

GUIDED
PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK
MODUL XI
DATA STORAGE BAGIAN 2



Disusun Oleh :
Maria Nathasya Desfera Pangestu / 2211104008
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

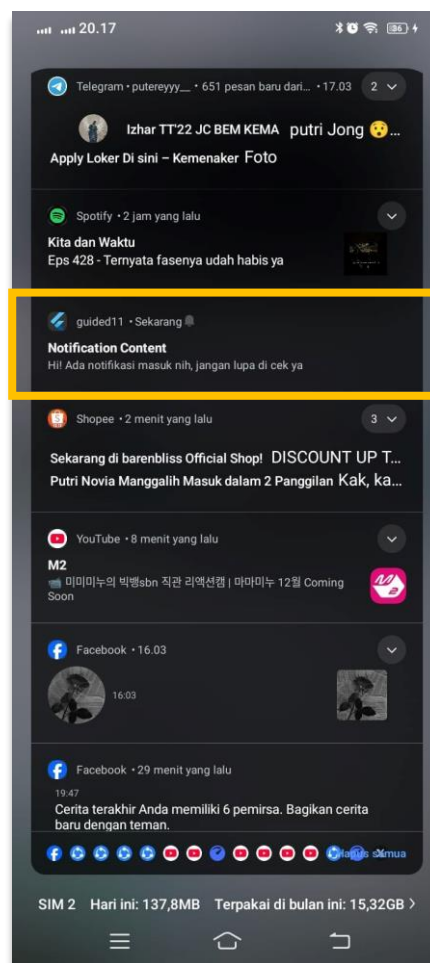
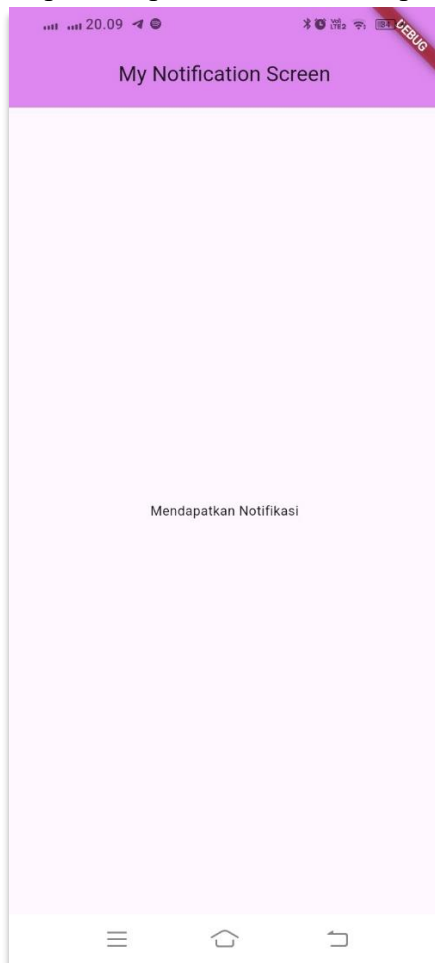
1. Source code main.dart

```
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:guided/notification_screen.dart';
6
7 String? token;
8 Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
9   await Firebase.initializeApp();
10  print('Handling a background message: ${message.messageId}');
11 }
12
13 const AndroidNotificationChannel channel = AndroidNotificationChannel(
14   'high_importance_channel', // ID Channel
15   'High Importance Notifications', // Nama Channel
16   description:
17     'This channel is used for important notifications.', //Deskripsi Channel
18   importance: Importance.high, // Prioritas
19 );
20
21 void main() async {
22   WidgetsFlutterBinding.ensureInitialized();
23   await Firebase.initializeApp();
24   FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
25   await FlutterLocalNotificationsPlugin()
26     .resolvePlatformSpecificImplementation<
27       AndroidFlutterLocalNotificationsPlugin>()
28     ?.createNotificationChannel(channel);
29   await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
30     alert: true,
31     badge: true,
32     sound: true,
33   );
34
35   runApp(const MyApp());
36 }
37
38 class MyApp extends StatelessWidget {
39   const MyApp({super.key});
40
41   // This widget is the root of your application.
42   @override
43   Widget build(BuildContext context) {
44     return MaterialApp(
45       title: 'Flutter Demo',
46       theme: ThemeData(
47         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
48         useMaterial3: true,
49       ),
50       home: const NotificationScreen(),
51     );
52   }
53 }
```

2. Source code notification_screen.dart

```
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:guided/main.dart';
5
6 class NotificationScreen extends StatefulWidget {
7   const NotificationScreen({super.key});
8
9   @override
10  State<NotificationScreen> createState() => _NotificationScreenState();
11 }
12
13 class _NotificationScreenState extends State<NotificationScreen> {
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     // Mendengarkan pesan saat aplikasi aktif
24     FirebaseMessaging.onMessage.listen((RemoteMessage message) {
25       RemoteNotification? notification = message.notification;
26       AndroidNotification? android = message.notification?.android;
27       // Jika notifikasi tersedia, tampilkan menggunakan notifikasi lokal
28       if (notification != null && android != null) {
29         FlutterLocalNotificationsPlugin().show(
30           notification.hashCode, // ID notifikasi (hashCode untuk unik)
31           notification.title, // Judul notifikasi
32           notification.body, // Isi notifikasi
33           NotificationDetails(
34             android: AndroidNotificationDetails(
35               channel.id,
36               channel.name,
37               channelDescription: channel.description,
38               color: const Color.fromARGB(255, 125, 196, 255),
39               icon: "@mipmap/ic_launcher",
40             ),
41           ),
42         );
43       }
44     });
45     // Menangani aksi ketika notifikasi dibuka
46     FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
47       RemoteNotification? notification = message.notification;
48       AndroidNotification? android = message.notification?.android;
49       // Jika notifikasi tersedia, tampilkan dialog
50       if (notification != null && android != null) {
51         showDialog(
52           context: context,
53           builder: (_) {
54             return AlertDialog(
55               title: Text(notification.title ?? ""), // Judul dialog
56               content: SingleChildScrollView(
57                 child: Column(
58                   crossAxisAlignment: CrossAxisAlignment.start,
59                   children: [Text(notification.body ?? "")], // Isi dialog
60                 ),
61               ),
62             );
63           },
64         );
65       }
66     });
67     // Memanggil metode untuk mengambil token FCM perangkat
68     getToken();
69   }
70
71   // Metode untuk mendapatkan token FCM
72   void getToken() async {
73     token = await FirebaseMessaging.instance
74       .getToken(); // Mendapatkan token FCM perangkat
75     print('FCM Token: $token'); // Menampilkan token di log
76   }
77
78   @override
79   Widget build(BuildContext context) {
80     return Scaffold(
81       appBar: AppBar(
82         title: const Text("Notification Screen"),
83         backgroundColor: const Color.fromARGB(255, 222, 135, 240),
84         centerTitle: true,
85       ),
86       body: const Center(
87         child: Text("Mendapatkan Notifikasi"),
88       ),
89     );
90   }
91 }
```

3. Output Tampilan awal dan Tampilan notifikasi masuk



Output Tampilan notifikasi ketika di buka

