

PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK

MODUL XI

FIREBASE NOTIFIKASI



Disusun Oleh:

Aorinka Anendya Chazanah / 2211104013

S1 SE-06-01

Asisten Praktikum:

MuhammadFazaZulian 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

Firestore Cloud Messaging (FCM) adalah layanan yang mempermudah pengiriman notifikasi ke banyak pengguna secara bersamaan. Dengan FCM, kita bisa mengirim pesan ke perangkat individu maupun ke kelompok pengguna yang berlangganan topik tertentu.

A. Jenis Pesan pada Firestore Cloud Messaging (FCM)

Firestore Cloud Messaging (FCM) menyediakan dua jenis pesan utama untuk kebutuhan notifikasi:

1. Notification Message

Jenis pesan ini dirancang untuk langsung menampilkan notifikasi di perangkat pengguna.

- **Ketika aplikasi berjalan di latar depan (foreground):** Pesan akan diteruskan ke fungsi `onMessage` untuk diproses lebih lanjut.
- **Ketika aplikasi berada di latar belakang (background) atau telah dimatikan (terminated):** Notifikasi otomatis muncul di tray notifikasi perangkat tanpa memerlukan kode tambahan.

2. Data Message

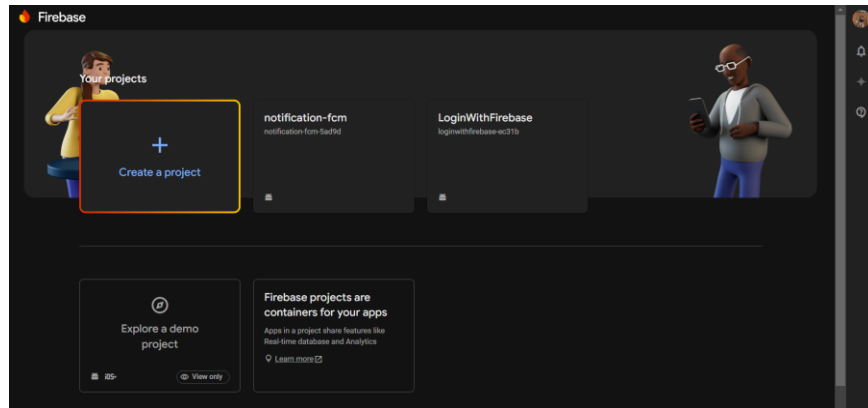
Pesan ini memberikan fleksibilitas lebih besar karena tidak langsung muncul sebagai notifikasi.

- Pesan akan diterima dan diproses oleh aplikasi, baik saat aplikasi aktif maupun tidak aktif.
- Untuk menampilkan notifikasi kepada pengguna, perlu ditambahkan kode khusus, biasanya dengan bantuan plugin seperti `flutter_local_notifications`.

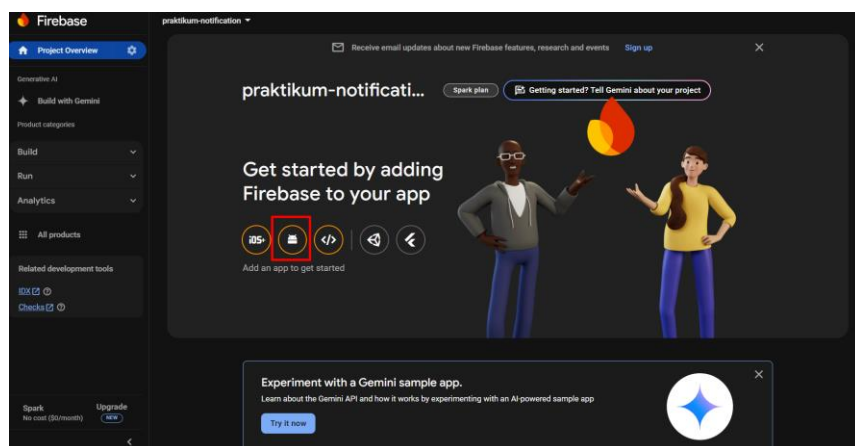
A. Integrasi Aplikasi dengan Firestore

Langkah awal untuk memulai adalah dengan membuat akun di Firestore. Berikut adalah tahapan yang perlu dilakukan untuk menyiapkan Google Firestore agar dapat digunakan untuk notifikasi push di Android:

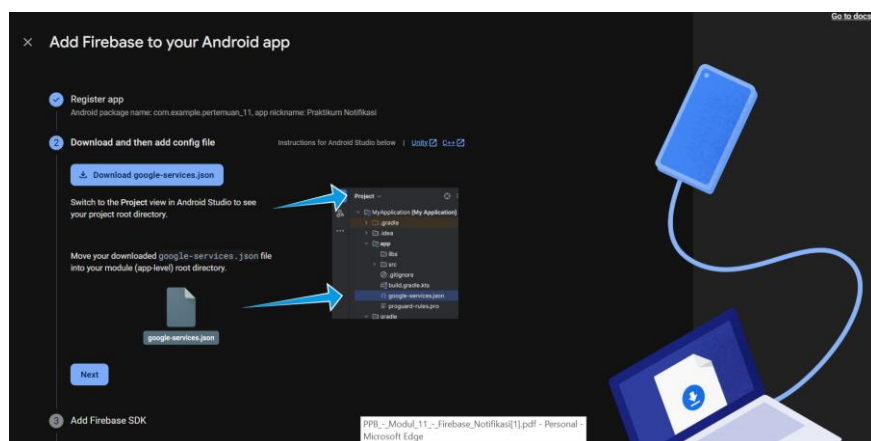
- a. Masuk ke Firestore Console, lalu daftar atau login, dan buat proyek baru.



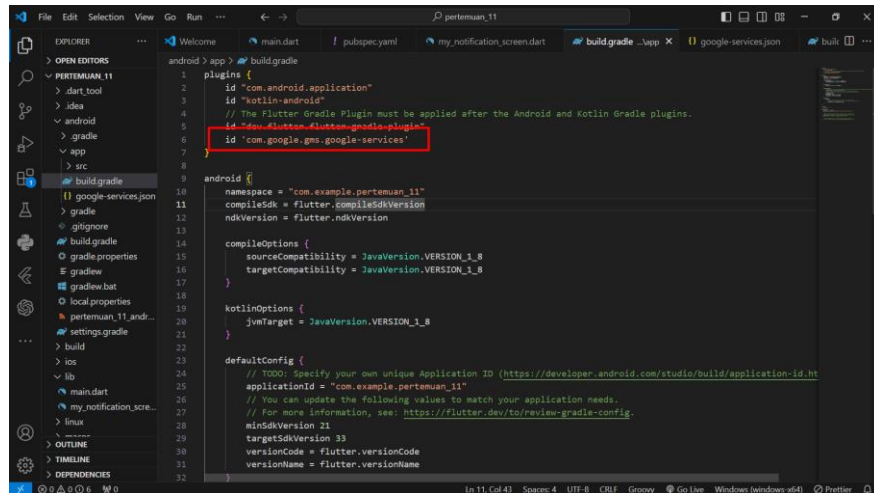
- b. Tambahkan android package name, app nickname and SHA-1 signing certificate key. Android package name dapat ditemukan di `./android/app/build.gradle`.



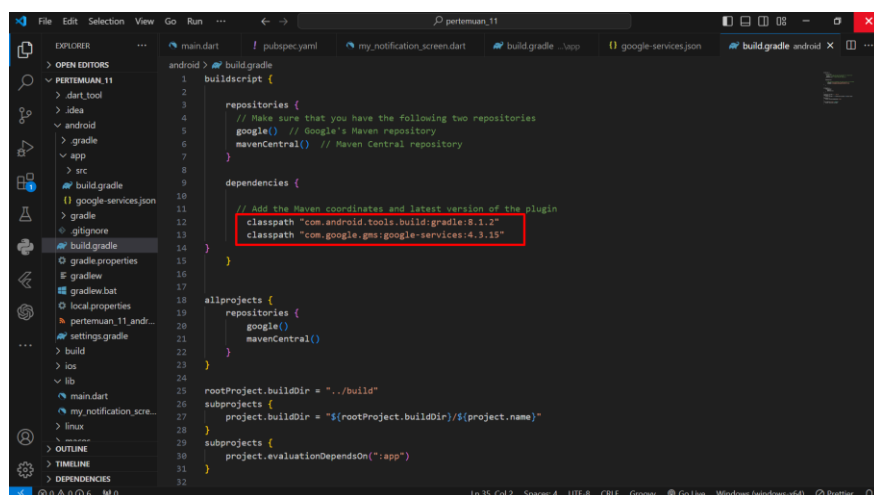
- c. Setelah berhasil mendaftarkan aplikasi ke firebase, download file `google-services.json` lalu simpan pada `./android/app/`



- 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.



```
1 plugins {
2     id "com.android.application"
3     id "kotlin-android"
4     // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
5     id "dev.flutter.flutter-gradle-plugin"
6     id "com.google.gms.google-services"
7 }
8
9 android {
10     namespace = "com.example.pertemuan_11"
11     compileSdk = flutter.compileSdkVersion
12     ndkVersion = flutter.ndkVersion
13
14     compileOptions {
15         sourceCompatibility = JavaVersion.VERSION_1_8
16         targetCompatibility = JavaVersion.VERSION_1_8
17     }
18
19     kotlinOptions {
20         jvmTarget = JavaVersion.VERSION_1_8
21     }
22
23     defaultConfig {
24         // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html)
25         applicationId = "com.example.pertemuan_11"
26         // You can update the following values to match your application needs.
27         // For more information, see: https://flutter.dev/to/review-gradle-config.
28         minSdkVersion 21
29         targetSdkVersion 33
30         versionCode = flutter.versionCode
31         versionName = flutter.versionName
32     }
33 }
```



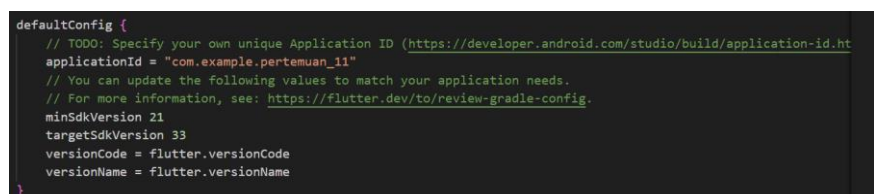
```
1 buildscript {
2     repositories {
3         // Make sure that you have the following two repositories
4         google() // Google's Maven repository
5         mavenCentral() // Maven Central repository
6     }
7
8     dependencies {
9         // Add the Maven coordinates and latest version of the plugin
10         classpath "com.android.tools.build:gradle:8.1.2"
11         classpath "com.google.gms:google-services:4.3.15"
12     }
13 }
14
15 allprojects {
16     repositories {
17         google()
18         mavenCentral()
19     }
20 }
21
22 rootProject.buildDir = "../build"
23 subprojects {
24     project.buildDir = "${rootProject.buildDir}/${project.name}"
25 }
26 subprojects {
27     project.evaluationDependsOn(":app")
28 }
29
30 dependencies {
31     implementation flutter.dependencies
32 }
```

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



```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.8
firebase_core: ^3.8.0
firebase_messaging: ^15.1.5
flutter_local_notifications: ^18.0.1
```

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



```
1 defaultConfig {
2     // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html)
3     applicationId = "com.example.pertemuan_11"
4     // You can update the following values to match your application needs.
5     // For more information, see: https://flutter.dev/to/review-gradle-config.
6     minSdkVersion 21
7     targetSdkVersion 33
8     versionCode = flutter.versionCode
9     versionName = flutter.versionName
10 }
```

B. Membuat Notifikasi Handler

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

```
WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp();

// Konfigurasi handler untuk pesan di background
FirebaseMessaging.onBackgroundMessage(_firebaseMe
ssagingBackgroundHandler);

// Konfigurasi Flutter Local Notifications
final FlutterLocalNotificationsPlugin
flutterLocalNotificationsPlugin =
    FlutterLocalNotificationsPlugin();

await flutterLocalNotificationsPlugin
    .resolvePlatformSpecificImplementation<
        AndroidFlutterLocalNotificationsPlugin>()
    ?.createNotificationChannel(channel);

// Konfigurasi untuk menampilkan notifikasi di
foreground
await
FirebaseMessaging.instance.setForegroundNotificatio
nPresentationOptions(
    alert: true,
    badge: true,
    sound: true,
);
```

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

```
String? token;

// Handler untuk pesan yang diterima saat aplikasi
di background
Future<void>
_firebaseMessagingBackgroundHandler(RemoteMessage
message) async {
    await Firebase.initializeApp();
    print('Handling a background message:
    ${message.messageId}');
```

```

}

// Konfigurasi channel notifikasi
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();

  var initializationSettingsAndroid =
    const
AndroidInitializationSettings('@mipmap/ic_launcher'
);
  var initializationSettings =
    InitializationSettings(android:
initializationSettingsAndroid);
  FlutterLocalNotificationsPlugin().initialize(in
itizationSettings);

  // Mendengarkan pesan saat aplikasi aktif
  FirebaseMessaging.onMessage.listen((RemoteMessa
ge 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
                        ),
                    ),
                );
            },
        );
    }
});

```

```

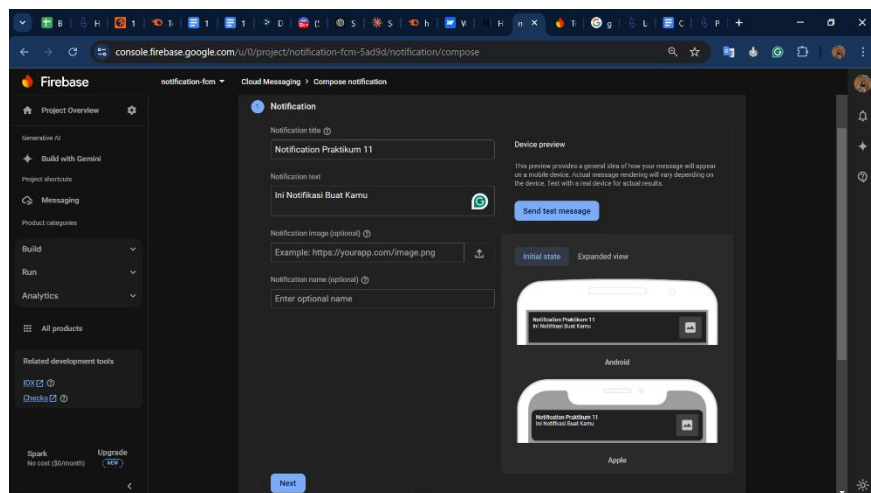
// Memanggil metode untuk mengambil token FCM
perangkat
getToken();
}

// 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
}

```

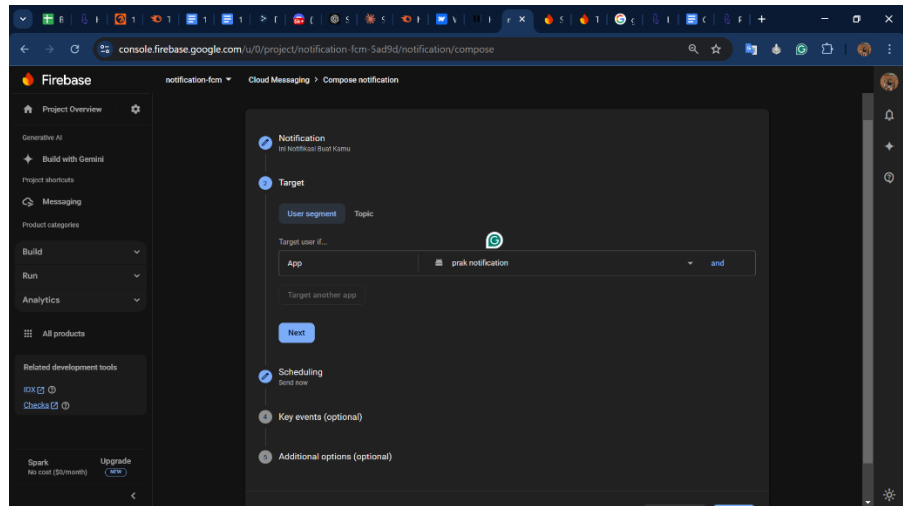
C. Mengirim Notifikasi Via FCM Console

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

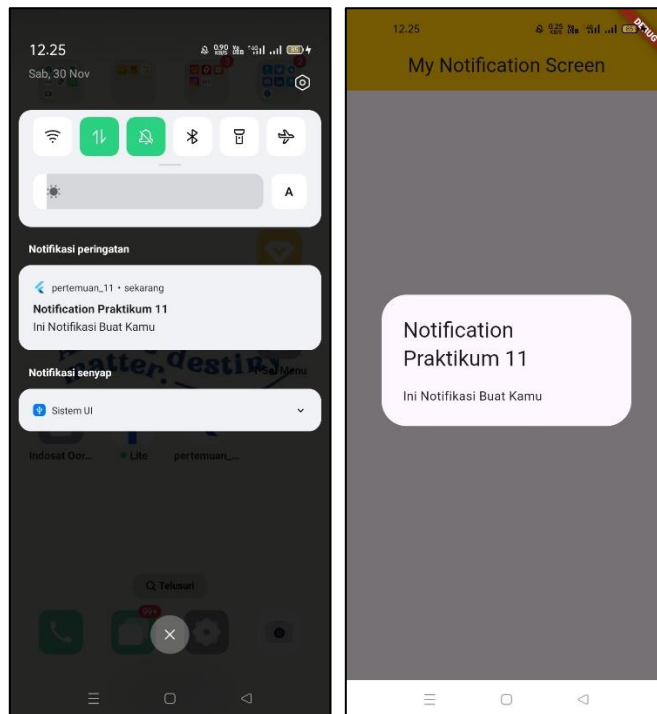


- Setelah itu, pilih aplikasi mana yang akan dikirimkan notifikasi dan setting waktu "Now" jika ingin langsung mengirimkan notifikasi.

Lalu klik publish jika isi dan settingan notifikasi sudah benar.



- d. Silakan tunggu notifikasi akan muncul pada aplikasi, di sini aplikasi di run pada perangkat pribadi. Apabila notifikasi di klik maka akan muncul detail halaman My Notification Screen.



UNGUIDED

Soal

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.

Jawaban

Source Code

1. Lengkapi android/app/build/gradle seperti berikut.

```
plugins {
    id "com.android.application"
    id "kotlin-android"
    // The Flutter Gradle Plugin must be applied after the
    Android and Kotlin Gradle plugins.
    id "dev.flutter.flutter-gradle-plugin"
    id 'com.google.gms.google-services'
}

android {
    namespace = "com.example.pertemuan_11"
    compileSdk = flutter.compileSdkVersion
    ndkVersion = flutter.ndkVersion

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget = JavaVersion.VERSION_1_8
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID
        (https://developer.android.com/studio/build/application-
        id.html).
        applicationId = "com.example.pertemuan_11"
        // You can update the following values to match your
        application needs.
        // For more information, see:
        https://flutter.dev/to/review-gradle-config.
        minSdkVersion 21
    }
}
```

```

        targetSdkVersion 33
        versionCode = flutter.versionCode
        versionName = flutter.versionName
    }

    buildTypes {
        release {
            // TODO: Add your own signing config for the
            // release build.
            // Signing with the debug keys for now, so
            // flutter run --release works.
            signingConfig = signingConfigs.debug
        }
    }
}

flutter {
    source = "../.."
}

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'

    // Add the dependencies for any other desired Firebase
    // products
    //
    // https://firebase.google.com/docs/android/setup#available-
    // libraries
}

```

2. Lengkapi build gradle pada root seperti berikut.

```

buildscript {

    repositories {
        // Make sure that you have the following two
        // repositories
    }
}

```

```

        google() // Google's Maven repository
        mavenCentral() // Maven Central repository
    }

    dependencies {

        // Add the Maven coordinates and latest version of the
plugin
        classpath "com.android.tools.build:gradle:8.1.2"
        classpath "com.google.gms:google-services:4.3.15"
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}

rootProject.buildDir = "../build"
subprojects {
    project.buildDir =
"${rootProject.buildDir}/${project.name}"
}
subprojects {
    project.evaluationDependsOn(":app")
}

tasks.register("clean", Delete) {
    delete rootProject.buildDir
}

```

3. Tambahkan package pada pubspec.yaml.

```

name: pertemuan_11
description: "A new Flutter project."
# The following line prevents the package from being
accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for
private packages.
publish_to: 'none' # Remove this line if you wish to publish
to pub.dev

# The following defines the version and build number for
your application.

```

```
# A version number is three numbers separated by dots, like
1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden
in flutter
# build by specifying --build-name and --build-number,
respectively.
# In Android, build-name is used as versionName while build-
number used as versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString
while build-number is used as CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major, minor, and
patch parts
# of the product and file versions while build-number is
used as the build suffix.
version: 1.0.0+1

environment:
  sdk: ^3.5.3

# Dependencies specify other packages that your package
needs in order to work.
# To automatically upgrade your package dependencies to the
latest versions
# consider running `flutter pub upgrade --major-versions`.
Alternatively,
# dependencies can be manually updated by changing the
version numbers below to
# the latest version available on pub.dev. To see which
dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your
application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.8
```

```
firebase_core: ^3.8.0
firebase_messaging: ^15.1.5
flutter_local_notifications: ^18.0.1

dev_dependencies:
  flutter_test:
    sdk: flutter

  # The "flutter_lints" package below contains a set of
  recommended lints to
  # encourage good coding practices. The lint set provided
  by the package is
  # activated in the `analysis_options.yaml` file located at
  the root of your
  # package. See that file for information about
  deactivating specific lint
  # rules and activating additional ones.
  flutter_lints: ^4.0.0

# For information on the generic Dart part of this file, see
the
# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter packages.
flutter:

  # The following line ensures that the Material Icons font
  is
  # included with your application, so that you can use the
  icons in
  # the material Icons class.
  uses-material-design: true

  # To add assets to your application, add an assets
  section, like this:
  # assets:
  #   - images/a_dot_burr.jpeg
  #   - images/a_dot_ham.jpeg

  # An image asset can refer to one or more resolution-
  specific "variants", see
  # https://flutter.dev/to/resolution-aware-images

  # For details regarding adding assets from package
  dependencies, see
  # https://flutter.dev/to/asset-from-package
```

```

# To add custom fonts to your application, add a fonts
section here,
# in this "flutter" section. Each entry in this list
should have a
# "family" key with the font family name, and a "fonts"
key with a
# list giving the asset and other descriptors for the
font. For
# example:
# fonts:
#   - family: Schyler
#     fonts:
#       - asset: fonts/Schyler-Regular.ttf
#       - asset: fonts/Schyler-Italic.ttf
#         style: italic
#   - family: Trajan Pro
#     fonts:
#       - asset: fonts/TrajanPro.ttf
#       - asset: fonts/TrajanPro_Bold.ttf
#         weight: 700
#
# For details regarding fonts from package dependencies,
# see https://flutter.dev/to/font-from-package

```

4. Buat file `my_notification_screen.dart` dan isi dengan kode berikut.

```

import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import
'package:flutter_local_notifications/flutter_local_notificat
ions.dart';
import 'package:pertemuan_11/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 =
            const
AndroidInitializationSettings('@mipmap/ic_launcher');
        var initializationSettings =
            InitializationSettings(android:
initializationSettingsAndroid);
        FlutterLocalNotificationsPlugin().initialize(initializat
ionSettings);

        // 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((RemoteMessa
ge 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: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                    Text("Nama: Aorinka Anendya", style:
                    TextStyle(fontSize: 16)),
                    Text("NIM: 2211104013", style:
                    TextStyle(fontSize: 16)),
                ],
            ),
        ),
    );
}

```

```

        Text("Kelas: SE0601", style: TextStyle(fontSize:
16)),
        Text("Prodi: Software Engineering", style:
TextStyle(fontSize: 16)),
    ],
),
),
);
}
}

```

5. Isikan file main.dart dengan kode berikut.

```

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_notificat
ions.dart';
import 'package:pertemuan_11/my_notification_screen.dart';

// Variabel global
String? token;

// Handler untuk pesan yang diterima saat aplikasi di
background
Future<void>
_firebaseMessagingBackgroundHandler(RemoteMessage message)
async {
    await Firebase.initializeApp();
    print('Handling a background message:
${message.messageId}');
}

// Konfigurasi channel notifikasi
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();

    // Konfigurasi handler untuk pesan di background
    FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBa
ckgroundHandler);

    // Konfigurasi Flutter Local Notifications
    final FlutterLocalNotificationsPlugin
flutterLocalNotificationsPlugin =
        FlutterLocalNotificationsPlugin();

    await flutterLocalNotificationsPlugin
        .resolvePlatformSpecificImplementation<
            AndroidFlutterLocalNotificationsPlugin>()
            ?.createNotificationChannel(channel);

    // Konfigurasi untuk menampilkan notifikasi di foreground
    await
FirebaseMessaging.instance.setForegroundNotificationPresenta
tionOptions(
    alert: true,
    badge: true,
    sound: true,
);

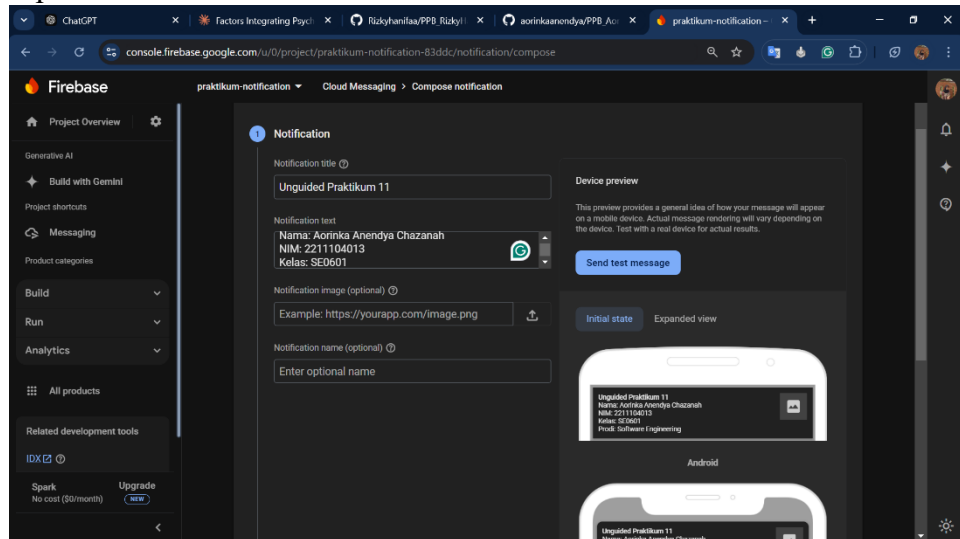
runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

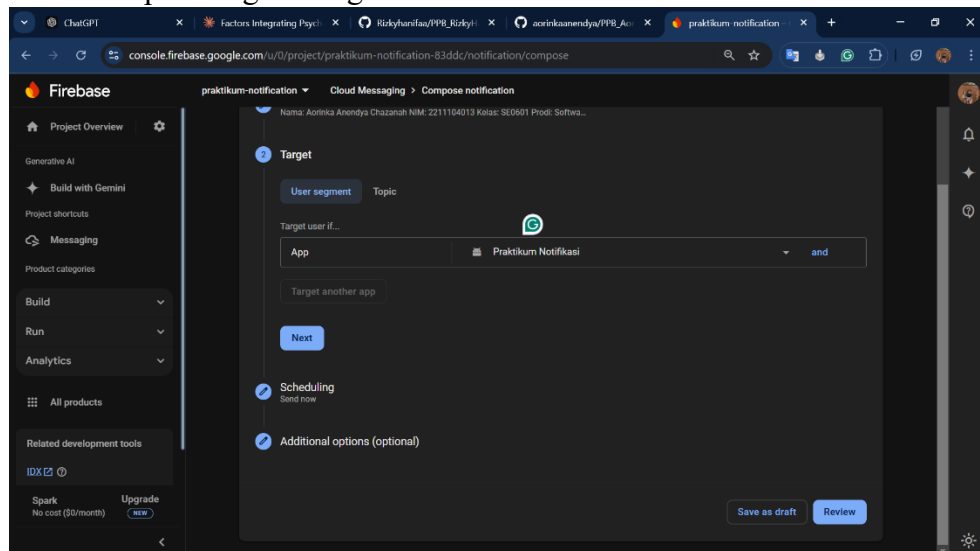
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Flutter Demo',
            theme: ThemeData(
                colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
                useMaterial3: true,
            ),
            home: const MyNotificationScreen(),
        );
    }
}

```

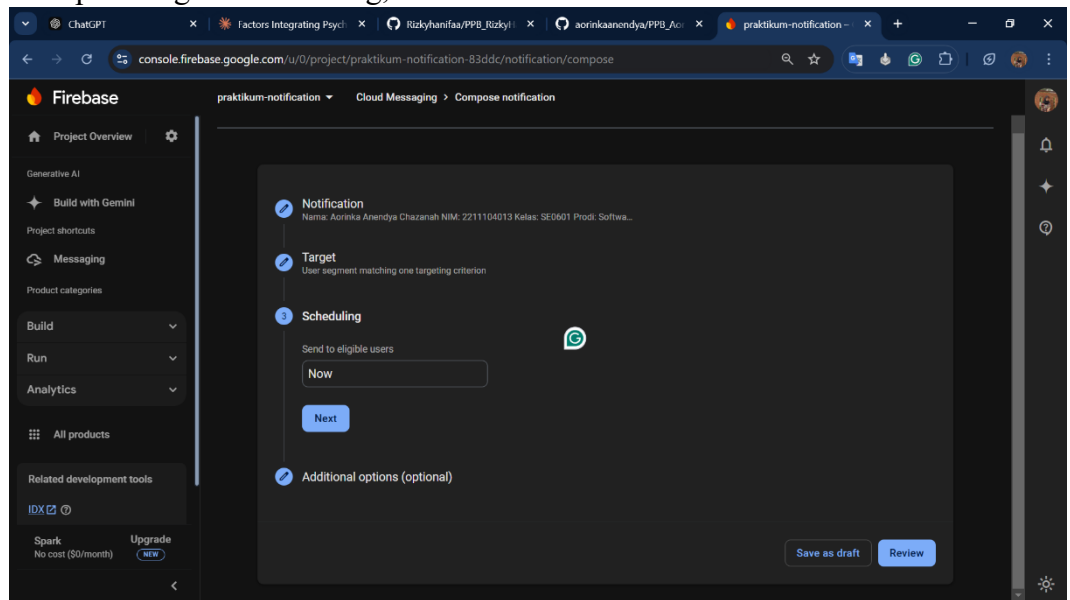
6. Kemudian, buat campaign baru pada firebase console. Caranya klik **All Product > Cloud Messaging > New Campaign**.
7. Buat campaign notifikasi baru dengan isi Nama, NIM, Kelas, dan Prodi seperti berikut.



8. Klik Next pada bagian Target.



9. Lalu pada bagian Scheduling, klik Next.

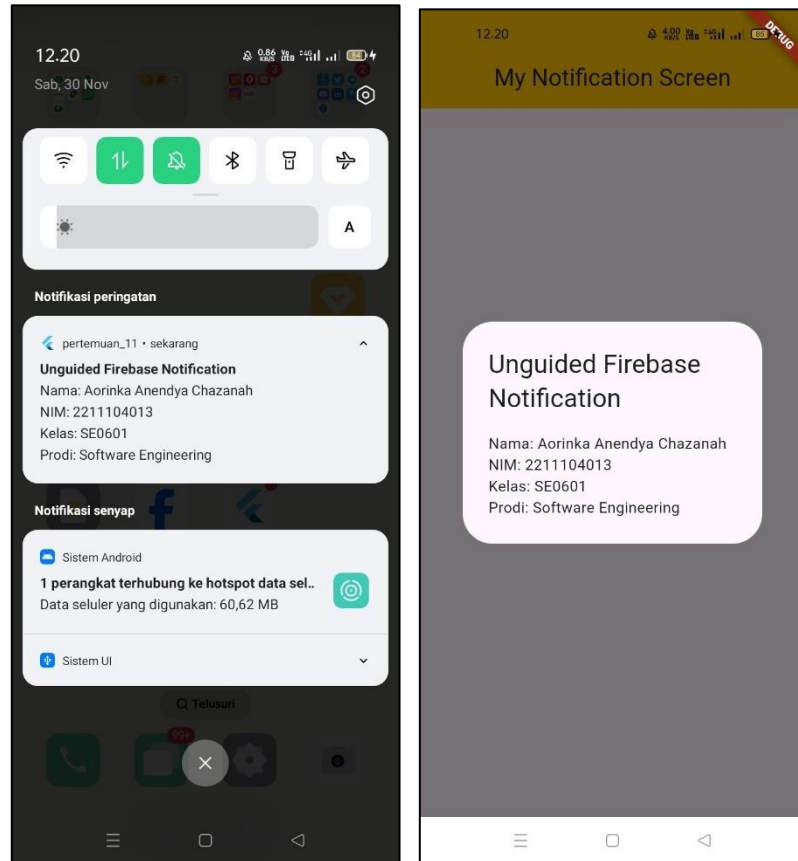


10. Jika berhasil maka muncul seperti ini

Unguided Firebase Notification						
	Nama: Aorinka Anendya Chazannah NIM: 2211104013 Kelas: SE0601 Prodi: Software Engineering		30 Nov 2024 12:19:19	Completed		30 Nov 2024
						<1,000 0%

Hasil Output Program

11. Running program pada VS Code, di sini saya menggunakan perangkat pribadi. Tunggu beberapa saat hingga muncul notifikasi seperti berikut.



Deskripsi Program

Program ini mengintegrasikan Firebase Cloud Messaging (FCM) untuk mengelola notifikasi push, menggunakan Firebase sebagai backend. Aplikasi ini juga menggunakan Flutter Local Notifications untuk menampilkan notifikasi secara lokal di perangkat.

1. Fungsi main.dart

- **Firestore Initialization:** Aplikasi melakukan inisialisasi Firestore melalui `Firestore.initializeApp()`, sehingga semua fitur Firestore (seperti FCM) bisa digunakan.
- **Background Message Handler**
Fungsi `_firebaseMessagingBackgroundHandler` menangani pesan yang diterima saat aplikasi berjalan di background.
- **Notifikasi Lokal**
`FlutterLocalNotificationsPlugin` dikonfigurasi untuk membuat channel notifikasi (`high_importance_channel`) dengan prioritas tinggi, memastikan semua notifikasi terlihat.
- **Foreground Notification**
Aplikasi mengatur agar notifikasi tetap ditampilkan saat aplikasi sedang aktif (`setForegroundNotificationPresentationOptions`).

2. Fungsi MyNotificationScreen

- **Menangani Pesan Masuk:** Aplikasi mendengarkan pesan masuk melalui `FirebaseMessaging.onMessage`:
 - Jika notifikasi diterima, ditampilkan menggunakan `Flutter Local Notifications`.
 - Aplikasi juga menangani pesan yang dibuka melalui `FirebaseMessaging.onMessageOpenedApp`, menampilkan detail pesan dalam dialog.
- **Menampilkan Informasi Pengguna**
Halaman ini menampilkan data pengguna (nama, NIM, kelas, dan prodi) menggunakan widget `Text` yang dirender di tengah layar.
- **Mendapatkan Token FCM**
Fungsi `getToken` digunakan untuk mengambil dan mencetak token FCM perangkat, yang penting untuk mengirim pesan ke perangkat spesifik.

3. Konfigurasi Firebase (Gradle File)

- **Integrasi Firebase**
Firebase dihubungkan ke proyek melalui `com.google.gms.google-services` di file Gradle. `Firebase BoM (firebase-bom)` memastikan semua dependensi Firebase memiliki versi yang kompatibel.
- **Dependencies**
`Firebase-analytics` digunakan sebagai salah satu layanan Firebase. Jika ingin menggunakan layanan Firebase lain, seperti FCM, dependensi tersebut harus ditambahkan.