

IMPLEMENTASI *HIGH AVAILABILITY SERVER* DENGAN TEKNIK *FAILOVER COMPUTER CLUSTER* PADA UPTSIM UNIVERSITAS BINA DARMA

Muhamad Syamsuhadi ¹, Vivi Sahfitri.S.Kom., MM²., Suryayusra.M.Kom³

¹) Mahasiswa Informatika Universitas Bina Darma

²) Dosen Ilmu Komputer ³) Dosen Ilmu Komputer. Jl Jend A. Yani No.12 Plaju,
Palembang 30264

Email: muhamadsyamsuhadi@gmail.com¹) vivisahfitri@binadarma.ac.id²),
suryayusra@binadarma.ac.id³)

Abstrak.Seiringnya waktu berjalan teknologi semakin canggih dan *server* pun semakin banyak di pakai oleh perusahaan-perusahaan untuk membangun *server clustering*. *Server cluster* merupakan teknologi yang keberadaannya sangat bermanfaat dan menggabungkan beberapa sumber daya yang bekerja bersama-sama sehingga tampak seperti satu sistem yang saling terintegrasi. *Cluster* memiliki kelebihan yang menghasilkan tingkat realibilitas tinggi dengan teknik *cluster* yang memiliki kebutuhan *availability* yang tinggi atau bisa disebut juga *high availability server* dan dengan menyediakan sistem dengan *availability* yang tinggi yaitu *failover cluster*. maka itu di butuhkan dua *server* yang tergabung dalam *clustering*. *Failover clustering* bertujuan untuk membantu menjawab *high availability server* untuk menjaga *akses client* ke sumber daya *server* yang ada. dalam penelitian ini akan dilakukan pengujian terhadap *virtualisasi server* dengan konsep *clustering server* dan melakukan pengujian *High Availability*. Metode yang akan digunakan adalah dengan menggunakan metode eksperimen

1. Pendahuluan

1.1 Latar Belakang

Keberadaan teknologi *cluster computer* merupakan salah satu teknologi yang keberadaannya sangat bermanfaat bagi beberapa perusahaan yang dapat melayani *request* dari *client* setiap saat sehingga tampak seperti satu sistem yang saling terintegrasi. Umumnya teknologi *cluster* memiliki kelebihan yang dapat menghasilkan suatu sistem dengan tingkat realibilitas tinggi sehingga sistem digunakan untuk menjawab kebutuhan yang memiliki tingkat *availability* tinggi atau bisa di sebut *high availability server* sehingga teknologi ini mulai banyak dikembangkan. Mengingat fungsi *server* adalah melayani *client* maka *server* di tuntut harus semaksimal mungkin tidak mengalami gangguan yang diberikan kepada *client*.

Server sendiri sering kali mengalami gangguan layanan dan kerusakan sehingga memerlukan adanya *server backup* yang menggantikan fungsi *server* utama

maka dibutuhkan dua *server* atau yang digabungkan ke dalam *cluster*. Teknologi ini digunakan sebagai antisipasi apabila ada kegagalan atau kerusakan dalam *devices* pada komputer *server* yang dapat mengganggu kinerja pada jaringan *server*. Gangguan yang muncul pada sistem jaringan disebabkan *server* utama yang aktif sedang mati dan tidak ada *server backup* yang menggantikan fungsi *server* utama yang mati sehingga komunikasi antar jaringan terganggu. Salah satu solusi utama mengatasi masalah tersebut adalah dengan menggunakan *failover clustering server*. Fungsi utama *failover clustering* bertujuan untuk dapat menjawab kebutuhan akan adanya *high availability server* dan untuk membantu menjaga akses *client* ke sumber daya *server*. Konsep ini merupakan salah satu solusi terbaik untuk diterapkan apabila adanya kegagalan atau kerusakan yang sering sekali di dalam *server*. Sistem *Fail Over* merupakan salah satu cara untuk mengatasi keadaan yang memaksa, apabila terjadi suatu kejadian yang mengakibatkan sistem utama *down*.

DRBD merupakan suatu aplikasi *replikasi storage block device* antar dua buah *server*, yang memungkinkan melakukan sinkronisasi dua *server*. Sedangkan *Heartbeat* adalah sebuah aplikasi yang dapat mendeteksi apabila *server* utama *down* maka *Heartbeat* akan secara otomatis mengarahkan peran *server* utama kepada *server backup*. Pada implementasi ini *server* utama dan *server backup* akan ditempatkan pada *data center* yang berbeda^[1]. adapun batasan permasalahan dalam penelitian ini sebagai berikut :

1. Menghasilkan rancangan jaringan *failover computer cluster* di sistem operasi linux ubuntu.
2. Penelitian dilakukan dengan menggunakan dua laptop dengan sistem operasi linux ubuntu 14.04.
3. Peneliti menggunakan DRBD (*Distributed Replicated Block Device*)
4. Implementasi dan pengujian aplikasi hanya sebatas pada pembuktian bahwa aplikasi dapat berjalan dengan baik pada sistem yang dibangun.

2. Metode

2.1 Metode

Dalam penelitian ini penulis menggunakan metode Eksperimental, menurut Wiersma (dalam Emzir, 2008: 60), mendefinisikan eksperimen sebagai suatu situasi penelitian yang sekurang-kurangnya satu variabel bebas, yang disebut sebagai variabel eksperimental, sengaja dimanipulasi oleh peneliti. Menurut Davis (dalam Emzir, 2008: 60), penelitian ekperimental didasarkan pada asumsi bahwa dunia berkerja menurut hukum-hukum kausal. Hukum-hukum ini esensinya adalah linear, meskipun bersifat komplikasi dan interaktif. Tujuan penelitian eksperimental adalah untuk menetapkan hukum sebab-akibat dengan mengisolasi variabel kausal. Pandangan yang lebih ringan tentang asumsi-asumsi filosofi di belakang penelitian eksperimental adalah bahwa “kadang-kadang” dan “dalam cara yang sama”, dunia bekerja menurut hukum-hukum kausal. Dengan demikian, hubungan sebab-akibat mungkin bukan pandangan lain dari realita, tetapi penunjukan sebab dan akibat bermanfaat dalam keadaan/kondisi yang sama (Davis, 2004: 3).

2.2 Metode Pengumpulan Data

- a) Pengamatan (Observasi)
penulis mengumpulkan data-data dengan langsung melakukan test atau eksperimen tentang objek yang ditinjau.
- b) Pengujian (*Testing*)
Penulis melakukan pengujian dimulai dari implementasi sampai hasil untuk mendapatkan data yang ingin ditinjau.
- c) Study pustaka
Penulis mendapatkan Informasi dari Buku, kuliah, dan *Internet* yang berhubungan dengan objek
- d) Wawancara
Penulis mendapatkan Informasi dari Buku, kuliah, dan *Internet* yang berhubungan dengan objek.

3 Tahap Perancangan

3.1 Tahap Cluster

Pada tahap ini akan di lakukan proses pembuatan *cluster* yang di mana nanti sebagai tempat dari *node server master* dan *node server slave*. Setiap *server* harus sudah terdaftar sebagai anggota *cluster* maka secara otomatis kedua *server* sudah pada satu *clustering*.

```
root@master:~# yum create-clusterhd
Restarting yum cluster [11mystm; yum-clusterhd] notice: wrote new cluster
config '/etc/cluster/cluster.conf'

Starting cluster:
  Checking if cluster has been disabled at boot... [ OK ]
  Checking Network Manager... [ OK ]
  Global setup... [ OK ]
  Loading kernel modules... [ OK ]
  Reading configs... [ OK ]
  Starting cman... [ OK ]
  Waiting for quorum... [ OK ]
  Starting fenced... [ OK ]
  Starting dlm_control... [ OK ]
  Tuning DLM kernel config... [ OK ]
  Initializing self... [ OK ]
root@master:~#
```

Gambar 1. Tahap Cluster

3.2 Penginstallan *Distributed Replicated Block Device (DRBD)*

Pada penelitian ini akan menggunakan *Distributed Replicated Block Device (DRBD)* dalam proses penginstallan

```
root@master:~# apt-get install drbd-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  smartmontools
The following NEW packages will be installed:
  drbd-utils
0 upgraded, 1 newly installed, 0 to remove and 20 not upgraded.
Need to get 258 kB of archives.
After this operation, 630 kB of additional disk space will be used.
Get:1 http://ftp.debian.org/debian wheezy/main drbd-utils amd64 2:8.3.13-2 [258 kB]
Fetched 258 kB in 3s (83.4 kB/s)
Selecting previously unselected package drbd-utils.
(Reading database ... 29152 files and directories currently installed.)
Unpacking drbd-utils (from .../drbd-utils_2:8.3.13-2_amd64.deb) ...
Processing triggers for man-db ...
Setting up drbd-utils (2:8.3.13-2) ...
root@master:~#
```

Gambar 4. Penginstallan DRBD pada server master dan slave

4 Hasil dan Pembahasan

4.1 hasil dari *Instalasi Distributed Replicated Block Device*

Hasil dari instalasi *Distributed Replicated Block Device* pada kedua server drbd akan melakukan replikasi block device melalui jaringan secara real time pada kedua server yang menghubungkan server secara bebas dengan kabel cross. Sehingga lebih menjamin *Replicated* data antar server berjalan lebih stabil

```
root@master:~# service drbd status
drbd driver loaded OK; device status:
version: 8.3.13 (api:88/proto:86-96)
GIT-hash: 83ca112086600faacab2f157bc5a9324f7bd7f77 build by root@sighted, 2012-10-09 12:47:51
```

Gambar 5. *Distributed Replicated Block Device* Server Master dan slave

4.2 Konfigurasi High Availability

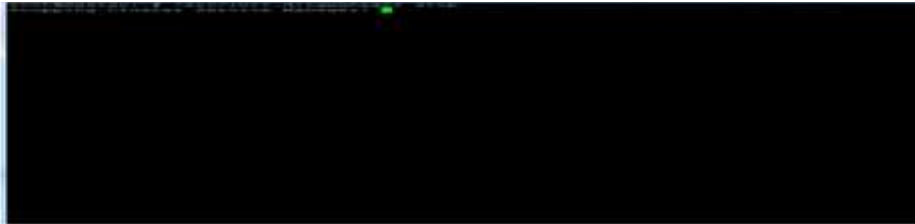
High Availability akan berfungsi jika suatu saat salah satu mesin server di ubuntu down, maka virtual machine yang ada didalamnya secara otomatis akan berpindah ke mesin server virtual yang lain. Dalam kasus ini jika “server master” down akan berpindah ke server slave.



Gambar 6. Konfigurasi High Availability

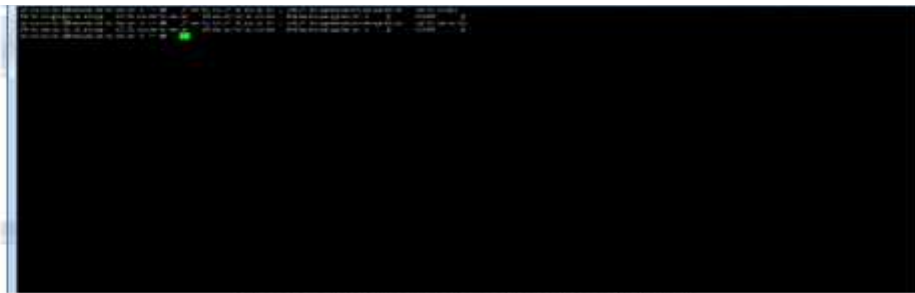
4.3 Uji Coba dan hasil *High Availability*

Untuk membuktikan dari hasil *high availability* pada tahap ini akan dilakukan uji coba *High Availability*. Dimana *Service* yang berada pada *Server Master* akan kita hentikan dan *server slave* berfungsi sebagai *backup*.



Gambar 7. Stop pada *Server Master*

Disaat *service server master* dihentikan secara otomatis *virtual machine* yang berada pada *linux ubuntu* akan berpindah ke *server slave*.



Gambar 8. Menjalankan *Service master*

Pada saat *service master* dijalankan kembali pada *server master*, secara otomatis *virtual machine* akan kembali berjalan pada *server master*

4.4 Pembahasan

Dari apa yang di dapatkan dalam penelitian ini penulis akan menjelaskan hasil dari implementasi ini secara rinci. Maka dari itu penulis sudah menyimpulkan bahwa hasil pengujian ini tidak berhasil karna penelitian ini dari luar kemampuan penulis penyebab tidak berhasilnya pengujian ini adalah karna pengujian ini masih menggunakan sistem operasi yang bersifat gratis dan konfigurasi juga masih banyak terdapat *bug* dan *error* pada kedua versi dengan menggunakan sistem operasi *linux ubuntu* dengan versi 12.04 dan versi 14.04 penyebabnya kegagalan dari versi 14.04 adalah *service* pada *distributed replicated block device (DRBD)* maka munculah tulisan *failed* dan saat melakukan perintah di terminal "*drbdadm primary r0*" perintah ini gagal dilakukan karena status *harddisk=Inconsistent* dan *primary* status datanya harus *UpToDate* bukan perintah itu saja yang mengalami *error* saat melakukan konfigurasi *drbd* pada terminal dan melakukan perintah "*service drbd status*" maka

keluarlah tulisan *drbd driver failed* dan seharusnya saat kita melakukan perintah “*service drbd status*” maka keluar tulisan “*drbd driver loaded ok*”. Di *server slave high availability* nya tidak berjalan sama sekali ketika *server master* mati maka *server slave* tidak bisa mem *backup server master* karna banyaknya *error* yang terjadi saat penulis melakukan penelitian ini dan *server slave* suka ngehang (*lag*). Dan penyebab kegagalan di *linux ubuntu* versi 12.04 adalah saat konfigurasi di *heartbeat* dan *drbd* hampir sama kejadian nya dengan konfigurasi *drbd* versi 14.04 saat memanggil *service heartbeat* dan *drbd* maka muncul lah *failure* sama seperti di versi sebelumnya jadi *web server* pun tidak bisa berjalan dengan baik dan *high availability* nya juga pun tidak bisa di jalankan dengan baik juga. Penulis masih harus banyak belajar tentang pembuatan server dan konfigurasi-konfigurasi server masih banyak yang tidak di mengerti oleh penulis

5 Kesimpulan

Kesimpulan yang telah di dapatkan dari penelitian adalah :

1. Dengan menggunakan *virtual* maka dapat mengoptimalkan *resource*
2. Dengan Menggunakan dua versi yang berbeda peneliti tidak berhasil melakukan *implementasi* karna masih terdapat banyak *error* dan *bug* pada kedua versi tersebut.
3. Masih terdapat *bug* pada *heartbeat* dan *DRBD* sehingga *konfigurasi* belum optimal pada *versi DRBD* 8.3.1.3 dan *Ubuntu Versi* 12.04 dan 14.04 LTS
4. Dapat mempermudah apabila *staff* atau *admin* ingin melakukan *maintance* pada *server master,server slave* akan mengambil alih *server master* apabila melakukan *maintance*
5. Dengan adanya sistem *High Availability*, *server* akan bisa memperkecil kegagalan apabila ada gangguan muncul

Referensi

1. Afif, Muhammad Faruq . 2013 *Implementasi Disaster Recovery Plan Dengan Sistem Failover Menggunakan Drbd dan Heartbeat Pada Data Center Fkip UNS*. Ketingan : Universitas Sebelah Maret
2. Emzir. 2008. *Metodologi Penelitian Pendidikan: Kuantitatif & Kualitatif*. Jakarta: PT RajaGrafindo Persada
3. Irfani. 2015. *Implementasi High Availability Server Dengan Teknik Failover Virtual Computer Cluster*. Surakarta : Universitas Muhammadiyah Surakarta.