06 Jurnal Sains Terapan : Wahana Informasi dan Alih Teknologi Pertanian
Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232

PEMBUATAN SISTEM DATABASE CLUSTER MENGGUNAKAN APLIKASI GALERA CLUSTER DI SEKOLAH VOKASI IPB UNIVERSITY

(*Development of Cluster Database System Using Galera Cluster Application at Vocational School of IPB University*)

Aep Setiawan, SSi, MSi[1], Wini Muthia Kansha[1]
[1]Program Studi Teknik Komputer, Sekolah Vokasi IPB University
Jl. Kumbang No. 14, Babakan, Bogor 16151

***E-mail* : [1]aepsetiawan@apps.ipb.ac.id, winimuthia7@gmail.com**

Diterima : 11 Juli 2021/Disetujui : 27 Oktober 2021

ABSTRAK

Server yang terdapat di Sekolah Vokasi Institut Pertanian Bogor (SV-IPB) diinstal *Modular Object-Oriented Dynamic Learning Environment* (MOODLE) yang merupakan aplikasi e-*learning* berbasis web. MOODLE membutuhkan *database* yang berfungsi sebagai sebuah sistem penyimpanan data dari berbagai aktivitas yang dilakukan. *Database server* yang tersedia di SV-IPB berupa *single database server*. SV-IPB memiliki 17 program studi, jumlah mahasiswa Sekolah Vokasi untuk tiga angkatan sekitar 6 300 mahasiswa, hal tersebut bisa menyebabkan pengaksesan layanan MOODLE dan beban yang ditanggung server menjadi tinggi. Mengatasi masalah tersebut salah satu solusinya diimplementasikan teknologi *database cluster* pada server SV-IPB serta membuat sebuah server *load balancing*. Pembuatan *database cluster* menggunakan aplikasi Galera Cluster, serta digunakan HAProxy sebagai *load balancer* yang berperan sebagai pembagi beban antar *database server* dalam *database cluster.* Ketika *database server down* maka ada *database server* lain yang dapat menggantikan tugas dari *database server* yang *down* tersebut.
Kata Kunci : *Database Cluster*, Galera Cluster, HAProxy, *Multi-master*, MySQL

ABSTRACT

The server in the Vocational School, Bogor Agricultural University (SV-IPB) is installed with Modular Object-Oriented Dynamic Learning Environment (MOODLE) which is a web-based e-learning application. MOODLE requires a database that functions as a data storage system for various activities carried out. The database server available at SV-IPB is a single database server. SV-IPB has 17 study programs, the number of Vocational School students for three batches is around 6 300 students, this can cause access to MOODLE services and the burden on the server to be high. One of the solutions to overcome this problem is to implement database cluster technology on the SV-IPB server and create a load balancing server. Creating a cluster database using the Galera Cluster application, and using HAProxy as a load balancer that acts as a load divider between database servers in the database cluster. When the database

Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232

server is down, there is another database server that can replace the task of the down database server.
Keyword: Database Cluster, Galera Cluster, HAProxy, Multi-master, MySQL

06 Jurnal Sains Terapan : Wahana Informasi dan Alih Teknologi Pertanian
Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232

PENDAHULUAN

Sekolah Vokasi IPB University adalah lembaga pendidikan tinggi dengan sistem jaringan komputer yang dilengkapi server. Sebelumnya SV-IPB menggunakan media kertas dalam sistem pembelajaran dan sistem ujian, kini sudah beralih menjadi pembelajaran *online* dan ujian *online* yang sering disebut dengan e-*learning*. *E-learning* sendiri merupakan salah satu bentuk model pembelajaran yang difasilitasi dan didukung pemanfaatan teknologi informasi dan komunikasi (Hanum 2013). LMS adalah sebuah sistem yang didesain untuk menyajikan, melacak, melaporkan dan mengatur konten pembelajaran, kemajuan siswa dan interaksi siswa (Ali 2011). Implementasi sistem *e-learning* dalam pembelajaran, server yang dimiliki oleh SV-IPB tersebut diinstal *Content Management System* (CMS). CMS adalah suatu sistem yang digunakan untuk mengelola dan memfasilitasi proses pembuatan, pembaharuan, dan publikasi *content* secara bersama (Elinawati *et al*. 2015).

Aplikasi CMS yang digunakan adalah MOODLE. MOODLE adalah sebuah nama untuk sebuah program aplikasi yang dapat mengubah sebuah media pembelajaran dalam bentuk web. Aplikasi ini memungkinkan siswa dan pengajar masuk dalam “ruang kelas digital” untuk mengakses materi-materi pembelajaran (Utami 2016).

Aplikasi MOODLE membutuhkan *database* yang berfungsi sebagai sebuah sistem penyimpanan data dari berbagai aktivitas yang dilakukan pada MOODLE, Pada saat ini *database server* yang tersedia di SV-IPB berupa *single database server*. SV-IPB memiliki 17 program studi, jumlah mahasiswa tiga Angkatan di sekolah Vokasi IPB sekitar 6 300 mahasiswa, hal tersebut menyebabkan pengaksesan layanan MOODLE menjadi tinggi, sehingga menyebabkan beban yang ditanggung server tinggi, hal tersebut dapat menyebabkan masalah. Jika *database server down*, maka layanan akan terhenti disebabkan oleh sistem yang dibangun tidak memiliki cadangan dan ketersediaan yang tinggi (*high availability*). *Database* sendiri adalah data yang terorganisir atau sistematis. sebuah *database* dapat memiliki sejumlah tabel atau relasi untuk menyimpan data. Data ini dapat diambil, dimodifikasi atau dihapus sesuai kebutuhan pengguna (Gouhar 2017). Pada umumnya penggunaan data base bisa menggunakan MySQL. MySQL merupakan suatu aplikasi yang sifatnya *open source* serta *database server* MySQL memiliki kinerja sangat cepat, *reliable*, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems* (Yuliansyah 2014).

Permasalahan tersebut diatasi dengan diterapkannya teknologi *database clustering. Database cluster* adalah kumpulan dari beberapa server yang berdiri sendiri kemudian bekerjasama sebagai suatu sistem tunggal. Hal ini secara langsung berdampak pada server *database* sebagai penyedia layanan terhadap akses data. Oleh sebab itu beban *database server* dibagi ke masing-masing *database server* dalam satu *cluster* menggunakan sebuah *load balancer* (Syamsu 2018). *Database cluster* melakukan replikasi dan sinkronisasi dari beberapa *database server* yang ada sehingga terbentuk sistem *database* yang bersifat *high available* dengan kemampuan pembagian beban. Pembagian beban

Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232

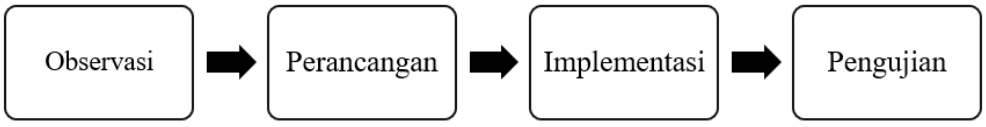
yang dimaksud yaitu pengaksesan *database* terbagi kebeberapa *database server* dalam sebuah *database cluster* menggunakan *load balancer.*

Penerapan *Database cluster* menggunakan Galera Cluster. Galera Cluster adalah aplikasi untuk *database cluster* yang menawarkan dukungan untuk *synchronous multi-master database replication based* pada *database* yang memiliki *storage engine* InnoDB (Pinheiro 2019). Galera Cluster merupakan sebuah *database server*. MySQL atau MariaDB menggunakan *Galera Replication plugin* dalam mengelola replikasi. Galera menggunakan *replication plugin Application Programming Interface* (API) untuk keperluan *multi-master* dan *synchronous replication*. API yang digunakan disebut dengan *Write-Set Replication* API atau *wsrep* API (Codership Oy 2020).

Spesifikasi server yang digunakan dengan kapasitas RAM 32 GB dan Hardisk 2 TB. Sekolah vokasi IPB University memiliki jumlah mahasiswa untuk semua Angkatan sekitar 6300 mahasiswa. Server yang digunakan cukup memadai untuk Ketika digunakan untuk mengakses server Ketika mengerjakan soal kuis atau kegiatan UTS dan UAS. Dengan database cluster diharapkan bisa membantu kinerja server Ketika diakses oleh mahasiswa.

Pengoptimalan *database cluster* perlu ditambahkan sebuah *load balancer* yang digunakan untuk membagi beban antar *database server* dalam sebuah *database cluster* menggunakan aplikasi *High Availability Proxy* (HA Proxy). *High Availability Proxy* (HAProxy) adalah *software open source* TCP/HTTP *load balancer* dan *proxying solution* yang dapat berjalan pada Linux, Solaris, dan FreeBSD. HAProxy biasa digunakan untuk meningkatkan kinerja dan *reliability* pada *database cluster* dengan cara membagi beban kerja kebeberapa *resource*/server. HAProxy juga banyak digunakan pada aplikasi *high-profile environments* seperti GitHub, Imgur, Instagram, dan Twitter (Le *et al*. 2015).

METODE PENELITIAN



Gambar 1 Prosedur kerja

Tahap observasi bertujuan untuk mengetahui permasalahan yang terjadi di lapangan, sehingga dapat direncanakan sebuah solusi untuk mengatasi masalah tersebut. Pada tahap ini juga dilakukan pengamatan untuk mendapatkan informasi perihal *resource* yang tersedia di instansi. Pada Tahap Observasi digali informasi mengenai jumlah user, kemampuan device yang digunakan, dan aplikasi MOODLE yang akan digunakan. Semakin banyak user yang akses ke aplikasi MOODLE tentunya akan menyebabkan beban server menjadi tinggi sehingga diperlukan device yang memadai untuk mengimbangi akses yang banyak supaya server tidak down.

Pada tahap perancangan dibuat sebuah rancangan untuk mengatasi permasalahan yang sudah diperoleh pada tahap observasi. Tahap perancangan

Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232

dimulai dengan membuat topologi jaringan untuk menentukan aplikasi yang digunakan.

Tahap implementasi adalah tahap pelaksanaan dari rancangan yang dibuat pada tahap perancangan. Implementasi dilakukan dengan menginstal dan melakukan konfigurasi berbagai aplikasi atau *service* sesuai dengan rancangan yang dirumuskan dalam tahap sebelumnya.

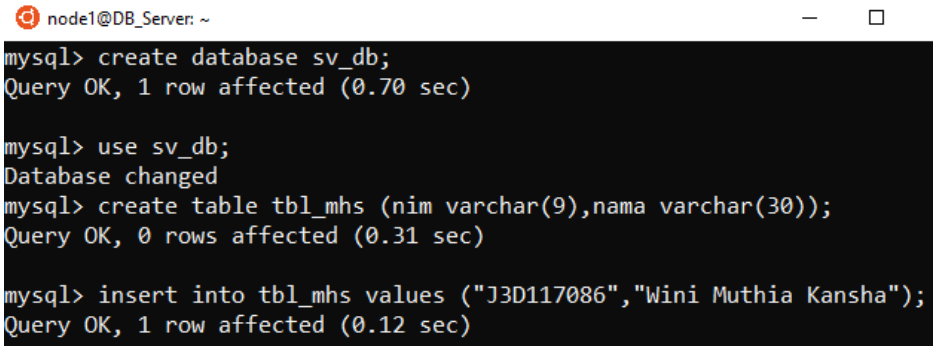
Tahap Pengujian merupakan tahap terakhir dari metode yang digunakan dalam pembuatan proyek ini. Tahap pengujian dilakukan untuk memastikan instalasi dan konfigurasi yang dilakukan sudah tepat sehingga hasil yang diperoleh sesuai dengan harapan. Pengujian tersebut antara lain memastikan semua *node database* tersinkronisasi satu sama lain serta dapat melakukan replikasi data antar *node*, menguji semua *node* dalam *database cluster* berstatus sebagai *master* dan menguji pembagian beban antar *database server* menggunakan *load balancer*.

HASIL DAN PEMBAHASAN

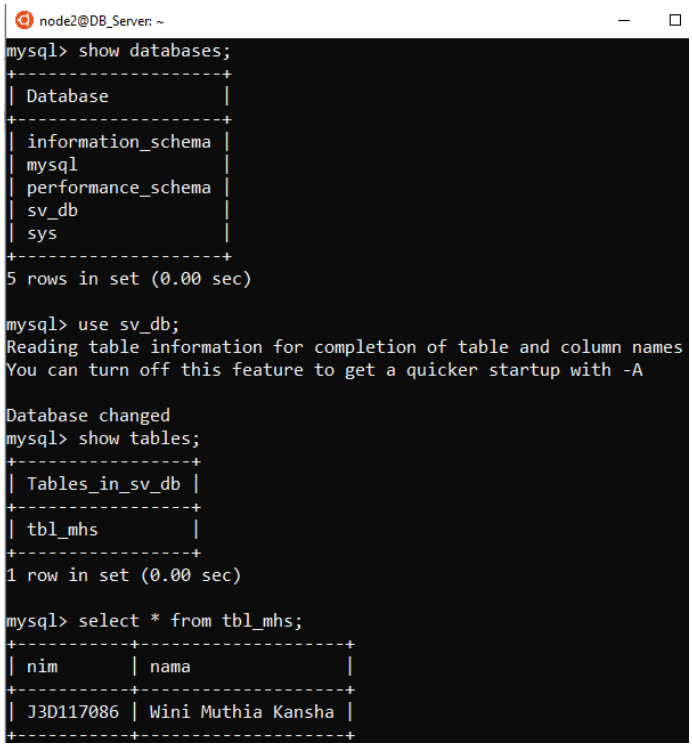
Pada bagian pembahasan pertama adalah melakukan observasi kebutuhan perangkat. Seperti yang sudah disebutkan sebelumnya sekolah vokasi IPB University mempunyai jumlah mahasiswa lebih kurang sekitar 6 300 mahasiswa. Jumlah mahasiswa tersebut sebagai client yang akan mengakses server. Kebutuhan perangkat yang digunakan harus bisa memberikan pelayanan yang baik untuk semua mahasiswa. Perangkat server yang dimiliki harus dirancang sebaik mungkin untuk bisa mengokomodasi semua client atau mahasiswa yang mengakses server. Langkah selanjutnya adalah membuat perancangan untuk konfigurasi server dimana tahap konfigurasi pertama adalah konfigurasi mengenai database server. Tahapan pertama konfigurasi database server adalah berikan alamat ip address dan install mysql. Pemberian ip address supaya database bisa diakses webserver dalam satu jaringan. Pada tahapan selanjutnya adalah konfigurasi *load balancer*. Pada tahapan ini dibagian menjadi tiga bagian yaitu node 1, node 2 dan node 3. Pembagian beban ini bertujuan jika salah satu node bermasalah maka node yang lain akan membackup. Pembahasan selanjutnya adalah akan dijabarkan proses dan hasil dari tahap pengujian antara lain memastikan semua *node database* tersinkronisasi satu sama lain serta dapat melakukan replikasi data antar *node*, menguji semua *node* dalam *database cluster* berstatus sebagai *master* dalam Galera Cluster, dan menguji pembagian beban antar *database server* menggunakan *load balancer* menggunakan HAProxy.

Pengujian ini dilakukan dengan cara membuat sebuah *database*, *table*, dan *record* pada salah satu *node*. Pada Gambar 2, dilakukan pengujian dengan membuat *database*, *table*, dan *record* pada *node* 1.

Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232



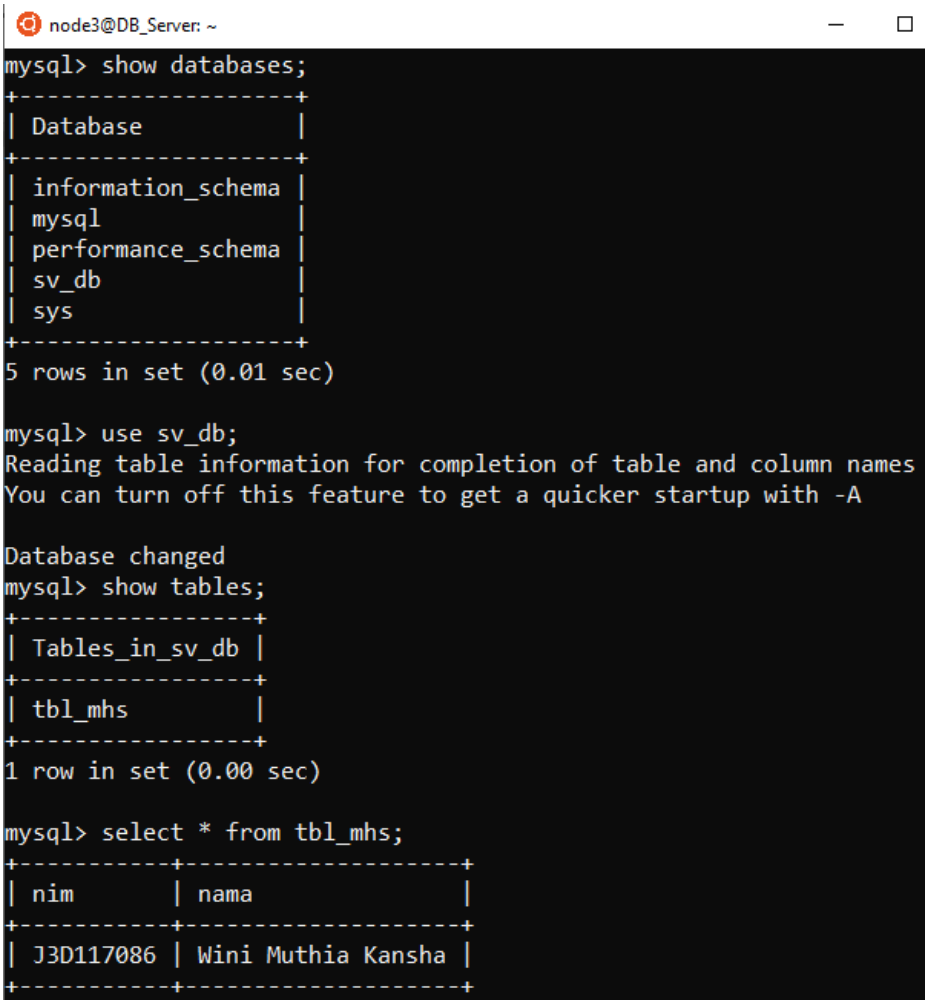
Gambar 2 *Insert* data pada *node* 1



Gambar 3 Hasil pada *node* 2

Gambar 3 menunjukkan bahwa data yang sebelumnya dimasukan pada *node* 1 telah terbuat secara otomatis pada *node* 2. Hal tersebut dibuktikan pada saat diketikan perintah *show databases*, *database* bernama sv_db terdapat dalam daftar kumpulan *database* dalam MySQL. Pada saat diketikan perintah *show tables*, tabel bernama tbl_mhs telah terbuat secara otomatis, begitu juga dengan *record* yang berisikan NIM dan nama telah berhasil dibuat secara otomatis. Pengecekan *record* dapat dilakukan dengan perintah select * from tbl_mhs. Data yang masuk pada *node* 2 telah sesuai dengan data yang dimasukan sebelumnya pada *node* 1.

Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232



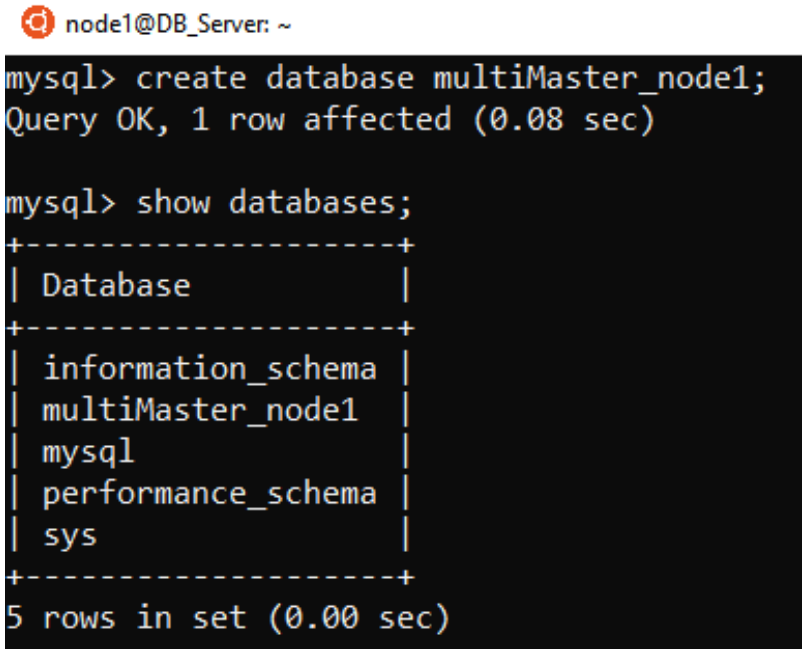
Gambar 4 Hasil pada *node* 3

Gambar 4 menunjukan bahwa data yang dimasukan pada *node* 1 telah otomatis dibuat pada *node* 3. Hal yang terjadi pada *node* 2 dan *node* 3 merupakan bukti bahwa setiap *node* dalam *database cluster* telah mampu melakukan replikasi dan sinkronisasi satu sama lain, selain itu hal tersebut merupakan bukti bahwa pembuatan *database cluster* telah berhasil dilakukan.

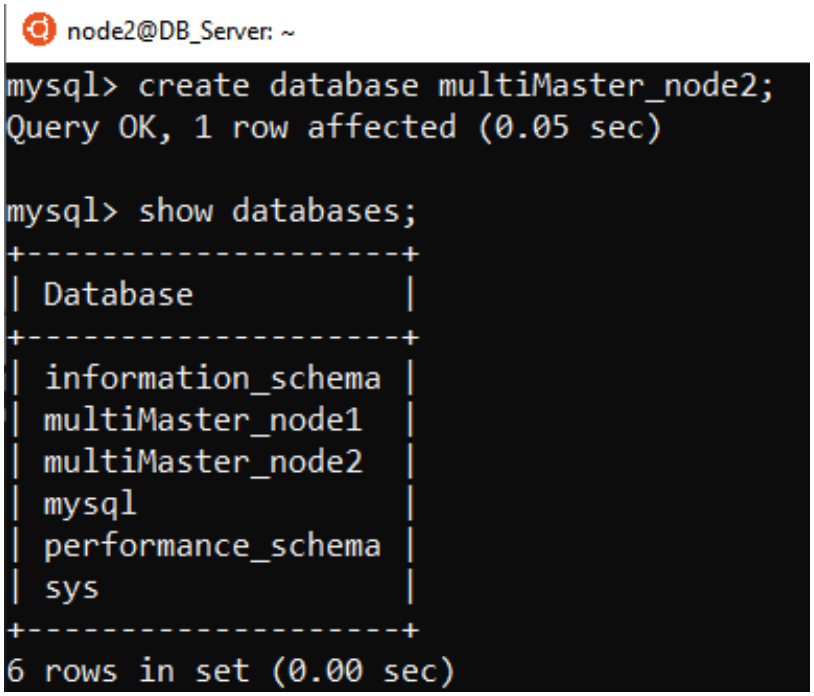
Multi-master menandakan bahwa setiap *node* dalam *database cluster* berstatus sebagai *master*. *Database server* dengan status m*aster* dapat melakukan perubahan data dalam *database* atau dalam arti lain *master* dapat melakukan penulisan atau *write* tidak hanya melakukan pembacaan atau *read* saja. Pengujian ini dilakukan untuk membuktikan bahwa masing-masing *database server* berstatus *master*.

Dilakukan pembuatan *database* pada *node* 1 seperti pada Gambar 5. Pada *node* 2 dan *node* 3 dilakukan juga pembuatan *database* seperti yang terlihat pada Gambar 6 dan Gambar 7. Keberhasilan penulisan *database* dapat dilihat pada respons MySQL yang bertuliskan *Query OK* dan *database* tersebut telah masuk ke dalam daftar *database* yang ditunjukkan dengan perintah *show databases*. Pada Gambar 5, Gambar 6, dan Gambar 7 membuktikan bahwa setiap *node* dalam *database cluster* dalam proyek ini berstatus sebagai *master*, sebab semua *node* dapat melakukan perubahan data.

Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232

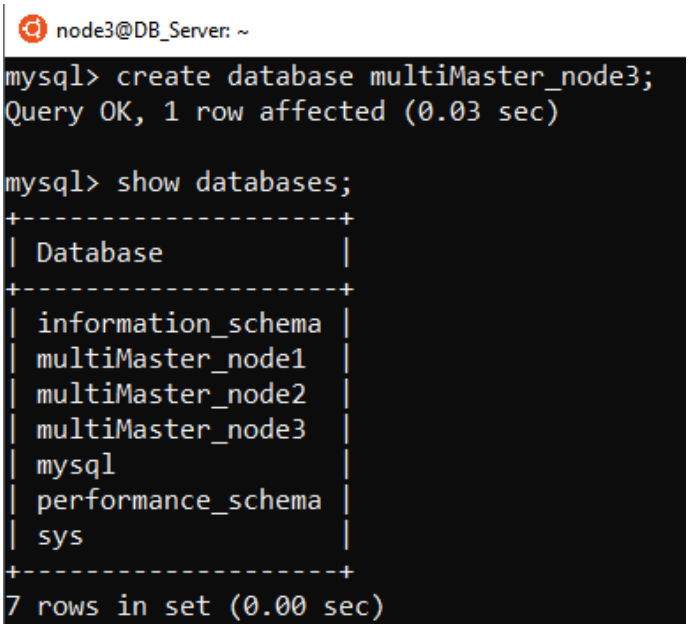


Gambar 5 Penulisan pada *node* 1



Gambar 6 Penulisan pada *node* 2

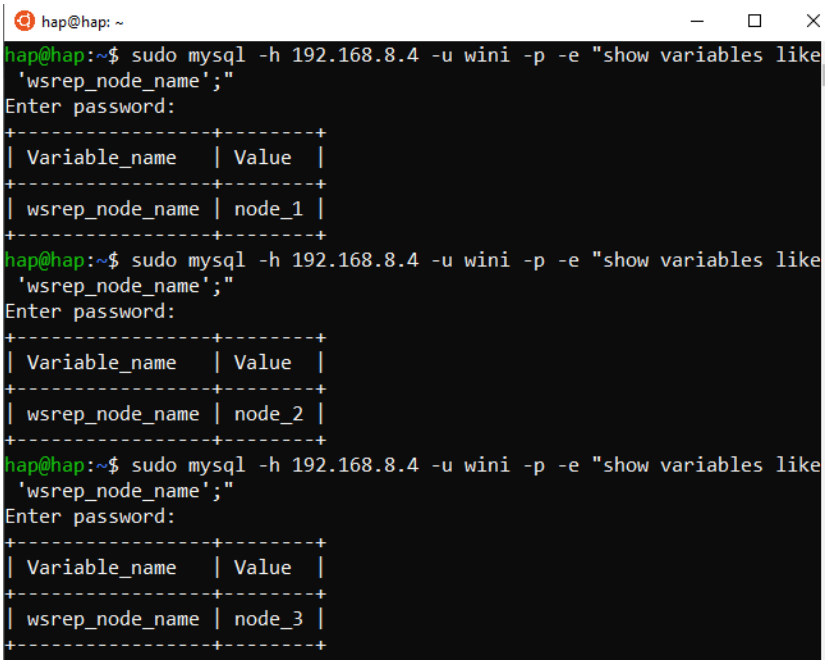
Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232



Gambar 7 Penulisan pada *node* 3

Instalasi mysql-client pada *client* diperlukan untuk melakukan proses pengujian *load balancing*. Pada pengujian *load balancing*, mysql-client dipasang pada server HAProxy. Penginstalan mysql-client dilakukan dengan perintah sudo apt-get install mysql-client. Pengaksesan *database cluster* dilakukan melalui *client* menggunakan mysql-client dengan perintah sudo mysql -h 192.168.8.4 -u wini -p -e “show variables like ‘wsrep_node_name’;” seperti yang terlihat pada Gambar 8. Pada Gambar 8 dibuktikan bahwa HAProxy dapat melakukan pembagian beban pada *database cluster*. Hal tersebut dapat terlihat di kolom *value* pada Gambar 8. Nilai pada *value* tersebut selalu berubah secara bergantian antar *node* yang terdapat dalam *database cluster*, baik *node_1*, *node_2*, dan *node_3*.

Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232



Gambar 8 Pengujian *load balancing* menggunakan HAProxy

SIMPULAN

Berdasarkan pengujian *load balancing* yang telah dilakukan, dapat disimpulkan semua *database server* sudah tergabung ke dalam satu *cluster* serta telah mampu melakukan replikasi data antar *node* (pembagian beban), dengan demikian ketika client atau mahasiswa secara bersamaan mengakses server maka akan dibagi-bagi sehingga kinerja server menjadi lebih optimal. Status pada setiap *node* dalam *cluster* adalah *master*, sehingga perubahan pada *database* dapat dilakukan di *node* manapun lalu direplikasikan keseluruh *node* dalam *cluster*. *Multi-master* mendukung *high available*, karena bila terjadi kerusakan pada salah satu *node* maka pengaksesan *database* dapat dialihkan ke *node* lain dalam *cluster* tersebut, artinya data client atau mahasiswa masih aman tidak akan hilang. Data-data tersebut contoh nya seperti nilai-nilai ataupun bahan materi yang sudah di upload di server. Penggunaan *load balancer* bertujuan untuk mengoptimalkan *database cluster*. Pengoptimalan yang dimaksud adalah penggunaan *node* 2 dan *node* 3 tidak hanya berfungsi ketika *node* 1 *down* atau mati saja, tapi semua *node* dalam *cluster* dapat menangani *request* secara bergantian dengan sistem pembagian beban menggunakan HAProxy.

SARAN

Penambahan *Firewall* dalam sistem jaringan perlu diterapkan. Firewall ini fungsinya salah satunya adalah untuk melindungi sistem jaringan ketika ada serangan dari luar. Penambahan firewall juga tentunya akan melindungi server tempat dimana *database cluster* diterapkan, sehingga penerapan *database cluster* menjadi lebih optimal.

Vol. 11 (2) : 49 - 59 (2021)
DOI : 10.29244/jstsv.11.2.49 - 59
P-ISSN : 2088-8732|E-ISSN : 2722-5232

DAFTAR PUSTAKA

Ali IT. 2011. Analisis Hubungan Implementasi Multimedia pada *Learning Management System* terhadap Kemampuan Mahasiswa dalam Penguasaan Materi Pembelajaran. *Jurnal Sains dan Teknologi*. 10(1):1–7.

Codership Oy. (2020, April 28). *Library: Codership Oy.* Diambil kembali dari Galera Cluster Web site: https://galeracluster.com/library/galera-documentation.pdf

Elinawati S, Muhammad A, Arlis S. 2015. Perancangan Content Management System (Cms) Dengan Studi Kasus E-Bisnis Pada Toko Alya Gorden. *Jurnal KomTekInfo Fakultas Ilmu Komputer*. 2(1):79–90.

Gouhar A. 2017. Database Management System. *International Journal of Engineering Science and Computing*. 7(5):649.

Hanum NS. 2013. Keefektifan E-Learning Sebagai Media Pembelajaran (Studi Evaluasi Model Pembelajaran E-Learning SMK Telkom Sandhy Putra Purwokerto). *Jurnal Pendidikan Vokasi*. 3(1):90–102.

Le QH, Xie J, Millington D, Waniss A. 2015. Comparative Performance Analysis of PostgreSQL High Availability Database Clusters through Containment. *IJARCCE*. 4(12):526–533.

Pinheiro HS. 2019. An Eventually Perfect Failure Detector in a High-Availability Scenario.

Syamsu S. 2018. Implementasi Cluster Database Berbasis MySQL Dan Haproxy Sebagai Pembagi Beban Kerja Server. *Inspiration : Jurnal Teknologi Informasi dan Komunikasi*. 8(1):48–58. DOI:10.35585/inspir.v8i2.2459

Utami IS. 2016. Implementasi E-Learning untuk Meningkatkan Aktivitas Belajar Siswa. *Jurnal Komputer Terapan*. 2(2):169–178.

Yuliansyah H. 2014. Perancangan Replikasi Basis Data Mysql Dengan Mekanisme Pengamanan Menggunakan Ssl *Encryption*. *Jurnal Informatika*. 8(1):826–836.