

Implementasi High Availability Cloud Storage Dengan Metode Replikasi Dan Failover Pada Laboratorium Teknik Informatika

Miftakhdin Kusuma Wijaya^{*1}, Zamah Sari², Mahar Faiqurahman³

^{1,2,3}Teknik Informatika/Universitas Muhammadiyah Malang

atfim25@gmail.com^{*1}, abdzamahsari@gmail.com², maharf@gmail.com³

Abstrak

Cloud storage merupakan salah satu bentuk dari cloud computing yang terpusat pada media penyimpanan data. Dalam cloud storage ada kemungkinan yang di alami seperti kehilangan data dengan sekala kecil maupun besar, ataupun tidak dapat di akses sama sekali. Kesalahan yang terjadi akibat bencana alam, kesalahan manusia, ataupun alat yang cukup usang. Dari permasalahan tersebut di lakukan backup dan sinkronisasi data untuk menanggulangi permasalahan yang ada. Replikasi adalah proses yang di gunakan untuk menyalin atau mendistribusikan data dari penyedia layanan ke perangkat backup. Replikasi yang di gunakan ada dua yaitu replikasi database MySql dan replikasi data Rsync, dan untuk menjaga agar cloud storage tetap menyediakan sumber daya untuk pengguna dengan menambahkan metode failover. Failover adalah peralihan dari sebuah perangkat penyedia layanan ke perangkat backup ketika mengalami permasalahan yang tidak di inginkan. Pada penelitian ini akan menjelaskan bagaimana cara membangun dan mengimplementasi infrastruktur cloud storage dengan replikasi untuk backup juga sinkronisasi data dan failover untuk memberikan ketersediaan sumber daya layanan untuk pengguna secara realtime.

Kata Kunci: Cloud Storage, Rsync, MySql, Failover

Abstract

Cloud storage is one form of cloud computing but is centered on data storage media. In cloud storage there is the possibility of data loss with small or large scale or cannot be accessed at all. It can happen from natural disasters, human error, or device oldness. From these problems we can do data backup and data synchronization to overcome these problems. Replication is a process that used to copy or distribute data from service providers to backup devices. This replication using two ways, there is replication database using MySql and data replication using Rsync. To keep cloud storage provide resource for user by adding failover. This failover is a transition from a service provider device when having problems to backup device. In this study will explain how to build and implementation cloud storage infrastructure with replication for backup as well as data synchronization and failover to provide real-time availability of service resources for users.

Keywords: Cloud Storage, Rsync, MySql, Failover

1. Pendahuluan

Cloud Storage merupakan salah satu bentuk dari cloud computing namun terpusat pada media penyimpanan yang dalam pengaksesannya memerlukan jaringan internet maupun intranet. Di cloud storage seseorang di beri hak untuk melakukan akses ke resource yang di miliki oleh penyedia layanan cloud storage. Sehingga pengguna tidak harus membawa storage dengan fisik besar seperti hardisk dan alat penyimpanan lain. Karena cloud storage sudah menyediakan layanan penyimpanan dan dapat lebih efisien waktu dan tempat di bandingkan penyimpanan fisik [1].

Owncloud merupakan aplikasi open source yang juga mendukung teknologi cloud computing. Owncloud menyajikan layanan cloud storage seperti dropbox, yaitu layanan penyimpanan dan pengambilan file secara online pada jaringan cloud computing [2]. Di dalam cloud storage memiliki suatu server database, dimana server database ini merupakan tempat penyimpanan seluruh data atau konfigurasi yang di lakukan pada cloud storage.

Dalam cloud storage ada kemungkinan terjadi kerusakan atau kehilangan file dalam skala besar, karena di sebabkan oleh berbagai faktor seperti hardisk bad sector atau hardisk mati. Oleh

sebab itu pada cloud storage perlu di tambah beberapa *backup* dan sinkronisasi data untuk mengantisipasi hal-hal tersebut. Replikasi adalah proses yang di gunakan untuk menyalin dan mendistribusikan data pada suatu *server database* agar dapat terhubung dengan baik antara *database* satu ke *database* lainnya. Untuk mengoptimalkan kerja suatu replikasi di butuhkan adanya *failover* [3].

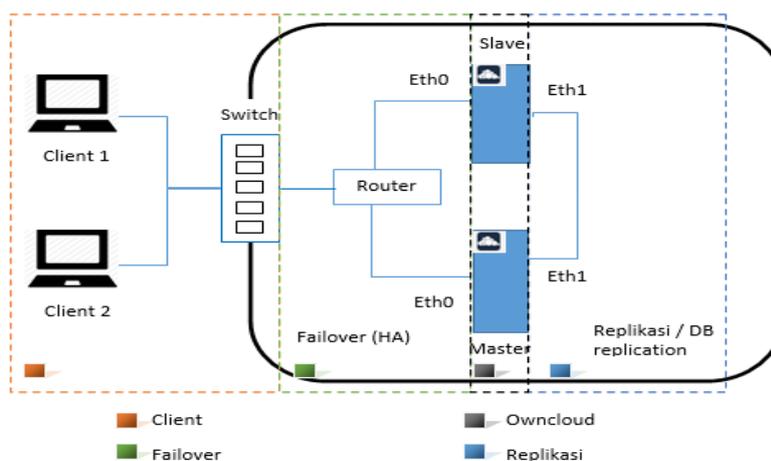
Failover adalah peralihan ke sebuah perangkat komponen atau *server* cadangan saat *server* utama mengalami gangguan. Dengan menggunakan *failover* kita dapat mengakses data yang di simpan pada saat *server* utama mengalami gangguan [4].

Failover clustering merupakan salah satu bentuk *failover* yang menyediakan solusi *high availability server* dimana jika terjadi kegagalan pada sistem *hardware* seperti *power supply* mati yang menyebabkan *server* mati total maka *server* lain anggota *cluster* yang akan mengambil alih fungsi dari *server* yang mati, sehingga pengguna tidak mengetahui terjadinya kegagalan pada *server* karena proses yang dilakukan pada *server* yang gagal atau mati akan dilanjutkan oleh *server backup* [5]. *Failover* memiliki beberapa metode dalam menangani kegagalan sistem atau komponen. *Warm Stanby* merupakan salah satu metode *failover* dimana metode ini membutuhkan replikasi dan *failover* otomatis untuk menggantikan komponen atau *service* utama yang gagal. Solusi ini dipakai untuk melakukan redundansi *hardware* dan *software*. Di dalam metode ini *server* utama secara teratur akan melakukan replikasi ke *server* cadangan. Ketika *server* utama mengalami kegagalan *server* cadangan akan menggantikan peran *server* utama berdasarkan replikasi terakhir yang tercatat dari *server* utama sebelum *server* utama mengalami kegagalan [6].

Karena berhubungan dengan Laboratorium Teknik Informatika, yang membutuhkan tempat penyimpanan untuk menyimpan dan mendistribusikan data secara *flexible* dari pihak administrasi ke mahasiswa teknik informatika maupun, dari pihak mahasiswa teknik informatika ke administrasi, dan juga membutuhkan *system back up*, dimana ketika *server* utama mati *server backup* akan mengambil alih peranan *server* utama. Dalam penelitian ini akan di implementasikan *high availability cloud storage* dengan metode replikasi dan *failover* agar memberi kinerja yang maksimal untuk *cloud private storage* pada Laboratorium Teknik Informatika.

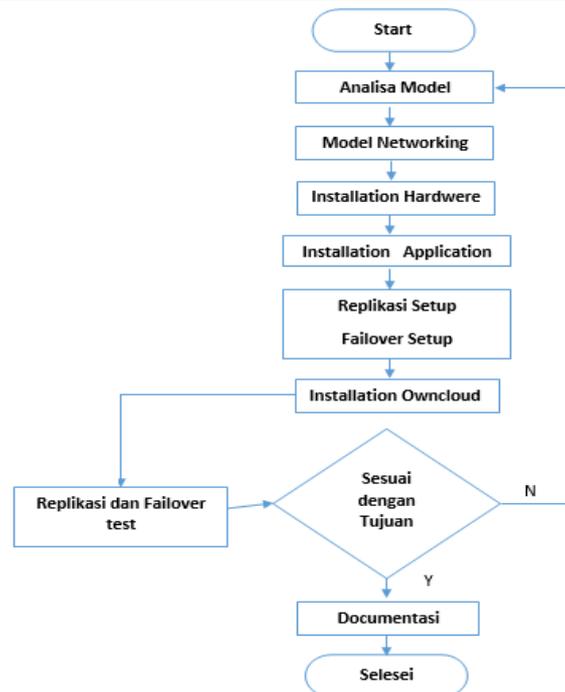
2. Metode Penelitian

Dalam perancangan ini saat kondisi di Lab Teknik Informatika belum tersedia *back up* data dan kontrol *server* ketika *server* utama mengalami gangguan. Efek dari kondisi di atas tidak menutup kemungkinan *server* tidak dapat di akses, *server* mati total dan bahkan kehilangan data pengguna atau user. Dengan permasalahan di atas di buatlah *high availability cloud storage* di mana ketika *server* mengalami gangguan akan di alihkan ke *server* cadangan dengan bantuan *failover*. Dan saat *server* utama mengalami gangguan peran *server* cadangan menangani data yang direquest oleh pengguna dengan bantuan replikasi. Gambar 1 merupakan arsitektur cloud storage yang berdasarkan perancangan sistem yang telah di buat dan di butuhkan sebelumnya



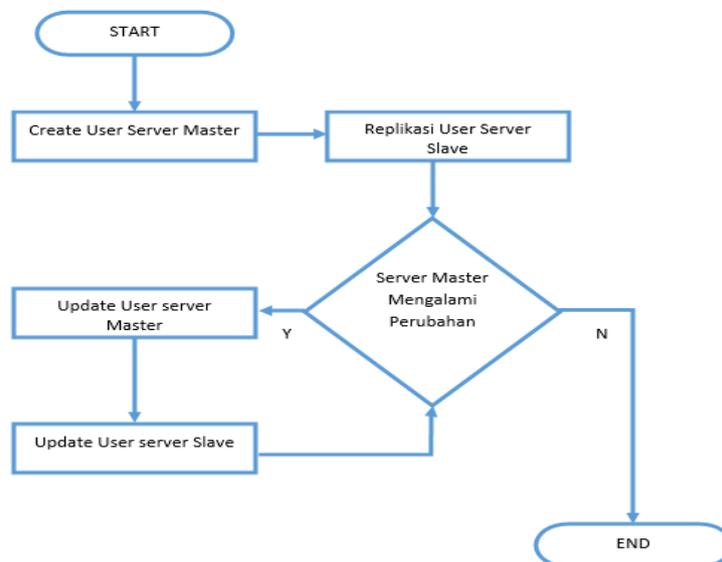
Gambar 1. Arsitektur Cloud Storage

Pada Gambar 2 merupakan alur perancangan sistem keseluruhan yang akan di buat dengan urutan penerapan yang terdapat pada asitektur cloud storage pada Gambar 1.



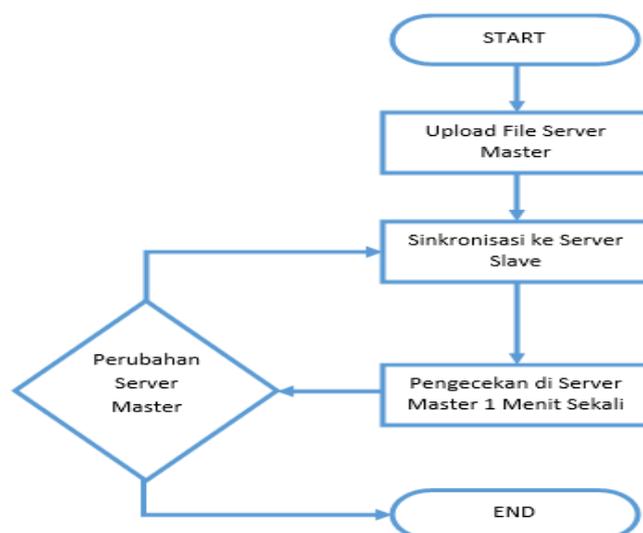
Gambar 2. Alur Perancangan Sistem

Gambar 2 merupakan alur proses replikasi database pada arsitektur cloud storage. Ketika *server master* mendapat instruksi *create user* maka akan dilanjutkan mereplikasi ke *server slave*. Dan dalam setiap waktu akan selalu ada pengecekan ketika *server master* mengalami perubahan, dan ketika *server* mengalami perubahan replikasi akan berjalan kembali dan di sinkronkan ke *server slave*, jika tidak ada perubahan data replikasi selesai.



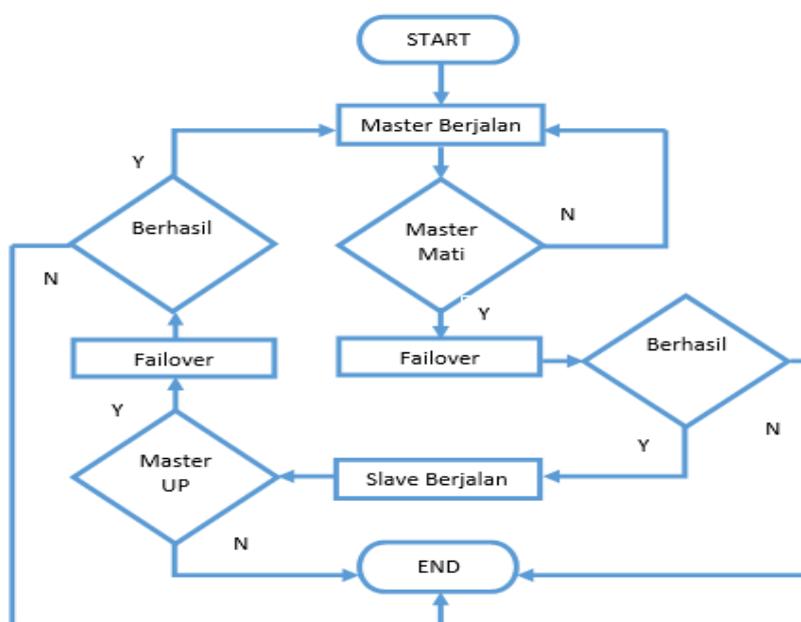
Gambar 3. Alur Proses Replikasi Database (MySQL)

Gambar 3 merupakan alur proses replikasi database pada arsitektur cloud storage. Ketika *user* memberikan instruksi untuk menambahkan *file* ke dalam *server master*, maka data yang sudah berhasil ditambahkan ke dalam *server master* akan di kirim atau di teruskan ke *server slave*. Proses replikasi *Rsync* menggunakan *cronjob* untuk memberikan waktu *realtime*, waktu minimal yang di butuhkan *cronjob* adalah setiap 1 menit dalam 1 jam untuk melakukan perintah replikasi.



Gambar 4. Alur Proses Replikasi Data (rsync)

Gambar 4 merupakan alur proses replikasi database pada arsitektur cloud storage. Ketika *server master* mati maka akan dilakukan *failover* ke *server slave*. *Server slave* akan mengambil alih semua beban dan *request* dari *client*. Ketika *server master* aktif, *failover* akan di alihkan kembali pada prioritas terbesar yaitu *server master*.



Gambar 5. Alur Proses Failover

Pada Gambar 5, dijelaskan tentang proses kerja dari *failover*. Ketika *server master* mati maka akan dilakukan *failover* ke *server slave*. *Server slave* akan mengambil alih semua beban dan *request* dari *client*. Ketika *server master* aktif, *failover* akan di alihkan kembali pada prioritas terbesar yaitu *server master*.

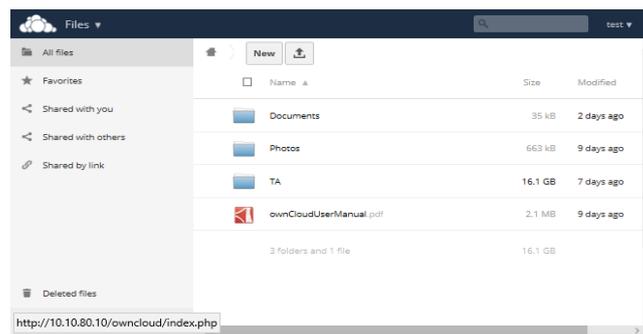
3. Hasil Penelitian dan Pembahasan

3.1 User Interface

Gambar 6 dan Gambar 7 merupakan tampilan cloud storage yang di akses oleh pengguna. Pengguna hanya dapat mengelola cloud mereka menggunakan user dan password pada Owcloud di web browser.



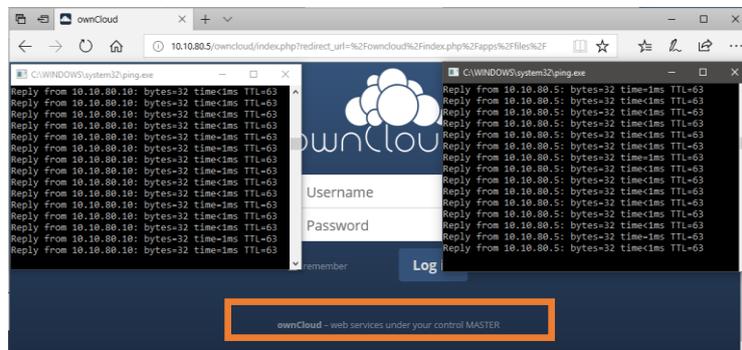
Gambar 6. Interface Login



Gambar 7. Interface Owncloud

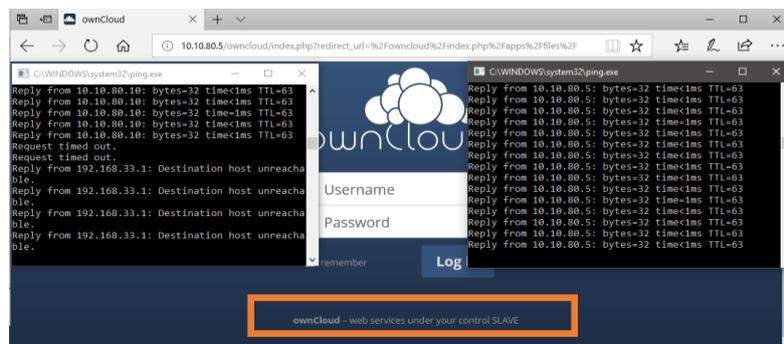
3.2 Pengujian Fungsional

Pengujian pertama yaitu pengujian failover menggunakan alamat Owncloud dengan melakukan login IP virtual dan cek ping IP address server master dan IP virtual.



Gambar 8. Pengujian Owncloud saat Master Up

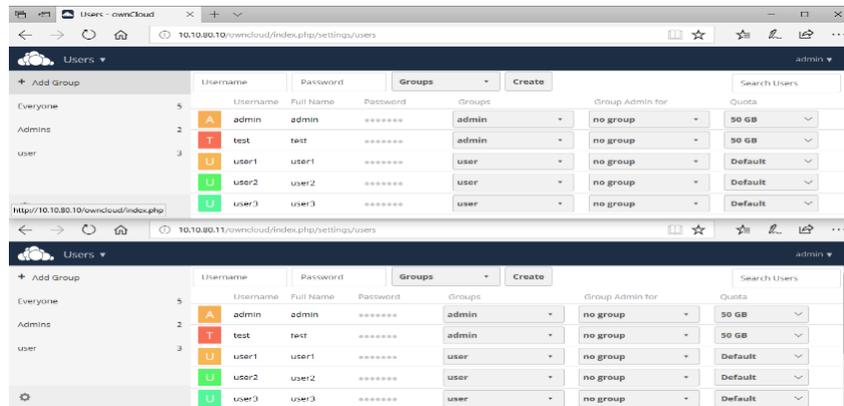
Terlihat pada Gambar 8, IP virtual dengan alamat 10.10.80.5/owncloud mengarahkan akses dari user ke server master, karena server master memiliki prioritas lebih besar dari slave.



Gambar 9. Pengujian Owncloud saat Master Down

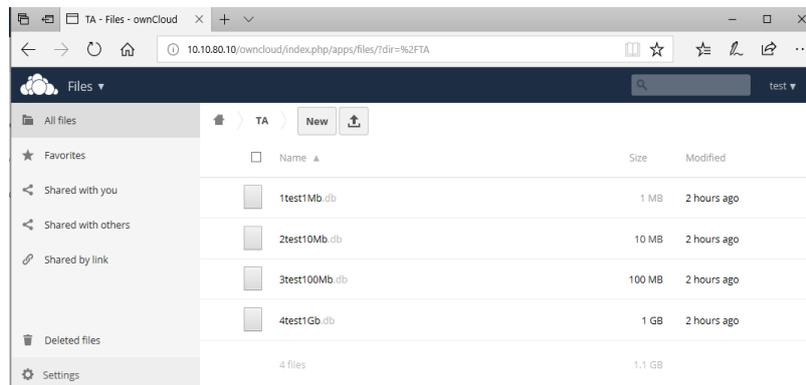
Terlihat pada Gambar 9 *IP virtual* dengan alamat 10.10.80.5 /owncloud mengarahkan akses dari *user* ke *server slave*. Karena *server master* yang memiliki prioritas tertinggi tidak dapat di akses maka, *server slave* yang memiliki prioritas di bawah *server master* yang akan menggantikan peran *master*.

Setelah pembuatan *user-user* pada *server master* melalui Owncloud menggunakan ID admin terlihat *user1*, *user2*, *user3*, selanjutnya melihat pada Owncloud di *server slave* dengan menggunakan *login ID admin*.



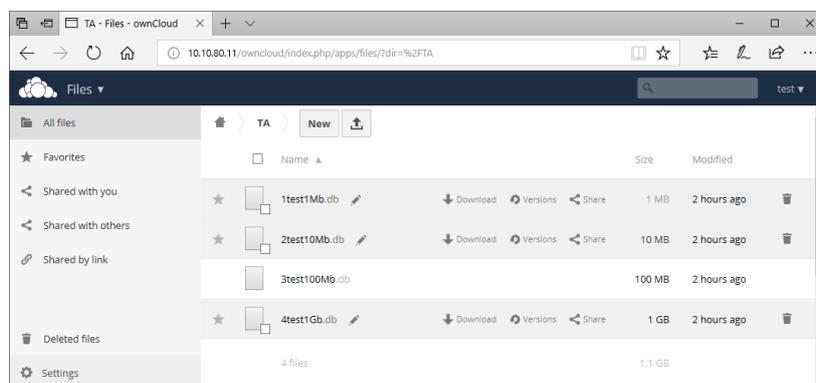
Gambar 10. User Sudah Replikasi

Sebelum pengujian dilakukan perlu dipersiapkan beberapa file yang berukuran berbeda dan di *upload* pada owncloud terlebih dahulu. Di pengujian ini menggunakan user test pada owncloud yang sudah dibuat sebelumnya, seperti pada Gambar 10.



Gambar 11 Beberapa File Upload

Gambar 11 di atas adalah beberapa file yang sudah di *upload* ke dalam server master dan siap untuk di lakukan pengujian.



Gambar 12. Data yang Sudah Terkirim

Gambar 12 merupakan hasil dari replikasi data menggunakan *rsync*, dapat dilihat jumlah dan besar *file* sama dengan jumlah atau besar *file* sebelum dilakukan replikasi pada *server master*.

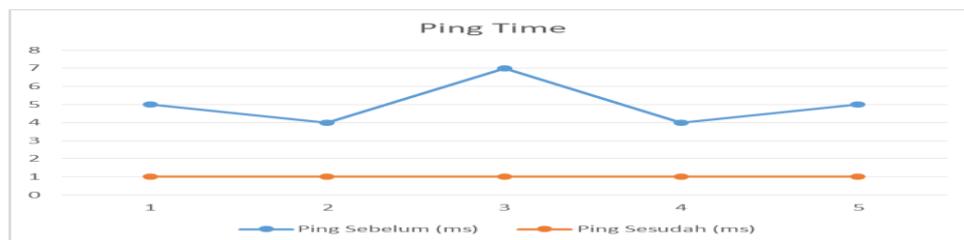
3.3 Pengujian Performasi

Pengujian performansi *failover* dengan mematikan *server master* untuk melihat perpindahan *master* ke *slave* dan juga diberi beban pada *server master*, beban yang dimaksud, yaitu *download* data 5GB dengan 5 kali percobaan yang terdapat pada Tabel 1.

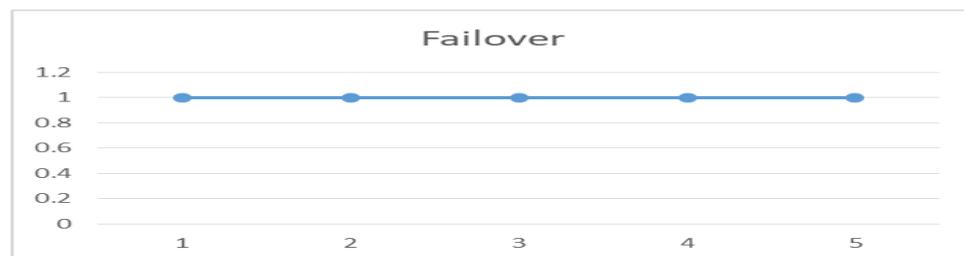
Tabel 1. Tabel Failover

Percobaan Master Down	Down time Failover	Ping Time Sebelum	Ping Time Sesudah
1	1 RTO	< 5 ms	< 1 ms
2	1 RTO	<4 ms	< 1 ms
3	1 RTO	< 7 ms	< 1 ms
4	1 RTO	< 4ms	< 1 ms
5	1 RTO	< 5 ms	< 1 ms

Pada Tabel 1 hasil dari *failover* sebanyak lima kali pengujian, *down time failover* adalah suatu proses *failover* untuk perpindahan antara *master* ke *slave* hasil tabel diambil dari proses *ping IP virtual* dengan alamat IP 10.10.80. Sebelum terjadi kegagalan dan dilakukan *failover* ke *server slave*, *ping time* kurang lebih 5ms-7ms dikarenakan *server master* memiliki beban *download* dan *ping time* sesudah dilakukan *failover* kurang lebih 1ms pada saat dilakukan *failover*, *server slave* tidak memiliki beban apapun, seperti grafik pada Gambar 13 dan Gambar 14.



Gambar 13. Grafik Ping Time Failover



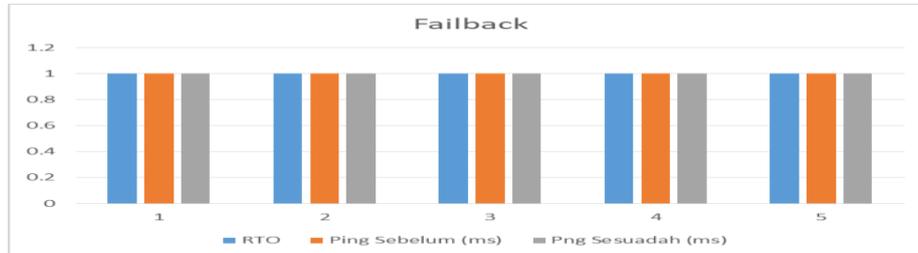
Gambar 14. Grafik RTO Failover

Pengujian *failback* ini di lakukan sebanyak 5 kali percobaan untuk mendapat data *delay failover failback*, dan *ping time* sebelum dan sesudah, dengan beban *download file* 10GB di *server master* dan *server slave*.

Tabel 2. Pengujian Failback

Percobaan Master Down	Failback	Ping Time Sebelum	Ping Time Sesudah
1	1 RTO	< 1 ms	< 1 ms
2	1 RTO	< 1 ms	< 1 ms
3	1 RTO	< 1 ms	< 1 ms
4	1 RTO	< 1 ms	< 1 ms
5	1 RTO	< 1 ms	< 1 ms

Tabel 2, merupakan beberapa pengujian yang di lakukan untuk melihat beberapa *failback*, *ping time* pada sistem *cloud storage* yang telah di buat. *Failback* merupakan perpindahan kembali dari *slave* ke *server* ketika *server* utama *UP* kembali. Dalam Tabel 2, pengujian dilakukan pada *IP server master* dan *IP virtual*. Dilihat ping time sebelum dan sesudah failback tidak mengalami perubahan dikarenakan, proses *download* yang terjadi pada *server master* telah terputus, maka trafik kembali kepada kondisi normal tanpa beban. *Ping time* sebelumnya adalah melakukan *ping* sebelum terjadinya *failback* dan *ping time* sesudah adalah *ping* sesudah *failback*, seperti pada Gambar 15.



Gambar 15. Grafik Failback

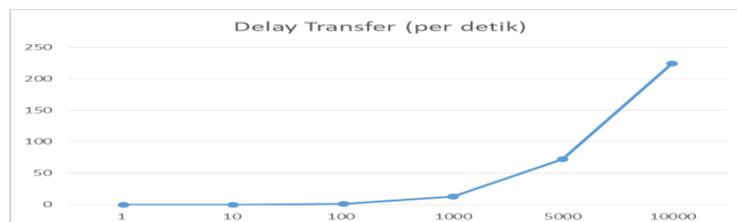
Pengujian performansi replikasi ini menggunakan *Rsync* untuk melihat kecepatan transfer saat terjadinya replikasi, pengujian di sini menggunakan 5 variasi data dari 1Mb sampai 10GB

Pada Tabel 3 serta Gambar 16 dan Gambar 17, menunjukkan kecepatan transfer menggunakan *rsync*, proses transfer dengan *rsync* berjalan satu per satu mulai dari atas, dan setelah dilakukan transfer, diperlukan berapa detik untuk dapat mentransfer beberapa file yang berbeda ukuran.

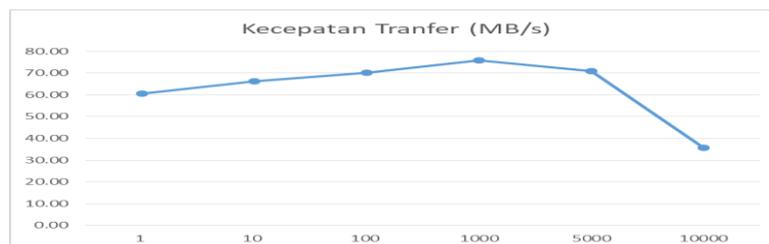
Tabel 3. Proses Replikasi Data atau File

Data	Ukuran Data	Kecepatan Transfer	Delay Pengiriman
1	1 Mb	60.55 MB/s	00:00:00
2	10 Mb	66.23 MB/s	00:00:00
3	100 Mb	70.32 MB/s	00:00:01
4	1 Gb	75.94 MB/s	00:00:13
5	5 Gb	71.00 MB/s	0:01:12
6	10 Gb	35.56 MB/s	0:03:44

Tabel 3 adalah hasil dari replikasi data menggunakan *rsync* dengan ukuran data yang berbeda. Data yang digunakan adalah data *binary* dengan asumsi, saat proses replikasi kondisi trafik jaringan tidak memiliki beban.



Gambar 16. Grafik Delay Transfer Data



Gambar 17. Grafik Kecepatan Transfer Data

Pengujian replikasi selanjutnya menggunakan beban pada saat terjadinya replikasi. Beban yang di maksud yaitu pada saat ada proses *download* data yang di lakukan pengguna dan ketika proses replikasi berjalan. Pada Gambar 18 adalah proses pengujian pada saat trafik sibuk dengan replikasi data 5GB.

```
test/files/Documents/5test5Gb.db
5,368,709,120 100% 56.47MB/s 0:01:30 (xfr#1, to-chk=23/61)
test/files/Photos/
test/files_trashbin/files/New folder.d1521372074/
sent 5,223,369 bytes received 1,421 bytes 45,631.35 bytes/sec
total size is 31,062,477,509 speedup is 5,945.21
```

Gambar 18. Proses Transfer Dengan Beban Trafik Jaringan

Gambar 18 merupakan hasil replikasi data dengan beban 5GB download, terlihat pada Tabel 3. Dimana kecepatan replikasi data 5GB pada saat tidak ada beban adalah 71.00MB/s dengan perbandingan kecepatan replikasi 5GB pada saat memiliki beban sekitar 65.47MB/s.

3.4 Pengujian Download Data

Pada pengujian ini dilakukan dengan cara mengunduh data dari Owncloud menggunakan *IP virtual, file* yang di *download* sebesar 10Gb agar pada proses pengunduhan memiliki waktu yang cukup untuk melihat kecepatan *download*. Pengujian yang di gunakan dengan memberi variasi pada jumlah *user* yang *download* secara bersamaan.

Tabel 4. Pengujian Download Data

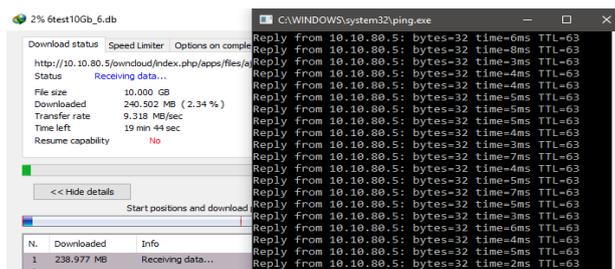
Pengujian	Jumlah User	Kecepatan Download
1	1 User	+ 10.209KB/s
2	2 User	+ 5.210KB/s
3	3 User	+ 3.120 KB/s
4	4 User	+ 2.120KB/s
5	5 User	+ 1.812 KB/s

Tabel 4 adalah pengujian *download* data 10 GB dengan variasi *user* dari 1 *user* sampai 5 *user*. Grafik pada Gambar 19 pada pengujian *download* ini menggunakan IDM agar *speed download* yang diperoleh lebih maksimal, dengan pengujian secara bertahap. serta didapatkan *range download* pada saat melakukan proses *download*.



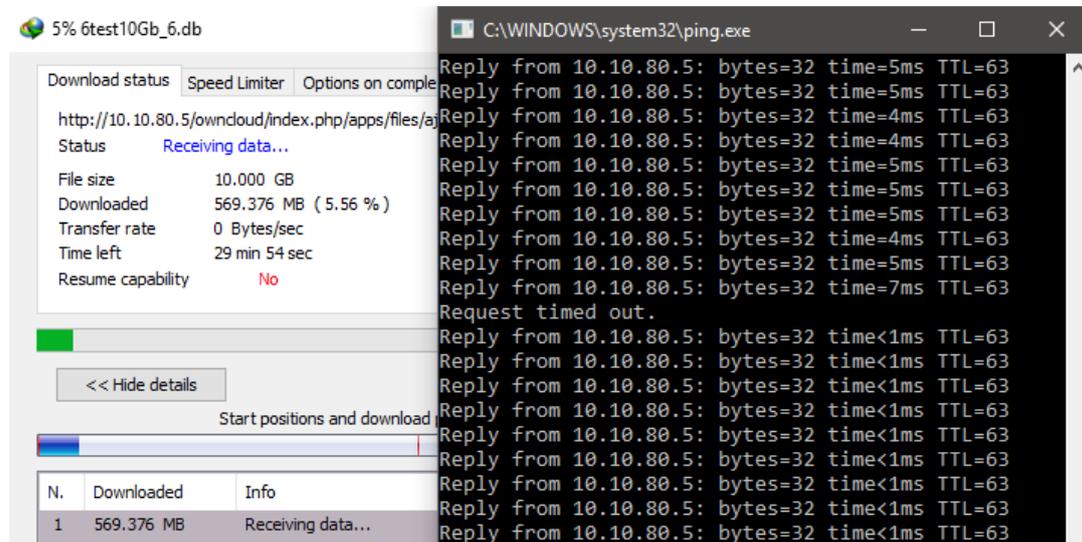
Gambar 19. Grafik Pengujian Variasi Download

Dalam setiap sistem pasti memiliki beberapa kelemahan, pada sistem yang di buat pada penelitian ini memiliki kelemahan saat *download* berlangsung. Gambar 20 merupakan pengujian kelemahan sistem yang di buat.



Gambar 20. Download IDM Master UP

Gambar 20 adalah pengujian dengan skenario ketika *server master UP* dan dilakukan *download data* lewat *IP virtual*.



Gambar 21. Download IDM Master Down

Gambar 21 adalah pengujian dengan skenario saat melakukan proses *download* pada saat *server master* mengalami kegagalan, dan dilakukan *failover* ke *server slave*. Proses *failover* bisa dilihat pada saat *ping* ketika mengalami *request time out* dan ping turun menjadi 1ms.

4. Kesimpulan

Dari penelitian yang telah dilakukan ada beberapa kesimpulan yang dapat diambil dari seluruh percobaan dan pengujian pada sistem yang telah dibangun yang berjudul "Implementasi High Availability Cloud Storage Dengan Metode Replikasi Dan Failover Pada Laboratorium Teknik Informatika", dapat disimpulkan sebagai berikut:

1. Penggunaan metode failover efektif diterapkan pada infrastruktur server private cloud storage terbukti berhasil diterapkan dengan baik saat menangani kegagalan sistem, dibuktikan pada saat server master mengalami kegagalan dan client atau pengguna layanan masih bisa mengakses melalui server slave, dan ketika server master aktif kembali sistem berhasil mengembalikan ke server master.
2. Penerapan sistem Owncloud pada server private cloud storage berjalan dengan baik saat melayani dan membuat account dari client, kemudian client berhasil melakukan upload data juga download data.
3. Pengujian failover dan failback dengan di berikan beban pada server master berjalan dengan baik pada proses perindahan.
4. Penerapan replikasi pada private cloud storage Owncloud menggunakan mysql tereplikasi dengan baik, replikasi data menggunakan Rsync memiliki transfer rate cukup kecil yakni 1GB /13 detik dan 10GB/3:44 menit.
5. Pengujian download file pada IP virtual dengan skenario mematikan server master mendapatkan kendala, karena ketika server master mati proses download akan berhenti walaupun sudah ada perpindahan failover.

5. Saran

Adapun saran dari penulis yang dapat di gunakan untuk pengembangan system ini untuk menjadi lebih baik, antara lain:

1. Perlu dibuat load balancer pada cloud storage dengan tujuan memaksimalkan akses upload dan download dari user atau pengguna layanan
2. Perlu adanya tambahan sistem keamanan dengan tujuan mengamankan segala transaksi atau pengiriman file
3. Penambahan sitem monitoring kecepatan, data yang di pakai atau yang di gunakan dan sisa ruang yang bisa di pakai.

Referensi

- [1] T. A. Nugroho, "Perancangan Private Cloud Storage Menggunakan ownCloud (Studi Kasus di Program Studi Magister Ilmu Lingkungan Universitas Sebelas Maret) Perancangan Private Cloud Storage Menggunakan ownCloud (Studi Kasus di Program Studi Magister Ilmu Lingkungan Unive," 2014.
- [2] T. R. Febriani, "Implementasi Dan Analisa Sistem Failover Virtual Computer Cluster," 2011.
- [3] C. Engelmann, "Active/active replication for highly available HPC system services," *Proceedings - First International Conference on Availability, Reliability and Security, ARES 2006*, vol. 2006, no. 8, pp. 639-645, 2006.
- [4] E. S. Purwanto, "Perbandingan Strategi Replikasi Pada Sistem Basis Data Terdistribusi," *Jurnal Informatika*, pp. 1-8, 2012.
- [5] S. M. Parasian D. P. Silitonga, "Replikasi Basis Data Pada Sistem Pengolahan Data Akademik," *Jurnal TIME* , Vol. III No 1 : 32-36, 2014 ISSN : 2337 – 3601, vol. III, no. 1, pp. 32-36, 2014.
- [6] N. Purnomo, "Pemanfaatan Failover Cluster Server Guna Recovery Sistem Pada Pt.Lintas Data Prima," *Naskah Publikasi diajukan*, 2012.

