PERANCANGAN *APPLICATION PROGRAMMING INTERFACE* (API) BUKU INFORMATIKA DENGAN MENGGUNAKAN *GOOGLE REMOTE PROCEDURE CALL* (GRPC) GOLANG SEBAGAI KOMUNIKASI (STUDI KASUS PERPUSTAKAAN JAKARTA GLOBAL UNIVERSITY)

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer

ABIYYU IHSAN
200111401023

PROGRAM STUDI TEKNIK INFORMATIKA
JAKARTA GLOBAL UNIVERSITY
2024

ABSTRAK

Saat ini ketersediaan buku di perpustakaan JGU memiliki 473 buku informatika yang ada. Penambahan buku informatika di perpustakaan JGU bisa terus bertambah di setiap tahunnya yang membutuhkan proses *query resource* dengan *database* dan sistem harus cepat, sehingga diperlukan suatu sistem untuk memperbaiki performa dari komunikasi *database* dengan sistem dengan API. *Application Programming Interface* (API) merupakan suatu antarmuka yang menghubungkan berbagai sistem aplikasi *software* yang diakses secara bersama, API sebagai program *serverside* yang berkomunikasi dengan basis data secara langsung. Perancangan API ini menggunakan SDLC *Methodology* untuk pengembangan sistemnya. Diawali dengan perencanaan, analisis, desain, implementasi, uji coba, dan pemeliharaan. Hasil *black box testing* dengan *endpoint* pengujian fungsional API, menunjukan API dapat berfungsi dengan baik. Metode gRPC berbasis Golang pada API buku informatika berdasarkan hasil pengujian API telah sesuai dengan yang diharapkan dengan tingkat keberhasilan 100%. Dengan hasil ini maka fungsi dari API buku informatika yang dibuat telah sesuai dengan perancangan.

Kata kunci: *Perpustakaan JGU, API, SDLC Methodology, gRPC, Golang*

ABSTRACT

The availability of books in the JGU library includes 473 informatics books. The addition of informatics books in the JGU library can continue to increase each year, which requires a resource query process with the database, and the system must be fast. Therefore, a system is needed to improve the performance of communication between the database and the system using an API. An Application Programming Interface (API) is an interface that connects various software application systems that are accessed together. The API serves as a server-side program that communicates directly with the database. The design of this API uses the SDLC Methodology for system development. It begins with planning, analysis, design, implementation, testing, and maintenance. The results of black box testing with functional API endpoint testing show that the API functions well. The gRPC method based on Golang in the informatics book API, based on API testing results, has met expectations with a 100% success rate. With this result, the function of the informatics book API that was created has met the design specifications.

Keywords: *JGU Library, API, SDLC Methodology, gRPC, Golang*

DAFTAR ISI

PERNYATAAN ORISINALITAS SKRIPSI ..... **ii**
HALAMAN PENGESAHAN PEMBIMBING ..... **iii**
HALAMAN PENGESAHAN DEWAN PENGUJI ..... **iv**
KATA PENGANTAR ..... **v**
PERNYATAAN PERSETUJUAN PUBLIKASI AKADEMIS ..... **vi**
ABSTRAK ..... **vii**
ABSTRACT ..... **viii**
DAFTAR ISI ..... **ix**
DAFTAR GAMBAR ..... **xi**
DAFTAR TABEL ..... **xiv**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan Penelitian 3
1.4 Manfaat Penelitian 3
1.5 Batasan Masalah 3
BAB II KAJIAN PUSTAKA **3**
2.1 Penelitian Sebelumnya 4
2.2 Landasan Teori 6
2.2.1 Rancangan 6
2.2.2 Komunikasi 7
2.2.3 Rancangan Konsep Dasar API 7
2.2.4 Bahasa Pemrograman Golang 8
2.2.5 Konsep gRPC sebagai Protokol Komunikasi 9
2.2.6 PostgreSQL sebagai *Database* 10
2.2.7 Visual Studio Code 11
2.2.8 Postman sebagai Dokumentasi API 12
2.2.9 Swagger 13
2.2.10 DBeaver 14
BAB III METODE PENELITIAN **15**

3.1 Metode Pengembangan Sistem 15
3.1.1 Perencanaan 15
3.1.2 Analisis 17
3.1.3 Desain 18
3.1.4 Implementasi Basis Data 18
3.1.5 Uji Coba Hasil yang Diharapkan 19
3.1.6 Pemeliharaan 20
BAB IV HASIL PEMBAHASAN 21
4.1 Hasil Implementasi 21
4.1.1 Hasil Implementasi gRPC dan Protobuf 21
4.1.2 Hasil Implementasi Swagger 28
4.1.2.1 Detail Parameter dan Respon API 28
4.1.3 Hasil Implementasi PostgreSQL di DBeaver 33
4.1.4 Hasil Implementasi API 41
4.1.4.1 Pembuatan Struktur Folder API 41
4.1.4.2 Pengujian Repository API 47
4.1.4.3 Dokumentasi Respon API di Terminal 48
4.1.5 Hasil Implementasi API di Postman 50
4.1.5.1 Dokumentasi API dengan Menggunakan HTTP 50
4.1.5.2 Dokumentasi API dengan Menggunakan gRPC 58
4.2 Hasil Pengujian *Black Box Testing* 63
BAB V KESIMPULAN DAN SARAN 65
5.1 Kesimpulan 65
5.2 Saran 65
DAFTAR PUSTAKA 66
LAMPIRAN-LAMPIRAN 70

DAFTAR GAMBAR

Gambar 2. 1 Logo Golang 8
Gambar 2. 2 Logo gRPC 9
Gambar 2. 3 Logo PostgreSQL 10
Gambar 2. 4 Logo VSCode 11
Gambar 2. 5 Logo Postman 12
Gambar 2. 6 Logo Swagger 13
Gambar 2. 7 Logo DBeaver 14
Gambar 3. 1 Tahapan Metode SDLC 15
Gambar 3. 2 Activity Diagram Menampilkan Data Lis Buku Informatika 17
Gambar 4. 1 Membuat Struktur Folder untuk gRPC 21
Gambar 4. 2 Membuat File Buf.yaml 25
Gambar 4. 3 Membuat Buf.gen.yaml 26
Gambar 4. 4 Hasil Generate Buf Mod Update 27
Gambar 4. 5 Membuat Makefile untuk Alur Proto 27
Gambar 4. 6 Endpoint yang Dihasilkan dari Generate Proto 28
Gambar 4. 7 Detail Parameter Get yang Ditampilkan 29
Gambar 4. 8 Lanjutan Detail Parameter Get yang Ditampilkan 30
Gambar 4. 9 Hasil Implementasi Status Respon 200 Successful Response 31
Gambar 4. 10 Hasil Implementasi Status 400 Bad Request 31
Gambar 4. 11 Hasil Implementasi Status 404 Not Found 32
Gambar 4. 12 Hasil Implementasi Status 500 Server Error 33
Gambar 4. 13 Menjalankan Localhost PostgreSQL 33
Gambar 4. 14 Membuat Koneksi PostgreSQL di DBeaver 34
Gambar 4. 15 Melakukan Download Driver PostgreSQL di DBeaver 35
Gambar 4. 16 Melakukan Tes Koneksi di DBeaver 35
Gambar 4. 17 Transfer Targets 36
Gambar 4. 18 Memilih Data Buku dari Perpustakaan JGU 36
Gambar 4. 19 Input File(s) 37
Gambar 4. 20 Tables Mapping 37
Gambar 4. 21 Configure Metadata Structure 38

Gambar 4. 22 Data Load Settings 39
Gambar 4. 23 Confirm Data Transfer 39
Gambar 4. 24 Tampilan Hasil Data Impor yang Digunakan 40
Gambar 4. 25 Perintah Query Database untuk Menampilkan isi Data 41
Gambar 4. 26 Struktur Folder API..... 41
Gambar 4. 27 Lanjutan Struktur Folder API..... 43
Gambar 4. 28 Akhir Struktur Folder API..... 45
Gambar 4. 29 Hasil Tes Pengujian Repository 47
Gambar 4. 30 Pengujian TestRepo_Close pada File Repository 48
Gambar 4. 31 Menjalankan File Localhost API..... 48
Gambar 4. 32 Respon Error pada File Postgresql.go 49
Gambar 4. 33 Hasil Respon Jika 200 Ok di Terminal 49
Gambar 4. 34 Hasil Respon Jika 404 Not Found di Terminal 50
Gambar 4. 35 Tampilan Utama Aplikasi Postman..... 50
Gambar 4. 36 Tampilan Import File di Postman 51
Gambar 4. 37 Memilih Folder Proto yang Digunakan..... 51
Gambar 4. 38 Impor Folder Berhasil Dipilih 52
Gambar 4. 39 Hasil Detail Impor Folder Protobuf 52
Gambar 4. 40 Daftar Query Params pada Postman..... 53
Gambar 4. 41 Daftar Headers pada Postman 53
Gambar 4. 42 Contoh Dokumentasi Jika Successful Response 54
Gambar 4. 43 Contoh Dokumentasi Jika 500 Internal Server Error 54
Gambar 4. 44 Hasil Implementasi jika 500 Internal Server Error 55
Gambar 4. 45 Hasil Implementasi Jika 404 Not Found 56
Gambar 4. 46 Hasil Implementasi Jika 400 Bad Request 56
Gambar 4. 47 Hasil Implementasi Jika Respon 200 OK 57
Gambar 4. 48 Membuat API dengan gRPC pada Postman..... 58
Gambar 4. 49 Tampilan Awal dari gRPC pada Postman..... 59
Gambar 4. 50 Proses Import File Proto..... 59
Gambar 4. 51 Hasil Import File Protobuf 60
Gambar 4. 52 Hasil Implementasi Status Respon Jika 2 Unknown..... 60
Gambar 4. 53 Hasil gRPC Jika 5 Not Found di gRPC 61

Gambar 4. 54 Hasil Respon Jika 3 Invalid Argument di Grpc 62
Gambar 4. 55 Hasil Implementasi Jika OK di gRPC..... 63

DAFTAR TABEL

Tabel 2. 1 Referensi Sebelumnya..... 4
Tabel 3. 1 Waktu Pelaksanaan Pembuatan API..... 16
Tabel 3. 2 Info Buku Informatika 18
Tabel 3.3 Uji Coba yang Diharapkan dari API Buku Informatika 19

BAB I
PENDAHULUAN

1.1 Latar Belakang

Di dunia teknologi, aplikasi modern biasanya mengadopsi arsitektur terdistribusi di berbagai komponen yang bekerja secara independen dan berkomunikasi melalui jaringan. Dalam lingkungan semacam itu, faktor penting yang mempengaruhi skalabilitas, kinerja, dan responsivitas di seluruh sistem. Protokol komunikasi adalah seperangkat konvensi yang digunakan dua komputer atau perangkat komunikasi untuk memvalidasi format dan isi pesan yang dipertukarkan (Popovic, 2018a).

Application Programming Interface (API), merupakan suatu antarmuka yang menghubungkan berbagai sistem aplikasi *software* yang diakses secara bersama (Triawan & Siboro, 2021). API sebagai program *serverside* yang berkomunikasi dengan basis data secara langsung. API memberikan respon dengan bentuk *JavaScript Object Notation* (JSON) sebagai bentuk format data yang bisa digunakan oleh *developer* di berbagai bahasa pemrograman. Tanpa ketentuan sebuah format data dari API, akan mengakibatkan penulisan program dari API menjadi tidak dapat dipahami, karena tidak memiliki format data dari API. Akibat tidak efisiennya sebuah kode program akan mengakibatkan performa dan waktu eksekusi API tidak menjadi efisien, jika program tidak sesuai format maka akan berdampak pada pengembangan dan pemeliharaan API.

Buku informatika merupakan bentuk literatur pada penjelasan dan pengembangan tentang teknologi informasi, pemrograman, *database*, *cyber security*, *data analyst*, dan topik lainnya (Novega Pratama Adiputra, 2020). Saat ini ketersediaan buku di perpustakaan JGU memiliki 473 buku informatika yang ada. Penambahan buku informatika di perpustakaan JGU bisa terus bertambah di setiap tahunnya yang membutuhkan proses *query resource* dengan *database* dan sistem harus cepat, sehingga diperlukan suatu sistem untuk memperbaikinya dari komunikasi *database* dengan sistem dan API. Buku informatika telah terjadinya perubahan ke dalam bentuk digital, kebutuhan pada pengembangan buku informatika memiliki tujuan supaya menjadi efisien, responsif, dan lebih interaktif

dalam penggunaan buku. Salah satu solusi dari tantangan ini, ialah melakukan perancangan API buku informatika untuk memberikan potensi yang terintegrasi dengan *software* pendukung. Supaya pembaca bisa berinteraksi dengan isi buku melalui simulasi, latihan interaktif dengan komponen yang disediakan secara dinamis yang dapat membantu pembaca pada pemahaman konsep-konsep yang lebih baik.

Google Remote Procedure Call (gRPC), merupakan suatu kerangka kerja *Remote Procedure Call* (RPC) *opensource* modern dan berkinerja tinggi yang bisa dijalankan di lingkungan manapun. gRPC bisa menghubungkan berbagai layanan secara efisien di dalam dan di seluruh pusat data dengan dukungan *pluggable* untuk penyeimbangan penelusuran, penyeimbangan kapasitas, dan autentikasi. Hal ini juga berlaku pada komputasi terdistribusi untuk menghubungkan aplikasi seluler, perangkat, dan *browser* ke layanan *backend* (gRPC, 2024). Dengan menggunakan *Protocol Buffers* atau disingkat menjadi Protobuf sebagai format data, gRPC menawarkan kemampuan skalabilitas dan performa tinggi.

Di lain sisi, bahasa pemrograman Golang telah mendapatkan popularitas karena kinerja tinggi, manajemen konkurensi yang efisien, dan kesederhanaan sintaksis. Penggunaan kombinasi gRPC dan Golang dapat memberikan kualitas untuk menciptakan sistem yang efisien, responsif, dan dapat diandalkan dalam komunikasi antar layanan. Berdasarkan uraian di atas "Perancangan *Application Programming Interface* (API) Buku Informatika dengan Menggunakan *Google Remote Procedure Call* (gRPC) Golang sebagai Komunikasi (Studi Kasus Perpustakaan Jakarta Global University)". Diharapkan mampu menjadi sebuah kontribusi untuk tantangan ini supaya pembaca dapat berinteraksi dengan isi buku melalui simulasi yang disediakan secara dinamis.

1.2 Rumusan Masalah

Perancangan API dengan menggunakan gRPC dan Golang sebagai cara kerja komunikasi dalam upaya peningkatan performa dan efisiensi sistem, terdapat beberapa masalah yang perlu diselesaikan:

1. Bagaimana perancangan API buku informatika dengan menggunakan gRPC dan Golang?

2. Bagaimana perancangan gRPC untuk mempermudah pembaca dalam mencari buku?

1.3 Tujuan Penelitian

Berikut beberapa aspek dari tujuan penelitian ini, sebagai berikut:

1. Merancang sebuah API untuk mengelola informasi tentang buku informatika, dengan memanfaatkan penggunaan teknologi gRPC sebagai kerangka kerja RPC dan Golang sebagai bahasa pemrogramannya dalam perancangan API.
2. Melakukan konfigurasi gRPC sebagai protokol komunikasi untuk pembuatan API buku informatika di Perpustakaan JGU dengan menggunakan format Protobuf sebagai skema data.

1.4 Manfaat Penelitian

Berikut aspek-aspek manfaat yang dapat dijelaskan, antara lain:

1. Memberikan kontribusi di bidang informatika dalam pengembangan API dengan menggunakan gRPC sebagai protokol komunikasi.
2. Penelitian membantu pembaca untuk memahami tentang penggunaan konsep gRPC sebagai protokol komunikasi dan Golang sebagai bahasa pemrograman dalam pembuatan API di Perpustakaan JGU.

1.5 Batasan Masalah

Batasan masalah yang diidentifikasi dalam penelitian ini, ialah:

1. Merancang API dengan menggunakan gRPC sebagai alat komunikasi dan Golang sebagai bahasa pemrogramannya.
2. Berfokus untuk menggunakan gRPC sebagai protokol komunikasi dan Golang sebagai bahasa pemrograman dalam pembuatan API buku informatika di Perpustakaan JGU.