

HIGH AVAILABILITY WEB SERVER MENGGUNAKAN PROTOKOL SECURE SHELL

I Gede Putu Krisna Juliharta¹⁾, I Putu Oka Wisnawa²⁾

Program Studi Sistem Informasi^{1), 2)}

Sekolah Tinggi Manajemen dan Ilmu Komputer (STMIK) STIKOM Bali^{1), 2)}

Jl. Raya Puputan Renon No. 86, Telp (0361) 244445 Denpasar Bali^{1), 2)}

krisna@stikom-bali.ac.id¹⁾, okawisnawa@stikom-bali.ac.id²⁾

ABSTRAK

Untuk mengurangi resiko terhadap lumpuhnya sistem dibutuhkan sebuah sistem yang menjaga availability dari sebuah sistem. Availability merupakan sebuah kemampuan dari sebuah server untuk dapat tetap memberikan layanan sepanjang waktu. High availability dengan secure shell merupakan teknologi yang dapat membuat server untuk selalu bertahan dan menghindari resiko pemadaman. Dan dengan secure shell dan teknik load balancer maka proses replikasi server dapat dilakukan dengan baik.

Kata Kunci : *server, load balancer, high availability*

1. Pendahuluan

Penerapan dari teknologi jaringan yang paling populer didunia adalah Internet. Penggunaannya sudah Keruntuhan infrastruktur jaringan pada sistem jaringan merupakan bencana, menyebabkan terhentinya kegiatan sehari-hari suatu entitas karena kehilangan sumberdaya dan alat komunikasi. Masa fasilitas kemudahan berkomunikasi lenyap, dan peta jalan masa depan tiba tiba buram. Pertanggungjawaban menjadi tidak mungkin diberi atau diminta. bencana dapat menyebabkan infrastruktur sistem yang telah dibangun hilang dalam waktu seketika.

Penerapan dari teknologi jaringan yang paling populer didunia adalah Internet. Penggunaannya sudah merambah semua kalangan, tua, muda, pribadi, perusahaan ataupun organisasi semuanya menggunakan internet. Teknologi yang awalnya digunakan untuk kebutuhan perang telah merambah dan menjadi tulang punggung kegiatan manusia dalam bekerja maupun gaya hidup. Ketergantungan terhadap teknologi informasi ini bukanlah tanpa resiko, banyak hal yang menyebabkan sistem bisa menjadi lumpuh. Lumpuhnya sistem tentunya mengakibatkan layanan terganggu.

Untuk mengurangi resiko terhadap lumpuhnya sistem dibutuhkan sebuah sistem yang menjaga availability dari sebuah sistem. High availability dengan secure shell merupakan teknologi yang diharapkan mengurangi resiko tersebut. Dengan secure shell proses replicasi data dapat dilakukan dengan konsep enkripsi. Karena data tersebut berupa salinan sehingga data tersebut sama dengan data yang ada di server utama, sehingga availability dari sistem dapat terjaga.

Berdasarkan pemaparan diatas maka peneliti akan membuat sebuah perancangan high availability web server dengan menggunakan protocol secure shell.

2. Metode Penelitian

2.1 Sistematika Penelitian

Dalam melakukan penelitian ini, penulis melakukan langkah-langkah penelitian sebagai berikut :

- a. Studi awal
Dalam melakukan studi awal, penulis melakukan : pencarian materi yang berkaitan dengan high availability server dan teknologi jaringan komputer.
- b. Pengumpulan data lapangan
Pada tahapan ini, penulis melakukan pengumpulan data yang diperoleh dengan cara melakukan observasi terhadap prosedur penerapan jaringan komputer dan melakukan wawancara mendalam untuk mengetahui tata kelola dari jaringan yang ada.
- c. Penyelidikan Sejarah
Pada tahap ini, penulis juga masih menggunakan teknik wawancara dan pengumpulan data mengenai sejarah insiden terhadap jaringan yang selama ini pernah dialami.
- d. Perancangan dan Implementasi High Availability

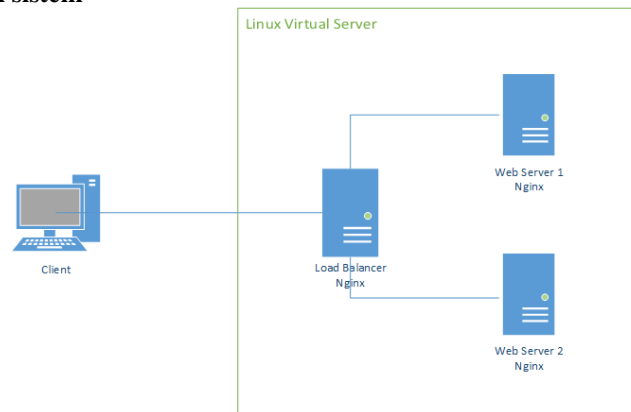
Merancang dan membangun sistem availability server berdasarkan hasil observasi

2.2 Instrumen Penelitian

Alat penelitian yang akan digunakan dalam penelitian dibagi dalam dua bagian utama yaitu:

- a. Perangkat Lunak
Perangkat lunak yang digunakan dalam perancangan sistem terdiri dari dua bagian yaitu :
 1. Sistem Operasi Windows Seven
 2. Aplikasi Microsoft Visio
- b. Perangkat Keras
Perangkat keras yang digunakan dalam penelitian adalah:
 1. Komputer dengan spesifikasi
 - Processor Intel Centrino Duo 1,8GHz
 - 2 GB RAM
 - Hardisk 160 GB
 - VGA 256 MB
 2. HDD External 500 GB

2.3 Perancangan sistem



Gambar 1. Perancangan sistem

Gambar 1. Menunjukkan proses high availability. Terdapat load balancer yang bertugas mengatur antara server utama dan server backup. Sehingga jika server utama mati dapat dibackup menggunakan server kedua. Server proses sinkronisasi menggunakan protocol secure shell, memodifikasi dari penggunaan secure shell (ssh)

3 Hasil dan Pembahasan

Proses awal adalah membangun web server dengan memiliki tiga file yaitu file1.mkv, file2.mp4 dan file3.mkv seperti terlihat pada gambar 2.

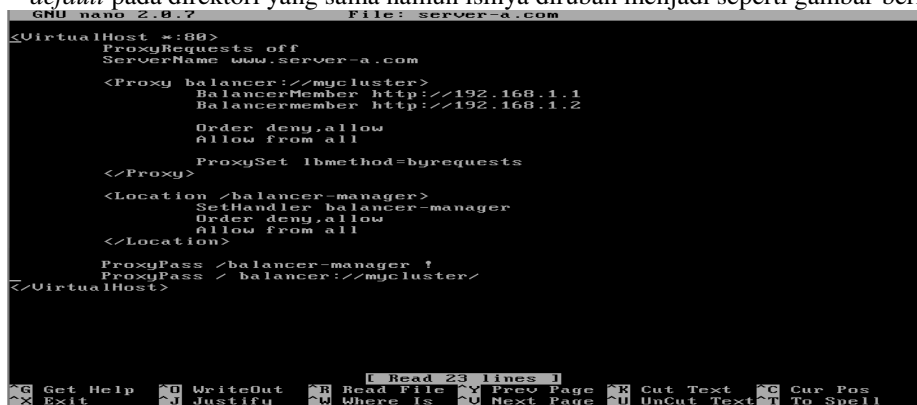


Gambar 2. Web page Apache web server 1

Proses selanjutnya adalah proses konfigurasi *load balancer Apache*, diperlukan instalasi, pengaktifan beberapa modul untuk *load balancing* dan konfigurasi *file load balancing*. Langkah – langkahnya sebagai berikut;

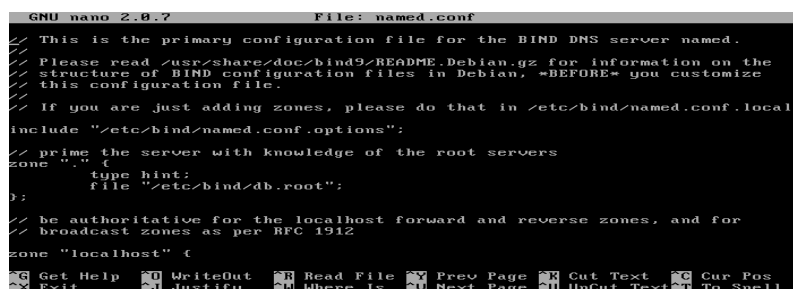
1. Instalasi paket *Apache2* dengan perintah `apt-get install Apache2`
2. Masuk ke direktori `/etc/apache2` dengan perintah `cd /etc/apache2/`
3. Aktifkan module yang diperlukan untuk mengkonfigurasi *load balancer* menggunakan perintah berikut;

```
a2enmod proxy           // mengaktifkan module proxy
a2enmod proxy_http      // mengaktifkan module proxy_http
a2enmod proxy_ajp        // mengaktifkan module proxy_ajp
a2enmod rewrite          // mengaktifkan module rewrite
a2enmod deflate          // mengaktifkan module deflate
a2enmod headers          // mengaktifkan module headers
a2enmod proxy_balancer   // mengaktifkan module proxy_balancer
a2enmod proxy_connect    // mengaktifkan module proxy_connect
a2enmod proxy_html       // mengaktifkan module proxy_html
```
4. Selanjutnya konfigurasi *load balancing* ditempatkan pada file konfigurasi *virtual host* yang terdapat di direktori `/etc/apache2/sites-available/`. File tersebut merupakan salinan dari file *default* pada direktori yang sama namun isinya dirubah menjadi seperti gambar berikut;



Gambar 3. konfigurasi load balancer

Langkah terakhir adalah memberikan atau mengkonfigurasi *domain name* untuk *load balancer Nginx*. Paket yang digunakan adalah BIND9. Proses pertama adalah mengkonfigurasi file `named.conf` di `bind9` sehingga terbentuk sebuah file `db.root` yang meakomodasi dari konfigurasi domain name server gambar 4.



Gambar 4. Konfigurasi default named.conf

Gambar 4. merupakan isi *default* dari file konfigurasi `named.conf`. Tambahkan zone untuk melakukan forward dan reverse *domain name* seperti pada gambar berikut;

```

GNU nano 2.8.7 File: named.conf

zone "www.server-a.com"{
    type master;
    file "/etc/bind/db.server-a";
};

zone "1.168.192.in-addr.arpa"{
    type master;
    file "/etc/bind/db.server-a.rev";
};

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^X Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell

```

Gambar 4. Konfigurasi named.conf untuk load balancer apache

Pada gambar 4, terdapat 2 buah zone yang didaftarkan, dimana zone forward merujuk ke *file* konfigurasi db.server-a dan zone reverse merujuk ke *file* konfigurasi db.server-a.rev. Simpan perubahan untuk *file* konfigurasi named.conf. Selanjutnya *copy file* db.local dengan nama db.server-a untuk menterjemahkan *IP address* menjadi *domain name*. Dan *copy file* db.127 dengan nama db.server-a.rev untuk mengembalikan *domain name* menjadi *IP address*.

```

GNU nano 2.8.7 File: db.server-a

; BIND data file for local loopback interface
$TTL      604800      SOA      www.server-a.com. root.server-a.com. (
;
; IN      IN      NS      www.server-a.com.
; IN      IN      A       192.168.1.5
;
; Read 13 lines
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^X Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell

```

Gambar 5. Konfigurasi db.server-a

Gambar 5 adalah tampilan isi dari *file* konfigurasi db.server-a yang merupakan *file* untuk menterjemahkan *Ip address* load balancer apache yaitu 192.168.1.5 menjadi *domain name* www.server-a.com.

```

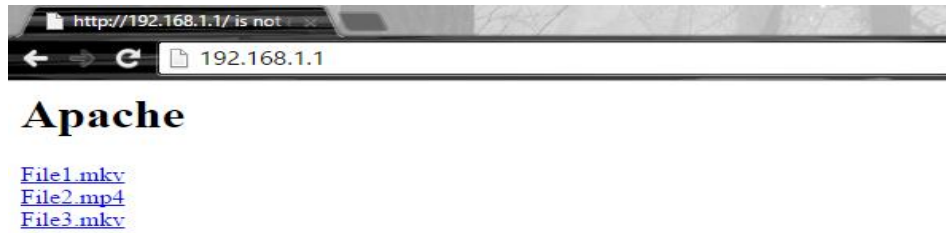
GNU nano 2.8.7 File: db.server-a.rev

; BIND reverse data file for local loopback interface
$TTL      604800      SOA      www.server-a.com. root.server-a.com. (
;
; IN      IN      NS      www.server-a.com.
; IN      IN      PTR     www.server-a.com.
;
; Read 13 lines
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^X Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell

```

Gambar 6. Konfigurasi db.server-a.rev

Gambar 6. merupakan konfigurasi db.server-a.rev yang merupakan *file* untuk menterjemahkan *domain name* www.server-a.com kembali menjadi *Ip address* yaitu 192.168.1.5. untuk proses pengujian *load balancer* dari client menggunakan *web browser* dengan mengakses *domain name* www.server-a.com yang diinputkan pada address bar *web browser*



Gambar 7. Hasil pengujian domain name www.server-a.com

Pada Gambar 7. terlihat tampilan *web* dari *load balancer apache* yang menampilkan *web page* dari *web server apache* 1. *Load balancer apache* sukses dikonfigurasi.

4 Simpulan

1. Implementasi High Availability Web Server telah berjalan dengan baik
2. Webserver yang dibangun menggunakan apache dan perangkat lunak yang digunakan adalah SSH dengan metode load balancer

Daftar Pustaka

- [1] Jefry Alvonsius Rabu, Joko Purwadi, Willy S. Raharjo. Implementasi *Load balancing Web Server* Menggunakan Metode LVS-NAT. *Informatika*. 2012; Vol. (8): 169 – 180.
- [2] Rahmad Saleh Lubis, Maksum Pinem. Analisis Quality Of Service (QoS) Jaringan Internet di SMK Telkom Medan. Singuda Ensikom. 2014: Vol. (7) No. 3: 131-136
- [3] Sirajuddin, Achmad Affandi, Eko Setijadi. Rancang Bangun *Server Learning Management System* Menggunakan *Load balancer* dan Reverse Proxy. *Jurnal Teknik ITS*. 2012; Vol. (1): A-50 – A-52.
- [4] Wahana Komputer. Konsep dan Implementasi Jaringan dengan Linux Ubuntu. Yogyakarta: C. V Andi Offset. 2014
- [5] Makinuddin, Sasongko Tri Hadiyanto. Analisis Sosial: Beraksi Dalam Advokasi Irigasi. Bandung: Yayasan Akatiga. 2006
- [6] Rijayani Iwan. Teknologi *Load balancing* Untuk Mengatasi Beban *Server*. Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNAT 2005). 2005; ISBN: 979-756-061-6
- [7] Cartealy Imam. *Linux Networking* Ubuntu, Kubuntu, Debian, dll. Jakarta: Jasakom. 2013