

Membangun Cloud Menggunakan Docker Pada Implementasi Load Balancing dan Pengujian Algoritma Round Robin Pada Web Server

Imammuddin¹⁾, Januar Al-Amien²⁾, Evans Fuad³⁾

^{1),3)}Program Studi Teknik Informatika Fakultas Ilmu Komputer, UMRI
imammudin@gmail.com

²⁾Program Studi Teknik Informatika Fakultas Ilmu Komputer, UMRI
januaralamien@umri.ac.id

Abstract

Along with the development of the world of technology and information, web site providers offer easy access and information provided by the cloud, which is inseparable from a server that is reliable and fast in calculating very large amounts of traffic, so there are still many problems in developing better services, such as new services, but the high price of physical devices is a reason for services that are not able to upgrade traffic demand into higher resources. Therefore Docker who supports container-based virtualization technology is the answer to the resources of physical devices. By using technology containers, physical servers can create multiple virtual servers using multiple resources such as memory and CPU load, different virtual container-based virtualization technologies. engine because it does not use the resources as a whole, but only uses the services that are needed. It is expected that implementing a technology based container docker with load balancing can increase the number of services with minimal resources.

Keywords: Virtualization, Docker, Load Balancing

Abstrak

seiring perkembangan dunia teknologi dan informasi, penyedia website menawarkan kemudahan dalam mengakses informasi dan mengelola informasi itu sendiri melalui ketersediaan penyimpanan cloud, yang tidak terlepas dari peranan server yang handal dan cepat dalam menangani jumlah trafik yang sangat besar, namun masih banyak permasalahan dalam mengembangkan layanan menjadi lebih baik, seperti penambahan service baru, tapi mahalnya harga suatu perangkat fisik menjadi alasan mengapa banyak layanan yang belum mampu mengupgrade trafik request menjadi lebih tinggi karena keterbatasan resource. Oleh karena itu hadirnya docker yang mengadopsi teknologi virtualisasi berbasis container menjadi jawaban atas keterbatasan resource dari suatu perangkat fisik. dengan menerapkan teknologi container, server fisik dapat membuat beberapa server virtual dengan menggunakan sedikit resource seperti memory dan CPU load, Teknologi virtualisasi berbasis container berbeda dengan virtual mesin karena tidak menggunakan resource secara keseluruhan, namun hanya menggunakan service yang di butuhkan saja. Diharapkan penerapan teknologi berbasis container docker dengan load balancing dapat meningkatkan jumlah layanan dengan minim resource.

Kata Kunci: *Virtualisasi, Docker, Load Balancing*

© 2019 Jurnal CTIA

1. Pendahuluan

Perkembangan teknologi yang dewasa ini sudah sangat mempermudah kegiatan manusia dalam melakukan kegiatan sehari harinya. Teknologi juga sangat berperan pada penghematan waktu kerja dalam melakukan tugas tugas yang awalnya memakan waktu lebih lama. Dengan teknologi juga manusia dapat bertukar informasi dan data dengan sangat cepat tanpa adanya batasan ruang dan waktu dengan jarak yang jauh sekalipun. Saat ini sudah banyak aplikasi berbagi

informasi, data dan *email* dengan akses internet, hanya saja untuk menggunakan aplikasi tersebut kita terlebih dahulu melakukan instalasi *software*, namun saat ini sudah banyak penyedia layanan aplikasi yang berbasis *web*. Terdapat banyak aplikasi berbasis *web* yang di tawarkan kepada *user*, dari versi *desktop* hingga *mobile phone*, membahas tentang aplikasi *web* tidak terlepas dari perangkat *server*. Dan akhir akhir ini banyak pengembang mengalihkan *service* dari server konvensional ke *cloud*. Pada awalnya menggunakan *microservice* ke teknik virtualisasi mesin, dan

sekarang ke teknologi berbasis virtualisasi *container*. Salah satu platform yang mendukung teknik *container* adalah *docker*

Docker adalah suatu *platform* terbuka bagi pengembang perangkat lunak dan pengelola sistem jaringan untuk membangun, mengirimkan dan menjalankan aplikasi-aplikasi terdistribusi. (Adiputra,2015). Definisi tersebut membawa pengertian praktis bahwa *Docker* merupakan suatu cara memasukkan layanan ke dalam lingkungan terisolasi bernama *container*, sehingga layanan tersebut dapat dipaketkan menjadi satu bersama dengan semua pustaka dan software lain yang dibutuhkan. Upaya yang dapat dilakukan untuk meningkatkan kualitas QOS pada *web server* adalah dengan menambahkan jumlah *server*, *Server server* yang berada dalam himpunan *server* disebut dengan *cluster server*. Penggunaan *cluster server* dapat menghilangkan hambatan dari kinerja *CPU*, penerapan pembagian beban kerja pada *server clustering* untuk meningkatkan kinerja sistem yaitu dengan menerapkan metode *load balancing*. *Server load balancing* atau di sebut juga dengan *director* mendistribusikan beban kerja ke *multiple server* dengan mempertimbangkan kapasitas dari setiap *server* untuk mengurangi terjadinya kegagalan *server* (Alimuddin & Ahmad, 2016). Penerapan *load balancing* dalam *web server* sangat penting dan dapat menjadi solusi yang tepat dan efektif untuk menangani beban *server* yang sibuk dan dapat meningkatkan skalabilitas pada sistem terdistribusi. Peningkatan kinerja *CPU* pada *server clustering* juga dapat dimaksimalkan dengan menerapkan metode algoritma penjadwalan sesuai dengan kebutuhan. penjadwalan berkaitan dengan permasalahan memutuskan proses mana yang akan dilaksanakan dalam suatu sistem, Proses yang belum mendapatkan bagian alokasi dari *CPU*. Algoritma penjadwalan berfungsi untuk menentukan proses mana yang akan di eksekusi oleh *CPU*. Universitas Muhammadiyah Riau adalah salah satu universitas swasta yang berdiri pada tahun 2008. Pada saat ini di tahun 2018 universitas muhammadiyah memiliki 8 fakultas dan 20 program studi. Dengan jumlah 8 fakultas dan 20 program studi tentunya jumlah mahasiswa dan dosen di lingkungan universitas muhammadiyah riau berjumlah ribuan. Hal ini juga yang mempengaruhi untuk menggunakan sistem informasi untuk mengakses informasi perkuliahan, jadwal kuliah dan lainnya. Di setiap aktivitas ini pun tak terlepas pula dengan jaringan yang terhubung di dalamnya seperti penggunaan *server*. Saat

ini universitas muhammadiyah riau belum menerapkan sistem virtualisasi mesin ataupun virtualisasi berbasis *container* pada *server*.

2. Metodologi Penelitian

a. Kerangka Konseptual Penelitian

Kerangka konseptual merupakan suatu bentuk kerangka berpikir yang dapat digunakan sebagai pendekatan dalam memecahkan masalah. Biasanya kerangka penelitian ini menggunakan pendekatan ilmiah dan memperlihatkan hubungan antar variabel dalam proses analisisnya.



Gambar 1 : kerangka Konseptual Penelitian

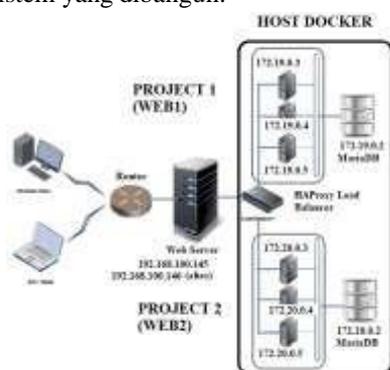
b. Metodologi Penelitian

- 1) Pengumpulan data merupakan langkah awal dalam melakukan suatu penelitian. Pengumpulan data yang digunakan pada penelitian ini menggunakan metode *observasi* dan *wawancara*.
- 2) *Observasi*, Teknik *Observasi* dilakukan untuk memperoleh data dengan cara mengamati secara langsung obyek yang berkaitan dengan kegiatan penelitian. Pada tahap ini peneliti melakukan pengamatan pada sistem web server yang ada pada server universitas muhammadiyah riau, lalu tahap pengembangannya dengan melakukan kegiatan penelitian terhadap kinerja *container* pada web server dan menerapkan pada server Universitas Muhammadiyah riau secara langsung.
- 3) *Wawancara*, merupakan teknik pengumpulan data dengan melakukan tanya-jawab langsung kepada pengguna (pegawai)

Universitas Muhammadiyah terhadap sistem yang dibuat sesuai dengan kebutuhan yang diharapkan. Untuk mendapatkan data dari hasil wawancara, peneliti meminta informasi dengan wawancara kepada Staff IT Support Universitas muhammadiyah yaitu saudara Suryanto

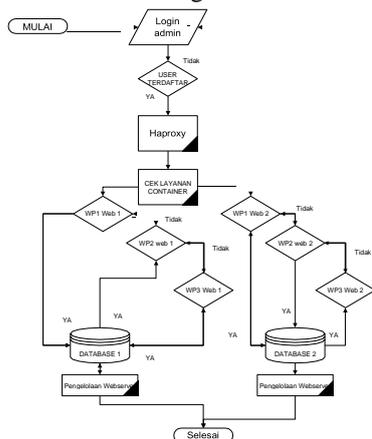
c. Perancangan Sistem

1) *Perancangan Topologi Jaringan* Topology jaringan menggambarkan bentuk konsep hubungan komputer yang saling terhubung dengan sistem. Memberikan gambaran kepada pengguna struktur jaringan yang bekerja pada sistem yang dibangun.



Gambar 2 : Topologi jaringan multiservice docker

a. *Perancangan Flowchart* merupakan sebuah jenis diagram yang mewakili algoritma, alir kerja atau proses yang menampilkan langkah langkah dalam bentuk symbol-simbol grafis. Gambar 4 merupakan flowchart dari alur sistem virtualisasi berbasis container dengan load balancing



Gambar 3 : Flowchart docker container dengan load balancing

3. Hasil dan Pembahasan

a. Instalasi Docker

Untuk dapat melakukan perancangan container, langkah awal adalah melakukan instalasi docker. Docker digunakan untuk menjalankan multi container FTP. Untuk melakukan instalasi docker pertama kali yang harus dilakukan adalah melakukan menambahkan repository docker kedalam repository milik host sever dengan menggunakan perintah berikut.

```

$ curl -fsSL https://download.docker.com/linux/ubuntu \
  > 'deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  > $(lsb_release -cs) \
  > stable' |
  
```

Gambar 4 : add repository docker

Setelah menambahkan repository docker kedalam list repository pada host server, langkah selanjutnya adalah melakukan instalasi docker dengan menggunakan perintah berikut.

```

~$ sudo apt-get install docker-ce
  
```

Gambar 5 : perintah instalasi docker

a. Instalasi Docker Compose

Docker compose merupakan sebuah *tool* untuk menentukan dan menjalankan multi-container pada docker. Dengan menggunakan compose, file dengan format YAML merupakan file untuk mengkonfigurasi service aplikasi yang akan dijalankan. Dan dengan menggunakan satu instruksi dapat membuat sekaligus menjalankan semua service yang sudah dikonfigurasi.

Untuk dapat menjalankan service pada docker compose, terlebih dahulu melakukan instalasi dengan menggunakan perintah berikut.

```

$ curl -L https://github.com/docker/compose/releases/download/1.18.0/docker-compose -s -o /usr/local/bin/docker-compose
  
```

Gambar 6 : perintah untuk instalasi docker compose

b. Membangun Container Webserver

Pada penelitian ini, penulis membangun suatu service webserver dengan multi-container dengan menggunakan docker-compose, yang mana ada 2 service yang di tempatkan pada masing masing directory.

Hal pertama sekali yang perlu di lakukan adalah memuat direktory PROJECT1 dan direktori PROJECT2 dan di tempatkan script yang bernama docker-compose.yml. berikut adalah isi script dari *docker-compose.yml* PROJECT1:

b. Instalasi Haproxy Load Balancer

HAProxy (High Availability Proxy) merupakan aplikasi yang digunakan sebagai load balancer dan reverse proxy yang biasa dijalankan pada system operasi linux. Pada penelitian ini HAProxy digunakan sebagai pembagi beban kinerja kedalam 3 container virtual server wordpress. Untuk dapat menggunakan HAProxy terlebih dahulu keluar dari direktori proyek1 lalu kemudian melakukan instalasi Haproxy dengan menggunakan perintah seperti berikut :

```
~$ sudo apt-get install haproxy
```

Gambar 7 : perintah install haproxy load balancer

Jika instalasi sudah selesai di lakukan, maka selanjutnya adalah mengkonfigurasi haproxy.cfg.

c. Hasil Pengujian

Pada pengujian ini mengacu pada skenario yang telah di rencanakan, yang mana di dapat hasil sebagai berikut :

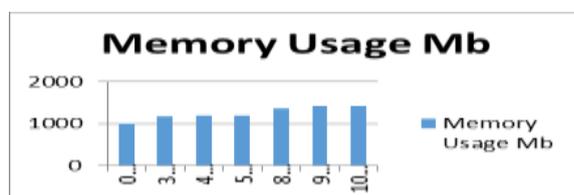
i. Hasil Pengujian Skenario satu



Gambar 8 : hasil pengujian penggunaan memory, CPU load pada service 1



Gambar 9 : hasil pengujian penggunaan memory, CPU load pada service 2



Gambar 10 : Hasil Penggunaan Memory pada 2 service webservice docker

4. Kesimpulan

Berdasarkan hasil yang diperoleh melalui pengamatan dan ujicoba, maka dapat ditarik kesimpulan sebagai berikut:

Dari penggunaan Virtualisasi berbasis container pada webserver di dapatkan hasil penggunaan memory pada service 1 hanya sebanyak 1.20Gb, dan jika menjalan 2 service sekaligus penggunaan memory hanya sebanyak 1.43 GB. Pengambilan resource ke kernel Host OS menjadikan Virtualisasi berbasis container sangat ringan dan hanya mengambil service service yang di perlukan sehingga resource dari Host OS tidak terlalu terbebani, dan service satu dengan service yang lain terisolasi sehingga service tidak akan terganggu. Sedangkan untuk VMware dengan pengujian 3 Virtual server Ubuntu mengkonsumsi jumlah memory berjumlah 1.73Gb.

Ucapan Terimakasih

Rekan-Rekan Seperjuangan Alumni Sarjana Komputer universitas Muhammadiyah Riau Fakultas Ilmu Komputer Tahun angkatan 2014

Sebutkan nama pemberi dana dan pemberi fasilitas yang membantu.

Daftar Rujukan

- [1] adiputra, firmansyah. (2015). container dan docker: teknik virtualisasi dalam pengelolaan banyak aplikasi web. jurnal simantec.
- [2] Afriansyah, Muhammad Faizal. Rochim, adian Fatrhur. Widiyanto, Eko Didik. (2015). Rancang Bangun Layanan Cloud Computing Berbasis IaaS Menggunakan Virtualbox. Jurnal Teknologi dan Teknik Komputer. Vol. 3 No. 1
- [3] Afdhal.(2013). Studi Perbandingan Layanan Coud Computing. Jurnal Rekayasa ElektriKa Vol.10, N0 4
- [4] Alimuddin, A., & Ashari, A.(2016). Peningkatan Kinerja Siakad Menggunakan Metode *Load Balancing* dan *Fault Tolerance* Di Jaringan Kampus Universitas Halu Oleo. IJCCS (*Indonesian Journal of Computing and Cybernetics Systems*), 10(1), 11-22.
- [5] Fauziah, Yuli. (2014). Arsitektur Cloud Computing Pada Sistem Informasi Desa Sebagai Layanan Akses Informasi Desa.Jurnal Seminar Nasional Informatika
- [6] Nugroho, M. A., & Kartadie, R. (2016). Analisis Kinerja Penerapan *Container* untuk *Load Balancing* Web *Server* Pada Raspberry Pi. Jurnal Ilmiah Penelitian dan Pembelajaran Informatika (JIPI), 1
- [7] Oktavianus, Y. L. (2013). Membangun Sistem Cloud Computing Dengan Implementasi *Load Balancing* dan

Pengujian Algoritma Penjadwalan Linux Virtual Server Pada FTP Server. Jurnal Nasional Teknik Elektro, 2

- [8] Kan, Chuanqi (2016) DoCloud : An Elastic Cloud Platform for Web Applications Based on Docker. ICACT
- [9] Sardi, Endah Sri Maulana (2017) Implementasi Teknik Virtualisasi Container dengan Docker untuk pengelolaan aplikasi Web di Dinas Komunikasi dan Informatika Kota Payakumbuh. Jurnal Teknik Politeknik Negeri Padang
- [10] **Zharfan, Anggie Afianda(2018)** Implementasi analisis kinerja container untuk load balancingFTP server(studi kasus PT.Yodya Karya(persero) cabang pekanbaru