

**IMPLEMENTASI GRAPHQL UNTUK MENGATASI
UNDER-FETCHING PADA SISTEM INFORMASI
PELACAKAN ALUMNI POLITEKNIK NEGERI MALANG**

PROPOSAL SKRIPSI

Oleh:

FANY ERVANSYAH NIM. 1641720080



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2019

**IMPLEMENTASI GRAPHQL UNTUK MENGATASI
UNDER-FETCHING PADA SISTEM INFORMASI
PELACAKAN ALUMNI POLITEKNIK NEGERI MALANG**

PROPOSAL SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:

FANY ERVANSYAH NIM. 1641720080



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2019

IMPLEMENTASI GRAPHQL UNTUK MENGATASI UNDER-FETCHING PADA SISTEM INFORMASI PELACAKAN ALUMNI POLITEKNIK NEGERI MALANG

Imam Fahrur Rozi, ST., MT.
NIP. 198406102008121004

DAFTAR ISI

DAFTAR ISI	iii
1. Judul Laporan Akhir	1
2. Latar Belakang	
3. Rumusan Masalah	3
4. Batasan Masalah	3
5. Tujuan	3
6. Landasan Teori	4
6.1. Konsep Dasar Membangun Aplikasi Berbasis Web	x
6.2. <i>Konsep Dasar Membangun Web Service</i>	x
6.3.	x
6.4.	x
7. Metodologi Penelitian	x
7.1. Deskripsi Sistem	x
7.2. Perancangan Sistem	x
7.3. Kebutuhan Perangkat Keras dan Perangkat Lunak Pendukung	x
7.4. Metode Pengumpulan Data	x
7.5. Uji Coba dan Implementasi	x
8. Relevansi	x
9. Sistematika Penulisan Laporan	x
10. Jadwal Kegiatan	x
DAFTAR PUSTAKA	x

- Problem :

1. Pada beberapa kasus pendistribusian data menggunakan REST API, terdapat masalah dimana dalam 1 kali get data, user harus request lebih dari 1 kali. Contoh: ketika user ingin menampilkan pekerjaan dan lulusan tahun berapa, user harus mengambil data dari /pekerjaan dan juga /mahasiswa. Hal ini disebut underfetching.
2. Pada REST API, hal ini bisa diatasi dengan cukup membuat 1 endpoint yang mengambil semua data yang akan ditampilkan. Hal ini memang menyelesaikan underfetching, namun menimbulkan 1 masalah baru dimana ketika ada perubahan data yang akan ditampilkan, atau penambahan kategori pencarian, bagian frontend dan backend harus mengubah sistem yang ada.

- Mengapa Problem tersebut Krusial

1. Jika masalah underfetching dibiarkan, dari sisi development, bagian frontend harus melakukan request lebih dari 1 kali. setelah itu, frontend juga harus menyeleksi data yang sudah diambil. Hal ini menyebabkan penulisan kode yang panjang dan penggunaan paket data untuk mengambil data yang tidak perlu.
2. pada bagian backend, backend harus menyediakan banyak endpoint untuk masing-masing pencarian dan kombinasinya. Hal itu akan sangat menguras tenaga dan waktu.

- Kondisi Sekarang bagaimana

1. Beberapa Website telah yang telah menerapkan GraphQL bisa dilihat di <https://graphql.org/users/>. Di Indonesia saat ini, website pada umumnya masih menggunakan framework dimana antara backend dan frontend masih menjadi satu. seperti CI, Wordpress dan Laravel. Contoh websitenya adalah: Blossomzones dan klikgalaxy (website peralatan gaming di Malang). Sementara pada beberapa website Industri IT seperti Tokopedia, Shopee dan Bukalapak telah menggunakan component-based untuk frontend seperti React.js dan Vue.js. Untuk tokopedia, telah menggunakan graphql.
2. Artinya, GraphQL saat ini lebih sering diterapkan pada website yang berkecimpung di dunia IT.

- Sejauh mana riset orang lain mengenai masalah ini (jurnal)

1. Saat ini, riset orang lain sudah sampai pada analisis bahasa Graph Query pada graphql
2. berdasarkan jurnal "*Performance of frameworks for declarative data fetching*" yang ditulis oleh Mattias Cederlund dari KTH (ICT) di Swedia tentang perbandingan antara GraphQL buatan Facebook dengan Falcor buatan netflix, didapat data bahwa Falcor lebih lambat dalam *fetching* data, baik pada *single request* ataupun *multiple request*. GraphQL lebih cepat jika

melakukan *fetching* data pada *multiple request*, dan REST API lebih cepat pada *single request*.

3. sejauh ini, belum ada yang membahas bagaimana GraphQL dari segi kenyamanan dalam pengembangan.

1 Judul Skripsi

Implementasi GraphQL untuk mengatasi *underfetching* pada Sistem Informasi Pelacakan Alumni Politeknik Negeri Malang.

2 Latar Belakang

Distribusi data merupakan hal yang sangat penting dalam pengembangan sebuah sistem informasi. Dalam pendistribusian data, beberapa *website* menerapkan metode yang berbeda-beda. Metode pertama adalah menggunakan metode yang menggabungkan antara bagian yang bertugas menampilkan informasi ke pengguna (*Frontend*) dan bagian yang mengatur bagaimana data - data diolah (*Backend*). Sehingga, distribusi data dan tampilan berada dalam 1 sistem. Contoh *website* yang menggunakan metode ini adalah blossomzones.com dan klikgalaxy.com, yaitu *website* yang berfokus pada penjualan peralatan komputer di daerah Malang.

Metode kedua adalah dengan memisahkan bagian *frontend* dan *backend*. Dengan begitu, pada fase pengembangan dan fase perbaikan sistem yang dikembangkan akan lebih mudah. Contoh dari *website* ini adalah Tokopedia, Bukalapak, dan Shopee. Karena bagian *frontend* dan *backend* dipisah, maka diperlukan suatu metode untuk menyediakan layanan distribusi data dari *backend* ke *frontend*.

Saat ini, berdasarkan situs jaxenter.com, pada tahun 2017, metode yang paling banyak digunakan dalam pendistribusian data adalah dengan menggunakan metode REST API. Namun, dalam metode REST API, terdapat suatu masalah yang disebut *underfetching*, dimana jika bagian *frontend* perlu untuk melakukan permintaan data lebih dari 1 kali ke bagian *backend*, maka hal itu akan meningkatkan latensi, yang membuat pengakses *website* harus menunggu lebih lama sebelum data dikirim pada bagian *frontend* dengan sempurna.

Pada tahun 2015, secara publik, Facebook meluncurkan sebuah *query language* yang menjadi metode baru dalam mengatur pendistribusian data. Nama *query language* tersebut adalah GraphQL. Salahsatu masalah yang dapat diatasi oleh GraphQL adalah masalah latensi ketika melakukan *request* lebih dari 1 kali.

Berdasarkan hasil riset oleh Mattias Cederlund tentang performa GraphQL dan Falcor, GraphQL memang dapat mengurangi latensi ketika melakukan *request* lebih dari 1 kali. Sehingga, performa pada *website* meningkat. Namun, ketika hanya melakukan request 1 kali saja, performa REST API masih lebih baik.

Oleh karena itu, kali ini penulis mencoba mengimplementasikan GraphQL untuk mengatasi *underfetching* pada Sistem Informasi Pelacakan Alumni Politeknik Negeri Malang yang akan penulis kembangkan. Dengan adanya GraphQL, diharapkan dapat meningkatkan performa *website* Sistem Informasi Pelacakan Alumni Politeknik Negeri Malang.

3 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang dapat diambil adalah sebagai berikut :

1. Apakah GraphQL dapat mengatasi masalah *underfetching* pada Sistem Informasi Pelacakan Alumni Politeknik Negeri Malang?
2. Apakah performa sistem informasi yang dibangun dengan GraphQL lebih baik daripada dengan REST API?

4 Batasan Masalah

Agar skripsi penulis yang berjudul “Implementasi GraphQL Untuk Mengatasi *Underfetching* pada Sistem Informasi Pelacakan Alumni” dapat berjalan sesuai rencana dan tujuan awal, maka penulis memberikan batasan - batasan masalah sebagai berikut:

- a. Sistem Informasi Pelacakan Alumni hanya menampilkan data hasil Scraping dari LinkedIn. Sehingga, alumni yang tidak memiliki akun LinkedIn dan tidak memasukkan Politeknik Negeri Malang sebagai riwayat pendidikan tidak dapat diambil datanya.
- b. Sistem Informasi Pelacakan Alumni memiliki fitur pencarian alumni dan detail dari alumni.

5 Tujuan

Tujuan dari pengimplementasian GraphQL pada Sistem Informasi Pelacakan Alumni adalah:

1. Untuk mengatasi *underfetching*,
2. Menguji performa dengan membandingkannya dengan REST API,

6 Tinjauan Pustaka

6.1. Penelitian Terdahulu

6.2. Tema

6.3. Studi Kasus

6.4. Algoritma Metode

7 Metodologi Penelitian

7.1. Data

7.2. Metode Pengambilan Data

7.3. Metode Pengolahan Data

7.4. Metode Pengembangan Perangkat Lunak

7.4.1. Analisis Kebutuhan

7.4.2. Perancangan

7.4.3. Implementasi

7.4.4. ... (Diisi sesuai SDLC-nya)

7.5. Metode Pengujian

8 Jadwal Kegiatan

DAFTAR PUSTAKA

- [1] Aji Supriyanto. 2005. *Pengantar Teknologi Informasi*. Salemba Infotek : Jakarta.
- [2] Wahyono, Teguh. 2004. *Sistem Informasi (Konsep Dasar, Analisis, Desain dan Implementasi)*. Graha Ilmu : Yogyakarta.
- [3] Elisabet Yunaeti A., Rita Irviani. 2017. *Pengantar Sistem Informasi*. Penerbit Andi. Yogyakarta.
- Boyd, Mark. 2015. "Pemrograman Javascript : Langkah Awal Belajar Javascript".
<https://www.petanikode.com/javascript-dasar/>
- Enterprise, Jubilee. 2017. "Mengenal Pemrograman ReactJS". Jakarta. PT. Elex Media Komputindo.
- Lutfi, Faisal. 2017. "Mengenal Node.js".
<https://www.codepolitan.com/mengenal-nodejs-5880234fe9ae3>
- Lipsus Internet. 2019. "Indonesia Digital".
<https://websindo.com/indonesia-digital-2019-tinjauan-umum/> (diakses pada 4 November 2019)