

1. Pendahuluan

Latar Belakang

Container dengan cepat mengganti Mesin Virtual (*VMs*) sebagai instance komputasi dalam penyebaran berbasis *cloud*. Biaya *overhead* yang jauh lebih rendah dibandingkan dengan *VMs* telah sering dikutip sebagai salah satu alasan kenapa *Container* lebih bagus dari pada Mesin Virtual (*VMs*) [1]. *Container* adalah cara atau metode menjalankan beberapa aplikasi perangkat lunak pada mesin yang sama. Masing-masing dijalankan dalam lingkungan terisolasi yang disebut *Container*. *Container* adalah lingkungan tertutup untuk perangkat lunak. Yang memungkinkan membungkus semua file dan *library* agar aplikasi berfungsi dengan benar[2].

Salah satu virtualisasi berbasis *Container* yang paling banyak digunakan saat ini adalah *Docker* [3]. *Docker* adalah salah satu alat yang berguna untuk menjalankan *Container*. Namun, mengelola *Container* untuk menciptakan banyak layanan dan menangani *user* yang banyak adalah tugas yang besar bagi *Docker* [4]. Proses *Scaling* adalah masalah utama dalam kasus ini. *Resource* yang tidak memadai yang tidak mampu mengatasi perubahan intensitas beban kerja dari waktu ke waktu, dalam hal ini aplikasi mengalami kinerja yang rendah atau permintaan pengguna yang terlalu banyak, dalam hal ini pemanfaatan sumber daya yang dialokasikan ke *Container* rendah. Oleh karena itu, penskalaan pada saat runtime diperlukan [5].

Dalam mewujudkan hal tersebut maka dibutuhkan suatu *Container Orchestration* yang merupakan sebuah sistem open source berfungsi sebagai *Controller* yang akan mengatur beberapa *Cluster Container* sehingga akan lebih memudahkan manajemen *Container*[4]. Salah satunya adalah penskalaan yang secara eksklusif didasarkan pada metrik infrastruktur level, seperti pemanfaatan *CPU Utilization*.

Oleh sebab itu penelitian ini dilakukan untuk membandingkan *Container Orchestration* manakah yang lebih baik dalam scaling menggunakan metode *Horizontal pod autoscaler*. *Container Orchestration* yang akan dibandingkan adalah *Kubernetes* dan *Docker Swarm* karena kedua *Container Orchestration* ini sangat populer di gunakan pada perusahaan besar seperti Google dan Ruang Guru[6], yang menggunakan *Kubernetes* serta *hepsiburada* dan *PinCAMP* [7], yang menggunakan *Docker Swarm*. Penelitian

Topik dan Batasannya

Rumusan masalah yang diangkat di dalam penelitian ini adalah *Container Orchestration* yang memiliki muatan yang sudah penuh atau permintaan pengguna yang terlalu banyak sehingga layanan tersebut tidak berjalan maksimal. Dan batasan masalah yang digunakan pada penelitian ini adalah sebagai berikut:

1. *Container Orchestration* yang digunakan adalah *Kubernetes* dan *Docker Swarm*.
2. *Container* yang digunakan adalah *Docker Container*.
3. Layanan yang digunakan adalah *webservice*.

Tujuan

Melakukan perbandingan performansi proses *scaling* pada *Kubernetes* dan *Docker Swarm*. Di dalam *Kubernetes* menggunakan metode *Horizontal Pod auto scaler* sedangkan di *Docker Swarm* menggunakan metode *scaling*. Dengan parameter pembandingan antara *Kubernetes* dan *Docker Swarm* adalah *Load Testing* untuk skalabilitas, waktu *scaling up* dan *scaling down* untuk performansi.

Organisasi Tulisan

Penulisan penelitian ini tersusun atas 5 bab. Pada bab 2 akan dijelaskan tentang Studi Terkait, bab 3 dijelaskan tentang sistem yang dibangun, yaitu arsitektur proses *scaling* pada *container orchestration*. Bab 4 akan dijelaskan tentang evaluasi, dan bab 5 dijelaskan kesimpulan dan saran.