

HIGH AVAILABILITY WEB SERVER BERBASIS OPEN SOURCE DENGAN TEKNIK FAILOVER CLUSTERING

I Gede Putu Krisna Juliharta¹⁾, Wayan Supedana²⁾, Dandy Pramana Hostiadi³⁾

^{1,2,3)} STMIK STIKOM Bali

*Jalan Raya Puputan Renon no. 86 Denpasar, Bali, Indonesia tlp. (0361) 244445 fax: (0361) 264773
E-mail:krisna@stikom-bali.ac.id¹⁾, suvedana88@gmail.com²⁾, dandy@stikom-bali.ac.id³⁾*

Abstrak

Dalam perkembangan teknologi jaman sekarang sudah semakin berkembang dengan tersedianya website yang digunakan untuk mengakses informasi dengan sangat mudah. website memerlukan web server agar dapat berjalan, jika web server dari website mengalami kegagalan seperti kerusakan pada hard disk atau yang lainnya tentunya web server tidak akan dapat di akses oleh konsumen yang ingin mendapatkan informasi yang ada. menghadapi masalah tersebut di atas, dibuatlah penelitian 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 yang di implementasikan dengan tujuan utama untuk meningkatkan availability layanan. Hasil dari penelitian ini adalah kemampuan dari system untuk tetap memberikan layanan walaupun system utama mengalami masalah atau lumpuh.

Kata kunci: Web Server, High Availability, Cluster

1. Pendahuluan

Perkembangan teknologi informasi saat ini sudah semakin berkembang dahulu seseorang hanya mendapatkan informasi melalui televisi dan berita dalam bentuk tulisan atau koran tapi seiring perkembangan jaman teknologi informasi saat ini yang banyak ketersediaan fasilitas akses internet maka seseorang berupaya menciptakan layanan website yang akan memudahkan seseorang untuk mencari sebuah , Dengan keadaan perangkat yang harus standby setiap hari maka seseorang tidaklah bisa untuk berbuat apa-apa jika sewaktu waktu server mengalami masalah dengan server tanpa down, maka kami akan mengimplementasikan suatu perangkat clustering server yang akan mengantisipasi jika server pusat mengalami masalah atau down maka server clustering yang langsung mengambil alih pusat data dan memberikan kesempatan untuk memperbaiki server pusat

Clustering server atau failover clustering adalah menggunakan lebih dari satu server yang menyediakan redundant interconnections, sehingga user hanya

mengetahui ada satu sistem server yang tersedia dan client tidak menyadari jika terjadi kegagalan pada sistem server karena tersedianya server sebagai redundant atau backup[1]. failover clustering yang akan di implementasikan adalah failover yang bersifat aktif-pasif yang artinya dua server yang bekerja bersama dan server secondary menjadi server cadangan jika server primary mati, untuk melaksanakan penelitian ini metode penelitian yang digunakan adalah metode waterfall, dengan tahapannya berupa analisa kebutuhan, desain sistem dan perancangan sistem, implementasi, dan pengujian.

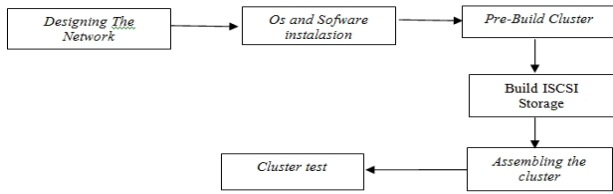
Penelitian yang dilakukan menggunakan referensi dari penelitian sebelumnya, yang sudah dilakukan oleh 2 orang yaitu Muhammad Taufik saenal dari universitas Pasundan Bandung berjudul Implementasi High Availability Cluster Dalam Penerapan Virtual Server Menggunakan Proxmox Virtual enviroment. Dari judul tersebut dan topik penelitian yang diangkat memiliki perbedaan pada tipe servernya, penelitian yang dilakukan menggunakan server asli dan DRDB sedangkan oleh taufik menggunakan server virtual dan Proxmox enviroment. Penelitian yang kedua oleh fikri hidayat dari Universitas Indonesia memiliki perbedaan objek penelitiannya. Penelitian yang dilakukan sebelumnya untuk Storage Area Network dan yang dilakukan saat ini untuk Web Server.

Hasil yang diharapkan adalah sebuah sistem high availability web server yang akan memberikan suatu kenyamanan bagi pengguna atau penyedia layanan untuk mengantisipasi suatu kejadian bencana yang akan mengakibatkan suatu kegagalan system.

2. Pembahasan

2.1 Perancangan Sistem

Agar sistem yang dibangun dapat berjalan dengan baik perlu adanya perencanaan yang baik. Komponen-komponen yang akan dibangun harus tepat sehingga dalam melakukan Implementasi high availability web server menjadi tepat. Berikut adalah Gambar alur dari perencanaan sistem cluster komputer sebagai berikut



Gambar 1 Block Diagram Perancangan sistem cluster komputer

Pada gambar 1 menunjukkan tahapan dalam mendesain suatu sistem cluster komputer dimana dapat dilihat pembuatan sistem cluster komputer dimulai dengan menentukan spesifikasi komputer yang akan digunakan dimulai dari memori processing dari power sampai dengan disk space [2]

2.1.1 Designing The Network

Pada tahapan ini adalah mendisain network yang akan digunakan rancangan cluster yang dibuat adalah suatu failover cluster dimana suatu failover cluster itu membutuhkan tingkat ketersediaan yang tinggi sehingga dibutuhkan lebih dari satu buah network jadi jika satu network mengalami kondisi down maka masih ada network lain yang akan menghubungkan cluster tersebut seperti yang dijelaskan akan ada tiga network yang akan dibangun seperti berikut :

- 1 192.168.1.10 Server 1 network yang menghubungkan ke semua node
- 2 192.168.1.11 server 2 network yang menghubungkan ke semua node
- 3 192.168.1.20 IP Virtual yang akan digunakan client untuk mengakses web server dengan domain name cluster.com

2.1.2 OS and Software Instalation

Instalasi OS dan software yang di gunakan dalam implementasi ini yaitu paket software lamp-server untuk penyimpanan data web server yang digunakan, paket ucarp, paket heartbeat, DRBD, DNS server pada tahap ini software di instal dengan fitur-fitur tambahan yang dibutuhkan

2.1.3 Pre-Build Cluster

pre-build cluster yaitu ditahap ini dibuatkan nodes-nodes cluster pada tahap ini dibuatkan implementasi serta konfigurasi dari masing-masing Server untuk dijadikan nodes ,konfigurasi yang dilakukan yaitu menghubungkan server1 ,server2 kedalam sebuah network yang dapat terhubung menjadi satu.dan menentukan domain user tiap nodes yang berpengaruh nantinya terhadap perannya di cluster[3].

2.1.4 Build ISCSI Storage

Membangun storage Drbd (distributed replicated block devices) adalah tools yang berguna dalam sinkronisasi data antar server. Tools ini berfungsi untuk me-mirror block device antar komputer anggota cluster karena sistem yang dibangun adalah menggunakan mirroring data jadi partisi dari HDD haruslah sama karena mirror (cermin)

2.1.5 Assembling the cluster

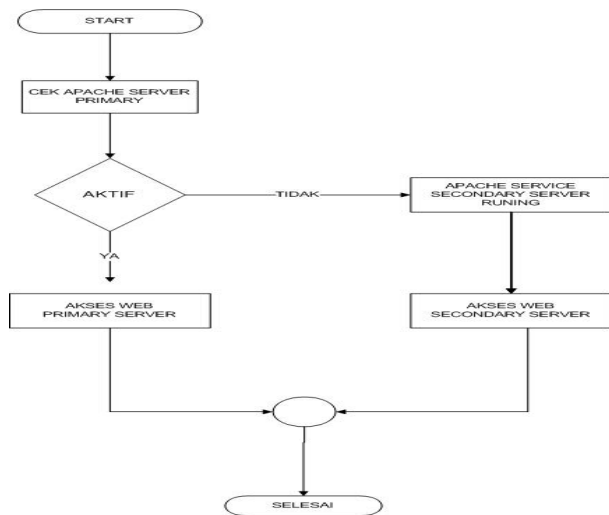
Perancangan ini adalah membangun cluster itu sendiri. tahap pembangunan cluster ini terdiri dari tahap validasi cluster yaitu tahap mengetahui apakah seluruh kebutuhan cluster sudah terpenuhi atau belum seperti nodes ,network dan storage dan menentukan nama cluster dan menentukan Ip cluster setelah berhasil dibuat tahap selanjutnya adalah instalasi service dan aplikasi dari cluster tersebut .

2.1.6 Cluster Test

Tahap terakhir dari rancangan pembuatan sisem ini yaitu cluster test cluster tes merupakan tahap dimana dilakukan test pada cluster untuk mengetahui apakah implementasi failover clustering mengalami kegagalan sistem apa mendapatkan hasil yang dicapai

2.2 Flowchart Sistem Cluster

Skema sistem yaitu dengan sistem primary diambil alih secondary system dan sistem primary mengambil alih sistem secara otomatis seperti berikut



Gambar 2. Sistem Cluster

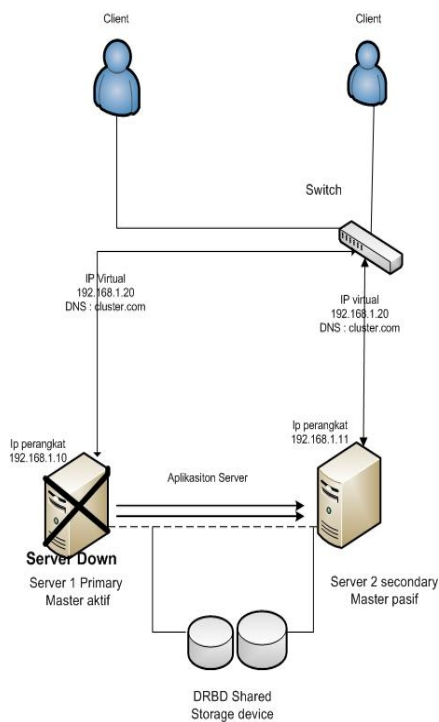
Dari Skema Flochart gambar 2 diatas dapat dijelaskan bahwa memulainya suatu sistem pada awal start primary dan secondary server dan dengan dilanjutkan dengan memproses dengan mengecek sistem apache2 dan jika sistem apache2 dalam keadaan aktif maka web server akan mengakses di primary server dan jika apache2 di primary server tidak aktif maka akan dialihkan ke

secondary server dan akses *web server* akan berjalan di *secondary server*.

2.3 Perancangan Sistem *high availability* dalam Keadaan *Down*

Dalam Perancangan skenario sistem dimana sistem mengalami masalah pada saat *client* sedang mengakses sistem pada saat bersamaan *server Primary* data mengalami masalah dan langsung dialihkan ke *secondary server* dan *client* tidak menyadari bahwa terjadi kegagalan pada sistem. Berikut akan dijelaskan simulasi sistem dimana *primary server* telah mengalami masalah dan dialihkan ke *secondary server*.

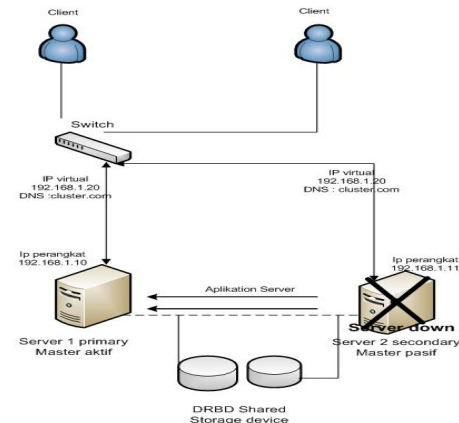
2.3.1 *Server Primary* Mengalami *down* sistem



Gambar 3. Perangkat *Primary* *down*

Seperti pada gambar 3. dimana *server1 primary* Mengalami *down* sistem pada saat *server* sedang aktif dan *client* mengakses data ke *server*, pada saat *server primary* mengalami masalah maka *ip virtual* pada *heartbeat* akan mendeteksi bahwa *server primary* dengan *IP* fisik 192.168.1.10 mengalami masalah dan *heartbeat* dengan *IP Virtual* 192.168.1.20 dengan otomatis akan mengalihkan sistem ke *IP* fisik 192.168.1.11 yang berada di *server2 secondary*, maka sistem akan tetap berjalan dengan keadaan *On* dan data yang telah *tersinkronisasi* ke *server secondary* akan berjalan sesuai data yang telah *dibackup*.

2.3.2 *Server Secondary* Mengalami *down* sistem



Gambar 4. Perangkat *Secondary* *Down*

Seperti Pada gambar 4. perangkat *secondary* mengalami masalah sistem maka *heartbeat* dengan *IP virtual* 192.168.1.20 atau domain name *cluster.com* akan mengalihkan secara otomatis sistem ke *server1 primary* dengan *ip* fisik 192.168.1.10 dan sistem akan tetap berjalan ini disebut dengan *failback* sistem.

2.4 Implementasi dan Pengujian Sistem

2.4.1 Konfigurasi DRBD

Untuk melakukan konfigurasi *DRBD* pastikan *harddisk* kedua *server* yang akan di *sinkronisasi* berukuran sama dan belum ada *file* sistem didalamnya sebelum melakukan konfigurasi, install paket *DRBD* dengan melakukan perintah `#apt-get install drbd8-utils drbdlinks` setelah selesai lakukan konfigurasi `#pico /etc/drbd.conf` dan isikan perintah berikut [4] :

- Resource r0* merupakan nama *resource* yang akan digunakan sebagai parameter untuk memanggil fungsi-fungsi *drbd*
- Protocol c* merupakan protocol yang digunakan oleh *drbd* untuk *sinkronisasi device* `/dev/drbd0` merupakan *device* yang akan digunakan ketika melakukan *mount*
- Opsi didalam `net{}` digunakan untuk menangani masalah *split brain*
- On *server1* {
`Address 192.168.1.10:7788;`
`Meta-disk internal;`
`Disk /dev/sda9;`
 Ini merupakan konfigurasi untuk tiap *nodenya*, setiap *node* disesuaikan dengan konfigurasi *server*
- `Address 192.168.1.10:7788` merupakan *IP address* yang digunakan untuk *sinkronisasi* dengan port standar *drbd*
- `Disk /dev/sda9` merupakan *partisi* yang digunakan oleh *drbd*

selanjutnya buat `drbdadm create-md r0` merupakan *create* pada *partisi sda9* untuk menentukan *drbd* tersimpan pada *partisi hardisk* jalankan *service* di semua *server* dan jalankan *server* dengan perintah `#!/service drbd start`. jadikan *server primary* sebagai *primary server* dengan mengetikkan perintah

```
# drbdsetup /dev/drbd0 primary --overwrite-data-of-peer.
```

Cek status service drbd status dan hasilnya block device akan tersinkronisasi tunggu sampai sinkronisasi berjalan hingga 100% selanjutnya

mounting kesalahan satu *directory format drive drbd* hanya dilakukan pada *server primary* atau *server1* saja dengan perintah `# mkfs.ext3 /dev/drbd0`, File sistem dibuat hanya pada aktif *server* karena pasif *server* akan otomatis menduplikasi disk pada aktif *server*. Setelah itu buat folder *dataSERVER* pada kedua *server* untuk meletakkan file-file yang akan disinkronisasi dengan perintah `#mkdir /dataSERVER`.

2.4.2 Pengujian sistem

Pada pengujian ini akan dilakukan *failover* dan *fallback* pada *ip address virtual* yang melayani *service* yang di *cluster*, sebelumnya seting IP client terlebih dahulu. Setelah itu lakukan ping ke server1 192.168.1.20 dan matikan service heartbeat pada server1 dengan perintah `#service heartbeat stop`, dan hasilnya seperti gambar

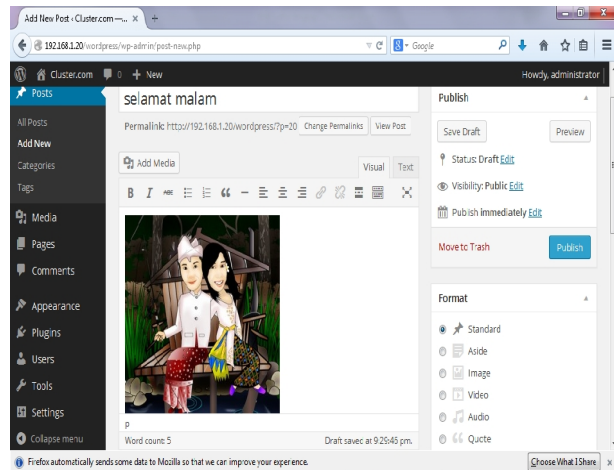
```
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Request timed out.
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
```

Gambar 5. Testing failover

Dari gambar 5. terlihat *request time out* di antara *reply* ini menandakan perpindahan pelayanan *service* dari *aktif server1* ke *pasif server2* sesuai konfigurasi *heartbeat* yaitu 5 detik

2.4.3 Pengujian Sistem web

Pada tahap test web cluster.com pengujian melakukan tes beberapa kali tes dengan mengupload data gambar file music dan file video dengan kapasitas size data berbeda beda, tahap pengujian *web website* sedang berjalan dan *service heartbeat* di *primary server1* akan di matikan dan pengujian akan mengakses *web cluster.com*, dan hasilnya seperti gambar.



Gambar 6. Test Cluster web

```
C:\Windows\System32\cmd.exe - ping 192.168.1.20 -t
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>ping 192.168.1.20 -t

Pinging 192.168.1.20 with 32 bytes of data:
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Request timed out.
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
Reply from 192.168.1.20: bytes=32 time<1ms TTL=64
```

Gambar 7. Tes Ip Virtual di client

Tes *cluster website cluster.com* masih tetap berjalan dan perpindahan data ke *secondary server* secara *realtime* walupun terjadi beberapa *RTO* pada *reply* itu menandakan perpindahan *failover* sudah berjalan (gambar 6. Dan gambar 7)

3. Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian akhir ini adalah

1. Dengan pemanfaatan *technology high availability web server* ketersediaan data pada *web server* dapat terjamin dengan menggunakan sistem *failover clustering* dengan didukung paket *heartbeat dan drbd disk*
2. Agar data antar server di dalam *cluster* tetap *sinkron* harus digunakan *redundansi* data yang *real time* pada kedua *server*.
3. *Secondary server* akan bekerja jika *primary server* mengalami masalah dengan layanan *heartbeat* dan *drbd* disk akan memindahkan data secara *realtime*
4. *Client* tidak akan mengalami gangguan jika ada *server* yang sedang tidak bekerja

Daftar Pustaka

- [1] Anjik Sukmaanji & Rianto, 2008. Jaringan Komputer, Penerbit Pustaka ANDI Yogyakarta
- [2] Designing a cluster computer. Diakses 20 Juli 2014 Dari ameslab
- [3] http://www.scl.ameslab.gov/project/parallel_computing/cluster_design.html

- [4] Hidayatno , 2008 , Perencanaan Jaringan Komputer, penerbit pustaka ,Jakarta: PT. Alex Media Komputindo
- [5] Rivai, Muhammad (2010). "Panduan Linux HA & Failover pada openSUSE/SLES". [Online]. Tersedia [uploads/panduan-high-availability-server menggunakan-opensuse-sles.pdf](https://www.opensuse.org/Uploads/panduan-high-availability-server-menggunakan-opensuse-sles.pdf) [mei. 20, 2011].

Biodata Penulis

I Gede Putu Krisna Juliharta, memperoleh gelar Sarjana Teknik (S.T), Jurusan Teknik Informatika UPN Veteran Yogyakarta, lulus tahun 2007. Memperoleh gelar Magister Teknik (M.T) Program Pasca Sarjana Magister Teknik Elektro Universitas Udayana Bali, lulus tahun 2010. Saat ini menjadi Dosen di STMIK STIKOM Bali.

I Wayan Supedana, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Komputer STMIK STIKOM Bali, lulus tahun 2014. Saat ini menjadi Staff ISP di Bali.

Dandy Pramana Hostiadi, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Komputer STMIK STIKOM Bali, lulus tahun 2010. Memperoleh gelar Magister Teknik (M.T) Program Pasca Sarjana Magister Teknik Elektro Universitas Udayana Bali, lulus tahun 2014. Saat ini menjadi Dosen di STMIK STIKOM Bali.

