

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XI
FIREBASE NOTIFICATION**



Disusun Oleh :

Devrin Anggun Saputri / 2211104001

SE0601

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

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.

A. 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:

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

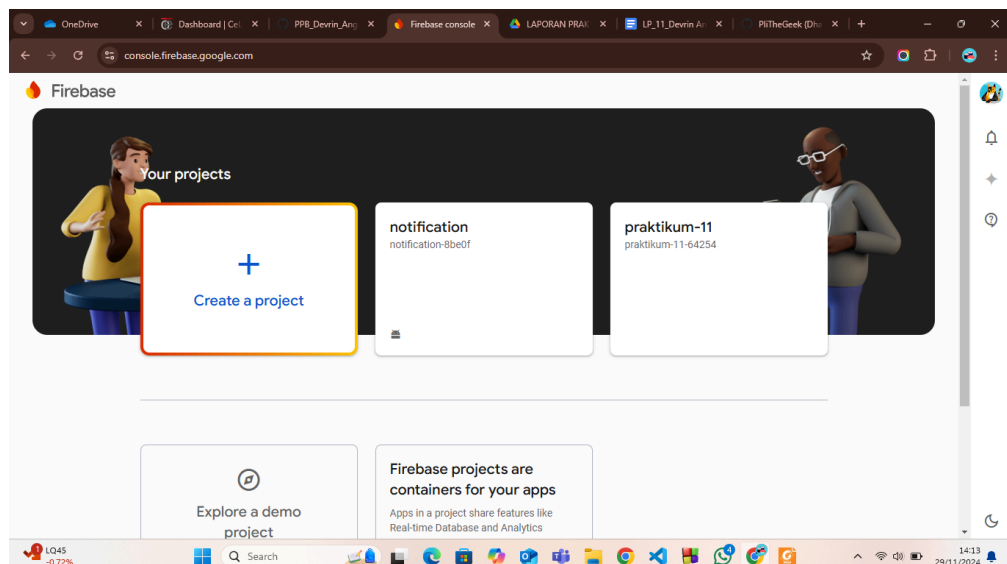
Cara kerja:

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

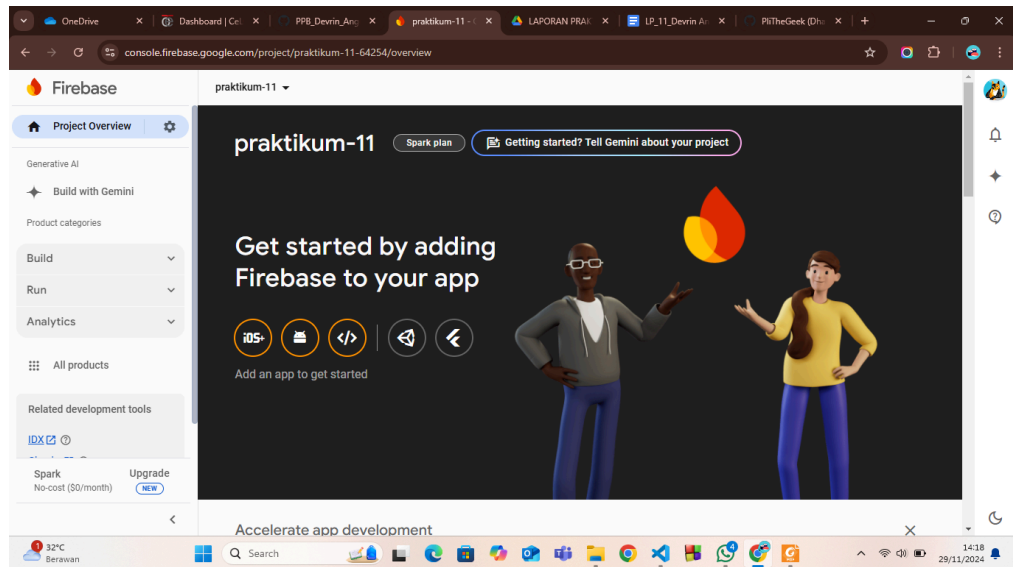
B. Integrasi Aplikasi dengan Firebase

Untuk memulai, 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:

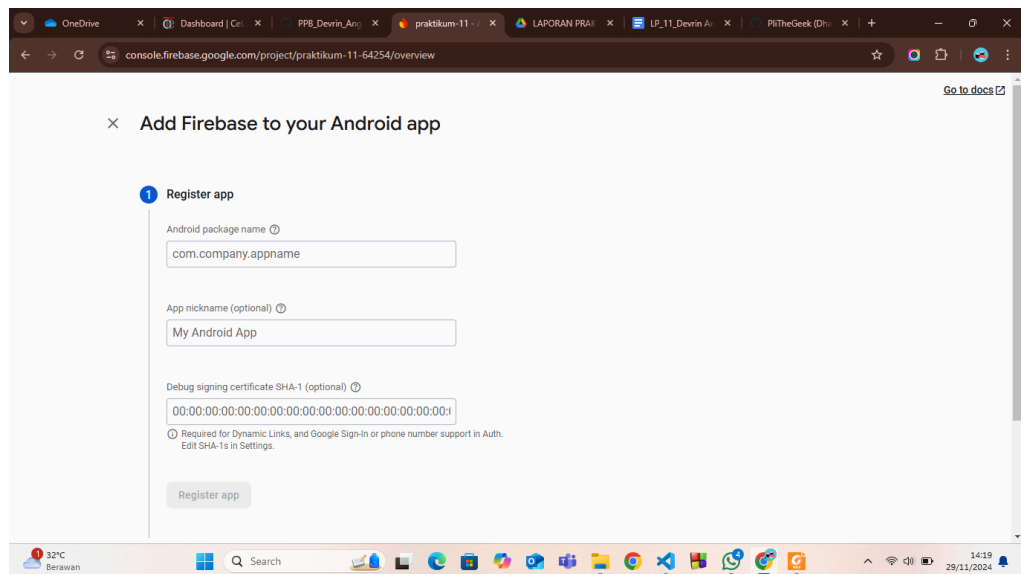
- a. Daftar dan login pada [firebase console](https://console.firebase.google.com) lalu buat sebuah proyek baru.



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



- 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 lalu lakukan sync dengan mengetik "flutter pub get" pada terminal.

```
dependencies {  
    classpath "com.android.tools.build:gradle:8.1.2"  
    classpath "com.google.gms:google-services:4.3.15"  
}
```

```
id "dev.flutter.flutter-gradle-plugin"  
id 'com.google.gms.google-services'
```

```
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'
```

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

```
// FOR MORE INFORMATION
minSdkVersion 21
targetSdkVersion 33
```

C. Membuat Notifikasi Handler

- a. Pertama, tambahkan kode di fungsi main() untuk memastikan Firebase dan konfigurasi lainnya sudah siap digunakan.

```
Run | Debug | Profile
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());
}
```

- b. Tambahkan kode berikut di bawah import sebagai variabel global.

```
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.

```
1  @override
2  State<MyNotificationScreen> createState() => _MyNotificationScreenState();
3  }
4
5  class _MyNotificationScreenState extends State<MyNotificationScreen> {
6    @override
7    void initState() {
8      super.initState();
9      // Membuat pengaturan inisialisasi notifikasi untuk Android
10     var initializationSettingsAndroid =
11       const AndroidInitializationSettings('@mipmap/ic_launcher');
12     var initializationSettings =
13       InitializationSettings(android: initializationSettingsAndroid);
14     FlutterLocalNotificationsPlugin().initialize(initializationSettings);
15   }
```

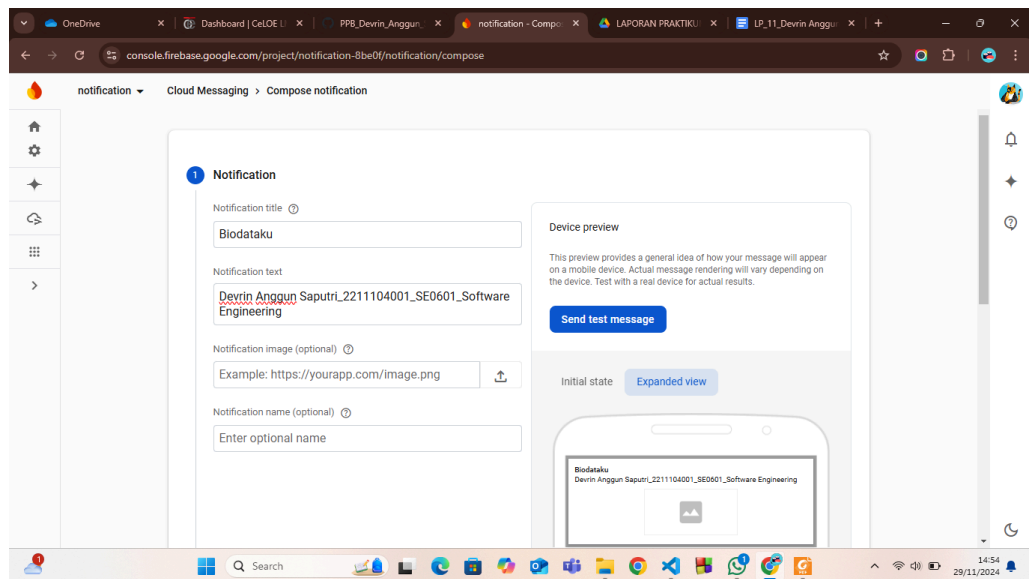
```

16 // Mendengarkan pesan saat aplikasi aktif
17 FirebaseMessaging.onMessage.listen((RemoteMessage message) {
18   RemoteNotification? notification = message.notification;
19   AndroidNotification? android = message.notification?.android;
20 // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
21   if (notification != null && android != null) {
22     FlutterLocalNotificationsPlugin().show(
23       notification.hashCode, // ID notifikasi (hashCode untuk unik)
24       notification.title, // Judul notifikasi
25       notification.body, // Isi notifikasi
26       NotificationDetails(
27         android: AndroidNotificationDetails(
28           channel.id,
29           channel.name,
30           channelDescription: channel.description,
31           color: Colors.blue,
32           icon: "@mipmap/ic_launcher",
33         ),
34       ),
35     );
36   }
37 });
38
39 // Menangani aksi ketika notifikasi dibuka
40 FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
41   RemoteNotification? notification = message.notification;
42   AndroidNotification? android = message.notification?.android;
43 // Jika notifikasi tersedia, tampilkan dialog
44   if (notification != null && android != null) {
45     showDialog(
46       context: context,
47       builder: (_) {
48         return AlertDialog(
49           title: Text(notification.title ?? ""), // Judul dialog
50           content: SingleChildScrollView(
51             child: Column(
52               crossAxisAlignment: CrossAxisAlignment.start,
53               children: [Text(notification.body ?? "")], // Isi dialog
54             ),
55           ),
56         );
57       },
58     );
59   }
60 });
61
62 // Memanggil metode untuk mengambil token FCM perangkat
63 getToken();
64 }
65
66 // Metode untuk mendapatkan token FCM
67 void getToken() async {
68   token = await FirebaseMessaging.instance
69     .getToken(); // Mendapatkan token FCM perangkat
70   print('FCM Token: $token'); // Menampilkan token di log
71 }
72
73 Widget build(BuildContext context) {
74   return Scaffold(
75     appBar: AppBar(
76       title: const Text('My Notification Screen'),
77       backgroundColor: Colors.amber,
78       centerTitle: true,
79     ),
80     body: Center(
81       child: Text('Mendapatkan Notifikasi'),
82     ),
83   );
84 }
85 }

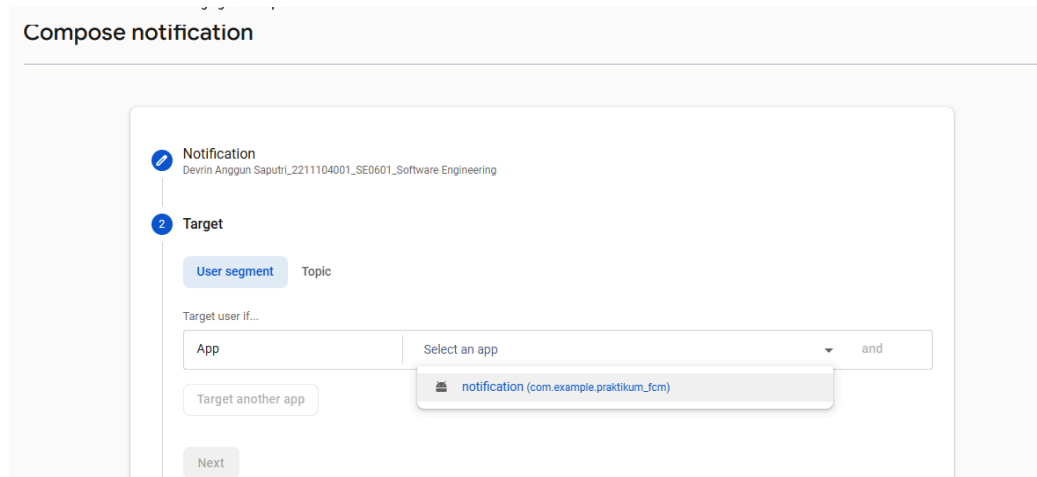
```

D. Mengirim Notifikasi Via FCM Console

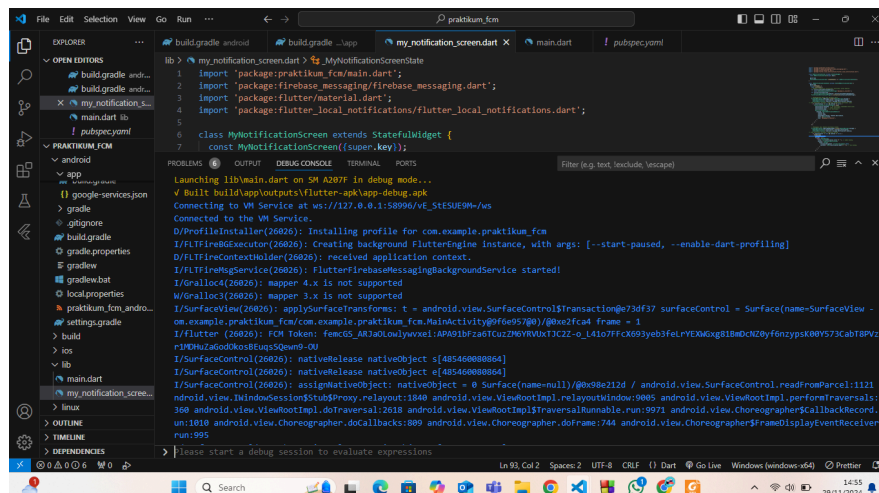
- Dimulai dengan membuka halaman all product pada tab firebase --> pilih menu cloud messaging --> lalu pilih send your first message.
- Selanjutnya, inputkan judul, pesan, dan foto dari notifikasi kalian sesuai dengan kebutuhan.



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



Hasil Praktikum:

- main.dart

```
1 import 'package:praktikum_fcm/my_notification_screen.dart';
2 import 'package:firebase_core/firebase_core.dart';
3 import 'package:firebase_messaging/firebase_messaging.dart';
4 import 'package:flutter/material.dart';
5 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
6
7 void main() async {
8   WidgetsFlutterBinding.ensureInitialized();
9   await Firebase.initializeApp();
10  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
11  await FlutterLocalNotificationsPlugin()
12    .resolvePlatformSpecificImplementation<
13      AndroidFlutterLocalNotificationsPlugin>()
14    ?.createNotificationChannel(channel);
15  await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
16    alert: true,
17    badge: true,
18    sound: true,
19  );
20  runApp(const MyApp());
21 }
22
23 String? token;
24 Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
25   await Firebase.initializeApp();
26   print('Handling a background message: ${message.messageId}');
27 }
28
29 const AndroidNotificationChannel channel = AndroidNotificationChannel(
30   'high_importance_channel', // ID Channel
31   'High Importance Notifications', // Nama Channel
32   description:
33     'This channel is used for important notifications.', // Deskripsi Channel
34   importance: Importance.high, // Prioritas
35 );
36
37 class MyApp extends StatelessWidget {
38   const MyApp({super.key});
39
40   @override
41   Widget build(BuildContext context) {
42     return MaterialApp(
43       title: 'Flutter Demo',
44       theme: ThemeData(
45         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
46         useMaterial3: true,
47       ),
48       home: const MyNotificationScreen(),
49     );
50   }
51 }
```

- my_notification_screen.dart

```
1 import 'package:praktikum_fcm/main.dart';
2 import 'package:firebase_messaging/firebase_messaging.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
5
6 class MyNotificationScreen extends StatefulWidget {
7   const MyNotificationScreen({super.key});
8
9   @override
10  State<MyNotificationScreen> createState() => _MyNotificationScreenState();
11 }
12
13 class _MyNotificationScreenState extends State<MyNotificationScreen> {
14   @override
15   void initState() {
16     super.initState();
17     // Membuat pengaturan inisialisasi notifikasi untuk Android
18     var initializationSettingsAndroid =
19       const AndroidInitializationSettings('@mipmap/ic_launcher');
20     var initializationSettings =
21       InitializationSettings(android: initializationSettingsAndroid);
22     FlutterLocalNotificationsPlugin().initialize(initializationSettings);
23
24     // Mendengarkan pesan saat aplikasi aktif
25     FirebaseMessaging.onMessage.listen((RemoteMessage message) {
26       RemoteNotification? notification = message.notification;
27       AndroidNotification? android = message.notification?.android;
28       // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
29       if (notification != null && android != null) {
30         FlutterLocalNotificationsPlugin().show(
31           notification.hashCode, // ID notifikasi (hashCode untuk unik)
32           notification.title, // Judul notifikasi
33           notification.body, // Isi notifikasi
34           NotificationDetails(
35             android: AndroidNotificationDetails(
36               channel.id,
37               channel.name,
38               channelDescription: channel.description,
39               color: Colors.blue,
40               icon: "@mipmap/ic_launcher",
41             ),
42           ),
43         );
44       }
45     });
46
47     // Menangani aksi ketika notifikasi dibuka
48     FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
49       RemoteNotification? notification = message.notification;
50       AndroidNotification? android = message.notification?.android;
51       // Jika notifikasi tersedia, tampilkan dialog
52       if (notification != null && android != null) {
53         showDialog(
54           context: context,
55           builder: (_) {
56             return AlertDialog(
57               title: Text(notification.title ?? ""), // Judul dialog
58               content: SingleChildScrollView(
59                 child: Column(
60                   crossAxisAlignment: CrossAxisAlignment.start,
61                   children: [Text(notification.body ?? "")], // Isi dialog
62                 ),
63               ),
64             );
65           },
66         );
67       }
68     });
69   }
```

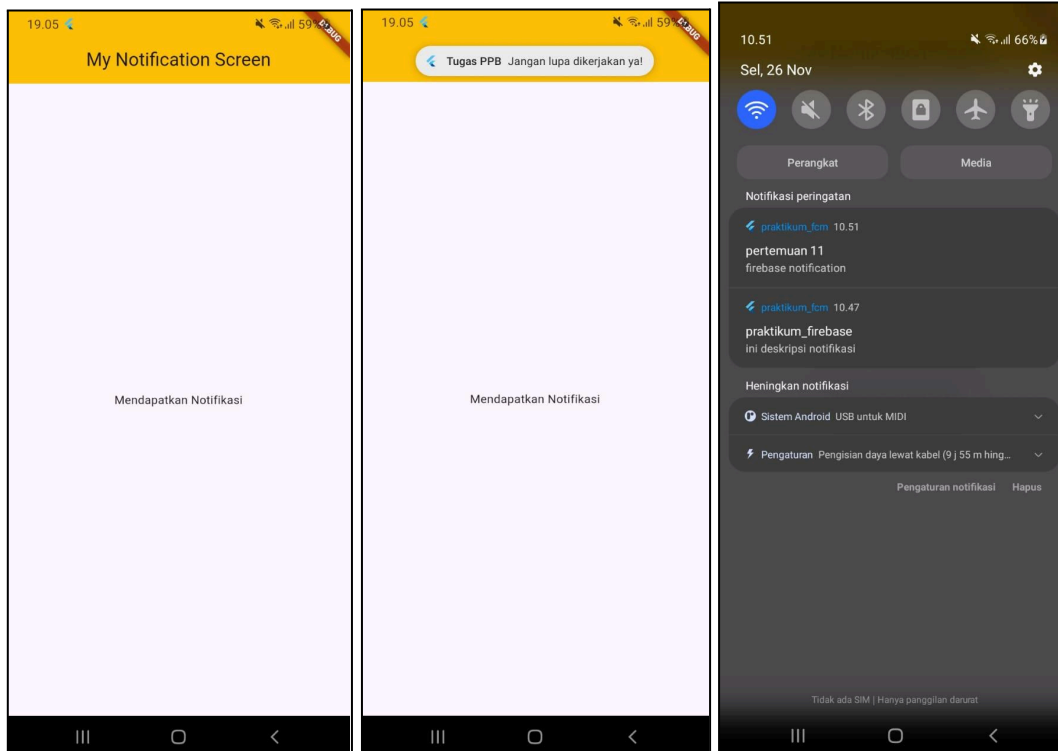


```

70 // Memanggil metode untuk mengambil token FCM perangkat
71 getToken();
72 }
73
74 // Metode untuk mendapatkan token FCM
75 void getToken() async {
76   token = await FirebaseMessaging.instance
77     .getToken(); // Mendapatkan token FCM perangkat
78   print('FCM Token: $token'); // Menampilkan token di log
79 }
80
81 Widget build(BuildContext context) {
82   return Scaffold(
83     appBar: AppBar(
84       title: const Text('My Notification Screen'),
85       backgroundColor: Colors.amber,
86       centerTitle: true,
87     ),
88     body: Center(
89       child: Text('Mendapatkan Notifikasi'),
90     ),
91   );
92 }
93 }

```

Output Hasil:



UNGUIDED

Tugas Mandiri

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.

1. Source Code:

- main.dart

```
1 import 'package:praktikum_fcm/my_notification_screen.dart';
2 import 'package:firebase_core/firebase_core.dart';
3 import 'package:firebase_messaging/firebase_messaging.dart';
4 import 'package:flutter/material.dart';
5 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
6
7 void main() async {
8   WidgetsFlutterBinding.ensureInitialized();
9   await Firebase.initializeApp();
10  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
11  await FlutterLocalNotificationsPlugin()
12    .resolvePlatformSpecificImplementation<
13      AndroidFlutterLocalNotificationsPlugin>()
14    ?.createNotificationChannel(channel);
15  await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
16    alert: true,
17    badge: true,
18    sound: true,
19  );
20  runApp(const MyApp());
21 }
22
23 String? token;
24 Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
25   await Firebase.initializeApp();
26   print('Handling a background message: ${message.messageId}');
27 }
28
29 const AndroidNotificationChannel channel = AndroidNotificationChannel(
30   'high_importance_channel', // ID Channel
31   'High Importance Notifications', // Nama Channel
32   description:
33     'This channel is used for important notifications.', // Deskripsi Channel
34   importance: Importance.high, // Prioritas
35 );
36
37 class MyApp extends StatelessWidget {
38   const MyApp({super.key});
39
40   @override
41   Widget build(BuildContext context) {
42     return MaterialApp(
43       title: 'Flutter Demo',
44       theme: ThemeData(
45         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
46         useMaterial3: true,
47       ),
48       home: const MyNotificationScreen(),
49     );
50   }
51 }
```

- my_notification_screen.dart

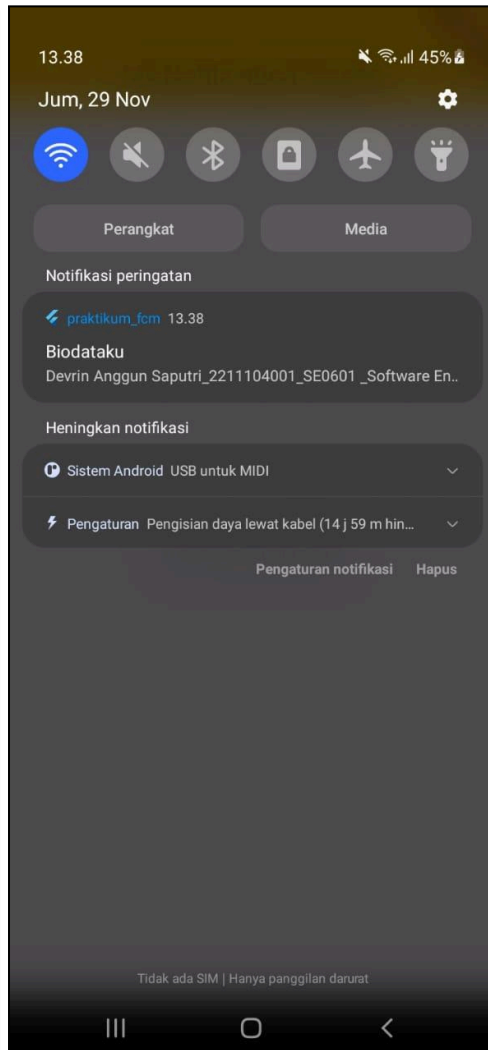
```
1 import 'package:praktikum_fcm/main.dart';
2 import 'package:firebase_messaging/firebase_messaging.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
5
6 class MyNotificationScreen extends StatefulWidget {
7   const MyNotificationScreen({super.key});
8
9   @override
10  State<MyNotificationScreen> createState() => _MyNotificationScreenState();
11 }
12
13 class _MyNotificationScreenState extends State<MyNotificationScreen> {
14   @override
15   void initState() {
16     super.initState();
17     // Membuat pengaturan inisialisasi notifikasi untuk Android
18     var initializationSettingsAndroid =
19       const AndroidInitializationSettings('@mipmap/ic_launcher');
20     var initializationSettings =
21       InitializationSettings(android: initializationSettingsAndroid);
22     FlutterLocalNotificationsPlugin().initialize(initializationSettings);
23
24     // Mendengarkan pesan saat aplikasi aktif
25     FirebaseMessaging.onMessage.listen((RemoteMessage message) {
26       RemoteNotification? notification = message.notification;
27       AndroidNotification? android = message.notification?.android;
28       // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
29       if (notification != null && android != null) {
30         FlutterLocalNotificationsPlugin().show(
31           notification.hashCode, // ID notifikasi (hashCode untuk unik)
32           notification.title, // Judul notifikasi
33           notification.body, // Isi notifikasi
34           NotificationDetails(
35             android: AndroidNotificationDetails(
36               channel.id,
37               channel.name,
38               channelDescription: channel.description,
39               color: Colors.blue,
40               icon: "@mipmap/ic_launcher",
41             ),
42           ),
43         );
44       }
45     });
46
47     // Menangani aksi ketika notifikasi dibuka
48     FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
49       RemoteNotification? notification = message.notification;
50       AndroidNotification? android = message.notification?.android;
51       // Jika notifikasi tersedia, tampilkan dialog
52       if (notification != null && android != null) {
53         showDialog(
54           context: context,
55           builder: (_) {
56             return AlertDialog(
57               title: Text(notification.title ?? ""), // Judul dialog
58               content: SingleChildScrollView(
59                 child: Column(
60                   crossAxisAlignment: CrossAxisAlignment.start,
61                   children: [Text(notification.body ?? "")], // Isi dialog
62                 ),
63               ),
64             );
65           },
66         );
67       }
68     });
69   }
```

```

70     // Memanggil metode untuk mengambil token FCM perangkat
71     getToken();
72 }
73
74 // Metode untuk mendapatkan token FCM
75 void getToken() async {
76     token = await FirebaseMessaging.instance
77         .getToken(); // Mendapatkan token FCM perangkat
78     print('FCM Token: $token'); // Menampilkan token di log
79 }
80
81 Widget build(BuildContext context) {
82     return Scaffold(
83         appBar: AppBar(
84             title: const Text('My Notification Screen'),
85             backgroundColor: Colors.amber,
86             centerTitle: true,
87         ),
88         body: Center(
89             child: Text('Mendapatkan Notifikasi'),
90         ),
91     );
92 }
93 }

```

Output Hasil:



2. Deskripsi Program

Program tersebut adalah aplikasi Flutter yang mengintegrasikan Firebase Cloud Messaging (FCM) untuk mengelola notifikasi push. Aplikasi ini mendukung notifikasi saat aplikasi berada dalam mode latar belakang, foreground, maupun saat dibuka dari notifikasi. Program menggunakan **Flutter Local Notifications** untuk menampilkan notifikasi lokal pada perangkat Android dan mengatur channel notifikasi dengan prioritas tinggi. Fitur utamanya meliputi:

1. **Inisialisasi Firebase** untuk mendukung pengiriman notifikasi.
2. **Listener FCM** untuk mendeteksi notifikasi saat aplikasi aktif, menampilkan dialog saat notifikasi dibuka, dan mencetak token perangkat.
3. **Pengaturan Notifikasi Lokal** untuk mengatur tampilan dan perilaku notifikasi.
4. Halaman utama dengan UI sederhana yang menampilkan teks "Mendapatkan Notifikasi" sebagai indikator fungsi dasar aplikasi.