

IMPLEMENTASI *CLUSTERING* DATABASE SERVER MENGUNAKAN *PGCLUSTER* UNTUK OPTIMALISASI KINERJA SISTEM BASIS DATA

Suryanto

Abstract-In the business world besides a customer service, availability and needs of data will also be crucial. With increasing time, data and user will also be increased, and the process of accessing a computer and the workload will be heavy and show. This triggers the emergence of a technology called clustering. The purpose of this research report is to build a system clustering database PostgreSQL using PGCluster and to test the performance by looking and comparing the processing time some query operations on the environment database PostgreSQL cluster and non-cluster. The methodology used in this study is to analyze the need that have been set by the company in the form of a fast response from the server, availability of data, and a hot backup on the database. Then the authors designed the architecture and implement the system clustering database PostgreSQL. From the performance test was done, the authors have concluded that the results of testing one user makes a connection and executes a query operation on each system, the time required on non-clustered system less. While on the test results of three users connect and execute a query operation on each system simultaneously, showed that the time required from the cluster system less.

Intisari-Didalam dunia bisnis selain pelayanan terhadap pelanggan, ketersediaan dan kebutuhan akan data juga menjadi hal yang sangat penting. Seiring bertambahnya waktu, maka data dan pengguna juga akan bertambah, dan proses pengaksesan serta beban kerja komputer akan menjadi berat dan lambat. Hal ini menjadi pemicu munculnya suatu teknologi yang bernama *clustering*. Tujuan dari laporan penelitian ini adalah membangun suatu sistem *clustering* basis data PostgreSQL menggunakan PGCluster dan melakukan uji kinerja dengan melihat dan membandingkan waktu pemrosesan beberapa operasi *query* dilingkungan basis data PostgreSQL *cluster* dan *non-cluster*. Metodologi yang digunakan dalam penelitian ini adalah melakukan analisa kebutuhan yang telah ditetapkan oleh perusahaan berupa respon yang cepat dari server, ketersediaan akan data, dan adanya *hot backup* pada basis data. Kemudian penulis merancang arsitektur dan mengimplementasikan sistem *clustering* basis data PostgreSQL tersebut. Dari uji kinerja yang telah dilakukan, penulis mendapatkan kesimpulan bahwa hasil pengujian satu buah pengguna melakukan koneksi dan mengeksekusi sebuah operasi *query* pada masing-masing sistem, waktu yang diperlukan pada sistem *non-cluster* lebih sedikit.

Sedangkan pada hasil pengujian dua buah pengguna melakukan koneksi dan mengeksekusi sebuah operasi *query* pada masing-masing sistem secara bersamaan, menunjukkan bahwa waktu yang diperlukan dari sistem *cluster* lebih sedikit.

Kata kunci: *Clustering, PostgreSQL, PGCluster*

I. PENDAHULUAN

Di dalam dunia bisnis selain pelayanan terhadap pelanggan, ketersediaan dan kebutuhan akan data juga menjadi hal yang sangat penting. Seiring bertambahnya waktu, maka data dan pengguna juga akan bertambah, dan proses pengaksesan serta beban kerja komputer akan menjadi berat dan lambat. Hal ini menjadi pemicu munculnya suatu teknologi yang bernama *clustering*.

Dalam jaringan yang dibangun pada sebuah perusahaan/instansi biasanya ada *server* yang digunakan sebagai tempat penyimpanan data perusahaan yang disebut sebagai database server. Data tersebut akan diakses oleh setiap *client* yang membutuhkannya, hal ini masih terlihat normal jika *server* yang digunakan sebagai tempat penyimpanan data tidak mengalami permasalahan seperti terjadinya kerusakan secara *software* atau *hardware*. Namun permasalahan akan muncul jika *server* tersebut tiba-tiba mengalami kerusakan pada aspek *software*-nya atau terjadi kerusakan pada media penyimpanan datanya. Permasalahan semacam ini akan mengakibatkan sistem pada *server* terputus atau mati dan kemungkinan besar data-data yang dimiliki tidak terselamatkan, sehingga proses pengaksesan data pada *server* tersebut tidak berfungsi sebagaimana mestinya. Hal inilah yang menjadi pemicu utama munculnya suatu teknologi yang bernama *clustering*.

Selama ini langkah yang digunakan untuk mengatasi masalah yang diilustrasikan di atas dilakukan secara konvensional. Cara yang digunakan yaitu dengan mengganti infrastruktur perangkat server seperti mengganti *hardwre* apabila ada perangkat yang lama (yang sedang digunakan) telah rusak atau mati total. Namun hal ini membutuhkan waktu yang sangat lama, tidak efisien dan masih dapat menyebabkan gangguan terhadap kontinuitas kerja para pemakai. Bertolak dari permasalahan ini maka perlu dibangun

Program Studi Manajemen Informatika AMIK BSI Jakarta, Jl.
RS Fatmawati No. 24 Jakarta Selatan, Telp.021-75914760;
e-mail: suryanto.syt@bsi.ac.id

sebuah *clustering database server* untuk optimalisasi jaringan perusahaan.

Dengan mengimplementasikan *web server cluster* dengan skema *load balancing* dapat meningkatkan performa system dengan lebih baik dibandingkan dengan menggunakan *web server* tunggal serta dapat juga memberikan *availabilitas* sistem yang terjaga dan skalabilitas cukup untuk dapat melayani setiap *request* dari pengguna.

Dengan menggunakan sistem (*clustering*) ini, maka *database* akan direplikasi pada server lain, sehingga akan memiliki server-server yang identik. Selain itu, sistem *clustering* ini juga dapat membagi beban kerja dari server. Salah satu teknologi *clustering* yang efektif dan dapat memberikan ketersediaan akan data saat ini adalah PGCluster. PGCluster ini merupakan sistem replikasi *multi-master* yang didesain untuk basis data *opensource PostgreSQL*.

Sejalan dengan permasalahan diatas maka ditentukanlah tujuan dari penelitian ini adalah membangun suatu sistem *clustering* basis data PostgreSQL menggunakan PGCluster dan melakukan uji kinerja dengan melihat dan membandingkan waktu pemrosesan beberapa operasi *query* dilingkungan basis data PostgreSQL *cluster* dan *non-cluster*.

II. KAJIAN LITERATUR

Penelitian mengenai *clustering* dengan skema *load balancing* sudah banyak dilakukan, diantaranya menjelaskan bahwa virtual server mempunyai skalabilitas dan ketersediaan yang tinggi yang dibangun diatas sebuah *cluster* dari beberapa *real server* dan *load balancer* akan saling terkoneksi baik dalam jaringan local kecepatan tinggi atau terpisah secara geografis. *Load balancer* dapat mengirim permintaan-permintaan ke *server* yang berbeda dan membuat *parallel service* dari sebuah *cluster* pada sebuah alamat IP tunggal dan meminta pengiriman dapat menggunakan teknologi IP *load balancing* atau level aplikasi teknologi *load balancing*. Oklilas dan Lukitasari (2010:31)

Ketersediaan sistem *cluster* memiliki pengertian bahwa sebuah sistem yang dibangun akan terus menjaga kontinyuitas sistemnya dengan cara melakukan *backup* sistem jaringan *server* komputer (*redundancy*), *failover* apabila ada salah satu perangkat jaringan *server* yang bermasalah atau mati total maka *load balancing* akan membagi jalur akses dari *client* untuk mengakses ke *server* sehingga pemrosesan data maupun aplikasi dapat dilakukan lebih cepat.

Untuk itulah penulis melakukan penelitian ini untuk sistem (*clustering*) ini, maka *database* akan direplikasi pada server lain, sehingga akan memiliki server-server yang identik. Dan kegunaan lain dari sistem *clustering* ini adalah untuk membagi beban kerja dari server sehingga dapat meningkatkan kinerja sistem. Salah satu teknologi *clustering* yang efektif dan dapat memberikan ketersediaan akan data saat ini adalah PGCluster. PGCluster ini merupakan sistem replikasi *multi-master* yang didesain untuk basis data *opensource PostgreSQL*.

2.1. Jaringan Komputer

Jaringan komputer adalah “gabungan antara teknologi komputer dan teknologi telekomunikasi. Gabungan teknologi

ini menghasilkan pengolahan data yang dapat didistribusikan, mencakup pemakaian database, software aplikasi dan peralatan hardware secara bersamaan”. Dua unit komputer dikatakan terkoneksi apabila keduanya bisa saling bertukar data/informasi, berbagi resource yang dimiliki, seperti file, printer, media penyimpanan (hardisk, floppy disk, cd-rom, flash disk, dan lain-lain). Data yang berupa teks, audio maupun video, bergerak melalui media kabel atau tanpa kabel (wireless) sehingga memungkinkan pengguna computer dalam jaringan komputer dapat saling bertukar file atau data, mencetak pada printer yang sama dan menggunakan hardware atau software yang terhubung dalam jaringan bersama-sama [3].

Database Cluster

Dalam ilmu komputer dan jaringan komputer, cluster adalah sekumpulan komputer (umumnya server jaringan) independen yang beroperasi serta bekerja secara erat dan terlihat oleh klien jaringan seolah-olah komputer-komputer tersebut adalah satu buah unit komputer.

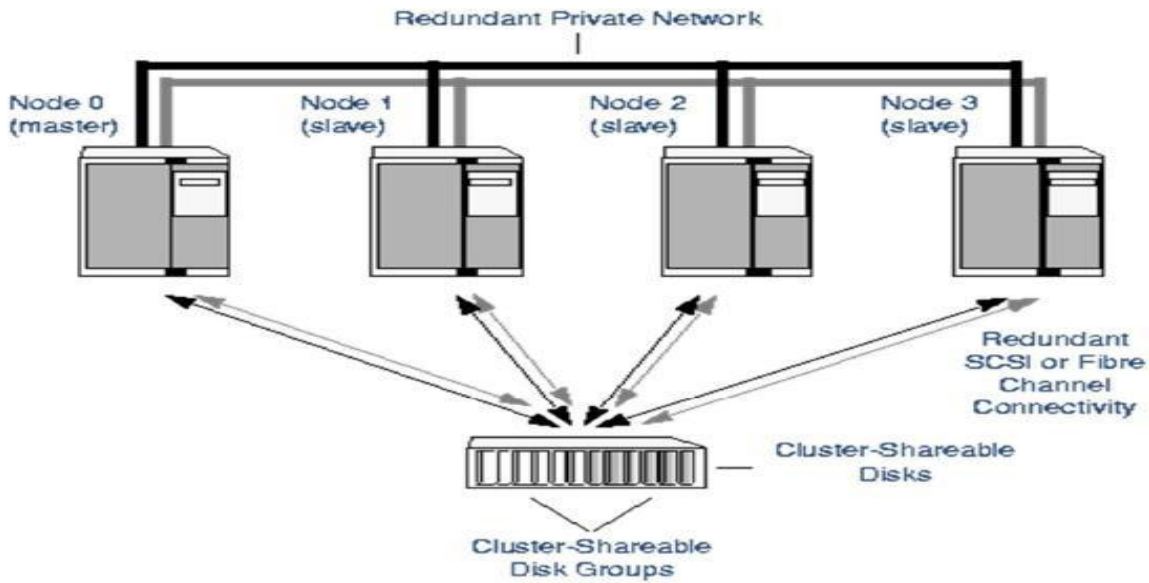
Proses menghubungkan beberapa komputer agar dapat bekerja seperti itu dinamakan dengan *clustering*. Komponen cluster biasanya saling terhubung dengan cepat melalui sebuah interkoneksi yang sangat cepat, atau bisa juga melalui jaringan LAN [5].

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 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 [2].

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 tempat penyimpanan data. Implementasi utama dari shared-disk clustering adalah bukan untuk scalability. Shared-disk clustering ini diimplementasikan untuk availability dan menambah node cadangan sebagai failover node.

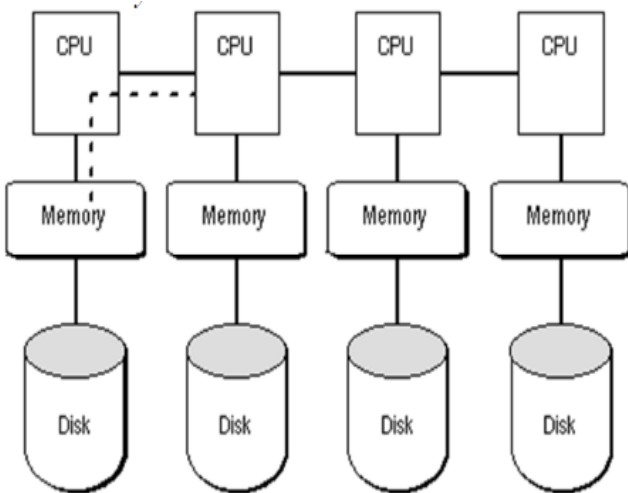


Sumber: Prabowo (2010)

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.



Sumber: Prabowo (2010)

Gambar 2. Shared Nothing Cluster

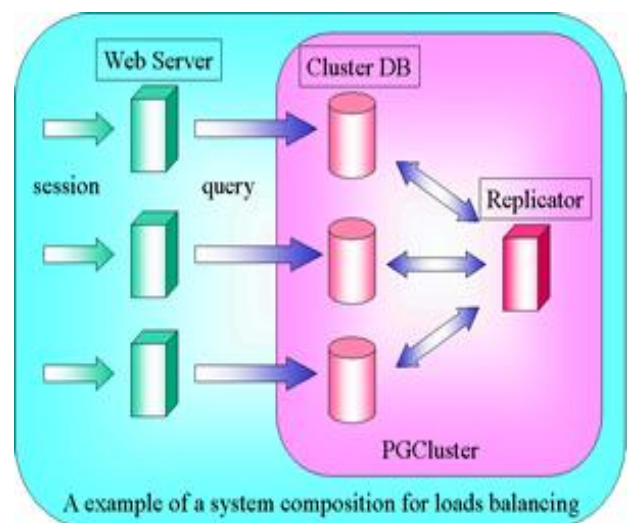
2.3.1 Replikasi Data

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 [2].

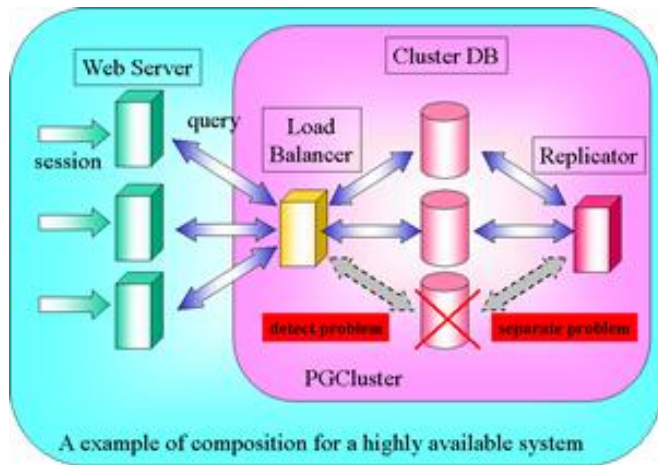
1.3.2 PGCluster

Cluster merupakan sistem replikasi multi-master yang didesain untuk database opensource PostgreSQL. Pada kondisi standar, PostgreSQL tidak memiliki sistem replikasi. Maka fungsi dari replikasi dan pendistribusian beban ini diperlukan. Ada beberapa software menawarkan solusi replikasi untuk database ini. Salah satunya adalah PGCluster.



Sumber: pgcluster.projects.pgfoundry.org (2007)

Gambar 3. Komponen PGCluster untuk Load Balancing Sistem



Sumber: pgcluster.projects.pgfoundry.org (2007)
Gambar 4. Komponen PGCluster untuk High-availability Sistem

Komponen-komponen tersebut diatas, seperti load balancer, server replikasi dan server cluster ini bukan berbentuk fisik, tetapi merupakan unit logika berupa software. Fungsi-fungsi komponen-komponen diatas dapat dilihat pada tabel 1.

Tabel 1. Fungsi dari Komponen PGCluster.

Komponen	Fungsi
Load-balancing	Menerima koneksi dari klien dan mendistribusikannya jika dibutuhkan
Replikasi	Menyebarkan query yang dibuat seperti <i>INSERT</i> , <i>UPDATE</i> dan <i>DELETE</i>
Cluster	Menyimpan data dan mengeksekusi query

Sumber: pgcluster.projects.pgfoundry.org (2007)

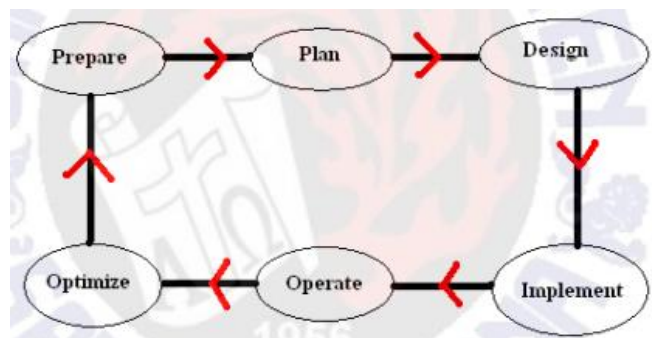
Pada dasarnya PGCluster memiliki dua (2) fungsi, yaitu:

- a. Membagi beban kerja (*Load Sharing*)
 - 1) Beban permintaan pada server akan didistribusikan. Hal ini sangat berguna pada aplikasi berbasis *web*, yang memiliki permintaan pada server *database* yang saling bereferensi.
 - 2) Objek yang direplikasi dapat dispesifikasikan per table. Ketika table yang menerima permintaan untuk *update* dan yang mereferensikan permintaan itu berbeda, PGCluster dapat mendistribusikan table yang menerima permintaan untuk *update* dan mengeluarkan hasil hanya table yang mereferensikan permintaan tersebut.
- b. *High-availability*
Ketika terjadi kegagalan atau masalah pada salah satu server *database cluster*, *load balancer* dan server replikasi akan memisahkan dari server, dan akan melanjutkan pelayanan dengan menggunakan server *database cluster* yang masih aktif. Server *database cluster* yang telah berhasil diperbaiki dapat dimasukkan kembali ke dalam sistem tanpa terlebih dahulu menghentikan layanan.

III. METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode yang dipakai pada perancangan ini adalah Cisco lifecycle services yang merupakan metode yang dirancang untuk mendukung perkembangan jaringan komputer. Cisco lifecycle services adalah pendekatan dengan enam fase. Setiap fase mendefinisikan aktifitas yang dibutuhkan untuk mensukseskan penyebaran dan pengoperasian teknologi jaringan. Fase-fase ini juga mendefinisikan bagaimana mengoptimalkan kinerja di seluruh siklus hidup suatu jaringan. Cisco Lifecycle Services juga biasa disebut dengan metode PPDIIO yang merupakan singkatan dari huruf pertama pada setiap fase yang dilalui.



Sumber: Wilkins (2011)
Gambar 5. Bagan PPDIIO Model

Berdasarkan Gambar 5, metode yang digunakan dalam proses penelitian dilakukan dengan mempersiapkan teknologi jaringan yang akan dibangun, merencanakan seperti apa jaringan yang akan dibangun, mendesain jaringan sesuai dengan yang direncanakan, mengimplementasikan desain yang sudah dibuat, pengoperasian sehari-hari untuk membantu mencapai skalabilitas maksimum, ketersediaan, keamanan dan pengelolaan jaringan yang dibuat, dan yang terakhir adalah mengoptimalkan jaringan yang bertujuan untuk meningkatkan kinerja jaringan dan kehandalan dengan mengidentifikasi dan mengatasi masalah jaringan sebelum terjadi kerusakan.

3.2 Metode Pengujian

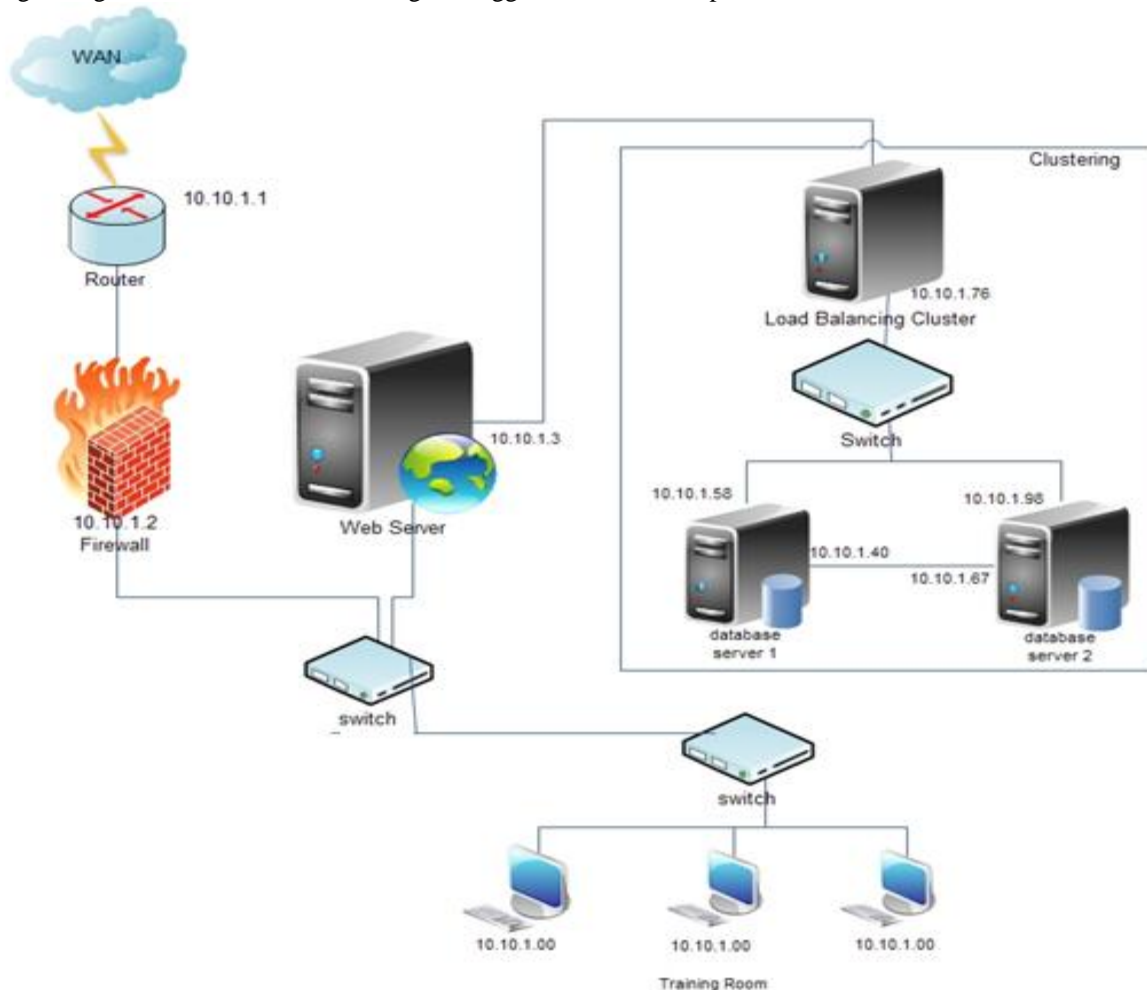
Pengujian yang pertama adalah dengan melakukan pengujian melihat performa dari database server cluster, dengan cara melihat waktu yang diperlukan saat web server melayani request dari user. Pengujian ini akan dilihat kinerja dari database server non cluster (tunggal), kemudian kinerja dari Database server cluster apakah terjadi peningkatan atau tidak. Pengujian yang terakhir adalah pengujian untuk melihat availabilitas dari web server cluster yang sudah berjalan. Pengujian ini dilakukan dengan mematikan salah satu database server pada sistem web server cluster. Setelah salah satu web server mati, dilakukan pengaksesan ke database yang sudah ada pada database server. Apabila user masih bisa mengakses database, maka availabilitas dari web server cluster sudah berjalan dengan baik.

IV. HASIL DAN PEMBAHASAN

4.1 Perancangan *Database Cluster*

Perancangan sistem yang dibangun untuk database server clustering dengan skema load balancing menggunakan

PGCluster adalah dengan menggunakan dua PC sebagai database server yang akan menyimpan database dari system yang tersedia. Satu PC sebagai load balancing yang berfungsi untuk membagi beban kerja antara dua database server. Desain atau perancangan sistem database server clustering dengan skema load balancing menggunakan PGCluster terlihat pada Gambar 6.



Sumber: Hasil Penelitian (2014)

Gambar 6. Desain Arsitektur Cluster

Gambar 6 merupakan gambaran desain sistem perancangan Database server clustering dengan skema load balancing menggunakan PGCluster, dengan menggunakan satu balancer. Balancer berfungsi untuk membagi request dari user untuk dikerjakan secara bersama oleh kedua database server (database1 dan database2). Kedua server database memiliki fungsi yang sama yaitu untuk penyimpanan database dari system dan diantara masing-masing database server dapat terjadi replikasi data. Sehingga seolah-olah semua request dari user hanya dikerjakan oleh satu database server. IP address dari masing PC sebagai berikut:

Web Server : 10.10.1.3/30
Blancer

LAN 1 : 10.10.1.4/30 (IP hubungan ke Web Server)
LAN 2 : 10.10.1.76/24 (IP untuk jaringan cluster)
Database Server1
Cluster1 : 10.10.1.58/24
Replika1 : 10.10.1.40/24
Database Server2
Cluster2 : 10.10.1.98/24
Replika2 : 10.10.1.67/24

Agar perancangan tersebut bisa berjalan dengan baik digunakan teknologi PGCluster (menggunakan PGCluster versi 1.9.0rc7) dengan masing-masing komputer clustering sudah terinstal system operasi linux Ubuntu (menggunakan Ubuntu versi 10.10) dan PostgreSQL. Selanjutnya lakukan

proses instalasi PGCluster dan diteruskan dengan melakukan konfigurasi pada kedua database server dan satu buah server

4.1 Implementasi Sistem

Pada bagian ini memuat hasil dan pembahasan yang meliputi implementasi dari perancangan, pengujian dan hasil analisa. Implementasi tersebut meliputi konfigurasi PGCluster untuk load balance, cluster dan replikasi serta pengaturan file postgresql.conf, dan pg_hba.conf. Sebelum melakukan konfigurasi pada PGCluster, yang harus dilakukan adalah menambah nama dan IP Address pada file /etc/hosts agar setiap IP Address dari server dapat dikenali dengan menggunakan nama. Lakukan perubahan isi file tersebut seperti dibawah ini:

```
10.10.1.58 Cluster1-desktop # Added by NetworkManager
127.0.0.1 localhost.localdomain localhost
::1 Cluster1-desktop localhost6.localdomain6 localhost6
127.0.1.1 Cluster1-desktop
10.10.1.58 cluster1
10.10.1.98 cluster2
10.10.1.76 lb
10.10.1.40 replicatel
10.10.1.67 replicate2
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Lakukan langkah diatas pada ketiga server (kedua server cluster dan satu buah server load balancing).

1. Konfigurasi PGCluster

a. Konfigurasi postgresql.conf

```
cluster1@Cluster1-desktop:~$ sudo -i
root@Cluster1-desktop:~# su postgres
postgres@Cluster1-desktop:~# cd ..
postgres@Cluster1-desktop:~/usr/local/src$ vim /usr/local/pgsql/data/postgresql.conf
```

b. Lakukan perubahan pada file postgresql.conf untuk mengatur alamat koneksi menggunakan port 5432.

```
# Connection Settings #
listen_addresses = '*' # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost', '*' = all
# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
```

load balancer.

c. Lakukan perubahan pada file pg_hba.conf yang bertujuan untuk mengatur autentifikasi pada semua host/user yang akan melakukan koneksi ke database server.

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
host all all 0.0.0.0/0 trust
host all all 10.10.1.58/32 trust
host all all 10.10.1.98/32 trust
host all all 10.10.1.76/32 trust
host all all 10.10.1.40/32 trust
host all all 10.10.1.67/32 trust
```

Lakukan perintah pada poin a sampai dengan poin c pada ketiga server yang diskemakan yaitu satu (1) buah server load balancing dan dua (2) server cluster dan replication.

2. Jalankan Server.

Setelah selesai melakukan instalasi dan konfigurasi pada ketiga (3) server, langkah selanjutnya adalah mengaktifkan atau menjalankan ketiga (3) server tersebut yaitu: Server Cluster Postgresql, Server Replicate Postgresql dan Server Load Balancing

4.2 Pengujian Sistem

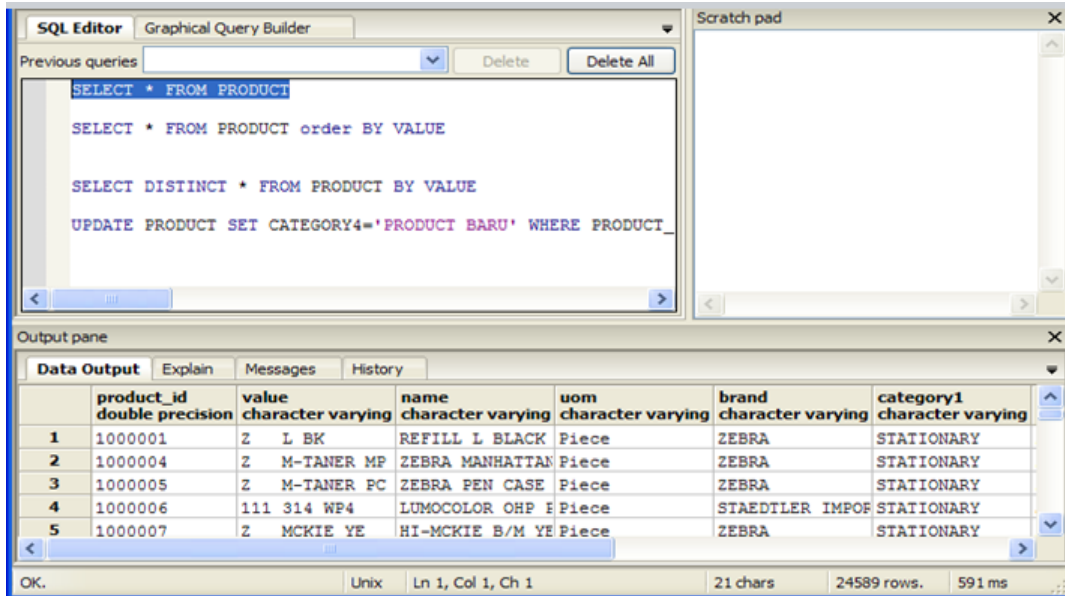
Tujuan dari pengujian sistem adalah untuk mengetahui perbandingan mengenai uji kinerja dan kecepatan akses jaringan dengan melihat dan membandingkan secara umum pemrosesan beberapa operasi (query) pada lingkungan PostgreSQL non-cluster dan lingkungan yang sudah menggunakan clustering.

1. Pengujian Sistem Tanpa Cluster

Untuk pengujian di lingkungan PostgreSQL non-cluster dapat dilakukan melalui koneksi dari komputer klien, dimana dua (2) computer klien secara bersamaan memasukan perintah query ke sistem basis data yang menggunakan database server non cluster (tunggal). Dengan menggunakan pgAdmin III dapat diketahui berapa besar waktu yang dibutuhkan dalam memproses query tersebut.

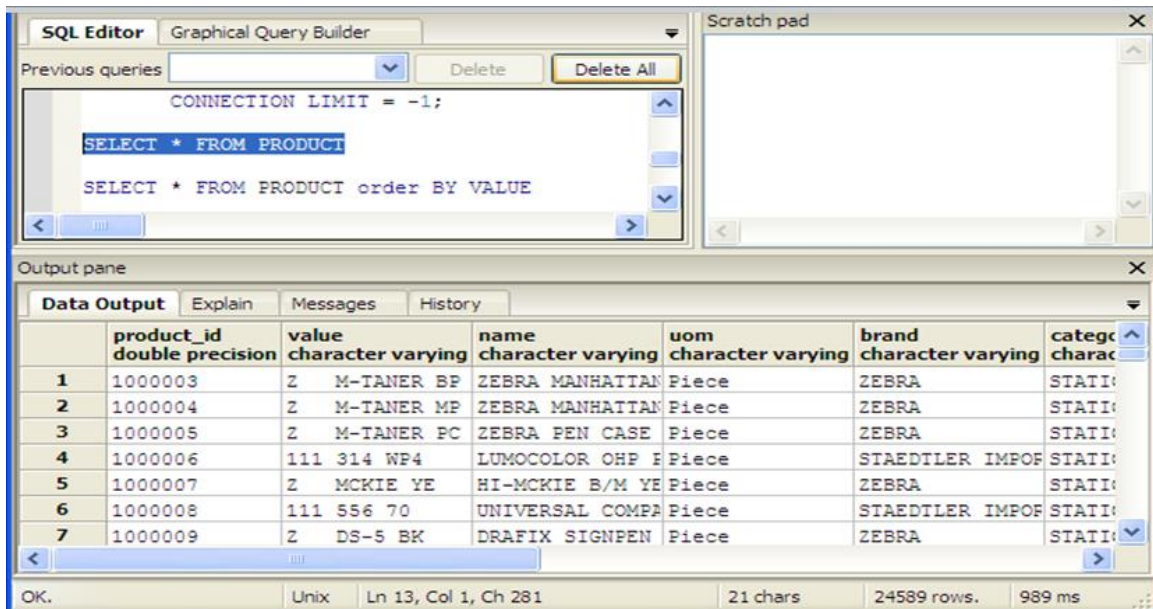
Pada saat pengujian dan setelah koneksi ke server non-cluster menggunakan pgAdminIII telah terhubung, selanjutnya adalah melakukan beberapa operasi query dan melihat hasil waktunya. Beberapa operasi query ke system basis data yang diujikan antara lain:

a. Operasi select * from product



Sumber: Hasil Penelitian (2014)

Gambar 7. Operasi *query* di lingkungan PostgreSQL non-cluster oleh user1



Sumber: Hasil Penelitian (2014)

Gambar 8. Operasi *query* di lingkungan PostgreSQL non-cluster oleh user2

Pada gambar 7 dan 8 terlihat waktu yang dibutuhkan dalam proses operasi query pada lingkungan PostgreSQL non-cluster oleh dua (2) user secara bersamaan. Lamanya waktu untuk proses query select * from product pada user1 adalah sebesar 591 ms, sedangkan pada user2 diperoleh waktu sebesar 989 ms

b. Operasi select * from product order by value

Lamanya waktu untuk proses query select * from product order by value pada user1 adalah sebesar 689 ms, sedangkan pada user2 diperoleh waktu sebesar 1250 ms.

c. Operasi update product set category4='PRODUCT BARU' where Product_id=1000002

Lamanya waktu untuk proses query update product set category4='PRODUCT BARU' where Product_id=1000002 pada user1 adalah sebesar 46 ms, sedangkan pada user2 diperoleh waktu sebesar 47 ms.

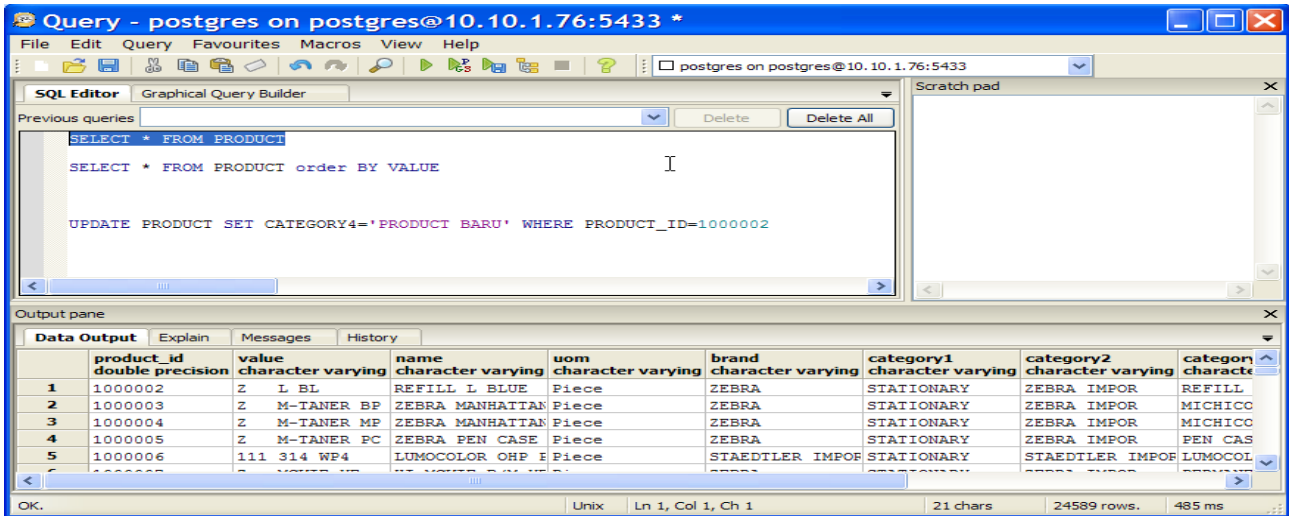
2. Pengujian Sisten dengan Clustering

Untuk pengujian di lingkungan PostgreSQL clustering dapat dilakukan melalui koneksi dari salah satu komputer klien menggunakan pgAdmin III ke server load balancer cluster.

Setelah terhubung ke server *load balancer cluster* menggunakan *pgAdminIII* maka selanjutnya adalah melakukan beberapa operasi *query* dan melihat hasil waktunya. Beberapa operasi *query* yang akan diujikan sama

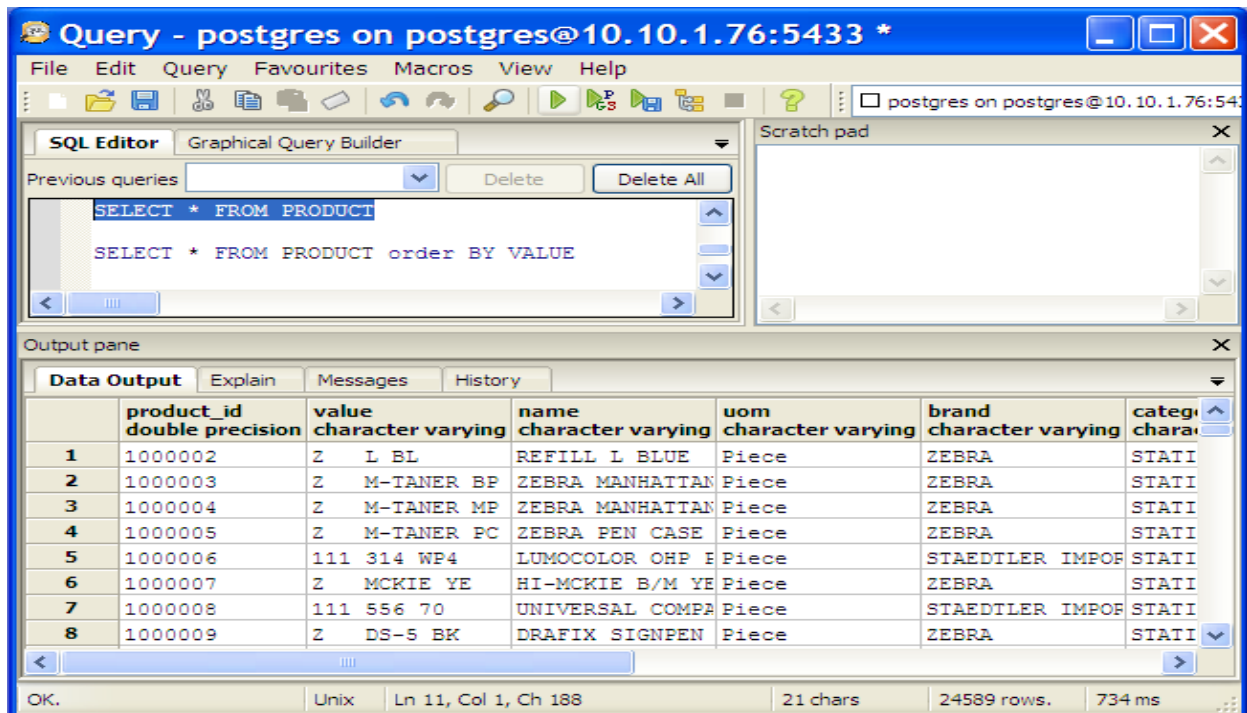
dengan yang telah diuji pada sistem yang telah berjalan sebelumnya. Operasi tersebut antara lain:

- a. Operasi *select * from product*



Sumber: Hasil Penelitian (2014)

Gambar 9. Operasi *query* di lingkungan PostgreSQL dengan *clustering*



Sumber: Hasil Penelitian (2014)

Gambar 10. Operasi *query* di lingkungan PostgreSQL dengan *clustering*

Pada gambar 9 dan 10 terlihat waktu yang dibutuhkan dalam proses operasi *query* pada lingkungan PostgreSQL dengan *clustering* oleh dua (2) *user* secara bersamaan. Lamanya waktu untuk proses *query* *select * from product* pada *user1* adalah sebesar 485 ms, sedangkan pada *user2* diperoleh waktu sebesar 734 ms

- b. Operasi *select * from product order by value*
Lamanya waktu untuk proses *query* *select * from product order by value* pada *user1* adalah sebesar 609 ms, sedangkan pada *user2* diperoleh waktu sebesar 891 ms.

c. Operasi update product set category4='PRODUCT BARU' where Product_id=1000002

Lamanya waktu untuk proses query update product set category4='PRODUCT BARU' where Product_id=1000002 pada user1 adalah sebesar 32 ms, sedangkan pada user2 diperoleh waktu sebesar 16 ms.

Dibawah ini adalah table hasil perbandingan waktu pemrosesan beberapa operasi query pada sistem yang telah berjalan (non-cluster) dan sistem yang telah dibangun (cluster) dengan dua (2) user mengakses masing-masing server secara bersamaan.

Tabel 2. Hasil perbandingan waktu pemrosesan query dengan 2 user akses secara bersamaan

No	Query yang di jalankan	Non-Cluster Farm	Cluster Farm	Rata-rata waktu Non-Cluster Farm	Rata-rata waktu Cluster Farm	Persentase Perbandingan Kedua Sistem
1	Select * from Product;	User1 : 591 ms	User1 : 485 ms	790 ms	609 ms	22,911%
		User2 : 989 ms	User2 : 734 ms			
2	select * from product order by value	User1 : 689 ms	User1 : 609 ms	969 ms	750 ms	22,600%
		User2 : 1250 ms	User2 : 891 ms			
3	update product set category4='PRODUCT BARU' where Product_id=1000002	User1 : 46 ms	User1 : 32 ms	46 ms	24 ms	47,826%
		User2 : 47 ms	User2 : 16 ms			

Sumber: Hasil Penelitian (2014)

3. Pengujian Availabilitas Web Server

Pengujian pada tahap ini dilihat dari *availabilitas* antara *database server non-cluster* (tunggal) dan *database server cluster*. Pengujian ini berfungsi untuk melihat apakah dalam *database server cluster* tetap dapat melayani *request* dan menyediakan ketersediaan data bagi *user*.

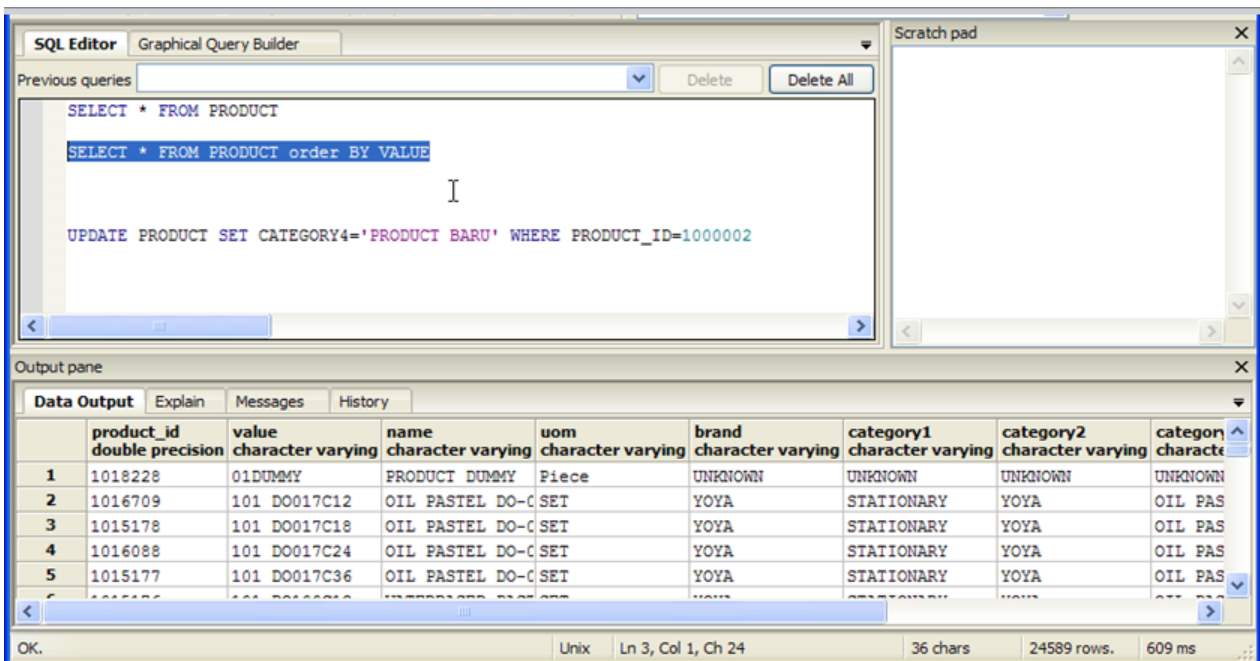
a. Database server non-cluster (tunggal)

Pengujian ini dilakukan dengan cara me-non-aktifkan NIC (*Network Interface Card*) pada *database server non-cluster* (tunggal) dengan berasumsi bahwa *web server* tunggal telah mati (*down*), kemudian dilakukan pengaksesan ke *web server* tersebut. Hasilnya *query* tidak berjalan.

menunjukkan bahwa pada *web server* tunggal apabila *web server down* maka *user* sudah tidak bisa melakukan *request* kepada *web server*.

b. Database Server Cluster

Pengujian ini dilakukan dengan cara yang sama pada pengujian *database server tunggal* yaitu dengan cara me-non-aktifkan NIC (*Network Interface Card*) pada salah satu *web server (node)* yang ada pada *cluster* dengan berasumsi bahwa *web server* telah mati (*down*). Hasil pengujian ini dapat dilihat pada Gambar 7.



Sumber: Hasil Penelitian (2014)

Gambar 11. Database Server Cluster dengan Satu Database Server mati (Down)

Gambar 9 memperlihatkan ternyata sistem *load balancing* dalam mengimplementasikan *availabilitas* sistem *cluster* dapat bekerja dengan baik, karena *request* dari *user* masih bisa dilayani dengan baik. Hal tersebut dikarenakan masih adanya *Database server (node)* yang masih tetap bekerja, sehingga *balancer* hanya mengalihkan *request* dari *user* tersebut ke *Database server (node)* yang masih aktif. Jika *web server* yang mengalami (*down*) sudah diperbaiki, maka *web server* tersebut akan langsung aktif kedalam anggota *cluster*.

4.3 Hasil Analisa

Dari semua hasil pengujian yang telah dilakukan, hasil analisa dari sistem *load balancing* untuk *Database server* dengan *PGCluster* adalah:

1. Dengan menggunakan aplikasi *PGCluster* dapat mengimplemntasikan *load balancer*, *request* dari *user* mampu dibagi ke semua *database server* yang menjadi anggota *cluster*.
2. *PGCluster* mampu memberikan layanan data secara *realtime*, karena sistem *replikasi* data yang ada pada setiap mesin *storage*.
3. Pengujian dengan menggunakan aplikasi *pgAdminIII*, sistem *database server cluster* mampu memberikan layanan data secara lebih cepat dari pada menggunakan sistem *web server non-cluster* (tunggal).
4. *Availabilitas database server* dapat terpenuhi dengan baik, hal ini dibuktikan dengan mematikan salah satu *datanbase server* dan *request* dari *user* masih bisa dilayani dengan baik. Pengujian ini juga membuktikan bahwa kinerja dari *load balancing* sudah berjalan dengan optimal.
5. Titik krusial pada sistem *cluster* terletak pada *server load balancer*. Hal ini terjadi karena apabila *server load balancer* mati, *database server* tidak dapat diakses.

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan tentang perancangan dan implementasi *database server clustering* dengan menggunakan *PGCluster*, dapat diambil kesimpulan sebagai berikut :

1. Sistem *load balancing* juga bisa dimanfaatkan sebagai *backup* atau *failover* yang artinya jika salah satu *Database server* mangalami kegagalan (*down*), salah satu *Database server* masih bisa melayani *request* dari *user*.
2. Pelayanan *request* data dari *user* dapat ditangani lebih cepat dengan menggunakan *database server clustering* daripada dengan menggunakan *database server* tunggal.
3. Implementasi *PGCluster* menggunakan metode *replication* mengakibatkan ketersediaan data pada suatu sistem basis data tetap terjaga.

REFERENSI

- [1] PGCluster Documentation. *Clustering System of PostgreSQL using Shared Data*. Diambil dari: <http://pgcluster.projects.postgresql.org/>. (05 Juni 2013). 2007.

- [2] Prabowo, Adityo. *Perancangan MySQL Cluster untuk Mengatasi Kegagalan Sistem Basis Data pada Sisi Server*. Semarang: Universitas Diponegoro. 2010.
- [3] Sopandi, Dede. *Instalasi dan Konfigurasi Jaringan Komputer*. Bandung: Informatika. 2008.
- [4] Wahyudi, Sri. *Pembuatan Sistem Cluster dan Load Balancing untuk Database Server dengan Engine MySQL*. Surakarta: FMIPA Universitas Sebelas Maret. 2010.
- [5] Williams, Rob. "Computer Systems Architecture A Network Approach, 2nd Edition". England: Pearson Education Limited. 2006.
- [6] Wilkins, Sean. *Designing for Cisco Internetwork Solutions (DESGN) Foundation Learning Guide: (CCDA DESGN 640-864), 3rd Edition*. Indiana: Cisco press. 2011.



Suryanto. M.Kom. Tahun 2002 lulus dari Program Strata Satu (S1) pada Fakultas Teknik UNKRIS, Jakarta. Tahun 2010 lulus Program Strata Dua (S2) 10 pada Pasca Sarjana Ilmu Komputer STMIK Nusa Mandiri, Jakarta. Saat ini bekerja sebagai dosen tetap di AMIK BSI Jakarta dengan Jabatan Fungsional Akademik Lektor di Program Studi Manajemen Informatika AMIK BSI Jakarta. Aktif mengikuti seminar dan menulis di beberapa jurnal ilmiah.