

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK  
MODUL XI  
Data Storage (Bagian 2)**



**Disusun Oleh :**

**Dwi Candra Pratama / 2211104035**

**SE06-02**

**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**

## **TUJUAN PRAKTIKUM**

1. Mahasiswa memahami cara mengintegrasikan Firebase ke dalam project Flutter.
2. Mahasiswa dapat mengimplementasikan Firebase Cloud Messaging untuk menerima notifikasi.

### **FIREBASE NOTIFIKASI**

#### **A. Firebase Cloud Messaging**

Firebase 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.

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 Firebase Cloud Functions.

#### **B. 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:

- a) Jika aplikasi berjalan di latar depan (foreground), pesan diteruskan ke fungsi `onMessage` untuk diproses.
- b) 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.

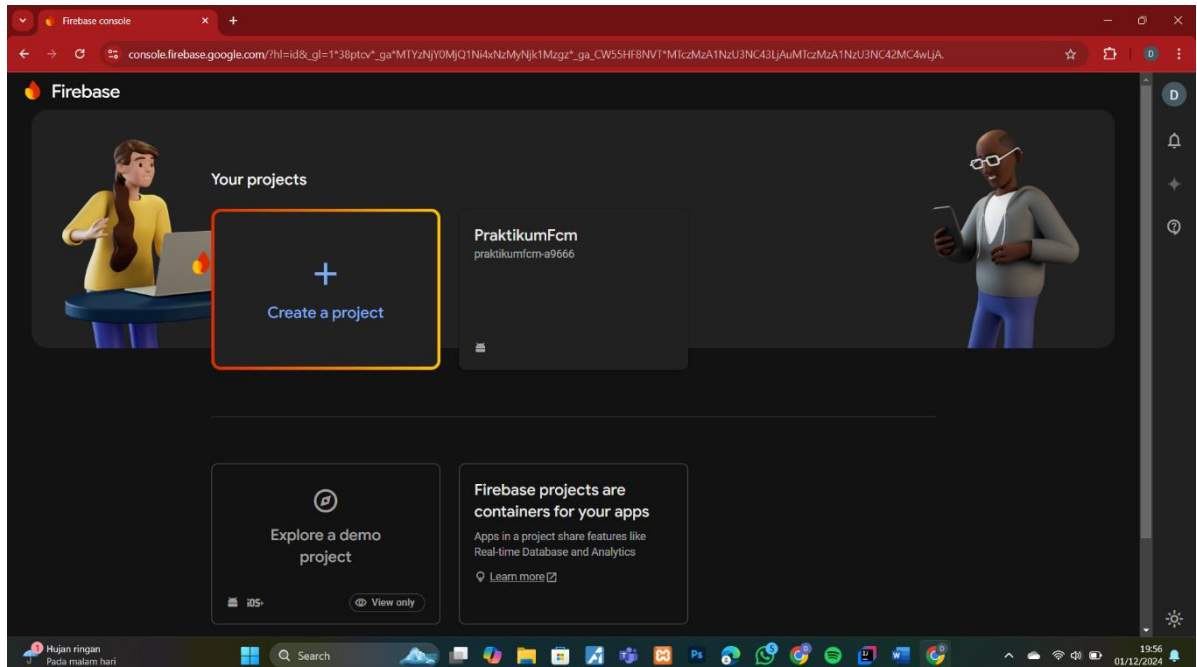
Cara kerja:

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

## C. Integrasi Aplikasi dengan Firebase

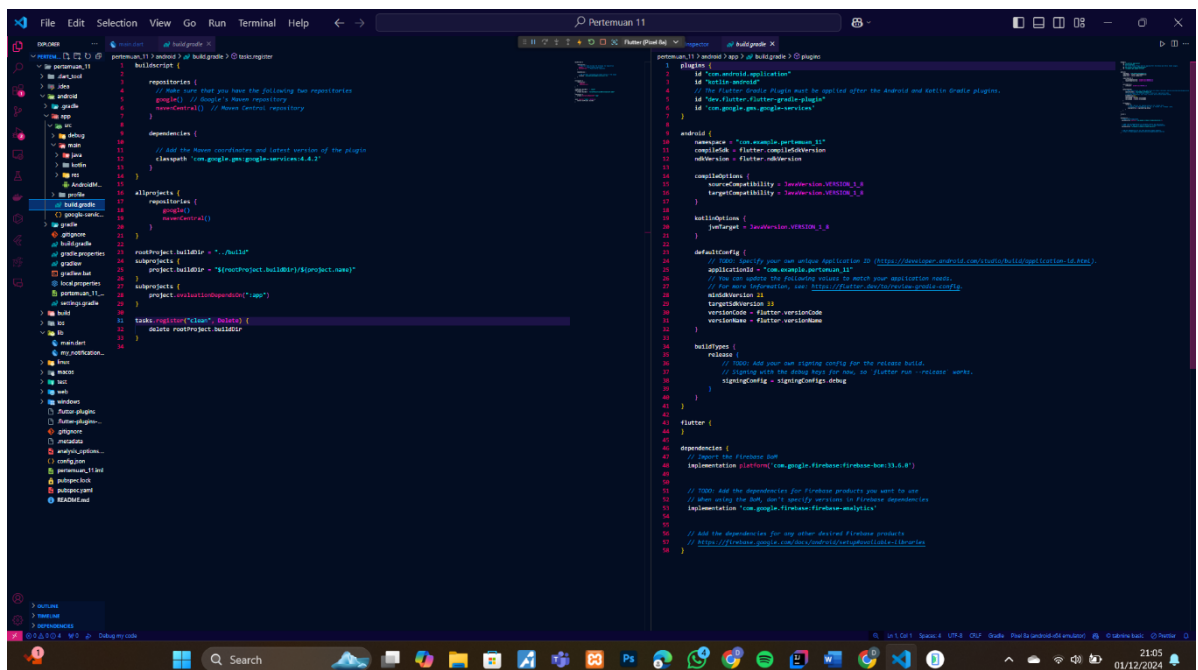
Pertama kita perlu menyiapkan akun di Firebase. Di bawah ini adalah langkah-langkah yang perlu kita ikuti untuk menyiapkan Google Firebase untuk notifikasi push di Android:

1. Daftar dan login pada firebase console lalu buat sebuah proyek baru.



Karena saya sudah buat jadi saya akan tampil seperti itu dan klik pada PraktikumFcm

2. 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.



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
```

Bisa dilihat pada line 28 dan 29

#### D. Membuat Notifikasi Handler

- Pertama, buat kode di fungsi main() menjadi seperti ini, untuk memastikan Firebase dan konfigurasi lainnya sudah siap digunakan

```
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 'package:pertemuan_11/my_notifications_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',
      debugShowCheckedModeBanner: false,
      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: MyNotificationsScreen(),
    );
  }
}

```

- kedua, buat kode di fungsi My\_Notifications\_Screen() menjadi seperti ini, agar bisa Membuat pengaturan inisialisasi notifikasi untuk Android, Mendengarkan pesan saat aplikasi aktif, Jika notifikasi tersedia bisa tampil menggunakan notifikasi lokal, Menangani aksi ketika notifikasi dibuka, Jika notifikasi tersedia, tampilkan dialog, dan Metode untuk mendapatkan token FCM. Berikut kodenya:

```
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:pertemuan_11/main.dart';

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

  @override
  State<MyNotificationsScreen> createState() =>
    _MyNotificationsScreenState();
}

class _MyNotificationsScreenState extends State<MyNotificationsScreen> {
  @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.pink,
        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();
}

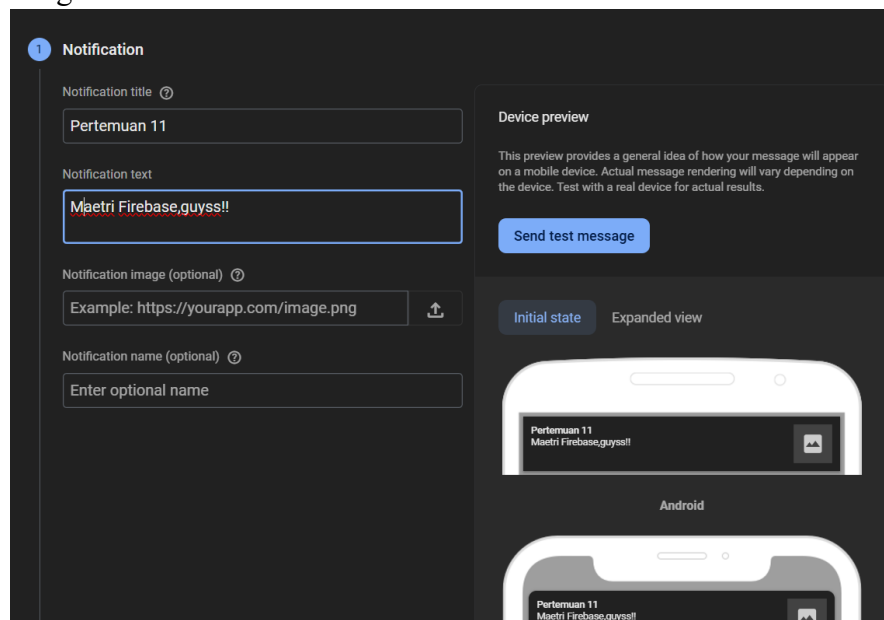
```

```
// 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
}

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

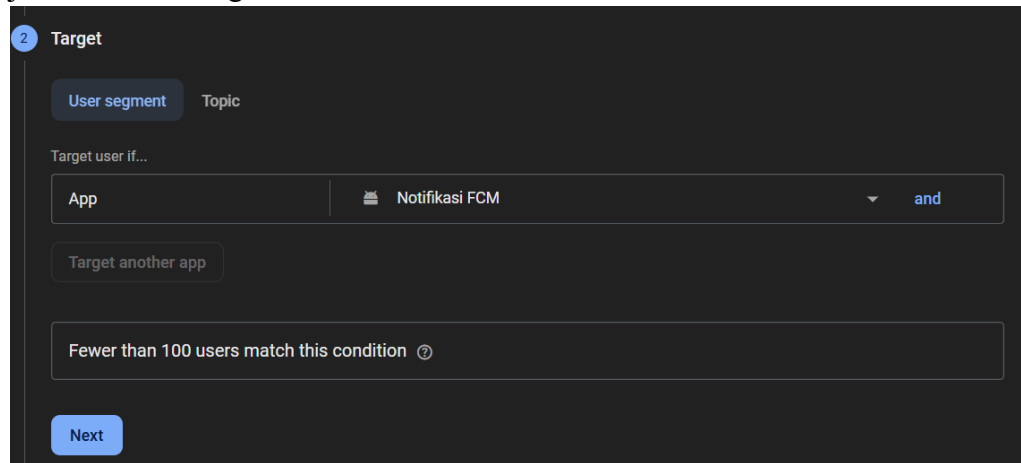
## E. Mengirim Notifikasi Via FCM Console

1. Selanjutnya, inputkan judul, pesan, dan foto dari notifikasi kalian sesuai dengan kebutuhan.

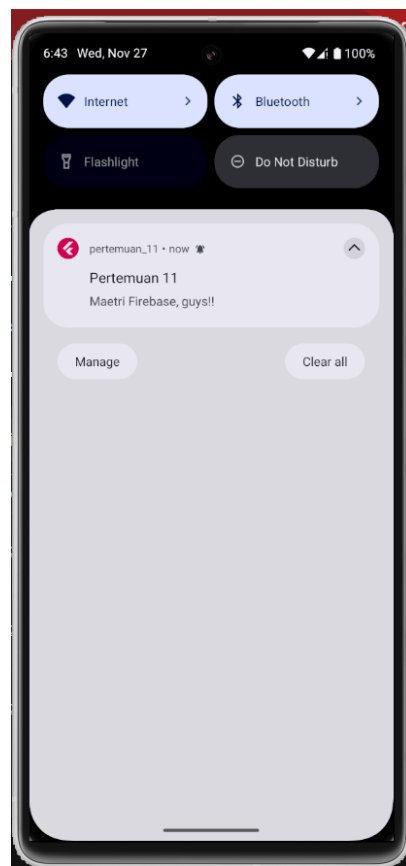




- Setelah itu, pilih aplikasi mana yang akan dikirimkan notifikasi dan setting waktu "Now" jika ingin langsung mengirimkan notifikasi. Lalu klik publish jika isi dan setingan notifikasi sudah benar.



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



## UNGUIDED

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

Source Codenya:

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 'package:pertemuan_11/my_notifications_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',
      debugShowCheckedModeBanner: false,
      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: MyNotificationsScreen(),
    );
  }
}

```

#### MyNotificationsScreen.dart

```

import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:pertemuan_11/main.dart';

```

```

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

  @override
  State<MyNotificationsScreen> createState() => _MyNotificationsScreenState();
}

class _MyNotificationsScreenState extends State<MyNotificationsScreen> {
  @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.pink,
              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
}

```

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("My Notifications"),

backgroundColor: Colors.pink,

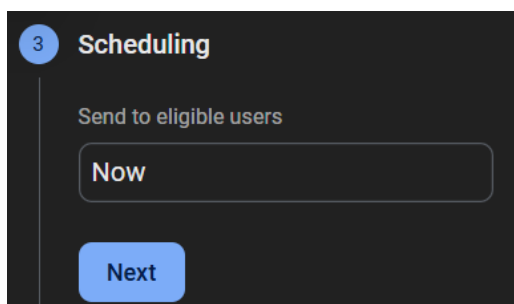
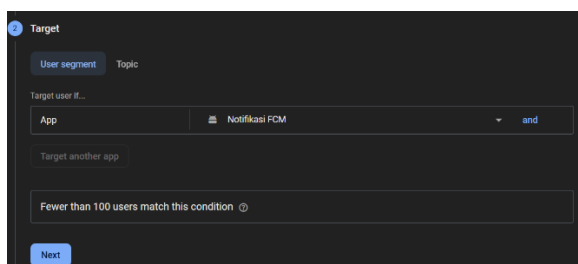
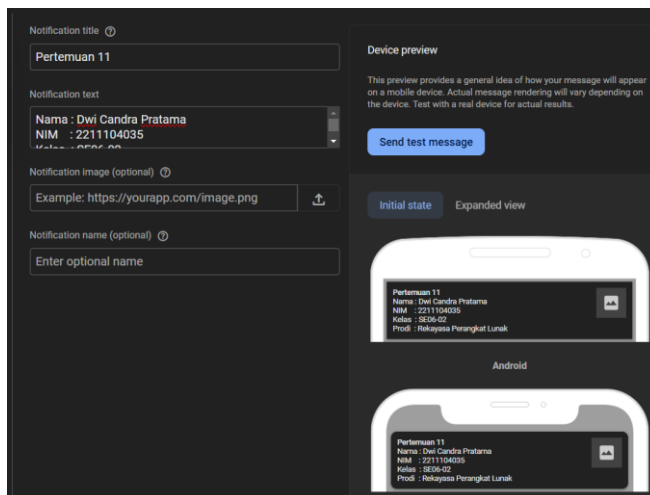
),

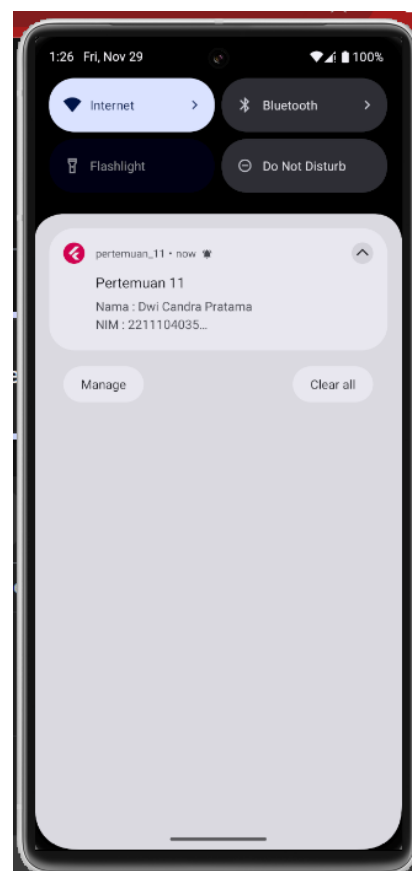
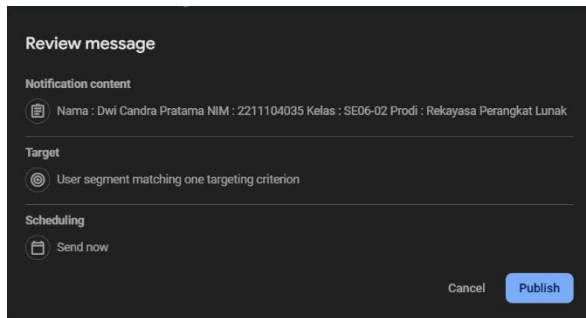
body: const Center(child: Text('Halaman untuk menerima notifikasi')),

);

}

}





Program ini adalah aplikasi Flutter yang mengintegrasikan Firebase Cloud Messaging (FCM) dan notifikasi lokal. Aplikasi ini memungkinkan pengguna menerima notifikasi penting, baik saat aplikasi aktif maupun di latar belakang. Notifikasi yang diterima saat aplikasi aktif ditampilkan menggunakan notifikasi lokal, sementara notifikasi di latar belakang dapat memunculkan dialog interaktif saat aplikasi dibuka melalui notifikasi tersebut. Fitur utama aplikasi ini meliputi pengelolaan kanal notifikasi dengan prioritas tinggi, penanganan notifikasi real-time, dan kemampuan untuk mendapatkan token unik perangkat yang dapat digunakan untuk mengirim pesan spesifik melalui FCM. Dengan antarmuka yang sederhana, aplikasi ini mempermudah pengguna untuk tetap terhubung dengan pemberitahuan penting secara andal dan responsif.