**LAPORAN PRAKTIKUM**



|  |  |  |
| --- | --- | --- |
| **NIM** | **:** | 2003073 |
| **Nama** | **:** | Ica Natasya |
| **Kelas** | **:** | D3TI.2C |
| **Mata Kuliah** | **:** | **Pemrograman Perangkat Bergerak (TIU3403)** |
| **Praktikum ke / Judul** | **:** | 10 / Firebase |
| **Tanggal Praktikum** | **:** | 16 Mei 2022 |
| **Dosen Pengampu** | **:** | Fachrul Pralienka Bani Muhamad, S.ST., M.Kom. |

**PROGRAM STUDI D3 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI INDRAMAYU**

**2022**

**PRAKTIKUM 13**

FIREBASE

1. **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 dapat

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

**Perbandingan Database Relational dan Non Relational**

* 1. Database relasional adalah database digital yang didasarkan pada model data relasional. Data disimpan dalam tabel yang berisi baris (yang mewakili entri) dan kolom (yang menyimpan dan mengurutkan jenis informasi tertentu). Singkatnya, database relasional digunakan untuk menyimpan data yang terstruktur ke dalam tabel.

**Kelebihan:**

* Dapat menangani banyak kueri kompleks, transaksi database, dan analisis data rutin.
* ACID (Atomity, Consistency, Isolation, Durability): Kumpulan properti yang memastikan transaksi database yang andal.

**Kekurangan:**

* Tidak dapat menyimpan gambar, angka, desain, dan produk multimedia yang kompleks atau sangat besar
* Bisa menjadi sangat mahal dengan pemeliharaan dan server baru

Contoh database relasional yaitu MySQL, PostgreSQL, SQLite3, Server MSSQL, dan mariaDB.

* 1. Database non-relasional ada pada akhir 1960-an, tetapi istilah itu tidak digunakan sampai 1998 oleh Carlo Strozzi yang memimpin pengembangan NoSQL. Database non-relasional lebih seperti dokumen Word. Daripada menggunakan tata letak terstruktur berbasis tabel untuk datanya, database non-relasional menyimpan data dalam koleksi tergantung pada jenis datanya. Mereka juga dikenal sebagai database NoSQL, atau ‘tidak hanya SQL’, karena mereka dapat menggunakan bahasa kueri lain bergantung pada tata letaknya.

**Kelebihan:**

* Data terstruktur, semi-terstruktur, dan tidak terstruktur dalam jumlah besar
* Pemrograman berorientasi objek yang mudah digunakan dan fleksibel
* Arsitektur berskala besar yang efisien, bukan arsitektur monolitik yang mahal

**Kekurangan:**

* Lebih sedikit dukungan karena database NoSQL biasanya open-source
* Dalam segi administrasi database NoSQL membutuhkan keterampilan teknis untuk menginstal dan memelihara.
* Database NoSQL masih berkembang dan banyak fitur masih diterapkan.

Contoh database non relasional yaitu mongoDB, Cassandra, dan Redis.

**Firebase**

(buatlah penjelasan singkat)

opsi referensi: <https://www.youtube.com/watch?v=EXp0gq9kGxI>

Firebase adalah salah satu database dengan basis NoSQL (Bukan SQL). Di Indonesia, Ia adalah salah satu tools yang mulai banyak digunakan oleh developer aplikasi dan web beberapa tahun belakangan ini. Firebase adalah tools yang terkenal sederhana dan powerful, sehingga banyak yang mengklaim dapat mempercepat pekerjaan. Firebase merupakan tools yang pertama kali rilis pada 2011 oleh Andrew Lee dan James Tamplin. Produk pertama Ia adalah Realtime Database yang berfungsi untuk menyimpan data dan menyinkronkannya ke beberapa user. Aplikasi ini selanjutnya berkembang menjadi berbagai layanan untuk developer aplikasi. Perusahaan ini Google ambil alih pada Oktober 2014.

**Service Pada Firebase**

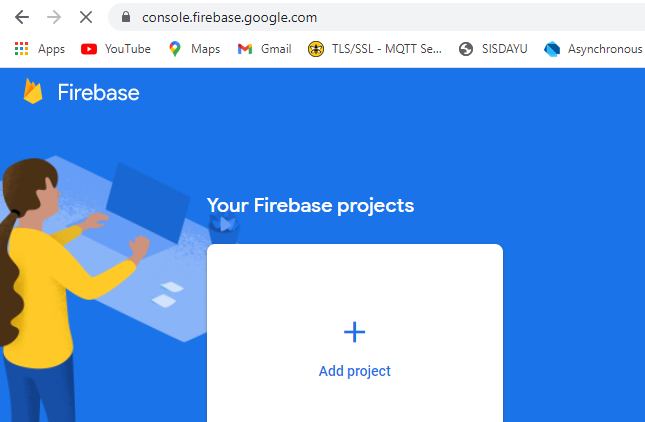
|  |  |  |
| --- | --- | --- |
| Nama Akun | Pengguunaan Akun | Kapan ditambahkan ke project |
| firebase-service-account@firebase-sa-management.iam.gserviceaccount.com | Untuk mengelola dan menautkan layanan Firebase ke project Google Cloud - diperlukan untuk semua project Firebase | Saat membuat project Firebase/menambahkan layanan Firebase ke project Google Cloud yang sudah ada  **Catatan:** Beberapa project Firebase yang dibuat sebelum September 2018 tidak berisi akun layanan ini. Namun, Firebase otomatis menambahkan akun layanan ini ke salah satu project tersebut saat tindakan yang akan dilakukan memerlukan akun layanan ini. |
| service-project-number@gcp-sa-firebaseappcheck.iam.gserviceaccount.com | Untuk memverifikasi token respons yang dikeluarkan oleh penyedia App Check reCAPTCHA Enterprise | Saat mendaftarkan aplikasi untuk menggunakan App Check dengan penyedia reCAPTCHA Enterprise |
| project-id@appspot.gserviceaccount.com | Untuk mengelola produk yang terkait dengan App Engine: Cloud Firestore, Cloud Storage for Firebase, dan Cloud Functions for Firebase | Saat membuat project Firebase/mengimpor project Google Cloud ke Firebase Mungkin sudah ada untuk project Google Cloud sebelum diimpor |
| service-project-number@firebase-rules.iam.gserviceaccount.com | Untuk mengelola Aturan Keamanan Firebase untuk Cloud Firestore, Cloud Storage for Firebase, dan Realtime Database | Saat membuat project Firebase/mengimpor project Google Cloud ke Firebase |
| service-project-number@gcf-admin-robot.iam.gserviceaccount.com | Untuk mengoperasikan Cloud Functions for Firebase | Saat membuat project Firebase/mengimpor project Google Cloud ke Firebase Mungkin sudah ada untuk project Google Cloud sebelum diimpor |

1. **PELAKSANAAN PRAKTIKUM**

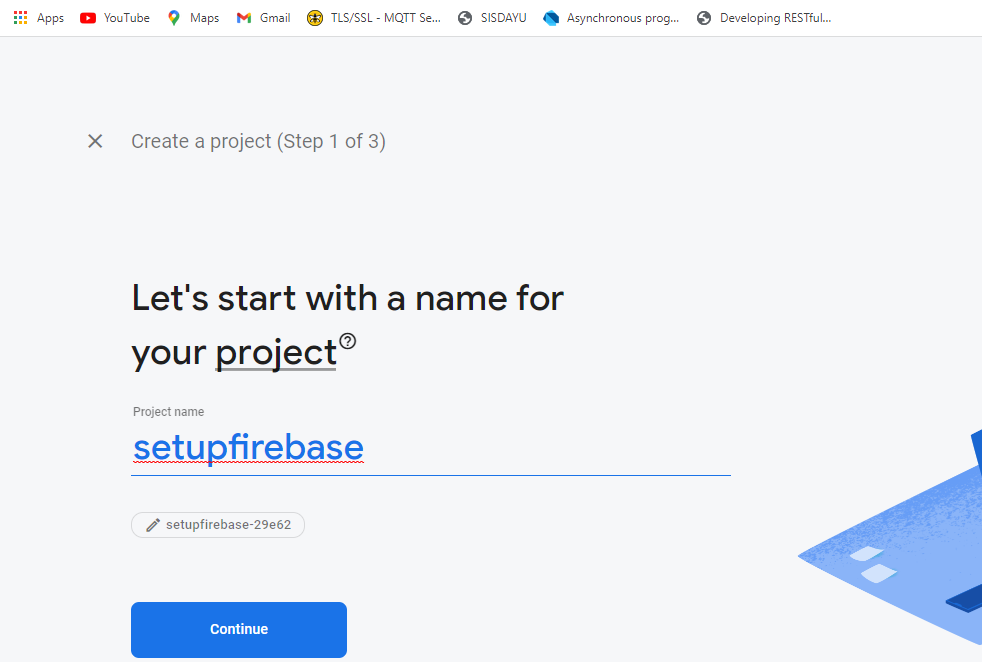
**Langkah-langkah praktikum Setup Firebase**

<https://www.youtube.com/watch?v=sz4slPFwEvs>

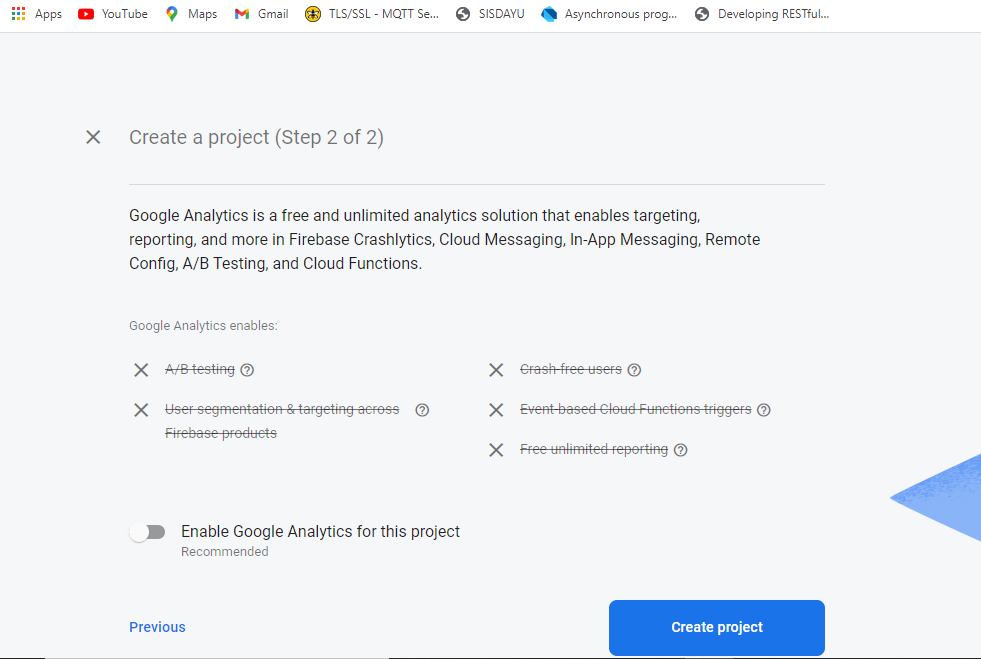
1. Buka console.firebase.google.com, kemudian login dengan menggunakan email, selanjutnya klik Add project.



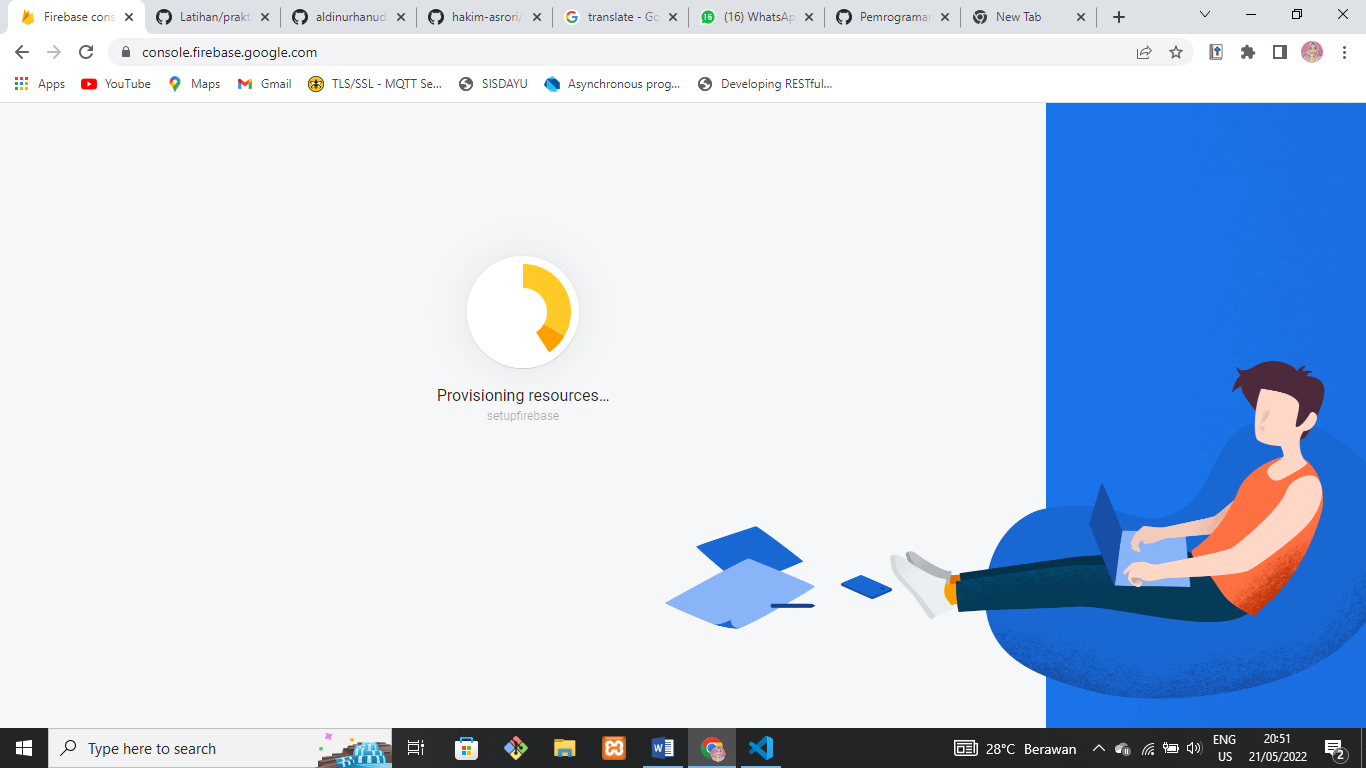
1. Beri nama project yang akan kita buat.



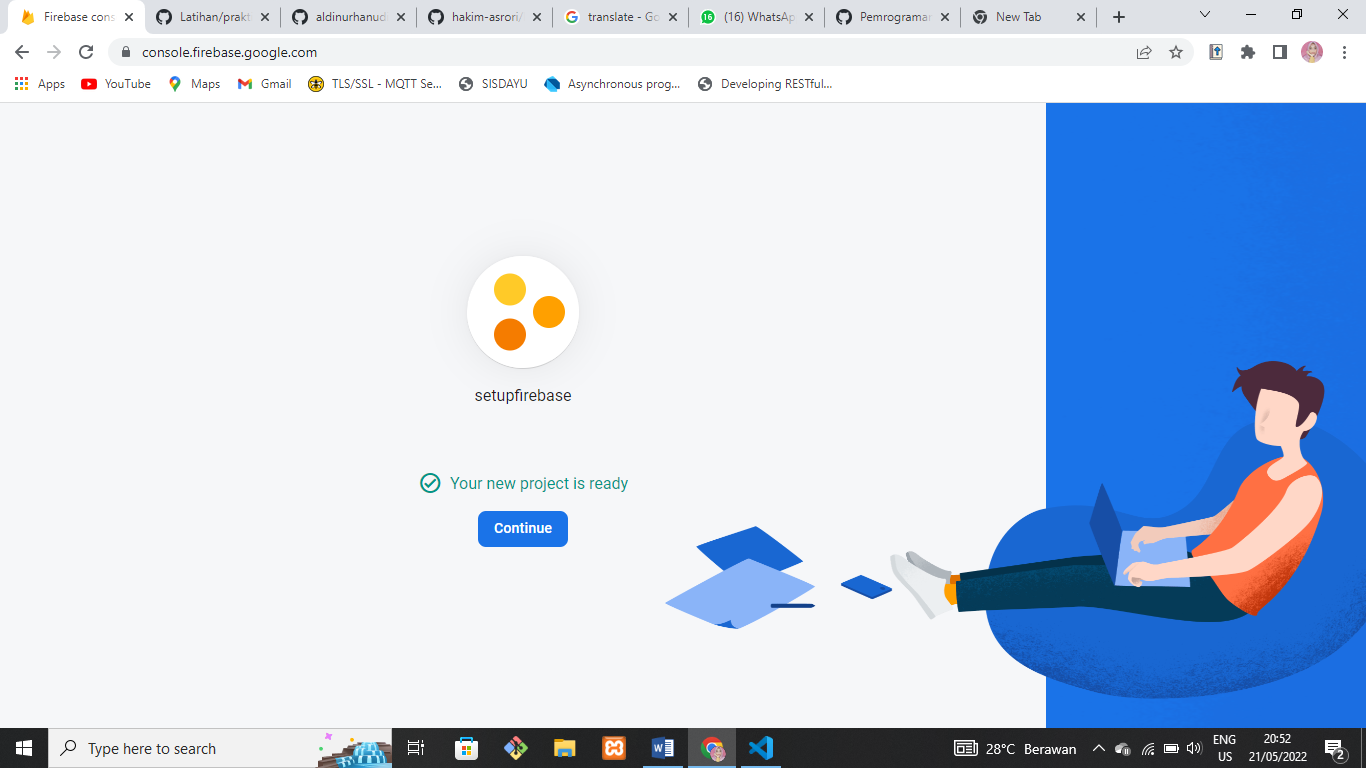
1. Matikan Enable Google Analytics for this projecet, selanjutnya klik create projeect.



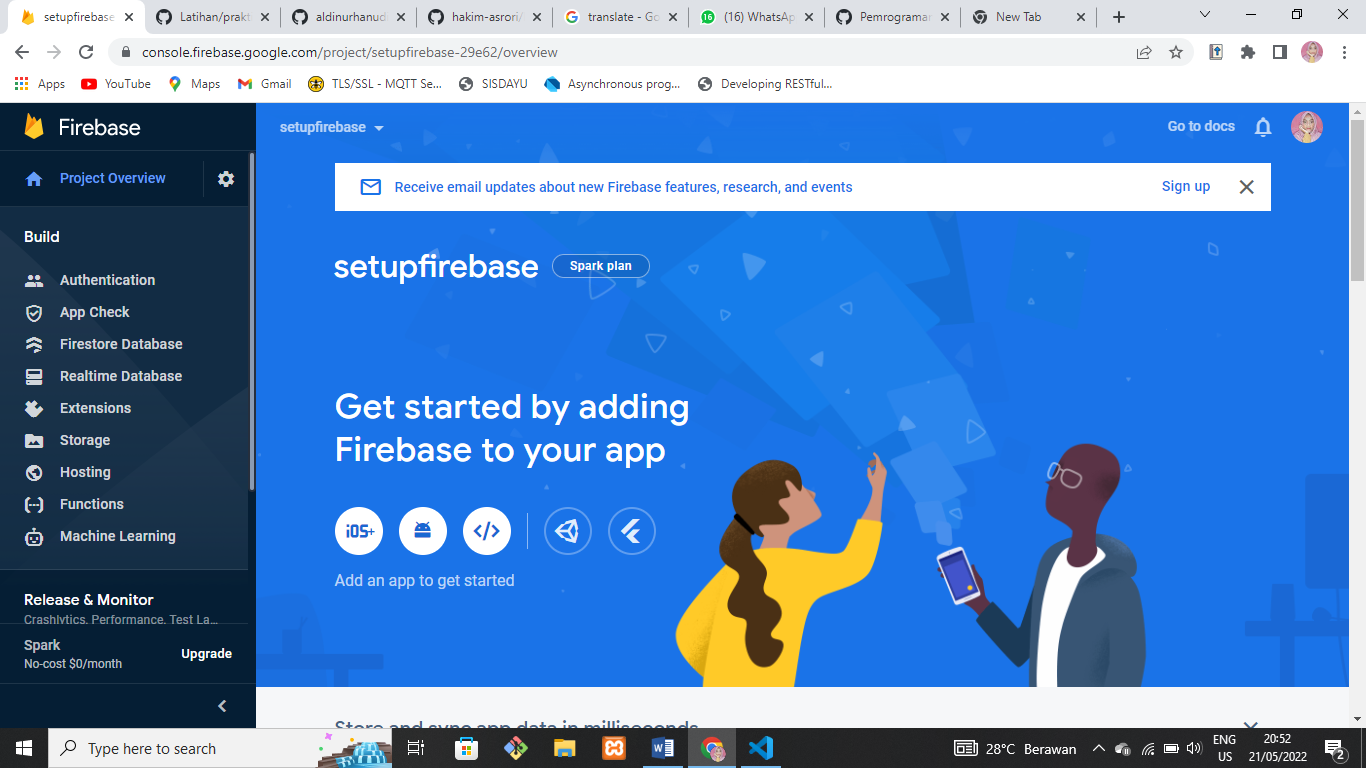
1. Tunggu proses pembuatan project sampai selesai.



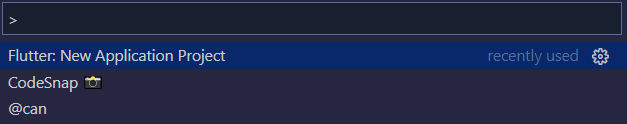
1. Setelah berhasil kemudian klik continue.



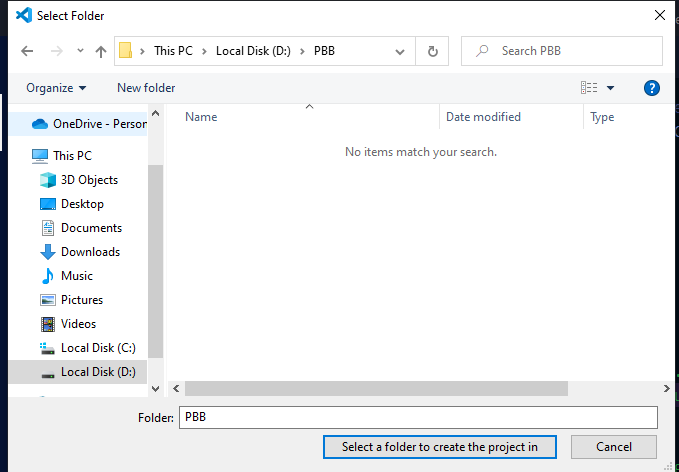
1. Kemudian klik app to get.



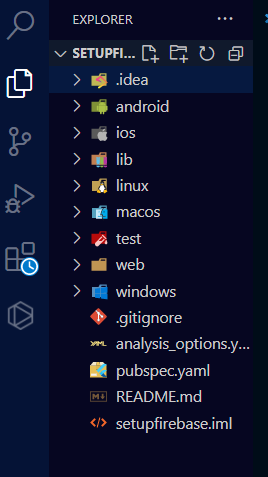
1. Buat projec baru pada flutter.



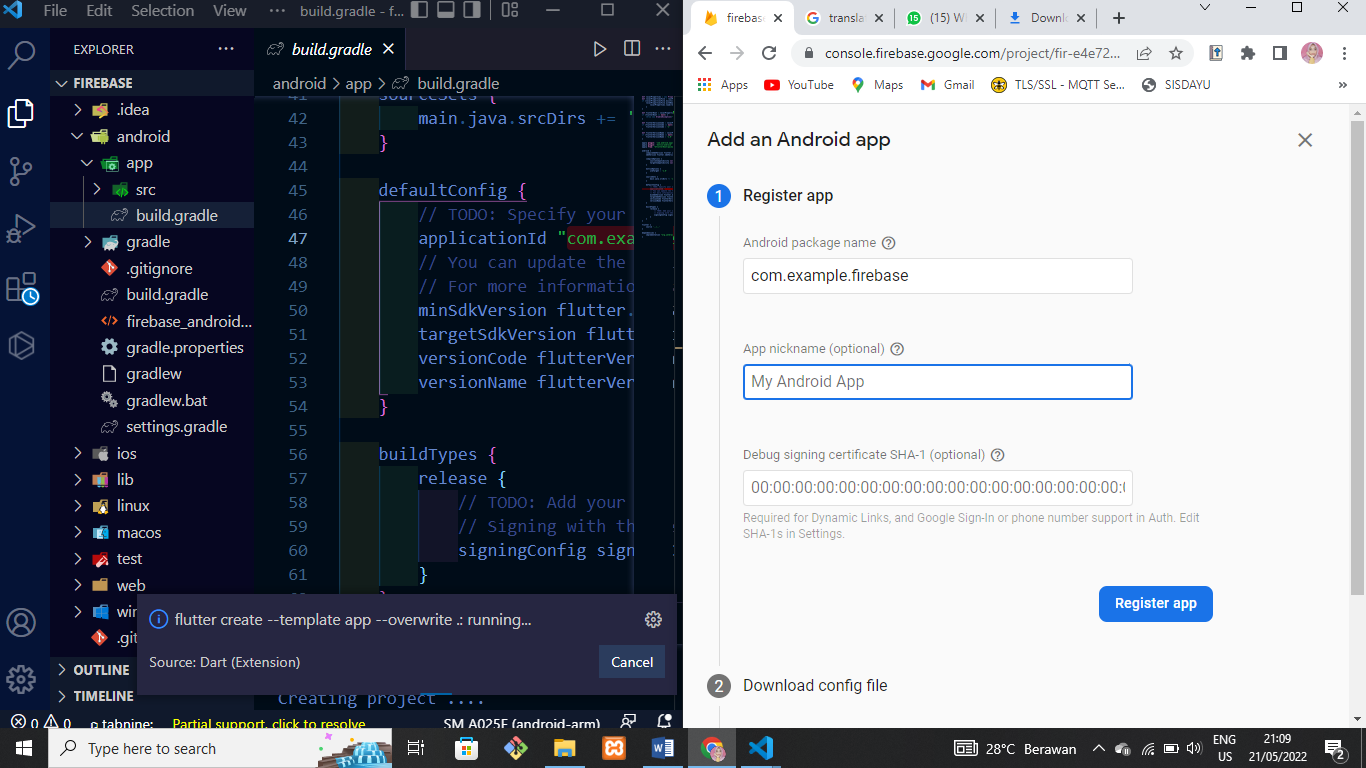
1. Pilih penyimpanan project.



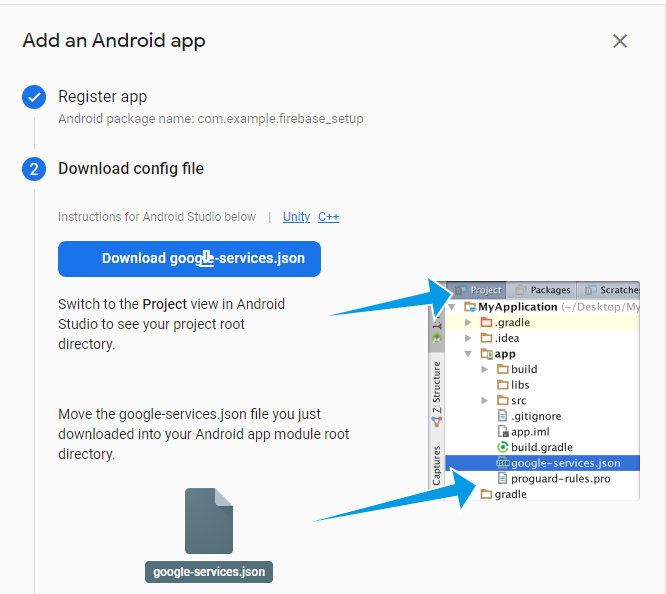
1. Project selesai dibuat.



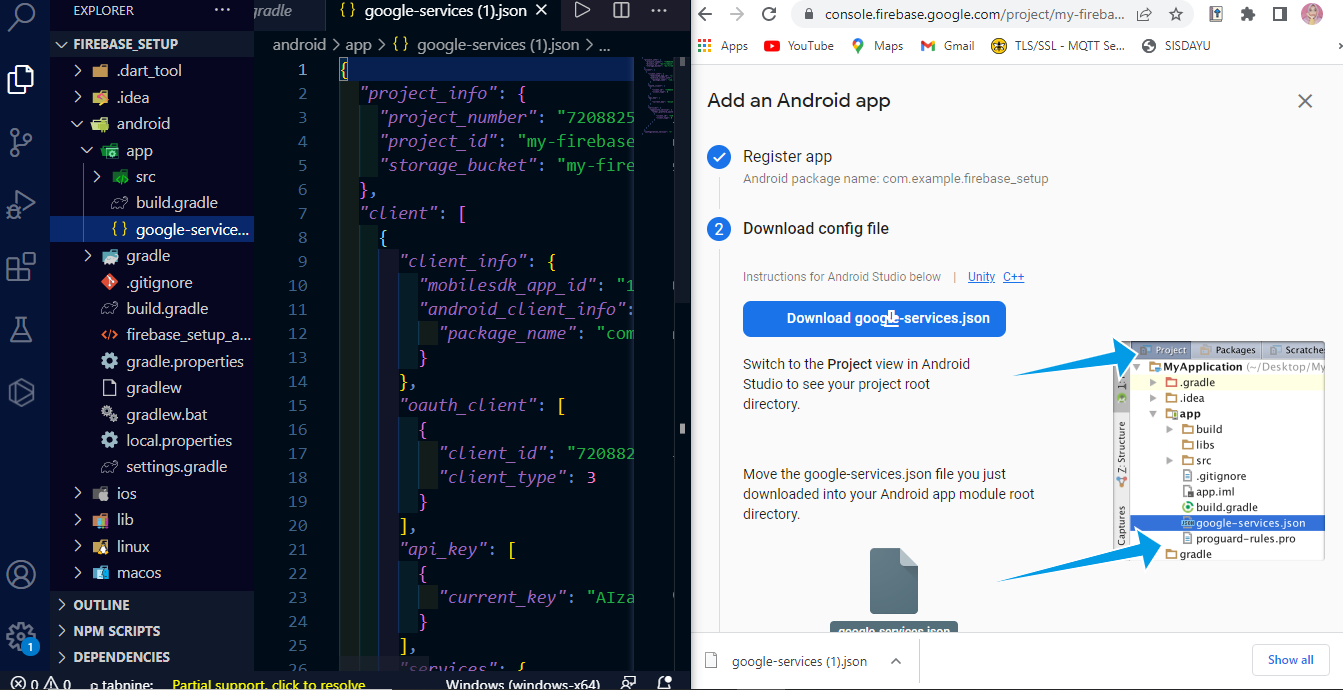
1. Copy applicationId pada flutter ke project firebase yang telah kita buat, kemudain klik Register app.



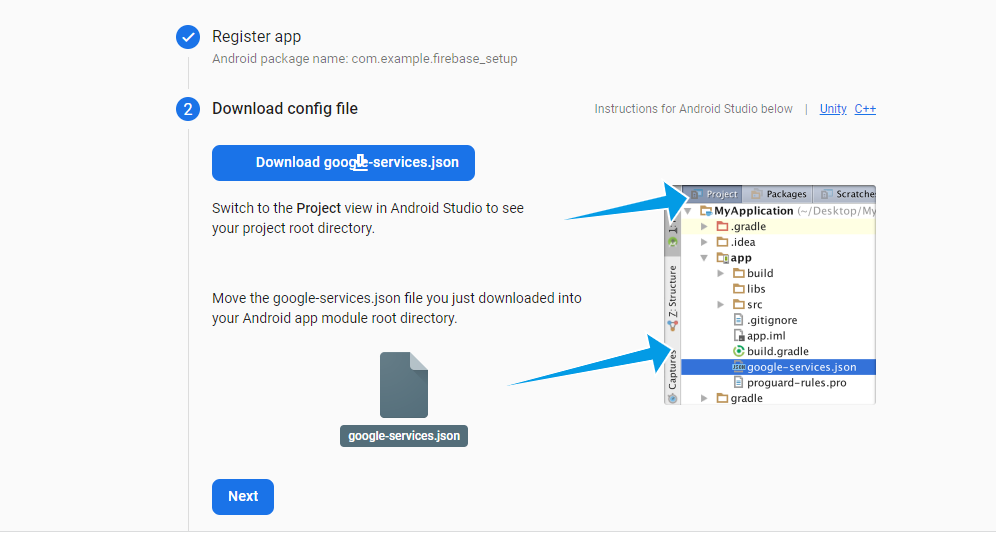
1. Download google-services.json.



1. Paste kan kedalam foldere app.

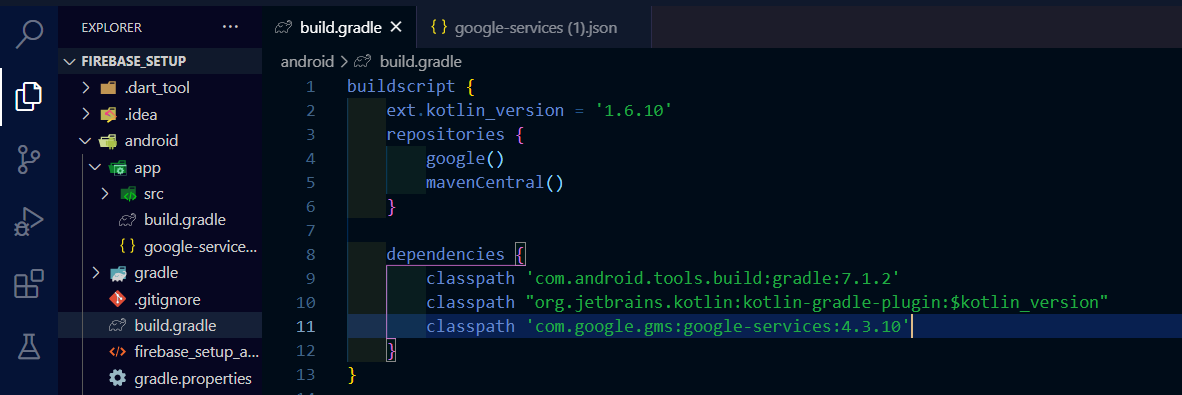


1. Kemudian klik next.

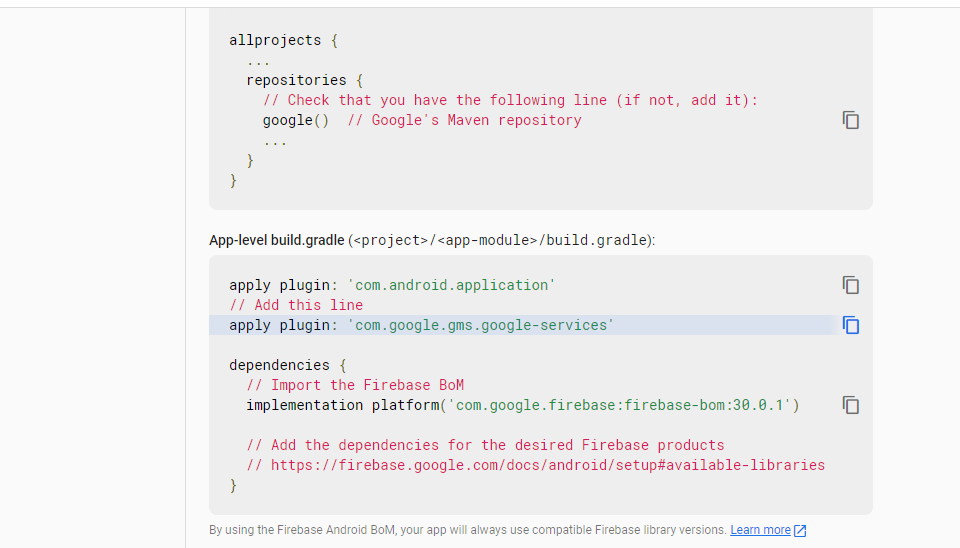


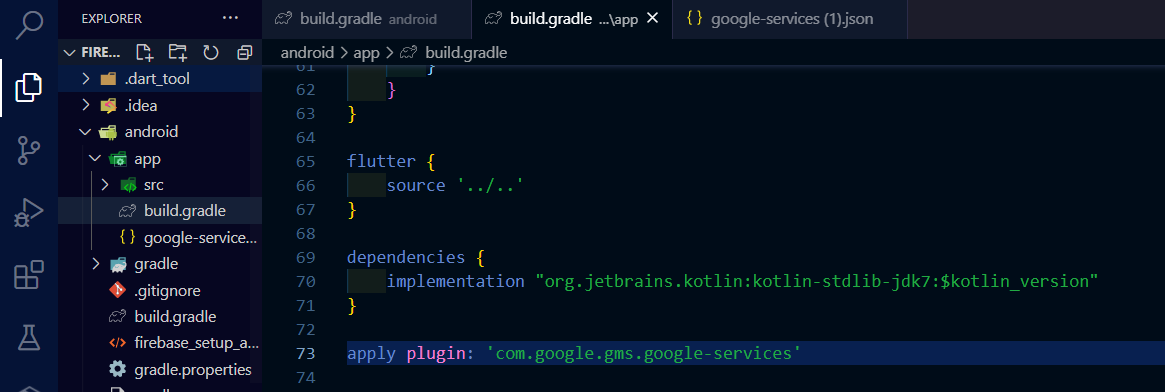
1. Paste kan alamat dependencies ke project flutter



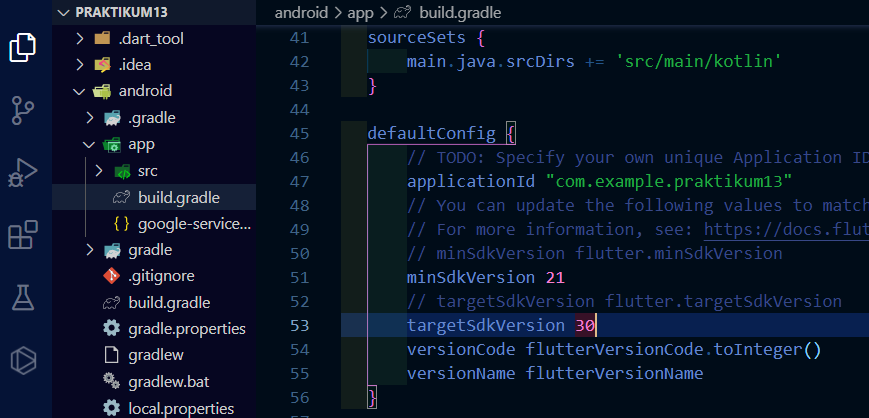


1. Paste kan appy pluigin ke dalam projeect flutter di package app-module>/build.gradle.

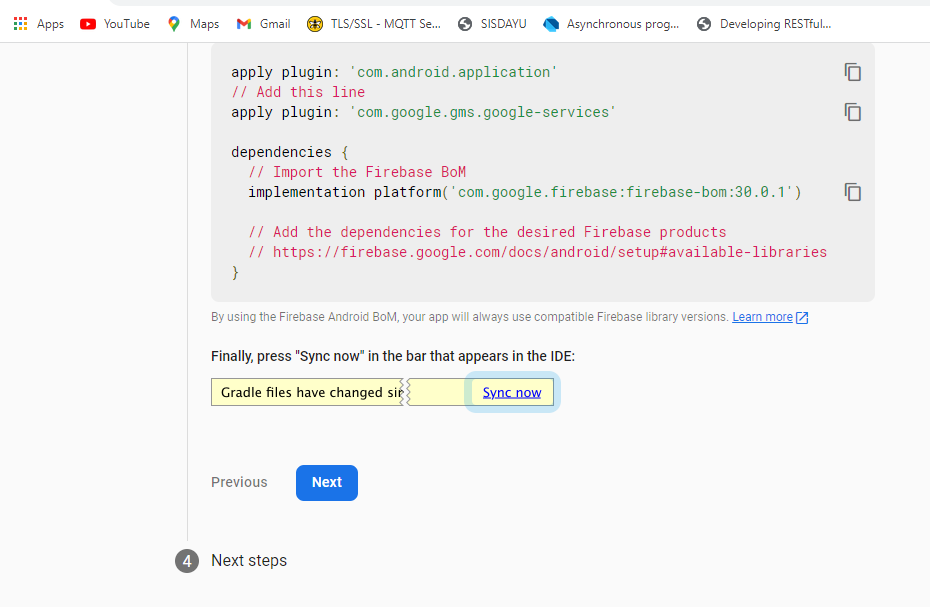




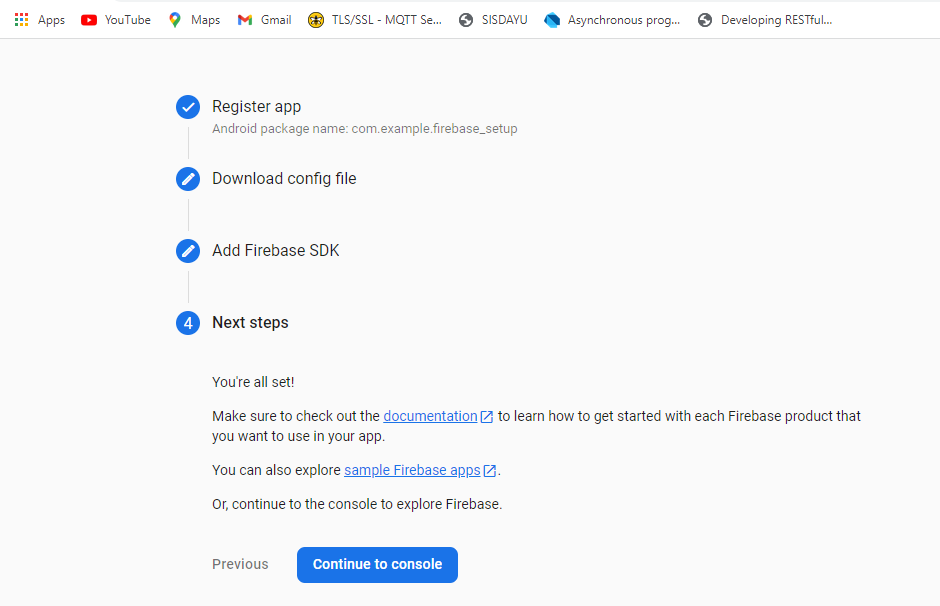
1. Tambahkan versi pada minSdkVersion dan targetSdkversion



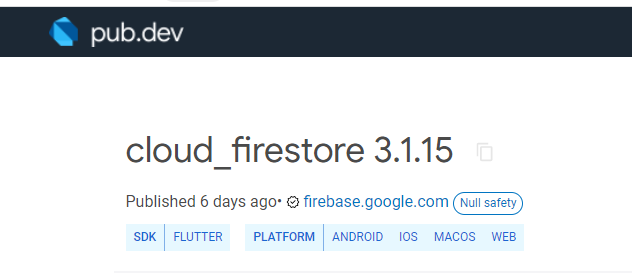
1. Kemudian klik next.

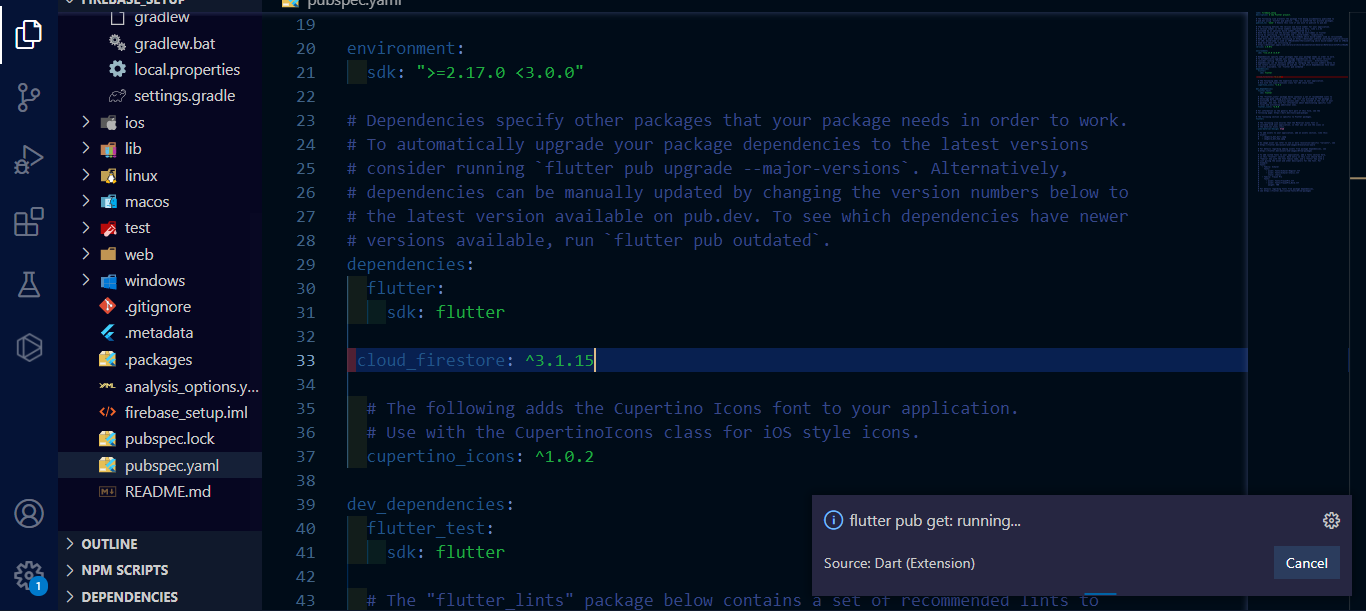


1. Klik continue to console.



1. Copy kan cloud\_firestore terbaru kedalam project flutter yang telah kita buat.



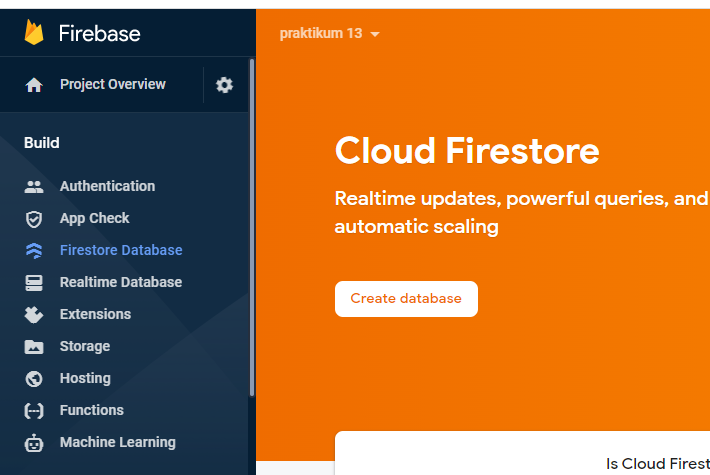


1. Selesai.

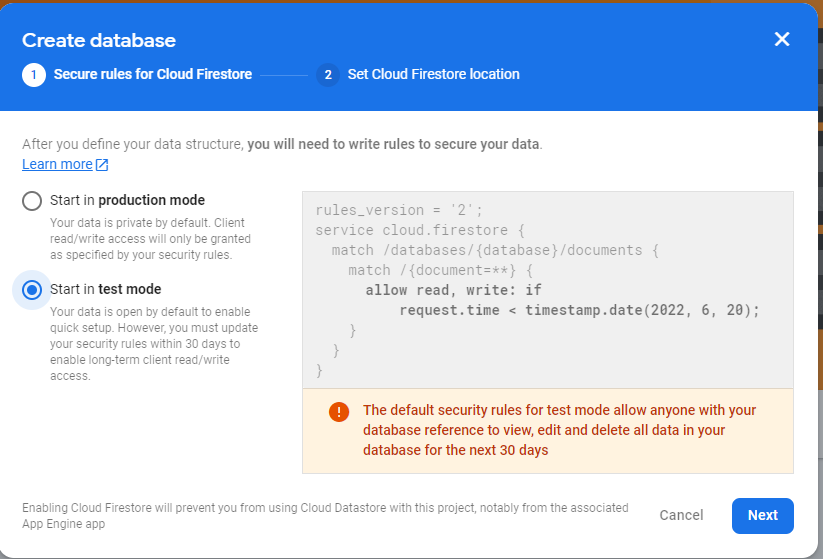
**Langkah-langkah praktikum CRUD Firebase**

<https://www.youtube.com/watch?v=ErP_xomHKTw>

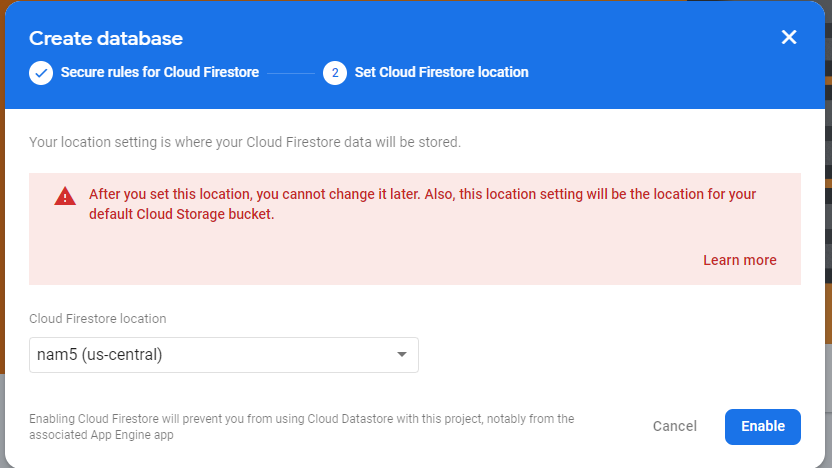
1. Lakukan praktikum sebelumnya menyesuaikan dengan project yang akan kita buat.
2. Create database pada firebase yang telah kita buat.



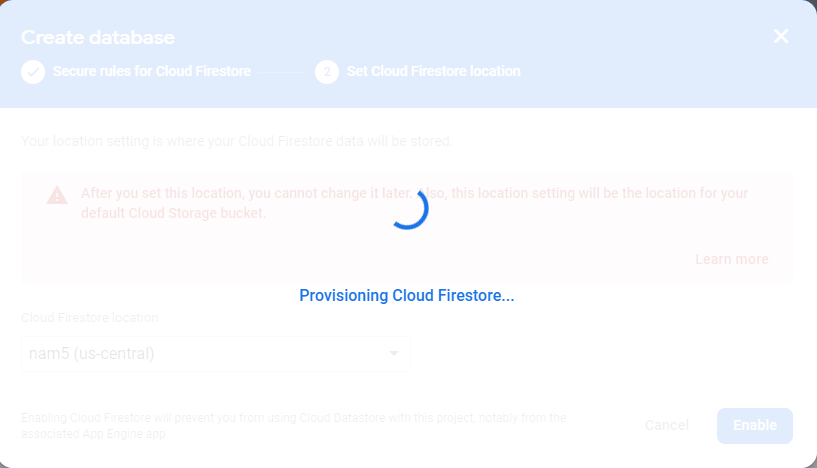
1. Pilih start in test mode kemudain klik next.



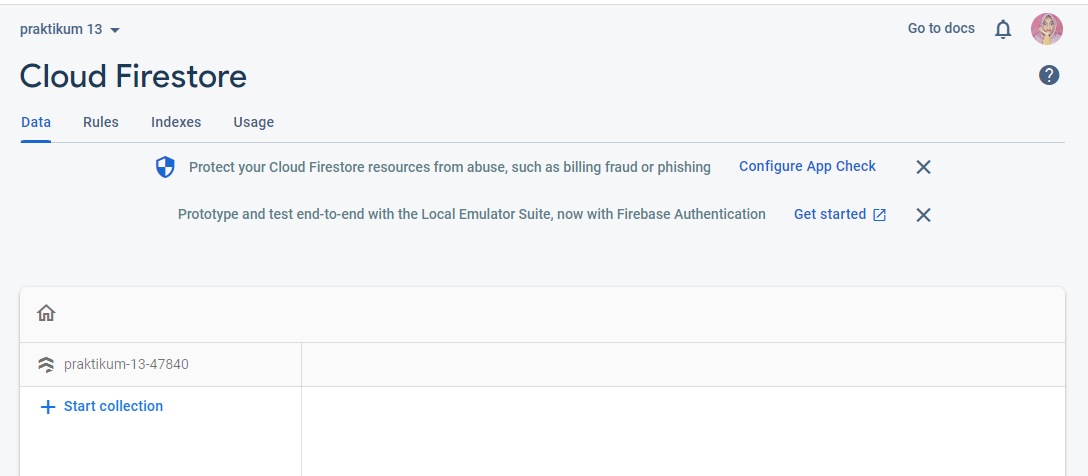
1. Pilih cloud firebase location kemudian klik enable.



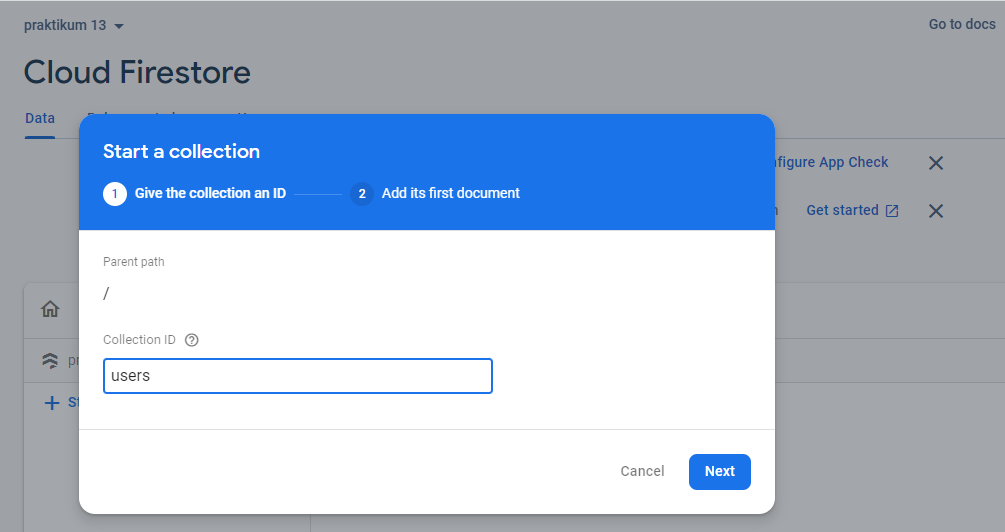
1. Tunggu proses provisioning cloud firestore.



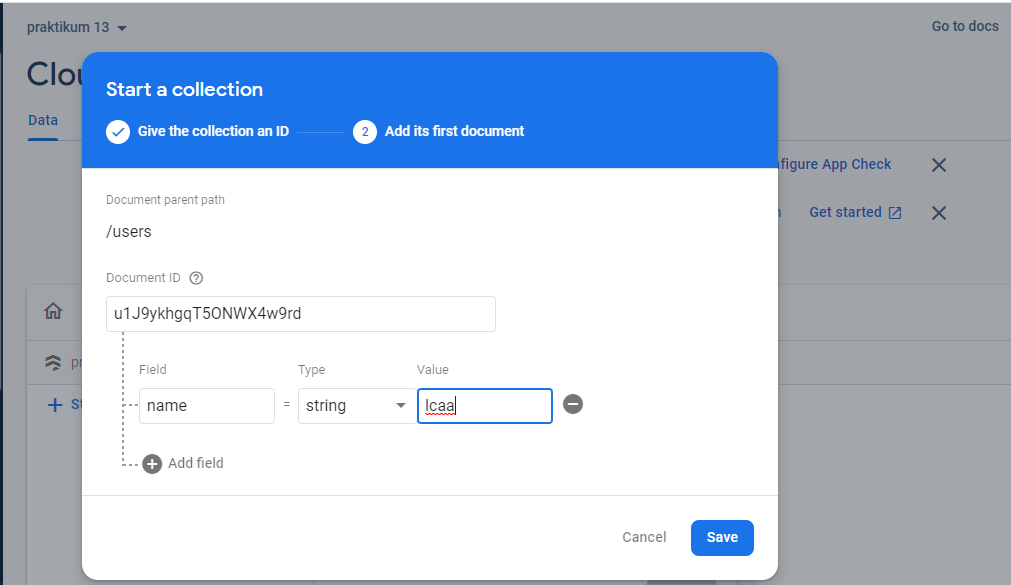
1. Database berhasil dibuat.



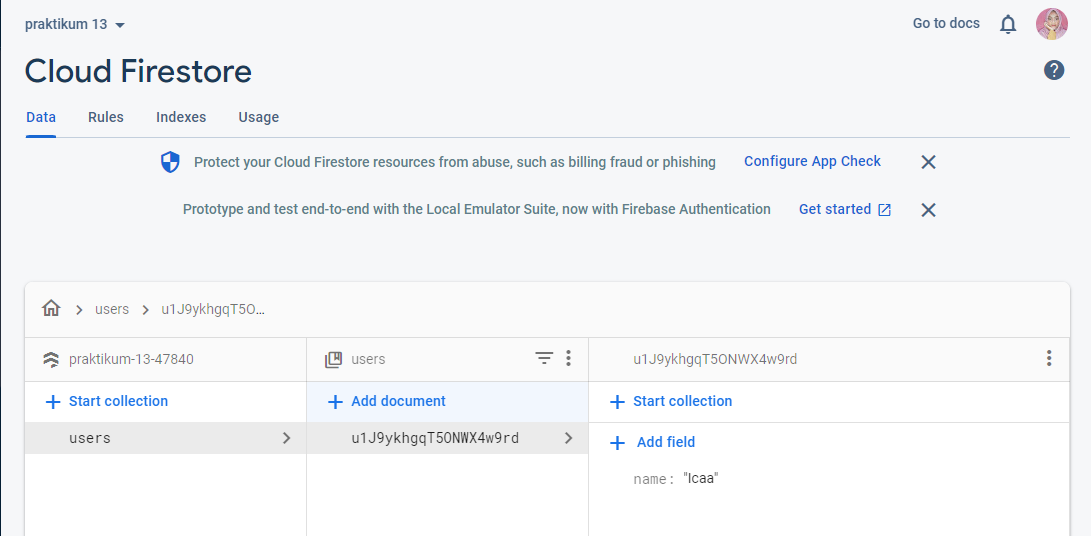
1. Buat collection baru dengan nama collection users.



1. Buat field pada collection users dengan field name type string kemudian value icaa, kemudian klik save.



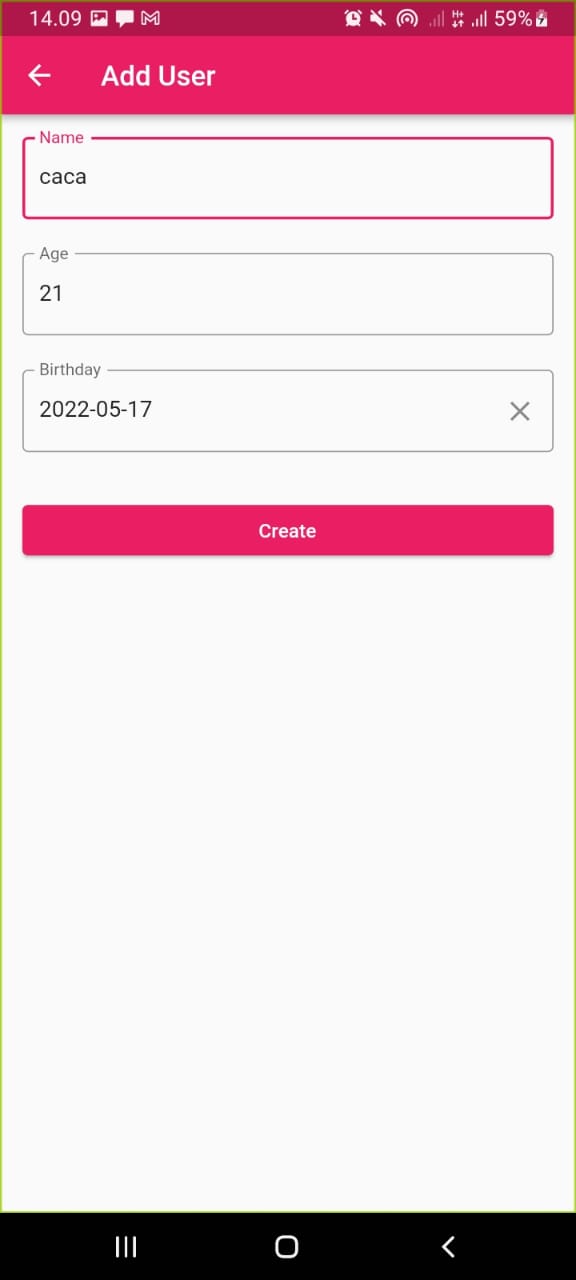
1. Data berhasil dibuat.



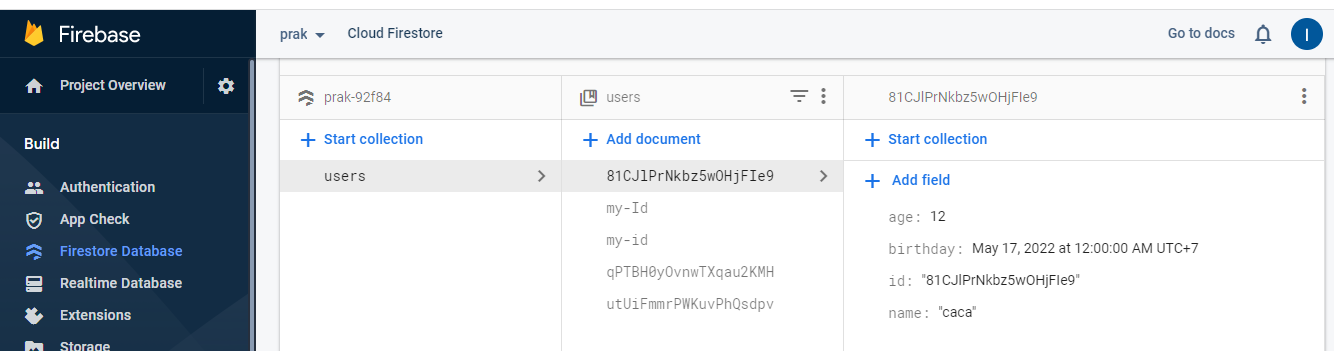
1. Create data pada project flutter.



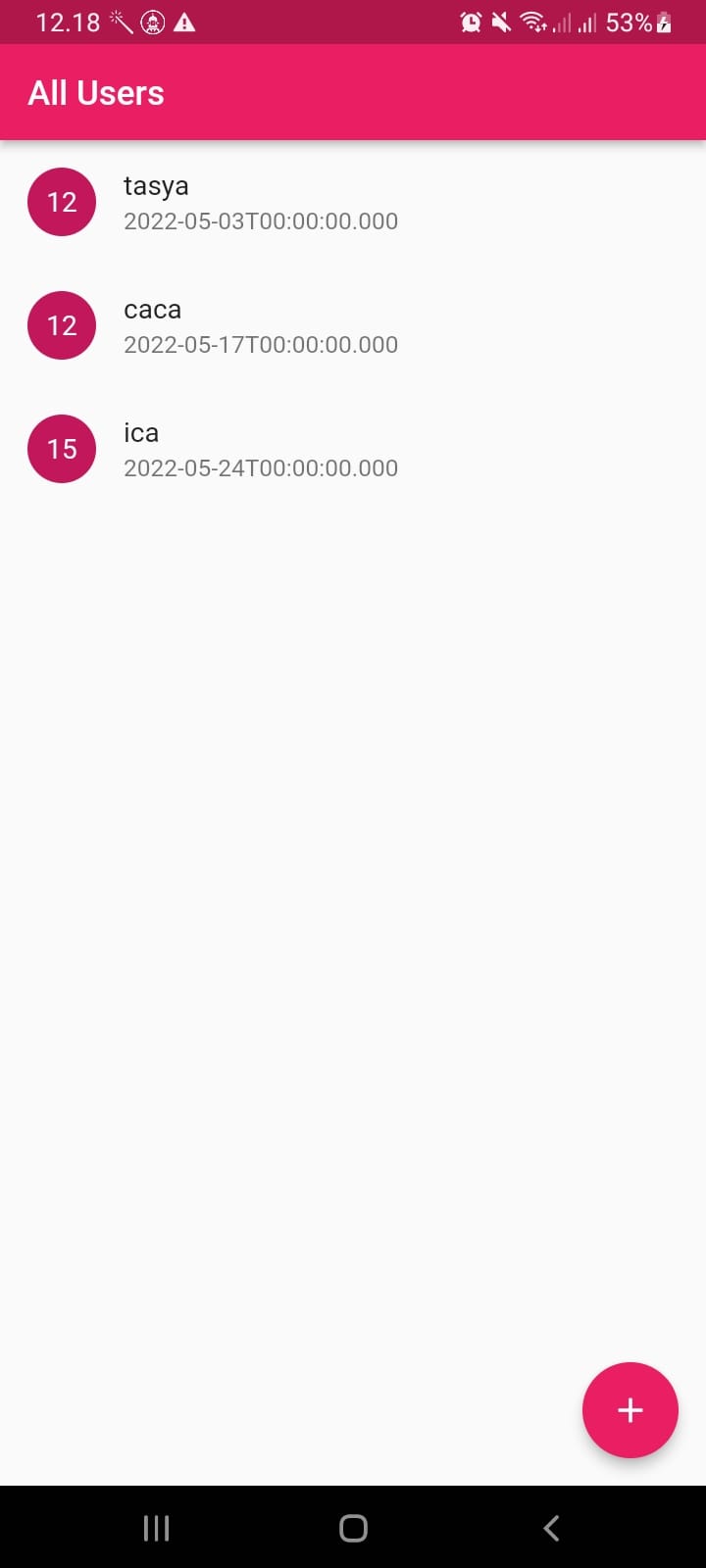
1. Isi data dengan name caca, age 21, dan birthday 2022-05-17 kemudian klik create.



1. Data berhasil dibuat dan masuk kedalam firebase.



1. Read data



1. Delete data.



1. Update data.



1. **LATIHAN**

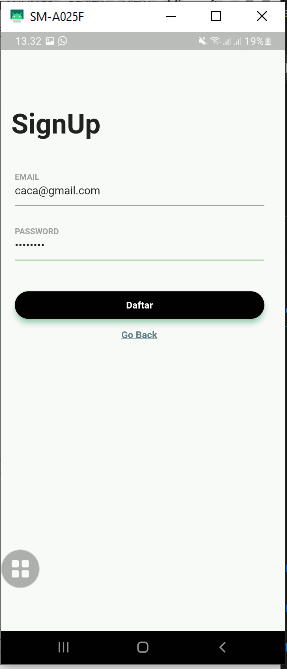
**Latihan Implementasi Firebase pada Studi kasus Authentication**

Buatlah implementasi penggunaan Firebase melalui tutorial berikut:

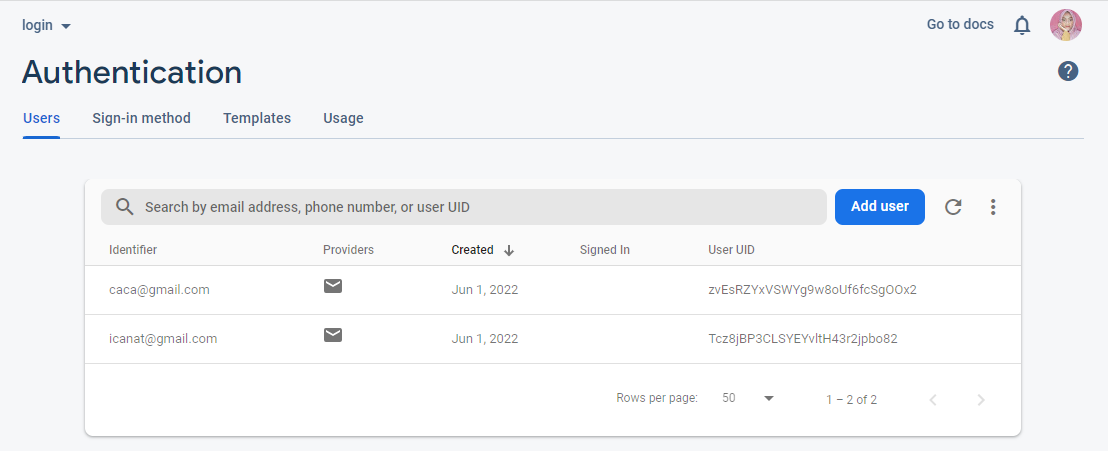
<https://www.youtube.com/watch?v=IPMIcGTzxGc>

Hasil kode program

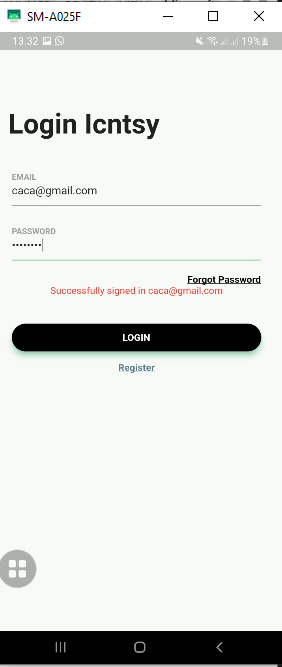
* + - 1. Buat projek flutter terbaru dengan nama project Auth, kemudian buat kode program.
      2. Buat firebase dengan nama project Auth kemudian buat Authentication.
      3. Running kode program.
      4. Halaman SignUp, masukkan email dan password.



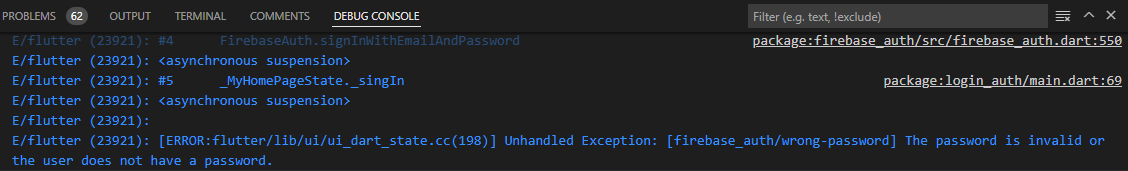
* + - 1. Cek pada firebase yang telah dibuat.



* + - 1. Halaman Login, masukkan email dan password kemudian login, jika berhasil maka akan muncul notifikasi Successfully signed in caca@gmail.com.



* + - 1. Jika gagal memasukkan email atau password maka akan munculerror the password is invalid or the user does not have a password.



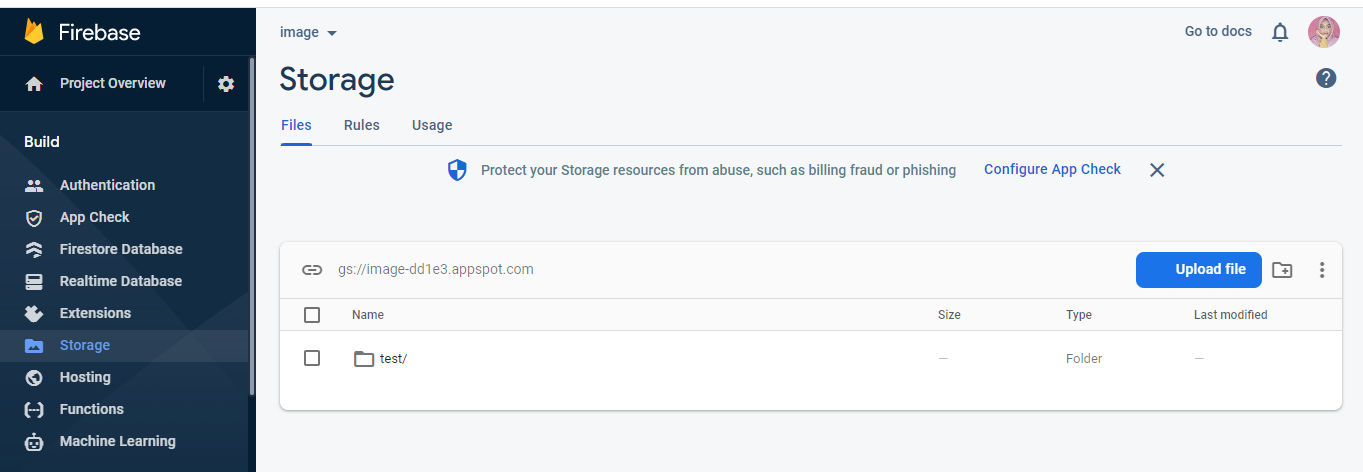
**Latihan Implementasi Firebase pada Studi kasus Upload & Retrieve Image**

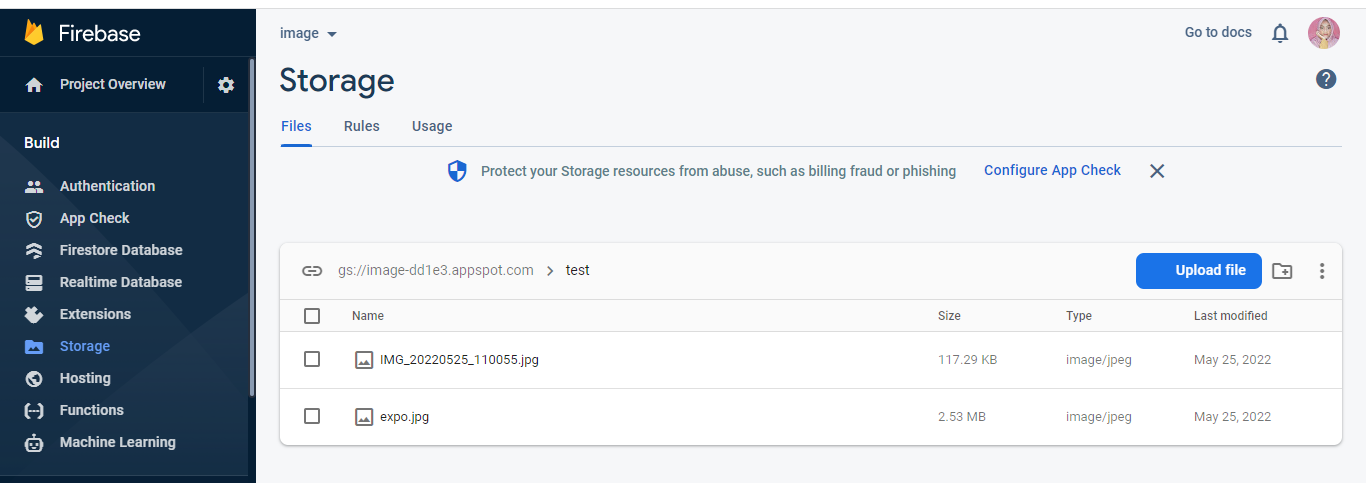
Buatlah implementasi penggunaan Firebase melalui tutorial berikut:

<https://www.youtube.com/watch?v=sM-WMcX66FI>

Hasil kode program

Buat database firebase dengan nama project image, kemudian buat storage pada firebase dengan nama folder test dan masukkan file foto.

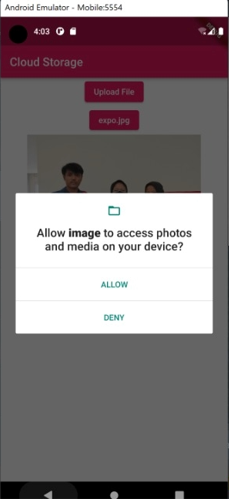




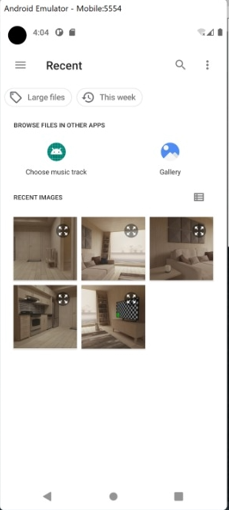
Tampilan mobile, data foto yang berada difirebase ada pada mobile aplication.



Jika kita ingin mengubah foto yang ditampilan, maka kita klik uoload foto, kemudian pilih allow.

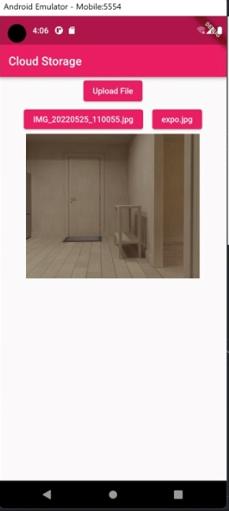


Pilih foto yang ingin kita upload.



Kemudian edit pada bagian FutureBuilder ganti dengan format foto yang terbaru, setelah itu foto akan berganti.





1. **TUGAS INDIVIDU**
2. Integrasi ke mata kuliah proyek
   1. Perubahan kode program harus dapat terlacak pada suatu repository terpusat (github, gitlab, atau bitbucket)
   2. Lampirkan tautan ke repository tersebut
3. **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.