

**TUGAS UNGUIDED & GUIDED
PEMROGRAMAN PERANGKAT BERGERAK
MODUL XI
FIREBASE NOTIFIKASI**



Disusun Oleh :

Arwin Nabiel Arroffif

2211104057

SE 06 2

Asisten Praktikum :

Ayu susiloawati

Noviana Rizki Anisa putri

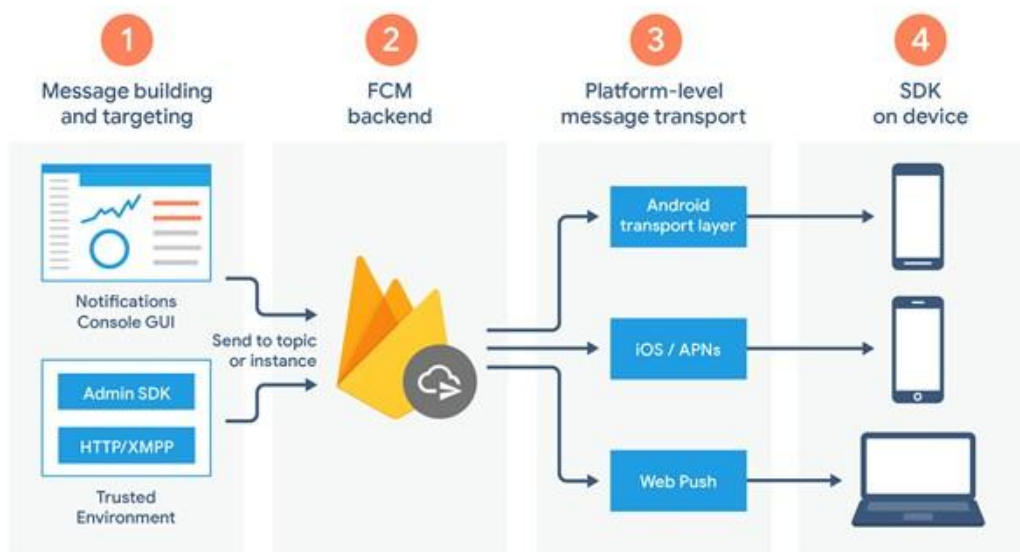
Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs.

**PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS
INFORMATIKA TELKOM UNIVERSITY PURWOKERTO 2024**

FIREBASE NOTIFIKASI

Firestore Cloud Messaging



Gambar 1. Cara Kerja FCM

Firestore Cloud Messaging (FCM) adalah layanan yang digunakan untuk mengirimkan notifikasi dengan mudah ke banyak pengguna sekaligus. Dengan FCM, Kita dapat mengirimkan pesan baik ke perangkat individu maupun ke grup pengguna yang berlangganan topik tertentu. Dokumentasi lengkapnya dapat diakses di <https://firebase.google.com/docs/flutter/setup?platform=android> Agar dapat mengirim pesan, Kita perlu:

1. Mendapatkan *token* unik dari setiap perangkat pengguna.
2. Atau mengelompokkan pengguna berdasarkan topik tertentu (misalnya "promo" atau "berita").

Setelah itu, notifikasi dapat dikirim menggunakan **Firestore Cloud Functions**.

A. Jenis Pesan pada Firestore Cloud Messaging (FCM)

FCM memiliki dua jenis pesan utama yang digunakan untuk kebutuhan notifikasi, yaitu:

1) Notification Message

Pesan ini digunakan untuk langsung menampilkan notifikasi di perangkat pengguna.

Cara kerja:

- Jika aplikasi berjalan di latar depan (foreground), pesan diteruskan ke fungsi onMessage untuk diproses.
- Jika aplikasi di latar belakang (background) atau dimatikan (terminated), notifikasi akan langsung muncul di tray notifikasi perangkat tanpa memerlukan kode tambahan.

2) Data Message

Pesan ini memberikan lebih banyak fleksibilitas karena isinya tidak langsung ditampilkan sebagai notifikasi.

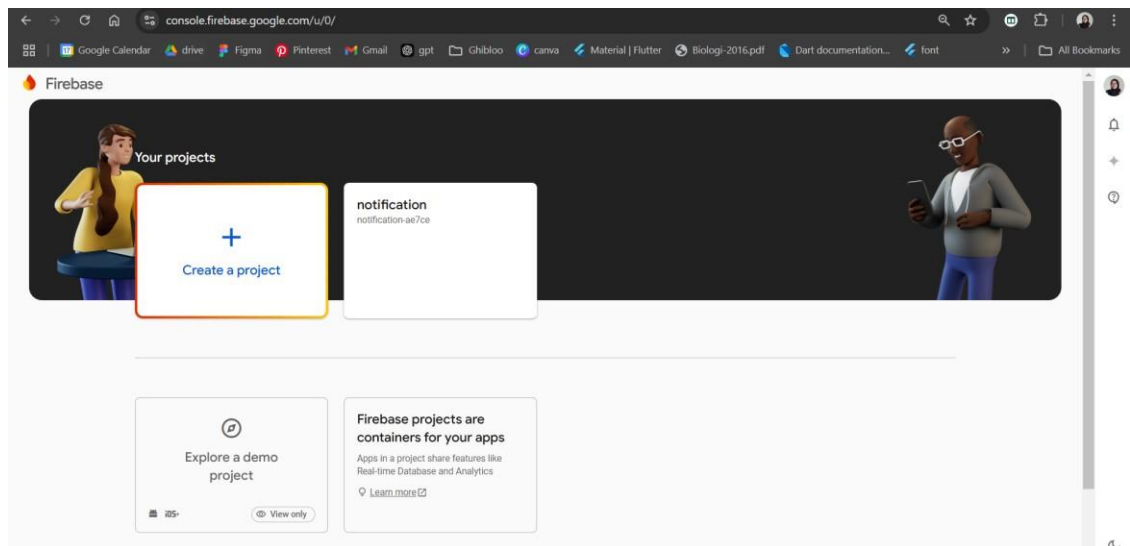
Cara kerja:

- Pesan ini akan diproses oleh aplikasi, baik saat aplikasi aktif maupun tidak aktif.
- Untuk menampilkan notifikasi ke pengguna, Kita perlu menambahkan kode khusus, biasanya menggunakan plugin seperti flutter_local_notifications.

A. Integrasi Aplikasi dengan Firebase

Untuk memulai, kita perlu menyiapkan akun di Firebase. Di bawah ini adalah langkah-langkah yang perlu kita ikuti untuk menyiapkan Google Firebase untuk notifikasi push di Android:

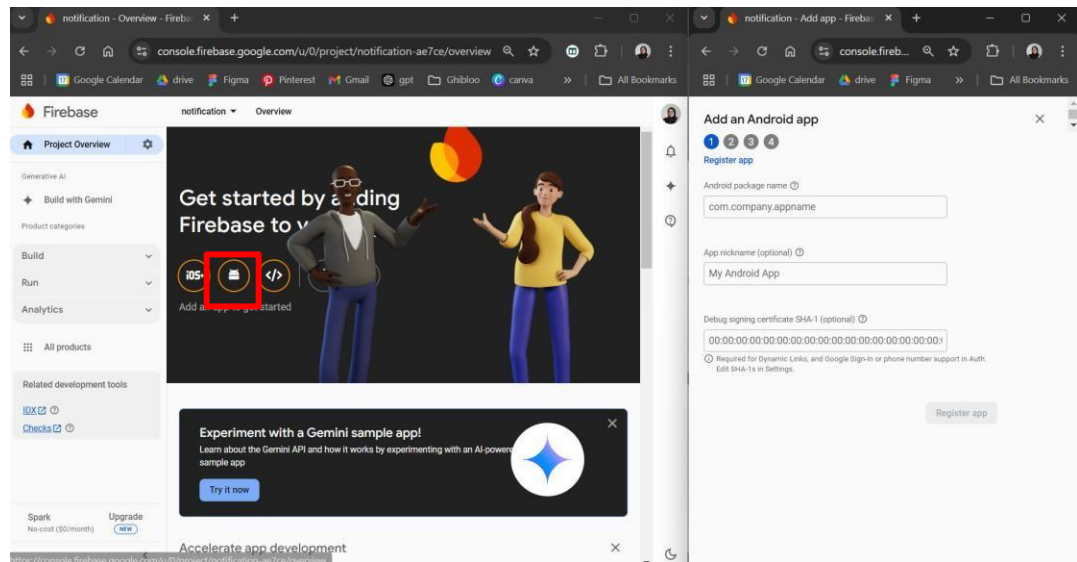
- Daftar dan login pada [firebase console](https://console.firebase.google.com/) lalu buat sebuah projek baru.



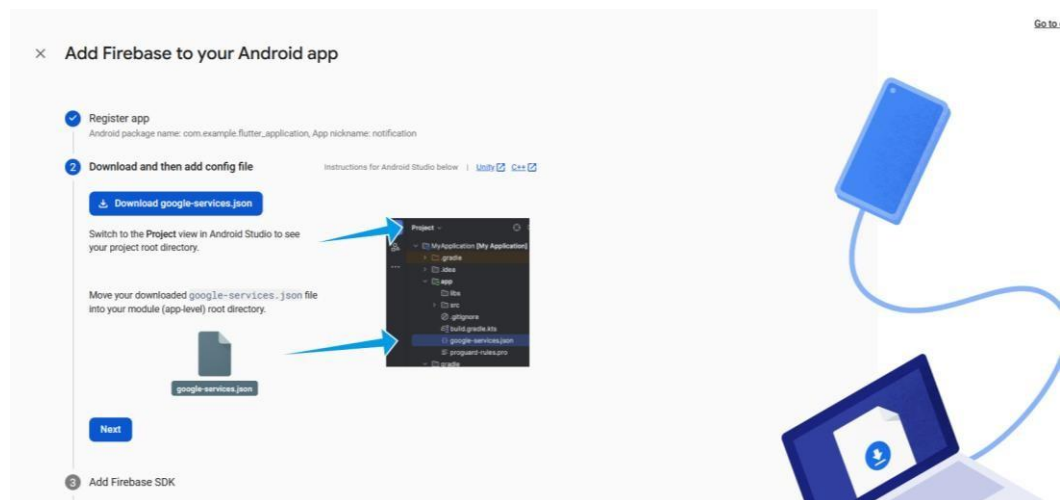
- Tambahkan android package name, app nickname and SHA-1 signing certificate key.

Android package name dapat ditemukan di **./android/app/ build.gradle**.

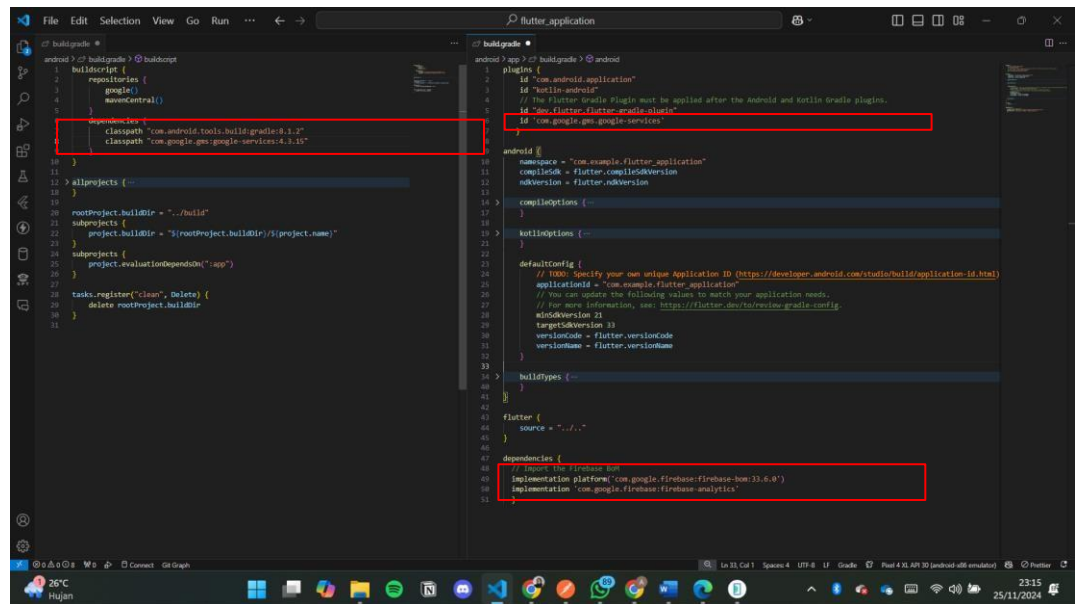
Catatan: App nickname and SHA-1 signing certificate key itu opsional.



- c. Setelah berhasil mendaftarkan aplikasi ke firebase, download file google-services.json lalu simpan pada **./android/app/**



- d. Tambahkan plugin dan sdk sesuai perintah pada **./android/build.gradle** dan **./android/app/build.gradle** lalu lakukan sync dengan mengetik **“flutter pub get”** pada terminal.



- e. Setelah memodifikasi file build.gradle, tambahkan beberapa package pada file pubspec.yaml dengan mengetik “flutter pub add **firebase_core** **firebase_messaging** **flutter_local_notifications**” pada terminal.
- f. Ubah compile sdk agar menggunakan versi 33 dan min sdk pada versi 21 yang terletak pada file **./android/app/build.gradle**

```
minSdkVersion 21
targetSdkVersion 33
```

B. Membuat Notifikasi Handler

- a. Pertama, tambahkan kode di fungsi main() untuk memastikan Firebase dan konfigurasi lainnya sudah siap digunakan

```
WidgetsFlutterBinding.ensureInitialized(); await
Firebase.initializeApp();

FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
await
FlutterLocalNotificationsPlugin()

.resolvePlatformSpecificImplementation<AndroidFlutterLocalNotificationsPlugin>()
?.createNotificationChannel(channel);
await
FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
alert: true, badge: true, sound: true,
);
```

- b. Tambahkan kode berikut di bawah import sebagai variabel global.

```
String? token;

Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message)
async {
  await Firebase.initializeApp();
  print('Handling a background message:
${message.messageId}'); }
const AndroidNotificationChannel channel = AndroidNotificationChannel(
  'high_importance_channel', // ID Channel
  'High Importance Notifications', // Nama Channel
  description: 'This channel is used for important notifications.', //
Deskripsi Channel
  importance: Importance.high, // Prioritas
);
```

c. Di dalam kelas State, tambahkan konfigurasi untuk menangani notifikasi

```
@override void
initState() {
  super.initState();

  // Membuat pengaturan inisialisasi notifikasi untuk Android
  var initializationSettingsAndroid = const
  AndroidInitializationSettings('@mipmap/ic_launcher'); var
  initializationSettings =
    InitializationSettings(android: initializationSettingsAndroid);
  FlutterLocalNotificationsPlugin().initialize(initializationSettings);
  // Mendengarkan pesan saat aplikasi aktif
  FirebaseMessaging.onMessage.listen((RemoteMessage message) {
    RemoteNotification? notification = message.notification;
    AndroidNotification? android = message.notification?.android;
    // Jika notifikasi tersedia, tampilkan menggunakan notifikasi
    lokal if (notification != null && android != null) {
      FlutterLocalNotificationsPlugin().show(
        notification.hashCode,
        // ID notifikasi (hashCode untuk unik) notification.title, //
        Judul notifikasi notification.body, // Isi notifikasi
        NotificationDetails(
          android: AndroidNotificationDetails(
            channel.id,
            channel.name,
            channelDescription: channel.description,
            color:
            Colors.blue,
            icon: "@mipmap/ic_launcher",
          ),
        ),
      ),
    ),
  ),
```

```

    );
}
});

// Menangani aksi ketika notifikasi dibuka
FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
    RemoteNotification? notification = message.notification;
    AndroidNotification? android = message.notification?.android;
    // Jika notifikasi tersedia, tampilkan dialog if
    (notification != null && android != null) {
        showDialog(
            context: context,
            builder: (_) {
                return
                AlertDialog(
                    title: Text(notification.title ?? ""), //
                    Judul dialog
                    content: SingleChildScrollView(
                        child: Column(
                            crossAxisAlignment:
                            CrossAxisAlignment.start,
                            children:
                            [Text(notification.body ?? "")], // Isi dialog
                        ),
                    ),
                );
            },
        );
    }
});

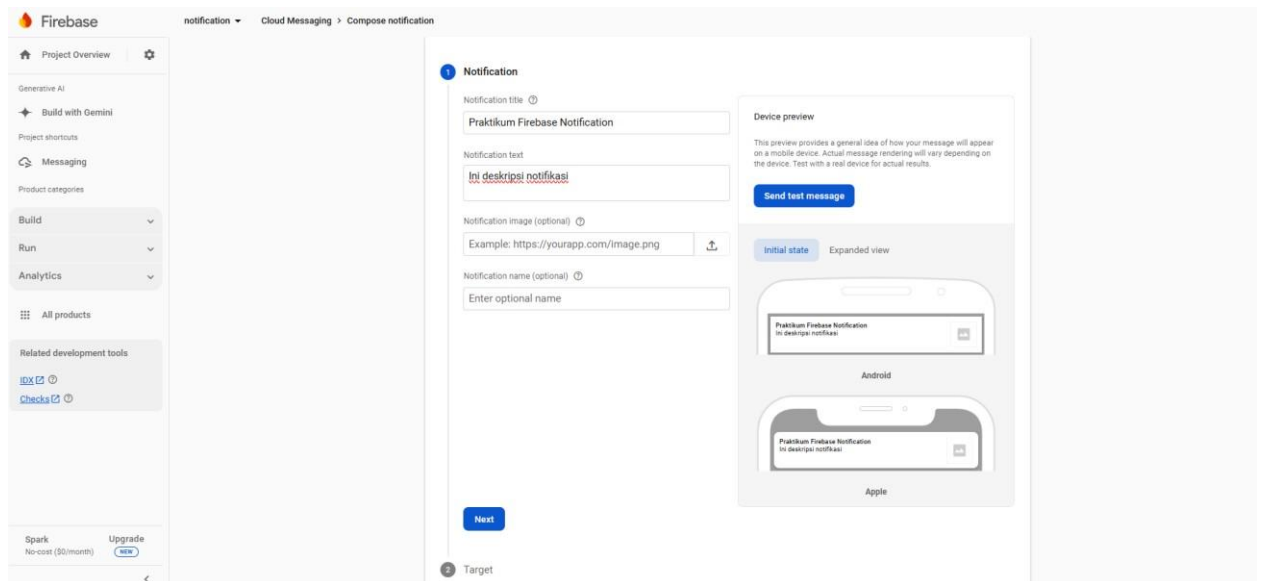
// Memanggil metode untuk mengambil token FCM perangkat
getToken();

// Metode untuk mendapatkan token FCM
void getToken() async {
    token = await
    FirebaseMessaging.instance
        .getToken(); // Mendapatkan token FCM perangkat
    print('FCM Token: $token'); // Menampilkan token di log }
}

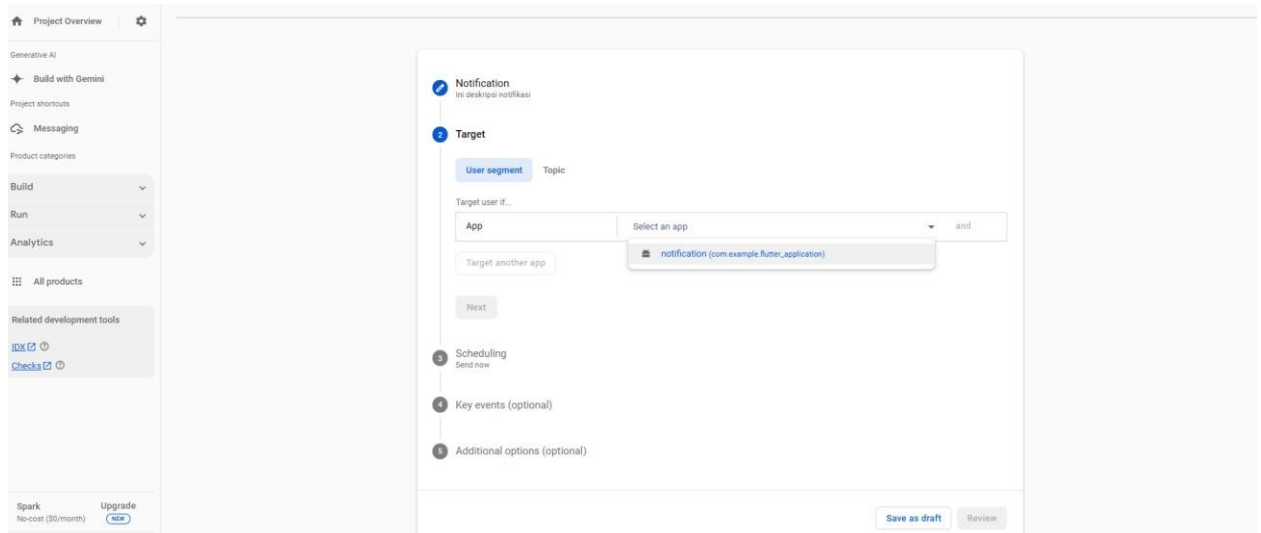
```

C. Mengirim Notifikasi Via FCM Console

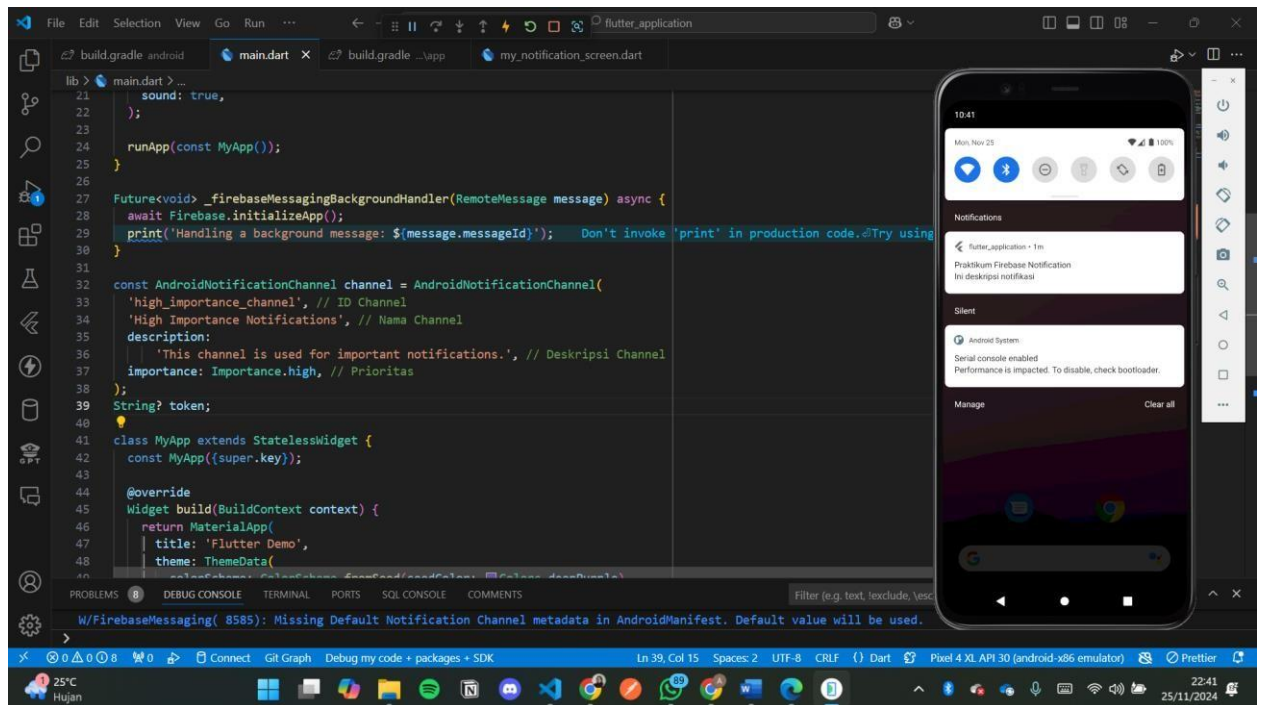
- a. Gaskeun waktunya testing notifikasi kalian. Dimulai dengan membuka halaman all product pada tab firebase --> pilih menu cloud messaging --> lalu pilih send your first message.
- b. Selanjutnya, inputkan judul, pesan, dan foto dari notifikasi kalian sesuai dengan kebutuhan.



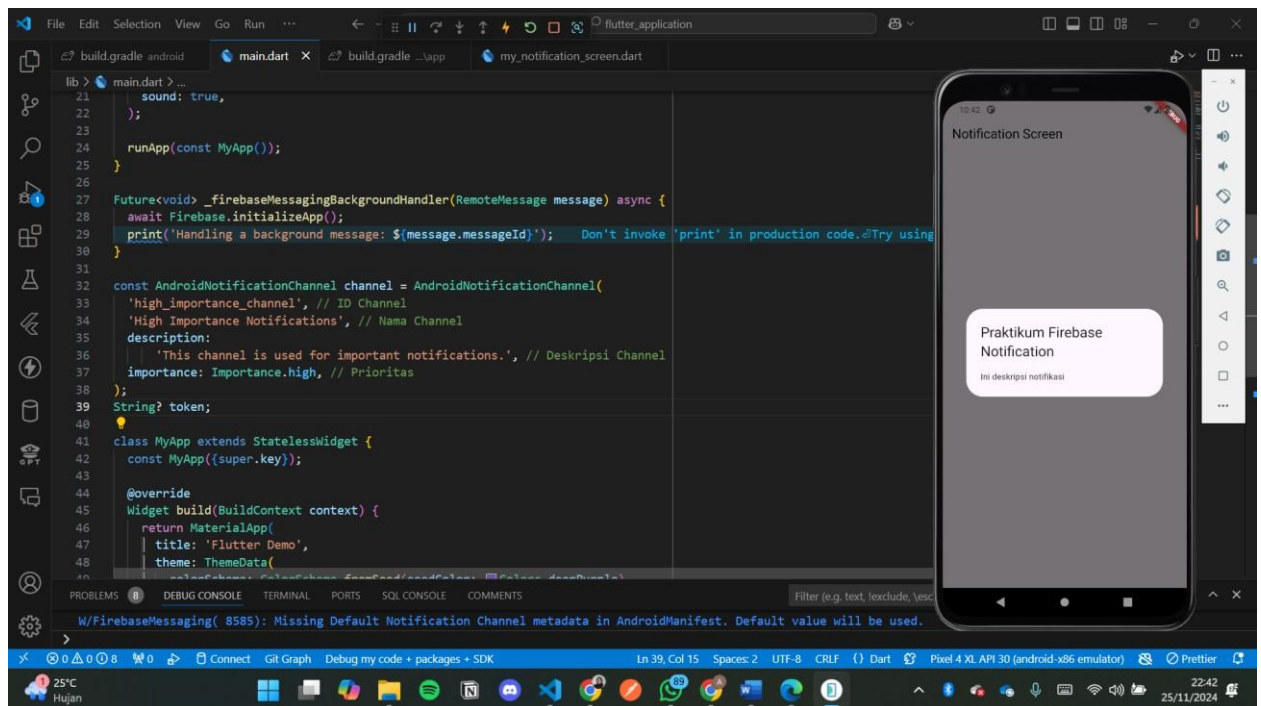
- c. Setelah itu, pilih aplikasi mana yang akan dikirimkan notifikasi dan setting waktu "Now" jika ingin langsung mengirimkan notifikasi. Lalu klik publish jika isi dan settingan notifikasi sudah benar.



- d. Silakan tunggu notifikasi muncul pada aplikasi. Catatan: pengiriman notifikasi memerlukan beberapa saat dan tidak akan langsung muncul setelah dikirim jadi silakan menunggu.



- e. Klik notifikasi maka akan di redirect ke aplikasi dan akan menampilkan seperti pada gambar dibawah.

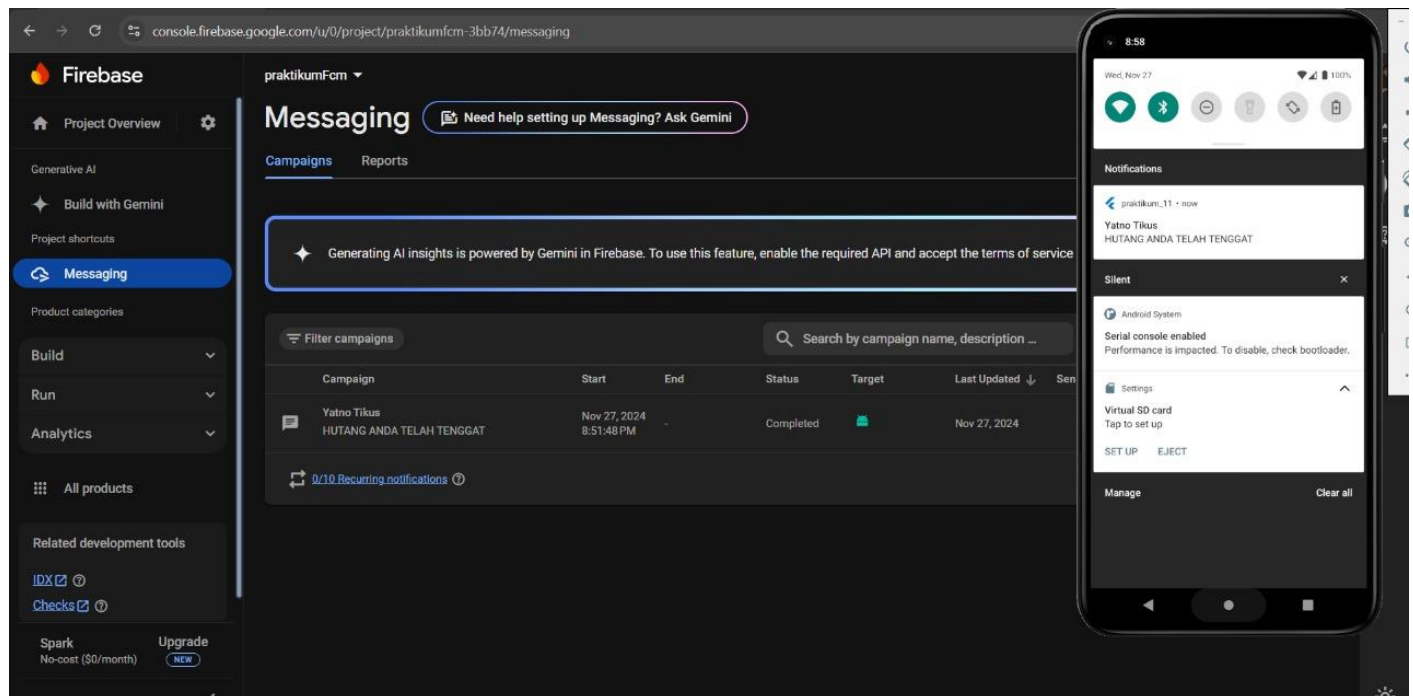


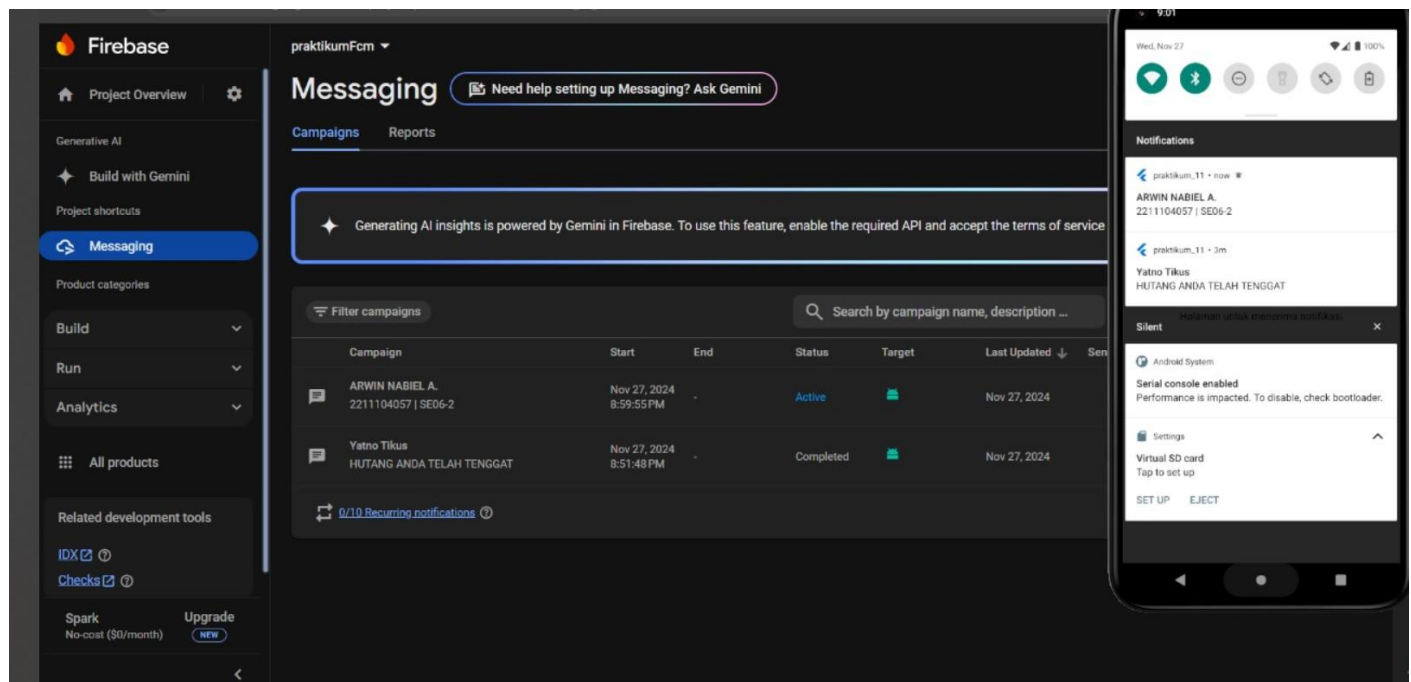
- f. Selamat notifikasi anda berjalan sesuai harapan. Keep coding and don't forget to take a rest...

Tugas Mandiri (Unguided)

Modifikasi Guided diatas bisa menampilkan Nama, Nim, Kelas, dan Prodi kalian ke dalam Notifikasi Flutter Cloud Messaging dan landing page notifikasinya.

Note: Jangan lupa sertakan source code, screenshot output, dan deskripsi program. Kreatifitas menjadi nilai tambah.





Untuk kodingan ada di folder masing – masing UNGUIDED & GUIDED

Deskripsi program:

Program ini merupakan implementasi Firebase Cloud Messaging (FCM) pada aplikasi Flutter untuk mengirim dan menerima notifikasi. Dengan menggunakan FCM, aplikasi dapat menerima pesan notifikasi baik dalam mode latar depan maupun latar belakang. Program ini mencakup pengaturan Firebase, pengelolaan notifikasi lokal menggunakan plugin `flutter_local_notifications`, serta pengujian pengiriman pesan melalui Firebase Console. Program dimodifikasi agar notifikasi dapat menampilkan data pribadi seperti Nama, NIM, Kelas, dan Prodi pengguna.