

LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL 11

DATA STORAGE (BAGIAN 2)



Disusun Oleh :

YOGI HAFIDH MAULANA / 2211104061

SE06-02

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

- **My Notification screen**

```
1  import 'package:firebase_messaging/firebase_messaging.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_local_notifications/flutter_local_notifications.dart';
4  import 'package:praktikum/main.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     // Membuat pengaturan inisialisasi notifikasi untuk Android
19     var initializationSettingsAndroid =
20       const AndroidInitializationSettings('@mipmap/ic_launcher');
21     var initializationSettings =
22       InitializationSettings(android: initializationSettingsAndroid);
23     FlutterLocalNotificationsPlugin().initialize(initializationSettings);
24
25     // Mendengarkan pesan saat aplikasi aktif
26     FirebaseMessaging.onMessage.listen((RemoteMessage message) {
27       RemoteNotification? notification = message.notification;
28       AndroidNotification? android = message.notification?.android;
29       // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
30       if (notification != null && android != null) {
31         FlutterLocalNotificationsPlugin().show(
32           notification.hashCode, // ID notifikasi (hashCode untuk unik)
33           notification.title, // Judul notifikasi
34           notification.body, // Isi notifikasi
35           NotificationDetails(
36             android: AndroidNotificationDetails(
37               channel.id,
38               channel.name,
39               channelDescription: channel.description,
40               color: Colors.blue,
41               icon: "@mipmap/ic_launcher",
42             ), // AndroidNotificationDetails
43           ), // NotificationDetails
44         );
45       }
46     });
```

```

47
48 // Menangani aksi ketika notifikasi dibuka
49 FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
50   RemoteNotification? notification = message.notification;
51   AndroidNotification? android = message.notification?.android;
52   // Jika notifikasi tersedia, tampilkan dialog
53   if (notification != null && android != null) {
54     showDialog(
55       context: context, Don't use 'BuildContext's across async gaps. Try rewriting th
56       builder: (_) {
57         return AlertDialog(
58           title: Text(notification.title ?? ""), // Judul dialog
59           content: SingleChildScrollView(
60             child: Column(
61               crossAxisAlignment: CrossAxisAlignment.start,
62               children: [Text(notification.body ?? "")], // Isi dialog
63             ), // Column
64           ), // SingleChildScrollView
65         ); // AlertDialog
66       },
67     );
68   }
69 });

71 // Memanggil metode untuk mengambil token FCM perangkat
72 getToken();
73 }
74
75 // Metode untuk mendapatkan token FCM
76 void getToken() async {
77   token = await FirebaseMessaging.instance
78     .getToken(); // Mendapatkan token FCM perangkat
79   print('FCM Token: $token'); // Menampilkan token di log Don't invoke 'print'
80 }
81
82 Widget build(BuildContext context) { The member 'build' overrides an inherited
83   return Scaffold(
84     appBar: AppBar(
85       title: const Text("My Notification"),
86       backgroundColor: Colors.yellow,
87     ), // AppBar
88     body: const Center(
89       child: Text("Halaman untuk menerima notifikasi"),
90     ), // Center
91   ); // Scaffold
92 }
93 }

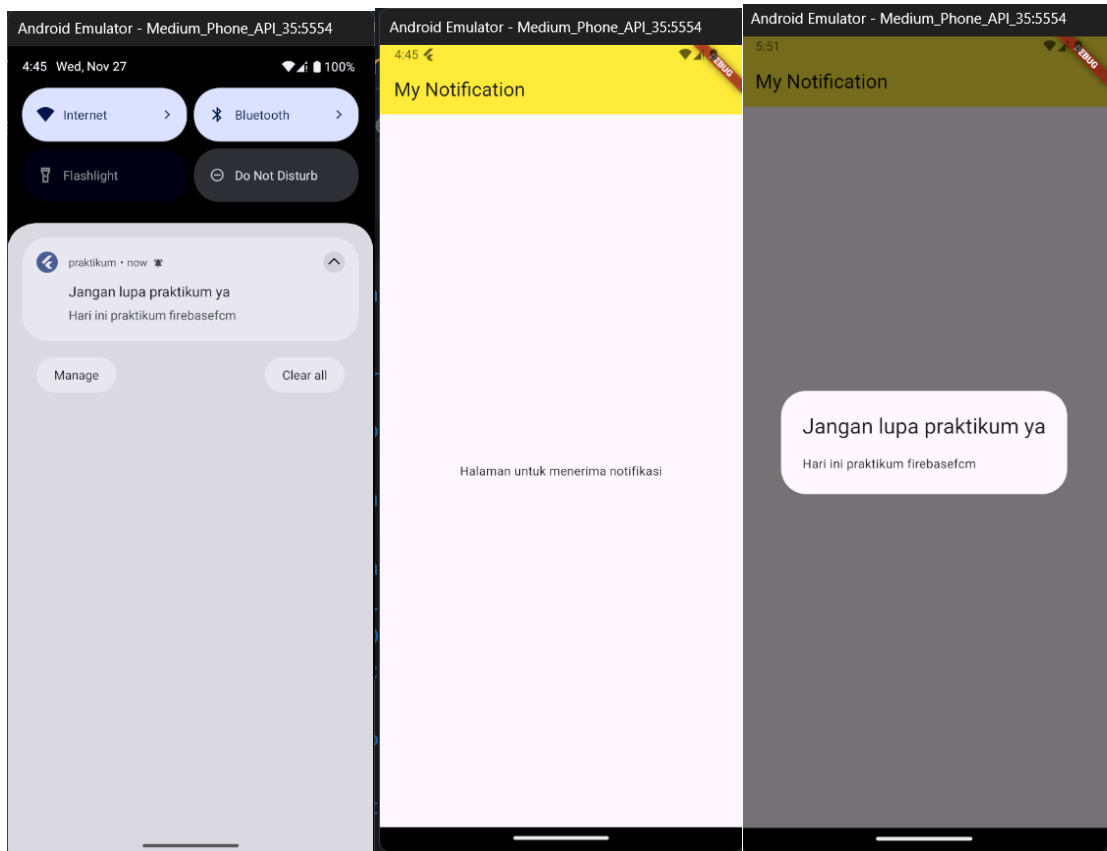
```

- Main

```
1  import 'package:firebase_core/firebase_core.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  import 'package:praktikum/my_notification_screen.dart';
6
7  Run | Debug | Profile
8  void main() async {
9    WidgetsFlutterBinding.ensureInitialized();
10   await Firebase.initializeApp();
11   FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
12   await FlutterLocalNotificationsPlugin()
13     .resolvePlatformSpecificImplementation<
14     AndroidFlutterLocalNotificationsPlugin>()
15     ?.createNotificationChannel(channel);
16   await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
17     alert: true,
18     badge: true,
19     sound: true,
20   );
21   runApp(MyApp());
22 }
23
```

```
24 String? token;
25
26 Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message)
27 async {
28   await Firebase.initializeApp();
29   print('Handling a background message: ${message.messageId}'); Don't invoke 'print' in prod
30 }
31
32 const AndroidNotificationChannel channel = AndroidNotificationChannel(
33   'high_importance_channel', // ID Channel
34   'High Importance Notifications', // Nama Channel
35   description: 'This channel is used for important notifications.', // Deskripsi Channel
36   importance: Importance.high, // Prioritas
37 );
38
39 class MyApp extends StatelessWidget { Constructors for public widgets should have a named
40   @override
41   Widget build(BuildContext context) {
42     return MaterialApp(
43       title: 'Firebase Notifikasi',
44       theme: ThemeData(primarySwatch: Colors.orange),
45       home: MyNotificationScreen(), Use 'const' with the constructor to improve performance
46     ); // MaterialApp
47   }
48 }
49
```

Output



Deskripsi Program

Kode di atas adalah implementasi sederhana notifikasi Firebase di aplikasi Flutter. Aplikasi ini menggunakan Firebase Messaging untuk menerima notifikasi push baik saat aplikasi aktif maupun di latar belakang. Saat aplikasi berjalan, notifikasi ditampilkan menggunakan `flutter_local_notifications` dengan mengatur saluran notifikasi khusus untuk memastikan pemberitahuan terlihat. Jika notifikasi diterima saat aplikasi aktif, notifikasi akan muncul sebagai pesan pop-up menggunakan `FlutterLocalNotificationsPlugin`, sedangkan jika notifikasi dibuka, sebuah dialog akan ditampilkan dengan judul dan isi notifikasi. Kode juga mencakup penanganan pesan latar belakang melalui fungsi `_firebaseMessagingBackgroundHandler` dan memperoleh token FCM unik perangkat, yang dicetak di log untuk digunakan dalam mengirim notifikasi. Komponen utama seperti `FirebaseMessaging`, `FlutterLocalNotificationsPlugin`, dan `AndroidNotificationChannel` berperan penting dalam pengelolaan notifikasi Firebase ini.

B. Unguided

Mynotification Screen file

```
1  import 'package:firebase_messaging/firebase_messaging.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_local_notifications/flutter_local_notifications.dart';
4  import 'package:praktikum/main.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      // Membuat pengaturan inisialisasi notifikasi untuk Android
19      var initializationSettingsAndroid =
20        const AndroidInitializationSettings('@mipmap/ic_launcher');
21      var initializationSettings =
22        InitializationSettings(android: initializationSettingsAndroid);
23      FlutterLocalNotificationsPlugin().initialize(initializationSettings);
24
25      // Mendengarkan pesan saat aplikasi aktif
26      FirebaseMessaging.onMessage.listen((RemoteMessage message) {
27        RemoteNotification? notification = message.notification;
28        AndroidNotification? android = message.notification?.android;
29        // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
30        if (notification != null && android != null) {
31          FlutterLocalNotificationsPlugin().show(
32            notification.hashCode, // ID notifikasi (hashCode untuk unik)
33            notification.title, // Judul notifikasi
34            notification.body, // Isi notifikasi
35            NotificationDetails(
36              android: AndroidNotificationDetails(
37                channel.id,
38                channel.name,
39                channelDescription: channel.description,
40                color: Colors.blue,
41                icon: "@mipmap/ic_launcher",
42              ), // AndroidNotificationDetails
43            ), // NotificationDetails
44          );
45        }
46      });
```

```

47
48 // Menangani aksi ketika notifikasi dibuka
49 FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
50     RemoteNotification? notification = message.notification;
51     AndroidNotification? android = message.notification?.android;
52     // Jika notifikasi tersedia, tampilkan dialog
53     if (notification != null && android != null) {
54         showDialog(
55             context: context, Don't use 'BuildContext's across async gaps. Try rewriting th
56             builder: (_) {
57                 return AlertDialog(
58                     title: Text(notification.title ?? ""), // Judul dialog
59                     content: SingleChildScrollView(
60                         child: Column(
61                             crossAxisAlignment: CrossAxisAlignment.start,
62                             children: [Text(notification.body ?? "")], // Isi dialog
63                         ), // Column
64                     ), // SingleChildScrollView
65                 ); // AlertDialog
66             },
67         );
68     }
69 });

71 // Memanggil metode untuk mengambil token FCM perangkat
72 getToken();
73 }

74
75 // Metode untuk mendapatkan token FCM
76 void getToken() async {
77     token = await FirebaseMessaging.instance
78         .getToken(); // Mendapatkan token FCM perangkat
79     print('FCM Token: $token'); // Menampilkan token di log Don't invoke 'print'
80 }

81
82 Widget build(BuildContext context) { The member 'build' overrides an inherited
83     return Scaffold(
84         appBar: AppBar(
85             title: const Text("My Notification"),
86             backgroundColor: Colors.yellow,
87         ), // AppBar
88         body: const Center(
89             child: Text("Halaman untuk menerima notifikasi"),
90         ), // Center
91     ); // Scaffold
92 }
93 }

```

Main File

```
1  import 'package:firebase_core/firebase_core.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  import 'package:praktikum/my_notification_screen.dart';
6
7  Run | Debug | Profile
8  void main() async {
9    WidgetsFlutterBinding.ensureInitialized();
10   await Firebase.initializeApp();
11   FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
12   await FlutterLocalNotificationsPlugin()
13     .resolvePlatformSpecificImplementation<
14     AndroidFlutterLocalNotificationsPlugin>()
15     ?.createNotificationChannel(channel);
16   await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
17     alert: true,
18     badge: true,
19     sound: true,
20   );
21   runApp(MyApp());
22 }
23
24 String? token;
25
26 Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message)
27 async {
28   await Firebase.initializeApp();
29   print('Handling a background message: ${message.messageId}'); Don't invoke 'print' in prod
30 }
31
32 const AndroidNotificationChannel channel = AndroidNotificationChannel(
33   'high_importance_channel', // ID Channel
34   'High Importance Notifications', // Nama Channel
35   description: 'This channel is used for important notifications.', // Deskripsi Channel
36   importance: Importance.high, // Prioritas
37 );
38
39 class MyApp extends StatelessWidget { Constructors for public widgets should have a named
40   @override
41   Widget build(BuildContext context) {
42     return MaterialApp(
43       title: 'Firebase Notifikasi',
44       theme: ThemeData(primarySwatch: Colors.orange),
45       home: MyNotificationScreen(), Use 'const' with the constructor to improve performance
46     ); // MaterialApp
47   }
48 }
```


Firebase message cloude

Filter campaigns

Search by campaign name, description ...

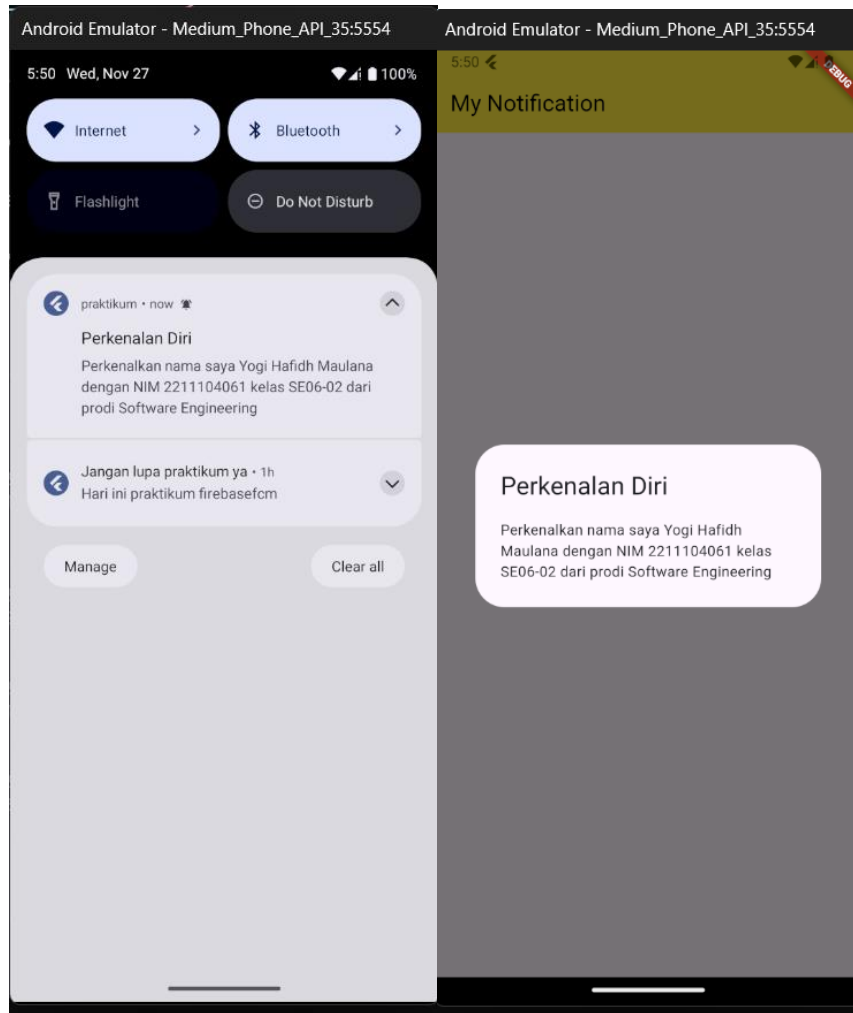
New experiment

New campaign

Campaign	Start	End	Status	Target	Last Updated ↓	Sends / Impressions	Clicks / Opens
<div> <div></div> <div>Perkenalan Diri</div> <div>Perkenalkan nama saya Yogi Hafidh Maulana dengan NIM 2211104061 kelas SED6-02 dari prodi Software Engineering</div> </div>	Nov 27, 2024	-	Completed	<div> <div></div> <div></div> </div>	Nov 27, 2024	<1000	0%
<div> <div></div> <div>Jangan lupa praktikum ya</div> <div>Hari ini praktikum firebasefcm</div> </div>	Nov 27, 2024	-	Completed	<div> <div></div> <div></div> </div>	Nov 27, 2024	<1000	0%
<div> <div></div> <div>Jangan lupa praktikum ya</div> <div>Hari ini praktikum firebasefcm</div> </div>	Nov 27, 2024	-	Completed	<div> <div></div> <div></div> </div>	Nov 27, 2024	<1000	0%
<div> <div></div> <div>Jangan lupa praktikum ya</div> <div>Hari ini praktikum firebasefcm</div> </div>	Nov 27, 2024	-	Completed	<div> <div></div> <div></div> </div>	Nov 27, 2024	<1000	0%

0/10 Recurring notifications

Output



Deskripsi Program

Kode di atas adalah implementasi sederhana notifikasi Firebase di aplikasi Flutter. Aplikasi ini menggunakan Firebase Messaging untuk menerima notifikasi push, baik saat aplikasi aktif maupun di latar belakang. Salah satu contoh notifikasi yang ditampilkan adalah kampanye "Perkenalan Diri," seperti terlihat pada gambar. Kampanye ini memiliki pesan berupa perkenalan pengguna dengan detail "Perkenalkan nama saya Yogi Hafidh Maulana dengan NIM 2211104061 kelas SE06-02 dari prodi Software Engineering," yang berhasil dikirimkan dan statusnya "Completed." Saat aplikasi berjalan, notifikasi semacam ini dapat muncul sebagai pesan pop-up dengan judul dan isi sesuai yang dikirim, menggunakan `FlutterLocalNotificationsPlugin`. Jika notifikasi dibuka, dialog akan muncul menampilkan detail notifikasi. Implementasi ini juga mencakup penanganan notifikasi di latar belakang melalui fungsi `_firebaseMessagingBackgroundHandler`, serta pengambilan token FCM unik perangkat untuk kebutuhan pengiriman notifikasi berikutnya.