

## Analisis Performansi Layanan Web Menggunakan Arsitektur Microservice dan Monolitik

Haris Hamdani Latif <sup>1</sup>, Siti Amatullah Karimah <sup>2</sup>, Sidik Prabowo <sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>harishl@students.telkomuniversity.ac.id, <sup>2</sup>karimahsiti@telkomuniversity.ac.id,

<sup>3</sup>pakwowo@telkomuniversity.ac.id

---

### Abstrak

Teknologi *Container-based virtualization* saat ini menjadi populer digunakan di seluruh platform *Cloud* dan diprediksi akan terus berlanjut untuk beberapa tahun ke depan, penggunaan teknologi *Container* ini akan memudahkan serta dapat menghemat *resource* yang digunakan untuk layanan. Ditambah lagi dengan perkembangan arsitektur web saat ini yang semakin berkembang dan digunakan untuk komersial diantaranya yaitu *Microservice* dan *Monolitik*. Arsitektur *Microservice* ini membagi layanannya menjadi bagian-bagian yang lebih kecil berdasarkan fungsionalitas. Sedangkan Arsitektur *Monolitik* ini disebut sebagai arsitektur konvensional karena di dalamnya layanan menjadi satu kesatuan utuh. Untuk itu dilakukan skenario pengujian untuk mengetahui performansi dari kedua arsitektur web. Pada penelitian ini dilakukan *load testing* dengan jumlah *request* 50, 100, 500, dan 1000 pada *Microservice* dan *Monolitik* untuk menunjukkan skalabilitas. Didapatkan hasil bahwa layanan *Monolitik* lebih unggul dengan rata – rata penggunaan *CPU* pada *AWS* sebesar 83% sedangkan *Microservice* di angka 99%. *CPU* pada *Docker Container Monolitik* sebesar 92% sedangkan *Microservice* 30% untuk setiap *service*. Untuk *Memory Usage* layanan *Microservice* mendapatkan hasil rata – rata 14% sedangkan untuk layanan *Monolitik* 12%. *Response Time* di dapatkan waktu 1497,31 ms untuk *Microservice* dan 89,02 ms untuk *Monolitik*. Pada pengujian availabilitas dengan melakukan *terminate/stop* pada *service* di layanan *Microservice* lalu diaktifkan kembali dan membutuhkan waktu 2 detik, sedangkan pada layanan *Monolitik* membutuhkan waktu 3 detik untuk mengembalikan *service*. Pada saat *service* di matikan layanan *Microservice* masih bisa berjalan dengan normal, hanya pada *service* yang mati saja yang akan mengalami gangguan, hal ini berbanding terbalik pada layanan *Monolitik* yang akan mati total saat *service* dimatikan.

**Kata kunci :** Microservice, Monolitik, Container, Virtualisasi, Skalabilitas, Availabilitas

---