

**GUIDED & UNGUIDED**  
**PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK**  
**MODUL XI**  
**DATA STORAGE BAGIAN 2**



**Disusun Oleh :**  
**Maria Nathasya Desfera Pangestu / 2211104008**  
**SE0601**

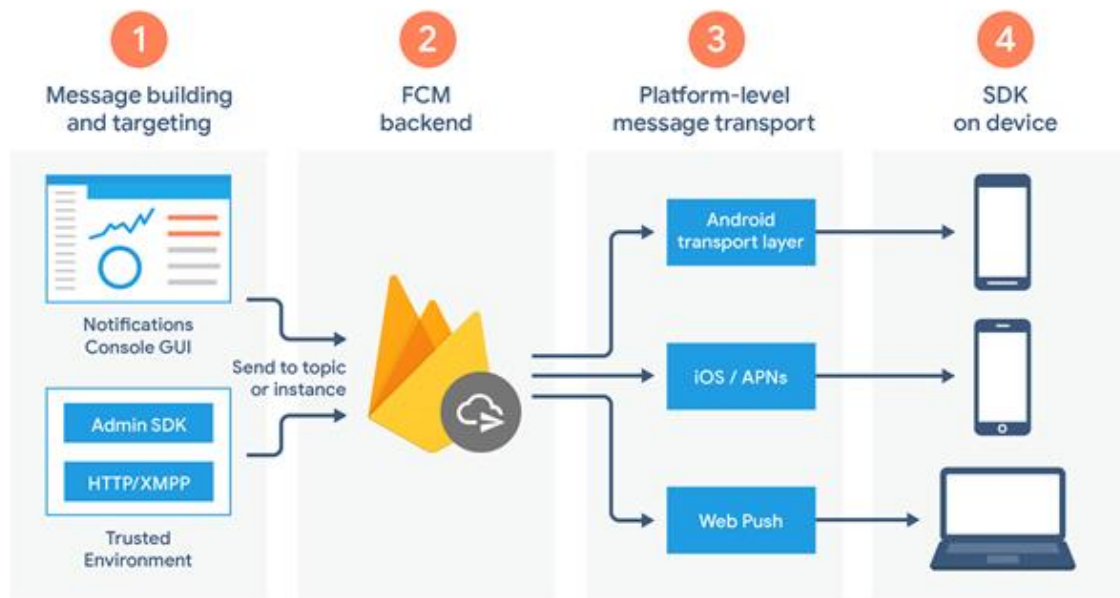
**Asisten Praktikum :**  
**Muhammad Faza Zulian Gesit Al Barru**  
**Aisyah Hasna Aulia**

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## GUIDED

### 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.

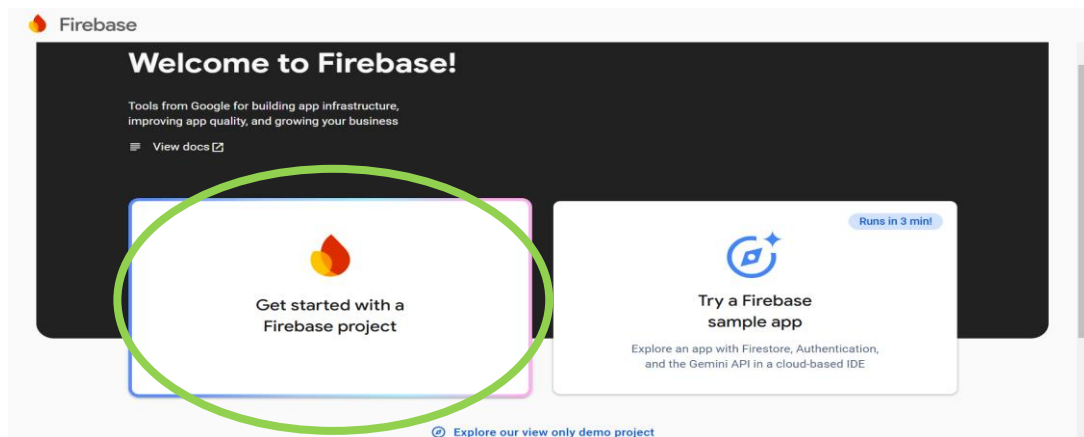
Cara kerja:

- 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:

- Pertama buka <https://console.firebase.google.com/> lalu daftar dan login pada firebase console



- Setelah itu buat projek dengan mengisi nama projek yang diinginkan.

×

Create a project

Let's start with a name for your project<sup>?</sup>

Enter your project name

my-awesome-project-id

☐ I accept the [Firebase terms](#).

☐ I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

- c. Lalu pilih lokasinya sesuai tempat kita berada yaitu Indonesia, setelah itu klik create project.

## Configure Google Analytics

Analytics location ⓘ

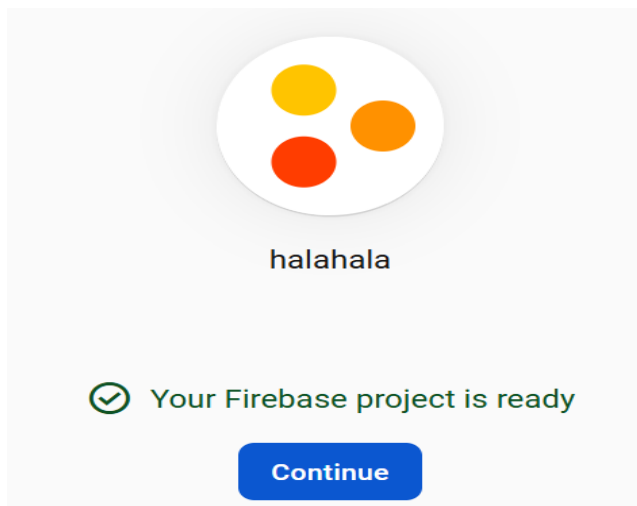
Indonesia ▼

Google Analytics is a business tool. Use it exclusively for purposes related to your trade, business, craft, or profession.

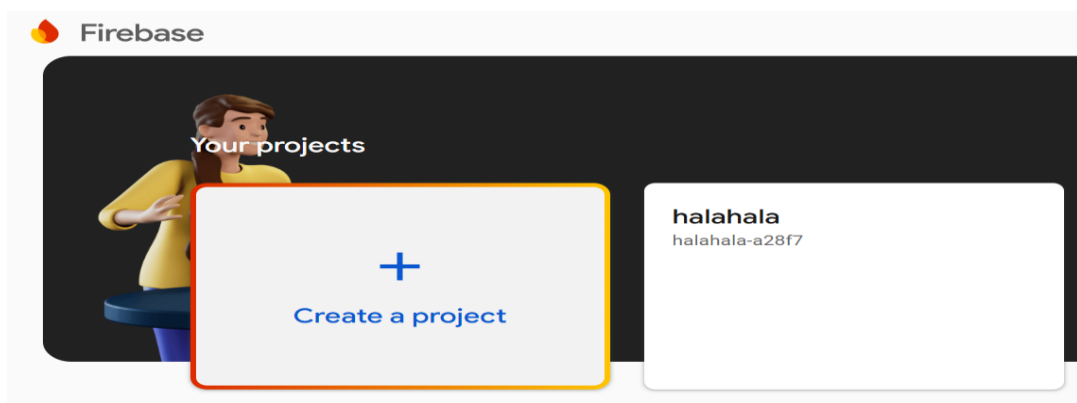
Data sharing settings and Google Analytics terms

☒ Use the default settings for sharing Google Analytics data. [Learn more](#) ⓘ

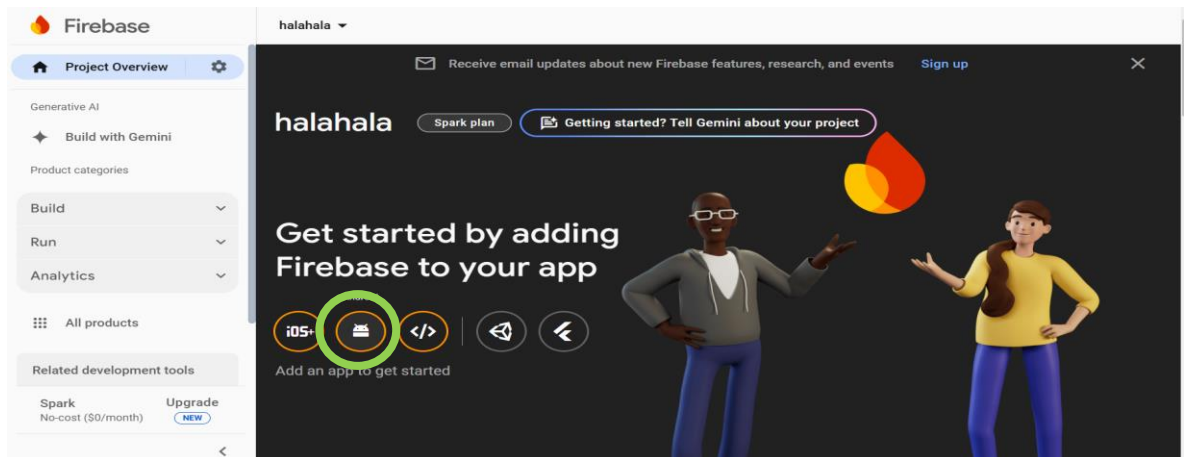
- d. Kemudian tunggu hingga proses selesai, lalu klik continue.



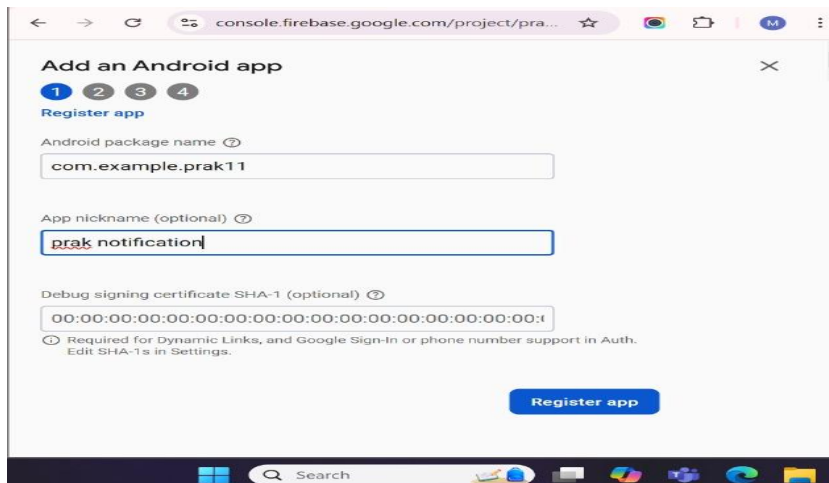
- e. Setelah itu buat sebuah projek firebase baru.



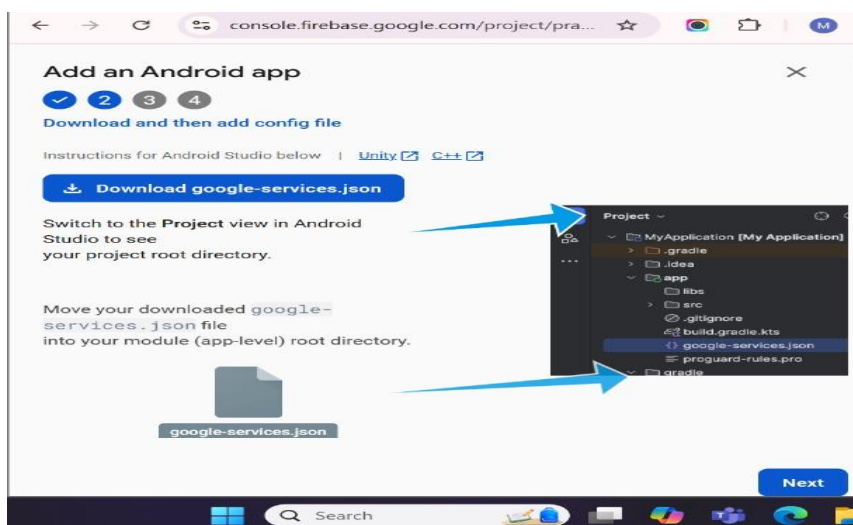
- f. Lalu pilih android, tambahkan android package name, app nickname and SHA-1 signing certificate key. Android package name dapat ditemukan di **./android/app/ build.gradle**.



- g. Kemudian lakukan langkah-langkah untuk menyambungkan firebasenya, seperti dibawah ini. Pertama isikan nama android packagenya bisa di lihat di vs code lalu sesuaikan dengan nama projek yang sudah dibuat. Untuk debug signingnya bisa di kosongkan saja dan klik register app.



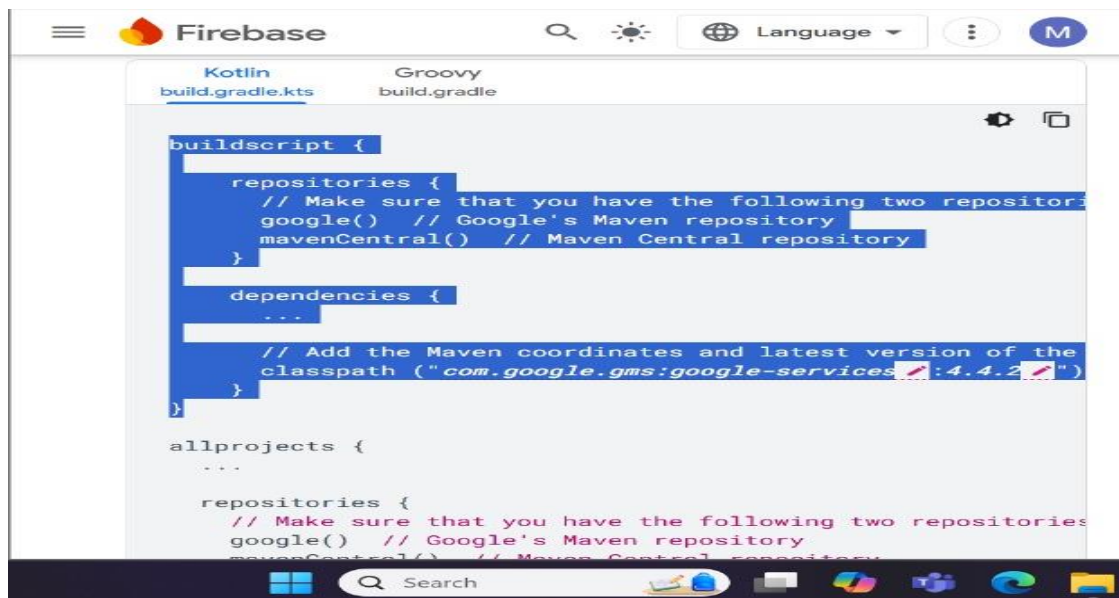
- h. Lalu klik download google-services.json setelah itu klik next, lalu simpan pada **./android/app/**.



- i. Kemudian pilih google play services plugin lalu copy.



- j. Lalu copykan ke bagian buildscript yang ada tanda pencil tersebut.



- k. 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.
- l. 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.
- m. 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

```
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

```

@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();
}

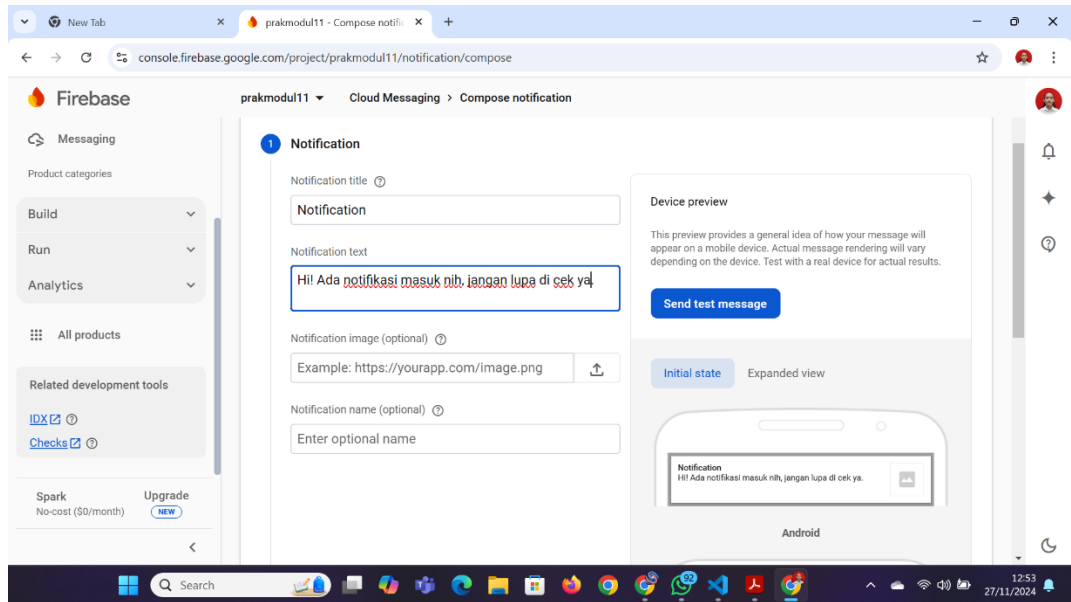
// Metode untuk mendapatkan token FCM
void getToken() async {
  token = await FirebaseMessaging.instance
    .getToken(); // Mendapatkan token FCM perangkat
  print('FCM Token: $token'); // Menampilkan token di log
}

```

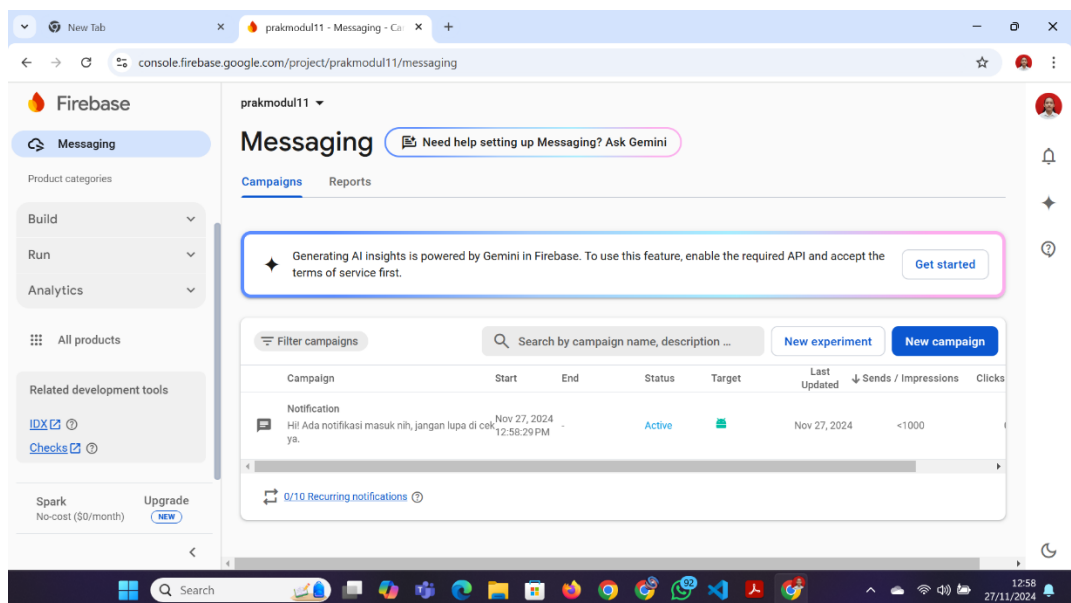


#### D. Mengirim Notifikasi Via FCM Console

- 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.



- 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.



## SOURCE CODE GUIDED

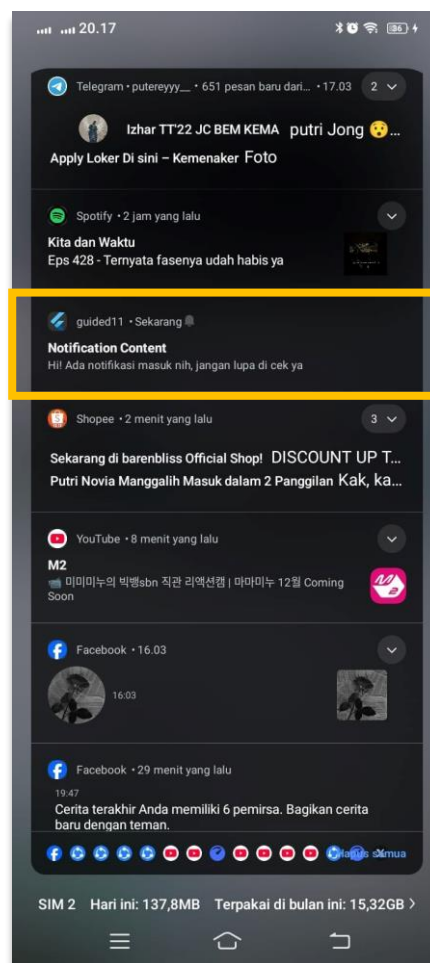
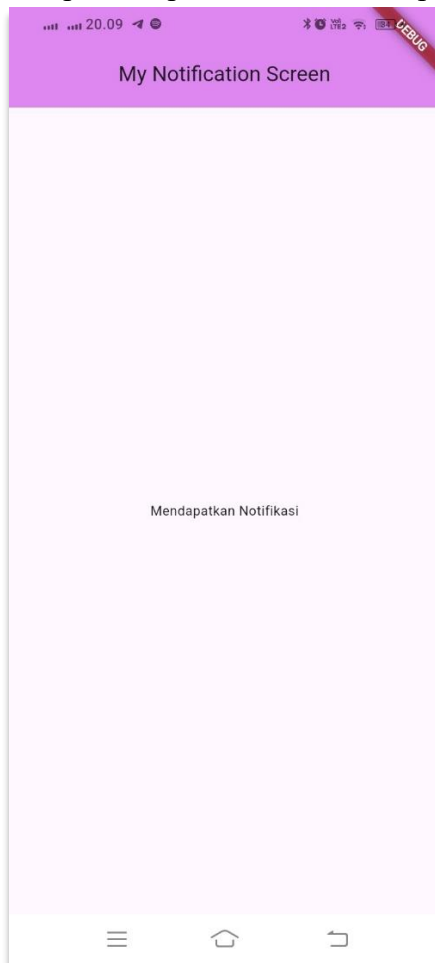
### 1. Source code main.dart

```
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:guided/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 NotificationScreen(),
51     );
52   }
53 }
```

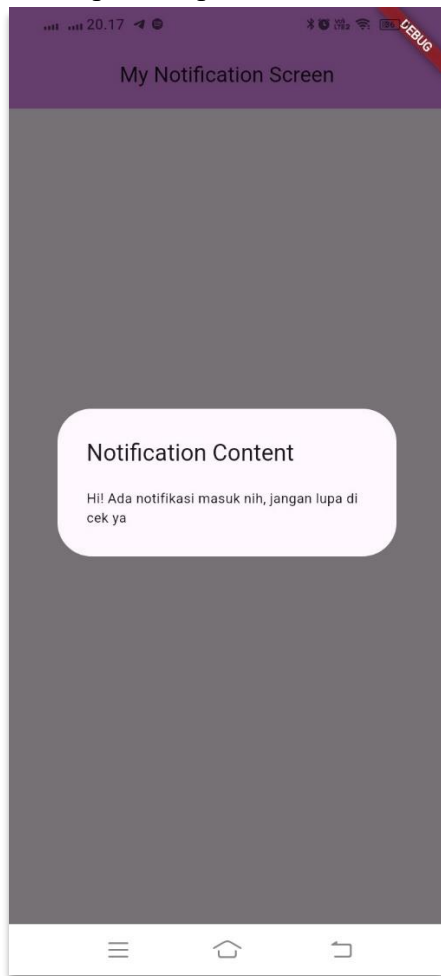
## 2. Source code notification\_screen.dart

```
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:guided/main.dart';
5
6 class NotificationScreen extends StatefulWidget {
7   const NotificationScreen({super.key});
8
9   @override
10  State<NotificationScreen> createState() => _NotificationScreenState();
11 }
12
13 class _NotificationScreenState extends State<NotificationScreen> {
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: const Color.fromARGB(255, 125, 196, 255),
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     // Memanggil 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("Notification Screen"),
83         backgroundColor: const Color.fromARGB(255, 222, 135, 240),
84         centerTitle: true,
85       ),
86       body: const Center(
87         child: Text("Mendapatkan Notifikasi"),
88       ),
89     );
90   }
91 }
```

### 3. - Output Tampilan awal dan Tampilan notifikasi masuk



- Output Tampilan notifikasi ketika di buka



## 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. 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.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 MyNotification(),
51     );
52   }
53 }
```

- Source code my\_notification.dart

```
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 MyNotification extends StatefulWidget {
7   const MyNotification({super.key});
8
9   @override
10  State<MyNotification> createState() => _MyNotificationState();
11 }
12
13 class _MyNotificationState extends State<MyNotification> {
14   @override
15   void initState() {
16     super.initState();
17     // 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: const Color.fromARGB(255, 222, 135, 240),
39               icon: "@mipmap/ic_launcher",
40             ),
41           ),
42         );
43       }
44     });
```

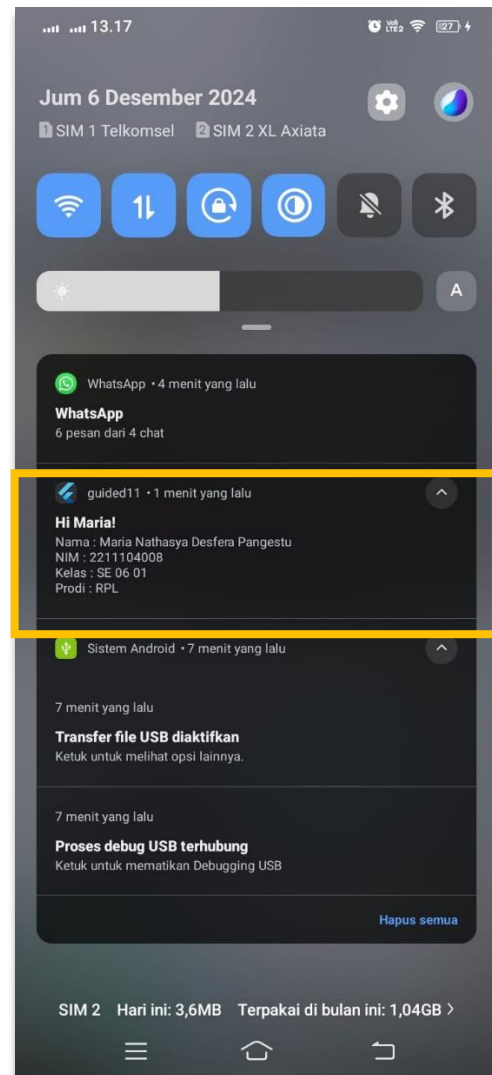
```

45 // Menangani aksi ketika notifikasi dibuka
46 FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
47     RemoteNotification? notification = message.notification;
48     AndroidNotification? android = message.notification?.android;
49     // Menampilkan dialog saat notifikasi tersedia
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 // Memanggil 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"),
83             backgroundColor: const Color.fromARGB(255, 222, 135, 240),
84             centerTitle: true,
85         ),
86         body: const Center(
87             child: Text("Mendapatkan Notifikasi"),
88         ),
89     );
90 }
91 }

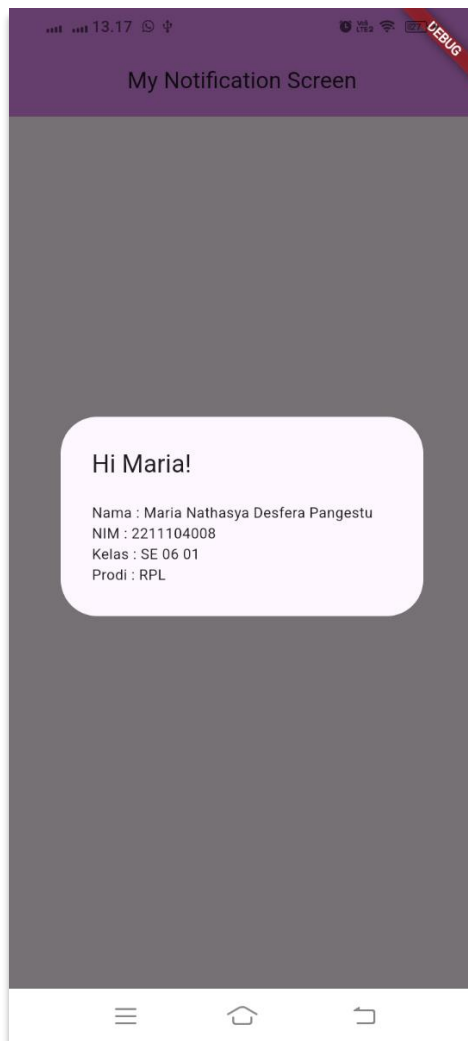
```



- Screenshot Hasil Output Tampilan awal dan saat notifikasi masuk



- Screenshot output tampilan notifikasi ketika dibuka di aplikasi flutter.



- Penjelasan  
Program tersebut untuk menerima dan menampilkan notifikasi menggunakan Firebase Cloud Messaging (FCM). Ada beberapa import seperti `firebase_messaging` untuk menangani notifikasi dari FCM dan `flutter_local_notifications` untuk menampilkan notifikasi lokal pada perangkat. Widget utama `MyNotification` dibuat sebagai widget stateful untuk mengelola perubahan status. Di `initState`, plugin notifikasi lokal diinisialisasi, dan aplikasi mendengarkan pesan dari FCM melalui `FirebaseMessaging.onMessage.listen`. Pesan yang diterima ditampilkan sebagai notifikasi lokal, sedangkan jika notifikasi dibuka, dialog dengan detail pesan ditampilkan. Metode `getToken` digunakan untuk mendapatkan token perangkat unik dari FCM. Antarmuka sederhana menampilkan judul "My Notification" dan teks "Mendapatkan Notifikasi" .