

ABSTRAK

Kubernetes adalah orkestrasi kontainer yang dapat menyediakan *service* pada server dan penskalaan (*scaling*) aplikasi terkontainer di seluruh *cluster*. Service Kubernetes dapat diakses atau diekspos oleh klien dengan IP *Cluster*, *Node Port*, dan *Load Balancer*. Namun pada IP cluster hanya dapat diakses dari dalam cluster, pada *Node Port* dan *Load Balancer* klien harus mengetahui semua IP *pod* dan IP *Load Balancer* yang diekspos ke *cloud* publik. Oleh karena itu, *Ingress* dapat digunakan klien untuk mengakses *service* dari luar *cluster* Kubernetes hanya menggunakan domain atau nama *ingress*. *Ingress* merupakan sekumpulan aturan untuk koneksi masuk suatu layanan (*service*) tertentu. Agar *ingress controller* berfungsi, *ingress controller* harus dijalankan di dalam cluster agar klien dapat mengakses *service* yang terdapat dalam kontainer di cluster Kubernetes. Untuk memiliki *controller ingress*, user harus menggunakan *controller* yang tersedia. Setiap *ingress controller* memiliki konfigurasi yang berbeda-beda yang mempengaruhi kinerja layanan. Konfigurasi tersebut menyebabkan setiap *ingress controller* memiliki keunggulan performansi yang berbeda juga pada setiap variasi jumlah *request* klien. Pada penelitian ini akan membandingkan kinerja dari *ingress controller* menggunakan Kong dan Istio. Penelitian ini menggunakan skenario pengujian peningkatan jumlah klien yaitu 50, 150, 250, 350, 450, 550, 650, dan 750 klien, serta skenario pengujian dengan Istio *ingress controller* dan Kong *ingress controller*. Parameter yang dibandingkan yaitu *delay*, *throughput*, CPU *usage*, dan *memory usage*. Berdasarkan hasil penelitian, didapatkan bahwa Istio *ingress controller* unggul dalam parameter *throughput*, CPU, dan *memory* dengan masing-masing hasilnya adalah 15206,631 Kbps, 10,13 %, serta 714,733 MB. Sedangkan Kong *ingress controller* hanya unggul pada parameter *delay* dengan hasil 0,481 ms.

Kata Kunci: Ingress Controller, Istio, Kong, Kontainer, Kubernetes.