

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XI
FIREBASE NOTIFICATION**



Disusun Oleh :

Zivana Afra Yulianto / 2211104039

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

Firestore Cloud Messaging (FCM) adalah layanan yang memungkinkan pengiriman pesan atau notifikasi secara real-time ke perangkat pengguna. Flutter Local Notifications digunakan untuk menampilkan notifikasi lokal pada perangkat, bahkan saat aplikasi berjalan di latar depan. Integrasi keduanya memungkinkan pengiriman notifikasi yang fleksibel, informatif, dan responsif.

B. MAKSUD DAN TUJUAN

Maksud:

Membuat aplikasi yang mengintegrasikan FCM dan Flutter Local Notifications untuk mengirim dan menampilkan notifikasi real-time.

Tujuan:

1. Menampilkan notifikasi real-time kepada pengguna.
2. Menyediakan halaman detail untuk menampilkan informasi notifikasi.
3. Meningkatkan pengalaman pengguna dengan antarmuka sederhana dan intuitif.

BAB II IMPLEMENTASI

A. GUIDED

Code 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:praktikum/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: MyNotificationScreen(),
    );
  }
}
```

Code my_notification_screen.dart :

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

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

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

class _MyNotificationScreenState extends State<MyNotificationScreen> {
  @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);

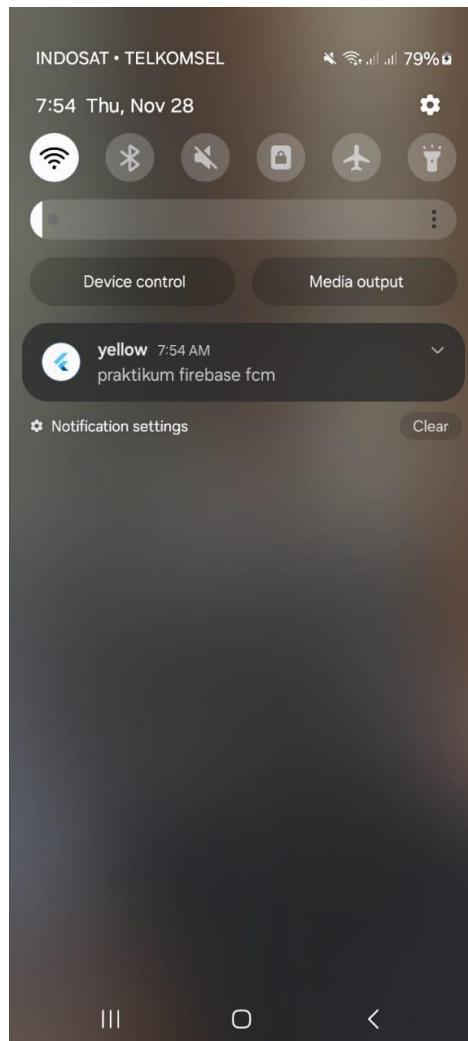
    // Mendengarkan pesan saat aplikasi aktif
    FirebaseMessaging.onMessage.listen((RemoteMessage message) {
      RemoteNotification? notification = message.notification;
      AndroidNotification? android = message.notification?.android;
      // Jika notifikasi tersedia, tampilkan menggunakan notifikasi 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
}

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

Screenshoot Output :



(notification)

Deskripsi :

Kode ini mengimplementasikan notifikasi di Flutter menggunakan Firebase Cloud Messaging (FCM) dan flutter_local_notifications.

1. Fungsi Utama:
 - Menerima Notifikasi: Mendukung notifikasi saat aplikasi aktif, di latar belakang, atau tertutup.
 - Notifikasi Lokal: Menampilkan pesan menggunakan kanal prioritas tinggi di Android.
2. Struktur:
 - `main.dart`: Inisialisasi Firebase, konfigurasi kanal notifikasi, dan pengaturan notifikasi latar depan.
 - `my_notification_screen.dart`:
 - Mendengarkan dan menampilkan notifikasi.
 - Menangani aksi klik notifikasi.
 - Mengambil token FCM perangkat.
3. Tujuan: Memberikan pengalaman real-time notifikasi untuk aplikasi berbasis Flutter.

B. UNGUIDED

Code 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:praktikum/my_notification_screen.dart';
import 'package:praktikum/notif_landing_page.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});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      initialRoute: '/',
      routes: {
        '/': (context) => const MyNotificationScreen(),
        '/landing': (context) => const NotifLandingPage(),
      },
    );
  }
}
```

Code my_notification_screen.dart :

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

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();

    // Inisialisasi pengaturan notifikasi lokal
    var initializationSettingsAndroid =
      const AndroidInitializationSettings('@mipmap/ic_launcher');
    var initializationSettings =
      InitializationSettings(android: initializationSettingsAndroid);

    flutterLocalNotificationsPlugin.initialize(
      initializationSettings,
      onDidReceiveNotificationResponse: (NotificationResponse response) {
        if (response.payload != null) {
          Navigator.pushNamed(context, '/landing', arguments: response.payload);
        }
      },
    );

    // 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', // Sesuaikan ID channel
              'Default Channel',
              channelDescription: 'Channel untuk notifikasi',
              color: Colors.blue,
              icon: "@mipmap/ic_launcher",
              importance: Importance.max,
              priority: Priority.high,
            ),
          ),
          payload: notification.body,
        );
      }
    });

    // Menangani klik notifikasi saat aplikasi dibuka
    FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
      Navigator.pushNamed(context, '/landing',
        arguments: message.notification?.body);
    });

    getToken();
  }
}
```

:


```

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

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Notifikasi Saya'),
      backgroundColor: Colors.amber,
    ),
    body: Center(
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Icon(Icons.notifications, size: 100, color: Colors.amber[700]),
            const SizedBox(height: 20),
            const Text(
              'Halaman Menunggu Notifikasi',
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.w500),
            ),
            const SizedBox(height: 10),
          ],
        ),
      ),
    ),
  );
}

```

Code notif_landing_page.dart :

```

import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    final String? data = ModalRoute.of(context)?.settings.arguments as String?;

    return Scaffold(
      appBar: AppBar(
        title: const Text('Detail Notifikasi'),
        backgroundColor: Colors.amber,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Card(
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(15.0),
          ),
          elevation: 4,
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              mainAxisAlignment: MainAxisAlignment.min,
              children: [
                const Text(
                  'Pesanan Notifikasi',
                  style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
                ),
                const Divider(),
                const SizedBox(height: 10),
              ],
            ),
          ),
        ),
      ),
    );
  }
}

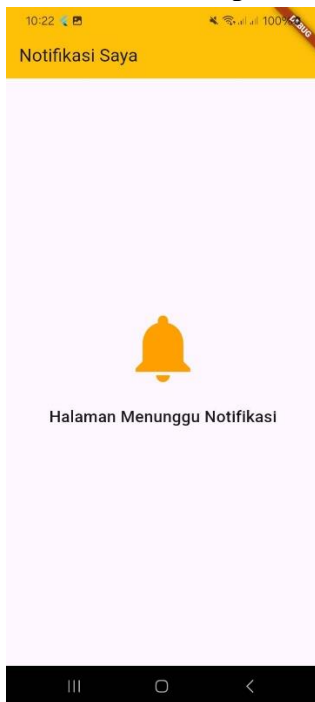
```

```

const SizedBox(height: 10),
Text(
  data ?? "Tidak ada pesan",
  style: const TextStyle(fontSize: 16),
),
),
),
),
),
),
);
}
}

```

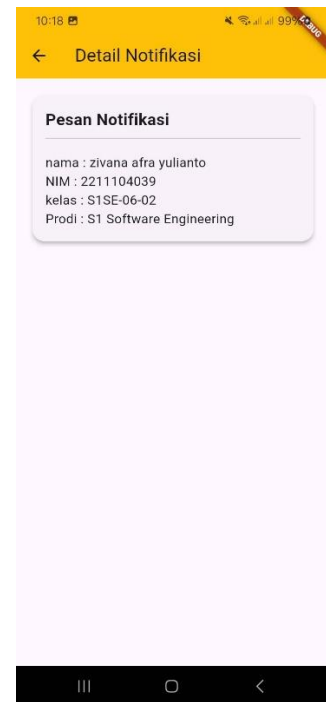
Screenshoot Output



(tampilan awal)



(notifikasi masuk)



(saat mengklik notif)

Deskripsi Program

Aplikasi notifikasi ini memungkinkan pengguna untuk menerima dan mengelola pemberitahuan secara real-time menggunakan Firebase Cloud Messaging (FCM) dan notifikasi lokal. Aplikasi dilengkapi dengan dua fitur utama:

1. Halaman Utama Notifikasi: Pengguna dapat melihat ikon notifikasi dengan tampilan minimalis.
2. Halaman Detail Notifikasi: Setelah mengklik notifikasi, pengguna diarahkan ke halaman detail yang menyajikan informasi terkait notifikasi secara terstruktur. Halaman ini juga menyertakan informasi pribadi pengguna untuk menambahkan konteks.

Dengan antarmuka yang sederhana dan user-friendly, aplikasi ini dirancang untuk memberikan pengalaman yang modern dan intuitif.

BAB III

KESIMPULAN

KESIMPULAN

Aplikasi ini berhasil mengintegrasikan FCM dan notifikasi lokal, memungkinkan pengiriman pesan real-time dengan tampilan yang ramah pengguna. Halaman detail memberikan informasi tambahan, menjadikan aplikasi efektif untuk komunikasi instan dan terorganisir.