

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Penelitian perbandingan dengan topik yang serupa pernah diteliti oleh Aleksi Ritsilä (2017) dengan topik “*GraphQL: The API Design Revolution*”. Penelitian berfokus perbandingan pertukaran data antara *klien* dan server dari masing-masing kedua metode tersebut.

Achmad Fauzi Harismawan, Agi Putra Kharisma dan Tri Afirianto (2018) dengan judul “Analisis Perbandingan Performa *Web Service* Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada klien Berbasis Android”. Penelitian ini berfokus pada performa *web service* pada *klien* yang menggunakan bahasa pemrograman *Python, PHP, dan Perl* berbasis android dengan parameter yang digunakan yaitu *response time*, penggunaan CPU dan RAM.

Danandjaya Saputra dan Rizal Fathoni Aji (2018) pernah melakukan penelitian dengan judul “Analisis Perbandingan Performa Web Service REST Menggunakan Framework Laravel, Django dan Ruby On Rails Untuk Akses Data Dengan Aplikasi Mobile (Studi kasus: Portal E-Kampus STT Indonesia Tanjungpinang )”. Penelitian ini menguji metode *web service REST* pada *framework Laravel, Django dan Ruby On Rails* dimana penelitian ini dilakukan untuk mengetahui perbedaan penggunaan CPU (Central Processing Unit),

memory dan kecepatan eksekusi web framework yang digunakan pada Portal E-Kampus.

M Gilvy Langgawan Putra dan M Ihsan Alfani Putera (2019) melakukan analisis perbandingan suatu *web service* dengan judul “Analisis Perbandingan Metode SOAP dan REST yang Digunakan Pada Framework FLASK untuk Membangun Web Service”. Penelitian ini bertujuan untuk mengetahui performa kedua metode tersebut dari segi *request* dan *response*.

Surya Aji Vitrianto (2019) dengan judul “Analisis Perbandingan antara *Framework Yii2* dengan *Framework Laravel* Pada Aplikasi Berbasis Web”. Penelitian ini berfokus pada perbandingan implementasi koneksi database, implementasi CRUD, performa aplikasi, dan fitur unggulan dari masing-masing kedua *framework* tersebut.

Tinjauan pustaka dalam bentuk tabel disajikan pada Tabel 2.1 yang berisi daftar peneliti, judul, metode dan hasil dari peneliti oleh peneliti sebelumnya.

**Tabel 2.1 Tinjauan Pustaka**

Peneliti	Judul	Metode	Hasil
Aleksi Ritsilä	GraphQL: The API Design Revolution	GraphQL & REST	Bahwa pertukaran data antara <i>klien</i> dan server dari metode GraphQL dan REST, GraphQL berhasil diatasi masalah REST dan memenuhi tuntutan yang dimiliki API saat ini.

Tabel 2.1 Tinjauan Pustaka (lanjutan)

Peneliti	Judul	Metode	Hasil
Achmad Fauzi Harismawan, Agi Putra Kharisma dan Tri Afirianto	Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada klien Berbasis Android		Bahasa pemrograman perl mempunyai rata-rata kecepatan eksekusi paling cepat, sedangkan bahasa pemrograman python mempunyai rata-rata penggunaan memory.
Danandjaya Saputra dan Rizal Fathoni Aji	Analisis Perbandingan Performa Web Service REST Menggunakan Framework Laravel, Django dan Ruby On Rails Untuk Akses Data Dengan Aplikasi Mobile (Studi kasus: Portal E-Kampus STT Indonesia Tanjungpinang )	REST	Web Service yang menggunakan framework Django diprediksi memiliki waktu respon yang cepat, penggunaan CPU dan RAM yang paling sedikit.
M Gilvy Langgawan Putra dan M Ihsan Alfani Putera	Analisis Perbandingan Metode SOAP dan REST yang Digunakan Pada Framework FLASK untuk Membangun Web Service	SOAP & REST	Bahwa untuk web service yang berbasis flask dengan menggunakan metode SOAP dan REST API, REST memiliki performa yang lebih bagus dibandingkan dengan SOAP dari segi pengujian request dan respon.
Surya Aji Vitrianto	Analisis Perbandingan antara Framework Yii2 dengan Framework Laravel Pada Aplikasi Berbasis		Implementasi koneksi database Mysql, Implementasi CRUD, performa aplikasi, dan

	Web		Implementasi fitur unggulan
--	-----	--	-----------------------------

## **2.2 Dasar Teori**

### **2.2.1 Analisis**

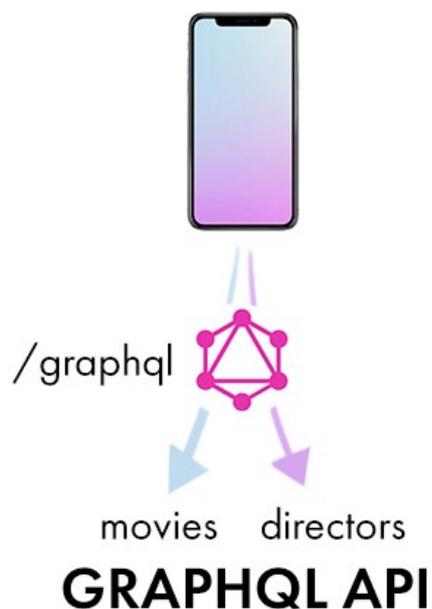
Analisis adalah aktivitas yang memuat sejumlah kegiatan seperti mengurai, membedakan, memilah sesuatu untuk digolongkan dan dikelompokkan kembali menurut kriteria tertentu kemudian dicari kaitannya dan ditafsirkan maknanya. Dalam pengertian yang lain, analisis adalah sikap atau perhatian terhadap sesuatu (benda, fakta, fenomena) sampai mampu menguraikan menjadi bagian-bagian, serta mengenal kaitan antar bagian tersebut dalam keseluruhan. Analisis dapat juga diartikan sebagai kemampuan memecahkan atau menguraikan suatu materi atau informasi menjadi komponen-komponen yang lebih kecil sehingga lebih mudah dipahami namun tetap terpadu (Komaruddin 2001:53).

### **2.2.2 Application Programming Interface (API)**

API adalah singkatan dari antarmuka pemrograman aplikasi. "A" dalam API dapat berupa bagian dari aplikasi, seluruh aplikasi atau seluruh server, atau hampir semua perangkat lunak. API menyediakan antarmuka untuk data yang disimpan sesuai dengan kebutuhan aplikasi yang memungkinkan perangkat lunak untuk berbicara satu sama lain. Ini bisa berupa server situs web yang meminta data dari yang lain server tempat data disimpan dalam database. Dan banyak server di luar sana hari ini memiliki semacam API. (Gazarov 2016).

### 2.2.3 GraphQL

GraphQL adalah suatu metode untuk mengakses API dengan memberikan deskripsi data yang lengkap sesuai dengan permintaan *klien* dan mudah dipahami (<https://graphql.org/>). Ini adalah bahasa yang dapat diajarkan kepada klien perangkat lunak aplikasi. Aplikasi kemudian dapat berkomunikasi dengan layanan backend, juga berbicara GraphQL, untuk meminta data. Bahasa ini dekat dengan JSON dan memiliki operasi untuk membaca (kueri) dan penulisan (mutasi) data. Operasi adalah string di mana GraphQL layanan kemudian dapat merespons dalam format yang diinginkan, seringkali JSON (Buna 2016,1).

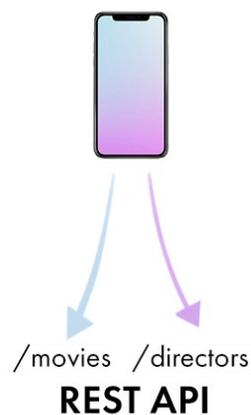


## Gambar 2.2 Akses data pada GraphQL

(Sumber : <https://davidwalsh.name/getting-started-with-graphql>)

### 2.2.4 REST

REST (REpresentational State Transfer) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web atau sistem terdistribusi. RESTful web service atau juga dikenal dengan nama RESTful Web API merupakan sebuah web service yang di implementasikan dengan menggunakan HTTP dengan menggunakan prinsip prinsip REST. Istilah REST diperkenalkan oleh Roy Fielding pada tahun 2000. Arsitektur gaya REST adalah arsitektur *klien server* di mana klien mengirim permintaan ke server, server kemudian memproses permintaan dan mengembalikan tanggapan. Umumnya menggunakan HTTP (Hypertext Transfer Protocol) sebagai protokol untuk komunikasi data (Danandjaya Saputra dan Rizal Fathoni Aji, 2018).



Gambar 2.3 Akses data pada REST

(Sumber : <https://davidwalsh.name/getting-started-with-graphql>)

### 2.2.5 NodeJS

Nodejs dikembangkan dari engine javaScript yang dibuat oleh Google untuk Browser Chrome / Chromium (V8) ditambah dengan libUV serta beberapa pustaka internal lainnya. Dengan menggunakan Nodejs semua pengembangan akan dilakukan dengan javaScript, baik pada sisi klien maupun server. Pengembangan aplikasi dengan menggunakan Node.js dapat dilakukan secara modular yaitu dengan memisahkan berbagai komponen kedalam pustaka (library). Pustaka tersebut dapat dikelola dengan npm yang terdapat di Node.js. Pada dasarnya, Node.JS sebuah runtime environment dan script library. Sebuah runtime environment adalah sebuah software yang berfungsi untuk mengeksekusi, menjalankan dan mengimplementasikan fungsi-fungsi serta cara kerja inti dari suatu bahasa pemrograman. Sedangkan script library adalah kumpulan, kompilasi atau bank data berisi skrip/kode-kode pemrograman. (Equan Pr.2013)

### 2.2.6 Express

*Express* merupakan framework *server side* yang dibangun dalam lingkup *node js*. Framework ini digunakan untuk mengelola *request* dari klien ke server termasuk untuk pengelolaan *routing* dan operasi HTTP (PUT, GET, POST, dll). *Express* dapat dikatakan sebagai suatu *middleware* yang bertanggungjawab dalam

mengelola siklus *request-response* dan menjamin tidak ada suatu request yang tertinggal (dibiarkan menggantung). (Ghifari Munawar)

### 2.2.7 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga bahasa perograman C. (JSON.org)

### 2.2.8 Web Service

*Web service* adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan (W3C, 2004). Menurut Kasman (2016) web service adalah aplikasi yang dibuat agar dapat dipanggil dan diakses oleh aplikasi lain melalui internet dengan menggunakan format pertukaran data sebagai format pengiriman pesan. Pengertian sederhana web service adalah aplikasi yang dibuat 2 agar dapat dipanggil atau diakses oleh aplikasi lain melalui internet atau intranet dengan menggunakan XML sebagai format pengiriman pesan. (Lucky, 2008.)  
perbedaan antara web service dengan website :

a) *Web service*

1. Tidak memiliki interface atau antarmuka.
2. Dibuat untuk berinteraksi langsung dengan aplikasi yang lain baik beda sistem operasi atau jenis platform.

b) *Website*

1. Memiliki interface atau antarmuka untuk user.
2. Dibuat untuk berinteraksi langsung dengan user.

### 2.2.9 Apache Benchmark

Apache Benchmark (AB) adalah alat untuk proses perbandingan Apache HTTP server. AB didesain untuk memberikan gambaran bagaimana performa instalasi Apache. Secara khusus akan menampilkan seberapa banyak request per second yang bisa dilayani oleh instalasi Apache. (Surya Aji Vitrianto, 2019) cara menggunakan Apache Benchmark : `$ ab -c 100 -n 1000 http://contohdomain.com/expamle/`.

Keterangan :

- *-c concurrency* adalah Berapa banyak koneksi bersamaan yang dibuka. Bisa dianalogikan dengan banyaknya pengunjung dalam satu waktu.. Disini terdapat 100 pengunjung secara bersamaan.
- *-n requests* adalah Jumlah permintaan konten yang akan dibuat ke web server. Disini terdapat 1000 request.
- `http://contohdomain.com/expamle/` adalah url yang akan diuji.

Setelah proses benchmark dijalankan maka akan menghasilkan beberapa keluaran diantaranya :

- *Total transferred* : jumlah byte yang diterima dari server.
- *Request per second* : Jumlah permintaan per detik. Nilai ini merupakan hasil dari membagi jumlah permintaan dengan total waktu yang dibutuhkan
- *Time per request* : Rata – rata waktu yang dihabiskan per permintaan.