



9th Applied Business and Engineering Conference

IMPLEMENTASI OPENSIFT CLUSTER MENGGUNAKAN KVM

Sari Fauzia Elza¹, Agus Urip Ari Wibowo²

¹Program Studi Teknik Informatika, Politeknik Caltex Riau, Jl. Umban Sari (Patin) No. 1 Rumbai, Pekanbaru, 28265

²Program Studi Teknik Informatika, Politeknik Caltex Riau, Jl. Umban Sari (Patin) No. 1 Rumbai, Pekanbaru, 28265

E-mail: sari17ti@mahasiswa.pcr.ac.id, agus@pcr.ac.id

Abstract

Now, technological developments are increasing rapidly, one of which is in cloud computing stored on servers. The more users accessing the service, the heavier the server load will be. Therefore, cluster technology is needed. With cluster technology, every incoming request will be divided equally or equally to each server. However, to run a cluster, a cluster orchestration software is needed. In this study, we compare two orchestrations, namely Openshift and Kubernetes. The virtualization technology used is Kernel-Based Virtual Machine (KVM). In this study, we compare latency, CPU usage, and memory usage when accessing the web between Openshift cluster and Kubernetes cluster technologies. The highest latency test results were 18201 ms by Kubernetes cluster and the lowest latency was 1374 ms by Openshift cluster technology. The latency of the Openshift cluster is lower because the nodes are divided into compute, infra, and controller. Good latency is low latency because it can open web pages faster. In standby, the lowest CPU usage and the lowest memory usage are Openshift clusters, namely 0.13% and 60%, this is because in Openshift cluster technology each node is divided into computing, infrared, and controller. In a busy state, the highest CPU usage and the highest memory are Kubernetes clusters, namely 75.14% and 98%, this is because in Kubernetes there is no node sharing.

Keywords: *Openshift cluster, Kubernetes cluster, Latency, CPU, Memory.*

Abstrak

Dewasa ini, perkembangan teknologi meningkat pesat salah satunya pada *Cloud computing* yang tersimpan pada *server*. Semakin banyak pengguna yang mengakses layanan tersebut akan mengakibatkan beban *server* menjadi lebih berat. Oleh karena itu, dibutuhkan teknologi *cluster*. Dengan teknologi *cluster* setiap *request* yang masuk akan dibagi secara merata atau seimbang kepada masing-masing *server*. Namun, untuk menjalankan sebuah *cluster*, maka dibutuhkan sebuah *software cluster orchestration*. Pada penelitian ini membandingkan dua *orchestration* yaitu *Openshift* dan *Kubernetes*. Teknologi virtualisasi yang digunakan adalah virtualisasi *Kernel-Based Virtual Machine (KVM)*. Pada penelitian ini membandingkan *latency*, pemakaian CPU, dan pemakaian *memory* saat mengakses halaman web antara teknologi *Openshift cluster* dan *Kubernetes cluster*. Diperoleh hasil pengujian *latency* tertinggi yaitu 18201 ms oleh *Kubernetes cluster* dan *latency* terendah yaitu 1374 ms oleh teknologi *Openshift cluster*. *Latency* dari *Openshift cluster* lebih rendah dikarenakan *node* dibedakan menjadi *compute*, *infra*, dan *controller*. *Latency* yang baik yaitu *latency* rendah karena dapat membuka halaman web lebih cepat Dalam keadaan *standby*, pemakaian CPU terendah dan pemakaian *memory* terendah adalah *Openshift cluster* yaitu 0,13% dan 60%, hal ini dikarenakan dalam teknologi *Openshift cluster* setiap



9th Applied Business and Engineering Conference

node dibedakan menjadi *compute*, *infra*, dan *controller*. Dalam keadaan *busy*, pemakaian CPU tertinggi dan *memory* tertinggi adalah *Kubernetes cluster* yaitu 75,14% dan 98%, hal ini disebabkan dalam *Kubernetes* tidak adanya pembagian *node*.

Kata Kunci: *Openshift Cluster, Kubernetes Cluster, Latency, CPU, Memory.*

PENDAHULUAN

Dewasa ini, perkembangan teknologi meningkat pesat khususnya pada penggunaan teknologi komputasi. Teknologi komputasi adalah aktivitas penggunaan dan pengembangan teknologi komputer, perangkat keras, dan perangkat lunak komputer. Jenis-jenis komputasi modern terbagi tiga macam, yaitu komputasi mobile, komputasi grid, dan cloud computing (Alkhawarizmi 2019). *Cloud Computing* merupakan sebuah teknologi yang menjadikan Internet sebagai pusat server untuk mengelola data dan juga aplikasi pengguna. *Cloud Computing* memudahkan penggunanya untuk menjalankan program tanpa harus menginstal aplikasi terlebih dahulu dan memudahkan pengguna untuk mengakses data dan informasi melalui Internet (IDCloudHost 2019). *Cloud computing* memiliki tiga model layanan seperti *Software as a Service (SaaS)*, *Infrastructure as a Service (IaaS)* dan *Platform as a Service (PaaS)*. Pada penelitian ini model yang digunakan adalah *Infrastructure as a Service (IaaS)*. IaaS merupakan layanan yang memungkinkan penggunaan server tanpa perlu membeli komputer dan peralatannya secara fisik, melakukan pemeliharaan rutin, dan melakukan konfigurasi perangkat (Intermedia 2020). Berdasarkan model penyebarannya, *cloud computing* memiliki empat pemodelan, yaitu *Private Cloud*, *Community Cloud*, *Public Cloud*, *Hybrid Cloud*. *Private cloud* adalah salah satu model komputasi yang menawarkan lingkungan kepemilikan yang didedikasikan untuk keperluan dan kebutuhan satu entitas bisnis (IdCloudHost 2020).

Layanan *cloud computing* tersimpan pada server. Semakin banyak pengguna yang mengakses layanan tersebut akan mengakibatkan beban server menjadi lebih berat, dikarenakan layanan harus bekerja secara terus menerus dalam menyediakan data yang dibutuhkan oleh pengguna. Oleh karena itu, untuk meningkatkan kualitas kerja server dibutuhkannya solusi ketika terjadi masalah dalam satu node maka mesin virtual didalamnya akan berpindah ke node lain untuk meminimalkan gangguan pada layanan



9th Applied Business and Engineering Conference

yang diakses client. Layanan-layanan server dijalankan pada mesin-mesin server virtual di dalam mesin server fisik. Jumlah layanan yang banyak, data-data penting, dan tingkat ketergantungan kinerja dari perusahaan, instansi, atau organisasi yang tinggi terhadap layanan server membuat server harus dapat melayani secara terus menerus (Yudi Restu Adi 2016). Oleh karena itu, diperlukanya teknologi *cluster* yang bertujuan mengurangi beban kerja dari server (Apriliana, Darusala and Nathasia 2018).

Dengan adanya teknologi *cluster* beban kerja sever akan dibagi dengan rata sesuai dengan jumlah server yang telah ter*cluster*. Setiap *request* yang masuk akan dibagi secara merata atau seimbang kepada masing-masing server. Dalam teknologi *cluster* ini dibutuhkan sebuah *platform* untuk manajemen. *Platform* yang dapat digunakan antara lain Kubernetes dan Openshift. Openshift adalah *platform* yang dioptimalkan untuk aplikasi web, memungkinkan membangun, menguji, dan menggunakan aplikasi web, tanpa menyediakan dan memelihara server khusus untuk setiap aplikasi. Kubernetes merupakan *platform open-source* untuk manajemen beban kerja aplikasi dalam container, serta menyediakan konfigurasi dan otomatisasi secara deklaratif. Kubernetes berada dalam ekosistem yang besar dan berkembang pesat. *Service, support*, dan perkakas Kubernetes tersedia secara meluas (Roziq 2020)

Teknologi virtualisasi server yang diterapkan pada server dengan prosesor berinti lebih dari satu, dapat dimanfaatkan untuk menjalankan aplikasi-aplikasi dan *services* secara virtualisasi. Dengan menggunakan teknologi virtualisasi permasalahan mengenai kebutuhan yang tinggi terhadap perangkat keras dapat teratasi. Hal ini tentu akan mengurangi jumlah biaya yang dialokasikan untuk pembelian perangkat keras. Sehingga penggunaan teknologi virtualisasi lebih menguntungkan dibandingkan penggunaan mesin yang bukan virtual. Salah satu teknologi yang memungkinkan Openshift dan Kubernetes dapat berjalan didalamnya adalah KVM. Kernel-Based Virtual Machine (KVM) adalah salah satu teknologi virtualisasi (*hypervisor*) yang dikembangkan oleh Linux. KVM merupakan *dedicated server* atau berdiri sendiri karena memiliki kinerja yang baik dan handal serta VPS lain tidak akan terganggu. Kelebihan virtualisasi KVM



9th Applied Business and Engineering Conference

antara lain: dapat menggunakan sistem operasi apa saja, dapat menggunakan SWAP, disertai dengan remote VNC, dan tidak dibatasi oleh node (Asrawan 2019).

Berdasarkan uraian diatas, perlu dilakukan analisis perbandingan antara Openshift *cluster* menggunakan KVM dengan Kubernetes *cluster* menggunakan KVM. Penelitian ini bertujuan untuk melihat perbedaan kinerja sumberdaya saat teknologi Openshift *cluster* dengan Kubernetes *cluster* diterapkan pada teknologi virtualisasi yang sama, maka dilakukan penelitian mengenai “Implementasi Openshift *Cluster* menggunakan KVM” dengan tujuan menginstalasi dan memajemen Openshift *cluster* dan Kubernetes *cluster* menggunakan virtualisasi KVM, dan membandingkan dan menganalisis perbedaan kinerja sumber daya perangkat keras (CPU dan memori) Openshift *cluster* dengan Kubernetes *cluster* menggunakan KVM.

METODE PENELITIAN

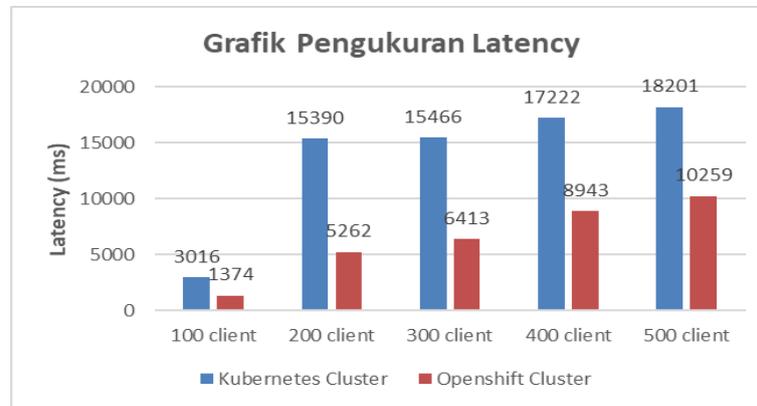
Metode penelitian yang dilakukan pada penelitian ini dengan menggunakan metode pengujian *performance testing*. Pengujian *performance* akan diuji dengan beberapa cara yaitu: pengujian *stress testing* bertujuan untuk mengetahui dan mengukur kinerja sebuah *web server* dalam menangani *request client* dalam jumlah yang sangat banyak secara bersamaan dalam satu waktu pada *server* dengan menggunakan *tools* JMeter. Pengujian kinerja virtualisasi *web server* dalam keadaan *standby* dan *busy* dilakukan untuk memantau persentase penggunaan CPU dan *memory* yang terpakai pada *web server* tidak di akses *client (standby)* dan di akses *client (busy)* pada *server Openshift cluster* dan *Kubernetes cluster* dengan menggunakan aplikasi protokol.

HASIL DAN PEMBAHASAN

1. Hasil *Stress Testing*

Berdasarkan pengujian yang dilakukan pada saat *stress testing* terhadap *web server* pada masing-masing *server*. Pengukuran tingkat *latency* yang dilakukan dengan 5 kali tahapan pengujian yaitu 100 *client*, 200 *client*, 300 *client*, 400 *client* dan 500

client dengan masing-masing pengujian dilakukan 5 kali percobaan dalam mengambil data.

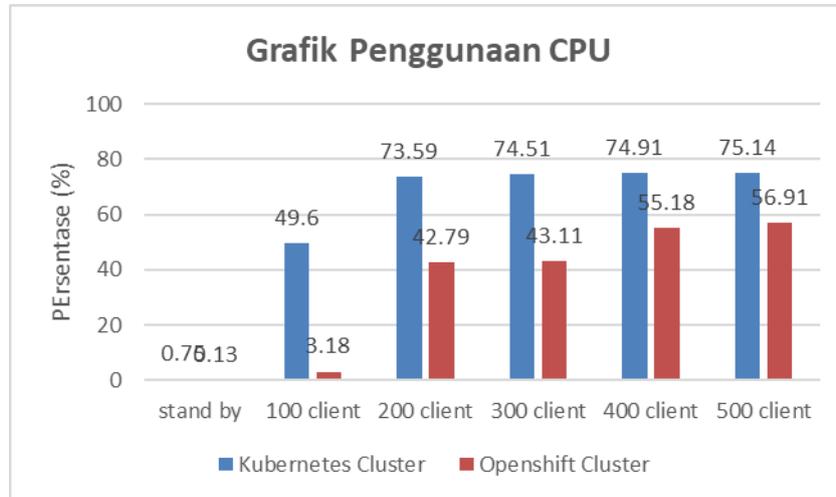


Gambar 1. Grafik Pengukuran *Latency*

Berdasarkan pengujian yang telah dilakukan diperoleh grafik seperti pada Gambar 4.1 yang merupakan hasil rata-rata jumlah *latency* dari semua pengujian yang dilakukan pada *server Openshift cluster* dan *Kubernetes cluster*. Pada teknologi *Openshift cluster* menunjukkan jumlah *latency* paling tinggi yaitu 10259 ms dan *latency* paling rendah yaitu 1374 ms. Sedangkan pada *Kubernetes cluster* jumlah *latency* paling tinggi yaitu 18201 ms, dan jumlah *latency* paling rendah yaitu 3016 ms. Hal ini dikarenakan pada *Openshift cluster node* dibedakan menjadi *compute*, *infra*, dan *controller*. Sehingga ketika layanan *web server* diakses oleh pengguna, *web server* akan dapat tampil lebih cepat dibandingkan dengan *Kubernetes cluster*. *Latency* merupakan berapa lama perjalanan data antara sumber dan tujuan, sehingga semakin rendah *latency* terhadap *web server* akan semakin baik.

2. Hasil CPU dan *Memory*

Berdasarkan pengujian yang dilakukan pada saat keadaan *server standby* dan *busy* diperoleh data penggunaan CPU. Penggunaan CPU dilakukan dengan keadaan *standby*, 100 *client*, 200 *client*, 300 *client*, 400 *client* dan 500 *client* dengan masing-masing pengujian dilakukan 5 kali percobaan dalam mengambil data.



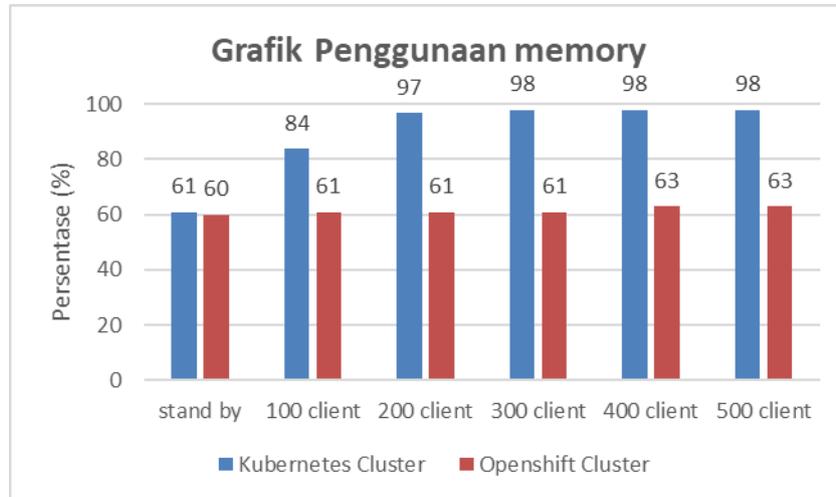
Gambar 2. Grafik penggunaan CPU saat *standby* dan *busy*

Berdasarkan pengujian yang telah dilakukan diperoleh grafik seperti pada Gambar 2. yang merupakan hasil rata-rata *persentase* penggunaan CPU dari semua pengujian yang dilakukan pada *server Openshift cluster* dan *Kubernetes cluster* dalam keadaan *standby* dan *busy*.

Pada saat keadaan *standby*, teknologi *Openshift cluster* menunjukkan penggunaan CPU terendah yaitu 0,13% sedangkan penggunaan CPU tertinggi adalah teknologi *Kubernetes cluster* yaitu 0,75%. Pada kedua *server* terdapat *node-node* yang berjalan di atas mesin virtual.

Pada saat keadaan *busy*, teknologi *Openshift cluster* menunjukkan penggunaan CPU terendah yaitu 3,18% karena pada *Openshift cluster* setiap *node* dibedakan menjadi *compute*, *infra* dan *controller* pada saat menerima *request client*. sedangkan penggunaan CPU tertinggi terdapat pada teknologi *Kubernetes cluster* yaitu 75,14% karena tidak adanya pembagian *control plane* (*compute*, *infra*, *controller*). Pengujian dengan membedakan jumlah *client* ternyata memberikan hasil penggunaan CPU yang berbeda. Semakin banyak *client* yang mengakses maka akan naik penggunaan CPU.

Selanjutnya adalah analisis penggunaan *memory* pada masing-masing *server* pada saat *standby* dan diakses oleh 100 *client*, 200 *client*, 300 *client*, 400 *client* dan 500 *client*.



Gambar 3. Grafik penggunaan *memory* saat *standby* dan *busy*

Berdasarkan pengujian yang telah dilakukan diperoleh grafik seperti pada Gambar 3. yang merupakan hasil rata-rata *persentase* penggunaan *memory* dari semua pengujian yang dilakukan pada *server Openshift cluster* dan *Kubernetes cluster* dalam keadaan *standby* dan *busy*.

Pada saat keadaan *standby*, teknologi *Openshift cluster* menunjukkan penggunaan *memory* terendah yaitu 60% karena pada *Openshift cluster* terdapat *control plane* yang membagi *node* menjadi *compute*, *infra* dan *controller*. Sedangkan penggunaan *memory* tertinggi adalah teknologi *Kubernetes cluster* yaitu 61% dikarenakan tidak adanya *control plane* pada teknologi *Kubernetes cluster*.

Pada saat keadaan *busy*, teknologi *Openshift cluster* menunjukkan penggunaan *memory* terendah yaitu 61% karena adanya *control plane* yang membagi *node* menjadi *compute*, *infra* dan *controller*, sedangkan penggunaan *memory* tertinggi terdapat teknologi *Kubernetes cluster* yaitu 98%. Pengujian dengan membedakan jumlah klien ternyata memberikan hasil penggunaan *memory* yang berbeda. Semakin banyak klien yang mengakses maka akan naik penggunaan *memory*.

SIMPULAN

Berdasarkan penelitian yang telah dilakukan, ada beberapa hal yang dapat disimpulkan dari penelitian ini, yaitu:



9th Applied Business and Engineering Conference

6. *Openshift cluster* dan *Kubernetes cluster* dapat dibangun di dalam virtualisasi KVM.
7. *Latency* tertinggi yaitu 18201 ms oleh *Kubernetes cluster* dan *latency* terendah yaitu 1374 ms oleh teknologi *Openshift cluster* berdasarkan semua pengujian yang dilakukan.
8. Dalam keadaan *standby*, pemakaian CPU terendah dan pemakaian *memory* terendah adalah *Openshift cluster* yaitu 0,13% dan 60% dibandingkan dengan teknologi *Kubernetes cluster* berdasarkan semua pengujian yang dilakukan.
9. Dalam keadaan *busy*, pemakaian CPU tertinggi dan *memory* tertinggi adalah *Kubernetes cluster* yaitu 75,14% dan 98% berdasarkan semua pengujian yang dilakukan.
10. Semakin banyak jumlah *client* yang mengakses *web server* maka semakin tinggi penggunaan sumber daya perangkat keras pada kedua *server*.

DAFTAR PUSTAKA

- Alkhawarizmi. (2019). *Komputasi Modern*. Retrieved 11 20, 2020, from mamz.weebly: <https://mamz.weebly.com/komputasi-modern.html>
- Apriliansa, L., Darusala, U., & Nathasia, N. D. (2018). Clustering Server Pada Cloud Computing. *Journal of Information Technology and Computer Science*, 6.
- Asrawan, M. A. (2019). ANALISIS PERFORMANSI WEB SERVER MENGGUNAKAN. *Pekanbaru: Politeknik Caltex Riau*.
- IDCloudHost. (2019, September 16). *Mengenal Apa itu Cloud Computing : Defenisi, Fungsi, dan Cara Kerja*. Retrieved 11 18, 2020, from idcloudhost.com: <https://idcloudhost.com/mengenal-apa-itu-cloud-computing-defenisi-fungsi-dan-cara-kerja/>
- IdCloudHost. (2020, August 11). *Apa itu Private Cloud : Pengertian, Fungsi, Keunggulan dan kelebihanannya*. Retrieved February 1, 2021, from IdCloudHost: <https://idcloudhost.com/apa-itu-private-cloud-pengertian-fungsi-keunggulan-dan-kelebihannya/>
- Intermedia, B. (2020, April 14). *Mengenal SaaS, PaaS, dan IaaS dalam Cloud Computing*. Retrieved February 1, 2021, from Jagoanhosting.com: <https://www.jagoanhosting.com/blog/mengenal-saas-paas-dan-iaas/>



9th Applied Business and Engineering Conference

- Roziq, M. (2020). *Apa Itu Kubernetes? Mengapa Harus Menggunakan Kubernete?*
Retrieved 2021, from Caraguna.com: <https://caraguna.com/apa-itu-kubernetes/>
- Yudi Restu Adi, O. D. (2016). Perancangan Sistem Cluster Server Untuk Jaminan Ketersediaan Layanan Tinggi pada Lingkungan Virtual. *JNTETI*.