LAPORAN PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK

MODUL 11 DATA STORAGE (BAGIAN 2)



Disusun Oleh : Dzikri Naufal Wisnu Pravida/2211104063 SE06-02

Asisten Praktikum : Muhammad Faza Zulian Aisyah Hasna Aulia

Dosen Pengampu : Yudha Islami Sulistya

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

PRAKTIKUM

A. GUIDED

1. Jenis Pesan Pada Firebase Cloud Messaging (FCM)

1.1. Notification Message

Pesan ini digunakan untuk langsung menampilkan notifikasi di perangkat pengguna. Cara kerja:

- Jika aplikasi berjalan di latar depan (foreground), pesan diteruskan ke fungsi onMessage untuk diproses.
- Jika aplikasi di latar belakang (background) atau dimatikan (terminated), notifikasi akan langsung muncul di tray notifikasi perangkat tanpa memerlukan kode tambahan.

1.2. Data Message

Pesan ini memberikan lebih banyak fleksibilitas karena isinya tidak langsung ditampilkan sebagai notifikasi.

Cara kerja:

- Pesan ini akan diproses oleh aplikasi, baik saat aplikasi aktif maupun tidak aktif.
- Untuk menampilkan notifikasi ke pengguna, Kita perlu menambahkan kode khusus, biasanya menggunakan plugin seperti flutter local notifications.

2. Hasil Praktikum

Source Code

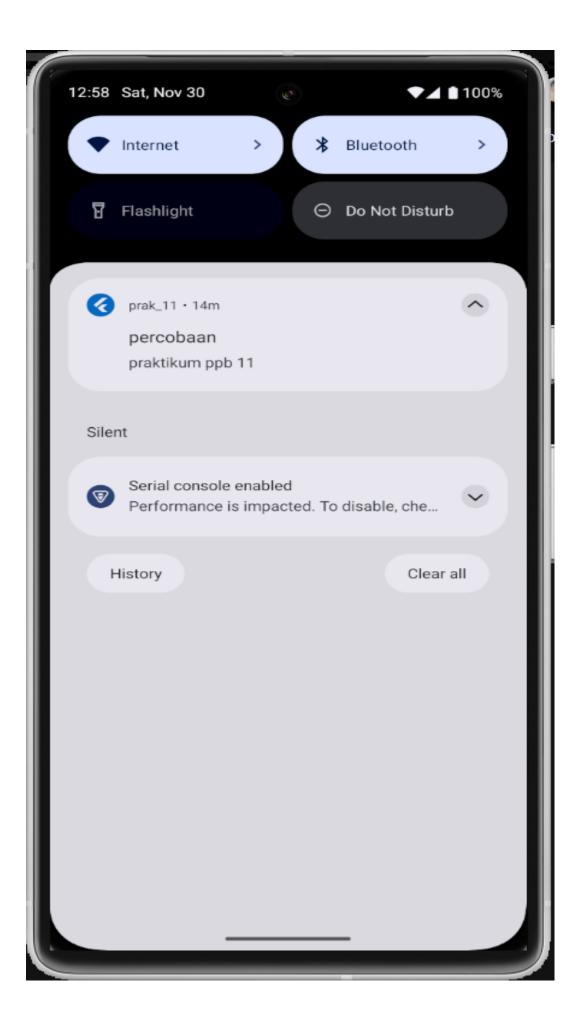
```
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:prak 11/my notification screen.dart';
String? token;
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.', //
  importance: Importance.high, // Prioritas
);
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  \label{lem:firebase} \textbf{Firebase} \textbf{Messaging.} on \textbf{Background} \textbf{Message} (\_\texttt{firebase} \textbf{MessagingBackgroundHandler});
  await FlutterLocalNotificationsPlugin()
      .resolvePlatformSpecificImplementation
          AndroidFlutterLocalNotificationsPlugin>()
      ?.createNotificationChannel(channel);
  await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
    alert: true,
    badge: true,
    sound: true,
  );
  runApp(const MyApp());
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(),
    );
  }
```

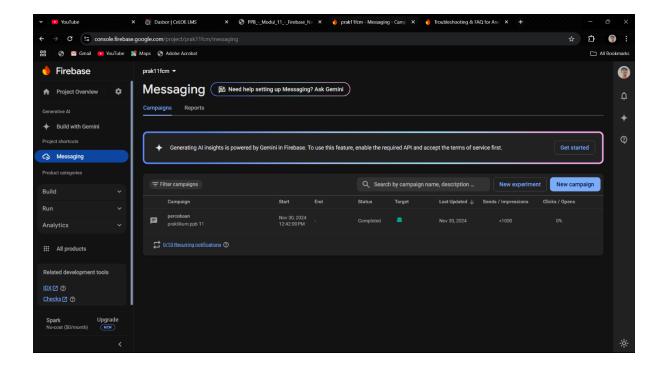
```
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
        import package:fideta_/main.dart';
        class MyNotificationScreen extends StatefulWidget {
  const MyNotificationScreen({super.key});
            @override
            State<MyNotificationScreen> createState() => _MyNotificationScreenState();
                    MyNotificationScreenState extends State<MyNotificationScreen> {
            @override
void initState()
                super.initState();
                       Membuat pengaturan inisialisasi notifikasi untuk Android
initializationSettingsAndroid =
  const AndroidInitializationSettings('@mipmap/ic_launcher');
19
20
21
22
               var initializationSettings =
    InitializationSettings(android: initializationSettingsAndroid);
FlutterLocalNotificationSPlugin().initialize(initializationSettings);
234567890123333333344444444455555555555555
                // Mendengarkan pesan saat aplikasi aktif
FirebaseMessaging.onMessage.listen((RemoteMessage message) {
   RemoteNotification? notification = message.notification;
   AndroidNotification? android = message.notification?.android;
                   android: AndroidNotificationDetails(
  channel.id,
                                   channel.name,
channelDescription: channel.description,
color: Colors.blue,
icon: "@mipmap/ic_launcher",
               });
// Menangani aksi ketika notifikasi dibuka
onMessageOpenedApp.lister
                // menangani aksi ketika notifikasi dibuka
FirebaseMessagging.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 dialog
  content: SingleChildScrollView(
    child: Column(
56
57
58
59
60
                                           crossAxisAlignment: CrossAxisAlignment.start,
children: [Text(notification.body ?? "")], // Isi dialog
                                    ),
                   } );
                               );
               // Memanggil metode untuk mengambil token FCM perangkat
getToken();
            void getToken() async {
   String? token = await FirebaseMessaging.instance.getToken();
   print(token);
            Widget build(BuildContext context) {
              return Scaffold(
appBar: AppBar(
title: const Text("Notification screen"),
backgroundColor: Colors.deepPurple,
                    ),
body: Center(
child: Text("halaman penerima notifikasi"),
```

Output:



Saat mencoba untuk mendapatkan notifikasi:





B. UNGUIDED

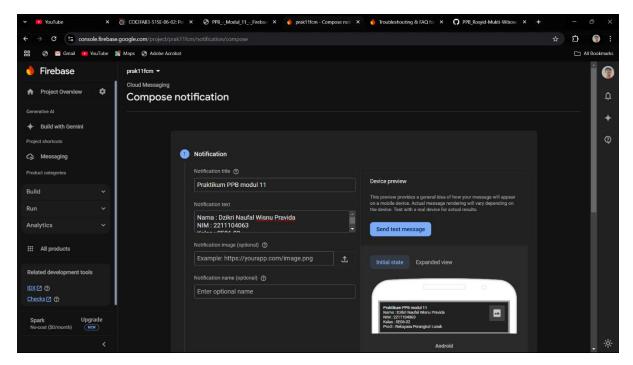
1. Soal Studi Case

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

Sourcecode

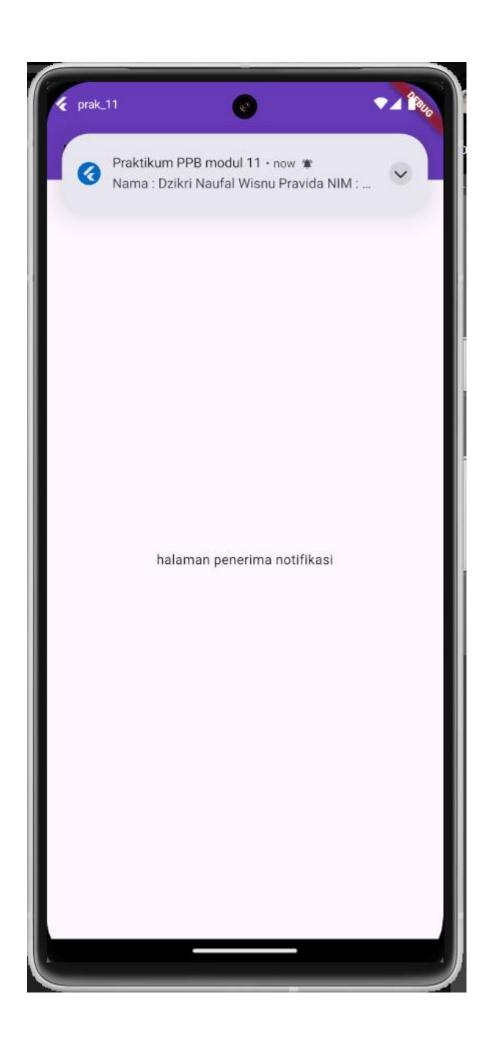
Tidak menambahkan coding apa apa didalamnya, hanya modifikasi isi dari pesan notifikasinya.

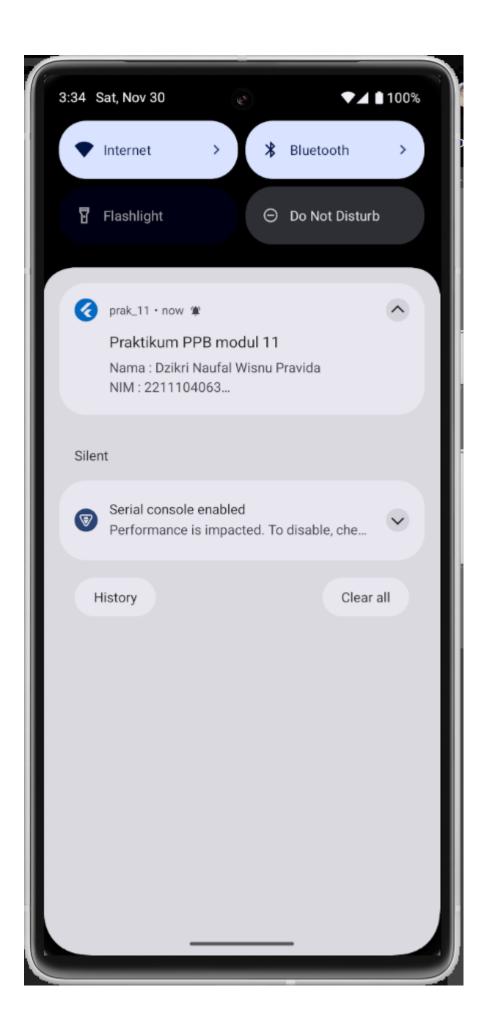
Screenshoot Output



Penjelasan : disini kita hanya mengisi notification tittle dengan text nya yang kemudian langsung 'send test message'

Maka akan beroutput seperti ini:







Deskripsi Program

Jadi di unguided ini tidak mengubah struktur isi dari program yang ada didalam file project flutter. Kita hanya mengisi notifikasi yang akan dikirimkan dengan menggunakan format sesuai yang ada pada soal unguided, yang mana ketika kita tekan notifikasinya akan langsung muncul isi yang kita isi.