

PENGEMBANGAN SISTEM REPLIKASI DAN REDUNDANSI UNTUK MENINGKATKAN KEHANDALAN BASISDATA MYSQL

Dolly Simon Kristian¹⁾, Adian Fatchur Rochim²⁾, Eko Didik Widiyanto²⁾
Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro,
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia
email : dolly23.dk@gmail.com

Abstract—, *With the development of information technology, humans make it easy to complete every duty. Any information about the work carried out to be very valuable, therefore the information should be stored properly, by organizing a reliable database system when performing data storage on the server information. Build a design application virtualization master slave database servers that are connected with the management node using the virtual application to get the test results replication system performance and redundancy in the design of the cluster system.*

Methodology of this research include the study of literature, collecting data by interview, observation, literature studies, system design, and testing of the system. In a literature study on the use of research methods to study the literature books, records that can be used as a support in the research. The design of this thesis using MySQL Cluster system with Ndbcluster engine. Last is testing this system on its performance on the server failure or failures occur and reliability in performance.

The results obtained are when there is a failure on the primary server, it will be immediately replaced by another server is a slave. And the replication of data between the main server and slave.

Keywords : *database, replication, MySQL Cluster, high availability, high scalability*

I. PENDAHULUAN

Basisdata (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basisdata tersebut. Konsep dasar dari basisdata adalah kumpulan dari uraian data. Basisdata memiliki penjelasan terstruktur dari jenis data yang tersimpan di dalamnya, atau disebut skema. Skema menggambarkan obyek yang mewakilkan suatu basisdata, serta hubungan di antara obyek tersebut.^[1]

Replikasi memiliki pengertian sebagai suatu proses perpindahan data dari suatu basisdata (*master*) ke basisdata (*slave*) lain yang tersimpan pada komputer yang berbeda. Untuk melakukan proses replikasi dibutuhkan dua buah komputer atau lebih yang berfungsi sebagai *server* dengan satu *server* sebagai *master* dan yang satu sebagai *slave*. Dengan replikasi, data dapat

didistribusikan ke lokasi yang berbeda dan pengguna yang jauh melalui LAN, WAN, *Dial-up Connection*, *wireless connections*, dan internet.

Tujuan penelitian ini adalah bertujuan untuk membangun sebuah rancangan virtualisasi aplikasi server master slave basisdata yang terhubung atau terkoneksi dengan manajemen node dan slave basisdata, menggunakan aplikasi VirtualBox untuk mendapatkan hasil uji performa sistem replikasi dan redundansi dalam rancangan sistem kluster.

Pembuatan basisdata kluster ini dilatarbelakangi oleh manfaat replikasi basisdata untuk menghindari kemungkinan tidak semua data tersalin karena saat proses penyalinan data manual dilakukan bisa saja terjadi perubahan data oleh klien. Apabila basisdata *master* mengalami kerusakan, basisdata bisa segera dialihkan ke basisdata *slave*. Replikasi *master-slave* berlangsung secara *realtime* dimana setiap perubahan pada data basisdata *master* akan otomatis merubah data pada basisdata *slave*.

II. METODE PENELITIAN

Tahapan-tahapan yang dilakukan dalam perancangan basisdata kluster ini adalah dengan menginstal virtualbox, kemudian membuat beberapa server pada virtualbox dengan Ubuntu Server 12.04 sebagai sistem operasinya, pembuatan mesin haproxy dengan menginstal haproxy dan apache, penginstalan wordpress dan lamp, penginstalan mysql cluster pada server node A dan node B, penginstalan httpperf sebagai aplikasi untuk pengujian ketahanan server.

Virtualisasi jaringan ini menggunakan aplikasi *VirtualBox*. *Oracle VM VirtualBox* adalah perangkat lunak virtualisasi yang dapat digunakan untuk mengeksekusi sistem operasi tambahan di dalam sistem operasi utama. Melakukan penginstalan virtualbox sebagai langkah pertamayang dilakukan.

Pembuatan server pada mesin virtual dengan menggunakan Ubuntu Server 12.04 sebagai sistem operasi tiap-tiap server. *Server Host* menggunakan *ubuntu server* sebagai sistem operasi. *Ubuntu server* versi saat ini adalah *Ubuntu 12.04 Long Term Support (LTS) release*, yang akan didukung hingga 2017.

Pembuatan mesin HAproxy dengan melakukan penginstalan Haproxy 1.4 terhadap salah satu server yang telah dibuat.^[10]

Pembuatan server aplikasi dengan penginstalan wordpress sebagai media untuk menguji apakah basisdata kluster yang dirancang dapat digunakan oleh aplikasi CMS. *WordPress* adalah sebuah aplikasi sumber terbuka (*open source*) yang digunakan sebagai CMS (*Content Management System*) pada situs web. Wordpress menggunakan bahasa php dan menggunakan MySQL sebagai basisdatanya.^[4]

Penginstalan MySQL pada server untuk menjadikan server tersebut sebagai Node A dan Node B yang berkerja dengan mesin kluster. *MySQL* merupakan perangkat lunak untuk manajemen basisdata dan sebagai tempat penyimpanan data pada server aplikasi. Pada sistem di sini menggunakan *mysql-kluster-gpl-7.3.7-linux-glibc2.5-x86_64.tar* yang basisdatanya direplikasi satu arah *master to slave*.^[5]

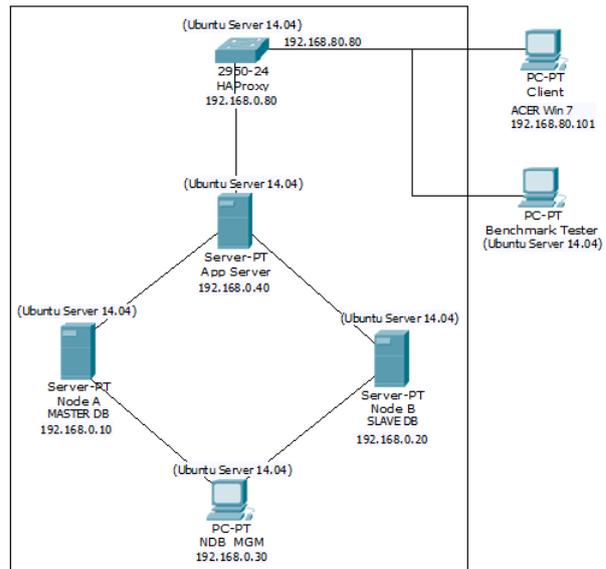
Penginstalan *LAMP* pada server aplikasi untuk mendukung kinerja server dalam menampilkan aplikasi yang telah tertanam dalam server. *LAMP* adalah sebuah aplikasi *web server Apache* yang di dalamnya sudah tersedia *database server MySQL* dan *support php programming*. Komponen *LAMP* yaitu *Apache*, *PHP*, *MySQL* dan *phpMyAdmin*. *LAMP* merupakan aplikasi yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support (PHP 4 dan PHP 5)* dan beberapa modul lainnya.

Penginstalan *httperf* sebagai aplikasi yang digunakan untuk mengukur ketahanan suatu server. *Httpperf* adalah program untuk mengukur kinerja atau performansi dari web server yang dibuat oleh David Mosberger dari HP Labs. *Httpperf* menyediakan fitur yang fleksibel dalam pembuatan beban kerja sesuai dengan parameter yang diberikan padanya. *Httpperf* merupakan sebuah aplikasi untuk sistem operasi turunan UNIX. *Httpperf* dapat membangkitkan sejumlah beban paket dan mendukung HTTP/1.0 dan HTTP/1.1. *Httpperf* dalam mengukur performansi sebuah *web server* adalah dengan mengirimkan permintaan atau paket ke server dengan rata tertentu dan mengukur waktu balasan yang diterima.

III. PERANCANGAN DAN IMPLEMENTASI

Sistem dirancang dengan menggunakan topologi jaringan, pada gambar 1 dijelaskan hubungan unsur dasar penyusun jaringan.

Berdasarkan topologi jaringan pada gambar 1 menjelaskan bahwa terdapat satu penyeimbang beban dengan nama *HAProxy*, satu *server* untuk aplikasi diberi nama *ServerApp*, dua *server* basisdata dengan nama *DB Master* dan *DB Slave*, satu PC untuk monitoring dengan nama *NDB_MGM* dan dua klien diberi nama *client* dan *benchmark tester*.



Gambar 1 Topologi Sistem

Server basisdata kluster digunakan untuk menyimpan data klien atau data yang akan digunakan oleh wordpress seperti *user name* dan *password*. Server basisdata yang digunakan pada penelitian ini adalah *mysql-kluster-gpl-7.3.7-linux-glibc2.5-x86_64.tar* membuat replikasi satu arah, yaitu *master to slave*.^[5]

Konfigurasi basisdata yang dibuat dalam konfigurasi Node adalah dengan membuat file *my.cnf* pada Node sebagai berikut:

```
# nano /etc/my.cnf
```

Untuk tiap data node dan sql node yang kita buat, *my.cnf* harus terlihat dan berisi pengaturan seperti ini:

```
[client]
port = 3306
socket = /tmp/mysql.sock
[mysqld]
port = 3306
socket = /tmp/mysql.sock
skip-locking
ndbcluster # run NDB storage engine
ndb-connectstring=192.168.0.30
[mysql_cluster]
ndb-connectstring=192.168.0.30
```

Konfigurasi manajemen node adalah untuk membuat direktori yang mana file konfigurasinya dapat ditemukan dan kemudian dapat membuat file itu sendiri.

```
# mkdir /var/lib/mysql-kluster
# cd /var/lib/mysql-kluster
# nano config.ini
```

Untuk pengaturan yang kita buat, file *config.ini* harus dibuat seperti ini:

```
[ndbd default]
NoOfReplicas=2
DataMemory=80M
IndexMemory=18M
[tcp default]
[ndb_mgmd]
hostname=192.168.0.30 # Hostname or IP
address of MGM node
```

```

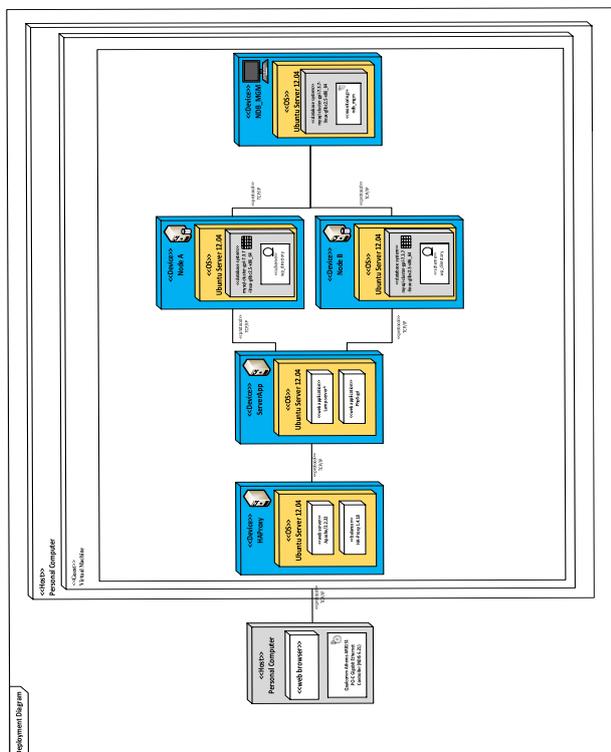
datadir=/var/lib/mysql-kluster #
Directory for MGM node log files
[ndbd]
hostname=192.168.0.10 # Hostname or IP
address
datadir=/usr/local/mysql/data #
Directory for this data node's data files
[ndbd]
hostname=192.168.0.20 # Hostname or IP
address
datadir=/usr/local/mysql/data #
Directory for this data node's data files
[mysqld]
hostname=192.168.0.10 # Hostname or IP
address
[mysqld]
hostname=192.168.0.20 # Hostname or IP
address

```

Perangkat lunak *Wordpress* dipasang dikedua server aplikasi Server App. Sebelum melakukan instalasi, terlebih dahulu membuat user dan basisdata pada mysql untuk data *Wordpress*. Terdapat pada `~/wordpress/wp-config`.

IV. PENGUJIAN DAN ANALISIS

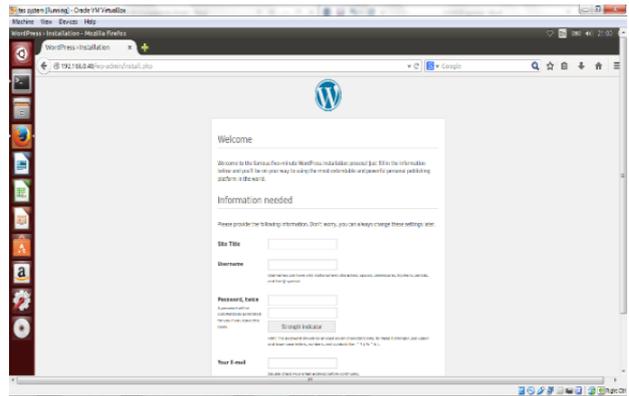
Pembuatan *deployment diagram* ini bertujuan untuk memudahkan kita dalam membaca topologi yang dibangun serta aplikasi apa saja yang terdapat pada sisi server, semua tergambar jelas dalam gambar 2.



Gambar 2 Deployment Diagram

Aplikasi *Wordpress* telah terpasang dan telah dilakukan konfigurasi yang diperlukan pada bab sebelumnya.

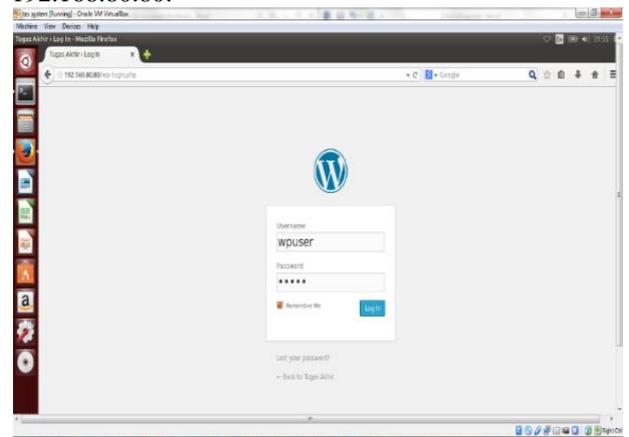
Tahapan selanjutnya adalah melakukan pengujian pada konfigurasi yang telah dibuat, apakah konfigurasi sudah benar atau masih ada *error* pada pelaksanaan pengujian *wordpress*. Pengujian dilakukan dengan cara klien mengakses ke *server* yang masih berada dalam *localhost* untuk mencoba apakah *wordpress* sudah terpasang dengan benar. *Wordpress* diakses oleh klien dengan menuliskan alamat IP *server* aplikasi pada browser, seperti `192.168.0.40/wp-admin`. Seperti yang di tunjukkan pada gambar 3.^[4]



Gambar 3 Tampilan 192.168.0.40/wp-admin

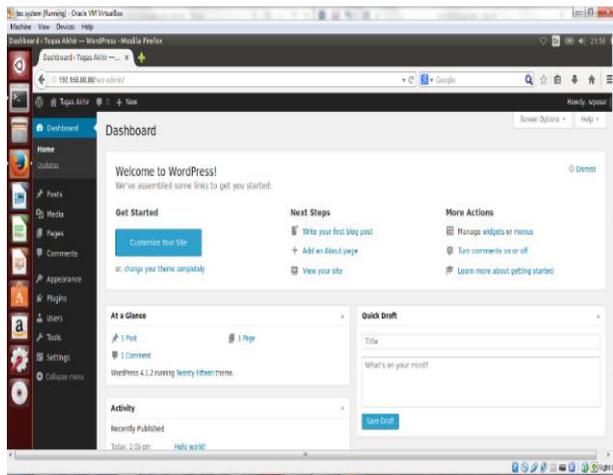
Wordpress telah diinstalasi tetapi data yang disimpan masih berada di *localhost*. Basisdata yang sudah terbentuk oleh *wordpress* dipindahkan ke *server* basisdata *MySQL* yang berada pada alamat `192.168.0.10` dan `192.168.0.20`.

Migrasi basisdata yang sudah dilakukan akan diuji melalui klien yang terhubung oleh *server* *HAProxy*, artinya klien tidak akan lagi melakukan uji coba permintaan melalui *server* aplikasi secara langsung tetapi melewati *server* *HAProxy* dahulu pada alamat `192.168.80.80`.



Gambar 4 Halaman Login

Pada gambar 4 ditunjukkan bahwa *wordpress* telah dapat diakses melalui ip dari *HAProxy* dengan permintaan `192.168.80.80/wp-admin` sehingga menampilkan halaman login pada *wordpress*.



Gambar 5 Halaman Setelah Berhasil Login

User berhasil login seperti yang di tunjukkan pada gambar 5. User berhasil login setelah memasukkan login pada alamat 192.168.80.80/wp-admin.

Pada penginstalan MySQL Cluster, berbeda dengan mysql server dan mysql client yang akan meminta pembuatan password untuk mengakses root. Pada pembuatannya hal tersebut tidak diminta oleh cluster.

Setelah node ndbd dijalankan keduadua node, maka teknologi MySQL Cluster sudah berjalan dengan lancar. Lakukan pengecekan pada manajemen client di NDB_MGM. Berikut adalah tampilannya.

```

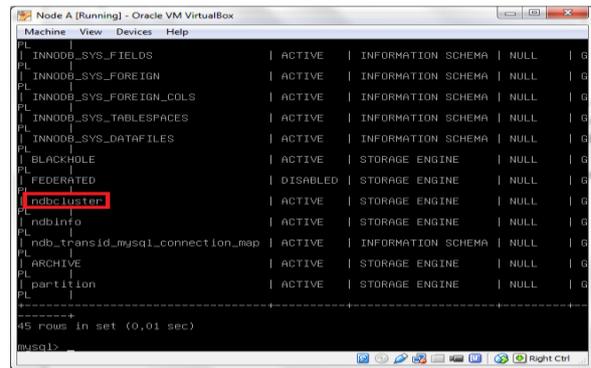
root@dolly:/home/dolly# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show

Connected to Management Server at:
localhost:1186
Kluster Configuration
-----
[ndbd(NDB)]      2 node(s)
id=2 @192.168.0.10 (mysql-5.6.21 ndb-
7.3.7, Nodegroup: 0, *)
id=3 @192.168.0.20 (mysql-5.6.21 ndb-
7.3.7, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.0.30 (mysql-5.6.21 ndb-
7.3.7)
[mysqld(API)]   2 node(s)
id=4 @192.168.0.10 (mysql-5.6.21 ndb-
7.3.7)
id=5 @192.168.0.20 (mysql-5.6.21 ndb-
7.3.7)

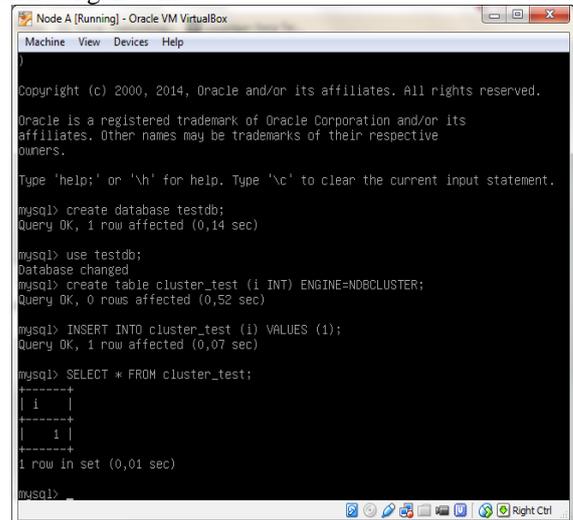
```

Masuk ke mysql untuk mengetahui apakah ndb cluster sudah aktif. Berikut cara mengetahui mesin ndb cluster telah aktif.



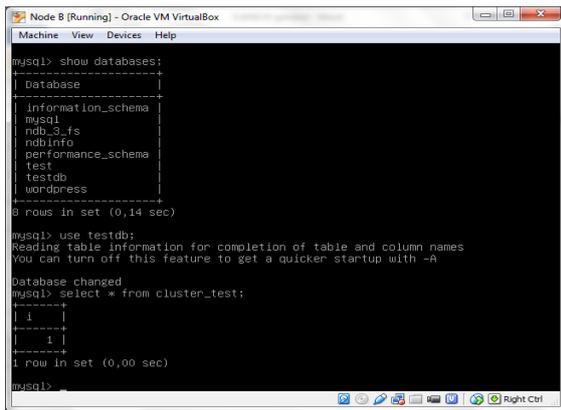
Gambar 6 NDB Kluster Telah Aktif

Pada gambar 6 dapat diketahui bahwa engine ndbcluster sudah aktif dengan berbagai keterangannya. Setelah engine ndbcluster sudah siap, lakukan pengujian yang akan mencoba apakah cluster sudah benar-benar berjalan dengan baik.



Gambar 7 Tampilan Pengujian Replikasi Node A

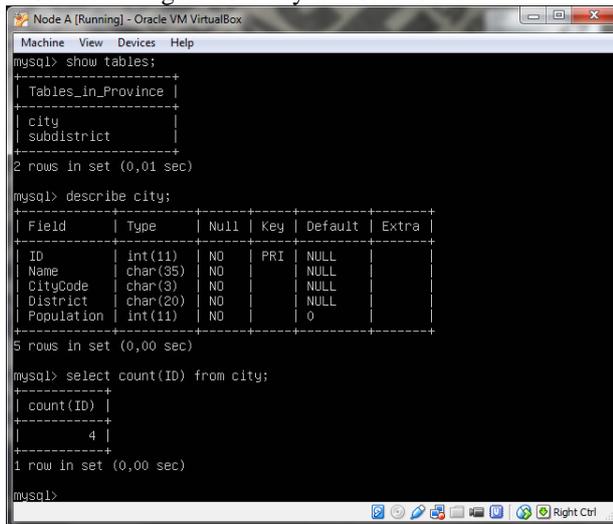
Pengujian yang dilakukan untuk melihat perubahan data pada basisdata cluster sudah berjalan dengan semestinya. Replikasi yang digunakan pada server basisdata menggunakan replikasi satu arah master to slave. Replikasi master to slave memungkinkan perubahan yang dilakukan pada Node A juga berdampak pada Node B secara otomatis dan sebaliknya. Pada gambar 7 di tunjukkan tampilan data pada Node A. Pada gambar 8 di tunjukkan tampilan data yang sama seperti yang terdapat pada Node A. Hal ini membuktikan bahwa replikasi pada ndbcluster ini telah berjalan dengan sempurna.



Gambar 8 Tampilan Test MySQL Cluster Node B

Jika replikasi pada Node A dan Node B sudah terjalin, maka teknologi pada MySQL Cluster sudah berjalan dengan baik.

Setelah dipastikan bahwa server MySQL Cluster sudah berjalan maka akan dicoba sifat MySQL Cluster yaitu *high availability*. Pertama-tama yang akan dibuat sebuah *database*. *Database* yang akan dibuat adalah *Province*. Dengan tabel *city* dan *subdistrict*.

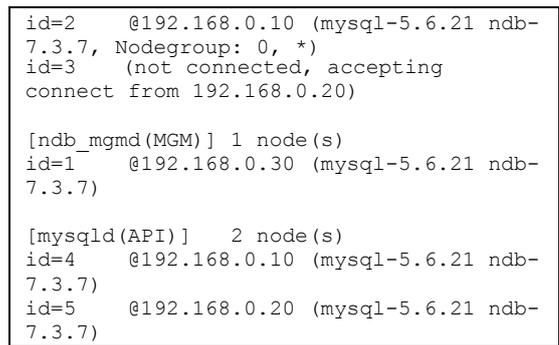
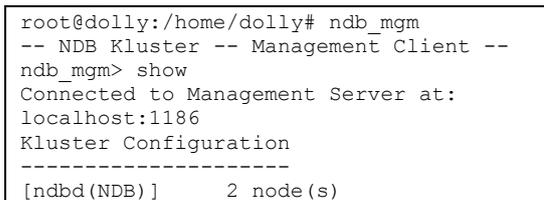


Gambar 9 Pengecekan Jumlah ID Pada Node A

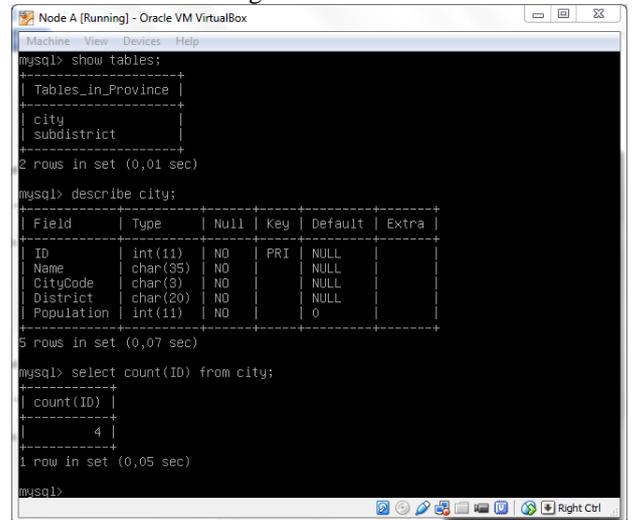
Dapat dilihat pada gambar 9 bahwa isi dari tabel *city* ada 4 id. Nilai id itu akan menjadi patokan apakah sifat *high availability* dapat berjalan dengan sempurna. Untuk itu kita harus mematikan node *ndbd* pada sisi *slave*.

```
[root@localhost ~]# killall ndbd
```

Bisa dilihat pada *ndb_mgm* server apakah node *ndbd* sudah tidak aktif.



Pada pemberitahuan tersebut dapat diketahui bahwa Node B sudah tidak aktif lagi.



Gambar 10 MySQL Cluster Memiliki Sifat *High Availability*

Pada gambar 10 dapat diketahui bahwa ID yang terdapat pada basisdata tetap berjumlah 4. Maka MySQL Cluster yang telah dibuat memiliki sifat *high availability*.

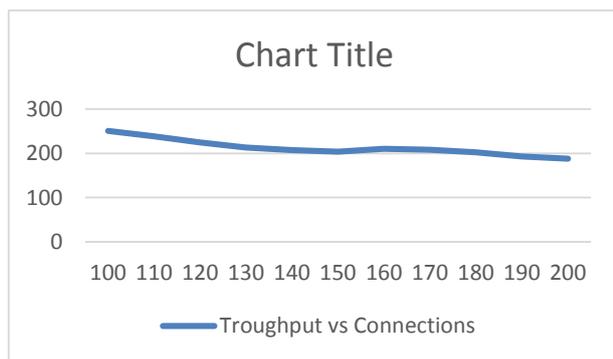
Pengujian *webench* dengan menggunakan *httperf* diperlukan untuk mendapatkan hasil berupa data perbandingan serta masukkan seberapa besar performa yang dimiliki oleh sebuah server.

Tabel 1 Hasil pengujian *Httpperf*

Conne	Conne	Conn	Reque	Reply	Troug
ction	ction	urrent	st	Rate(r	hput
	Rate	Conne	Rate(r	ep/s)	
		ctions	eq/s)		
100	36.1	78	24.2	20.2	250.4
110	35.3	84	25.3	22.4	238.6
120	34.1	93	26.7	23.1	224.7
130	34.0	102	25.8	24.3	213.1

140	33.8	113	27.4	24.8	207.6
150	31.9	132	28.0	25.1	203.8
160	29.7	130	28.7	24.7	210.3
170	28.6	148	29.3	26.4	207.8
180	29.7	162	29.7	27.1	202.2
190	30.0	168	30.4	24.3	192.7
200	29.2	177	28.6	26.8	187.9

Tabel 1 adalah hasil pengujian saat server up, server yang dirancang masih mampu melayani permintaan sampai 200 koneksi. Kemampuan optimal saat server dalam keadaan *idle* berada diantara 150 - 170 koneksi yang menghasilkan *throughput* sebesar 203.8 - 210.3 KB/s tanpa adanya permintaan yang *error*. Jika server memiliki perangkat yang lebih besar, bukan tidak mungkin server yang dirancang ini mampu melayani permintaan yang lebih banyak.



Gambar 11 Grafik Troughput vs. Connections

Grafik hasil dari pengujian koneksi yang diberikan terhadap server dan *throughput* yang diterima oleh klien menggunakan *httperf*. Karena menggunakan server tunggal, maka pengujian *httperf* tidak menggunakan berbagai kondisi. Pada gambar 10 bisa dilihat bahwa hasil uji yang diberikan mesin *ndbcluster* cenderung lebih stabil dalam memberikan *feedback* kepada klien. Sinyal *throughput* yang dikirimkan dengan respon yang diminta juga terlihat stabil, walaupun grafik memperlihatkan performa yang menurun, tetapi masih lebih stabil dibandingkan dengan *engine* basisdata yang lain.

V. PENUTUP

Berdasarkan hasil pengujian dan analisis, dapat diambil beberapa kesimpulan. Basisdata kluster yang diimplementasikan dengan mesin *ndbcluster* bersifat *high availability*. Basisdata kluster yang dirancang mampu melakukan replikasi dan pembaharuan data secara *realtime*. Hasil pengujian menunjukkan bahwa

kemampuan optimal saat server dalam keadaan *idle* berada diantara 150 - 170 koneksi yang menghasilkan *throughput* sebesar 203.8 - 210.3 KB/s tanpa adanya permintaan yang *error*. Node manajemen dirancang untuk melakukan monitoring dan mesin *ndbcluster* untuk melakukan replikasi diantara 2 server basisdata. Basisdata kluster dengan mesin *ndbcluster* memiliki keunggulan dalam performansi karena tidak memerlukan piranti lunak tambahan pada implementasinya.

Beberapa saran yang dapat diberikan sehubungan penelitian ini yaitu, rancangan basisdata kluster dapat diimplementasikan pada perangkat fisik dengan menambahkan kapasitas memori dan prosesor agar lebih handal dan optimal. Rancangan basisdata kluster akan berkerja lebih maksimal jika dikolaborasi pada sisi server aplikasi dengan teknologi server kluster. Penelitian lebih lanjut mengenai mesin *ndbcluster* yang digunakan untuk mengoptimalkan kinerja server basisdata bisa dilakukan dengan menerapkan replikasi *master-to-master* serta menambahkan jumlah node yang dijadikan sebagai basisdata kluster.

DAFTAR PUSTAKA

- [1] Adelheid, A., & Nst, K. (2012). BUKU PINTAR menguasai PHP MySQL. In A. Adelheid, & K. Nst, *BUKU PINTAR menguasai PHP MySQL*. mediakita.
- [2] Charles Bell, M. K. (2010). *MySQL High Availability: Tools for Building Robust Data Centers*. In M. K. Charles Bell, *MySQL High Availability: Tools for Building Robust Data Centers* (p. Juni). O'Reilly Media.
- [3] Davies, A. (2010). *High Availability MySQL Cookbook*. In A. Davies, *High Availability MySQL Cookbook*. PACKT.
- [4] Ellingwood, J. (2014, April 24). *DigitalOcean™ Inc*. Retrieved from How To Install Wordpress on Ubuntu 14.04: <https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-on-ubuntu-14-04>
- [5] Gultom, H. (2009, Juni 11). *mysql-kluster*. Retrieved from *mysql-kluster*: <http://henry.gultom.or.id/2009/06/11/membuat-mysql-kluster/#more-286>
- [6] Hariyanto, D. B. (2007). *Sistem Operasi*. In D. B. Hariyanto, *Sistem Operasi*. Informatika Bandung.

- [7] *Shared nothing kluster vs shared disk kluster.* (n.d.). Retrieved from Shared nothing kluster vs shared disk kluster:
<http://www.evidian.com/products/high-availability-software-for-application-klustering/shared-nothing-kluster-vs-shared-disk-kluster/>
- [8] Stopford, B. (2009, November 24). *Shared Nothing v.s. Shared Disk Architectures: An Independent View.* Retrieved from Shared Nothing v.s. Shared Disk Architectures: An Independent View:
<http://www.benstopford.com/2009/11/24/understanding-the-shared-nothing-architecture/>
- [9] Sumathi, D. S., & Esakkirajan, S. (2007). Fundamentals of Relational Database Management Systems. In D. S. Sumathi, & S. Esakkirajan, *Fundamentals of Relational Database Management Systems.* Springer.
- [10] Tarreau, W. (2015, 02 01). *HAProxy Configuration Manual.* Retrieved from HAProxy Configuration Manual:
<http://www.haproxy.org/>