

LAPORAN PRAKTIKUM



NIM : 2003071

Nama : Hakim Asrori

Kelas : D3TI.2C

Mata Kuliah : [Pemrograman Perangkat Bergerak \(TIU3403\)](#)

Praktikum ke / Judul : 11 / PUSH NOTIFICATION

Tanggal Praktikum : 19 Mei 2022

Dosen Pengampu : Fachrul Pralienka Bani Muhamad, S.ST.,
M.Kom

**PROGRAM STUDI D3 TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU
2022**



A. Tujuan Praktikum

Tujuan Umum

Mahasiswa dapat memanfaatkan layanan Firebase pada penyajian notifikasi di aplikasi mobile

Tujuan Khusus

Mahasiswa mampu:

1. Menjelaskan kegunaan notifikasi
2. Menyajikan tampilan notifikasi
3. Mengimplementasikan service yang disediakan Firebase untuk menyajikan notifikasi

B. Rangkuman Teori Singkat

Kegunaan Notifikasi pada Aplikasi Mobile

Notifikasi adalah pesan yang ditampilkan oleh Android di luar UI aplikasi yang berfungsi sebagai pemberi pengingat, komunikasi dari orang lain atau informasi tepat waktu lainnya dari aplikasi. Pengguna dapat mengetuk notifikasi untuk membuka aplikasi atau melakukan tindakan langsung dari notifikasi.

Penjelasan Cara Kerja Push Notification

Cara kerja dari push notifikasi untuk web dan mobile terbilang cukup mirip, berikut ini cara kerjanya :

1. Meminta persetujuan pada user untuk pengiriman push notifikasi
2. Mengirim pesan push



C. Langkah dan Hasil Pelaksanaan Praktikum

Langkah-langkah praktikum Push Notification Firebase

<https://www.youtube.com/watch?v=2tjuUwNx6qk>

1. Buka halaman firebase console melalui url berikut.
<https://console.firebase.google.com/u/0/>
2. Buat project baru terlebih dahulu di console firebase
3. Setelah buat project firebase, selanjutnya buat project flutter baru.
4. Sebelum melakukan perubahan kode program. Kita tambahkan terlebih dahulu library yang dibutuhkan seperti firebase_core dan firebase_messaging dengan mendefinisikan nama library tersebut seperti dibawah ini didalam file pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  # The following adds the Cupertino Icons font to your application.  
  # Use with the CupertinoIcons class for iOS style icons.  
  cupertino_icons: ^1.0.2  
  firebase_messaging: ^11.4.1  
  firebase_core: ^1.18.0
```

5. Selanjutnya secara otomatis flutter akan menambahkan library yang didefinisikan diatas kedalam project
6. Selanjutnya install library flutterfire_cli secara global menggunakan terminal dengan perintah berikut

```
C:\Users\ACER\OneDrive\Kuliah\Semester 4\Pemrograman Perangkat Bergerak\Kode Program\Tugas Praktikum 30 Mei 2022\Praktikum 11\firebase_cloud_messaging> dart pub global activate flutterfire_cli
```

Penjelasan: library diatas berguna untuk menghubungkan antara project flutter dengan project dari firebase sehingga saling terintegrasi

7. Selanjutnya jalankan perintah berikut untuk mengintegrasikan project firebase dengan project flutternya

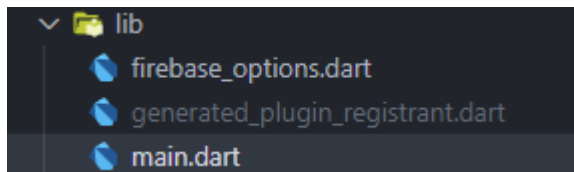
```
C:\Users\ACER\OneDrive\Kuliah\Semester 4\Pemrograman Perangkat Bergerak\Kode Program\Tugas Praktikum 30 Mei 2022\Praktikum 11\firebase_cloud_messaging> flutterfire configure
```

8. Selanjutnya akan ditampilkan project firebase yang telah dibuat, lalu pilih project dan tunggu hingga selesai



```
i Found 7 Firebase projects.  
? Select a Firebase project to configure your Flutter application with >  
coba-laravel-1b3da (coba-laravel)  
flutter-login-with-fireb-60255 (flutter-login-with-firebase)  
flutter-storage-41e9c (flutter-storage)  
my-firebase-project-e2f7a (my-firebase-project)  
nama-apa (nama apa)  
project-cloud-messaging-5aea7 (project-cloud-messaging)  
punya-lely (punya-lely)  
<create a new project>
```

9. Setelah selesai proses diatas, akan dibuatkan otomatis sebuah file baru yaitu dengan nama `firebase_otomatis.dart`



10. Selanjutnya tambahkan kode program berikut pada file `main.dart`

```
import 'package:flutter/material.dart';  
import 'package:firebase_core/firebase_core.dart';  
import 'firebase_options.dart';  
import 'package:firebase_messaging/firebase_messaging.dart';
```



```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp(  
    options: DefaultFirebaseOptions.currentPlatform,  
  );  
  FirebaseMessaging messaging = FirebaseMessaging.instance;  
  
  NotificationSettings settings = await messaging.requestPermission(  
    alert: true,  
    announcement: false,  
    badge: true,  
    carPlay: false,  
    criticalAlert: false,  
    provisional: false,  
    sound: true,  
  );  
  
  print('User granted permission: ${settings.authorizationStatus}');  
  
  FirebaseMessaging.onMessage.listen((RemoteMessage message) {  
    print('Got a message whilst in the foreground!');  
    print('Message data: ${message.data}');  
  
    if (message.notification != null) {  
      print('Message also contained a notification: ${message.notification}');  
    }  
  });  
  
  runApp(MyApp());  
}
```

```
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      theme: ThemeData(  
        primarySwatch: Colors.blue,  
      ), // ThemeData  
      home: const MyHomePage(title: 'Flutter Demo Home Page'),  
    ); // MaterialApp  
  }  
}
```



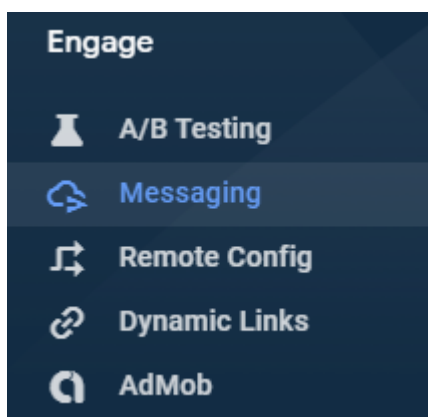
```
class MyHomePage extends StatefulWidget {  
  const MyHomePage({Key? key, required this.title}) : super(key: key);  
  
  final String title;  
  
  @override  
  State<MyHomePage> createState() => _MyHomePageState();  
}
```

```
class _MyHomePageState extends State<MyHomePage> {  
  int _counter = 0;  
  
  void _incrementCounter() {  
    setState(() {  
      _counter++;  
    });  
  }  
}
```

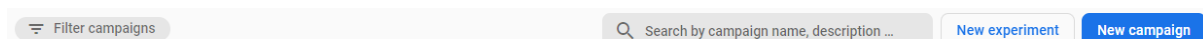


```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ), // AppBar
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          const Text(
            'You have pushed the button this many times:',
          ), // Text
          Text(
            '$_counter',
            style: Theme.of(context).textTheme.headline4,
          ), // Text
        ], // <Widget>[]
      ), // Column
    ), // Center
    floatingActionButton: FloatingActionButton(
      onPressed: _incrementCounter,
      tooltip: 'Increment',
      child: const Icon(Icons.add),
    ), // FloatingActionButton
  ); // Scaffold
}
```

11. Selanjutnya buat pesan notifikasi menggunakan service yang disediakan firebase yaitu
Firebase Cloud Messaging



12. Selanjutnya buat campaign baru





13. Selanjutkan isikan seperti berikut

1 Notification

Notification title ?

Test

Notification text

Halo Hakim Asrori, ini ada pesan baru

Notification image (optional) ?

Example: <https://yourapp.com/image.png>



Notification name (optional) ?

Enter optional name

2 Target

User segment

Topic

Target user if...

App

[firebase_cloud_messaging \(android\)](#) (com.example.firebase_cloud_messagi...

and

[firebase_cloud_messaging_tugas \(android\)](#) (com.example.firebase_cloud_...

iOS- [firebase_cloud_messaging \(ios\)](#) (com.example.firebaseCloudMessaging)

iOS- [firebase_cloud_messaging_tugas \(ios\)](#) (com.example.firebaseCloudMessa...

</> [firebase_cloud_messaging \(web\)](#)

[Target another app](#)

Next

3 Scheduling

Send to eligible users

Now

Next



4 Conversion events (optional)

Sent | | Opened | |

Analytics Label ?

Next

5 Additional options (optional)

All fields optional

Android Notification Channel ?

Custom data ?

Key

Value

Sound

Disabled

Expires ?

4

Weeks

14. Setelah itu tekan tombol review dibawah

Save as draft


Review

15. Selanjutnya akan muncul pop up box seperti dibawah ini, lalu tekan tombol publish




Review message


Notification content

 Halo Hakim Asrori, ini ada pesan baru

Target

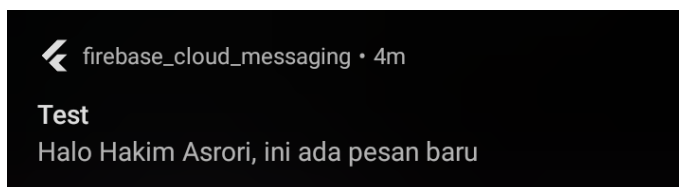
 User segment matching one targeting criterion

Scheduling

 Send now

Cancel **Publish**

16. Tunggu beberapa saat akan muncul notifikasi pada emulator mobile tersebut, dan terdapat pesan "broadcast received for message" didalam console



```
D/FLTFireMsgReceiver(19764): broadcast received for message
W/FLTFireMsgService(19764): A background message could not be handled in Dart
as no onBackgroundMessage handler has been registered.
W/FirebaseMessaging(19764): Unable to log event: analytics library is missing
W/FirebaseMessaging(19764): Missing Default Notification Channel metadata in
AndroidManifest. Default value will be used.
```

17. Selesai.



D. Langkah dan Hasil Latihan

Latihan Implementasi Push Notification Firebase pada Studi Individu

1. Buat flutter baru
2. Selanjutnya ubah kode program main.dart

```
import 'dart:convert';

import 'package:firebase_cloud_messaging_tugas/views/coba_page.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:firebase_cloud_messaging_tugas/constants/firebase_helper.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Firebase Messaging',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ), // ThemeData
      debugShowCheckedModeBanner: false,
      home: const MyHomePage(),
    ); // MaterialApp
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key}) : super(key: key);

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}
```



```
class _MyHomePageState extends State<MyHomePage> {  
  String dataNotification = "Example Data";  
  String titleNotification = "Example Title";  
  String bodyNotification = "Example Body";  
  
  @override  
  void initState() {  
    final firebaseMessaging = FCM();  
    firebaseMessaging.setNotification();  
    firebaseMessaging.dataControl.stream.listen(_updateDataNotification);  
    firebaseMessaging.titleControl.stream.listen(_updateTitleNotification);  
    firebaseMessaging.bodyControl.stream.listen(_updateBodyNotification);  
  
    super.initState();  
  }  
}
```

```
_updateDataNotification(String message) {  
  var temporaryData = jsonDecode(message);  
  setState(() => dataNotification = temporaryData['message']);  
}
```

```
_updateTitleNotification(String message) =>  
  setState(() => titleNotification = message);  
_updateBodyNotification(String message) =>  
  setState(() => bodyNotification = message);
```

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    floatingActionButton: FloatingActionButton(  
      child: Icon(Icons.add),  
      onPressed: () {  
        Navigator.push(context, MaterialPageRoute(builder: (context) {  
          return CobaPage();  
        })); // MaterialPageRoute  
      },  
    ), // FloatingActionButton  
    appBar: AppBar(  
      title: Text("Firebase Messaging"),  
    ), // AppBar  
  );  
}
```



```
body: Container(  
  margin: EdgeInsets.symmetric(  
    horizontal: MediaQuery.of(context).size.width * 0.15), // EdgeInsets.symmetric  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      Row(  
        children: [  
          Text(  
            "Title : $titleNotification",  
            style: TextStyle(fontSize: 25),  
          ), // Text  
        ],  
      ), // Row  
      Row(  
        children: [  
          Text(  
            "Body : $bodyNotification",  
            style: TextStyle(fontSize: 25),  
          ), // Text  
        ],  
      ), // Row  
    ],  
  ),  
),
```

```
      Row(  
        children: [  
          Text(  
            "Data : $dataNotification",  
            style: TextStyle(fontSize: 25),  
          ), // Text  
        ],  
      ), // Row  
    ],  
  ), // Column  
), // Container  
); // Scaffold  
}
```

3. Setelah itu, buat file baru dengan nama firebase_helper.dart

```
import 'dart:async';  
import 'dart:convert';  
  
import 'package:firebase_messaging/firebase_messaging.dart';  
  
class FCM {  
  FirebaseMessaging _firebaseMessaging = FirebaseMessaging.instance;  
  
  final dataControl = StreamController<String>.broadcast();  
  final titleControl = StreamController<String>.broadcast();  
  final bodyControl = StreamController<String>.broadcast();  
}
```



```
setNotification() async {  
  NotificationSettings settings = await _firebaseMessaging.requestPermission(  
    alert: true, sound: true, badge: true, provisional: false);  
  
  if (settings.authorizationStatus == AuthorizationStatus.authorized) {  
    print("user granted permission");  
  
    foregroundNotification();  
    backgroundNotification();  
    terminateNotification();  
  }  
  
  final token = _firebaseMessaging  
    .getToken()  
    .then((value) => print("FCM Token : $value"));  
}
```

```
void foregroundNotification() {  
  FirebaseMessaging.onMessage.listen((event) {  
    print("Foreground Notification");  
  
    if (event.data.isNotEmpty) {  
      dataControl.sink.add(jsonEncode(event.data));  
    }  
  
    if (event.notification != null) {  
      titleControl.sink.add(event.notification!.title!);  
      bodyControl.sink.add(event.notification!.body!);  
    }  
  });  
}
```

```
void backgroundNotification() {  
  FirebaseMessaging.onMessageOpenedApp.listen((event) {  
    print("Background Notification");  
  
    if (event.data.isNotEmpty) {  
      dataControl.sink.add(jsonEncode(event.data));  
    }  
  
    if (event.notification != null) {  
      titleControl.sink.add(event.notification!.title!);  
      bodyControl.sink.add(event.notification!.body!);  
    }  
  });  
}
```



```
void terminateNotification() async {  
    RemoteMessage? initialMessage =  
        await FirebaseMessaging.instance.getInitialMessage();  
  
    if (initialMessage != null) {  
        print("Terminate Notification");  
  
        if (initialMessage.data.isNotEmpty) {  
            dataControl.sink.add(jsonEncode(initialMessage.data));  
        }  
  
        if (initialMessage.notification != null) {  
            titleControl.sink.add(initialMessage.notification!.title!);  
            bodyControl.sink.add(initialMessage.notification!.body!);  
        }  
    }  
}
```

```
@override  
void dispose() {  
    dataControl.close();  
    titleControl.close();  
    bodyControl.close();  
}
```

4. Setelah selesai, lakukan testing
 - a. Foreground Message



ppb / foreground

POST <https://fcm.googleapis.com/fcm/send> [Send](#)

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings Cookies

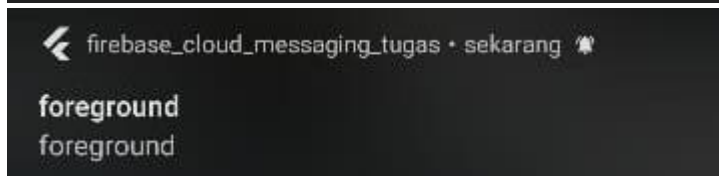
☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL [JSON](#) [Beautify](#)

```
2  {
3    "notification": {
4      "body": "foreground",
5      "title": "foreground",
6      "sound": "default"
7    },
8    "priority": "high",
9    "data": {
10     "click_action": "FLUTTER_NOTIFICATION_CLICK",
11     "message": "foreground",
12     "type": "type message"
13   }
14 }
```

Body Cookies Headers (12) Test Results [Status: 200 OK](#) [Time: 179 ms](#) [Size: 703 B](#) [Save Response](#)

[Pretty](#) [Raw](#) [Preview](#) [Visualize](#) [JSON](#) [Copy](#)

```
1  {
2    "multicast_id": 2763444147840991517,
3    "success": 1,
4    "failure": 0,
5    "canonical_ids": 0,
6  }
```



Firestore Messaging

Title : foreground
Body : foreground
Data : foreground

b. Background Message



ppb / background

POST https://fcm.googleapis.com/fcm/send Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

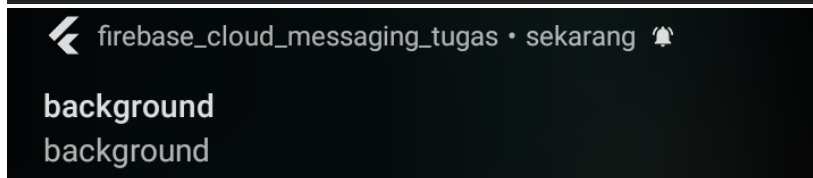
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "notification": {
3     "body": "background",
4     "title": "background",
5     "sound": "default"
6   },
7   "priority": "high",
8   "data": {
9     "click_action": "FLUTTER_NOTIFICATION_CLICK",
10    "message": "background",
11    "type": "type: message"
12  }
13 }
```

Body Cookies Headers (12) Test Results Status: 200 OK Time: 82 ms Size: 703 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "multicast_id": 5578565773283930037,
3   "success": 1,
4   "failure": 0,
5   "canonical_ids": 0
6 }
```



Firestore Messaging

Title : background
Body : background
Data : background

c. Terminate Message



The screenshot shows a Postman interface for a POST request to `https://fcm.googleapis.com/fcm/send`. The request body is a JSON object:

```
{  "notification": {    "body": "terminate",    "title": "terminate",    "sound": "default"  },  "priority": "high",  "data": {    "click_action": "FLUTTER_NOTIFICATION_CLICK",    "message": "terminate",    "type": "type message"  }}
```

The response status is `200 OK` with a time of `118 ms` and a size of `702 B`. The response body is:

```
{  "multicast_id": 325463143777871642,  "success": 1,  "failure": 0,  ...}
```

The screenshot shows a mobile app interface with a notification bar at the top. The notification text is `firebase_cloud_messaging_tugas • sekarang` followed by a bell icon. Below the notification, the text `terminate` is displayed twice.

Firestore Messaging

Title : terminate
Body : terminate
Data : terminate

5. Selesai.



Program Studi D3 Teknik Informatika
Jurusan Teknik Informatika
Politeknik Negeri Indramayu

NIM : 2003071

Nama : Hakim Asrori

E. Hasil dan Penjelasan Tugas



F. Kesimpulan

Notifikasi dapat digunakan untuk memberikan informasi kepada pengguna, dan ini sangat menunjang dalam pengembangan aplikasi mobile.

G. Referensi

- Alberto Miola. "Flutter Complete Reference Create Beautiful, Fast and Native Apps for Any Device". Independently Published. 2020.
- Simone Alessandria, Brian Kayfirz. "Flutter Cookbook: Over 100 proven techniques and solutions for app development with Flutter 2.2 and Dart". Packt Publishing. Birmingham - Mumbai. 2021.
- Dieter Meiller. "Modern App Development with Dart and Flutter 2: A Comprehensive Introduction to Flutter". Walter de Gruyter GmbH. Berlin - Boston. 2021.
- Priyanka Tyagi. "Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop". CRC Press Taylor & Francis Group, LLC. London - New York. 2022.
- <https://developer.android.com/guide/topics/ui/notifiers/notifications?hl=id>