

ABSTRAK

Container adalah perkembangan teknologi virtualisasi yang banyak digunakan untuk mengembangkan dan mengirimkan perangkat lunak untuk layanan *cloud computing*. *Container* memungkinkan untuk berbagi sumber daya langsung di *host* secara efisien diantara *platform* penyewa *cloud*, sehingga *container* lebih ringan dan lebih cepat daripada mesin virtual. Dalam arsitektur *container*, *container runtime* bertanggung jawab untuk memuat *image container* dari repositori, memantau sumber daya sistem lokal, mengisolasi sumber daya sistem untuk penggunaan *container*, dan mengelola siklus hidup *container*. *Host* dengan jumlah *container* yang banyak menyebabkan penurunan performansi yang signifikan, baik *container* yang dipasangkan pada *platform* penyewa *cloud* maupun pada *local host*. Masalah ini muncul karena *container runtime* bertanggung jawab menjalankan *container* pada *host* sistem operasi. Pemilihan *container runtime* menjadi sangat krusial untuk mengimbangi penambahan jumlah *container* yang digunakan. Pada penelitian ini berfokus menganalisis performansi dari tiga *container runtime* yaitu *Containerd*, CRI-O, dan Kata *Containers* pada orkestrasi Kubernetes. Skenario penelitian ini menggunakan skalabilitas dengan jumlah *container* yang berbeda yaitu 10, 20, dan 40 *container*. Parameter yang akan dianalisis yaitu performansi dari CPU, memory, *throughput*, dan *latency*. Berdasarkan hasil pengujian diketahui bahwa *container runtime* *Containerd* memiliki kinerja yang unggul dan optimal berdasarkan hasil dari parameter *throughput*, *latency*, dan *memory* dengan hasil masing-masing parameter yaitu 2301,75 MB/s, 0,359 ms, dan 13001,8 MB/s.

Kata Kunci: *Container runtime*, *Containerd*, CRI-O, Kata *Containers*, Kubernetes.