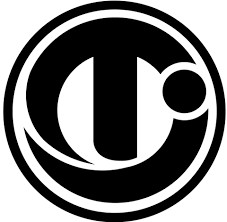
**Laporan Praktek Topik Khusus IV**

MongoDB



**SEMESTER VI**

**DISUSUN OLEH :**

MUHAMMAD ABEL AL-FAHREZI (2211083034)

**PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI PADANG**

**2025**

1. **Landasan Teori**
2. **Golang (Go Language)**

Golang, yang dikembangkan oleh Google, merupakan bahasa pemrograman sumber terbuka yang dirancang untuk efisiensi, kesederhanaan, dan keandalan. Dengan dukungan bawaan untuk konkurensi melalui goroutine dan channel, Golang sangat cocok untuk pengembangan aplikasi web, sistem terdistribusi, dan aplikasi jaringan. Fitur garbage collection otomatis mengurangi beban pengelolaan memori, sementara sifat statis diketiknya membantu mendeteksi kesalahan pada waktu kompilasi. Kinerja tinggi Golang menjadikannya pilihan populer untuk membangun API dan layanan web, dengan framework seperti Gin dan Echo yang mempermudah pengembangan aplikasi web.

1. **API**

API (Application Programming Interface) merupakan fondasi utama dalam pengembangan aplikasi web modern. API memungkinkan aplikasi yang berbeda untuk berkomunikasi dan bertukar data. Dalam konteks ini, RESTful API menjadi pilihan populer karena arsitekturnya yang berbasis pada prinsip-prinsip HTTP, memungkinkan operasi CRUD (Create, Read, Update, Delete) dilakukan dengan mudah dan efisien. Operasi CRUD sendiri merupakan inti dari manipulasi data, di mana 'Create' digunakan untuk membuat data baru, 'Read' untuk membaca data, 'Update' untuk memperbarui data, dan 'Delete' untuk menghapus data.

1. **MongoDB**

MongoDB adalah sistem manajemen basis data NoSQL (Not Only SQL) yang berorientasi pada dokumen, yang berarti data disimpan dalam format dokumen mirip JSON yang fleksibel, bukan dalam tabel baris dan kolom seperti pada database relasional tradisional. Ini memberikan fleksibilitas dan skalabilitas yang lebih besar untuk menangani data yang tidak terstruktur atau semi-terstruktur, serta cocok untuk aplikasi web modern yang membutuhkan kecepatan dan kemampuan untuk menangani volume data yang besar.

1. **RabbitMQ**

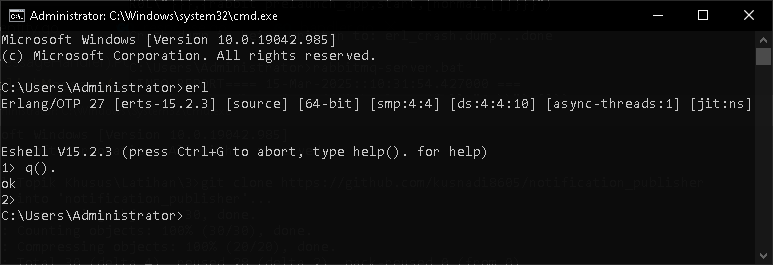
RabbitMQ adalah sebuah message broker yang bersifat open-source yang mengimplementasikan protokol Advanced Message Queuing Protocol (AMQP). Fungsinya adalah untuk menerima dan meneruskan pesan. RabbitMQ memungkinkan aplikasi-aplikasi untuk berkomunikasi dan berbagi data secara efisien, bahkan dalam lingkungan yang terdistribusi. RabbitMQ mendukung berbagai pola pertukaran pesan, seperti point-to-point dan publish/subscribe, sehingga sangat fleksibel untuk berbagai kasus penggunaan.

1. Tools

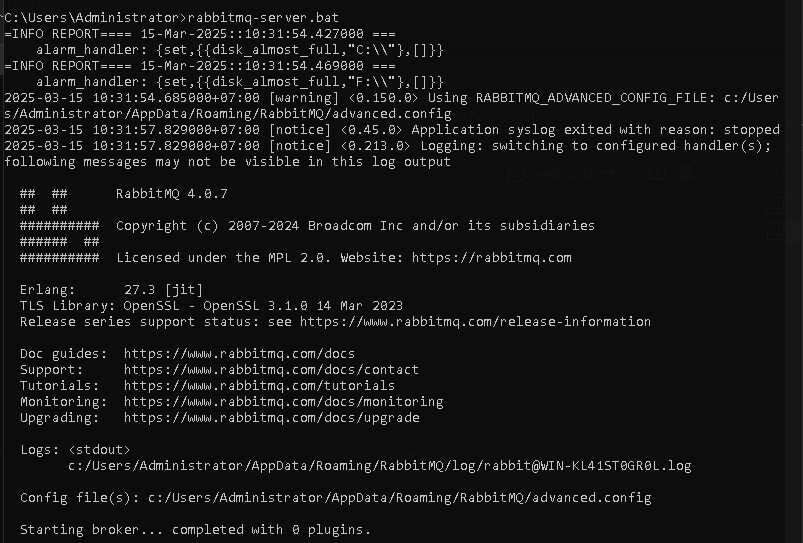
* Erlang
* Golang
* Visual Studio Code
* RabbitMQ

1. **Langkah Kerja**
2. **Install Erlang dan RabbitMQ**

* Download dan Install Erlang (RabbitMQ memerlukan erlang)
* Pastikan erlang terinstall

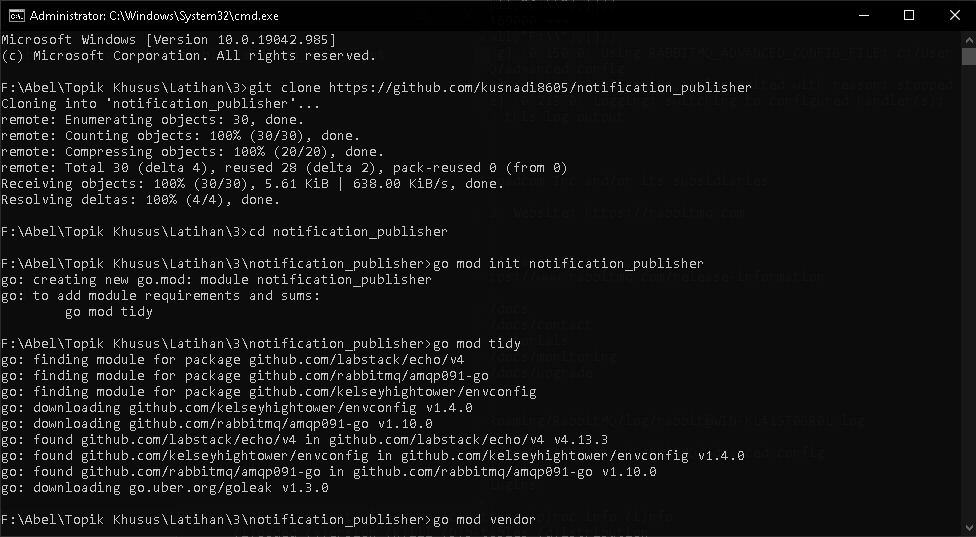


* Download dan Install RabbitMQ
* Menambahkan RabbitMQ ke PATH
* Menjalankan RabbitMQ



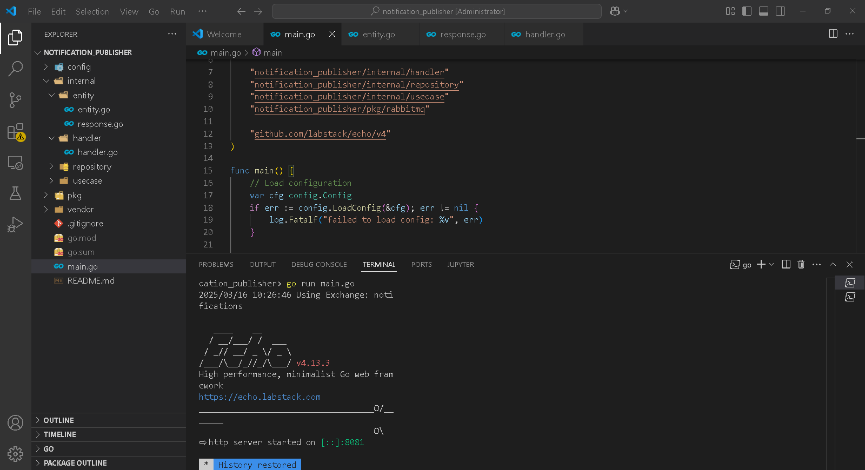
1. **Clone Repository Notification Publisher**

* 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.



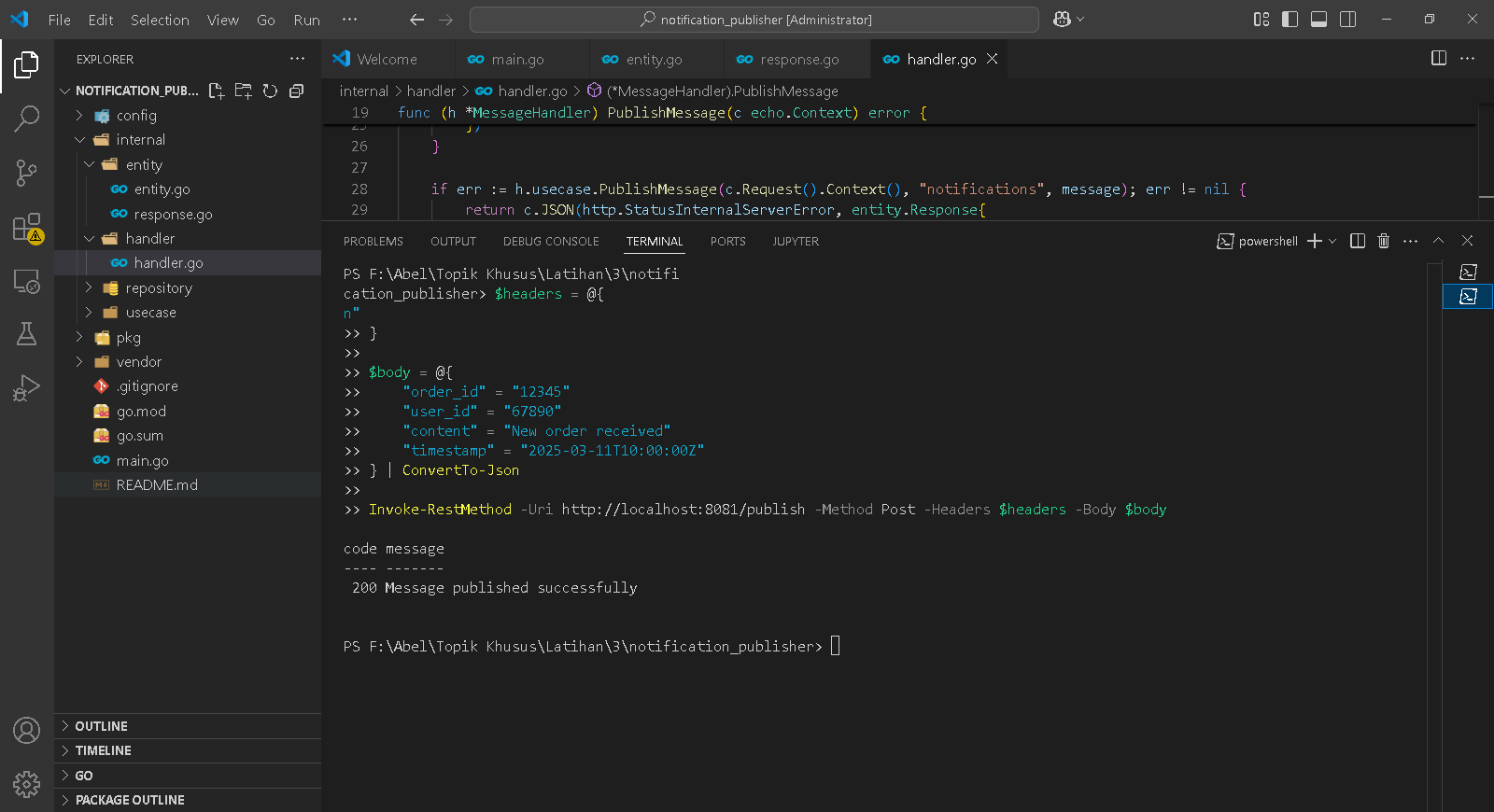
1. **Jalankan Aplikasi**

* Run file main.go untuk menjalankan aplikasi.



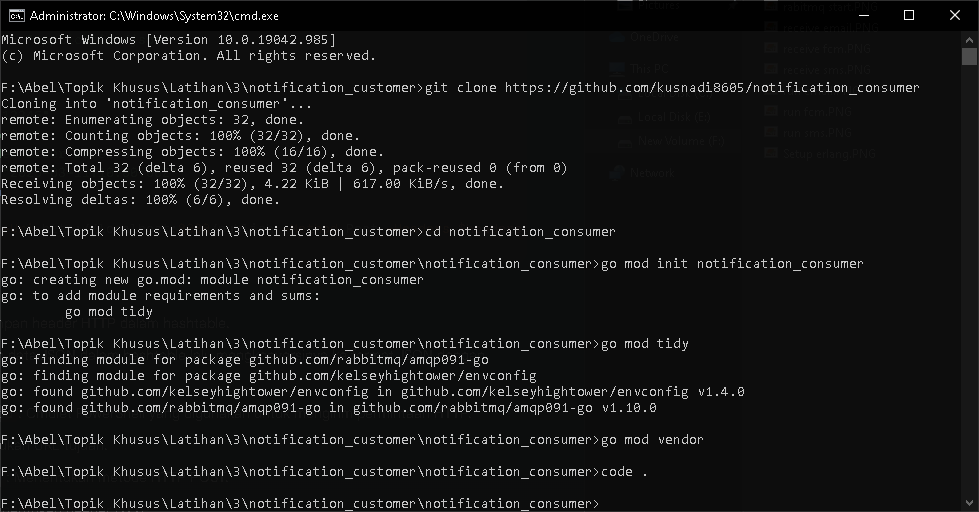
1. **Menguji Aplikasi**

* Mengirim pesan ke server
* Dan berikut adalah output nya.

****

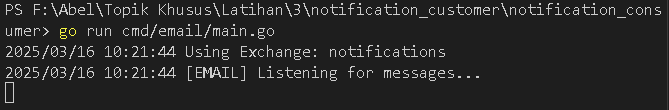
1. **Clone Aplikasi Consumer**

* 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.

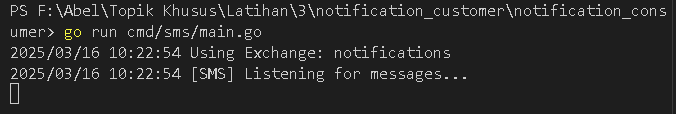


1. **Menjalankan Aplikasi Consumer**

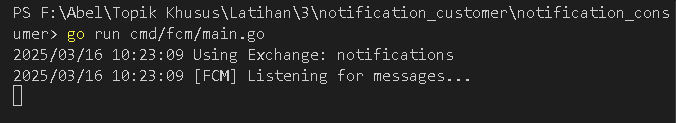
* Run file main.go pada folder email

****

* Run file main.go pada folder SMS

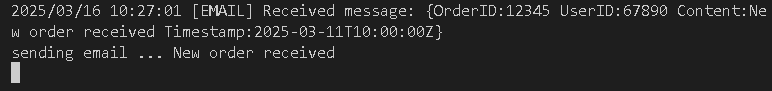
****

* Run file main.go pada folder fcm

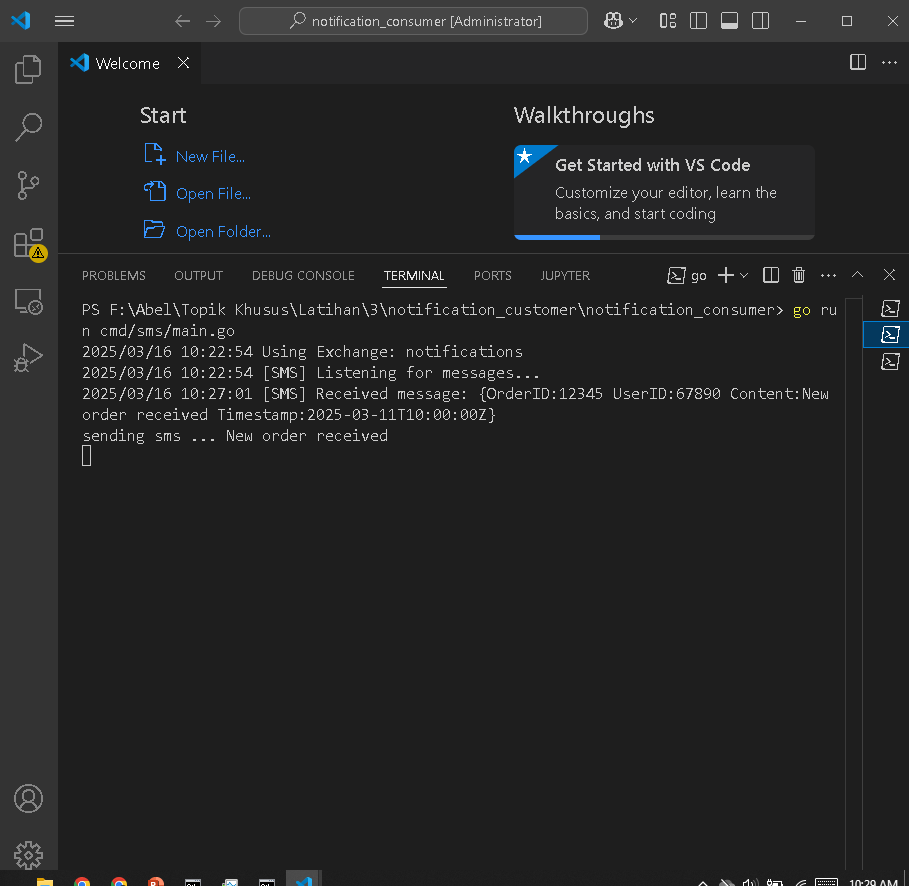
****

1. **Respon atau outputnya**

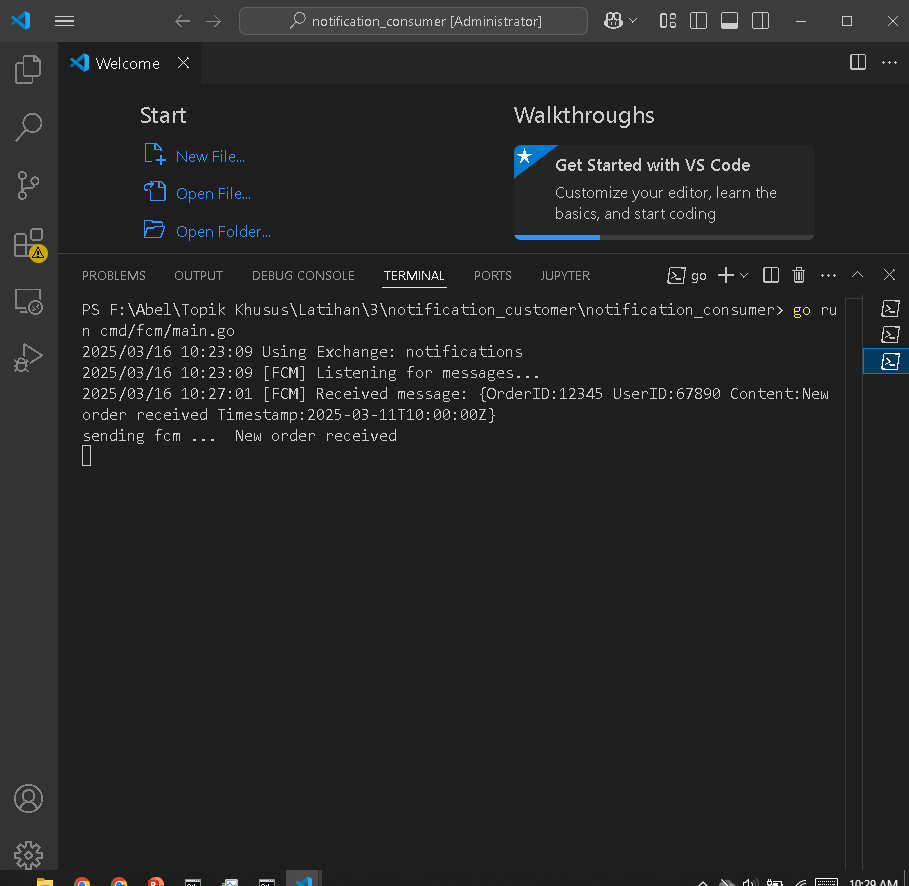
* Respon yang ada pada email consumer

****

* Respon yang ada pada SMS consumer

****

* Respon yang ada pada FCM consumer

****

1. **Kesimpulan**

Penggunaan RabbitMQ sebagai *message queue* dengan Go menghadirkan solusi yang kuat untuk membangun aplikasi yang membutuhkan komunikasi asinkron yang andal dan skalabel. Dengan RabbitMQ, kita dapat memisahkan komponen aplikasi menjadi produsen dan konsumen, memungkinkan komunikasi yang efisien dan fleksibel. Go, dengan pustaka RabbitMQ-nya, memungkinkan implementasi berbagai pola pesan dan penanganan kesalahan yang efektif, memastikan aplikasi dapat menangani kegagalan dan masalah lainnya dengan baik. Keandalan dan skalabilitas RabbitMQ, ditambah dengan efisiensi Go, menjadikan kombinasi ini sangat cocok untuk aplikasi yang memerlukan kinerja tinggi dan ketersediaan yang tinggi..