

LAPORAN PRAKTIKUM



NIM : 2003071

Nama : Hakim Asrori

Kelas : D3TI.2C

Mata Kuliah : [Pemrograman Perangkat Bergerak \(TIU3403\)](#)

Praktikum ke / Judul : 10 / FIREBASE

Tanggal Praktikum : 19 Mei 2022

Dosen Pengampu : Fachrul Pralienka Bani Muhamad, S.ST.,
M.Kom

**PROGRAM STUDI D3 TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU
2022**



A. Tujuan Praktikum

Tujuan Umum

Mahasiswa dapat memanfaatkan service yang disediakan Firebase untuk keperluan pembuatan aplikasi mobile dengan pengalaman pengguna yang lebih baik

Tujuan Khusus

Mahasiswa mampu:

1. Menjelaskan perbedaan database NoSQL dan relational
2. Menjelaskan keunggulan dan kelemahan Firebase
3. Mengimplementasikan service yang disediakan Firebase

B. Rangkuman Teori Singkat

Perbandingan Database Relational dan non-Relational

Database Relational adalah database digital yang didasarkan pada model data relational. Database Relational digunakan untuk menyimpan data yang terstruktur ke dalam tabel. Beberapa tabel juga dapat berisi kolom yang sama.

Database Non-Relational merupakan penyimpanan data dalam format kunci-nilai, dalam dokumen atau dengan metode lain tanpa menggunakan tabel seperti database relational.

Perbandingannya ialah:

1. Database Relational
 - 1.1. Difokuskan pada kumpulan dokumen
 - 1.2. Basis data yang berpasangan nilai-kunci
 - 1.3. Penyimpanan data hirarkis
2. Database Non-Relational
 - 2.1. Menggunakan SQL untuk mendefinisikan
 - 2.2. Basis data yang berbentuk tabel
 - 2.3. Tidak ada penyimpanan hirarkis

Firestore

Firestore merupakan salah satu layanan dari Google untuk memudahkan para developer aplikasi dalam pengembangannya. Dengan menggunakan firestore, para developer bisa fokus dalam mengembangkan aplikasi tanpa memberikan *effort* yang besar untuk urusan *backend*.

Service Pada Firestore

1. Firestore Analytics. Merupakan service yang digunakan sebagai koleksi data dan reporting untuk aplikasi Android maupun iOS.
2. Firestore Cloud Messaging and Notifications. Merupakan service yang digunakan untuk mengirim dan menerima pesan serta notifikasi di Android, iOS, dan web tanpa perlu biaya.



3. Firebase Authentication. Merupakan service yang digunakan sebagai backend antara pengguna dengan sistem melalui perantara login
4. Firebase Cloud Firestore. Merupakan service yang digunakan sebagai penyimpanan data yang bersifat fleksibel dan terukur.
5. Firebase Realtime Database. Merupakan service yang digunakan sebagai database yang di hosting elalui cloud. Data akan disimpan dan dieksekusi dama bentuk JSON dan disinkronkan secara realtime ke setiap user yang terkoneksi.

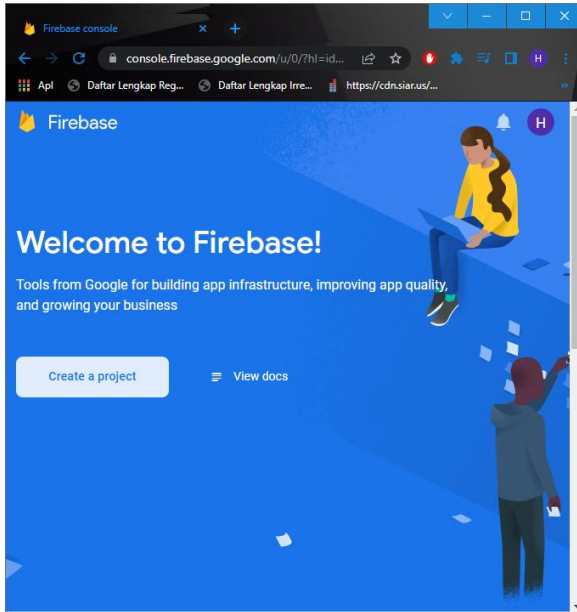


C. Langkah dan Hasil Pelaksanaan Praktikum

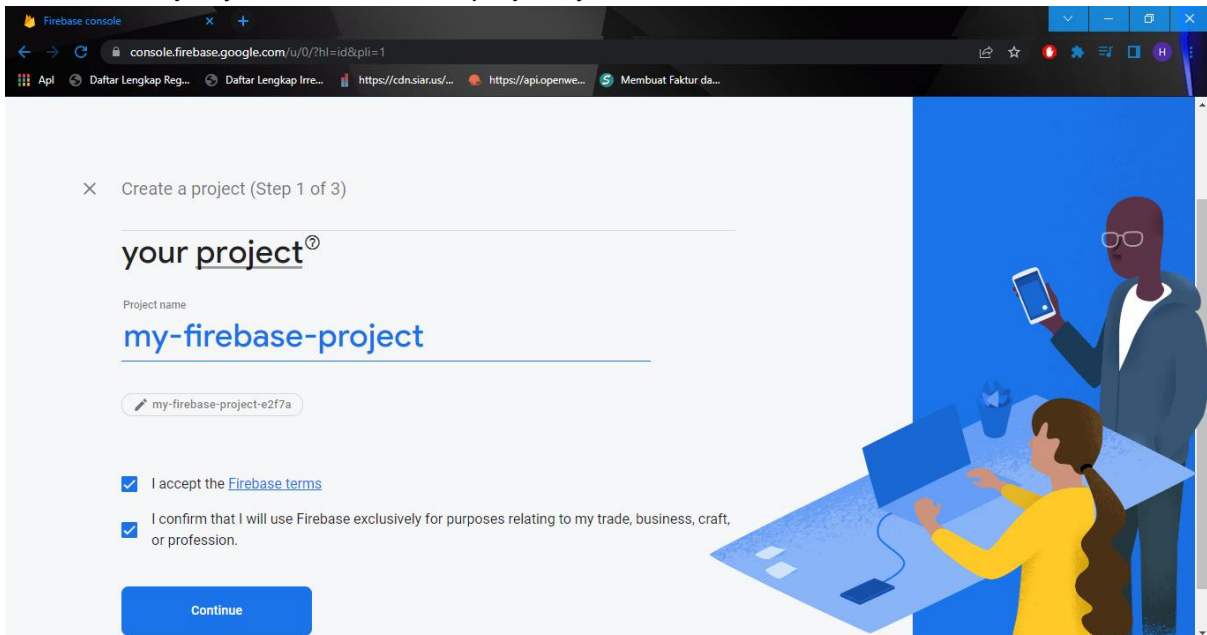
Langkah-langkah praktikum Setup Firebase

1. Buka halaman firebase console melalui url berikut.

<https://console.firebase.google.com/u/0/>

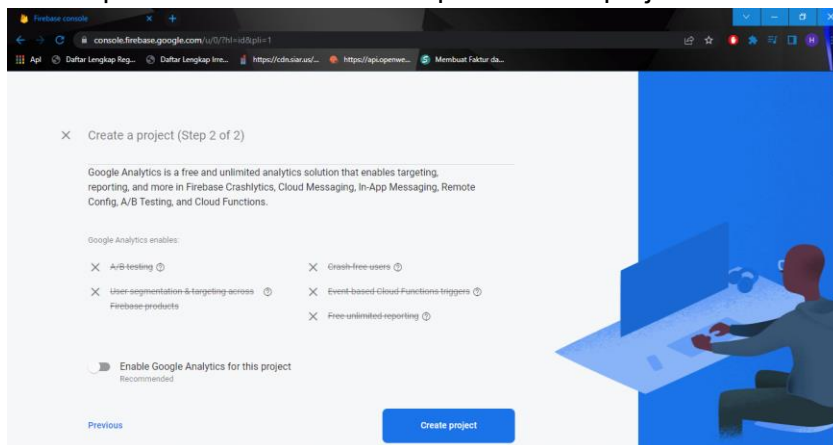


2. Untuk membuat project pertama kali, pilih “Create a project”
3. Selanjutnya masukan nama projectnya.

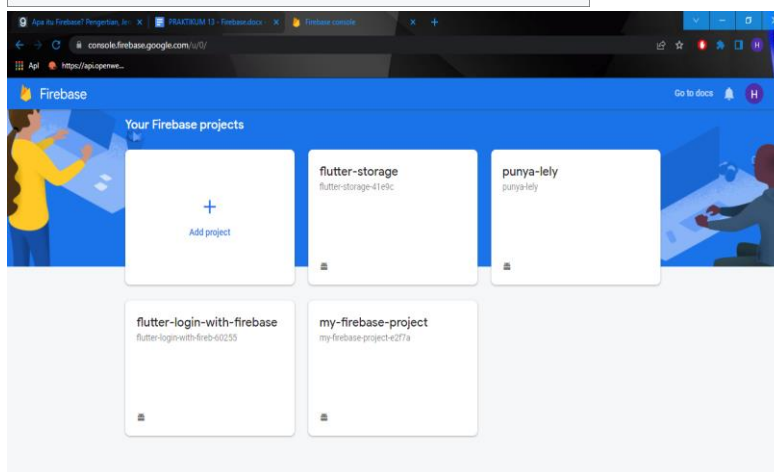
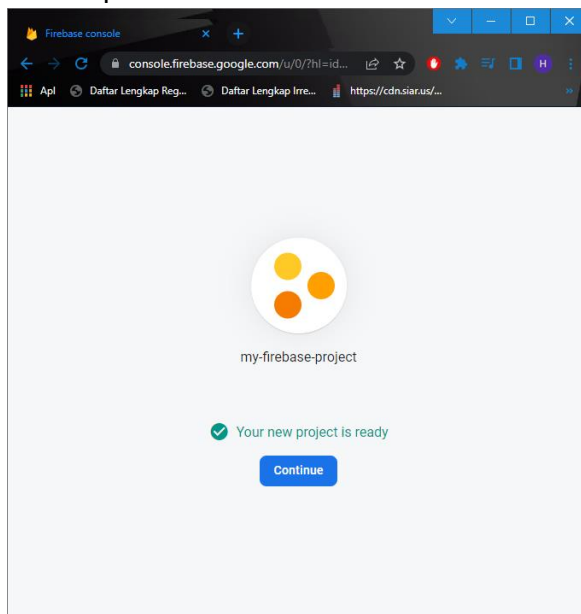




4. Selanjutnya pilihan optional, secara default “Enable Google Analytics” ini aktif atau posisi enable. Setelah itu pilih “Create project”.

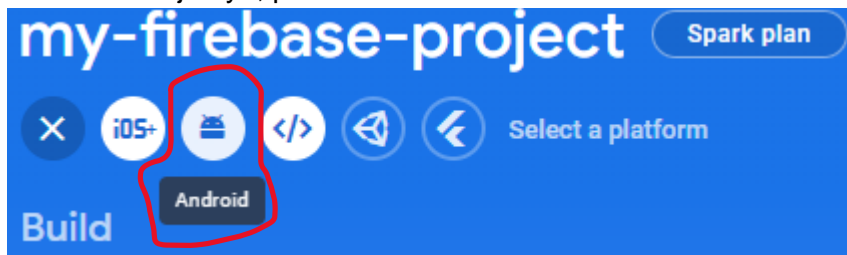


5. Setelah tahapan diatas selesai, maka hasilnya akan seperti berikut ini. Selanjutnya pilih “Continue”

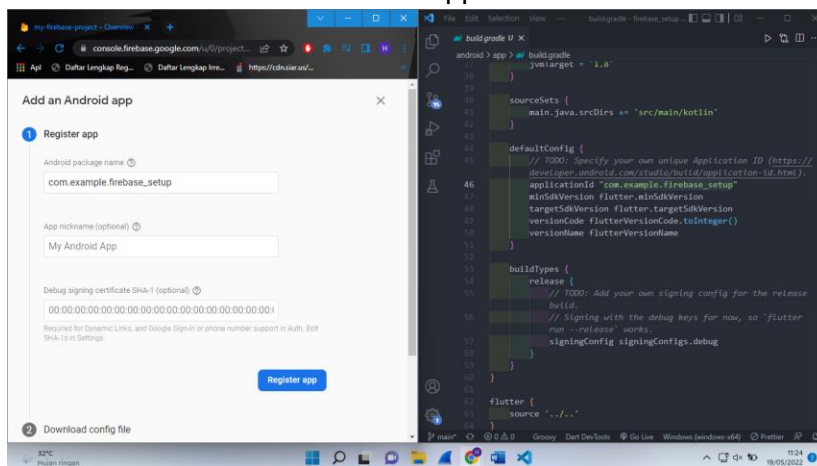




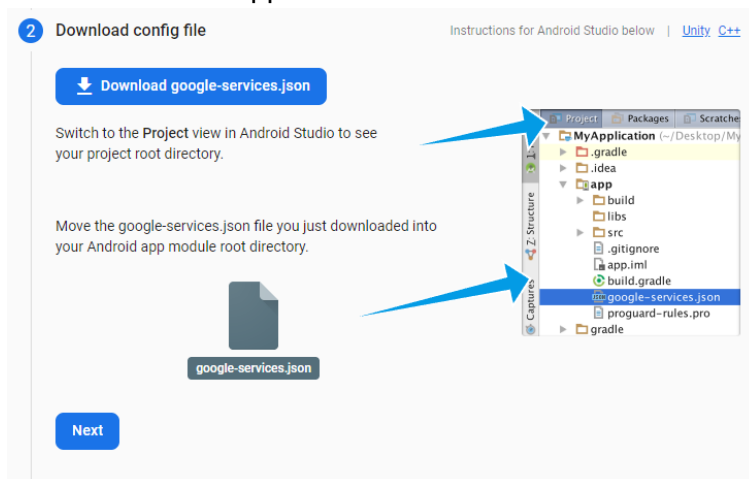
6. Selanjutnya, kita pilih project yang telah dibuat.
7. Selanjutnya, pilih icon android.

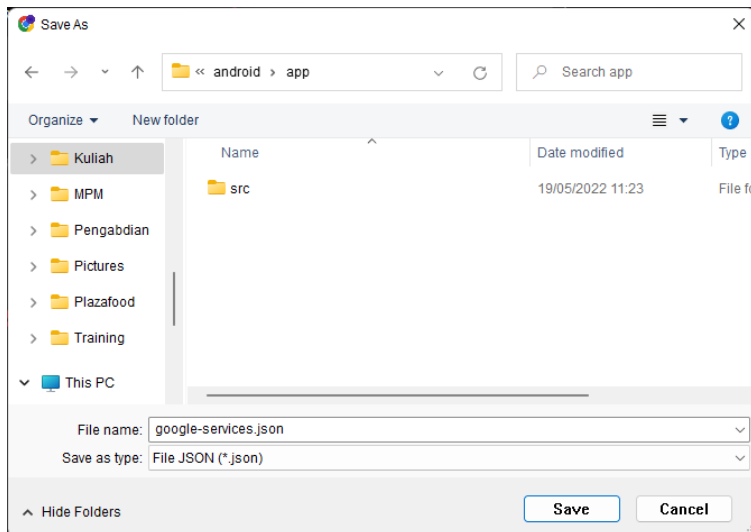


8. Selanjutnya, masukan package name yang sesuai dengan build. gradle app yang ada di flutter > android > app.

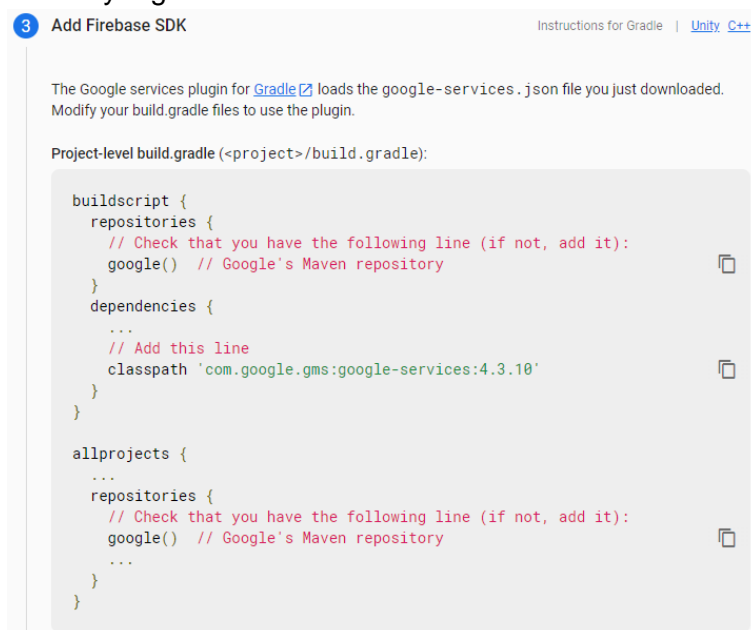


9. Selanjutnya, download file json tersebut dan masukan kedalam folder flutter > android > app.





10. Selanjutnya, masukan classpath dependencies tersebut kedalam file build.gradle yang berada didalam folder flutter > android.





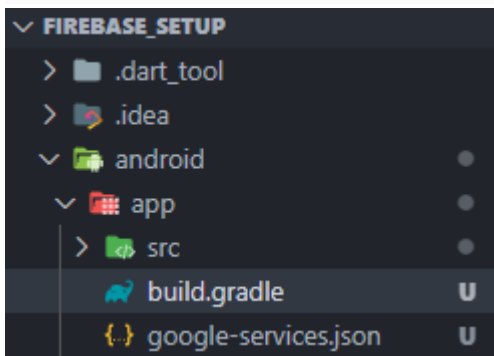
11. Selanjutnya, masukan apply plugin tersebut kedalam file build. gradle yang berada didalam folder flutter > android > app.

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

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

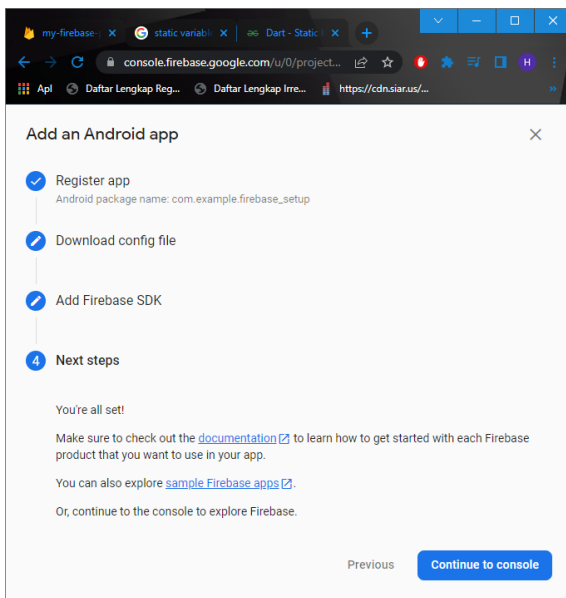
dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:30.0.1')

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



```
66 dependencies {
67     implementation "org.jetbrains.
68         kotlin:kotlin-stdlib-jdk7:$kotlin_version"
69 }
70 apply plugin: 'com.google.gms.google-services'
```

12. Setelah tahapan diatas selesai, maka tampilannya seperti berikut. Selanjutnya pilih “continue to console”.





13. Setelah tahapan diatas selesai, maka telah selesai semua konfigurasi dari firebase tersebut.

14. Selanjutnya, percobaan firebase yang akan diintegrasikan ke dalam flutter.

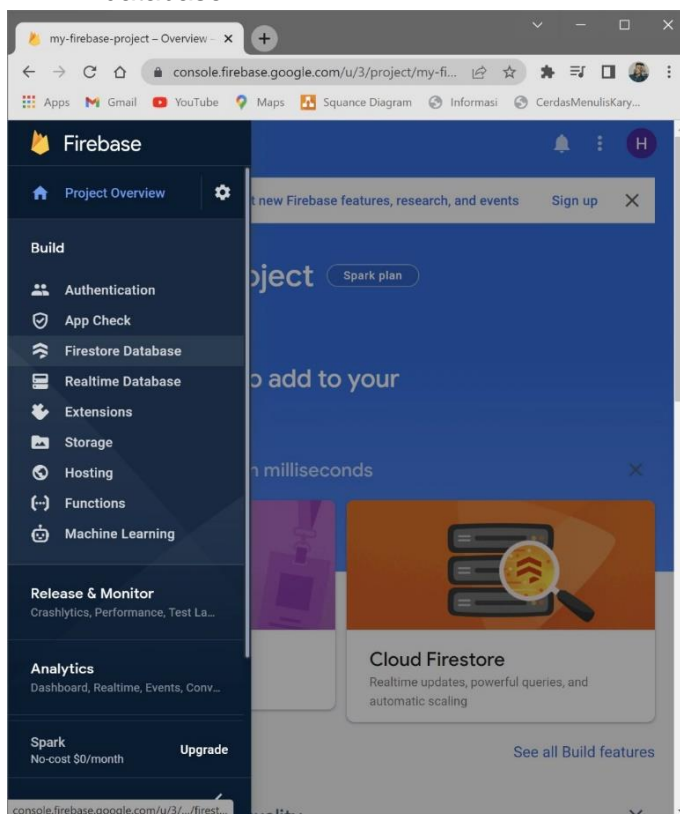
```
34 # The following adds the Cupertino Icons font to your
    application.
35 # Use with the CupertinoIcons class for iOS style icons.
36 cupertino_icons: ^1.0.2
37 cloud_firestore: ^3.1.15

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

Run | Debug | Profile
Future main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}
```

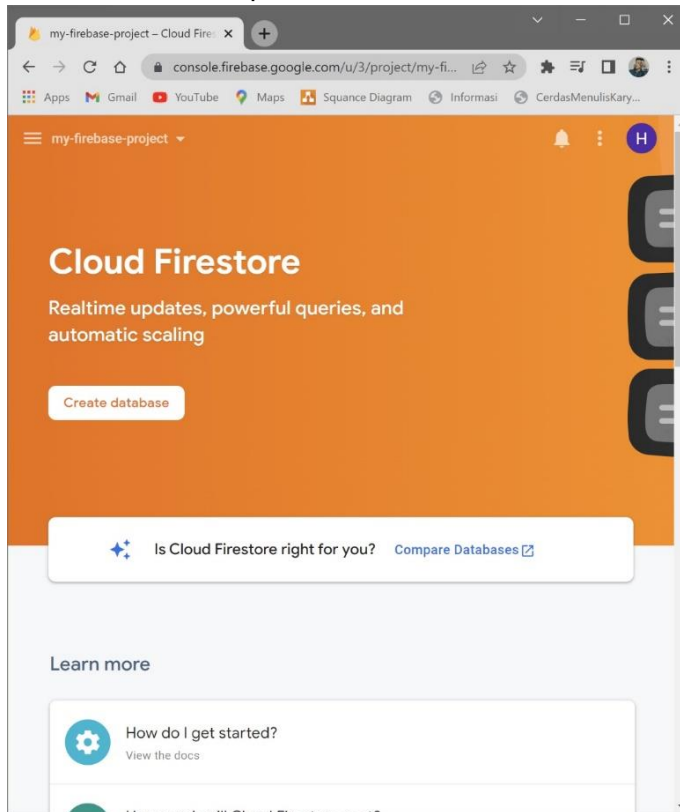
Langkah-langkah praktikum CRUD Firebase

1. Untuk melakukan praktikum CRUD Firebase, disini saya menggunakan project dari praktikum sebelumnya. Dan studi kasus yang saya pakai merupakan studi kasus dari authors lumen sebelumnya.
2. Buat penyediaan dari CRUD nanti di firebase yaitu menggunakan fitur firestore database.

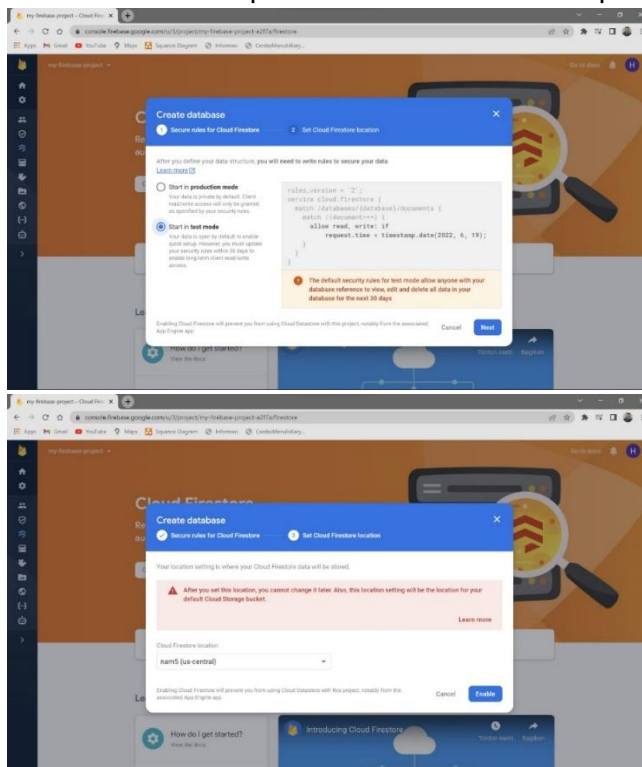


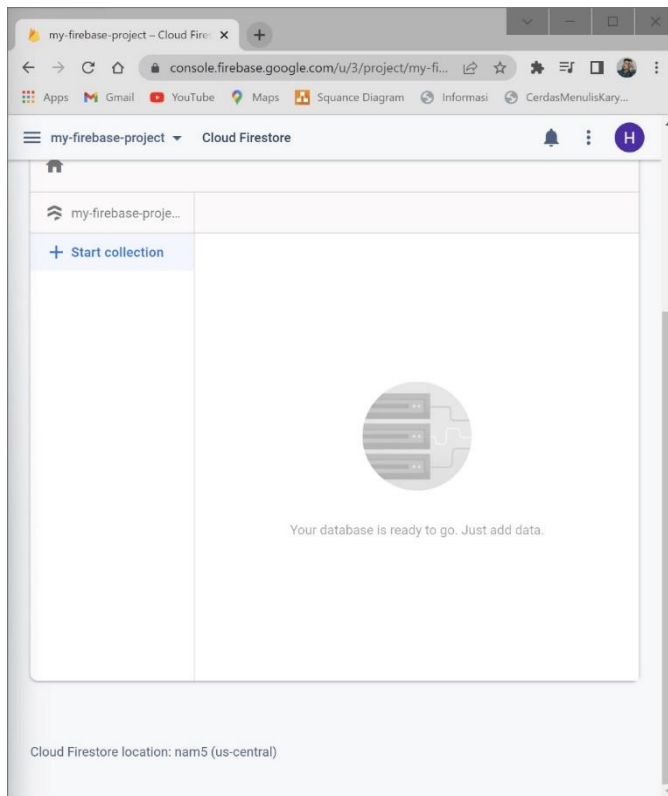


3. Setelah itu pilih create database.



4. Setelah itu pilih create database. Dan proses berikutnya seperti yang ada digambar





5. Setelah itu buka kode program sebelumnya, dan tambahkan kode program berikut didalam main.dart yang berfungsi untuk mengarahkan home nya ke HomeScreen()

```
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: "Authors",  
      debugShowCheckedModeBanner: false,  
      home: HomeScreen(),  
      localizationsDelegates: [GlobalMaterialLocalizations.delegate],  
      supportedLocales: [const Locale('id', 'ID')],  
    );  
  }  
}
```

6. Setelah itu tambahkan kode program berikut didalam class HomeScreen, yang berfungsi untuk mengambil data dari firebase

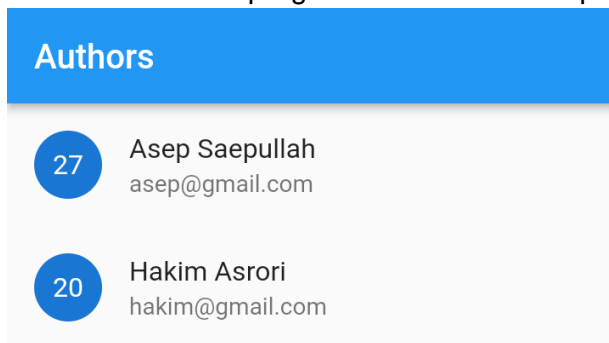


```
Stream<List<Author>> readAuthors() => FirebaseFirestore.instance
    .collection('authors')
    .snapshots()
    .map((snapshot) =>
        snapshot.docs.map((doc) => Author.fromJson(doc.data())).toList());
```

7. Setelah itu tambahkan kode program berikut didalam class HomeScreen, yang berfungsi untuk memunculkan data yang sudah diambil dari firebase

```
Widget buildAuthor(Author author) {
    return ListTile(
        leading: CircleAvatar(child: Text(author.age.toString())),
        title: Text(author.name.toString()),
        subtitle: Text(author.email.toString()),
        onTap: () {
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => DetailScreen(
                        id: author.id.toString(),
                    ),
                ),
            );
        },
    );
}
```

8. Hasil kode program diatas adalah seperti berikut



9. Selanjutnya adalah pembuatan kode program yang akan memunculkan masing-masing data yang kita pilih

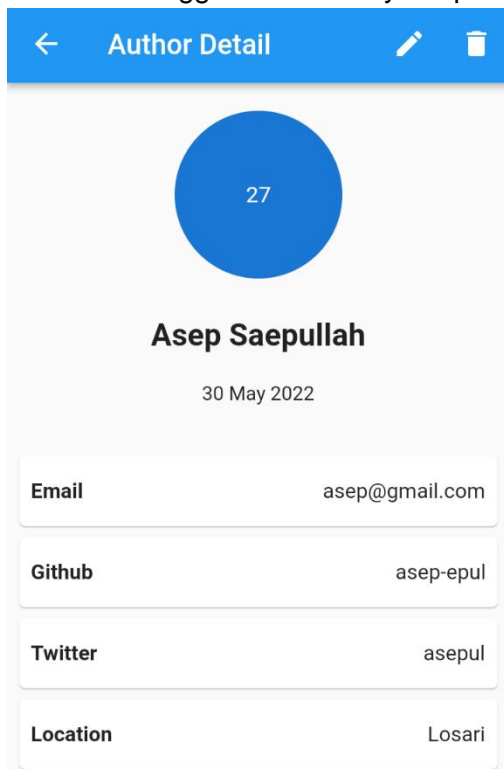


```
DetailScreen({required this.id});

Future<Author?> readAuthor() async {
  final docUser =
    FirebaseFirestore.instance.collection('authors').doc(this.id);
  final snapshot = await docUser.get();

  if (snapshot.exists) {
    return Author.fromJson(snapshot.data());
  }
}
```

10. Sehingga hasil akhirnya seperti berikut



11. Selanjutnya adalah kode program yang akan mengeksekusi penambahan data ke firebase



```
Future<void> saveAuthor() async {  
    final docAuthor = FirebaseFirestore.instance.collection('authors').doc();  
  
    final author = Author(  
        id: docAuthor.id,  
        name: _nameController.text,  
        email: _emailController.text,  
        github: _githubController.text,  
        twitter: _twitterController.text,  
        age: _ageController.text,  
        birthday: DateTime.parse(_birthdayController.text),  
        location: _locationController.text,  
    );  
  
    final json = author.toJson();  
  
    await docAuthor.set(json);  
}
```

12. Untuk tampilan dari upload datanya yaitu seperti berikut

← Add Author

Name

Email

Github

Twitter

Age

Birthday

Location

Save

13. Selanjutnya adalah kode program yang akan mengeksekusi perubahan data ke firebase



```
final docAuthor = FirebaseFirestore.instance
    .collection('authors')
    .doc("${author.id}");

docAuthor.update({
    "name": _nameController.text,
    "email": _emailController.text,
    "github": _githubController.text,
    "twitter": _twitterController.text,
    "age": _ageController.text,
    "birthday":
        DateTime.parse(_birthdayController.text),
    "location": _locationController.text,
});
```

14. Selanjutnya adalah kode program yang akan mengeksekusi penghapusan data ke firebase

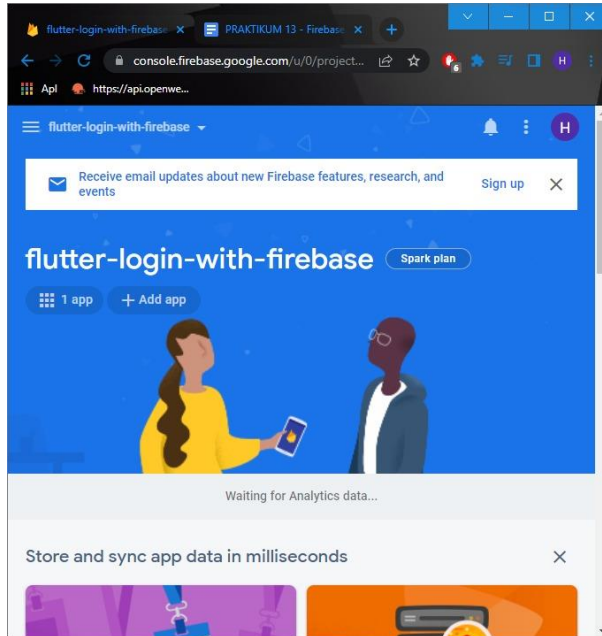
```
final docAuthor = FirebaseFirestore.instance
    .collection('authors')
    .doc(this.id);
docAuthor.delete();
```



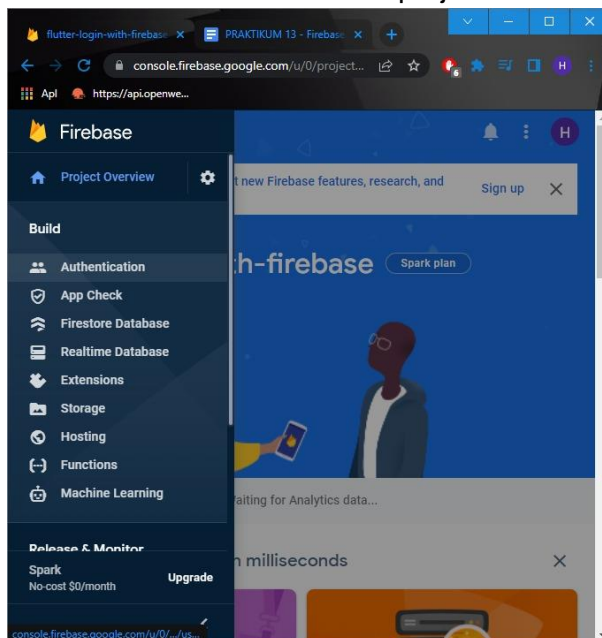
D. Langkah dan Hasil Latihan

Latihan Implementasi Firebase pada Studi kasus Authentication

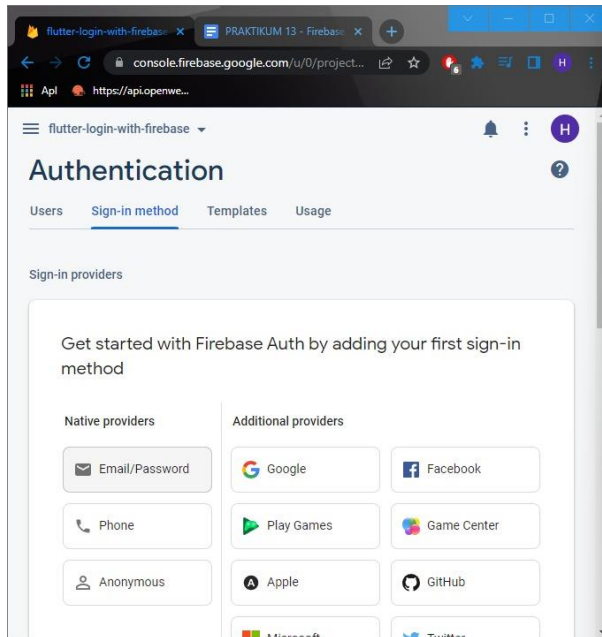
1. Untuk melakukan latihan authentication, kita perlu menyiapkan project firebase terlebih dahulu



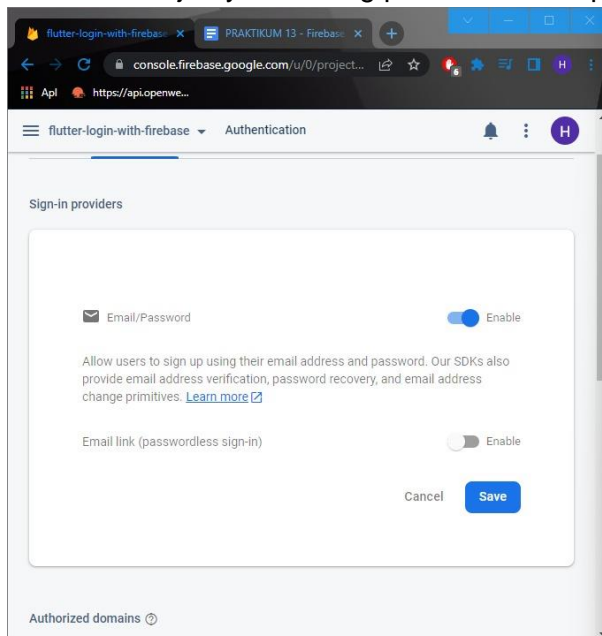
2. Setelah selesai dibuat project firebase-nya, selanjutnya kita pilih menu authentication



3. Setelah itu pilih tab menu sign-in method
4. Lalu pilih native providers Email/Password



5. Selanjutnya enabling providers email/password



6. Untuk melakukan latihan authentication ini, saya menggunakan project flutter yang baru dan sudah terintegrasi dengan library-library berikut.

```
firebase_core: ^1.17.0  
firebase_auth: ^3.3.18
```

7. Berikut kode program untuk instansiasi dari FirebaseAuth yang sudah terinstall

```
final FirebaseAuth _auth = FirebaseAuth.instance;
```

8. Selanjutnya merupakan kode program dari registrasi menggunakan FirebaseAuth dengan mengirimkan parameter email dan password



```
void _register() async {  
  final User? user = (await _auth.createUserWithEmailAndPassword(  
    email: _emailController.text, password: _passwordController.text))  
    .user;  
  
  if (user != null) {  
    setState(() {  
      _success = true;  
      _userEmail = user.email!;  
    });  
  } else {  
    setState(() {  
      _success = false;  
    });  
  }  
}
```

9. Selanjutnya merupakan kode program dari login yang akan dieksekusi oleh FirebaseAuth

```
void _signIn() async {  
  final User? user = (await _auth.signInWithEmailAndPassword(  
    email: _emailController.text, password: _passwordController.text))  
    .user;  
  
  if (user != null) {  
    setState(() {  
      _success = 2;  
      _userEmail = user.email!;  
    });  
  } else {  
    setState(() {  
      _success = 3;  
    });  
  }  
}
```

10. Berikut adalah hasil dari authentication menggunakan firebase. Dan dibawah ini merupakan hasil ketika login berhasil



Hello Kavit

Email
hakim@gmail.com

Password
.....

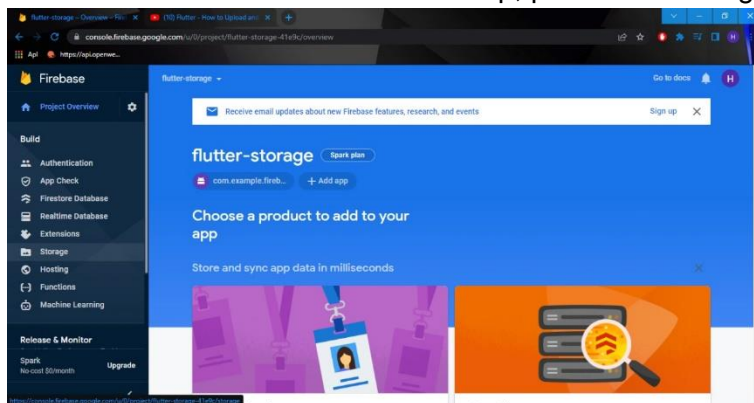
[Forgot Password](#)
Successfully signed in hakim@gmail.com

LOGIN

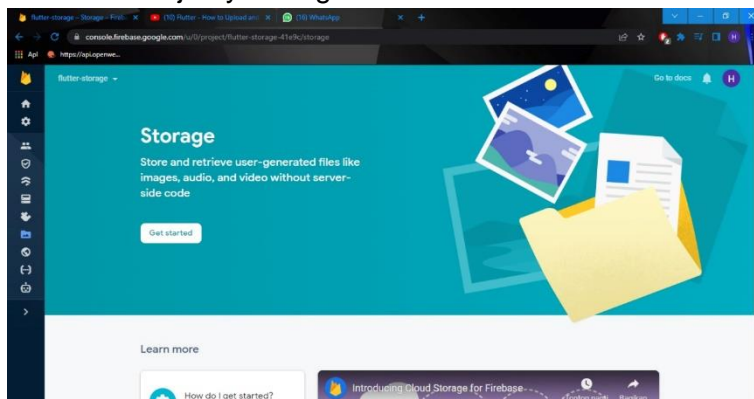
[Register](#)

Latihan Implementasi Firebase pada Studi kasus Upload & Retrieve Image

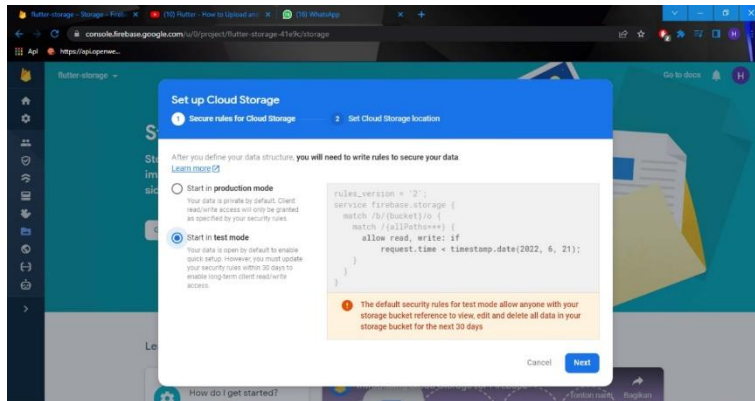
1. Untuk melakukan latihan firebase storage ini, sebelumnya kita siapkan project firebase terlebih dahulu. Jika sudah siap, pilih menu storage



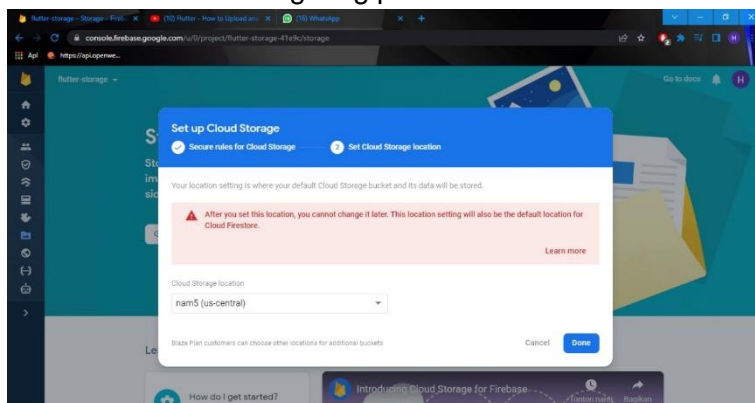
2. Selanjutnya klik get started



3. Pilih start in test mode



4. Setelah itu langsung pilih done.



5. Untuk melakukan latihan ini, saya menggunakan project flutter yang baru, dan sudah terintegrasi dengan library berikut

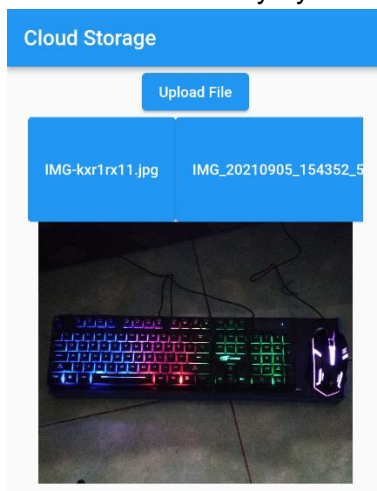
```
firebase_core: ^1.17.0
firebase_storage: ^10.2.16
file_picker: ^4.5.1
```

6. Berikut kode program class buatan yang berfungsi untuk menyimpan gambar dan menampilkan gambar dari firebase storage



```
class Storage {  
    final firebase_storage.FirebaseStorage storage =  
        firebase_storage.FirebaseStorage.instance;  
  
    Future<void> uploadFile(String filePath, String fileName) async {  
        File file = File(filePath);  
  
        try {  
            await storage.ref('test/$fileName').putFile(file);  
        } on firebase_core.FirebaseException catch (e) {  
            print(e);  
        }  
    }  
  
    Future<firebase_storage.ListResult?> listFiles() async {  
        firebase_storage.ListResult results = await storage.ref('test').listAll();  
  
        results.items.forEach((firebase_storage.Reference ref) {  
            print("Found file: $ref");  
        });  
  
        return results;  
    }  
  
    Future<String> downloadURL(String imgName) async {  
        String downloadURL = await storage.ref('test/$imgName').getDownloadURL();  
  
        return downloadURL;  
    }  
}
```

7. Hasil akhirnya yaitu sebagai berikut





Program Studi D3 Teknik Informatika
Jurusan Teknik Informatika
Politeknik Negeri Indramayu

NIM : 2003071

Nama : Hakim Asrori

E. Hasil dan Penjelasan Tugas



F. Kesimpulan

Firebase merupakan penyimpanan data yang di sediakan oleh google untuk memudahkan para programmer. Firebase sendiri memiliki arsitektur microservice.

G. Referensi

- Alberto Miola. "Flutter Complete Reference Create Beautiful, Fast and Native Apps for Any Device". Independently Published. 2020.
- Simone Alessandria, Brian Kayfirz. "Flutter Cookbook: Over 100 proven techniques and solutions for app development with Flutter 2.2 and Dart". Packt Publishing. Birmingham - Mumbai. 2021.
- Dieter Meiller. "Modern App Development with Dart and Flutter 2: A Comprehensive Introduction to Flutter". Walter de Gruyter GmbH. Berlin - Boston. 2021.
- Priyanka Tyagi. "Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop". CRC Press Taylor & Francis Group, LLC. London - New York. 2022.
- Resa Risyan. <https://www.monitorteknologi.com/perbedaan-database-relasional-dan-non-relasional/#penci-Apa-Itu-Database-Relasional>. 2020. Diakses tahun 2022.
- Hasil Copa. <https://hasilcopa.com/perbandingan-antara-basis-data-relasional-dan-basis-data-non-relasional>. 2022. Diakses tahun 2022
- Dicoding Intern. <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>. 2020. Diakses tahun 2022