

IMPLEMENTASI MySQL CLUSTER PADA BASIS DATA TERDISTRIBUSI

I Gde Budi Rinanta Putra
Program Studi Teknik Informatika,
Jurusan Ilmu Komputer,
Fakultas Matematika Dan Ilmu Pengetahuan Alam,
Universitas Udayana
Email: rinanta.putra@cs.unud.ac.id

ABSTRAK

Perusahaan harus memiliki sistem informasi yang lengkap dan mudah diakses oleh siapapun. Dengan adanya sistem informasi itu perusahaan menunjukkan kredibilitasnya untuk go public dan untuk lebih dikenal oleh khalayak umum. Dalam praktiknya sistem informasi itu ketika diakses oleh client maupun khalayak umum kadang terjadi kegagalan. Itu dikarenakan karena di sisi server terjadi failure. Kegagalan atau failure itu sendiri disebabkan karena server mati dan tidak ada backup dari server lain yang langsung menggantikan ketika server utama mati. Salah satu solusi untuk mengatasi masalah diatas adalah dengan menggunakan teknologi MySQL Cluster. Itu dikarenakan di dalam teknologi MySQL Cluster terdapat replikasi database juga terdapat sistem yang mampu mengatasi failure sistem database itu sendiri.

Kata kunci : Cluster Server, *Basis Data Terdistribusi*, *MySQL Cluster*

ABSTRACT

The Companies must have a complete information system and easily accessible by anyone. With the company's information system that shows credibility to go public and to better known by the general public. In practice the information system when it is accessed by the client and the general public sometimes failure. That is because failure occurred on the server side. Failure or failure caused by the server itself dies and there is no backup from another server when the primary server directly replace dead. One solution to overcome the above problems is to use the MySQL Cluster technology. That is because in the MySQL Cluster technology there are also database replication system capable of dealing with failure, the database system itself.

Keywords : Cluster Server, Distributed Database, MySQL Cluster

PENDAHULUAN

Sebuah instansi atau perusahaan yang namanya cukup dikenal luas oleh berbagai pihak pada umumnya telah terdapat Sistem Informasi yang cukup memadai. Sistem informasi itu sendiri digunakan untuk menunjang kredibilitas sebuah perusahaan itu sendiri dan juga untuk mempermudah client dalam mencari info dan seluk beluk perusahaan itu itu sendiri. Dengan adanya sistem informasi yang kompleks itu dibutuhkan penyimpanan database untuk menunjang berjalannya sistem informasi.

Dalam kapasitasnya sebagai sebuah perusahaan tentunya perusahaan harus memiliki sistem informasi yang lengkap dan mudah diakses oleh siapapun. Dengan adanya sistem informasi itu perusahaan menunjukkan kredibilitasnya untuk go public dan untuk lebih dikenal oleh khalayak umum. Di dalam sistem informasi itu terdapat berbagai macam info tentang perusahaan itu sendiri. Misalnya, profile, struktur organisasi, produk yang ditawarkan, promo dan pengumuman pengumuman yang berhubungan dengan perusahaan tersebut.

Dalam praktiknya sistem informasi itu ketika diakses oleh client maupun khalayak umum kadang terjadi kegagalan. Itu dikarenakan karena di sisi server terjadi failure. Kegagalan atau failure itu sendiri disebabkan karena server mati dan tidak ada backup dari server lain yang langsung

menggantikan ketika server utama mati. Oleh karena itu penyimpanan database yang baik harus bersifat *high availability*.

Salah satu solusi untuk mengatasi masalah diatas adalah dengan menggunakan teknologi MySQL Cluster. Itu dikarenakan di dalam teknologi MySQL Cluster terdapat replikasi database juga terdapat sistem yang mampu mengatasi failure sistem database itu sendiri. Oleh karena itu dengan implementasi MySQL cluster ini diharapkan sistem penyimpanan database itu bersifat *high availability*. Sehingga apabila terjadi sistem *failure* pada server utama bisa langsung digantikan dengan *secondary server*.

MATERI DAN METODE

2.1 *Distributed Database*

Basis data terdistribusi (*Distributed Database*) adalah kumpulan database yang disimpan di banyak komputer pada lokasi yang berbeda-beda dan menampilkan ke user sebagai single database. Dalam sebuah database terdistribusi, database disimpan pada beberapa komputer. Komputer-komputer dalam sebuah sistem terdistribusi berhubungan satu sama lain melalui bermacam-macam media komunikasi seperti high-speed buses atau telephone line.

Sebuah sistem database terdistribusi berisikan sekumpulan site, di mana tiap-tiap site dapat berpartisipasi dalam pengekseskuan transaksi-transaksi

yang mengakses data pada satu site atau beberapa site. Tiap-tiap site dapat memproses transaksi lokal yaitu sebuah transaksi yang mengakses data pada satu site di mana transaksi telah ditentukan. Sebuah site juga dapat mengambil bagian dalam mengeksekusi transaksi global yaitu transaksi yang mengakses data pada site yang berbeda di mana transaksi telah ditentukan, atau transaksi yang mengakses data pada beberapa site yang berbeda.

2.2 Database Cluster

Database clustering adalah kumpulan dari beberapa server yang berdiri sendiri yang kemudian bekerjasama sebagai suatu sistem tunggal. Saat ini aplikasi database semakin berkembang, baik dalam hal kegunaan, ukuran, maupun kompleksitas. Hal ini secara langsung berdampak pada server *database* sebagai penyedia layanan terhadap akses *database*, konsekuensi dari semua itu adalah beban database server akan semakin bertambah berat dan mengakibatkan kurang optimalnya kinerja dari *server* tersebut.

Oleh karena itu diperlukan perancangan yang tepat dan handal dalam membangun *database server*. *Database* pada masa sekarang ini dituntut agar dapat berjalan dengan cepat, mempunyai kehandalan dan keseterdaian yang tinggi, dengan *clustering database* yang disimpan

dapat terbagi ke beberapa mesin dan pada saat aplikasi berjalan, semua mesin yang menyimpan data tersebut dianggap sebagai satu kesatuan. Metode clustering seperti ini sangat baik untuk *loadbalancing* dan penanganan *system failure* karena kemampuan tiap mesin akan digunakan dan jika ada salah satu mesin yang mengalami *failure* maka sistem tidak akan langsung terganggu karena mesin lain akan tetap berfungsi. Kemampuan *clustering* memungkinkan sebuah *database* tetap hidup dalam waktu yang lama.

2.3 MySQL Cluster

MySQL Cluster merupakan sebuah tipe basis data (*database*) yang dapat beroperasi dalam ukuran data yang relatif besar (maksimal dalam skala beberapa ratus gigabyte). MySQL Cluster adalah sebuah teknologi baru untuk memungkinkan clustering di dalam memory database dalam sebuah sistem share-nothing. Arsitektur share-nothing mengijinkan sistem dapat bekerja dengan hardware/perangkat keras yang sangat murah, dan tidak membutuhkan perangkat keras dan lunak dengan spesifikasi khusus. Arsitektur tersebut juga handal karena masing-masing komponen mempunyai memory dan disk tersendiri.

MySQL Cluster menggabungkan MySQL Server biasa dengan sebuah mesin penyimpanan in-memory tercluster yang dinamakan NDB. NDB berarti bagian dari suatu rangkaian yang dikhususkan sebagai

mesin penyimpanan, sedangkan MySQL Cluster diartikan sebagai kombinasi atau gabungan dari MySQL dan mesin penyimpanan yang baru tersebut.

2.4 Arsitektur MySQL Cluster

MySQL Cluster merupakan sebuah database yang menggunakan arsitektur shared-nothing dan antarmuka SQL yang telah umum digunakan. Sistem database ini terdiri dari beberapa node yang dapat didistribusikan ke beberapa perangkat keras dan ke beberapa wilayah/zona yang berbeda sekaligus untuk tetap menjaga ketersediaan data meskipun jaringan ataupun salah satu node sedang mengalami kegagalan (failure). Ada tiga node yang menyusun MySQL Cluster, yakni:

1. Data Nodes, digunakan untuk menyimpan semua data yang menjadi milik MySQL Cluster. Semua data direplikasi di node-node ini.
2. Management Server Nodes, digunakan untuk mengendalikan konfigurasi sistem ketika startup. Selain itu, node ini juga dapat digunakan sebagai pengidentifikasi setiap perubahan setting yang terjadi pada cluster.
3. MySQL Server Nodes, berfungsi sebagai pintu akses untuk masuk ke dalam node-node data yang ter-cluster.

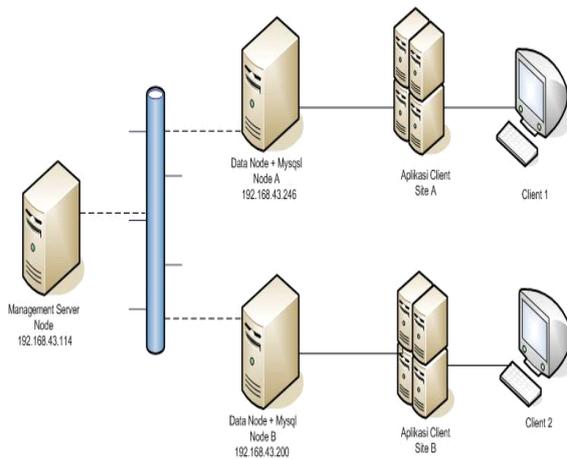
HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

3.1.1 Perancangan MySQL Cluster

Dalam perancangan ini dibutuhkan tiga server. Server pertama untuk Management Server Node, server kedua untuk Data Node dan server ketiga MySQL Server Nodes. Management Server Nodes, digunakan untuk mengendalikan konfigurasi sistem ketika startup. Selain itu, node ini juga dapat digunakan sebagai pengidentifikasi setiap perubahan setting yang terjadi pada cluster. Data Nodes, digunakan untuk menyimpan semua data yang menjadi milik MySQL Cluster. Semua data direplikasi di node-node ini. MySQL Server Nodes, berfungsi sebagai pintu akses untuk masuk ke dalam node-node data yang tercluster. Management Server Node dan Data Node dihubungkan tujuannya agar bisa terjadi replikasi data antara kedua server tersebut dan juga berfungsi untuk mengatasi kegagalan sistem database pada salah satu sisi server agar server yang lain bisa menggantikan tugas server yang lainnya.

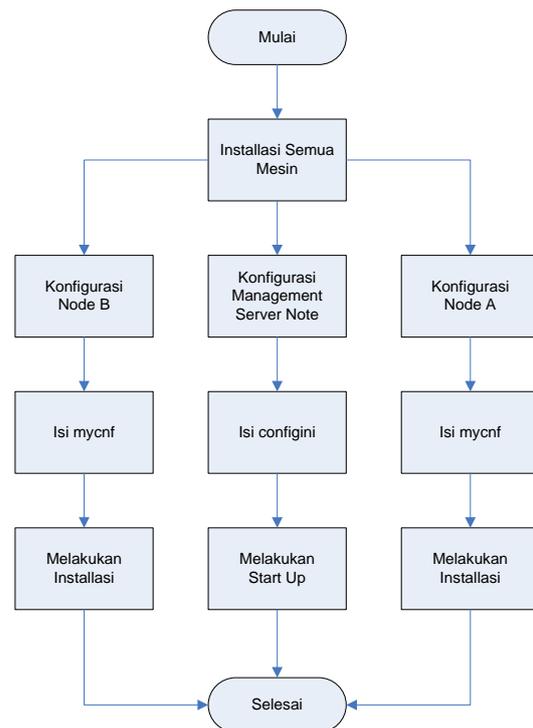
Agar perancangan itu bisa dilakukan dengan baik digunakan teknologi MySQL Cluster. Pada tahapan pembuatan sistem cluster dibagi tiga bagian utama yaitu Management Server Node, Data Node dan MySQL Server Nodes. Penggambaran sistem secara umum dapat dilihat pada gambar 3.1 :



Gambar 3. 1 Gambaran Umum Perancangan Sistem

Dari gambar tersebut Data Node dan MySQL Node digabungkan menjadi satu Node. Kemudian untuk menghubungkan Node A dan Node B dibutuhkan Management Server Node. Jadi fungsi utama Management Server Node adalah menghubungkan kedua node tersebut sehingga fungsi dari MySQL cluster itu sendiri dapat berjalan dengan fungsi sebagai replikasi data dan high availability.

Inti dari konfigurasi Mysql cluster adalah menginstal mysql cluster secara manual, bukan langsung menginstal dari paket yang tersedia di repository yang sudah disediakan oleh linux. Tetapi berbeda halnya ketika kita memakai windows, cara menginstal bisa secara manual maupun secara langsung dengan menggunakan .exe nya. Konfigurasi ini sendiri terbagi menjadi tiga yaitu: pengaturan konfigurasi di sisi Management Server Node, pengaturan konfigurasi di sisi Node A dan pengaturan di sisi Node B. Berikut adalah Flowchartnya :



Gambar 3. 2 Diagram Alir Proses Konfigurasi Perangkat Keras

3.1.2 Tahap-Tahap Instalasi MySQL Cluster

3.1.2.1 Tahap Instalasi Data dan SQL node di node1 dan node2

Pada masing-masing node yang ditunjuk untuk menjadi rumah data atau node SQL (dalam kasus node01 dan node02), melakukan langkah-langkah berikut sebagai user root sistem :

1. Membuat grup MySQL pengguna baru, dan kemudian menambahkan user MySQL :


```
shell> groupadd mysql
shell> useradd-g mysql mysql
```
2. Mengubah lokasi ke direktori yang berisi file yang didownload, membongkar arsip, dan menciptakan symlink ke direktori mysql bernama

mysql. Perhatikan bahwa file yang sebenarnya dan nama direktori bervariasi sesuai dengan jumlah cluster versi MySQL.

```
shell> cd / var / tmp
shell> tar-C / usr / local-
xzvf mysql-cluster-gpl-7.1.5-
linuxx86_
64-glibc23.tar.gz
shell> ln-s /
usr/local/mysql-cluster-gpl-
7.1.5-linux-i686-glibc23
/ usr / local / mysql
shell> export PATH = $ PATH:
/ usr / local / mysql / bin
shell> echo "export PATH = \
$ PATH: / usr / local / mysql
/
bin">> / etc / bash.bashrc
```

3. Mengubah lokasi ke direktori mysql dan menjalankan script yang disediakan untuk menciptakan sistem database :

```
shell> cd mysql
shell>. / scripts /
mysql_install_db-user = mysql
```

4. Mengatur izin yang diperlukan untuk server MySQL dan direktori data :

```
shell> chown-R root.
shell> chown-R mysql data
shell> chgrp-R mysql.
```

5. Salin *script startup MySQL* ke direktori yang sesuai, membuatnya menjadi executable, dan set untuk memulai ketika sistem beroperasi :

```
shell> cp support-
files/mysql.server / etc /
init.d / mysql
shell> chmod + x / etc /
init.d / mysql
shell> update-rc.d mysql
default
```

3.1.2.2 Instalasi dari node Manajemen pada node03

Pemasangan node manajemen tidak memerlukan biner mysqld. Hanya

memerlukan manajemen server MySQL Cluster (ndb_mgmd), kami berasumsi bahwa telah menempatkan mysql-cluster-gpl-7.1.5-linux-i686-glibc23.tar.gz di /var / tmp. Sistem sebagai root melakukan langkah-langkah sebagai berikut untuk menginstal ndb_mgmd dan ndb_mgm pada host Cluster.

1. Ubah lokasi ke / var / tmp direktori, dan ekstrak ndb_mgm dan ndb_mgmd dari arsip ke direktori yang sesuai seperti / usr / local / bin :

```
shell> cd / var / tmp
shell> tar-zxvf mysql-
cluster-gpl-7.1.5-linux-i686-
glibc23.tar.gz
shell> cd / usr/local/mysql-
cluster-gpl-7.1.5-linux-i686-
glibc23
shell> cp bin / ndb_mgm * /
usr / local / bin
```

2. Mengubah lokasi ke direktori di mana tempat pada saat menyalin file, dan kemudian membuat keduanya dieksekusi :

```
shell> cd / usr / local / bin
shell> chmod + x ndb_mgm *
```

3.1.2.3 Konfigurasi Manajemen Node

Langkah pertama dalam mengkonfigurasi manajemen node adalah untuk membuat direktori di mana file konfigurasi dapat ditemukan dan kemudian untuk membuat file itu sendiri :

```
shell> mkdir / var / lib /
mysql-cluster
shell> cd / var / lib / mysql-
cluster
shell> vi config.ini
```

Untuk setup file config.ini diisi scrip sebagai berikut:

```
[ndbd default]
```

```

NoOfReplicas=2
DataMemory=80M
IndexMemory=18M
[tcp default]
[ndb_mgmd]
hostname=192.168.43.114 #
Hostname or IP address of MGM
node
datadir=/var/lib/mysql-cluster #
Directory for MGM node log files
[ndbd]
hostname=192.168.43.200 #
Hostname or IP address
datadir=/usr/local/mysql/data #
Directory for this data node's
data files
[ndbd]
hostname=192.168.43.246 #
Hostname or IP address
datadir=/usr/local/mysql/data #
Directory for this data node's
data files
[mysqld]
hostname=192.168.43.200 #
Hostname or IP address
[mysqld]
hostname=192.168.43.246 #
Hostname or IP address

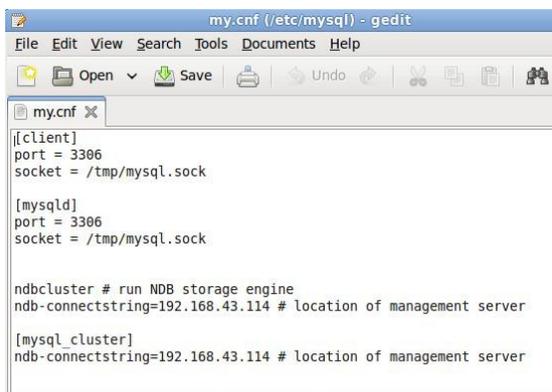
```

3.1.2.4 Konfigurasi Data dan SQL Node

Konfigurasi Data dan SQL Node dilakukan mengedit file my.cnf di direktori /etc/

```
shell> vi /etc/my.cnf
```

Untuk setiap data node dan SQL node di setup pada my.cnf akan terlihat seperti gambar 3.3 :



Gambar 3. 3 Konfigurasi my.cnf pada data node

Setelah memulai sebuah proses mysqld dengan NDBCLUSTER dan ndbconnectstring parameter dalam [mysqld] di file my.cnf seperti yang ditunjukkan sebelumnya, User tidak dapat mengeksekusi CREATE TABLE atau ALTER TABLE tanpa harus benar-benar mulai cluster.

3.1.2.5 Memulai MySQL Cluster

Memulai cluster ini tidak terlalu sulit setelah dikonfigurasi sebelumnya. Setiap proses node cluster harus dimulai secara terpisah, dan di host di mana dia berada. Manajemen node harus dimulai terlebih dahulu, diikuti oleh node data, dan kemudian terakhir oleh setiap node SQL :

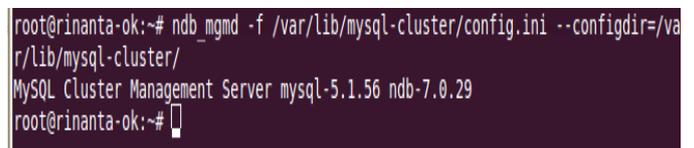
1. Pada host manajemen (node03), mengeluarkan perintah berikut dari shell sistem untuk memulai proses manajemen node :

```

shell> ndb_mgmd -f
/var/lib/mysql-
cluster/config.ini-
configdir=/var/lib/mysql-
cluster/

```

jika berhasil akan muncul gambar seperti gambar di bawah ini.



Gambar 3. 4 Start management node

2. Pada masing-masing Data / host SQL node, jalankan perintah ini untuk memulai ndbd dan proses mysql server :

```
shell> / usr / local / mysql
/ bin / ndbd
```

Jika berhasil akan keluar tampilan seperti gambar 3.5 :

```
root@baglug-laptop:/etc/mysql# /usr/
2012-01-09 18:32:24 [ndbd] INFO
2012-01-09 18:32:24 [ndbd] INFO
root@baglug-laptop:/etc/mysql#
```

Gambar 3. 5 Menjukan bahwa data note terkoneksi dengan management note

Setelah itu mengaktifkan mysql pada data note dengan perintah seperti di bawah ini.

```
shell> / etc / init.d / mysql
start
```

Kemudian diuji pada client node dengan perintah `ndb_mgm`. Jika berhasil output yang dihasilkan akan ditunjukkan seperti gambar 3.6 di bawah ini.

```
root@rinanta-ok:~# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.43.200 (mysql-5.1.56 ndb-7.0.29, Nodegroup: 0)
id=3 @192.168.43.246 (mysql-5.1.56 ndb-7.0.29, Nodegroup: 0, Master)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.43.114 (mysql-5.1.56 ndb-7.0.29)

[mysqld(API)] 2 node(s)
id=4 @192.168.43.200 (mysql-5.1.56 ndb-7.0.29)
id=5 @192.168.43.246 (mysql-5.1.56 ndb-7.0.29)

ndb_mgm>
```

Gambar 3. 6 Kedua Data Note Saling Terkoneksi

3.2 Pengujian Sistem

3.2.1 Tampilan Database

Sebelum menampilkan aplikasi, yang di konfigurasi pertama kali adalah database pada site masing-masing yang terdapat pada SQL node.

Gambar 3. 7 Setting Database untuk tiap Site

```
database.php
*/
$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = '192.168.43.246';
$db['default']['username'] = 'root';
$db['default']['password'] = 'password';
$db['default']['database'] = 'baster1';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

Gambar 3. 8 Tampilan Konfigurasi Site1

```
database.php
$db['default']['hostname'] = '192.168.43.200';
$db['default']['username'] = 'root';
$db['default']['password'] = 'password';
$db['default']['database'] = 'baster1';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

Gambar 3. 9 Tampilan Konfigurasi Site2

3.2.2 Tampilan Aplikasi

Berikut tampilan aplikasi :

http://site2.com/baster/index.php/nasabah

Nama :

Alamat :

Saldo :

Data User

No	Nama	Alamat	Saldo	Edit	Delete
1	oka gartria	tabanan	20000	Edit	Delete
2	surya	denpasar	10000	Edit	Delete
3	firman	gianyar	20099	Edit	Delete
4	budi rinanta	gianyar	100000	Edit	Delete

Gambar 3. 10 Tampilan Aplikasi

3.2.3 Pengujian Kinerja Replikasi Jika Terjadi Kegagalan

Dalam pembuatan MySQL Cluster, tujuannya adalah untuk mengatasi dalam hal ketersediaan data. Disini dicoba untuk menampilkan apabila terjadi kegagalan data tetap tersimpan di database. Berikut tampilan server yang masih hidup:

```

ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.43.200 (mysql-5.1.56 ndb-7.0.29, Nodegroup: 0)
id=3 @192.168.43.246 (mysql-5.1.56 ndb-7.0.29, Nodegroup: 0, Master)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.43.114 (mysql-5.1.56 ndb-7.0.29)

[mysqld(API)] 2 node(s)
id=4 @192.168.43.200 (mysql-5.1.56 ndb-7.0.29)
id=5 @192.168.43.246 (mysql-5.1.56 ndb-7.0.29)

ndb_mgm>

```

Gambar 3. 11 Tampilan Server

Jika sebuah node di matikan, yaitu pada contoh ini adalah node 2, perintah yang dimasukkan :

```

ndb_mgm> 2 stop
Node 2: Node shutdown initiated

Node 2 has shutdown.

ndb_mgm>
ndb_mgm> Node 2: Node shutdown complete

```

Gambar 3. 12 Shutdown Node 2

Sehingga hasil yang dapat dilihat di management Node :

```

ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 (not connected, accepting connect from 192.168.43.200)
id=3 @192.168.43.246 (mysql-5.1.56 ndb-7.0.29, Nodegroup: 0, Master)

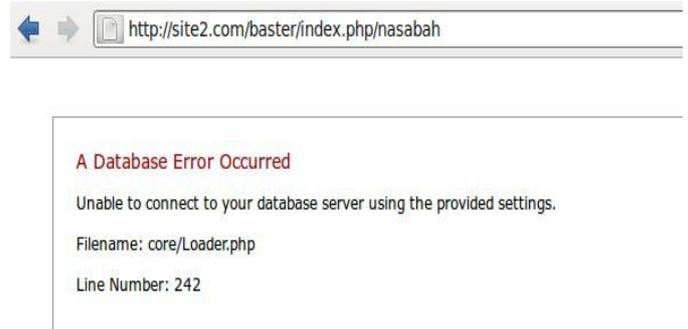
[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.43.114 (mysql-5.1.56 ndb-7.0.29)

[mysqld(API)] 2 node(s)
id=4 (not connected, accepting connect from 192.168.43.200)
id=5 @192.168.43.246 (mysql-5.1.56 ndb-7.0.29)

ndb_mgm>

```

Gambar 3. 13 Node 2 Disconnected



Gambar 3. 14 Site 2 Disconnected

Setelah Node 2 disconnected, maka dicoba memasukkan data pada site 1 :

Nama :

Alamat :

Saldo :

Data User

No	Nama	Alamat	Saldo	Edit	Delete
1	oka gartria	tabanan	20000	Edit	Delete
2	surya	denpasar	10000	Edit	Delete
3	budi rinanta	gianyar	100000	Edit	Delete

Gambar 3. 15 Tambah Data Site 1

Setelah site 2 kembali di sinkronkan, maka data yang sebelumnya di masukan dari site 1 pada saat site 2 disconnected akan terlihat, berikut tampilannya :

Nama :

Alamat :

Saldo :

Data User

No	Nama	Alamat	Saldo	Edit	Delete
1	oka gartria	tabanan	20000	Edit	Delete
2	surya	denpasar	10000	Edit	Delete
3	budi rinanta	gianyar	100000	Edit	Delete
4	firman	gianyar	50000	Edit	Delete

Gambar 3. 16 Tampilan Site2

SIMPULAN

Simpulan yang dapat diambil dari penelitian yang telah dilakukan adalah sebagai berikut,

1. Server mysql cluster berjalan di protocol TCP/IP.
2. Engine database yang digunakan dalam server mysql cluster ini menggunakan ndbcluster.
3. Kofigurasi mysql cluster ini terdiri dari tiga node, yaitu Management node (*ndbd_mgmd process*), Data node (*ndbd process*), SQL node (*mysqld process*)
4. MySQL cluster merupakan sebuah database yang menggunakan arsitektur shared-nothing.

UCAPAN TERIMA KASIH

Sehubungan dengan telah terselesaikannya penelitian ini, maka diucapkan terima kasih dan penghargaan kepada berbagai pihak yang telah membantu, antara lain:

1. Bapak Drs. I Wayan Santiyasa, M.Si., selaku Ketua Jurusan Ilmu Komputer Fakultas MIPA Universitas Udayana.
2. Seluruh dosen, staf pegawai, dan rekan rekan mahasiswa di Jurusan Ilmu Komputer Fakultas MIPA Universitas Udayana yang telah meluangkan waktu untuk memberikan saran dan masukan

untuk menyempurnakan penelitian ini.

KEPUSTAKAAN

- [1] Basis Data Terdistribusi. [Online] Tersedia : <http://lenidian.staff.gunadarma.ac.id/Downloads/files/26235/distdb1.pdf>. (diakses tanggal 9 September 2011, jam 17.04 Wita)
- [2] BAB 1 DBMS Terdistribusi – Konsep dan Desain. [Online] Tersedia : <http://irmarr.staff.gunadarma.ac.id/Downloads/files/11629/BAB+1+DDBMS+Konsep+dan+Design.doc>. (diakses tanggal 9 September 2011, jam 17.06 Wita)
- [3] Mysql-cluster. [Online] Tersedia : <http://wimpermana.web.ugm.ac.id/blog/wp-content/mysql-cluster.pdf> (diakses tanggal 7 Januari 2012, jam 18.14 Wita)
- [4] Prabowo, Adityo.2010.*Perancangan MySQL Cluster Untuk Mengatasi Kegagalan Sistem Basis Data Pada Sisi Server* . Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro : Semarang [Online] Tersedia : <http://eprints.undip.ac.id/25272/1/Makalahku.pdf> (diakses tanggal 7 Januari 2012, jam 18.23 Wita)