

## IMPLEMENTASI PERBANDINGAN METODE GRAPHQL DAN REST API PADA TEKNOLOGI NODEJS

### COMPARATIVE IMPLEMENTATION OF GRAPHQL AND REST API METHODS IN NODEJS TECHNOLOGY

Muntahanah<sup>1</sup>, Yulia Darmi<sup>2</sup>, Keken Pinandita<sup>3</sup>

<sup>1,2,3</sup>Universitas Muhammadiyah Bengkulu

<sup>1</sup>muntahanah@umb.ac.id, <sup>2</sup>yuliadarmi10juli@gmail.com, <sup>3</sup>kekenpinandita7@gmail.com

#### ABSTRACT

*This research aims to compare the implementation of the GraphQL and REST API methods on Node.js technology. GraphQL is a flexible and efficient query language, while the REST API has become the de facto standard in web application development. In this research, we conducted an in-depth analysis of both methods by comparing performance, usability, and developer experience. The method used in this research was experimental, where we developed two Node.js applications that implemented GraphQL and REST API methods separately. measure the performance of both methods using metrics such as response time and scalability. The research results show that both methods have different strengths and weaknesses. GraphQL offers greater flexibility in retrieving data by querying only the required parts, reducing repetition and optimizing resource usage. However, using GraphQL requires deeper learning and a better understanding of data structures.*

**Keywords:** Comparison, GraphQL, REST API, Node.js Technology, Flexibility.

#### ABSTRAK

Penelitian ini bertujuan untuk membandingkan implementasi metode GraphQL dan REST API pada teknologi Node.js. GraphQL adalah bahasa query yang fleksibel dan efisien, sedangkan REST API telah menjadi standar de facto dalam pengembangan aplikasi web. Dalam penelitian ini, kami melakukan analisis mendalam terhadap kedua metode dengan membandingkan kinerja, kegunaan, dan pengalaman pengembang. Metode yang digunakan dalam penelitian ini adalah eksperimen, di mana kami mengembangkan dua aplikasi Node.js yang menerapkan metode GraphQL dan REST API secara terpisah. mengukur kinerja kedua metode menggunakan metrik seperti waktu respon dan skalabilitas. Hasil penelitian menunjukkan bahwa kedua metode memiliki kelebihan dan kelemahan yang berbeda. GraphQL menawarkan fleksibilitas yang lebih tinggi dalam mengambil data dengan hanya meminta bagian yang diperlukan, mengurangi pengulangan dan mengoptimalkan penggunaan sumber daya. Namun, penggunaan GraphQL membutuhkan pembelajaran yang lebih mendalam dan pemahaman yang lebih baik tentang struktur data.

**Kata Kunci :** Perbandingan, GraphQL, REST API, Teknologi Node.js, Fleksibilitas.

#### PENDAHULUAN

Dengan perkembangan teknologi yang pesat, banyak metode yang muncul untuk mengkomunikasikan data antara backend dan frontend, seperti metode GraphQL dan REST API. Komunikasi data antara backend dan frontend adalah kegiatan menghubungkan pengguna (*frontend*) ke server. Ini agar data mencapai tujuan yang dituju dan mudah ditemukan dan diakses. Ketika membuat aplikasi web server, salah satu hal yang harus dipertimbangkan adalah pemilihan metode API (*Application Programming Interface*) yang akan digunakan. Kedua metode API yang sering digunakan adalah

REST API dan GraphQL. REST API adalah metode API yang sering digunakan dan memiliki banyak pengguna. Namun, pengembangan aplikasi menggunakan REST API sering memiliki masalah seperti *over-fetching* atau *under-fetching* data. Sementara itu, GraphQL adalah metode API baru yang menawarkan solusi untuk masalah ini.

Meskipun GraphQL menawarkan solusi yang baik untuk masalah yang terjadi pada REST API, banyak pengembang masih memilih untuk menggunakan REST API karena lebih terbiasa dan memiliki banyak sumber belajar dan referensi. Dengan demikian,

penelitian ini sangat penting dilakukan, karena hasil dari penelitian ini dapat memberikan informasi dan solusi untuk membuat aplikasi web server yang lebih baik dan memenuhi kebutuhan pengguna seperti Scalability dan Flexibilitas. Pengguna aplikasi web membutuhkan aplikasi yang dapat bertumbuh dan berkembang sesuai dengan permintaan, dan penelitian ini dapat membantu mereka memahami bagaimana metode tertentu dapat memastikan skalabilitas aplikasi.

Dalam pengembangan aplikasi web, metode pengelolaan dan pertukaran data antara client dan server menjadi sangat penting. Tradisionalnya, Representational State Transfer (REST) API telah menjadi metode yang umum digunakan. Namun, dengan munculnya GraphQL sebagai alternatif yang lebih fleksibel, muncul pertanyaan tentang apakah harus menggunakan GraphQL atau tetap menggunakan REST API. Oleh karena itu, penelitian ini dilakukan untuk membandingkan implementasi GraphQL dan REST API pada teknologi Node.js. Dengan demikian, penelitian ini sangat penting dilakukan, karena hasil dari penelitian ini dapat memberikan informasi dan solusi untuk membuat aplikasi web server yang lebih baik dan memenuhi kebutuhan pengguna seperti Scalability dan Flexibilitas. Pengguna aplikasi web membutuhkan aplikasi yang dapat bertumbuh dan berkembang sesuai dengan permintaan, dan penelitian ini dapat membantu mereka memahami bagaimana metode tertentu dapat memastikan skalabilitas aplikasi.

### **Implementasi**

Implementasi secara umum mengacu pada penerapan atau penerapan dalam Kamus Besar Bahasa Indonesia. Kata "implementasi" biasanya mengacu pada tindakan tertentu yang diambil untuk mencapai suatu tujuan. Implementasi merupakan salah satu upaya yang ingin diwujudkan dalam suatu sistem. Pedoman yang ditentukan sebelumnya diperlukan

karena ide tidak akan pernah menjadi kenyataan tanpa implementasi (susrama, diyasa, amarul dan ade (2021).

Kamus Webster, merumuskan secara pendek bahwa to implement (mengimplementasikan) berarti to provide the means for carrying out (menyediakan sarana untuk melaksanakan sesuatu), to give practical effect to (menimbulkan dampak atau akibat terhadap sesuatu). Pengertian tersebut mempunyai arti bahwa untuk mengimplementasikan sesuatu harus disertai sarana yang mendukung yang nantinya akan menimbulkan dampak atau akibat terhadap sesuatu itu.

### **Perbandingan**

Suatu bentuk perbandingan antara dua hal atau lebih dengan memperhatikan perbedaan atau kesamaan di antara mereka. Dalam bahasa Indonesia, perbandingan bisa dilakukan dengan menggunakan kata-kata seperti lebih, kurang, sama, sebanding, atau tidak sebanding.

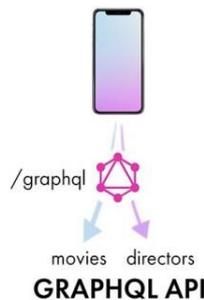
Perbandingan sering digunakan untuk membandingkan sifat, karakteristik, ukuran, atau kualitas dari dua atau lebih benda atau konsep yang berbeda. Dalam bahasa matematika, perbandingan dapat dijelaskan sebagai perbandingan antara dua bilangan atau besaran dalam bentuk pecahan, di mana dua bilangan atau besaran tersebut ditempatkan dalam posisi numerator dan denominator, seperti  $a/b$  atau  $c/d$ .

### **GraphQL**

Menurut A.R (2017), GraphQL adalah bahasa kueri untuk implementasi API berbasis situs web. Tidak seperti REST, GraphQL memiliki keunggulan dalam menyajikan data secara dinamis sesuai dengan aplikasi klien GraphQL menyediakan satu titik akhir untuk mengakses semua sumber data di server. Pada 7 November 2018, proyek GraphQL dipindahkan dari Facebook ke Yayasan GraphQL yang baru didirikan, diselenggarakan oleh Yayasan Linux

nirlaba. Sejak 2012, kebangkitan GraphQL mengikuti garis waktu adopsi seperti yang ditetapkan oleh Lee Byron, pencipta GraphQL Tujuan Byron adalah membuat GraphQL ada di mana-mana di seluruh platform web.

GraphQL menyediakan pendekatan untuk mengembangkan API web dan telah dibandingkan dan dikontraskan dengan REST dan arsitektur layanan web lainnya. Ini memungkinkan klien untuk menentukan struktur data yang diperlukan, dan struktur data yang sama dikembalikan dari server. Hal ini mencegah pengembalian data dalam jumlah yang sangat besar, tetapi dapat menghambat caching web dari hasil kueri. Fleksibilitas dan kekayaan bahasa kueri juga menambah kerumitan yang mungkin tidak berguna untuk API sederhana. Terlepas dari namanya, GraphQL tidak memberikan kekayaan operasi grafik yang mungkin ditemukan dalam bahasa kueri grafik lengkap seperti SPARQL, atau bahkan dalam dialek SQL yang mendukung penutupan transitif. Misalnya, antarmuka GraphQL yang melaporkan induk dari individu tidak dapat mengembalikan, dalam satu kueri.



**Gambar 1. Akses Data Pada Graphql**

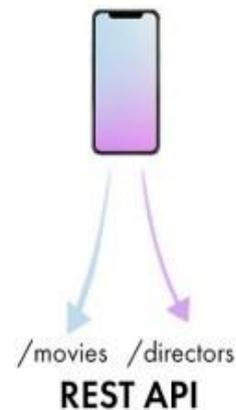
## Rest

Standar arsitektur untuk komunikasi berbasis web yang dikenal sebagai REST (*REpresentational State Transfer*) sering digunakan dalam pengembangan layanan untuk sistem berbasis web atau terdistribusi. Layanan web yang menggunakan HTTP dan prinsip-prinsip REST disebut layanan web RESTful, sering dikenal dengan moniker RESTful

Web API. Roy Fielding menciptakan frase "REST" pada tahun 2000, Putra dan Putera, 2019).

Klien mengirimkan permintaan ke server, yang memprosesnya, dan kemudian mengembalikan respons, yang merupakan arsitektur klien-server yang dikenal sebagai gaya REST. Memanfaatkan protokol HTTP (Hypertext Transfer Protocol) sebagai metode default transmisi data, saputra (2018).

REST secara spesifik merujuk pada suatu koleksi prinsip-prinsip arsitektur jaringan yang menggariskan pendefinisian dan pengalamatan sumber daya. Istilah ini sering digunakan dengan longgar untuk mendeskripsikan semua antarmuka sederhana yang menyampaikan data dalam domain spesifik melalui HTTP tanpa tambahan lapisan pesan seperti SOAP atau pelacakan sesi menggunakan cookie HTTP. Dua pengertian ini dapat menimbulkan konflik dan juga tumpang tindih. Dimungkinkan untuk merancang suatu sistem perangkat lunak besar sesuai dengan gaya arsitektur REST Fielding tanpa menggunakan HTTP dan tanpa berinteraksi dengan WWW.



**Gambar 2. Akses data pada REST**

## Application Programming Interface (Api)

Antarmuka pemrograman aplikasi disebut sebagai API. "A" di API dapat berupa komponen, program lengkap, server, atau hampir semua perangkat lunak. API menawarkan antarmuka untuk menyimpan data sesuai dengan kebutuhan aplikasi, memungkinkan perangkat lunak

untuk berkomunikasi satu sama lain. Mungkin saja data tersebut diminta oleh server situs web dari server lain tempat data tersebut disimpan dalam database. Dan banyak server saat ini memiliki semacam API.

### **Node.js**

Node.js adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. adalah runtime untuk lingkungan JavaScript di luar peramban web di atas mesin JavaScript V8 Node.js dapat dijalankan pada Windows, Mac OS X, dan Linux. Node.js pertama kali diciptakan dan diperkenalkan untuk pengguna pada sistem Linux pada tahun 2009. Node.js dikembangkan oleh Ryan Dahl dan disponsori oleh Joyent.

Dengan menggunakan Node.js 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 (Studiawan, Iqbal dan Husni (2012)).

### **JSON**

JSON adalah sebuah format untuk berbagi data. Seperti dapat kita lihat dari namanya, JSON diturunkan dari bahasa pemrograman JavaScript, akan tetapi format ini tersedia bagi banyak bahasa lain termasuk Python, Ruby, PHP, dan Java. JSON biasanya dilafalkan seperti nama "Jason." JSON menggunakan ekstensi .json saat ia berdiri sendiri. Saat didefinisikan di dalam format file lain (seperti di dalam .html), ia dapat tampil didalam tanda petik sebagai JSON string, atau ia dapat dimasukkan kedalam sebuah variabel. Format ini sangat mudah untuk ditransfer antar server web dengan klien atau browser, (Kurniawan, Humaira da Rozi (2012)).

### **Web Service**

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. 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.

### **METODE**

Implementasikan perbandingan metode GraphQL dan REST API pada teknologi Node.js. Metode ini dipilih karena sesuai dengan tujuan penelitian yaitu untuk membandingkan dan menentukan metode yang lebih baik dalam skalabilitas, dan kemudahan pemeliharaan, Metode ini di lakukan dengan metode deskriptif komparatif dari metode tersebut dapat memberikan informasi dan wawasan tentang suatu fenomena atau masalah, yang kemudian dapat digunakan sebagai dasar untuk membandingkan dengan fenomena atau masalah perbedaan dan persamaan antara dua atau lebih kondisi Suyanto, (2010).

Kebutuhan bahan/data yang digunakan pada penelitian ini yaitu :

### **Komponen Sistem**

Kebutuhan sistem yang digunakan untuk menganalisis perbandingan metode GraphQL dan REST Api sebagai berikut :

1. Dua aplikasi web yang menggunakan metode GraphQL dan REST API.
2. Data simulasi yaitu data User yang terdiri dari Id, Email, Dan Password.

Dalam hal ini akan memberikan kajian terhadap 3 parameter yaitu :

1. Fleksibilitas klien melakukan kustomisasi terhadap kebutuhan data.
2. Fleksibilitas klien melakukan kustomisasi operasi CRUD
3. Response time (ms) Perangkat lunak pendukung lainnya adalah :
  - a) NodeJS v18.13.0
  - b) Npm
  - c) MongoDB
  - d) Mongoose
  - e) Express
  - f) VSCode
  - g) Google chrome

Pada parameter fleksibilitas klien melakukan kustomisasi terhadap kebutuhan kesesuaian data yang ditampilkan metode dengan melakukan pengoperasian operasi read “Adapun sebagai hasil atas pengukuran terhadap baik adalah jika hasilnya metode dapat dibutuhkan klien dan yang disebut kurang baik adalah metode yang menampilkan data yang berlebihan sehingga data yang tidak dibutuhkan klien juga akan di tampilkan, Pada parameter fleksibilitas klien melakukan kustomisasi terhadap kebutuhan data menggunakan tools postman

Pada parameter fleksibilitas klien melakukan kustomisasi operasi CRUD yang akan diukur adalah berapa jumlah URL yang dibutuhkan agar dapat mengoperasikan semua operasi CRUD dengan melakukan langkah-langkah eksperimen terhadap pengoperasian operasi CRUD dengan menggunakan tools post man. Adapun sebagai hasil atas pengukuran terhadap parameter adalah yang disebut baik adalah jika hasilnya metode yang hanya membutuhkan satu jenis URL agar dapat melakukan semua operasi CRUD dan yang disebut kurang baik adalah metode yang membutuhkan URL yang berbeda-beda untuk masing-masing operasi CRUD, Pada parameter fleksibilitas klien melakukan kustomisasi operasi CRUD menggunakan tools postman, memenuhi, (2021).

Pada parameter Peforma response time (ms) melakukan pengujian pada masing–masing metode menggunakan tools postman adalah salah satu alat pengujian yang dapat digunakan untuk menguji performa GraphQL dan REST API pada teknologi Node.js

### **Kebutuhan Keluaran**

Pada parameter fleksibilitas klien melakukan kustomisasi terhadap kebutuhanyang akan diukur adalah data yang ditampilkn dari operasi *read* dalam bentuk data user berupa teks (dalam fomate JSON).

Pada parameter fleksibilitas klien melakukan kustomisasi operasi CRUD yang akan diukur adalah output yang dihasilkan dari masing-masing operasi CRUD dalam bentuk data user berupa teks (dalam fomate JSON).

### **Prosedur dan Pengumpulan Data**

Cara yang dilakukan untuk mengukur setiap parameter pada masing-masing metode yang dikaji adalah sebagai berikut:

1. prosedur membandingkan pada parameter fleksibilitas klien melakukan kustomisasi terhadap kebutuhan data adalah sebagai berikut:
  - a) melakukan pengoperasian operasi read dengan menggunakan *tools postman*
  - b) dan dari hasil operasi read yang menggunakan *postman* akan memberikan output data user berupa JSON
  - c) dalam JSON tersebut dapat diketahui bahwa data yang ditampilkan sesuai atau tidak sesuai dengan apa yang diminta.
2. Prosedur membandingkan pada parameter fleksibilitas klien melakukan kustomisasi operasi CRUD dengan melalui langkah-langkah sebagai berikut:
  1. langkah-langkah metode GraphQL
    - a) inputkan URL pada kolom inputan URL di aplikasi *postman*

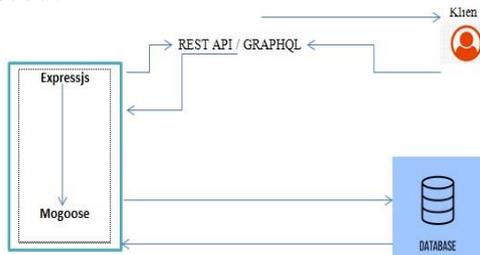


### Analisis Dan Perancangan Sistem

Pada penelitian ini perancangan sistem memiliki dua tahapan yaitu perancangan sistem arsitektur sistem aplikasi web dan rancangan API.

### Perancangan Arsitektur Sistem Aplikasi Web

Menggambarkan cara kerja sistem aplikasi web yang akan dibangun untuk sebagai bahan perbandingan kedua metode tersebut.

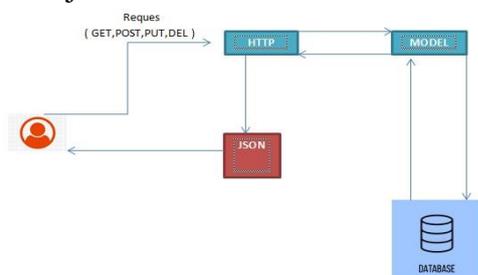


**Gambar 7. Arsitektur Sistem Aplikasi Web (GRAPHQL Dan REST API)**

Pada gambar ini menjelaskan bahwa nantinya klien akan regues ke server dengan menggunakan metode RES API atau GRAPHQL dimana servernya menggunakan *expressJS* yang merupakan salah satu *package* yang dilingkup nodeJS setelah itu akan diteruskan sampai ke *database* (mongodb) dan ketika itu dari *database* memberikan hasil / data yang berdasarkan permintaan klien

### Perancangan Api

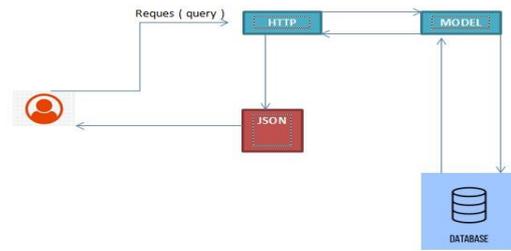
Perancangan API bertujuan untuk menunjukkan alur komunikasi data antara



**Gambar 8. Komunikasi Data REST API**

Gambar 8 menjelaskan bahwa komunikasi data REST API dengan cara klien akan meriques data ke server berdasarkan *methot* yang dibutuhkan kemudian dari server akan mengirim data dari hasil requestnya klien dimana data

yang akan dikirim ke klien atau *reponse* kepada klie berupa JSON.



**Gambar 9. Komunikasi Data GRAPHQL**

Gambar menjelaskan yang dimana terlihat perbedaan komunikasi dara antara REST API dan GRAPHQL yaitu terletak pada *reques* nya klien, pada *reques* klien di GRAPHQL menggunakan *query* yang telah disediakan oleh aplikasi web sedangkan REST API mengguakan *method* yang telah disediakan apikasi web

### HASIL DAN PEMBAHASAN

#### A. Hasil Query Dan Analisis Parameter

##### Fleksibilitas Klien Melakukan

##### Kostumisasi Terhadap Kebutuhan Data

Pada bagian ini hasil query yang telah di implementasikan serta di analisis terhadap parameter fleksibilitas klien melakukan kostomisasi terhadap kebutuhan data dengan cara melakukan pengoprasian operasi *read* dengan menggunakan *tools postman* yang memiliki tolak ukur jika hasilnya metode dapat menampilkan data sesuai dengan yang dibutuhkan klien maka itu disebut baik dan yang didebut kurang baik adalah metode yang menampilkan data yang berlebihan sehingga data yang tidak dibutuhkan klien juga akan di tampilkan. Hasil pengukuran dan pembahasannya sebagai berikut :

**Table 1. Implementasi Dengan Parameter Fleksibilitas Klien GRAPHQL**

	Operasi read	Hasil operasi
Menampilkan data ID dan email	mutation { createUser(email: "kekepinandital7@email.com", password: "password1234") { id email } }	{ "data": { "createUser": { "id": "645d2379d12f1fe542f08e6e", "email": "kekepinandital7@email.com" } } }
Menampilkan id	mutation { createUser(email: "kekepinandital7@email.com", password: "password1234") { id } }	{ "data": { "createUser": { "id": "645d2379d12f1fe542f08e6e", } } }
Menampilkan email	mutation { createUser(email: "kekepinandital7@email.com", password: "password1234") { email } }	{ "data": { "createUser": { "email": "kekepinandital7@email.com" } } }

Melakukan kustomisasi terhadap kebutuhan data pada metode GraphQL terlihat bahwa output yang dihasilkan sesuai dengan apa yang di request klien atau yang dibutuhkan seperti jika klien membutuhkan id dari data user maka GraphQL dapathanya menampilkan id dari data user yang sesuai yang di request klien dan jika klien membutuhkan email dari data user maka GraphQL juga dapat hanya menampilkan email dari data user yang sesuai yang di request klien dan jika klien membutuhkan id dan email GraphQL dapat menampilkan kedua data tersebut yang sesuai di request klien seperti yang telah diimplementasikan di table 1.

B. Hasil Query REST API Dengan Parameter Menampilkan Data User Berdasarkan Kebutuhan Klien

**Table 2. Implementasi Dengan Parameter Fleksibilitas Klien REST API**

	Operasi read	Hasil operasi
Menampilkan data ID dan email		{ "_id": "645d2a12539cf44009599c", "email": "abelbengtulul728@gmail.com", "password": "52b6106d83e5f0d3yQRSVbxPwLPerUc54aKpao7/uFJ.HCYr9aKjo3R2aTa", "user": 0 }
Menampilkan id		{ "_id": "645d2a12539cf44009599c", "email": "abelbengtulul728@gmail.com", "password": "52b6106d83e5f0d3yQRSVbxPwLPerUc54aKpao7/uFJ.HCYr9aKjo3R2aTa", "user": 0 }
Menampilkan email		{ "_id": "645d2a12539cf44009599c", "email": "abelbengtulul728@gmail.com", "password": "52b6106d83e5f0d3yQRSVbxPwLPerUc54aKpao7/uFJ.HCYr9aKjo3R2aTa", "user": 0 }

Tabel 2 adalah implementasi melakukan kustomisasi terhadap kebutuhan data pada metode REST API terlihat bahwa output yang dihasilkan

REST API menampilkan data yang tidak sesuai yang dibutuhkan klien jika hanya membutuhkan id dan email saja REST API tidak bisa menampilkannya seperti yang telah di implementasikan di table 2

Berdasarkan table 1 dan 2 yang telah di implementasikan dengan menggunakan parameter fleksibilitas klien melakukan kostumisasi terhadap kebutuhan data, dapat di simpulkan bahwa metode GraphQL feleksibel dalam mengijinkan klien dalam melakukan kostumisasi terhadap kebutuhan data.sehingga data yang ditampilkan menjadi efisien karena menampilkan data yang sesuai dengan kebutuhan klien.

**Hasil perbandingan dan analisis parameter felsibilitas klien melakukan oprasi crud.**

Pada bagian ini hasil perbandingan yang telah di implemantasikan serta di analisis dan di catat terhadap parameter fleksibilitas klien melakukan kostumisasi oprasi CRUD dengan melakukan langkah-langkah experimen terhadap masing-masing operasi CRUD pada kedua metode dengan mnggunakan *tools posman* yang memiliki tolak ukur jika metode yang hanya membutuhkan satu jenis url agar dapat melakukan operasi CRUD makan disebut baik dan yang di sebut kurang baik adalah metode yang membutuhkan URL yang berbeda – beda untuk masing masing oprasi CRUD.

Hasil pengukuran dan pembahasannya sebagai berikut :

**Tabel 3. Merupakan Hasil Jumlah Url**

	OPERA SI	URL
Graph ql	Operasi CRUD	<a href="http://localhost:3000/graphql">http://localhost:3000/graphql</a>
Rest api	Operasi CRUD	<a href="http://localhost:8080/user/create">http://localhost:8080/user/create</a>
		<a href="http://localhost:8080/user/read">http://localhost:8080/user/read</a>
		<a href="http://localhost:8080/user/update/">http://localhost:8080/user/update/</a>
		<a href="http://localhost:8080/user/delete/">http://localhost:8080/user/delete/</a>

Terlihat pada tabel 3 bahwa metode GRAPHQL hanya membutuhkan satu jenis URL agar dapat menggunakan semua operasi CRUD, sedangkan REST API membutuhkan URL yang berbeda – beda

untuk masing – masing oprasi CRUD, jadi dapat disimpulkan bahwa GRAPHQL felseibilitas klien melakukan kontumisasi terhadap operasi CRUD hanya membutuhkan satu jenis URL agar dapat melakukan semua operasi CRUD sehingga metode graphql menjadi tolak ukur yang baik, sedangkan metode rest api tidak dapat mengijinkan klien untuk kostumisasi terdapa operasi CRUD karena membutuhkan beberapa URL yang membedakan setiap masing-masing operasi CRUD.

### Hasil Pengukuran Dan Analisis Parameter Performa Response Time

Pada bagian ini merupakan hasil pengukuran yang telah di implementasikan serta dianalisis terdagap parameter performa response time maing masing metode dengan menggunakan *tool postman*, yang menjadi tolak ukur jika hasilnya cepatnya dalam menanggapi hasil dari satu reques maka di sebut baik dan yang disebut kutrang baik adalah metode yang paling lambat dalam menanggapi hasil dari suatu *reques*.

Hasil pengukuran dan pembahasannya sebagai berikut :

**Tabel 4. Hasil Pengukuran Performa Response Time Pada Metode REST API**

REQUEST	SOCKET	1.11 MS
	INITIALIZATION	
	DNS LOOKUP	0.42 MS
	TPC HANDSHAKE	0.28 MS
	TRANSFER START	8.9 MS
	DOWNLOAD	1.64 MS
	TOTAL	14.51 MS

**Tabel 5. Hasil Pengukuran Performa Response Time Pada GRAPHQL**

REQUEST	SOCKET	1 MS
	INITIALIZATION	
	DNS LOOKUP	0.26 MS
	TPC HANDSHAKE	0.32 MS
	TRANSFER START	88.3 MS
	DOWNLOAD	2.1 MS
	TOTAL	109.89 MS

Tabel 4 dan 5 adalah data yang dihasilkan dari parameter performa response time dari kedua metode. jumlah hasil terlihat bahwa REST API lebih cepat

dalam menanggapi suatu *request* dibandingkan GRAPHQL yang membutuhkan waktu lebih lama daam menanggapi suatu *request*.

### SIMPULAN

Berdasarkan pada implementasi perbandingan serta pembahasan pada bab-bab sebelumnya memiliki beberapa kesimpulan yaitu :

1. Metode GraphQL sangat fleksibel dan efesien karena dapat menginjinkan klien untuk kustomnisasi terhadap kebutuhan data sehingga nantinya data yang ditampilkan ke klien akan sesuai yang di request atau dibutuhkan klien, sedangkan metode REST API tidak dapat melakukan hal yang dilakukan GraphQL karena REST API tidak dapat mengijinkan klien untuk kustomnisasi terhadap kebutuhan data.
2. Metode GraphQL hanya membutuhkan satu jenis URL agar dapat mengoperasikan semua operasi CRUD, sedangkan metode REST API membutuhkan URL yang berbeda-beda untuk masing-masing operasi CRUD.
3. REST API lebih cepat dalam menanggapi suatu request dibandingkan Graphql yang membutuhkan waktu yang lebih lama dalam menanggapi suatu request seperti yang terlihat pada tabel 4 dan 5

Berdasarkan dari penelitian yang telah dilakukan terhadap metode GraphQL dan metod REST API ada beberapa saran sebagai berikut :

Analisis perbandingan antara metode REST API dan metode GraphQL agar dapat dikembangkan lagi dengan parameter pembanding yang lain. Perlunya pengujian dengan menggunakan jaringan online atau dihosting pada penyedia yang menyediakan hosting.

### DAFTAR PUSTAKA

- G. Susrama, M. Diyasa, G. S. Budiwitjacksono, H. Amarul, and I. Ade. (2021). Comparative Analysis of Rest and GraphQL Technology on

- Nodejs-Based Api Development,1–9.
- M. G. L. Putra and M. I. A. Putera. (2019). Analisis Perbandingan Metode Soap Dan Rest Yang Digunakan Pada Framework Flask Untuk Membangun Web Service. *Jurnal Teknologi Informasi dan Komunikasi*, 14(2):1–7, 2019
- D. Saputra. (2018). Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data Dengan. *Jurnal Bangkit Indonesia*, 7(2):17-28.
- A. R. (2017). GraphQL: T. A. D. Revolution”.P. berfokus perbandingan pertukaran data antara klien dan server Dari, and masing-masing kedua metode Tersebut., “GraphQL: The API Design Revolution. pp. 1–37.
- D. U. Memenuhi et al. (2021). Perbandingan Web Service RESTful dan GraphQL pada.
- B. Suyanto. (2010). Masalah Sosial Anak. *jurnal Kencana Prenada Media Group*. 1–10.
- H. Studiawan, M. C.R., Iqbal, and M. Husni. (2012). Implementasi Klien SIP Berbasis Web. *Jurnal Teknik ITS*, vol 1. 242–245.
- I. Kurniawan, Humaira, and F. Rozi. (2020). REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa *Elektronik Berbasis Android. Jurnal JITS*. 1(4):127–132.