

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XII
DATA STORAGE (BAGIAN II)**



Disusun Oleh :

Dimas Cahyo Margono / 2211104060

SE-06-02

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

2024

GUIDED

1. LOGIC

Sebelum kita pergi ke tampilan flutter project, Langkah yang pertama kita lakukan adalah mengatur file-services untuk notifikasi Firebase dan lain-lain

A. PUBSPEC.YAML

Sourcecode

```
name: pertemuan11
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# versions available, run `flutter pub outdated`.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/Schlyer-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
```

Deskripsi Program

Berkas pubspec.yaml ini berisi konfigurasi untuk proyek Flutter yang dinamai pertemuan11, dengan deskripsi proyek berupa aplikasi Flutter baru dan tidak akan dipublikasikan ke pub.dev. Versi aplikasi ditetapkan sebagai 1.0.0+1, yang menunjukkan versi utama 1, patch 0, dan build 1. SDK Dart yang digunakan adalah versi 3.5.3 atau yang lebih baru. Pada bagian dependencies, proyek ini mengandalkan beberapa paket seperti flutter untuk framework dasar, cupertino_icons untuk ikon bertema iOS, serta firebase_core, firebase_messaging, dan flutter_local_notifications untuk integrasi dengan Firebase dan pengelolaan notifikasi lokal. Selain itu, bagian dev_dependencies mencakup paket flutter_test untuk pengujian dan flutter_lints untuk memastikan praktik kode yang baik.

B. BUILD.GRADLE (ANDROID/APP)

Kita akan melakukan konfigurasi untuk build.gradle pada Android/app

Sourcecode

```
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.pertemuan11"
    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.pertemuan11"
        // 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
}

```

Deskripsi Program

Berkas build.gradle ini mengonfigurasi proyek Android dalam aplikasi Flutter, termasuk pengaturan plugin Android, Kotlin, dan Flutter, serta integrasi dengan Firebase. Menetapkan versi SDK, ID aplikasi, dan konfigurasi build, serta menggunakan Firebase BoM untuk mengelola versi Firebase secara otomatis dan menambahkan pustaka seperti firebase-analytics. Konfigurasi ini mendukung pengembangan aplikasi Flutter dengan layanan Firebase.

C. GOOGLE-SERVICES.JSON

Sourcecode

```
{
  "project_info": {
    "project_number": "204651033466",
    "project_id": "fir-praktikum-68b0a",
    "storage_bucket": "fir-praktikum-68b0a.firebaseio.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id":
"1:204651033466:android:45e1275f20a566566b9cc3",
        "android_client_info": {
          "package_name": "com.example.pertemuan11"
        }
      },
      "oauth_client": [],
      "api_key": [
        {
          "current_key": "AIzaSyBQuGsxITtliOQLvdcfd3eDknXx9KzqdzU"
        }
      ],
      "services": {
        "appinvite_service": {
          "other_platform_oauth_client": []
        }
      }
    }
  ],
  "configuration_version": "1"
}
```

Deskripsi Program

Berkas JSON ini berisi konfigurasi proyek Firebase untuk aplikasi Android, dengan informasi proyek seperti nomor proyek (204651033466), ID proyek (fir-praktikum-68b0a), dan bucket penyimpanan Firebase (fir-praktikum-68b0a.firebaseio.com). Di dalam bagian client, terdapat ID aplikasi Firebase untuk Android (mobilesdk_app_id) yang menghubungkan aplikasi dengan Firebase, serta nama paket aplikasi (com.example.pertemuan11). Kunci API (AIzaSyBQuGsxITtliOQLvdcfd3eDknXx9KzqdzU) juga disertakan untuk mengakses layanan Firebase, dan bagian services mengindikasikan bahwa aplikasi ini menggunakan layanan undangan aplikasi (appinvite_service), meskipun tidak ada klien OAuth untuk platform lain yang dikonfigurasi.

D. BUILD.GRADLE (ANDROID)

Sourcecode

```
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.google.gms:google-services:4.4.2'
    }
}

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

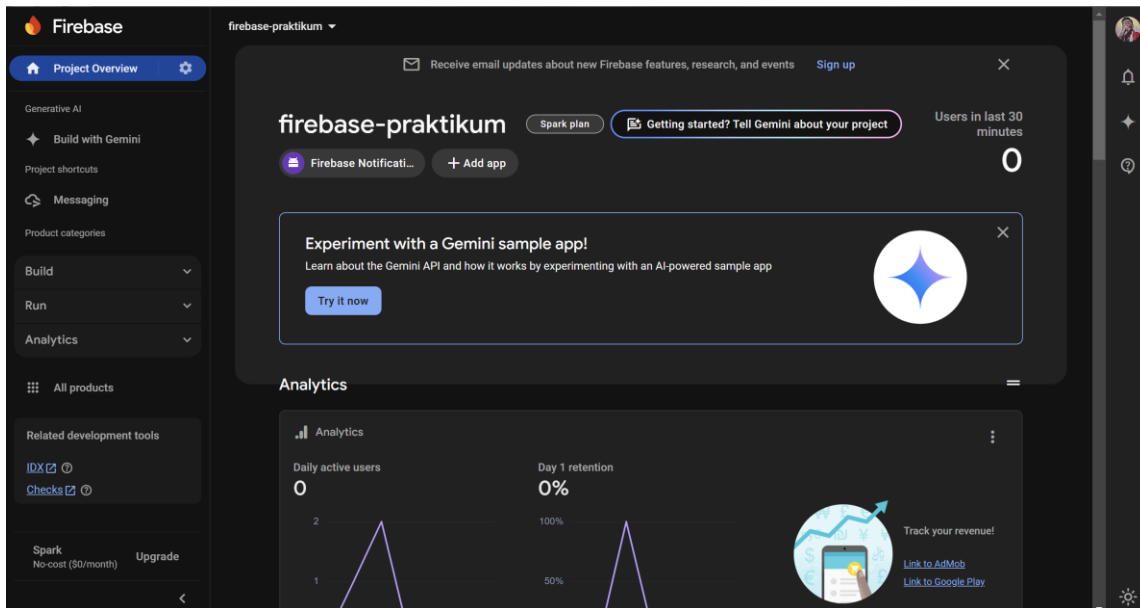
rootProject.buildDir = "../build"
subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
}
subprojects {
    project.evaluationDependsOn(":app")
}

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

Deskripsi Program

Berkas `build.gradle` ini mengonfigurasi repositori untuk Google dan Maven Central, serta menambahkan plugin `google-services` untuk integrasi Firebase. Menetapkan direktori build untuk proyek dan subproyek, serta memastikan subproyek `:app` dievaluasi terlebih dahulu. Tugas `clean` juga didefinisikan untuk menghapus direktori build.

E. PROJECT FIREBASE



Deskripsi Program

Gambar ini menunjukkan konfigurasi Firebase untuk aplikasi Android dengan informasi proyek seperti nomor proyek, ID proyek, dan bucket penyimpanan Firebase. Terdapat juga ID aplikasi Android (mobilesdk_app_id) dan nama paket aplikasi (com.example.pertemuan11). Kunci API digunakan untuk mengakses layanan Firebase, dan konfigurasi layanan mencakup undangan aplikasi, meskipun tidak ada klien OAuth untuk platform lain yang dikonfigurasi. Ini semua mengatur integrasi Firebase dengan aplikasi Android.

2. TAMPILAN APLIKASI

Setelah kita melakukan konfigurasi untuk Firebase Notification, tidak lupa kita melakukan coding untuk tampilan aplikasinya

A. MY_NOTIFICATION_SCREEN.DART

Sourcecode

```
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';

// 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
);

class MyNotificationScreen extends StatefulWidget {
  const MyNotificationScreen({super.key});

  @override
  State<MyNotificationScreen> createState() =>
    _MyNotificationScreenState();
}

class _MyNotificationScreenState extends State<MyNotificationScreen> {
  final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
    FlutterLocalNotificationsPlugin();

  @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.onMessageOpenedApp.listen((RemoteMessage message) {
      print("Notifikasi diklik: ${message.notification?.title}");
      RemoteNotification? notification = message.notification;
      AndroidNotification? android = message.notification?.android;

      if (notification != null && android != null) {
        showDialog(
          context: context,
          builder: (_) => AlertDialog(
            title: const Text("Detail Notifikasi"),
            content: SingleChildScrollView(
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text("Judul: ${notification.title ?? 'Tidak Ada Judul'}"),
                  Text("Isi: ${notification.body ?? 'Tidak Ada Isi'}"),
                  const SizedBox(height: 10),
                  const Text("Informasi Pengirim:",
                    style: TextStyle(fontWeight: FontWeight.bold)),

```

```

        const Text("Nama: Dimas Cahyo Margono"),
        const Text("NIM: 2211104060"),
        const Text("Kelas: SE-06-02"),
        const Text("Prodi: S1 Software Engineering"),
    ],
),
),
actions: [
    TextButton(
        onPressed: () {
            Navigator.of(context).pop();
        },
        child: const Text("Tutup"),
    ),
],
),
);
}
});

// 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: Colors.amber,
        ),
        body: const Center(
            child: Text("Halaman untuk menerima notifikasi"),
        ),
    );
}
}
}

```

Deskripsi Program

Kode ini mengimplementasikan penerimaan dan penanganan notifikasi Firebase Cloud Messaging (FCM) dalam aplikasi Flutter. Di dalam `MyNotificationScreen`, aplikasi menginisialisasi notifikasi lokal untuk Android menggunakan `FlutterLocalNotificationsPlugin`. Ketika aplikasi menerima pesan melalui FCM, jika aplikasi dibuka dari notifikasi, pesan tersebut akan ditampilkan dalam sebuah dialog yang berisi detail notifikasi seperti judul, isi, dan informasi pengirim.

Selain itu, token FCM perangkat diambil melalui metode `getToken` dan dicetak di log. Notifikasi dikonfigurasi dengan saluran prioritas tinggi menggunakan `AndroidNotificationChannel` dan dapat diproses saat aplikasi berjalan di latar depan. Fungsi `FirebaseMessaging.onMessageOpenedApp` digunakan untuk menangani tindakan pengguna ketika notifikasi diklik.

B. MAIN.DART

Sourcecode

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import
'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:pertemuan11/my_notification_screen.dart';
import 'package:firebase_messaging/firebase_messaging.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandle
r);
  await FlutterLocalNotificationsPlugin()
    .resolvePlatformSpecificImplementation<
      AndroidFlutterLocalNotificationsPlugin>()
    ?.createNotificationChannel(channel);
  await
FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
  alert: true,
  badge: true,
  sound: true,
);
  runApp(const MyApp());
}

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
);

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

```

// This widget is the root of your application.
@override
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Flutter Demo',
    theme: ThemeData(
      // This is the theme of your application.
      //
      // TRY THIS: Try running your application with "flutter run".
You'll see
      // the application has a purple toolbar. Then, without quitting
the app,
      // try changing the seedColor in the colorScheme below to
Colors.green
      // and then invoke "hot reload" (save your changes or press the
"hot
      // reload" button in a Flutter-supported IDE, or press "r" if you
used
      // the command line to start the app).
      //
      // Notice that the counter didn't reset back to zero; the
application
      // state is not lost during the reload. To reset the state, use
hot
      // restart instead.
      //
      // This works for code too, not just values: Most code changes can
be
      // tested with just a hot reload.
      colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
      useMaterial3: true,
    ),
    home: const MyNotificationScreen(),
  );
}
}

```

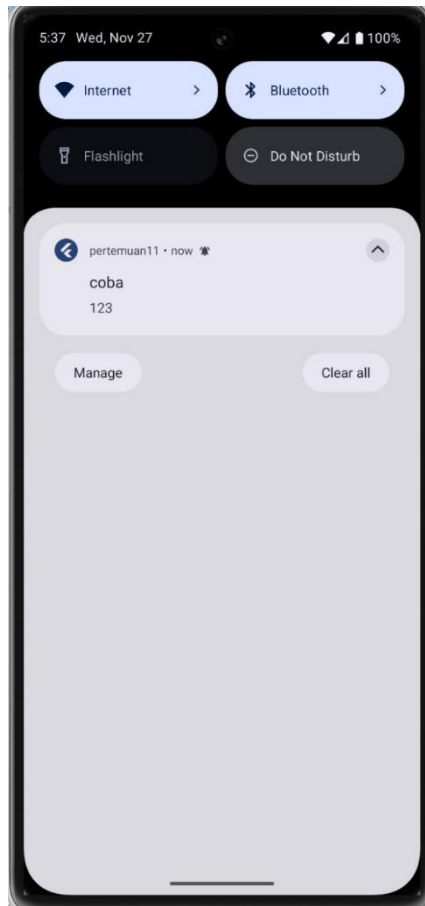
Deskripsi Program

Kode ini menginisialisasi Firebase di aplikasi Flutter, mengonfigurasi notifikasi lokal dan Firebase Cloud Messaging (FCM). Pada latar belakang, pesan FCM ditangani dengan `FirebaseMessaging.onBackgroundMessage`, sementara di latar depan, notifikasi lokal diatur dengan `FlutterLocalNotificationsPlugin` untuk menampilkan pesan penting. Aplikasi menggunakan tema Material3 dan menampilkan layar utama `MyNotificationScreen` untuk menerima dan menampilkan notifikasi.

Screenshoot Output



Tampilan aplikasi



Panel notifikasi



Alert dialog

Deskripsi Program

Gambar ini menunjukkan konfigurasi Firebase di dalam aplikasi Flutter untuk mengelola notifikasi melalui Firebase Cloud Messaging (FCM). Firebase diinisialisasi dan mengonfigurasi saluran notifikasi Android dengan prioritas tinggi untuk menampilkan pesan yang relevan. Selain itu, aplikasi juga mengonfigurasi pengaturan untuk menerima notifikasi baik saat aplikasi berada di latar depan maupun latar belakang, serta menampilkan antarmuka pengguna (UI) yang responsif untuk menampilkan notifikasi yang diterima.

UNGUIDED

Modifikasi Guided diatas bisa menampilkan Nama, Nim, Kelas, dan Prodi kalian ke dalam Notifikasi Flutter Cloud Messaging dan landing page notifikasinya.

Berikut merupakan sourcecode dari program

A. MY_NOTIFICATION_SCREEN.DART

Sourcecode

```
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';

// 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
);

class MyNotificationScreen extends StatefulWidget {
  const MyNotificationScreen({super.key});

  @override
  State<MyNotificationScreen> createState() =>
_MyNotificationScreenState();
}

class _MyNotificationScreenState extends State<MyNotificationScreen> {
  final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
    FlutterLocalNotificationsPlugin();

  @override
  void initState() {
    super.initState();
  }
}
```

```

// Membuat pengaturan inisialisasi notifikasi untuk Android
var initializationSettingsAndroid =
    const AndroidInitializationSettings('@mipmap/ic_launcher');
var initializationSettings =
    InitializationSettings(androidId: initializationSettingsAndroid);
FlutterLocalNotificationsPlugin().initialize(initializationSettings);

// Mendengarkan pesan saat aplikasi aktif
FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
    print("Notifikasi diklik: ${message.notification?.title}");
    RemoteNotification? notification = message.notification;
    AndroidNotification? android = message.notification?.android;

    if (notification != null && android != null) {
        showDialog(
            context: context,
            builder: (_) => AlertDialog(
                title: const Text("Detail Notifikasi"),
                content: SingleChildScrollView(
                    child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            Text("Judul: ${notification.title ?? 'Tidak Ada Judul'}"),
                            Text("Isi: ${notification.body ?? 'Tidak Ada Isi'}"),
                            const SizedBox(height: 10),
                            const Text("Informasi Pengirim:",
                                style: TextStyle(fontWeight: FontWeight.bold)),
                            const Text("Nama: Dimas Cahyo Margono"),
                            const Text("NIM: 2211104060"),
                            const Text("Kelas: SE-06-02"),
                            const Text("Prodi: S1 Software Engineering"),
                        ],
                    ),
                ),
                actions: [
                    TextButton(
                        onPressed: () {
                            Navigator.of(context).pop();
                        },
                        child: const Text("Tutup"),
                    ),
                ],
            ),
        );
    }
});

// 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: Colors.amber,
            ),
            body: const Center(
                child: Text("Halaman untuk menerima notifikasi"),
            ),
        );
    }
}

```

B. MAIN.DART

Sourcecode

```

import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import
'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:pertemuan11/my_notification_screen.dart';
import 'package:firebase_messaging/firebase_messaging.dart';

Future<void> main() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp();
    FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandle
r);
    await FlutterLocalNotificationsPlugin()
        .resolvePlatformSpecificImplementation<
            AndroidFlutterLocalNotificationsPlugin>()
            ?.createNotificationChannel(channel);
    await
FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
        alert: true,
        badge: true,
        sound: true,
    );
    runApp(const MyApp());
}

String? token;
Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message)
async {
    await Firebase.initializeApp();
    print('Handling a background message: ${message.messageId}');
}

```



```

}

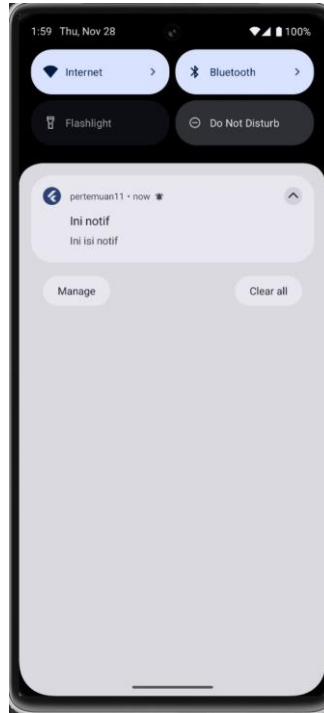
const AndroidNotificationChannel channel = AndroidNotificationChannel(
  'high_importance_channel', // ID Channel
  'High Importance Notifications', // Nama Channel
  description:
    'This channel is used for important notifications.', // Deskripsi
  importance: Importance.high, // Prioritas
);

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run".
        // You'll see
        // the application has a purple toolbar. Then, without quitting
        // the app,
        // try changing the seedColor in the colorScheme below to
        Colors.green
        // and then invoke "hot reload" (save your changes or press the
        "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you
        used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the
        application
        // state is not lost during the reload. To reset the state, use
        hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can
        be
        // tested with just a hot reload.
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MyNotificationScreen(),
    );
  }
}

```

Screenshoot output



Deskripsi Program

Program ini mengonfigurasi Firebase Cloud Messaging (FCM) dan Flutter Local Notifications untuk menangani notifikasi dalam aplikasi Flutter. Aplikasi ini menginisialisasi Firebase, mengatur channel notifikasi dengan prioritas tinggi, dan menampilkan notifikasi ketika aplikasi berada di latar depan atau latar belakang. Selain itu, aplikasi juga mengambil token perangkat FCM untuk mengirim notifikasi secara spesifik ke perangkat pengguna.