

# Rancang Bangun Server *Learning Management System* Menggunakan *Load Balancer* dan *Reverse Proxy*

Sirajuddin, Achmad Affandi, dan Eko Setijadi

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

*E-mail*:: affandi@ee.its.ac.id

**Abstrak** — Internet yang semakin mudah dijangkau masyarakat ditandai dengan jumlah pengguna yang semakin besar. Seiring dengan hal tersebut trafik yang diterima *web server* juga terus meningkat dan menyebabkan beban kerja *web server* tersebut semakin berat. Terjadinya lonjakan trafik yang tidak mampu diatasi oleh *web server* menyebabkan *web server* tersebut mengalami kelebihan beban (*overload*) dan membuat *website* menjadi tidak dapat diakses. Masalah tersebut juga terjadi pada *server* situs *e-learning*.

Salah satu solusi untuk mengatasi masalah beban pada *web server* adalah dengan menggunakan lebih dari satu *web server*. Trafik internet akan dikendalikan oleh *load balancer* yang akan membagi *request* yang diterimanya ke masing-masing *web server*. Selain *load balancer* bisa juga dipasang *reverse proxy* untuk *cache* halaman *web* sehingga bisa dimuat lebih cepat lagi. Rancang bangun sistem yang diuji dalam tugas akhir ini meliputi dua *web server Learning Management System (LMS)* yang dikontrol oleh satu *load balancer* dan *reverse proxy (cluster)*. Pengujian juga dilakukan pada *server LMS single* untuk dijadikan tolok ukur perbandingan. Pengujian dilakukan untuk mengamati perbedaan kinerja yang dicapai *server LMS cluster* dibandingkan dengan *server LMS single*. Dari hasil pengujian didapatkan bahwa *server LMS cluster* mampu meningkatkan kinerja *website* ketika jumlah *user* yang mengakses berjumlah besar. Peningkatan tersebut terutama terlihat pada parameter *throughput*, *packet retransmission* dan *page load time* yang mampu meningkat sampai 57.93%.

**Kata kunci:** *e-learning*, *load balancer*, *packet retransmission*, *page load time*, *reverse proxy*, *through put*

## I. PENDAHULUAN

**W**EB SERVER sebagai komponen utama sebuah situs pembelajaran online (*e-learning*) seringkali mendapat beban yang semakin berat seiring dengan bertambahnya *user* yang mengaksesnya [1]. Dibutuhkan *web server* yang handal untuk mengatasi peningkatan jumlah *user* dan kompleksitas konten *e-learning* seperti penambahan bahan berupa kuis dan konten *video* yang semakin banyak dijumpai. Ketika

penambahan jumlah *user* dan konten tidak mampu diatasi oleh *web server* maka *web server* tersebut akan mengalami kelebihan beban (*overload*) dan situs *e-learning* tersebut akan *down*. Pemilihan sistem *e-learning* yang tepat akan meningkatkan skalabilitas dan ketahanan dari situs *e-learning* tersebut[2].

Ketika mengakses situs *e-learning* secara bersama-sama sedikit maka cukup dipakai satu *web server* saja, tetapi ketika jumlah mengakses bertambah besar maka ini menyebabkan terjadinya peningkatan *request* dari mengakses (*klien*) yang tidak mampu ditangani oleh *web server* sehingga mengakibatkan *web server* mengalami kelebihan beban. Permasalahan tersebut umumnya dihadapi oleh *web server* yang hanya memiliki mesin tunggal atau *single web server* [2]. Salah satu solusi untuk permasalahan tersebut adalah dengan menggunakan *load balancer* [3].

*Load balancing* adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi [4]. Salah satu jenis *load balancer* adalah *HaProxy*, yaitu sebuah software *load balancer* untuk aplikasi *TCP* dan *HTTP*, *HaProxy* dipilih karena memiliki kemampuan mengontrol trafik dari masing-masing *request* data dari *klien*, bukan hanya berdasarkan jumlah koneksi yang masuk, selain itu *HaProxy* dipilih karena terdapat fitur untuk menampilkan statistik dari penggunaan masing-masing *web server* yang dikontrolnya[5].

Pada penelitian sebelumnya[3], terdapat pembahasan tentang penggunaan *load balancer* untuk mengatasi trafik yang harus dilayani *web server*, tetapi hasil penelitian menunjukkan bahwa penggunaan *single server* membuat *website* lebih mampu melayani jumlah *user* yang lebih besar[3], selain itu penggunaan parameter *packet loss* untuk *QoS* sebuah aplikasi *TCP* adalah kurang tepat, karena setiap paket yang hilang atau tidak sampai ke *server* akan dikirim ulang lagi oleh protokol *TCP* tersebut. Tugas akhir ini dibuat dengan melakukan perbaikan dari penelitian sebelumnya, disamping juga menambah metode lain untuk mengatasi trafik pada *web server*.

Metode lain yang bisa digunakan adalah dengan cara *cache* data-data *static* seperti gambar dan teks kedalam sebuah *server* lain yang disebut *reverse proxy* [6]. *Cache* adalah

sebuah penyimpanan data sementara, sebagai contoh, ketika *user* mengakses sebuah halaman yang terdapat data berupa gambar maka gambar tersebut tidak perlu diambil dari *web server* tetapi langsung dikirim oleh *reverse proxy*, sehingga bisa akan mengurangi beban *web server*. Salah satu jenis *reverse proxy server* adalah Nginx, sebuah software bersifat modular[7]. Nginx dipilih karena karakternya yang ringan, bisa digunakan sebagai *server* khusus reverse proxy tanpa membebani *hardware* yang juga dipasang load balancer.

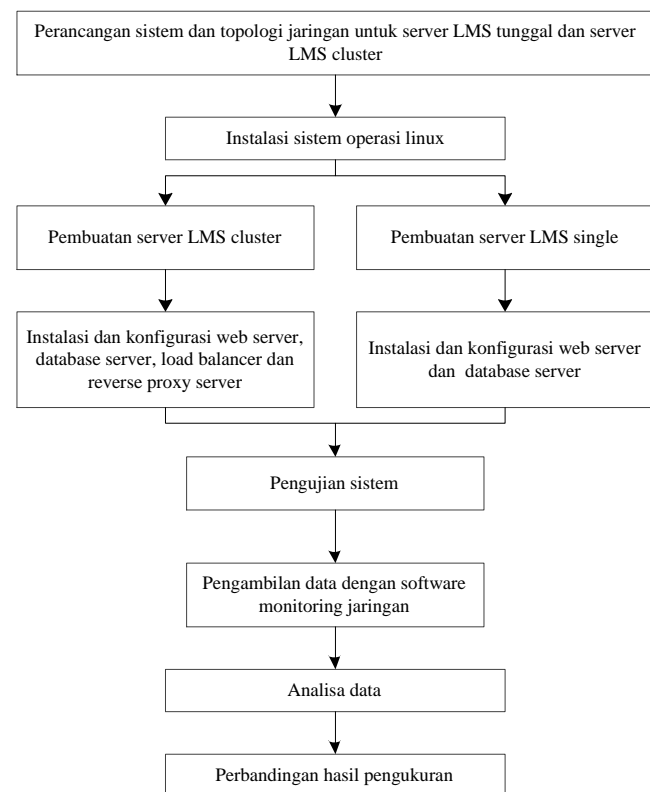
## II. URAIAN PENELITIAN

### A. Tahap Perancangan

Sistem yang akan dibangun terdiri dari dua bagian yaitu, *server LMS single* dan *server LMS cluster*, sistem dengan *server LMS single* terdiri dari bagian utama berupa *web server* sebagai tempat instalasi Moodle dan *database server*. Sistem *server single* terdiri dari satu komputer yang berperan sebagai *web server LMS*. *Web Server LMS* adalah *server* yang berfungsi sebagai tempat menyimpan data dari konten dan obyek pembelajaran klien. *Database server* berfungsi sebagai tempat pengaturan klien dengan datanya, manajemen *user* dsb. Klien adalah pengguna atau mengakses situs LMS yaitu mahasiswa (*student*) maupun pengajar (*teacher*).

Sistem kedua, *server LMS cluster* terdiri dari bagian-bagian berupa, dua *web server*, *database server*, *load balancer* dan *reverse proxy server*.

Perancangan sistem dan topologi jaringan untuk *server LMS single* dan *server LMS cluster* dimulai dari instalasi linux dan pembuatan masing-masing konfigurasi server. konfigurasi



Gambar. 1. Blok diagram perancangan dan pengujian sistem

untuk server LMS single meliputi Apache *web server* dan MySQL *database server* sebagai *database*-nya. Konfigurasi *server LMS cluster* berupa konfigurasi dua Apache *web server*, MySQL *database server*, HaProxy *load balancer* dan Nginx *reverse proxy server*.

Konfigurasi *server LMS single* dan *server LMS cluster* kemudian diuji dengan menggunakan alat uji dan parameter yang sama untuk kemudian diambil data hasil pengujian tersebut menggunakan software monitoring jaringan.

Hasil pengujian kedua konfigurasi server kemudian dibandingkan dan dianalisa untuk diketahui perbedaan kualitas situs *e-learning* ketika menggunakan kedua konfigurasi tersebut. Blok diagram perancangan dan pengujian sistem ada pada Gambar 1.

### B. Parameter Performansi

Parameter performansi jaringan dan kinerja server yang akan diamati adalah *throughput* dan *page load time*. *Throughput* adalah jumlah data/bit yang berhasil diterima tanpa ada kesalahan dalam satu detik dalam selang waktu pengamatan. *Throughput* dinyatakan dengan satuan bit per detik. *Page load time* adalah waktu yang dibutuhkan sebuah *web* untuk menampilkan halaman sebuah halaman *web* sampai selesai.

Tabel 1.  
Estimasi perbandingan jumlah *user* maksimal dan throughput antara *server LMS single* dan *server LMS cluster*

Konfigurasi server	Bandwidth (kbps)	Jumlah <i>user</i> maksimal	Throughput maksimal per <i>user</i> (kbps)
<i>Single</i>	256	262	0,41
	512	173	0,32
	1024	148	1,35
<i>Cluster</i>	256	617	0,18
	512	342	0,19
	1024	279	0,35

## III. HASIL DAN DISKUSI

Perbandingan nilai dari *throughput* dilakukan dengan membandingkan jumlah *user* maksimal yang mampu dilayani oleh masing-masing konfigurasi *server* beserta dengan *throughput* pada saat itu. Hasil dari perbandingan ditunjukkan pada Tabel 1.

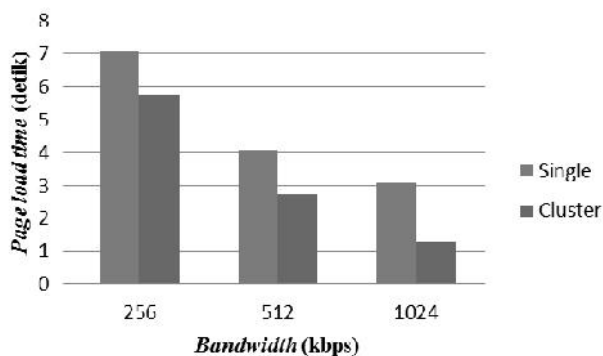
Dari Tabel 1 bisa diketahui bahwa penggunaan konfigurasi *cluster* mampu menambah jumlah *user* yang mampu dilayani secara bersamaan oleh *server LMS*. Ketika *server LMS single* pada bandwidth 256 kbps hanya mampu melayani 262 *user* secara bersamaan dengan throughput 0,41 kbps, *server LMS cluster* mampu melayani sampai 617 *user* dengan throughput 0,18 kbps. Hal yang sama terjadi pada bandwidth 512 dan 1024 yaitu 173 *user* dengan throughput 0,32 kbps pada *server single* dibandingkan dengan 342 *user* dengan throughput 0,19 kbps pada *server cluster*. Selanjutnya 148 *user* dengan 1,35 kbps pada *server single* dibandingkan dengan 279 *user* dengan 0,35 kbps pada *server cluster*.

Setelah dilakukan pengukuran *page load time* pada *server LMS*

single dan server LMS cluster, pada bagian ini hasil pengukuran page load time akan dibandingkan dan dikalkulasi peningkatannya. Hal ini ditunjukkan di Tabel 2 dan Gambar 2.

Tabel 2.  
Perbandingan page load time (detik) antar konfigurasi server

Konfigurasi server	Bandwidth (kbps)		
	256	512	1024
Single	7,11	4,06	3,09
Cluster	5,73	2,77	1,3
Penurunan	19,41 %	31,77 %	57,93 %



Gambar. 2. Perbandingan page load time antara server LMS single dengan server LMS cluster.

Dari grafik pada Gambar 2, terlihat bahwa page load time pada pengujian server LMS single lebih lama dibandingkan server LMS cluster. Penggunaan server LMS cluster mampu menurunkan page load time sebesar 19.41%, 31.77% dan 57.93% seperti yang terlihat pada kondisi ketika bandwidth yang dipakai adalah 256 kbps, 512 kbps dan 1024 kbps.

#### IV. KESIMPULAN

Berdasarkan hasil rancang bangun server Learning Management System (LMS) menggunakan load balancer dan reverse proxy, juga dari data-data yang didapatkan dari pengukuran performansi jaringan dan kinerja sistemnya, maka dapat diambil beberapa kesimpulan.

Konfigurasi server LMS menggunakan load balancer dan reverse proxy mampu memberi batas maksimal jumlah user yang mampu dilayani lebih besar dari pada server single, yaitu 262 user pada bandwidth 256 kbps, 173 user pada 512 kbps dan 148 user pada 1024 kbps, dibandingkan dengan server LMS cluster yang mampu melayani 617 user pada 256 kbps, 342 user pada 512 kbps dan 279 user pada 1024 kbps.

Hasil pengujian page load time menunjukkan efektifitas penggunaan reverse proxy untuk meng-cache halaman web, hal ini terbukti dari penurunan page load time 19.41% pada bandwidth 256 kbps, 31.77% pada bandwidth 512 kbps sampai 57.93% pada 1024 kbps ketika menggunakan konfigurasi server LMS dengan menggunakan load balancer dan reverse proxy saat memuat halaman web LMS yang tersusun dari gambar dan teks HTML.

#### DAFTAR PUSTAKA

- [1] Iskandar, M., "Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education", Springer Science+Business Media, New York, (2008).
- [2] Taryana, A., Siswanto, H., "Penerapan E-Learning OLAT Dengan Webserver Ter-cluster Untuk Peningkatan Kapasitas Akses E-Learning", Proceedings Seminar Nasional Aplikasi Teknologi Informasi (SNATI) 2010, Yogyakarta, Juni, (2010).
- [3] Pranata, A., "Rancang Bangun Server Learning Management System Dengan Metode Load Balancing", Tugas Akhir Jurusan Teknik Elektro FTI-ITS, (2011).
- [4] Rina, Marter, (2011, April). Pengertian Moodle. Available: <http://marter-rina-ilmukomputer.blogspot.com/2011/04/pengertian-moodle.html>
- [5] Willy, T., (2011, Agustus). HAProxy, The Reliable, High Performance TCP/HTTP Load Balancer. Available: <http://haproxy.1wt.eu/#desc>
- [6] Kew N. (2009, Desember). Running a Reverse Proxy in Apache. Available: <http://www.apachetutor.org/admin/reverseproxies>
- [7] Vpsnginx, (2012, Februari). Nginx Sebagai Web Server. Available: <http://vpsnginx.com/webserver/nginx/nginx-sebagai-web-server/>