

PEMROGRAMAN PERANGKAT BERGERAK
MODUL XI
FIREBASE NOTIFIKASI



Disusun oleh:
Salman alfa rizzi
2211104056

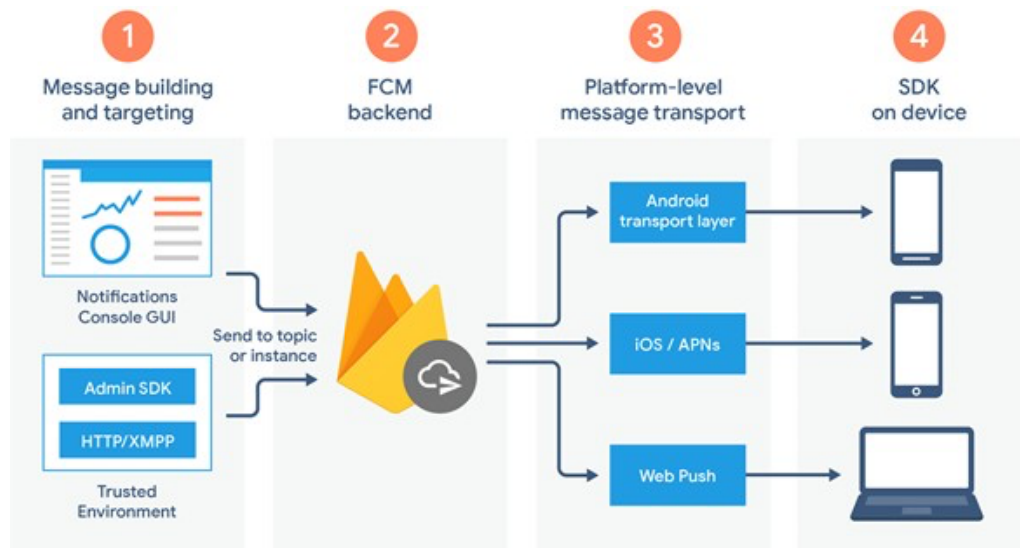
Asisten Praktikum :
Muhammad Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia
Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.
PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

Tujuan Praktikum

Mahasiswa memahami cara mengintegrasikan Firebase ke dalam project Flutter.

Mahasiswa dapat mengimplementasikan Firebase Cloud Messaging untuk menerima 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 Firebase 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.

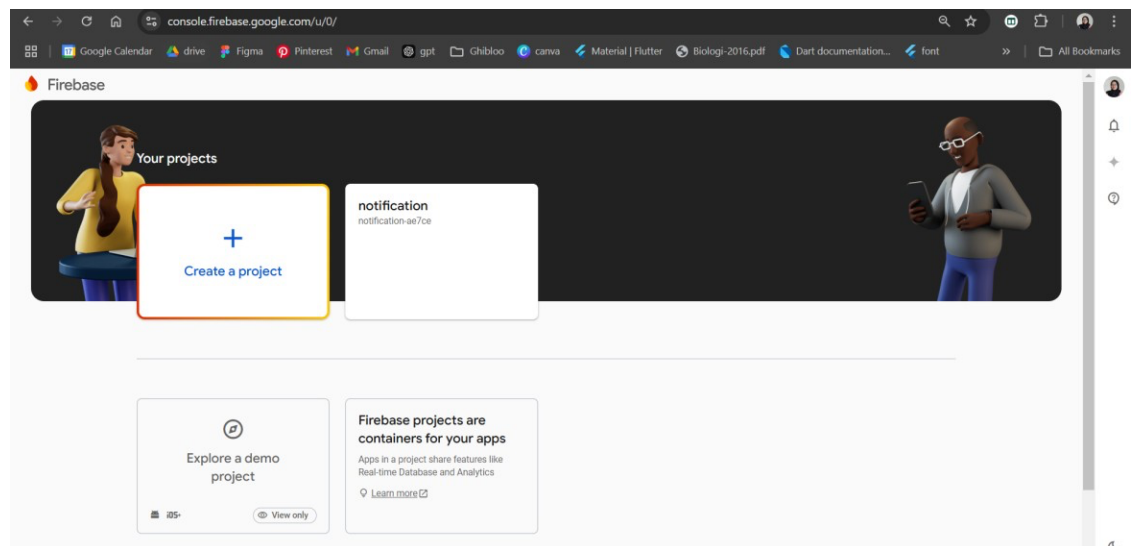
Cara kerja:

- 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

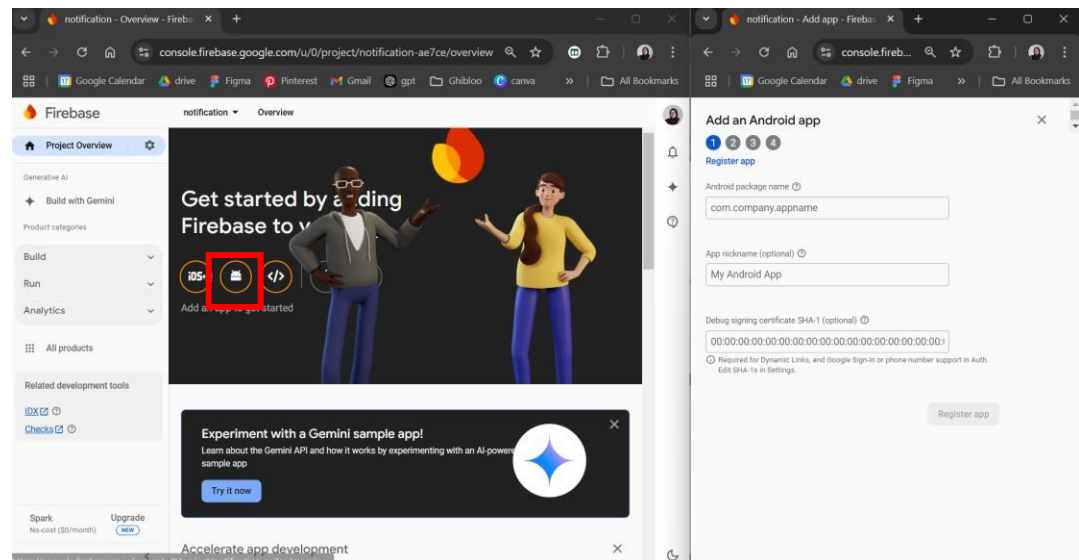
Untuk memulai, kita perlu menyiapkan akun di Firebase. Di bawah ini adalah langkah-langkah yang perlu kita ikuti untuk menyiapkan Google Firebase untuk notifikasi push di Android:

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

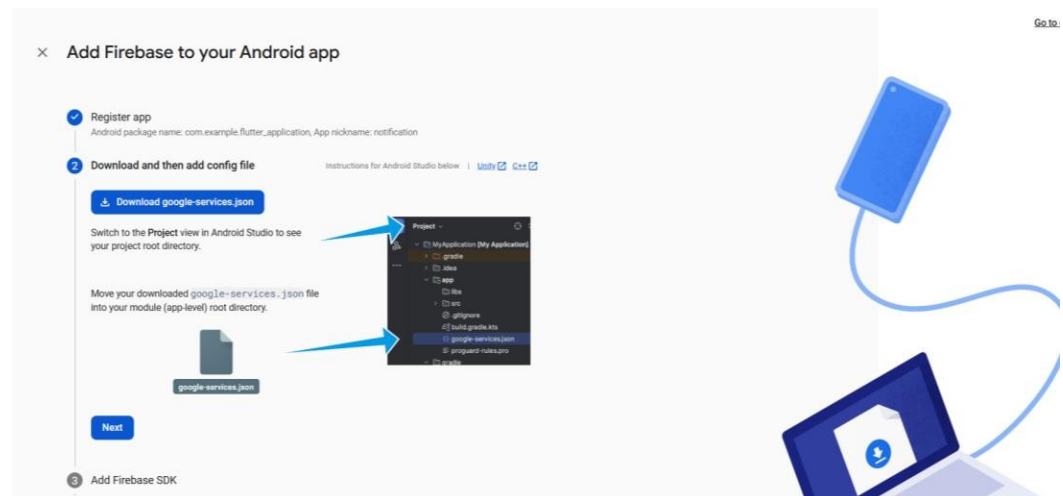


- b. Tambahkan android package name, app nickname and SHA-1 signing certificate key.
Android package name dapat ditemukan di `./android/app/build.gradle`.

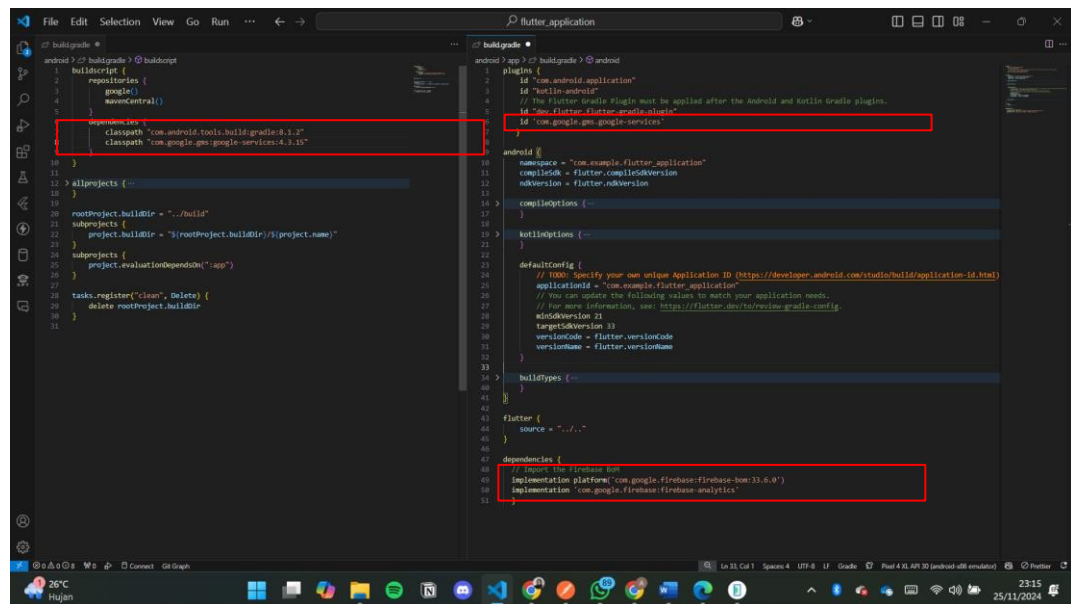
Catatan: App nickname and SHA-1 signing certificate key itu opsional.



- 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
            ),
          ),
        );
      },
    );
  }
});

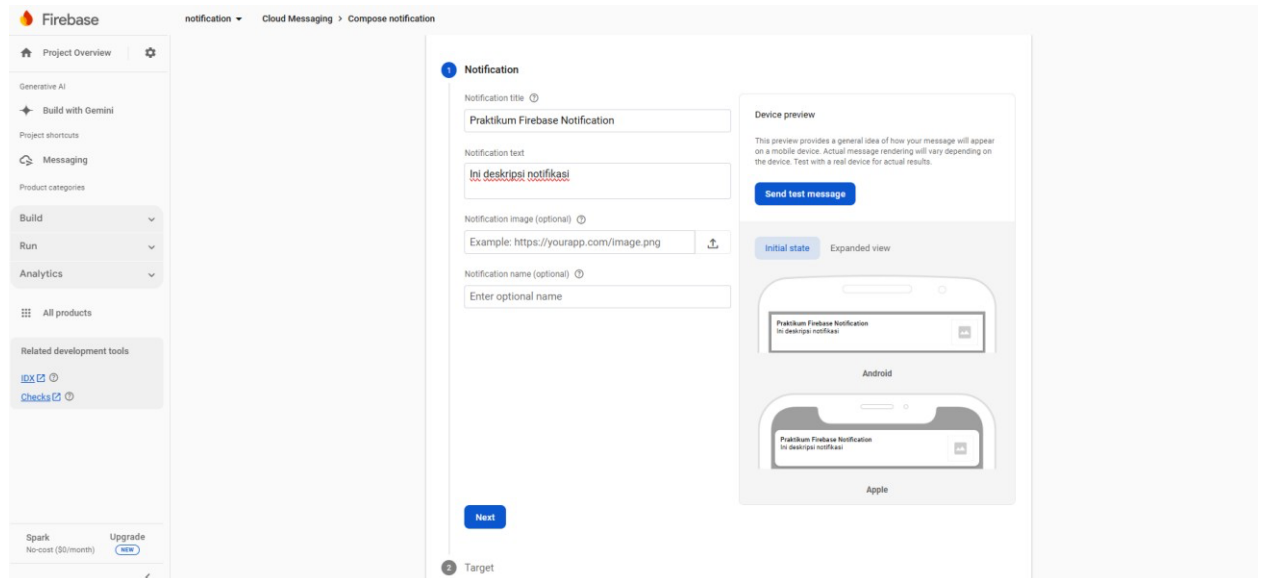
// Memanggil metode untuk mengambil token FCM perangkat
getToken();

// 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
}

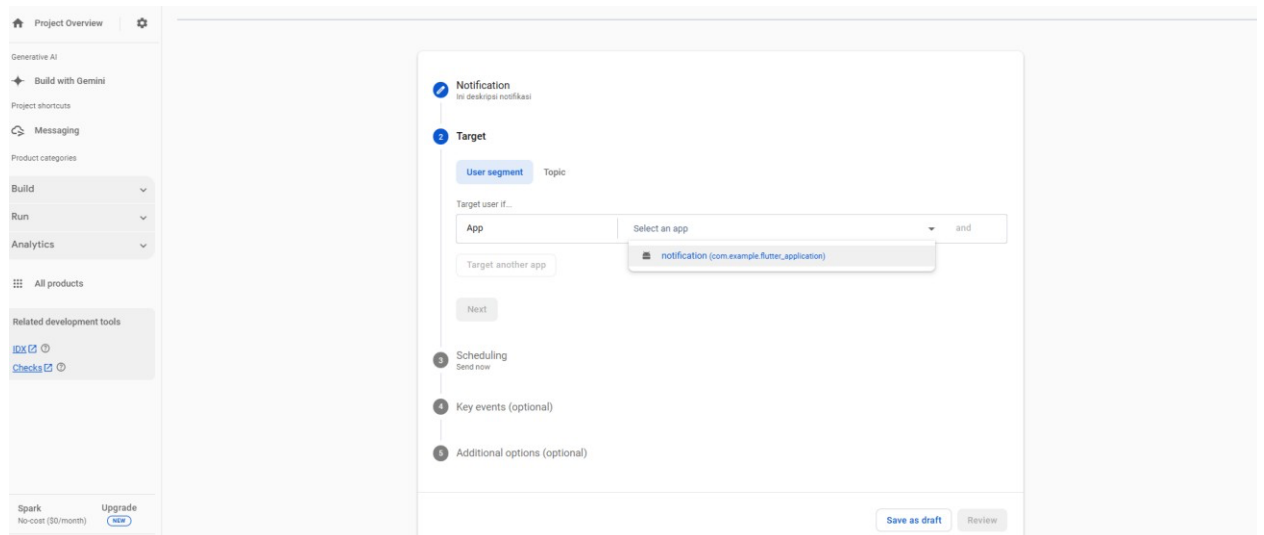
```

C. Mengirim Notifikasi Via FCM Console

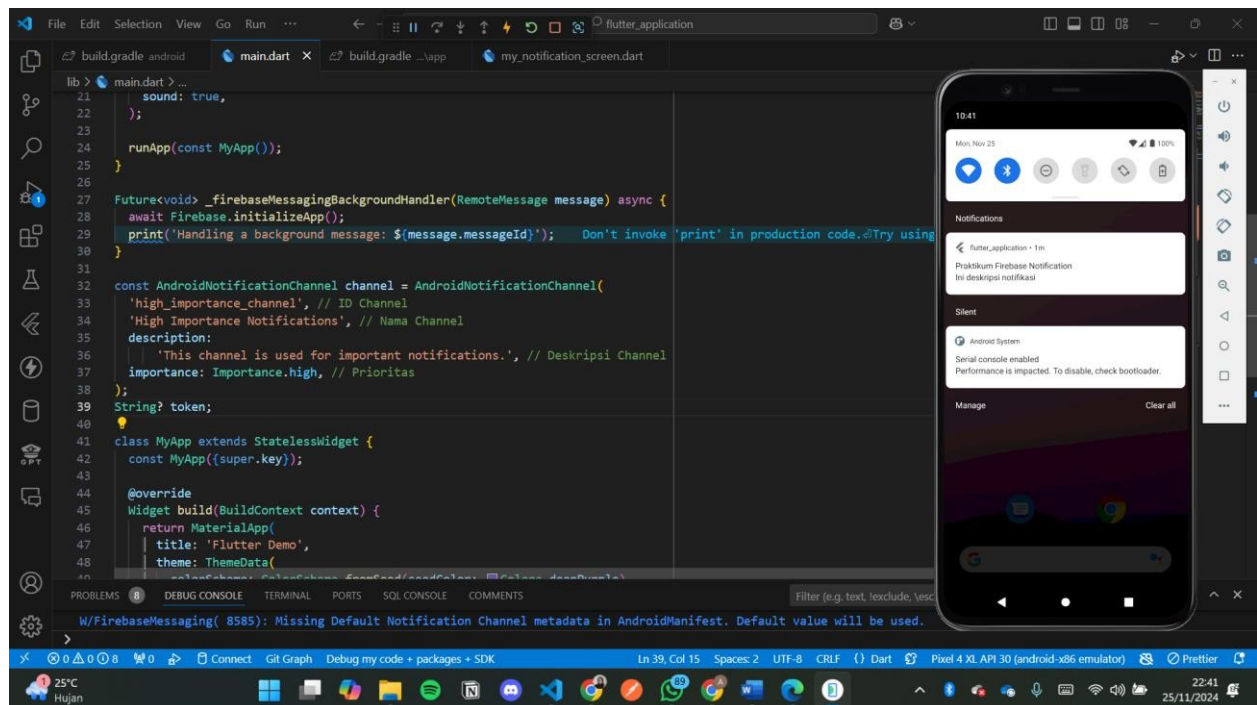
- a. Gaskeun waktunya testing notifikasi kalian. Dimulai dengan membuka halaman all product pada tab firebase --> pilih menu cloud messaging --> lalu pilih send your first message.
- b. Selanjutnya, inputkan judul, pesan, dan foto dari notifikasi kalian sesuai dengan kebutuhan.



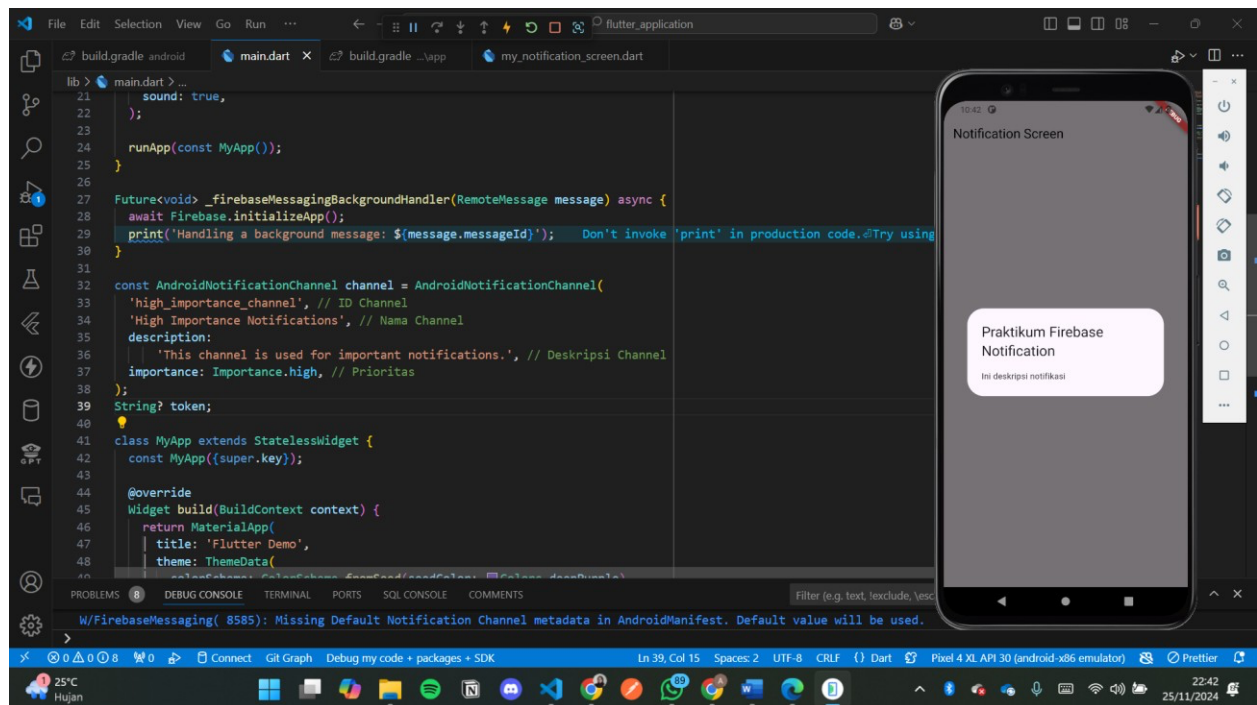
- c. Setelah itu, pilih aplikasi mana yang akan dikirimkan notifikasi dan setting waktu "Now" jika ingin langsung mengirimkan notifikasi. Lalu klik publish jika isi dan setingan notifikasi sudah benar.



- 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.

sourcecode:

main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:pertemuan11/my_notification_screen.dart';
import 'package:firebase_messaging/firebase_messaging.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
  await FlutterLocalNotificationsPlugin()
    .resolvePlatformSpecificImplementation<
      AndroidFlutterLocalNotificationsPlugin>()
    ?.createNotificationChannel(channel);
  await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
    alert: true,
    badge: true,
    sound: true,
  );
  runApp(const MyApp());
}

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
);

```

```

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a purple toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can be
        // tested with just a hot reload.
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),

```

```

        useMaterial3: true,
    ),
    home: const MyNotificationScreen(),
);
}
}

```

my_notification.dart

```

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

// 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
);

class MyNotificationScreen extends StatefulWidget {
  const MyNotificationScreen({super.key});

  @override
  State<MyNotificationScreen> createState() => _MyNotificationScreenState();
}

```

```
}
```

```
class _MyNotificationScreenState extends State<MyNotificationScreen> {  
  final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =  
    FlutterLocalNotificationsPlugin();  
  
  @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.onMessageOpenedApp.listen((RemoteMessage message) {  
      print("Notifikasi diklik: ${message.notification?.title}");  
      RemoteNotification? notification = message.notification;  
      AndroidNotification? android = message.notification?.android;  
  
      if (notification != null && android != null) {  
        showDialog(  
          context: context,  
          builder: (_) => AlertDialog(  
            title: const Text("Detail Notifikasi"),  
            content: SingleChildScrollView(  
              child: Column(  
                crossAxisAlignment: CrossAxisAlignment.start,  
                children: [  
                  Text("Judul: ${notification.title ?? 'Tidak Ada Judul'}"),  
                  Text("Isi: ${notification.body ?? 'Tidak Ada Isi'}"),  
                ],  
              ),  
            ),  
          ),  
        );  
      }  
    });  
  }  
}
```

```

const SizedBox(height: 10),
const Text("Informasi Pengirim:",
  style: TextStyle(fontWeight: FontWeight.bold)),
const Text("Nama: Dimas Cahyo Margono"),
const Text("NIM: 2211104060"),
const Text("Kelas: SE-06-02"),
const Text("Prodi: S1 Software Engineering"),
],
),
),
actions: [
  TextButton(
    onPressed: () {
      Navigator.of(context).pop();
    },
    child: const Text("Tutup"),
  ),
],
),
);
}
});

```

```

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

```

```

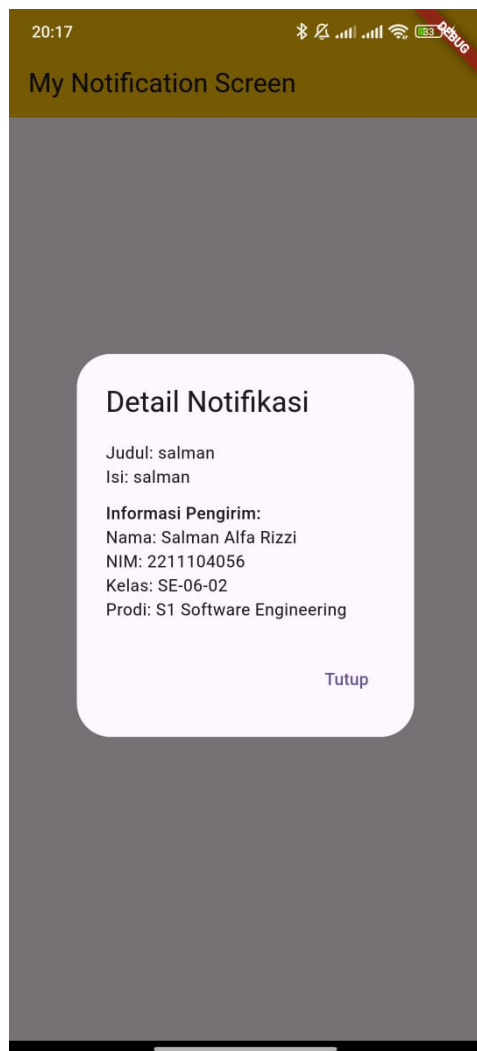
// Metode untuk mendapatkan token FCM
void getToken() async {
  token = await FirebaseMessaging.instance
    .getToken(); // Mendapatkan token FCM perangkat
  print('FCM Token: $token'); // Menampilkan token di log
}

```

@override

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: const Text("My Notification Screen"),  
      backgroundColor: Colors.amber,  
    ),  
    body: const Center(  
      child: Text("Halaman untuk menerima notifikasi"),  
    ),  
  );  
}
```

Output:



Deskripsi program:

Program di atas adalah aplikasi Flutter yang mengintegrasikan Firebase Cloud Messaging (FCM) dan notifikasi lokal untuk mengirim dan menampilkan notifikasi. Aplikasi ini menggunakan FCM untuk menerima dan mengirim notifikasi push, dengan token perangkat yang diperoleh melalui `FirebaseMessaging.instance.getToken()` untuk mempermudah pengiriman pesan. Ketika aplikasi menerima notifikasi, baik saat aktif atau dalam keadaan latar belakang, notifikasi lokal akan ditampilkan menggunakan plugin `flutter_local_notifications`. Pengguna dapat berinteraksi dengan notifikasi yang muncul, yang kemudian akan menampilkan dialog dengan detail notifikasi saat diklik. Program ini juga mengonfigurasi saluran notifikasi Android dengan prioritas tinggi untuk notifikasi penting dan menangani pesan FCM saat aplikasi berada di latar belakang dengan fungsi `_firebaseMessagingBackgroundHandler`. Selain itu, aplikasi ini mengambil token FCM perangkat untuk pengiriman pesan lebih lanjut. Secara keseluruhan, aplikasi ini bertujuan untuk memberikan pengalaman notifikasi yang interaktif dan mendukung pengiriman pesan melalui FCM.

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