



Program Studi Teknik Informatika  
Institut Teknologi Sumatera

## **PROPOSAL PENELITIAN AWAL**

# **PERBANDINGAN EFISIENSI REST API DAN GRAPHQL PADA SISTEM PENJADWALAN PERKULIAHAN BERBASIS CLOUD-NATIVE DENGAN DOCKER, KUBERNETES, DAN LOAD BALANCING**

Tugas Mata Kuliah Metodologi Penelitian (IF25-41029)  
Program Studi Teknik Informatika  
Institut Teknologi Sumatera



# **ITERA**

Oleh:  
**Reyhan Capri Moraga**  
NIM: 123140022

PROGRAM STUDI TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI SUMATERA  
2025

# 1 Pendahuluan

## 1.1 Latar Belakang

Transformasi digital di sektor pendidikan tinggi telah mendorong peningkatan ekspektasi terhadap ketersediaan dan performa layanan akademik. Sistem informasi krusial, seperti sistem penjadwalan perkuliahan, dituntut untuk tidak hanya fungsional tetapi juga harus skalabel dan responsif, terutama saat menghadapi lonjakan lalu lintas data (traffic) secara tiba-tiba, seperti pada masa pengisian Rencana Studi Mahasiswa (RSM)[1].

Untuk menjawab tantangan ini, arsitektur aplikasi modern telah beralih dari sistem monolitik menuju arsitektur cloud-native yang memanfaatkan teknologi seperti kontainerisasi (Docker) dan orkestrasi (Kubernetes) untuk mencapai skalabilitas elastis dan ketahanan sistem (resilience)[2].

Inti dari sistem terdistribusi ini adalah Application Programming Interface (API), yang berfungsi sebagai jembatan komunikasi antar layanan. Selama bertahun-tahun, REST (Representational State Transfer) telah menjadi standar de-facto dalam perancangan API karena kesederhanaan dan adopsinya yang luas[3]. Namun, seiring dengan meningkatnya kompleksitas relasi data pada aplikasi modern—seperti sistem penjadwalan yang melibatkan entitas mahasiswa, dosen, mata kuliah, dan ruangan—kelemahan REST mulai terlihat, terutama masalah over-fetching (mengambil data berlebih) dan under-fetching (memerlukan banyak request untuk mendapatkan data lengkap)[4].

Sebagai alternatif, GraphQL muncul sebagai query language untuk API yang menawarkan fleksibilitas tinggi, memungkinkan klien untuk meminta data yang spesifik hanya dengan satu request[5]. Sejumlah penelitian telah berupaya membandingkan efisiensi kedua arsitektur ini. Banyak studi awal, seperti yang dilakukan oleh Hartanto[3] dan Cerny and Donahoo[4], berfokus pada perbandingan performa dalam lingkungan aplikasi monolitik atau lingkungan pengujian lokal.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana rancangan dan implementasi prototipe backend sistem penjadwal perkuliahan yang mengadopsi dua arsitektur API (REST dan GraphQL) di atas platform Docker dan Kubernetes?
2. Bagaimanakah perbandingan kinerja—ditinjau dari throughput, latency, ukuran payload, dan penggunaan resource (CPU/Memori)—antara REST API dan GraphQL dalam skenario akses data standar?
3. Bagaimana pengaruh skenario beban tinggi (high load) dan mekanisme load balancing pada platform Kubernetes terhadap performa dan skalabilitas masing-masing arsitektur API?

## 1.3 Tujuan Penelitian

1. Merancang dan mengimplementasikan prototipe backend sistem penjadwal perkuliahan dengan dua arsitektur API yang berbeda (REST dan GraphQL) menggunakan teknologi Docker dan Kubernetes dalam waktu 3 bulan.
2. Menganalisis dan membandingkan secara kuantitatif metrik efisiensi kedua API dengan target minimal 1000 request per second, latency di bawah 100ms, dan utilisasi CPU di bawah 80%.
3. Mengevaluasi dan mendokumentasikan performa kedua arsitektur API dalam skenario beban tinggi (10.000 concurrent users) dengan target availability 99.9%.

## 1.4 Batasan Masalah

- Penelitian berfokus pada perbandingan performa API di sisi backend. Aspek frontend (UI/UX) dari aplikasi penjadwal tidak akan dikembangkan atau dievaluasi.
- Logika bisnis atau algoritma untuk generating jadwal perkuliahan berada di luar cakupan penelitian. Sistem menggunakan data jadwal yang sudah ada.
- Pengujian dilakukan dalam lingkungan cluster Kubernetes yang terkontrol dengan 3-5 node.
- Aspek keamanan API seperti otentikasi dan otorisasi tidak menjadi fokus utama pengukuran efisiensi.

## 1.5 Kontribusi Penelitian

1. Kontribusi Akademik: Menyajikan data empiris perbandingan performa REST API vs. GraphQL dalam konteks arsitektur cloud-native, sebuah area yang masih terbatas di literatur saat ini.
2. Kontribusi Praktis: Memberikan rekomendasi berbasis data bagi pengembang dalam memilih arsitektur API untuk aplikasi berskala besar dengan karakteristik relasi data kompleks.
3. Artefak: Menghasilkan prototipe sistem dan dataset pengujian yang dapat direplikasi untuk penelitian lanjutan tentang optimasi API di lingkungan cloud-native.

## References

- [1] Y. Pratama and A. B. Santoso, "Analisis kinerja sistem informasi akademik berbasis cloud computing saat terjadi lonjakan trafik," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 8, no. 3, pp. 501–510, 2021.
- [2] L. Wang and J. Li, "Performance comparison of graphql and rest apis in microservices architecture," in *2022 IEEE International Conference on Web Services (ICWS)*, Shanghai, China, 2022, pp. 310–318.
- [3] D. Hartanto, "Analisis performa graphql dan restful api pada aplikasi e-commerce monolitik," *Jurnal Teknika: Jurnal Sains dan Teknologi*, vol. 17, no. 1, pp. 45–54, 2021.
- [4] T. Cerny and M. Donahoo, "GraphQL vs. rest: A performance comparison," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 10, no. 1, pp. 1–15, 2021.
- [5] A. Suryadi, "Payload size optimization using graphql in mobile applications," *International Journal of Computer Science and Network Security*, vol. 22, no. 4, pp. 112–119, 2022.

## A Worksheet Tugas Proposal Awal

### A.1 Catatan Review Platform AI

### A.2 Catatan Review Teman Sejawat

### A.3 Changelog dan Perbaikan

### A.4 Refleksi