

**PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK**  
**TUGAS GUIDED & UNGUIDED**

**MODUL XI**  
**DATA STORAGE (BAGIAN 2)**  
**FIREBASE NOTIFIKASI**



**Disusun Oleh :**

Nadia Putri Rahmaniar / 2211104012

S1 SE-06-01

**Asisten Praktikum :**

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

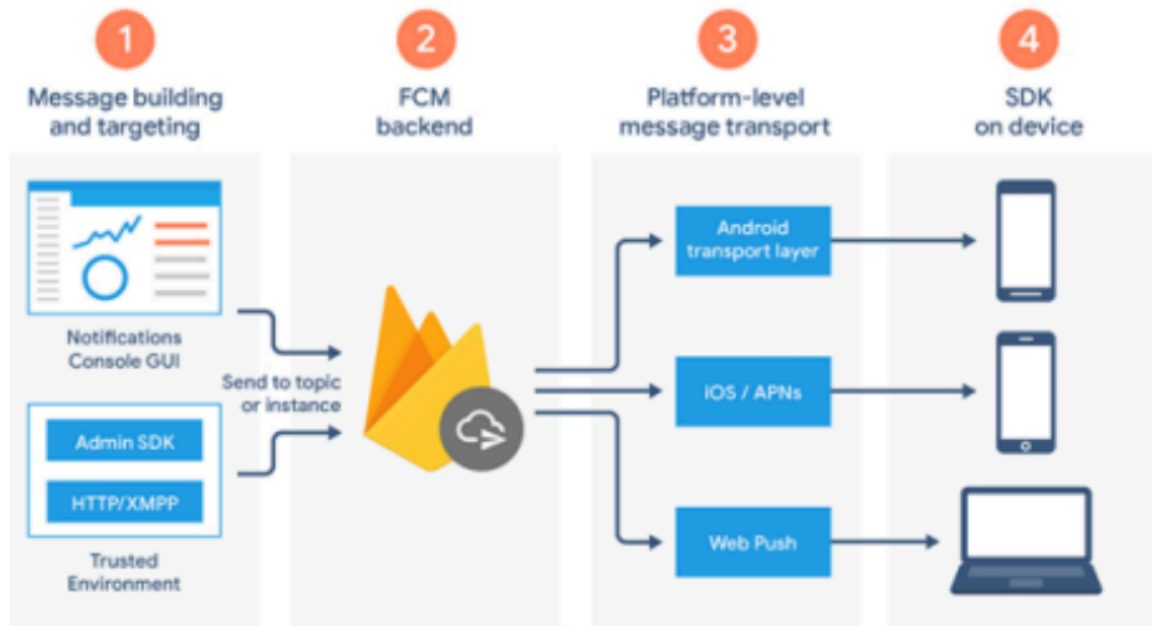
**Dosen Pengampu :**

Yudha Islami Sulistya, S.Kom., M.Cs.

**PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS**  
**INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO 2024**

## GUIDED



### 1. Pengenalan Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) adalah layanan yang memungkinkan pengiriman notifikasi secara serentak ke banyak pengguna. Dengan FCM, kita dapat mengirim pesan langsung ke perangkat tertentu atau kepada sekelompok pengguna yang berlangganan pada topik tertentu. Untuk mengirim pesan, diperlukan langkah-langkah berikut:

1. Mendapatkan token unik dari setiap perangkat pengguna.
2. Mengelompokkan pengguna berdasarkan topik tertentu, seperti "event" atau "informasi penting".

Setelah persiapan tersebut, notifikasi dapat dikirim menggunakan Firebase Cloud Functions.

### 2. Jenis Pesan pada Firebase Cloud Messaging (FCM)

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

1. Notification Message

Jenis pesan ini digunakan untuk langsung menampilkan notifikasi di perangkat pengguna. Cara kerjanya:

- a. Jika aplikasi berjalan di latar depan (foreground), pesan akan diteruskan

ke fungsi `onMessage` untuk diproses lebih lanjut.

- b. Jika aplikasi berada di latar belakang (background) atau telah ditutup (terminated), notifikasi secara otomatis akan muncul di tray perangkat tanpa memerlukan konfigurasi tambahan.

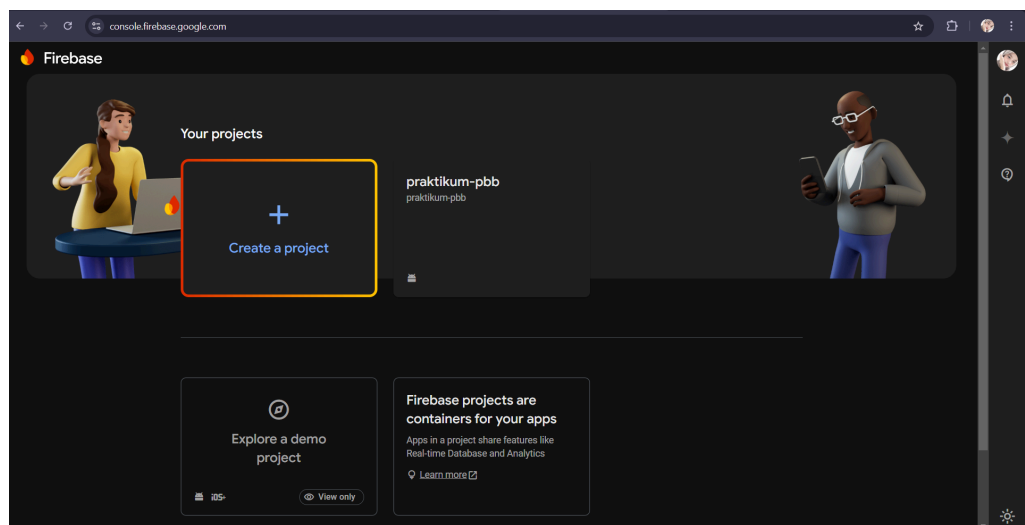
## 2. Data Message

Jenis pesan ini memberikan fleksibilitas lebih karena tidak langsung ditampilkan sebagai notifikasi. Cara kerjanya:

- a. Pesan ini diproses oleh aplikasi, baik saat aplikasi aktif maupun tidak aktif.
- b. Untuk menampilkan notifikasi kepada pengguna, diperlukan implementasi tambahan, biasanya dengan memanfaatkan plugin seperti `flutter_local_notifications`.

## 3. Integrasi Aplikasi dengan Firebase

- a. Masuk ke [Firebase Console](https://console.firebase.google.com), kemudian daftar atau login dan buat proyek baru.

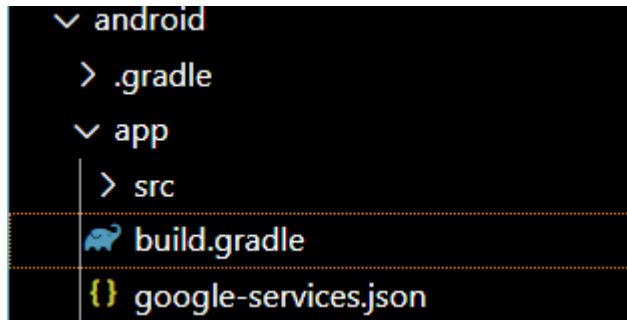


- b. Masukkan Android Package Name, App Nickname, dan SHA-1 Signing Certificate Key. Android Package Name dapat ditemukan di file `./android/app/build.gradle`.

```
android > app > build.gradle

 9  android {
10      namespace = "com.example.guide11"
11      compileSdk = flutter.compileSdkVersion
12      ndkVersion = flutter.ndkVersion
```

- c. Setelah aplikasi berhasil terdaftar, unduh file `google-services.json`. Simpan file tersebut di direktori `./android/app/`.



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

- File `./android/build.gradle`

```
android > build.gradle
1  buildscript {
2      repositories {
3          google()
4          mavenCentral()
5      }
6      dependencies {
7          classpath "com.android.tools.build:gradle:8.1.2"
8          classpath "com.google.gms:google-services:4.3.15"
9      }
10 }
```

- Menambahkan 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 after the Android and Kotlin
5      id "dev.flutter.flutter-gradle-plugin"
6      id "com.google.gms.google-services"
7  }
```

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

```
android > app > build.gradle
47  dependencies {
48      implementation platform('com.google.firebase:firebase-bom:32.2.0')
49      implementation 'com.google.firebase:firebase-analytics'
50  }
```

- e. Setelah mengubah file `build.gradle`, tambahkan paket ke file `pubspec.yaml`

dengan mengetikkan "flutter pub add firebase\_core firebase\_messaging flutter\_local\_notifications" di terminal.

```
! pubspec.yaml
30 dependencies:
31   flutter:
32     sdk: flutter
33
34
35   # The following adds the Cupertino Icons font to your application.
36   # Use with the CupertinoIcons class for iOS style icons.
37   cupertino_icons: ^1.0.8
38   firebase_core: ^3.8.0
39   firebase_messaging: ^15.1.5
40   flutter_local_notifications: ^18.0.1
```

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

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + v
PS D:\School\PPB\cobacode\guide11> flutter pub get
Resolving dependencies... (1.0s)
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)
  http_parser 4.0.2 (4.1.1 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)
  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 (0.7.4 available)
  vm_service 14.2.5 (14.3.1 available)
Got dependencies!
19 packages have newer versions incompatible with dependency constraints.
Try `flutter pub outdated` for more information.
PS D:\School\PPB\cobacode\guide11> 
```

#### 4. Integrasi Aplikasi dengan Firebase

- a. Tambahkan kode di dalam fungsi `main()` pada file `main.dart` untuk memastikan Firebase dan konfigurasi lainnya telah siap digunakan.

```
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. Di file main.dart, masukkan kode 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. Sehingga code lengkap main.dart menjadi seperti ini

```

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:guidell/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(_firebaseMessagingBack
groundHandler);
  await FlutterLocalNotificationsPlugin()
    .resolvePlatformSpecificImplementation<
      AndroidFlutterLocalNotificationsPlugin>()
    ?.createNotificationChannel(channel);

  await
  FirebaseMessaging.instance.setForegroundNotificationPresenta
tionOptions(
    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(
    debugShowCheckedModeBanner: false,
    title: 'Flutter Demo',
    theme: ThemeData(
      colorScheme: ColorScheme.fromSeed(seedColor:
Colors.yellow),
      useMaterial3: true,
    ),
    home: const MyNotificationScreen(),
  );
}

```

- d. Tambahkan konfigurasi untuk menangani notifikasi ke file `my_notification_screen.dart` di kelas State.

```

@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(initializationS
ettings);
  // 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
}

```

e. Code lengkap 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:guidell/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();

        var initializationSettingsAndroid =
        AndroidInitializationSettings('@mipmap/ic_launcher');
        var initializationSettings =

```

```

InitializationSettings(android:
initializationSettingsAndroid);

FlutterLocalNotificationsPlugin().initialize(initializationS
ettings);
// 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

```

```

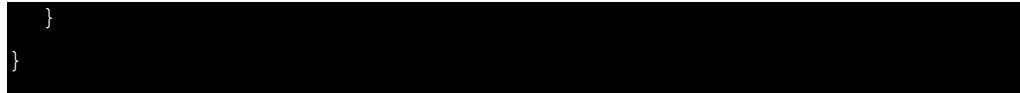
        if (notification != null && android != null) {
            showDialog(
                context: context,
                builder: (_) {
                    return AlertDialog(
                        title: Text(notification.title ?? ""), //
Judul dialog
                        content: SingleChildScrollView(
                            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
}

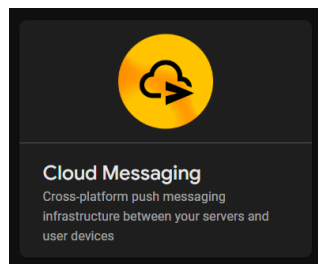
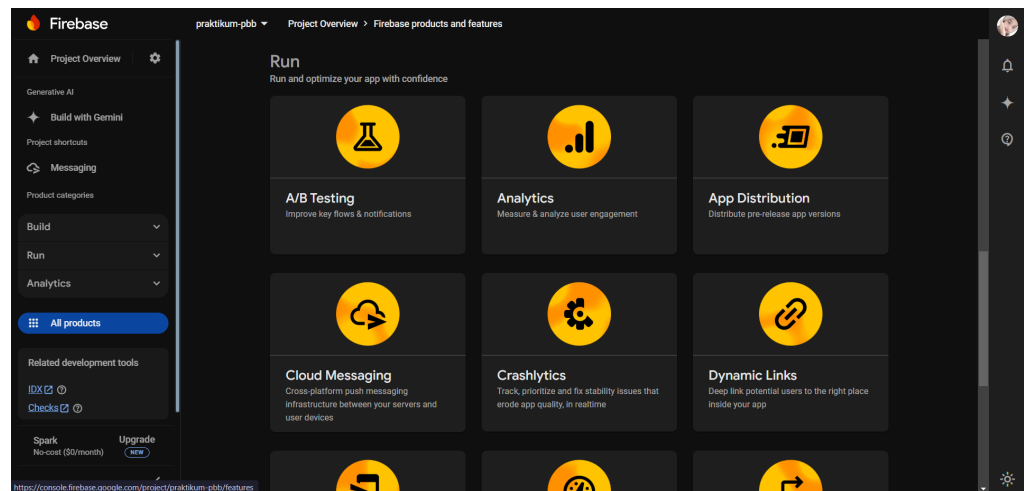
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text('My Notification Screen'),
            backgroundColor: Colors.amber,
            centerTitle: true,
        ),
        body: Center(
            child: Text('Mendapatkan Notifikasi'),
   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

```



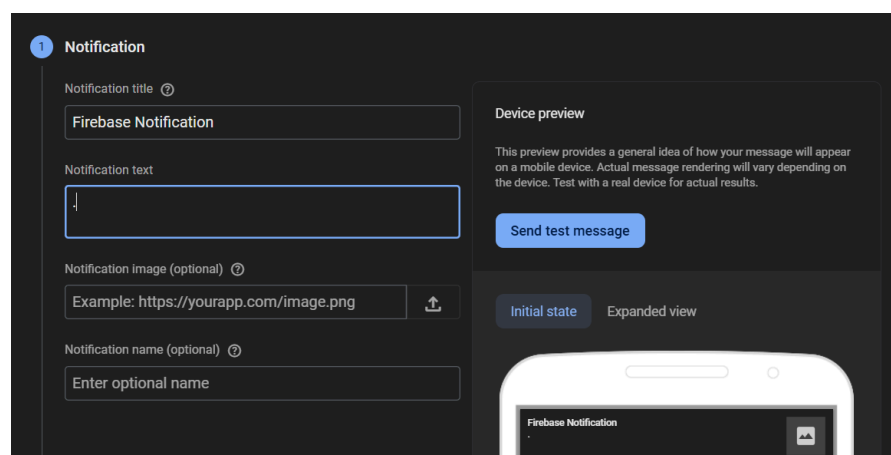
## 5. Mengirim Notifikasi Via FCM Console

- a. Langkah berikutnya adalah menguji notifikasi. Buka halaman All Product di tab Firebase, lalu pilih menu Cloud Messaging dan klik opsi Send Your First Message.



- b. Berikutnya, masukkan judul, pesan, dan untuk gambar itu opsional sesuai dengan kebutuhan.

- Judul



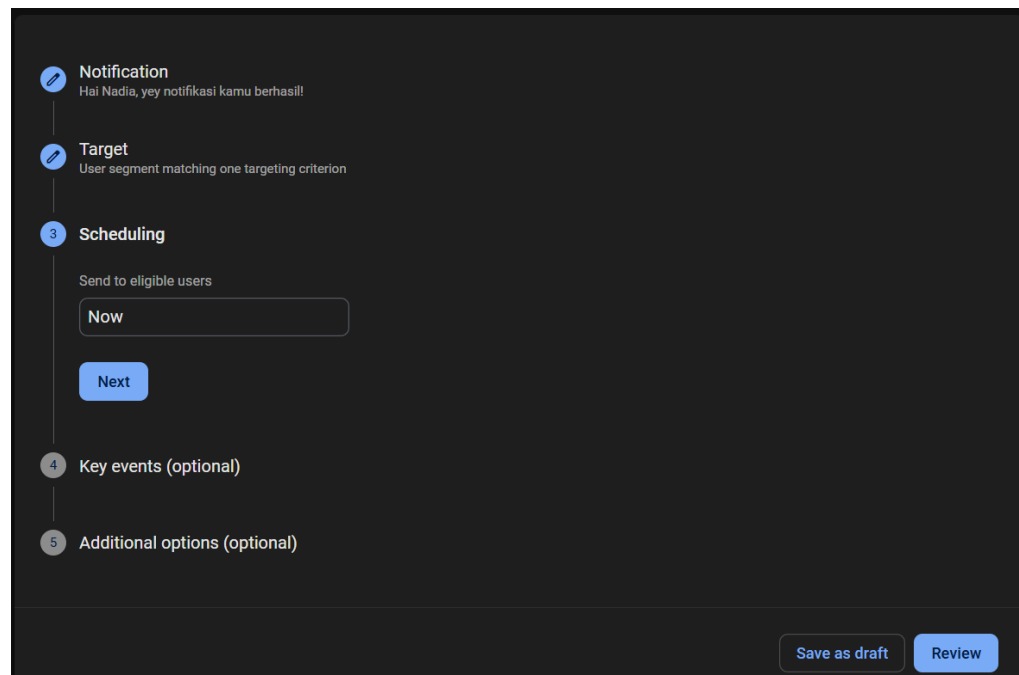
- Pesan

The screenshot shows the 'Notification' configuration screen in the Firebase console. It includes fields for 'Notification title' (set to 'Firebase Notification'), 'Notification text' (set to 'Hai Nadia, yey notifikasi kamu berhasil!'), 'Notification image (optional)' (with a placeholder URL and an upload icon), and 'Notification name (optional)' (with a placeholder 'Enter optional name'). On the right, there is a 'Device preview' section with a 'Send test message' button and a visual representation of the notification on a mobile device. Below the preview, there are tabs for 'Initial state' and 'Expanded view'.

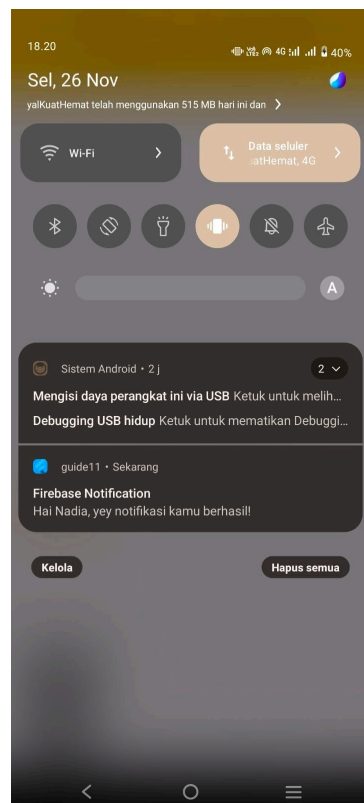
c. Kemudian, pilih aplikasi yang akan menerima notifikasi

The screenshot shows the 'Target' configuration screen in the Firebase console. It has two tabs: 'User segment' and 'Topic'. Under 'User segment', there is a 'Target user if...' section with a dropdown menu set to 'App' and a text input field containing 'notifications (com.example.guide11)'. Below this is a 'Next' button. On the left side, there is a vertical list of steps: '2 Target', '3 Scheduling' (with a sub-option 'Send now'), '4 Key events (optional)', and '5 Additional options (optional)'. At the bottom right, there are 'Save as draft' and 'Review' buttons.

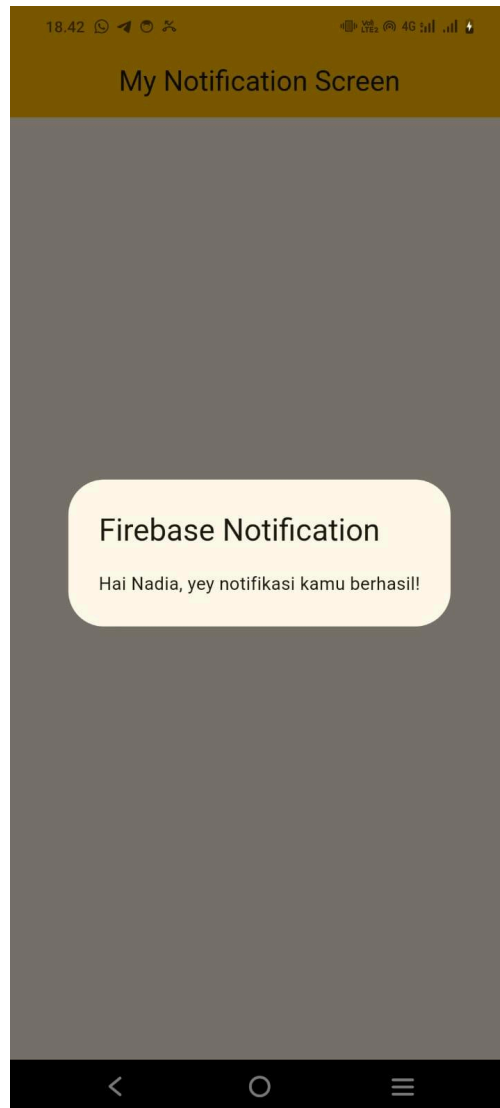
- d. Atur waktu pengiriman ke "Now" jika ingin notifikasi dikirim segera. Setelah semua isi dan pengaturan notifikasi sudah sesuai, klik Review lalu klik Publish.



- e. Tunggu hingga notifikasi muncul di aplikasi. Perlu dicatat bahwa pengiriman notifikasi mungkin membutuhkan waktu beberapa saat dan tidak akan langsung muncul setelah dikirim, jadi harap bersabar.



- f. Klik notifikasi, dan Anda akan diarahkan ke aplikasi, yang kemudian akan menampilkan tampilan seperti yang terlihat pada gambar di bawah ini.





## 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, screenshot 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:guidell/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(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.yellow),
        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:guidell/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
                        ),
                    ),
                );
            },
        );
    }
});

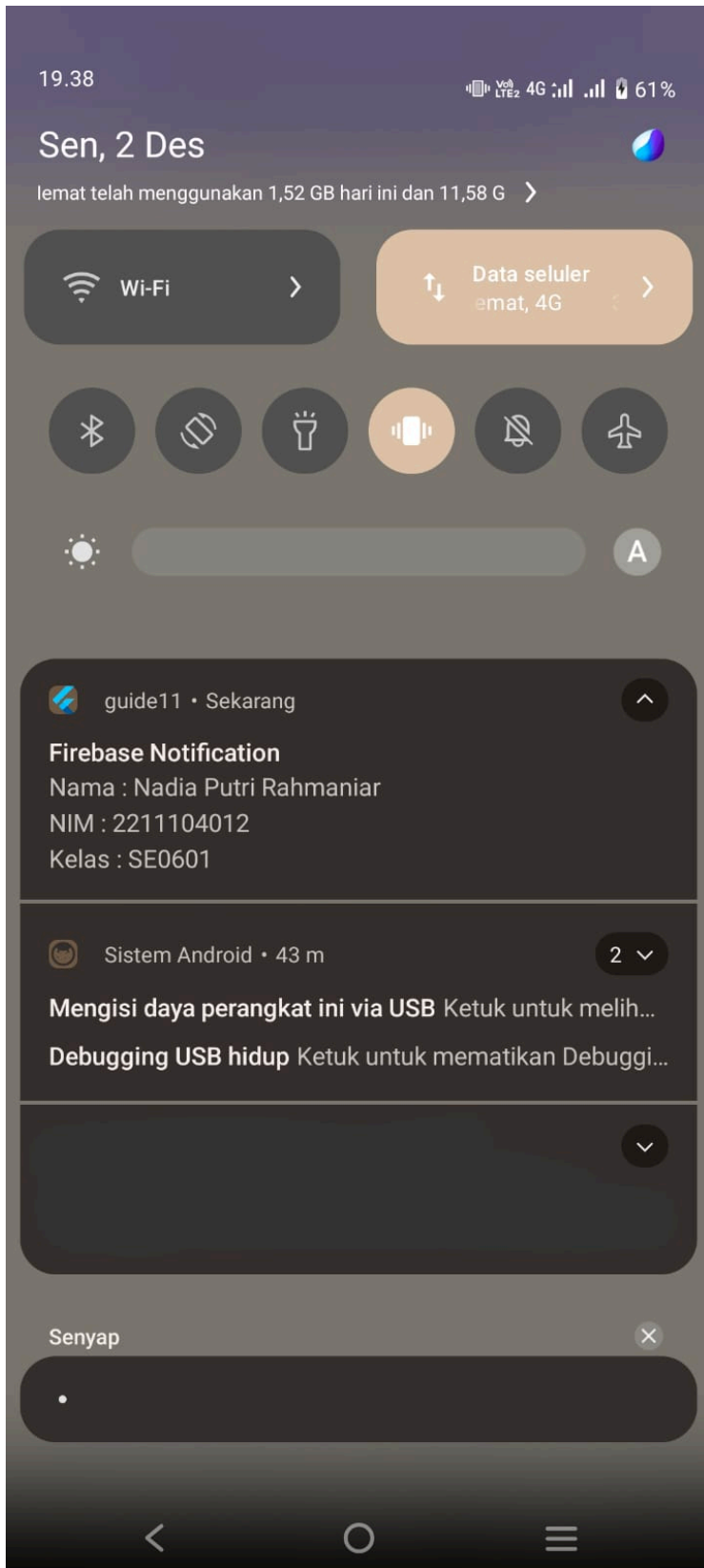
// Memanggil metode untuk mengambil token FCM perangkat

```

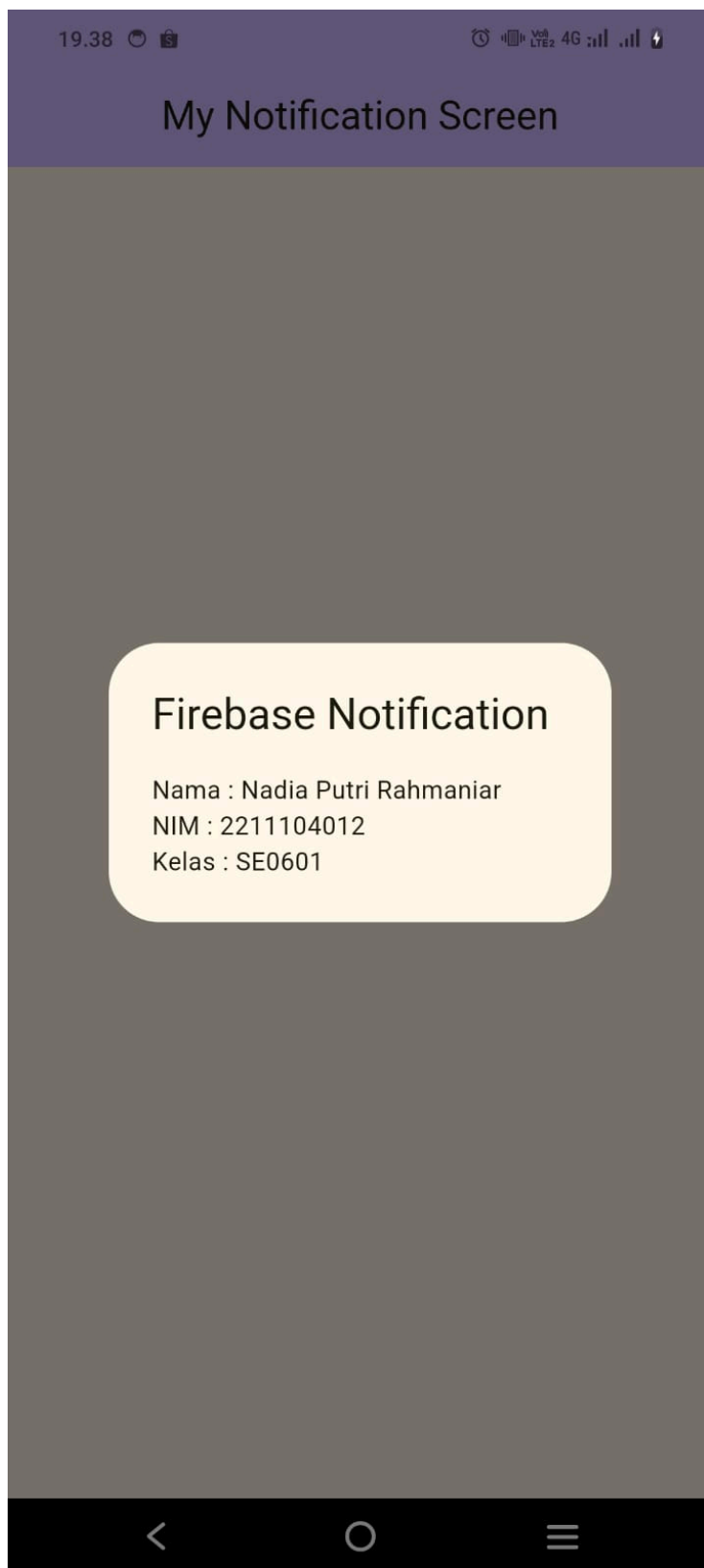
```
    getToken();  
  }  
  
  // Metode untuk mendapatkan token FCM  
  void getToken() async {  
    token = await FirebaseMessaging.instance  
      .getToken(); // Mendapatkan token FCM perangkat  
    print('FCM Token: $token'); // Menampilkan token di log  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text("My Notification Screen"),  
        backgroundColor: const Color.fromARGB(255, 205, 190, 255),  
        centerTitle: true,  
      ),  
      body: const Center(  
        child: Text("Mendapatkan Notifikasi"),  
      ),  
    );  
  }  
}
```

## Output

Tampilan saat muncul notifikasi



Tampilan setelah notifikasi di klik



## **Deskripsi Program**

Program di atas memanfaatkan Firebase Cloud Messaging (FCM) untuk mengelola pengiriman dan penerimaan notifikasi. Firebase mengirimkan pesan ke perangkat tujuan menggunakan token FCM, yang dapat diperoleh melalui `FirebaseMessaging.instance.getToken()`.

Ketika aplikasi berjalan dalam kondisi aktif (foreground), pesan yang diterima akan ditampilkan menggunakan `flutter_local_notifications`. Namun, jika aplikasi berada dalam mode tidak aktif (background atau terminated), notifikasi akan otomatis muncul di layar perangkat tanpa perlu penulisan kode tambahan.

Pesan dikirimkan melalui Firebase Console dengan judul "Notifikasi Aplikasi". Ketika pengguna mengetuk notifikasi, mereka akan diarahkan kembali ke aplikasi untuk melihat detail pesan yang berisi informasi berikut:

- Nama : Nadia Putri Rahmانيar
- NIM : 2211104012
- Kelas : SE0601