

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XI
FIREBASE NOTIFIKASI**



Disusun Oleh :
Muhammad Abdul Aziz / 2211104026
SE0601

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

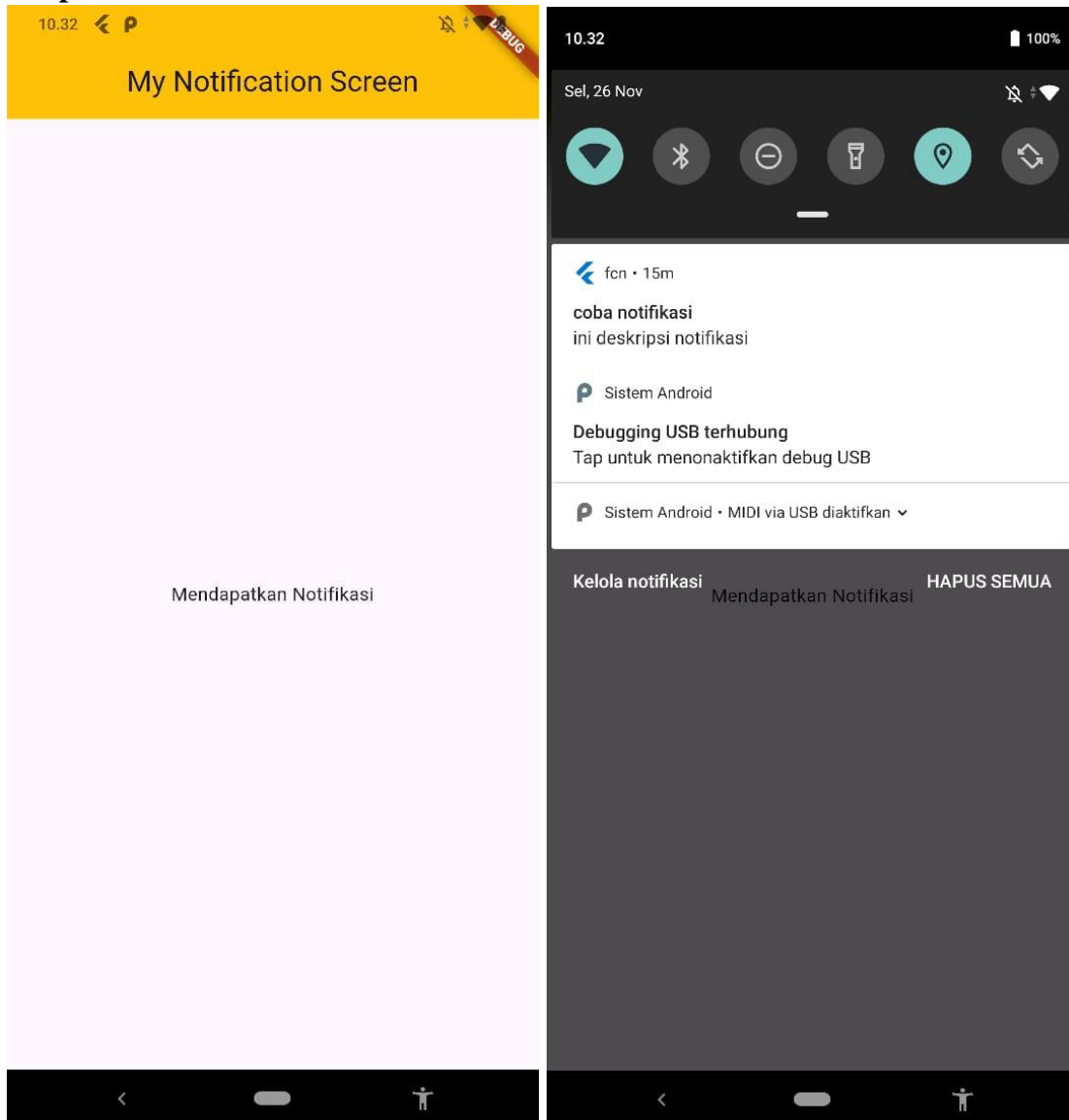
File main.dart :

```
1  import 'package:fcn/my_notification_screen.dart';
2  import 'package:firebase_core/firebase_core.dart';
3  import 'package:firebase_messaging/firebase_messaging.dart';
4  import 'package:flutter/material.dart';
5  import 'package:flutter_local_notifications/flutter_local_notifications.dart';
6
7  void main() async {
8    WidgetsFlutterBinding.ensureInitialized();
9    await Firebase.initializeApp();
10   FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
11   await FlutterLocalNotificationsPlugin()
12     .resolvePlatformSpecificImplementation<
13       AndroidFlutterLocalNotificationsPlugin>()
14     ?.createNotificationChannel(channel);
15   await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
16     alert: true,
17     badge: true,
18     sound: true,
19   );
20   runApp(const MyApp());
21 }
22
23 String? token;
24 Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
25   await Firebase.initializeApp();
26   print('Handling a background message: ${message.messageId}');
27 }
28
29 const AndroidNotificationChannel channel = AndroidNotificationChannel(
30   'high_importance_channel', // ID Channel
31   'High Importance Notifications', // Nama Channel
32   description:
33     'This channel is used for important notifications.', // Deskripsi Channel
34   importance: Importance.high, // Prioritas
35 );
36
37 class MyApp extends StatelessWidget {
38   const MyApp({super.key});
39
40   @override
41   Widget build(BuildContext context) {
42     return MaterialApp(
43       title: 'Flutter Demo',
44       theme: ThemeData(
45         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
46         useMaterial3: true,
47       ),
48       home: const MyNotificationScreen(),
49     );
50   }
51 }
52
```

File my_notification_screen.dart :

```
1 import 'package:fcn/main.dart';
2 import 'package:firebase_messaging/firebase_messaging.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
5
6 class MyNotificationScreen extends StatefulWidget {
7   const MyNotificationScreen({super.key});
8
9   @override
10  State<MyNotificationScreen> createState() => _MyNotificationScreenState();
11 }
12
13 class _MyNotificationScreenState extends State<MyNotificationScreen> {
14   @override
15   void initState() {
16     super.initState();
17     // Membuat pengaturan inisialisasi notifikasi untuk Android
18     var initializationSettingsAndroid =
19       const AndroidInitializationSettings('@mipmap/ic_launcher');
20     var initializationSettings =
21       InitializationSettings(android: initializationSettingsAndroid);
22     FlutterLocalNotificationsPlugin().initialize(initializationSettings);
23
24     // Mendengarkan pesan saat aplikasi aktif
25     FirebaseMessaging.onMessage.listen((RemoteMessage message) {
26       RemoteNotification? notification = message.notification;
27       AndroidNotification? android = message.notification?.android;
28       // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
29       if (notification != null && android != null) {
30         FlutterLocalNotificationsPlugin().show(
31           notification.hashCode, // ID notifikasi (hashCode untuk unik)
32           notification.title, // Judul notifikasi
33           notification.body, // Isi notifikasi
34           NotificationDetails(
35             android: AndroidNotificationDetails(
36               channel.id,
37               channel.name,
38               channelDescription: channel.description,
39               color: Colors.blue,
40               icon: "@mipmap/ic_launcher",
41             ),
42           ),
43         );
44       }
45     });
46
47     // Menangani aksi ketika notifikasi dibuka
48     FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
49       RemoteNotification? notification = message.notification;
50       AndroidNotification? android = message.notification?.android;
51       // Jika notifikasi tersedia, tampilkan dialog
52       if (notification != null && android != null) {
53         showDialog(
54           context: context,
55           builder: (_) {
56             return AlertDialog(
57               title: Text(notification.title ?? ""), // Judul dialog
58               content: SingleChildScrollView(
59                 child: Column(
60                   crossAxisAlignment: CrossAxisAlignment.start,
61                   children: [Text(notification.body ?? "")], // Isi dialog
62                 ),
63               ),
64             );
65           },
66         );
67       }
68     });
69
70     // Memanggil metode untuk mengambil token FCM perangkat
71     getToken();
72   }
73
74   // Metode untuk mendapatkan token FCM
75   void getToken() async {
76     token = await FirebaseMessaging.instance
77       .getToken(); // Mendapatkan token FCM perangkat
78     print('FCM Token: $token'); // Menampilkan token di Log
79   }
80
81   Widget build(BuildContext context) {
82     return Scaffold(
83       appBar: AppBar(
84         title: const Text('My Notification Screen'),
85         backgroundColor: Colors.amber,
86         centerTitle: true,
87       ),
88       body: Center(
89         child: Text('Mendapatkan Notifikasi'),
90       ),
91     );
92   }
93 }
94
```

Output :



Penjelasan :

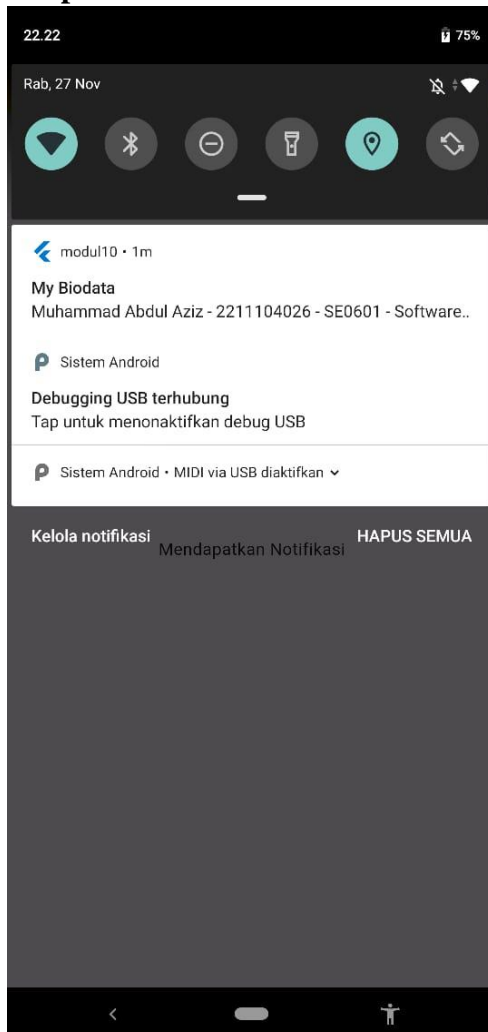
Kode tersebut adalah implementasi notifikasi pada aplikasi Flutter menggunakan Firebase Cloud Messaging (FCM) dan Flutter Local Notifications. Pada fungsi main, aplikasi diinisialisasi dengan Firebase melalui `Firebase.initializeApp()` dan mengatur handler untuk pesan di latar belakang menggunakan `FirebaseMessaging.onBackgroundMessage`. Selain itu, sebuah channel notifikasi lokal bernama `high_importance_channel` dibuat untuk menampilkan notifikasi dengan prioritas tinggi. Kelas `MyApp` adalah titik masuk aplikasi yang menampilkan halaman utama bernama `MyNotificationScreen`, dengan desain tema `Material3`. Pada kelas `MyNotificationScreen`, notifikasi diatur pada metode `initState`, di mana `FlutterLocalNotificationsPlugin` digunakan untuk menerima dan menampilkan notifikasi lokal saat pesan diterima ketika aplikasi aktif. Pesan yang diterima ditampilkan sebagai notifikasi menggunakan `FirebaseMessaging.onMessage`, dan ketika pengguna membuka notifikasi, dialog akan muncul dengan informasi dari notifikasi tersebut melalui `FirebaseMessaging.onMessageOpenedApp`. Selain itu, aplikasi mengambil token perangkat FCM menggunakan `FirebaseMessaging.instance.getToken()` untuk mengidentifikasi perangkat, yang dicetak di log. Secara keseluruhan, aplikasi ini siap untuk menerima dan menampilkan notifikasi secara langsung maupun ketika aplikasi berjalan di latar belakang, dengan halaman utama yang menampilkan teks sederhana sebagai indikasi kesiapan menerima notifikasi.

UNGUIDED

File main.dart : sama seperti guided

File my_notification_screen.dart : sama seperti guided

Output :



Penjelasan :

Kode tersebut adalah implementasi notifikasi pada aplikasi Flutter menggunakan Firebase Cloud Messaging (FCM) dan Flutter Local Notifications. Pada fungsi main, aplikasi diinisialisasi dengan Firebase melalui `Firebase.initializeApp()` dan mengatur handler untuk pesan di latar belakang menggunakan `FirebaseMessaging.onBackgroundMessage`. Selain itu, sebuah channel notifikasi lokal bernama `high_importance_channel` dibuat untuk menampilkan notifikasi dengan prioritas tinggi. Kelas `MyApp` adalah titik masuk aplikasi yang menampilkan halaman utama bernama `MyNotificationScreen`, dengan desain tema `Material3`. Pada kelas `MyNotificationScreen`, notifikasi diatur pada metode `initState`, di mana `FlutterLocalNotificationsPlugin` digunakan untuk menerima dan menampilkan notifikasi lokal saat pesan diterima ketika aplikasi aktif. Pesan yang diterima ditampilkan sebagai notifikasi menggunakan `FirebaseMessaging.onMessage`, dan ketika pengguna membuka notifikasi, dialog akan muncul dengan informasi dari notifikasi tersebut melalui `FirebaseMessaging.onMessageOpenedApp`. Selain itu, aplikasi mengambil token perangkat FCM menggunakan `FirebaseMessaging.instance.getToken()` untuk mengidentifikasi perangkat, yang dicetak di log. Secara keseluruhan, aplikasi ini siap untuk menerima dan menampilkan notifikasi secara langsung maupun ketika aplikasi berjalan di latar belakang, dengan halaman utama yang menampilkan teks sederhana sebagai indikasi kesiapan menerima notifikasi.