

**IMPLEMENTASI CLUSTER PADA WEB SERVER
BERBASIS CLOUD COMPUTING**

RESUME SKRIPSI

untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1

Program Studi Teknik Informatika



disusun oleh:

SETIYA BUDI

07650018

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2013

KATA PENGANTAR

Puji syukur bagi Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini. Shalawat serta salam semoga tetap tercurah ke haribaan Rasulullah SAW yang telah menunjukkan kebenaran hakiki dan kebahagiaan sejati bagi umat manusia, ahli bait-nya yang suci, dan sahabat-sahabatnya yang mulia.

Penulisan skripsi ini tidak lepas dari bantuan dan dukungan serta bimbingan berbagai pihak. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan yang sebesar – besarnya kepada seluruh pihak yang telah memberikan bantuan dan dukungan serta bimbingan.

Penulis menyadari masih banyak kekurangan dalam penyusunan laporan skripsi ini. Semoga penyusunan skripsi ini dapat menjadi pengalaman yang berharga bagi penulis.

Yogyakarta, 9 Juli 2013

Penulis

IMPLEMENTASI CLUSTER PADA WEB SERVER BERBASIS CLOUD COMPUTING

Setiya Budi
07650018

INTISARI

Salah satu teknologi dibalik *Cloud Computing* merupakan *Platform* virtualisasi dan *Cluster Server* yang menjamin ketersediaan layanan yang tinggi dikarenakan semua data ditampung pada *High Available Server*. *Cloud Computing* di buat dengan tujuan agar semua layanan bisa di akses secara *realtime*, kapanpun dan dimanapun. Pada dasarnya skema pelayanan *Cloud Computing* berjalan berbasis web, sehingga hal ini merupakan solusi komputasi *multi platform*, mulai dari komputer *desktop* hingga perangkat mikro seperti *smartphone* bisa mengakses layanan ini.

Penelitian penulis lebih menekankan implementasi Mesin *virtual Web Server* dengan teknik *High-Availibility* atau *Failover Cluster* yang berjalan pada *Hypervisor Proxmox Virtual Environment (Proxmox VE)*. *Platform* virtualisasi menggunakan *Kernel-Based Virtual Machine (KVM)*. Pada penelitian ini, penulis menggunakan metode pengumpulan data berupa studi pustaka dan wawancara langsung kepada narasumber. Sedangkan tahapan penelitian terbagi dalam beberapa langkah yakni Perancangan sistem dan jaringan, Implementasi rancangan sistem dan pengujian sistem.

Hasil akhir dari penelitian ini diukur sebagai nilai pertimbangan terhadap sistem *High-Availibility Cluster* yang mampu menangani isu ketersediaan layanan serta menimalisir kemungkinan *downtime* sehingga meningkatkan *uptime Server* yang cukup signifikan dibandingkan menggunakan *Server* konvensional.

Kata Kunci : *Web Server Cluster, Cloud Computing, High-Availibility, Virtualisasi, Proxmox VE*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Meningkatnya kebutuhan belanja Teknologi Informasi (TI) seiring dengan perkembangan infrastruktur TI secara global menuntut perusahaan melakukan inovasi terkait penghematan anggaran. Tidak heran bila perusahaan melakukan pemangkasan pada beberapa unit investasi, bahkan perusahaan juga menekan pengeluaran serta memaksimalkan investasi Teknologi & Informasi (TI).

Wacana komputasi awan (yang selanjutnya lebih dikenal dengan istilah *Cloud Computing*) semakin gencar di gembor-gemborkan sebagai solusi memaksimalkan *resource* dalam dunia TI, terutama dalam isu penghematan anggaran perusahaan. Salah satu teknologi dibalik *Cloud Computing* adalah *Platform* virtualisasi dan *Cluster Server* yang menjamin ketersediaan layanan yang tinggi karena semua data ditampung kedalam *Server* data yang *High Available*.

CV Berkah Abadi sebagai salah satu perusahaan yang bergerak dibidang TI telah menerapkan sebuah webserver untuk keperluan pelayanan aplikasi berbasis web. Penggunaan webserver tunggal tentunya beresiko dengan tidak adanya mekanisme backup serta rawan *downtime*. Skenario tersebut merupakan salah satu objek yang bisa diselesaikan dengan implementasi konsep *Cloud Computing* yang mudah dan murah dengan model penyebaran *Private Cloud* dengan tidak mengurangi performa pelayanan sistem.

1.2 Rumusan Masalah

Berdasar latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang suatu sistem *Cluster* pada *Web Server* dengan skema *Cloud Computing* yang mampu meminimalisir *downtime*.
2. Bagaimana mengimplementasikan sistem *Cluster* pada *Web Server* tersebut menggunakan Platform Virtualisasi pada *Proxmox VE*

1.3 Batasan Masalah

1. Sistem *Cluster Web Server* ini berada pada jaringan lokal (LAN) di CV.Berkah Abadi dan *client* sebatas mengakses data pada *Web Server*.
2. Penelitian ini menggunakan dua unit server dengan sistem operasi *Proxmox VE 3.0* dan masing-masing terpasang mesin *Web Server* virtual berbasis sistem operasi *Debian Squeeze*.
3. Mesin *Server* virtual dibangun diatas *Platform KVM (Kernel-Based Virtual Machine)*.
4. Implementasi *Cluster* pada *Web Server* ini menggunakan teknik *High-availability/Failover Cluster* dengan skema *Active-Passive* dan belum mendukung *Load-balancing*. Sehingga layanan *Fallover* masih terbatas pada Alamat IP, belum sampai service aplikasi *web server*.
5. Redundansi data pada *database mysql* digunakan teknik *mysql-replication* dengan konfigurasi *Master-Master*.
6. Pengujian dalam penelitian ini menggunakan data *dummy*.
7. Aspek keamanan sistem tidak dibahas dalam penelitian ini.

1.4 Tujuan Penelitian

1. Merancang suatu sistem *Cluster* pada *Web Server* dengan skema *Cloud Computing* yang mampu meminimalisir *downtime* guna menjaga tingginya ketersediaan layanan.
2. Mengimplementasikan sistem *Cluster* pada *Web Server* tersebut menggunakan sistem operasi *Proxmox VE* sebagai solusi virtualisasi yang handal dan murah di CV Berkah Abadi.

1.5 Manfaat Penelitian

1. Memberi manfaat bagi instansi terkait mengenai mudah dan murah nya implementasi *Cloud Computing* sebagai solusi atas resiko umum penggunaan *Web Server* konvensional.
2. Memberikan solusi terhadap tuntutan ketersediaan layanan yang tinggi (*High-Avalibility*) di instansi terkait yaitu di CV Berkah Abadi.
3. Memberikan sumbangsih bagi ilmu pengetahuan terutama dalam rumpun keilmuan TI dan dunia *Cloud Computing*.

1.6 Keaslian Penelitian

Penelitian umum terkait *Cluster* pada *Web Server* sudah pernah dilakukan sebelumnya. Akan tetapi penelitian penulis lebih menekankan tentang bagaimana mengimplementasikan mekanisme *High-Avalibility Cluster* pada *Web Server* dengan memanfaatkan skema *Cloud Computing*, yang mana sejauh belum pernah dilakukan sebelumnya, khususnya di UIN Sunan Kalijaga Yogyakarta.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian terdahulu yang menjadi pertimbangan dari penelitian ini disajikan sebagai perbandingan seperti tersaji pada Tabel 2.1

Tabel 2.1 Tabel daftar penelitian

No	Nama	Tahun	Metode	Hasil
1	Maitimu	2008	<i>Web Server Cluster</i> di implementasikan menggunakan <i>Load Balancing</i> pada <i>Linux Virtual Server (LVS)</i> memanfaatkan NAT	Sistem <i>Cluster</i> mampu memberikan performa merata serta jaminan <i>high-available</i> .
2	Afif	2012	<i>Disaster Recovery Plan</i> di implementasikan menggunakan <i>Heartbeat</i> sebagai detektor <i>failover</i> , serta aplikasi <i>DRBD</i> untuk replikasi data,	Sistem <i>Failover cluster</i> mampu menjaga ketersediaan layanan serta keidentikan data dengan nilai <i>downtime server</i> yang minimum
3	Muchtar	2012	Perbandingan <i>Failover Clustering</i> yang diterapkan pada <i>Platform Windows Server</i> dan <i>Ubuntu server</i> serta Virtualisasi dengan <i>Proxmox VE</i> . Pengujian masing-masing <i>platform</i> menggunakan aplikasi <i>httperf</i> .	Sistem <i>Failover Cluster</i> mampu menangani isu kegagalan <i>server</i> serta menjamin ketersediaan layanan, serta meningkatkan nilai <i>uptime server</i> yang cukup signifikan dibandingkan <i>server konvensional</i> .
4	Purnomo	2012	Sistem <i>Cluster</i> dibangun dengan koneksi <i>VPN</i> melalui fitur <i>EoIP</i> pada <i>Mikrotik</i> . Mekanisme <i>Failover</i> ditangani oleh <i>heartbeat</i> sedangkan Sinkronisasi <i>database</i> menggunakan metode <i>master-master</i>	Sistem <i>Failover Cluster</i> mampu menjaga ketersediaan layanan yang tinggi serta terjaganya redundansi <i>database</i> pada masing-masing <i>server billing</i> .
5	Budi	2013	<i>Cluster</i> pada <i>web server</i> dibangun diatas mesin <i>Virtual</i> pada <i>Proxmox VE</i> , Mekanisme <i>failover</i> menggunakan <i>ucarp</i> sedangkan replikasi <i>database</i> menggunakan <i>mysql-replication</i>	Sistem yang dibangun berupa kombinasi <i>Cloud Computing</i> serta <i>Cluster</i> yang <i>High-Available</i> . Sistem mampu menekan <i>downtime</i> serta menjaga keidentikan data pada kedua <i>server cluster</i> .

2.2. Landasan Teori

2.2.1. Cluster

Konsep *Cluster* dilahirkan saat orang mencoba menyebarkan job berbeda ke sejumlah komputer dan memperoleh informasi yang dihasilkan dari permosesan data. Dengan makin murah dan tersedianya Hardware komputer secara luas, maka kesenjangan kinerja antara superkomputer dan Cluster sejumlah komputer makin tipis. Secara mendasar terdapat tiga macam *Cluster* yaitu *High-availability Cluster*, *Load-balancing* dan *High Performance Computing Cluster* (HPCC) (Utdiraratmo, 2004).

2.2.2. Web Server

Web Server dapat bermakna dua hal, sebuah komputer di mana situs Website di pasang atau suatu program yang berjalan pada suatu komputer. Jadi Web Server merujuk pada Hardware dan Software (http://www.webdevelopersnotes.com/basics/what_is_web_server.php).

2.2.4. Cloud Computing

Cloud Computing adalah sebuah model komputasi / *Computing*, dimana sumber daya seperti *Processor / Computing power, storage, network*, dan *Software* menjadi abstrak dan diberikan sebagai layanan di jaringan / *internet* menggunakan pola akses *remote*. (Purbo , 2011).

Cloud Computing menyediakan tiga model layanan, yaitu *Software as a Service* (SaaS), *Platform as a Service* (PaaS) dan *Infrastructure as a Service* (IaaS) (Mell, 2009).

BAB III

METODE PENELITIAN

3.1. Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan penulis adalah sebagai berikut:

3.1.1. Studi Pustaka

Penulis melakukan studi pustaka dengan mencari, membaca dan mengumpulkan dokumen-dokumen sebagai referensi tentang *Cluster* dan *Cloud Computing* melalui buku referensi, artikel, buku elektronik (*e-book*), *Web-site*, jurnal publikasi dan literatur tugas akhir yang berhubungan dengan topik yang dipilih tentang pembuatan sistem *Cluster* dan *Cloud Computing*.

3.1.2. Wawancara

Merupakan metode yang penulis lakukan dengan mengajukan pertanyaan langsung kepada narasumber yang berkompeten dibidangnya untuk mendapatkan informasi yang dibutuhkan terkait dengan penelitian penulis.

3.2. Langkah Kerja Penelitian

Langkah kerja rentetan langkah taktis setelah data yang diperlukan sudah terkumpul. Uraian langkah kerja yang dilakukan terbagi dalam perancangan sistem dan jaringan, Implementasi rancangan sistem dan pengujian sistem.

3.2.1 Perancangan jaringan dan sistem

Tahap perancangan sistem dan jaringan merupakan tindak lanjut terhadap data yang didapat sebelumnya. Perancangan jaringan ini menjadi acuan desain sistem dan jaringan yang akan diterapkan dalam penelitian agar tidak melenceng dari tujuan awal sistem dibangun.

3.2.2 Implementasi Rancangan sistem

Tahap Implementasi merupakan tahap mempersiapkan sistem *Cluster* pada *Web Server* untuk selanjutnya diuji coba. Dalam penelitian ini *Proxmox VE* bertindak sebagai *Hypervisor* untuk mesin *Web Server* Virtual diatasnya. Dengan mekanisme *High-Availibility Cluster* yang diterapkan, akan mengurangi *downtime* seminimal mungkin ketika skenario sistem sudah di implementasikan.

3.2.3 Pengujian sistem

Pengujian merupakan langkah penting dalam penelitian berupa pembuktian terhadap relevansi fungsi dari sistem. Pengujian fungsional sistem *High-Availibility Cluster* ini menggunakan lima metode:

1. Pengujian Failover
2. Pengujian Redundansi data
3. Pengujian HTTP *Service*
4. Pengujian MySQL-Replication
5. Pengujian Stress tes

BAB IV

HASIL DAN PEMBAHASAN

4.1. Analisa Kebutuhan Sistem

Pada penelitian ini digunakan beberapa kebutuhan yang penulis sajikan dalam dua rincian, yaitu kebutuhan hardware dan kebutuhan software.

4.1.1. Perangkat keras (*Hardware*)

1. Komputer Server Master dan Backup

Komputer server master dan Backup dengan spesifikasi Processor setara Intel Dual Core dengan RAM 2 GB, Hardisk dengan kapasitas 160GB serta kartu jaringan LAN Card terintegrasi.

2. Perangkat kebutuhan jaringan

Kebutuhan perlengkapan jaringan menggunakan sebuah Wireless Router TL-WR340G, 2 buah kabel UTP-Cat5 lengkap dengan Konektor RJ-45 untuk dihubungkan dari komputer server master ke Router.

3. Komputer Klien

Komputer klien yang digunakan dengan spesifikasi processor setara dengan Intel Dual Core dengan RAM 1 GB, Hardisk dengan kapasitas 160GB serta kartu jaringan LAN Card terintegrasi.

4.1.2 Perangkat Lunak (*Software*)

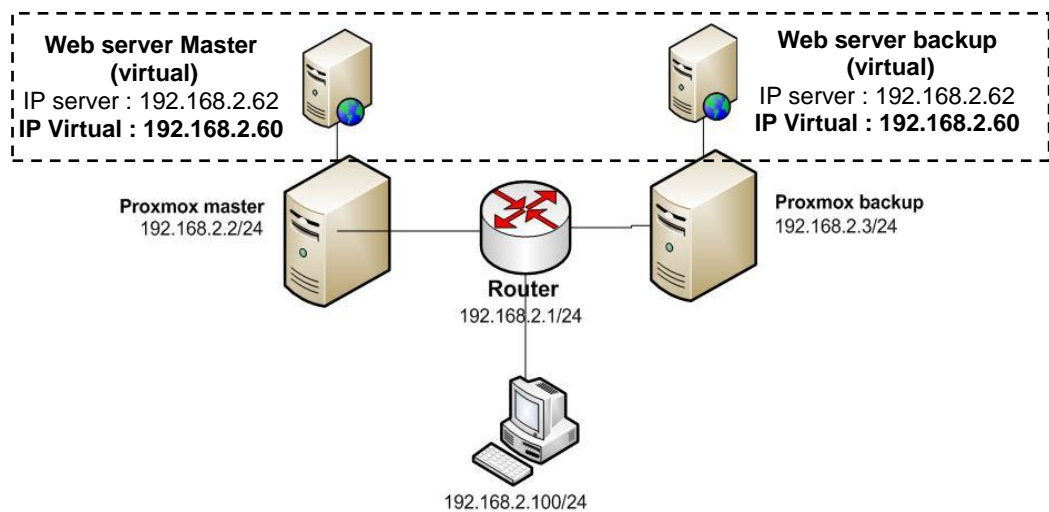
1. Komputer *Server Master* dan *Backup* menggunakan Proxmox VE 3.0

2. *Web Server* virtual menggunakan sistem Operasi Debian Squeeze, aplikasi apache2 versi 2.2.16-6, mysql-server versi 5.1.49-3 serta ucarp versi 1.5.1-1.

4.2. perancangan jaringan dan sistem

4.2.1. Perancangan Topologi

Topologi jaringan merupakan gambaran hubungan antar komputer dalam suatu jaringan komputer. Topologi jaringan penelitian yang penulis tawarkan adalah sebagaimana ditunjukkan pada Gambart 4.1



Gambar 4.1. Topologi jaringan penelitian

4.2.2. Perancangan Jaringan

Pada penelitian ini seluruh komputer baik server maupun klien terhubung ke sebuah *router* . Ketika klien ingin melakukan koneksi maka *router* akan meneruskan paket tersebut. Tipe jaringan yang digunakan untuk simulasi *cloud computing* adalah pada jaringan lokal atau LAN.

4.2.3. Perancangan Sistem

Pada perancangan sistem ini, Kedua *Server Proxmox VE* dibentuk *Cluster* untuk memudahkan administrasi sistem dari komputer klien melalui *web browser*. Mekanisme *failover webserver* ditangani oleh aplikasi *ucarp* sedangkan untuk replikasi *database* menggunakan MySQL-

Replication. Service yang berjalan ketika antara lain:

- a. Proxmox VE sebagai Hypervisor sistem operasi virtual
- b. apache2 sebagai daemon untuk layanan *Web Server*
- c. mysqld sebagai daemon untuk layanan replikasi *database*
- d. ucarp sebagai daemon yang memantau mekanisme failover
- e. sshd daemon untuk layanan secure shell (SSH)

4.2.4 Perancangan Redundansi

Perancangan redundansi merupakan perancangan yang menunjukkan konsep backup terhadap suatu sistem. Pada penelitian ini sistem yang dibuat redundant adalah aplikasi apache2 menggunakan aplikasi ucarp dengan skema *master-slave* serta *redundant database* menggunakan fitur MySQL-Replication dengan skema *master-master*.

4.3 Implementasi Rancangan Sistem

Tahap ini merupakan tahap tindak lanjut dari perancangan yang telah dibuat. Terdiri dari beberapa uraian langkah implementasi sistem dari awal hingga pengujian. Tahapan implementasi penulis jabarkan sebagai berikut:

4.3.1 Instalasi Hypervisor Proxmox VE

Instalasi *Proxmox* VE terdiri beberapa tahap, yaitu Pemilihan *Target Disk* Instalasi, *Timezone* dan *Region*, Administrasi *Password*, Informasi alamat IP dan Instalasi *paket* utama *Proxmox*. Informasi jaringan pada masing-masing server proxmox penulis sajikan pada Tabel 4.1

Tabel 4.1 Informasi IP Address Proxmox VE

No	Hostname	IP address	Netmask	Gateway	DNS
1	Proxmox1. uin.edu	192.168.2.2	255.255.255.0	192.168.2.1	8.8.8.8
2	Proxmox2. uin.edu	192.168.2.3	255.255.255.0	192.168.2.1	8.8.8.8

4.3.2 Konfigurasi Cluster pada Hypervisor Proxmox VE

```
proxmox1# pvecm create uin
proxmox2# pvecm add 192.168.2.2
```

4.3.3 Instalasi VM Sistem Operasi Debian menggunakan KVM

Instalasi Mesin virtual dengan cara login ke dashboard proxmox dengan alamat <https://192.168.2.2:8006>, lalu mengikuti langkah berikut:

- Upload file iso Debian* melalui menu upload pada tab storage
- Memasang mesin virtual Debian Squeeze melalui menu Create VM

4.3.4 Konfigurasi Jaringan

Konfigurasi alamat IP pada kedua *web server* tersaji pada Tabel 4.2

Tabel 4.2. Informasi alamat IP mesin virtual Web Server Debian

No	Hostname	IP address	Netmask	Gateway	DNS
1	webserver1. uin.edu	192.168.2.61	255.255.255.0	192.168.2.1	8.8.8.8
2	webserver2. uin.edu	192.168.2.62	255.255.255.0	192.168.2.1	8.8.8.8

4.3.5 Instalasi Paket webserver, database server, ucarp dan ntp

```
webserver# apt-get install apache2 php5 mysql-server mqsq-  
client phpmyadmin ucarp ntp
```

4.3.6 Konfigurasi Redundansi pada aplikasi ucarp

- Konfigurasi interface jaringan ucarp (webserver1 dan webserver2)

Konfigurasi dilakukan pada berkas `/etc/network/interfaces` dengan

menambahkan beberapa parameter sebagai berikut:

```
# Konfigurasi UCARP
ucarp-vid 1
ucarp-vip 192.168.2.60
ucarp-password "lulus"
ucarp-advskew 1
ucarp-advbase 1
ucarp-master yes
# Konfigurasi interface nic carp
auto eth0:ucarp
iface eth0:ucarp inet static
    address 192.168.2.60
    netmask 255.255.255.0
```

b. Konfigurasi sinkronisasi aplikasi *web server*

Konfigurasi sinkronisasi aplikasi *web server* penulis lakukan menggunakan sebuah skrip bash-shell bernama sync-apache-conf.sh.

Skrip utama bisa dilihat sebagai berikut:

```
#inisiasi server master
master_server="192.168.2.61"
current_date=$(date)
log_file="/var/log/sync-apache-conf.log"
#skrip utama sinkronisasi data
result=$(rsync -ai --delete --force
${master_server}:/etc/apache2/ /etc/apache2/)
rsync -ai --delete --force ${master_server}:/etc/ssl/
/etc/ssl/
rsync -ai --delete --force ${master_server}:/etc/hosts
/etc/hosts
rsync -ai --delete --force ${master_server}:/var/www/
/var/www/
```

4.3.7 Konfigurasi skrip pengujian Failover webserver

Untuk keperluan pengujian penulis membuat sebuah skrip php yang berfungsi menunjukkan IP Webserver dan IP pengakses. File php ini penulis letakkan pada webserver1 (otomatis tereplikasi ke webserver2. Skrip utama sebagai berikut:

```
<?php
    $ipwebserver = $_SERVER ['SERVER_ADDR'];
    $ipclient = $_SERVER['REMOTE_ADDR'];
?>
```

4.3.8 Konfigurasi redundansi MySQL-Replication

4.3.8.1. Konfigurasi Setting MySQL

Konfigurasi pertama replikasi database *master-master* terdapat pada file `/etc/mysql/my.cnf` pada `webserver1` sebagai berikut:

```
port=3306
server_id      = 1
log_bin        = /var/log/mysql/mysql-bin.log
log_bin_index  = /var/log/mysql/mysql-bin.log.index
log_slave_updates = 1
```

sedangkan konfigurasi pada `webserver2` dengan merubah `server_id = 2`

4.3.8.2. Konfigurasi Setup Replikasi database

Setup replikasi merupakan langkah untuk menentukan *agent* yang berperan dalam sinkronisasi *database* sebagaimana penulis rincikan sebagai berikut:

1) Membuat *user agent* replikasi pada `webserver1`

```
mysql> create user 'mysqlchkuser'@'localhost'
identified by 'mysql321';
mysql> create user 'mmm_monitor'@'%' identified by
'monitor_password';
mysql> create user 'agent_monitor'@'%' identified by
'agent_password';
mysql> create user 'replication'@'%' identified by
'replication_password';
mysql> GRANT REPLICATION CLIENT ON *.* TO
'mmm_monitor'@'%' IDENTIFIED BY 'monitor_password';
mysql> GRANT SUPER, REPLICATION CLIENT, PROCESS ON *.*
TO 'mmm_agent'@'%' IDENTIFIED BY 'agent_password';
mysql> GRANT REPLICATION SLAVE ON *.* TO
'replication'@'%' IDENTIFIED BY
'replication_password';
mysql> flush privileges;
mysql> flush tables with read lock;
```


Selanjutnya dump database pada webserver1 lalu restore ke database pada webserver2.

```
webserver1# ssh root@192.168.2.61
webserver1# mysqldump -u root -p --all-databases >
/tmp/database-backup.sql
webserver1# scp /tmp/database-backup.sql
root@192.168.2.62:/root/
webserver1# ssh root@192.168.2.62
webserver1# mysqldump -u root -p --all-databases
< /root/database-backup.sql
```

2) Setup database webserver2

```
mysql> flush privileges;
mysql> CHANGE MASTER TO master_host='192.168.2.61',
master_port=3306, master_user='replication',
master_password='replication_password',
master_log_file='mysql-bin.000043',
master_log_pos=1110;
mysql> start slave;
```

3) Setup database webserver1

```
mysql>CHANGE MASTER TO master_host='192.168.2.62',
master_port=3306, master_user='replication',
master_password='replication_password',
master_log_file='mysql-bin.000004',
master_log_pos=27957968;
mysql> start slave;
```

4.3.10 Konfigurasi Client

Konfigurasi jaringan *client* di dengan merubah alamat IP menjadi 192.168.2.100 dengan netmask 255.255.255.0 dengan gateway 192.1.68.2.1.

4.4. Pengujian Sistem

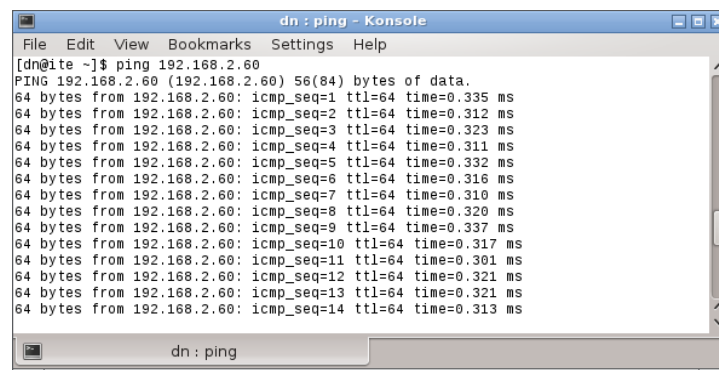
Pengujian sistem ini melewati lima tahap, yaitu Failover, Redundansi data, HTTP Service, MySQL-Replication dan Stress test.

4.4.1. Pengujian Failover

Pengujian failover merupakan pengujian untuk mengetahui fungsional IP Cluster yang terdiri dari sub uji Failover IP serta Failback.

4.4.1.1 Pengujian Failover IP

Guna memastikan IP *cluster* 192.168.2.60 sudah berjalan dilakukan pengujian ping ke webserver seperti pada gambar 4.2

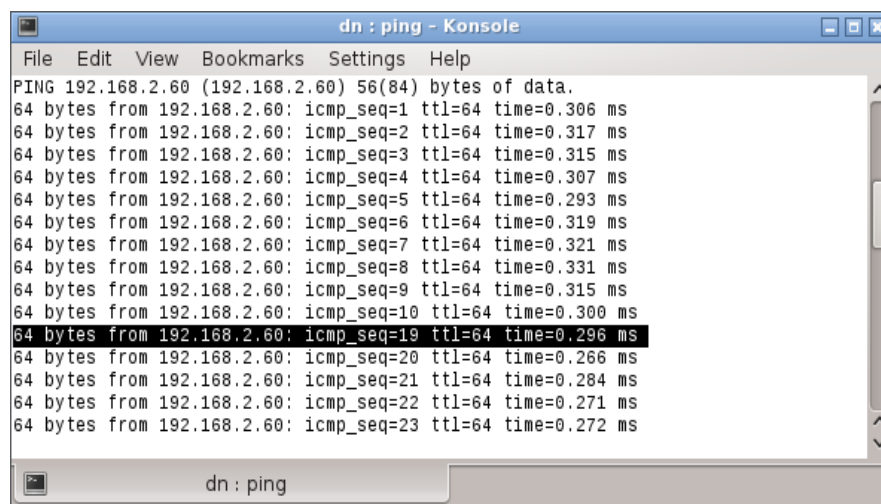


Gambar 4.2. Pengujian koneksi IP Cluster

Selanjutnya untuk menguji mekanisme *failover* berjalan lancar, penulis mematikan daya pada webserver1 menggunakan perintah berikut:

```
webserver1# poweroff
```

untuk memastikan terjadinya perpindahan IP dari webserver1 ke webserver2, pengujian dapat dilakukan dengan menggunakan utilitas ping seperti pada Gambar 4.3. Tampak disana terjadi jeda 10 detik.



Gambar 4.3. Proses jalannya failover

4.4.1.2 Pengujian IP Failback

Pengujian IP Failback bertujuan untuk memastikan IP virtual kembali ke webserver1 (master) setelah webserver1 kembali *online*. Ketika sudah berhasil kembali ke webserver1 maka interface akan tersaji seperti pada gambar 4.4

```

dn@ite:/home/dn
root@webserver1:~# ifconfig
eth0      Link encap:Ethernet  HWaddr da:30:7e:4a:6a:89
          inet addr:192.168.2.61  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::d830:7eff:fe4a:6a89/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:319  errors:0  dropped:0  overruns:0  frame:0
          TX packets:340  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:27900 (27.2 KiB)  TX bytes:31867 (31.1 KiB)

eth0:ucarp Link encap:Ethernet  HWaddr da:30:7e:4a:6a:89
          inet addr:192.168.2.60  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8  errors:0  dropped:0  overruns:0  frame:0
          TX packets:8  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:0
          RX bytes:560 (560.0 B)  TX bytes:560 (560.0 B)

root@webserver1:~#

```

Gambar 4.4. Interface jaringan pada webserver1

4.4.2. Pengujian Redundansi data

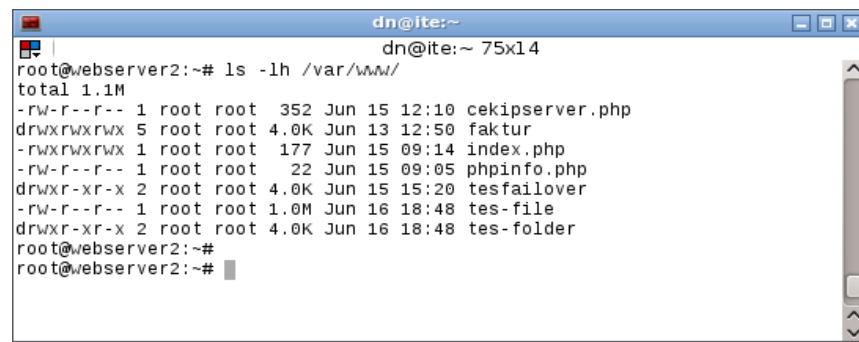
Pengujian redundansi data bertujuan untuk memeriksa sinkronisasi data antara webserver1 dan webserver2. Pada pengujian ini penulis membuat sebuah folder bernama tes-folder dan file dummy bernama tes-file pada webserver1 melalui perintah berikut:

```

webserver1# mkdir /var/www/tes-folder
webserver1# dd if=/dev/zero of=/var/www/tes-file bs=1M
count=10

```

Guna memastikan fungsi redundansi data pada folder dan file tersebut, penulis memastikan dengan perintah seperti pada Gambar 4.5



```

dn@ite:~
root@webserver2:~# ls -lh /var/www/
total 1.1M
-rw-r--r-- 1 root root 352 Jun 15 12:10 cekipserver.php
drwxrwxrwx 5 root root 4.0K Jun 13 12:50 faktur
-rwxrwxrwx 1 root root 177 Jun 15 09:14 index.php
-rw-r--r-- 1 root root 22 Jun 15 09:05 phpinfo.php
drwxr-xr-x 2 root root 4.0K Jun 15 15:20 tesfailover
-rw-r--r-- 1 root root 1.0M Jun 16 18:48 tes-file
drwxr-xr-x 2 root root 4.0K Jun 16 18:48 tes-folder
root@webserver2:~#
root@webserver2:~#

```

Gambar 4.5. Memeriksa keberadaan file dan folder pada webserver1

4.4.3. Pengujian HTTP Service

Pengujian *HTTP Service* bertujuan untuk mengetahui sejauh mana fungsi aplikasi apache2 dapat dijalankan oleh sistem *Failover cluster*. Pengujian ini menggunakan sebuah file utama cek-ip.php dengan tampilan menggunakan file cek-ip.html. Alamat URL 192.168.2.60/cek-ip.html akan menampilkan isi dari webserver1 seperti pada gambar 4.6.

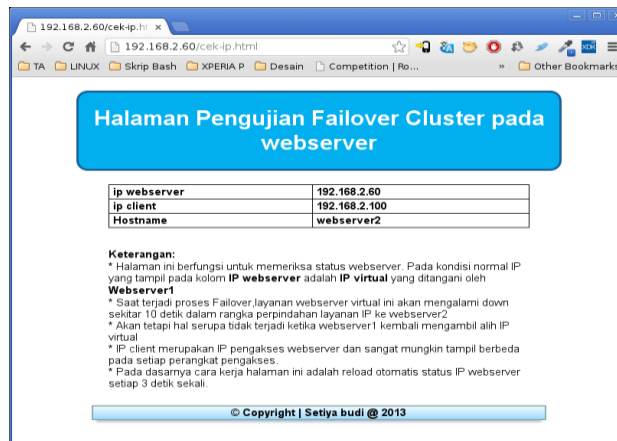


Gambar 4.6. Halaman pengujian webserver sebelum webserver1 mati

Pengujian selanjutnya adalah *Failover Webserver* dengan cara mematikan daya pada webserver1 menggunakan perintah berikut:

```
webserver1# poweroff
```

Setelah mengalami kurun waktu sekitar 10 detik, halaman pengujian akan menampilkan alamat IP Cluster dengan *hostname* webserver2 seperti pada gambar 4.6.



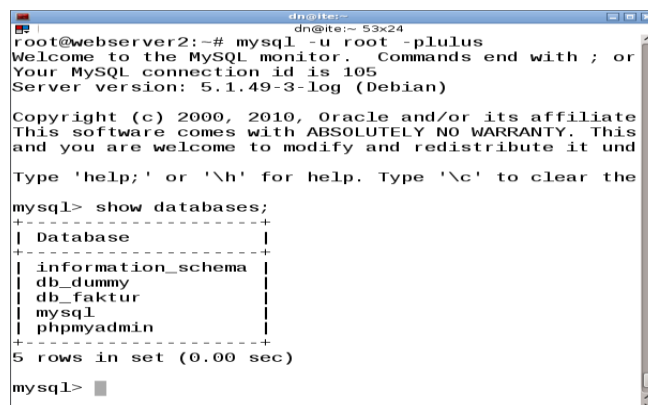
Gambar 4.6. Halaman pengujian webserver setelah webserver2 mengambil alih IP Cluster

4.4.4. Pengujian MySQL Replication

Skenario pengujian dengan membuat database dummy pada konsol MySQL webserver1 dengan perintah berikut:

```
mysql> create database db_dummy;
```

Selanjutnya periksa keberadaan database db_dummy pada webserver2 seperti tersaji pada Gambar 4.7



Gambar 4.7. Memeriksa keberadaan database dummy pada webserver2

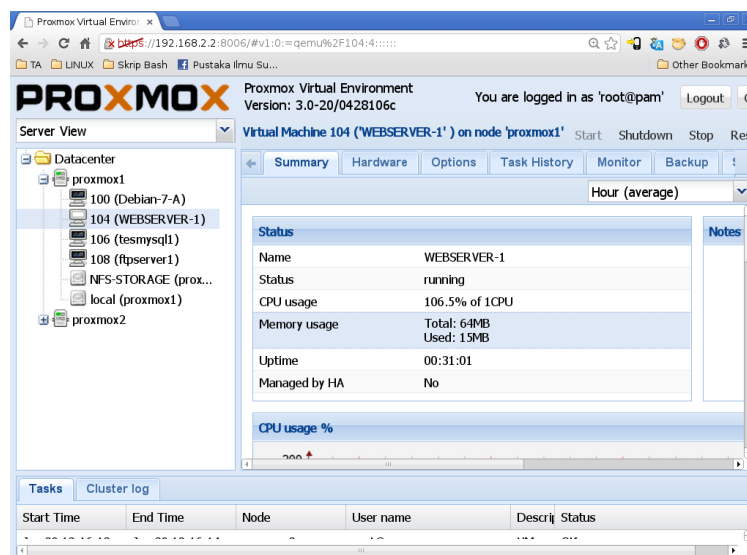
Pada gambar 4.7 tampak bahwa operasi replikasi database sudah berjalan dengan baik dari webserver1 ke webserver2.

4.4.5 Pengujian Stress Test

Stress test adalah pengujian untuk mengetahui kemampuan software dalam menangani kondisi tidak normal dari sisi volume/ kuantitas. Stress test pada pengujian ini menggunakan aplikasi stress. Mekanisme pengujian stress test menggunakan perintah sebagai berikut.

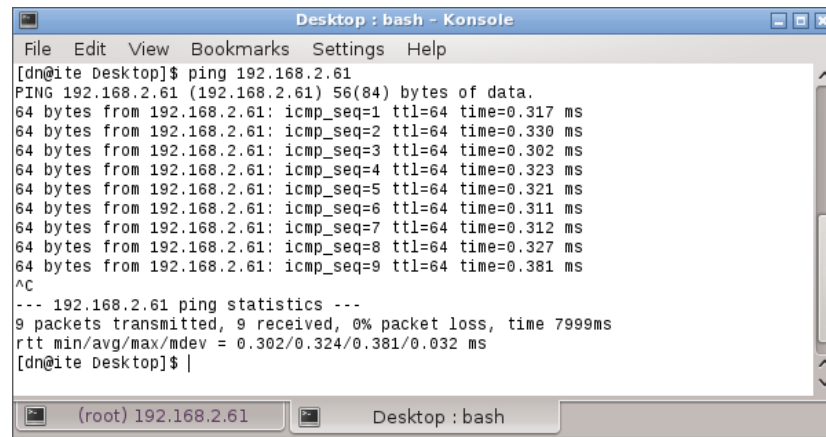
```
webserver1 #stress -v --cpu 2 --io 8 --vm 8 --vm-bytes 600M --
timeout 1m
```

Hasil dari pengujian stress test tersebut menunjukkan bahwa load CPU naik menjadi 100% lebih. Seperti ditunjukkan pada Gambar 4.8



Gambar 4.8. Hasil pengujian stress test

Pada gambar 4.8, pengujian stress test menunjukkan load CPU sebesar 106.5 %, nilai load CPU akan bertahan selama sekitar 1 menit. Pada saat yang bersamaan dilakukan pengujian pada jaringan dengan melakukan ping seperti pada gambar 4.9.



```

Desktop : bash - Konsole
File Edit View Bookmarks Settings Help
[dn@ite Desktop]$ ping 192.168.2.61
PING 192.168.2.61 (192.168.2.61) 56(84) bytes of data:
64 bytes from 192.168.2.61: icmp_seq=1 ttl=64 time=0.317 ms
64 bytes from 192.168.2.61: icmp_seq=2 ttl=64 time=0.330 ms
64 bytes from 192.168.2.61: icmp_seq=3 ttl=64 time=0.302 ms
64 bytes from 192.168.2.61: icmp_seq=4 ttl=64 time=0.323 ms
64 bytes from 192.168.2.61: icmp_seq=5 ttl=64 time=0.321 ms
64 bytes from 192.168.2.61: icmp_seq=6 ttl=64 time=0.311 ms
64 bytes from 192.168.2.61: icmp_seq=7 ttl=64 time=0.312 ms
64 bytes from 192.168.2.61: icmp_seq=8 ttl=64 time=0.327 ms
64 bytes from 192.168.2.61: icmp_seq=9 ttl=64 time=0.381 ms
^C
--- 192.168.2.61 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 7999ms
rtt min/avg/max/mdev = 0.302/0.324/0.381/0.032 ms
[dn@ite Desktop]$

```

Gambar 4.9. Hasil ping ketika dilakukan stress test

Berdasarkan Seluruh parameter pengujian tersebut, Pengujian sistem ini melibatkan praktisi ahli dari berbagai kalangan serta penguji user biasa yang akrab dengan linux, server, jaringan komputer serta para pengembang program. Daftar penguji ahli dapat dilihat pada Tabel 4.3.

Tabel 4.3. Daftar penguji ahli

No	Nama	Pekerjaan
1	Irwan Dwi Siswanto	Penanggungjawab IT CV Berkah Abadi
2	Abdur Rahman	Guru TKJ SMK TI Kartika Cendekia
3	Galih Setyawan	Web Developer / Programmer
4	M.Iqbal Jalaludin	IT Admin Gameloft Yogyakarta
5	Timor Bayu S	OSS Helpdesk Pemprov DIY

Sedangkan daftar penguji user biasa penulis sajikan pada Tabel 4.4.

Tabel 4.4. Daftar penguji user biasa

No	Nama	Pekerjaan
1	Dicky Martin Pramanta	Mahasiswa UAD / Linuxer
2	Estu Fardani	Mahasiswa UIN / Linuxer
3	Fendi Tri Cahyono	Web Developer / Programmer
4	Fathurrahman	Mahasiswa UIN / Linuxer

Tabel 4.4. Daftar penguji user biasa (Lanjutan)

No	Nama	Pekerjaan
5	Fuad Hasan	Mahasiswa UIN / Linuxer
6	Isnan Nugraha	Mahasiswa UIN / Linuxer
7	Irfak Dwi Lutfi	Mahasiswa UIN / Linuxer
8	Koespradono	Mahasiswa Akprind / Linuxer
9	M.Husna Mubarak	Mahasiswa UIN / Linuxer
10	R. Nur Faizin	Mahasiswa UIN / Linuxer

Total penguji pada Tabel 4.3 dan Tabel 4.4 berjumlah 15 orang Penguji yang terdiri dari penguji ahli dan user biasa. Seluruh penguji sistem melakukan Pengujian fungsional serta pengujian stress. Hasil Pengujian fungsional disajikan pada Tabel 4.5

Tabel 4.5 Daftar pertanyaan uji fungsional

No	Pengujian	Pilihan	
		Ya	Tidak
1	Apakah skenario pengujian IP Service dengan menonaktifkan webserver1 sebagai master IP Cluster berjalan lancar di webserver2?	15	0
2	Apakah skenario pengujian failback berjalan lancar ketika webserver1 ketika webserver1 kembali diaktifkan?	15	0
3	Apakah skenario Uji redundansi data dengan membuat folder dan data dummy pada htdocs berjalan lancar?	15	0
4	Apakah skenario pengujian Layanan apache2 dengan menonaktifkan webserver1 dapat di ambil alih oleh webserver2 dengan lancar?	15	0
5	Apakah skenario pengujian proses Failback layanan webserver dapat diambil alih oleh webserver1 ketika webserver1 kembali diaktifkan?	15	0
6	Apakah skenario pengujian replikasi database MySQL dari webserver1 ke webserver2 dan sebaliknya dapat berjalan dengan lancar?	15	0
Jumlah		90	0

Sedangkan hasil pengujian stress test disajikan pada Tabel 4.6

Tabel 4.6. Daftar pertanyaan stress test

No	Aspek	Pilihan			
		SS	S	KS	TS
1	Apakah hasil pengujian stress test sudah sesuai dengan harapan?	12	3		
2	Apakah solusi Cluster pada Webserver menjadi solusi yang tepat terhadap masalah downtime?	13	2		
3	Apakah konsep Cluster pada Webserver ini sudah berjalan sesuai harapan?	14	1		
Jumlah		39	6		

Keterangan:

SS : Sangat setuju

S : setuju

KS : Kurang Setuju

TS : Tidak setuju

Berdasar hasil dari dari pengujian yang dilakukan terhadap 15 responden yang terdiri dari penguji ahli dan user biasa pada Tabel 4.5 dan Tabel 4.6, hasil pengujian dapat dihitung sebagai berikut:

Pengujian Fungsional

Ya = $90/90 \times 100\% = 100\%$

Tidak = $0/100 \times 100\% = 0\%$

Pengujian Stress Test

$$SS = 39/45 \quad \times 100\% = 86.67 \%$$

$$S = 6/45 \quad \times 100\% = 13.33\%$$

$$KS = 0/45 \quad \times 100\% = 0\%$$

$$TS = 0/45 \quad \times 100\% = 0\%$$

Berdasarkan prosentase penilaian tersebut, uji fungsional dari responden menyatakan 100% sistem bekerja dengan lancar. Sedangkan pada uji stress test, 86.67 % dari 15 responden menyatakan Sangat Setuju dengan Sistem yang dibuat serta 13.33% menyatakan setuju. Dapat disimpulkan bahwa sistem *Cluster* pada *Web Server* berbasis *Cloud Computing* dapat berjalan dengan baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut:

- 1) Rancangan sistem Cluster pada *web server* berbasis *Cloud Computing* ini berhasil dilakukan dengan tahapan penelitian yakni Perancangan jaringan, Implementasi rancangan sistem dan pengujian sistem. Pada implementasinya sistem ini mampu meminimalisir masalah *downtime* dengan nilai 10 detik pada saat terjadi proses *failover*.
- 2) Sistem *Cluster* pada *web server* telah berhasil dibangun diatas platform virtual berbasis Proxmox yang didalamnya terpasang sistem operasi Debian Squeeze pada platform KVM. Hasil yang muncul cukup mumpuni untuk skala mesin virtual dalam hal pelayanan yang mendekati mesin server riil.

5.2 Saran

- 1) Berdasarkan penelitian yang telah dilakukan untuk sistem Cluster pada Webserver berbasis Cloud Computing ini disarankan pada penelitian kedepan menggunakan Hardware yang lebih mumpuni sekelas intel Xeon untuk menambah node *Cluster virtual* guna implementasi lebih luas, serta pemanfaatan jaringan cadangan (*redundant network*).
- 2) Dalam hal menangani masalah *downtime* seminimal mungkin, pada penelitian selanjutnya bisa digunakan mekanisme *Fencing* sebagai *agent*

pihak ketiga guna mendukung sistem *Cluster* dengan model Active-Active. Sistem ini kedepannya bisa digunakan untuk keperluan pelayanan sistem yang lebih luas dan request tinggi seperti Akademisi, kantor perusahaan besar maupun penyedia layanan *Cloud Computing*.

- 3). Pada penelitian selanjutnya bisa di kembangkan penelitian dengan kebutuhan replikasi data MySQL tidak hanya dengan kondisi master-master dengan dua node database saja. Pada kasus tertentu, kebutuhan replikasi data bisa berupa tipe multi master-multi slave, atau multi master- multi master.

DAFTAR PUSTAKA

- Afif, M.F. 2012. "*Implementasi Disaster Recovery plan dengan sistem Failover menggunakan DRBD dan Heartbeat pada Data Center FKIP UNS*". Surakarta: Universitas Sebelas Maret.
- Azikin, Askari. 2007. "*Debian GNU/Linux 2nd Edition*". Debian Indonesia. Sumber:
<http://directory.umm.ac.id/Operating%20System%20Ebook/Debian/Debian+GNULinux+2nd+Edition.pdf>
- Bel, Laurent, 2012. "*Simple Apache failover cluster on Ubuntu with config synchronization*". Sumber : <http://laurentbel.com/2012/04/12/simple-apache-failover-cluster-on-ubuntu-with-config-synchronization/>
- Kadir, Abdul, dan Terra Ch. Triwahyuni. 2003. "*Pengenalan Teknologi Informasi*". Yogyakarta: Penerbit Andi
- Komputer, Wahana. 2003. "*Konsep jaringan komputer dan pengembangannya*". Jakarta: Salemba Infotek
- Maitimu, Theddy R. 2008. "*Perancangan dan Implementasi Web Server Clustering dengan Skema Load Balance Menggunakan Linux Virtual Server Via NAT*". Salatiga: Universitas Kristen Satya Wacana.
- Mitchell, Bradley. "*Apache*". About.com. sumber:
http://compnetworking.about.com/cs/web_servers/g/bldef_apache.htm
- Mell, P., and Grance, T. 2011. "*The NIST Definition of Cloud Computing*". Computer Security Division Information Technology Laboratory National Institute of Standards and Technology. Gaithersburg USA , Sumber:
<http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf>.
- Muchtar, Akhyar. 2012. "*Implementasi Failover Clustering pada dua Platform yang berbeda untuk mengatasi kegagalan fungsi Server*". Makassar: Universitas Negeri Makassar.
- Natsirudin, M.A. 2011. "*Analisis pemanfaatan teknologi Cloud Computing pada jaringan Thin Client*". Yogyakarta: Amikom.
- Nugroho, Bunafit. 2004. "*PHP & mySQL dengan Editor Dreamweaver MX*". Yogyakarta: Penerbit Andi
- Oetomo, Budi Sutedjo Dharma. 2003. "*Konsep dan perancangan jaringan komputer*". Yogyakarta: Penerbit Andi

- Periyadi, dkk. 2009. "*Sistem Tersebar*". Bandung: Politeknik Telkom, Sumber: <http://repository.politekniktelkom.ac.id/Courseware/Semester%204/Sistem%20Tersebar/Produce/sister%20book.pdf>
- Purnomo, Nanang. 2012. "*Pemanfaatan Failover Cluster Server guna Recovery sistem pada PT.Lintas Data Prima*". Yogyakarta: Amikom
- Ramadhani, Graifhan. 2003. "*Modul Pengenalan internet*" sumber: http://directory.umm.ac.id/tik/pengenalan_internet.pdf
- Ridwan, M. 2012. "*Modul pembelajaran praktek Basis Data (MySQL) UNILA.*" Universitas Langlangbuana. Sumber: http://www.akmi-baturaja.ac.id/wp-content/uploads/2012/07/modul_mysql.pdf
- Rouse, Margaret. 2009. "*Ping*" Sumber: <http://searchnetworking.techtarget.com/definition/ping>
- Rouse, Margaret. 2009. "*Secure Copy (SSH)*". Sumber: <http://searchsecurity.techtarget.com/definition/Secure-Shell>
- Rouse, Margaret. 2009. "*Windows Azure*" sumber: searchCloudComputing.techtarget.com/definition/Windows-Azure
- Rouse, Margaret. 2006. "*hypervisor*" sumber: <http://searchServervirtualization.techtarget.com/definition/hypervisor>
- Santoso, Berkah. I. 2012. "*Perkembangan Virtualisasi*" sumber: <http://www.Cloudindonesia.or.id/wp-content/uploads/2012/06/E-Book-Perkembangan-Virtualisasi.pdf>
- Solichin, Achmad. 2009." *Pemrograman Webdengan PHP dan MySQL*". Universitas Budi Luhur, Jakarta. Sumber: <http://achmatim.net/2009/04/15/buku-gratis-pemrograman-web-dengan-php-dan-mysql/>
- Sufehmi, Harry. 2009. "*Pegenalan Virtualisasi*". Asosiasi Open Source Indonesia (AOSI). Sumber: <http://harry.sufehmi.com/wp-content/uploads/2009/06/pengenalan-virtualisasi.pdf>
- Sugianto, Masim Vavai. 2011. "*Belajar Teknologi Virtualisasi : VMWare vSphere Hypervisor™ ESXi (1)*". sumber: <http://www.excellent.co.id/product-services/vmware/belajar-teknologi-virtualisasi-vmware-vsphere-hypervisor%E2%84%A2-esxi-1/>

- Sugianto, Masim Vavai. 2010. "*Panduan Linux HA & Failover pada openSUSE/SLES*". Sumber: <http://www.vavai.com/wp-content/uploads/panduan-high-availability-server-menggunakan-opensuse-sles.pdf>
- Sugianto, Masim Vavai. 2011. "*Panduan Virtualisasi & Cloud Computing pada Sistem Linux*". Sumber: <http://vavai.com/wp-content/uploads/2011/01/Implementasi-Virtualisasi-Cloud-Computing-pada-Sistem-Linux.pdf>
- Syafii, Ahmad. 2005. "*Aplikasi Database dengan PHP5 MySql PostgreSQL Oracle*". Yogyakarta: Penerbit Andi.
- Syafrizal, Melwin. 2005. "*Pengantar Jaringan Komputer*", Yogyakarta: Penerbit Andi,
- Utdirartatmo, Firrar. 2004. "*Clustering PC di Linux dengan OpenMosix & Cluster Knoppix*". Yogyakarta: Penerbit Andi,
- http://compnetworking.about.com/cs/webservers/g/bldef_apache.htm diakses 1 maret 2013
- <http://opennebula.org/about:about> diakses pada 2 januari 2013
- http://pve.proxmox.com/wiki/proxmox_ve_2.0_cluster diakses pada 3 januari 2013.
- <http://www.freebsd.org/doc/handbook/carp.html> diakses pada 2 januari 2013
- <http://www.netbsd.org/docs/guide/en/chap-carp.html> diakses pada 2 januari 2013
- <http://www.openstack.org/Software/openstack-compute/> diakses pada 16 maret 2013
- <http://www.proxmox.com/products/proxmox-ve> diakses pada 1 april 2013.
- <http://www.pureftpd.org/project/ucarp> diakses pada 28 mei 2013
- <http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap9sec95.html> diakses pada 13 maret 2013
- <http://www.vmware.com/files/pdf/VMware-ESX-and-VMware-ESXi-DS-EN.pdf> diakses pada 12 maret 2013.
- http://www.webdevelopersnotes.com/basics/what_is_web_Server.php diakses pada 5 maret 2013