

Clustering Server Pada Cloud Computing Berbasis Proxmox VE Menggunakan Metode High Availability

Linda Apriliana¹, Ucuk Darusala², Novi Dian Nathasia³

¹lindaaprilianastyo@gmail.com, ²ucuk.darusalam@gmail.com, ³ivanovic77@yahoo.com

^{1,2,3}Teknik Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional

Abstract— Cloud Computing technology Service and data stored on machine, the server this makes important factor engine server as a supporting service availability. More and more users are accessing the service will result in engine performance server load becomes heavier and less than optimal, because the service must work continuously to provide data that can be accessed anytime by the user via the network connected. Server hardware has a performance period. Likewise with software that may crash. With functionality server which provides services to the client, the server is required to have level of availability. A high this enables the engine server to experience downtime. Server machine must also be turned off for maintenance purposes. The purpose of this study is build Clustering Server to work together as a single system which over the virtual environment. This is a solution to overcome these problems. In this study researcher use a server virtualization proxmox, FreeNAS is a NAS server and DRBD to support high availability of services in the sphere of HA, data synchronization in a High Availability to perform mirroring another system machine. With the implementation of the method of HA and synchronization DRBD and NFS (Network File System) on a system cluster obtained results average migration time on node1 towards node2 reach 9.7(s), 3.7(s) for node2 towards node3, and 3(s) on node3 towards node1. Also obtained downtime less amounting 0.58 ms on node1, on node2 0.02 ms, and 0.02 ms on node3.

Keywords- cluster, High Availability, proxmox VE, server, virtualization.

Intisari— Layanan dan data teknologi Cloud Computing tersimpan pada server, hal ini menjadikan faktor pentingnya server sebagai pendukung ketersediaan layanan. Semakin banyak pengguna yang mengakses layanan tersebut akan mengakibatkan beban kinerja mesin server menjadi lebih berat dan kurang optimal, karena layanan harus bekerja menyediakan data terus-menerus yang dapat diakses kapanpun oleh penggunanya melalui jaringan terkoneksi. Perangkat keras server memiliki masa performa kinerja. Hal serupa dengan perangkat lunak yang dapat mengalami crash. Dengan fungsi server yang memberikan layanan kepada client, server dituntut untuk memiliki tingkat availability yang tinggi. Hal tersebut memungkinkan mesin server mengalami down. Server juga harus dimatikan untuk keperluan pemeliharaan. Penelitian bertujuan ini membangun Clustering Server yang dapat bekerja bersama yang seolah merupakan sistem tunggal diatas lingkungan virtual. Hal ini merupakan solusi untuk mengatasi permasalahan tersebut. Pada penelitian ini penulis menggunakan server virtualisasi proxmox, FreeNAS sebagai server NAS dan DRBD untuk pendukung

ketersediaan layanan tinggi dalam lingkup HA, sinkronisasi data dalam High Availability (HA) yang dapat melakukan mirroring sistem kemesin lain. Dengan diterapkannya metode HA dan sinkronisasi DRBD serta penggunaan NFS (Network File System) pada sistem cluster didapatkan hasil rata-rata waktu migrasi sebesar 9.7(s) pada node1 menuju node2, 3.7(s) node2 menuju node3, dan 3(s) pada node3 menuju node1. Didapatkan juga waktu downtime yang lebih sedikit yaitu sebesar 0.58 ms pada node1, 0.02 ms pada node2, dan 0.02 ms pada node3.

Kata Kunci— Klaster, High Availability, proxmox VE, server, virtualisasi

I. PENDAHULUAN

Cloud Computing merupakan komputasi dimana kapabilitas terkait teknologi informasi disajikan sebagai suatu layanan (as a service), sehingga pengguna dapat mengaksesnya melalui internet tanpa mengetahui apa yang ada didalamnya beserta kendali terhadap infrastruktur teknologi yang membantunya.

Cloud computing mengacu pada paradigma berorientasi layanan di mana penyedia layanan menawarkan komputasi awan mengacu pada paradigma berorientasi layanan di mana penyedia layanan menawarkan komputasi sumber daya seperti perangkat keras, perangkat lunak, penyimpanan dan platform sebagai layanan sesuai dengan tuntutan pengguna[8]. Salah satu kendali yang berada didalam cloud computing adalah mesin server. Layanan harus dapat diakses kapanpun client ingin mengaksesnya, sedangkan server yang menampung data pada cloud harus dilakukan maintenance dan upgrade Oleh karena itu, diperlukannya clustering server yang bertujuan untuk mengurangi beban kerja server.

Untuk meningkatkan kualitas kerja server dibutuhkannya solusi Ketika terjadi masalah dalam satu node maka mesin virtual didalamnya akan berpindah ke node lain untuk meminimalkan gangguan pada layanan yang di akses client. Layanan-layanan server dijalankan pada mesin-mesin server virtual di dalam mesin server fisik. Jumlah layanan yang banyak, data-data penting, dan tingkat ketergantungan kinerja dari perusahaan, instansi, atau organisasi yang tinggi terhadap layanan server membuat server harus dapat melayani secara terus menerus[1].

High availability server digunakan untuk mengantisipasi kegagalan atau kerusakan devices pada komputer server yang dapat mengganggu kinerja sistem jaringan[3]. Downtime adalah waktu (period of time) dimana sistem tidak dapat digunakan untuk menjalankan fungsinya sesuai yang diharapkan. Downtime sangat berpengaruh pada nilai availability dari suatu equipment[2].

Penelitian bertujuan untuk mengoptimalkan kerja server dengan digunakannya sistem clustering pada server serta

digunakannya metode *high availability* guna meningkatkan dan membantu kinerja mesin *server* agar beban yang dikerjakan mesin *server* tidak terlalu berat

II. PENELITIAN TERKAIT

Masalah pada penelitian sebelumnya didapatkan bahwa DRBD tidak berfungsi seperti yang diharapkan. DRBD hanya bisa melakukan replikasi data hanya untuk 2 node. Proses replikasi hanya dapat dilakukan dengan menggunakan 2 node sementara adanya node 3 hanya sebagai cadangan (*Backup*). Sehingga fungsi replikasi tidak berjalan sebagaimana mestinya pada node ke-3[7].

Untuk meningkatkan dan mengoptimalkan kinerja server perlu dibangun *clustering server* agar ketika salah satu mesin *server* sedang mengalami gangguan ataupun sedang dilakukan pemeliharaan, layanan *cloud* tetap bisa diakses tanpa harus dilakukan pemberhentian layanan sementara. Walau sedang dilakukan *maintenance* pada *server*, pengguna dapat melakukan proses akses semua data yang ada di *cloud* dalam waktu, ruang dan tempat yang tidak dibatasi. Dari hasil uji coba pemanfaatan metode *high availability* yang didukung dengan bantuan *distributed replicated block device* didapatkan hasil *downtime* yang lebih rendah dan waktu migrasi yang lebih cepat.

Pada penelitian ini dilakukan perbandingan antara HA dengan bantuan DRBD pada node2 dan node3 dan HA tanpa bantuan DRBD pada node1. Pada *downtime* yang didukung metode HA dengan bantuan *distributed replicated block device* didapatkan nilai 0.02 ms pada node2, dan 0.02 ms pada node3. Pada uji coba untuk waktu migrasi yang didukung oleh metode HA dengan bantuan *distributed replicated block device* didapatkan hasil rata-rata 3.7(s) node2 menuju node3, dan 3(s) pada node3 menuju node1. Sedangkan dari hasil uji coba *downtime* pada node 1 yang tidak menggunakan bantuan DRBD didapatkan nilai 0.58 ms dan untuk waktu migrasi didapatkan nilai rata-rata 9.7(s) pada node1 menuju node2.

Kekurangan dari virtualisasi *server* adalah pengumpulnya semua *service* pada 1 mesin, sehingga apabila secara fisik mesin tersebut rusak atau *error* maka akan semua sistem yang berjalan di atasnya akan fail. Hal ini dapat diatasi dengan membuat mekanisme *redundant server* atau *fail over server* sebagai cadangan[5].

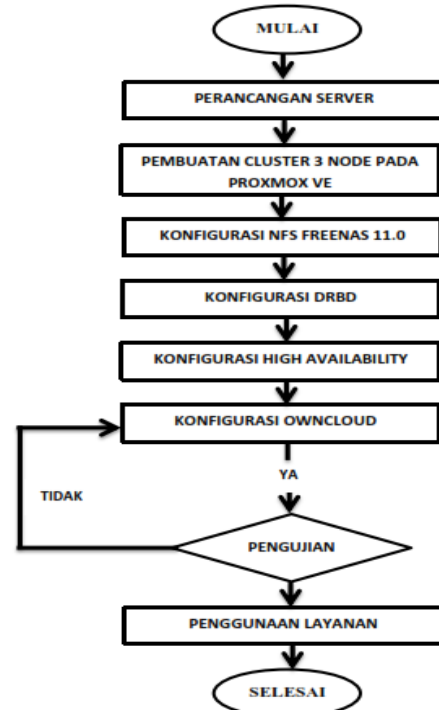
Waktu yang dibutuhkan pada saat migrasi *server*, dikarenakan node bermasalah, atau dikondisikan ada node yang mengalami kegagalan sistem, adalah : 1 menit 39 detik[6]. Pada perbandingan tabel penelitian sebelumnya didapatkan dalam 3 kali pengujian waktu migrasi melalui *server 1* ke *server 2* pengujian 1 didapatkan jumlah migrasi sebesar 7(s), pengujian ke-2 didapatkan waktu migrasi sebesar 13(s) dan pengujian ke-3 didapatkan waktu migrasi 11(s) yang total rata-ratanya adalah 10.3 *second*. Lalu pada *server 2* menuju ke *server 1* didapatkan waktu migrasi pada pengujian 1 sebesar 9(s), pengujian ke-2 sebesar 11(s) dan pengujian ke-3 sebesar 9(s), dengan total rata-rata sebesar 10.3 *second*. Salah satu *downtime* yang diambil dari jurnal acuan terdapat waktu *downtime* sebesar 0.7(s) pada *server 1* menuju *server 2* dan 0.7(s) dari *server 2* ke *server 1*[4].

Berdasarkan beberapa perbandingan migrasi selanjutnya terdapat hasil pengujian sebesar 51(s) pada pengujian 1, 23(s) pada pengujian ke-2 dan 42(s) pada pengujian ke-3 pada *server 1*, pada *server 2* terdapat hasil migrasi sebesar 56(s) pada pengujian 1, 24(s) pada pengujian ke-2 dan 44(s) pada pengujian ke-3. *Server 3* mendapatkan waktu migrasi sebesar

58(s) pada pengujian 1, 23(s) pada pengujian ke-2 dan 40(s) pada pengujian ke-3. Hasil *downtime* sebesar 1.3(s) pada *server 1*, 0.2(s) pada *server 2* dan 0.4(s) pada *server 3*[1].

III. METODE PENELITIAN

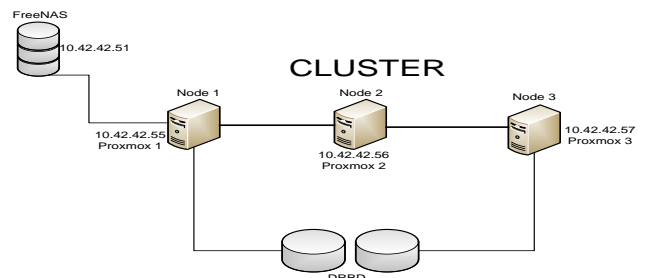
Alur Perancangan



Gambar 1 Flowchart Perancangan

Pada Gambar 1 menjelaskan perancangan sistem *clustering* menggunakan metode *High Availability*. Tahapan pertama dilakukan penginstalan proxmox VE, yang akan digunakan sebagai platform dalam pembuatan *clustering server*. Perancangan dalam sistem ini menggunakan 3 node yang akan dijadikan sebuah *cluster*. penulis menginstal *freenas* untuk penggabungan *storage* pada *cluster proxmox* dan menerapkan konfigurasi DRBD untuk mirroring pada setiap node dalam lingkungan HA guna menghindari gangguan layanan. Didalam proxmox penulis menginstal *owncloud* dalam virtual mesin. *Owncloud* telah dilakukan konfigurasi agar *client* dapat mengakses layanan pada *cloud*.

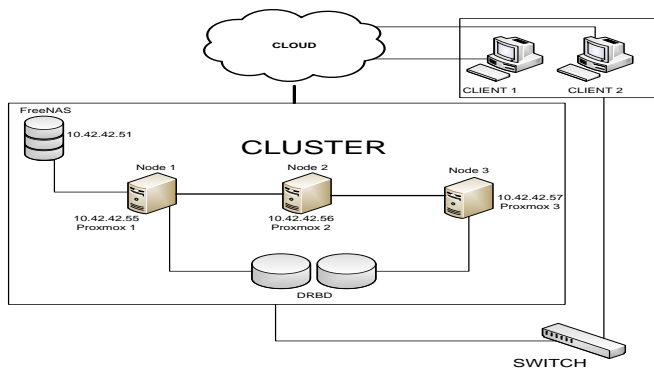
Desain Topologi



Gambar 2 Desain Topologi

Dari desain topologi diatas menjelaskan sebuah *server* virtual yang masing-masing memiliki ip berbeda yang sudah di konfigurasi menjadi node-node yang sudah ter *cluster* dengan layanan tinggi dapat diakses *client* melalui layanan *cloud computing*.

Konfigurasi Sistem Cluster



Gambar 3 Konfigurasi Cluster

Gambar diatas menggambarkan proses pembentukan 3 node menjadi cluster dengan bantuan DRBD untuk kestabilan layanan HA dan menggunakan storage freenas.

Tabel 1
Perangkat Keras

No	Hardware	Spesifikasi Hardwere
1	PC Proxmox 1,2,3	Intel Core i7-4500U CPU @1.80GHz (4 CPUs), ~2.4GHZ Windows 8.1 Pro 64-bit Memory 4096MB RAM Hardisk 1TB
2	PC NAS	Intel Core i5 Windows 7 Ultimate 64-bit Memory 4GB RAM Hardisk 500GB

Tabel 2
Perangkat Lunak

No	Software	Spesifikasi
1	Proxmox VE 5.0	Untuk membuat virtualisasi server cluster
2	FreeNAS 11.0	Untuk Server Network Attached Storage (NAS)
3	Owncloud	Sebagai Layanan yang akan diakses Client
4	Fping	Sebagai uji downtime

IV. HASIL DAN PEMBAHASAN

Pada tahapan pembuatan cluster seperti gambar dibawah menjelaskan:

```
#pvecm create mycluster, sebagai nama dari cluster yang dibuat.
#pvecm add, merupakan penambahan node cluster.
#pvecm status, merupakan pengecekan status apakah server telah tergabung.
```

```
Votequorum information
-----
Expected votes: 1
Highest expected: 1
Total votes: 1
Quorum: 1
Flags: Quorate

Membership information
-----
Nodeid Votes Name
0x00000001 1 10.42.42.55 (local)
root@node1:~# pvecm status
Quorum information
-----
Date: Mon Oct 2 14:56:18 2017
Quorum provider: corosync_votequorum
Nodes: 3
Node ID: 0x00000001
Ring ID: 1/16
Quorate: Yes

Votequorum information
-----
Expected votes: 3
Highest expected: 3
Total votes: 3
Quorum: 2
Flags: Quorate

Membership information
-----
Nodeid Votes Name
0x00000001 1 10.42.42.55 (local)
0x00000002 1 10.42.42.56
0x00000003 1 10.42.42.57
root@node1:~#
```

Gambar 4 Penggabungan Cluster

Menjadikan node1 sebagai server utama. Lalu gabungkan node 2 dan node 3 untuk dijadikan cluster. lakukan pengecekan dengan PING dan pvecm status untuk mengetahui semua node yang sudah terdaftar pada cluster.

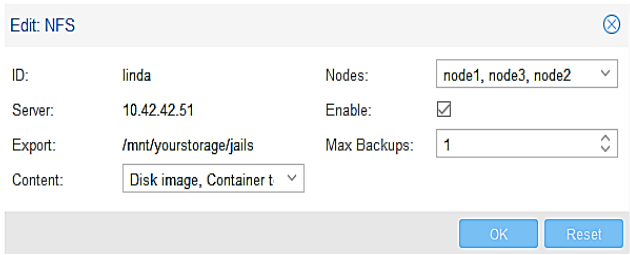
```
#apt-get install drbd8-utils, proses instal distributed duplicated block device.
#fdisk /dev/sdb, untuk mempartisi new virtual disk untuk drbd.
#nano etc/drbd.d/global_common.conf, untuk membuat konfigurasi DRBD global dan umum.
#nano /etc/drbd.d/r0.res, satu bagian sumber daya masing masing.
#resource r0, untuk memanggil fungsi drbd.
#protocol C, sinkronasi
pvcreate /dev/drbd0 (membuat physical volume disk)
#address 10.42.42.55:7788, sinkronasi standar port DRBD
#drbdadm--overwrite-data-of-peer primary r0, cek status service dan hasilnya akan tersinkronasi.
#watch cat /proc/drbd, proses sinkronasi tunggu hingga 100%.
#service drbd start, memulai DRBD.
#service drbd restart, mengulang kembali DRBD.
#service drbd stop, memberhentikan DRBD.
```

Gambar 5 ditunjukkan bagaimana cara menginstal DRBD (Distributed Replicated Block Device) sebagai backup data pada server disaat salah satu server sedang dalam keadaan mati atau maintenance.

Pada Gambar 6 ditunjukkan Setelah melakukan penginstalan freeNAS (Network Attached Storage), lakukan konfigurasi Storage NFS pada proxmox untuk penggabungan freeNAS dengan proxmox

```
root@node1:~# apt-get install drbd8-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

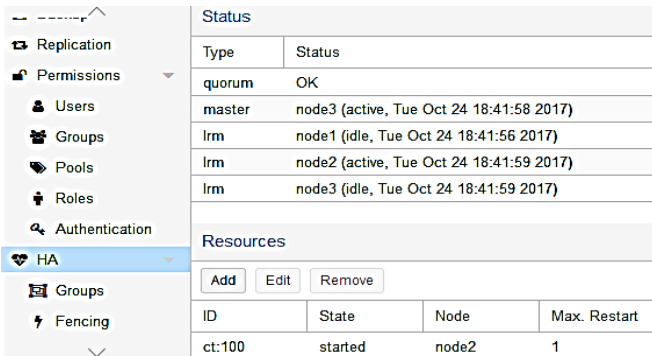
Gambar 5 Instalasi DRBD



Gambar 6 Konfigurasi Storage NAS

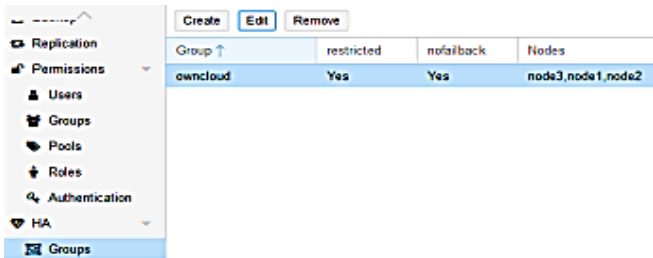
Tahapan perancangan *High Availability*:

- 1) Membuat *group* HA, untuk penggabungan ketiga *cluster*.
- 2) Memasukkan *vm* ID yang akan dimasukkan kedalam *group* HA.
- 3) Melakukan migrasi *vm* kepada ketiga *cluster*.



Gambar 7 Perancangan *High Availability*

Perancangan layanan ketersediaan tinggi yang bertujuan membantu perpindahan mesin virtual disaat salah satu *server* sedang mati.

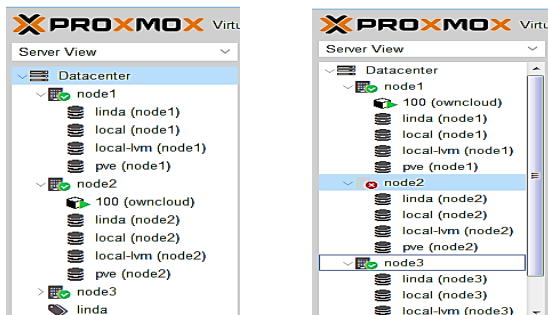


Gambar 8 Pengelompokan *server* HA

Pengelompokan *server* yang diberikan ketersediaan layanan tinggi untuk sistem *backup* pada mesin virtual agar bisa selalu diakses.

(a)

(b)



Gambar 9 Perpindahan Mesin Virtual *Owncloud*

Dari gambar diatas menjelaskan tahap cara kerja *High Availability* (HA) pada sistem *cluster*, berikut adalah penjelasannya:

Tahapan awal, *cloud* yang akan diakses terdapat di *server* node 2, dan tahap b menjelaskan ketika *server* node 2 dimatikan, maka *owncloud* akan berpindah ke *server* lainnya dalam keadaan yang stabil.

Tabel 3
Perpindahan Virtual Server

No	Node	Waktu Migrasi (s)			Rata-rata
		1	2	3	
1	Node 1 ke Node 2	10	9	10	9.7
2	Node 2 ke Node 3	4	4	3	3.7
3	Node 3 ke Node 1	3	3	3	3

Berdasarkan pengukuran waktu pada migrasi saat jalur aktif terputus, pengukuran waktu migrasi dilakukan pada *server* 1 menuju *server* 2, lalu dari *server* 2 menuju *server* 3 dan dimigrasikan kembali ke *server* 1.

Tabel 4

Infrastruktur as a Service

No	Ukuran File (KB)	CPU	Memory	Network Traffic (s)
100	164.1	2.1%	185.2Mb	46.45Kb
100	2915	8.18%	215.02Mb	58.06Kb

Berdasarkan penggunaan *Infrastruktur* pada *vm* 100 (*owncloud*) yang terletak pada node1. Uji coba memasukkan *file* sebesar 164.1Kb dan 2915Kb pada *owncloud*, maka hasil dari penggunaan kinerja pada CPU, *Memory* dan *Network Traffic* sesuai dengan tabel diatas.

Tabel 5

Penggunaan Kapasitas Node

N	CPU Delay	CPU Usage	Server Load	Memory	Kapasitas Memory	Network Traffic (s)
n1	68.79%	9.65%	3.43	1.01Gb	22%	10.73MB/s
n2	22.16%	5.24%	0.64	1.01Gb	25%	4.31Mb/s
n3	2.66%	6.46%	0.7	715.22Mb	35%	21.58 Kb/s

Berdasarkan penggunaan CPU ketiga node ketika *server* sedang bekerja. Terdapat perubahan pada masing masing diagram di *server*.

Tabel 6 Pengukuran *Downtime*

No	IP Address	Status	Waktu (ms)
1	10.42.42.55 (Node1)	Alive	0.58 ms
2	10.42.42.56 (Node2)	Alive	0.02 ms
3	10.42.42.57 (Node3)	Alive	0.02 ms

Pengukuran waktu pengukuran *downtime* pada saat migrasi dilakukan, terlihat hasil sesuai pada tabel pada saat uji *elapsed time on return packets* menggunakan *Fping*.

V. KESIMPULAN

Berdasarkan hasil pengujian dapat diambil beberapa kesimpulan. Hasil dari implementasi sistem *cluster* yang menggunakan metode *High Availability* yang didukung oleh lingkup DRBD mampu memberikan waktu migrasi dan *downtime* yang sangat rendah pada *server* 2 dan 3. Berbeda dengan *server* 1 yang tidak diberikan konfigurasi DRBD yang mendapatkan waktu migrasi dan *downtime* yang lebih tinggi yaitu dengan rata-rata 9.7(s) untuk waktu migrasi dan 0.58 ms

untuk waktu *downtime*. Dengan adanya *resource* yang tinggi, mampu mengurangi *downtime* yang terlalu lama pada *server*. Dapat disimpulkan waktu migrasi dari *server* 1 ke *server* lainnya sebanyak rata-rata 9.7 *second* pada node1 ke node2. Lalu didapatkan rata-rata waktu migrasi 3.7 *second* dari node2 ke node3 dan rata-rata 3 *second* dari node3 kembali ke node1. Dengan waktu *downtime* 0.58 ms pada node1, 0.02 ms pada node2 dan 0.02 ms pada node3.

VI. DAFTAR PUSTAKA

- [1] Yudi Restu Adi, Oki Dwi Nurhayati, Eko Didik Widiyanto “Perancangan Sistem Cluster Server Untuk Jaminan Ketersediaan Layanan Tinggi Pada Lingkungan Virtual”, JNTECI, Vol. 5, No. 2, Mei 2016.
- [2] Irfani, “Implementasi High Availability Server Dengan Teknik Failover Virtual Computer Cluster”, <http://eprints.ums.ac.id/35179/1/NASKAH%20PUBLIK%20ASI.pdf>.
- [3] I Gede Putu Krisna Juliharta, Wayan Supedana, Dandy Pramana Hostiadi, “High Availability Web Server Berbasis Open Source Dengan Teknik Failover Clustering”, Seminar Nasional Teknologi Informasi dan Multimedia 2015 *STMIK AMIKOM Yogyakarta, 6-8 Februari 2015*.
- [4] Muhammad Hafiz Zidni, “Cloud Storage Dengan Implementasi Network Attached Storage Metode Cluster Pada Server Virtual”.
- [5] Arief Arfriandi, “Perancangan, Implementasi, Dan Analisis Kinerja Virtualisasi Server Menggunakan Proxmox, Vmware Esx, Dan Openstack”, JURTEK http://jurtek.akprind.ac.id/sites/default/files/182_192_A%20RIEF.pdf.
- [6] Agni Isador Harsapranata, Ahmad Rais Ruli “Pemanfaatan Open Source Dalam Clustering Untuk Penerapan High Availability Pada Server”, Konferensi Nasional Ilmu Sosial & Teknologi (KNiST) Maret 2017, pp. 132~136.
- [7] Sugeng Purwantoro E.S.G.S, Muhammad Arief Fadhly, “Distributed Replicated Block Device (DRBD) Sebagai Alternatif High Availability Data Replication Pada Cloud Computing”, Jurnal Politeknik Caltex Riau <http://jurnal.pcr.ac.id>.
- [8] A. Kovari, P. Dukan, “KVM & OpenVZ virtualization based IaaS Open Source Cloud Virtualization Platforms: OpenNode, Proxmox VE”, College of Dunaújváros/Institute of Informatics, Dunaújváros, Hungary, <http://ieeexplore.ieee.org/document/6339540/>.

