

ABSTRAK

Aplikasi Sidang Online Fakultas Rekayasa Industri (SOFI) bermigrasi dari arsitektur *monolithic* ke *microservices* untuk memulihkan performa dan skalabilitas yang menurun karena lonjakan pengguna. Tantangan berikutnya ialah memilih protokol komunikasi paling efisien. Penelitian ini mengkuantifikasi kelemahan REST pada modul pendaftaran sidang, menilai sejauh mana GraphQL menanganinya, dan merumuskan pedoman strategis pemilihan API bagi SOFI. Metodologi penelitian menggunakan eksperimen terkontrol dalam kerangka rekayasa perangkat lunak empiris. Dua layanan identik, yaitu REST dan GraphQL dibangun menggunakan bahasa pemrograman Go, dikemas dalam Docker, dan diuji beban menggunakan Apache JMeter dengan skenario 10, 50, dan 100 user virtual. Evaluasi kinerja dilakukan terhadap lima parameter utama, yaitu *response time*, *throughput*, *error rate*, *CPU usage*, dan *memory usage*. Hasil eksperimen menunjukkan adanya *trade-off* yang jelas antara latensi dan efisiensi *resource*. Kedua arsitektur menunjukkan kapasitas pemrosesan (*throughput*) dan keandalan (*error rate* 0.00%) yang setara di semua skenario. API REST unggul dalam hal latensi untuk permintaan data tunggal yang sederhana, dengan *response time* minimal 6 ms. Sebaliknya, GraphQL menunjukkan efisiensi *resource* yang jauh lebih superior. Penggunaan CPU GraphQL sangat stabil (di bawah 0.4%) dan konsumsi memory-nya secara konsisten hanya sekitar 265 MB, hampir setengah dari yang dibutuhkan oleh REST yang mencapai 452 MB. Implikasi dari temuan ini, yaitu REST tetap ideal untuk modul layanan ringan yang memprioritaskan latensi minimal, sedangkan GraphQL optimal untuk layanan dengan agregasi data kompleks atau yang diimplementasikan pada sumber daya yang terbatas.

Kata Kunci: REST, GraphQL, Microservices, Evaluasi Performa, Efisiensi Resource, Response Time.