

PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK
TUGAS GUIDED & UNGUIDED

MODUL XI
FIREBASE NOTIFIKASI



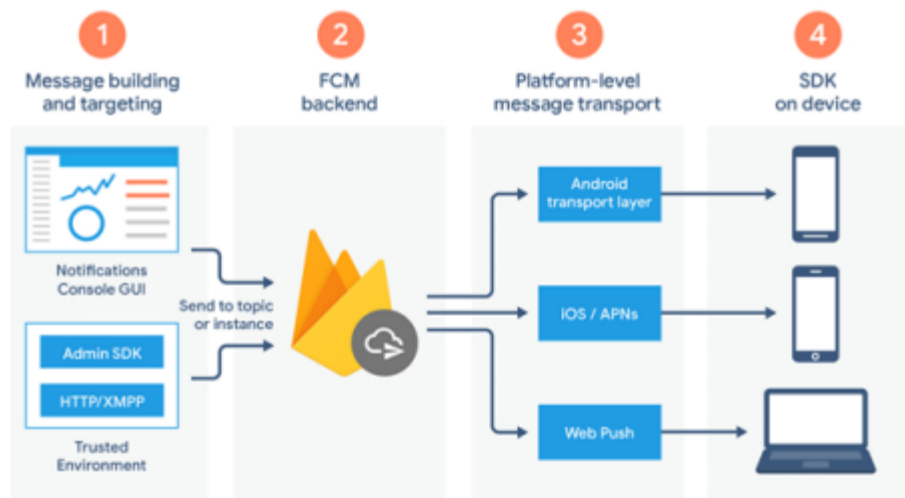
Disusun Oleh :
Rizky Hanifa Afania / 2211104017
SE-06-01

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

GUIDED



Firestore Cloud Messaging (FCM) adalah sebuah layanan yang memudahkan pengiriman notifikasi ke banyak pengguna secara bersamaan. Melalui FCM, kita dapat mengirim pesan ke perangkat tertentu atau ke sekelompok pengguna yang telah berlangganan pada topik tertentu.

Agar dapat mengirim pesan, maka 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 Firestore Cloud Functions.

Jenis Pesan pada Firestore Cloud Messaging (FCM)

Firestore Cloud Messaging (FCM) menyediakan dua jenis pesan utama untuk mendukung pengiriman notifikasi, yaitu:

1. Notification Message

Pesan ini digunakan untuk langsung menampilkan notifikasi pada perangkat pengguna.

Cara kerja:

- Jika aplikasi sedang berjalan di latar depan (foreground), pesan akan diteruskan ke fungsi onMessage untuk diproses.
- Jika aplikasi berada di latar belakang (background) atau sudah dimatikan (terminated), notifikasi akan otomatis muncul di tray perangkat tanpa memerlukan pengaturan tambahan.

2. Data Message

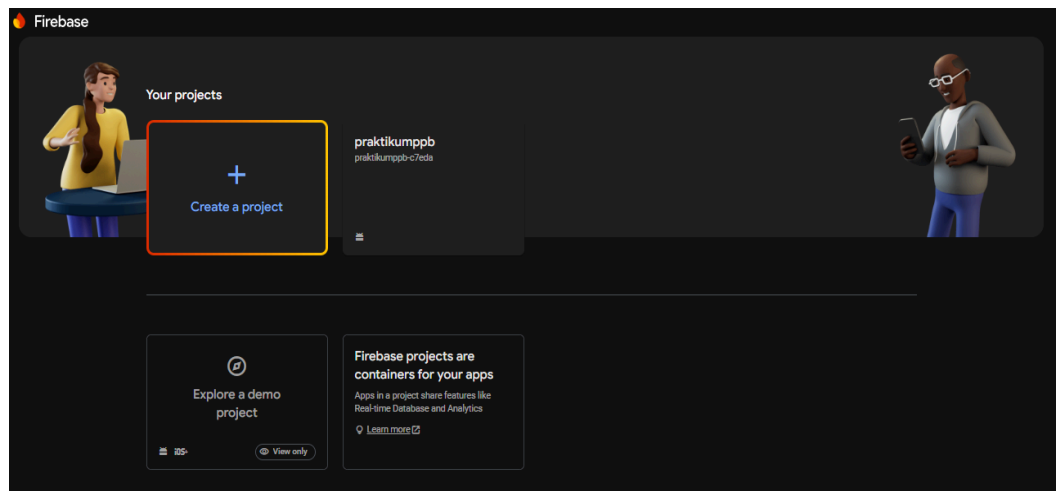
Pesan ini memberikan fleksibilitas lebih karena tidak langsung ditampilkan sebagai notifikasi.

Cara kerja:

- Pesan ini akan diproses oleh aplikasi, baik saat aplikasi sedang aktif maupun tidak aktif.
- Untuk menampilkan notifikasi kepada pengguna, diperlukan kode tambahan, biasanya menggunakan plugin seperti `flutter_local_notifications`.

A. Integrasi Aplikasi dengan Firebase

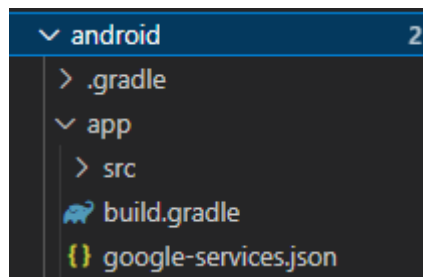
- a. Daftarkan dan login pada [firebase console](#) lalu buat sebuah proyek baru



- b. Tambahkan android package name, app nickname and SHA-1 signing certificate key. Android package name dapat ditemukan di `./android/app/build.gradle`.

```
android > app > build.gradle > ...
9  android {
10      namespace = "com.example.modul_11_firebase"
11      compileSdk = flutter.compileSdkVersion
12      ndkVersion = flutter.ndkVersion
```

- c. Setelah berhasil mendaftarkan aplikasi ke firebase, download file `google-services.json` lalu simpan pada `./android/app/`



- d. Tambahkan plugin dan sdk sesuai perintah pada `./android/build.gradle` dan `./android/app/build.gradle`

- File ./android/build.gradle

```
android > build.gradle > buildscript
1  buildscript {
2
3      repositories {
4          // Make sure that you have the following two repositories
5          google() // Google's Maven repository
6          mavenCentral() // Maven Central repository
7      }
8
9      dependencies {
10         classpath ("com.google.gms:google-services:4.4.2")
11     }
12 }
```

- Tambahkan google service pada plugins di file ./android/app/build.gradle

```
android > app > build.gradle > ...
1  plugins {
2      id "com.android.application"
3      id "kotlin-android"
4      // The Flutter Gradle Plugin must be applied
5      id "dev.flutter.flutter-gradle-plugin"
6      id 'com.google.gms.google-services'
7  }
```

- Tambahkan firebase-bom dan firebase analytics di dependency dalam file ./android/app/build.gradle

```
android > app > build.gradle > android
47  dependencies {
48      // Import the Firebase BoM
49      implementation platform('com.google.firebase:firebase-bom:33.6.0')
50
51      // TODO: Add the dependencies for Firebase products you want to use
52      // When using the BoM, don't specify versions in Firebase dependencies
53      implementation 'com.google.firebase:firebase-analytics'
54
55      // Add the dependencies for any other desired Firebase products
56      // https://firebase.google.com/docs/android/setup#available-libraries
57  }
```

- Setelah memodifikasi file build.gradle, tambahkan beberapa package pada file pubspec.yaml dengan mengetik “flutter pub add firebase_core firebase_messaging flutter_local_notifications” pada terminal.

```
! pubspec.yaml
30 dependencies:
31   flutter:
32     sdk: flutter
33
34   cupertino_icons: ^1.0.8
35   firebase_core: ^3.8.0
36   firebase_messaging: ^15.1.5
37   flutter_local_notifications: ^18.0.1
```

- f. Ubah compile sdk agar menggunakan versi 33 dan min sdk pada versi 21 yang terletak pada file `./android/app/build.gradle`

```
android > app > build.gradle > android
9   android {
25     defaultConfig {
26       applicationId = "com.example.modul_11_firebase"
27
28       minSdkVersion 21
29       targetSdkVersion 33
30       versionCode = flutter.versionCode
31       versionName = flutter.versionName
32     }
```

- g. Lalu lakukan sync dengan mengetik **“flutter pub get”** pada terminal.

```
modul_11_firebase
EXPLORER
  MODUL_11_FIREBASE
    .dart_tool
    .idea
    android
    app
      build.gradle
      google-services.json
      gradle-wrapper.jar
      gradle-wrapper.properties
      .gitignore
      build.gradle
      gradle.properties
      gradlew
      gradlew.bat
      local.properties
      modul_11_firebase...
      settings.gradle
    ios
    lib
      main.dart
    linux
    macos
    test
    web
    OUTLINE
    TIMELINE
    DEPENDENCIES
  build.gradle android
  build.gradle .\app
1 plugins {
2   id "com.android.application"
3   id "kotlin-android"
4   // The Flutter Gradle Plugin must be applied after the Android
  PROBLEMS
  TERMINAL
  powershell
  PS F:\PPB\1. Project saat Praktikum\Pert 11\GUIDED\modul_11_fir
  ebase> flutter pub get
  Resolving dependencies... (1.6s)
  Downloading packages...
  async 2.11.0 (2.12.0 available)
  boolean_selector 2.1.1 (2.1.2 available)
  characters 1.3.0 (1.3.1 available)
  clock 1.1.1 (1.1.2 available)
  collection 1.18.0 (1.19.1 available)
  fake_async 1.3.1 (1.3.2 available)
  flutter_lints 4.0.0 (5.0.0 available)
  leak_tracker 10.0.5 (10.0.8 available)
  leak_tracker_flutter_testing 3.0.5 (3.0.9 available)
  lints 4.0.0 (5.1.0 available)
  matcher 0.12.16+1 (0.12.17 available)
  material_color_utilities 0.11.1 (0.12.0 available)
  meta 1.15.0 (1.16.0 available)
  path 1.9.0 (1.9.1 available)
  stack_trace 1.11.1 (1.12.0 available)
  string_scanner 1.2.0 (1.4.0 available)
  test_api 0.7.2 (0.7.3 available)
  vm_service 14.2.5 (14.3.1 available)
  Got dependencies!
  18 packages have newer versions incompatible with dependency co
  nstraints.
  Try 'flutter pub outdated' for more information.
  PS F:\PPB\1. Project saat Praktikum\Pert 11\GUIDED\modul_11_fir
  ebase>
```

B. Integrasi Aplikasi dengan Firebase

- Pertama, tambahkan kode di fungsi main() untuk memastikan Firebase dan konfigurasi lainnya sudah siap digunakan, berada di file **main.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,
  );
}
```

- Tambahkan kode berikut di bawah import sebagai variabel global, berada di file **main.dart**

```
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
  );
```

- c. Di dalam kelas State, tambahkan konfigurasi untuk menangani notifikasi, berada di file **my_notification_screen.dart**

```
@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
```

```

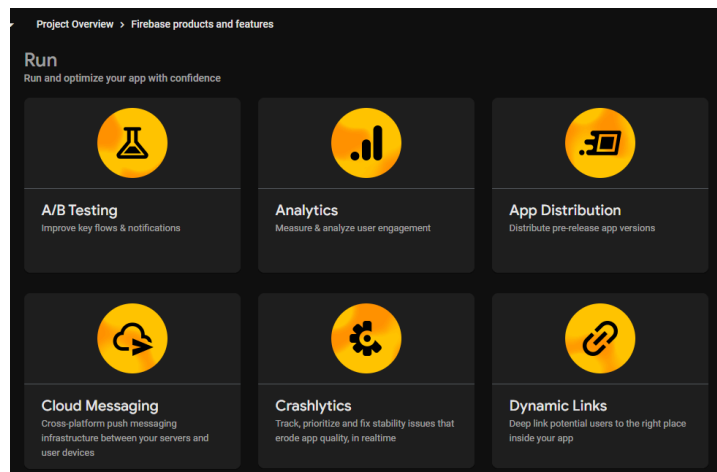
message) {
    RemoteNotification? notification = message.notification;
    AndroidNotification? android =
message.notification?.android;
    // Jika notifikasi tersedia, tampilkan dialog
    if (notification != null && android != null) {
        showDialog(
            context: context,
            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
}

```

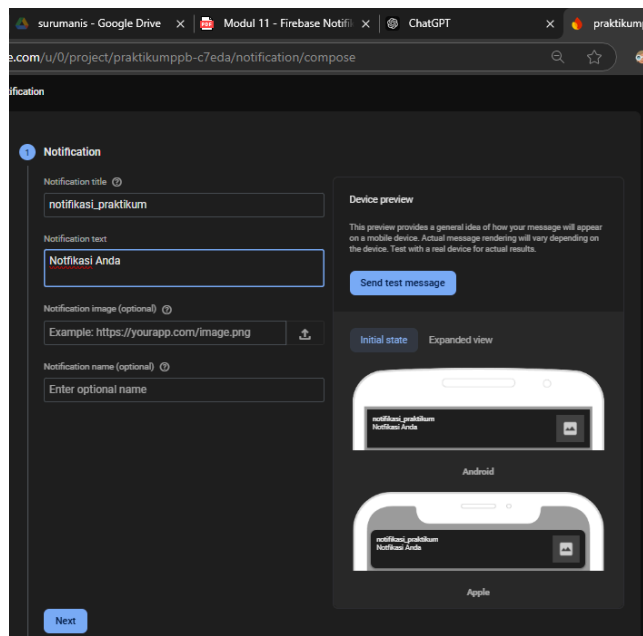
C. Mengirim Notifikasi Via FCM Console

- a. Selanjutnya testing notifikasi, dimulai dengan membuka halaman all product pada tab firebase --> pilih menu cloud messaging --> lalu pilih send your first message.

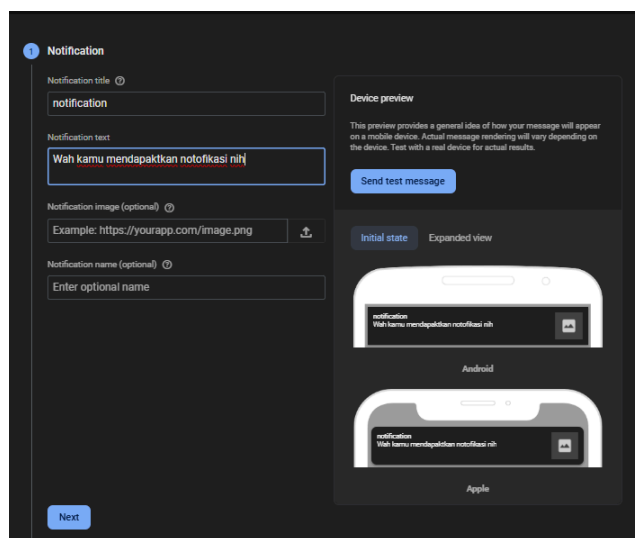


b. Selanjutnya, inputkan judul, pesan, dan foto dari notifikasi kalian sesuai dengan kebutuhan.

- Judul



- Pesan

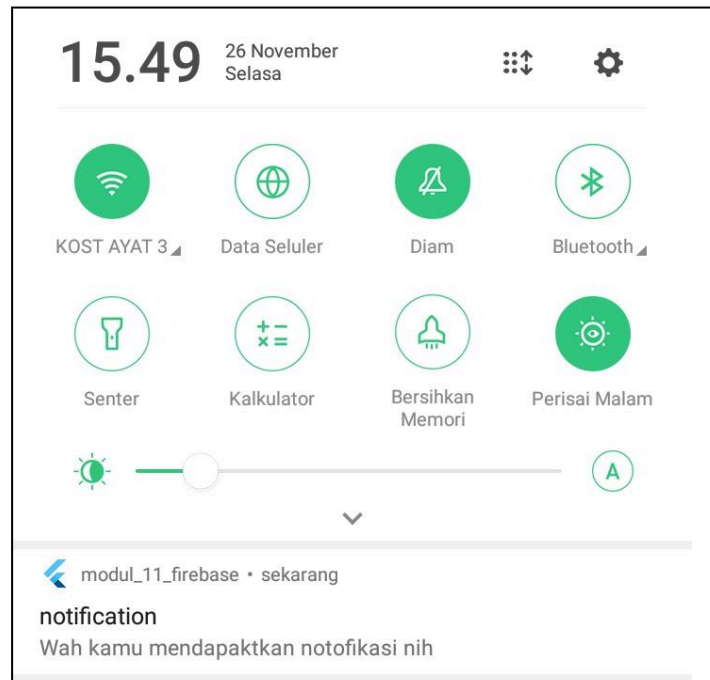


- c. Setelah itu, pilih aplikasi mana yang akan dikirimkan notifikasi dan setting waktu "Now" jika ingin langsung mengirimkan notifikasi. Lalu klik publish jika isi dan settingan notifikasi sudah benar.

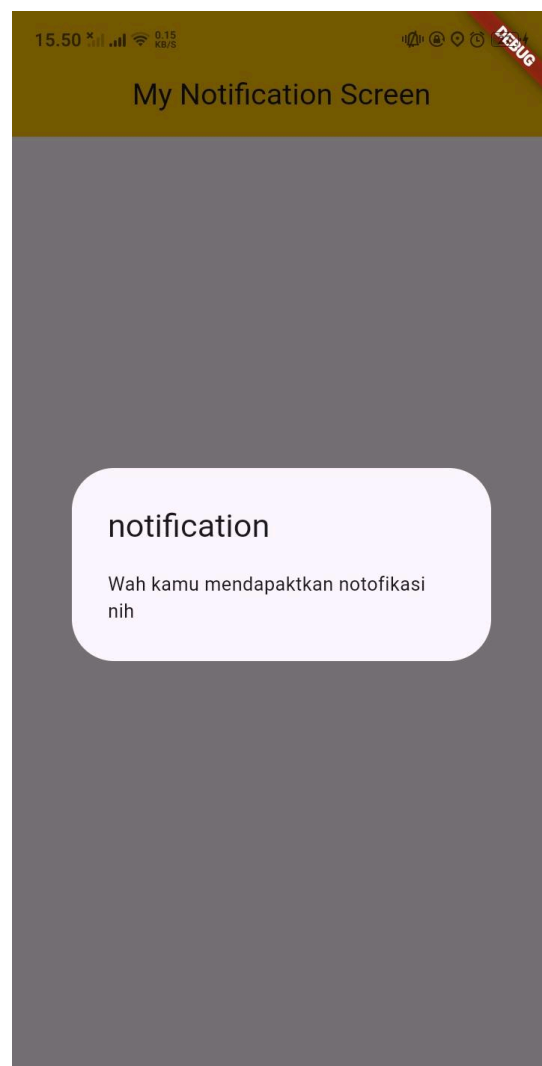
The screenshot shows a dark-themed interface with a vertical progress bar on the left. The steps are: 1. Notification (Wah kamu mendapatkan notifikasi nih), 2. Target (User segment matching one targeting criterion), 3. Scheduling (Send to eligible users), 4. Key events (optional), and 5. Additional options (optional). The 'Scheduling' step is active, and a dropdown menu is open showing 'Now' as the selected option. A 'Next' button is visible below the dropdown. At the bottom right, there are 'Save as draft' and 'Review' buttons.

The screenshot shows a 'Review message' dialog box overlaid on the 'Scheduling' step. The dialog box contains the following information: Notification content (Wah kamu mendapatkan notifikasi nih), Target (User segment matching one targeting criterion), and Scheduling (Send now). There are 'Cancel' and 'Publish' buttons at the bottom right of the dialog. In the background, the 'Scheduling' step is visible, and the 'Additional options (optional)' step is also visible. At the bottom right of the main interface, there are 'Save as draft' and 'Review' buttons.

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



- e. Klik notifikasi maka akan di redirect ke aplikasi dan akan menampilkan seperti pada gambar dibawah



UNGUIDED

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

Note: Jangan lupa sertakan source code, screenshoot output, dan deskripsi program.

Kreatifitas menjadi nilai tambah

Source 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:modul_11_firebase/my_notification_screen.dart';

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
);

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());
}

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(),
    );
  }
}

```

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:modul_11_firebase/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();
}

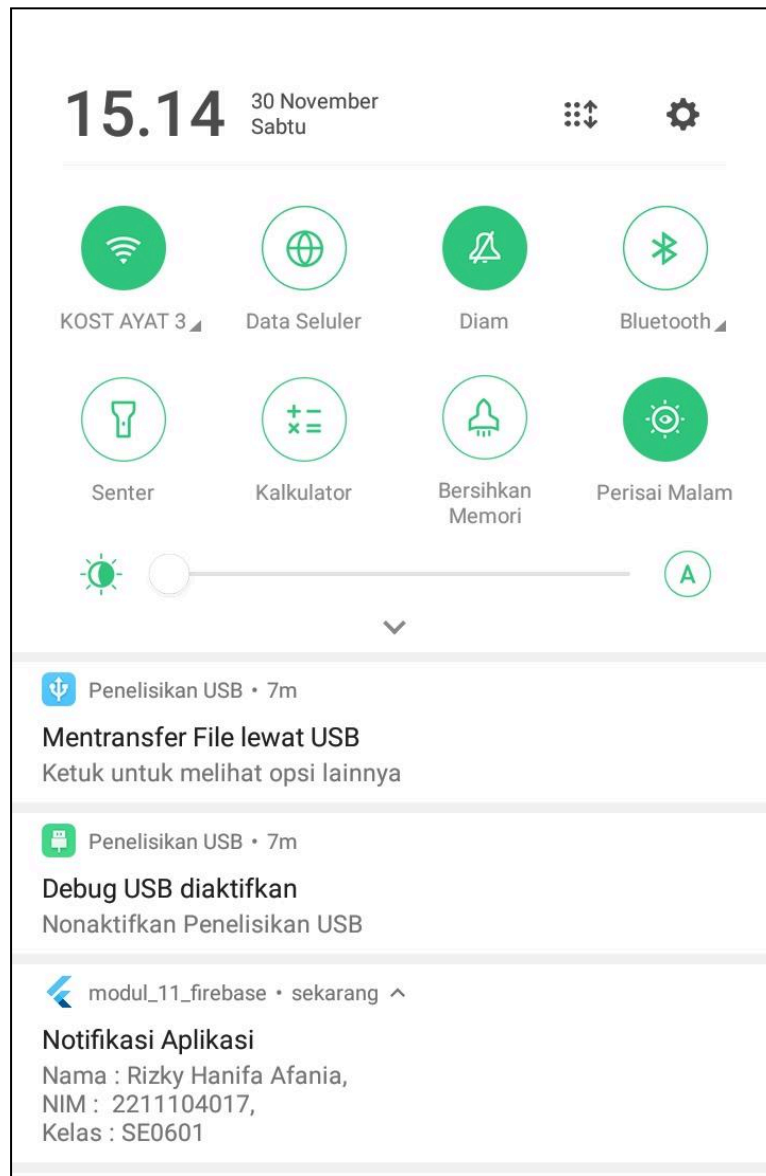
// Metode untuk mendapatkan token FCM
void getToken() async {
  token = await FirebaseMessaging.instance
    .getToken(); // Mendapatkan token FCM perangkat
  print('FCM Token: $token'); // Menampilkan token di log
}

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

```

Screenshoot output

- a. Tampilan saat muncul notifikasi



b. Tampilan setelah notifikasi di klik



Deskripsi Program:

Program di atas mengimplementasikan Firebase Cloud Messaging (FCM) yang digunakan untuk mengirim dan menerima notifikasi. Firebase akan mengirimkan pesan ke perangkat target menggunakan token FCM. Lalu aplikasi mendapatkan token menggunakan `FirebaseMessaging.instance.getToken()`.

Saat aplikasi berjalan, FCM akan menampilkan notifikasi langsung melalui `flutter_local_notifications` jika pesan diterima ketika aplikasi aktif (foreground). Sedangkan jika aplikasi dalam kondisi tidak aktif (background atau terminated) notifikasi akan otomatis muncul di atas layar perangkat tanpa kode tambahan.

Pesan dikirim dari Firebase Console dengan judul "Notifikasi Aplikasi". Jika pengguna mengetuk notifikasi, maka akan di redirect ke aplikasi untuk menampilkan pesan yang lebih detail, disini berupa:

Nama: Rizky Hanifa Afania,

NIM: 2211104017,

Kelas: SE0601