

**LAPORAN PRAKTIKUM PEMROGRAMAN
PERANGKAT BERGERAK
MODUL 11**



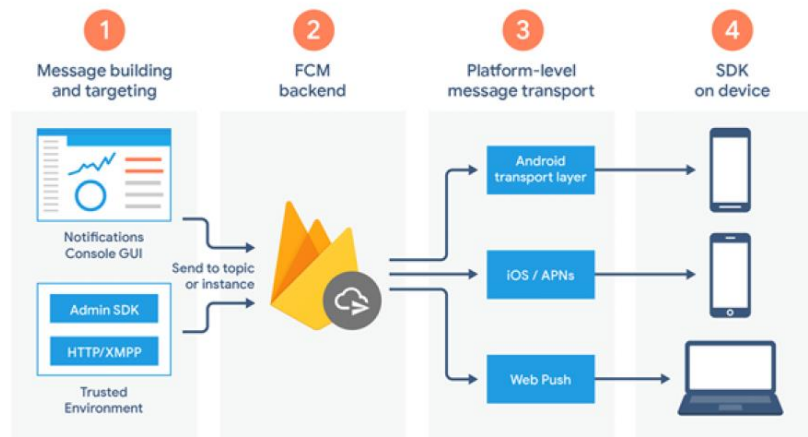
Oleh: Lintang Suminar Tyas Wening
2211104009
SE-06-01

**PRODI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

GUIDED

FIREBASE NOTIFIKASI

Firestore Cloud Messaging



Gambar 1. Cara Kerja FCM

Firestore 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. Dokumentasi lengkapnya dapat diakses di <https://firebase.google.com/docs/flutter/setup?platform=android>

Agar dapat mengirim pesan, Kita 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**.

A. Jenis Pesan pada Firestore 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:

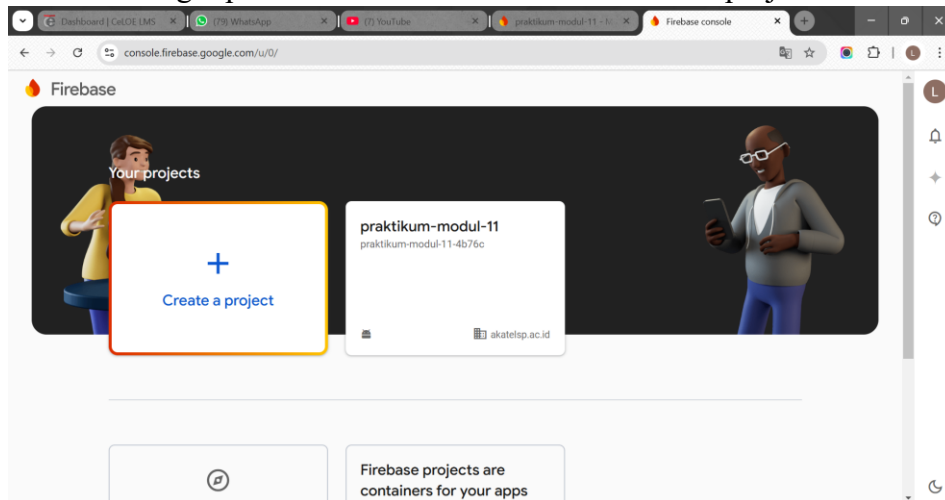
Pesan ini akan diproses oleh aplikasi, baik saat aplikasi aktif maupun tidak aktif.

Untuk menampilkan notifikasi ke pengguna, Kita perlu menambahkan kode khusus, biasanya menggunakan plugin seperti `flutter_local_notifications`.

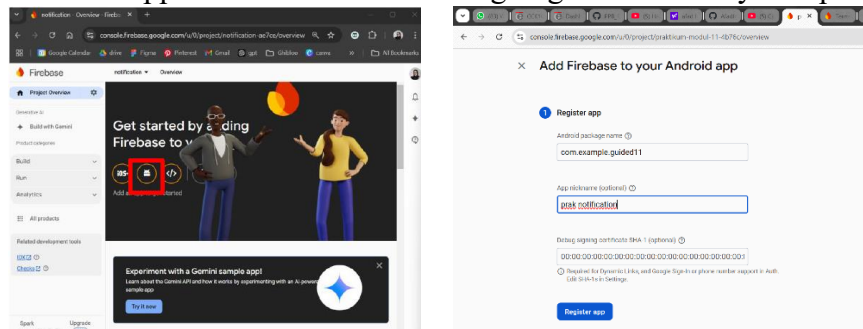
B. Integrasi Aplikasi dengan Firebase

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:

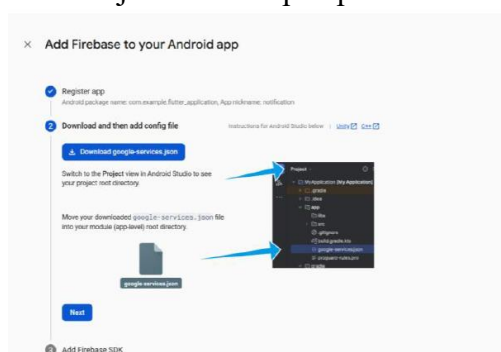
- a. Daftar dan login pada firebase console lalu buat sebuah proyek baru.



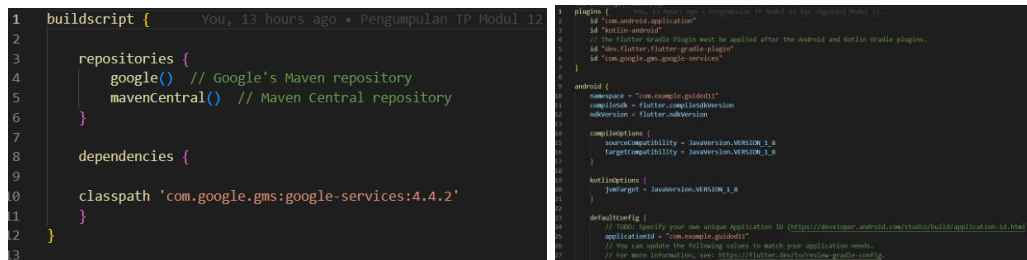
- b. Tambahkan android package name, app nickname and SHA-1 signing certificate key. Android package name dapat ditemukan di `./android/app/build.gradle`. Catatan: App nickname and SHA-1 signing certificate key itu opsional.



- 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.

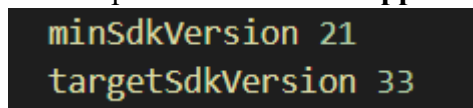


```

1 buildscript {
2
3     repositories {
4         google() // Google's Maven repository
5         mavenCentral() // Maven Central repository
6     }
7
8     dependencies {
9
10        classpath 'com.google.gms:google-services:4.4.2'
11    }
12 }
13
14 plugins {
15     id "com.android.application"
16     id "kotlin-android"
17     // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
18     id "dev.flutter.flutter-gradle-plugin"
19     id "com.google.gms.google-services"
20 }
21
22 android {
23     namespace = "com.example.guided11"
24     compileSdk = flutter.compileSdkVersion
25     ndkVersion = flutter.ndkVersion
26
27     compileOptions {
28         sourceCompatibility = JavaVersion.VERSION_1_8
29         targetCompatibility = JavaVersion.VERSION_1_8
30     }
31
32     kotlinOptions {
33         jvmTarget = JavaVersion.VERSION_1_8
34     }
35
36     defaultConfig {
37         // You can specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
38         applicationId = "com.example.guided11"
39         // You can update the following values to match your application needs.
40         // For more information, see: https://flutter.dev/to/android-manifest
41     }
42 }

```

- 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`



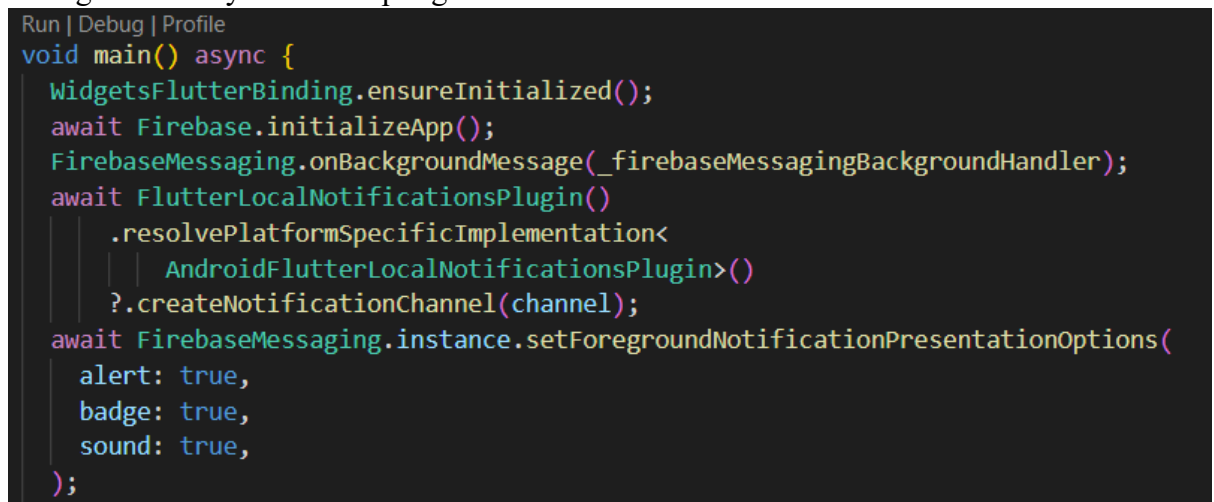
```

minSdkVersion 21
targetSdkVersion 33

```

C. Membuat Notifikasi Handler

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

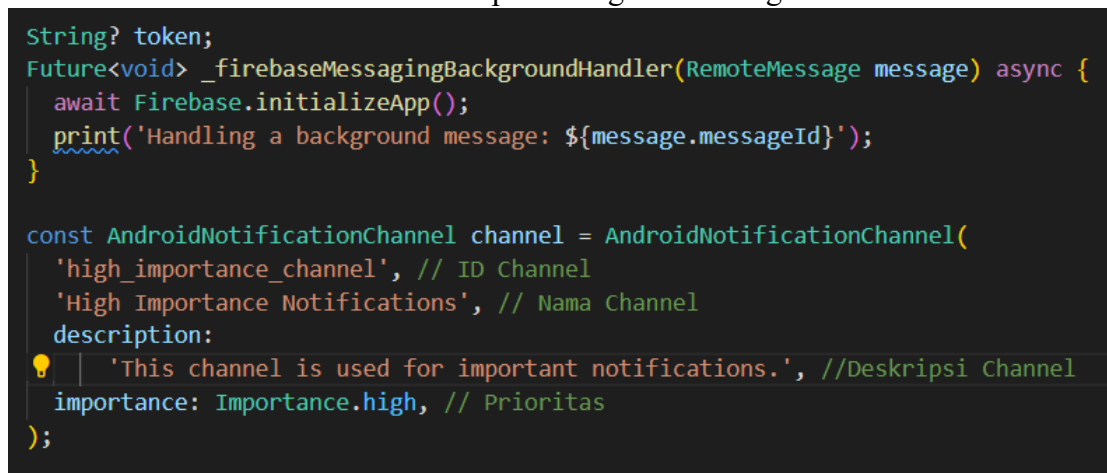


```

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,
    );
}

```

- 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
);

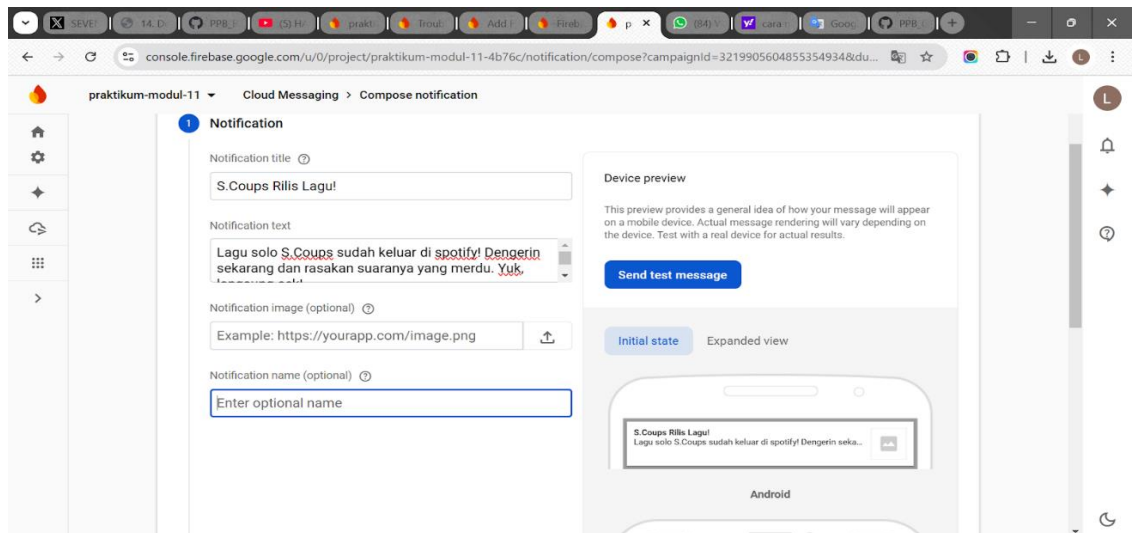
```

- c. Di dalam kelas State, tambahkan konfigurasi untuk menangani notifikasi

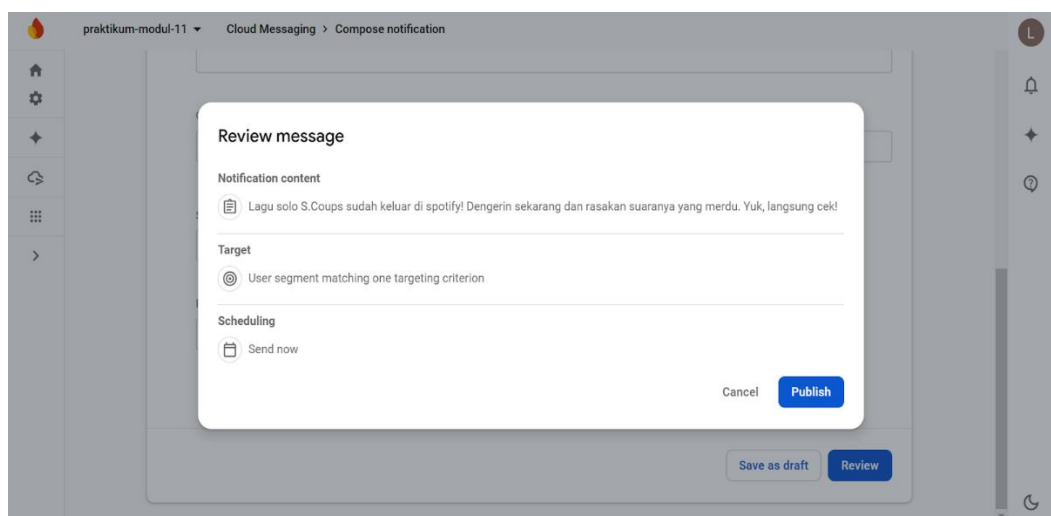
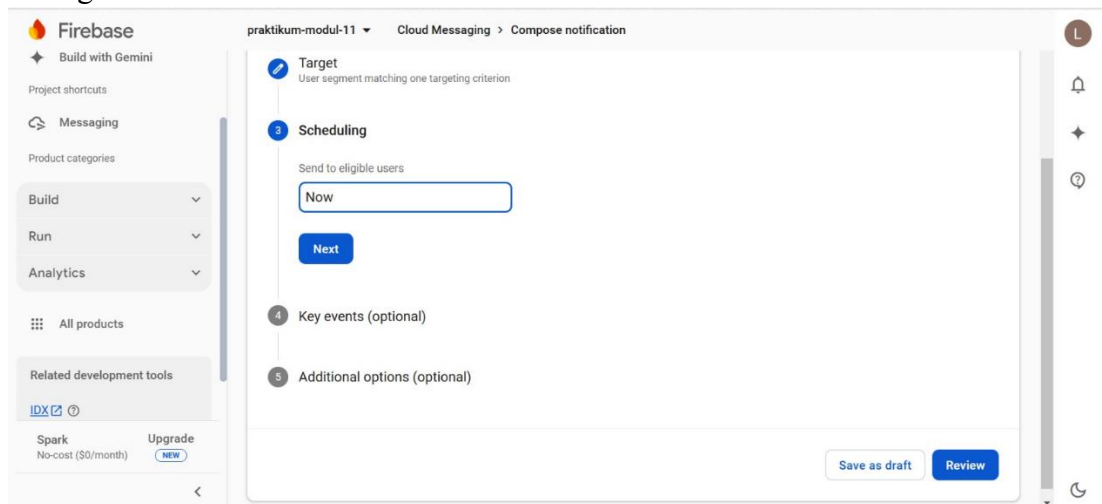
```
1  @override
2  void initState() {
3    super.initState();
4    // Membuat pengaturan inisialisasi notifikasi untuk Android
5    var initializationSettingsAndroid =
6      const AndroidInitializationSettings('@mipmap/ic_launcher');
7    var initializationSettings =
8      InitializationSettings(android: initializationSettingsAndroid);
9    FlutterLocalNotificationsPlugin().initialize(initializationSettings);
10   // Mendengarkan pesan saat aplikasi aktif
11   FirebaseMessaging.onMessage.listen((RemoteMessage message) {
12     RemoteNotification? notification = message.notification;
13     AndroidNotification? android = message.notification?.android;
14     // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
15     if (notification != null && android != null) {
16       FlutterLocalNotificationsPlugin().show(
17         notification.hashCode, // ID notifikasi (hashCode untuk unik)
18         notification.title, // Judul notifikasi
19         notification.body, // Isi notifikasi
20         NotificationDetails(
21           android: AndroidNotificationDetails(
22             channel.id,
23             channel.name,
24             channelDescription: channel.description,
25             color: Colors.blue,
26             icon: "@mipmap/ic_launcher",
27           ),
28         ),
29       );
30     }
31   });
32   // Menangani aksi ketika notifikasi dibuka
33   FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
34     RemoteNotification? notification = message.notification;
35     AndroidNotification? android = message.notification?.android;
36     // Jika notifikasi tersedia, tampilkan dialog
37     if (notification != null && android != null) {
38       showDialog(
39         context: context,
40         builder: (_) {
41           return AlertDialog(
42             title: Text(notification.title ?? ""), // Judul dialog
43             content: SingleChildScrollView(
44               child: Column(
45                 crossAxisAlignment: CrossAxisAlignment.start,
46                 children: [Text(notification.body ?? "")], // Isi dialog
47               ),
48             ),
49           );
50         },
51       );
52     }
53   });
54   // Memanggil metode untuk mengambil token FCM perangkat
55   getToken();
56 }
57
58 // Metode untuk mendapatkan token FCM
59 void getToken() async {
60   token = await FirebaseMessaging.instance
61     .getToken(); // Mendapatkan token FCM perangkat
62   print('FCM Token: $token'); // Menampilkan token di log
63 }
```

D. Mengirim Notifikasi Via FCM Console

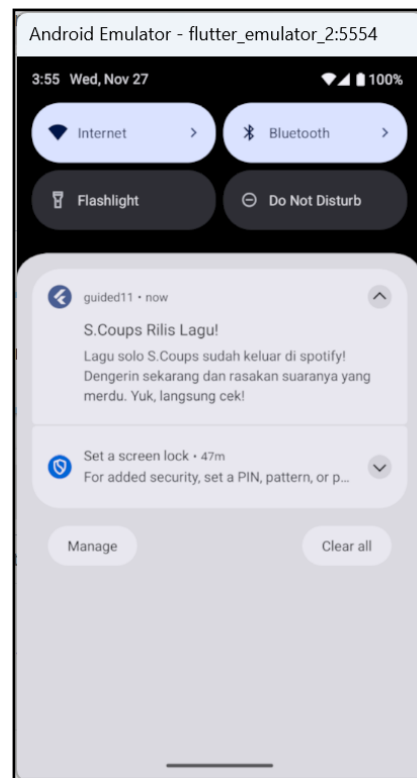
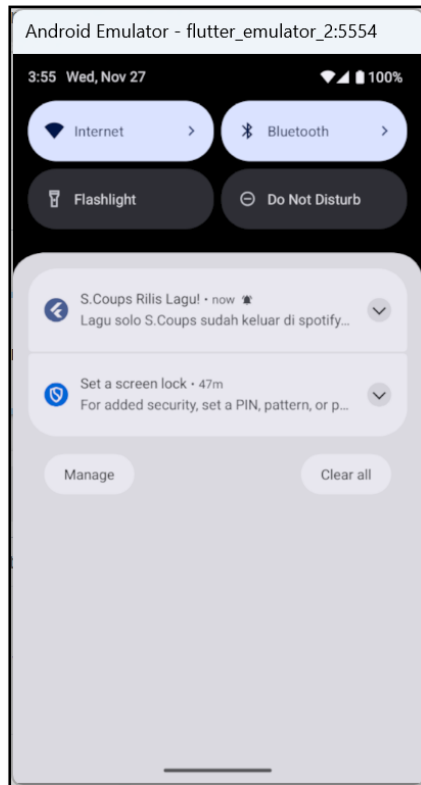
- Waktunya testing notifikasi kalian. Dimulai dengan membuka halaman all product pada tab firebase --> pilih menu cloud messaging --> lalu pilih send your first message.
- Selanjutnya, inputkan judul, pesan, dan foto dari notifikasi kalian sesuai dengan kebutuhan.



- 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.



- 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

Tugas Mandiri

1. 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. Kreativitas menjadi nilai tambah.

Jawab :

Source Code :

- Main dart

```
1 import 'package:firebase_core/firebase_core.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 import 'package:unguided/my_notification_screen.dart';
6
7 String? token;
8 Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
9   await Firebase.initializeApp();
10  print('Handling a background message: ${message.messageId}');
11 }
12
13 const AndroidNotificationChannel channel = AndroidNotificationChannel(
14   'high_importance_channel', // ID Channel
15   'High Importance Notifications', // Nama Channel
16   description:
17     'This channel is used for important notifications.', //Deskripsi Channel
18   importance: Importance.high, // Prioritas
19 );
20
21 void main() async {
22   WidgetsFlutterBinding.ensureInitialized();
23   await Firebase.initializeApp();
24   FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
25   await FlutterLocalNotificationsPlugin()
26     .resolvePlatformSpecificImplementation<
27       AndroidFlutterLocalNotificationsPlugin>()
28     ?.createNotificationChannel(channel);
29   await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
30     alert: true,
31     badge: true,
32     sound: true,
33   );
34
35   runApp(const MyApp());
36 }
37
38 class MyApp extends StatelessWidget {
39   const MyApp({super.key});
40
41   // This widget is the root of your application.
42   @override
43   Widget build(BuildContext context) {
44     return MaterialApp(
45       title: 'Flutter Demo',
46       theme: ThemeData(
47         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
48         useMaterial3: true,
49       ),
50       home: const MyNotificationScreen(),
51     );
52   },
53 }
54
```


- my_notification_screen

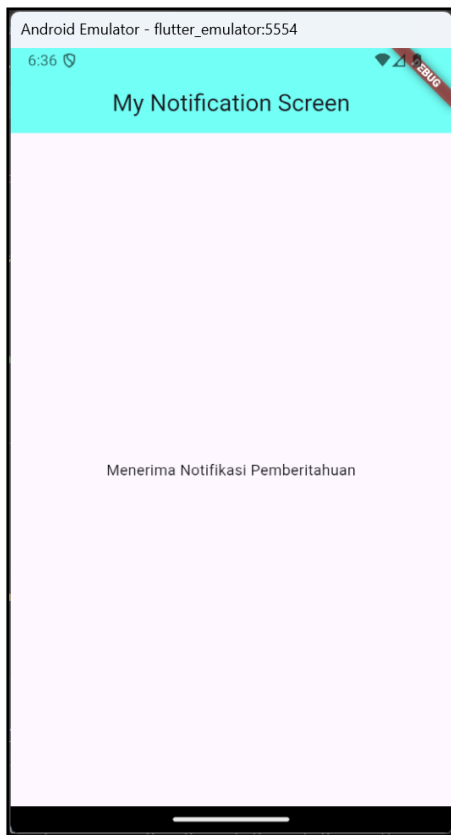
```

1 import 'package:firebase_messaging/firebase_messaging.dart';
2 import 'package:flutter/material.dart';
3 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
4 import 'package:unguided/main.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     // Mendengarkan pesan saat aplikasi aktif
24     FirebaseMessaging.onMessage.listen((RemoteMessage message) {
25       RemoteNotification? notification = message.notification;
26       AndroidNotification? android = message.notification?.android;
27       // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
28       if (notification != null && android != null) {
29         FlutterLocalNotificationsPlugin().show(
30           notification.hashCode, // ID notifikasi (hashCode untuk unik)
31           notification.title, // Judul notifikasi
32           notification.body, // Isi notifikasi
33           NotificationDetails(
34             android: AndroidNotificationDetails(
35               channel.id,
36               channel.name,
37               channelDescription: channel.description,
38               color: Colors.blue,
39               icon: "@mipmap/ic_launcher",
40             ),
41           ),
42         );
43       }
44     });
45     // Menangani aksi ketika notifikasi dibuka
46     FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
47       RemoteNotification? notification = message.notification;
48       AndroidNotification? android = message.notification?.android;
49       // Jika notifikasi tersedia, tampilkan dialog
50       if (notification != null && android != null) {
51         showDialog(
52           context: context,
53           builder: (_) {
54             return AlertDialog(
55               title: Text(notification.title ?? ""), // Judul dialog
56               content: SingleChildScrollView(
57                 child: Column(
58                   crossAxisAlignment: CrossAxisAlignment.start,
59                   children: [Text(notification.body ?? "")], // Isi dialog
60                 ),
61               ),
62             );
63           },
64         );
65       }
66     });
67     // Mengambil metode untuk mengambil token FCM perangkat
68     getToken();
69   }
70
71   // Metode untuk mendapatkan token FCM
72   void getToken() async {
73     token = await FirebaseMessaging.instance
74       .getToken(); // Mendapatkan token FCM perangkat
75     print('FCM Token: $token'); // Menampilkan token di log
76   }
77
78   @override
79   Widget build(BuildContext context) {
80     return Scaffold(
81       appBar: AppBar(
82         title: const Text("My Notification Screen"),
83         backgroundColor: const Color.fromARGB(255, 11, 255, 246),
84         centerTitle: true,
85       ),
86       body: const Center(
87         child: Text("Menerima Notifikasi Pemberitahuan"),
88       ),
89     );
90   }
91 }
92

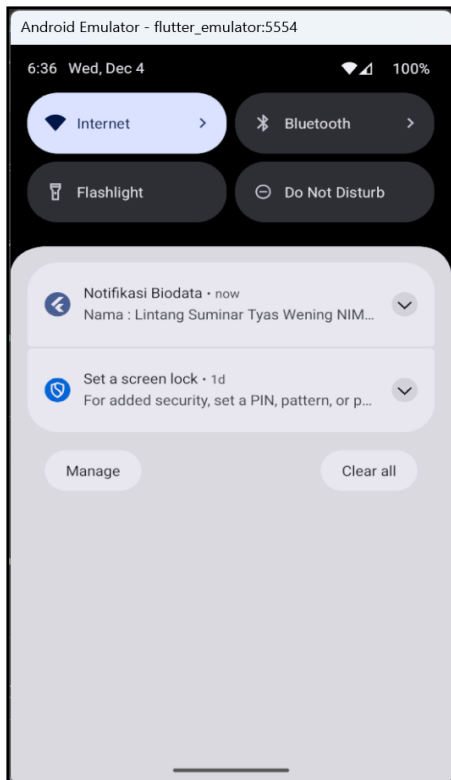
```

Screenshot Output :

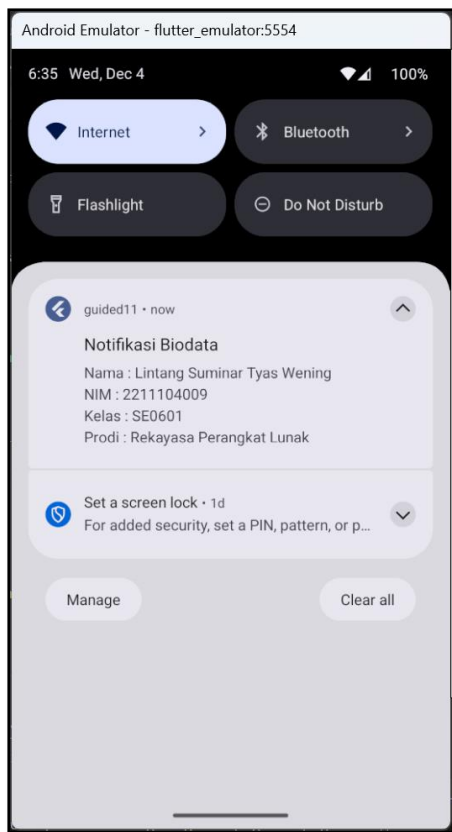
- Tampilan awal atau Halaman muka



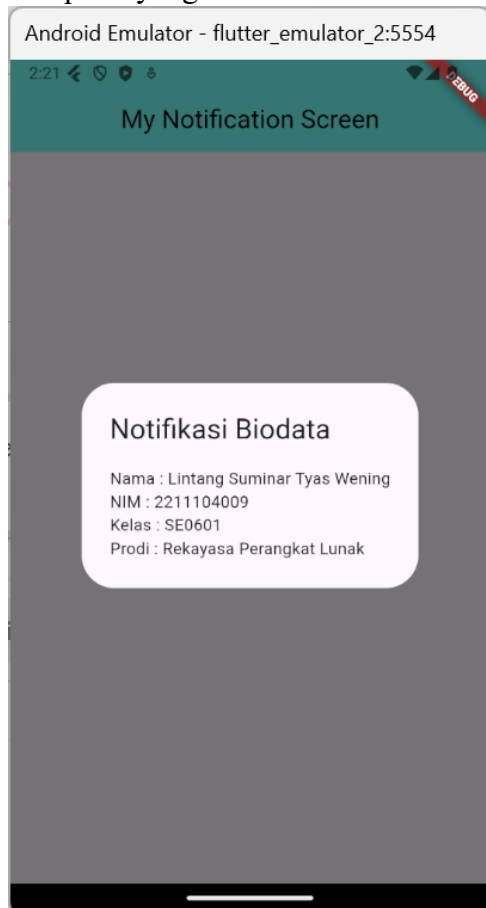
- Tampilan saat notifikasi muncul di layar



- Tampilan ketika notifikasi nya di klik ”^” untuk melihat detail notifikasi



- Tampilan yang muncul setelah sebuah notifikasi diketuk atau diklik



Deskripsi Program :

Program ini menggunakan Firebase Cloud Messaging (FCM) dan Flutter Local Notifications untuk menerima dan menampilkan notifikasi pada perangkat Android. Saat aplikasi aktif, pesan masuk ditangani oleh `FirebaseMessaging.onMessage` dan ditampilkan sebagai notifikasi lokal dengan judul, isi, dan ikon. Jika pengguna membuka notifikasi dari tray, aplikasi memprosesnya melalui `FirebaseMessaging.onMessageOpenedApp` dan menampilkan dialog dengan detail pesan.

Aplikasi juga mengambil token FCM perangkat melalui `FirebaseMessaging.instance.getToken` untuk pengiriman notifikasi yang ditargetkan. Antarmuka aplikasi sederhana, menampilkan teks "Menerima Notifikasi Pemberitahuan" di layar utama, dengan pengaturan notifikasi diinisialisasi melalui `AndroidInitializationSettings`. Program ini memerlukan konfigurasi Firebase dan izin perangkat untuk mendukung notifikasi.