Seminar Nasional Instrumentasi, Kontrol dan Otomasi (SNIKO) 2015
Bandung, Indonesia, 10-11 Desember 2015

Perancangan dan Implementasi *High Availability Cluster Web Server Berbasis DBRD dan Heartbeat*

[1]Mohammad Iwan Wahyuddin*), [2]Andrianingsih, [3]Muhammad Jejen

[1]FTKI, Universitas Nasional

[2]FTKI, Universitas Nasional

iwan_wyd@yahoo.com)*

Abstrak

Sebuah perusahaan sering kali mempromosikan dan bahkan menjual produk- produknya melalui website dengan memanfaatkan internet dengan tujuan memperbesar pangsa pasarnya. Website memerlukan server berkualitas agar dapat berjalan dengan baik, jika server dari website mengalami kegagalan seperti kerusakan pada harddisk atau yang lainnya, situs pada server tersebut pasti tidak akan dapat di akses oleh konsumen yang ingin mendapatkan informasi yang disediakan. Untuk menghadapi masalah tersebut, dibuatlah suatu usulan untuk membangun suatu sistem yang dapat menjamin ketersediaan informasi pada server web dengan menggunakan teknologi cluster. Cluster mempunyai berbagai metode salah satunya High Availability Cluster yaitu cluster yang di implementasikan dengan tujuan utama untuk meningkatkan Availabilitas layanan yang disediakan cluster. Penulis akan membuat virtualisasi web server di linux Ubuntu yang di dalamnnya terdapat DRBD dan Heartbeat dengan menggunakan sistem operasi Proxmox VE. DRBD merupakan suatu aplikasi replikasi storage block device antar 2 buah server, yang memungkinkan melakukan sinkronisasi 2 server dengan metode Uptime, Synchronous dan Asynchronous. Sedangkan Heartbeat adalah sebuah aplikasi yang dapat mendeteksi apabila server utama down maka Heartbeat akan secara otomatis mengarahkan peran server utama kepada server backup. Dengan adanya DRBD sebagai backup data dan Heartbeat sebagai pengecek kegagalan fungsi server membuat data menjadi aman.

Kata Kunci: High Availability Cluster, DRBD, Heartbeat, Virtualisasi

1. Pendahuluan

Pemanfaatan teknologi *high availability cluster* ini dirancang untuk menunjang layanan infrastruktur *web server. Web server* merupakan *software* yang memberikan layanan data yang berfungsi menerima permintaan *HTTP* atau *HTTPS* dari klien atau *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman–halaman *web* yang umumnya berbentuk dokumen HTML. Pada penelitian ini akan dirancang virtualisasi *web server* di linux Ubuntu dengan menggunakan sistem operasi *Proxmox VE*. Sistem operasi *Proxmox VE* merupakan distribusi berbasis debian yang mempunyai model penyimpanan sangat fleksibel dan tidak ada batasan untuk melakukan instalasi sistem operasi dan juga dapat mengkonfigurasi banyak definisi penyimpanan yang diinginkan oleh admin. Beberapa kelebihan dari sistem *server* yang bersifat *virtual server* yaitu sistem tidak terlalu berlebihan dalam menggunakan *source CPU-usage* dan kemudahan dalam melakukan *Backup dan Recovery*.

High Availability cluster atau sering disebut sebagai *Failover Cluster System,* pada umumnya diimplementasikan untuk tujuan meningkatkan ketersediaan layanan yang diberikan oleh *cluster*. *Cluster* merupakan kolaborasi beberapa komputer independen yang digabungkan menjadi satu sistem dengan menggunakan jaringan dan *software*. Dalam pengertian mendasar, sistem cluster diartikan dengan dua atau lebih komputer yang digunakan untuk menyelesaikan satu masalah secara bersama-sama. Elemen *cluster* akan bekerja dengan memiliki node redundan, yang kemudian digunakan untuk menyediakan layanan saat salah satu elemen *cluster* mengalami kegagalan. Ukuran yang paling umum dari kategori ini adalah dua node, yang merupakan syarat minimum untuk melakukan redudansi. Dimana implementasi *cluster* jenis ini nantinya akan diaplikasikan dengan menggunakan redudansi komponen *cluster* untuk menghilangkan kegagalan di suatu titik menggunakan Linux Ubuntu *server* yang didalamnya terdapat *Heartbeat* dan *DRBD (Distributed Replicated Block Device)* untuk meningkatkan ketersediaan layanan yang disediakan oleh *web server*.

1.1. Permasalahan

Database storage pada sebuah server tunggal masih rentan terhadap kerusakan yang mengakibatkan kehilangan data, untuk mengatasinya perlu dibuat suatu sistem agar kehilangan data akibat adanya kegagalan sistem server dapat dihindari. Dengan menggunakan beberapa server sebagai backup, secara ideal hal

ini memerlukan penggunaan *resource hardware* yang cukup besar dalam perancangan sistem *high availability* pada server fisik. Untuk itu, maka pada penelitian ini dibuat beberapa server virtual dan akan dianalisis sejauh mana respon server backup dalam menangani masalah yang terjadi pada server utama (primer). Baik pada keadaan normal saat proses maintenance atau update perangkat, maupun saat terjadi gangguan dari luar sistem.

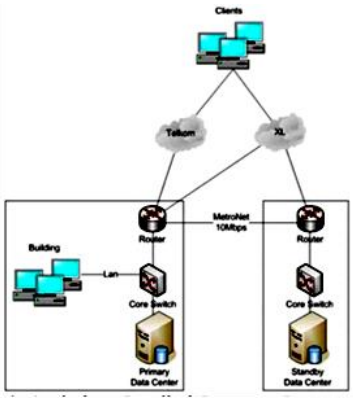
1.2. Tujuan dan Manfaat

Adapun tujuan peneltitian ini adalah untuk memanfaatkan package DRDB Linux untuk memudahkan backup data *server*, sehingga dapat dibangun web server yang memiliki ketersediaan layanan tinggi (*High availability*) yang bertujuan agar user dapat mengakses *web server* tanpa adanya masalah.

Disisi lain, ke-2 unit *web server tersebut* selalu sinkron satu sama lain, ketika *server-1* update, maka *server*-2 akan *update* secara *otomatis*. Sehinga terdapat *content* data yang selalu melakukan sinkronisasi pada masing-masing server. Untuk keperluan ini maka perlu dibuat mekanisme redudansi data untuk menguji sinkronisasi kedua server. Perancangan sistem *Heartbeat* sebagai *failover* dapat mengatasi adanya kegagalan pada fungsi *server*, yaitu dengan cara menggantikan server-1 jika tiba-tiba mati atau *down,* sehingga secara otomatis server-2 akan mengagantikan fungsi *server-1* agar sistem tetap berjalan.

2. Tinjauan Pustaka

Solusi *high availability* dapat meminimalkan dampak gangguan terhadap proses bisnis pada suatu perusahaan yang merupakan dampak dari gangguan *database server* dan *storage server* [2].



Gambar 1. Arsitektur logikal *disaster recovery center (DRC)*

Gambar 1. Adalah sebuah contoh yang memperlihatkan bagaimana kantor-kantor cabang terhubung dengan data center utama melalui jaringan WAN yang disediakan oleh dua provider telekomunikasi. Dalam keadaan normal, dua jaringan ini berperan sebagai *load balancer* dan apabila terjadi gangguan pada salah satu jaringan, maka seluruh arus informasi dan komunikasi akan ditangani oleh jaringan yang tidak bermasalah. Data yang berasal dari *cloud provider* diteruskan masuk ke *router* pada *data center* utama, dan jaringan yang masuk ke router DRC. Sinkronisasi database antara data center utama dan DRC dilakukan melalui koneksi antar router. Apabila sewaktu-waktu terjadi gangguan pada *data center* utama, yang mengakibatkan kantor cabang tidak dapat mengakses *data center* utama, maka jaringan *standby* dari *cloud provider* ke DRC akan diaktifkan dan *Data Guard* akan mengubah peran *standby database* pada DRC menjadi *primary database*. Selanjutnya koneksi dari kantor cabang ke *data center* utama akan dialihkan menuju DRC.

Penelitian High Availability Cluster yang dilakukan oleh Muhammad Taufik Saenal [3], menggunakan komponen dalam sistem terhubung melalui jaringan dimana ada dua buah *server* yang menyediakan *service* untuk *client*. Kedua *server* saling terhubung melalui dua buah jaringan. Salah satu jaringan merupakan jaringan publik yang menghubungkan kedua *server* dengan *client*, sedangkan jaringan lainnya merupakan jaringan *private* yang menghubungkan *server* secara langsung untuk duplikasi data antar kedua *server*. Aktif *server* adalah *server* utama yang melayani service untuk *client* dan pasif *server* adalah *backup* dari aktif *server*. Kedua *server* di atas adalah satu kesatuan sistem yang melayani permintaan service dari *client*, dimana jika terjadi kesalahan pada aktif *server* maka pasif *server* akan mengambil alih peran dari aktif *server.*

Penggunaan *Distributed Replicated Database System* (DBRD) pada *High Data Availability* menunjukkan suatu program replikasi secara otomatis dan multi database. Replikasi akan dilakukan, dan akan ditampilkan pada *interface server* yang memperlihatkan data setiap saat perubahan-perubahan yang terjadi pada database di *server utama (primer)* dan *server* backup. Replikasi database adalah membuat cadangan *database* pada server lainnya yang dikonfigurasi sebagai server cadangan [4].

Server-server fisik tentunya memiliki banyak kekurangan khususnya dalam masalah *single point of failure,* ketika suatu layanan mati dan tidak memperhitungkan kesalahan yang berhubungan dengan redundansi. Proses Replikasi antar server dilakukan dengan cara mereplikasi database pada masing-masing

server. Terdapat *database master* dan *database slave* yang memiliki fungsi yang sama yaitu untuk penyimpanan *database* dari suatu sistem. kedua *server* itu dihubungkan dengan tujuan agar bisa terjadi replikasi data antara kedua *server.* Agar perancangan itu bisa dilakukan dengan baik digunakan teknologi MySQL Cluster. Pada tahapan pembuatan sistem *cluster* dibagi tiga bagian utama yaitu *server master, server slave,* dan manajemen *server*. Antara node A yang berfungsi sebagai *server master* dan node B sebagai *server slave* saling terhubung dengan suatu *management node* (MGM Node). *Heartbeat* bekerja untuk jaringan yang akan mengkomunikasikan status server (node) setiap waktu. Ketika suatu node mengalami gangguan sistem *database*, proses migrasi mesin virtual akan dilakukan ke *node* lainnya. Ketika *node* yang mengalami gangguan tersebut sudah diperbaiki dan kembali hidup proses migtasi kembali dilakukan untuk mengembalikan mesin *virtual* yang telah dimigrasikan ke *node* asalnya [5].

Jadi fungsi utama dari MGM Node adalah menghubungkan antara kedua node tersebut sehingga fungsi dari MySQL cluster itu sendiri dapat berjalan. Di samping fungsi lainnya yang telah disebutkan seperti replikasi data, dan *high availability*.

3. Perancangan Sistem

Implementasi pada perancangan sistem ini menggunakan server untuk keperluan virtualisasi, adapun perangkat keras yang dibutuhkan sebagai berikut:

a. Komputer server:
   - Processor Intel Core i3 2320 TM 3,3 MHz
   - Memory DDR3 4 GB
   - Hardisk 250 GB

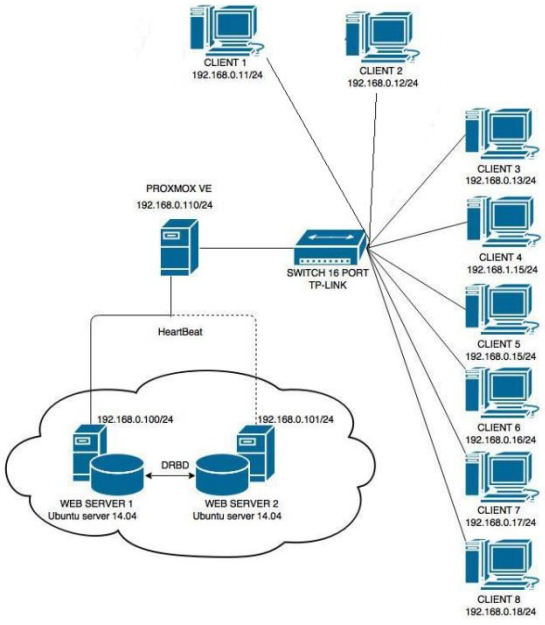
b. Komputer Client minmal Pentium IV dengan dilengkapi browser web.
c. Switch 16 Port
d. Kabel Jaringan (UTP)

Perangkat lunak yang diperlukan adalah sebagai berikut:

a. Proxmox VE yang digunakan untuk membuat Virtualisasi OS Server sebagai konfigurasi High Availiability cluster.
b. Ubuntu Server 14.04 sebagai OS dalam virtualisasi server.
c. MYSQL Sebagai database web server.
d. Apache Sebagai Web Server.
e. PHP5 Sebagai bahasa pemrograman web
f. Heartbeat *software* untuk melakukan HA *Cluster*.
g. DRBD *Software* untuk sinkronisasi data pada kedua server berbasis web.

3.1. Topologi Jaringan

Topologi jaringan yang digunakan pada perancangan *High Availability Web server* ini menggunakan topologi star. Topologi star adalah bentuk topologi jaringan yang berupa konvergensi dari node tengah ke setiap node atau pengguna. Beberapa kelebihan yang dimiliki topologi star diantaranya adalah meminimalisir kerusakan pada satu saluran dan hanya akan mempengaruhi jaringan pada saluran tersebut dari stasion yang terpaut. Tingkat keamanan termasuk tinggi dan tahan terhadap lalu lintas jaringan yang sibuk. Di samping itu penambahan dan pengurangan stasion dapat dilakukan dengan mudah karena mempunyai akses Kontrol terpusat. Kelebihan lainnya yaitu kemudahan deteksi dan isolasi dalam hal pengelolaan jaringan termasuk jika terjadi kerusakan pada jaringan. Namun di samping kelebihan yang dimiliki, terdapat juga kelemahan seperti jika node tengah mengalami kerusakan, maka seluruh rangkaian akan berhenti, boros dalam pemakaian kabel, HUB/SWITCH jadi elemen kritis karena kontrol terpusat dan peran hub sangat sensitif sehingga ketika terdapat masalah dengan hub maka jaringan tersebut akan down biaya jaringan lebih mahal dari pada Bus atau Ring



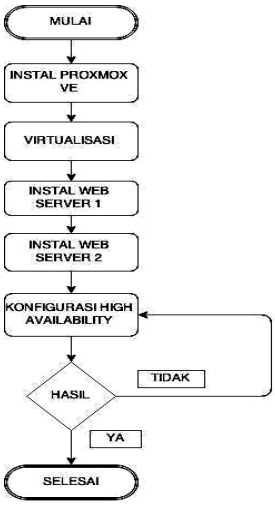
Gambar 2. Topologi Jaringan

Pada gambar 2 menjelaskan topologi jaringan High Avaibility pada web server pada jaringan *local area network* (LAN). Dari topologi tersebut terdapat komputer *server* yang terinstal Proxmox VE dengan IP 192.168.0.110 yang didalamnya

terdapat 2 server atau 2 node virtual yang berfungsi sebagai *web server*. *Node1* sebagai *Master* atau *Primary* dengan alamat IP 192.168.0.101 dan *node2* sebagai *Slave* atau *Secondary* dengan alamat IP 192.168.0.102. Kedua *node* tersebut saling berhubungan dan saling sinkron satu sama lain oleh DRBD didalam *cluster*. Fungsi dibuatnya 2 *node* pada jaringan diatas adalah ketika *node1* mengalami kegagalan fungsi *server* maka secara otomatis *node2* menggantikan fungsi dari *node*1. *Heartbeat* akan senantiasa mengecek setiap saat jika sebuah *node* mengalami kegagalan fungsi *server.*

Diagram alir dari perancangan sistem diperlihatkan pada gambar 3. Adapun langkah-langkahnya dijelaskan sebagai berikut:

1. Pada proses pertama, install OS PROXMOX VE untuk menginstal virtualisasi *Web server* 1 dan *Web server* 2.
2. Berikutnya proses Instalasi *Web server* 1 dan *Web server* 2 menggunakan Virtualisasi Proxmox VE hypervisor.
3. Selanjutnya konfigurasi *High Availability* atau *Failover Cluster* untuk menggabungkan *Web server* 1 dan *Web server* 2 kedalam *cluster*.
4. Melakukan instalasi dan konfigurasi *Heartbeat* dan DRBD pada *Web server* 1 dan *web server* 2.
5. Konfigurasi *High Availability* pada kedua *Web server*.

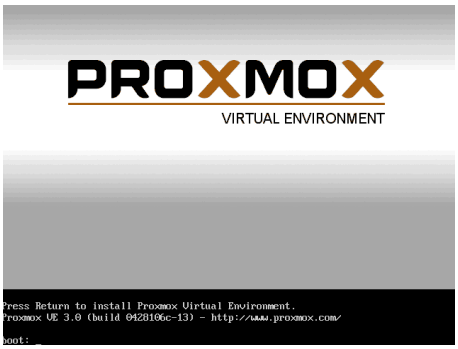


Gambar 3. Flowchart perancangan sistem

3.2. Instalasi PROXMOX VE

Proxmox merupakan operating sistem yang berfungsi sebagai hypervisor dan dapat didownload secara gratis pada situsnya di alamat http://www.proxmox.com/downloads/proxmox dalam bentuk format .ISO kemudian dari file tersebut *burning* ke dalam CD menggunakan CD burner, seperti Nero, dll

Proses instalasi Proxmox tidak berbeda dengan cara instalasi pada disribusi linux pada umumnya, hanya saja, pada Proxmox diperlukan mesin 64 bit dengan dukungan virtualisasi. Untuk *storage* diperlukan hardisk yang partisinya dipergunakan secara utuh karena pada saat instalasi, secara otomatis seluruh isi harddisk akan terhapus. Gambar 4 dibawah adalah tampilan saat pertama kali instalasi Proxmox.



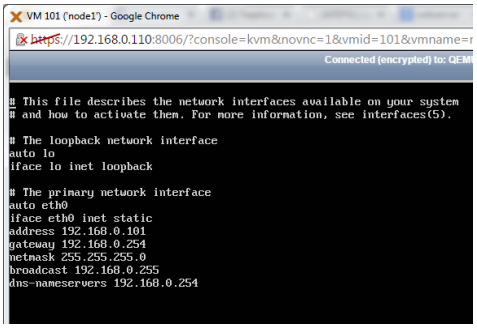
Gambar 4. Tampilan awal instalasi Proxmox

3.3 Instalasi *Heartbeat*

Setelah menginstall 2 buah virtual machine linux Ubuntu server 14.04 pada Proxmox VE. Tahap selanjutnya adalah menginstall heartbeat untuk komponen pendukung dalam membangun *High Availability,* yang berfungsi sebagai failover. Langkah pertama yang dilakukan adalah update dan upgrade package sources pendukung untuk mempermudah peroses penginstallan. Dengan perintah sebagai berikut:

```
#sudo apt-get update
#sudo apt-get upgrade
```

Tahap berikutnya adalah setting network di masing-masing server node1 dan node2. Setel IP node1 dengan IP *192.168.0.100* dan IP node2 dengan IP *192.168.0.101* seperti pada gambar 5 di bawah.



Gambar 5. Setting network node server

Tahap selanjutnya install aplikasi Heartbeat pada masing – masing server, dengan perintah:

```
#Sudo-apt install heartbeat
```

3.4. Konfigurasi *Heartbeat*

Setelah install *heartbeat* tahap selanjutnya adalah tahap konfigurasi heartbeat pada ke dua server node1 dan node 2 sebagai berikut :

a. Membuat berkas baru yaitu ha.cf dengan perintah:

```
#Nano/etc/ha.d/hacf
```

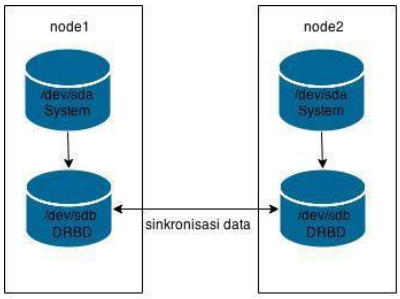
b. Kemudian Isi berkas di atas dengan script di bawah ini agar Heartbeat mengenali kedua node server tersebut:

```
Keeplive 2
Warntime 5
Deadtime 15
Initdead 90
Udpport 694
Auto failback on
Bcast eth0
Node node 1 node 2
```

Setelah itu dibuat berkas untuk IP virtual pada Heartbeat di dalam folder */etc/ha.d/haresources*. *Heartbeat* nantinya akan memberikan IP virtual tersebut pada server node1 dan node2, dan client hanya akan dilayani oleh IP virtua tersebut. Heartbeat selalu mengecek antara dua server, bila salah satu down maka IP virtual akan pindah Ke server yang aktif.

3.5. Instalasi DRBD (Distributed Replicated Block Device)

Proses instalasi DRBD (Distributed Replicated Block Device) diperlukan penambahan hardisk virtual Proxmox VE pada server node1 dan node2 dengan ukuran yang sama. Hardisk ini akan di gunakan sebagai hardisk *cluster* DRBD.



Gambar 6. Harddisk pada DRBD

Terlihat di gambar 6, di mana terdapat dua buah hardisk pada setiap node yang terdiri dari /dev/sda sebagai file system dari Ubuntu server 14.04 dan /dev/sdb sebagai cluster DRBD. Kemudian Install paket DRBD pada masing-masing server node1 dan node2 dengan perintah :

```
#apt-get install drbd82-utils
 drbdlinks\
```

Perintah di atas akan menginstal paket DBRD 8 dari repository server linux Ubuntu. Karena semua proses dilakukan secara online, maka pada saat instalasi server harus terhubung ke jaringan Internet.

3.6. Konfigurasi DRBD (Distributed Replicated Block Device)

Selanjutnya dilakukan proses konfigurasi server dengan pengeditan pada file drbd.conf. Konfigurasi ini dimaksudkan untuk membuat sinkronisasi antar server. Kemudian tentukan server node1 menjadi primer dan server node2 sebagai secondary atau backup untuk perangkat yang akan memuat file konfigurasi paket LAMP dan dilakukan sinkronisasi penuh untuk pertama kali antara kedua server. Jika proses sinkronisasi telah selesai, kita perlu memformat drive drbd dan mounting ke salah satu direktori yang dilakukan di server primer.

```
#[node1] mkfs.ext4 /dev/drbd0
#[node1] mount /dev/drbd0 /srv/data
```

3.7. Instalasi LAMP (Linux, Apache, Mysql, Php)

Paket instalasi LAMP (Linux, Apache, Mysql, PHP) merupakan komponen penting pendukung untuk membuat Web server.Tujuannya adalah memiliki service heartbeat yang mengatur service dari paket rangkaian LAMP dan mencegah rangkaian LAMP ini dari aturan normal yang diatur dalam init. Selama proses penginstallan lakukan pengisian user dan password untk mengakses database *MySQL*.

Setelah instalasi LAMP server selesai, lakukan pemindahan semua direktori tempat database dan web servernya ke DRBD storage, hal ini dibutuhkan agar web server dapat saling sinkron dari server node1 dan server node2 tapi sebelumnya hentikan terlebih dahulu service-nya. Lakukan pada server primary atau node1 terlebih dahulu baru node 2 setelahnya. Lakukan pengapusan LAMP dari script init dan pindahkan kembali semua paket LAMP ke direkrori /srv/data yang berada pada storage DRBD. Hal yang sama dilakukan untuk server

node2. Dengan melepas mounting /dev/drbd0 dari /srv/data, membuat kedua node sebagai secondary DRBD. Jalankan kembali Heartbeat. Lakukan restart atau reboot untuk kedua server yang telah dikonfigurasi tersebut.

4. Pengujian sistem

Pengujian sistem pada perancangan *High Availibility web server* ini akan dilaksanakan dalam beberapa tahap pengujian sebagai berikut:

4.1. Pengujian Sinkronisasi Data

Sebelum pengujian sinkronisasi dilakukan terlebih dahulu dicek kecepetan menulis dan membaca pada harddisk / penyimpanan, dan diperoleh bahwa kecepatan membaca memory 4,398.42 MB/s, sementara Kecepatan rata – rata membaca harddisk 210.51 Mb/s. yang menggambarkan bahwa kecepatan membaca pada memori sekitar 20x lebih tinggi daripada kecepatan membaca hardisk.

Pada pengujian sinkronisasi data yang telah diuji sebesar 100MB, 500MB dan 1GB didapatkan hasil sebagai berikut:

Table 4.2 Sinkronisasi data sebesar 100MB

No	Data	Kecepatan sinkronisasi (Mb/s)
1	100MB	64.10
2	100MB	51.29
3	100MB	53.48
4	100MB	64.10
5	100MB	42.47
6	100MB	45.25
7	100MB	56.18
8	100MB	46.51
9	100MB	37.41
10	100MB	46.37

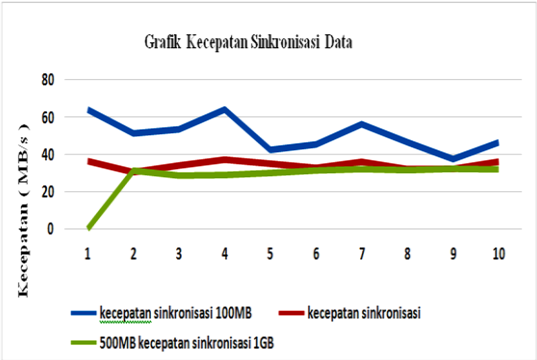
Table 4.3 Sinkronisasi data sebesar 500MB

No	Data	Kecepata sinkronisasi (Mb/s)
1	500MB	36.3
2	500MB	30.4
3	500MB	34.0
4	500MB	37.2
5	500MB	35.0
6	500MB	32.8
7	500MB	36.0
8	500MB	32.1
9	500MB	32.2
10	500MB	36.1

Table 4.4 Sinkronisasi data sebesar 1GB

No	Data	Kecepata sinkronisasi (Mb/s)
1	1GB	32.4
2	1GB	31.3
3	1GB	28.7
4	1GB	28.9
5	1GB	30.0
6	1GB	31.4
7	1GB	32.1
8	1GB	31.5
9	1GB	32.2
10	1GB	31.9

Dari data di atas kita dapat menganalisa situasi yang terjadi pada *storage cluster*. Semakin besar data yang dirubah semakin lama waktu *sinkronisasi*. Pada proses penyalinan data dengan metode sederhana, dengan menyalin file yang besar secara langsung akan berdampak pemakaian memori yang besar. Dibandingkan dengan perubahan yang kecil kemudian dibackup akan mendapatkan hasil lebih cepat.



Gambar 7. Grafik kecepatan data sinkronisasi

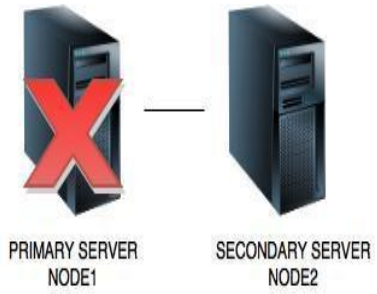
Pada gambar 7. dapat dilihat waktu kecepatan sinkronisasi data kedua *server* bergantung dengan ukuran data yang diubah pada server utama, semakin kecil data akan semakin cepat selesai melakukan sinkronisasi, begitupun sebaliknya.

4.2. Pengujian *Failover* dan *Failback*

Pengujian ini dilakukan dengan percobaan *failover* dan *failback* menggunakan *Heartbeat* yang menghasilkan IP address virtual yang melayani *service* pada server yang di *cluster*.

Pengujian *Failover* terjadi jika pada salah satu *server failure*. *Heartbeat* akan memberikan IP virtual pada server node1 dan node2. Yang akan diakses oleh client. *Heartbeat* mempunyai peran penting dalam *failover*. Karena *Heartbeat*

selalu mengecek antara dua server bila salah satu down.



Gambar 8. *Failover* pada node1 dan node2

Gambar 8. Di atas memperlihatkan bagaimana proses fail over ini dilakukan. Dari hasil pengujian didapatkan perpindahan server yang dilakukan oleh heartbeat memakan waktu rata-rata selama 3 detik. Pada pengujian *failback* atau proses pengembalian fungsi server *primary* (node1) dari server *secondary* (node2) meliputi perubahan yang terjadi pada server node2 selama menggantikan fungsi dari server node1. Layanan yang diberikan aplikasi Heartbeat memudahkan proses *failback* agar data selalu sinkron satu sama lain. Proses *failback* berjalan dengan memindahkan pelayanan *service* dari node2 ke server node1. Proses ini sesuai konfigurasi *Heartbeat* yaitu 5 detik. Pada peristiwa *failback* system akan bekerja seperti semula. Dan node yang dijadikan node *primary* yaitu node1 akan kembali pulih menjadi node primary dan mensinkronkan data dengan node *secondary* yaitu node2.

5. Kesimpulan

1. Dengan adanya DRBD sebagai backup data dan Heartbeat sebagai pengecek kegagalan fungsi server membuat data menjadi aman.
2. Dengan memanfaatkan teknologi cluster ketersedian informasi pada pada web server dapat terjamin.
3. Agar data antar server di dalam cluster tetap sinkron harus digunakan redudansi data yang realtime pada kedua server.
4. Server node2 hanya akan bekerja jika server utama node1 mengalami kegagalan
5. Client tidak akan mengalami gangguan jika ada server yang sedang tidak berfungsi

6. Daftar Pustaka

[1] Adityo Prabowo, Kodrat Iman Satoto, Maman Soemantri, 2012. "Perancangan MySQL *Cluster* Untuk Mengatasi Kegagalan Sistem Basis Data Pada Sisi *Server",* Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro.

[2] Indrajani, Johan, 2010. "Analisis Dan Perancangan Sistem *High Availability* Pada PT. A", Bina Nusantara University, Jakarta.

[3] Muhammad Taufik Saenal, Solikin, Setia Juli Irzal Ismail, 2013. "Membangun *High Availability Cluster* pada Web *Server* Dengan Sistem Operasl Linux Ubuntu *Server* Menggunakan *Heartbeat"*, Departemen Teknik Komputer Politeknik Telkom, Bandung.

[4] Muhammad Rais , Muhammad Tola, Armin Lawi, 2013. *"High Data Availability System* Dengan *Distributed Replicated Database System"*, Fakultas Teknik Informatika Program Studi, Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Handayani, Makassar.

[5] William, 2010. "Perancangan dan Implementasi Sistem *High Availability* dengan *Virtualisasi"*, Program Studi Teknik Elektro Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

[6] Nugroho, Prasetyo."Analisis Pemenuhan Carrier Grade Linux Standard 4.0 Aspek Cluster Pada Fedora 7". Proyek Akhir, Institut Teknologi Bandung, Bandung, 2009.

[7] Periyadi, dkk. "Sistem Tersebar" Bandung: Politeknik Telkom, 2009.

[8] Rivai,Muhammad(2010)."Panduan Linux HA & Failover pada openSUSE/SLES". Tersedia:http://www.vavai.com/wp-content/uploads/panduan-high-availability-server-menggunakan-opensuse- sles.pdf [mei. 20, 2015].