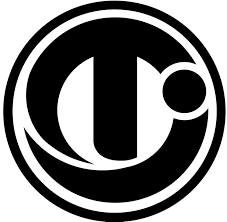
**Laporan Praktek Topik Khusus 3**

Message Queue



**SEMESTER VI**

**DISUSUN OLEH :**

KURNIAWAN ALEXANDER

2211083030

**DOSEN PENGAMPU :**

YULHERNIWATI, S.Kom.,MT

YUNUS SUPRIADI WIJAYA

**PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI PADANG**

**2025**

**A. Landasan Teori**

1. **Golang (Go Language)**  
   Golang, atau Go, adalah bahasa pemrograman open-source yang dikembangkan oleh Google dengan tujuan utama menyediakan efisiensi, kesederhanaan, dan keandalan dalam pengembangan perangkat lunak. Bahasa ini memiliki dukungan bawaan untuk pemrograman konkuren melalui fitur goroutine dan channel, menjadikannya sangat ideal untuk membangun aplikasi berbasis jaringan, sistem terdistribusi, dan layanan web. Golang juga memiliki garbage collector otomatis yang membantu pengelolaan memori tanpa perlu intervensi manual. Karena bersifat statis dan dikompilasi, kesalahan dapat terdeteksi lebih awal pada saat kompilasi. Dengan performa yang tinggi dan framework populer seperti Gin dan Echo, Golang sering dipilih dalam pengembangan REST API maupun aplikasi web modern.
2. **API (Application Programming Interface)**  
   API merupakan antarmuka yang memungkinkan satu aplikasi berinteraksi dengan aplikasi lain. Dalam konteks pengembangan aplikasi web, API memainkan peran penting dalam pertukaran data antar sistem. Salah satu bentuk API yang umum digunakan adalah RESTful API, yang memanfaatkan protokol HTTP untuk memfasilitasi operasi dasar terhadap data, seperti **Create (membuat data baru), Read (mengambil data), Update (memperbarui data),** dan **Delete (menghapus data)** — yang dikenal dengan istilah CRUD. Pendekatan ini membuat pengelolaan dan integrasi data menjadi lebih mudah dan terstruktur.
3. **Erlang**  
   Erlang adalah bahasa pemrograman berparadigma fungsional yang dikembangkan untuk menciptakan sistem yang mampu menangani banyak proses secara bersamaan (high concurrency), tahan terhadap kesalahan (fault-tolerant), dan mudah diskalakan (scalable). Dengan pendekatan proses ringan yang saling berkomunikasi melalui pengiriman pesan, Erlang dapat menjalankan banyak tugas secara paralel secara efisien. Filosofi “let it crash” atau “biarkan gagal” memungkinkan sistem untuk menangani kegagalan tanpa menghentikan keseluruhan layanan. Selain itu, Erlang dilengkapi dengan OTP (Open Telecom Platform), sebuah kerangka kerja yang menyediakan kumpulan pustaka dan prinsip arsitektural untuk membangun sistem yang andal dan tahan banting.
4. **RabbitMQ**  
   RabbitMQ adalah perangkat lunak open-source yang berfungsi sebagai message broker dan mendukung protokol AMQP (Advanced Message Queuing Protocol). Peran utamanya adalah menerima, menyimpan sementara, dan mengirimkan pesan antar aplikasi. RabbitMQ memungkinkan komunikasi antar sistem berjalan dengan efisien, terutama dalam lingkungan yang tersebar secara geografis atau sistem mikroservis. Dengan mendukung berbagai pola komunikasi seperti point-to-point (satu pengirim ke satu penerima) dan publish/subscribe (satu pengirim ke banyak penerima), RabbitMQ menjadi solusi fleksibel untuk kebutuhan distribusi pesan dalam skala besar.

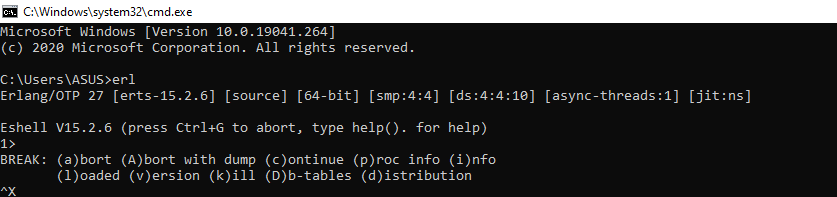
**B.** Tools

* Erlang
* Golang
* Vs Code
* RabbitMQ

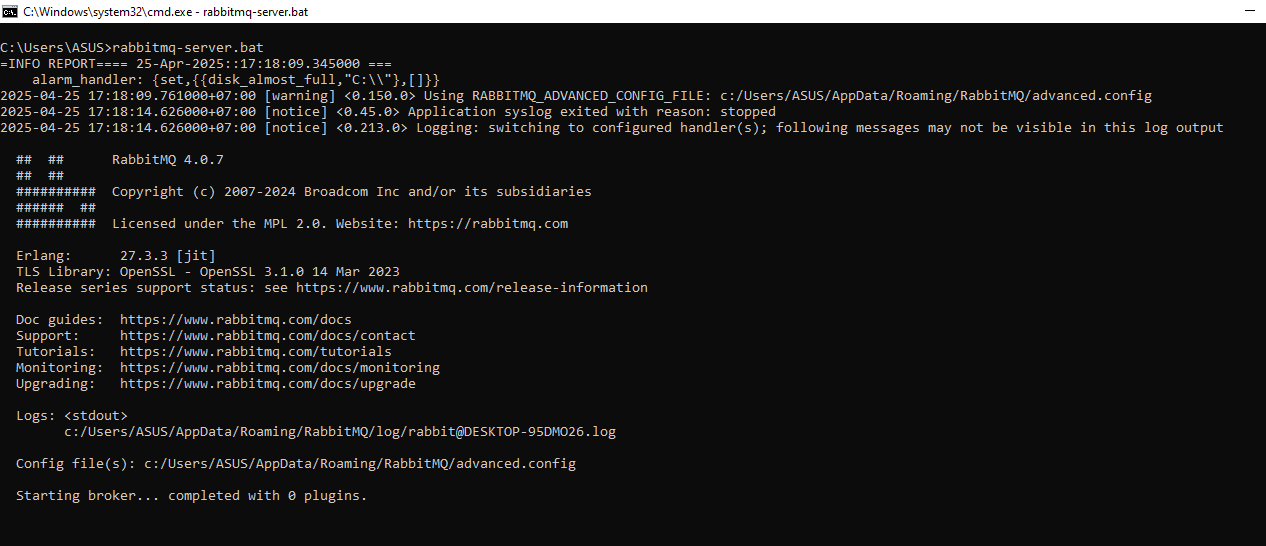
C. Langkah Kerja

1. Install Erlang dan RabbitMQ

* Download dan Install Erlang (RabbitMQ perlu erlang)

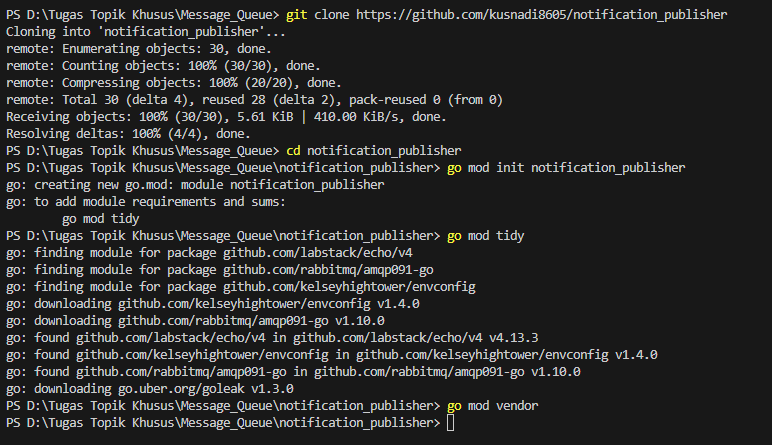


* Download dan Install RabbitMQ
* Menambahkan RabbitMQ ke Path
* Menjalankan RabbitMQ



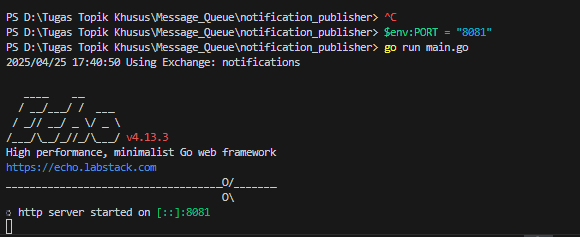
1. Clone Repository Notification Publisher

* Lakukan clone repository yang akan dijadikan bahan praktek
* Membuat file go.mod di direktori Anda, yang akan melacak dependensi proyek Anda.
* Menambahkan dependensi yang diperlukan yang ditemukan dalam kode Anda ke file go.mod.
* Menyalin semua dependensi proyek ke direktori vendor di dalam direktori proyek Anda.



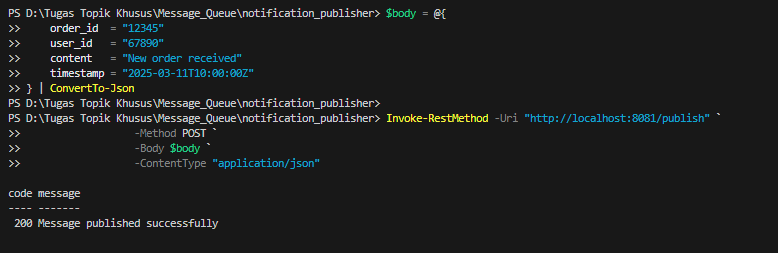
1. Jalankan Aplikasi

* Run file main.go untuk menjalankan aplikasi



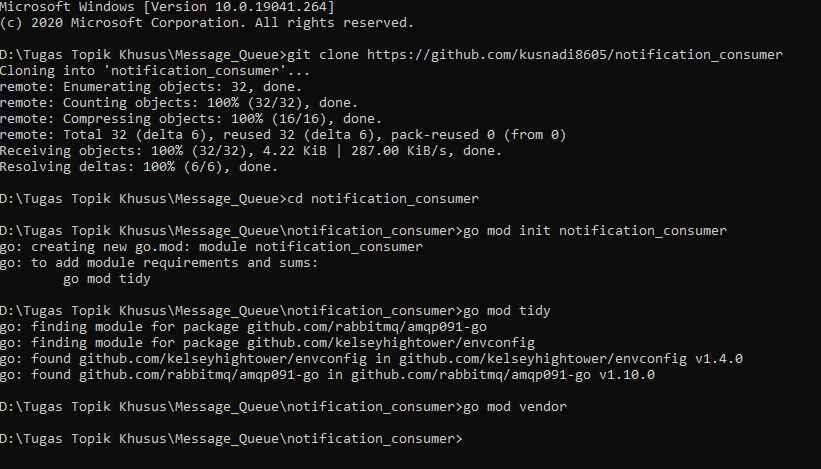
1. Menguji Aplikasi

* Mengirim pesan ke server dan hasil outputnya



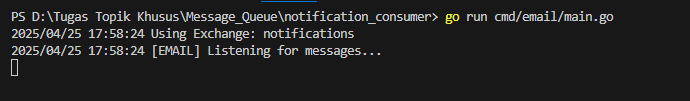
1. Clone Aplikasi Consumer

* Lakukan clone repository yang akan dijadikan bahan praktek
* Membuat file go.mod di direktori Anda, yang akan melacak dependensi proyek Anda.
* Menambahkan dependensi yang diperlukan yang ditemukan dalam kode Anda ke file go.mod.
* Menyalin semua dependensi proyek ke direktori vendor di dalam direktori proyek Anda.

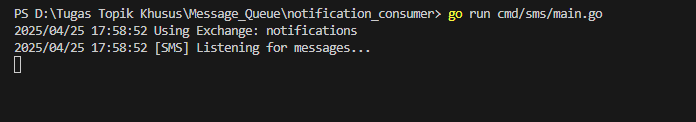


1. Menjalankan Aplikasi Consumer

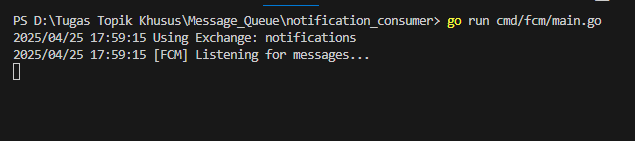
* Run file main.go pada folder email



* Run file main.go pada folder sms

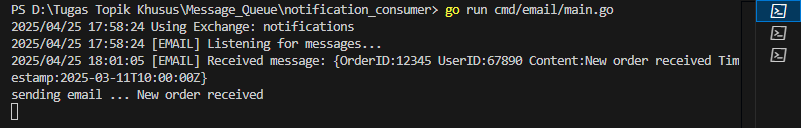


* Run file main.go pada folder fcm

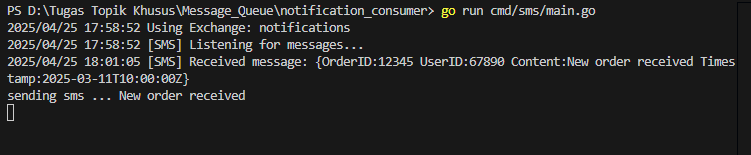


1. Outputnya

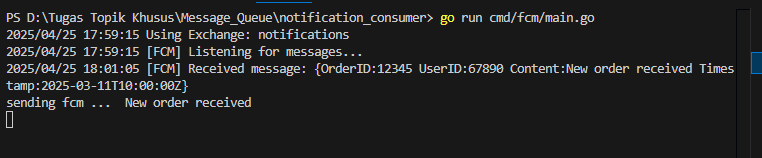
* Output Email



* Output SMS



* Output FCM



D. Kesimpulan

**Penggunaan RabbitMQ dengan Go sebagai Message Queue**

Mengintegrasikan RabbitMQ sebagai sistem antrian pesan (message queue) dalam pengembangan aplikasi menggunakan Go memberikan solusi yang andal dan mudah diskalakan, terutama untuk kebutuhan komunikasi asinkron. RabbitMQ memungkinkan pemisahan logika aplikasi menjadi dua peran utama, yaitu produsen (producer) dan konsumen (consumer), sehingga proses pertukaran data dapat dilakukan secara efisien dan tidak saling bergantung secara langsung.

Bahasa Go, melalui pustaka pendukung RabbitMQ, memudahkan penerapan berbagai pola komunikasi pesan serta menyediakan mekanisme penanganan kesalahan yang tangguh. Hal ini memungkinkan aplikasi untuk terus berjalan meskipun terjadi gangguan atau kegagalan pada salah satu bagian sistem. Kombinasi kekuatan RabbitMQ dalam pengelolaan pesan dan performa tinggi yang dimiliki Go menjadikan keduanya sangat ideal untuk membangun aplikasi dengan kebutuhan throughput tinggi, keandalan sistem yang kuat, serta arsitektur yang fleksibel dan mudah diperluas.