STUDI KOMPARASI PERFORMA NGINX DAN HAPROXY SEBAGAI LOAD BALANCER DI CLOUD MENGGUNAKAN TEKNOLOGI KONTAINER

Skripsi

Diajukan oleh:

Joshua Gibeon Mulyana

71160004

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS KRISTEN DUTA WACANA

YOGYAKARTA

2020

STUDI KOMPARASI PERFORMA NGINX DAN HAPROXY SEBAGAI LOAD BALANCER DI CLOUD MENGGUNAKAN TEKNOLOGI KONTAINER

Skripsi

Diajukan kepada Fakultas Teknologi Informasi Program Studi Informatika
Universitas Kristen Duta Wacana
Sebagai salah satu syarat dalam memperolah gelar Sarjana Komputer

Diajukan oleh:
Joshua Gibeon Mulyana
71160004

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA
2020

HALAMAN PERSETUJUAN

Saya yang bertanda tangan dibawah ini:

NIM : 71160004

Nama : Joshua Gibeon Mulyana

Prodi / Fakultas : Informatika / Teknologi Informasi

Judul Tugas Akhir : Studi komparasi performa NGINX dan HAPROXY sebagai load balancer di cloud menggunakan teknologi kontainer

bersedia menyerahkan Tugas Akhir kepada Universitas melalui Perpustakaan untuk keperluan akademis dan memberikan Hak Bebas Royalti Non Ekslusif (Non-exclusive Royalty-free Right) serta bersedia Tugas Akhirnya dipublikasikan secara online dan dapat diakses secara lengkap (full access).

Dengan Hak Bebas Royalti Non ekslusif ini Perpustakaan Universitas Kristen Duta Wacana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk database, merawat dan memublikasikan Tugas Akhir saya selama tetap mencamtumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenar-benarnya.

Yogyakarta, 17 Juli 2020

Yang menyatakan,

(71160004 – Joshua Gibeon Mulyana)

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

STUDI KOMPARASI PERFORMA NGINX DAN HAPROXY SEBAGAI LOAD BALANCER MENGGUNAKAN TEKNOLOGI KONTAINER

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar kesarjanaan saya.

Yogyakarta, 12 Juli 2020

JOSHUA GIBEON MULYANA
71160004

HALAMAN PENGESAHAN

STUDI KOMPARASI PERFORMA NGINX DAN HAPROXY SEBAGAI LOAD BALANCER MENGGUNAKAN TEKNOLOGI KONTAINER

Oleh: JOSHUA GIBEON MULYANA / 71160004

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 17 Juni 2020

Yogyakarta, 10 Juli 2020
Mengesahkan,

Dewan Penguji:

1. Willy Sudiarto Raharjo, S.Kom.,M.Cs. .....
2. Gani Indriyanta, Ir. M.T. .....
3. Budi Susanto, SKom.,M.T. .....
4. Prihadi Beny Waluyo, SSi., MT. .....

Dekan	Ketua Program Studi
(Restyandito,S.Kom.,MSIS.,Ph.D.)	(Gloria Virginia,Ph.D.)

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas penyertaan-Nya sehingga penulis dapat menyelesaikan penelitian ini dengan baik.

Pembuatan laporan ini dimaksudkan untuk memenuhi persyaratan tugas mata kuliah Skripsi di Program Studi Informatika, Fakultas Teknologi Informasi Universitas Kristen Duta Wacana. Laporan ini menjelaskan proses dan hasil penelitian yang sudah penulis lakukan.

Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang sudah membantu penulis, baik dalam proses penyusunan maupun dukungan moral. Pihak-pihak tersebut adalah sebagai berikut:

1. Bapak Willy Sudiarto Raharjo, S.Kom., M.Cs dan Bapak Ir. Gani Indriyanta, MT. selaku dosen pembimbing mata kuliah Skripsi.
2. Bapak Restyandito, S.Kom., MSIS, Ph.D. selaku Dekan Fakultas Teknologi Informasi Universitas Kristen Duta Wacana.
3. Ibu Gloria Virginia, Ph.D. selaku Kepala Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana.
4. Orang tua, keluarga, dan teman yang sudah memberikan dukungan moral kepada penulis.

Laporan ini dibuat dengan segala keterbatasannya, tetapi penulis sudah berusaha untuk membuat laporan ini dengan sebaik mungkin. Semoga laporan ini dapat memberikan manfaat untuk semua pihak.

INTISARI

High availability dan *scalability* sudah menjadi suatu keharusan bagi server yang dimiliki oleh sektor industri. Salah satu konsep yang dipakai oleh pelaku teknologi adalah *horizontal scaling*. *Horizontal scaling* ini dapat dicapai oleh *load balancer*. Meskipun para pelaku teknologi mulai mengadaptasi teknologi kontainerisasi, para pelaku teknologi ini tetap memakai *load balancer*. Penelitian ini bertujuan untuk membandingkan performa NGINX dan HAPROXY sebagai *load balancer* dari sisi *response time* dan *error rate*, yang dijalankan diatas docker yang berada di *virtual machine* AWS dan GCP. Perbandingan ini dilakukan di berbagai spesifikasi *virtual machine*. Hasil penelitian menunjukkan bahwa berdasarkan konfigurasi yang sudah dibuat oleh penulis, AWS dapat menangani beban 8.93 persen baik daripada GCP. Hasil penelitian juga menyimpulkan bahwa jika dilihat dari sisi *response time*, maka secara umum NGINX mempunyai performa yang lebih baik dibandingkan dengan HAPROXY. Tetapi ketika *load balancer* dijalankan diatas VM yang mempunyai spesifikasi yang kecil, HAPROXY memilik *error rate* yang lebih stabil dibandingkan dengan NGINX.

Kata Kunci: *Load balancer*, *Cloud Computing*, *Docker*, *Container*, *Load Testing*.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Kontainerisasi adalah sebuah teknologi ringan dan portabel yang mempunyai kemampuan untuk mengembangkan, melakukan uji coba, dan melakukan *deployment* terhadap sebuah aplikasi ke server dalam jumlah besar, serta kemampuan untuk terhubung antara satu kontainer dengan kontainer yang lain (Pahl, 2015). Karena kemampuannya tersebut, sudah banyak pelaku teknologi yang mulai berpindah dari teknologi VM (*virtual machine)* ke teknologi kontainerisasi ini. Menurut data yang diperoleh Diamanti (Diamanti, 2018), muncul pertanda bahwa teknologi kontainerisasi sudah mulai menjadi teknologi *mainstream,* 44% responden berencana untuk mengganti VM dengan kontainer. Walaupun adanya perubahan penggunaan teknologi dari teknologi VM menjadi teknologi kontainerisasi, beberapa prinsip yang digunakan masih sama.

High availability dan *scalability* menjadi suatu keharusan bagi server yang dimiliki oleh beberapa sektor industri seperti perbankan. Salah satu konsep yang dipakai oleh pelaku teknologi adalah *horizontal scalability*. *Horizontal scalability* menurut Anandhi (Anandhi & Chitra, 2012) adalah penambahan beberapa unit *resource* ke dalam sistem dan memperlakukannya sebagai satu kesatuan. Untuk memperlakukannya sebagai satu kesatuan dibutuhkan sesuatu agar semua unit tersebut dapat diperlakukan sebagai satu sistem yang sama.

Load balancer merupakan sesuatu yang dibutuhkan untuk memperlakukan beberapa unit *resource* tersebut sebagai satu kesatuan, *load balancer* biasanya berada di antara client dan kumpulan server (Salchow, 2007). *load balancer* bekerja dengan cara membagi *traffic* atau permintaan yang masuk ke sebuah kumpulan server. Ada beberapa alat yang dapat digunakan untuk melakukan *load balancing*, beberapa diantaranya adalah NGINX dan HAPROXY.

Dengan mulainya teknologi kontainerisasi menjadi teknologi *mainstream* saat ini, dengan kebutuhan *high availability* dan *scalability*. Penulis akan membandingkan performa dari NGINX dan HAPROXY sebagai *load balancer*, dilihat dari sisi *error rate* dan *response time*.

1.2 Rumusan Masalah

Poin yang akan menjadi rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana perbandingan performa antara NGINX dan HAPROXY sebagai *load balancer* di teknologi kontainerisasi?
2. Bagaimana perbandingan performa *load balancer* dari sisi *error rate* dan *response time* dengan berbagai VM *instance type* yang ada di GCP (*Google Cloud Platform*) dan AWS (*Amazon Web Service*)?

1.3 Batasan Masalah

Penggunaan *load balancer* di dunia nyata sangatlah rumit dan kompleks. Sehingga dalam penelitian ini, penulis membatasi sistem yang akan digunakan. Batasan-batasan yang akan digunakan dalam penelitian ini adalah sebagai berikut:

1. Kontainer akan dijalankan di atas docker.
2. Algoritma *load balancer* yang digunakan adalah *round robin.*
3. Jaringan yang digunakan mempunyai *throughput* yang tidak stabil.
4. Parameter yang digunakan untuk perbandingan adalah *error rate* (respon dengan status 4xx) dan *response time*.
5. *Backend service* yang akan digunakan dalam penelitian ini adalah kontainer yang dibuat oleh penulis, yang akan diisi oleh aplikasi *open source*.
6. Docker akan dijalankan diatas VM *instance* di GCP dan AWS.
7. *Virtual machine* akan menggunakan sistem operasi Ubuntu 18.04 yang disediakan oleh masing-masing GCP dan AWS.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengukur dan membandingkan performa kedua *load balancer*, NGINX dan HAPROXY sebagai *load balancer* yang dijalankan di atas kontainer.
2. Membandingkan performa *load balancer* dari sisi *error rate* dan *response time* di berbagai VM *instance type* yang ada di masing-masing GCP dan AWS.

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan oleh penulis pada penelitian ini adalah:

1. Pengkajian Literatur

   Penulis akan mengumpulkan informasi yang berkaitan dengan konfigurasi HAPROXY, konfigurasi NGINX, konfigurasi docker, arsitektur jaringan server untuk penelitian, cara pengambilan data dari docker, metrik yang dipakai untuk perbandingan performa antar *load balancer*, dan alat untuk melakukan pengujian terhadap *load balancer*. Lalu penulis akan melakukan uji coba awal untuk mencari kombinasi alat simulasi *traffic*, arsitektur jaringan server, dan konfigurasi dari *load balancer* terbaik untuk dipakai dalam penelitian.
2. Pengambilan Sample Data

   Penulis akan membuat server berdasarkan informasi yang sudah penulis dapatkan ketika penulis melakukan pengkajian literatur, lalu penulis akan mengambil sampel data sebanyak 10 kali untuk masing-masing kelas. Penulis akan menggunakan alat *load testing* dan mencoba mengganti alat dan konfigurasi *load balancer*, mengganti VM *instance type*, dan mengganti *cloud provider* untuk mendapatkan sampel data yang dibutuhkan dalam penelitian ini.
3. Pengujian dan Analisis

Penulis akan mengolah sampel data yang sudah didapatkan dari langkah sebelumnya. Penulis akan membuat tabel dan membandingkan performa dari tiap objek penelitian. Pengolahan dilakukan hingga dapat ditarik kesimpulan.

4. Penarikan Kesimpulan

   Penulis akan merangkum semua hasil pengolahan data yang dilakukan oleh penulis pada langkah sebelumnya untuk menghasilkan sebuah kesimpulan dari penelitian ini.

1.6 Sistematika Penulisan

Sistematika penulisan dari penelitian ini adalah sebagai berikut:

1. BAB 1 PENDAHULUAN

   Pada bab ini, penulis akan membahas tentang gambaran umum penelitian ini. Bab ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan.

2. BAB 2 TINJAUAN PUSTAKA DAN LANDASAN TEORI

   Penulis akan membahas tentang kajian literatur dari berbagai sumber yang sudah dipelajari oleh penulis, termasuk penelitian-penelitian sebelumnya yang terkait dengan penelitian yang akan dilakukan oleh penulisan dan dasar teori yang digunakan oleh penulis dalam penelitian ini.

3. BAB 3 PERANCANGAN SISTEM

   Pembahasan tentang bagaimana penelitian ini akan dilakukan, termasuk rancangan sistem dan proses analisa yang akan digunakan, akan penulis bahas dalam bab ini.

4. BAB 4 IMPLEMENTASI DAN ANALISIS DATA

   Bab ini akan membahas tentang hasil implementasi yang telah dilakukan oleh penulis, hasil data yang didapatkan dari implementasi tersebut lalu

akan dianalisa secara kualitatif sesuai dengan apa yang telah penulis tulis di bab sebelumnya.

5. BAB 5 KESIMPULAN DAN SARAN

   Setelah implementasi dan analisis data dilakukan, penulis akan membahas hasil keseluruhan secara singkat tentang hasil penelitian dan analisa yang didapatkan. Saran juga akan diberikan oleh penulis sebagai pertimbangan untuk penelitian serupa yang akan dilakukan di kemudian hari.

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil yang didapatkan, dapat menyimpulkan bahwa *cloud provider* AWS dapat menangani beban 8.93 persen baik daripada *cloud provider* GCP.

Penulis menyimpulkan bahwa spesifikasi *virtual machine* berpengaruh terhadap *response time* dan *error rate*. Semakin besar vCPU dan memori yang dimiliki oleh VM, maka VM akan mempunyai *response time* dan *error rate* yang lebih kecil. VM dengan 2 vCPU dan 4 GB memori mempunya rata-rata performa 8.77 persen lebih baik jika dibandingkan dengan performa VM yang memiliki 1 vCPU dan 2 GB memori, VM yang memiliki 4 vCPU dan 16 GB memori mempunyai rata-rata performa 19.27 persen lebih baik jika dibandingkan dengan performa VM yang memiliki 2 vCPU dan 16 GB memori, dan VM yang memiliki 4 vCPU dan 16 GB memori memiliki rata-rata performa 26.12 persen lebih baik jika dibandingkan dengan VM yang memiliki 1 vCPU dan 2 GB memori.

Penulis menyimpulkan untuk VM yang mempunyai 1 vCPU dan 2 GB memori dan 2 vCPU dan 4 GB, NGINX mempunyai rata-rata *response time* 41.39 persen lebih baik daripada HAPROXY. Tetapi HAPROXY mempunyai rata-rata *error rate* 36.05 persen lebih baik daripada NGINX. Untuk VM yang mempunyai 4 vCPU dan 16 GB memori, NGINX mempunyai rata-rata *response time* 11.22 persen lebih baik daripada HAPROXY.

5.2 Saran

Penulis menyarankan beberapa hal untuk pengembangan penelitian kedepan, beberapa saran tersebut adalah:

1. Menggunakan konfigurasi yang berbeda untuk masing-masing HAPROXY dan NGINX.

2. Membandingkan algoritma *load balancing* yang berbeda.
3. Membandingkan *cloud provider* lain seperti Microsoft Azure dan Alibaba Cloud.

DAFTAR PUSTAKA

Anandhi, R., & Chitra, K. (2012). A Challenge in Improving the Consistency of Transactions in Cloud Databases - Scalability. *International Journal of Computer Applications*, *52*(2), 12–14. https://doi.org/10.5120/8172-1485

Dadi, S. (2017). Evaluasi Metode Load Balancing dan Fault Tolerance pada Chatting Server Jaringan Sosial.

Dhar, R. (2016). Comparative evaluation of Virtual Environments: Virtual Machines and Containers. *Journal of Mental Science*.

Diamanti. (2018). 2018 Container Adoption. Retrieved from https://diamanti.com/wp-content/uploads/2018/07/WP_Diamanti_End-User_Survey_072818.pdf

Islam, S. (2017). Network Load Balancing Methods : Experimental Comparisons and Improvement. *ArXiv Preprint ArXiv:1710.06957*.

Majid, F. N. A. (2019). *Studi Komparasi Performa Load Balancer dengan Menggunakan Teknologi Docker Container*. Yogyakarta.

Mell, P., & Grance, T. (2011). The NIST-National Institute of Standars and Technology- Definition of Cloud Computing. *NIST Special Publication 800-145*, 7.

Menascé, D. A. (2002). Load testing of Web sites. *IEEE Internet Computing*, *6*(4), 70–74. https://doi.org/10.1109/MIC.2002.1020328

Pahl, C. (2015). Containerization and the PaaS Cloud. *IEEE Cloud Computing*, *2*(3), 24–31. https://doi.org/10.1109/MCC.2015.51

Salchow, K. (2007). Load balancing 101: Nuts and bolts. *F5 Networks, Inc.*, 1–6.

Santosa, M. W. I., Primananda, R., & Yahya, W. (2018). *Implementasi Load Balancing Server Basis Data Pada Virtualisasi Berbasis Kontainer*. Malang.

Waterman, R., Lahaya, B., D, R., S, W., INS, Lucent Technologies, … Allot Networks Inc. (1999). Remote Network Monitoring MIB Extensions for Switched Networks.

Xavier, M. G., Neves, M. V, & Rose, C. A. F. De. (2014). A Performance Comparison of Container-Based Virtualization Systems for MapReduce Clusters, (February). https://doi.org/10.1109/PDP.2014.78