

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 11  
Firebase Notifikasi**



**Disusun Oleh :  
Arzario Irsyad Al Fatih/2211104032  
SE 06 2**

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

**Dosen Pengampu :  
Yudha Islami Sulistya**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## 1. GUIDED

### 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_notification
s.dart';
import 'package:praktikum_11/my_notification_screen.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackg
roundHandler);
  await FlutterLocalNotificationsPlugin()
    .resolvePlatformSpecificImplementation<
      AndroidFlutterLocalNotificationsPlugin>()
    ?.createNotificationChannel(channel);
  await
FirebaseMessaging.instance.setForegroundNotificationPresentatio
nOptions(
  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(
      colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
      useMaterial3: true,
    ),
    home: MyNotificationScreen(),
  );
}
}

```

- my\_notification\_screen.dart

```

import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import
'package:flutter_local_notifications/flutter_local_notification
s.dart';
import 'package:praktikum_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();
    // Membuat pengaturan inisialisasi notifikasi untuk Android
    var initializationSettingsAndroid =
      const
AndroidInitializationSettings('@mipmap/ic_launcher');
    var initializationSettings =
      InitializationSettings(android:
initializationSettingsAndroid);
    FlutterLocalNotificationsPlugin().initialize(initialization
Settings);

    // 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
                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
                        content: SingleChildScrollView(
                            child: Column(
                                crossAxisAlignment: CrossAxisAlignment.start,
                                children: [Text(notification.body ?? "")], //
                            ),
                        ),
                    );
                },
            );
        }
    });

    // 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
}

```

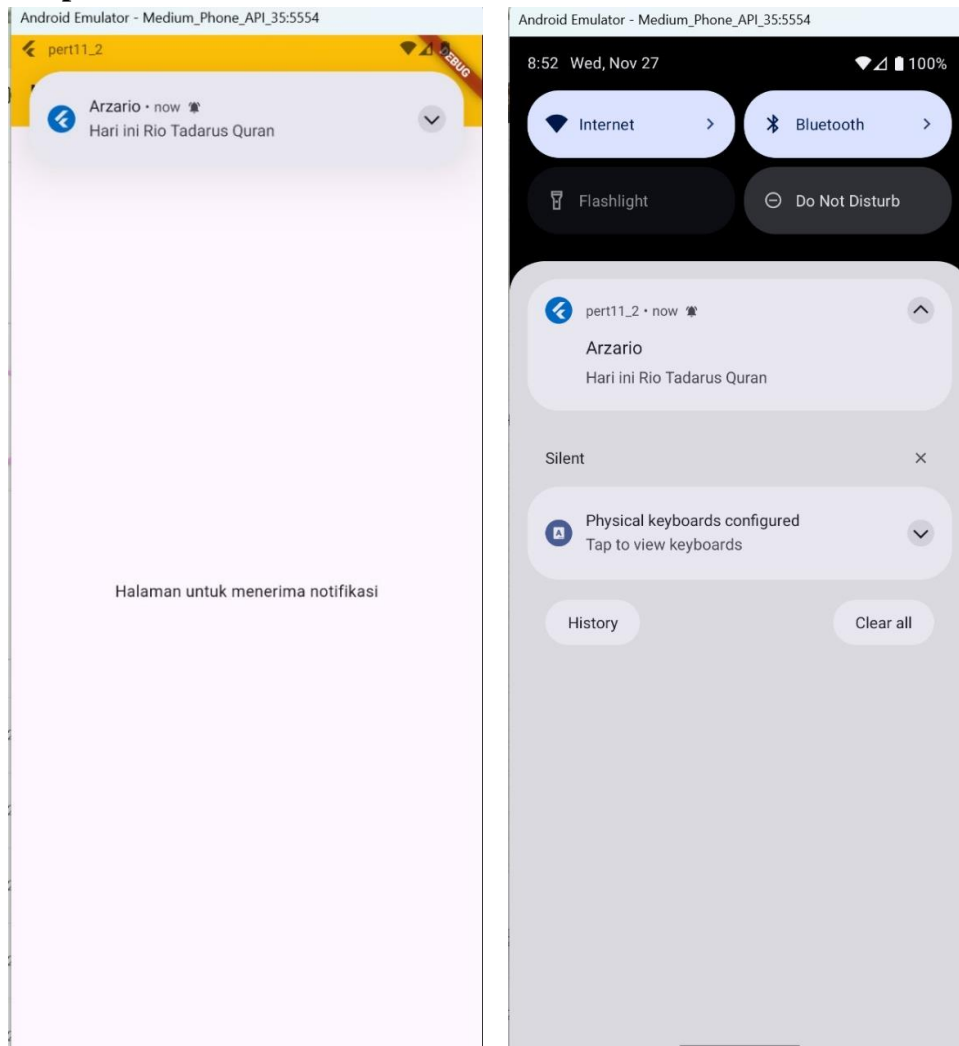
```

}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('My Notification Screen'),
      backgroundColor: Colors.amber,
    ),
    body: Center(
      child: Text('Halaman untuk menerima notifikasi'),
    ),
  );
}
}

```

## Output



## Deskripsi

Kode di atas mengimplementasikan aplikasi Flutter yang menggunakan Firebase Cloud Messaging (FCM) untuk menerima notifikasi push dan menampilkan detail notifikasi menggunakan FlutterLocalNotificationsPlugin. Pada bagian main.dart, aplikasi diinisialisasi dengan Firebase dan notifikasi lokal, serta menangani notifikasi yang diterima saat aplikasi berada di latar belakang.

menggunakan `FirebaseMessaging`. Ketika pengguna menerima notifikasi, aplikasi akan menampilkan tampilan detail notifikasi pada halaman `MyNotificationScreen`, yang menerima payload notifikasi yang berisi data seperti nama, NIM, kelas, dan prodi. Data payload yang diterima diparse menggunakan `Uri.splitQueryString()` untuk menampilkan informasi di halaman tersebut. Aplikasi juga menggunakan `GlobalKey<NavigatorState>` untuk melakukan navigasi ke halaman detail saat notifikasi ditekan.

## 2. UNGUIDED

### 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.da
rt';
import 'package:praktikum_11/my_notification_screen.dart';
import 'notification_channel.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();

  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroun
dHandler);

  // Setup Notification Channel
  await FlutterLocalNotificationsPlugin()
    .resolvePlatformSpecificImplementation<
      AndroidFlutterLocalNotificationsPlugin>()
    ?.createNotificationChannel(channel);

  await
FirebaseMessaging.instance.setForegroundNotificationPresentationOpt
ions(
  alert: true,
  badge: true,
  sound: true,
);

  runApp(const MyApp());
}

// Background message handler
Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage
message) async {
  await Firebase.initializeApp();
  print('Handling a background message: ${message.messageId}');
}
```

```

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(),
    );
  }
}

```

my\_notification\_screen.dart

```

import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import
'package:flutter_local_notifications/flutter_local_notifications.da
rt';
import 'notification_channel.dart';

class MyNotificationScreen extends StatefulWidget {
  const MyNotificationScreen({Key? key}) : super(key: key);

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

class _MyNotificationScreenState extends
State<MyNotificationScreen> {
  late FlutterLocalNotificationsPlugin
flutterLocalNotificationsPlugin;

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

    flutterLocalNotificationsPlugin =
FlutterLocalNotificationsPlugin();

    // Handle when app is terminated
    FirebaseMessaging.instance.getInitialMessage().then((message) {
      if (message != null) {
        _navigateToLandingPage(message.data);
      }
    });
  }
}

```

```

// Handle when app is in foreground
FirebaseMessaging.onMessage.listen((RemoteMessage message) {
  if (message.notification != null) {
    _showNotification(message);
  }
});

// Handle when app is in background and opened via notification
FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage
message) {
  _navigateToLandingPage(message.data);
});
}

void _showNotification(RemoteMessage message) {
  flutterLocalNotificationsPlugin.show(
    0,
    message.notification?.title ?? 'Notifikasi',
    message.notification?.body ?? '',
    NotificationDetails(
      android: AndroidNotificationDetails(
        channel.id,
        channel.name,
        channelDescription: channel.description,
        importance: Importance.high,
      ),
    ),
  );
}

void _navigateToLandingPage(Map<String, dynamic> data) {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => LandingPage(
        nama: data['nama'] ?? 'Unknown',
        nim: data['nim'] ?? 'Unknown',
        kelas: data['kelas'] ?? 'Unknown',
        prodi: data['prodi'] ?? 'Unknown',
      ),
    ),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('My Notification Screen'),
    ),
    body: const Center(
      child: Text('Menunggu notifikasi...'),
    ),
  ),

```



```

    );
  }
}

class LandingPage extends StatelessWidget {
  final String nama;
  final String nim;
  final String kelas;
  final String prodi;

  const LandingPage({
    Key? key,
    required this.nama,
    required this.nim,
    required this.kelas,
    required this.prodi,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Landing Page'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text('Nama: $nama', style: const TextStyle(fontSize:
16)),
            Text('NIM: $nim', style: const TextStyle(fontSize:
16)),
            Text('Kelas: $kelas', style: const TextStyle(fontSize:
16)),
            Text('Prodi: $prodi', style: const TextStyle(fontSize:
16)),
          ],
        ),
      ),
    );
  }
}

```

notification\_screen.dart

```

import
'package:flutter_local_notifications/flutter_local_notifications.da
rt';

const AndroidNotificationChannel channel =
AndroidNotificationChannel(
  'high_importance_channel',

```

```
'High Importance Notifications',  
description: 'This channel is used for important notifications.',  
importance: Importance.high,  
);
```

## Output



## Deskripsi

Kode di atas merupakan implementasi aplikasi Flutter yang menggunakan Firebase Cloud Messaging (FCM) untuk mengirimkan notifikasi push dan menampilkan notifikasi lokal menggunakan Flutter Local Notifications. Aplikasi ini menangani notifikasi dalam tiga keadaan: ketika aplikasi berada di latar depan, latar belakang, dan ketika aplikasi dalam keadaan terminated. Ketika notifikasi diterima, aplikasi menampilkan pemberitahuan melalui flutterLocalNotificationsPlugin dan mengarahkan pengguna ke halaman LandingPage dengan data yang diterima dari notifikasi. Selain itu, aplikasi juga mengonfigurasi saluran notifikasi Android dengan tingkat kepentingan tinggi untuk memastikan bahwa notifikasi yang diterima tidak terlewat. Konfigurasi notifikasi dan pengaturan Firebase Messaging dilakukan pada main.dart, sementara logika penanganan notifikasi ada di dalam MyNotificationScreen.