

PERANCANGAN DAN IMPLEMENTASI REPLIKASI SERVER MENGUNAKAN METODE FAILOVER CLUSTER

Hoga Prasetya¹, Erna Kumalasari², Joko Triyono³

Program Studi Teknik Informatika, Fakultas Teknologi Industri
Institut Sains & Teknologi AKPRIND Yogyakarta

Email : ¹ haracopo@gmail.com, ²ernakumala@akprind.ac.id, ³jack@akprind.ac.id

ABSTRACT

The need for practicum information service in campus laboratory Institut Akprind Yogyakarta is very important, therefore it needs a system that can manage the information data practicum well. The service process on web applications should not stop / interrupt due to the system.

Database replication is a technique for copying and distributing data and database objects from one database to another and performing synchronization between databases so that data consistency can be guaranteed. The database system is required to be always ready to provide data with increasingly frequent quantities and data types that are increasingly large and varied.

By utilizing the database replication system using failover cluster, the performance of the database system can be enhanced by utilizing each client as master and as a slave at once, so that if one server is interrupted it will be available other server acting as new server so data availability can be guaranteed.

Keywords: Replication, Failover, Cluster, Server

INTISARI

Kebutuhan layanan informasi praktikum di laboratorium kampus Institut Akprind Yogyakarta sangat penting, oleh karena itu dibutuhkan sebuah sistem yang dapat mengelola data informasi praktikum dengan baik. Proses layanan pada aplikasi web tidak seharusnya berhenti / terganggu karena sistem.

Replikasi database merupakan suatu teknik untuk melakukan copy dan pendistribusian data serta objek-objek database dari satu database ke database lain dan melaksanakan sinkronisasi antara database sehingga konsistensi data dapat terjamin. Sistem database dituntut agar selalu siap untuk menyediakan data dengan kuantitas yang makin sering dan jenis data yang semakin besar dan bervariasi.

Dengan memanfaatkan sistem replikasi database menggunakan failover cluster, maka performa sistem database dapat ditingkatkan dengan memanfaatkan setiap klien sebagai master dan sebagai slave sekaligus, sehingga jika satu server mengalami gangguan maka akan tersedia server lainnya yang bertindak sebagai server baru sehingga ketersediaan data dapat terjamin.

Kata Kunci: Replikasi, Failover, Cluster, Server

PENDAHULUAN

Server mempunyai peran yang sangat penting dalam lalu-lintas suatu jaringan, karena serverlah yang mengatur besar kecilnya jalur tempat para client dalam mengakses server tersebut guna mendapatkan informasi. Apabila sebuah server diakses oleh banyak client, tentunya akan mempunyai beban yang sangat berat, sedangkan kemampuan dari server tersebut terbatas. Meningkatnya kebutuhan akan informasi menuntut akses yang cepat untuk mendapatkan informasi-informasi terkini, salah satunya yang paling dominan mempengaruhi adalah kecepatan akses pada saat suatu data diinputkan ke dalam alamat situs web.

Menurut (Rahardiantoro 2008) Replikasi adalah sebagai suatu proses mengcopy atau mentransfer data dari suatu database ke database lain yang tersimpan pada komputer berbeda. Replikasi data dapat meningkatkan kinerja dan ketersediaan aplikasi, karena adanya berbagai pilihan alternatif terhadap akses data yang disediakan. Sebagai contoh, sebuah aplikasi mungkin biasanya mengakses data secara lokal dari pada melalui server yang lokasinya jauh dengan tujuan untuk meminimalkan lalu lintas jaringan dan mencapai performa yang maksimal. Jika server lokal mengalami kegagalan, aplikasi tersebut masih dapat terus berfungsi, karena server yang lainnya memiliki replikasi data

dari server lokal. Dengan konsep seperti ini, ketersediaan terhadap aplikasi maupun data yang dibutuhkan dapat tercapai. Fungsi utama failover clustering (Hirt, 2009) bertujuan untuk membantu menjaga akses client ke aplikasi dan sumber daya server, bahkan ketika terjadi kegagalan software, ataupun kegagalan fungsi server yang mengakibatkan server berhenti bekerja. Teknologi ini diharapkan dapat menjadi solusi dalam mengatasi kegagalan server ketika terjadi gangguan ataupun perawatan (maintenance) dan mengimplementasikan suatu sistem failover virtual computer cluster untuk mengatasi kegagalan server.

TINJAUAN PUSTAKA

Penelitian ini menggunakan referensi hasil-hasil penelitian sebelumnya. Noviyanto (2015) telah melakukan perancangan dan implementasi *load balancing reserve proxy* menggunakan *haproxy* pada aplikasi *web*. Penelitian tersebut menjelaskan pemecahan masalah pembagian beban *server* menggunakan metode *load balancing* dan *failover*. Penelitian tersebut tidak menjelaskan tentang cara konfigurasi *failover* secara lengkap. Referensi selanjutnya adalah referensi penelitian dari, Nanang (2012) telah melakukan analisis pemanfaatan *failover cluster*. Penelitian tersebut menjelaskan kegunaan *recovery* sistem pada sistem *backup server*. Dari penelitian tersebut penelitian dapat dikembangkan lagi dan diimplementasikan juga pada *virtual server*. Selanjutnya adalah referensi dari penelitian menurut Feri,(2012) yaitu membandingkan efektivitas sinkronisasi antara dua database dengan menggunakan teknik multi master, yaitu replikasi 2 arah. Penelitian tersebut mempunyai kekurangan, dimana replikasi tersebut hanya dapat berjalan diaplikasi *mysql*. Referensi terakhir adalah referensi dari Andi (2013) yang telah melakukan tentang perancangan dan implementasi jaringan sebelumnya, maka perancangan infrastruktur dan implementasi jaringan metode *load balancing* dan *failover router* ini, yang akan diimplementasikan pada jaringan komputer untuk menggabungkan 2 line koneksi Speedy. Penelitian tersebut juga tidak menjelaskan secara lengkap cara kerja *failover* yang akan disimulasikan. Berdasarkan referensi penulisan-penulisan tentang *failover cluster* dan replikasi data sebelumnya, maka dalam penelitian ini dirancang dan dibuat simulasi metode baru mengenai Replikasi *Server* menggunakan *Failover Cluster*.

Landasan teori yang digunakan dalam penelitian ini adalah penjelasan teori dari buku yang berhubungan dengan penelitian ini, diantaranya:

1. Pengertian Replikasi

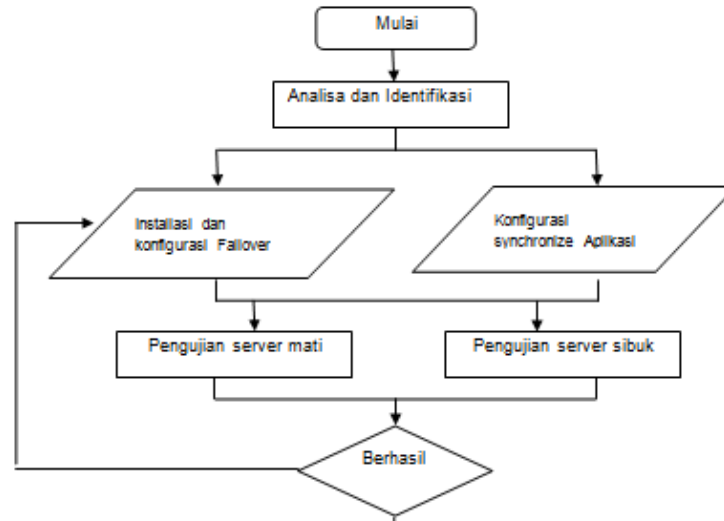
Replikasi adalah sebagai suatu proses mengcopy atau mentransfer data dari suatu database ke database lain yang tersimpan pada komputer berbeda. Replikasi dapat difahami sebagai teknik meng-copy database dan pengelolaan obyek-obyek database dalam suatu jaringan komputer yang dapat membentuk suatu sistem database terdistribusi untuk menjaga konsistensi data secara otomatis. Sistem client-server mempunyai satu proses klien atau lebih dan satu proses server atau lebih, dan proses klien dapat mengirim sebuah query pada proses server manapun. Client bertanggung jawab terhadap proses user inter face dan server mengatur data dan mengeksekusi transaksi. Jadi, proses klien dapat menjalankan komputer personal dan mengirim query pada server yang berjalan pada kerangka utama. Triyono (2012)

2. Failover cluster

failover dalam istilah computer internetworking adalah kemampuan sebuah sistem untuk dapat berpindah secara manual maupun otomatis jika salah satu sistem mengalami kegagalan sehingga menjadi backup untuk sistem yang mengalami kegagalan. Dengan menerapkan teknik data replication dan redundancy pada jaringan cluster, maka dapat dipastikan untuk konsistensi, ketersediaan, dan aksesibilitas data di antara masing-masing node dapat terjaga dengan baik. Dalam kasus failover, node yang menggantikan tugas dari node yang down akan tetap memiliki data yang sama seperti node yang down tersebut. (Bell Charles, dkk, 2014)

Dalam penelitian ini, langkah yang dilakukan dalam implementasi sinkronise database dengan metode *failover computer cluster* adalah sebagai berikut:

1. Mengidentifikasi kebutuhan data yang ingin digunakan dalam penelitian
2. Menginstall serta konfigurasi yang dibutuhkan oleh *failover* komputer *cluster*.
3. Melakukan konfigurasi pada aplikasi yang akan digunakan agar dapat *synchronize* dengan *server* yang akan digunakan.
4. Melakukan pengujian terhadap jaringan untuk mengetahui keberhasilan pada sistem *failover*nya.
5. Jika sistem *failover* belum berfungsi dengan baik, perlu dicermati tahapan dalam konfigurasi.
6. Melakukan analisa beban *traffic* yang terjadi pada *server* dan dalam *database*
7. Melaporkan hasil pengujian dan analisis mengenai *sinkronise database* dengan metode *failover* komputer *cluster* yang diteliti.

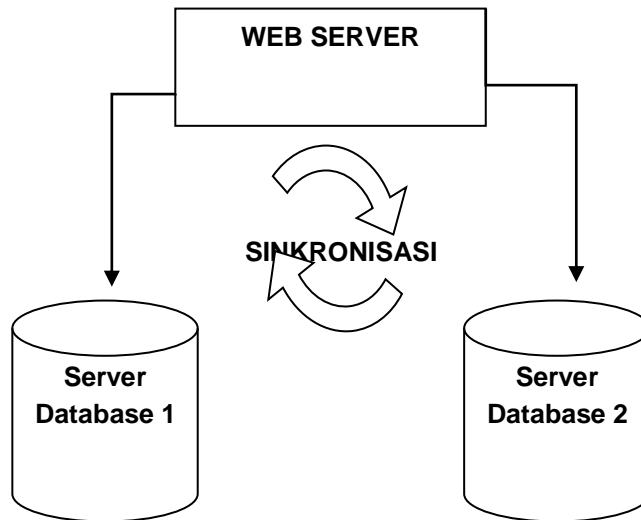


Gambar 1 Diagram Alur Penelitian

Gambar 1 Diagram Alur Penelitian

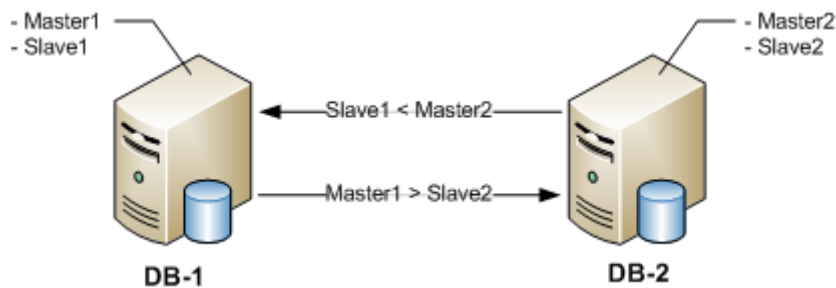
PEMBAHASAN

Hasil penelitian ini meliputi rancangan jaringan server, konfigurasi jaringan komputer antar server, implementasi replikasi, pengujian data, dan monitoring. Setelah tahap analisis sistem selesai dilakukan, maka selanjutnya adalah tahapan perancangan yang dibuat meliputi tahapan-tahapan sistem yang akan dibangun meliputi arsitektur sistem pada gambar 1



Gambar 2 Rancangan Jaringan

Pada gambar 2 merupakan rancangan arsitektur sistem yang akan dibangun untuk replikasi database. Terdapat beberapa komponen yang di perlukan untuk replikasi dapat berjalan, master dan slave. Dimana di dalam gambar 2 di jelaskan bahwa jika sebuah master mengalami kerusakan atau tidak dapat diakses maka akan dapat langsung dialihkan kepada master baru yang merupakan slave sebelumnya. Perencanaan IP address dalam sistem yang akan dibangun menggunakan IP DHCP (dynamic host konfigurasi protocol), sehingga IP yang diberikan secara acak.



Gambar 3 Skema umum master slave

Setelah melakukan installasi pada server yang digunakan, langkah berikutnya adalah melakukan installasi aplikasi dan konfigurasi database server. Berikut adalah langkah yang dilakukan untuk install aplikasi dan konfigurasi yang akan dipakai pada database server yang akan digunakan

Setting Database Server

Seperti installasi pada web server umumnya, langkah pertama yang dilakukan untuk setting konfigurasi database-server yaitu adalah setting interface pada database server. Berikut adalah isi konfigurasi interface pada database server :

Konfigurasi server

Langkah pertama mengganti konfigurasi di my.cnf , masuk menggunakan syntax berikut :
`sudo nano /etc/mysql/my.cnf`

akan tampil seperti berikut ini :

```
server-id      = 1
log_bin       = /var/log/mysql/mysql-bin.log
binlog_do_db  = labora
# bind-address = 127.0.0.1
```

server id adalah untuk mengidentifikasi nama dari server. log_bin adalah lokasi terakhir dimana data disimpan. Binlog_do_db adalah nama database yang dibuat, bin-address diberi tanda # karena tidak menggunakan localhost.
Langkah selanjutnya restart mysql

```
sudo /etc/init.d/mysql restart
```

Login kedalam mysql

```
mysql -u root -p
```

membuat user replikasi di mysql

```
create user 'replicator'@'%' identified by 'replicator';
```

mengganti permission user replicator

```
grant replication slave on *.* to 'replicator'@'%';
```

melihat status master

```
show master status;
```

mematikan service di slave

```
stop slave;
```

konfigurasi slave ke master, jangan lupa samakan master_log_file, dan master_log_pos pada master

```
CHANGE MASTER TO MASTER_HOST = '190.190.190.6', MASTER_USER = 'replicator', MASTER_PASSWORD = 'replicator', MASTER_LOG_FILE = 'mysql-bin.000006', MASTER_LOG_POS = 107;
```

Implementasi Failover

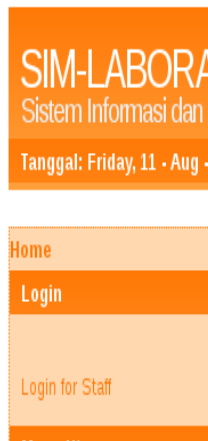
Tahap selanjutnya adalah melakukan pengujian terhadap implementasi failover pada jaringan yang telah dibangun. pengujian dilakukan dengan menggunakan laptop sebagai client yang terhubung dengan laptop yang dijadikan server. tujuan dari pengujian ini adalah untuk membuktikan apakah client dapat menggunakan aplikasi dan tetap dapat menjalankan aplikasi jika terdapat gangguan pada master server dengan fungsi failover yang telah dibangun.

pengujian dilakukan dalam beberapa kondisi sebagai berikut.

1. Server 1 dan server 2 dalam keadaan hidup dan kondisi link terhubung kedalam jaringan (kondisi 1).
 2. Server 1 dalam keadaan tidak bekerja (mati, link putus ke jaringan), sementara server 2 dalam kondisi hidup dan link terhubung ke jaringan (kondisi 2).
- berdasarkan pengujian (kondisi 1 dan kondisi 2) dapat disimpulkan bahwa proses failover cluster sudah berjalan, dimana server 2 dapat mengambil alih tugas server 1 begitu pula sebaliknya.

Pada gambar 3 adalah gambar dari implementasi failover yang ditampilkan didalam SIM-LABORATORY IST AKPRIND. Terlihat pada status diatas table daftar asisten, koneksi pada server 190.190.190.5 telah mati/down dan kemudian dialihkan ke server

190.190.190.6



fresh off | Refresh 5 | Refresh 30 | Expert

Slave Status

Slave_IO_State	Reconnecting after a failed master event read
Master_Host	190.190.190.5
Master_User	replicator
Master_Port	3306
Connect_Retry	60
Master_Log_File	mysql-bin.000007
Read_Master_Log_Pos	107
Relay_Log_File	mysql-relay-bin.000008
Relay_Log_Pos	253
Relay_Master_Log_File	mysql-bin.000007
Slave_IO_Running	Connecting
Slave_SQL_Running	Yes
Replicate_Do_DB	
Replicate_Ignore_DB	
Replicate_Do_Table	
Replicate_Ignore_Table	
Replicate_Wild_Do_Table	
Replicate_Wild_Ignore_Table	
Last_Errno	0
Last_Error	
Skip_Counter	0
Exec_Master_Log_Pos	107
Relay_Log_Space	556
Until_Condition	None
Until_Log_File	
Until_Log_Pos	0
Master_SSL_Allowed	No
Master_SSL_CA_File	
Master_SSL_CA_Path	
Master_SSL_Cert	
Master_SSL_Cipher	
Master_SSL_Key	
Seconds_Behind_Master	

Koneksi

koneksi gambar dengan

Gambar 4
Keterangan
Pada Aplikasi

Untuk menghasilkan failover pada IV.3 dilakukan

cara mengubah file koneksi yang ada dalam aplikasi SIM-LABORATORY IST-AKPRIND. Berikut ini isi dari file koneksi sebelum di ubah.

```
<?php
//buka file
$con=mysql_connect("localhost","labora","labora");
$db=mysql_select_db("labora");
?>
```

Dan setelah diubah dan ditambah dengan keterangan kedalam file koneksi dari aplikasi tersebut

```
<?php
$con = mysql_connect('190.190.190.5','hoga','hoga','labora');
if ($con == false){
    echo "Koneksi pada server 190.190.190.5 telah DOWN <br/>";
    $con = mysql_connect('190.190.190.6','hoga','hoga','labora');
    if ($con== false){
        echo "gagal localhost <br/>";
    }else{
        echo "Sementara koneksi dialihkan pada 190.190.190.6 <br/><br/>";
    }
}
}
echo "Koneksi pada server 190.190.190.5 telah UP <br/><br/><br/>";
}

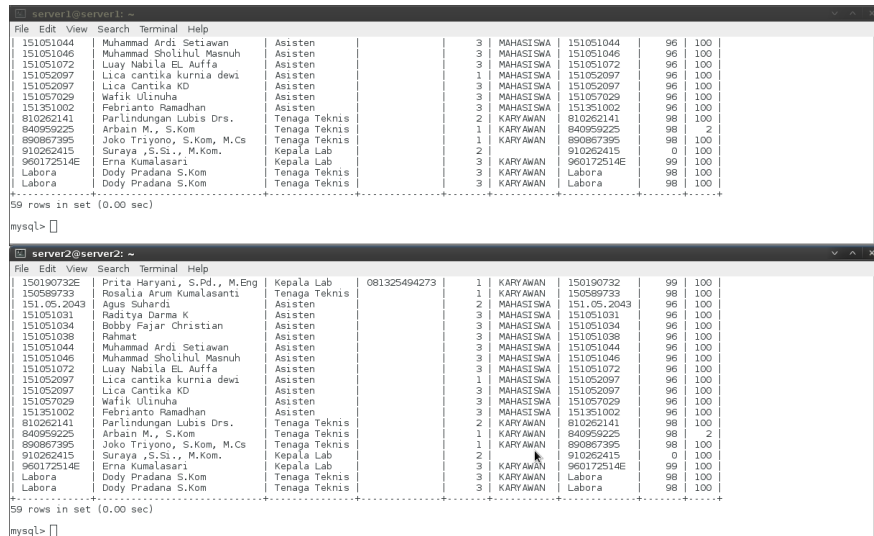
$db = mysql_select_db("labora");

?>
```

Gambar 4 merupakan *monitoring* jika salah satu *server* dimatikan. Table *slave status* pada *running io slave* akan menyala merah. Itu dikarenakan salah satu *server* mati dan tidak dapat sinkron dengan menunjukkan status *connecting*.

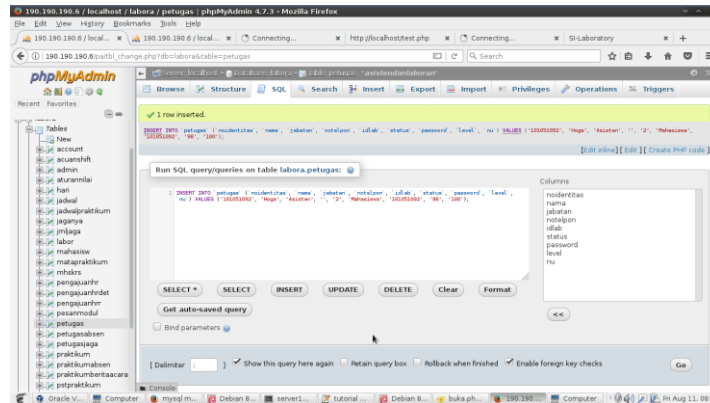
Gambar 5 Monitoring Status dengan kondisi Salah satu server mati

Pengujian Data, data yang digunakan dalam penelitian ini berupa data pendaftaran praktikum mahasiswa IST Akprind Yogyakarta diambil dari laborat.akprind.ac.id. Data pengujian diambil dari tabel petugas dalam database laborat. Pada gambar 5 melihat isi data dari tabel petugas menggunakan terminal dengan meremote server 1 dan server 2.



Gambar 6 Isi dari Table Petugas menggunakan terminal

Dalam pengujian data penelitin ini menggunakan phpmysqladmin untuk tindakan create, update dan delete data kedalam database yang ada. Dalam gambar 6 terlihat penginputan data kedalam table petugas sebanyak 1 buah data dan dapat diterima database dengan sukses,



Gambar 7 Input data kedalam database

Selanjutnya melakukan pengujian terhadap waktu yang dibutuhkan untuk melakukan jenis tindakan seperti select, update, delet, dan insert.

Table 1 Hasil Pengujian Replikasi Pada Server Database 1

No	Jenis Eksekusi	Jumlah Eksekusi	Rata – rata waktu
1	Select	10	0.26 detik
2	Update	10	0.296 detik
3	Delete	10	0.059 detik
4	Insert	10	0.65 detik

Pada tabel 1 adalah hasil dari pengujian dengan jenis perintah select, update, delete dan insert kedalam server 1 dengan jumlah perintah 10 kali pengujian. Rata – rata waktu yang dihasilkan dari pengujian tersebut kurang dari 1 detik .

Tabel 2 Hasil Pengujian Replikasi Pada Server Database 2

No	Jenis Eksekusi	Jumlah Eksekusi	Rata – rata waktu
1	Select	10	0.36 detik
2	Update	10	0.396 detik
3	Delete	10	0.079 detik
4	Insert	10	0.75 detik

Pada tabel 2 hasil pengujian dari server 2 dengan jenis perintah dan jumlah pengujian yang sama dengan server 1. Dan menghasilkan waktu rata – rata 0.317 detik dari server 2 sedangkan pada server 1 menghasilkan waktu rata – rata 0.253 detik

Table 3 Hasil Pengujian Perubahan Log Bin

Pengujian	Time	Host	Status	File	Position	Database
	10 Agustus 2017	190.190.190.5	Master	Bin.000009	101	labora
		190.190.190.6	Slave	Bin.000028	395	labora
		190.190.190.5	Master	Bin.000009	101	labora
		190.190.190.6	Slave	Bin.000028	395	labora
1	11 Agustus 2017	190.190.190.5	Master	Bin.000009	107	labora
		190.190.190.6	Slave	Bin.000028	417	labora
		190.190.190.5	Slave	Bin.000009	107	labora
		190.190.190.6	Master	Bin.000028	417	labora
2	12 Agustus 2017	190.190.190.5	Master	Bin.000009	1525	labora
		190.190.190.6	Slave	Bin.000028	461	labora
		190.190.190.5	Slave	Bin.000009	1525	labora
		190.190.190.6	Master	Bin.000028	461	labora

Pada tabel 3 merupakan hasil tampilan perubahan log bin dan posisi penyimpanan data pada mysql. Pengujian dilakukan dalam waktu yang berbeda yaitu pengujian pertama pada tanggal 11 Agustus 2017 dilanjutkan pada 12 Agustus 2017.

KESIMPULAN

Dari pengujian sistem yang dibuat pada tugas akhir ini dapat ditarik beberapa kesimpulan tentang pembangunan sistem replikasi database dengan menggunakan metode failover cluster yaitu :

1. Telah berhasil dibangun sebuah sistem replikasi database dengan menggunakan metode master-slave dengan mysql pada sistem operasi linux Debian versi 8
2. Sistem replikasi berjalan dengan baik pada sistem yang memiliki 2 buah server database yang masing-masing bertindak sebagai master server dan slave server.
3. Hasil pengujian dengan jenis perintah dan jumlah pengujian yang sama menghasilkan waktu rata – rata 0.317 detik dari server 2 sedangkan pada server 1 menghasilkan waktu rata – rata 0.253 detik
4. Karena masih menggunakan trigger based replication sehingga membutuhkan resource tambahan yang kurang lebih setara dengan resource untuk transkripsinya sendiri sehingga membutuhkan resource lebih yang kurang efisien.

Saran

Perancangan dari replikasi server dengan failover cluster ini sangat jauh dari kesempurnaan dan dibutuhkan lagi beberapa pengembangan untuk penelitian lebih lanjut, diantaranya sebagai berikut:

1. Membuat sistem baru dengan metode lain, kondisi terjadi kepadatan request kedalam server database maka secara otomatis berpindah ke database lain
2. Merancang sistem dengan kondisi terjadi kegagalan (false) dalam penginputan data kedalam database server maka akan secara otomatis berpindah ke database server lain

DAFTAR PUSTAKA

- Bell, C., dkk. 2014. *MySQL High Availability*. Edisi Kedua. Sebastopol:O'Reilly Media.
- Buyya, Rajkumar, dkk. 2011. *Cloud Computing: Principles and Paradigms*. Hoboken:Wiley.
- Noviyanto, A.B., 2015. *Perancangan Dan Implementasi Load Balancing Reserve Proxy Menggunakan Haproxy Pada Aplikasi Web*. Jurusan Teknik Informatika, FTI, IST AKPRIND, Yogyakarta
- Purnomo, N., 2012. *Pemanfaatan Failover Cluster Server Guna Recovery Sistem Pada Pt.Lintas Data Prima.*, Jurusan Teknik Informatika, FTI, IST AKPRIND, Yogyakarta
- Purnama, F., 2012. *Perbandingan Efektifitas Sinkronisasi Instance Dua Database Menggunakan Teknik Replikasi Dan Program Aplikasi (Menggunakan Mysql)*. Skripsi, Jurusan Teknik Informatika, FTI, IST AKPRIND, Yogyakarta
- Alhasan , A.J., 2013. *Perancangan Dan Implementasi Jaringan Komputer Perumahan Pasir Ridge Dengan Metode Loadbalancing Dan Failover Pt Chevron Indonesia Company*. Yogyakarta.
- Triyono, J., 2012. *Repikasi Untuk Meningkatkan Kinerja Dan Ketersediaan Data (Study Kasus Sistem Informasi Akademik)*. Yogyakarta. Technoscientia