

LAPORAN PRAKTIKUM
Modul 11
“Data Storgae (Bagian 2)”



Disusun Oleh:
Muhammad Ralfi - 2211104054
Kelas SE-06-2

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

- a. Mahasiswa mampu memahami konsep firebase
- b. Mahasiswa dapat mengimplementasikan firebase pada flutter

2. Landasan Teori

Firestore

Firestore adalah suatu layanan dari Google untuk memberikan kemudahan bahkan mempermudah para developer aplikasi dalam mengembangkan aplikasinya. Firestore alias BaaS (*Backend as a Service*) merupakan solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan developer. Dengan menggunakan Firestore, apps developer bisa fokus dalam mengembangkan aplikasi tanpa memberikan *effort* yang besar untuk urusan *backend*.

Jenis-jenis atau fitur pada firestore ada banyak diantaranya adalah sebagai berikut:

- a. Firestore Analytics
- b. Firestore Cloud Messaging and Notifications
- c. Firestore Authentication
- d. Firestore Cloud Firestore
- e. Firestore Realtime Database
- f. Firestore Hosting

3. Guided

- a. Instalasi Paket **sqflite** dan **pat**
- b. File 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.d
  art';
import 'package:gd11/my_notification.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized(); await
    Firebase.initializeApp();
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHand
    ler);
  await FlutterLocalNotificationsPlugin()

  .resolvePlatformSpecificImplementation<AndroidFlutterLocalNotification
    sPlugin>()?.createNotificationChannel(channel);
  await
  FirebaseMessaging.instance.setForegroundNotificationPresentationOption
    s(
    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',
      'High Importance Notifications',
      description: 'This channel is used for important notifications.',
      importance: Importance.high,
    );

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

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

```

c. File my_notifications.dart

```

import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import
  'package:flutter_local_notifications/flutter_local_notifications.d
    art';
import 'package:gd11/main.dart';

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

  @override
  State<MyNotification> createState() => _MyNotificationState();
}

class _MyNotificationState extends State<MyNotification> {
  void initState() {
    super.initState();

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

```

```
// 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();
}

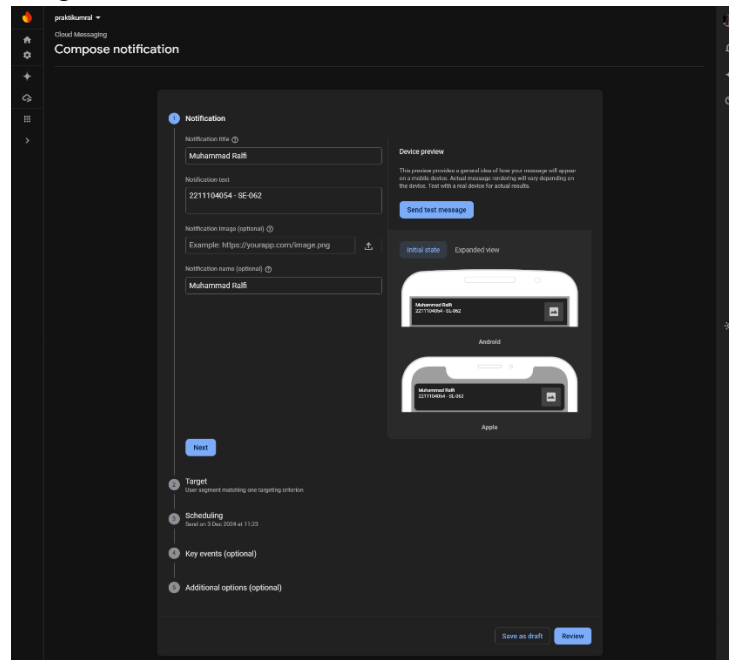
// Metode untuk mendapatkan token FCM
void getToken() async {
  token = await FirebaseMessaging.instance
```

```
.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'),
      backgroundColor: Colors.amber,
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text('This is my notification screen'),
        ],
      ),
    ),
  );
}
```

d. Memanggil / membuat notifikasi

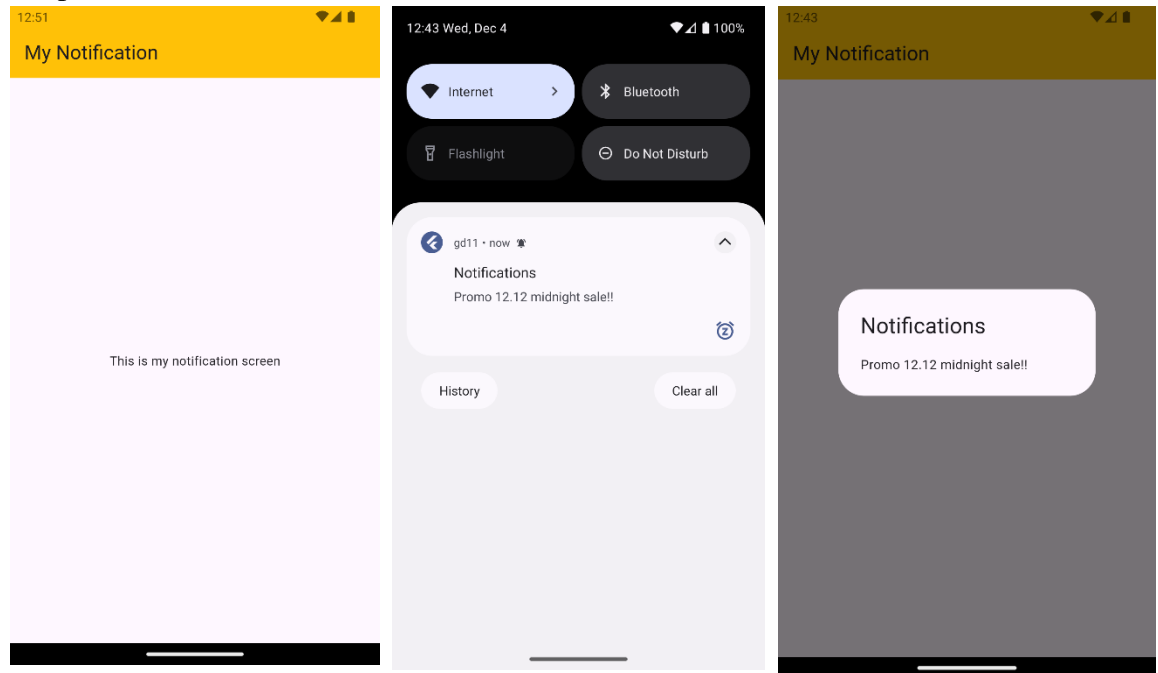
- Dimulai dengan membuka halaman all product pada tab firebase --> pilih menu cloud messaging --> lalu pilih send your first message.
- Selanjutnya, inputkan judul, pesan, dan foto dari notifikasi kalian sesuai dengan kebutuhan



- 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.

e. Output



4. Unguided

a. File 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.d
  art';
import 'package:gd11/my_notification.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();          await
  Firebase.initializeApp();
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHand
    ler);
  await FlutterLocalNotificationsPlugin()

  .resolvePlatformSpecificImplementation<AndroidFlutterLocalNotificatio
    nPlugin>().createNotificationChannel(channel);
  await
  FirebaseMessaging.instance.setForegroundNotificationPresentationOption
    s(
    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',
    'High Importance Notifications',
    description: 'This channel is used for important notifications.',
    importance: Importance.high,
  );

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

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

```

b. File my_notifications.dart

```

import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import
  'package:flutter_local_notifications/flutter_local_notifications.d
    art';
import 'package:gd11/main.dart';

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

  @override
  State<MyNotification> createState() => _MyNotificationState();
}

class _MyNotificationState extends State<MyNotification> {
  void initState() {
    super.initState();

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

```

```

// 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
            ),
          ),
        );
      },
    );
  }
});
// 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
}

```



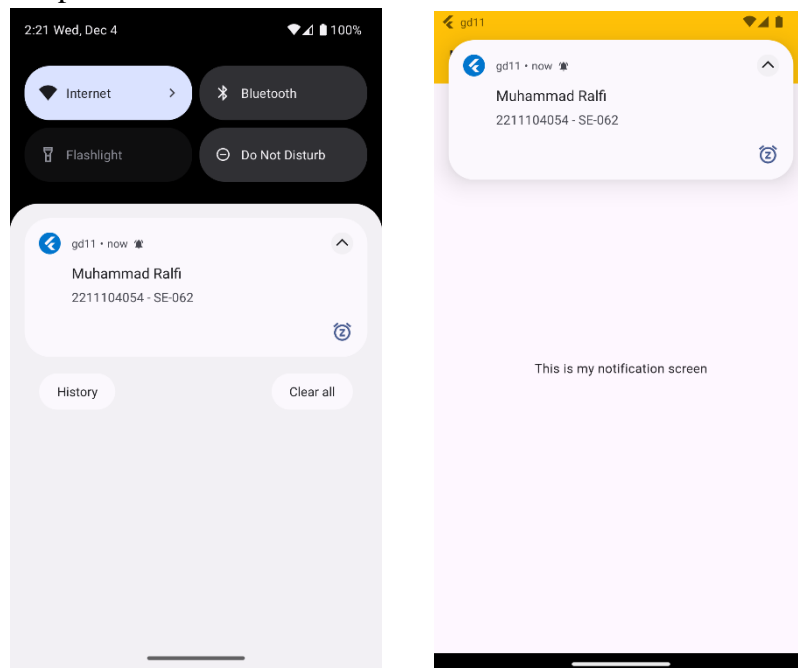
```

    }

    @override
    Widget build(BuildContext context) {
      return Scaffold(
        appBar: AppBar(
          title: const Text('My Notification'),
          backgroundColor: Colors.amber,
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text('This is my notification screen'),
            ],
          ),
        ),
      );
    }
  }
}

```

c. Output



5. Kesimpulan

Kesimpulannya, implementasi Firebase Cloud Messaging (FCM) pada aplikasi Flutter memungkinkan pengiriman notifikasi yang efisien dan real-time kepada pengguna. Dengan memanfaatkan fitur seperti *Notification Message* untuk menampilkan notifikasi langsung dan *Data Message* untuk fleksibilitas data, pengembang dapat memberikan pengalaman interaktif yang optimal. Selain itu, integrasi Firebase dengan Cloud Firestore memungkinkan penyimpanan dan pengambilan data secara real-time, seperti data mahasiswa, yang dapat ditampilkan di halaman utama aplikasi. Dengan langkah-langkah konfigurasi yang tepat, fitur notifikasi dan data real-time ini dapat meningkatkan fungsionalitas dan user experience aplikasi secara keseluruhan.