

# Analisis Perbandingan Layanan Data *Server* Menggunakan *Failover Cluster* pada Platform *Nginx* dan *Apache*

Hayudin Hasnanda Mail<sup>a,1,\*</sup>, Dolly Indra<sup>a,2</sup>, Ramdan Satra<sup>a,3</sup>

<sup>a</sup>Universitas Muslim Indonesia, Jalan Urip Sumoharjo, Makassar, 90231, Indonesia  
<sup>1</sup>hayudin.hasnanda97@gmail.com; <sup>2</sup>dolly.indra@umi.ac.id; <sup>3</sup>ramdan.satra@umi.ac.id  
\*corresponding author

## INFORMASI ARTIKEL

Diterima : 14 – 05 – 2020  
Direvisi : 23 – 05 – 2020  
Diterbitkan : 30 – 05 – 2020

Kata Kunci:  
Analisis Perbandingan  
*Failover Cluster*  
*Apache*  
*Platform Nginx*

## ABSTRAK

Dalam penerapannya, data dan informasi disimpan di suatu layanan yang disebut *server*. Salah satu sarana untuk mendapatkan data dan informasi yaitu menggunakan web, dimana aktivitas pelayanannya ditanggapi dan diproses oleh *web server*. Terdapat dua *web server* berbasis *open source* yang biasa digunakan yaitu *Apache* dan *Nginx*. Kinerja *web server* dan database sebagai media penyedia konten diharapkan selalu dapat memenuhi semua kebutuhan dari pengguna. Penelitian ini bertujuan untuk membandingkan ketersediaan layanan data *server* menggunakan teknik *failover cluster* pada *web server Apache* dan *Nginx* berdasarkan waktu saat *load* atau migrasi *server*. Hasil penelitian ini dengan melakukan perbandingan waktu saat *load* sistem atau perpindahan layanan *server* bahwa *web server Nginx* memiliki kualitas layanan lebih baik dari pada *web server Apache*. Melalui skenario pengujian yang telah dilakukan pada mesin *Apache* menghasilkan total rata-rata waktu kecepatan perpindahan layanan sebesar 25,82 s sedangkan pada mesin *Nginx* menghasilkan total rata-rata waktu kecepatan perpindahan layanan sebesar 22,98 s.

This is an open access article under the [CC-BY-SA](#) license.



## I. Pendahuluan

Penerapan *web server* dalam dunia perkantoran belakangan ini menjadi hal yang cukup penting, mengingat saat ini perkembangan aplikasi-aplikasi pendukung untuk perkantoran sudah mulai bergeser dari aplikasi konvensional (*Desktop-Based Application*) menuju ke aplikasi berbasis web (*Web-Based Applications*) [1]. *Web server* merupakan *software* yang memberikan layanan berbasis data yang berfungsi menerima permintaan dari HTTP atau HTTPS pada web browser dan mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web atau berbentuk dokumen HTML [2]. Terdapat dua *web server* berbasis *open source* yang biasa digunakan yaitu *Apache* dan *Nginx*. Kinerja *web server* dan database sebagai media penyedia konten diharapkan selalu dapat memenuhi semua kebutuhan dari pengguna [3]. Jika tidak ditanggapi dengan serius, ini bisa saja berakibat pada *server-server* yang kelebihan beban permintaan (request) dari pengguna [4]. Hal ini disebabkan permintaan dari pengguna lebih besar dari kemampuan *server* untuk memberikan layanan [5]. Dengan banyaknya yang mengakses suatu web akan berdampak pada beban *web server* [6]. Kemudian dari pada itu, kinerja *server* akan berkurang ketika melayani permintaan dari berbagai client karena pelayanan data untuk *client* hanya terpusat disatu *server*. Hal ini disebabkan permintaan dari pengguna lebih besar dari kemampuan *Server* untuk memberikan layanan. Oleh karena itu, pemilik website akan kebingungan dalam memilih layanan *Server* yang tepat. Salah satu solusi untuk menangani masalah tersebut adalah dengan menggunakan sebuah *server* yang andal dengan performa yang tinggi [6]. Selain itu dibutuhkan *web server* yang efektif untuk mendukung teknik *failover* serta perlu dibangun suatu *high availability web server cluster* yang dapat menjalankan teknik *failover*, sehingga dapat meminimalisir resiko kegagalan *service* pada *web server* yang dapat menghambat efektifitas kerja penyedia layanan atau *server* [1].

Penelitian serupa dilakukan oleh Intan Ferinja (2017) yang menganalisis dan membandingkan kinerja dari kedua *Web server* tersebut yaitu *Apache* dan *Nginx*, sehingga pengguna dapat memilih *Web Server* yang terbaik. Penulis hanya membandingkan parameter *throughput*, koneksi, *request*, *reply* dan *error* dengan memberikan beban pada setiap pengujian dan dilakukan pada atribut-atribut yang ada pada *beritalinux.com* secara virtual. Setelah dilakukan pengujian didapatkan hasil dimana dalam merespon dan menghubungkan data yang diminta oleh *client web application server Nginx* lebih unggul dari *apache*. Dari hasil tersebut, disarankan kepada admin sistem untuk menggunakan *web server Nginx* untuk kinerja website yang lebih baik [3].

## II. Metode

Metode penelitian yang diterapkan berisi tahapan atau prosedur penelitian dan algoritma yang digunakan dalam penelitian, formula permasalahan yang diteliti dengan lebih rinci, serta perancangan sistem jika dibutuhkan. Jenis penelitian ini adalah penelitian observasi dan percobaan. Sebagai metode penelitian yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap yang lain dalam kondisi yang terkendali [7]. Eksperimen merupakan modifikasi kondisi yang dilakukan secara sengaja dan terkontrol dalam menentukan peristiwa atau kejadian, serta pengamatan terhadap perubahan yang terjadi pada peristiwa itu sendiri [8].

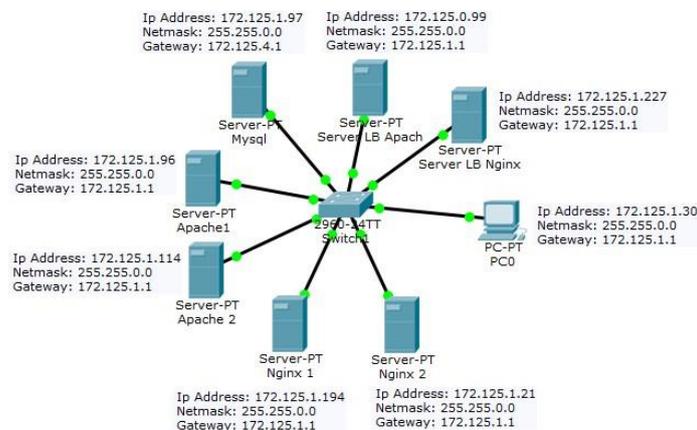
### a. Variabel yang diteliti

Variabel yang diteliti dalam objek penelitian sistem adalah

1. Waktu kecepatan perpindahan layanan. *Response Time* disini mengacu pada seberapa cepat suatu sistem/aplikasi dapat merespon suatu aksi. *Throughput* menunjukkan banyaknya tugas yang dapat diselesaikan dalam satu satuan waktu. Misalnya web server harus bisa melayani minimal 10 pengunjung dalam satu detik [9].
2. Ketersediaan data, proses pengiriman sekaligus penerimaan suatu data dan informasi melalui dua atau lebih perangkat yang tersedia.
3. Hasil nilai uji waktu kecepatan perpindahan layanan.

### b. Sistem Usulan

Sistem ini dibangun untuk membandingkan layanan yang diberikan oleh *web server Apache* dan *Nginx* menggunakan *failover clustering* yang merupakan salah satu solusi untuk menangani kegagalan *server* misalnya ketika salah satu *server* mati total atau tidak bisa memberikan layanan maka *server* lain yang akan mengambil alih fungsi dari *server* yang mati [9]. Sehingga komputer *client* tidak mengetahui jika terjadi kegagalan *server*, karena proses yang akan dilakukan oleh *server* yang gagal atau mati akan dilanjutkan oleh *server* cadangan [10]. Berdasarkan hasil observasi dan pengumpulan data ditempat penelitian dan masalah yang telah dikemukakan, maka penulis akan melakukan percobaan dengan membangun sistem yang dimaksud.



Gambar 1. Performansi sistem yang diusulkan

### a. Kebutuhan Perangkat

#### i. Perangkat Keras (*Hardware*)

1. Laptop LENOVO G40
2. Processor Intel Inside
3. RAM 4 GB
4. Switch
5. 2 PC yang akan menjalankan *Apache* dan PC yang akan menjalankan *Nginx*
6. 1 PC untuk Database
7. 1 PC yang akan menjalankan load balancing *Apache* dan 1 PC yang akan menjalankan load balancing *Nginx*

#### ii. Perangkat Lunak (*Software*)

1. Sistem operasi Windows 10.

2. XAMPP v3.2.2
3. Web Server Apache 2.4.41 dan Nginx 1.17.6
4. OS Linux Ubuntu versi 16.04
5. Browser Chrome versi 77.0.3865.120
6. Haproxy 4.5.5
7. Putty 0.73

Tabel 1. Perbandingan Total Average dari mesin Apache dan Nginx

Mesin	Average Waktu Migrasi Server (dtk)			
	Skenario 1	Skenario 2	Skenario 3	Skenario 4
Apache	6,898	10,044	6,472	9,634
Nginx	6,184	9,376	5,064	9,429

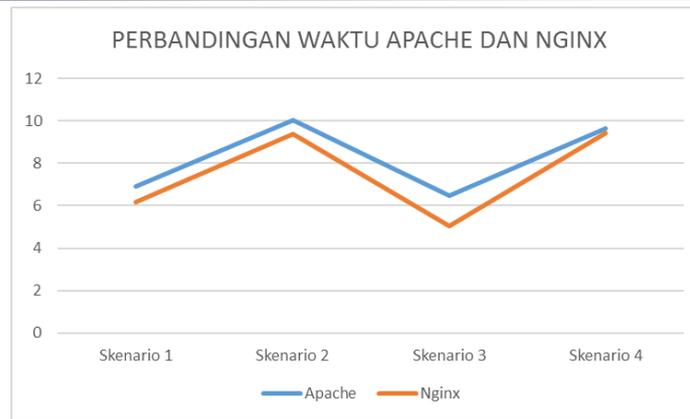
### III. Hasil dan Pembahasan

Pengujian sistem ini dengan melakukan konfigurasi untuk mengetahui waktu kecepatan perpindahan layanan dan ketersediaan data saat *load* atau migrasi *server*. Saat pengujian, akan dilakukan pengamatan kepada kedua *web server* tersebut. Salah satu *server* yang aktif memberikan layanan akan dimatikan. Kemudian mengecek ketersediaan data saat *load* sistem dan menghitung hasil nilai uji waktu kecepatan perpindahan layanan.

Beberapa skenario pengujian yang dilakukan sebagai berikut:

1. Pencabutan Kabel Power  
Pengujian dilakukan dengan mencabut kabel *power* pada mesin *server* (Apache 1 dan Nginx 1) bersamaan dengan itu dihitung waktu kecepatan perpindahan atau migrasi layanan *server* yang diberikan kepada mesin (Apache 2 dan Nginx 2) menggunakan *stopwatch* dan mengecek apakah sistem *failover* berhasil dilakukan yang diketahui melalui aplikasi *Haproxy* dan pengaksesan database. Demikian juga sebaliknya akan dilakukan pengujian dengan pencabutan kabel *power* pada mesin (Apache 2 dan Nginx 2).
2. Pencabutan Kabel Jaringan.  
Pengujian dilakukan dengan mencabut kabel jaringan pada mesin (Apache 1 dan Nginx 1) bersamaan dengan itu dihitung waktu kecepatan perpindahan atau migrasi layanan *server* yang diberikan kepada mesin (Apache 2 dan Nginx 2) menggunakan *stopwatch* dan mengecek apakah sistem *failover* berhasil dilakukan yang diketahui melalui aplikasi *Haproxy* dan pengaksesan database. Demikian juga sebaliknya akan dilakukan pengujian dengan pencabutan kabel jaringan pada mesin Apache 2.
3. Mematikan Servis Mesin  
Pengujian dilakukan dengan mematikan *service* pada mesin (Apache 1 dan Nginx 1) menggunakan perintah terminal dari sistem operasi *Linux Ubuntu Desktop* 16.04. Kemudian bersamaan dengan itu dihitung waktu kecepatan perpindahan atau migrasi layanan *server* yang diberikan kepada mesin (Apache 2 dan Nginx 2) menggunakan *stopwatch* dan mengecek apakah sistem *failover* berhasil dilakukan yang diketahui melalui aplikasi *Haproxy* dan pengaksesan database. Demikian juga sebaliknya akan dilakukan pengujian dengan mematikan *service* pada mesin (Apache 2 dan Nginx 2).
4. *Disable Ethernet Network*  
Pengujian dilakukan dengan *disable ethernet network* pada mesin (Apache 1 dan Nginx 1) menggunakan perintah terminal dari sistem operasi *Linux Ubuntu Desktop* 16.04. Kemudian bersamaan dengan itu dihitung waktu kecepatan perpindahan atau migrasi layanan *server* yang diberikan kepada mesin (Apache 2 dan Nginx 2) menggunakan *stopwatch* dan mengecek apakah sistem *failover* berhasil dilakukan yang diketahui melalui aplikasi *Haproxy* dan pengaksesan database. Demikian juga sebaliknya akan dilakukan pengujian dengan *disable ethernet network* pada mesin (Apache 2 dan Nginx 2).

Pada tahapan analisa akhir ini diperoleh dari skenario pengujian yang telah dilakukan pada mesin Apache dan Nginx. Berdasarkan pengujian tersebut kedua mesin sama-sama memberikan layanan ketika salah-satunya terjadi *down*. Setelah itu diambil nilai total *average* waktu perpindahan layanan dari masing-masing pengujian untuk dilakukan perbandingan yang disajikan pada tabel 1.



Gambar 3. Grafik Perbandingan Mesin Apache dan Nginx

Dengan melihat Gambar 3 maka diperoleh hasil perbandingan mesin Apache dan Nginx. Grafik pada mesin Nginx melalui skenario-skenario yang telah dilakukan memiliki waktu lebih rendah dari pada waktu pada grafik mesin Apache, dimana ketika waktu itu semakin rendah berarti semakin cepat saat terjadi perpindahan layanan atau migrasi. Dengan demikian dapat diketahui hasil perbandingan waktu saat load sistem atau perpindahan layanan server bahwa mesin Nginx memiliki kualitas layanan lebih baik dari pada mesin Apache. Perbedaan waktu kecepatan perpindahan layanan ini, dimana Nginx lebih unggul dari pada Apache dipengaruhi oleh sistem kerja dari kedua web server tersebut. Web server Apache bekerja hanya pada satu thread menggunakan konsep based on process, dimana web server Apache hanya akan bekerja atau memproses suatu pekerjaan saat ada permintaan. Sedangkan web server Nginx menggunakan konsep kerja multitasking, dimana web server Nginx akan bekerja atau memberikan layanan secara bersamaan dengan satu thread sehingga mengkonsumsi RAM lebih rendah dan respon waktu lebih cepat dalam memberikan layanan dibandingkan web server Apache.

#### IV. Kesimpulan

Kedua platform web server yaitu Apache dan Nginx dapat meningkatkan ketersediaan layanan setelah di implementasikan dengan melakukan failover, dimana ketika salah satu server mengalami down maka server yang lain dapat aktif memberikan layanan. Pada penelitian ini, diperoleh hasil perbandingan kualitas layanan dari kedua web server berdasarkan waktu saat load atau waktu kecepatan migrasi layanan server. Web server Nginx memiliki waktu migrasi layanan lebih cepat dengan total rata-rata waktu kecepatan perpindahan layanan sebesar 22,98 s dibandingkan dengan web server Apache dengan total rata-rata waktu kecepatan perpindahan layanan sebesar 25,82 s. Penelitian selanjutnya, diharapkan dapat melakukan pengujian dengan menggunakan parameter yang berbeda dan menggunakan lebih dari satu database. Dapat menerapkan hasil penelitian ini pada infrastruktur produksi pada suatu instansi atau organisasi.

#### Daftar Pustaka

- [1] A. Hakim, M. Ridha, S. Heristian, A. Selawati, and P. Paramita, "Impelementasi Failover Clustering Server untuk mengurangi Resiko Downtime Pada Web Server," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [2] Hastanti, Rulia Puji, and Bambang Eka Purnama. "Sistem penjualan berbasis web (e-commerce) pada tata distro kabupaten pacitan." *Bianglala Informatika* 3.2, 2015.
- [3] Ferina Intan, Zulhendra, Efrizon, "Analisis Perbandingan Kinerja Web Server Apache dan Nginx Menggunakan Httpperf Pada Portal Berita", *Jurnal Vokasional Teknik Elektronika & Informatika* Vol. 5, No. 2, 2017.
- [4] Schroeder, Bianca, and Mor Harchol-Balter. "Web servers under overload: How scheduling can help." *ACM Transactions on Internet Technology (TOIT)* 6.1, 2006, pp. 20-52.
- [5] F. Adnan and Kusnawi, "Analisis Perbandingan Performa Web Server Apache dan Nginx menggunakan Httpperf pada VPS dengan Sistem Operasi CentOS," *Stmik Amikom Yogyakarta*, p. 6, 2016.
- [6] R. Dani and F. Suryawan, "Perancangan dan Pengujian Load Balancing dan Failover Menggunakan NginX," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 3, no. 1, p. 43, 2017.
- [7] Sugiyono. 2006. *Metode Penelitian Kuantitatif, Kualitatif dan R & D*. Bandung: Alfabeta. A. Susilo, *Administrasi Server 1.pdf*, Edisi Pert. Jakarta: IMrs Ghost1, SMK Negeri 2 Kalianda, 2013.
- [8] Ali, Muhammad. 1993. *Penelitian Kependidikan Prosedur dan Strategi*. Bandung: Angkas.
- [9] Arsa, Wikranta, and Khabib Mustofa. "Perancangan dan Analisis Kinerja Private Cloud Computing dengan Layanan Infrastructure-As-A-Service." *BIMIPA* 24.3: 225-237, 2014.

- 
- [10] Wijaya, Miftakhudin Kusuma, Zamah Sari, and Mahar Faiqurahman. "Implementasi High Availability Cloud Storage Dengan Metode Replikasi Dan Failover Pada Laboratorium Teknik Informatika." *Jurnal Repositor 2.2* (2020), pp.165-176.