

ACARA 37 : AUTHENTICATION

Pokok Bahasan	: Authentication
Acara Praktikum/Pertemuan	: Minggu 11 / 1
Tempat	: Daring / Luring Politeknik Negeri Jember
Alokasi Waktu	: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

- a. Mahasiswa mampu memahami konsep Authentication.
- b. Mahasiswa mampu menerapkan Authentication.

b. Indikator

Keberhasilan mahasiswa memahami konsep Authentication dan menerapkan Authentication pada flutter.

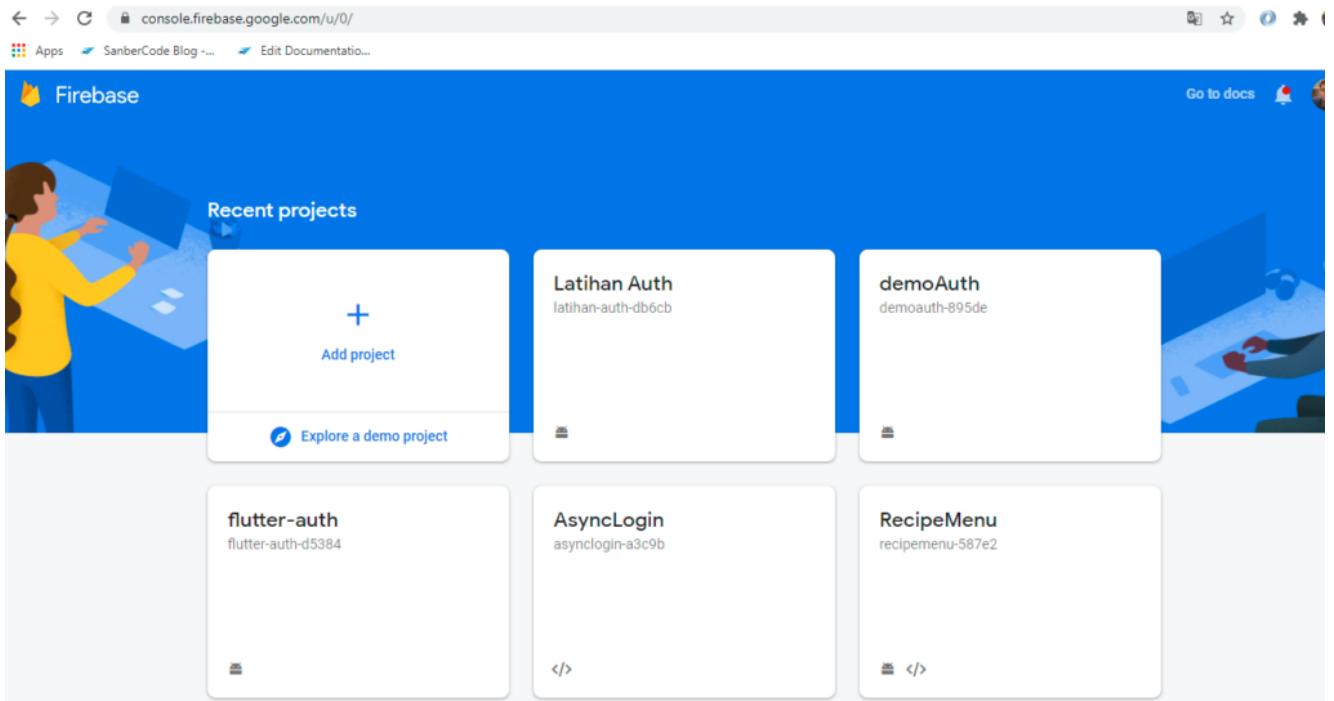
c. Dasar Teori

Selain menggunakan API yang telah disediakan back-end developer maupun oleh pihak ketiga (third party), Anda juga dapat membuat API sendiri, misalnya dalam bentuk database dengan menggunakan Firebase. Authentication adalah metode keamanan yang diterapkan saat pengguna akan memasuki sebuah sistem atau aplikasi dengan cara harus memasukan email/username dan password untuk memastikan bahwa pengguna memang memiliki hak akses ke sebuah sistem.

Untuk mulai menggunakannya Anda cukup menuju <https://firebase.google.com/> dan login menggunakan akun Google Anda. Langkah-langkah yang diperlukan selanjutnya adalah sebagai berikut:

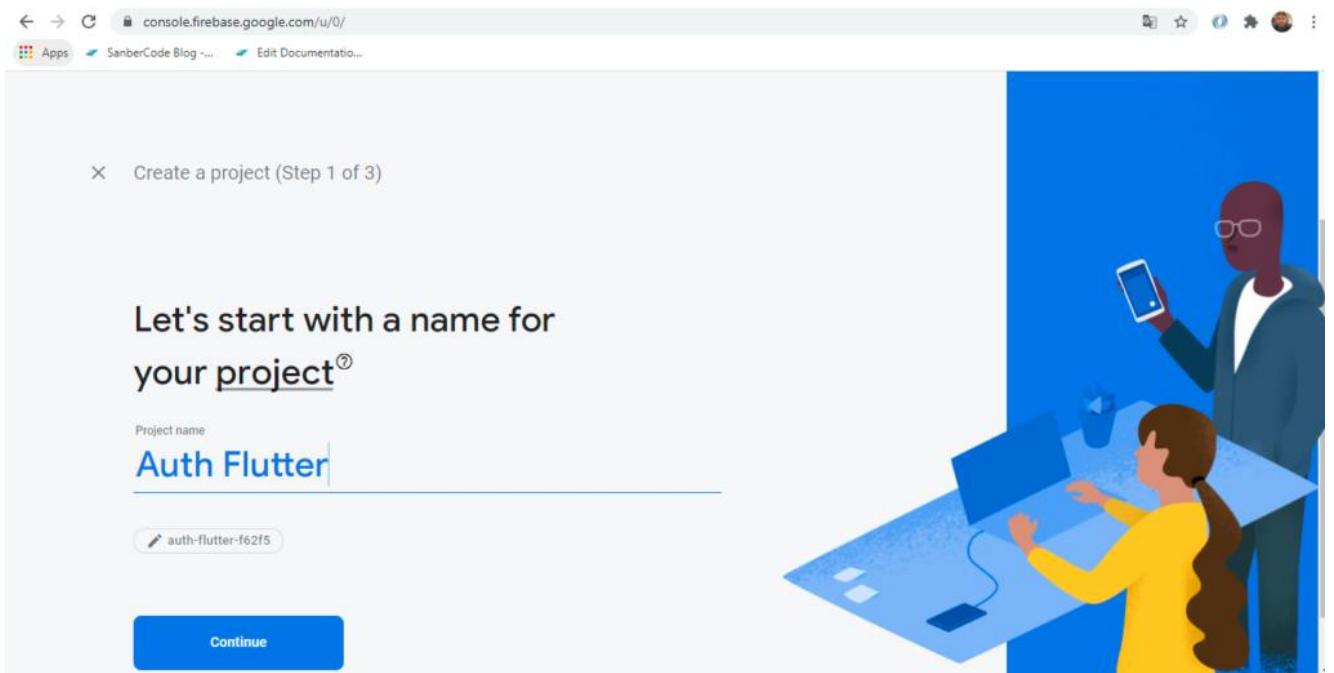
1. Klik “Go to Console” yang berada pada bagian kanan atas.
2. Buat Project baru dengan klik “Create Project” atau “Add Project” jika sudah pernah membuat project sebelumnya.
3. Masukkan nama project Anda dan klik “Continue”.
4. Anda bisa meng-enable “Google Analytics” dan klik “Continue”.
5. Kemudian Anda akan diminta untuk memilih akun yang digunakan untuk menerima data dari Google Analytics, pilih “Default Account for Firebase” saja, selanjutnya klik “Create Project”.

6. Atau Anda bisa meng-disable “Google Analytic” dan klik “Create Project”.
7. Selanjutnya Anda tinggal menunggu project selesai dibuat.



tampilan awal firebase

untuk pembuatan project klik tombol tambah



setelah add akan muncul seperti ini dan bisa di continue

The screenshot shows the Firebase Project Overview page for a project named 'Auth Flutter'. The left sidebar lists various services: Authentication, Firestore, Realtime Database, Storage, Hosting, Functions, and Machine Learning. The main content area features a blue background with the text 'Auth Flutter' and 'Spark plan'. It includes a call-to-action 'Get started by adding Firebase to your app' with icons for iOS, Android, web, and Windows. Below this is a button 'Add an app to get started' and an illustration of two people interacting with a screen.

dan ini adalah tampilan dashboard project firebase

The screenshot shows the main navigation menu of the Firebase Project Overview. The menu items include 'Project Overview' (selected), 'Build' (with a dropdown arrow), 'Authentication', 'Firestore', 'Realtime Database', 'Storage', 'Hosting', 'Functions', and 'Machine Learning'. Each item has a corresponding icon next to it.

dan ini merupakan macam macam service yang di tawarkan oleh firebase

1. Authentication

Authentication mencakup proses authentication mulai dari masuk menggunakan email dan password, google, facebook dapat di handle oleh servis ini

2. Firestore

firestore mencakup database sekaligus storage atau gabungan dari realtime database dan storage

3. Realtime Database

realtime database adalah database yang memiliki bentuk json dalam penyimpanannya sehingga sangat fleksibel dengan data, dan tidak terbatas dengan table dan relational nya

4. Storage

storage merupakan proses untuk menyimpan asset yang berupa data gambar, video, music dan lain lain, dapat disimpan di servis ini, dan setiap item yang di simpan memiliki url sendiri sendiri berdasarkan id yang telah di generate oleh firebase

5. Hosting

6. Hosting adalah servis yang disediakan firebase untuk kita mengupload file web kita yang berupa html

7. Functions

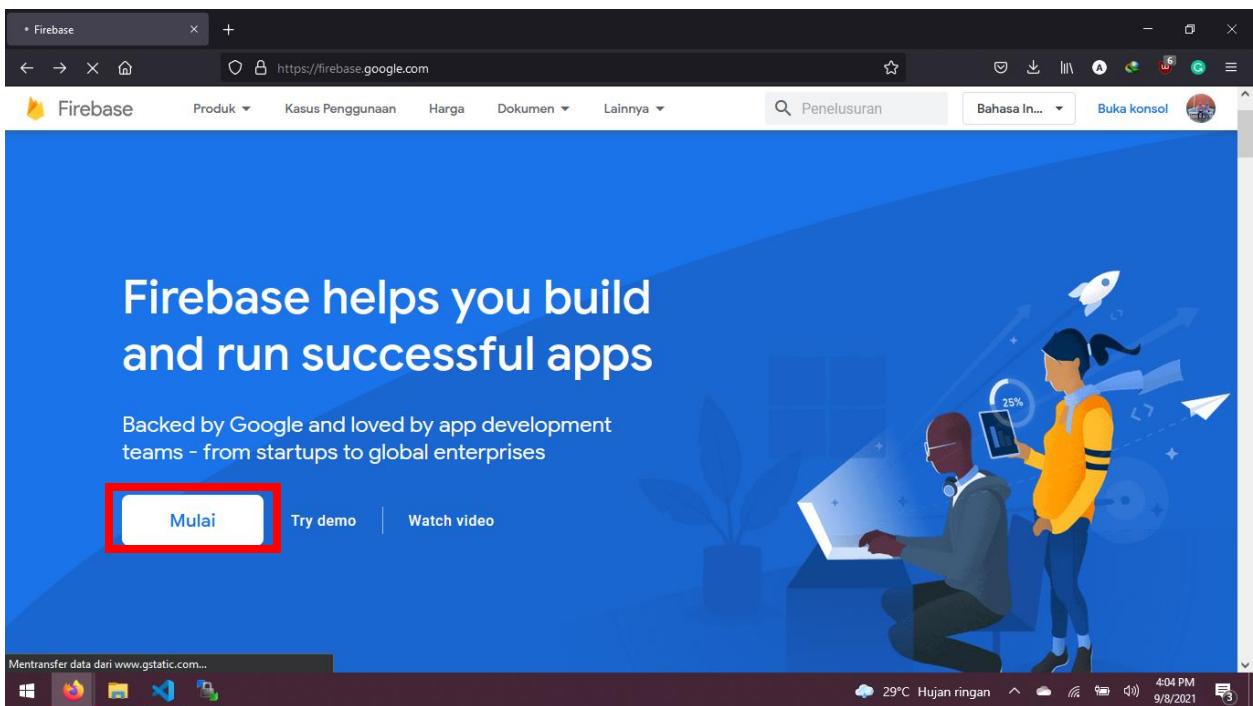
8. Machine Learning

d. Alat dan Bahan

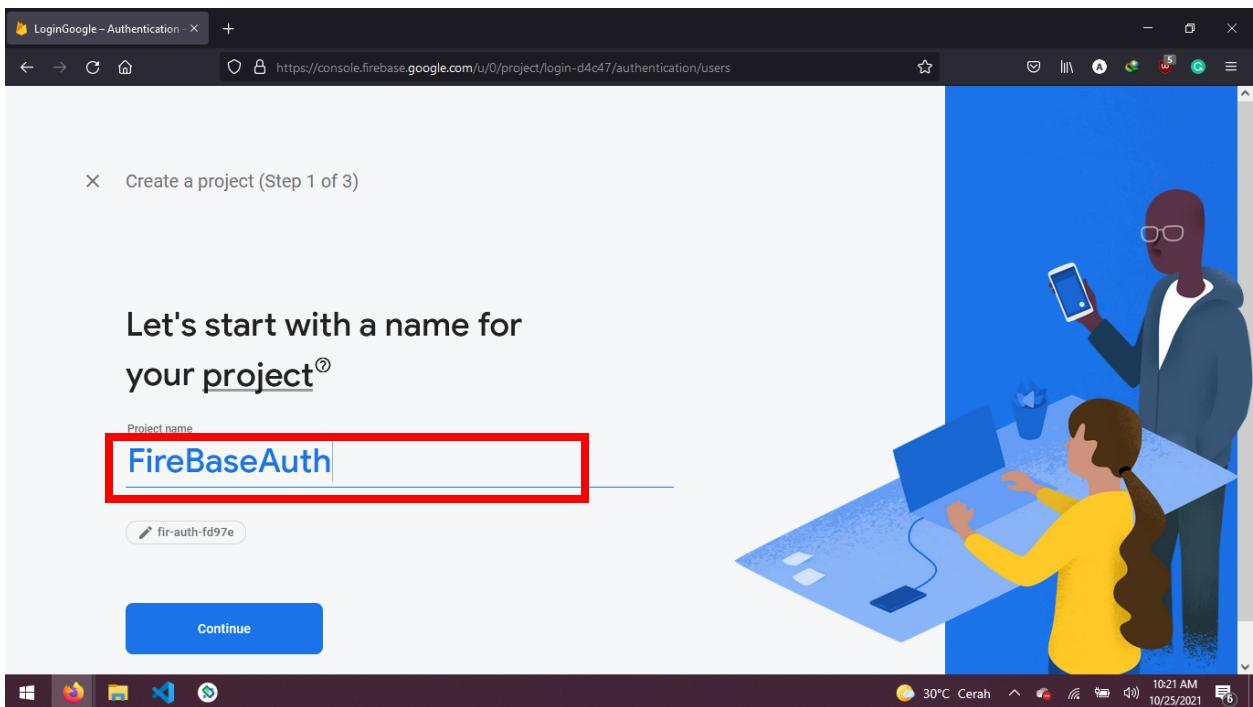
1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

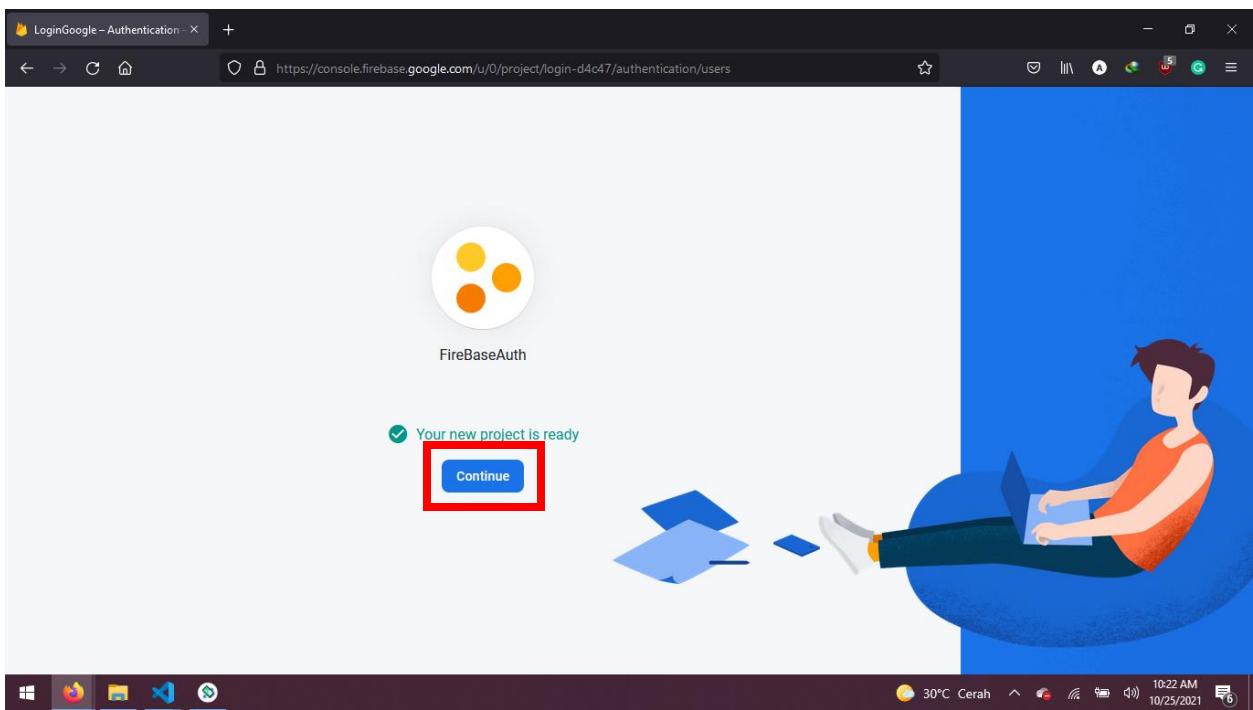
1. Membuat project flutter dengan nama minggu_11
2. Buka link berikut : <https://firebase.google.com/>



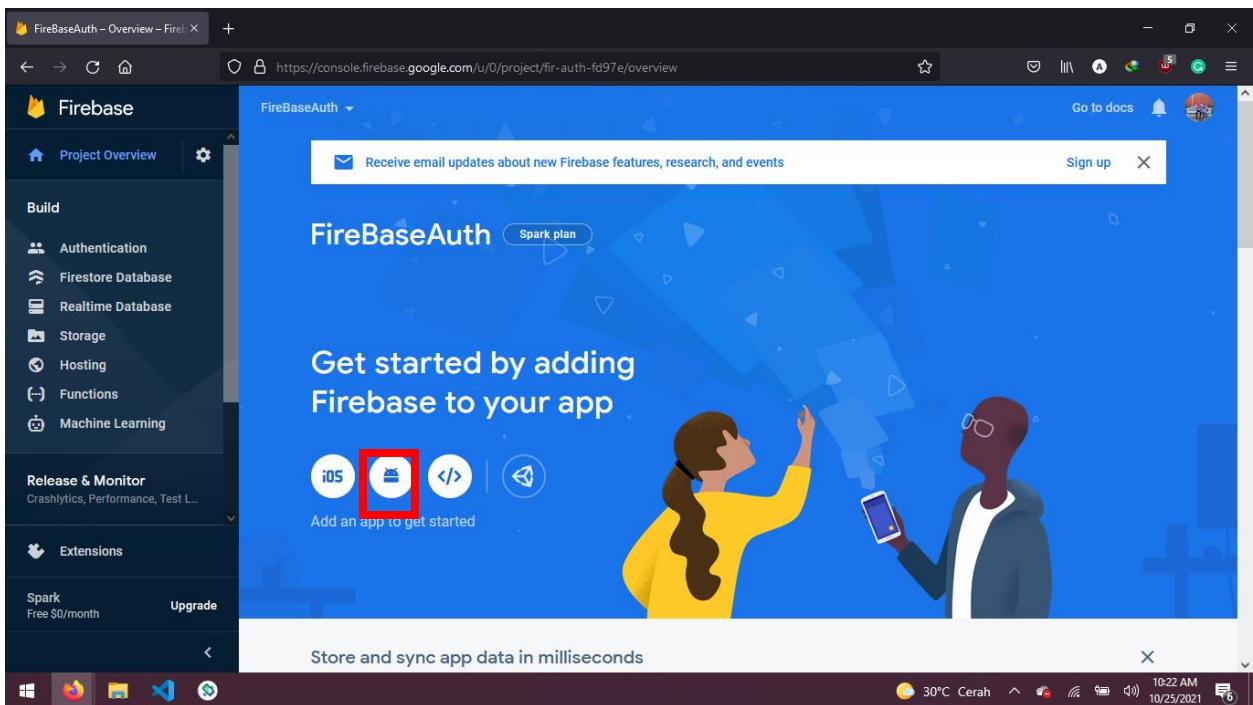
2. Membuat Project dengan nama “LoginGoogle”, kemudian tekan tombol **continue**.



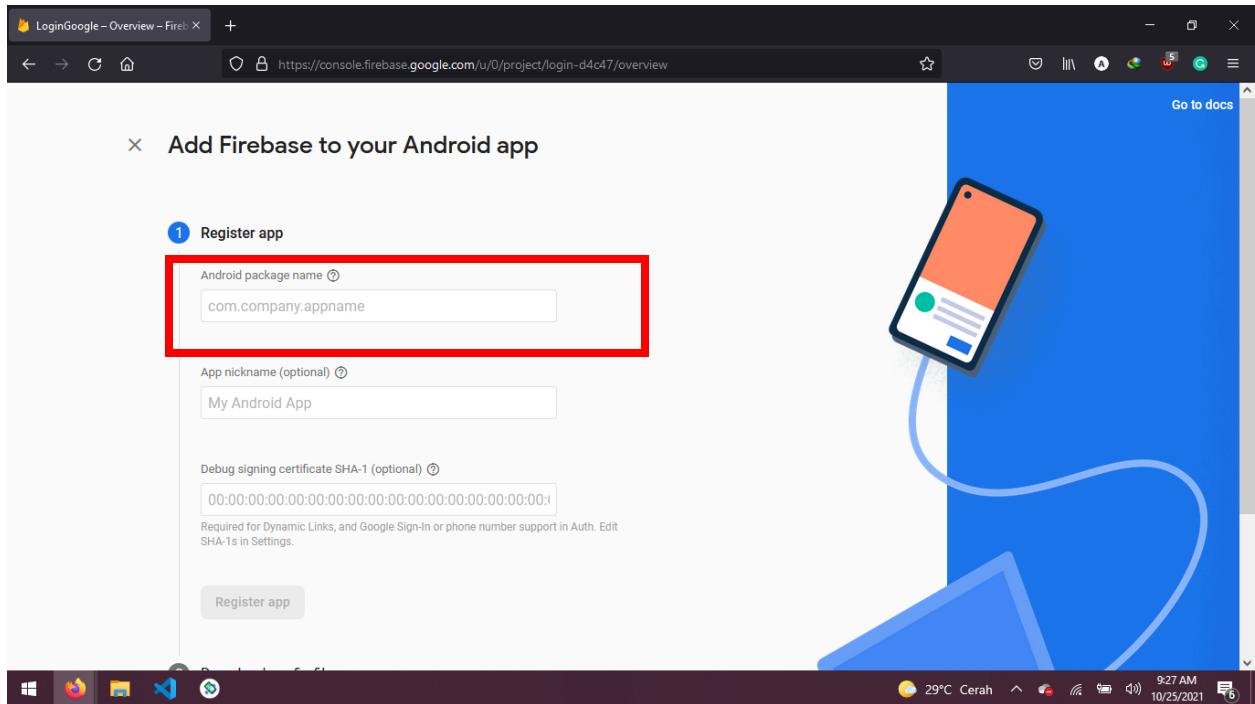
3. Tekan tombol **continue** hingga step 3.



4. Memilih platform android.



5. Memasukkan android packagenya.



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.minggu_11">
    <application>
        <activity
            android:name=".MainActivity"
            android:label="minggu_11"
            android:icon="@mipmap/ic_launcher">
            <activity
```

The screenshot shows the Android Studio interface. On the left, the project structure for 'MINGGU_11' is displayed, with 'main' selected. In the center, the code editor shows the 'AndroidManifest.xml' file. The line 'package="com.example.minggu_11"' is highlighted with a red box. The code editor also shows other XML attributes for the application and activity tags.

package com.example.minggu_11 digunakan untuk mendaftarkan apps kita di firebase dan mendapatkan google-service.json

6. download google-services.json

2 Download config file

Instructions for Android Studio below | [Unity](#) [C++](#)

Switch to the Project view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.

[Download google-services.json](#)

[Next](#)

MyApplication (~/Desktop/MyApplication)

Project Packages Scratch

MyApplication (~/Desktop/MyApplication)

build.gradle .idea

app build libs src .gitignore app.iml build.gradle google-services.json proguard-rules.pro gradle

capture

kemudian pindahkan file ke dalam **C:\Users\ARVITA\dev\project\minggu_11\android\app** sesuaikan **C:\Users\ARVITA\dev\project** dengan lokasi project kalian masing-masing

ARVITA > dev > project > minggu_11 > android > app >			
Name	Date modified	Type	Size
src	2/17/2022 8:40 AM	File folder	
build.gradle	2/17/2022 8:40 AM	GRADLE File	2 KB
google-services.json	11/5/2021 11:30 AM	JSON File	1 KB

7. Tekan tombol next

3 Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)

The Google services plugin for [Gradle](#) loads the google-services.json file you just downloaded. Modify your build.gradle files to use the plugin.

Project-level build.gradle (<project>/build.gradle):

```
buildscript {  
    repositories {  
        // Check that you have the following line (if not, add it):  
        google() // Google's Maven repository  
    }  
    dependencies {  
        ...  
        // Add this line  
        classpath 'com.google.gms:google-services:4.3.10'  
    }  
}
```

```
allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        ...
    }
}
```

App-level build.gradle (<project>/<app-module>/build.gradle):

App-level build.gradle (<project>/<app-module>/build.gradle):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:28.4.2')

    // Add the dependencies for the desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

Finally, press "Sync now" in the bar that appears in the IDE:

Gradle files have changed since last sync [Sync now](#)

Previous

Next

8. Tekan tombol continue to console

4 Next steps

You're all set!

Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

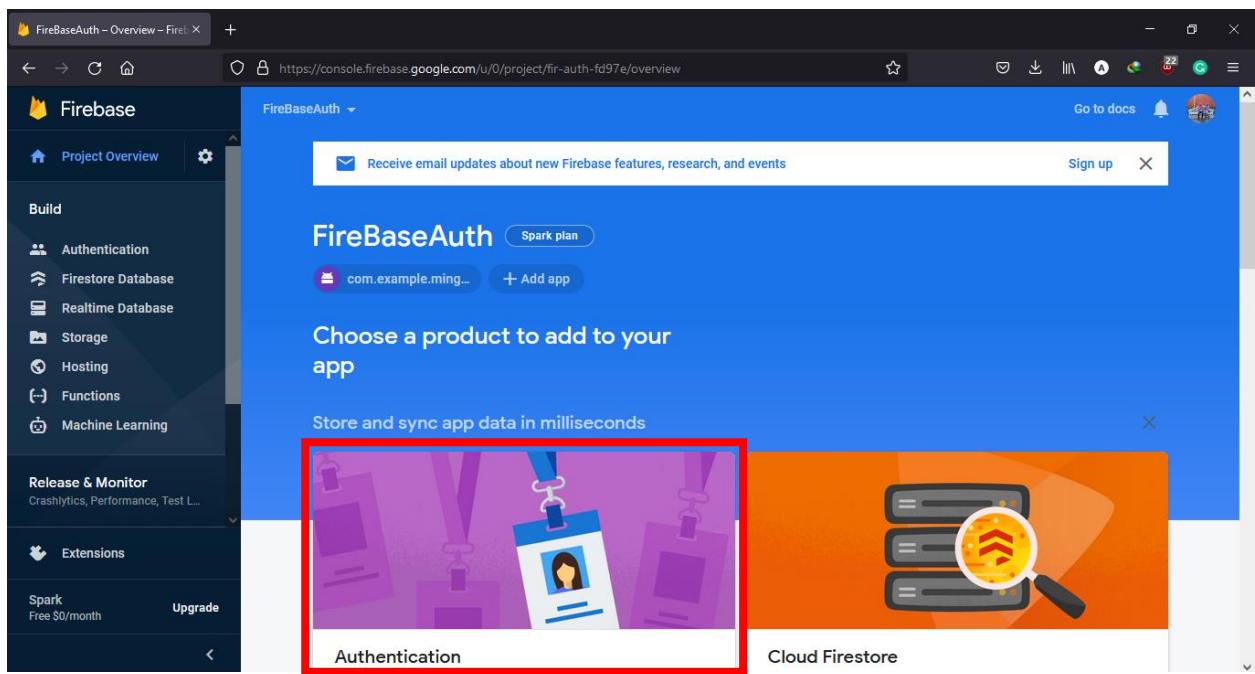
You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

Previous

Continue to console

9. Memilih product Authentication



10. Pilih sign-in methode kemudian enable Email/Password dan kemudian Save.

A screenshot of the 'Authentication' settings page in the Firebase console. The 'Sign-in method' tab is selected. Under 'Sign-in providers', the 'Email/Password' provider is listed with its 'Enable' switch turned on. Below it, the 'Email link (passwordless sign-in)' provider has its switch turned off. At the bottom right, there are 'Cancel' and 'Save' buttons, with 'Save' being highlighted with a red box.

11. Tambahkan User dan password

The screenshot shows the Firebase Authentication interface under the 'Users' tab. A search bar at the top allows searching by email address, phone number, or user UID. Below the search bar is a table with columns: Identifier, Providers, Created, Signed In, and User UID. One user is listed: aqwuarvitha@gmail.com, with a dropdown arrow next to the Providers column, created on Oct 25, 2021, and a User UID of k3SYXI9jkgVldGOuWMO2DZlxD5j2. At the bottom of the table, there are pagination controls for 'Rows per page: 50' and '1 - 1 of 1'.

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS.

g. Kesimpulan

Mahasiswa mendapatkan memahami Authentication, serta implementasi code Authentication.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan implementasi pada konsep Authentication	1	2	3	(4)
2	Ketepatan waktu dan ketepatan implementasi pada konsep Authentication	1	2	(3)	4
3	Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Authentication	1	(2)	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Authentication	(1)	2	3	4
Jumlah skor					

ACARA 38 : AUTHENTICATION

Pokok Bahasan	: Pengambilan API
Acara Praktikum/Pertemuan	: Minggu 11 / 2
Tempat	: Daring / Luring Politeknik Negeri Jember
Alokasi Waktu	: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Authentication.
2. Mahasiswa mampu menerapkan Authentication

b. Indikator

Keberhasilan mahasiswa memahami konsep Authentication dan menerapkan Authentication pada flutter.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. minggu_11->android->build.gradle

```
8     dependencies {  
9         classpath 'com.android.tools.build:gradle:4.1.0'  
10        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
11        classpath 'com.google.gms:google-services:4.3.10'  
12    }  
13 }
```

2. Minggu_11->android->app->build.gradle

```
54 flutter {  
55     source '.../...'  
56 }  
57  
58 dependencies {  
59     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
60     implementation platform('com.google.firebaseio:firebase-bom:28.4.2')  
61 }  
62
```

3. Minggu_11->pubspec.yaml

```
environment:  
|   sdk: ">=2.12.0 <3.0.0"  
  
dependencies:  
|   cupertino_icons: ^1.0.2  
|   firebase_auth: ^3.1.2  
|   firebase_core: ^1.7.0  
|   flutter:  
|       sdk: flutter  
|       http: ^0.13.3  
  
dev_dependencies:  
|   flutter_test:  
|       sdk: flutter
```

4. Merubah main.dart seperti berikut :

```
7 Future<void> main() async {  
8     WidgetsFlutterBinding.ensureInitialized();  
9     await Firebase.initializeApp();  
10    runApp(MyApp());  
11 }  
12  
13 class MyApp extends StatelessWidget {  
14     // This widget is the root of your application.  
15     @override  
16     Widget build(BuildContext context) {  
17         return MaterialApp(  
18             title: 'Flutter Demo',  
19             theme: ThemeData(  
20                 primarySwatch: Colors.blue,  
21             ), // ThemeData  
22             // home: LoginScreen(),  
23             home: LoginScreen(),  
24         ); // MaterialApp  
25     }  
26 }
```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Authentication, serta implementasi code Authentication.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan implementasi pada konsep Authentication	1	2	3	(4)
2	Ketepatan waktu dan ketepatan implementasi pada konsep Authentication	1	2	(3)	4
3	Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Authentication	1	(2)	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Authentication	(1)	2	3	4
Jumlah skor					

ACARA 39 : AUTHENTICATION

Pokok Bahasan	: Authentication
Acara Praktikum/Pertemuan	: Minggu 11 / 3
Tempat	: Daring / Luring Politeknik Negeri Jember
Alokasi Waktu	: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Authentication.
2. Mahasiswa mampu menerapkan Authentication

b. Indikator

Keberhasilan mahasiswa memahami konsep Authentication dan menerapkan Authentication pada flutter.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Menambahkan file dart baru yaitu LoginScreen.dart seperti berikut :

```
6  class LoginScreen extends StatefulWidget {
7    @override
8    _LoginScreenState createState() => _LoginScreenState();
9  }
10
11 class _LoginScreenState extends State<LoginScreen> {
12   final TextEditingController _emailController = TextEditingController();
13   final TextEditingController _passwordController = TextEditingController();
14
15   final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;
```

```

16  registerSubmit() async {
17      try {
18          await _firebaseAuth.createUserWithEmailAndPassword(
19              email: _emailController.text.toString().trim(),
20              password: _passwordController.text);
21      } catch (e) {
22          print(e);
23          SnackBar(content: Text(e.toString()));
24      }
25  }
26
27  loginSubmit() async {
28      try {
29          _firebaseAuth
30              .signInWithEmailAndPassword(
31                  email: _emailController.text, password: _passwordController.text)
32              .then((value) => Navigator.of(context).pushReplacement(
33                  MaterialPageRoute(builder: (context) => HomeScreen())));
34      } catch (e) {
35          print(e);
36          SnackBar(content: Text(e.toString()));
37      }
38  }
39
40  @override
41  Widget build(BuildContext context) {
42      return Scaffold(
43          body: Padding(
44              padding: EdgeInsets.all(10),
45              child: ListView(
46                  children: <Widget>[
47                      Container(
48                          alignment: Alignment.center,
49                          padding: EdgeInsets.all(10),
50                          margin: const EdgeInsets.only(top: 40),
51                          child: Text(
52                              "Latihan Auth",
53                              style: TextStyle(
54                                  color: Colors.blue,
55                                  fontWeight: FontWeight.w500,
56                                  fontSize: 30), // TextStyle
57                          ), // Text
58                      ), // Container
59                      Padding(
60                          padding: const EdgeInsets.all(8),
61                          child: Image.asset(
62                              "assets/img/flutter.png",
63                              height: 100,
64                              width: 100,

```

```

65     ), // Image.asset
66     ], // Padding
67     Container(
68       padding: EdgeInsets.all(10),
69       child: TextField(
70         controller: _emailController,
71         decoration: InputDecoration(
72           border: OutlineInputBorder(), labelText: "Username"), //
73         ), // TextField // Container
74     Container(
75       padding: EdgeInsets.all(10),
76       child: TextField(
77         controller: _passwordController,
78         obscureText: true,
79         decoration: InputDecoration(
80           border: OutlineInputBorder(), labelText: "Password"), //
81         ), // TextField // Container
82     TextButton(onPressed: () {}, child: Text("Forgot password")),
83     Container(
84       height: 50,
85       padding: EdgeInsets.fromLTRB(10, 0, 10, 0),
86       child: ElevatedButton(
87         style: raisedButtonStyle,
88         child: Text("Register"),
89         onPressed: () {
90           registerSubmit();
91         },
92       ), // ElevatedButton
93     ), // Container
94     SizedBox(height: 10),
95     Container(
96       height: 50,
97       padding: EdgeInsets.fromLTRB(10, 0, 10, 0),
98       child: ElevatedButton(
99         style: raisedButtonStyle,
100        child: Text("Login"),
101        onPressed: () {
102          loginSubmit();
103        },
104      ), // ElevatedButton
105    ), // Container
106  ], // <Widget>[]
107 ), // ListView
108 ), // Padding
109 ); // Scaffold
110 }
111 }

```

```

113   final ButtonStyle raisedButtonStyle = ElevatedButton.styleFrom(
114     onPrimary: Colors.grey[300],
115     primary: Colors.blue[300],
116     minimumSize: Size(88, 36),
117     padding: EdgeInsets.symmetric(horizontal: 16),
118     shape: const RoundedRectangleBorder(
119       borderRadius: BorderRadius.all(Radius.circular(6)),
120     ); // RoundedRectangleBorder
121

```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Authentication, serta implementasi code Authentication.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan implementasi pada konsep Authentication	1	2	3	(4)
2	Ketepatan waktu dan ketepatan implementasi pada konsep Authentication	1	2	(3)	4
3	Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Authentication	1	(2)	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Authentication	(1)	2	3	4
Jumlah skor					

ACARA 40 : AUTHENTICATION

Pokok Bahasan	: Pengujian JSON API MovieDB melalui Postman
Acara Praktikum/Pertemuan	: Minggu 11 / 4
Tempat	: Daring / Luring Politeknik Negeri Jember
Alokasi Waktu	: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Authentication.
2. Mahasiswa mampu menerapkan Authentication

b. Indikator

Keberhasilan mahasiswa memahami konsep Authentication dan menerapkan Authentication pada flutter.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Menambahkan file dart baru yaitu HomeScreen.dart seperti berikut :

```
5
6  class HomeScreen extends StatelessWidget {
7    Future<void> _signOut() async {
8      await FirebaseAuth.instance.signOut();
9    }
10
11   @override
12   Widget build(BuildContext context) {
13     FirebaseAuth auth = FirebaseAuth.instance;
14     if (auth.currentUser != null) {
15       print(auth.currentUser!.email);
16     }
17     return Scaffold(
18       body: Padding(
```

```

19     padding: const EdgeInsets.all(30.0),
20     child: Column(
21       crossAxisAlignment: CrossAxisAlignment.start,
22       children: [
23         SizedBox(height: 60),
24         Row(
25           mainAxisAlignment: MainAxisAlignment.end,
26           children: [
27             IconButton(icon: Icon(Icons.notifications), onPressed: () {}),
28             IconButton(icon: Icon(Icons.extension), onPressed: () {})
29           ],
30         ), // Row
31         SizedBox(height: 37),
32         Text.rich(
33           TextSpan(
34             children: <TextSpan>[
35               TextSpan(
36                 text: "Welcome, \n",
37                 style: TextStyle(color: Colors.blue[300]),
38               ), // TextSpan
39               TextSpan(
40                 text: auth.currentUser!.email,
41                 style: TextStyle(color: Colors.blue[900]),
42               ), // TextSpan
43             ], // <TextSpan>[]
44           ), // TextSpan
45           style: TextStyle(fontSize: 30),
46         ), // Text.rich
47         SizedBox(height: 20),
48         TextField(
49           decoration: InputDecoration(
50             prefixIcon: Icon(Icons.search, size: 18),
51             border: OutlineInputBorder(
52               borderRadius: BorderRadius.circular(10)), // OutlineInputBorder
53             hintText: "Search"), // InputDecoration
54         ), // TextField
55         SizedBox(height: 10),
56         Text(
57           "Recomended Place",
58           style: TextStyle(fontWeight: FontWeight.w600, fontSize: 20),
59         ), // Text
60         Container(
61           child: ElevatedButton(
62             onPressed: () {
63               signOut().then((value) => Navigator.of(context)
64                 .pushReplacement(MaterialPageRoute(
65                   builder: (context) => LoginScreen())));
66             },
67             child: Text('Logout'),
68           ), // ElevatedButton // Container
69         ],
70       ), // Column
71     ), // Padding
72   ); // Scaffold
73 }
74 }
```

```

75
76     final countries = ["Tokyo", "Berlin", "Roma", "Monas"];
77

```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Authentication, serta implementasi code Authentication.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan implementasi pada konsep Authentication	1	2	3	(4)
2	Ketepatan waktu dan ketepatan implementasi pada konsep Authentication	1	2	(3)	4
3	Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Authentication	1	(2)	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Authentication	(1)	2	3	4
Jumlah skor					