

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dengan perkembangan teknologi, kebutuhan sistem informasi semakin meningkat. Kebutuhan sistem yang meningkat dan jumlah pengguna yang juga terus meningkat perlu diimbangi dengan adanya pembaruan sistem yang lebih efisien, skalabilitas tinggi, cepat, *secure*, ketersediaan data tinggi dan tidak terbatas pada platform yang digunakan. Banyak sistem terdistribusi skalabilitasnya harus tinggi, artinya sistem tersebut harus dapat diterapkan secara ekonomis dalam berbagai ukuran dan konfigurasi (Prasad Jogalekar, 2000).

Pada pengembangannya terdapat arsitektur yang disebut dengan *Service Oriented Architecture* (SOA). Untuk memahami konsep SOA, kita akan terlebih dahulu membahas mengenai Service. Dalam konteks SOA, Service merujuk pada serangkaian fungsi atau prosedur yang memberikan respons ketika diminta oleh pengguna (*client*). SOA merupakan suatu pendekatan arsitektur teknologi yang mengadopsi prinsip-prinsip Orientasi Layanan (*Service-Orientation*). Pendekatan ini mengurai proses bisnis kompleks menjadi serangkaian layanan yang sederhana dengan tujuan menyelesaikan masalah tertentu. SOA tidak terpacu pada teknologi spesifik, namun lebih mengarah pada pendekatan modular dalam pengembangan perangkat lunak (Erl, 2005).

Dengan menggunakan web service, memungkinkan pengembangan *software* untuk dapat lebih efisien karena pada prosesnya pengembangan tidak akan terbatas pada *operating system* ataupun platform sehingga proses pertukaran informasi tidak perlu melakukan pergantian platform ataupun *operating sistem* menyesuaikan

kebutuhan. Dalam pengembangannya web service atau disebut *API* perlu menyertakan dokumentasi dan penjelasan dari fungsi/rutin yang disediakan dan cara pemakaiannya seperti penjelasan *end-point* dan *method* yang dapat digunakan. Web Service merupakan sistem perangkat lunak yang dirancang untuk memfasilitasi interoperabilitas antar mesin dalam sebuah jaringan. Antarmuka dari Web Service dijelaskan dalam format mesin seperti *WSDL*. Interaksi dengan Web Service dilakukan oleh sistem lain melalui antarmuka menggunakan pesan seperti *SOAP*. Umumnya, pesan-pesan ini menggunakan protokol *HTTP* dan format *XML*, yang merupakan standar web. Saat ini, penelitian dan pengembangan untuk meningkatkan Web Service terus dilakukan, baik dari segi konseptual maupun teknis. Secara umum, pertukaran data menjadi fokus utama dalam konteks Internet saat ini. Sejalan dengan perkembangan tuntutan dan teknologi, pertukaran data juga dapat dilakukan pada tingkat layanan. (Lily Wulandari, 2006).

*API* memiliki beragam jenis arsitektur, salah satunya adalah *REST* (*Representational State Transfer*). *REST* merupakan kumpulan prinsip arsitektur yang memfasilitasi transmisi data melalui antarmuka standar seperti *HTTP*. Banyak *API* yang dibangun dengan menggunakan pendekatan *REST*. Untuk mengakses sumber daya, *URI* (*Uniform Resource Identifier*) digunakan sebagai identifikasi saat melakukan permintaan ke server *HTTP*. Terdapat beberapa metode *HTTP* yang digunakan seperti *GET*, *PUT*, *DELETE*, dan *POST*. Dalam implementasinya, output yang dihasilkan oleh *REST* lebih fleksibel karena dapat berupa format *JSON*. (Rizqi Kartika Safitri, 2021).

Dalam perkembangan arsitektur komunikasi *API*, *GraphQL* adalah teknologi yang dikembangkan oleh *Facebook* dengan tujuan menciptakan *API* yang lebih

fleksibel dan efisien berdasarkan schema. Pendekatan ini membantu dalam menyelesaikan tantangan yang dihadapi saat menggunakan metode komunikasi berbasis *REST* untuk mengembangkan *API*. Dengan *GraphQL*, pengembang dapat menentukan secara spesifik data yang mereka perlukan dalam sebuah query atau respons. Hal ini memungkinkan pihak klien untuk mengontrol bentuk data yang diterima. (Antonio Quiña-Mera, 2023).

Dikarenakan dalam membangun sebuah web service dapat dipastikan akan memiliki banyak client sehingga optimasi sangat perlu dilakukan baik dalam hal kebutuhan efisiensi dari segi teknologi yang mengacu pada response time dari permintaan dan pemberian data antara server-client dan penggunaan memori pada server. Dengan melakukan pengujian dengan metode stress testing pada web service untuk mendapatkan Response Time, Throughput dan Error pada web service berbasis *REST* dan *GraphQL* diharapkan dapat mengetahui metode komunikasi *API* yang memiliki efisiensi paling optimal dengan skenario pengujian berupa *package size* dan waktu inflasi data yang diatur sesuai dengan pengujian statistik yang akan dilakukan dengan bantuan actuator berupa *Apache Jmeter*. Sebuah sistem yang berkualitas harus dilengkapi dengan *API* yang handal. Untuk memastikan kualitas *API* dalam sistem, pengujian diperlukan, salah satunya adalah pengujian stres. Pengujian stres adalah metode pengujian yang digunakan untuk menguji stabilitas dan keandalan sebuah sistem. Tujuan utamanya adalah untuk menilai ketahanan sistem dan kemampuannya dalam menangani kesalahan saat terpapar pada beban yang sangat tinggi. (Ni Luh Ayu Sonia Ginasaria, 2021).

## 1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian, maka rumusan masalah dalam penelitian ini dapat dijabarkan sebagai berikut:

1. Bagaimana analisis performa pada metode komunikasi *API* berbasis *REST*?
2. Bagaimana analisis performa pada metode komunikasi *API* berbasis *GraphQL*?
3. Manakah metode komunikasi *API* yang paling optimal antara *REST* dan *GraphQL*?

## 1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah untuk mendapatkan hal – hal berikut:

1. Mengetahui metode komunikasi atau arsitektur *API* selain *REST API*.
2. Mengetahui penyebab metode komunikasi berupa *REST API* lebih familiar dan lebih banyak digunakan pada pengembangan web service.
3. Mengetahui performa pada metode komunikasi *API* berbasis *REST*.
4. Mengetahui performa pada metode komunikasi *API* berbasis *GraphQL*.
5. Mengetahui metode komunikasi *API* yang paling optimal berdasarkan komparasi dan pertimbangan pada pengujian yang sudah ditentukan.

#### 1.4 Batasan Penelitian

Agar pembahasan dari penelitian ini tidak terlalu luas dan keluar dari konteks yang telah ditentukan, maka batasan penelitian hanya terkait pada:

1. Metode komunikasi atau arsitektur *API* yang diuji hanya berbasis *REST* dan *GraphQL*.
2. Acuan yang menjadi tolak ukur komparasi hanya pada, *Response Time*, *Throughput* dan *Error*.
3. *Package size* atau data payload yang digunakan sebagai beban berukuran kurang dari 1mb per row data dan hanya berupa data *JSON*.
4. Database yang digunakan menggunakan database *PostgreSQL* yang terinstall didalam komputer server.
5. Pengujian dilakukan pada aplikasi edukasi pernikahan bernama HaFy.
6. Testing hanya dilakukan pada 1 service yang mencakup pengambilan data, dan manipulasi data pada service artikel.
7. Testing method menggunakan 1 juta data dummy yang sudah tersedia di dalam database.
8. Framework yang digunakan menggunakan *Java Springboot*.

#### 1.5 Manfaat Penelitian

Manfaat yang diharapkan dari dilakukannya penelitian adalah sebagai berikut:

1. Bagi para pengembang perangkat lunak, diharapkan dapat membantu menentukan metode komunikasi atau arsitektur *API* yang paling optimal untuk digunakan dalam membangun web service.

2. Bagi peneliti selanjutnya, diharapkan dapat digunakan sebagai acuan untuk melakukan pengujian metode komunikasi atau arsitektur *API* yang lainnya.