

Modul 6

Integrating Firebase into a Flutter App (1)

Firestore Integration with Analytics

Module Overview

Membuat proyek yang terhubung dengan Firebase Analytics

Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

- Mengetahui mengenai firebase
- Dapat bekerja dengan proyek firebase (*analytics*) di dalam Flutter

Firestore adalah seperangkat alat yang digunakan untuk membangun aplikasi yang skalabel di cloud seperti otentikasi, database, pemberitahuan, dan hosting. Karena ini adalah layanan cloud, Anda tidak perlu menginstal perangkat lunak apa pun di perangkat Anda. Anda hanya perlu membuat akun Google untuk dapat menggunakan proyek Firestore. Ini dikarenakan Firestore dioperasikan oleh Google. Kelebihan menggunakan Firestore adalah alat yang disediakan Firestore mencakup sebagian besar layanan yang Anda butuh kan untuk membangun sebuah aplikasi, termasuk otentikasi, database, dan penyimpanan file. Anda cukup bertindak sebagai klien yang terhubung ke Firestore (dalam kasus ini, klien berupa aplikasi Flutter yang

akan anda bangun) dan berinteraksi dengan layanan back-end secara langsung, tanpa layanan sisi server middleware. Artinya ketika anda menggunakan database Firestore, Anda hanya cukup menulis kueri langsung di Aplikasi Anda. Ini adalah salah satu keuntungan utama menggunakan Firebase saat mengembangkan aplikasi yang membutuhkan layanan back-end. Anda tidak perlu untuk menulis, menginstal, atau memelihara layanan web menggunakan PHP, atau Java, atau C#. Anda akan berurusan dengan Firebase langsung dari aplikasi Flutter Anda.

Setiap proyek yang melibatkan penggunaan Firebase dimulai dengan konsol Firebase. Anda dapat melihat konsol tersebut pada:

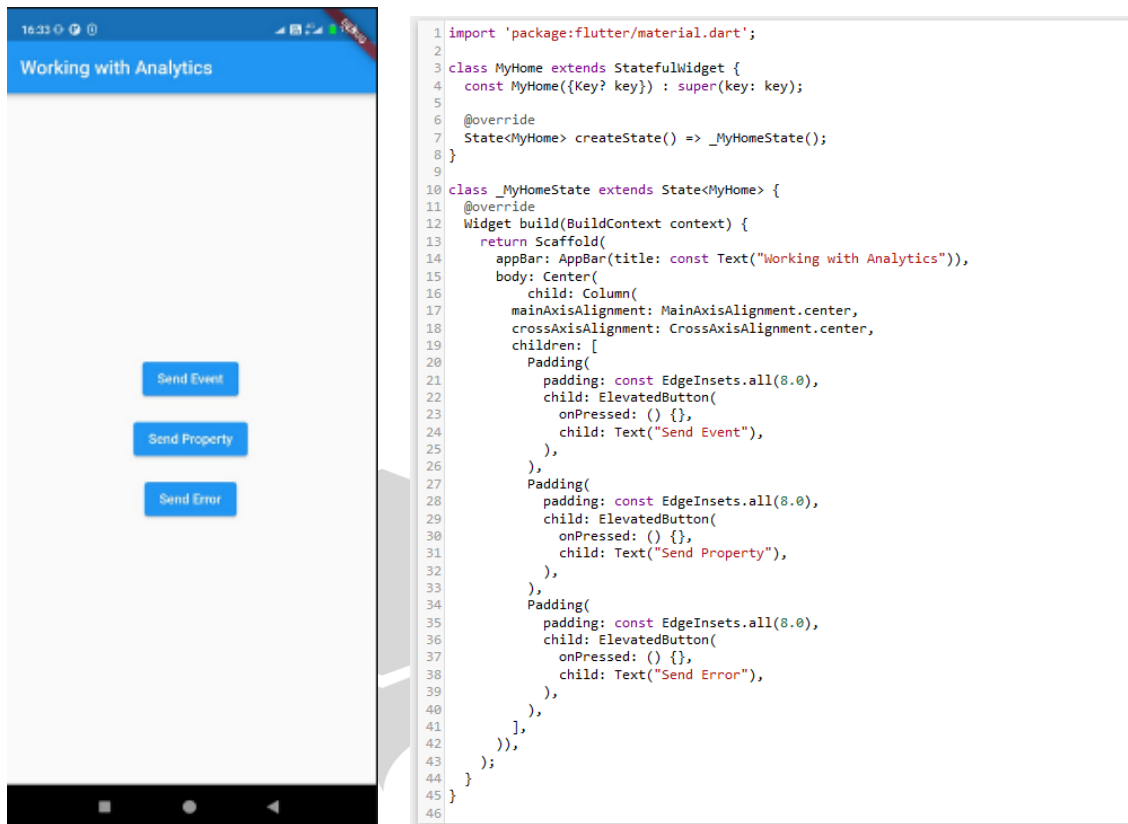
<https://console.firebase.google.com/>

Anda akan diminta untuk mengautentikasi diri Anda sebelum mengakses konsol. Jika anda tidak memiliki akun Google, Anda dapat membuatnya secara gratis. Sebagai contoh awal, anda dapat mencoba menghubungkan aplikasi anda di Flutter dengan Firebase, dengan memanfaatkan *firebase analytics*.

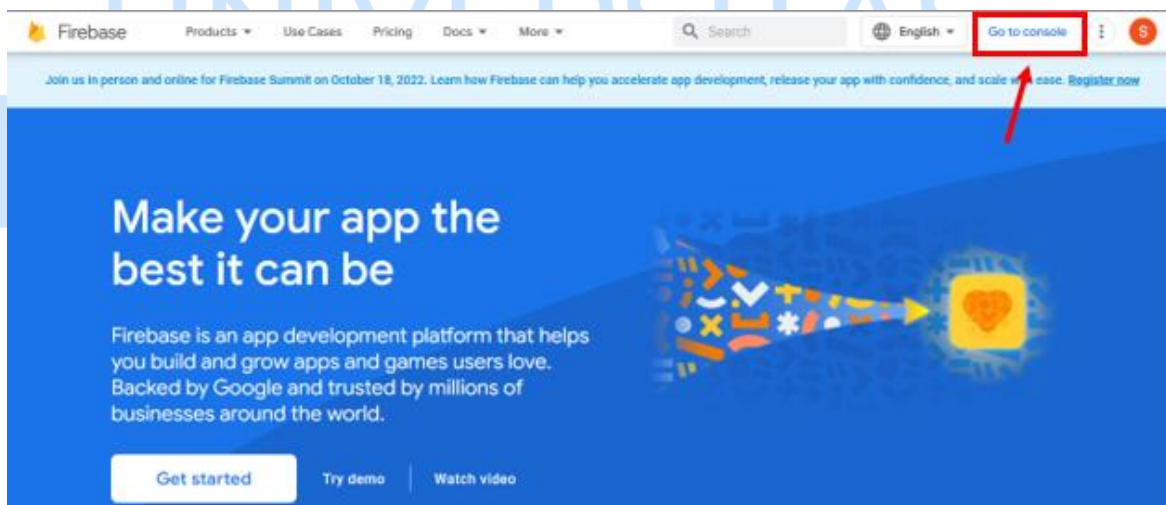
Fitur *analytics* adalah salah satu fitur pada Firebase yang digunakan sebagai koleksi data dan reporting penggunaan aplikasi Android maupun iOS. Sebagai contoh, anda bisa mendapatkan berapa banyak pengguna aplikasi di negara Indonesia, atau mungkin negara lain. Anda juga dapat melihat bagian dari aplikasi yang paling sering digunakan oleh user.

Dengan adanya fitur ini, anda dimungkinkan untuk dapat membuat segmentasi user yang bertujuan untuk mengidentifikasi aktivitas user seperti barang apa yang terbanyak dibeli di negara tertentu, dan bahkan dapat mencari tahu jam berapa transaksi dilakukan. Jadi, mari kita mulai mencoba untuk menghubungkan aplikasi kita dengan firebase, yaitu sebagai berikut:

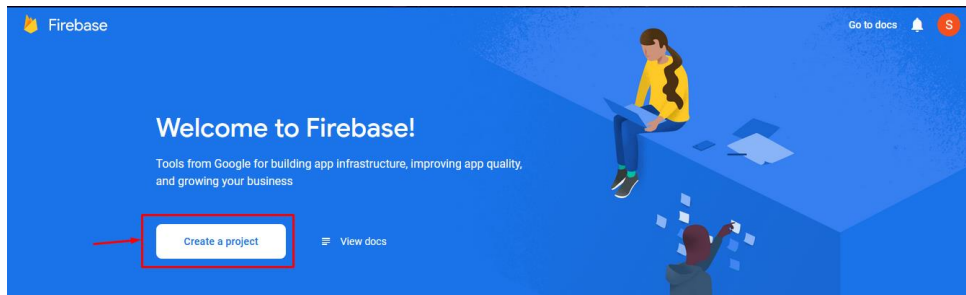
1. Anda boleh membuat proyek baru di dalam flutter anda, dan rancang tampilan seperti berikut ini:



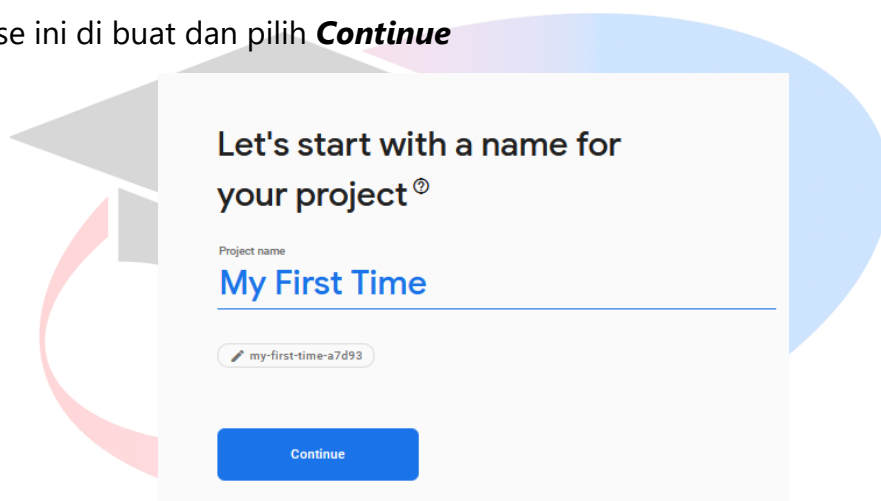
2. Pastikan anda login ke dalam <https://firebase.google.com/> setelah anda login menggunakan akun google anda, dan pilih **Go to Console / Buka Konsol**



3. Buatlah Proyek pertama firebase anda dengan memilih **Create a project**.

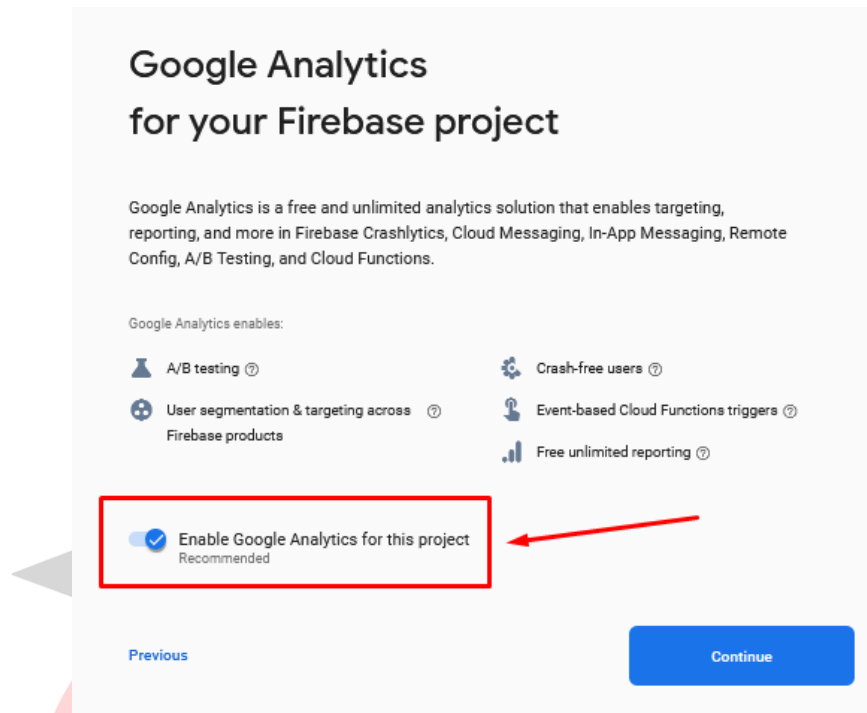


4. Berikan **nama proyek** anda, bebas selama anda mengingat tujuan dari proyek Firebase ini di buat dan pilih **Continue**

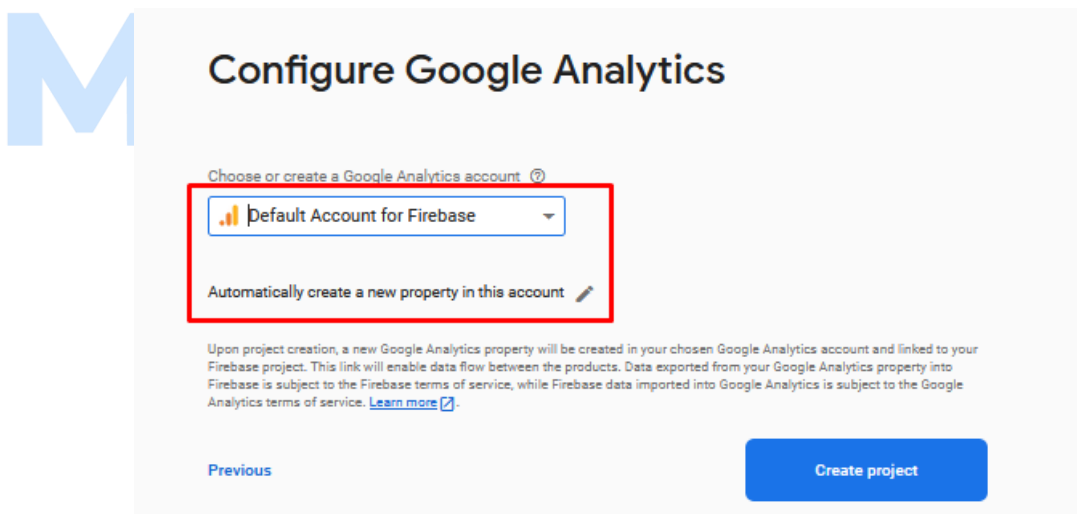


Catatan : nilai unik yang muncul pada chips di bawah nama proyek anda adalah sebuah id unik yang digunakan untuk mengenali Proyek anda

5. Langkah kedua merupakan Langkah yang penting dalam mengaktifkan analytics di dalam proyek anda. Pastikan **menyalakan analytics pada pilihan switch (secara default)**. Jika anda mematikan layanan ini, anda tetap dapat mengaktifkannya kembali di dalam pengaturan proyek anda di kemudian hari.

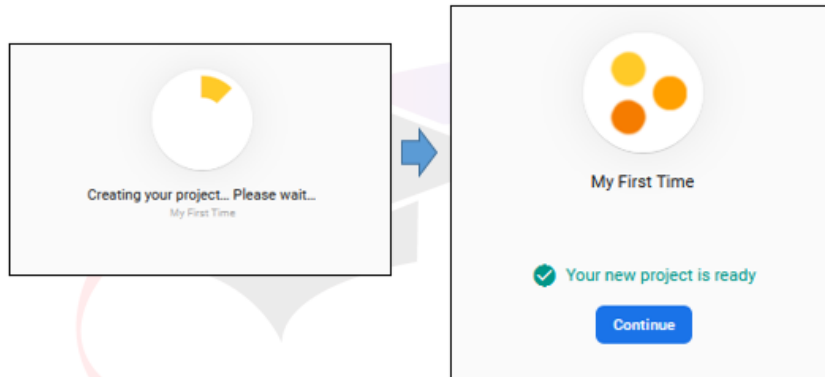


6. Ketikan anda mengaktifkan analytics, anda dapat menggabungkan hasil analytics anda dengan proyek yang sudah ada. Akan tetapi, jika kedua proyek tidak saling berhubungan, sangat disarankan untuk memisahkan proyek tersebut dengan memilih **Default Account for Firebase**, yang mana akan membuat proyek baru khusus untuk akun Google anda. Setelah itu anda dapat melanjutkan dengan klik **Create Project**.

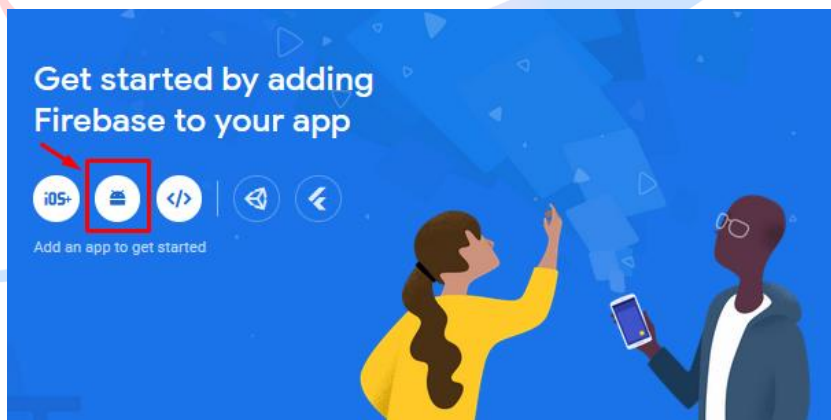


7. Silakan menunggu proses inialisasi proyek firebase anda siap untuk digunakan.

Klik **Continue**

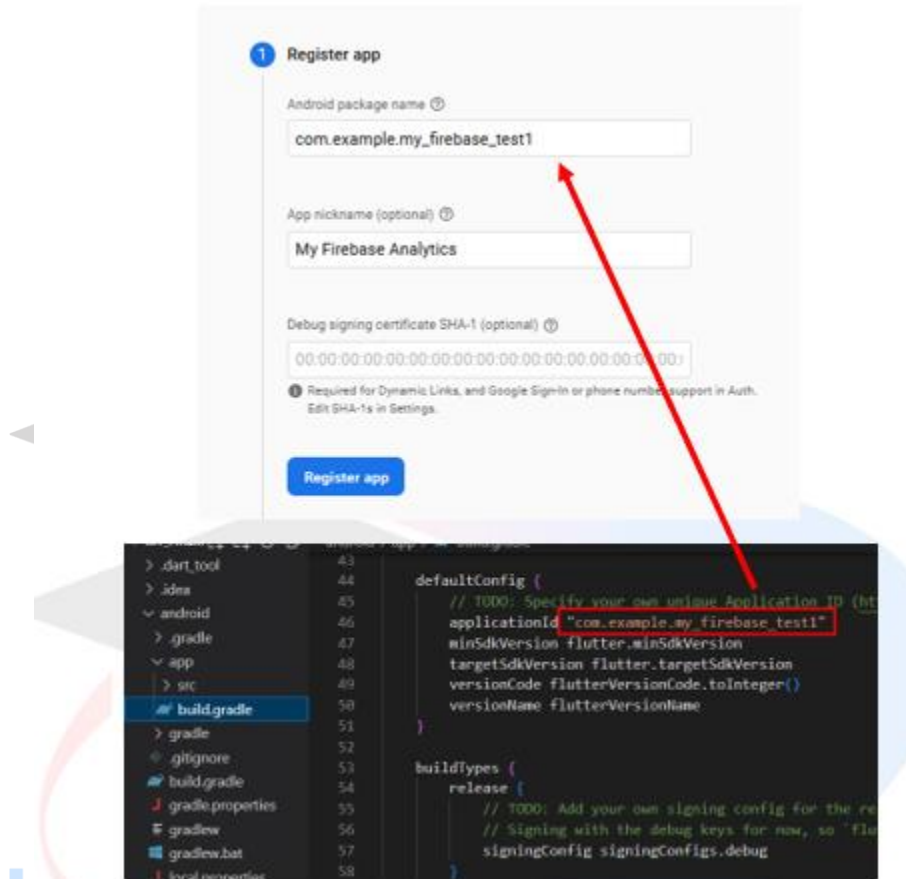


8. Anda telah berhasil membuat sebuah proyek firebase anda. Akan tetapi saat ini proyek firebase anda belum terhubung dengan aplikasi flutter yang anda buat. Klik **logo Android** untuk menghubungkan aplikasi android anda dengan firebase anda.

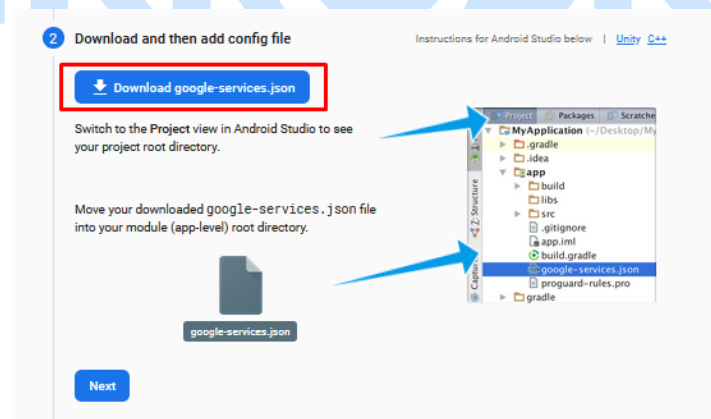


9. Pertama, anda harus memasukkan nama aplikasi id unik anda yang ada dalam pengaturan aplikasi anda. Nama aplikasi anda, dapat anda temukan pada File **Android/app/build.gradle**, pada **applicationId**. Adapun untuk **App Nickname**, bersifat opsional, hanya digunakan untuk memudahkan anda dalam mengingat nama aplikasi yang anda buat dengan lebih sederhana. Untuk **nilai SHA**, dapat anda masukkan, ketika aplikasi anda telah siap dikembangkan dan siap untuk

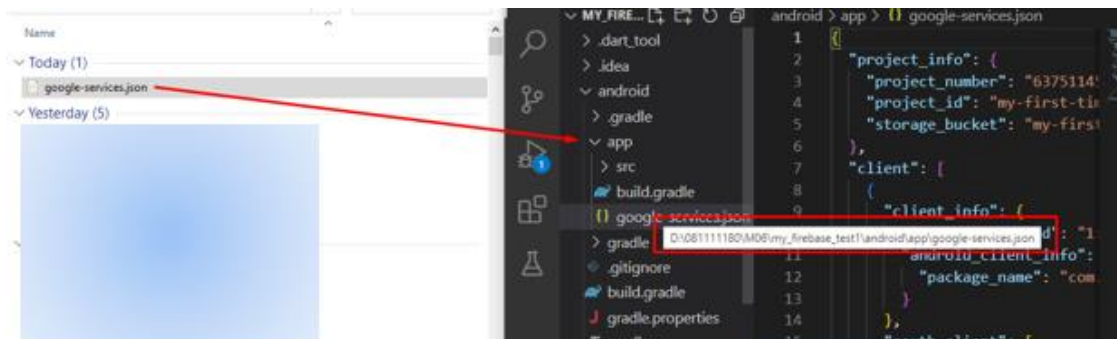
dipublikasi pada **Play Store** sehingga dapat anda kosongkan terlebih dahulu. Setelah mengisi data tersebut, klik **Register app**.



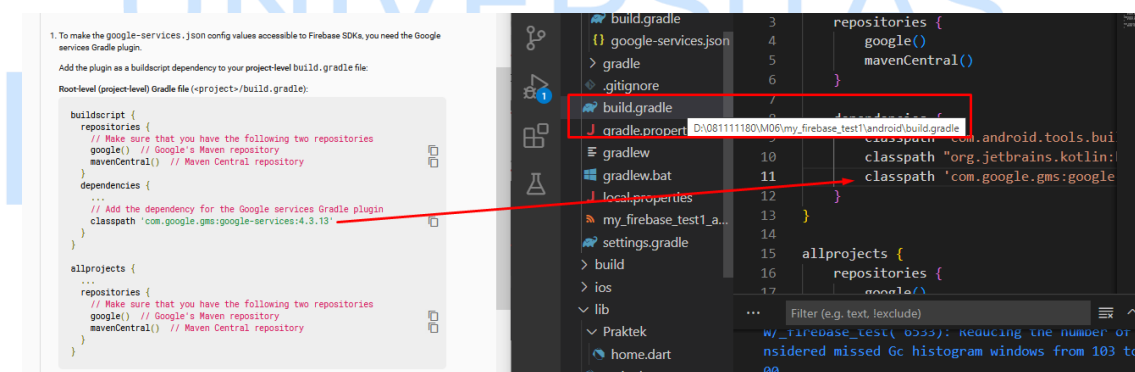
10. Langkah berikutnya, anda diwajibkan untuk mendownload file JSON yang diberikan. File ini berisikan pengaturan untuk menghubungkan aplikasi anda dengan Firebase.



11. Letakkan file yang telah anda download pada folder **Android/app** dengan cara mengklik dan drag file tersebut ke dalam folder **Android/app di Flutter**, atau anda dapat melakukan **Cut and Paste** pada directory project flutter anda di **...//<nama_proyek>/android/app/**. Pastikan anda tidak salah meletakkan file pengaturan ini, kesalahan peletakan file ini dapat mengakibatkan aplikasi anda tidak terhubung dengan firebase nantinya.

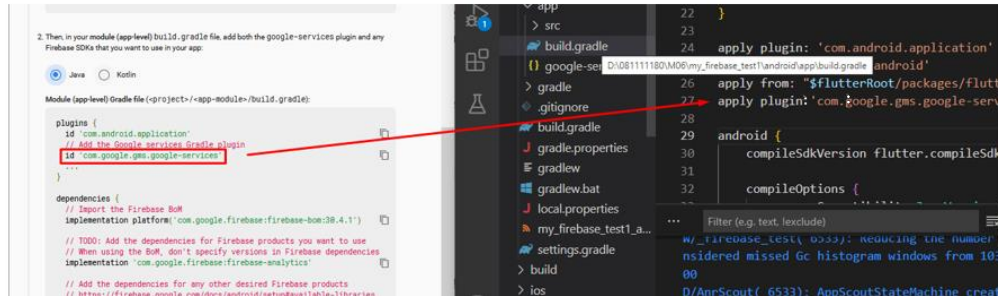


12. Tahap berikutnya, pastikan anda mencopy **classpath 'com.google.gms:google-services:4.3.13'**, dan memasukkan teks tersebut pada **build.gradle** level project anda pada bagian **dependencies**, Bukan pada level app, perhatikan letak dari build.gradle yang dibuka pada gambar di bawah. Jangan lupa untuk menyimpan file anda (**Ctrl+S**)

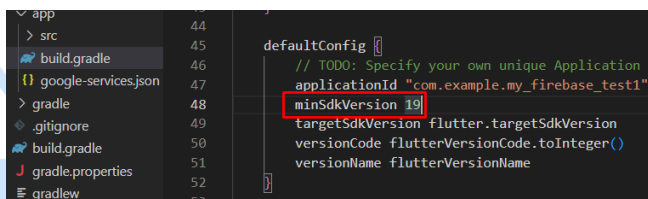


13. Berikutnya, anda perlu menambahkan plugin pada **build.gradle** level app anda, dengan **id 'com.google.gms.google-services'**. Untuk **dependencies implementation platform('com.google.firebase:firebase-bom:30.4.1')** dan **implementation 'com.google.firebase:firebase-analytics'** tidak perlu anda

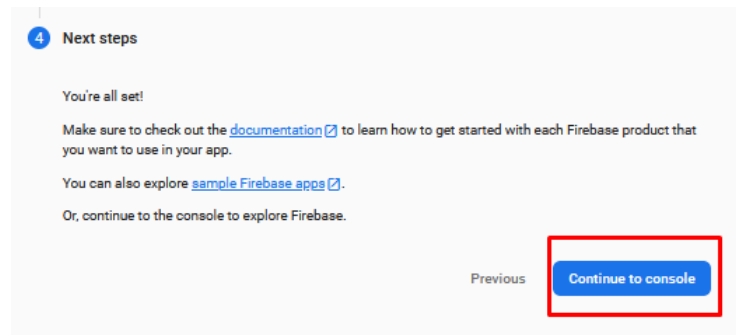
tambahkan, karena anda bekerja menggunakan **pubspec.yaml**. Jangan lupa untuk menyimpan file anda (**Ctrl+S**)



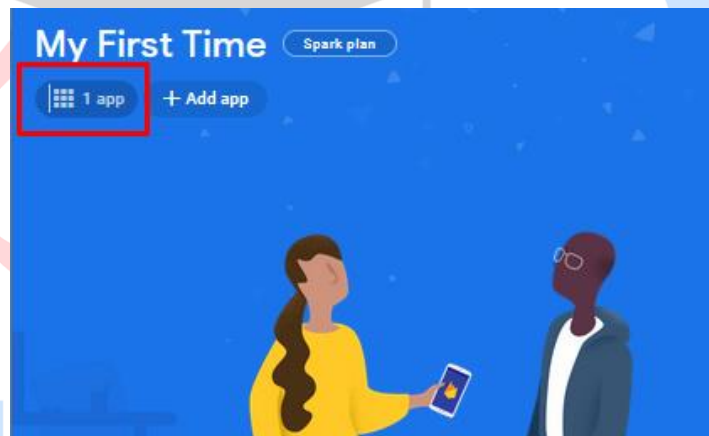
Pada **build.gradle** level app, anda juga harus mengatur Sdk Minimal dalam menjalankan aplikasi. Dikarenakan analitik hanya dapat berjalan dari SDK 19, maka ubah minSdkVersion anda pada **build.gradle** level app menjadi minSdkVersion 19 atau lebih besar.



14. **Klik Next.** Anda telah selesai melakukan konfigurasi firebase dengan aplikasi anda. Anda dapat menyelesaikan tahapan ini dengan kembali ke konsol aplikasi anda dengan klik **continue to console**.



Anda dapat melihat bahwa proyek anda telah terhubung dengan firebase pada tampilan konsol anda. Dengan melakukan pengaturan ini, anda telah dapat menggunakan fasilitas firebase di dalam aplikasi anda, baik firestore, firebase realtime database, autentikasi dan lainnya.



Setelah berhasil menghubungkan aplikasi dengan firebase. Langkah ini tetap harus dilakukan untuk setiap proyek baru yang anda buat dan ingin dihubungkan ke dalam firebase. Saatnya anda untuk menggunakan fitur analytics pada project flutter anda. Yang perlu anda lakukan adalah:

1. Lakukan konfigurasi pada **pubspec.yaml** file dengan menggunakan pubspec assists, untuk menambahkan **firebase_core** and **firebase_analytics**. **Firestore_core**, merupakan library penting pada setiap firebase, karena digunakan sebagai pengaturan awal dalam setiap operasi di dalam firebase. Sedangkan **firebase_analytics** digunakan untuk menjalankan fungsi analitik di dalam firebase melalui proyek anda.

```
firebase_core, firebase_analytics
```

Enter package names, separated by commas. (Press 'Enter' to confirm or 'Escape' to cancel)

2. Firebase analytics tidak langsung memberikan Informasi setiap aktivitas, bahkan mungkin memerlukan waktu hingga 24 jam. Sebelum menjalankan aplikasi anda, harus mengatur pengaturan untuk mengaktifkan DebugView di konsol firebase menggunakan perintah ini pada **adb.exe** pada **command prompt** anda

adb shell setprop debug.firebase.analytics.app <<ApplicationID>>

```

Microsoft Windows [Version 10.0.19042.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sunaryo Winardi>cd AppData\Local\Android\Sdk\platform-tools
C:\Users\Sunaryo Winardi\AppData\Local\Android\Sdk\platform-tools>adb shell setprop debug.firebase.analytics.app com.example.my_firebase_test1
C:\Users\Sunaryo Winardi\AppData\Local\Android\Sdk\platform-tools>

```

Ini akan mengaktifkan **DebugView** di konsol firebase sehingga anda dapat langsung mendapatkan info analytics aplikasi anda.

3. Lakukan inisialisasi Firebase ke dalam aplikasi dengan menggunakan fungsi firebase berikut pada fungsi main() anda.

```

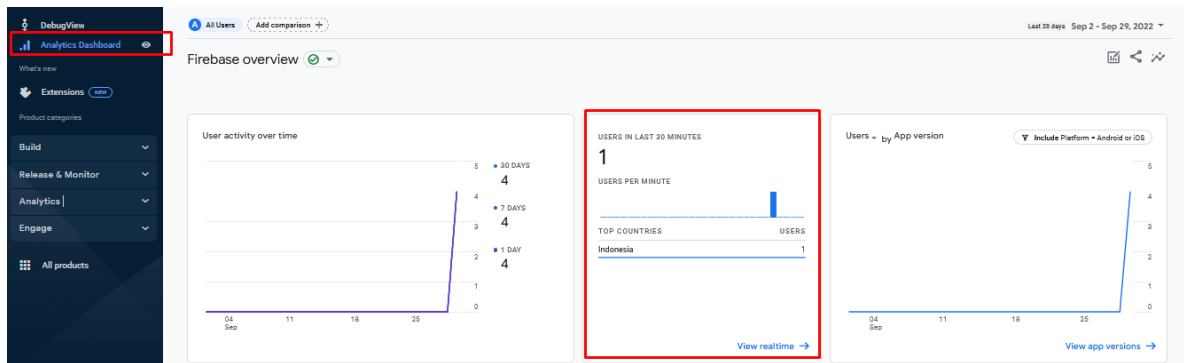
Run | Debug | Profile
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

```

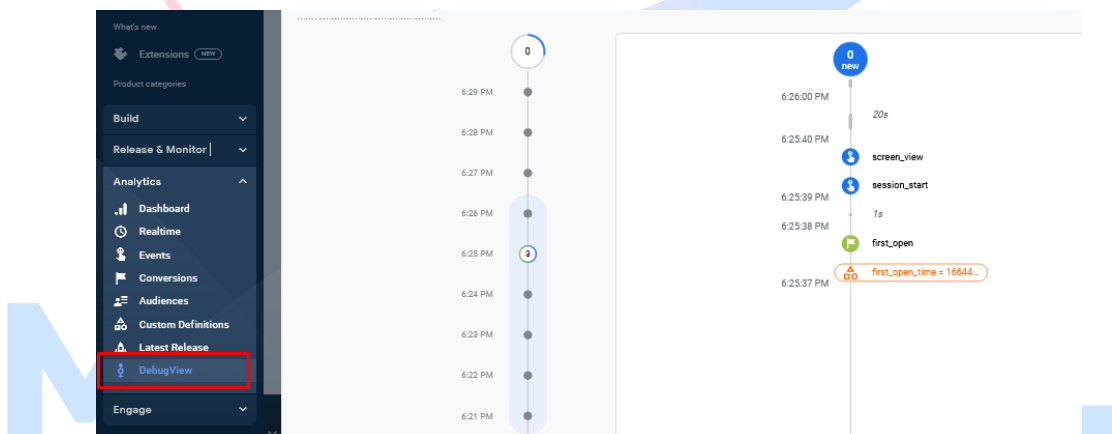
Fungsi **Firebase.initializeApp()** digunakan untuk melakukan pengaturan dalam menghubungkan firebase dengan aplikasi anda. Proses ini harus dilakukan dengan proses asinkron, sehingga kita menggunakan Fungsi **Future**. Proses ini dapat menyebabkan proses build widget aplikasi pada terganggu. Oleh karena itu anda harus memastikan proses build widget selesai sebelum menjalankan perintah `Firebase.initializeApp()` dengan menggunakan fungsi **WidgetsFlutterBinding.ensureInitialized();**

4. Dengan menjalankan perintah ini, anda dapat melihat analytics anda telah berjalan. Hasilnya dapat anda lihat pada Menu Dashboard Analytics, bahwa ada

1 user yang aktif pada 30 menit terakhir (ingat proses ini kadang memakan waktu beberapa menit sebelum firebase menerima report)



- Anda juga dapat melihat pada bagian **DebugView**, beberapa Informasi analitik telah didapatkan seperti, pertama kali aplikasi di buka oleh pengguna baru (**first_open**), operasi dimulai (**session_start**), dan pertama kali screen terbuka di aplikasi (**screen_view**). Ini merupakan event otomatis yang akan selalu dicatat oleh Firebase Analytics.



- Untuk mengatur analytics (mengirimkan data sesuai dengan kebutuhan anda), anda dapat membuat sebuah file terpisah dengan membuat sebuah class dengan nama **MyAnalyticsHelper** dan menginisialisasi setiap fungsi analytics yang dapat anda gunakan dari **FirebaseAnalytics**.

```

1  import 'package:firebase_analytics/firebase_analytics.dart';
2
3  class MyAnalyticsHelper {
4    FirebaseAnalytics analytics = FirebaseAnalytics.instance;
5  }
6

```

7. Setelah mengonfigurasi instance **FirebaseAnalytics**, Anda dapat mulai mencatat peristiwa di dalam aplikasi ke dalam log dengan metode `logEvent()`. Setiap proses di dalam **FirebaseAnalytics** harus dijalankan di dalam fungsi `async Future`. Masukkan fungsi berikut, dengan catatan, setiap **name dari logEvent, wajib lowercase**. Untuk properti lainnya dapat anda sesuaikan.

```

1  import 'package:firebase_analytics/firebase_analytics.dart';
2
3  class MyAnalyticsHelper {
4    FirebaseAnalytics analytics = FirebaseAnalytics.instance;
5
6    Future<void> testEventLog(_value) async {
7      await analytics
8        .logEvent(name: '${_value}_click', parameters: {'Value': _value});
9      print('Send Event');
10   }

```

8. Untuk menjalankan fungsi `Future` yang telah kita buat, pastikan anda membuat instance dari `MyAnalyticsHelper` sebagai berikut:

```

11 class _MyHomeState extends State<MyHome> {
12   MyAnalyticsHelper fbAnalytics = MyAnalyticsHelper();
13   @override
14   Widget build(BuildContext context) {

```

9. Jalankanlah setiap event Ketika button di klik, dengan mengirimkan event button yang dijalankan seperti berikut ini

```

22   Padding(
23     padding: const EdgeInsets.all(8.0),
24     child: ElevatedButton(
25       onPressed: () {
26         fbAnalytics.testEventLog("send_event");
27       },
28       child: Text("Send Event"),
29     ), // ElevatedButton
30   ), // Padding

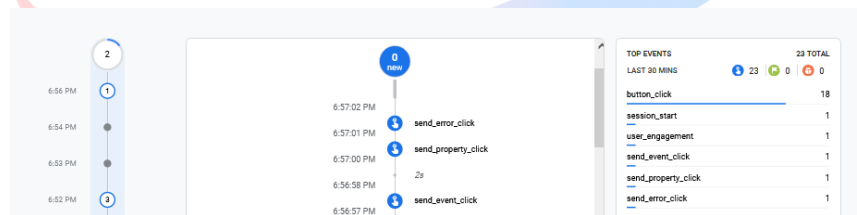
```

```

31      Padding(
32        padding: const EdgeInsets.all(8.0),
33        child: ElevatedButton(
34          onPressed: () {
35            fbAnalytics.testEventLog("send_property");
36          },
37          child: Text("Send Property"),
38        ), // ElevatedButton
39      ), // Padding
40      Padding(
41        padding: const EdgeInsets.all(8.0),
42        child: ElevatedButton(
43          onPressed: () {
44            fbAnalytics.testEventLog("send_error");
45          },
46          child: Text("Send Error"),
47        ), // ElevatedButton
48      ), // Padding

```

10. Jalankan aplikasi, dan klik button tersebut. Jika aplikasi anda sudah terbaca pada **DebugView**, maka anda akan melihat setiap log yang anda kirimkan telah muncul pada **DebugView**.



11. Anda juga dapat mencoba beberapa fungsi dalam Firebase analytics lainnya seperti, mengirimkan **id user** (mengirimkan user yang sedang menggunakan aplikasi) dan **property** (digunakan untuk menerima inputan dari sebuah variable ke analitik anda), dengan menggunakan fungsi berikut:

```

12      Future<void> testSetUserId(_value) async {
13        await analytics.setUserId(id: '$_value');
14        print('setUserId succeeded');
15      }
16
17      Future<void> testSetUserProperty() async {
18        await analytics.setUserProperty(name: 'regular', value: 'indeed');
19        print('setUserProperty succeeded');
20      }

```

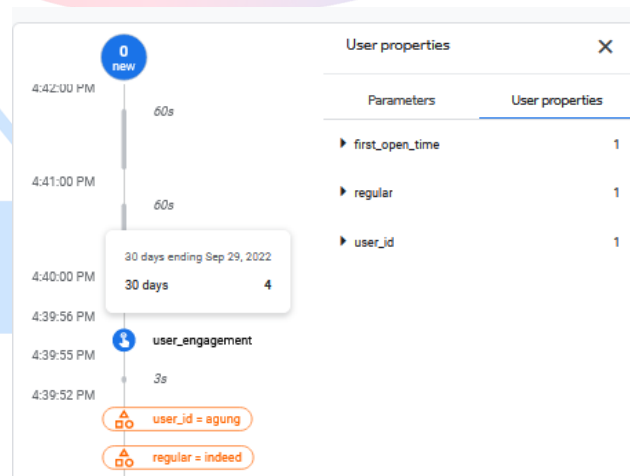
12. Jalankanlah kedua fungsi dengan mengganti fungsi yang akan dijalankan pada button event onPressed **tombol Send Event**, dan **tombol Send Property** seperti berikut:

```

22      Padding(
23        padding: const EdgeInsets.all(8.0),
24        child: ElevatedButton(
25          onPressed: () {
26            fbAnalytics.testSetUserId("agung");
27          },
28          child: Text("Send Event"),
29        ), // ElevatedButton
30      ), // Padding
31      Padding(
32        padding: const EdgeInsets.all(8.0),
33        child: ElevatedButton(
34          onPressed: () {
35            fbAnalytics.testSetUserProperty();
36          },
37          child: Text("Send Property"),
38        ), // ElevatedButton
39      ), // Padding

```

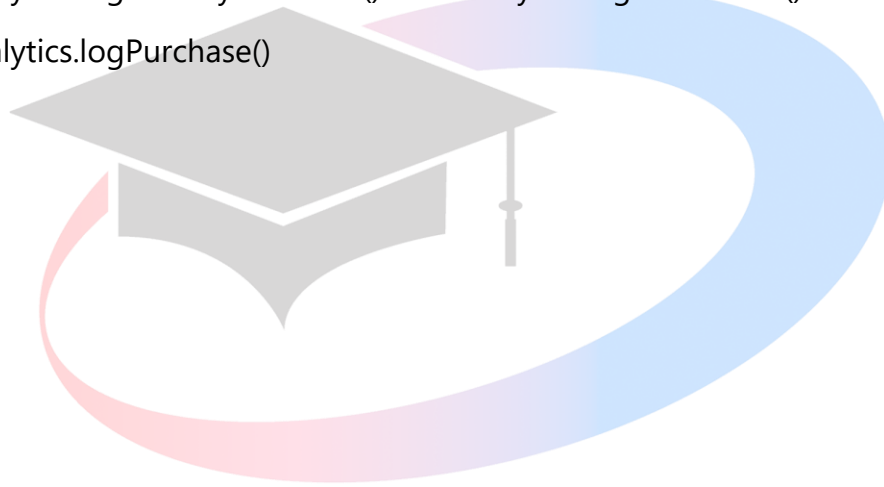
13. Perhatikan pada **DebugView**, maka anda telah berhasil mengirimkan data user id dan properti ke dalam analitik anda.



Anda dapat menambahkan Informasi log lainnya, dengan melihat penggunaan firebase analytics pada https://pub.dev/packages/firebase_analytics/example

Latihan

1. Setelah membuat proyek yang terhubung dengan Firebase Analytics, tambahkan beberapa event dan jelaskan fungsi dari:
 - a. `analytics.setSessionTimeoutDuration`
 - b. `analytics.appInstanceId`
 - c. `analytics.resetAnalyticsData()`
 - d. `.analytics.logAddPaymentInfo()` dan `.analytics.logAddToCart()`
 - e. `.analytics.logPurchase()`



UNIVERSITAS
MIKROSKIL