

**IMPLEMENTASI GRAPHQL UNTUK MENGATASI
UNDER-FETCHING PADA PENGEMBANGAN 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 PENGEMBANGAN 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**

HALAMAN PENGESAHAN

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

Disusun oleh :

FANY ERVANSYAH NIM. 1641720080

Proposal Skripsi ini telah diuji pada 20....

Disetujui oleh:

- | | | | |
|-----------------|---|---|-------|
| 1. Penguji I | : | _____ | |
| | | NIP. 19..... | |
| 2. Penguji II | : | _____ | |
| | | NIP. 19..... | |
| 3. Penguji III | : | _____ | |
| | | NIP. 19... | |
| 4. Pembimbing I | : | <u>Putra Prima Arhandi, S.T., M.Kom</u> | |
| | | NIP. 198611032014041001 | |

Mengetahui,

Ketua Jurusan
Teknologi Informasi,

Ketua Program Studi
Teknik Informatika,

Rudy Ariyanto, ST., M.Cs
NIP. 19711110 199903 1 002

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

DAFTAR ISI

DAFTAR ISI	3
Judul Skripsi	4
Latar Belakang	4
Rumusan Masalah	5
Batasan Masalah	6
Tujuan	6
Tinjauan Pustaka	6
Penelitian Terdahulu	6
Tema	8
Studi Kasus	8
Algoritma Metode	8
Metodologi Penelitian	8
Data	8
Metode Pengambilan Data	8
Metode Pengolahan Data	8
Metode Pengembangan Perangkat Lunak	8
Membuat User Story	9
Menentukan Durasi Tiap User Story	9
Memberi Prioritas Tiap User Story	9
Implementasi	10
Menangani Permintaan Perubahan/Update	10
Metode Pengujian	10
Jadwal Kegiatan	10
DAFTAR PUSTAKA	12

1 Judul Skripsi

Implementasi GraphQL untuk mengatasi under-fetching pada Pengembangan 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. Berdasarkan analisa penulis menggunakan *tool* Wappalyzer, 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 menyediakan layanan permintaan data ke *backend*. Dengan begitu, bagian *frontend* berada pada sistem yang terpisah dengan bagian *backend*. Sehingga, ketika terdapat suatu sistem informasi baru dan membutuhkan data yang sama dengan sistem informasi sebelumnya, pengembang aplikasi cukup membuat bagian *frontend* dan mengambil data pada *backend* yang sebelumnya dibuat. 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*¹, pada tahun 2017, metode yang paling banyak digunakan dalam pendistribusian data adalah dengan

¹ (2017, August 16). The State of API Integration: SOAP vs. REST ... - JAXenter. Retrieved December 2, 2019, from <https://jaxenter.com/state-of-api-integration-report-136342.html>

menggunakan metode REST API. Namun, dalam metode REST API, terdapat suatu masalah yang disebut *under-fetching*², 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. Selain itu, kompleksitas program juga semakin bertambah karena bagian *frontend* harus menambahkan 1 permintaan data lagi ke bagian *backend*.

Pada tahun 2015, secara publik, Facebook meluncurkan sebuah *query language* yang menjadi metode baru dalam mengatur pendistribusian data. Nama *query language* tersebut adalah GraphQL³. Salah satu masalah yang dapat diatasi oleh GraphQL adalah masalah latensi ketika melakukan *request* lebih dari 1 kali atau yang biasa disebut *under-fetching*⁴.

Oleh karena itu, kali ini penulis mencoba mengimplementasikan GraphQL untuk mengatasi *under-fetching* pada Pengembangan 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. Bagaimana cara mengimplementasikan GraphQL pada pengembangan Sistem Informasi Pelacakan Alumni Politeknik Negeri Malang?
2. Bagaimana GraphQL dapat mengatasi masalah *under-fetching* pada sistem informasi yang dikembangkan?

² (n.d.). GraphQL is the better REST - How to GraphQL. Retrieved December 2, 2019, from <https://www.howtographql.com/basics/1-graphql-is-the-better-rest/>

³ (n.d.). The GraphQL Foundation. Retrieved December 2, 2019, from <https://foundation.graphql.org/>

⁴ (2019, April 16). 3 major REST API problem can solve using GraphQL - Dev.to. Retrieved December 2, 2019, from <https://dev.to/mshossain110/3-major-rest-api-problem-can-solve-using-graphql-1dg2>

4 Batasan Masalah

Agar skripsi penulis yang berjudul “Implementasi GraphQL Untuk Mengatasi *Under-fetching* pada Pengembangan 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. Mengimplementasikan GraphQL pada Sistem Informasi Pelacakan Alumni Politeknik Negeri Malang.
2. Mengatasi masalah *under-fetching* pada sistem yang dikembangkan.

6 Tinjauan Pustaka

6.1. Penelitian Terdahulu

Sejauh ini, penulis menemukan 5 jurnal yang membahas mengenai GraphQL. Jurnal - jurnal tersebut adalah:

1. “*An Initial Analysis of Facebook’s GraphQL Language*” bertujuan untuk memahami bahasa Graph Language milik GraphQL dan menunjukkan bahwa bahasa tersebut memiliki kompleksitas yang

rendah. Jurnal tersebut ditulis oleh Olaf Hartig dari Dept. of Computer and Information Science (IDA), Linköping University, Swedia dan Jorge Pérez dari Department of Computer Science, Universidad de Chile.

2. “*API Design in Distributed Systems: A Comparison between GraphQL and REST*” yang bertujuan untuk membandingkan REST API dan GraphQL. Jurnal ini ditulis oleh Thomas Eizinger, BSc dari University of Applied Sciences Technikum Wien.
3. “*Improving the OEEU’s data-driven technological ecosystem’s interoperability with GraphQL*” yaitu menerapkan GraphQL pada *Observatory of Employment and Employability*, sebuah grup riset lulusan universitas di Spanyol⁵. Hasil dari penerapan GraphQL menunjukkan peningkatan performa, *flexibility*, dan *maintainability*.
4. “*Performance of frameworks for declarative data fetching: An evaluation of Falcor and Relay+GraphQL*” membahas tentang perbandingan performa pengambilan data menggunakan Falcor, dan Relay+GraphQL. Falcor merupakan produk buatan Netflix yang juga dapat membantu pengambilan data dari *backend* ke *frontend*. Sedangkan Relay adalah sebuah *framework* untuk mempermudah pengambilan data dari sisi *client*. Jurnal ini ditulis oleh Mattias Cederlund dari KTH, Skolan För Informations- Och Kommunikationsteknik, Stockholm, Swedia.
5. “*Using GraphQL for Content Delivery in Kentico Cloud*” bertujuan untuk meriset GraphQL sebagai alternatif untuk mengantar konten pada Kentico Cloud, selain menggunakan API yang sudah disediakan oleh Kentico Cloud, yaitu sebuah *CMS online* yang menyediakan konten sebagai *service*. Kentico menyediakan REST API untuk melakukan pengambilan data, namun di sini, dicoba

⁵ (n.d.). Observatorio de Empleabilidad y Empleo Universitarios. Retrieved December 2, 2019, from <https://oeeu.org/>

diterapkan GraphQL. Thesis ini ditulis oleh David Čechák dari Masaryk University, Faculty Of Informatics.

6.2. Tema

Tema yang penulis angkat pada skripsi ini adalah Integrasi Sistem dan Arsitektur.

6.3. Studi Kasus

Studi kasus yang penulis ambil adalah pengembangan sistem informasi pelacakan alumni Politeknik Negeri Malang.

6.4. Algoritma Metode

Cara kerja dari GraphQL adalah dengan mendeklarasikan skema di *backend*, kemudian dari skema tersebut, bagian frontend dapat mengetahui data apa yang dapat diambil dan memilih sesuai kebutuhan.

7 Metodologi Penelitian

7.1. Data

Data yang digunakan adalah data para alumni mahasiswa Politeknik Negeri Malang.

7.2. Metode Pengambilan Data

Pengambilan data dilakukan dengan cara *scraping* pada situs LinkedIn, halaman Politeknik Negeri Malang.

7.3. Metode Pengolahan Data

Pengolahan data dilakukan dengan menggunakan framework Node.js, dan menggunakan mongoDB sebagai penyimpanan data (*database*).

7.4. Metode Pengembangan Perangkat Lunak

Metode pengembangan aplikasi yang penulis gunakan adalah metode Agile dengan framework Kanban. Alasan penulis memilih metode

Kanban karena metode ini termasuk pada metode Agile yang memberikan kebebasan pada pengembang untuk melakukan penambahan fitur (*incremental*)⁶. Selain itu, kanban memiliki WIP (*Work-In-Progress*) yang memberi batasan berapa banyak tugas yang bisa diselesaikan⁷. Sehingga, pengembang dapat fokus pada beberapa tugas terlebih dahulu.

7.4.1. Membuat *User Story*

User story merupakan suatu kasus berupa cerita ketika pengguna sedang mengakses website yang dibuat. Pengembang memperkirakan kira-kira apa saja yang ingin dilakukan oleh seorang pengguna.

User story yang dibuat diharapkan dapat menimbulkan masalah under-fetching. seperti dalam 1 halaman, website akan menampilkan data-data mahasiswa sekaligus jumlah data mahasiswa yang ada.

7.4.2. Menentukan Durasi Tiap *User Story*

Penentuan durasi dapat diperkirakan dengan besar/banyaknya fitur yang akan dikembangkan pada 1 *user story*.

7.4.3. Memberi Prioritas Tiap *User Story*

Penentuan prioritas pada *user story* dilakukan dengan memperkirakan pengaruh suatu user story dengan fitur-fitur lain, apakah *user story* itu merupakan syarat untuk mengerjakan *user story* lainnya atau tidak.

⁶ (n.d.). How is it different? - What is Agile?. Retrieved December 2, 2019, from http://www.agilenutshell.com/how_is_it_different

⁷ (2018, July 19). How To Use Kanban To Become Insanely Productive: A Short Retrieved December 2, 2019, from <https://www.forbes.com/sites/bryancollinseurope/2018/07/19/how-to-use-kanban-to-become-insanely-productive-a-short-guide/>

7.4.4. Implementasi

Langkah implementasi dilakukan dengan mengikuti *kanban board*. *Kanban Board* adalah suatu papan dimana kita menaruh *user story - user story* yang akan kita kerjakan. Batasan *Doing/WIP* (*Work-In-Progress*) adalah 3 fitur.

7.4.5. Menangani Permintaan Perubahan/*Update*

Permintaan perubahan dilakukan ketika ada perubahan pada sistem. Dalam Agile, ada 2 langkah dalam menangani perubahan. Langkah pertama adalah menolak perubahan, atau menerima perubahan dengan meminta tambahan waktu⁸.

7.5. Metode Pengujian

Metode pengujian akan dilakukan dengan cara membuat suatu sistem informasi pelacakan alumni Politeknik Negeri Malang dengan menggunakan metode pendistribusian data REST API.

Pengujian akan dilakukan dengan membandingkan performa (kecepatan akses website), dan membandingkan kode yang digunakan untuk menyelesaikan 1 masalah yang sama, yaitu *under-fetching*.


8 Jadwal Kegiatan

Untuk jadwal kegiatan selengkapnya, bisa dilihat pada alamat yang sudah disediakan berikut:

<https://docs.google.com/spreadsheets/d/1u9IAJGN0iW9O4uOi5nfhe1jiLGZQ9LqKHu4-nE0Sdk0/edit?usp=sharing>

⁸ (n.d.). How does it work? - What is Agile?. Retrieved December 2, 2019, from http://www.agilenutshell.com/how_does_it_work

Nomor	Pekerjaan Pengembangan Perangkat Lunak	Jadwal Per Minggu															
		Januari				Februari				Maret				April			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Membuat <i>User Story</i>																
2	Menentukan Durasi tiap <i>User Story</i>																
3	Memberi prioritas tiap <i>User Story</i>																
4	Implementasi																
4	Penerimaan Perubahan/Update																

 = **Relatif** (Jika tidak ada perubahan, maka tidak akan dilakukan)

 = **Pekerjaan** (Berkaitan dengan Kanban Board)

DAFTAR PUSTAKA

1. Hartig, O., & Pérez, J. (n.d.). *An Initial Analysis of Facebook's GraphQL Language*.
2. Eizinger, T. (2017). *Api Design in Distributed Systems: A Comparison between GraphQL and Rest*.
3. Vazquez, A., Cruz, J., & García, F. J. (2017). *Improving the Oeeu's data-driven technological ecosystem's interoperability with GraphQL*.
4. Cederlund, M. (2016). *Performance of frameworks for declarative data fetching An evaluation of Falcor and Relay GraphQL*.
5. Čechák, D. (2017). *Using GraphQL for Content Delivery in Kentico Cloud*.