

# Perancangan MySQL Cluster Untuk Mengatasi Kegagalan Sistem Basis Data Pada Sisi Server

Adityo Prabowo<sup>1)</sup>, Kodrat Iman Satoto<sup>2)</sup>, Maman Soemantri<sup>2)</sup>  
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,  
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

## ABSTRACT

*With the development of information technology, changing human in completing all works. Not only in working but in all aspects of human life, such as when storing data on a server information. In the modern technology needed reliable server availability data. In consequencethe, server it must be high availability and high scalability. In order for such a server can be used with MySQL Cluster technology.*

*The research methodology of this thesis, among others, with library research, system design, and testing of the system. In the literature study used methods of research library of reference books related. This final design uses a system of MySQL Cluster. Last is testing this system on it's performance on a server failure.*

*The results obtained are in case of failure or major failure on the side of the server, in terms of availability of the data will be directly secured by another server is a slave. And the presence of data replication between the primary and the slave servers.*

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

## I. PENDAHULUAN

### Latar Belakang

Sebuah universitas 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 universitas itu sendiri dan juga untuk mempermudah mahasiswa dalam mencari info dan seluk beluk universitasnya itu sendiri. Dengan adanya sistem informasi yang kompleks itu dibutuhkan penyimpanan *database* untuk menunjang berjalannya sistem informasi.

Dalam kapasitasnya sebagai universitas terkemuka tentunya Universitas Diponegoro memiliki sistem informasi yang lengkap dan mudah diakses oleh mahasiswanya. Dengan adanya sistem informasi itu Undip menunjukkan kredibilitasnya *untuk go public* dan untuk lebih dikenal oleh khalayak umum. Di dalam sistem informasi itu terdapat berbagai macam info tentang Undip itu sendiri. Misalnya, SIA, info beasiswa, dan pengumuman-pengumuman yang berhubungan dengan mahasiswa.

Dalam praktiknya sistem informasi itu ketika diakses oleh mahasiswa 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 adanya Tugas Akhir ini diharapkan sistem penyimpanan *database* itu bersifat *high availability*. Sehingga apabila terjadi sistem *failure* pada *server* utama bisa langsung digantikan dengan *server*

*slave* sehingga di sisi *client* nyaman-nyaman saja dalam mengakses sistem informasi.

### Tujuan

Tujuan dari tugas akhir ini adalah mempelajari penggunaan, cara kerja dan fungsi dari system MySQL Cluster untuk mengatasi kegagalan dalam hal ketersediaan data.

### Batasan Masalah

Agar pembahasan atau analisis tidak melebar dan terarah, maka permasalahan dibatasi pada :

- Tugas akhir ini akan dirancang pada bagian *database* dengan menggunakan MySQL Cluster.
- Tugas akhir ini akan membahas / menganalisis setiap kinerja replikasi *database* dan apabila *server* utama mengalami kegagalan.
- Tugas Akhir ini dirancang untuk berjalan di atas Sistem Operasi LINUX dan Windows.
- Tugas Akhir ini tidak membahas tentang hacking dan system keamanan pada sisi server dan client.
- Tugas akhir ini tidak membahas *coding* aplikasi yang digunakan untuk *testing* mysql cluster

## II. LANDASAN TEORI

### Database (Basis Data)

*Database* adalah sebuah koleksi dari data yang saling berelasi, dimana data tersebut disimpan pada komputer sedemikian hingga sebuah program komputer dapat berinteraksi dan menggunakan data yang disimpan tersebut untuk menyelesaikan masalah ataupun menjawab pertanyaan (Elmasri 1994). Data perlu disimpan dalam suatu *database* untuk keperluan penyediaan informasi lebih lanjut. Data didalam *database* perlu diorganisasikan sedemikian rupa

1) Mahasiswa Teknik Elektro UNDIP

2) Dosen Teknik Elektro UNDIP

sehingga informasi yang terkandung didalamnya mudah diakses.

Database merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di *database*. Pengelolaan *database* yang buruk dapat mengakibatkan ketidakterediaan data penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan.

### Database Terdistribusi

*Database* Terdistribusi bisa diartikan sebagai kumpulan dari data-data dengan berbagai bagian yang ditangani DBMS (*Data Base Management System*) secara terpisah dan berjalan pada sistem komputer.

Semua komputer saling terhubung dan setiap sistem mempunyai hak kemampuan memproses untuk melayani permintaan lokal. Setiap sistem berpartisipasi baik dalam melaksanakan satu atau lebih permintaan. Beberapa permintaan memerlukan data tidak hanya dari satu tempat melainkan beberapa tempat atau lebih. *Database* asli yang terdistribusi itu sendiri tidak terlihat oleh user dan transparansi dari data tersebut dapat ditunjukkan dalam berbagai cara.

Distribusi data dalam DBMS ada 2 bentuk :

- a. Data *partitioning* (data yang terpisah-pisah)
- b. Data *replication* (replikasi data)

Kedua bentuk distribusi data di atas merupakan transparan artinya tampak nyata oleh user dalam sistem distribusi *database*.

Pada database terdistribusi (*distributed database*), data disimpan pada beberapa tempat (*site*), setiap tempat diatur dengan suatu DBMS (*Database Management System*) yang dapat berjalan secara independent. Berikut adalah contoh arsitektur dari DBMS itu sendiri.

- a. *Client-Server*
- b. *Collaboration Server*
- c. *Fragmentasi*
- d. Replikasi

### Database Cluster

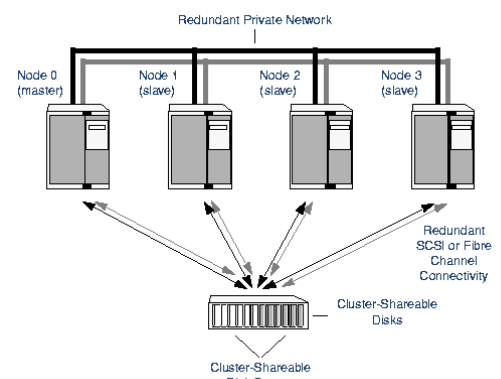
*Database clustering* adalah kumpulan dari beberapa server yang berdiri sendiri yang kemudian bekerjasama sebagai suatu sistem tunggal (Hodges, 2007). Saat ini aplikasi database semakin berkembang, baik dalam hal kegunaan, ukuran, maupun kompleksitas. Hal ini secara langsung berdampak pada server *database* 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 keseterediaan 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 *load-balancing* 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. Berikut adalah contoh arsitektur dari database cluster itu sendiri.

#### a. Shared Disk Clusters

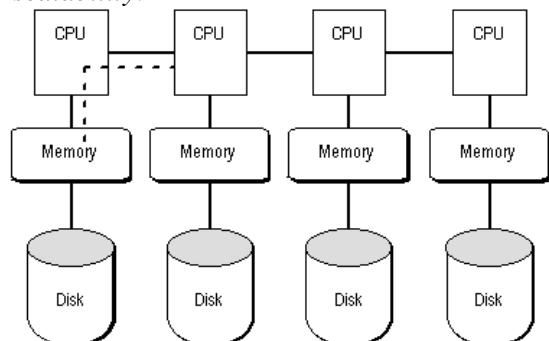
Arsitektur *shared disk clusters* menggunakan server-server independent dan berbagi sebuah sistem penyimpanan tunggal. Setiap server mempunyai prosesor dan memori sendiri, tetapi berbagi *disk resources*. Implementasi utama dari *shared-disk clustering* adalah bukan untuk *scalability*. *Shared-disk clustering* ini diimplementasikan untuk *availability* dan menambah node cadangan sebagai *failover node*.



Gambar 1. Shared Disk Clusters

#### b. Shared Nothing Cluster

Dalam arsitektur *shared nothing cluster*, tiap server dalam cluster menangani prosesor, memori, storage, record locks dan transaksi yang terpisah dan melakukan koordinasi dengan server lain melalui jaringan dengan menggunakan *high speed, low-latency interconnect technology*. Dalam proses permintaan data suatu node harus mengirimkan pesan ke node yang lain yang memiliki data yang diakses. Hal ini juga dilakukan saat koordinasi data yang dilakukan pada node yang lain seperti *insert, select, update* dan *delete*. Berbeda dengan shared disk, shared nothing didisain untuk *high availability* dan *scalability*.



Gambar 2 Shared Nothing

## Replikasi

Replikasi adalah suatu teknik untuk melakukan *copy* dan pendistribusian data dan objek-objek *database* dari satu *database* ke *database* lain dan melaksanakan sinkronisasi antara *database* sehingga konsistensi data dapat terjamin.

Dengan menggunakan teknik replikasi ini, data dapat didistribusikan ke lokasi yang berbeda melalui koneksi jaringan lokal maupun internet. Replikasi juga memungkinkan untuk mendukung kinerja aplikasi, penyebaran data fisik sesuai dengan penggunaannya, seperti pemrosesan transaksi *online* dan DSS (*Decision Support System*) atau pemrosesan *database* terdistribusi melalui beberapa server. Terdapat beberapa jenis – jenis replikasi diantaranya adalah sebagai berikut :

- Snapshot replication*
- Transactional replication*
- Merge replication*

## MySQL Cluster

MySQL Cluster merupakan sebuah tipe basis data (*database*) yang dapat beroperasi dalam ukuran data yang besar. 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.

Sebuah MySQL Cluster terdiri dari sekumpulan komputer, masing-masing menjalankan sejumlah proses mencakup beberapa MySQL server, *node-node* penyimpanan untuk cluster NDB, *server-server* manajemen dan program-program pengakses data yang khusus. Semua program-program tersebut bekerja bersama-sama untuk membentuk MySQL Cluster. Ketika data disimpan di dalam mesin penyimpan media NDB cluster, tabel-tabel disimpan didalam *node-node* penyimpanan pada NDB Cluster. Tabel-tabel seperti itu dapat diakses secara langsung dari semua MySQL server yang lain di dalam cluster tersebut.

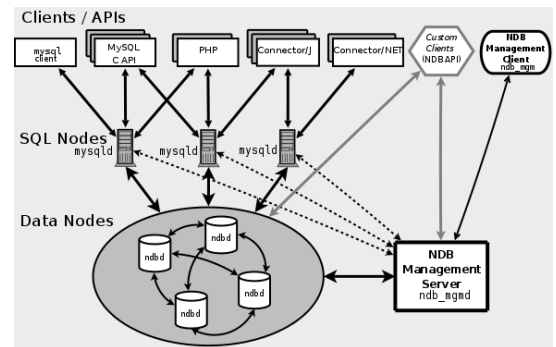
Data yang disimpan di dalam *node-node* penyimpanan pada MySQL Cluster dapat di *mirror* (dicerminkan), cluster tersebut dapat menangani kegagalan dari *node-node* penyimpanan *individual* dengan tidak ada dampak lain dari sejumlah transaksi dihentikan karena kegagalan proses transaksi.

Di dalam konfigurasi mysql cluster itu sendiri terdapat tiga *node cluster*, ketiga *node cluster* itu adalah:

- Management node (ndb\_mgmd process)*

- Data node (ndbd process)*
- SQL node (*mysqld process*)

Ketiga node tersebut dapat dilihat di gambar berikut



Gambar 3. Node dalam MySQL Cluster

## Pengaturan node

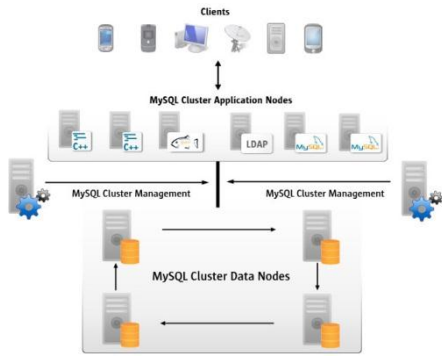
Seperti replikasi yang ada di MySQL, clustering menawarkan banyak fleksibilitas. MySQL cluster sendiri menyertakan beberapa contoh pengaturan untuk memberi anda beberapa ide tentang bagaimana cara untuk mengatur node dapat

Pengaturan node yang akan dilakukan itu sendiri berdasarkan dari adanya ketersediaan data, dan apa hardware yang diperlukan untuk memenuhi kebutuhan sistem anda. Ketika kita telah memutuskan bahwa MySQL Cluster dapat memenuhi keinginan kita, kita perlu menentukan berapa banyak mesin yang dibutuhkan untuk memenuhi kebutuhan dari sistem yang akan kita rancang dan memberikan jumlah yang tepat untuk redundansi. Berikut adalah contoh dari beberapa pengaturan node.

- Simple Arrangement*
- Robust Arrangement*
- Minimalist Arrangement*

## 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*).



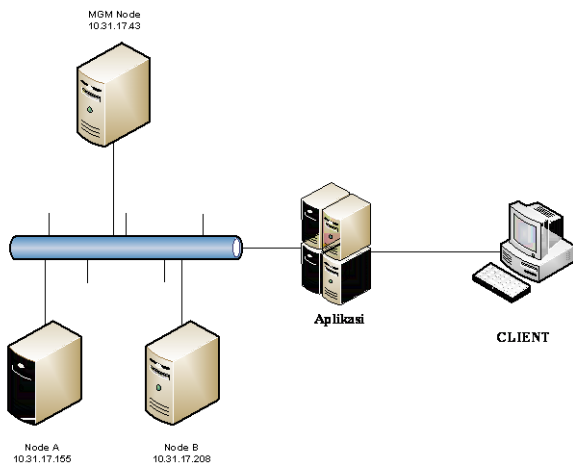
Gambar 4 Arsitektur MySQL Cluster

### III. PERANCANGAN SISTEM

Dalam perancangan ini dibutuhkan tiga *server*. *Server* pertama untuk manajemen node, *server* 2 untuk *database master* dan *server* satunya untuk *database slave*. Kedua *server master* dan *slave* memiliki fungsi yang hampir sama yaitu untuk penyimpanan *database* dari suatu sistem. Satu komputer yang lain digunakan untuk menghubungkan antara *server master* dan *server slave*, kedua *server* itu 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 *server master*, *server slave*, dan manajemen *server*.

Penggambaran sistem secara umum dapat dilihat pada gambar dibawah :



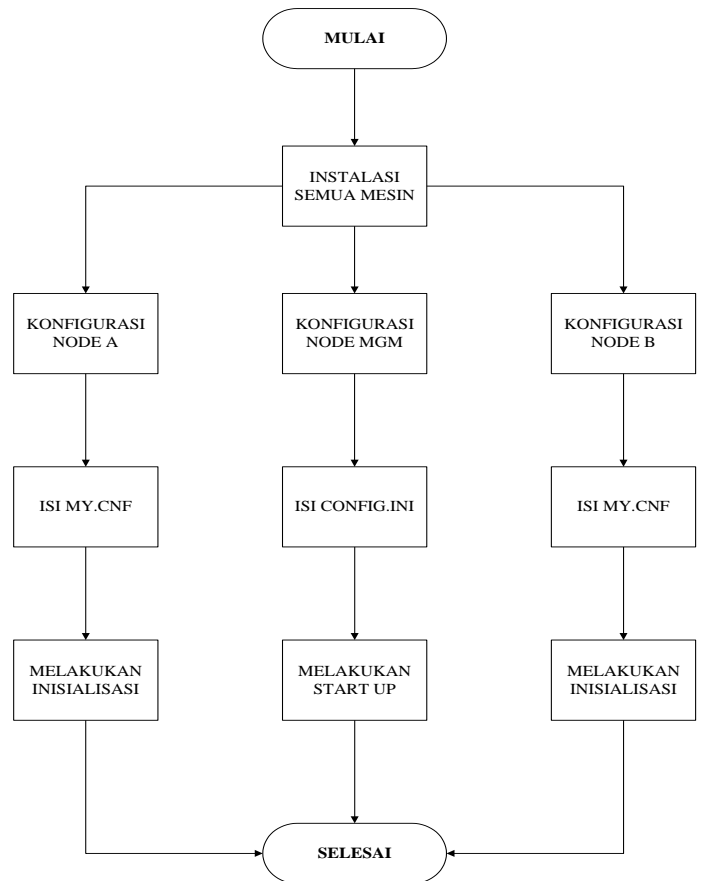
Gambar 5 Perancangan sistem secara umum

Dari gambar 5 dapat dilihat, antara node A yang berfungsi sebagai *server master* dan node B sebagai *master slave* saling terhubung. Untuk menghubungkan kedua node tersebut dibutuhkan MGM Node (*management node*). Jadi fungsi utama dari MGM Node adalah menghubungkan antara kedua node tersebut sehingga fungsi dari MySQL cluster itu sendiri dapat berjalan. Sedangkan fungsi-fungsinya antara lain adalah replikasi data, dan *high availability*.

### Perancangan Konfigurasi Perangkat Keras

Sebenarnya 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 MGM node, pengaturan konfigurasi di sisi node A atau *server master*, dan pengaturan konfigurasi di sisi node B atau *server slave*.

Berikut adalah flowchartnya :



Gambar 6 Diagram Alir Proses Konfigurasi Perangkat Keras

### IV. IMPLEMENTASI DAN PENGUJIAN

Sebelum menanamkan sistem MySQL Cluster perlu disiapkan terlebih dahulu sistem operasi *server*nya. Sistem operasi yang dipilih adalah turunan dari Debian, yaitu Ubuntu *client* 9.04 Alasan digunakan turunan dari debian yaitu Ubuntu adalah, karena memiliki repository *server* diberbagai negara termasuk Indonesia yang memungkinkan melakukan update sistem maupun paket-paket terbaru. Versi 9.04 dipilih karena versi tersebut memiliki paket-paket untuk dukungan MySQL Cluster yang lebih baik, dibanding versi sesudahnya. Sama dengan instalasi kebanyakan sistem operasi, instalasi Ubuntu menggunakan paket instalasi yang berupa CD.



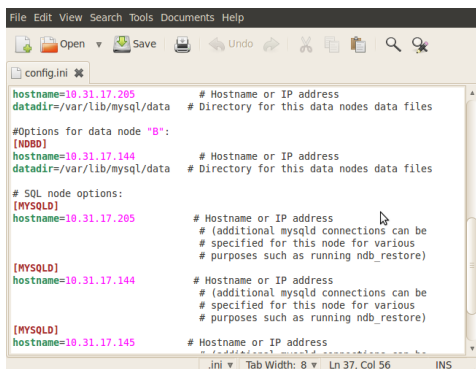
Setelah sistem operasi terinstall dengan baik, dilakukan pengaturan IP pada *server* tersebut serta daftar DNS agar *server* tersebut dapat terhubung dengan internet dan jaringan lokal. Setelah *server* dapat terhubung dengan baik ke internet dan jaringan lokal, pengaturan terhadap *server* tersebut dapat dilakukan melalui fasilitas *remote*, hal ini memungkinkan *server* dapat di akses darimana saja.

Pada Ubuntu instalasi program dapat dilakukan dengan menjalankan perintah “`apt-get install <nama program>`”. Dengan syarat file `/etc/apt/source.list` telah berisi daftar repository. Penyedia repository yang digunakan pada implementasi ini adalah `repo.undip.ac.id`.

### Instalasi dan Konfigurasi MySQL Cluster di Manajemen Node di Linux

Hal pertama yang dilakukan adalah mendownload paket `mysql cluster` yang sudah disediakan `mysql` di websitenya. Manajemen node itu sendiri saya set ipnya yaitu `10.31.17.43`. Kemudian `Copy` paket `mysql-cluster-XXX-VERSION-OS.tar.gz` ke `/home` directory anda, kemudian lakukan `'sudo -i'` atau login setara `'root'`.

Setelah kita melakukan konfigurasinya kemudian kita akan membuat `file config.ini`, file tersebut sangat penting karena apabila kita ingin menjadikan manajemen `node` harus membuat file tersebut. Itu dikarenakan karena manajemen `node` itu mengatur secara keseluruhan `mysql cluster` itu sendiri.



Gambar 7 Konfigurasi config.ini

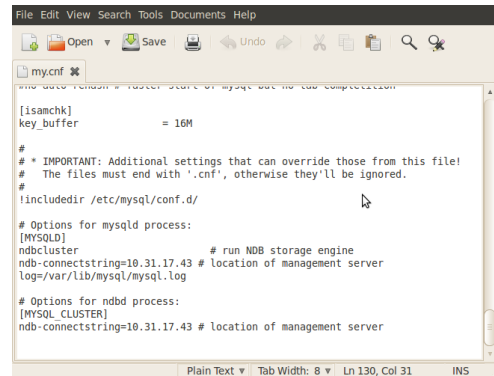
Dari isi file tersebut, `config.ini` berisi seperti `IP addres node master` dan `slave`, Memori data dan memori indeks yang dialokasikan ke setiap `node` data, jumlah replikasi, dan direktori tempat menyimpan setiap `node file` data dan data log juga direktori tempat menyimpan `node` manajemen file log-nya.

### Instalasi dan Konfigurasi MySQL Cluster di server master dan slave

Dalam konfigurasi ini `server master` atau `node A` saya set di IP `10.31.17.205` sedangkan `server slave` atau `node B` saya set di IP `10.31.17.144`. Seperti halnya instalasi di manajemen node hal pertama yang dilakukan adalah `meny-copy` paket `mysql-cluster-XXX-VERSION-OS.tar.gz` ke `/home` directory anda, kemudian lakukan `'sudo -i'` atau login setara `'root'`.

Setelah kita melakukan konfigurasinya kemudian kita mengedit konfigurasi `database` yang sudah tersedia.

Disini yang membedakan antara manajemen node, kalau di manajemen node kita membuat di data node ini kita tidak membuat tetapi hanya mengedit file `my.cnf`. File `my.cnf` itu sendiri berisi tentang `mysql`.

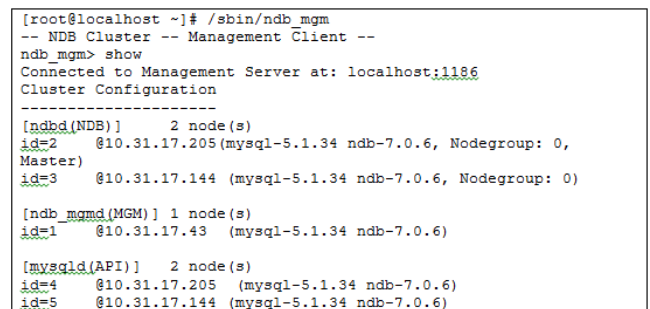


Gambar 8 Konfigurasi my.cnf

Isi dari file tersebut adalah IP dari manajemen node itu. `ndb-connectstring` itu sendiri berfungsi agar antara `server master` dan `slave` bisa terkoneksi dengan manajemen node dan akhirnya bisa melakukan teknologi `MySQL Cluster` dengan benar.

### Pengujian Mysql Cluster untuk sifat high availability

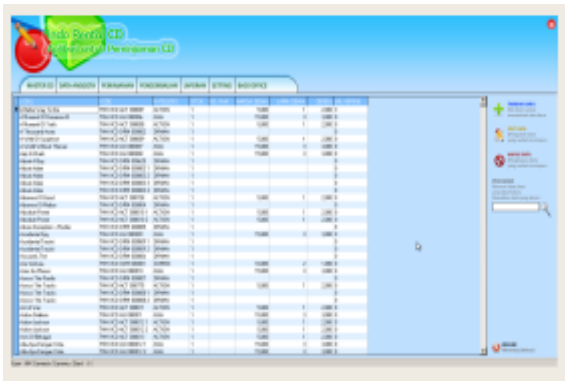
Setelah kita melakukan instalasi dan konfigurasi di tiga komputer, saatnya kita melakukan pengujian. Pertama kita menjalankan manajemen node dengan perintah `/sbin/ndb_mgmd -f /usr/local/mysql/mysql-cluster/config.ini`. Kemudian kita menjalankan node A atau node B dengan perintah `ndbd—initial`. Setelah semua dijalankan kemudian kita cek apakah `MySQL Cluster` sudah berjalan atau belum dengan perintah `ndb_mgm`. Berikut tampilannya.



Gambar 9 Tampilan manajemen client

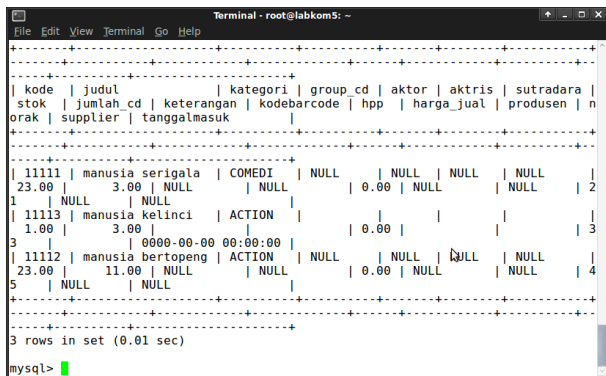
Setelah semua berjalan dengan baik sekarang kita akan menguji ketersediaan data di database dengan sebuah aplikasi. Aplikasi yang saya akan gunakan adalah persewaan `vcd`.

Setelah itu kita mengubah salah satu table di aplikasi itu agar bisa dijalankan ke `mysql cluster`. Table yang akan diganti engingnya yaitu `tcid`. Cara menggantinya cukup mudah yaitu dengan menggunakan kata `alter`. Tetapi perlu diingat bahwa di sisi `slave` dibuat dulu database yang sama dengan sisi `master`.



Gambar 10 aplikasi rental vcd

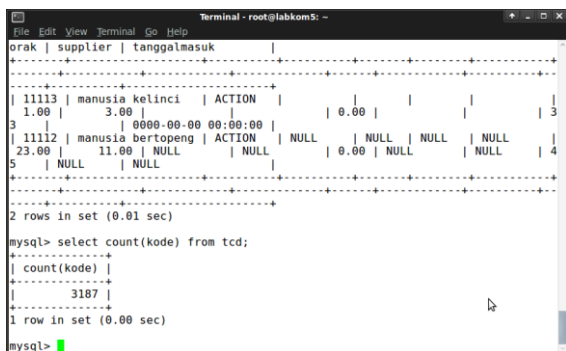
Setelah kita mengganti engine dari salah satu tabelnya kita akan mencoba menambah data di buku menggunakan aplikasi. Kita akan menambah cd dengan judul manusia sebanyak 3 film. Setelah kita buat dengan aplikasi maka kita akan check di sisi slave apakah sudah ada apa belum



Gambar 11 pengecekan di sisi slave setelah adanya penambahan data

Dari gambar 11 Dapat dilihat bahwa penambahan bukunya berhasil. Terbukti di sisi slave sudah ter record. Kemudian kita akan mengetest juga apakah mysql cluster bersifat high availability apa tidak. Kita akan mematikan salah satu node, tetapi sebelum itu kita akan mengecek jumlah barisnya terlebih dahulu.

Setelah kita menegetahui jumlah barisnya adalah 3187, maka kita akan mematikan salah satu nodenya dan mengecek apakah jumlah barisnya masih sama atau tidak.



Gambar 12 pengecekan jumlah baris data di sisi slave

Dari gambar 12 Terlihat bahwa jumlah baris dari table tcd tetap 3187 berarti sudah terbukti bahwa mysql cluster bersifat high availability.

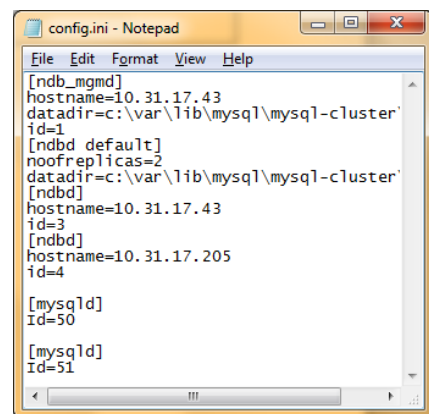
Setelah kita melakukan konfigurasi di system operasi linux selanjutnya kita akan melakukan di system operasi windows. Di sini kita akan memakai windows vista dan seven. Sama halnya dengan di linux kita juga akan mempersiapkan system operasi dengan cara menginstalnya. Setelah selesai menginstal kita lanjut dengan konfigurasi di manajemen node, node master, dan node slave.

### Instalasi dan Konfigurasi MySQL Cluster di Manajemen Node dan Server Master

Tidak seperti di linux, di konfigurasi kali ini kita menggabungkan antara manajemen node dengan server master. Itu disebabkan karena keterbatasan hardware.

Hal pertama yang dilakukan adalah mendownload paket mysql cluster yang sudah disediakan mysql di websitenya. Manajemen node dan server master itu sendiri saya set IP-nya yaitu 10.31.17.43. Kemudian ekstrak paket mysql-cluster-XXX-VERSION-OS.rar ke /home directory anda. Jangan lupa menambahkan mysql ke path system di windows dan juga firewallnya dimatikan agar mysql cluster bisa berjalan.

Setelah itu seperti yang kita lakukan di linux yaitu kita akan membuat file config.ini, file tersebut sangat penting karena apabila kita ingin menjadikan manajemen node harus membuat file tersebut. Itu dikarenakan karena manajemen node itu mengatur secara keseluruhan mysql cluster itu sendiri

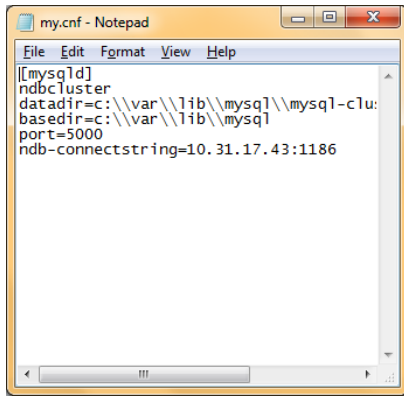


Gambar 13 Konfigurasi config.ini pada windows

Dari isi file tersebut, config.ini berisi seperti IP address node master dan slave, Memori data dan memori indeks yang dialokasikan ke setiap node data, jumlah replikasi, dan direktori tempat menyimpan setiap node file data dan data log juga direktori tempat menyimpan node manajemen file log-nya.

Karena kita menyatukan antara manajemen node dan master server atau node A maka kita juga membuat my.cnf. isi dari my.cnf adalah ip dari

manajemen node itu sendiri. Berikut isi filenya dan tampilannya.

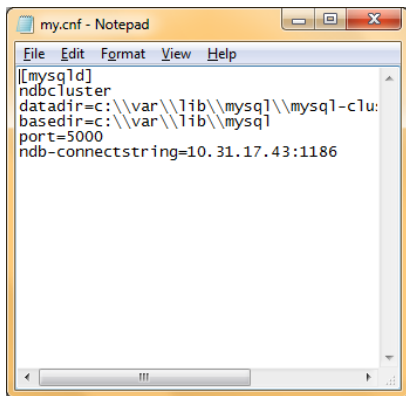


Gambar 14 Konfigurasi my.cnf pada windows

### Instalasi dan Konfigurasi MySQL Cluster di server slave

Seperti halnya instalasi dan konfigurasi di sisi server master hal pertama yang kita lakukan adalah mendownload paket mysql cluster yang sudah disediakan mysql di websitenya. Server slave itu sendiri saya set IP-nya yaitu 10.31.17.205. Kemudian ekstrak paket mysql-cluster-XXX-VERSION-OS.rar ke /home directory anda. Jangan lupa menambahkan mysql ke path system di windows dan juga firewallnya dimatikan agar mysql cluster bisa berjalan.

Disini kita hanya membuat my.cnf. Itu dikarenakan karena kita hanya akan membuat node data dan node sql saja. Isi dari my.cnf adalah ip dari manajemen node itu sendiri. Berikut isi filenya dan tampilannya.

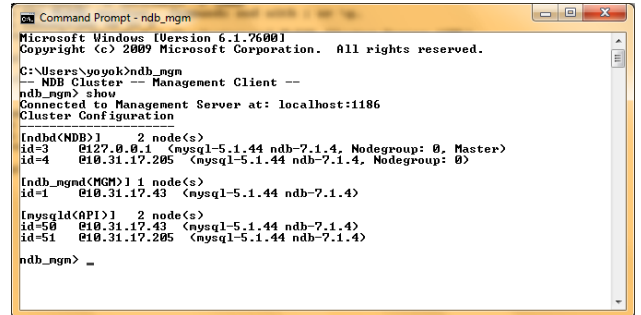


Gambar 15 Konfigurasi my.cnf pada sisi slave di windows

### Pengujian Mysql Cluster untuk sifat high availability

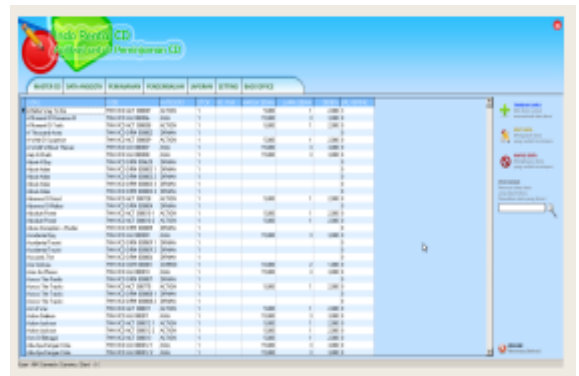
Setelah kita melakukan instalasi dan konfigurasi di tiga komputer, saatnya kita melakukan pengujian. Pertama kita menjalankan manajemen node dengan perintah start /B c:\var\lib\mysql\bin\ndb\_mgmd -f conf\config.ini --initial configdir=c:\var\lib\mysql\mysql-cluster\conf. Kemudian kita menjalankan node A dengan perintah start /B c:\var\lib\mysql\bin\ndbd -c localhost:1186 dan node B dengan perintah ndbd -c 10.31.17.43:1186 --initial. Setelah semua dijalankan kemudian kita akan menjalankan service mysql di kedua node dengan perintah start /B c:\var\lib\mysql\bin\mysqld

--defaults-file=conf\my.cnf. setelah itu kita cek apakah MySQL Cluster sudah berjalan atau belum dengan perintah ndb\_mgm. Berikut tampilannya.



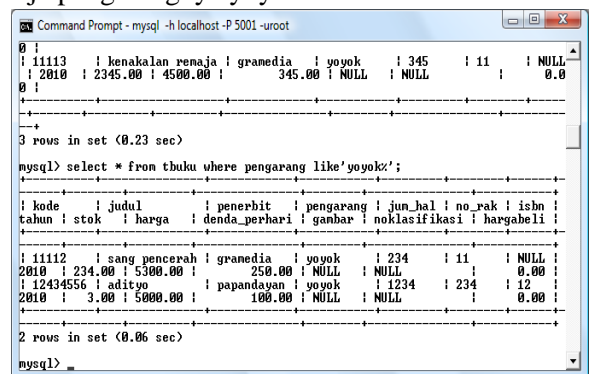
Gambar 16 Manajemen Cluster Client di Windows

Dari tampilan diatas kita bisa lihat, bahwa semua node dari server master, server slave, dan manajemen node semua terkoneksi. Itu berarti bahwa system atau teknologi mysql cluster sudah bisa berjalan dengan baik dan dapat digunakan. Kemudian kita akan menguji ketersediaan data di database dengan sebuah aplikasi. Aplikasi yang saya akan gunakan adalah persewaan buku. Berikut tampilannya



Gambar 17 Aplikasi rental buku

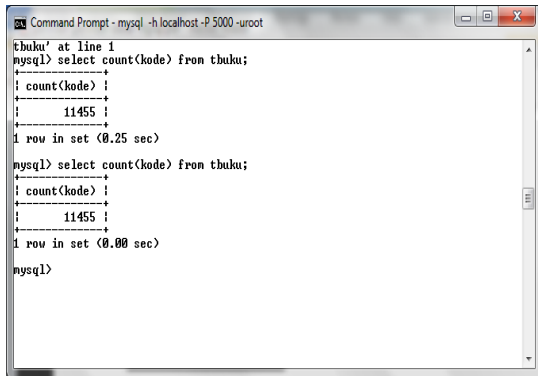
Setelah itu kita mengubah salah satu table di aplikasi itu agar bisa dijalankan ke mysql cluster. Table yang akan diganti engingnya yaitu tbuku. Cara menggantinya cukup mudah yaitu dengan menggunakan kata alter. Tetapi perlu diingat bahwa di sisi slave dibuat dulu database yang sama dengan sisi master. Kita akan mencoba menghapus salah satu buku. Kita akan menghapus judul buku kenakalan remaja pengarangnya yoyok.



Gambar 18 Pengecekan pengurangan data di sisi slave

Dari gambar 18 kita dapat melihat bahwa buku kenakalan remaja dengan pengarang yoyok juga sudah ikut terhapus.

Kemudian kita akan mengetest juga apakah mysql cluster bersifat *high availability* apa tidak. Kita akan mematikan salah satu *node*, lalu kita akan mengecek apakah datanya masih utuh atau sama.



**Gambar 19** Pengecekan jumlah baris di sisi master

Dapat terlihat bahwa tidak ada perubahan dalam ketersediaan bukunya. Dan juga kita akan mengecek jumlah bukunya. Dari gambar 4.38 Kita lihat bahwa jumlah buku masih tetap 1145. Berarti terbukti bahwa mysql cluster bersifat *high availability*.

Setelah itu kita akan melakukan konfigurasi MySQL Cluster lintas sistem operasi antara linux dan windows. Caranya adalah perpaduan antara keduanya, akan tetapi yang perlu diingat agar MySQL cluster berjalan dengan baik tempat penginstalan dan tempat penyimpanan data antara keduanya harus di tempat yang sama. Dan juga perlu diperhatikan *path system* dan firewallnya di sisi windows

**Evaluasi**

Dalam pengerjaannya, timbul masalah-masalah yang dihadapi sehingga teknologi mysql cluster itu sendiri tidak dapat berjalan dengan semestinya. Berikut adalah beberapa masalah yang ada.

1. Manajemen node tidak mau dijalankan.

Hal pertama yang di cek adalah config.ini. apakah sudah sesuai aturan apa belum. Dan yang kedua adalah cek ip dari computer yang berperan sebagai manajemen node itu sendiri. Apakah sudah sesuai dengan ip yang ada di config.ini.

2. Node master atau slave tidak mau terkoneksi atau muncul kata-kata seperti berikut: `ndbd Unable to connect with connect string: nodeid=0,10.0.0.1:1186`

Hal pertama yang kita check adalah my.cnf, apakah sudah benar konfigurasinya. Dan juga kita harus samakan antara my.cnf di sisi node dengan config.ini di sisi manajemen node. Aturan ipnya harus sama. Di windows harus dipastikan juga bahwa firewallnya harus dimatikan, dan kita harus menambahkan path mysql ke sistem

windowsnya. Tambahan ketika kita akan menjalankan ndbd di windows kita harus masuk ke folder mysql cluster. Apabila kita menjalankan selain di folder mysql cluster pasti akan gagal.

3. Ndbd tidak mau start atau berjalan sehingga akan muncul kata-kata sebagai berikut: `Cannot become daemon: /var/lib/mysql-cluster/ndb_X.pid: open for write failed: no such file or directory`

Hal yang pertama adalah membuat folder mysql-cluster terlebih dahulu. Setelah itu tempat penyimpanan program mysql clusternya antara node master dan slave harus tersimpan di tempat yang sama

4. Tidak terjadi replikasi antara node master ke node slave atau sebaliknya.

Hal pertama yang dilakukan adalah mengecek engine dari table yang akan kita gunakan. Apakah sudah menjadi ndbcluster atau belum. Apabila belum kita harus mengganti engine dari table yang akan kita gunakan menjadi engine ndbcluster.

**PENUTUP**

**Kesimpulan**

Dari hasil analisa dan pembahasan dapat disimpulkan bahwa :

1. Server mysql cluster dapat dilakukan di system operasi mana saja. Dan sistem operasinya dapat dikombinasikan.
2. Server mysql cluster berjalan di protocol TCP/IP.
3. Pada saat menggunakan dua sistem operasi yang berbeda perlu diperhatikan tempat penyimpanan mysql clusternya sendiri harus di tempat yang sama.
4. Pada saat penggunaan sistem operasi windows firewall perlu dimatikan dan penambahan path mysql pada sistem agar mysql berjalan dengan baik.
5. Engine database yang digunakan dalam server mysql cluster ini menggunakan ndbcluster.
6. Konfigurasi mysql cluster ini terdiri dari tiga node, yaitu *Management node (ndb\_mgmd process)*, *Data node (ndbd process)*, *SQL node (mysqld process)*.
7. MySQL Cluster merupakan sebuah *database* yang menggunakan arsitektur *shared-nothing*

**Saran**



Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah :

1. Sistem mysql cluster tidak hanya bisa berjalan pada jaringan lokal saja. Pengembangan sistem mysql cluster ini dapat berjalan antar jaringan public yang berbeda.
2. Sistem mysql cluster perlu ditambah system load balancer agar dapat meningkatkan kinerja dari mysql cluster itu sendiri.
3. Perancangan sistem mysql cluster dengan lingkup yang lebih luas dengan jenis data yang dikhususkan kepada klien tertentu membutuhkan arsitekturs mysql cluster yang baik serta perlunya membahas tentang faktor keamanan.
4. Perlu adanya penambahan jumlah server di setiap nodenya agar lebih jelas pengamatan mysql cluster itu sendiri.

#### DAFTAR PUSTAKA

- [1] Fisk, Harrison. 2008. **MySQL Clustering**, 1st edition. MySQL Press
- [2] Davies, Alex. and Fisk, Harrison. 2006. **MySQL® Clustering**. MySQL Press
- [3] Elmasri, R. dan Navathe, S.B. 1994. **Fundamentals of Database Systems**. 2nd Edition, Prentice Hall.
- [4] Korth, H.F. dan Silberchatz. 1998. **A Database System Concept**, 3rd Editions, McGraw Hill.
- [5] Alex, Davies. 2010. **High Availability MySQL Cookbook**. Packt Publishing Ltd. Birmingham
- [6] Hodges, R. 2007. **Database High Availability and Scalability**, CTO Continuent, Inc.  
  
<URL: <http://www.pgday.it/files/scalability.pdf>>  
(Diakses pada tanggal 27 Mei 2010)
- [7] MySQL AB. 2009. **A MySQL Technical White Paper**.  
<URL: <http://www.mysql.com/why-mysql/white-papers/>>  
(Diakses pada tanggal 7 Juni 2010)
- [8] DuBois, Paul .2008. **MySQL Fourth Edition**. USA: Addison-Wesley
- [9] MySQL AB. 2009. **MySQL Cluster Evaluation Guide**.  
<URL: [http://www.mysql.com/why-mysql/white-papers/mysql\\_cluster\\_eval\\_guide.php](http://www.mysql.com/why-mysql/white-papers/mysql_cluster_eval_guide.php)>  
(Diakses pada tanggal 29 Juni 2010)



Adityo Prabowo, lahir di Semarang tanggal 7 Juni 1987. Menempuh pendidikan dasar di SD Negeri 02 Candibaru Semarang. Melanjutkan ke SLTP N 5 Semarang, Dan Pendidikan tingkat atas di SMU N 1 Semarang lulus tahun 2005. Dari tahun 2005 sampai saat ini masih menyelesaikan studi Strata-1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang, konsentrasi Komputer dan Informatika.

Menyetujui,  
Dosen Pembimbing I

Ir. Kodrat Iman Satoto M.T.  
**NIP. 19631028 1993031 002**

Dosen Pembimbing II

Maman Somantri, S.T., M.T.  
**NIP. 19740627 19999031 002**