

## Perbandingan Uji Analisa Performansi Owncloud Dengan Menggunakan Network Attached Storage Dan Distributed Replicated Blocked Device

Desy Mahmuda<sup>\*1</sup>, Diah Risqiwati<sup>2</sup>, Denar Regata Akbi<sup>3</sup>

<sup>1,2,3</sup>Universitas Muhammadiyah Malang

dessymahmuda@webmail.umm.ac.id<sup>\*1</sup>, risqiwati@umm.ac.id<sup>2</sup>, dnarregata@umm.ac.id<sup>3</sup>

### Abstrak

Kebutuhan pada jaringan dengan mengutamakan performa untuk mendukung sebuah efisiensi jaringan merupakan hal penting. Salah satu kesulitan yang di alami yaitu dalam hal pengarsipan. Untuk membuat sstem High Availability disini digunakan metode failover dimana metode failover merupakan metode ketersediaan siste jika salah satu server down atau mati. Tujuan dari penelitian ini yaitu membangun sistem penyimpanan berbasis cloud yang handal guna mencegah terjadinya down pada server dan membandingkan performansi sistem penyimpanan cloud computing menggunakan Network Attached Storage dan Distributed Replicated Bloced Device. Hasil dari penelitian ini yaitu menunjukkan bahwa sistem penyimpanan cloud menggunakan Distributed Replicated Block Device lebih baik dibandingkan dengan Network Attached Storage dilihat dari beberapa pengujian yaitu pengujian High Availability, CPU, kehandalan I/O dan MySQL.

**Kata Kunci:** Network Attached Storage, Distributed Replicated Blocked Device, Failover

### Abstract

The need for networks to prioritize performance to support a network efficiency is important. One of the difficulties experienced is in terms of archiving. To create a High Availability system here the failover method is used where the failover method is the system availability method if one of the servers is down or dead. The purpose of this research is to build a reliable cloud-based storage system to prevent server downtime and compare the performance of cloud computing storage systems using Network Attached Storage and Distributed Replicated Bloced Devices. The results of this study indicate that cloud storage systems using Distributed Replicated Block Devices are better than Network Attached Storage seen from several tests namely testing High Availability, CPU, I / O and MySQL reliability.

**Keywords:** Network Attached Storage, Distributed Replicated Blocked Device, Failover

### 1. Pendahuluan

Media penyimpanan data digital saat ini sudah memasuki era yang baru. Jika dulu hanya mengenal media penyimpanan data seperti *harddisk* dalam perangkat komputer atau *flashdisk* yang *portable* sehingga bisa dibawa kemana-mana, kini perkembangan teknologi telah menawarkan media pengolah dan penyimpanan data secara *online* yang dikenal dengan *cloud computing*. *Cloud computing* adalah metafora dari jaringan komputer/internet, dimana cloud (awan) merupakan penggambaran dari jaringan komputer/internet yang diabstraksi dari infrastruktur kompleks yang disembunyikan [1]. Secara sederhana dapat dikatakan bahwa para pengguna komputer bisa menggunakan *resource* tanpa perlu membeli, memiliki atau menginstal program, namun cukup dengan menyewa sumber daya dari *server* inti sesuai dengan kebutuhan. Media penyimpanan secara *online* yang disimpan pada alat penyimpanan *virtual* pada *cloud computing* disebut *cloud storage*.

Pada dasarnya teknologi *cloud storage* merupakan pengembangan dari sistem *cloud computing*. *Cloud computing* merupakan konsep dasar dari adanya layanan *cloud storage*. Dengan penerapan teknologi *cloud computing*, penyedia layanan *cloud storage* bisa membangun media penyimpanan secara *online* tersebut. Mengenai *cloud computing*, teknologi ini merupakan salah satu teknologi jaringan internet yang memiliki sejarah pengembangan yang cukup panjang. Tidak seperti media *offline* yang membutuhkan perangkat khusus, kini dengan adanya teknologi tersebut, dapat bisa lebih mudah mengakses data digital hanya berbekal perangkat yang telah

dilengkapi akses internet. Kelebihan yang ditawarkan *cloud storage* tentunya lebih banyak. Selain data yang disimpan terjaga keamanannya, tidak perlu khawatir juga jika tiba-tiba terjadi masalah pada perangkat elektronik. Semua data penting yang di miliki pun tetap tersimpan aman di dalam *cloud storage*.

*Private cloud storage* adalah jenis mekanisme penyimpanan yang menyimpan data organisasi di *server* penyimpanan *internal* dengan menerapkan *cloud computing* dan teknologi penyimpanan. Penyimpanan *private cloud* mirip dengan penyimpanan *public cloud* karena menyediakan kegunaan, skalabilitas, dan fleksibilitas arsitektur penyimpanan. Namun tidak seperti penyimpanan *public cloud*, *private cloud* tidak dapat diakses publik dan dimiliki oleh satu organisasi dan mitra *eksternal* resminya.

Salah satu aplikasi media pelayanan *cloud storage* yaitu *Owncloud*. *Owncloud* adalah perangkat lunak *server client* untuk membuat dan menggunakan layanan *file hosting*. *OwnCloud* secara fungsional sangat mirip dengan *Dropbox* yang banyak digunakan, dengan perbedaan fungsional utama bahwa *Server ownCloud* adalah gratis dan *open-source*, dan dengan demikian memungkinkan siapa pun untuk menginstal dan mengoperasikannya tanpa biaya pada *server* pribadi [5]. Ini juga mendukung ekstensi yang memungkinkannya berfungsi seperti *Google Drive*, dengan pengeditan dokumen *online*, kalender dan sinkronisasi kontak, dan banyak lagi. Keterbukaannya menghindari kuota yang diberlakukan pada ruang penyimpanan atau jumlah *client* yang terhubung, alih-alih memiliki batas keras (seperti pada ruang penyimpanan atau jumlah pengguna) yang hanya ditentukan oleh kemampuan fisik *server*.

Seiring dengan berkembangnya internet, penyedia layanan juga harus mempertimbangkan banyaknya jumlah pengguna yang terus bertambah. Sebuah *server* pasti memiliki keterbatasan dalam kemampuan menangani pengguna. Salah satu cara untuk mengatasi hal tersebut adalah dengan penambahan jumlah *server*. Dengan penambahan *server*, layanan yang sudah ada tetap berjalan ketika proses konfigurasi sedang dilakukan sehingga tidak akan sampai mematikan layanan. Untuk mendukung ketersediaan sebuah sistem yaitu dengan membangun infrastruktur terdistribusi, dimana peran satu *server* dibagi kepada beberapa *server* sehingga ketersediaan sistem yang telah dibangun menjadi meningkat.

Untuk membangun sistem *high availability*, dibutuhkan metode *failover*. *Failover* berarti bahwa ada sistem cadangan yang dikonfigurasi sebagai sistem yang standby untuk sistem yang utama. Ketika terjadi kegagalan, sistem cdangan akan melakukan takeover.

## 2. Metode Penelitian

Metode penelitian yang digunakan adalah *Network Development Life Cycle* (NDLC), karena hasil penelitian berupa desain (perancangan) yang membutuhkan *planning* (perencanaan) yang melibatkan analisa kebutuhan perangkat keras dan perangkat lunak, analisa kebutuhan pengguna serta analisa topologi jaringan tempat penelitian [3]. Tahapan penelitian NDLC terdiri dari enam tahapan, yaitu analisa, desain, simulasi, implementasi, *monitoring* dan manajemen.

Tahap analisa adalah tahapan mengenai analisa kebutuhan perangkat, analisa permasalahan yang muncul, dan analisa topologi atau jaringan yang akan dibangun. Kebutuhan sistem yang perlu dibangun adalah *Data Center* dan *Online Storage*. *Data Center* digunakan untuk keperluan pendistribusian data yang dinamis berupa penyimpanan data dan pengolahan data. *Online Storage* digunakan sebagai adalah media penyimpanan data *online* dengan teknologi *Cloud Computing* yaitu layanan *Cloud Storage* yang dilengkapi sistem *Management User* dan *Management Files* [4].

Tahap desain adalah pembuatan gambar desain topologi jaringan dan arsitektur sistem yang akan dibangun. Topologi dan arsitektur sistem yang dirancang mengacu pada hasil dari data yang didapat pada tahap analisa.

Tahap simulasi yang dilakukan adalah penggunaan alat bantu sebuah PC *Virtual* untuk mencegah kesalahan yang mungkin terjadi. Tahap sismulasi dilakukan dengan menggunakan aplikasi *Oracle Virtual Box* versi *Windows*. Tahap simulasi bisa dilakukan bersamaan dengan tahap implementasi.

Tahap implementasi adalah tahapan penerapan semua yang telah direncanakan dan didesain sebelumnya. Penerapan yang dilakukan yaitu pemasangan dan konfigurasi *server virtual*, serta pemasangan perangkat sesuai dengan desain jaringan yang telah dilakukan.

Tahap *monitoring* adalah tahapan evaluasi dan pengawasan terhadap sistem yang telah dibuat. Evaluasi yang dilakukan adalah pengujian terhadap sistem yang telah diimplementasikan.

Pengujian terhadap sistem terbagi menjadi beberapa pengujian, yaitu pengujian sistem *Network Attached Storage* (NAS), pengujian kemampuan HA, dan pengujian fungsi-fungsi *OwnCloud* menggunakan *web browser*.

Tahap manajemen mengatur mengenai pengaturan pengguna yang bisa mengakses *server virtual*. Pembatasan hak akses bertujuan agar sistem yang telah dibuat bisa dikelola dengan baik dan tertata.

### 3. Hasil Penelitian dan Pembahasan

#### 3.1 Pengujian

##### 3.1.1 Pengujian High Availability

Pengujian HA merupakan pengujian terhadap kemampuan *Server 1* dan *Server 2* dalam menjalankan *failover* VM yang sedang berjalan. Ping dilakukan dengan menggunakan interval waktu yaitu 0.1 detik. Pengujian ini dapat mengukur berapa lama waktu down server NAS dengan cara melihat *icmp\_seq* yang hilang dari barisan ping.

Tabel 1. Percobaan waktu *Failover* dan Migrasi

Percobaan	NAS (detik) Dari Server 1 ke Server 2	DRBD (detik) Dari Server 1 ke Server 2
	Migrasi	Migrasi
1	6	4
2	7	5
3	4	6
Waktu rata-rata (detik)	5,88	5

Dari Tabel 1 di atas dapat disimpulkan bahwa waktu *down* selama *failover* terjadi pada arsitektur NAS lebih lama dibandingkan dengan arsitektur DRBD. Dari 3 percobaan yang telah dilakukan diketahui bahwa selesai waktu yang dihasilkan antara kedua arsitektur tersebut sangat kecil. Migrasi yang dilakukan menghasilkan sebuah waktu rata-rata down pada DRBD dari server 1 ke server 2 selama 5 detik, sedangkan pada NAS dari server 1 ke server 2 selama 5,88 detik.

##### 3.1.2 Keandalan CPU

Uji keandalan CPU dilakukan dengan menggunakan aplikasi *sysbench* yang dipasang pada Server DRBD dan server NAS. Metode yang dilakukan yaitu dengan melihat waktu yang dibutuhkan CPU untuk memproses bilangan prima dengan batas 1000.

Tabel 2. Pengujian performa CPU

	DRBD (detik)	NAS (detik)
Percobaan 1	22,834	21,0331
Percobaan 2	20,2585	19,9430
Percobaan 3	19,8527	19,9361
Waktu rata-rata (detik)	20,98173333	20,3041

Tabel 2 diatas menunjukkan hasil dari pengujian keandalan yang dilakukan oleh server DRBD dan server NAS. Waktu rata-rata yang dibutuhkan CPU untuk memproses bilangan prima dari server DRBD adalah 20,9817 dan waktu rata-rata yang dibutuhkan CPU untuk memproses bilangan prima dari server NAS adalah 20,3041. Server NAS memiliki kecepatan processor yang lebih tinggi dibanding server DRBD.

##### 3.1.3 Keandalan I/O

Pengujian keandalan I/O dilakukan pada server DRBD dan server NAS dengan menggunakan aplikasi *sysbench*. Metode yang dilakukan oleh *sysbench* yaitu dengan melakukan operasi baca, tulis, dan lainnya yang dilakukan selama 100 detik terhadap I/O dari server NAS dan DRBD.

Tabel 3. Pengujian performa I/O

	Waktu (detik)	Banyaknya Operasi				Ukuran Berkas (Mb)	Kecepatan rata-rata (Mb/detik)
		Baca	Tulis	Lainnya	Jumlah		
DRBD	100	5567	3711	11776	21054	144,97	1,4495
NAS		5473	3648	11648	20769	142,52	1,4251

Hasil yang didapatkan berupa kecepatan rata-rata server dalam membaca dan menulis melalui I/O. Tabel 3 menunjukkan hasil uji kehandalan I/O server. Hasil yang didapatkan yaitu server DRBD dengan kecepatan rata-rata 1,4495Mb/detik sedangkan server NAS dengan kecepatan rata-rata 1,4251Mb/detik. Tidak terlihat perbedaan yang signifikan dari pengujian performa I/O pada kedua arsitektur, baik pada hasil pengujian *read* (baca), *write* (menulis), dan juga kecepatan rata-rata.

### 3.1.4 MYSQL

Pengujian MySQL dilakukan untuk mengukur performa dari MySQL pada kedua arsitektur. Pengujian ini dilakukan dengan menggunakan aplikasi *sysbench* yang dipasang pada server DRBD dan server NAS. Metode yang digunakan yaitu menghitung kecepatan rata-rata server DRBD dan server NAS dalam memproses transaksi tiap detiknya. Transaksi yang dilakukan adalah sebanyak 500.000 baris perintah. Dan dilakukan bertahap dengan menambah jumlah *threads* mulai dari 1 sampai dengan 128 *threads*.

Tabel 4. Pengujian MySQL

THREADS	DRBD		NAS	
	Transaction Per Second (TPS)	Avg. Response Time	Transaction Per Second (TPS)	Avg. Response Time
1	364,18	2,74	343,93	2,9
2	347,4	5,75	343,6	5,81
4	358,58	11,15	339,75	11,76
8	353,09	22,64	333,2	23,99
16	343,97	46,47	335,09	47,7
32	344,29	92,81	334,12	95,64
64	350,69	182,03	326,68	195,25
128	334,44	381,37	321,32	396,48

Berdasarkan hasil pengujian Tabel 4 di atas, terlihat bahwa arsitektur DRBD memiliki performa MySQL lebih baik dari pada arsitektur NAS. Hal itu ditandai dengan waktu *Transaction Per Second* (TPS) di tiap pengujian lebih kecil pada arsitektur DRBD dibandingkan dengan waktu TPS pada arsitektur NAS.

### 3.2 Kesimpulan

Berdasarkan seluruh rangkaian proses implementasi yang sudah dilakukan, maka dapat disimpulkan bahwa performansi *OwnCloud* dengan konsep DRBD menunjukkan hasil yang lebih baik dibandingkan performansi *OwnCloud* dengan konsep NAS. Pengujian performansi yang menunjukkan hasil paling signifikan antara konsep DRBD dan NAS yaitu: Pengujian *High Availability* dan Pengujian MySQL. Sehingga dapat disimpulkan bahwa arsitektur DRBD menghasilkan performa yang lebih baik dibandingkan *OwnCloud* dengan konsep NAS.

### Referensi

- [1] Hurwitz, Judith, et.al. 2010. *Cloud Computing for Dummies*. Wiley Publishing. Indiana.
- [2] Luka Perkov, 2011, "High Availability using Open Source software"
- [3] P. Jay Salzman, 2001, "The Linux Kernel Kernel Module Programming Guide"
- [4] Jim Gray, Daniel P. Siewiorek, 1991, "High Availability Computer Systems"
- [5] Steven J. Vaughan-Nichols (2012-10-11). "OwnCloud: Build your own or manage your public cloud storage services". ZDNet. Retrieved 2016-09-20.

- 
- [6] Wijaya, Miftakhudin Kusuma, 2018, Skripsi "Implementasi *High Availability Cloud Storage* dengan Metode Replikasi dan Failover pada Laboratorium Teknik Informatika"
- [7] Defini, dan Prabowo, C. 2013. Perancangan dan Implementasi Data *Loss Prevention System* dengan menggunakan *Network Attached Storage*. Jurnal TEKNOIF. 1(2): 45-60.
- [8] Sigh, Sukhdeep, 2014. *Implementing Storage as a Service Cloud using Network Attached Storage*. *International Journal of Computer Applications*, Volume 108-No.13(Print), Desember 2014.
- [9] Yafid, Muhammad Said, 2017, Skripsi "Implementasi Freenas Sebagai Penyimpanan Pada Proxmox Menggunakan Metode Failover: Metode Failover"

