

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XI
FIREBASE NOTIFIKASI**



**Disusun Oleh :
Dewi Atika Muthi / 2211104042
SE-06-02**

**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

BAB I

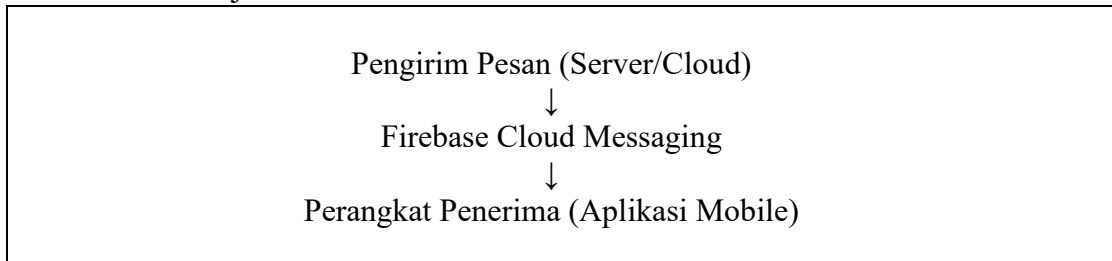
PENDAHULUAN

A. DASAR TEORI

Firebase Cloud Messaging (FCM) adalah layanan canggih dari Google yang memungkinkan pengembang mengirim notifikasi dengan mudah ke berbagai perangkat dalam jumlah besar. Konsep dasar FCM mencakup dua jenis utama pesan notifikasi:

1. Notification Message
 - Digunakan untuk menampilkan notifikasi langsung di perangkat pengguna
 - Berlaku untuk kondisi aplikasi:
 - Foreground: Pesan diteruskan ke fungsi onMessage
 - Background/Terminated: Notifikasi muncul di tray notifikasi otomatis
2. Data Message
 - Memberikan fleksibilitas lebih dalam pengolahan pesan
 - Memerlukan kode tambahan untuk menampilkan notifikasi
 - Dapat diproses kapan pun, baik aplikasi aktif maupun tidak

Ilustrasi Cara Kerja FCM



B. MAKSUD DAN TUJUAN

Praktikum Firebase Notifikasi bertujuan untuk:

- Memahami integrasi Firebase ke dalam proyek Flutter
- Mengimplementasikan Firebase Cloud Messaging (FCM)
- Mempelajari cara mengirim dan menerima notifikasi pada aplikasi mobile
- Mengembangkan kemampuan dalam menggunakan layanan cloud messaging

BAB II IMPLEMENTASI

A. PRAKTIKUM

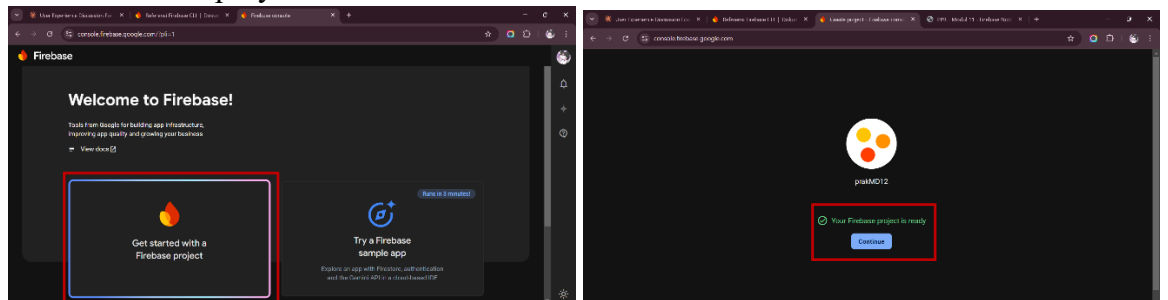
Membuat program untuk mengirimkan notifikasi melalui Firebase Claude Messaging.

Soal Studi Case

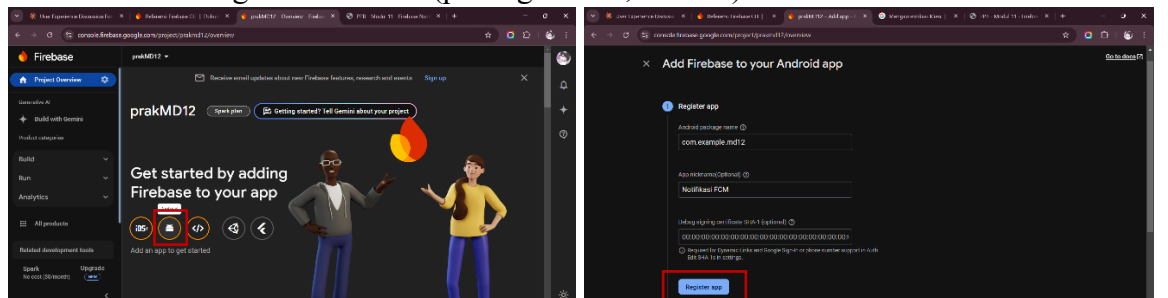
Langkah-langkah implementasi Firebase Notifikasi:

1. Persiapan Awal

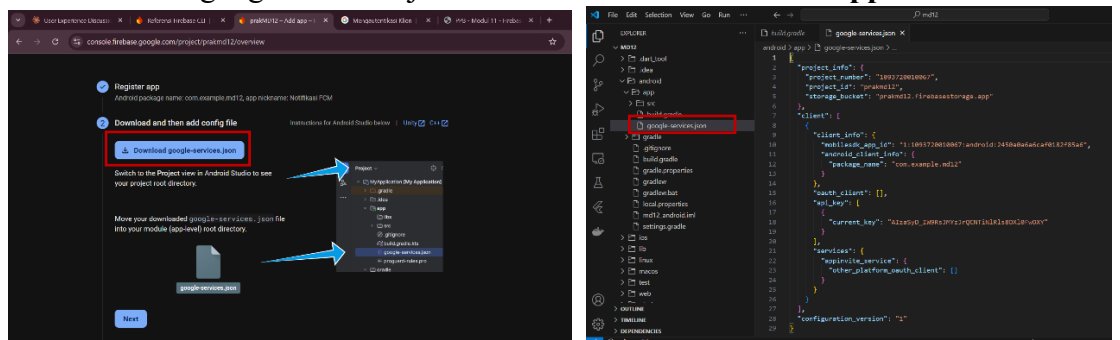
a) Daftar dan buat proyek di Firebase Console



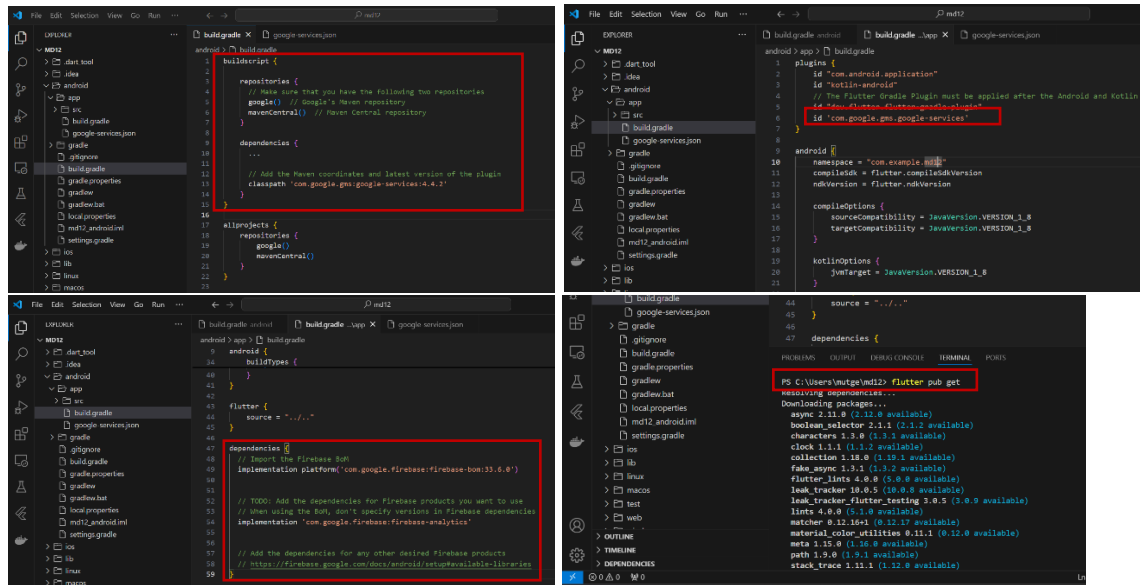
b) Tambahkan konfigurasi Android (package name, SHA-1)



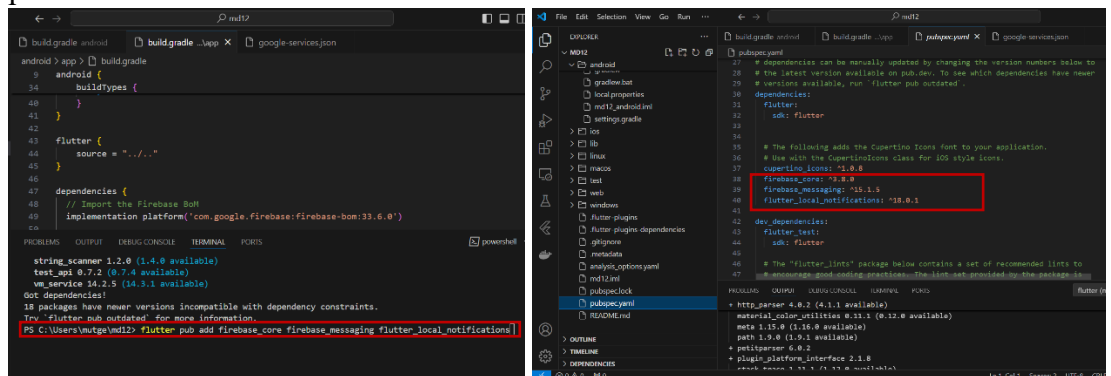
c) Download file google-services.json dan masukkan ke ./android/app/



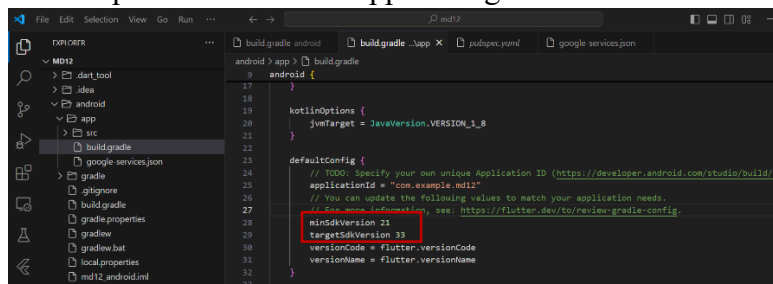
d) Tambahkan plugin dan sdk sesuai perintah pada ./android/build.gradle dan ./android/app/build.gradle lalu lakukan sync dengan mengetik “flutter pub get” pada terminal.



- e) Menambahkan 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



2. Implementasi Kode Notifikasi

SourceCode my_notification_screen.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();
    _initializeFirebase();
    _setupFirebaseMessaging();
  }

  Future<void> _initializeFirebase() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp();

    // Inisialisasi kanal notifikasi Android
    await flutterLocalNotificationsPlugin
      .resolvePlatformSpecificImplementation<
        AndroidFlutterLocalNotificationsPlugin>()
      ?.createNotificationChannel(channel);

    // Pengaturan tampilan notifikasi foreground
    await FirebaseMessaging.instance
      .setForegroundNotificationPresentationOptions(
        alert: true,
        badge: true,
        sound: true,
      );

    // Mendapatkan token FCM
    getToken();
  }

  void _setupFirebaseMessaging() {
    // Mendengarkan pesan saat aplikasi aktif
    FirebaseMessaging.onMessage.listen((RemoteMessage message) {
      RemoteNotification? notification = message.notification;
      AndroidNotification? android = message.notification?.android;

      if (notification != null && android != null) {
        flutterLocalNotificationsPlugin.show(
          notification.hashCode,
          notification.title,
          notification.body,
          NotificationDetails(
            android: AndroidNotificationDetails(
              channel.id,
              channel.name,
              channelDescription: channel.description,
              color: Colors.blue,
              icon: "@mipmap/ic_launcher",
            ),
          ),
        );
      }
    });
  }
}

```

```

// Menangani aksi notifikasi yang dibuka
FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
  RemoteNotification? notification = message.notification;
  AndroidNotification? android = message.notification?.android;

  if (notification != null && android != null) {
    showDialog(
      context: context,
      builder: (_) => AlertDialog(
        title: Text(notification.title ?? ""),
        content: SingleChildScrollView(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [Text(notification.body ?? "")],
          ),
        ),
      ),
    );
  }
});

void getToken() async {
  token = await FirebaseMessaging.instance.getToken();
  print('FCM Token: $token');
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("My Notification Screen"),
      backgroundColor: Colors.pink[200],
    ),
    body: const Center(
      child: Text("Halaman untuk menerima notifikasi"),
    ),
  );
}
}

```

SourceCode main.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';
import 'my_notification_screen.dart';

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
);

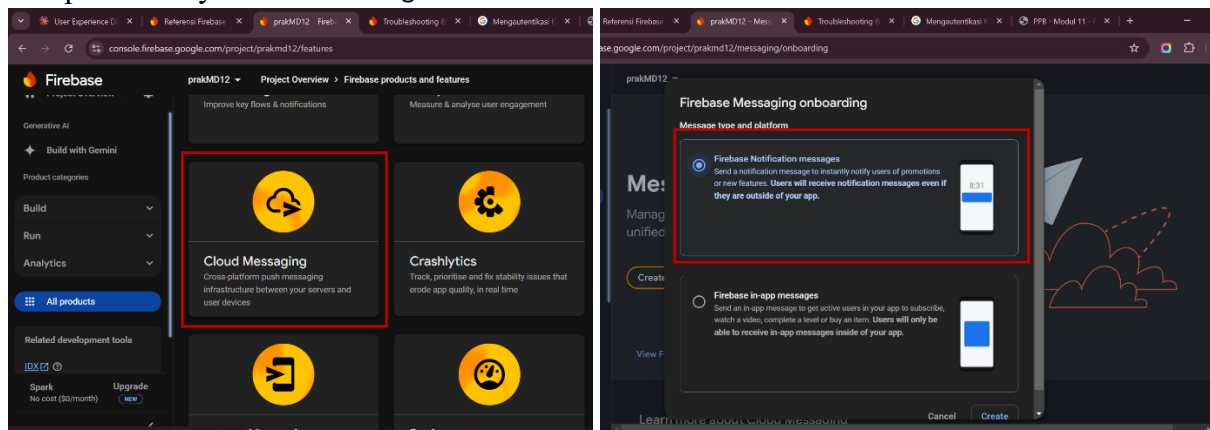
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MyNotificationScreen(),
    );
  }
}

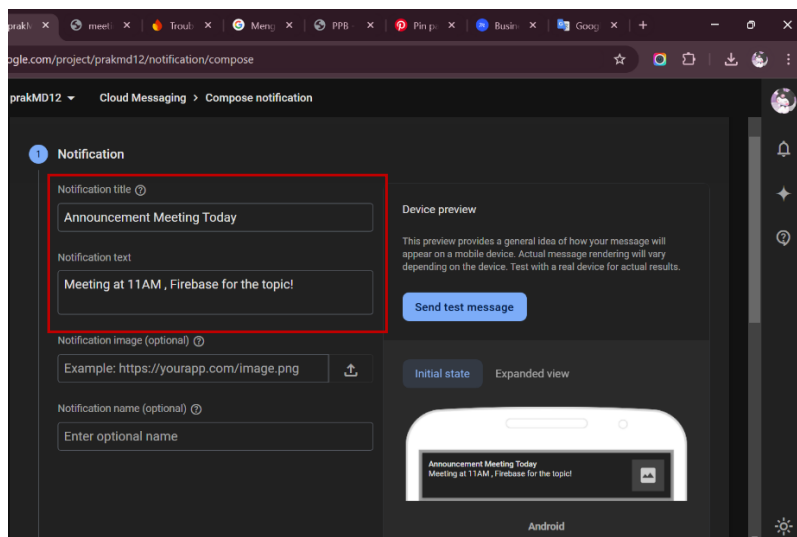
```

3. Mengirim Notifikasi Via FCM Console

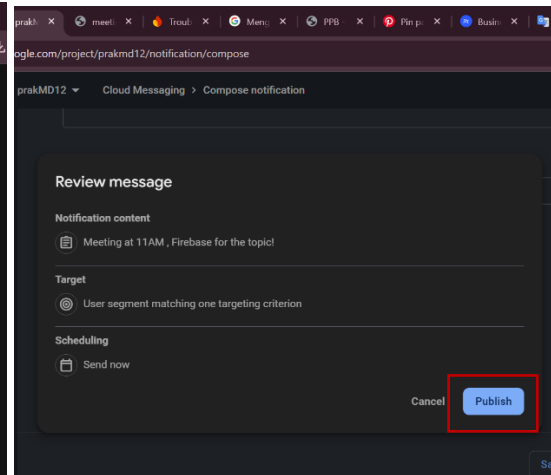
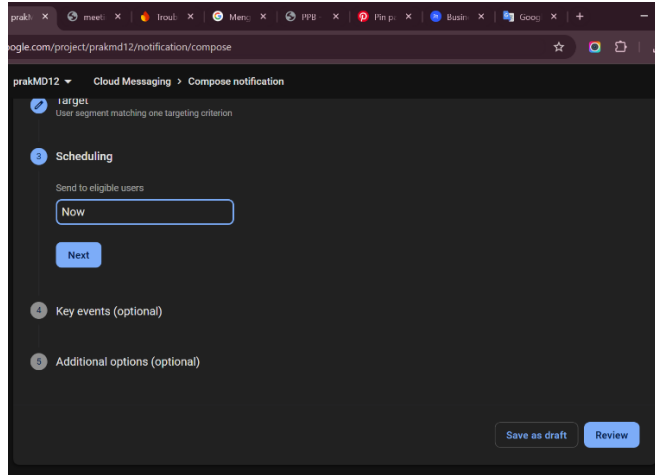
- a) Buka halaman all product pada tab firebase --> pilih menu cloud messaging --> lalu pilih send your first message



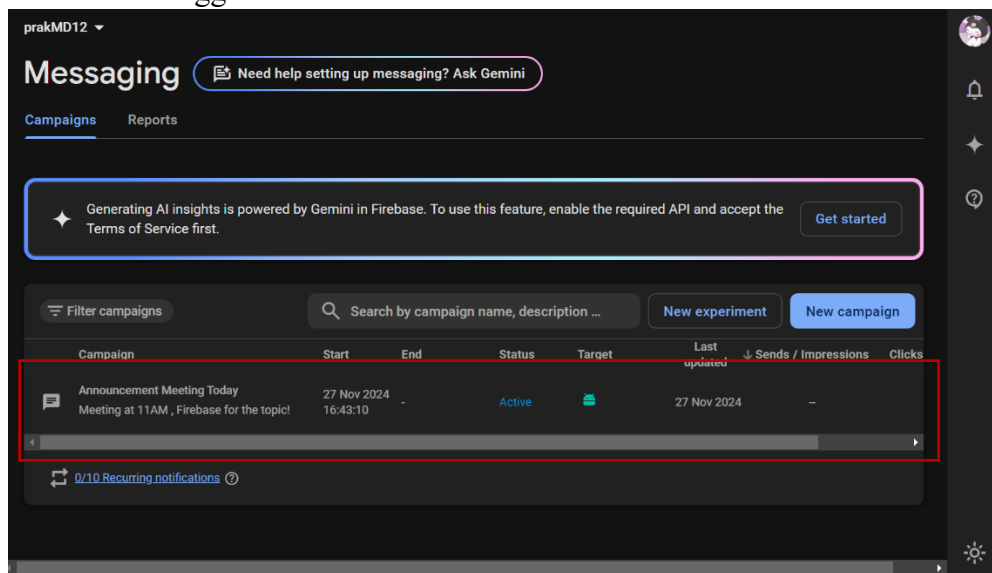
- b) Selanjutnya, inputkan judul, pesan, dan foto dari notifikasi kalian sesuai dengan kebutuhan.



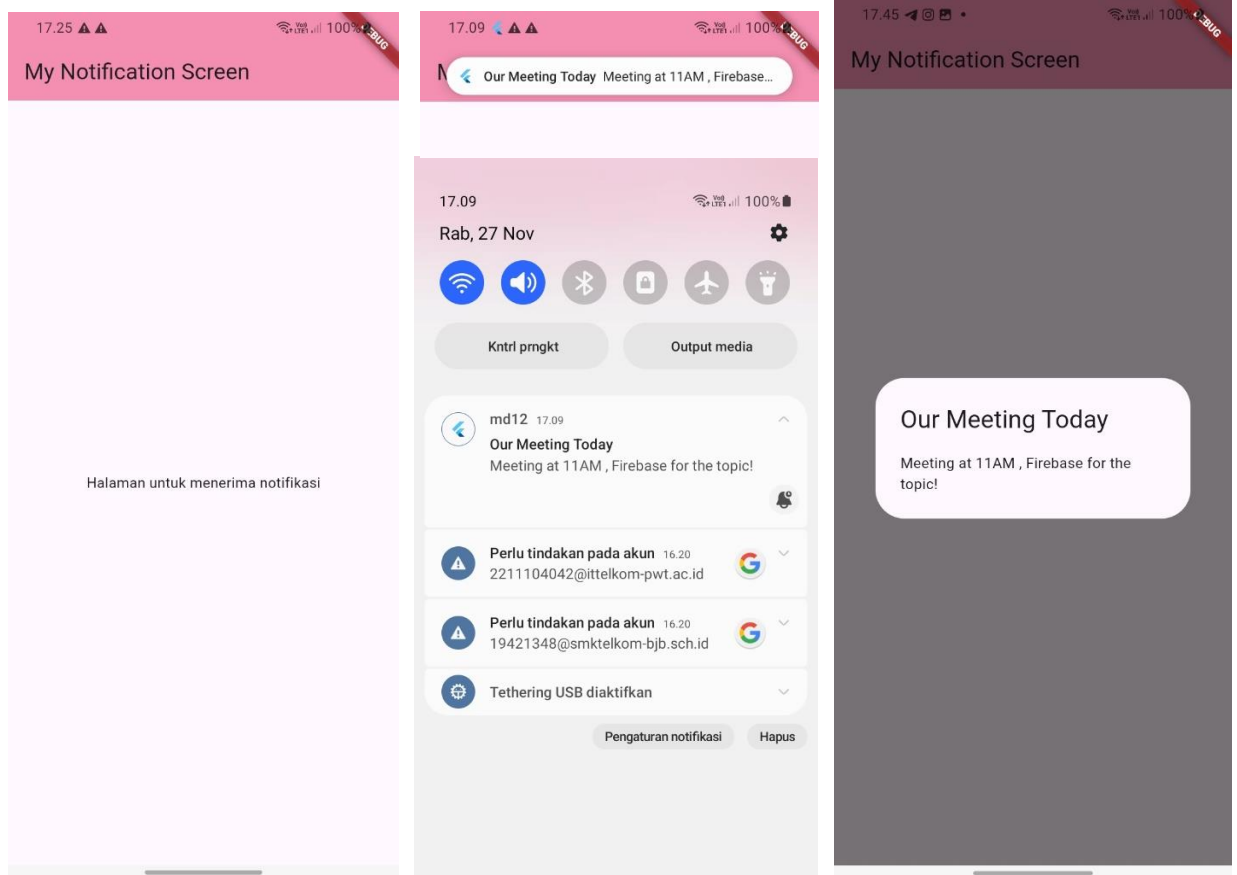
- c) Setelah itu, pilih aplikasi mana yang akan dikirimkan notifikasi dan setting waktu "Now" jika ingin langsung mengirimkan notifikasi. Lalu klik publish jika isi dan setingan notifikasi sudah benar.



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



Screenshoot Output



Deskripsi Program

Program ini adalah aplikasi Flutter yang mengimplementasikan sistem notifikasi push menggunakan Firebase Cloud Messaging (FCM). Intinya terdiri dari beberapa komponen utama:

a) Konfigurasi Awal Firebase:

- Pada fungsi main(), Firebase diinisialisasi
- Diatur cara menangani pesan background
- Dibuat saluran notifikasi untuk Android
- Diatur opsi presentasi notifikasi foreground

b) Manajemen Token FCM:

- Metode getToken() digunakan untuk mendapatkan unique token perangkat
- Token ini diperlukan untuk mengirim notifikasi ke perangkat spesifik

c) Penanganan Notifikasi:

- FirebaseMessaging.onMessage.listen(): Menangani notifikasi saat aplikasi sedang aktif
- Menampilkan notifikasi menggunakan FlutterLocalNotificationsPlugin
- FirebaseMessaging.onMessageOpenedApp.listen(): Menangani aksi saat notifikasi dibuka
- Menampilkan dialog dengan detail notifikasi

d) Saluran Notifikasi:

- Dibuat `AndroidNotificationChannel` untuk mengatur prioritas dan karakteristik notifikasi
- Saluran bernama `'high_importance_channel'` dengan prioritas tinggi

Singkatnya, program ini untuk:

- Menerima notifikasi push dari Firebase
- Menampilkan notifikasi saat aplikasi aktif
- Menangani notifikasi yang dibuka pengguna
- Mendapatkan token unik perangkat untuk komunikasi

B. UNGUIDED (Tugas Mandiri)

Soal Studi Case

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

Sourcecode `my_notification_screen.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();
    _initializeFirebase();
    _setupFirebaseMessaging();
  }

  Future<void> _initializeFirebase() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp();

    // Inisialisasi kanal notifikasi Android
    await flutterLocalNotificationsPlugin
      .resolvePlatformSpecificImplementation<
        AndroidFlutterLocalNotificationsPlugin>()

```

```

        ?.createNotificationChannel(channel);

// Pengaturan tampilan notifikasi foreground
await FirebaseMessaging.instance
    .setForegroundNotificationPresentationOptions(
        alert: true,
        badge: true,
        sound: true,
    );

// Mendapatkan token FCM
getToken();
}

void _setupFirebaseMessaging() {
    // Mendengarkan pesan saat aplikasi aktif
    FirebaseMessaging.onMessage.listen((RemoteMessage message) {
        RemoteNotification? notification = message.notification;
        AndroidNotification? android = message.notification?.android;

        if (notification != null && android != null) {
            flutterLocalNotificationsPlugin.show(
                notification.hashCode,
                notification.title,
                notification.body,
                NotificationDetails(
                    android: AndroidNotificationDetails(
                        channel.id,
                        channel.name,
                        channelDescription: channel.description,
                        color: Colors.blue,
                        icon: "@mipmap/ic_launcher",
                    ),
                ),
            );
        }
    });

// Menangani aksi notifikasi yang dibuka
FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
    RemoteNotification? notification = message.notification;
    AndroidNotification? android = message.notification?.android;

    if (notification != null && android != null) {
        showDialog(
            context: context,
            builder: (_) => AlertDialog(
                title: Text(notification.title ?? ""),
                content: SingleChildScrollView(
                    child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [Text(notification.body ?? "")],
                    ),
                ),
            ),
        );
    }
});
}

void getToken() async {
    token = await FirebaseMessaging.instance.getToken();
    print('FCM Token: $token');
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text("My Notification Screen"),
            backgroundColor: Colors.pink[200],
        ),
        body: const Center(
            child: Text("Halaman untuk menerima notifikasi"),
        ),
    );
}

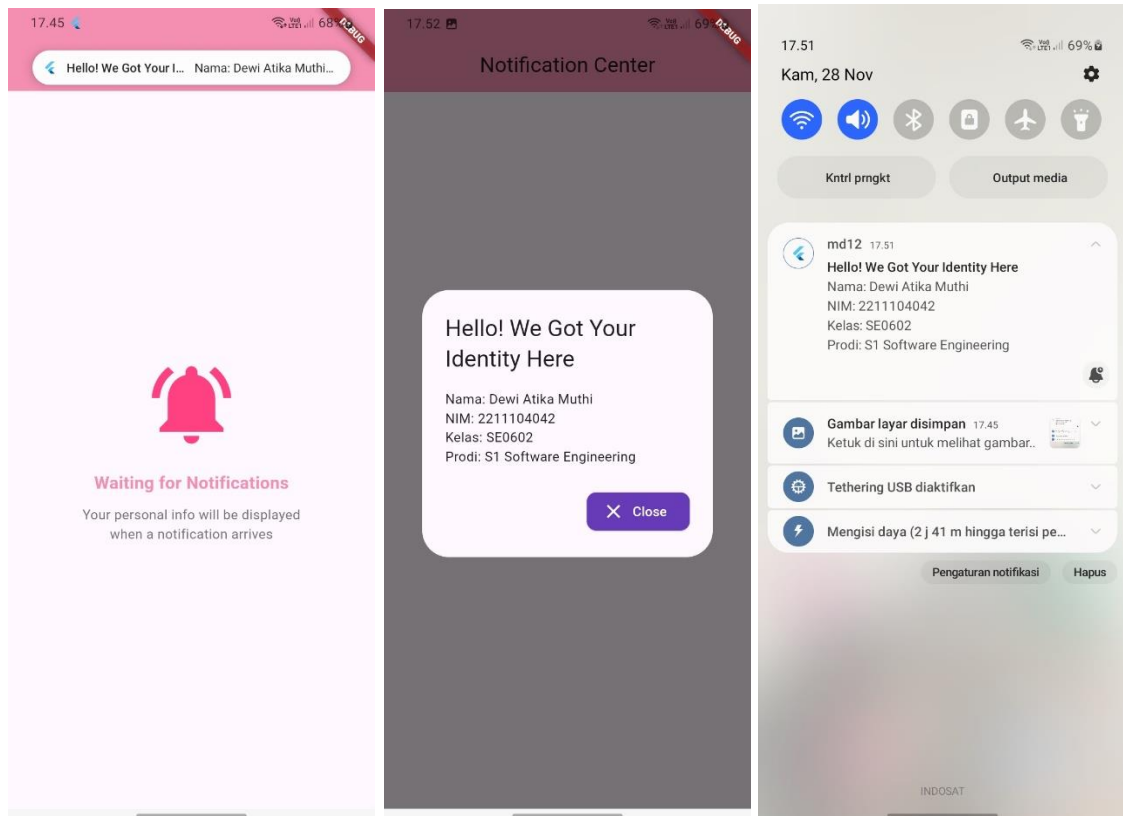
```

```
}  
}
```

Sourcecode main.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';  
import 'my_notification_screen.dart';  
  
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(  
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),  
        useMaterial3: true,  
      ),  
      home: const MyNotificationScreen(),  
    );  
  }  
}
```

Screenshoot Output



Deskripsi Program

Program ini bertujuan untuk mengimplementasikan fitur notifikasi Firebase Cloud Messaging (FCM) di aplikasi Flutter. Menggunakan beberapa package utama, seperti `firebase_core` untuk inisialisasi Firebase, `firebase_messaging` untuk menangani pesan notifikasi dari FCM, dan `flutter_local_notifications` untuk menampilkan notifikasi lokal pada perangkat, program ini mencakup berbagai skenario penerimaan notifikasi.

Pada awalnya, aplikasi mendefinisikan variabel global token, yang digunakan untuk menyimpan FCM token sebagai identitas unik perangkat. Program juga mencakup fungsi `_firebaseMessagingBackgroundHandler` untuk menangani notifikasi yang diterima ketika aplikasi berada di latar belakang.

Fungsi `_initializeFirebase` menginisialisasi Firebase, membuat kanal notifikasi, dan mengatur tampilan notifikasi ketika aplikasi sedang aktif (foreground). Kemudian, metode `_setupFirebaseMessaging` mendengarkan notifikasi masuk. Saat aplikasi berada di foreground, notifikasi ditampilkan menggunakan `flutterLocalNotificationsPlugin.show`. Jika notifikasi diterima saat aplikasi berada di latar belakang atau dibuka dari tray notifikasi, aplikasi akan menampilkan dialog menggunakan `showDialog`. Dialog ini memuat judul dan isi notifikasi serta menyediakan opsi `Cancel` untuk menutup dialog tanpa tindakan lebih lanjut.

Program ini juga mencakup fungsi `getToken`, yang mengambil dan mencetak token perangkat untuk digunakan dalam mengirim notifikasi.

BAB III

KESIMPULAN DAN SARAN

A. KESIMPULAN

Firebase Cloud Messaging (FCM) merupakan teknologi penting dalam pengembangan aplikasi mobile yang memungkinkan pengiriman notifikasi secara efisien. Melalui praktikum ini, kami memahami bahwa FCM tidak hanya sekadar mengirim pesan, tetapi juga memberikan fleksibilitas dalam penanganan notifikasi pada berbagai kondisi aplikasi. Konsep notification message dan data message memberikan kemampuan berbeda dalam mengelola komunikasi dengan pengguna. Integrasi FCM membutuhkan konfigurasi bertahap mulai dari pendaftaran proyek, pengaturan plugin, hingga implementasi kode handler. Praktikum ini juga mengajarkan pentingnya memahami alur kerja dan metode pengiriman notifikasi dalam konteks pengembangan aplikasi mobile modern.

B. REFERENSI

Dokumentasi Resmi Firebase: <https://firebase.google.com/docs/cloud-messaging>
Flutter Firebase Documentation