**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN EVALUASI KECEPATAN**

**PENGAMBILAN DATA PADA API GRAPHQL VS. REST**

*Laporan ini disusun untuk memenuhi tugas 2 mata kuliah Pengembangan Web (Teori)*



Disusun oleh kelompok B4:

Asri Husnul Rosadi 221524035

Faris Abulkhoir 221524040

Mahardika Pratama 221524044

Muhamad Fahri Yuwan 221524047

Najib Alimudin Fajri 221524053

Septyana Agustina 221524058

**Sarah 221524059**

Dosen Pengampu:

Joe Lian Min, M.Eng.

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2024**

# **DAFTAR ISI**

[DAFTAR ISI i](#_Toc175471873)

[A. IDENTIFIKASI PROBLEM 1](#_Toc175471874)

[B. DESKRIPSI PROBLEM 1](#_Toc175471875)

[C. METODOLOGI EKSPERIMEN 1](#_Toc175471876)

[D. PELAKSANAAN EKSPERIMEN 2](#_Toc175471877)

[E. ANALISIS HASIL EKSPERIMEN 2](#_Toc175471878)

# **IDENTIFIKASI PROBLEM**

Pengembang seringkali menghadapi tantangan dalam memilih API terbaik antara REST dan GraphQL untuk menangani pengambilan data, terutama ketika aplikasi melibatkan data skala besar, kompleksitas relasi antar data, serta kebutuhan performa yang optimal. Keduanya memiliki kelebihan dan kekurangan dalam skenario yang berbeda, namun pilihan yang salah bisa berdampak pada kecepatan, efisiensi, dan pengalaman pengguna.

# **DESKRIPSI PROBLEM**

REST API secara tradisional digunakan untuk pengambilan data berbasis resource dengan endpoint spesifik, namun sering kali terjadi over-fetching (mengambil terlalu banyak data yang tidak diperlukan) atau under-fetching (mengambil terlalu sedikit data, sehingga memerlukan request tambahan). Di sisi lain, GraphQL memungkinkan pengambilan data yang lebih fleksibel dan spesifik dalam satu permintaan, namun bisa rentan terhadap query yang terlalu kompleks. Tantangannya adalah memahami dalam skenario apa masing-masing pendekatan lebih cocok digunakan, terutama dalam konteks skala data dan relasi antar entitas.

# **METODOLOGI EKSPERIMEN**

1. **Desain Eksperimen**

Eksperimen ini membandingkan performa pengambilan data antara REST API dan GraphQL API dalam tiga skenario:

* Pengambilan data kecil
* Pengambilan data besar
* Pengambilan data dengan relasi kompleks

1. **Variabel Eksperimen**

* Variabel bebas: Jenis API yang digunakan (REST vs. GraphQL).
* Variabel terikat: Waktu respon pengambilan data (ms), ukuran data yang diambil (KB), kompleksitas query.
* Variabel kontrol: Ukuran dataset, struktur data, koneksi internet, perangkat keras.

1. **Prosedur Eksperimen**
2. Membangun REST API dan GraphQL API menggunakan Node.js dengan dataset yang sama.
3. Melakukan pengambilan data pada tiga skenario: data kecil, data besar, dan data dengan relasi.
4. Mengukur waktu respon API pada setiap percobaan.
5. Mengumpulkan hasil untuk dibandingkan.
6. **Alat dan Bahan**

* Alat: Node.js, Express.js, GraphQL, database dummy.
* Bahan: Komputer, koneksi internet stabil.

1. **Pengukuran dan Pengamatan**

* Waktu respon dalam milidetik (ms).
* Ukuran data yang dikirim (KB).
* Perbandingan jumlah request yang diperlukan untuk query yang sama.

1. **Analisis Data**

Data dikumpulkan berdasarkan waktu respon API untuk REST dan GraphQL pada setiap skenario. Analisis dilakukan dengan membandingkan rata-rata waktu respon dan jumlah request antara REST dan GraphQL.

# **PELAKSANAAN EKSPERIMEN**

1. Membangun REST API dan GraphQL API dengan dataset pengguna yang memiliki relasi ke postingan.
2. Melakukan pengujian pada tiga skenario:
3. Pengambilan data kecil: Mengambil informasi dari 10 pengguna.
4. Pengambilan data besar: Mengambil data dari 1.000 pengguna.
5. Pengambilan data relasional: Mengambil data pengguna beserta postingan yang terkait.
6. Mengukur waktu respon untuk setiap pengujian, serta mencatat ukuran data dan jumlah request.
7. Mencatat hasil pengujian ke dalam tabel..

# **ANALISIS HASIL EKSPERIMEN**

1. Hasil Pengamatan (Tabel)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | API Type | Rata-rata Waktu Respon (ms) | Ukuran Data | Jumlah Request |
| Pengambilan Data Kecil | REST | 124 | 10 | 5 |
| GraphQL | 148 | 10 | 5 |
| Pengambilan Data Besar | REST | 316 | 1000 | 3 |
| GraphQL | 283 | 1000 | 3 |
| Pengambilan Data Relasi Kompleks | REST | 447 | 100 | 4 |
| GraphQL | 351 | 100 | 5 |

1. Pros dan Cons

* REST API:
* Kelebihan: Mudah digunakan untuk pengambilan data sederhana, stabil, mudah di-cache.
* Kekurangan: Menghasilkan over-fetching pada query kompleks, memerlukan banyak request untuk data relasional.
* GraphQL API:
* Kelebihan: Efisien dalam menangani query kompleks dengan relasi data, hanya mengambil data yang dibutuhkan.
* Kekurangan: Memerlukan query yang lebih rumit untuk pengambilan data sederhana, lebih rentan terhadap query besar yang memperlambat performa.

1. Kesimpulan dan Saran

Kesimpulan dari eksperimen ini menunjukkan bahwa REST API lebih cocok digunakan untuk aplikasi dengan pengambilan data sederhana atau resource-based, di mana setiap endpoint memiliki struktur yang jelas dan tidak memerlukan banyak relasi antar data. Sementara itu, GraphQL API lebih efektif untuk menangani query data yang kompleks dan relasional, serta memungkinkan fleksibilitas dalam mengambil hanya data yang diperlukan. Rekomendasinya, REST API sebaiknya digunakan untuk aplikasi yang stabil dan membutuhkan pengambilan data yang relatif sederhana, sedangkan GraphQL cocok untuk aplikasi dengan kebutuhan query dinamis dan data yang lebih kompleks.