Cyberspace: Jurnal Pendidikan Teknologi Informasi
Volume 4, Nomor 1, Maret 2020, 40-49

ANALISA PROSES MIGRASI MYSQL *NON-CLUSTER* KE *CLUSTER* DALAM MENANGANI *FAIL-OVER* SISTEM AKADEMIK UNIVERSITAS SYIAH KUALA

Alim Misbullah[1], Nazaruddin[2], Rasudin[3], Zulfan[4]
[1,2,3,4]Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Syiah Kuala, Darussalam, Banda Aceh, 23111, Indonesia
E-mail: misbullah@unsyiah.ac.id, anzaro@unsyiah.ac.id, rasudin@unsyiah.ac.id, zulfan@unsyiah.ac.id

Abstract

The number of academic data which increase rapidly in higher education institutions need to be managed effectively so that the requesting data by academic information system become faster and easier. One of Database Management System (DBMS) technology that is still often used in higher education institutions is MySQL. Generally, the higher institutions use MySQL application on single server to process query in case overload access in peak hours can cause academic information system unstable and timeout. This research is aimed to analyze migration process of MySQL non-cluster to MySQL cluster to solve fail-over of academic system. The optimal configuration is obtained on MySQL cluster in case can be used widely on any MySQL server. In addition, load- balancing is also tested in this experiment to distribute request among MySQL server.

Keywords: *MySQL cluster, academic data, information system, migration, access time*

Abstrak

Jumlah data akademik yang terus meningkat pada institusi perguruan tinggi perlu dikelola dengan efektif sehingga pengaksesan data melalui sistem informasi akademik menjadi lebih mudah dan cepat. Salah satu teknologi *Database Management System* (DBMS) yang masih sering digunakan pada institusi perguruan tinggi adalah MySQL. Umumnya, aplikasi MySQL yang digunakan untuk memproses *query* dari pengguna masih bersifat tunggal pada sebuah server sehingga *overload* akses pada waktu tertentu dapat menyebabkan sistem informasi menjadi tidak stabil bahkan sering mengalami *timeout*. Penelitian ini bertujuan untuk menganalisa proses migrasi MySQL *non-cluster* ke MySQL *cluster* dalam menangani *fail-over* sistem akademik. Hasil penelitian yang diperoleh yaitu berupa dokumen konfigurasi yang tepat pada MySQL *cluster* sehingga dapat digunakan secara umum pada berbagai MySQL server. Selain itu, *load-balancing* juga diuji dalam penelitian ini untuk mendistribusikan *request* antara *MySQL server*.

Kata Kunci: *MySQL cluster, data akademik, sistem informasi, migrasi, waktu akses*

1. Pendahuluan

Beberapa tahun terakhir, peningkatan jumlah data yang begitu pesat telah menjadi perhatian penting pada setiap instansi pemerintah maupun swasta. Pada akhir tahun 2019, jumlah data di dunia telah mencapai 40 *zettabytes* yang terdiri dari data terstruktur dan

tidak terstruktur. Diperkirakan, data tersebut akan terus meningkat sampai dengan 175 *zettabytes* pada tahun 2025 [1]. Perguruan tinggi merupakan salah satu instansi yang merasakan dampak secara langsung dari perkembangan data tersebut.

Data akademik yang dikelola oleh perguruan tinggi semakin meningkat setiap tahunnya. Pada Universitas Syiah Kuala, jumlah mahasiswa aktif sampai tahun 2020 berjumlah hampir 30 ribu mahasiswa yang tersebar di 13 fakultas [2]. Umumnya, mahasiswa akan mengambil 16 mata kuliah dalam dua semester atau setahun yang menyebabkan jumlah *record* menjadi semakin banyak. Penambahan jumlah data mahasiswa dan pegawai pada perguruan tinggi tersebut menyebabkan pengaksesan sistem akademik yang tersedia menjadi semakin lambat ketika diakses pada waktu bersamaan.

Sistem basis data yang digunakan pada perguruan tinggi seperti Universitas Syiah Kuala masih bersifat tunggal, artinya hampir semua sistem informasi akademik masih mengakses satu server yang menyimpan basis data sehingga *overload* sangat mungkin terjadi. Pengelolaan data akademik perlu dilakukan dengan baik sehingga memungkinkan adanya *load-balancing* pada sistem basis data. Selain itu, *fail-over* juga menjadi bagian penting yang perlu diterapkan ketika *request* yang ditangani semakin banyak. Penelitian ini dilakukan oleh menganalisa proses migrasi *MySQL non-cluster* ke *MySQL cluster* dan menguji *load-balancing* serta *fail-over* menggunakan *HAProxy* pada *server MySQL cluster*.

2. Kajian Pustaka

A. *Database Management System (DBMS)*

Database Management System (*DBMS*) merupakan sebuah paket *software* yang didesain untuk membuat, memanipulasi, menerima, dan mengatur data dalam sebuah *database*. Sebuah *DBMS* umumnya dapat memanipulasi data, format data, nama tabel, *record* dan *file* strukturnya. *DBMS* digunakan oleh pengguna untuk mempermudah pengaturan data dalam sebuah *database* [3].

B. *Cluster*

Cluster merupakan kumpulan beberapa komputer yang saling terhubung melalui jaringan yang bekerja dengan cepat secara bersamaan sehingga kumpulan tersebut dapat dilihat sebagai sistem komputer tunggal. Sistem *cluster* dapat diterapkan menggunakan metode *shared-nothing* dan *shared-disk*. Metode *shared-disk* memanfaatkan media penyimpan secara bersana-sama. Sedangkan, metode *shared-nothing* digunakan untuk mengatasi keterbatasan yang ada pada metode *shared-disk*. Pada metode *shared-nothing*, setiap komputer (*server*) akan memiliki media penyimpan (*disk*), namun sistem *cluster* dapat mengontrol dengan baik setiap media penyimpan tersebut sehingga memungkinkan adanya transaksi antar komputer. Salah satu *DBMS* yang menggunakan metode *shared-nothing* adalah *MySQL Cluster* [4].

C. *MySQL*

MySQL merupakan salah satu *DBMS* yang gratis dan sangat popular digunakan oleh berbagai instansi dan industri. *MySQL* digunakan untuk membangun *database* untuk berbagai macam *website* dan aplikasi [5]. Ada beberapa macam edisi *MySQL* yang dirilis pada *website* resminya[1] seperti *MySQL Enterprise Edition, MySQL Standard Edition, MySQL Classic Edition*, dan *MySQL Cluster CGE*. Edisi yang dirilis tersebut mempunyai

[1] https://www.mysql.com/products/

kelebihan yang berbeda-beda, namun masih memiliki fungsi yang sama. Salah satu edisi *MySQL* yang akan digunakan pada penelitian ini adalah *MySQL Cluster CGE*.

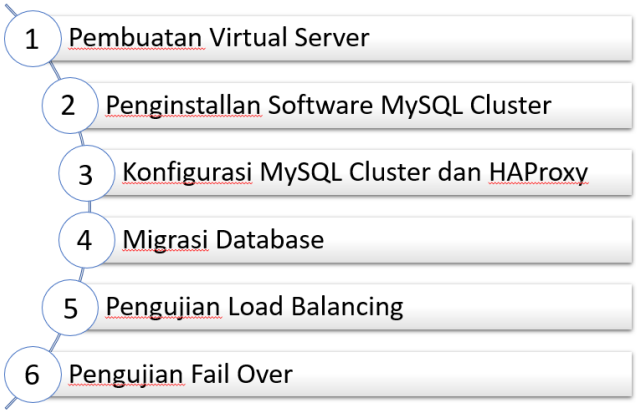
D. *High-Availability Proxy* (*HAProxy*)

HAProxy merupakan salah satu *software* gratis yang dikembangkan oleh HAProxy Technologies LLC yang digunakan sebagai *load balancer proxy* untuk aplikasi *Transmission Control Protocol* (*TCP*) dan *Hypertext Transfer Protocol* (*HTTP*) [6]. *HAProxy* dapat dimanfaatkan untuk meningkatkan performa *server* dengan mendistribusikan beban kerja (*workload*). Performa yang dapat ditingkatkan yaitu termasuk meminimalisir waktu merespon (*response time*) dan meningkatkan hasil. *HAProxy* sering digunakan pada *website* yang memiliki banyak pengunjung dan trafik yang tinggi seperti GitHub dan Twitter.

Salah satu fitur yang dimiliki oleh *HAProxy* adalah *load balancing* masing-masing pada layer 4 dan 7 yaitu *TCP* dan *HTTP*. Fitur ini memungkinkan *server* untuk membagi *workload* jika diakses oleh banyak pengguna dalam waktu bersamaan. Saat ini, *HAProxy* tersedia pada banyak distribusi Linux yang merupakan sistem operasi *open source* yang paling banyak digunakan sebagai *server* [7].

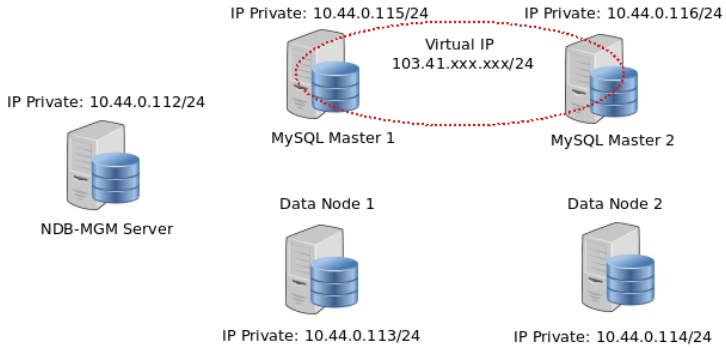
3. Metode Penelitian

Pada penelitian ini, tahapan yang dilakukan dimulai dari pembuatan *server virtual* sampai dengan pengujian *fail over* seperti yang terlihat pada Gambar 1. Pada tahapan pertama, *virtual server* dibuat dari mesin *server* fisik yang akan digunakan untuk pengujian *MySQL Cluster*. Jumlah *virtual server* yang dibuat berjumlah 5 (lima) yang terdiri dari 1 (satu) *MySQL management cluster server*, 2 (dua) *MySQL cluster master server* dan 2 (dua) *MySQL cluster data node server*. Spesifikasi *virtual server* yang digunakan dalam penelitian ini dirangkum pada Tabel 1.



Gambar 1. Skema Metode Penelitian

Semua *virtual server* yang digunakan dalam penelitian ini terhubung satu sama lain menggunakan alamat *IP* (*Internet Protocol*) *private* agar keamanan dari *server* tersebut lebih terjamin. Topologi *virtual server* yang digunakan dapat dilihat pada Gambar 2. Pada Gambar 2, terdapat *virtual IP* yaitu sebuah *IP* publik yang dimiliki oleh *server MySQL Master 1* dan *MySQL Master 2* secara sekaligus sehingga dapat digunakan untuk menerima permintaan secara bergiliran. *Virtual IP* tersebut dikonfigurasi pada kedua *server* yang akan diinstall *HAProxy*.



Gambar 2. Topologi *MySQL Cluster*

Tahapan kedua adalah penginstallan *software MySQL Cluster* yang diunduh dari situs resmi *MySQL*[2]. Meskipun sudah ada versi terbaru untuk *MySQL Cluster* di situs resminya, namun yang digunakan adalah versi 7.6.13 karena kestabilan dari versi terbaru masih belum rilis ketika penelitian ini dilakukan. Terdapat 3 jenis paket *software* yang akan diinstall pada *server* berbeda seperti yang ditampilkan pada Tabel 2.

Table 1. Spesifikasi *Virtual Server*

Name Spesifikasi	Spesifikasi Satuan
CPU/Threads	Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz/10 Threads
Memory	12 GB
Harddisk	100 GB
Sistem Operasi	Ubuntu 18.04.1 LTS

Tahapan penginstallan yang pertama dilakukan adalah pada server *MySQL Cluster Management*. *Server* ini merupakan penghubung antara *MySQL Cluster Master* dan *MySQL Cluster Data Node* sehingga diperlukan konfigurasi yang berisi informasi dari kedua server tersebut seperti yang terlihat pada Gambar 3.

Table 2. Paket *Software MySQL Cluster*

Nama Server	Name Paket Software
MySQL Cluster Management	mysql-cluster-community-management-server_7.6.13-1ubuntu18.04_amd64.deb
MySQL Cluster Master	mysql-cluster_7.6.13-1ubuntu18.04_amd64.deb-bundle.tar
MySQL Cluster Data Node	mysql-cluster-community-data-node_7.6.13-1ubuntu18.04_amd64.deb

Pada konfigurasi tersebut, `[ndbd default]` memuat informasi yang diperlukan oleh *server MySQL Cluster Data Node* dalam menangani proses penyimpanan data. Blok `[ndb_mgmd]` menampilkan informasi mesin yang digunakan sebagai *MySQL Cluster Management Server*, blok `[ndbd]` menampilkan informasi mesin yang digunakan sebagai *MySQL Cluster Data Node*, dan blok `[mysqld]` berisi informasi mesin yang digunakan sebagai *MySQL Cluster Master*.

```
File: /var/lib/mysql-cluster/config.ini
```

[2] https://dev.mysql.com/downloads/cluster/7.6.html

```
[ndbd default]
NoOfReplicas=2
MaxNoOfAttributes=5000000
MaxNoOfOrderedIndexes=10000
DataMemory=3G
NoOfFragmentLogFiles=50
TimeBetweenLocalCheckpoints=2

[ndb_mgmd]
hostname=10.44.0.112
datadir=/var/lib/mysql-cluster

# Data Node 1
[ndbd]
hostname=10.44.0.113
NodeId=2
datadir=/usr/local/mysql/data

# Data Node 2
[ndbd]
hostname=10.44.0.114
NodeId=3
datadir=/usr/local/mysql/data


[mysqld]
# SQL node 1
hostname=10.44.0.115

[mysqld]
# SQL node 2
hostname=10.44.0.116
```

Gambar 3. Konfigurasi *MySQL Cluster Management Server*

Pada sisi *MySQL Cluster Data Node*, konfigurasi yang dilakukan adalah menghubungkan mesin *data node* ke mesin *management server* menggunakan *IP address* dari mesin *management server* seperti yang terlihat pada Gambar 4. Konfigurasi yang digunakan pada mesin *MySQL Cluster Master* diperlihatkan pada Gambar 5. Blok `[mysql_cluster]` menampilkan informasi *IP address* yang menghubungkan *MySQL Cluster Master* ke *MySQL Cluster Management Server*.

```
File: /etc/my.cnf
[mysql_cluster]
ndb-connectstring=10.44.0.112
```

Gambar. 4. Konfigurasi *MySQL Cluster Data Node*

```
File: /etc/mysql/my.cnf
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/

[mysqld]
ndbcluster
sql-mode="NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,
NO_AUTO_CREATE_USER"

bind-address        = 10.44.0.115

server_id           = 1
log_bin             = /var/log/mysql/mysql-bin.log
log_bin_index       = /var/log/mysql/mysql-bin.log.index
relay_log           = /var/log/mysql/mysql-relay-bin
relay_log_index     = /var/log/mysql/mysql-relay-bin.index
```

```
expire_logs_days    = 10
max_binlog_size     = 100M
log_slave_updates   = 1
auto-increment-increment = 2
auto-increment-offset = 1

[mysql_cluster]
ndb-connectstring=10.44.0.112
```

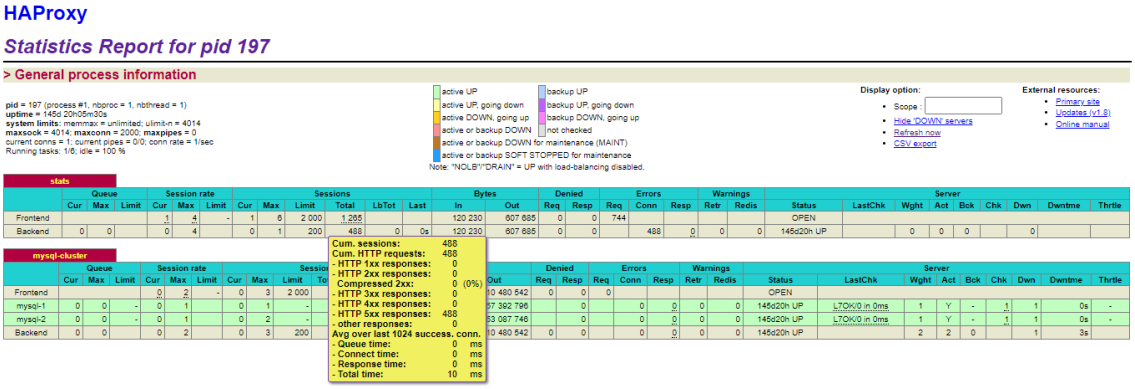
Gambar 5. Konfigurasi *MySQL Cluster Master*

Selanjutnya, *HAProxy* dinstall pada kedua *MySQL Cluster Master* sebagai *load balancer* untuk mengatur pembagian tugas antara kedua *server* ketika sedang menerima permintaan (*request*). Ketika terjadi permintaan ke *server*, maka *HAProxy* akan melihat *server* yang sedang tidak bekerja dan memberikan permintaan tersebut kepadanya. Proses ini diistilahkan dengan nama metode *Round Robin* [8]. Konfigurasi yang digunakan pada *HAProxy* ditampilkan pada Gambar 6.

```
File: /etc/haproxy/haproxy.cfg
global
    user haproxy
    group haproxy
defaults
    mode http
    log global
    retries 2
    timeout connect 3000ms
    timeout server 5000ms
    timeout client 5000ms
listen stats
    bind 103.xx.xx.xx:9999
    stats enable
    stats hide-version
    stats uri /stats
    stats auth statadmin:statadminpass
listen mysql-cluster
    bind 103.xx.xx.xx:3306
    mode tcp
    option mysql-check user haproxy_check
    balance roundrobin
    server mysql-1 10.44.0.115:3306 check
    server mysql-2 10.44.0.116:3306 check
```

Gambar 6. Konfigurasi *HAProxy*

Pada Gambar 6, terdapat 2 (dua) nama *listen* menggunakan *IP address* yang sama namun *port* berbeda. Hal ini dilakukan agar *HAProxy* dapat menangani *request* sekaligus melihat statistik dari *request* tersebut menggunakan *browser* seperti yang terlihat pada gambar 7.



Gambar 7. Tampilan statistik *HAProxy* di *browser*

Tahapan selanjutnya dari penelitian ini adalah migrasi *database* dan pengujian *load-balancing* serta *fail-over* yang akan dibahas pada bagian hasil dan pembahasan.

4. Hasil dan Pembahasan

Tahapan migrasi *database* merupakan tahapan yang awal dilaporkan pada bagian pembahasan. Migrasi *database* dilakukan dari *server MySQL Non-Cluster* ke server *MySQL Cluster* pada *database* yang sering diakses oleh pengguna di lingkungan Universitas Syiah Kuala. Migrasi *database* juga melibatkan proses pergantian *engine* pada *MySQL* yaitu *MyISAM/InnoDB* menjadi *NDBCluster* dengan menggunakan perintaha *alter*. Proses ini perlu dilakukan agar *database* tersebut dapat diakses dari *server MySQL Cluster* berbeda. Contoh tabel pada *database* yang digunakan dalam penelitian ini ditampilkan pada Tabel 3.

Table 3. Contoh Tabel pada *MySQL Cluster*

Nama Tabel	*Engine*	Version	Row Format	Rows
access_token	ndbcluster	10	Dynamic	0
batas_set_semester	ndbcluster	10	Dynamic	140
blok	ndbcluster	10	Dynamic	36
daftar_hadir_mahasiswa	ndbcluster	10	Dynamic	68
daftar_hadir_mahasiswa_copy	InnoDB	10	Dynamic	233

**Engine InnoDB* adalah contoh tabel yang belum dikonversi menjadi *ndbcluster*

Jika proses migrasi *database* dilakukan pada mesin *MySQL Cluster Master 1* dan *engine* tabelnya masih dalam bentuk *MyISAM/InnoDB* maka tabel tersebut tidak dapat diakses dari *MySQL Cluster Master 2*. Contoh tampilan perbedaan status tabel antara *MySQL Cluster Master 1* dan *MySQL Cluster Master 2* diperlihat pada Gambar 8.

```
mysql> show table status;
+---------------------------+------------+---------+------------+---------+----------------+-------------+--------------
---+--------------+-----------+----------------+---------------------+---------------------+---------------------+-----------
---+----------+----------------+---------+
| Name                      | Engine     | Version | Row_format | Rows    | Avg_row_length | Data_length | Max_data_leng
th | Index_length | Data_free | Auto_increment | Create_time         | Update_time         | Check_time          | Collation
   | Checksum | Create_options | Comment |
+---------------------------+------------+---------+------------+---------+----------------+-------------+--------------
---+--------------+-----------+----------------+---------------------+---------------------+---------------------+-----------
---+----------+----------------+---------+
| access_token              | ndbcluster |      10 | Dynamic    |       0 |              0 |           0 |
 0 |            0 |         0 |           NULL | NULL                | NULL                | NULL                | latin1_swedish_c
i  |     NULL | partitioned    |         |
| batas_set_semester        | ndbcluster |      10 | Dynamic    |       3 |            140 |      131072 |
 0 |            0 |         0 |           NULL | NULL                | NULL                | NULL                | latin1_swedish_c
i  |     NULL | partitioned    |         |
| blok                      | ndbcluster |      10 | Dynamic    |      33 |             36 |      131072 |
 0 |            0 |         0 |             52 | NULL                | NULL                | NULL                | latin1_swedish_c
i  |     NULL | partitioned    |         |
| daftar_hadir_mahasiswa    | ndbcluster |      10 | Dynamic    | 2011677 |             68 |   360710144 |
 0 |            0 |         0 |        2017862 | NULL                | NULL                | NULL                | latin1_swedish_c
i  |     NULL | partitioned    |         |
| daftar_hadir_mahasiswa_copy | InnoDB     |      10 | Dynamic    |    6806 |            233 |     1589248 |
 0 |            0 |   4194304 |           6940 | 2020-06-26 08:03:49 | 2020-06-26 08:03:49 | NULL                | latin1_swedish_c
i  |     NULL |                |         |
| daftar_skill              | ndbcluster |      10 | Dynamic    |      11 |             32 |      131072 |
 0 |            0 |         0 |             16 | NULL                | NULL                | NULL                | latin1_swedish_c
i  |     NULL | partitioned    |         |
| data_jadwal               | ndbcluster |      10 | Dynamic    |       0 |              0 |           0 |
 0 |            0 |         0 |              1 | NULL                | NULL                | NULL                | latin1_swedish_c
i  |     NULL | partitioned    |         |
| data_pengajar             | ndbcluster |      10 | Dynamic    |    6479 |            228 |     2621440 |
 0 |            0 |         0 |           7101 | NULL                | NULL                | NULL                | latin1_swedish_c
```

(a) Tabel pada *MySQL Cluster Master 1*

```
mysql> show table status;
+---------------------------+------------+---------+------------+---------+----------------+-------------+---------------
-+--------------+-----------+----------------+---------------------+---------------------+---------------------+----------
----------+---------+
| Name                      | Engine     | Version | Row_format | Rows    | Avg_row_length | Data_length | Max_data_length
 | Index_length | Data_free | Auto_increment | Create_time         | Update_time         | Check_time          | Collation         | Checksum | Cre
ate_options | Comment |
+---------------------------+------------+---------+------------+---------+----------------+-------------+---------------
-+--------------+-----------+----------------+---------------------+---------------------+---------------------+----------
----------+---------+
| access_token              | ndbcluster |      10 | Dynamic    |       0 |              0 |           0 |             0
 |            0 |         0 |           NULL | NULL                | NULL                | NULL                | latin1_swedish_ci |     NULL | par
titioned    |         |
| batas_set_semester        | ndbcluster |      10 | Dynamic    |       3 |            140 |      131072 |             0
 |            0 |         0 |           NULL | NULL                | NULL                | NULL                | latin1_swedish_ci |     NULL | par
titioned    |         |
| blok                      | ndbcluster |      10 | Dynamic    |      33 |             36 |      131072 |             0
 |            0 |         0 |             52 | NULL                | NULL                | NULL                | latin1_swedish_ci |     NULL | par
titioned    |         |
| daftar_hadir_mahasiswa    | ndbcluster |      10 | Dynamic    | 2011677 |             68 |   360710144 |             0
 |            0 |         0 |        2017862 | NULL                | NULL                | NULL                | latin1_swedish_ci |     NULL | par
titioned    |         |
| daftar_skill              | ndbcluster |      10 | Dynamic    |      11 |             32 |      131072 |             0
 |            0 |         0 |             16 | NULL                | NULL                | NULL                | latin1_swedish_ci |     NULL | par
titioned    |         |
| data_jadwal               | ndbcluster |      10 | Dynamic    |       0 |              0 |           0 |             0
 |            0 |         0 |              1 | NULL                | NULL                | NULL                | latin1_swedish_ci |     NULL | par
titioned    |         |
| data_pengajar             | ndbcluster |      10 | Dynamic    |    6479 |            228 |     2621440 |             0
 |            0 |         0 |           7101 | NULL                | NULL                | NULL                | latin1_swedish_ci |     NULL | par
titioned    |         |
| data_pengajar_backup      | ndbcluster |      10 | Dynamic    |      10 |            228 |      131072 |             0
 |            0 |         0 |             26 | NULL                | NULL                | NULL                | latin1_swedish_ci |     NULL | par
```

(b) Tabel pada *MySQL Cluster Master 2*

Gambar 8. Tampilan tabel pada *MySQL Cluster Master*

Semua tabel pada *database* dalam penelitian ini berhasil dikonversi dari *non-cluster* menjadi *cluster* sehingga sudah dapat dilakukan pengujian untuk *load-balancing*. Pada Gambar 9, pengujian dilakukan menggunakan perintah *mysql* dengan mengirimkan *request* berupa "*show variables like 'server_id*'" ke *virtual IP* yang telah dikonfigurasi pada *HAProxy*. Terlihat bahwa nilai dari *server_id* menjadi berubah setiap dilakukan *request* secara berulang-ulang.

```
alim@ndb-sqlnode1:~$ mysql -h 103.xx.xx.xx -u haproxy_root -p -e "show
variables like 'server_id'"
Enter password:
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| server_id     | 1     |
+---------------+-------+
```

```
alim@ndb-sqlnode1:~$  mysql  -h  103.xx.xx.xx  -u  haproxy_root  -p  -e  "show
variables like 'server_id'"
Enter password:
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| server_id     | 2     |
+---------------+-------+
```

Gambar 9. Tampilan pengujian *load-balancing* dari *HAProxy*

Pengujian *fail-over* juga merupakan bagian dari analisa yang dilakukan pada penelitian ini. Skenarionya adalah dengan cara mematikan salah satu *server MySQL Cluster Master* yaitu *server* dengan nilai *server_id*-nya *2*. Pada Gambar 10, hasil pengujian terlihat bahwa *fail-over* dapat ditangani dengan baik ketika salah satu *server* dimatikan. *Request* yang dikirim hanya akan ditangani oleh *server* dengan nilai *server_id*-nya 1.

```
alim@ndb-sqlnode1:~$ mysql -h 103.xx.xx.xx -u haproxy_root -p -e "show
variables like 'server_id'"
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| server_id     | 1     |
+---------------+-------+
alim@ndb-sqlnode1:~$ mysql -h 103.xx.xx.xx -u haproxy_root -p -e "show
variables like 'server_id'"
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| server_id     | 1     |
+---------------+-------+
alim@ndb-sqlnode1:~$ mysql -h 103.xx.xx.xx -u haproxy_root -p -e "show
variables like 'server_id'"
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| server_id     | 1     |
+---------------+-------+
```

Gambar 10. Tampilan pengujian *fail-over*

5. Kesimpulan

Penelitian ini dilakukan untuk menguji dan menganalisa proses migrasi *MySQL Non-Cluster* ke *MySQL Cluster* sekaligus menggunakan *HAProxy* sebagai *load-balance* dan *fail-over*. Hasil pengujian menunjukkan bahwa *MySQL Cluster* dapat digunakan untuk menggantikan *MySQL Non-Cluster* setelah dilakukan perubahan pada *engine*. Selain itu, *HAProxy* juga sangat efektif diterapkan untuk menangani proses *load-balancing* dan *fail-over* yang terjadi ketika *request* terlalu banyak pada waktu bersamaan. Pengujian terhadap berbagai nilai pada tahapan konfigurasi juga dilakukan sehingga diperoleh dokumen konfigurasi yang optimal seperti yang ditampilkan pada penelitian ini.

Penelitian lanjutan yang mungkin dapat dilakukan adalah pengujian pada sisi aplikasi sistem akademik sehingga akan diperoleh hasil yang lebih akurat. Selain itu, penambahan *server* untuk *MySQL Cluster Master* juga perlu dilakukan untuk melihat proses *load-balancing* pada mesin yang lebih banyak.

Daftar Pustaka

[1] T. Coughlin, “175 Zettabytes By 2025”, Forbes Publishing, https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025
[2] Portal Data, “Mahasiswa Aktif”, Universitas Syiah Kuala, https://data.unsyiah.ac.id/mahasiswa-aktif
[3] S. Kedar, “Database Management System”, Technical Publications; 2009.
[4] V. Alex, “Linux cluster architecture. Sams”, 2002 Jul 1.
[5] P. DuBois, “MySQL”, Addison-Wesley Professional, ISBN: 9780768678673, 2008
[6] M. Rouse, “High-Availability Proxy (HAProxy)”, WhatIs.com, https://searchnetworking.techtarget.com/definition/HAProxy
[7] Ramirez, Nick. “Load Balancing with HAProxy: Open-source technology for better scalability, redundancy and availability in your IT infrastructure”. Independently published, 2016.
[8] Prasetijo, Agung B., Eko D. Widianto, and Ersya T. Hidayatullah. "Performance comparisons of web server load balancing algorithms on HAProxy and Heartbeat." In *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 393-396. IEEE, 2016.