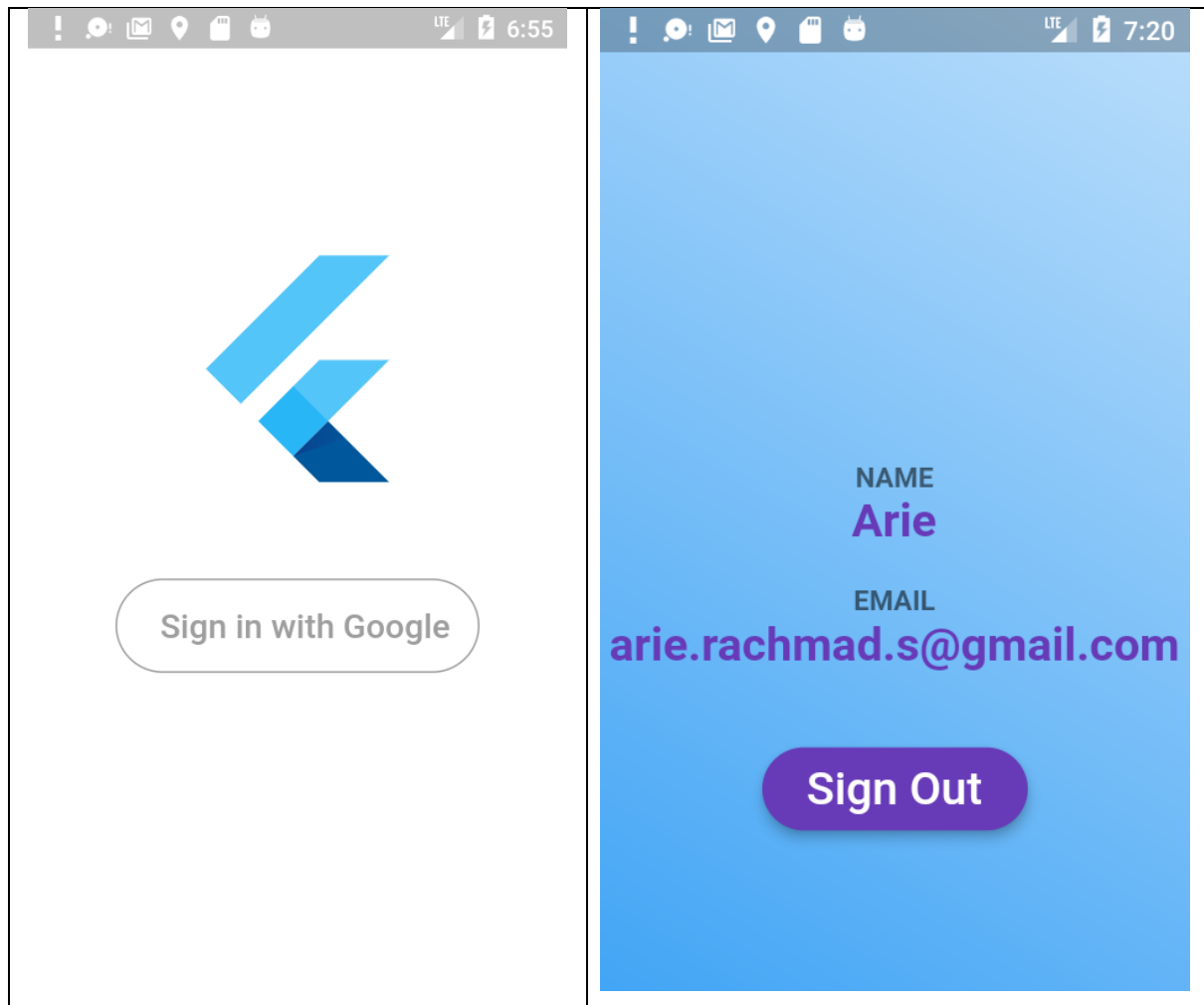
Untuk lebih detail tentang fitur yang dimiliki firebase dapat dilihat pada tautan berikut:

1. <https://firebase.google.com/products-build?hl=en>
2. <https://firebase.google.com/products-release?hl=en>
3. <https://firebase.google.com/products-engage?hl=en>

8.2 Desain Aplikasi

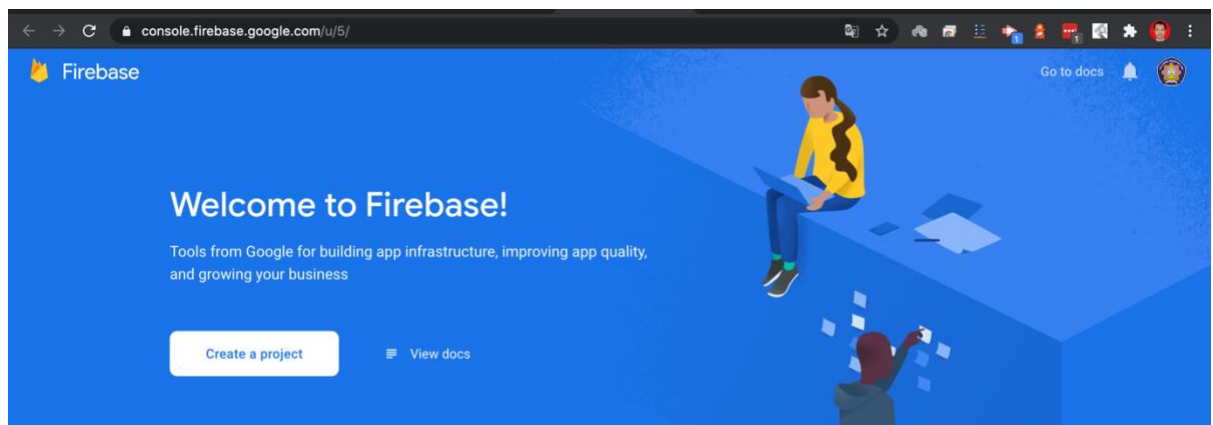
Aplikasi flutter yang akan dibuat pada modul ini adalah memanfaatkan salah satu fitur firebase yaitu firebase authentication yang dapat digunakan untuk melakukan login.

Tampilan aplikasi akhir yang diharapkan adalah:

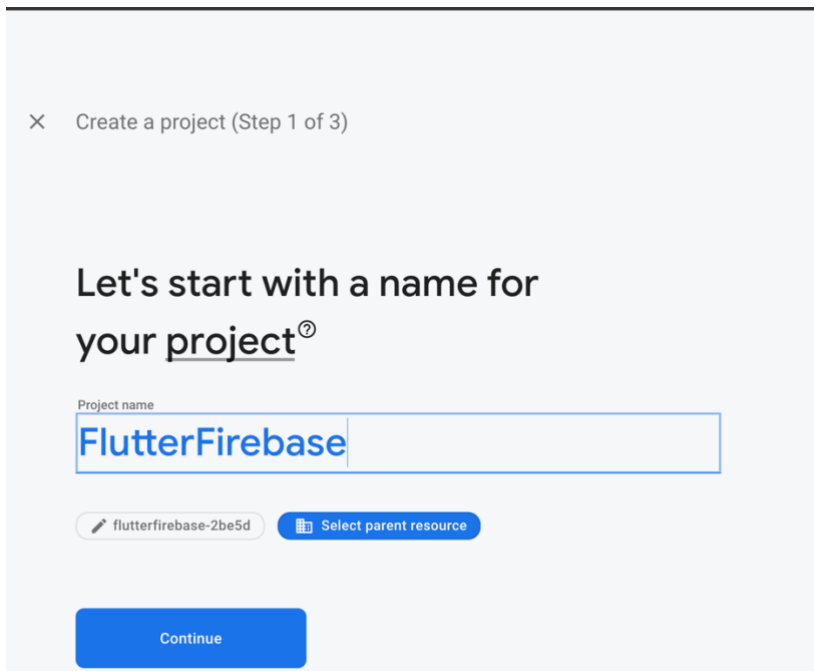


8.3 Persiapan Project

1. Membuat project di firebase console <https://console.firebase.google.com/>



2. Tekan create project dan diisi nama project sesuai dengan project yang akan dibuat, misalnya seperti ini.



× Create a project (Step 1 of 3)

Let's start with a name for your project[?]

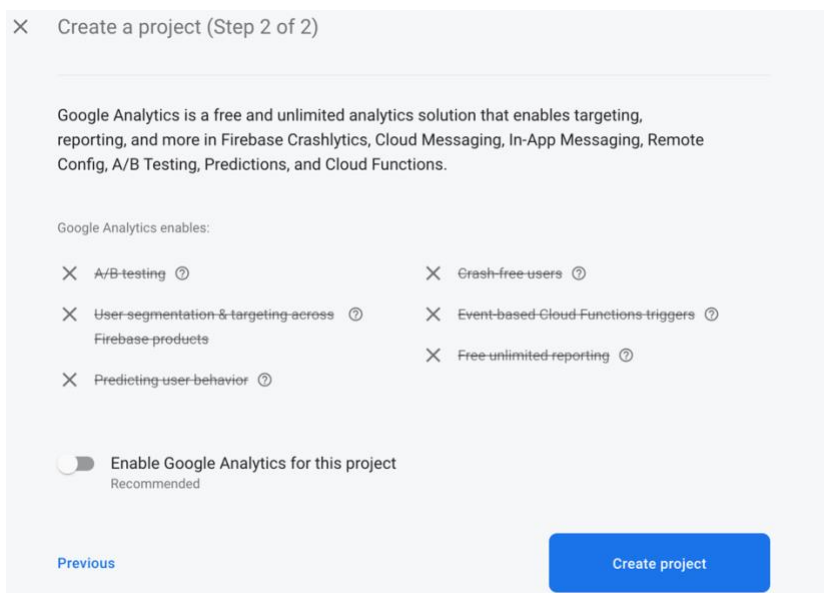
Project name

FlutterFirebase

flutterfirebase-2be5d Select parent resource

Continue

3. Tekan continue maka selanjutnya akan ada tampilan seperti berikut:



× Create a project (Step 2 of 2)

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

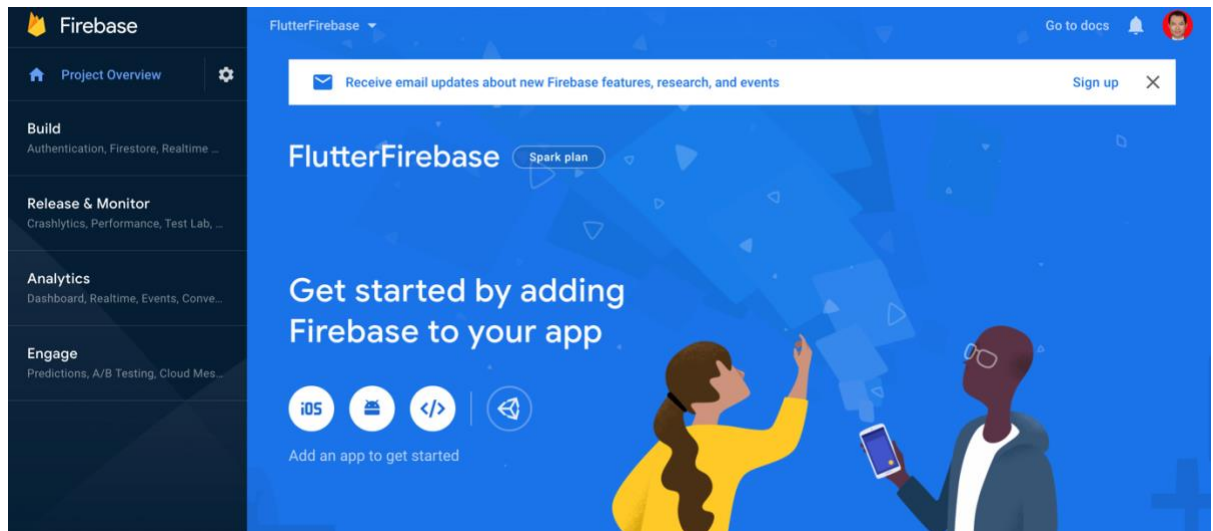
Google Analytics enables:

- × A/B-testing[?]
- × User-segmentation & targeting across Firebase-products[?]
- × Predicting user behavior[?]
- × Crash-free users[?]
- × Event-based Cloud Functions triggers[?]
- × Free unlimited reporting[?]

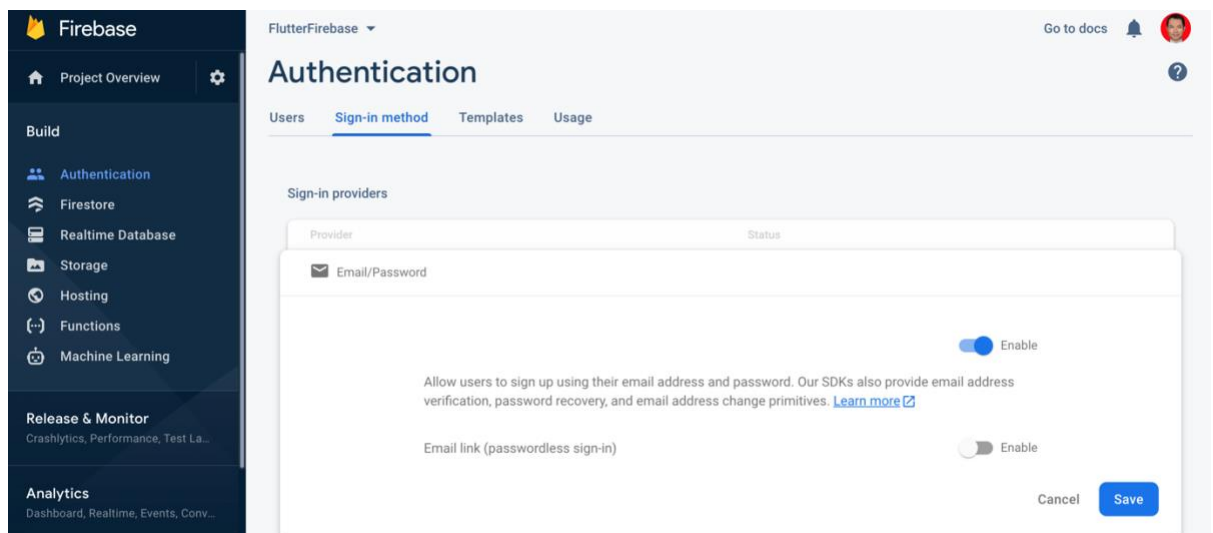
☐ Enable Google Analytics for this project
Recommended

Previous Create project

4. Pada Firebase console, expand Build menu pada panel sebelah kiri.



5. Klik Authentication, selanjutnya klik button Get Started, kemudian pilih tab Sign-in method dan terakhir Save pengaturan yang sudah dilakukan, seperti pada ilustrasi di bawah ini

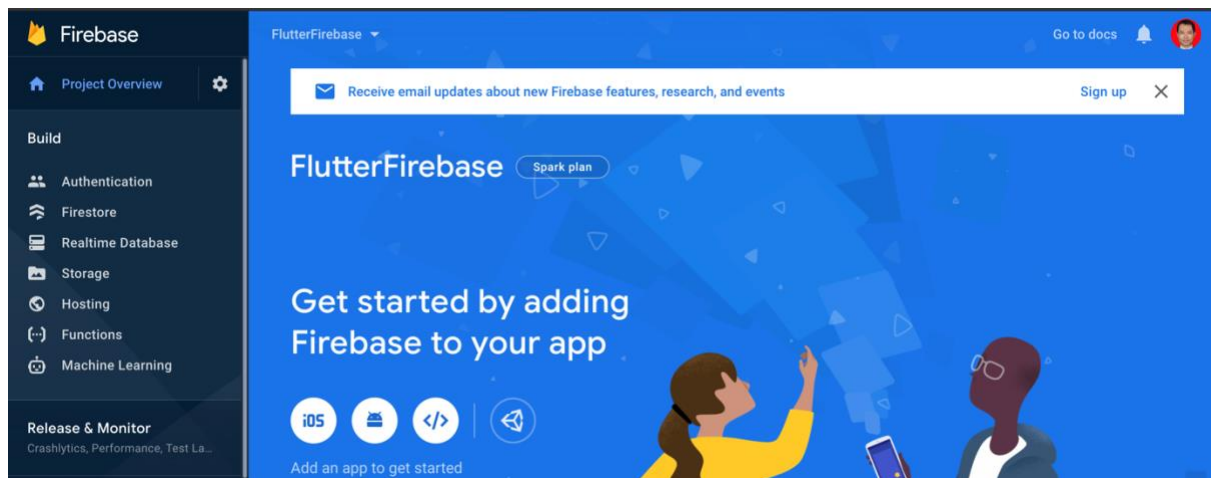


6. Setup menjadi enable pengaturan autentikasi menggunakan login yang terintegrasi dengan aplikasi lain misalnya google. Tampilannya adalah sebagai berikut:

Provider	Status
Email/Password	Enabled
Phone	Disabled
Google	Enabled

8.4 Konfigurasi Firebase

1. Klik Project Overview maka terdapat tampilan seperti dibawah ini. Yang dapat digunakan untuk menambahkan firebase pada aplikasi yang akan dibuat.



2. Klik logo android karena kita akan fokus untuk membuat aplikasi android. Setelah itu akan muncul tampilan seperti berikut ini.

The image shows a dialog box titled 'Add Firebase to your Android app'. It has a close button (X) in the top left. The first step is '1 Register app'. There are three input fields: 'Android package name' with the value 'com.company.appname', 'App nickname (optional)' with the value 'My Android App', and 'Debug signing certificate SHA-1 (optional)' with a long hexadecimal string. Below the third field is a note: 'Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.' At the bottom is a 'Register app' button.

Tambahkan android packages name yang ada di AndroidManifest.xml pada tampilan di atas. Packages name juga dapat diubah dengan mengikuti langkah pada tautan berikut.

- <https://qastack.id/programming/51534616/how-to-change-package-name-in-flutter>

note : Untuk nicknames optional diisi atau tidak.

3. Untuk Debug Signing certificate dapat menggunakan perintah berikut ini di terminal visual code (<https://developers.google.com/android/guides/client-auth>).

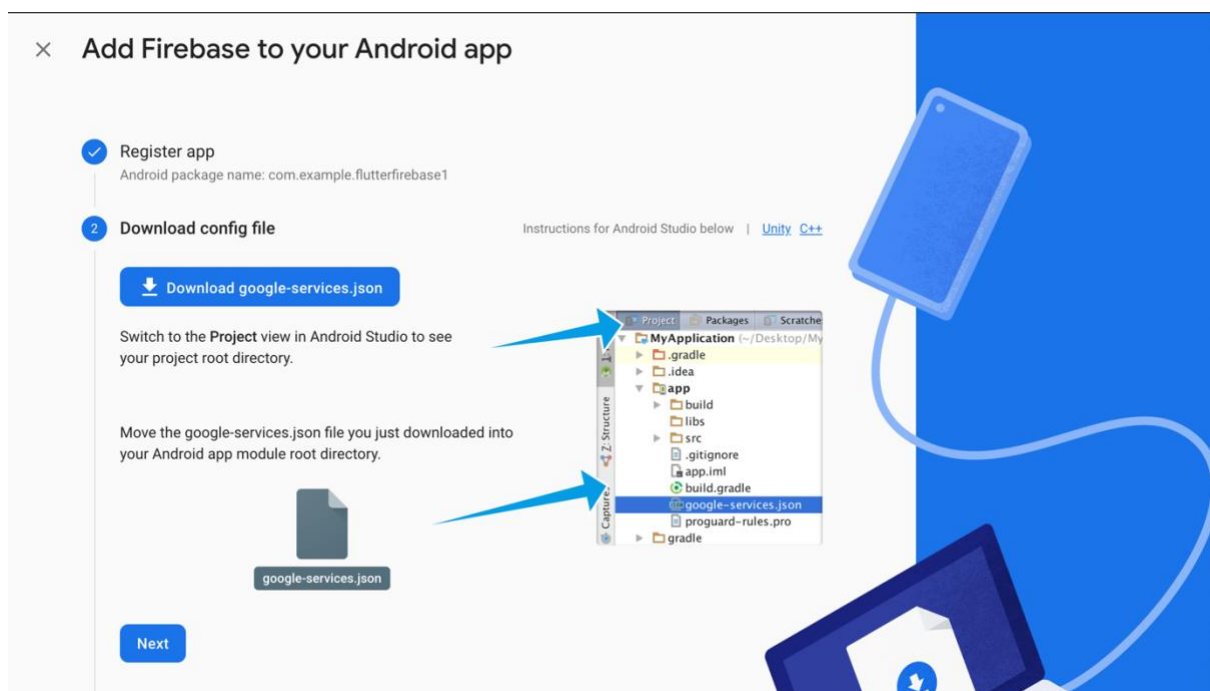
Windows

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

Unix/Linux

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

4. Menambah google-services.json pada folder android seperti berikut



5. Edit android/build.gradle untuk menambah google-services plugin dependency seperti berikut.

```
dependencies {  
    classpath 'com.android.tools.build:gradle:4.1.0'
```

```
classpath "org.jetbrains.kotlin:kotlin-gradle-  
plugin:$kotlin_version"  
  
classpath 'com.google.gms:google-services:4.3.5' // new  
}
```

6. Edit android/app/build.gradle untuk mengenable google-services plugin.

```
apply plugin: 'com.android.application'  
apply plugin: 'kotlin-android'  
apply from:  
"$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"  
apply plugin: 'com.google.gms.google-services' // new
```

7. Firebase memerlukan Multidex untuk di enabled, dan satu-satunya cara untuk melakukan hal itu adalah dengan mengupdate minimum supported SDK yaitu 21 atau versi yang lebih baru.

```
defaultConfig {  
    applicationId "com.example.gtk_flutter"  
    minSdkVersion 21 // Updated  
    targetSdkVersion 30  
    versionCode flutterVersionCode.toInteger()  
    versionName flutterVersionName  
}
```

8. Terakhir mengubah pubspec.yaml dengan menambah packages firebase yang diperlukan pada project flutter, seperti berikut.

```
dependencies:  
  flutter:  
    sdk: flutter  
  firebase_auth: ^1.0.0 # new  
  google_sign_in: ^4.5.3 # new
```

8.5 Membuat login_page.dart

Buatlah halaman login page yang digunakan untuk login menggunakan akun google.

```
import 'package:flutter/material.dart';
import 'package:flutterfirebase/sign_in.dart';
import 'package:flutterfirebase/first_screen.dart';

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        color: Colors.white,
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.max,
            mainAxisSize: MainAxisSize.max,
            children: <Widget>[
              FlutterLogo(size: 150),
              SizedBox(height: 50),
              _signInButton(),
            ],
          ),
        ),
      ),
    );
  }

  Widget _signInButton() {
    return OutlineButton(
      splashColor: Colors.grey,
      onPressed: () {
        signInWithGoogle().then((result) {
          if (result != null) {
            Navigator.of(context).push(
              MaterialPageRoute(
                builder: (context) {

```



```

        return FirstScreen();
      },
    ),
  );
}
});
},
shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(40)),
highlightElevation: 0,
borderSide: BorderSide(color: Colors.grey),
child: Padding(
  padding: const EdgeInsets.fromLTRB(0, 10, 0, 10),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.min,
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Image(image: AssetImage("assets/google_logo.png"), height: 35.0),
      Padding(
        padding: const EdgeInsets.only(left: 10),
        child: Text(
          'Sign in with Google',
          style: TextStyle(
            fontSize: 20,
            color: Colors.grey,
          ),
        ),
      ),
    ],
  ),
),
);
}
}

```

8.6 Membuat sign_in.dart

Pada file sign_in inilah yang memegang peranan penting pada modul firebase autentikasi.

```

import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:google_sign_in/google_sign_in.dart';

final FirebaseAuth _auth = FirebaseAuth.instance;

```

```

final GoogleSignIn googleSignIn = GoogleSignIn();

String name;
String email;
String imageUrl;

Future<String> signInWithGoogle() async {
    await Firebase.initializeApp();

    final GoogleSignInAccount googleSignInAccount = await googleSignIn.signIn();
    final GoogleSignInAuthentication googleSignInAuthentication =
        await googleSignInAccount.authentication;

    final AuthCredential credential = GoogleAuthProvider.credential(
        accessToken: googleSignInAuthentication.accessToken,
        idToken: googleSignInAuthentication.idToken,
    );

    final UserCredential authResult =
        await _auth.signInWithCredential(credential);
    final User user = authResult.user;

    if (user != null) {
        // Checking if email and name is null
        assert(user.email != null);
        assert(user.displayName != null);
        assert(user.photoURL != null);

        name = user.displayName;
        email = user.email;
        imageUrl = user.photoURL;

        // Only taking the first part of the name, i.e., First Name
        if (name.contains(" ")) {
            name = name.substring(0, name.indexOf(" "));
        }

        assert(!user.isAnonymous);
        assert(await user.getIdToken() != null);

        final User currentUser = _auth.currentUser;
        assert(user.uid == currentUser.uid);

        print('signInWithGoogle succeeded: $user');
    }
}

```



```

    ),
    SizedBox(height: 40),
    Text(
      'NAME',
      style: TextStyle(
        fontSize: 15,
        fontWeight: FontWeight.bold,
        color: Colors.black54),
    ),
    Text(
      name,
      style: TextStyle(
        fontSize: 25,
        color: Colors.deepPurple,
        fontWeight: FontWeight.bold),
    ),
    SizedBox(height: 20),
    Text(
      'EMAIL',
      style: TextStyle(
        fontSize: 15,
        fontWeight: FontWeight.bold,
        color: Colors.black54),
    ),
    Text(
      email,
      style: TextStyle(
        fontSize: 25,
        color: Colors.deepPurple,
        fontWeight: FontWeight.bold),
    ),
    SizedBox(height: 40),
    RaisedButton(
      onPressed: () {
        signOutGoogle();

```

```

Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (context)
{return LoginPage();}), ModalRoute.withName('/'));

```

```

    },
    color: Colors.deepPurple,
    child: Padding(
      padding: const EdgeInsets.all(8.0),
      child: Text(
        'Sign Out',
        style: TextStyle(fontSize: 25, color: Colors.white),

```

```

        ),
      ),
      elevation: 5,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(40)),
    )
  ],
),
),
),
);
}
}

```

8.8 Mengedit main.dart

```

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Login',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LoginPage(),
    );
  }
}

```

8.9 Tugas

Edit aplikasi sehingga dapat login menggunakan email dan password. Sehingga terdapat 2 cara untuk login menggunakan email dan akun google.