

# Implementasi *Framework* Gin dan gRPC pada Pengembangan *Back-end Web*

Markvinson Avatara<sup>#1</sup>, Robby Tan<sup>\*2</sup>

<sup>#</sup>*Program Studi Sarjana Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Maranatha  
Jln. Prof. Drg. Surya Sumantri No. 65, Sukawarna, Bandung, Indonesia  
12073011@maranatha.ac.id*

<sup>\*</sup>*Program Studi Sarjana Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Maranatha  
Jln. Prof. Drg. Surya Sumantri No. 65, Sukawarna, Bandung, Indonesia  
2robbly.tan@it.maranatha.edu*

**Abstract** — The objective of this journal is to gain a deeper understanding of the professional landscape, particularly in the realm of backend development for websites. This endeavor is undertaken at PT Gits Indonesia, an IT-focused company. Throughout the MSIB, two projects have been assigned by the company for completion. Both projects revolve around backend development for a website and utilize the Golang programming language. These projects entail the incorporation of frameworks as per the company's specifications. The initial project involves a library and makes use of the Gin framework, while the second project is centered around e-commerce and uses the gRPC framework. Despite the prior utilization of both frameworks in past projects, their implementations in these assignments can be continuously refined to optimize their efficacy. Upon the conclusion of this MSIB, the aspiration is to broaden insights into the professional sphere and potentially contribute positively to the company and others considering participation in MBKM internships.

**Keywords**— Backend, Framework, Golang

## I. PENDAHULUAN

Seiring perkembangan zaman, seluruh rangka aspek kehidupan ikut berkembang terutama pada teknologi dan kehidupan dunia kerja. Dalam menanggapi perkembangan ini pemerintah mengadakan program Kampus Merdeka. Kegiatan yang diadakan Kampus Merdeka ada berbagai macam mulai dari studi independen sampai dengan kegiatan kerja praktik atau magang. PT Gits Indonesia menjadi salah satu mitra program Kampus Merdeka.

PT Gits Indonesia merupakan perusahaan yang bergerak di bidang teknologi informasi yang bermitra dengan Kampus Merdeka di kegiatan magang. Proses kegiatan magang di PT Gits Indonesia melibatkan pembelajaran dan proyek nyata seperti di dunia kerja. PT Gits Indonesia akan mengarahkan bagaimana cara mengimplementasikan satu atau lebih teknologi yang baru ataupun teknologi yang sudah ada tetapi baru diimplementasikan ke dalam suatu proyek internal. Pada masa magang ini terdapat dua penerapan teknologi yang akan diimplementasikan ke dalam proyek yang diberikan PT Gits Indonesia. Selama magang dilakukan terdapat dua proyek yang akan dikerjakan. Selama masa magang terdapat dua studi kasus yang diteliti.

Studi kasus pertama adalah sebuah perpustakaan. Perpustakaan ini sedang mengalami kendala terhadap sistem pengelolaan data dan layanan yang belum bisa mengelola data dengan baik dan efektif. Studi kasus yang kedua merupakan *e-commerce* yang sudah memiliki beberapa tampilan untuk pengguna namun belum memiliki bagian *back-end* sehingga tidak memiliki layanan dan fitur yang dapat digunakan oleh serta tidak dapat melakukan pengelolaan terhadap data yang digunakan.

Adapun rumusan masalah yang akan diteliti dari kedua proyek ini adalah Bagaimana cara mengembangkan *back-end website* perpustakaan dengan menerapkan *framework* Gin dengan *database* PostgreSQL dan Bagaimana cara mengembangkan *back-end website e-commerce* dengan menerapkan *framework* gRPC dengan *database* PostgreSQL. Untuk menjawab kedua rumusan masalah perlu dilakukan penelitian dengan melakukan implementasi *framework* Gin dengan *database* PostgreSQL dalam pengerjaan proyek perpustakaan dan implementasi *framework* gRPC dengan *database* PostgreSQL dalam pengerjaan proyek *e-commerce*.

## II. PROFIL PERUSAHAAN

PT Gits Indonesia adalah sebuah perusahaan agensi konsultasi IT yang didirikan pada 13 September 2008 dengan nama C19 House oleh lima pendiri ITB yang tinggal di rumah bernomor C19. Awalnya bernama Ganesha IT Solution, kemudian menjadi Gits Indonesia dengan moto "Get It Simple." Gits telah mengembangkan lebih dari 170 aplikasi, memberikan kontribusi pada kenyamanan 15 juta pengguna.

## III. LANDASAN TEORI

### A. Golang (Go)

Golang, yang juga dikenal sebagai Go, adalah bahasa pemrograman yang dirancang oleh Google. Bahasa ini dikembangkan oleh Rob Pike, Robert Griesemer, dan Ken Thompson. Go dibuat dengan dasar inspirasi dari bahasa pemrograman C dan C++, sehingga sintaksnya mirip dengan kedua bahasa tersebut [1]. Pengembang perangkat lunak memanfaatkan bahasa Golang dalam pembuatan aplikasi berbasis *web*, *cloud*, dan berbagai jenis aplikasi lainnya. [2]. Golang memiliki beberapa kelebihan, antara lain [3]:

- Golang merupakan bahasa yang cepat untuk melakukan compile.
- Golang merupakan bahasa yang simple sehingga mudah untuk dipelajari.
- Bisa dikerjakan secara cross-platform. Golang dapat digunakan di berbagai macam platform seperti UNIX, LINUX, Windows, dan *operating system* lainnya yang dapat bekerja pada perangkat seluler.

### B. PostgreSQL

PostgreSQL, dikenal sebagai "pg" atau "psql," adalah sistem manajemen basis data relasional dengan sifat *open source*. Sistem ini menggunakan model data relasional, di mana data disimpan dalam tabel yang memiliki baris dan kolom. Dalam model ini, data dapat dihubungkan melalui relasi antar tabel [4]. Pengelolaan data di PostgreSQL dilakukan melalui penggunaan bahasa SQL (Structured Query Language). PostgreSQL adalah salah satu sistem manajemen basis data terpopuler di dunia dan mendukung beberapa bahasa pemrograman, termasuk Golang.

### C. Gin Framework

Gin, atau Gin Gonic, adalah sebuah *framework* yang dikembangkan oleh komunitas Golang. Sebagai HTTP *web framework* yang ditulis dalam bahasa Go, Gin didesain untuk memfasilitasi pengembangan aplikasi *web*. Kelebihannya dari Gin meliputi berbagai fitur yang dapat mempercepat dan menyederhanakan proses pengembangan aplikasi *web*. [5]. Dalam *framework* Gin, sebagian kode dapat diperluas dan digunakan kembali, meningkatkan efisiensi dalam pengembangan kode [6].

### D. Google Remote Procedure Calls (gRPC)

gRPC adalah sebuah *framework* yang dikembangkan oleh Google untuk mempermudah pengembangan aplikasi berbasis layanan [7]. *Framework* ini dapat diimplementasikan dalam beberapa bahasa, termasuk Golang, dan telah menjadi pilihan beberapa perusahaan terkemuka, termasuk Netflix. [8].

Dalam gRPC, aplikasi klien dapat memanggil metode langsung pada aplikasi *server* di mesin yang berbeda seakan-akan itu adalah objek lokal, memberikan kemudahan dalam pengembangan aplikasi dan layanan terdistribusi. gRPC didasarkan pada konsep mendefinisikan layanan dengan menentukan metode-metode yang dapat dipanggil secara *remote*, termasuk parameter dan tipe pengembaliannya. Dalam implementasinya, gRPC menggunakan *protocol buffer*. [9].

*Protocol buffer* adalah mekanisme yang dikembangkan oleh Google yang bersifat netral pada bahasa pemrograman dan *platform*, dan dapat diperluas untuk melakukan serialisasi data terstruktur. Dengan menggunakan *protocol buffer*, pengguna dapat menentukan struktur data sesuai kebutuhan, memastikan keberlanjutan dan kemudahan dalam penulisan serta pembacaan data di berbagai bahasa pemrograman. [10].

## IV. HASIL PEKERJAAN

### A. Tahap Implementasi

Proses implementasi dalam kegiatan magang di PT. Gits terdiri dari beberapa langkah, dimulai dengan tahap pembelajaran, diikuti oleh tahap implementasi, dan diakhiri dengan tahap evaluasi.

Tahap awal dalam kegiatan magang adalah tahap pembelajaran. Pada tahap ini penambahan pengetahuan yang diperlukan untuk menangani proyek yang akan diberikan oleh perusahaan akan dilakukan. Pembelajaran ini mencakup pemahaman terhadap bahasa pemrograman yang akan digunakan, penggunaan instrumen, keterampilan komunikasi dalam tim, serta penguasaan sistem yang akan digunakan untuk pelaksanaan proyek.

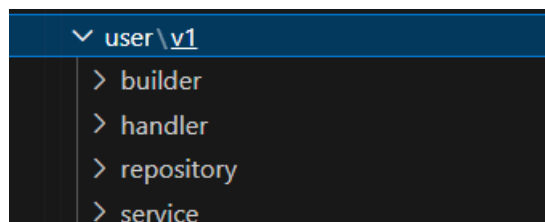
Tahapan implementasi adalah tahap yang melibatkan penerapan konsep-konsep yang telah dipelajari ke dalam pembuatan proyek dalam suatu tim. Implementasi ini terdiri dari dua sub-tahap, yaitu pembuatan kode dan pengujian kode. Dalam tahap

pembuatan kode, digunakan bahasa pemrograman Golang yang akan diintegrasikan dengan *database* PostgreSQL Contoh integrasi antara Golang dengan *database* PostgreSQL dapat dilihat pada gambar berikut.

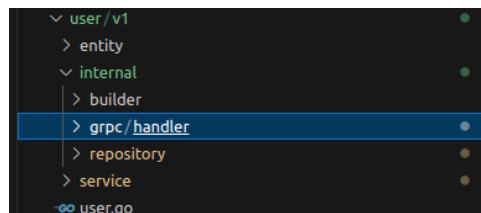
```
func (ur *DeleteUserRepository) DeleteUsers(ctx
context.Context, id uuid.UUID) error {
    if err := ur.db.WithContext(ctx).
        Model(&entity.User{}).
        Where(`id = ?`, id).
        Delete(&entity.User{}, "id = ?",
id).Error; err != nil {
        return errors.Wrap(err, "[UserRepository-
DeleteUsers] error when updating user data")
    }

    return nil
}
```

Gambar 1. Fungsi Menghapus Data Pengguna pada database



Gambar 2. Struktur Folder Gin Framework



Gambar 3. Struktur Folder gRPC Framework

Langkah pembuatan kode akan melanjutkan ke implementasi *framework* Gin dan gRPC. Struktur *folder framework* Gin terlihat pada Gambar 2, sementara untuk *framework* gRPC terlihat pada Gambar 3. Perbedaan utama antara kedua *framework* ini terletak pada bagian *routes* dan struktur *folder*. Pada Gin, *folder entity* atau model dibuat terpisah di luar modul yang sedang dikerjakan, berbeda dengan gRPC di mana *folder* model ditempatkan di dalam *folder* modul yang sedang dikerjakan. Selain itu, dalam penggunaan Gin, rute harus didefinisikan secara terpisah untuk setiap fitur, sementara pada gRPC, pengguna dapat menggunakan *protocol buffer*. Gambar 4 menunjukkan contoh *file protocol buffer*.

```
syntax = "proto3";

package test.user.v1;

option go_package = "test-grpc/api/user/v1;userv1";

import "google/api/annotations.proto";
import "google/api/field_behavior.proto";
import "validate/validate.proto";

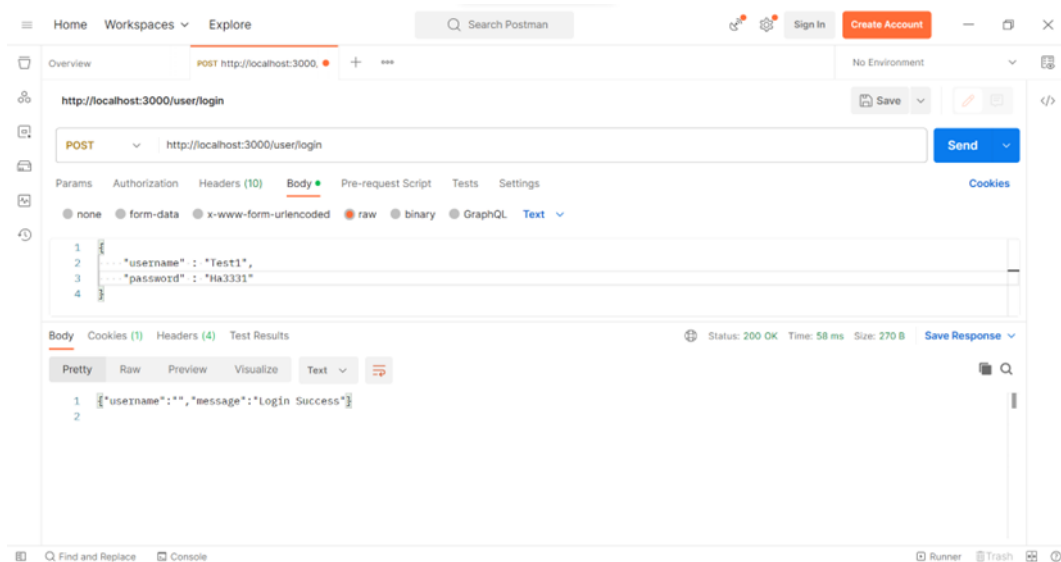
service UserService {
  rpc Login(LoginRequest) returns (LoginResponse)
  {
    option (google.api.http) = {
      post : "/v1/auth/login",
      body : "*"
    };
  }
}

message LoginRequest {
  string email = 1;
  string password = 2;
}

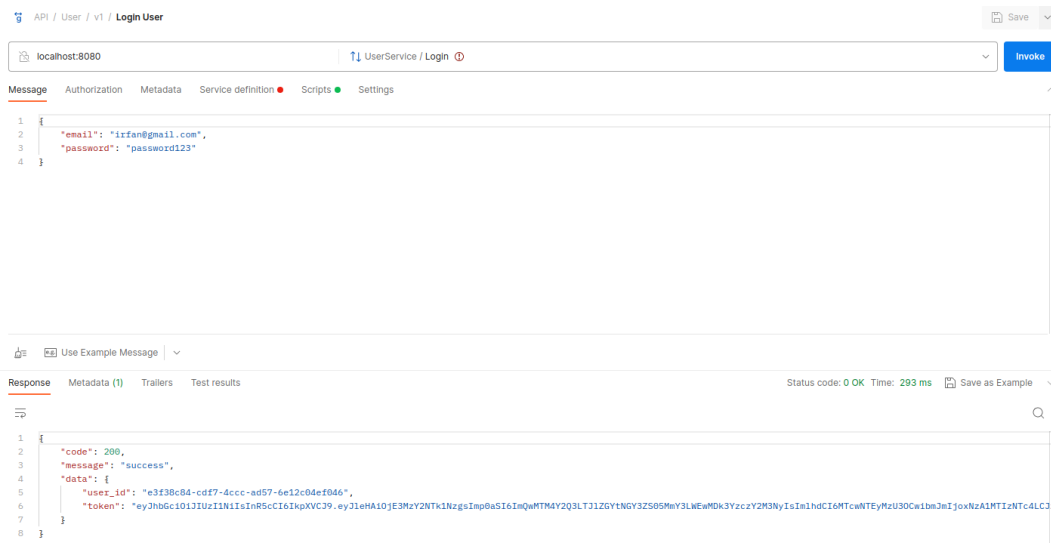
message LoginResponse {
  uint32 code = 1;
  string message = 2;
  TokenData data = 3;
}
```

Gambar 4. File Protocol Buffer

Uji kode dilaksanakan menggunakan Postman, sebuah *tools* yang dapat digunakan untuk menguji fitur-fitur yang telah dibuat pada tahap pembuatan kode sebelumnya. Gambar 5 adalah contoh dari uji coba *Gin framework* dengan menggunakan Postman dan Gambar 6 adalah versi gRPCnya.



Gambar 5. Uji Coba Fitur Login Gin Framework



Gambar 6. Uji Coba Fitur Login gRPC Framework

Tahap evaluasi adalah tahap terakhir pada tahapan implementasi. Pada tahap ini seluruh pekerjaan akan dievaluasi dan dilaporkan kepada mentor atau co-mentor. Tahap ini berfokus pada perbaikan atau penyempurnaan terhadap pekerjaan yang telah dilakukan selama proses implementasi

Selama masa magang di PT. Gits Indonesia, terdapat dua proyek *capstone* yang dikerjakan. Proyek pertama adalah proyek pengembangan *back-end* perpustakaan, sedangkan proyek kedua berkaitan dengan pengembangan *back-end* untuk *platform e-commerce*. Proses pembuatan kedua proyek melibatkan beberapa langkah, termasuk pendefinisian *backlog*, pembuatan skema *Entity Relationship Diagram (ERD)*, pengembangan kode program, dan tahap evaluasi.

#### B. Implementasi Framework Gin pada Proyek Perpustakaan

Proyek *capstone* pertama yang diberikan oleh perusahaan adalah pembuatan *back-end* perpustakaan, yang dilakukan dalam sebuah tim. Setiap anggota tim memiliki tanggung jawab individu dalam mengembangkan fitur-fitur proyek ini. *Framework* yang digunakan dalam pelaksanaan proyek ini adalah Gin.

Fitur-fitur yang dikembangkan dalam proyek ini mencakup pengelolaan data pengguna, data buku, data peminjaman, dan proses transaksi peminjaman. Pembuatan fitur-fitur ini melibatkan proses seperti pembuatan *query*, penentuan metode yang akan digunakan, dan implementasi langkah-langkah tersebut.

Pembuatan *endpoint* atau fitur dimulai dengan merancang struktur tabel *database*. Pada fase ini, semua elemen fitur dan modul yang dibutuhkan dijelaskan secara rinci. Setelah pendefinisian telah terdefinisi dengan lengkap, langkah berikutnya adalah membuat tabel-tabel *database*, dan setelah semua tabel terbentuk, proses dilanjutkan ke tahap pemrograman yang melibatkan pembuatan beberapa *folder*.

Langkah awal adalah pembuatan *folder repository* yang mengandung kode pemrograman Golang yang terintegrasi dengan *database*, seperti yang terlihat pada Gambar 1. *Folder* ini berperan sebagai penghubung antara Golang dan *database* untuk mengambil informasi dari *database*, dengan tujuan untuk mendukung implementasi fitur atau *endpoint* tertentu.

Setelah *folder query* terbentuk, langkah selanjutnya adalah pembentukan *folder service*. *Folder service* akan mengaitkan *query* yang telah dibuat dengan *folder handler* yang nantinya akan dibuat. Di dalam *folder handler*, fitur akan dikembangkan untuk menerima data yang masuk dari *endpoint* yang dituju.

Setelah terbentuk ketiga *folder* tersebut, langkah selanjutnya adalah penyusunan *folder builder*. *Folder builder* diciptakan untuk memberikan struktur dan kerapian pada modul sehingga lebih mudah dihubungkan dengan *folder routes*. Setelah semua *folder* pada modul selesai, pembuatan routes dapat dilakukan dengan memanggil fungsi-fungsi *handler* yang sudah diorganisir dengan rapi di dalam *builder*, dan menentukan *method* yang diinginkan untuk setiap *endpoint* yang akan dibuat. Struktur *folder* Gin yang disajikan pada Gambar 2 memberikan gambaran lebih jelas.

#### C. Implementasi Framework gRPC pada Proyek Perpustakaan

Proyek berikutnya yang diberikan oleh perusahaan adalah pengembangan *back-end* untuk situs *e-commerce*. Tim yang bertanggung jawab atas proyek ini terdiri dari dua anggota, di mana setiap individu memiliki tugas dan tanggung jawab yang

spesifik. Hasil proyek pada jurnal ini mencakup pengembangan modul pengguna, peran (*role*), dan *dashboard* untuk menampilkan data tertentu. *Framework* yang akan diterapkan dalam proyek ini adalah gRPC.

Pembuatan *endpoint* atau fitur dalam proyek ini hampir serupa dengan proyek sebelumnya, dimulai dari perancangan tabel *database* hingga pembuatan seluruh *folder* selama proses pemrograman. Dalam pembuatan proyek ini, struktur *folder* gRPC memiliki perbedaan yang cukup mencolok dengan Gin. Detail struktur gRPC dapat dilihat pada Gambar 3.

Setelah semua *folder* terbentuk, mulai dari *folder repository* hingga *folder handler*, langkah selanjutnya adalah membuat *folder protocol buffer* yang berisi *file proto*, seperti yang terlihat pada Gambar 4. *File* ini memiliki fungsi untuk menghasilkan *file* yang ada di dalam *folder routes*. Meskipun secara tampilan terdapat perbedaan yang signifikan antara *routes* pada Gin dan gRPC, keduanya memiliki fungsi yang serupa. Melalui penggunaan *protocol buffer*, proses pembuatan *file* dalam *folder routes* dapat dilakukan secara otomatis.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Dalam implementasi *framework* Gin, kinerja Gin sangat optimal dan sangat mudah digunakan oleh pengembang yang masih pemula.

Gambar 5 menunjukkan performa *framework* ini dengan *response time* yang cukup cepat. *Framework* ini dianggap mudah digunakan karena tidak memerlukan pemahaman mendalam tentang Golang, dan memiliki struktur arsitektur yang efisien untuk memproses fitur, seperti terlihat pada Gambar 2.

Dalam implementasi *framework* gRPC, performanya juga terlihat baik, seperti yang tergambar pada Gambar 6, namun tidak begitu ramah pengguna bagi sebagian pengembang. Hal ini terlihat dari struktur *folder* yang terpisah dan penggunaan *protocol buffer*. Meskipun penggunaan *protocol buffer* membuat pembuatan *back-end* lebih praktis, namun diperlukan pemahaman mendalam terkait *protocol buffer* agar pengembang dapat memanfaatkan *framework* ini secara maksimal.

### B. Saran

Setelah pelaksanaan magang ini, ada beberapa saran yang dapat disampaikan. Berikut adalah sarannya:

1. Pembuatan proyek disertai data *table* atau *pagination* agar data yang ditampilkan bisa diberi batas per halamannya
2. Pembuatan proyek lebih banyak mengimplementasikan Redis Cache agar kinerja *database* berkurang

## DAFTAR PUSTAKA

- [1] A. N. Prayogo, "Dasar Pemrograman Golang," 4 April 2021. [Online]. Available: <https://dasarpemrogramangolang.novalagung.com>. [Accessed 14 April 2023].
- [2] H. B. Noorazman, *Golang for Jobseekers: Unleash the power of Go programming for career advancement*, BPB Publications, 2023.
- [3] S. b. Uzayr, *GoLang: The Ultimate Guide*, Boca Raton: CRC Press, 2022.
- [4] H.-J. Schönig, *Mastering PostgreSQL 15: Advanced techniques to build and manage scalable, reliable, and fault-tolerant database applications*, Birmingham: Packt Publishing, 2023.
- [5] Gin Team, "Introduction: Gin Web Framework," Gin Team, 29 April 2022. [Online]. Available: <https://gin-gonic.com/docs/introduction/>. [Accessed 5 Mei 2023].
- [6] M. Labouardy, *Building Distributed Applications in Gin*, Birmingham: Packt Publishing Ltd., 2021.
- [7] K. Indrasiri and D. Kuruppu, *gRPC: Up & Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes*, 2020: O'Reilly Media.
- [8] gRPC Authors, "About gRPC: Who is using gRPC and why," gRPC, 2023. [Online]. Available: <https://grpc.io/about/>. [Accessed 7 November 2023].
- [9] gRPC Authors, "Basics tutorial: A basic tutorial introduction to gRPC in Go.," gRPC, 13 September 2023. [Online]. Available: <https://grpc.io/docs/languages/go/basics/>. [Accessed 10 November 2023].
- [10] Protocol Buffer Team, "Protocol Buffers: Protocol Buffers are language-neutral, platform-neutral extensible mechanisms for serializing structured data.," Google LLC All, 2023. [Online]. Available: <https://protobuf.dev>. [Accessed 21 November 2023].