

LAPORAN PRAKTIKUM
Modul 11
“DATA STORAGE (BAGIAN II)”



Disusun Oleh:
Ganesha Rahman Gibran -2211104058
Kelas S1SE-06-02

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

Tujuan

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

Landasan Teori

Firebase Cloud Messaging (FCM) adalah layanan yang disediakan oleh Firebase untuk mengirimkan notifikasi atau pesan secara efisien ke berbagai perangkat. Layanan ini memungkinkan pengembang aplikasi untuk mengelola komunikasi dengan pengguna, baik secara individu maupun berdasarkan kelompok tertentu. FCM mendukung dua jenis pesan utama, yaitu **Notification Message** dan **Data Message**, yang dapat disesuaikan dengan kebutuhan aplikasi.

1. Notification Message

Pesan ini secara otomatis menampilkan notifikasi di perangkat pengguna.

- **Foreground:** Pesan diteruskan ke fungsi `onMessage` untuk diproses secara manual.
- **Background/Terminated:** Notifikasi langsung muncul di tray perangkat tanpa memerlukan kode tambahan.

2. Data Message

Pesan ini menyediakan fleksibilitas lebih karena kontennya tidak langsung ditampilkan sebagai notifikasi.

- Memerlukan pengolahan manual menggunakan kode, biasanya dengan bantuan plugin seperti `flutter_local_notifications`.

Integrasi Firebase pada Aplikasi

Untuk menggunakan FCM, aplikasi perlu dihubungkan ke Firebase melalui beberapa langkah, seperti mendaftarkan aplikasi di Firebase Console, menambahkan file konfigurasi `google-services.json`, dan menyertakan dependensi yang relevan dalam proyek Flutter. Selanjutnya, aplikasi perlu diatur agar dapat menangani notifikasi di berbagai kondisi (`foreground`, `background`, atau `terminated`). Pengaturan ini mencakup penggunaan kanal notifikasi, pemrosesan pesan di latar belakang, serta pengambilan token FCM perangkat.

Jenis Pesan pada Firebase Cloud Messaging

Pesan FCM terdiri dari dua jenis:

1. Notification Message: Digunakan untuk notifikasi instan di perangkat.
2. Data Message: Digunakan untuk pengolahan data sebelum menampilkan notifikasi.

Mengirim Notifikasi

Pengiriman pesan dapat dilakukan melalui Firebase Console atau Firebase Cloud Functions. Firebase Console memungkinkan pengembang untuk membuat, mengatur, dan mengirim pesan langsung ke perangkat berdasarkan token atau topik tertentu.

Guided

Input :

- 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:guided11/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(),  
      );  
    }  
  }
```

- my_notification_screen.dart

```
import 'dart:ffi';  
  
import 'package:firebase_messaging/firebase_messaging.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter_local_notifications/flutter_local_notifications.dart';  
import 'package:guided11/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",  
            ),  
          ),  
        );  
      }  
    });  
  }  
}
```

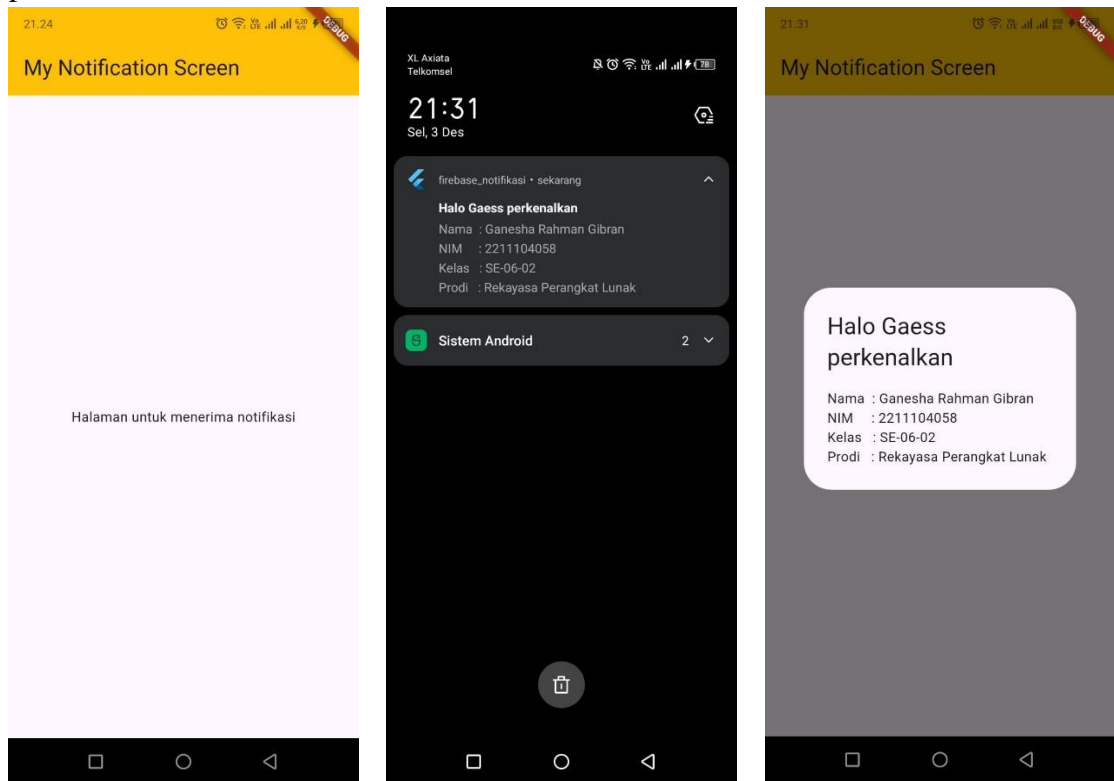
```
// Menangani aksi ketika notifikasi dibuka
FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
  RemoteNotification? notification = message.notification;
  AndroidNotification? android = message.notification?.android;
  // Jika notifikasi tersedia, tampilkan dialog
  if (notification != 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: const Text('My Notification Screen'),
      backgroundColor: Colors.amber,
    ),
    body: const Center(child: Text('Halaman untuk menerima notifikasi')),
  );
}
}
```

Output :



Unguided

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

Input :

- Lib/main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:firebase_notifikasi/my_notification_screen.dart';
import 'package:flutter/material.dart';
import
'package:flutter_local_notifications/flutter_local_notifications.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',
  'High Importance Notifications',
  description: 'This channel is used for important notifications.',
  importance: Importance.high,
);

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

```

- Lib/my_notification_screen.dart

```

import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:firebase_notifikasi/main.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> {
  @override
  void initState() {
    // Fixed the method name from initState to 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,
  notification.title,
  notification.body,
  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 ?? ""),
          content: SingleChildScrollView(
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [Text(notification.body ?? "")],
            ),
          ),
        );
      },
    );
  }
});

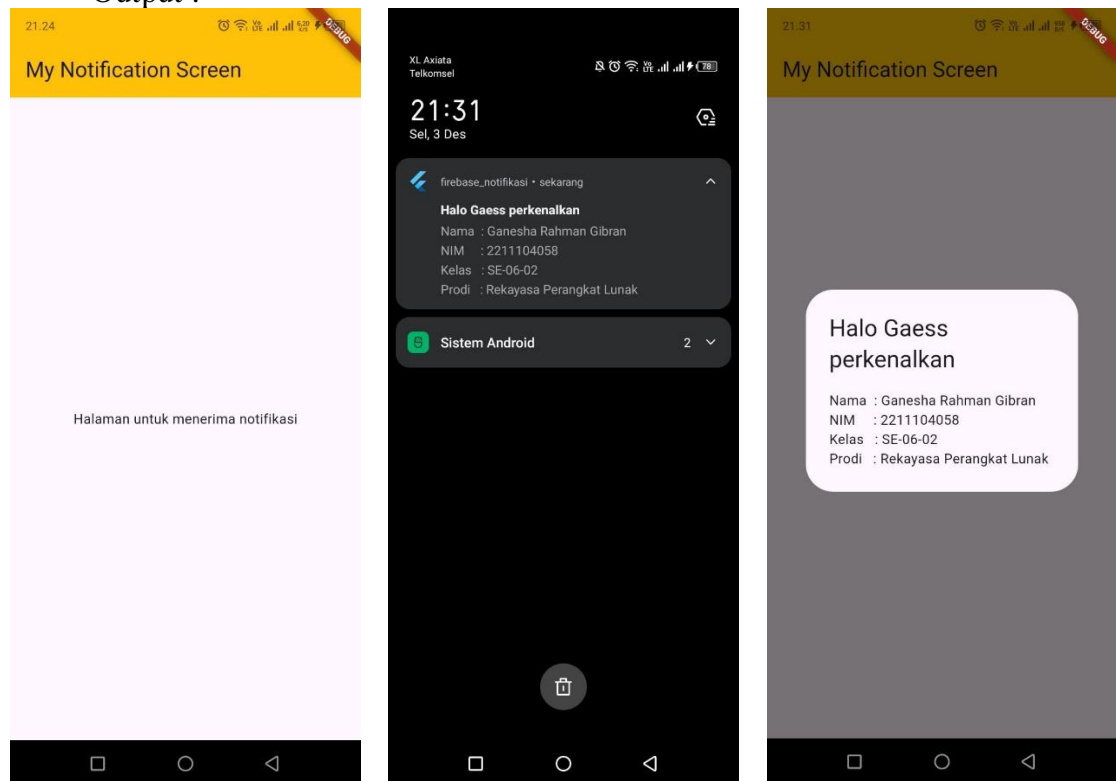
// Memanggil metode untuk mengambil token FCM perangkat
getToken();

// Metode untuk mendapatkan token FCM
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.amber,
    ),
    body: const Center(
      child: Text('Halaman untuk menerima notifikasi'),
    ),
  );
}

```


Output :



Deskripsi kode

Implementasi halaman notifikasi menggunakan Flutter dan Firebase Cloud Messaging (FCM), yang memungkinkan aplikasi menerima dan menampilkan notifikasi. Ketika aplikasi aktif, notifikasi yang diterima ditampilkan melalui `FlutterLocalNotificationsPlugin` menggunakan pengaturan Android spesifik. Selain itu, juga menangani aksi pengguna saat membuka notifikasi, dengan menampilkan detail notifikasi dalam dialog. Token unik perangkat FCM diperoleh melalui metode `getToken()` untuk digunakan dalam pengiriman pesan. Halaman ini menampilkan teks statis sebagai placeholder dan memiliki AppBar dengan judul "My Notification Screen."

Kesimpulan

Firebase Cloud Messaging adalah solusi efektif untuk mengelola pengiriman notifikasi di aplikasi, mendukung kebutuhan komunikasi dengan pengguna secara terstruktur dan efisien. Dengan memahami jenis pesan yang tersedia, pengembang dapat memilih metode pengiriman yang sesuai. Selain itu, integrasi yang mendalam antara Firebase dan Flutter memastikan notifikasi dapat ditampilkan dengan konsisten, baik dalam kondisi aplikasi aktif maupun tidak. Hal ini membantu meningkatkan keterlibatan pengguna dan memastikan pengalaman aplikasi yang optimal.