



# Push Notification

PBP, SEMESTER GANJIL T.A. 2022-  
2023



Pemrograman Berbasis Platform

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ATMA JAYA YOGYAKARTA

## TUJUAN

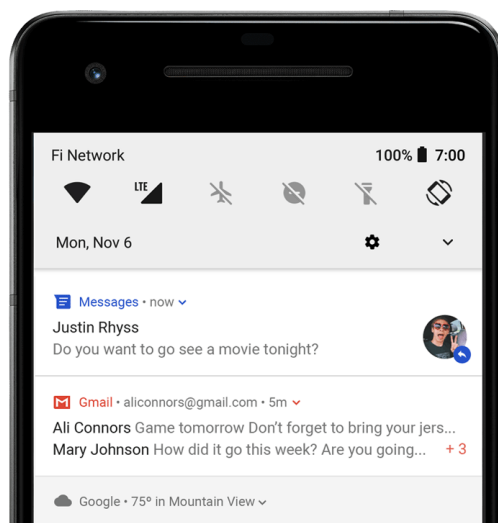
Setelah menyelesaikan modul ini, praktikan diharapkan mampu:

1. Memahami apa yang dimaksudkan dengan push notification dan cara mengimplementasikannya.
2. Memahami apa yang dimaksudkan dengan push notification memakai firebase dan cara mengimplementasikannya.

## TEORI

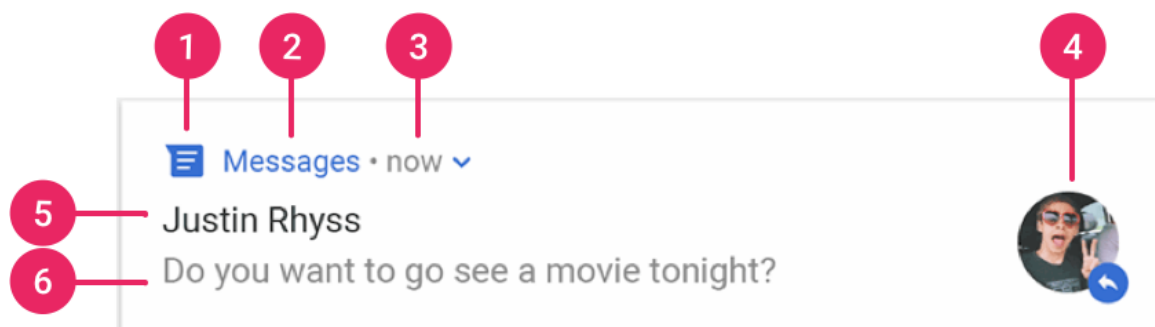
### A. Notifikasi

Notifikasi adalah pesan yang ditampilkan oleh Android diluar UI aplikasi anda untuk memberikan pengingat, komunikasi dari orang lain, atau informasi tepat waktu lainnya dari aplikasi. Simplenya adalah sebuah pengingat yang muncul di luar aplikasi anda ketika sedang melakukan aktifitas luar aplikasi yang dijadikan notification. Lalu push notification adalah pesan yang didapatkan diluar UI aplikasi, tetapi akan didapatkan terus menerus selama aplikasi yang dipasang tetap ada. Contoh dari Push Notification seperti berikut.



*Gambar 1 Push Notification di bar notif*

Anatomi pada Notification adalah :

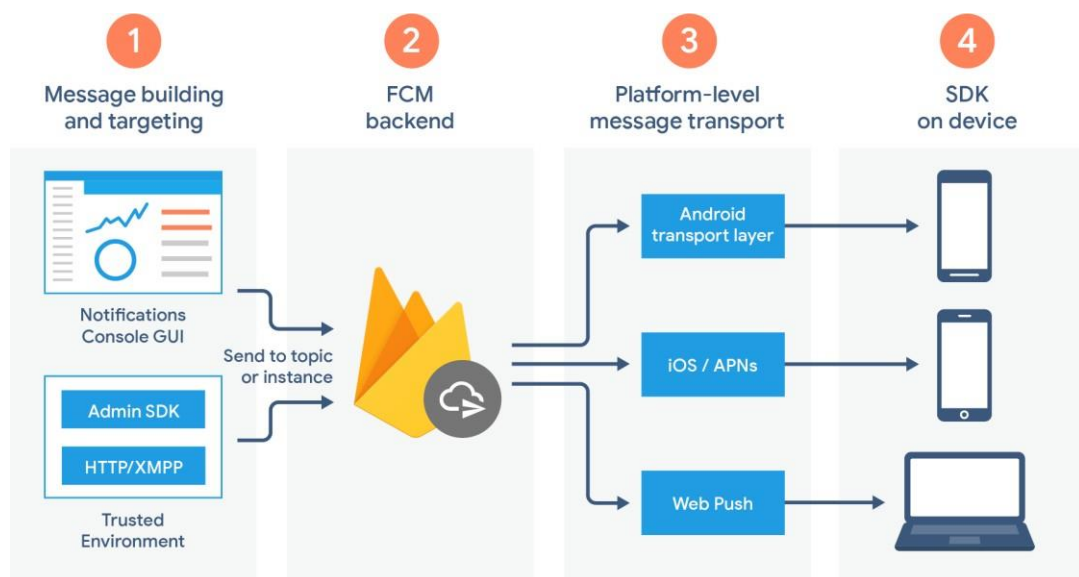


Bagian paling umum dari notifikasi ditunjukkan dalam gambar 7 sebagai berikut:

1. Ikon kecil: Wajib diisi dan disetel dengan [`setSmallIcon\(\)`](#).
2. Nama aplikasi: Disediakan oleh sistem.
3. Stempel waktu: Ini diberikan oleh sistem, tetapi Anda dapat menggantinya dengan [`setWhen\(\)`](#) atau menyembunyikannya dengan [`setShowWhen\(\)`](#).
4. Ikon besar: Ini opsional (biasanya hanya digunakan untuk foto kontak; jangan gunakan untuk ikon aplikasi Anda) dan disetel dengan [`setLargeIcon\(\)`](#).
5. Judul: Ini opsional dan disetel dengan [`setContentTitle\(\)`](#).
6. Teks: Ini opsional dan disetel dengan [`setContentText\(\)`](#).

## B. Firebase

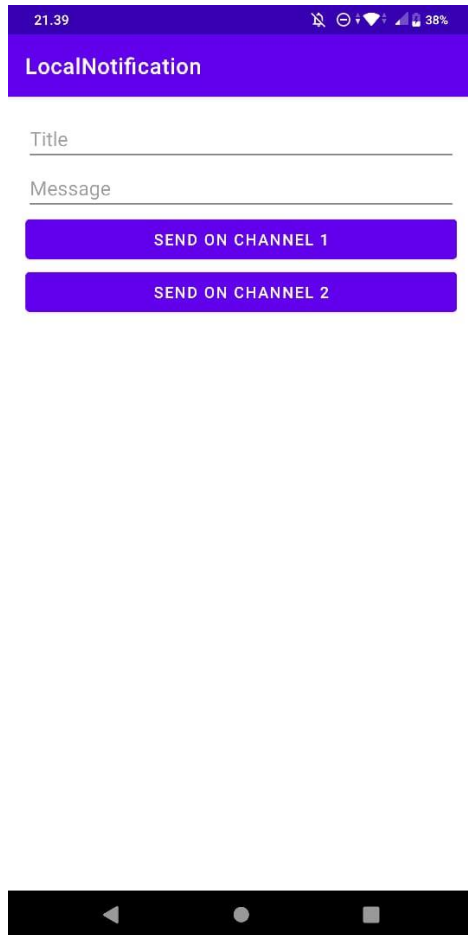
Firebase adalah layanan dari Google untuk memberikan kemudahan bahkan mempermudah para developer aplikasi mengembangkan aplikasi tersebut baik web, iOS, Android, IoT, maupun Kecerdasan Buatan. Firebase merupakan juga sebuah Backend as a Service (BaaS) yang NoSQL. Salah satu fitur dari Firebase adalah FCM (Firebase Cloud Messaging), yaitu layanan koneksi untuk mengirim pesan serta notifikasi dari satu device ke multiple device lainnya baik Android, iOS, dan web tanpa keluar biaya alias Free. Pesan notifikasi ini terintegrasi sepenuhnya dengan Google Analytics for Firebase, sehingga kamu memiliki akses pada interaksi dan tracking konversi secara detail. Nah, Anda dapat memantau suatu efektivitas dari satu dashboard tidak perlu coding atau membuat program sendiri. Arsitektur dari FCM sebagai berikut :



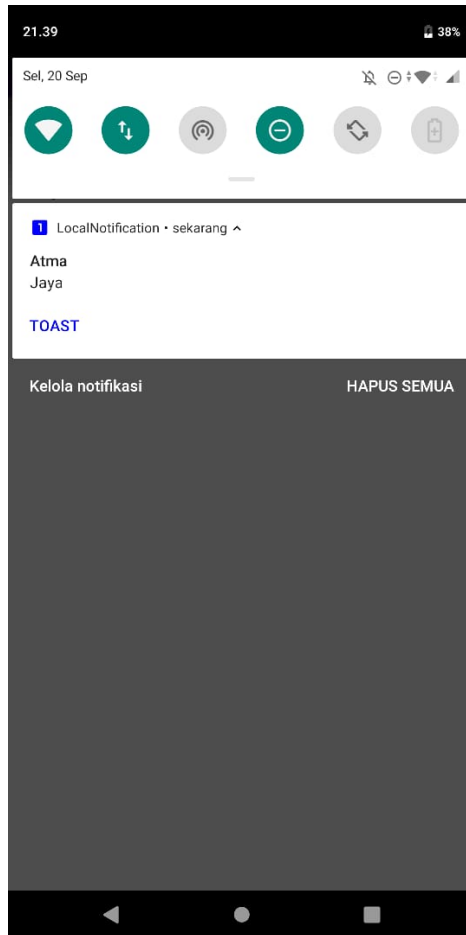
Gambar 2 Arsitektur FCM

### Guided

Pada guided ini akan mempelajari mengenai push notification dengan firebase dan tidak. Berikut tampilan jadinya.

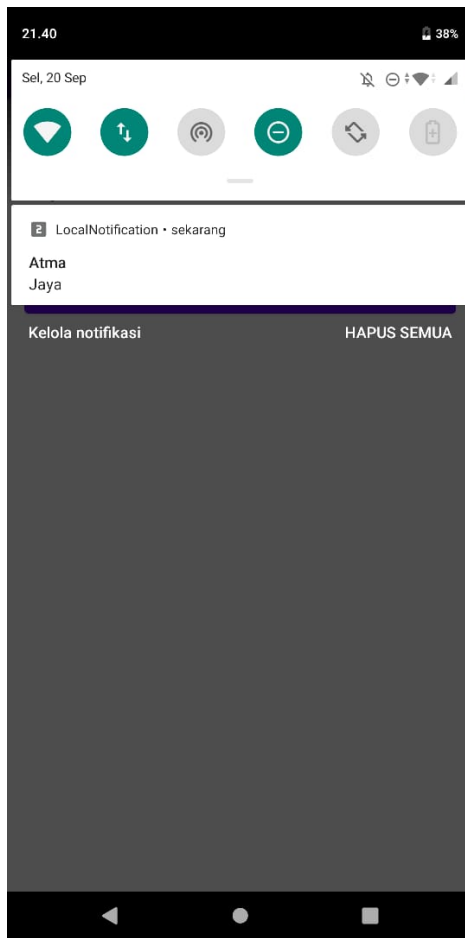


Gambar 3 Tampilan Aplikasi

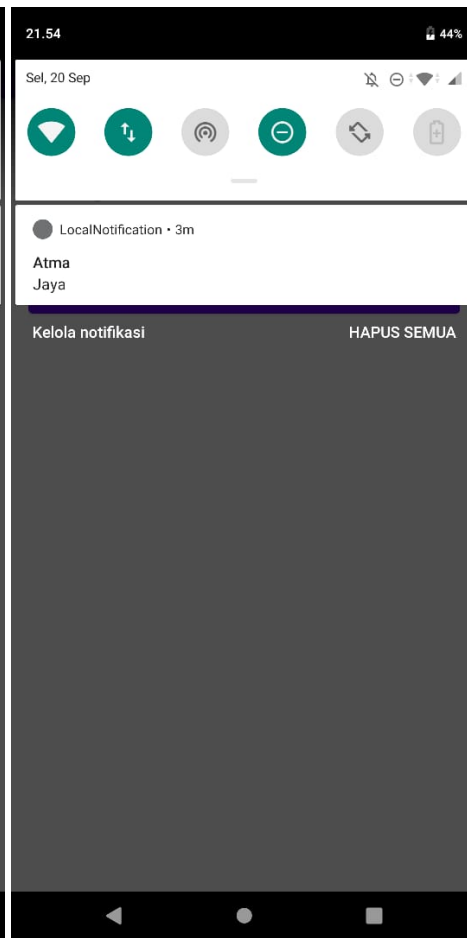


Gambar 4 Tampilan Notifikasi 1

## PUSH NOTIFICATION

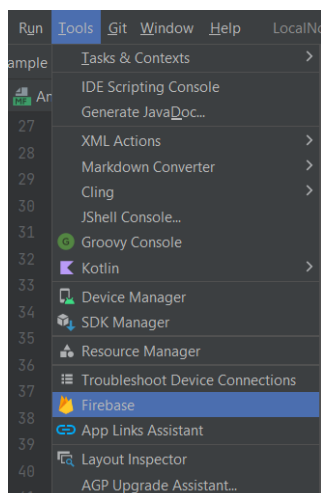


Gambar 5 Tampilan Notifikasi 2

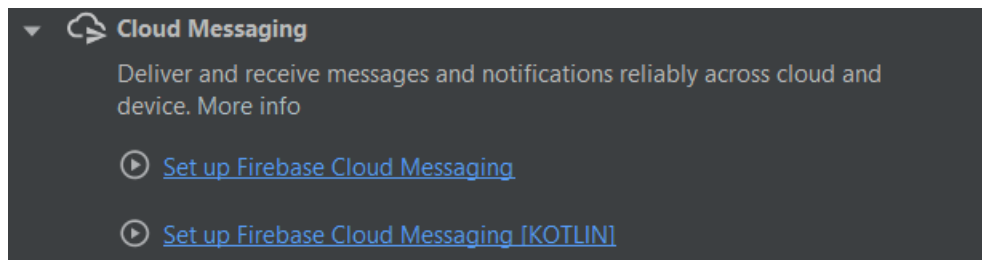


Gambar 6 Tampilan Notifikasi Firebase

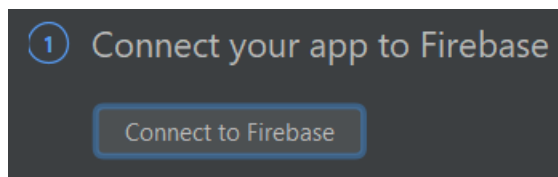
1. Buatlah proyek dengan spesifikasi proyek sebagai berikut:
  - a. Template: **Empty Activity**
  - b. Activity Name: **GD6\_Y\_XXXX**, dengan Y adalah kelas dan X adalah NPM.
  - c. Language: **Kotlin**
  - d. Minimum SDK: **API 26: Android 8.0 (Oreo)**
2. Pada bagian atas klik **Tools -> Firebase**.



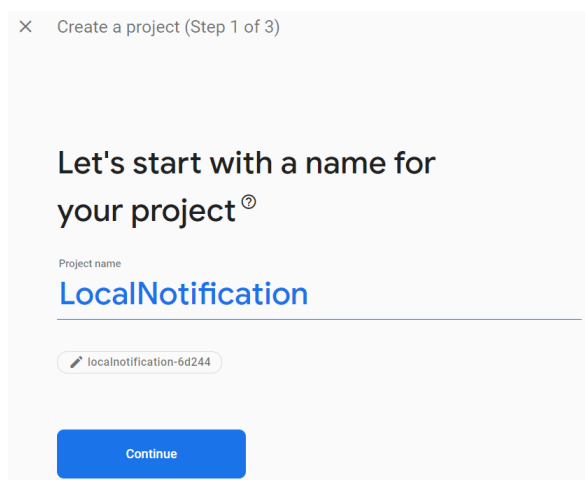
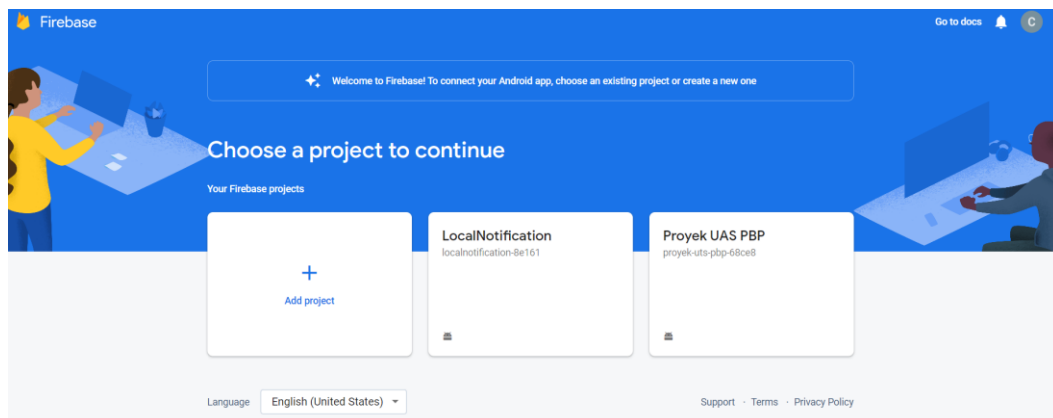
3. Pada **Cloud Messaging** klik **Set up Firebase Cloud Messaging [KOTLIN]**.



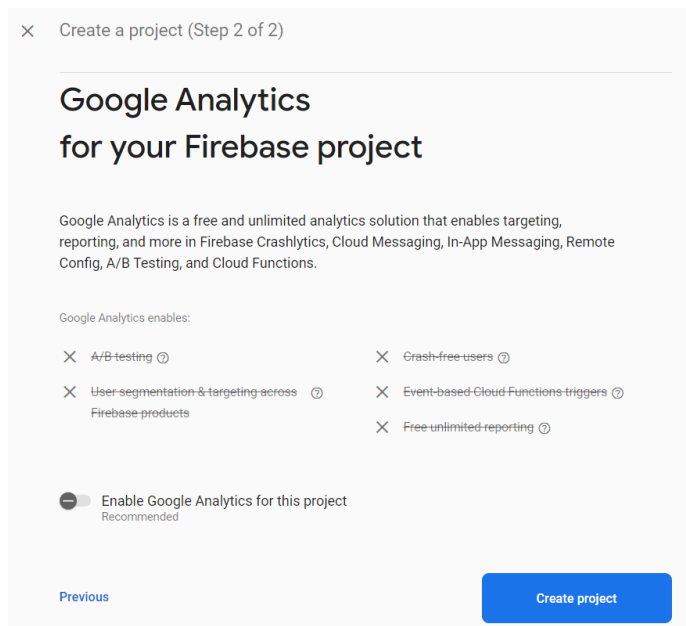
4. Pertama, kalian tekan **“Connect your app to Firebase”**. Lalu kalian akan dibawa ke Console Firebase. Pertama jika kalian dibawa ke google login, masukkan email dan password email google yang sudah terdaftar. Jika tidak kalian diwajibkan untuk daftar.



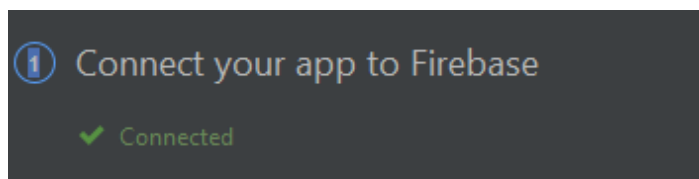
5. Setelah itu, kalian akan dibawa ke list project pada Firebase. Lalu klik Add Project dan tidak perlu diubah nama projectnya karena akan otomatis terisi. Klik **continue**.



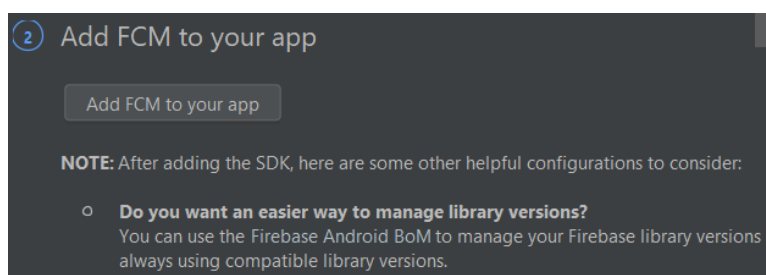
6. Lalu disable Google Analytics saja lalu klik create. Tunggu proses hingga selesai.



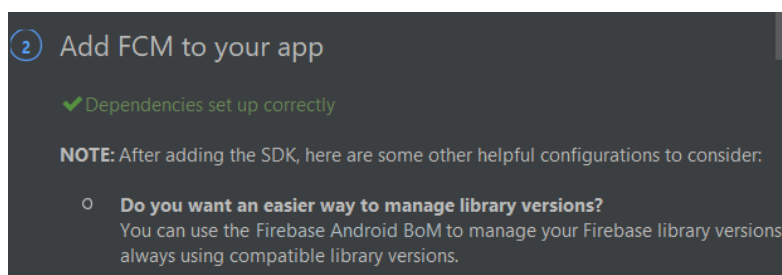
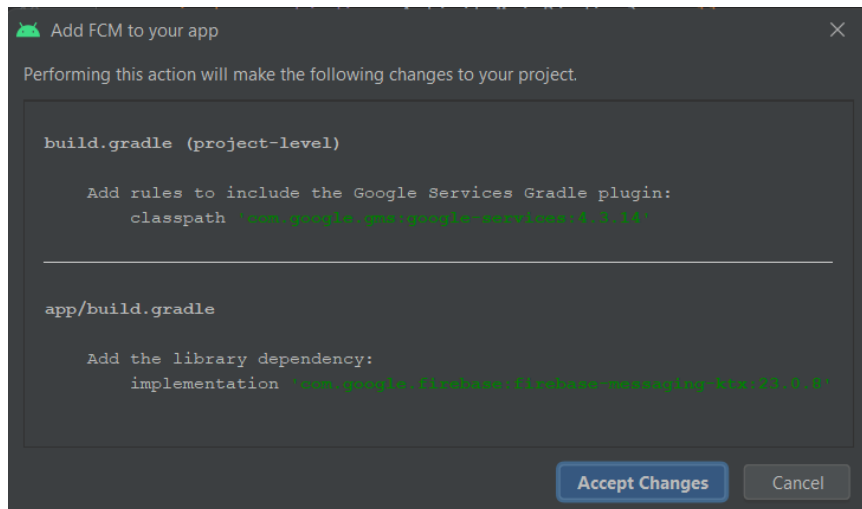
7. Setelah itu, kalian akan muncul notifikasi bahwa kalian sudah terkoneksi dengan project android kalian. Kembali ke project Android dan lihat apakah yang nomor 1 sudah menjadi connected.



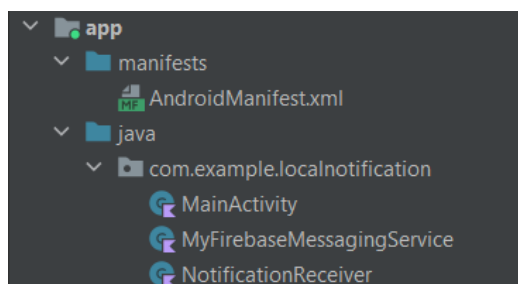
8. Lalu kalian tekan tombol add FCM to your app. Akan muncul notifikasi list yang mana yang harus diisi otomatis oleh android studio. Klik Accept Changes dan tunggu hingga nomor 2 Firebase sudah menjadi connected. Habis itu ditutup atau minimize saja.



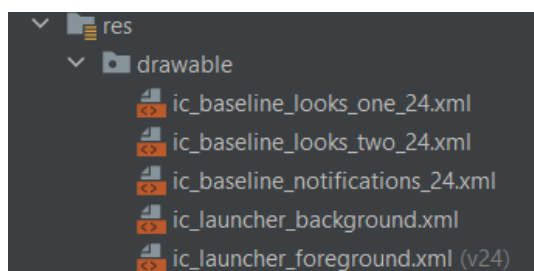
## PUSH NOTIFICATION



9. Buatlah kelas **Kotlin** baru pada project kalian bernama **MyFirebaseMessagingService** dan **NotificationReceiver**.



10. Lalu kalian pergi ke **res-> drawable** lalu klik kanan menuju **new->Vector Asset**. Pilihlah **ic\_baseline\_looks\_one\_24.xml**, **ic\_baseline\_looks\_two\_24.xml**, dan **ic\_baseline\_looks\_notifications\_24.xml** dengan cara ketika muncul kotak dialog, klik gambar pada android lalu cari di search one untuk gambar **ic\_baseline\_looks\_one**, two di search untuk **ic\_baseline\_looks\_two**, dan notifications di search untuk **ic\_baseline\_looks\_notifications**.





11. Pada **build.gradle** yang **Module**, tambahkan kode view binding sehingga bisa memakai view binding dalam aplikasinya. Taruhnya di dalam **android{}** sehabis **kotlinOptions**. Sehabis itu klik **Sync Now**.

```

30      kotlinOptions {
31          jvmTarget = '1.8'
32      }
33
34      viewBinding{
35          enabled=true
36      }
37  }

```

12. Pada **AndroidManifest.xml**, tambahkan kode berikut. Jika ketemu error, diabaikan saja dulu tunggu semua langkah selesai baru errornya hilang. Jika semua langkah sudah selesai tapi errornya belum hilang maka memang ada suatu kesalahan.

```

6      <application
7          android:allowBackup="true"
8          android:dataExtractionRules="@xml/data_extraction_rules"
9          android:fullBackupContent="@xml/backup_rules"
10         android:icon="@mipmap/ic_launcher"
11         android:label="@string/local_notification"
12         android:roundIcon="@mipmap/ic_launcher_round"
13         android:supportsRtl="true"
14         android:theme="@style/Theme.LocalNotification"
15         tools:targetApi="31">
16         <activity
17             android:name=".MainActivity"
18             android:exported="true">
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />
21
22                 <category android:name="android.intent.category.LAUNCHER" />
23             </intent-filter>
24         </activity>
25         <service
26             android:name=".MyFirebaseMessagingService"
27             android:exported="false">
28             <intent-filter>
29                 <action android:name="com.google.firebase.MESSAGING_EVENT" />
30             </intent-filter>
31         </service>
32         <receiver android:name=".NotificationReceiver"/>
33     </application>

```

13. Pada **activity\_main.xml** tambahkan kode berikut.

## PUSH NOTIFICATION

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   android:padding="16dp"
9   tools:context=".MainActivity">
10
11   <EditText
12     android:id="@+id/et_title"
13     android:layout_width="match_parent"
14     android:layout_height="wrap_content"
15     android:hint="Title"/>
16
17   <EditText
18     android:id="@+id/et_message"
19     android:layout_width="match_parent"
20     android:layout_height="wrap_content"
21     android:hint="Message"/>
22
23   <Button
24     android:id="@+id/btn1"
25     android:layout_width="match_parent"
26     android:layout_height="wrap_content"
27     android:text="Send on Channel 1"/>
28
29   <Button
30     android:id="@+id/btn2"
31     android:layout_width="match_parent"
32     android:layout_height="wrap_content"
33     android:text="Send on Channel 2"/>
34
35 </LinearLayout>
```

14. Pada kelas **NotificationReceiver.kt** tambahkan kode berikut.

```
1 package com.example.localnotification
2
3 import android.content.BroadcastReceiver
4 import android.content.Context
5 import android.content.Intent
6 import android.widget.Toast
7
8 class NotificationReceiver : BroadcastReceiver() {
9     override fun onReceive(context: Context?, intent: Intent) {
10         val message = intent.getStringExtra("toastMessage")
11         Toast.makeText(context, message, Toast.LENGTH_LONG).show()
12     }
13 }
```

15. Pada kelas **MainActivity.kt** tambahkan kode berikut.

## PUSH NOTIFICATION

```
1 package com.example.localnotification
2
3 import android.app.NotificationChannel
4 import android.app.NotificationManager
5 import android.app.PendingIntent
6 import android.content.Context
7 import android.content.Intent
8 import android.graphics.Color
9 import android.os.Build
10 import android.os.Bundle
11 import androidx.appcompat.app.AppCompatActivity
12 import androidx.core.app.NotificationCompat
13 import androidx.core.app.NotificationManagerCompat
14 import com.example.localnotification.databinding.ActivityMainBinding
15
16 class MainActivity : AppCompatActivity() {
17
18     private var binding: ActivityMainBinding? = null
19     private val CHANNEL_ID_1 = "channel_notification_01"
20     private val CHANNEL_ID_2 = "channel_notification_02"
21     private val notificationId1 = 101
22     private val notificationId2 = 102
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         binding = ActivityMainBinding.inflate(layoutInflater)
27         setContentView(binding!!.root)
28
29         createNotificationChannel()
```

```
30
31         binding!!.btn1.setOnClickListener { // it View!
32             sendNotifiacion1()
33         }
34
35         binding!!.btn2.setOnClickListener { // it View!
36             sendNotifiacion2()
37         }
38     }
39
40     private fun createNotificationChannel() {
41         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
42             val name = "Notification Title"
43             val descriptionText = "Notification Description"
44
45             val channel1 = NotificationChannel(CHANNEL_ID_1, name, NotificationManager.IMPORTANCE_DEFAULT).apply { // this: NotificationChannel
46                 description = descriptionText
47             }
48
49             val channel2 = NotificationChannel(CHANNEL_ID_2, name, NotificationManager.IMPORTANCE_DEFAULT).apply { // this: NotificationChannel
50                 description = descriptionText
51             }
52
53             val notificationManager: NotificationManager =
54                 getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
55             notificationManager.createNotificationChannel(channel1)
56             notificationManager.createNotificationChannel(channel2)
57         }
58     }
```

## PUSH NOTIFICATION

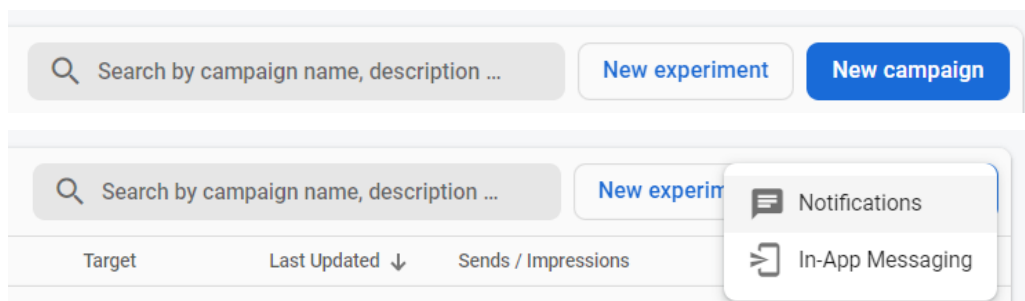
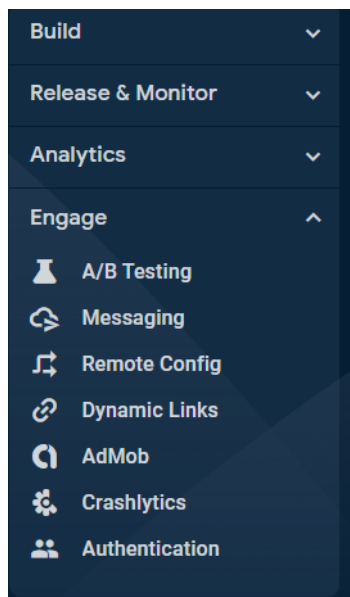
```
59
60 private fun sendNotifiaction1() {
61
62     val intent : Intent = Intent( packageContext: this, MainActivity::class.java).apply { this: Intent
63         flags = Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK
64     }
65     val pendingIntent: PendingIntent = PendingIntent.getActivity( context: this, requestCode: 0, intent, flags: 0)
66
67     val broadcastIntent : Intent = Intent( packageContext: this, NotificationReceiver::class.java)
68     broadcastIntent.putExtra( name: "toastMessage", binding?.etMessage?.text?.toString())
69     val actionIntent = PendingIntent.getBroadcast( context: this, requestCode: 0, broadcastIntent, PendingIntent.FLAG_UPDATE_CURRENT)
70
71     val builder = NotificationCompat.Builder( context: this, CHANNEL_ID_1)
72     .setSmallIcon(R.drawable.ic_baseline_looks_one_24)
73     .setContentTitle(binding?.etTitle?.text?.toString())
74     .setContentText(binding?.etMessage?.text?.toString())
75     .setCategory(NotificationCompat.CATEGORY_MESSAGE)
76     .setColor(Color.BLUE)
77     .setAutoCancel(true)
78     .setOnlyAlertOnce(true)
79     .setContentIntent(pendingIntent)
80     .addAction(R.mipmap.ic_launcher, title: "Toast", actionIntent)
81     .setPriority(NotificationCompat.PRIORITY_DEFAULT)
82
83     with(NotificationManagerCompat.from( context: this)) { this: NotificationManagerCompat
84         notify(notificationId1, builder.build())
85     }
86 }
87
88 private fun sendNotifiaction2() {
89
90     val builder = NotificationCompat.Builder( context: this, CHANNEL_ID_1)
91     .setSmallIcon(R.drawable.ic_baseline_looks_two_24)
92     .setContentTitle(binding?.etTitle?.text?.toString())
93     .setContentText(binding?.etMessage?.text?.toString())
94     .setPriority(NotificationCompat.PRIORITY_LOW)
95
96     with(NotificationManagerCompat.from( context: this)) { this: NotificationManagerCompat
97         notify(notificationId2, builder.build())
98     }
99 }
100
101 }
```

16. Pada **MyFirebaseMessagingService.kt** tambahkan kode berikut.

## PUSH NOTIFICATION

```
1 package com.example.localnotification
2
3 import android.app.NotificationChannel
4 import android.app.NotificationManager
5 import android.app.PendingIntent
6 import android.content.Intent
7 import android.media.RingtoneManager
8 import android.os.Build
9 import androidx.core.app.NotificationCompat
10 import androidx.core.app.NotificationManagerCompat
11 import com.google.firebase.messaging.FirebaseMessagingService
12 import com.google.firebase.messaging.RemoteMessage
13
14 class MyFirebaseMessagingService : FirebaseMessagingService() {
15     override fun onMessageReceived(remoteMessage: RemoteMessage) {
16         if (remoteMessage.notification != null) {
17             sendNotification(remoteMessage.notification!!.title.toString(), remoteMessage.notification!!.body.toString())
18         }
19     }
20
21     private fun sendNotification(messageTitle: String, messageBody: String) {
22         val intent = Intent(packageContext: this, MyFirebaseMessagingService::class.java).apply { this: Intent
23             flags = Intent.FLAG_ACTIVITY_CLEAR_TOP
24         }
25         val pendingIntent = PendingIntent.getActivity(context: this, requestCode: 0, intent, PendingIntent.FLAG_ONE_SHOT)
26
27         val channelId = "CHANNEL"
28         val defaultSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION)
29
30         val notificationBuilder = NotificationCompat.Builder(context: this, channelId)
31             .setSmallIcon(R.drawable.ic_baseline_notifications_24)
32             .setContentTitle(messageTitle)
33             .setContentText(messageBody)
34             .setAutoCancel(true)
35             .setSound(defaultSoundUri)
36             .setContentIntent(pendingIntent)
37
38         // Since android Oreo notification channel is needed.
39         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
40             val channel = NotificationChannel(channelId, name: "Notification Title", NotificationManager.IMPORTANCE_HIGH)
41             val notificationManager = getSystemService(NOTIFICATION_SERVICE) as NotificationManager
42             notificationManager.createNotificationChannel(channel)
43             with(NotificationManagerCompat.from(context: this)) { this: NotificationManagerCompat
44                 notify(id: 0, notificationBuilder.build())
45             }
46         }
47     }
48 }
```

17. Proyek telah selesai. Coba di run-kan aplikasinya dan dicoba apakah sudah berhasil atau belum.
18. Untuk mengetes notification dari firebase bisa ikutin langkah berikut.
19. Kalian klik site <https://console.firebase.google.com/u/0/>, lalu klik project firebase yang sudah kalian buat lalu klik. Anda akan menuju ke Dashboard Firebase. Kalian cari Messaging lalu klik. Lalu klik New Campaign dan pilih Notifications.



20. Disini kalian akan mencoba kirim sebuah pesan ke aplikasi kalian. Kirimnya bisa saat aplikasinya terbuka atau tertutup. Pada bagian 1 Notification ketik isi sesuai dengan contoh. Lalu klik Next.

**1 Notification**

Notification title ⓘ

Atma

Notification text

Jaya

Notification image (optional) ⓘ

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

Notification name (optional) ⓘ

Enter optional name

21. Lalu pada Target, pilih target user sesuai nama project kalian. Lalu klik Next. Contohnya seperti ini.

## PUSH NOTIFICATION

**2 Target**

**User segment** Topic

Target user if...

App com.example.localnotification and

Target another app

**Next**

22. Untuk Scheduling pastikan bentuk nilainya Now lalu klik next.

**3 Scheduling**

Send to eligible users

Now

**Next**

23. Pada bagian Additional options (optional) biarkan default dan klik Review.

24. Pada Review Message, klik Publish. Kalian akan dibawa ke dashboard Cloud messaging apakah message kalian sudah terkirim atau belum.

**Review message**

**Unable to reserve a user property for Notification conversion funnel analysis**

**Notification content**

Jaya

**Target**

User segment matching one targeting criterion

**Scheduling**

Send now

Cancel **Publish**

25. Terakhir, check status di notification hp kalian dan kalian juga bisa lihat hasil akhir dari Aplikasi kalian.

26. Jika masih kebingungan, silahkan tanya kakak-kakak asdos atau dosen masing-masing.

27. Semoga beruntung 😊.