

LAPORAN GUIDED & UNGUIDED
PEMROGRAMAN PERANGKAT BERGERAK
MODUL XI
FIREBASE NOTIFIKASI



Disusun Oleh :
Ahmad Junaidi / 2211104002
SE-06-01

Asisten Praktikum :
Ayu Susilowati
Noviana Rizki Anisa Putri

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

GUIDED

Source Code build.gradle di app :

```
plugins {  
    id "com.android.application"  
    id "kotlin-android"  
    // The Flutter Gradle Plugin must be applied after the Android and  
    Kotlin Gradle plugins.  
    id "dev.flutter.flutter-gradle-plugin"  
    id 'com.google.gms.google-services'  
}  
  
android {  
    namespace = "com.example.modul10"  
    compileSdk = flutter.compileSdkVersion  
    ndkVersion = flutter.ndkVersion  
  
    compileOptions {  
        sourceCompatibility = JavaVersion.VERSION_1_8  
        targetCompatibility = JavaVersion.VERSION_1_8  
    }  
  
    kotlinOptions {  
        jvmTarget = JavaVersion.VERSION_1_8  
    }  
  
    defaultConfig {  
        // TODO: Specify your own unique Application ID  
        (https://developer.android.com/studio/build/application-id.html).  
        applicationId = "com.example.modul10"  
        // You can update the following values to match your application  
        needs.  
        // For more information, see:  
        https://flutter.dev/to/review-gradle-config.  
        minSdkVersion 21  
        targetSdkVersion 33  
        versionCode = flutter.versionCode  
        versionName = flutter.versionName  
    }  
  
    buildTypes {  
        release {  
            // TODO: Add your own signing config for the release build.  
            // Signing with the debug keys for now, so `flutter run --release`  
            works.
```

```

        signingConfig = signingConfigs.debug
    }
}

flutter {
    source = "../.."
}
dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:33.6.0')

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase
    dependencies
        implementation 'com.google.firebase:firebase-analytics'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}

```

Source Code build.gradle di android:

```

buildscript {
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        classpath "com.android.tools.build:gradle:8.1.2"
        classpath "com.google.gms:google-services:4.3.15"
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}

rootProject.buildDir = "../build"

```

```

subprojects {
  project.buildDir = "${rootProject.buildDir}/${project.name}"
}
subprojects {
  project.evaluationDependsOn(":app")
}

tasks.register("clean", Delete) {
  delete rootProject.buildDir
}

```

Source Code main.dart

```

import 'package:modul10/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';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();

  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackground
  Handler);
  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,
      ),
      home: const MyNotificationScreen(),
    );
  }
}

```

Source Codemy_notification_screen.dart

```

import 'package:modul10/main.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> {
  @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
    ),
  ),
);
},
);
});

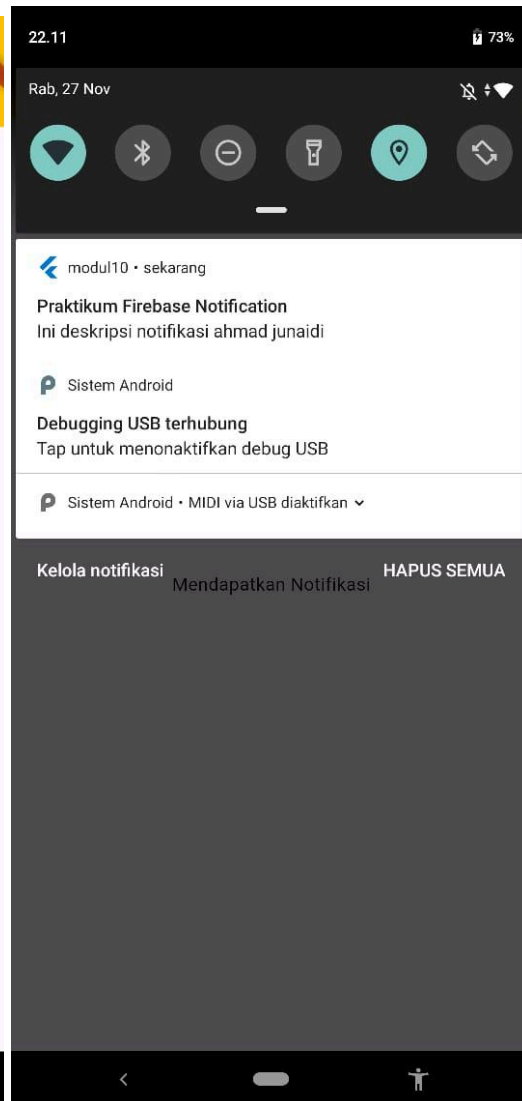
// Memanggil metode untuk mengambil token FCM perangkat
getToken();
}

// 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
}

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('My Notification Screen'),
      backgroundColor: Colors.amber,
      centerTitle: true,
    ),
    body: Center(
      child: Text('Mendapatkan Notifikasi'),
    ),
  );
}
}

```

Output Programs GUIDED:



UNGUIDED

Jawab:

GUIDED

Source Code build.gradle di app :

```
plugins {  
    id "com.android.application"  
    id "kotlin-android"  
    // The Flutter Gradle Plugin must be applied after the Android and  
    Kotlin Gradle plugins.  
    id "dev.flutter.flutter-gradle-plugin"  
    id 'com.google.gms.google-services'  
}  
  
android {  
    namespace = "com.example.modul10"  
    compileSdk = flutter.compileSdkVersion  
    ndkVersion = flutter.ndkVersion  
  
    compileOptions {  
        sourceCompatibility = JavaVersion.VERSION_1_8  
        targetCompatibility = JavaVersion.VERSION_1_8  
    }  
  
    kotlinOptions {  
        jvmTarget = JavaVersion.VERSION_1_8  
    }  
  
    defaultConfig {  
        // TODO: Specify your own unique Application ID  
        (https://developer.android.com/studio/build/application-id.html).  
        applicationId = "com.example.modul10"  
        // You can update the following values to match your application  
        needs.  
        // For more information, see:  
        https://flutter.dev/to/review-gradle-config.  
        minSdkVersion 21  
        targetSdkVersion 33  
        versionCode = flutter.versionCode  
        versionName = flutter.versionName  
    }  
  
    buildTypes {
```

```

    release {
        // TODO: Add your own signing config for the release build.
        // Signing with the debug keys for now, so `flutter run --release`
works.
        signingConfig = signingConfigs.debug
    }
}

flutter {
    source = "../.."
}
dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:33.6.0')

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase
dependencies
    implementation 'com.google.firebase:firebase-analytics'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}

```

Source Code build.gradle di android:

```

buildscript {
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        classpath "com.android.tools.build:gradle:8.1.2"
        classpath "com.google.gms:google-services:4.3.15"
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}

```

```

    }
  }

  rootProject.buildDir = "../build"
  subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
  }
  subprojects {
    project.evaluationDependsOn(":app")
  }

  tasks.register("clean", Delete) {
    delete rootProject.buildDir
  }

```

Source Code main.dart

```

import 'package:modul10/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';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();

  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackground
  Handler);
  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,
      ),
      home: const MyNotificationScreen(),
    );
  }
}

```

Source Codemy_notification_screen.dart

```

import 'package:modul10/main.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> {
  @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
          ),
        ),
      );
    },
  );
}

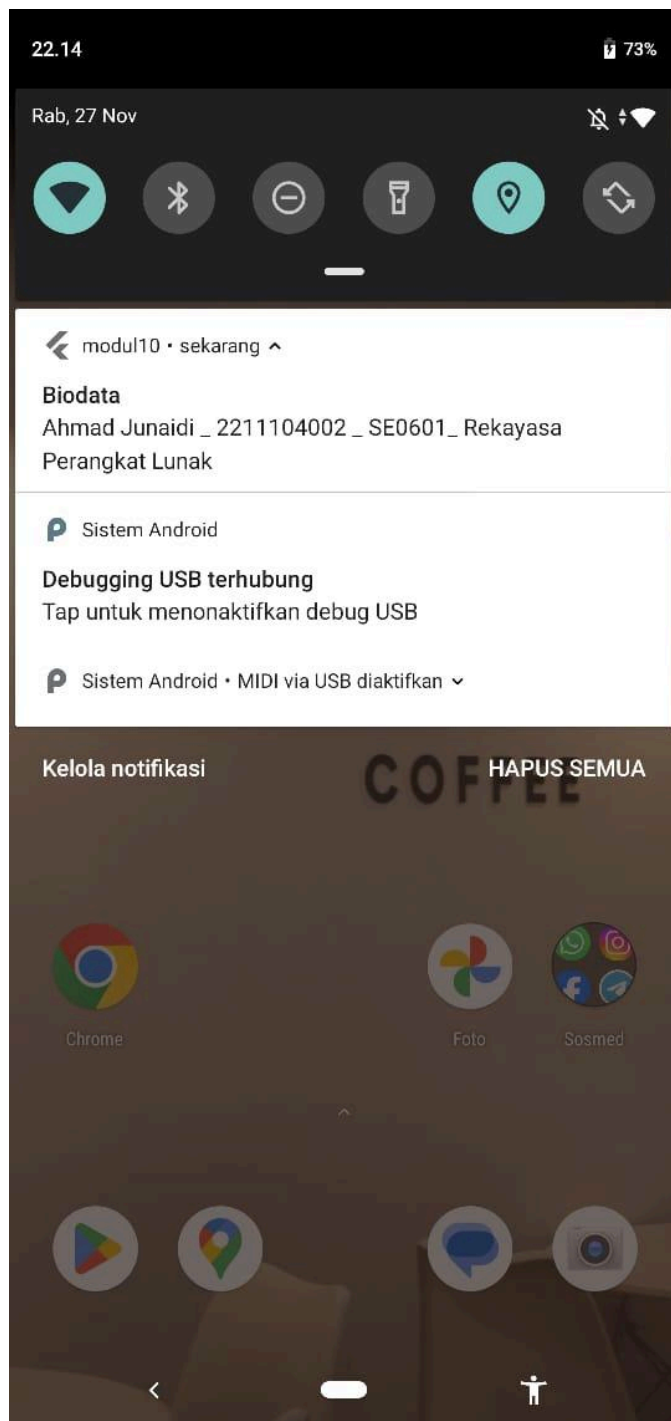
// Memanggil metode untuk mengambil token FCM perangkat
getToken();
}

// 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
}

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('My Notification Screen'),
      backgroundColor: Colors.amber,
      centerTitle: true,
    ),
    body: Center(
      child: Text('Mendapatkan Notifikasi'),
    ),
  );
}
}

```

Output Programs UNGUIDED



Deskripsi Program

Deskripsi Program Firebase Notifikasi

Program ini adalah implementasi Firebase Cloud Messaging (FCM) pada platform Flutter untuk menerima dan mengelola notifikasi di aplikasi mobile. Firebase Cloud Messaging adalah layanan dari Google yang memungkinkan pengembang mengirim pesan atau pemberitahuan kepada perangkat pengguna baik secara individu maupun ke grup tertentu berdasarkan topik langganan. Berikut adalah rincian program:

Tujuan Utama

1. Mengintegrasikan Firebase ke dalam project Flutter.
2. Mengimplementasikan FCM untuk menerima notifikasi di aplikasi.

Fitur Utama

1. Jenis Pesan Notifikasi:
 - Notification Message: Ditampilkan langsung di tray notifikasi perangkat. Pesan akan diterima baik ketika aplikasi berada di latar depan (foreground) maupun latar belakang (background).
 - Data Message: Tidak langsung ditampilkan sebagai notifikasi. Memerlukan kode tambahan untuk diproses menggunakan plugin seperti flutter_local_notifications.
2. Inisialisasi Firebase:
 - Memastikan Firebase sudah terintegrasi dengan aplikasi.
 - Mengatur channel notifikasi dengan tingkat prioritas tinggi.
3. Mekanisme Notifikasi:
 - Mendengarkan pesan yang masuk ketika aplikasi berjalan di latar depan menggunakan `FirebaseMessaging.onMessage`.
 - Menangani aksi ketika pengguna membuka aplikasi dari notifikasi melalui `FirebaseMessaging.onMessageOpenedApp`.
 - Mendapatkan token FCM unik untuk setiap perangkat.
4. Pengiriman Notifikasi:
 - Pesan dikirim melalui Firebase Console. Pengembang dapat mengatur judul, isi pesan, gambar, dan waktu pengiriman notifikasi.