

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XI  
DATA STORAGE(BAGIAN 2)**



**Disusun Oleh :**

**Dhiya Ulhaq Ramadhan 2211104053**

**Kelas :**

**SE-06-02**

**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**

## A. GUIDED

### 1. Main.dart

Source code :

```
firebase_notifikasi > lib > main.dart
1  import 'package:firebase_core/firebase_core.dart';
2  import 'package:firebase_messaging/firebase_messaging.dart';
3  import 'package:firebase_notifikasi/my_notification_screen.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',
31   'High Importance Notifications',
32   description: 'This channel is used for important notifications.',
33   importance: Importance.high,
34 );
35
36 class MyApp extends StatelessWidget {
37   const MyApp({super.key});
38
39   @override
40   Widget build(BuildContext context) {
41     return MaterialApp(
42       title: 'Flutter Demo',
43       theme: ThemeData(
44         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
45         useMaterial3: true,
46       ), // ThemeData
47       home: const MyNotificationScreen(),
48     ); // MaterialApp
49   }
50 }
```

## 2. My\_notification\_screen.dart

Source code :

```
firebase_notifikasi > lib > my_notification_screen.dart > _MyNotificationScreenState > initState
1  import 'package:firebase_messaging/firebase_messaging.dart';
2  import 'package:firebase_notifikasi/main.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     // Fixed the method name from initState to initState
17     super.initState();
18
19     // Membuat pengaturan inisialisasi notifikasi untuk Android
20     var initializationSettingsAndroid =
21       const AndroidInitializationSettings('@mipmap/ic_launcher');
22     var initializationSettings =
23       InitializationSettings(android: initializationSettingsAndroid);
24     FlutterLocalNotificationsPlugin().initialize(initializationSettings);
25
26     // Mendengarkan pesan saat aplikasi aktif
27     FirebaseMessaging.onMessage.listen((RemoteMessage message) {
28       RemoteNotification? notification = message.notification;
29       AndroidNotification? android = message.notification?.android;
30       // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
31       if (notification != null && android != null) {
32         FlutterLocalNotificationsPlugin().show(
33           notification.hashCode,
34           notification.title,
35           notification.body,
36           NotificationDetails(
37             android: AndroidNotificationDetails(
```

```

38         channel.id,
39         channel.name,
40         channelDescription: channel.description,
41         color: Colors.blue,
42         icon: "@mipmap/ic_launcher",
43     ), // AndroidNotificationDetails
44 ), // NotificationDetails
45 );
46 }
47 });
48
49 // Menangani aksi ketika notifikasi dibuka
50 FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
51     RemoteNotification? notification = message.notification;
52     AndroidNotification? android = message.notification?.android;
53     // Jika notifikasi tersedia, tampilkan dialog
54     if (notification != null && android != null) {
55         showDialog(
56             context: context,
57             builder: (_) {
58                 return AlertDialog(
59                     title: Text(notification.title ?? ""),
60                     content: SingleChildScrollView(
61                         child: Column(
62                             crossAxisAlignment: CrossAxisAlignment.start,
63                             children: [Text(notification.body ?? "")],
64                         ), // Column
65                     ), // SingleChildScrollView
66                 ); // AlertDialog
67             },
68         );
69     }
70 });
71
72 // Memanggil metode untuk mengambil token FCM perangkat
73 getToken();
74

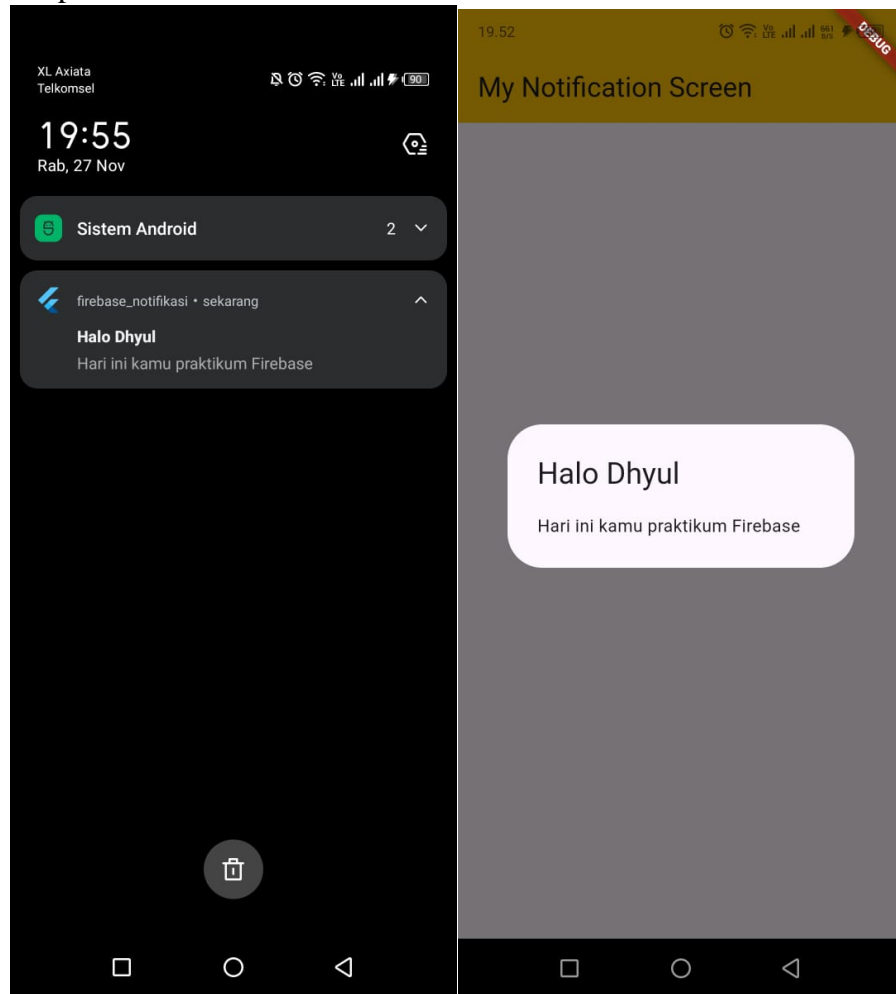
```

```

76 // Metode untuk mendapatkan token FCM
77 void getToken() async {
78     token = await FirebaseMessaging.instance.getToken();
79     print('FCM Token: $token');
80 }
81
82 @override
83 widget build(BuildContext context) {
84     return Scaffold(
85         appBar: AppBar(
86             title: const Text('My Notification Screen'),
87             backgroundColor: Colors.amber,
88         ), // AppBar
89         body: const Center(
90             child: Text('Halaman untuk menerima notifikasi'),
91         ), // Center
92     ); // Scaffold
93 }
94 }
95

```

Output :



### Penjelasan Program

Firestore diinisialisasi menggunakan `Firestore.initializeApp()`. Inisialisasi ini diperlukan sebelum kita bisa menggunakan layanan Firestore apapun. Aplikasi kemudian mendaftarkan fungsi `_firebaseMessagingBackgroundHandler` untuk menangani pesan yang diterima ketika aplikasi berada di latar belakang. Ketika aplikasi berpindah ke `MyNotificationScreen`, beberapa hal penting terjadi di dalam `initState()`. Pertama, aplikasi mengkonfigurasi `FlutterLocalNotificationsPlugin` dengan mengatur ikon default menggunakan `'@mipmap/ic_launcher'`. Ini adalah ikon yang akan muncul di notification tray Android.

Aplikasi kemudian mulai mendengarkan dua jenis event notifikasi:

1. `FirebaseMessaging.onMessage` digunakan untuk menangani notifikasi yang diterima ketika aplikasi sedang aktif (foreground). Ketika pesan diterima, aplikasi memeriksa apakah ada notifikasi dan data Android yang valid. Jika ada, aplikasi menampilkan notifikasi lokal menggunakan `FlutterLocalNotificationsPlugin` dengan judul dan isi pesan yang diterima dari Firestore.
2. `FirebaseMessaging.onMessageOpenedApp` menangani kasus ketika pengguna mengklik notifikasi untuk membuka aplikasi. Dalam kasus ini, aplikasi akan menampilkan `AlertDialog` yang berisi judul dan isi pesan notifikasi.

Terakhir, aplikasi memanggil metode `getToken()` untuk mendapatkan token FCM (Firebase Cloud Messaging) perangkat. Token ini adalah identifikasi unik yang digunakan Firestore untuk mengirim notifikasi ke perangkat spesifik. Token ini dicetak ke konsol untuk keperluan debugging.

## B. UNGUIDED

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

Source code :

lib/main.dart

Untuk kode main.dart masih sama dengan yang ada di Guided, karena saya hanya mengubah tampilan UI nya di my\_notification\_screen.dart

lib/my\_notification\_screen.dart

```
firebase_notifikasi > lib > my_notification_screen.dart > _MyNotificationScreenState > build
1  import 'package:firebase_messaging/firebase_messaging.dart';
2  import 'package:firebase_notifikasi/main.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
18     var initializationSettingsAndroid =
19       const AndroidInitializationSettings('@mipmap/ic_launcher');
20     var initializationSettings =
21       InitializationSettings(android: initializationSettingsAndroid);
22     FlutterLocalNotificationsPlugin().initialize(initializationSettings);
23
24     FirebaseMessaging.onMessage.listen((RemoteMessage message) {
25       RemoteNotification? notification = message.notification;
26       AndroidNotification? android = message.notification?.android;
27
28       if (notification != null && android != null) {
29         FlutterLocalNotificationsPlugin().show(
30           notification.hashCode,
31           notification.title,
32           notification.body,
33           NotificationDetails(
34             android: AndroidNotificationDetails(
35               channel.id,
36               channel.name,
37               channelDescription: channel.description,
```

```

38         color: Color(0xFF6200EE),
39         icon: "@mipmap/ic_launcher",
40         importance: Importance.high,
41         priority: Priority.high,
42         styleInformation: const BigTextStyleInformation(''),
43     ), // AndroidNotificationDetails
44 ), // NotificationDetails
45 );
46 }
47 });
48
49 FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
50     RemoteNotification? notification = message.notification;
51     AndroidNotification? android = message.notification?.android;
52
53     if (notification != null && android != null) {
54         showDialog(
55             context: context,
56             builder: () {
57                 return AlertDialog(
58                     shape: RoundedRectangleBorder(
59                         borderRadius: BorderRadius.circular(16),
60                     ), // RoundedRectangleBorder
61                     title: Row(
62                         children: [
63                             const Icon(
64                                 Icons.notifications,
65                                 color: Color(0xFF6200EE),
66                             ), // Icon
67                             const SizeBox(width: 8),
68                             Expanded(
69                                 child: Text(
70                                     notification.title ?? "",
71                                     style: const TextStyle(
72                                         fontSize: 20,
73                                         fontWeight: FontWeight.bold,
74                                         color: Color(0xFF1F1F1F),

```

```

75         ), // TextStyle
76     ), // Text
77 ), // Expanded
78 ],
79 ), // Row
80 content: SingleChildScrollView(
81   child: Column(
82     crossAxisAlignment: CrossAxisAlignment.start,
83     mainAxisAlignment: MainAxisAlignment.min,
84     children: [
85       Text(
86         notification.body ?? "",
87         style: const TextStyle(
88           fontSize: 16,
89           color: Colors(0xFF424242),
90         ), // TextStyle
91       ), // Text
92     ],
93   ), // Column
94 ), // SingleChildScrollView
95 actions: [
96   TextButton(
97     onPressed: () => Navigator.of(context).pop(),
98     child: const Text(
99       'Tutup',
100     style: TextStyle(
101       color: Colors(0xFF6200EE),
102       fontWeight: FontWeight.bold,
103     ), // TextStyle
104   ), // Text
105 ), // TextButton
106 ],
107 ); // AlertDialog
108 },
109 );
110 }
111 });

```

```

113   getToken();
114 }
115
116 void getToken() async {
117   token = await FirebaseMessaging.instance.getToken();
118   print('FCM Token: $token');
119 }
120
121 @override
122 Widget build(BuildContext context) {
123   return Scaffold(
124     appBar: AppBar(
125       elevation: 0,
126       title: const Text(
127         'Notifikasi',
128         style: TextStyle(
129           fontWeight: FontWeight.bold,
130           color: Colors.white,
131         ), // TextStyle
132       ), // Text
133       backgroundColor: Colors(0xFF6200EE),
134       actions: [
135         IconButton(
136           icon: const Icon(Icons.refresh, color: Colors.white),
137           onPressed: () => getToken(),
138         ), // IconButton
139       ],
140     ), // AppBar
141     body: Container(
142       decoration: const BoxDecoration(
143         gradient: LinearGradient(
144           begin: Alignment.topCenter,
145           end: Alignment.bottomCenter,
146           colors: [Colors(0xFFE7F6), Colors.white],
147         ), // LinearGradient
148       ), // BoxDecoration
149       child: Center(

```

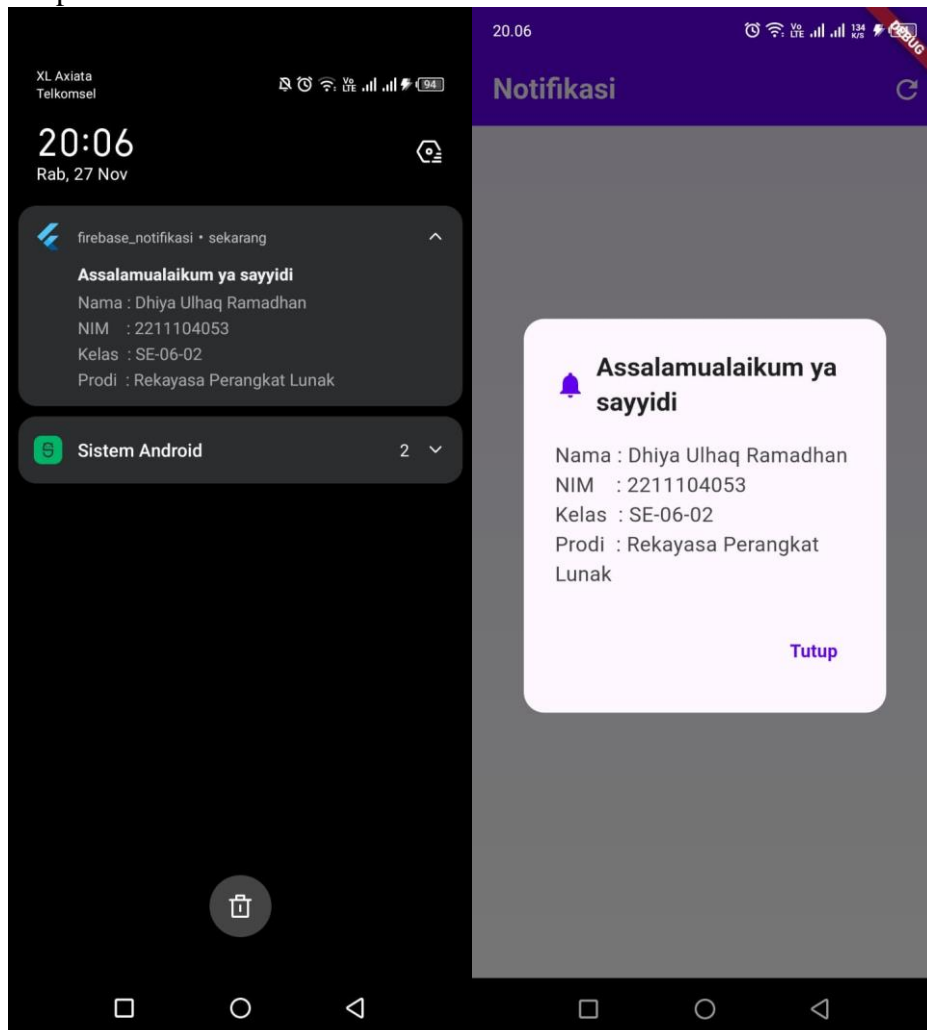


```

150         child: Column(
151           mainAxisAlignment: MainAxisAlignment.center,
152           children: [
153             const Icon(
154               Icons.notifications_active,
155               size: 80,
156               color: Color(0xFF6200EE),
157             ), // Icon
158             const SizedBox(height: 24),
159             const Text(
160               'Notifikasi Aktif',
161               style: TextStyle(
162                 fontSize: 24,
163                 fontWeight: FontWeight.bold,
164                 color: Color(0xFF1F1F1F),
165               ), // TextStyle
166             ), // Text
167             const SizedBox(height: 12),
168             const Padding(
169               padding: EdgeInsets.symmetric(horizontal: 32),
170               child: Text(
171                 'Anda akan menerima pemberitahuan untuk setiap pembaruan penting',
172                 textAlign: TextAlign.center,
173                 style: TextStyle(
174                   fontSize: 16,
175                   color: Color(0xFF424242),
176                 ), // TextStyle
177               ), // Text
178             ), // Padding
179           ], // Column
180         ), // Center
181       ), // Container
182     ); // Scaffold
183   }
184 }
185

```

Output :



### Penjelasan Program

Aplikasi memastikan bahwa Flutter binding sudah diinisialisasi dengan memanggil `WidgetsFlutterBinding.ensureInitialized()`. Ini penting karena akan menggunakan fitur platform native untuk notifikasi. Setelah itu, aplikasi menginisialisasi Firebase dengan `Firebase.initializeApp()`. Inisialisasi ini diperlukan agar aplikasi dapat terhubung dengan layanan Firebase Cloud Messaging (FCM) yang akan digunakan untuk mengirim dan menerima notifikasi. Aplikasi juga mendaftarkan handler untuk pesan background, sehingga dapat memproses notifikasi bahkan ketika aplikasi sedang berjalan di latar belakang.

Aplikasi menjalankan `MyNotificationScreen` yang merupakan layar utama aplikasi. Pada layar ini, terdapat beberapa komponen penting untuk menangani notifikasi. Di dalam `initState`, aplikasi menginisialisasi plugin notifikasi lokal Flutter dengan mengatur ikon yang akan digunakan untuk notifikasi (`@mipmap/ic_launcher`). Aplikasi ini memiliki kemampuan untuk menangani notifikasi dalam tiga skenario berbeda. Pertama, ketika aplikasi sedang aktif (foreground), notifikasi akan ditampilkan menggunakan `FlutterLocalNotificationsPlugin`. Kedua, ketika aplikasi dibuka dari notifikasi, sebuah dialog akan muncul menampilkan detail notifikasi. Ketiga, aplikasi juga dapat menangani notifikasi yang diterima saat aplikasi berada di latar belakang.

Aplikasi berhasil menampilkan notifikasi dengan desain yang rapi. Notifikasi tersebut berisi informasi mahasiswa dengan format yang terstruktur, dimulai dengan salam "Assalamualaikum ya sayyidi" diikuti dengan detail saya yaitu Nama(Dhiya Ulhaq Ramadhan), NIM (2211104053), kelas (SE-06-02), dan program studi (Rekayasa Perangkat Lunak).