

LAPORAN TOPIK KHUSUS

(Message Queue dengan RabbitMQ)



O l e h:

Muhammad Amir Shafwan

2211083042

PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI PADANG

2025

A. DASAR TEORI

1. Konsep Sistem Notifikasi

Sistem notifikasi adalah mekanisme yang memungkinkan aplikasi untuk mengirim pesan atau pemberitahuan kepada pengguna atau sistem lain secara real-time atau terjadwal. Dalam konteks ini, notifikasi dapat dikirim melalui berbagai media seperti email, SMS, atau push notification.

Sistem ini umumnya terdiri dari dua bagian utama:

- a. Notification Publisher – Bertanggung jawab untuk mengirimkan pesan ke sistem perantara (broker).
- b. Notification Consumer – Bertugas untuk menerima dan memproses pesan dari broker sesuai dengan kebutuhan.

2. Arsitektur Message Queue

Message Queue adalah pola komunikasi asinkron yang memungkinkan komponen aplikasi untuk bertukar pesan tanpa perlu saling mengetahui satu sama lain. Beberapa keuntungan utama dari Message Queue adalah:

- a. Decoupling – Publisher dan consumer tidak perlu berinteraksi langsung.
- b. Scalability – Proses bisa didistribusikan ke beberapa consumer tanpa mempengaruhi publisher.
- c. Reliability – Pesan yang masuk ke dalam queue tetap tersimpan hingga berhasil diproses oleh consumer.

3. RabbitMQ sebagai Message Broker

RabbitMQ adalah message broker berbasis protokol AMQP (Advanced Message Queuing Protocol) yang digunakan untuk mengelola antrian pesan. Dalam sistem notifikasi ini, RabbitMQ bertindak sebagai perantara antara publisher dan consumer.

Komponen Utama RabbitMQ:

- a. Exchange: Tempat masuknya pesan dari publisher sebelum dikirim ke queue.

- b. Queue: Menyimpan pesan hingga ada consumer yang mengambilnya.
- c. Binding: Menghubungkan exchange dengan queue menggunakan routing key.

RabbitMQ memungkinkan implementasi berbagai pola komunikasi, seperti fanout, direct, dan topic exchange untuk mendistribusikan pesan.

B. PERCOBAAN DAN HASIL

Sebelum melakukan percobaan ini ada beberapa hal yang perlu diperhatikan:

- Install RabbitMQ
- Install Golang

1. Clone project dari GitHub

https://github.com/kusnadi8605/notification_publisher

https://github.com/kusnadi8605/notification_consumer

```
amir@LAPTOP-R7VISQ0T:/mnt/c/Users/Lenovo/Topik_Khusus/Pertemuan4_rabbitmq$ git clone https://github.com/kusnadi8605/notification_consumer
Cloning into 'notification_consumer'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 32 (delta 6), reused 32 (delta 6), pack-reused 0 (from 0)
Receiving objects: 100% (32/32), 4.22 KiB | 159.00 KiB/s, done.
Resolving deltas: 100% (6/6), done.
amir@LAPTOP-R7VISQ0T:/mnt/c/Users/Lenovo/Topik_Khusus/Pertemuan4_rabbitmq$ git clone https://github.com/kusnadi8605/notification_publisher
Cloning into 'notification_publisher'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 30 (delta 4), reused 28 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (30/30), 5.61 KiB | 302.00 KiB/s, done.
Resolving deltas: 100% (4/4), done.
```

2. Masuk ke dalam direktori dan install dependensi Go

- go mod init notification_consumer
- go mod tidy
- go mod vendor

3. Jalankan kedua aplikasi dan RabbitMQ

a. Perintah untuk menjalankan RabbitMQ

- sudo systemctl enable rabbitmq-server
- sudo systemctl start rabbitmq-server

b. Perintah untuk menjalankan server cunsomer:

- go run cmd/consumer/email/main.go
- go run cmd/consumer/sms/main.go
- go run cmd/consumer/fcm/main.go

```
amir@LAPTOP-R7VISQ0T:/mnt/c/Users/Lenovo/Topik_Khusus/Pertemuan4_rabbitmq/notification_consumer$ go run cmd/email/main.go
2025/03/15 13:31:46 Using Exchange: notifications
2025/03/15 13:31:46 [EMAIL] Listening for messages...
```

```

amir@LAPTOP-R7VISQ0T:/mnt/c/Users/Lenovo/Topik_Khusus/Pertemuan4_rabbitmq/notification_consumer$ go run cmd/sms/main.go
2025/03/15 13:31:25 Using Exchange: notifications
2025/03/15 13:31:25 [SMS] Listening for messages...

```

```

amir@LAPTOP-R7VISQ0T:/mnt/c/Users/Lenovo/Topik_Khusus/Pertemuan4_rabbitmq/notification_consumer$ go run cmd/fcm/main.go
2025/03/15 13:31:00 Using Exchange: notifications
2025/03/15 13:31:00 [FCM] Listening for messages...

```

c. Perintah untuk menjalankan server publisher:

- go run main.go

```

amir@LAPTOP-R7VISQ0T:/mnt/c/Users/Lenovo/Topik_Khusus/Pertemuan4_rabbitmq/notification_publisher$ go run main.go
2025/03/15 13:17:32 Using Exchange: notifications

      _ _ _ _ _
     / _ _ _ _ \
    / _ _ _ _ \ v4.13.3
   / _ _ _ _ \
  / _ _ _ _ \
 / _ _ _ _ \
/_ _ _ _ _ \
High performance, minimalist Go web framework
https://echo.labstack.com
-----o/-----
      o\
=> http server started on [::]:8080

```

4. Menguji Endpoint publish dengan cURL

```
curl -X POST http://localhost:8080/publish \
```

```
-H "Content-Type: application/json" \
```

```
-d '{
```

```
  "order_id": "12345",
```

```
  "user_id": "67890",
```

```
  "content": "New order received",
```

```
  "timestamp": "2025-03-11T10:00:00Z"
```

```
}'
```

```

amir@LAPTOP-R7VISQ0T:/mnt/c/Users/Lenovo$ curl -X POST http://localhost:8080/publish -H "Content-Type: application/json" -d '{
  "order_id": "12345",
  "user_id": "67890",
  "content": "New order received",
  "timestamp": "2025-03-11T10:00:00Z"
}'
{"code":200,"message":"Message published successfully"}

```

Jika pengujian berhasil maka server email, sms, dan fcm akan menerima notifikasi order

```
2025/03/15 13:32:00 [EMAIL] Received message: {OrderID:12345 UserID:67890 Content:New order received Timestamp:2025-03-11T10:00:00Z}  
sending email ... New order received
```

```
2025/03/15 13:32:00 [SMS] Received message: {OrderID:12345 UserID:67890 Content:New order received Timestamp:2025-03-11T10:00:00Z}  
sending sms ... New order received
```

```
2025/03/15 13:32:00 [FCM] Received message: {OrderID:12345 UserID:67890 Content:New order received Timestamp:2025-03-11T10:00:00Z}  
sending fcm ... New order received
```

C. REFERENSI

https://github.com/kusnadi8605/notification_publisher

https://github.com/kusnadi8605/notification_consumer