

SIMULASI LOAD BALANCING SERVER BLOG UMY MENGGUNAKAN NGINX WEB SERVER

NASKAH PUBLIKASI

Untuk memenuhi sebagian persyaratan

Mencapai derajat Sarjana S-1



Diajukan oleh

Nurbawono Auliajati

20130140076

Kepada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH YOGYAKARTA

2018

**SIMULASI LOAD BALANCING SERVER BLOG UMY
MENGUNAKAN NGINX WEB SERVER
(STUDI KASUS BIRO SISTEM INFORMASI UNIVERSITAS
MUHAMMADIYAH YOGYAKARTA)**

Oleh: Nurbawono Auliajati [1], Eko Prasetyo [2], Reza Giga Isnanda [3]

[1] Mahasiswa Program Studi Teknik Informatika, Fakultas Teknik, Universitas
Muhammadiyah Yogyakarta

[2] Dosen Program Studi Teknik Informatika, Fakultas Teknik, Universitas
Muhammadiyah Yogyakarta

[3] Dosen Program Studi Teknik Informatika, Fakultas Teknik, Universitas
Muhammadiyah Yogyakarta

INTISARI

Universitas Muhammadiyah Yogyakarta (UMY) merupakan salah satu universitas swasta terbaik di Indonesia dan memiliki mahasiswa lebih dari 10.000 orang. Mahasiswa yang begitu banyak tentu harus didukung oleh tenaga karyawan kampus serta fasilitas yang memadai, sehingga pelayanan terhadap kebutuhan mahasiswa terpenuhi. Layanan teknologi informasi juga disediakan untuk mendukung kegiatan operasional di UMY, baik dari segi layanan *network infrastruktur* sampai dengan *service infrastruktur* seperti blog, KRS, sistem kepegawaian, dan lainnya. Blog UMY menjadi salah satu sarana yang sangat penting, dimana blog ini menjadi wadah para mahasiswa/i serta staf dan dosen UMY untuk mengumpulkan artikel, tulisan, maupun tugas perkuliahan. Maka dari itu blog sudah selayaknya memiliki *performance* yang mumpuni dalam *handle* kegiatan sehari-hari di universitas. akan tetapi seiring dengan meningkatnya *user* atau pengguna harian pada blog universitas, blog menjadi sering kewalahan atau tidak mampu melayani permintaan *user* yang membengkak,

maka dari itulah dilakukan penelitian yang dimana akan memanfaatkan Nginx sebagai *web server default* yang memang sudah digunakan oleh blog tersebut. Implementasi *load balance* sendiri telah dilakukan oleh staf Biro Sistem Informasi (BSI) selaku pihak yang bertanggung jawab terhadap sarana universitas, namun metode yang digunakan belumlah optimal, bisa dilihat dari masih seringnya blog tidak bisa diakses pada jam-jam padat perkuliahan. untuk itu penulis disini telah melakukan serangkaian ujicoba dimana *load balance* akan digunakan dari sisi *web server*.

Kata Kunci: *Nginx, Load Balance, Server, User, System, web server, blog, network*

LOAD BALANCING UMY BLOG SIMULATION WITH NGINX WEB SERVER

(CASE STUDY: INFORMATION SYSTEM AGENT OF UNIVERSITY MUHAMMADIYAH OF YOGYAKARTA)

By: Nurbawono Auliajati [1], Eko Prasetyo [2], Reza Giga Isnanda [3]

[1] Student of Informatics Engineering, Faculty of Engineering, Muhammadiyah
University of Yogyakarta

[2] Lecture of Informatics Engineering, Faculty of Engineering, Muhammadiyah
University of Yogyakarta

[3] Lecture of Informatics Engineering, Faculty of Engineering, Muhammadiyah
University of Yogyakarta

ABSTRACT

The Muhammadiyah University of Yogyakarta (UMY) is one of the best universities in Indonesia and has a student more than 10,000 people. So many students certainly should be supported by employees of the campus as well as adequate facilities, so that the service to the needs of students are met. Information technology

services are also provided to support their operations in UMY, both in terms of network infrastructure to service with service infrastructure such as blogs, KRS, the staffing system, and more. UMY blog become one means very important, where this blog into a container of the students as well as staff and lecturers UMY to collect articles, writings, or task and associated costs. Thus, the blog should already have qualified in handling big performance of daily activities at the University. but with increasing users or users daily on his blog, the blog became often overwhelmed or incapable of serving the user demand swelled, then from that conducted research which will utilize the Nginx as the default Web server that was already used by the blog. Implementation of load balance itself has been carried out by the staff of the Bureau of information systems as a responsible party means a University, but the method used is far from optimal, can still be seen from often inaccessible on the blog solid hours of coursework. for that the author here has done a series of tests where the load balance will be used from the web server.

Keywords: *Nginx, Load Balance, Server, User, System, web server, blog, network.*

A. PENDAHULUAN

Universitas Muhammadiyah Yogyakarta (UMY) merupakan salah satu universitas swasta terbesar di Yogyakarta. Berdasarkan wawancara dengan kepala urusan program Biro Sistem Informasi (BSI) UMY, yang memiliki jumlah mahasiswa aktif pada tahun 2016 mencapai 18.444 mahasiswa. Dengan jumlah mahasiswa yang begitu banyak, sudah selayaknya UMY memiliki sejumlah fasilitas pendukung kinerja baik staf maupun mahasiswa/i, salah satunya yaitu media blog.

Media blog universitas (Wordpress) berguna sebagai sarana pendukung pembelajaran yang dimanfaatkan para mahasiswa/i UMY sebagai media pengumpulan tugas baik itu artikel maupun *sharing data*. Selain itu, blog juga dipergunakan untuk memposting hasil-hasil karya tulis ilmiah ataupun hasil dari lomba-lomba dan kegiatan lainnya. Membicarakan *sharing data*, blog Universitas Muhammadiyah Yogyakarta (UMY) juga dipergunakan sebagai media tempat berbagi artikel yang dapat diunduh dalam bentuk pdf (*sharing data*) maupun dibaca secara *online*. Sehingga blog UMY memiliki peranan penting dalam mendukung para mahasiswa/i maupun dosen dan staf universitas lainnya dalam proses perkuliahan. Dalam melakukan kegiatan keseharian di Universitas Muhammadiyah Yogyakarta (UMY).

Akan tetapi, seiring dengan meningkatnya jumlah mahasiswa-mahasiswi di Universitas Muhammadiyah Yogyakarta (UMY), blog yang dikelola oleh Biro Sistem Informasi (BSI) di UMY sering kewalahan dalam menampung jumlah permintaan atau *request* dari *client* blog itu sendiri, baik dari kalangan mahasiswa/i maupun dari kalangan staf akademisi UMY.

Masalah selanjutnya yang muncul adalah banyaknya *spam* atau *backlink spam* yang masuk ke dalam *database resources server* blog, sehingga mengakibatkan terkurasnya jumlah *bandwidth* yang mampu di *handle* server. Upaya penanganan telah dilakukan diantaranya dengan mengupdate *operating system* server yang sudah *out of date*, memperbaiki beberapa konfigurasi server dan web server, serta menghapus beberapa *spam* dan *backlink spam* pada wordpress dan *database server*. Akan tetapi dikarenakan jumlah *spam* pada Wordpress dan *database server* sudah tergolong sangat banyak, maka sangat tidak mungkin menghapus satu per satu *spam* yang ada.

Maka dari itu, untuk menangani masalah tersebut, penulis merancang dan mengembangkan *load balance* pada *server* yang baru dengan konfigurasi yang *up to date*, yaitu dengan menerapkan *Nginx* (baca: Engine X) *loadbalance* yang diimplementasikan pada *network*, *server*, *web server* (*Nginx*), *Wordpress*, serta *database server* yang baru.

B. METODE

Dalam penelitian ini metode yang digunakan adalah metode PPDIIO (*Prepare and Plan, Design, Implementation, Operate and Optimize*). PPDIIO merupakan metode yang digunakan *vendor Cisco system (Cisco Proprietary)* untuk membangun *network topology*.

Terdapat beberapa alasan mengapa memilih menggunakan metode PPDIIO dalam mengembangkan *server load balance* di UMY, antara lain:

1. Dapat menentukan biaya oprasional jika harus membangun dari awal
 2. Mampu memperkirakan ketersediaan *resource* yang ada
 3. Mampu memperkirakan jumlah dan kapasitas *user*
 4. Topologi yang akan digunakan akan lebih terstruktur
 5. *Maintenance* atau *troubleshooting* yang lebih mudah karena topologi yang terstruktur
 6. Waktu yang digunakan dalam menyelesaikan penelitian bisa diperhitungkan
- Meningkatkan kecepatan akses kepada *user* layanan



Gambar 4.26 Backend Server 2

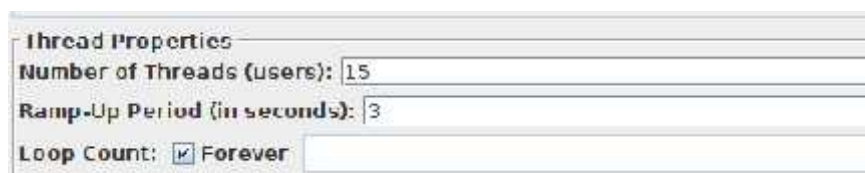


Gambar 4.27 Backend Server 3

Terlihat dari ketiga gambar yaitu, gambar 4.25, 4.26, dan 4.27, *client* mengakses *load balancer server* pada waktu yang berbeda dan mendapatkan *backend server* yang berbeda pula, hal ini menunjukkan kerja *load balance* telah berhasil.

2. DOS/DDOS Attack Simulasi

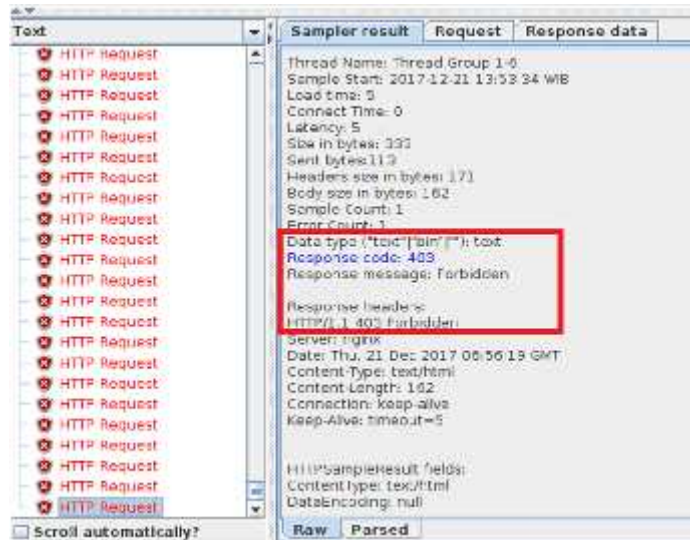
Pengujian dengan cara *DOS attack* atau dengan simulasi DOS ini bertujuan untuk mengetahui tingkat ketahanan *server* dalam *handle* atau menerima jumlah *request* dari pengguna layanan blog UMY, serta menguji beberapa *module* konfigurasi yang digunakan dalam mengatasi *request* atau permintaan terhadap layanan blog dari *client* yang tidak wajar atau dalam jumlah yang banyak dalam waktu yang singkat. Pada simulasi DOS ini jumlah dari *client* yang akan melakukan *request* terhadap *web server* telah ditentukan yaitu sebanyak 15 *user* (Terlihat pada gambar 4.28) dikarenakan jumlah *user* ini dirasa telah cukup dalam melakukan simulasi DOS skala kecil berdasarkan parameter *module* pencegahan DOS pada gambar 4.19. Sedangkan waktu yang dibutuhkan untuk melakukan *reload request* sebanyak 3 detik (Terlihat pada gambar 4.28) sekali juga dirasa cukup dikarenakan telah melampaui batas yang ditentukan pada *module limit_conn* dan *limit_req* sebelumnya yaitu pada gambar 4.19.



Gambar 4.28 Pengaturan jumlah user/request DDOS

Hasil dari pengujian ini akan terlihat pada gambar 4.29, dimana pada gambar tersebut terlihat Nginx melakukan penolakan atau *rejecting* terhadap proses *http request* yang tidak normal atau melebihi kapasitas yang sudah ditentukan. Pada jumlah permintaan yang meningkat, layanan blog masih dapat berjalan dikarenakan *request* telah ditolak dengan *http respon 403 (forbidden)* dikarenakan telah membatasi jumlah

request per second yang akan diterima oleh *web server* Nginx, dengan cara menambahkan module *limit_conn* seperti yang terlihat pada gambar 4.19.



Gambar 4.29 HTTP Request DOS



Gambar 4.30 Nginx Connection Request Result

Pada gambar 4.30 terlihat juga Nginx *connection* yang terhubung serta Nginx *request* yang terhubung, terlihat dari Nginx *request* yang begitu tinggi pada jam 10:15 sampai dengan 10.58 waktu percobaan ini dilakukan, yang menandakan telah terjadinya *request* atau permintaan terhadap layanan blog yang tinggi pada waktu tersebut.

3. Mencegah DOS/DDOS Attack dengan *Limit Connection*

Limit connection dan *limit request connection* pada Nginx dapat dipergunakan untuk membatasi jumlah akses *user* terhadap *web server*, hal ini diperuntukan untuk menjaga agar *web server* Nginx tidak terjadi *overload request* dan memastikan Nginx tetap

berkerja sebagai mana mestinya. Ada beberapa *rules* yang akan diterapkan pada *module limit_conn* dan *limit_req* yang akan membatasi akses terhadap *web server* seperti pada gambar 4.31.

```
#charset koi8-r;
access_log /var/log/nginx/access.log main;
error_log /var/log/nginx/error.log;
location / {
    limit_conn backend1_limit_conn 1;
    limit_req zone=mybackend1 burst=10 nodelay;
    limit_req_status 403;
    try_files $uri $uri/ /index.php?$args;
}
```

Gambar 4.31 Nginx *Limit Connection and Request*

Seperti yang terlihat pada gambar 4.31, *limit connection (limit_conn)* yang terapkan pada *directory root path* dari Nginx bernama *backend 1* dengan *limit connection 1*, sedangkan pada *limit request (limit_req)* bernama *mybackend1* dan hanya akan mampu melayani jumlah dari *request* sebanyak 10 *request* per detik tanpa jeda, dan apabila melampaui ketetapan, maka akan ditampilkan status *forbidden* seperti gambar 4.29.

D. KESIMPULAN

Membahas mengenai kesimpulan, alangkah baiknya terlebih dahulu membahas mengenai *security issue* yang terjadi pada blog UMY. Pada saat pertama kali akan memulai mengerjakan *load balance* pada blog UMY di BSI UMY, langkah pertama yaitu dengan mengidentifikasi masalah yang sering dirasakan para mahasiswa/i di UMY ketika berada dalam masa aktif perkuliahan atau ketika akan menggunakan blog UMY.

Setelah dilakukan analisa, ternyata terdapat beberapa masalah yang menjadi penyebab timbulnya masalah-masalah pada blog UMY, diantaranya:

1. Reputasi *ip address* UMY yang buruk di *internet*, hal ini disebabkan oleh banyaknya *malware*, *botnet*, ataupun *scam link* pada jaringan internal UMY yang mengarah ke *internet*.
2. Banyaknya *spam* baik *back-link spam* maupun *comment spam* yang terdapat pada *database* blog UMY dan juga pada Wordpress yang digunakan pada blog UMY.

Terdapat *security issue* lainnya seperti *bug* atau *vulnerability* pada Wordpress, *database*, serta *security issue* yang terkait dengan *DNS hijack*, *malware*, *botnet*, dan juga *security header website* blog UMY.

Oleh karena itu, berdasarkan masalah yang berhasil ditemukan selama penelitian, dapat diambil kesimpulan pertama, bahwa blog UMY sering tidak dapat diakses jika pada jam-jam padat perkuliahan adalah, karena disebabkan meningkatnya ukuran *database* yang disebabkan oleh banyaknya *spam* yang masuk melalui Wordpress *comment* pada blog, serta terdapat beberapa *spam* yang disebabkan oleh *malware*.

E. SARAN

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan sebagai acuan dalam penelitian selanjutnya, yaitu:

1. Lakukanlah *hardening* dan bila perlu sewalah seorang *penetration tester* yang benar-benar telah berkompeten dan telah teruji.
2. Lakukanlah pengamanan baik dari segi program maupun aplikasi.

3. Lakukanlah *maintenance* secara berkala terhadap *software*, dan *system* yang telah dibangun.
4. Edukasi *user* atau pengguna layanan terhadap masalah IT *security* atau apapun yang berkaitan dengan keamanan data atau privasi.
5. Sebagai seorang pengelola pada BSI UMY yang melakukan *maintenance* terhadap *system* yang berjalan pada universitas hendaklah memiliki perhatian terkait dengan *issue* IT baik teknologinya, *software*, maupun *security issue* yang terjadi.

Untuk menghindari hal-hal yang tidak diinginkan seperti penyebaran *botnet*, *malware* maupun sampai pada kasus peretasan terhadap website maupun server UMY, sangat disarankan agar universitas membentuk kelompok yang benar-benar fokus terhadap layanan teknologi yang digunakan pada saat ini.

F. DAFTAR PUSTAKA

- [1] A. Rahmatulloh and F. MSN, "Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi," Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi, vol. 03, pp. 241–248, Aug. 2017.
- [2] D. Sharma and V. B. Aggarwal, "Performance Evaluation Of Dynamic Load Balancing System By Using Number Of Effective Parameters," vol. 8, pp. 678–681, Aug. 2017.

[3] N. B. Umkabu, B. Erfianto, and E. Ariyanto, “Analisis Performansi Dispatcher Load Balancing Pada Web Server Menggunakan Algoritma Least Connection,” Telkom University. 2010.

[4] A. B. Noviyanto, E. K. N, and A. Hamzah, “Perancangan dan Implementasi Load Balancing Reverse Proxy Menggunakan HAProxy Pada Aplikasi Web,” vol. 3, pp. 21–31, Dec. 2015.

[5] J. Ellingwood, Apache vs Nginx: Practical Considerations, 28-Jan-2015.

[Online]. Available: <https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>. [Accessed: 03-Jan-2018].

[6] NGINX Inc., “NGINX vs. Apache: Our View of a Decade-Old Question,” NGINX, 03-Aug-2017. [Online]. Available: <https://www.nginx.com/blog/nginx-vs-apache-our-view/>. [Accessed: 03-Jan-2018].

[7] NGINX Inc., “Why Use NGINX?,” Web Server Load Balancing with NGINX Plus. [Online]. Available: https://www.nginx.com/resources/wiki/community/why_use_it/. [Accessed: 04-Jan-2018].

[8] A. Lerner, J. Skorupa, and D. Cisco, Magic Quadrant for Application Delivery Controllers, 29-Aug-2016. [Online]. Available: <https://www.gartner.com/doc/3426420/magic-quadrant-application-delivery-controllers>. [Accessed: 04-Jan-2018].

[9] I. Rijayana, “Teknologi Load Balancing Untuk Mengatasi Beban Server,” in Seminar Nasional Aplikasi Teknologi Informasi, 2005, pp. 35–39.

- [10] Supramana and I. G. L. P. Eka Prisma, “Implementasi Load Balancing Pada Web Server Dengan Menggunakan Apache,” vol. 5, pp. 117–125, 2016.
- [11] NGINX Inc., What Is Layer 4 Load Balancing? | NGINX Load Balancer. [Online]. Available: <https://www.nginx.com/resources/glossary/layer-4-load-balancing/>. [Accessed: 04-Jan-2018].
- [12] NGINX Inc., What is NGINX? [Online]. Available: <https://www.nginx.com/resources/glossary/nginx/>. [Accessed: 06-Jan-2018].
- [13] NGINX Inc, Module ngx_http_fastcgi_module. [Online]. Available: http://nginx.org/en/docs/http/ngx_http_fastcgi_module.html. [Accessed: 06-Sep-2017].
- [14] NGINX Inc, Using nginx as HTTP load balancer. [Online]. Available: http://nginx.org/en/docs/http/load_balancing.html. [Accessed: 12-Nov-2017].
- [15] KEMP Corporate HQ., Load Balancer. (n.d.). Load Balancing Algorithms. [online] Available at: <https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques/> [Accessed 26 Jan. 2018].
- [16] VMWare Inc., Virtualization Technology & Virtual Machine Software: What is Virtualization?, 16-Jan-2018. [Online]. Available: <https://www.vmware.com/solutions/virtualization.html>. [Accessed: 21-Jan-2018].
- [17] Proxmox Server Solutions GmbH, Open-Source Virtualization Platform. [Online]. Available: <https://www.proxmox.com/en/proxmox-ve>. [Accessed: 21-Jan-2018].

- [18] Proxmox Server Solutions GmbH. (2018). Compare Proxmox VE vs VMware vSphere, Hyper-V, XenServer. [online] Available: <https://www.proxmox.com/en/proxmox-ve/comparison> [Accessed 21 Jan. 2018].
- [19] Mitchell, B. (2017). What Is a Server in Computer Networking?. [online] Lifewire. Available at: <https://www.lifewire.com/servers-in-computer-networking-817380> [Accessed 11 Jan. 2018].
- [20] Hewlett Packard Enterprise. (2018). HPE ProLiant MicroServer Gen10. [online] Available: <https://www.hpe.com/us/en/product-catalog/servers/proliant-servers/pip.hpe-proliant-microserver-gen10.1009955118.html> [Accessed 21 Jan. 2018].
- [21] Pluralsight. (2017). Linux hardening: A 15-step checklist for a secure Linux server. [online] Available: <https://www.pluralsight.com/blog/it-ops/linux-hardening-secure-server-checklist> [Accessed 19 Aug. 2017].
- [22] Canonical Ltd., (2018). About Ubuntu. [online] Ubuntu.com Available at: <https://www.ubuntu.com/about/about-ubuntu> [Accessed 21 Jan. 2018].
- [23] GNU Nano., "nano Command Manual", Nano-editor.org. [Online]. Available: <https://www.nano-editor.org/dist/v2.2/nano.html>. [Accessed: 14- Oct- 2017].
- [24] The Php Group., "PHP: Migrating from PHP 5.6.x to PHP 7.0.x - Manual", Php.net. [Online]. Available: <http://php.net/manual/en/migration70.php>. [Accessed: 17- Nov- 2017].

- [25] Nginx Inc., "PHP-Fpm and FastCGI Example", github.com, 2017. [Online]. Available: <https://github.com/nginxinc/nginx-wiki/blob/master/source/start/topics/examples/phpfcgi.rst>. [Accessed: 20- Jan- 2018].
- [26] Wordpress., "About Us", WordPress.com, 2005. [Online]. Available: <https://wordpress.com/about/>. [Accessed: 21- Jan- 2018].
- [27] Tatham, S. (1997). PuTTY User Manual. [online] The.earth.li. Available at: <https://the.earth.li/~sgtatham/putty/0.70/html/putty070.html> [Accessed 21 Jan. 2018].
- [28] MariaDB Corporation Ab., "About MariaDB - MariaDB.org", MariaDB.org, 2018. [Online]. Available: <https://mariadb.org/about/>. [Accessed: 21- Jan- 2018].
- [29] Cisco Systems Inc., Cisco Unified Communications System for Contact Center. (2008). 7th ed. [ebook] San Jose, California, USA: Cisco Systems Inc, pp.1-122. Available at: https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/UC7-1-3/TIS/tis701cc.pdf [Accessed 22 Jan. 2018].
- [30] Mars.merhot.dk. (2018). CCNP TSHOOT 642-832/Chapter 1 - House of Technology. [online] Available at: http://mars.merhot.dk/w/index.php/CCNP_TSHOOT_642-832/Chapter_1 [Accessed 26 Jan. 2018].