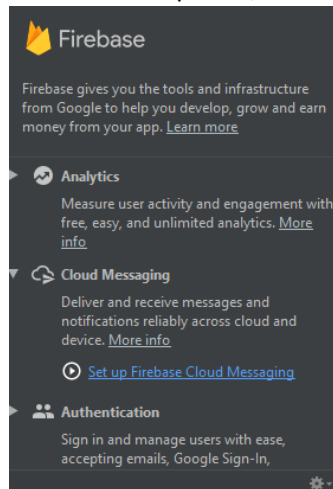
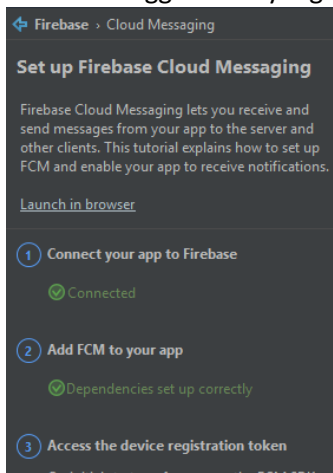


1. Koneksi ke Firebase Cloud Messaging & Menerima Push Notification dari Firebase Console
  - a. Buat project baru dengan nama "Platform 05 FCM" dengan sebuah empty activity bernama MainActivity. Setelah proses building gradle selesai, klik menu Tools > Firebase. Pastikan Plugin Firebase sudah aktif dan terinstal, jika tidak, menu Firebase pada Tools tidak akan muncul. Cek di Preferences > Plugins.
  - b. Di tab Firebase yang muncul, klik tautan Set up Firebase Cloud Messaging. Jika tidak melihat tautan yang ditunjukkan oleh tanda panah, klik dulu di bagian Cloud Messaging.



Selanjutnya klik tombol Set up Firebase Cloud Messaging & klik Connect to Firebase. Kita akan dibawa ke halaman login dengan menggunakan akun Gmail.

- c. Setelah berhasil login akan muncul jendela baru di Android Studio yang meminta kita membuat Firebase project baru atau menggunakan yang sudah ada



Ada kalanya, Firebase mengalami error koneksi, coba lagi dengan klik ulang Connect to Firebase pada panel Assistant, lalu Sync. Klik tombol Add FCM to your app lalu Klik tombol Accept Changes untuk mengaplikasikan penambahan dependensi ke file build.gradle. Jika kemudian muncul error Failed to resolve: firebase-messaging-15.0.0, maka buka file gradle (app/build.gradle) dan ubah ke versi :17.3.3

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0-alpha1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.google.firebase:firebase-messaging:17.3.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
```

- d. Download file config google-services.json, lalu salin isinya ke dalam file google-services.json yang sudah ada di folder app. Sync ulang dengan meng-klik Connect to Firebase. buat sebuah file Java baru dan beri nama MyFirebaseMessagingService yang meng-extend class FirebaseMessagingService. Pada file AndroidManifest.xml, tambahkan <service> di dalam tag <application>

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.yogatama.myapplication">

    <supports-screens android:smallScreens="false" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/icon"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".Home">
            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />

                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MainActivity" />
        <activity android:name=".DetailActivity" />

        <service
            android:name=".MyFirebaseMessagingService"
            android:stopWithTask="false">
            <intent-filter>
                <action
                    android:name="com.google.firebase.MESSAGING_EVENT" />
            </intent-filter>
        </service>

        <activity android:name=".Main2Activity"></activity>
    </application>

</manifest>
```

pada file MyFirebaseMessagingService.java, tambahkan method baru bernama onNewToken() untuk menampilkan token pada Logcat.

```
package com.example.yogatama.myapplication;

import android.util.Log;

import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.messaging.FirebaseMessagingService;
```

```
import com.google.firebase.messaging.RemoteMessage;

public class MyFirebaseMessagingService extends
FirebaseMessagingService {
    private static final String TAG = "FCM_GUE";

    @Override
    public void onNewToken(String s) {
        super.onNewToken(s);
        Log.e("NEW_TOKEN", s);
    }
}
```

Jalankan aplikasi dan amati pada panel Logcat dan cari NEW\_TOKEN.

Buat sebuah method bernama onMessageReceived() dalam class MyFirebaseMessagingService.java

```
@Override
public void onMessageReceived(RemoteMessage remoteMessage) {
    super.onMessageReceived(remoteMessage);

    Log.d(TAG, "Pengirim: " + remoteMessage.getFrom());
}
```

Jangan lupa definisikan TAG, Lalu jalankan aplikasi kembali. Kembali ke Firebase Console, klik tombol Send your first message. Klik Send message dan perhatikan bahwa pada device/ emulator notifikasi telah terkirim tetapi tidak memicu notifikasi.

2. Menerima Push Message dari FCM dengan tipe Data Message melalui Postman
  - a. Dapatkan Server key dari Firebase console. Download & install Postman sesuai sistem operasi. Setelah terinstall, buat request Post.

The screenshot shows the Postman interface for a POST request to `https://fcm.googleapis.com/fcm/send`. The **Headers** tab is selected, showing two headers: `Authorization` with value `key=AAAAAT6FFtIQ:APA91bE9OCz44ujm9ILKWr...` and `Content-Type` with value `application/json`. The **Body** tab is also selected, showing a JSON payload for a push message. The payload includes a `to` field with a device ID, a `collapse_key` of `"type_a"`, and a `data` object with `body`, `title`, and `key` fields.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Authorization	key=AAAAAT6FFtIQ:APA91bE9OCz44ujm9ILKWr...	
<input checked="" type="checkbox"/> Content-Type	application/json	
Key	Value	Description

```
{
  "to": "eqUW2-MZq0Y:APA91bEfGaP-FwIlTnm6fYqICznmrUZVtZu-uDu9GH2HZ0hNmgUhj4tmBsnIsG0tzwrY3kke-UhZCZ6Mux33q90MKCTfuPASjVboDMhf910wLw3oXICGkyz_WBNFLuRijlLmvDLht5_w",
  "collapse_key": "type_a",
  "data": {
    "body": "AUTO2000",
    "title": "Mohon maaf, anda belum terdaftar dalam list antrian service. Terimakasih",
    "key_1": "testing key1",
    "key_2": "testing key2"
  }
}
```

buka file MyFirebaseMessagingService.java, lalu jalankan aplikasi via emulator/ device

```

@Override
public void onMessageReceived(RemoteMessage remoteMessage) {
    super.onMessageReceived(remoteMessage);

    Log.d(TAG, "Pengirim: " + remoteMessage.getFrom());

    if (remoteMessage.getData().size() > 0) {
        Log.d(TAG, "Pesennya bang: " +
            remoteMessage.getData().get("body"));
    }
}

```

Pastikan response dari server Google FCM sukses (success: 1).

Pretty Raw Preview JSON

```

1 {
2   "multicast_id": 8497640912131927360,
3   "success": 1,
4   "failure": 0,
5   "canonical_ids": 0,
6   "results": [
7     {
8       "message_id": "0:1539755673072823%97a27b38cccfb49c"
9     }
10  ]
11 }

```

perhatikan terdapat log Data message

Logcat

Samsung Nexus Android 4.4.2 com.example.yogatama.myapplication Verbose Q pesel

10-17 12:54:33.291 1769-1853/com.example.yogatama.myapplication D/FCM\_GUE: Pengirim: 342008116356

Pesennya bang: AUTO2000

### 3. Menghandle Push Notification untuk Foreground dan Background

- a. Buat icon notifikasi melalui File > Image Asset, pilih Notification Icons dan nama ic\_stat\_name, Buat class baru bernama MyNotificationManager.

```

4. package com.example.yogatama.myapplication;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.media.RingtoneManager;
import android.net.Uri;
import android.support.v4.app.NotificationCompat;

public class MyNotificationManager {
    private Context mContext;
    private static MyNotificationManager mInstance;

    private MyNotificationManager(Context context){
        mContext=context;
    }

    public static synchronized MyNotificationManager
    getInstance(Context context){
        if (mInstance==null){
            mInstance=new MyNotificationManager(context);
        }
    }
}

```

```

    }
    return mInstance;
}

public void displayNotification(String title, String body){
    String CHANNEL_ID = "my_channel_01";
    Intent resultIntent = new Intent(mCtx,
Main2Activity.class);
    resultIntent.putExtra("wira",title);
    resultIntent.putExtra("ganteng",body);

    PendingIntent pendingIntent =
PendingIntent.getActivity(mCtx, 0, new
Intent[]{resultIntent},PendingIntent.FLAG_ONE_SHOT);

    Uri defaultSoundUri=
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION)
;

    NotificationCompat.Builder mBuilder= new
NotificationCompat.Builder(mCtx,CHANNEL_ID)
        .setSmallIcon(R.drawable.ic_stat_name1)
        .setContentTitle(title)
        .setContentText(body)
        .setAutoCancel(true)
        .setSound(defaultSoundUri)
        .setContentIntent(pendingIntent);

    NotificationManager
mNotifyMgr=(NotificationManager)mCtx.getSystemService(Context.NO
TIFICATION_SERVICE);
    if (mNotifyMgr != null){
        mNotifyMgr.notify(1,mBuilder.build());
    }

}
}

```

Pada class MyFirebaseMessagingService, panggil method displayNotification()

```

@Override
public void onMessageReceived(RemoteMessage remoteMessage) {
    super.onMessageReceived(remoteMessage);

    Log.d(TAG, "Pengirim: " + remoteMessage.getFrom());

    if (remoteMessage.getData().size() > 0) {
        Log.d(TAG, "Pesennya bang: " +
remoteMessage.getData().get("body"));
    }

    MyNotificationManager.getInstance(this).displayNotification(
        remoteMessage.getData().get("body"),
        remoteMessage.getData().get("title")
    );
}

```

Lalu jalankan aplikasi via device/ emulator

