

BAB IV

PERANCANGAN

Pada bab ini menjelaskan tentang perancangan sistem *High Availability* menggunakan DRBD (*Distributed Replicate Block Device*) dan *Heartbeat* pada server CentOS. Perancangan ini dilakukan untuk mengetahui apabila dua buah server digabungkan menjadi satu untuk bekerja secara bergantian membentuk suatu layanan data yang lebih optimal mulai dari pembuatan server, instalasi dan konfigurasi DRBD (*Distributed Replicate Block Device*) dan *Heartbeat* dan bagaimana komunikasi yang terjadi pada jaringan tersebut.

Perancangan sistem *High Availability* ini dibagi menjadi tiga tahap, yaitu : analisis kebutuhan, perancangan *hardware* dan perancangan *software*. Analisis kebutuhan dilakukan untuk mengetahui segala kebutuhan yang sudah ada dan belum ada pada jaringan Teknik Informatika Universitas Brawijaya. Untuk perancangan *hardware* terdiri atas perancangan server, spesifikasi server, dan fungsi dari server yang dijalankan pada lingkungan Teknik Informatika Universitas Brawijaya. Sedangkan untuk perancangan *software* terdiri instalasi OS, DRBD (*Distributed Replicate Block Device*), dan *Heartbeat*, dan konfigurasi server yang berisikan komponen *High Availability*.

Dalam praktik dilapangan *IP Address* yang dimiliki oleh Teknik Informatika Universitas Brawijaya adalah 172.20.15.xxx untuk gedung kuliah perpustakaan dan 172.20.19.xxx untuk gedung kuliah poltek. Dalam topologi jaringan di Teknik Informatika Universitas Brawijaya letak server baru ada pada gedung kuliah perpustakaan, oleh karena itu *IP Address* yang digunakan adalah 172.20.15.xxx dan akan diambil 3 buah IP statis untuk penerapan sistem *High Availability* ini.

4.1. Analisis kebutuhan

Proses analisis kebutuhan server *High Availability* menggunakan DRBD (*Distributed Replicate Block Device*) dan *Heartbeat* meliputi lingkungan perancangan dan implementasi sistem dan analisis topologi jaringan pada Teknik Informatika Universitas Brawijaya.

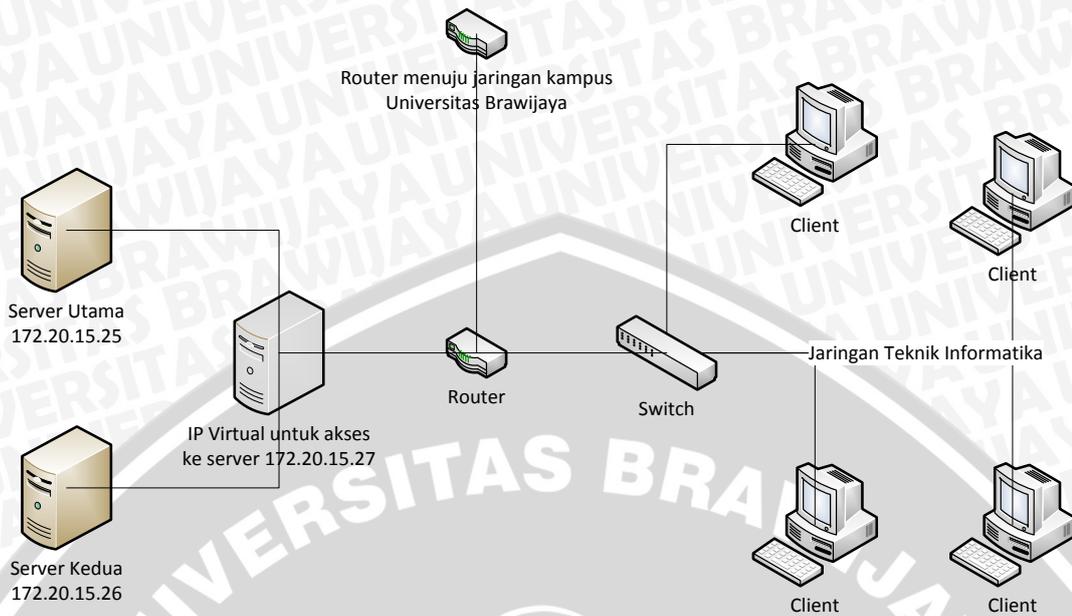
4.1.1. Lingkungan Perancangan dan Implementasi Sistem

Lingkungan Perancangan dan Implementasi Server *High Availability* ini adalah sebagai berikut :

- a. Terdiri dari dua server yang berada pada gedung Teknik Informatika Universitas Brawijaya.
- b. Pada jaringan yang digunakan sebagai IP dari kedua server adalah 172.20.15.25 dan 172.20.15.26 dan satu IP virtual yaitu 172.20.15.27 yang digunakan sebagai jembatan penghubung antara server dan user.
- c. Pada masing-masing server menggunakan Sistem Operasi Centos 5.
- d. Setiap user yang mengakses server hanya mengakses IP virtual dari server, bukan IP server yang sebenarnya.
- e. Lingkungan jaringan yang dibuat dianggap dalam kondisi yang *default* (dianggap stabil dalam koneksi antar jaringan) dan tanpa konfigurasi khusus di sisi router.

4.1.2. Analisis Topologi Jaringan yang Digunakan

Topologi jaringan komputer yang digunakan dalam jaringan server High Availability ini adalah seperti yang digambarkan pada gambar 4.1 pada halaman berikutnya, dimana server yang dibuat dilewatkan pada sebuah router yang menuju jaringan pada Teknik Informatika Universitas Brawijaya dan dilanjutkan menuju jaringan kampus Universitas Brawijaya.



Gambar 4.1 Topologi Jaringan Server *High Availability*

4.2. Perancangan Hardware

Hardware yang dibutuhkan jelas lebih dari satu buah komputer untuk merancang topologi ini. Seperti yang sudah dijelaskan pada bahasan sebelumnya dan berdasarkan pada topologi yang telah dibuat, maka dapat disimpulkan rincian hardware yang digunakan dalam perancangan jaringan *High Availability* ini. Pada tabel 4.1 di bawah ini dijelaskan tentang perangkat yang digunakan, fungsi dari perangkat-perangkat tersebut, dan alokasi IP yang diberikan pada masing-masing perangkat agar dapat saling berhubungan.

Tabel 4.1 Tabel Hardware Jaringan *High Availability*

Perangkat	Keterangan	Alokasi IP
PC pada lantai 5	Sebagai simulasi server	172.20.15.25
PC pada lantai 5	Sebagai simulasi server	172.20.15.26
PC pada lantai 5	Sebagai client	172.20.15.30

Sistem *hardware* yang dibutuhkan dalam membangun sistem *High Availability* ini minimal sehingga dapat berjalan dengan sempurna adalah sebagai berikut :

Tabel 4.2 Spesifikasi *Hardware Primary Server*

No.	Deskripsi	Spesifikasi
1	CPU	Intel® Core™2 Duo CPU T6600 @ 2.20GHz / di atasnya
2	RAM	2 GB / di atasnya
3	<i>Platform</i>	Linux 32bit
4	Sistem Koneksi	LAN 100Mbps / di atasnya
5	Penyimpanan Data	Minimal 8GB isi <i>harddisk</i> dan harus identik antara kedua server

Tabel 4.3 Spesifikasi *Hardware Standby Server*

No.	Deskripsi	Spesifikasi
1	CPU	Intel® Pentium® 4 CPU @ 2.40GHz / di atasnya
2	RAM	2 GB / di atasnya
3	<i>Platform</i>	Linux 32bit
4	Sistem Koneksi	LAN 100Mbps / di atasnya
5	Penyimpanan Data	Minimal 8GB isi <i>harddisk</i> dan harus identik antara kedua server

Pada tabel 4.2 adalah spesifikasi untuk *primary server*, dimana kebutuhan minimum dalam perancangan sistem server tersebut adalah menggunakan prosesor Intel® Core™2 Duo CPU T6600 @ 2.20GHz / di atasnya, dengan jumlah memori RAM minimal 2GB, dan koneksi data pada jaringan LAN menggunakan kecepatan minimal 100Mbps. Sedangkan pada tabel 4.3 adalah spesifikasi untuk *secondary server*, dimana kebutuhan minimum dalam perancangan sistem server tersebut adalah menggunakan prosesor Intel® Pentium® 4 CPU @ 2.40GHz / di atasnya, dengan jumlah memori RAM minimal 2GB, dan koneksi data pada jaringan LAN menggunakan kecepatan minimal 100Mbps. Dan untuk kapasitas penyimpanan yang dibutuhkan minimal adalah 8GB untuk instalasi sistem operasi dan beberapa data yang akan direplikasi nantinya.

Data di atas hanya sebagai data kebutuhan minimum, misalkan ingin menggunakan keseluruhan sistem yang sama persis (identik) dengan spesifikasi minimal adalah spesifikasi di atas, akan lebih baik dalam performa yang dijalankan

kedua server tersebut. Sedangkan untuk media penyimpanan data, harus menggunakan *harddisk* dan format partisi yang sama (identik), karena nantinya data akan direplikasi pada kedua *harddisk* yang digunakan.

4.3. Perancangan Software

4.3.1. Instalasi CentOS pada Kedua PC Server

Sistem operasi yang digunakan pada percobaan jaringan *High Availability* ini adalah CentOS 5.4, karena sistem operasi ini memang sudah dipakai dalam server pada jaringan Teknik Informatika Universitas Brawijaya. Sehingga nantinya tidak perlu terlalu merubah sistem yang sudah ada.

Dalam proses perancangannya sistem operasi ini membutuhkan beberapa *service* bawaan dari CentOS yang harus dijalankan pada saat startup sistem operasi tersebut. Misalnya *httpd*, *mysqld*, *php*, dan beberapa *service networking*.

Untuk memperoleh kemampuan sistem yang maksimal, perlu diperhatikan juga penggunaan *harddisk* yang digunakan pada percobaan ini. Karena inti dari sistem yang akan dibuat adalah replikasi data, dimana *harddisk* adalah sumber daya utama dalam pengoperasiannya. Hal yang perlu diperhatikan dalam penggunaan *harddisk* untuk sistem ini adalah kecepatan baca/tulis yang sama, jumlah dan ukuran partisi yang sama, dan jenis *harddisk* yang sama. Jadi untuk pemilihan *harddisk* dalam sistem ini sangat direkomendasikan untuk menggunakan *harddisk* yang identik atau satu pabrikan dengan jenis yang sama.

4.3.2. Instalasi DRBD dan Heartbeat pada Kedua PC Server

DRBD dan *Heartbeat* ini dapat diinstal setelah semua proses instalasi sistem operasi berhasil dilakukan, yaitu dengan cara menggunakan perintah instalasi standar *yum* pada terminal di CentOS. Setelah itu dilanjutkan dengan konfigurasi DRBD dan *Heartbeat* tersebut.

Instalasi DRBD dapat dilakukan tanpa merubah *repository* bawaan dari CentOS, dalam paket DRBD ini akan diambil dua buah aplikasi yaitu *drbd* dan *drbd-*

kmod. Sedangkan pada instalasi *heartbeat* diambil paket utama saja, yaitu *heartbeat*. Selain kedua komponen utama di atas, masih dibutuhkan beberapa tambahan aplikasi untuk menjalankan sistem *High Availability* ini secara optimal, seperti *mysqld*, *httpd*, *php*, dan sebagainya. Semua aplikasi di atas sudah ada dalam paket instalasi CentOS 5.4 hanya saja perlu dilakukan *update* dari aplikasi-aplikasi tersebut.

Aplikasi *mysqld* digunakan untuk menyimpan database dari data-data yang akan disimpan kedalam server, dimana *mysqld* ini nantinya akan diletakkan pada *replication disc*, jadi akan dilakukan perpindahan dari tempat aplikasi *default*-nya. Sedangkan aplikasi *httpd* digunakan untuk menyimpan data dari *web* atau *localhost* yang diakses melalui port 80 atau dari *browser*. File ini bisa saja dipindah ke *replication disc*, namun untuk baiknya tetap diletakkan pada *default*-nya karena dengan begitu bisa dipantau server mana yang sedang mengakses data pada *replication disc*. Ini sangat berguna bagi server yang letaknya saling berjauhan atau *administrator* yang jauh dari server, sehingga akan dapat dilakukan penanganan masalah dengan cepat. Untuk aplikasi *php* nantinya akan digunakan untuk pengembangan selanjutnya, yaitu untuk menjalankan sebuah *web file* yang mengandung data dari *database* yang telah dibuat dalam *mysqld*.

Dalam konfigurasi DRBD dan Heartbeat, diperlukan sebuah aturan (*rules*) sehingga kinerja dari sistem menjadi optimal dan sesuai harapan. Dalam penerapan aturan tersebut konfigurasi sistem menjadi hal yang paling krusial, karena dalam konfigurasi sistem ini terdapat perintah-perintah yang bisa digunakan untuk mendeklarasikan aturan yang akan diterapkan. Langkah pertama adalah menentukan *IP Address* yang akan digunakan, dalam penelitian ini digunakan *IP Address* dari Teknik Informatika Universitas Brawijaya yaitu 172.20.15.xxx untuk gedung kuliah perpustakaan, dan 172.20.19.xxx untuk gedung kuliah poltek. Karena server Teknik Informatika Universitas Brawijaya terletak pada gedung perpustakaan, maka *IP Address* yang akan digunakan adalah 172.20.15.xxx, dengan netmask 255.255.255.0 atau 172.20.15.xxx/24. Dalam percobaan ini server utama dan server kedua akan menggunakan *IP Address* 172.20.15.25 dan 172.20.15.26 dan untuk Virtual *IP Address* akan digunakan 172.20.15.27 sebagai penghubung antara *client* dan server.

DRBD mempunyai aturan tersendiri dalam melakukan replikasi datanya, yaitu dengan aturan protokol, ada tiga jenis protokol dalam DRBD [AN4-11], yaitu :

- **Protokol A** : Protokol replikasi *asynchronous* yang menganggap penulisan data replikasi pada server utama telah selesai dilakukan setelah penulisan dalam media penyimpanan server utama telah selesai ditulis, dan paket replikasi data telah ditaruh pada *TCP send buffer*. Umumnya protokol A ini dipakai untuk skenario replikasi jarak jauh (tidak berada dalam jaringan lokal).
- **Protokol B** : Protokol replikasi *semi-synchronous* yang menganggap penulisan data replikasi pada server utama telah selesai dilakukan setelah penulisan dalam media penyimpanan server utama telah selesai ditulis, dan paket replikasi data telah sampai pada server kedua.
- **Protokol C** : protokol replikasi *synchronous* yang menganggap penulisan data replikasi pada server utama telah selesai dilakukan setelah penulisan dalam media penyimpanan server utama dan penyimpanan server kedua telah diverifikasi selesai penulisan datanya.

METHOD DRBD PARAMETER *resources, protocol, startup, disk, net, syncer, on*

DECLARATION :

Resources IS string
Protocol IS string
Startup IS int
Disk IS string
Net IS int
Syncer IS byte
On IS string

DESCRIPTION :

```
1 resource drbdku { #nama drbd yang dibuat
2 protocol C; #protokol yang digunakan
3 startup { #pengaturan pada saat startup
4   degr-wfc-timeout 30; # 30 seconds }
5 disk { #konfigurasi disk
6   on-io-error detach; }
7 net { #konfigurasi saat terjadi kesalahan drbd
8   timeout 120;
9   connect-int 20;
10  ping-int 20;
```

```

11     ko-count 30;
12     cram-hmac-alg "sha1";
13     shared-secret "GILELUNDRO";
14     }
15     syncer { #konfigurasi replikasi data
16         rate 30M; # synchronization data transfer rate
17         al-extents 257;
18     }
19     on ha1 { #konfigurasi pada server utama
20         device /dev/drbd0;
21         disk /dev/sda3;
22         address 172.20.15.25:7789;
23         meta-disk internal;
24     }
25     on ha2 { #konfigurasi pada server kedua
26         device /dev/drbd0;
27         disk /dev/sda3;
28         address 172.20.15.26:7789;
29         meta-disk internal;
30     }
31 }
END DRBD

```

2Gambar 4.2 Perancangan pseudocode file konfigurasi DRBD

Konfigurasi DRBD pada gambar 4.2 di atas menggunakan aturan yang telah dibuat sehingga kinerja DRBD dapat menjadi optimal, berikut adalah keterangan yang lebih lengkap tentang data pada gambar 4.2 :

- Pembuatan nama dari DRBD yang akan digunakan, pada percobaan ini menggunakan nama “drbdku”.
- Protokol yang digunakan dalam percobaan ini adalah protokol C, dimana data yang dituliskan pada disk replikasi baru akan disinkronisasikan setelah file tersebut diverifikasi bahwa penulisannya telah selesai.
- Pada saat sistem operasi melakukan *startup*, DRBD akan menunggu pasangan replikasinya hingga *online*, waktu tunggu yang dialokasikan pada DRBD ini adalah 30 detik. Setelah lebih dari itu, maka DRBD tidak dinyalakan dan server akan berjalan seperti biasa namun dengan *service* DRBD yang tidak aktif.
- Pada saat terjadi error pada *replication disk*, maka partisi tersebut akan dilepas secara otomatis oleh sistem DRBD.

- Jumlah *data rate* yang digunakan pada sistem DRBD untuk melakukan proses sinkronisasi dengan pasangan *replication disk*-nya adalah 30MBps. Sehingga *data traffic* pada jaringan tidak terpenuhi oleh sinkronisasi data yang terjadi pada *replication disk*.
- Pada ha1, digunakan *device /dev/drbd0* dengan disk */dev/sda3* sebagai media penyimpanan datanya dan pengalokasian *IP Address* pada 172.20.15.25 dengan port 7789.
- Pada ha 2, digunakan *device /dev/drbd0* dengan disk */dev/sda3* sebagai media penyimpanan datanya dan pengalokasian *IP Address* pada 172.20.15.26 dengan port 7789.

Setelah konfigurasi di atas selesai dilakukan, maka langkah selanjutnya adalah melakukan *umount* pada */dev/sda3* karena disk tersebut akan digunakan untuk media replikasi data dalam DRBD. Kemudian pastikan juga dalam */etc/fstab* disk tersebut sudah ter-*umount*, karena apabila belum ter-*umount* disk tersebut akan tetap ter-*mount* saat sistem restart nantinya. Selanjutnya adalah membuat sebuah data semu pada *replication disk* tersebut yaitu dengan melakukan perintah “*dd*”, setelah data semu tersebut dibuat maka DRBD dapat dibuat dengan melakukan perintah “*create-md*”. Setelah DRBD yang dibuat dari hasil konfigurasi di atas tadi dibuat, maka lakukan format pada partisi */dev/drbd0* dengan menggunakan perintah *mkfs*. Setelah semua perintah di atas dipastikan sudah dilakukan, maka *service* dari DRBD sudah dapat dijalankan.

METHOD : Heartbeat ha.cf PARAMETER logfacility, node, keepalive, warntime, deadtime, initdead, udpport, bcast, baud, auto_failback

DECLARATION :

Logfacility IS string
 Node IS string
 Keepalive IS int
 Warntime IS int
 Deadtime IS int
 Initdead IS int
 Udpport IS int
 Bcast IS int
 Baud IS int

DESCRIPTION :

```

1 #Heartbeat logging configuration
2   logfacility daemon
3 #Heartbeat cluster members
4   node ha1
5   node ha2
6 #Heartbeat communication timing
7   keepalive 1
8   warntime 10
9   deadtime 30
10  initdead 120
11 #Heartbeat communication paths
12  udpport 694
13  bcast eth0
14  baud 19200
15 #Fail back automatically
16  auto_failback on
END ha.cf

```

Gambar 4.3 Perancangan *pseudocode* file konfigurasi Heartbeat ha.cf

Konfigurasi Heartbeat juga memerlukan beberapa aturan yang dapat ditetapkan sehingga dapat berjalan secara optimal. Ada tiga buah file yang harus dikonfigurasi dalam pengaturan Heartbeat ini yaitu ha.cf, haresources, dan authkeys.

Berikut ini adalah keterangan yang lebih lengkap tentang aturan yang diterapkan dari gambar 4.3 :

- *Logging* yang digunakan dalam percobaan ini menggunakan fasilitas dari *daemon* yang sudah ada dalam sistem operasi CentOS 5.4.
- *Node* yang akan ditentukan pada kedua server harus dideklarasikan pada *Heartbeat*, sehingga *Heartbeat* mengenali server mana saja yang akan dilakukan replikasi. Dalam percobaan ini *node* yang digunakan adalah *node ha1* dan *node ha2*.
- Untuk komunikasi antar *node* digunakan beberapa aturan sebagai berikut yang tiap nilainya adalah ditentukan dengan satuan detik, *keepalive* menggunakan 1 sebagai satuannya agar saat sistem terjadi eror maka *Heartbeat* hanya akan menjalankan sistem tersebut selama 1 detik yang selanjutnya berubah menjadi *warnitime*. Nilai ini dirasa sangat efektif, karena dapat memangkas waktu tunggu saat terjadi *failover*.
- *Warnitime* menggunakan 10 sebagai satuannya, setelah lebih dari 10 detik maka sistem akan dialihkan kekeadaan mati.
- *Deadtime* menggunakan 30 sebagai satuannya, setelah lebih dari 30 detik maka sistem akan dinyatakan mati.
- Pada *Heartbeat* ini menggunakan *port UDP 694* sebagai jalur komunikasi antar server, dan pengecekan ketiap-tiap server dilakukan secara *broadcast* pada satu jaringan *IP Address* yang telah dialokasikan.
- Fitur *auto_failback* pada percobaan ini diaktifkan karena untuk memudahkan proses *failback* saat server utama telah dalam kondisi aktif kembali.

METHOD : Heartbeat haresources PARAMETER node, IPaddr, drbddisk, filesystem, services, error report

DECLARATION :

Node IS string
 IPaddr IS int
 Drbddisk IS string
 Filesystem IS string
 Services IS string
 Error report IS string

DESCRIPTION :

```

1  #node server utama
2  ha1
3  #IPaddr
4  172.20.15.27/24
5  #drbddisk
6  drbdku
7  #Filesystem
8  device /dev/drbd0 pada disk /repdata
9  #Services
10 Httpd
11 Mysqld
12 #Error Report
13 MailTo yasa\_46@live.com DRBDFAILURE
END haresources

```

4Gambar 4.4 Perancangan *pseudocode* file konfigurasi Heartbeat haresources

Gambar 4.4 menjelaskan aturan yang dibuat saat terjadi proses *failover* dan *failback*, berikut adalah penjelasan yang lebih detail dari gambar 4.4 di atas :

- *Node* ha1 dideklarasikan sebagai server utama dalam sistem *High Availability* yang akan dibangun.
- Pengalokasian *IP Address* 172.20.15.27/24 sebagai *IP Virtual* yang digunakan sebagai jalur akses antara *client* dan server.
- *Services* httpd dan mysqld dimasukkan kedalam konfigurasi ini karena kedua *services* tersebut nantinya akan digunakan pada komunikasi antara *client* dan server.
- *Drbddisk* yang digunakan adalah drbd yang akan dibuat yaitu **drbdku**. *Filesystem* yang digunakan adalah /dev/drbd0 yang akan di-mount pada file “/repdata”.

- Jika terjadi kegagalan dalam sistem (*failover*) maka akan dikirimkan sebuah e-mail peringatan yang ditujukan pada e-mail yang telah dideklarasikan pada konfigurasi yang akan dibuat nantinya.

Setelah melakukan konfigurasi di atas, langkah selanjutnya adalah memberikan *soft-link* untuk *mysqld* dan *httpd* sehingga kedua *services* tersebut dapat berpindah menuju ke server kedua saat terjadi proses *failover*. Setelah itu adalah membuat sebuah *authkeys*, dimana fungsi dari file tersebut adalah untuk memberikan sebuah *security* pada server yang dibuat. Setelah file tersebut dibuat, maka hak akses untuk file tersebut juga harus dirubah menggunakan perintah "*chmod*" dimana nilai yang diberikan pada file tersebut adalah 600. Setelah semua perintah di atas dipastikan sudah dilakukan, maka *service* dari Heartbeat sudah dapat dijalankan.

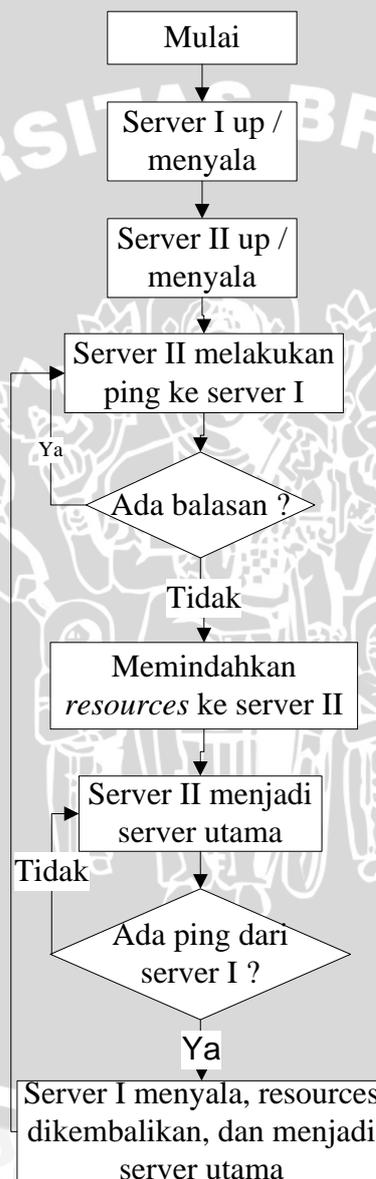
Selanjutnya adalah konfigurasi dari *service* *httpd* dan *mysqld* yang akan digunakan sebagai penunjang performa dari sistem yang akan dibangun. Pada instalasi CentOS sudah dibawakan *service* berupa *mysqld* dan *httpd*, namun versi yang dibawakan adalah versi yang lama, sehingga dalam percobaan ini file-file tersebut akan di-*update* menjadi versi terbaru.

File pada *httpd* yang perlu dilakukan konfigurasi adalah **httpd.conf** yang terletak pada "*/etc/httpd/conf/httpd.conf*", yaitu penggunaan *listening IP Address* dan *port* yang akan digunakan pada percobaan ini. Mengganti baris perintah *DocumentRoot* menjadi "*/var/www/html*", dan baris perintah *<Directory* "*/var/www/html*".

File pada *mysqld* yang perlu dilakukan konfigurasi adalah **my.cnf** yang terletak pada "*/etc/my.cnf*", yaitu penggunaan direktori data dan soket yang akan digunakan dalam percobaan ini serta konfigurasi untuk *log-error*. Direktori data diubah menjadi "*/repdata/mysql*", lalu soket dirubah menjadi "*/var/lib/mysql/mysql.sock*", dan *log-error* dirubah menjadi "*/repdata/mysql/mysqld.log*".

Setelah konfigurasi kedua *services* di atas selesai dilakukan, maka langkah selanjutnya adalah memberikan *soft-link* untuk kedua *services* tersebut. Hal itu dilakukan dengan melakukan perintah “*ln -sf*”, perintah tersebut diikuti dengan letak file *services* yang akan diberikan *soft-link*.

Jadi algoritma dari proses High Availability di atas dapat dijabarkan pada gambar 4.5 berikut ini.



Gambar 4.5 Flowchart algoritma proses High Availability