

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi yang berkembang pesat pada saat ini memerlukan komunikasi yang cepat untuk pengiriman data antar berbagai *device*, salah satu teknologi tersebut adalah *Web service*. *Web service* merupakan mekanisme komunikasi dua aplikasi atau mesin terlepas dari arsitektur dan teknologi yang digunakan [1].

*REST API* adalah pertukaran data atau informasi antara satu aplikasi dan aplikasi lainnya yang menggunakan protokol HTTP [2]. Sedangkan *GraphQL* sendiri adalah *query language* yang telah terbukti di kalangan pengembang aplikasi. Pada tahun 2015 kerangka kerja *GraphQL* diperkenalkan oleh facebook sebagai cara alternatif untuk mengakses data [3].

Dalam implementasinya penulis menggunakan *Runtime Environment Nodejs* sebagai alat penelitian. *Nodejs* menjalankan mesin *JavaScript V8*, inti dari google chrome, diluar *browser*. Ini memungkinkan *Nodejs* memiliki performa yang sangat baik. Aplikasi *nodejs* dijalankan dalam satu proses tanpa membuat *thread* baru untuk setiap *request*. *Nodejs* menyediakan metode *asynchronous I/O* di pustaka standarnya yang mencegah *JavaScript* dari *blocking code* [4].

Munculnya teknologi tersebut menimbulkan masalah bagi para *developer* salah satu contohnya adalah *Netflix*. Masalah tersebut berupa *URL* yang terlalu banyak untuk setiap *request* ke *server*. Performa kecepatan mana yang lebih baik antara *REST API* dan *GraphQL* [5]. Oleh karena itu penulis bermaksud membuat sebuah kajian untuk membantu *developer* untuk menentukan teknologi yang paling sesuai untuk proses pengembangan aplikasi.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan diatas maka penulis merumuskan masalah dalam penelitian ini berupa “Apa perbedaan performa kecepatan antara *REST API* dan *GraphQL* menggunakan *runtime environment node js?*”.

## 1.2 Batasan Masalah

Adapun batasan ruang lingkup dari penelitian ini sebagai berikut:

1. Pengujian dilakukan di *server local (localhost)*.
2. *Database* menggunakan *Mongodb*.
3. Data di *database* yang digunakan berupa *dummy data*.
4. Sistem yang dibangun baik untuk *REST API* maupun *GraphQL* menggunakan *nodejs* dengan *framework Expressjs*.
5. Pengujian *Gorilla Test* menggunakan aplikasi *Apache JMeter*.
6. Pengujian *Whitebox Test* menggunakan *package mocha* dan *chai*.
7. Menggunakan *ES6* untuk penulisan *syntax*-nya
8. Menggunakan *package Nodemon* untuk *log*.
9. *Package Monggose* untuk skema *MongoDB*.
10. *Package Dotenv* untuk pengambilan konfigurasi file pada *project*.
11. Menggunakan *GraphQL-yoga* untuk *GraphQL*.
12. Komputer yang digunakan memiliki CPU 3,8 GHz AMD Ryzen 5 3600x.
13. Komputer memiliki RAM 16GB.
14. Komputer server menggunakan sistem operasi windows 10.
15. Output dari *REST API* dan *GraphQL* berupa *JSON*.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menentukan performa kecepatan mana yang lebih optimal dalam menerima *request* dan memberikan *response* antara *REST API* dan *GraphQL* menggunakan *Runtime Environment Nodejs*.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi *Developer*:

Penelitian ini dapat memberikan informasi performa yang lebih optimal antara kedua teknologi tersebut dan dapat pertimbangan untuk memilih skema dari *web service* untuk pengembangan aplikasi.

2. Bagi Peneliti

Manfaat yang didapat bagi peneliti adalah mengimplementasikan ilmu yang telah didapat agar dapat digunakan oleh orang lain.

## 1.6 Metode Penelitian

Metode penelitian yang digunakan adalah metode kuantitatif dengan menggunakan analisis perbandingan antara *REST API* dengan *GraphQL* yang didasarkan dari performa kecepatan data. *Dataset* untuk mengukur performa kedua teknologi tersebut *degenerate* menggunakan [www.mockaroo.com](http://www.mockaroo.com).

## 1.7 Sistematika Penulisan

### BAB I PENDAHULUAN

Pada bab ini akan diuraikan mengenai latar belakang masalah, rumusan masalah, batasan masalah, maksud dan tujuan penelitian, metode penelitian dan sistematika penulisan.

### BAB II LANDASAN TEORI

Pada bab ini akan diuraikan secara teoritis tentang konsep dasar dari *REST API* dan *GraphQL* menggunakan *runtime environment* Nodejs.

### BAB III METODOLOGI PENELITIAN

Pada bab ini akan diuraikan gambaran umum tentang objek penelitian, analisis masalah, analisis kebutuhan, perancangan *REST API*, serta perancangan *GraphQL*.

#### **BAB IV IMPLEMENTASI DAN PEMBAHASAAN**

Pada bab ini menjelaskan tentang proses hasil dari pengujian performa kedua teknologi tersebut.

#### **BAB V PENUTUP**

Kesimpulan dan saran dari hasil komparasi dari *REST API* dan *GraphQL* akan diuraikan dalam bab ini.

#### **DAFTAR PUSTAKA**

Bagian ini berisikan daftar pustaka yang dipakai sebagai acuan referensi literatur dalam penelitian ini.

