

Implementasi Load Balancer Aktif Standby Menggunakan Protokol VRRP (Virtual Router Redudancy Protocol) Pada Web Server Kluster

Iwan^{1*}, Alan Adi Prastyo²

¹iwan@staff.gunadarma.ac.id

²alanprastyo95@gmail.com

ABSTRAK

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. Load balancing digunakan pada saat sebuah server telah memiliki jumlah user yang telah melebihi maksimal kapasitasnya. Load balancing juga mendistribusikan beban kerja secara merata di dua atau lebih komputer, link jaringan, CPU, hard drive, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal. bahwa loadbalance tidak akan menambah besar bandwidth yang kita peroleh, tetapi hanya bertugas untuk membagi trafik dari kedua bandwidth tersebut agar dapat terpakai secara seimbang. Ketika server atau jaringan kita diakses oleh banyak pengguna, maka disinilah keuntungan load balancing yang paling dirasakan. Atau ketika sebuah aplikasi yang sangat penting yang ada di sebuah server, tiba-tiba tidak bisa diakses karena server-nya mengalami gangguan, maka dengan adanya load balancing bisa dialihkan ke server lain. Dengan metode ini maka masalah yang selama ini terjadi yakni koneksi yang lambat ketika banyak user yang terkoneksi akan bisa teratasi, karena dengan metode ini koneksi internet terasa lebih cepat karena konsep load balancing adalah membagi rata beban koneksi ke beberapa jalur.

Kata Kunci : Load Balancer, Haproxy, Keepalived, web server, clustering.

I. PENDAHULUAN

Perubahan pola hidup manusia dewasa ini turut serta berpengaruh pada penerapan teknologi informasi. Saat ini manusia dalam hal ini adalah pengguna jaringan komputer membutuhkan sebuah kemudahan dalam mengakses suatu informasi pada jaringan internet. Tidak dapat dipungkiri lagi bahwa jaringan komputer berbasis internet mampu mempermudah dan membantu manusia mulai dari mencari informasi sampai penelitian yang kompleks.

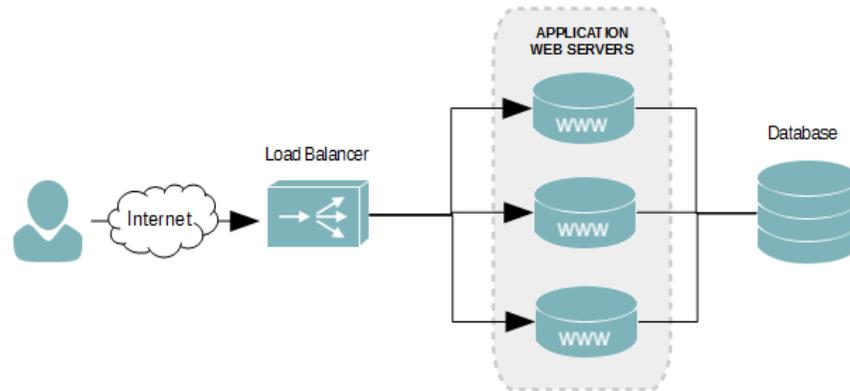
Pertumbuhan jaringan internet dewasa ini semakin pesat dan jumlah klien yang harus dilayani oleh server meningkat secara drastis. Beberapa situs bahkan sudah menerima ratusan ribu koneksi dari klien secara simultan.

Belakangan ini banyak perusahaan atau perorangan yang khawatir dengan sistem yang mereka miliki dari waktu ke waktu. Selain itu respon cepat dan ketersediaan 24x7 merupakan persyaratan utama, sebagai situs yang bersaing untuk menawarkan akses ke pengguna dengan layanan yang terbaik. Suatu situs biasanya hanya memiliki sebuah server tunggal yang biasanya tidak cukup untuk menangani beban permintaan yang meningkat secara drastis. Proses upgrade server sangat kompleks, dan single server adalah titik tunggal kegagalan (single point of failure). Semakin tinggi server di-upgrade maka biaya yang akan dikeluarkan semakin tinggi. sudah banyak perusahaan sekarang menggunakan konsep load balancer untuk membagi beban secara merata terhadap server yang dimiliki, namun masih ada masalah yaitu jika load balancer-nya mati maka semua node server di backend akan tidak bisa diakses.

Pada kenyataannya banyak perusahaan yang menggunakan load balancer untuk mengatasi masalah down pada single server. Maka dari itu penelitian ini akan mempelajari desain sistem server load balancing aktif standby yang memiliki beberapa keuntungan termasuk fleksibilitas, skalabilitas, high availability dan keamanan.

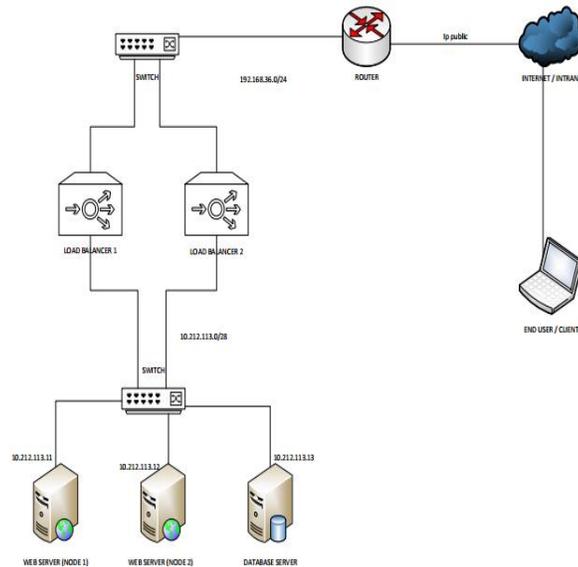
Load balancing adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan throughput, memperkecil waktu tanggap dan menghindari overload pada salah satu jalur koneksi. Load balancing digunakan pada saat sebuah server telah memiliki jumlah user yang telah melebihi

maksimal kapasitasnya. Load balancing juga mendistribusikan beban kerja secara merata di dua atau lebih komputer, link jaringan, CPU, hard drive, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal.



Gambar 1 . *Load Balancer*

Untuk mempermudah ketika mengimplementasi jaringan yang diperlukan dalam perancangan load balancer yang tepat dan efisien, supaya semua bagian dapat bekerja secara maksimal, perancangan jaringan ini nantinya akan menjadi landasan dalam implementasi load balancer aktif standby menggunakan protokol VRRP pada web server kluster.



Gambar 2. Topologi Jaringan Load Balancer aktif standby

Dalam perancangan load balancer aktif standby membutuhkan beberapa tahapan dasar yang harus dilakukan diantaranya konfigurasi ip address, konfigurasi Haproxy dan keepalived, konfigurasi web server , database server dan konfigurasi router kerio control.

II. METODE PENELITIAN

Metode yang digunakan dalam mencapai tujuan penelitian adalah sebagai berikut : Melakukan konfigurasi baik itu konfigurasi terhadap load balancer, server web , sistem operasi maupun perangkat jaringan lainnya. Untuk mempermudah ketika mengimplementasi jaringan yang diperlukan dalam perancangan load balancer yang tepat dan efisien, supaya semua bagian dapat bekerja secara maksimal, perancangan jaringan ini nantinya akan menjadi landasan dalam implementasi load balancer aktif standby menggunakan protokol VRRP pada web server kluster.

III. HASIL DAN PEMBAHASAN

Setelah pembuatan sistem load balancer aktif standby selesai, maka harus dilakukan suatu percobaan terlebih dahulu, untuk melihat apakah suatu konfigurasi yang sudah dilakukan dapat berjalan dengan baik atau tidak.

A. Implementasi load balancer pada server web

Implementasi ini bertujuan untuk menguji seberapa cepat load balancer dalam membagikan beban yang masuk terhadap jumlah server backend yang dimiliki. Untuk implementasi ini terdapat 2 server web yaitu server web 1 (node 1) dan server web 2(node 2) dan pada load balancer menunakan algoritma roundrobin dalam pembagian beban, karena sifat dari roundrobin yang sama rata dalam pembagian beban ke server.

1. Mengakses <https://datacenter.id/> pada saat pertama kali, pada load balancer akan membaca paket data, kemudian akan mengkalkulasi banyaknya permintaan yang masuk, kemudian akan di arahkan ke server yang pertama.

Routecloud Network Home Forum Info Server 2

Data Center Routecloud Network Indonesia

No	Kode Datacenter	Nama Datacenter	Jumlah Server	Status	Action
1	DPK01	Datacenter Gunadarma	80	aktif	Lihat Edit Hapus
2	BDG01	Institute Teknologi Bandung	70	aktif	Lihat Edit Hapus
3	DPK02	Universitas Indonesia	100	aktif	Lihat Edit Hapus
4	DIY01	Universitas Gadjah Mada	90	aktif	Lihat Edit Hapus
5	JKT01	Universitas Bina Nusantara	60	aktif	Lihat Edit Hapus

Tambah Data Center

Gambar 3. Server web 1

- Mengakses <https://datacenter.id/> pada saat kedua kalinya, pada load balancer akan membaca paket data, kemudian paket data tersebut akan di arahkan pada server yang kedua.

Routecloud Network Home Forum Info Server 1

Data Center Routecloud Network Indonesia

No	Kode Datacenter	Nama Datacenter	Jumlah Server	Status	Action
1	DPK01	Datacenter Gunadarma	80	aktif	Lihat Edit Hapus
2	BDG01	Institute Teknologi Bandung	70	aktif	Lihat Edit Hapus
3	DPK02	Universitas Indonesia	100	aktif	Lihat Edit Hapus
4	DIY01	Universitas Gadjah Mada	90	aktif	Lihat Edit Hapus
5	JKT01	Universitas Bina Nusantara	60	aktif	Lihat Edit Hapus

Tambah Data Center

Gambar 4. Server web 2

B. Implementasi load balancer aktif standby

Implementasi ini bertujuan untuk menguji seberapa cepat load balancer dalam melakukan *convergence* dari load balancer aktif ke load balancer standby.

1. Mengakses <http://datacenter.id:81/>, pada saat load balancer aktif masih berjalan maka, pada statistic akan muncul lb01 yang berarti load balancer tersebut masih bekerja di load balancer 1 (aktif).

Statistics Report for pid 1279 on lb01: Routecloud Load Balancer Status

> General process information

pid = 1279 (process #1, nbroc = 1)
 uptime = 0d 0h31m23s
 system limits: memmax = unlimited; ulim-n = 531
 maxsock = 531; maxconn = 256; maxpipes = 0
 current conns = 1; current pipes = 0/0; conn rate = 0/sec
 Running tasks: 1/14; idle = 100 %

Legend:
 active UP (green), backup UP (blue), active UP, going down (yellow), backup UP, going down (purple), active DOWN, going up (orange), backup DOWN, going up (pink), active or backup DOWN (red), not checked (grey), active or backup DOWN for maintenance (MAINT) (dark red), active or backup SOFT STOPPED for maintenance (dark blue), Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option: Scope: []
 External resources:
 - Primary site
 - Updates (v1.5)
 - Hide "DOWN" servers
 - Disable refresh
 - Refresh now
 - CSV export

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
Frontend	0 287 - 0	53 2 000	2 451	197 771	2 343 707	0 0	45 OPEN

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
Frontend	0 164 - 0	101 2 000	3 248	484 522	2 266 926	0 0	32 OPEN

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
Frontend	0 255 - 0	44 2 000	3 318	190 580	2 145 122	0 0	40 OPEN

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
Frontend	0 258 - 0	47 2 000	3 366	191 061	2 190 760	0 0	41 OPEN

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server	
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft	
web-node1	0 0 - 0	400 0 55 -	5 447 5 447	49 533 818	4 623 400	0 0	0 0 0 0 0 31m23s UP L4OK in 0ms 1 Y - 0 0 0s -	
web-node2	0 0 - 0	400 0 60 -	5 446 5 446	49 530 116	4 293 494	0 0	0 0 0 0 0 31m23s UP L4OK in 1ms 1 Y - 0 0 0s -	
Backend	0 0 - 0	800 0 101	800 10 941	10 893 49	1 063 934	8 922 878	0 0	0 0 0 0 0 31m23s UP 2 2 0 0 0 0s -

Gambar 5. Load balancer1

2. Mengakses <http://datacenter.id:81/>, pada saat load balancer aktif masih tidak berfungsi maka, pada statistic akan muncul lb02 yang berarti load balancer tersebut sudah bekerja pada load balancer 2 (aktif).

Statistics Report for pid 4214 on lb02: Routecloud Load Balancer Status

> General process information

pid = 4214 (process #1, nbroc = 1)
 uptime = 0d 0h01m05s
 system limits: memmax = unlimited; ulim-n = 531
 maxsock = 531; maxconn = 256; maxpipes = 0
 current conns = 2; current pipes = 0/0; conn rate = 0/sec
 Running tasks: 1/15; idle = 100 %

Legend:
 active UP (green), backup UP (blue), active UP, going down (yellow), backup UP, going down (purple), active DOWN, going up (orange), backup DOWN, going up (pink), active or backup DOWN (red), not checked (grey), active or backup DOWN for maintenance (MAINT) (dark red), active or backup SOFT STOPPED for maintenance (dark blue), Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option: Scope: []
 External resources:
 - Primary site
 - Updates (v1.5)
 - Hide "DOWN" servers
 - Disable refresh
 - Refresh now
 - CSV export

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
Frontend	0 0 - 0	0 2 000	0	0 0	0 0	0 0	0 0 0 0 0 0s -

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
Frontend	0 1 - 1	1 2 000	1	479 10 481	0 0	0 0	0 0 0 0 0 0s -

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
Frontend	0 0 - 0	0 2 000	0	0 0	0 0	0 0	0 0 0 0 0 0s -

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
Frontend	0 0 - 0	0 2 000	0	0 0	0 0	0 0	0 0 0 0 0 0s -

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Server
Cur Max Limit	Cur Max Limit	Cur Max Limit	Total LbTot Last	In Out	Req Resp Req Conn Resp	Retr Redis	Status LastChk Wght Act Bck Chk Dwn Downtime Thrft
web-node1	0 0 - 0	1 0 1 -	1 1 3s 479	10 481	0 0	0 0	0 0 0 0 0 1m5s UP L4OK in 0ms 1 Y - 0 0 0s -
web-node2	0 0 - 0	0 0 0 -	0 0 ?	0 0	0 0	0 0	0 0 0 0 0 1m5s UP L4OK in 0ms 1 Y - 0 0 0s -
Backend	0 0 - 0	2 1 1 800	2 1 3s 479	10 481	0 0	0 0	0 0 0 0 0 1m5s UP 2 2 0 0 0 0s -

Gambar 6. Load balancer2

- Kecepatan konvergance pada protokol vrrp secara default adalah 1 detik, ketika load balancer 1 mengalami kegagalan sistem, maka 1 detik kemudian virtual IP dari load balancer aktif akan di serahkan ke load balancer 2.

```

root@AlanPrastyo:/home/alanprastyo# ping datacenter.id
PING datacenter.id (192.168.43.100) 56(84) bytes of data.
64 bytes from datacenter.id (192.168.43.100): icmp_seq=1 ttl=64 time=0.382 ms
64 bytes from datacenter.id (192.168.43.100): icmp_seq=2 ttl=64 time=0.385 ms
64 bytes from datacenter.id (192.168.43.100): icmp_seq=3 ttl=64 time=0.351 ms
64 bytes from datacenter.id (192.168.43.100): icmp_seq=4 ttl=64 time=0.457 ms
64 bytes from datacenter.id (192.168.43.100): icmp_seq=5 ttl=64 time=0.394 ms
From datacenter.id (192.168.43.100): icmp_seq=6 Redirect Host(New nexthop: datacenter.id (192.168.43.100))
From datacenter.id (192.168.43.100): icmp_seq=7 Redirect Host(New nexthop: datacenter.id (192.168.43.100))
64 bytes from datacenter.id (192.168.43.100): icmp_seq=6 ttl=64 time=1362 ms
64 bytes from datacenter.id (192.168.43.100): icmp_seq=7 ttl=64 time=362 ms
64 bytes from datacenter.id (192.168.43.100): icmp_seq=8 ttl=64 time=0.281 ms
64 bytes from datacenter.id (192.168.43.100): icmp_seq=9 ttl=64 time=0.370 ms

```

Gambar 7. proses ping pada virtual IP

Dari hasil gambar tersebut, ketika proses pemberian virtual dari load balancer 1(aktif) ke load balancer 2 (standby) tidak mengalami request time out atau terjadi putus koneksi, namun hanya terjadi delay yang cukup besar sesaat virtual ip tersebut di berikan ke load balancer 2.

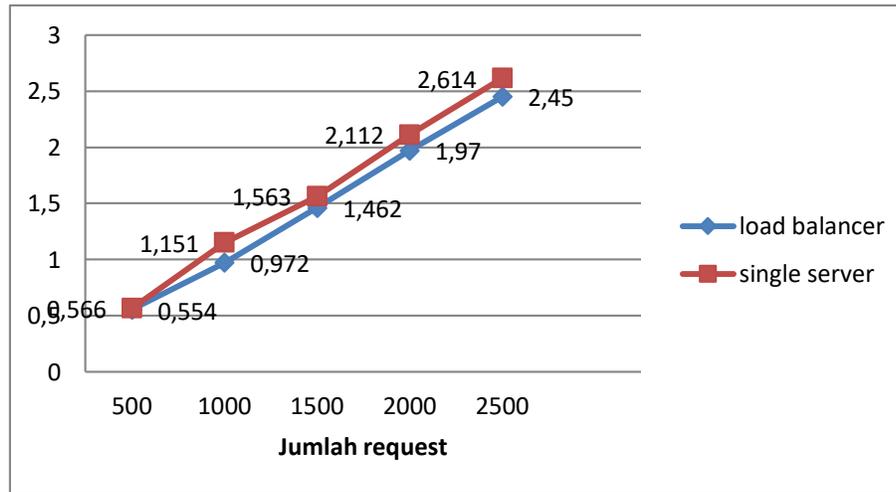
C. Hasil kecepatan stress testing pada load balancer

Pada hasil uji coba ini dilakukan stress testing pada load balancer server dimulai 500 request sampe 2500 request dengan setiap kali pengulangan mengalami kenaikan 500 request. Pengujian ini membandingkan antara server load balancer dengan server tunggal dengan hasil seperti tabel IV.1

balancer dengan Tabel 1. Perbandingan waktu load balancer dengan server tunggal

No	Jumlah <i>request</i>	Load Balancer (detik)	Server Tunggal (detik)
1	500	0.554	0.566
2	1000	0.972	1.151
3	1500	1.462	1.563
4	2000	1.970	2.112

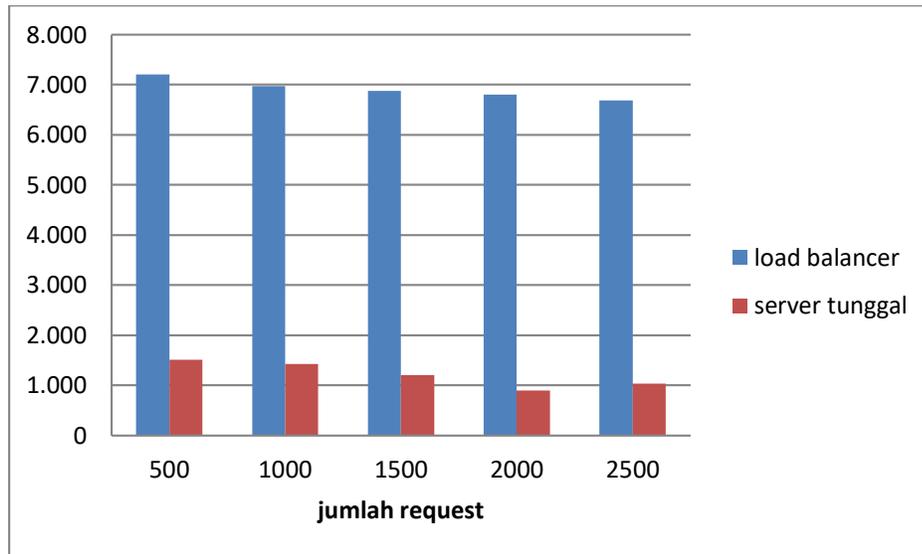
5	2500	2.450	2.614
---	------	-------	-------



Gambar 8. Grafik Perbandingan waktu load server tunggal

Tabel 2. Perbandingan troughput load balancer dengan server tunggal

No	Jumlah request	Load Balancer (KBps)	Server Tunggal (KBps)
1	500	7,206	1,507
2	1000	6,978	1,425
3	1500	6,873	1,204
4	2000	6,799	892
5	2500	6,688	1,030



Gambar 9. Grafik Troughput Load Balancer Dengan Server Tunggal

IV. Kesimpulan

Dari hasil uji coba implementasi load balancer aktif standby pada web server kluster yang di lakukan maka diperoleh kesimpulan sebagai berikut:

1. Secara fungsional, sistem load balancing server telah berhasil diimplementasikan, sistem dapat menyeimbangkan beban dan memiliki tingkat ketersediaan yang tinggi, serta dibangun dari komputer-komputer yang memiliki spesifikasi biasa yang menggunakan sistem operasi Linux serta perangkat lunak yang bersifat opensource.
2. Sistem kerja load balancer aktif standby sudah sesuai dengan tujuan implementasi yaitu jika load balancer master mengalami kegagalan dalam sistem maka akan dialihkan pada load balancer backup, dengan waktu perpindahan dari load balancer master ke backup membutuhkan 1 detik, sehingga proses permintaan dari klien ke server tidak mengalami gangguan seperti terputus koneksi, hanya saja mengalami

delay yang cukup besar pada awal perpindahan koneksi dari load balancer master ke backup.

3. Sistem load balancing server yang dirancang memiliki kinerja yang lebih baik dibandingkan dengan penggunaan server tunggal, yaitu:
 - Kemampuan sistem load balancing server untuk melayani permintaan dari klien jauh lebih besar, sehingga throughput pada sistem kluster lebih bagus dibandingkan server tunggal.
 - Waktu respon sistem load balancing server secara keseluruhan lebih baik, dimana waktu respon sistem load balancing lebih singkat dibandingkan waktu respon server tunggal.

Saran

1. Untuk mengetahui karakteristik kinerja sistem load balancing, diperlukan adanya penelitian lebih lanjut mengenai sistem load balancing dengan penggunaan jumlah server yang lebih banyak.
2. Penelitian lebih lanjut pada sistem load balancing server dengan kedua load balancer dalam keadaan aktif sehingga bersifat load sharing.
3. Perlu adanya penelitian lebih lanjut mengenai algoritma penjadwalan yang digunakan dalam load balancing untuk mengoptimalkan kinerja sistem.

Daftar Pustaka

Bourke, Tony., *Server Load Balancing*, O'Reilly & Associates, Inc., 2001.

Mallett, Andrew., *CentOS System Administration Essential*, Packt Publishing, Ltd., 2014.

Nedelcu, Clement., *Nginx HTTP Server Second Edition*, Packt Publishing, Ltd., 2013.

Anonim. "Centos". <http://gudanglinux.com/glossary/centos/>. Tanggal Akses : 29 Maret 2016, 12:45 WIB.

Anonim. "Mengenal Teknologi Load Balancing". <http://fxekobudi.net/networking/mengenal-teknologi-load-balancing/> Tanggal Akses : 27 Maret 2016, 10:00 WIB.

Anonim. "Pengertian Web Server dan Fungsinya". <https://idcloudhost.com/pengertian-web-server-dan-fungsinya/> Tanggal Akses : 29 Maret 2016, 11:48 WIB.

Arisyi. "Install NGINX 1.10, PHP-FPM 7.0, dan MariaDB 10.1 di CentOS 7".
<https://www.routecloud.net/blog/install-nginx-1-10-php-fpm-7-0-dan-mariadb-10-1/>
Tanggal Akses : 1 Mei 2016, 07:30 WIB

Seltzer, Brian. "Redundant Load Balancers HAProxy and Keepalived".
<http://behindtheracks.com/2014/04/redundant-load-balancers-haproxy-and-keepalived/>
Tanggal Akses : 1 Mei 2016, 06:00 WIB

Tarreau, Willy. "Haproxy Configuration Manual".
<http://www.haproxy.org/download/1.5/doc/configuration.txt> Tanggal Akses : 15 April
2016, 19:09 WIB