 ISSN : 1858 – 3938

IMPLEMENTASI SISTEM REPLIKASI DATA BASE

POSTGRESQL MASTER-SLAVE REPMGR DENGAN AUTO PROMOTE MASTERDB

Erwin Asriyar, Teten Sutendi

Sekolah Tinggi Teknologi Informasi NIIT, Jakarta

ABSTRAK

Replikasi merupakan proses penyalinan dan pemeliharaan objek basis data, seperti tabel-tabel, dalam banyak basis data yang membentuk suatu sistem basis data terdistribusi. Perubahan-perubahan yang dilakukan pada satu tempat dicatat dan disimpan secara lokal sebelum diteruskan dan diterapkan pada setiap lokasi lain. Proses replikasi basis data merupakan sebuah proses yang banyak dilakukan pada sistem basis data terdistribusi. Sistem basis data terdistribusi terdiri dari kumpulan site-site, masing masing site ini dapat berpartisipasi dalam pemrosesan transaksi yang mengkases data pada suatu site atau beberapa site. Beberapa alasan untuk membangun basis data terdistribu-si, seperti pemakaian bersama (*share*), kehandalan (*reliability*), ketersediaan (*availability*) dan kecepatan pemrosesan query. Keuntungan utama dari basis data terdistribusi adalah kemampuan untuk pemakaian dan pengaksesan data se-cara bersama dengan cara yang handal dan efisien. Salah satu metode replikasi data adalah dengan menggunakan replikasi master-slave. Replikasi master-slave merupakan sebuah metode penduplikasi basis data yang memungkinkan data untuk disimpan di sejumlah *site/node* dan di-*update* dari *server master database* dan slave *stanby*. Pada bentuk *master-slave master* sebagai server aktif melakukan *query update delete* dan *slave* berperan server *standby* dan mereplikasi *update* dari *server master*.

Kata Kunci:postgresql,replikasi,repmgr,auto promote,pgbouncer.

 ISSN : 1858 – 3938

1. PENDAHULUAN

Dengan semakin besarnya kebutuhan masyarakat akan informasi dan semakin kompleksnya informasi yang harus disajikan, secara tidak langsung mengakibatkan perkembangan teknologi informasi menjadi semakin kompleks dengan beragam platform dan sumber daya yang berbeda. Disatu sisi perkembangan teknologi memberikan kemudahan bagi penyajian dan pengolahan informasi, namun disisi lain dengan segala kecanggihan teknologi informasi yang telah ada masih memiliki keterbatasan apabila diintergrasikan dengan aplikasi atau *platform* yang berbeda.

Kebutuhan akan tersedianya data yang baik setiap saat menjadi permasalahan yang rumit. Hal ini berkaitan dengan kinerja sistem, jumlah pengakses data, serta lalu lintas jaringan data yang setiap saat bertambah padat. Hal ini memicu munculnya teknologi replikasi data.

Replikasi merupakan proses penyalinan dan pemeliharaan objek basis data seperti tabel-tabel dalam banyak basis data yang membentuk suatu sistem basis data terdistribusi. Perubahan-perubahan yang dilakukan pada satu tempat dicatat dan disimpan secara lokal sebelum diteruskan dan diterapkan pada setiap lokasi lain.

Proses replikasi basis data merupakan sebuah proses yang banyak dilakukan pada sistem basis data terdistribusi. Sistem basis data terdistribusi terdiri dari kumpulan site-site, masing-masing site ini dapat berpartisipasi dalam pemrosesan transaksi yang mengakases data pada suatu site atau beberapa site. Beberapa alasan untuk membangun basis data terdistribusi adalah seperti pemakaian bersama (*share*), kehandalan (*reliability*), ketersediaan (*availability*) dan kecepatan pemrosesan query. Keuntungan utama dari basis data terdistribusi adalah kemampuan untuk pemakaian dan pengaksesan data secara bersama dengan cara yang handal dan efisien.

Salah satu metode replikasi data adalah dengan menggunakan replikasi *master-slave*. Replikasi *master-slave* merupakan sebuah metode penduplikasi basis data yang memungkinkan data untuk disimpan di sejumlah *site/node* dan di-*update* dari *server master database* dan slave *stanby*. Pada bentuk *master-slave master* sebagai server aktif melakukan *query update delete* dan

 ISSN : 1858 – 3938

slave berperan server *standby* dan mereplikasi *update* dari *server master*.

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.

Replikasi basis data pada laporan ini dilakukan dengan menggunakan postgresql 9.6.. Teknik replikasi yang dilakukan adalah dengan menggunakan replikasi *Master-slave*.

Model replikasi pada postgresql adalah synchronous, sehingga server dengan type slave selalu terkoneksi secara permanen untuk menerima berbagai update pada *database server master*.

Saat ini di subag PBJ (Pengadaan Barang dan Jasa) Kemdikbud melakukan pengembangan pada aplikasi lpse untuk mempermudah dan dapat mempercepat pemenuhan kebutuhan dan keakuratan informasi yang diperlukan oleh seluruh peserta lelang barang dan jasa.

Sistem yang di kembangkan saat ini menghadapi banyak tantangan terutama di ketersediaan data, kerusakan saat *server master database error* menjadi penghambat saat proses lelang karena harus menunggu *recovery* data pada *server master database.*

Untuk menjaga aplikasi tetap berjalan di saat server master database *down* maka harus dibuat *autopromote server slave* menjadi *server master* sehingga tidak perlu menunggu perbaikan server master selesai. Perbaikan saat *error database* bisa di gantikan dengan *server slave* menjadi *backup database master.*

Identifikasi Masalah

Berdasarkan latar belakang di atas perlu dibuat suatu server basis data secara terdistribusi sehingga identifikasi masalah yang akan diselesaikan dalam penelitian ini :

1. Belum di buat *server database*

2. secara terdistribusi atau replikasi.
3. Kerusakan data pada *server database* mengakibakan terhentinya aktivitas pada aplikasi.
4. Tidak ada *auto switch ove*r atau *fileover* database *master* ke *slave* saat terjadi *failure* mesin.

Rumusan Masalah

Berdasarkan identifikasi masalah di atas perlu dibuatkan server data base yang bisa mereplikasi data base secara *realtime* sehingga identifikasi masalah yang akan diselesaikan dalam penelitian ini:

1. Bagaimana mensimulasikan replikasi database dengan *model master-slave* dengan metode *syncronius*.
2. Bagaimana mensimulasikan *auto promote server slave* menjadi *server master*.

Pembatasan Masalah

Berdasarkan rumusan masalah di atas, pembahasan permasalahan di batasi adalah:

1. Simulasi dilakukan dalam tiga server database.
2. Basis data yang digunakan dan dibuat hanya untuk memberikan gambaran secara umum mengenai replikasi database dengan model *master-slave* memakai metode syncronius dan mecoba *auto promote server slave* menjadi *master.*
3. Server yang di gunakan dengan spesifikasi sebagai berikut:

   a. *Software database* menggunakan postgresql 9.6.

   b. *System* operasi menggunakan CentOs 7.

   c. Aplikasi virtual server menggunakan Virtualbox.

Tujuan dan Manfaat Penelitian

Tujuan dari pembuatan Laporan Tugas Akhir :

1. Menganalisa dan meracang replikasi database *master-slave* dengan postgresql.

 ISSN : 1858 – 3938

2. Basis data yang di replikasi tersingkronisasi
3. Membuat a*uto promote server slave* menjadi *master* di saat *server master mengalami error*.

Manfaat dari hasil penelitian yang diharapkan adalah :

1. Dapat mengatasi backup database yang kurang sempurna karena data di replikasi dengan secara realtime dengan metode *syncronus*.
2. Meningkatkan avabilitas data.
3. Menghindari kemungkinan tidak semua data terbackup karena saat proses backup data manual dilakukan bisa saja terjadi perubahan data oleh client

Apabila *server master* mengalami kerusakan, database bisa segera dialihkan ke *server slave*.

Landasan Teori

secara *realtime*.

Menurut Nasution, 2013:3 replikasi adalah suatu teknik untuk melakukan copy dan pengiriman data dan objek-objek database dari satu database ke database lain yang lokasinya terpisah secara fisik. Dengan menggunakan teknik replikasi ini, data dapat dikirimkan ke lokasi yang berbeda melalui koneksi jaringan lokal maupun internet. Model Replikasi terdiri dari *one master, one slave, one master, many slave, master/slave circular relationship* dan *master/slave "daisy chain"*. Dengan menggunakan teknik replikasi ini, data dapat dikirimi ke lokasi yang berbeda melalui koneksi jaringan lokal maupun internet.

Bentuk Replikasi

Perbedaan paling menonjol dalam kasus replikasi adalah pada bentuk yang digunakan.

 ISSN : 1858 – 3938

Bentuk-bentuk replikasi yaitu: replication.

a. Full Replication

replication. Dalam bentuk ini seluruh database yang ada pada satu lokasi diduplikasikan ke beberapa lokasi database server lain.

b. Partial Replication

Partial Replication adalah replikasi yang hanya melibatkan beberapa database saja. Meskipun dalam ketersediaan data tidak sebaik full replication, namun mampu meningkatkan performansi dengan lebih baik.

Modifikasi Data pada Replikasi

Dalam proses modifikasi data pada replikasi, penarikan atau pengambilan kembali data (retrieve) yang direplikasi dapat dilakukan dengan dua cara, yaitu :

1.*Synchronous*

Model synchronous memungkinkan pemrosesan perintah yang berupa permintaan full replication dan partial

Replikasi yang melibatkan seluruh database yang ada pada suatu lokasi disebut dengan full (request) dilakukan pada sistem terdistribusi. Dalam hal ini semua data dimodifikasi selama terjadi proses transaksi, dan request berikutnya akan terkunci sampai request sebelumnya selesai dilakukan.

2.*Asynchoronous*

Pada model ini, modifikasi pada suatu data akan disebarkan pada lokasi lain dalam beberapa waktu kemudian setelah file sumber dimodifikasi (*update*) meski tidak dalam transaksi yang sama.

Synchronous

Menurut Jeisha, 2009:1 *synchronous* transmission ini dikenal dengan istilah synchronous transfer mode (STM). Proses pengirim dan penerima diatur sedemikian rupa agar memiliki pengaturan yang sama, sehingga dapat dikirimkan dan diterima dengan baik antar

 ISSN : 1858 – 3938

alat tersebut. Umumnya pengaturan ini didasarkan terhadap pewaktuan dalam mengirimkan sinyal.

PostgreSQL

Basis Data Untuk SIG PostgreSQL merupakan salah satu DBMS yang digunakan untuk menyimpan data dan bersifat *open source*. Untuk dapat menyimpan data spasial, PostgreSQL membutuhkan plugin tambahan yaitu PostGIS. PostGreSQL dilengkapi dengan berbagai variasi tipe data seperti numerik (integer dan numeric), string atau teks (char dan VarChar), interval, timestamp, dan date. Selain itu PostGreSQL juga menyediakan tipe data geometrik terutama dalam kaitannya untuk PostGIS dan byte (Eddy Prahasta, 2012).

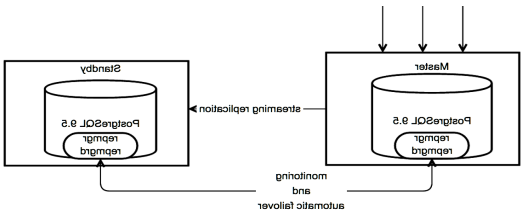
Master-Slave Replikasi

Dengan metode ini, salah satu komputer berfungsi sebagai master dan yang lainnya berfungsi sebagai slave. Pada prosesnya, komputer yang digunakan sebagai server akan dapat read dan write ke dalam database.Sedangkan komputer yang berfungsi sebagai slave, hanya akan read saja kedalam Basis Datatersebut. Apabila kita melakukan perubahan data pada master, maka otomatis data pada slave akan berubah. Tetapi jika kita melakukan perubahan data pada slave, Basis Datapada master tidak akan berubah (Eddy Purwanto, S.Kom,2012).

Dalam replikasi ada server yang di jadikan Master dan Slave. server master adalah sebuah server yang akan di jadikan sebagai Induk/Utama sedangkan server slave adalah server yang di jadikan sebagai server kopian dari server master. sedangkan proses replikasinya

sendiri, Server yang sebelumnya sudah di konfigurasi sebagai master akan seperti biasa beroperasi,saat server master itu menerima sebuah query dari user/client sesudah query itu di terapkan menjadi sebuah record/data maka secara otomatis query itu di teruskan oleh server master ke server yang sebelumnya sudah di konfigurasi sebagai slave melalui jaringan, dan terakhir query itupun di diterapkan di

server slave dan seperti itu terus, replikasi akan terus berjalan melakukan backup sempurna secara realtime asalkan jaringan antara server master dan slave tidak ternganggu.jadi intinya setiap query yang dijalankan di server master makan dijalankan juga pada server slave.



gambar 2.1.master-slave replikasi database postgresql

REPMGR

Repmgr adalah alat opensource untuk mengelola replikasi dan *failover* server databes postgresql serta untuk meningkatkan kemampuan pada mode *hot-standby* replikasi pada postgresql,alat ini mampu mengatur sendiri,memantau serta melaksanakan tugas administrasi *failover* baik secara otomatis atau secara manual(repmgr.org).

Untuk melakukan switching master saat terjadi error di server master maka repmgr akan melakukan promote ke server slave yag standby untuk di jadikan server master.

Failover

Definisi failover dalam istilah computer internetworking adalah kemapuan sebuah sistem untuk dapat berpindah secara manual maupun otomatis jika salah satu sistem mengalami kegagalan sehingga menjadi backup untuk sistem yang

dapat disimpulkan bahwa tujuan dari failover adalah sistem backup server yang terputus dengan server yang lainnya.

PGbouncer

pgbouncer adalah koneksi pooler PostgreSQL . Setiap target aplikasi dapat terhubung ke pgbouncer seolah-olah itu sebuah server PostgreSQL, dan pgbouncer akan membuat koneksi ke server yang sebenarnya, atau akan menggunakan kembali salah satu dari koneksi yang ada.Tujuan dari pgbouncer adalah untuk menurunkan kinerja dampak dari pembukaan baru koneksi ke PostgreSQL.Dalam hal transaksi

semantik untuk connection pooling, pgbouncer mendukung beberapa jenis pooling koneksi:

Session pooling:

Ketika klien terhubung, koneksi server akan tetap terhubung. Ketika klien terputus, koneksi server akan dimasukkan kembali ke dalam pooling. Ini adalah metode standar dalam pgbouncer.

Transaction pooling:

Koneksi klien hanya terhubung selama transaksi. Ketika PgBouncer pemberitahuan bahwa transaksi selesai, koneksi server akan dimasukkan kembali ke dalam pooling.

Statement pooling:

Metode ini sangat agresif. Koneksi server akan dimasukkan kembali ke pooling segera setelah permintaan selesai. Multi-pernyataan transaksi yang dilarang dalam mode ini.

5. PERANCANGAN SISTEM

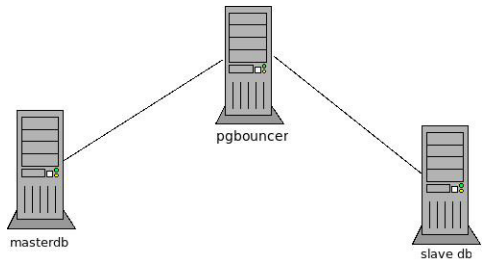
Komponen Perangkat Virtual Server dan Lunak

Perangkat-perangkat yang dibutuhkan untuk pengembangan sistem meliputi komponen virtual server, dua virtual server, yaitu dua buah virtual server yang di buat di virtualbox, Sistem Operasi Linux , postgresql server.

Topologi Jaringan

Topologi jaringan yang digunakan dalam pembangunan system master-slave database dengan menggunakan postgresql ini dilakukan dalam sebuah jaringan virtual dengan menggunakan Virtualbox yang dapat juga dihubungkan dengan jaringan fisik

(external). Simulasi master-slave database ini menggunakan sistem operasi CentOs7 Server dengan menggunakan virtual machine Virtualbox.



Gambar 3.1 Topologi Jaringan

Pada gambar menerangkan topologi jaringan yang digunakan untuk proses replikasi database oleh postgresql di dalam Postgresql proses replikasi dapat dilakukan jika terdapat sebuah master dan satu buah slave untuk proses replikasi dapat berjalan dengan maksimal. Untuk lebih jelasnya dapat melihat tabel 1 berikut:

Tabel 3.1 spesifikasi virtual server

Networking dalam simulai ini menggunakan dua jaringan, yaitu nat dan host only adapter yang mana dalam implementasi akan menggunakan ip publik dan privat . Jaringan nat berfungsi sebagai translasi alamat IP public ke alamat IP private atau sebaliknya sehingga dengan adanya NAT ini setiap virtual server pada virtualbox dapat mengakses internet dengan mudah.dan jaringan host only adapter difungsikan untuk komunikasi backend, dan khusus untuk komunikasi antar guest OS (sistem operasi) itu sendiri. Berikut ringkasan network dapat terlihat dalam Tabel 2.

Nama server	Prosesor	Ram	Hardisk	Eth0	Eth1
Master	1	512	8 gb	Enp0s3	Enp0s8
Slave	1	512	8 gb	Enp0s3	Enp0s8
Pgbouncer	1	512	8 gb	Enp0s3	Enp0s8

Tabel 3.2 konfigurasi IP pada virtualbox

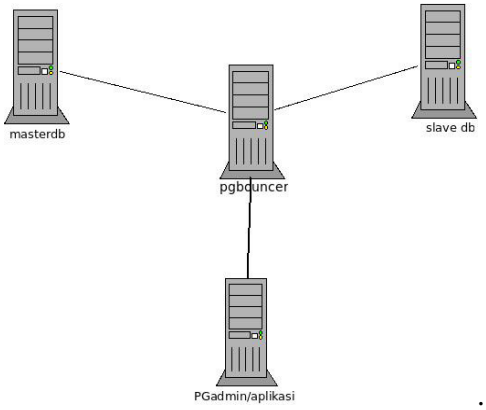
Name	Virtual card	Network	Netmask	Gateway
Nat	Enp0s3	10.0.2.15	255.255.255.0	10.0.2.1
Host only adapter	Enp0s8	192.168.56.0	255.255.255.0	192.168.56.1

Pada table di atas terlihat ip address untuk host only adapter menggunakan dhcp yang di atur oleh network

 ISSN : 1858 – 3938

virtualbox begitu juga dengan Nat menggunakan ip virtualbox yang terhubung dengan internet

Untuk melakukan koneksi dari client ke server melalui pgbouncer yang menjadi perantara antara masterdb dan slave db sehingga client tidak lagi melakukan koneksi langsung ke server database sehingga pada saat *server master down* tidak perlu melakukan perubahan konfigurasi pada sisi aplikasi cukup di sisi server pgboucer bisa di lihat pada gambar berikut:



Gambar 3.2 Topologi Jaringan client ke pgbouncer

Pada gambar di tersebut bisa di lihat koneksi pgadmin melakukan koneksi ke server pgbouncer sebagai perantara koneksi antara slavedb dan masterdb.

Promote slave server ke master server

Di saat terjadi error pada server master maka akan otomatis slave menjadi master yag di promote oleh tools repmgr seperti di gambarkan di bawah ini:

Gambar 3.3.*promote slave database* ke *master database*

Pada gambar di atas merupakan contoh rancangan arsitektur sistem yang akan dibangun untuk replikasi database. Terdapat beberapa komponen yang di perlukan untuk replikasi dapat berjalan, master dan slave. Dimana di dalam gambar tersebut di jelaskan bahwa jika sebuah master mengalami kerusakan atau tidak dapat diakses maka akan dapat langsung dialihkan kepada master baru yang merupakan slave sebelumnya. Permpindahan dari slave kemaster database akan di atur oleh repmgr secara otomatis.

Implementasi Dan Pengujian Systen

1. Prosedur

   Prosedur akan mendefinisikan urutan-urutan pengerjaan dari metode dan alat bantu yang digunakan di dalam pemecahan atau pembuatan perangkat lunak.

Rancangan-rancangan pada tugas akhir ini adalah:

Simulasi Sistem

Pada bagian ini akan menerapkan apa yang telah direncanakan dan didesain sebelumnya. Selanjutnya sistem diimplementasikan dalam beberapa hal,seperti :

a. Instalasi postgresql,repmgr dan konfigurasi
   Pada tahap ini melakukan instalasi postgresql dan konfigurasi service postgresql.
b. Konfigurasi master-slave database
   Pada tahap ini melakukan konfigurasi master dan slave postgresql.
c. Konfigurasi repgmgr dan auto promote
   Pada tahap ini melakukan instalasi repmgr dan konfigurasi auto promote standby server ke primary server.

Instalasi Postgresql Pada master dan slave sever database

Instalasi paket dependensi paket postgres postgressql pada server master dan slave sbb:

Unduh file postgresql di https://yum.postgresql.org/repopackages.php
#curl –O https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-centos93-9.6-3.noarch.rpm
Extrack paket rpm postgresql
#rpm –ivh pgdg-centos93-9.6-3.noarch.rpm
Cek list versi postgresql
#yum list postgres*
Install postgresql
#yum install posgrest-server
Inisialisasi service postgresql
#/usr/pgsql-9.6/bin/postgresql96-setup initdb
Config servise postgresql hidup saat boot
#chkconfig postgresql-9.6 on
Jalankan service postgresql
#service postgresql-9.6 start
Masuk ke dalam postgresql
#su – postgres
Untuk memulai pembuatan database haru masuk ke psql postgres
#psql

Konfigurasi server master dan slave postgresql

1. Konfigurasi master postgresql

Pada tahap ini melakukan konfigurasi server master postgresql pada serverr ini menjadi aktif melakukan query seperti update,delet,insert di lakukan di server master sbb:

a. Edit file /var/lib/pgsql/data/postgresql.conf
   # nano /var/lib/pgsql/data/postgresql.conf
   # line 59: uncomment and change
   listen_addresses = '*'
   # line 165: uncomment and change
   wal_level = hot_standby
   # line 168: uncomment and change
   # on ⇒ sync
   # remote_write ⇒ memory sync
   # local ⇒ slave is asynchronous
   # off ⇒ asynchronous

 ISSN : 1858 – 3938

synchronous_commit = local
# line 194: uncomment and change (enable archive_mode)
archive_mode = on
# line 196: uncomment and change (command to get archives)
archive_command = 'cp %p /var/lib/pgsql/archive/%f'
# line 212: uncomment and change (slave servers + 1)
max_wal_senders = 2
# line 214: uncomment and change
wal_keep_segments = 10
# line 221: uncomment and change (any name you like)
synchronous_standby_names = 'slave01'

b. Edit file /var/lib/pgsql/data/pg_hba.conf
add to the end
host replication [replication user] [allowed IP addresses] password
Host replication replica 192.168.56.0/24 trust

c. Edit user replica pada postgresql
su - postgres
-bash-4.2$ createuser --replication -P replica

d. setting firewal buka port 5432
firewall-cmd --permanent --add-port=5432/tcp
firewall-cmd --reload

e. buat folder archive
mkdir /var/lib/pgsql/archive
chmod -R 700 /var/lib/pgsql/archive
chown -R postgres:posgres /var/lib/pgsql/archive

2. konfigurasi server slave postgresql
pada tahap ini melakukan konfigurasi slave server postgresql,server slave ini merupakan server *standby* yang hanya menerima update dari server master postgresql di saat server master *error* maka server slave ini bisa untuk menggantikan server master tanpa melakukan dumping data dari server master postgre karena data telah *syncron* dengan master sebelum server master error konfigurasi sebagai berikut:

a. matikan service postgresql
#service postgresql-9.6 stop

b. masuk ke user postgres
#su - postgres

c. lakukan backup dari server master
pg_basebackup -h 192.68.56.1 -U replica -D /var/lib/pgsql/data -P --xlog
Password: # "replica" user's password

d. config file /var/lib/pgsql/data/postgresql.conf
# line 230: uncomment and change
hot_standby = on

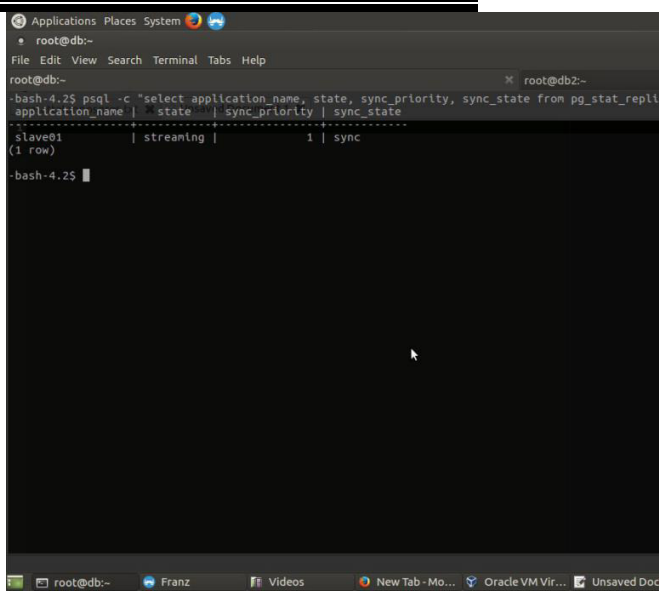
e. buat file recovery.conf
touch /var/lib/pgsql/9.6/recovery.conf
isi kan sbg berikut
restore_command = 'scp 192.168.56.1:/var/lib/pgsql/archive/%f %p'
standby_mode = on
primary_conninfo = 'host=192.168.56.1 port=5432 user=replica password=password application_name=slave01'

f. buat folder archive
mkdir archive
chmod -R 700 archive
chown postgres:postgres archive

g. hidupkan service postgresql
service postgresql-9.6 start

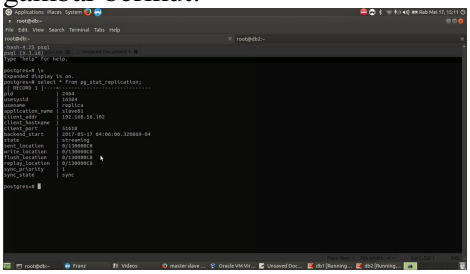
h. masuk ke user postgres
#su - postgres
jalankan perintah berikut mengecek proses syncronisasi/replikasi pada master

-bash-4.2$ psql -c "select application_name, state, sync_priority, sync_state from pg_stat_replication;"
Maka output tamplian sebagai berikut:



Gambar 4.1 pengecekan setreming databas

Untuk lebih detail bisa di lihat pada gambar berikut:



Gambar 4.2 metode streaming syncronisasi

Dari gambar di atas kita telah membuat replikasi dengan
Username : replica
Aplicationname : slave01
Alamat client/slave : 192.168.56.102
Syncronisasi mode : sync

Konfigurasi repmgr

Sebelum melakukan konfigurasi repmgr untuk auto promot harus melakukan instalasi terlebih dahulu di kedua node1 dengan perintah instalasi sebagai berikut:

yum install repmgr-9.6 -y

Setelah instalasi selesai lanjutkan dengan konfigurasi repmgr di kedua node

Node1 master atau primary

Buka file /etc/repmgr/9.6/repmgr.conf lakukan uncoment pada bagian berikut:

node=1
node_name=node1
conninfo='host=192.168.56.103 user=’repmgr dbname=repmgr'

selanjutnya buat beberapa symlink untuk command repmgr yang kita butuhkan

ln -s /usr/pgsql-9.6/bin/repmgr /usr/sbin/repmgr
ln -s /usr/pgsql-9.6/bin/repmgrd /usr/sbin/repmgrd

masuk menjadi user postgres

su - postgres

buka file postgresql.conf

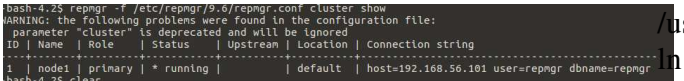
nano /var/lib/pgsql/9.6/data/postgresql.conf

uncomment paramater dibawah dan rubah value nya seperti dibawah

listen_addresses = '*'
max_connections = 1000
max_wal_senders = 10
wal_level = 'hot_standby'
hot_standby = on

 ISSN : 1858 – 3938

archive_mode = on
archive_command = '/bin/true'
wal_keep_segments = 5000
jika sudah save, lalu buka file pg_hba.conf
Nano /var/lib/pgsql/9.6/data/pg_hba.conf
uncomment paramater dibawah dan rubah value nya menjadi seperti dibawah
local replication repmgr trust
host replication repmgr 127.0.0.1/32 trust
host replication repmgr 192.168.56.0/24 trust
local repmgr repmgr trust
ALTER USER repmgr SET search_path TO repmgr_sekolahlinux, "$user", public;
selanjutnya daftarkan node1 menjadi master dengan perintah dibawah
repmgr -f /etc/repmgr/9.6/repmgr.conf master register
untuk melihat apakah kita sudah berhasil mendaftarkan node1 menjadi master bisa dengan cara dibawah ini:
repmgr -f /etc/repmgr/9.6/repmgr.conf cluster show



Gambar 4.3 node1 menjadi *primary*

Gambar di atas menunjukan bahwa node1 sudah berhasil menjadi *primary*

Konfigurasi node2 sebagai slave di repmgr

host repmgr repmgr 127.0.0.1/32 trust
host repmgr repmgr 192.168.56.0/24 trust
restart service postgresql
service postgresql-9.6 restart
selanjutnya buat user & db repmgr pada mode user postgres
createuser -s repmgr
createdb repmgr -O repmgr
pada mode user postgres, buat schema dengan penamaan seperti dibawah, karena nama clusternya sekolahlinux maka saya buat scheme dengan nama repmgr_sekolahlinux seperti dibawah:
psql

Untuk mendaftarkan node2 sebagai slave di repmgr ada beberapa yang harus di konfigurasi sebagai berikut:
Buka file repmgr.conf
Nano /etc/repmgr/9.6/repmgr.conf
uncomment paramater dibawah dan rubah value nya menjadi seperti dibawah:
cluster=sekolahlinux
node=2
node_name=node2
conninfo='host=192.168.56.102 user=repmgr dbname=repmgr'
buat beberapa symlink untuk command repmgr yang kita butuhkan
ln -s /usr/pgsql-9.6/bin/repmgr /usr/sbin/repmgr
ln -s /usr/pgsql-9.6/bin/repmgrd /usr/sbin/repmgrd
masuk menjadi user postgres
su - postgres
Pada tahap ini akan melakukan cloning data dari node1 ke node2 sebelum melakukan cloning untuk login antar server mengimplementasikan ssh authkey

pada node1 dan node2 ikuti perimtah berikut:
ssh-keygen -t rsa
cd .ssh/
chmod 700 ~/.ssh
copy file id_rsa.pub ke dalam file~/.ssh/authorized_keys (buat jika belum ada) di server lainnya. Misal id_rsa.pub node1, maka copykan isinya ke dalam file authorized_keys di node2. Begitu juga sebaliknya saling tukar. Change mode file dan set SElinux untuk permit SSH ini.
chmod 644 authorized_keys id_rsa.pub
restorecon -r /var/lib/pgsql/.ssh/
clone node1/master ke server node2/standby
repmgr -h 192.168.56.101 -U repmgr -d repmgr -D /var/lib/pgsql/9.6/data/

service postgresql-9.6 restart
daftarkan node2 menjadi standby pada mode user postgres
repmgr -f /etc/repmgr/9.6/repmgr.conf standby register
jika sudah selanjutnya coba cek apakah node2 sudah standby
#repmgr -f /etc/repmgr/9.6/repmgr cluster show

```
-bash-4.2$ repmgr -f /etc/repmgr/9.6/repmgr.conf cluster show
WARNING: the following problems were found in the configuration file:
  parameter "cluster" is deprecated and will be ignored
 ID | Name  | Role    | Status    | Upstream | Location | Connection string
----+-------+---------+-----------+----------+----------+-------------------------------------------------------
 1  | node1 | primary | * running |          | default  | host=192.168.56.101 user=repmgr dbname=repmgr
 2  | node2 | standby |   running | node1    | default  | host=192.168.56.102 user=repmgr dbname=repmgr
```

Gambar 4.4 node2 menjadi *standby*

Install PGbouncer

Instalasi pgbouncer install repository terlebih dahulu ikuti perintah berikut;
#install repo postgresql 9.6 pada centos 7
rpm -Uvh https://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-centos96-9.6-3.noarch.rpm
#install postgresql 9.6
yum install postgresql96-server postgresql96 repmgr96 -y
-f /etc/repmgr/9.6/repmgr.conf standby clone
hasil dari command diatas akan seperti dibawah
NOTICE: destination directory '/var/lib/pgsql/9.6/data/' provided
NOTICE: starting backup (using pg_basebackup)...
HINT: this may take some time; consider using the -c/--fast-checkpoint option
NOTICE: standby clone (using pg_basebackup) complete
NOTICE: you can now start your PostgreSQL server
HINT: for example : pg_ctl -D /var/lib/pgsql/9.6/data start
HINT: After starting the server, you need to register this standby with "repmgr standby register"
restart service postgresql
#install pgbouncer dan pgcontrib
yum install -y postgresql92-contrib pgbouncer
selanjutnya buat symlink agar perintah pgbench nantinya bisa digunakan dengan mudah
ln -s /usr/pgsql-9.6/bin/pgbench /usr/sbin/pgbench
buka file pgbouncer.ini lalu lakukan perubahan jika ada paramater yang isinya berbeda dan juga penambahan jika paramater tersebut belum ada, menjadi seperti dibawah
[databases]
test_db = host=192.168.56.103 port=5432 dbname=test_db

[pgbouncer]
logfile = /var/log/pgbouncer/pgbouncer.log
pidfile = /var/run/pgbouncer/pgbouncer.pid

listen_addr = *
listen_port = 6432

auth_type = plain
auth_file = /etc/pgbouncer/userlist.txt
admin_users = test_user
stats_users = stats, postgres, test_user

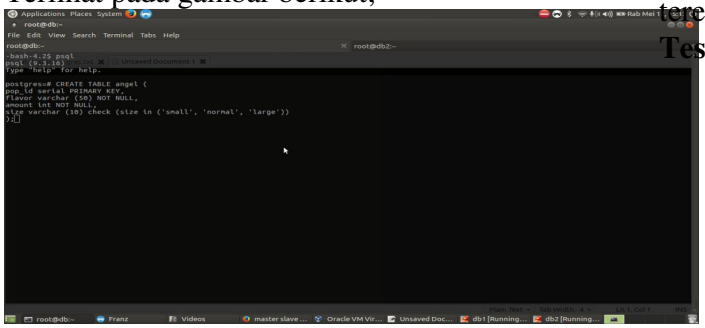
pool_mode = session
max_client_conn = 1000
default_pool_size = 20
buat file userlist.txt
nano /etc/pgbouncer/userlist.txt
tambahkan paramater ini didalamnya
"test_user""123456"
save lalu restart service pgbouncer
service pgbouncer restart

Test Replikasi Master-Slave

Pada tahap ini kita akan melakukan pengujian server database telah di replikasi dengan metode master-slave pada simulasi ini membuat satu database dangan nama angle dengan perintah sebagai berikut;

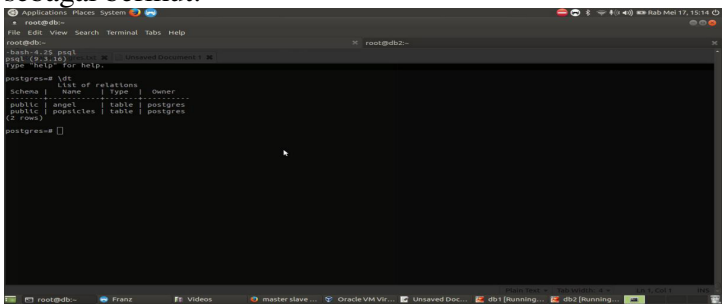
```
CREATE TABLE angel (
pop_id serial PRIMARY KEY,
flavor varchar (50) NOT NULL,
amount int NOT NULL,
size varchar (10) check (size in ('small', 'normal', 'large'))
);
```

Terlihat pada gambar berikut;



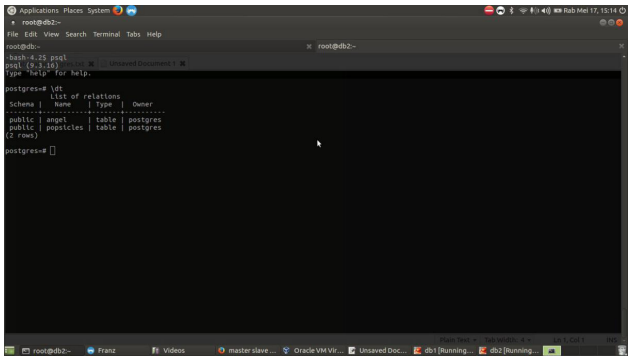
Gambar 4.5 pembuatan table test replikasi

Setelah melakukan pembuatan database maka cek kembali database telah di buat apakah sudah tercreate di server master sebagai berikut:



Gambar 4.6 list table yang di buat

Terlihat di gambar tersebut database telah berhasil di buat maka kita melakukan pengecekan apakah database yang di buat pada server master berhasil teriplikasi dengan baik ke server master sebagai berikut:



Gambar 4.7 pengecekan table di sisi slave server

Pada gambar di atas terlihat database angel yang di create di server master telah tereplikasi dengan baik.

Test Failover REPMGR

Pada tahap ini melakukan pengujian failover database postgres dalam pengujian ini database server master akan di matikan untuk mengetahui apakah failover server berjalan dengan baik dan server postgres yang menjadi standby akan automatis menjadi master atau primary.

Gunakan peritah berikut untuk mematikan server primary atau master

service postgresql-9.6 stop

Kita lihat log pada server slave atau standby server dengan perintah berikut

tail -f repmgr.log

akan nampak log aktifitas seperti berikut:

```
-bash-4.2$ tail -f repmgr.log
[2018-01-22 00:29:23] [NOTICE] using provided configuration file "/etc/repmgr/9.6/repmgr.conf"
[2018-01-22 00:29:23] [NOTICE] starting monitoring of node "node2" (ID: 2)
[2018-01-22 00:29:57] [WARNING] unable to connect to upstream node "node1" (node ID: 1)
[2018-01-22 00:30:47] [WARNING] unable to reconnect to node 1 after 6 attempts
[2018-01-22 00:30:47] [NOTICE] this node is the only available candidate and will now promote itself
[2018-01-22 00:30:47] [NOTICE] promoting standby to primary
[2018-01-22 00:30:47] [DETAIL] promoting server "node2" (ID: 2) using "/usr/pgsql-9.6/bin/pg_ctl  -w -D '/var/lib/pgsql/9.6/data' promote"
[2018-01-22 00:30:49] [NOTICE] STANDBY PROMOTE successful
[2018-01-22 00:30:49] [DETAIL] server "node2" (ID: 2) was successfully promoted to primary
[2018-01-22 00:30:49] [NOTICE] 0 followers to notify
[2018-01-22 00:30:49] [NOTICE] monitoring cluster primary "node2" (node ID: 2)
```

Gambar 4.8 proses pembentukan master baru dari *standby* server

#repmgr -h 192.168.56.102 -U repmgr -d repmgr -D /var/libpgsql/9.6/data -f /etc/repmgr/9.6/repmgr.conf clone

Setelah proses cloning data selesai hidupkan kembali service postgressql

Service postgresql-9.6 restart

Daftarkan kembali node1 menjadi server standby dengan masuk ke user postgres telebih dahulu ketikan perintah berikut:

#repmgr -f /etc/repmgr/9.6/repmgr.conf -D /var/lib/pgsql/9.6/data -h 192.168.56.102 -U repmgr -d repmgr standby follow

Pendaftaran stanby baru telah selesai pada node1 lakukan perintah berikut untuk otomatis promote menjadi primary atau master server

#repmgr -d -f /etc/repmgr/9.6/repmgr.conf --verbose >> $HOME/repmgr/repmgr.log 2>&1

Perintah tersebut akan menulis pada log repmgr.log apabila server primary terjadi error atau mati dan akan melakukan auto promot master atau primary pada server slave atau standby pada node1

Pada gambar di atas terlihat ketika server primary atau master mati maka server standby akan langsung di promot menjadi primary dan server primary atau master akan menjadi failed seperti gambar berikut:

```
-bash-4.2$ repmgr -f /etc/repmgr/9.6/repmgr.conf cluster show
ID | Name  | Role    | Status    | Upstream | Location | Connection string
---+-------+---------+-----------+----------+----------+-----------------------------------------------------
1  | node1 | primary | - failed  |          | default  | host=192.168.56.101 user=repmgr dbname=repmgr
2  | node2 | primary | * running |          | default  | host=192.168.56.102 user=repmgr dbname=repmgr
```

Gambar 4.9 node1 failed node menjadi master baru

Pada gambar di atas menunjukan bahwa primary atau master baru sudah berubah ke node2 atau server standby dan server master yang lama atau node1 menjadi failed atau error karena server postgres mati untuk merubah server node1 menjadi slave di lakukan perintah berikut:

Masuk ke user postgres

Su - postgres

Lakukan backup folder data

mv data dataold

Clone folder data dari primary baru yaitu node 2

```
-bash-4.2$ mv data/ dataold
-bash-4.2$ repmgr -h 192.168.56.102 -U repmgr -d repmgr -D /var/lib/pgsql/9.6/data/ -f /etc/repmgr/9.6/repmgr.conf standby clone
NOTICE: destination directory "/var/lib/pgsql/9.6/data" provided
NOTICE: starting backup (using pg_basebackup)...
HINT: this may take some time; consider using the -c/--fast-checkpoint option
NOTICE: standby clone (using pg_basebackup) complete
NOTICE: you can now start your PostgreSQL server
HINT: for example: pg_ctl -D /var/lib/pgsql/9.6/data start
HINT: after starting the server, you need to re-register this standby with "repmgr standby register --force" to update the existing node rec
-bash-4.2$ logout
[root@node1 ~]# service postgresql-9.6 restart
Redirecting to /bin/systemctl restart postgresql-9.6.service
[root@node1 ~]# su - postgres
Last login: Sen Jan 22 00:32:03 EST 2018 on pts/0
-bash-4.2$ repmgr -f /etc/repmgr/9.6/repmgr.conf -D /var/lib/pgsql/9.6/data/ -h 192.168.56.102 -U repmgr -d repmgr standby follow
NOTICE: setting node 1's primary to node 2
NOTICE: restarting server using "/usr/pgsql-9.6/bin/pg_ctl  -w -D '/var/lib/pgsql/9.6/data' restart"
waiting for server to shut down.... done
server stopped
waiting for server to start....< 2018-01-22 00:39:23.391 EST > LOG:  redirecting log output to logging collector process
< 2018-01-22 00:39:23.391 EST > HINT:  Future log output will appear in directory "pg_log".
 done
server started
NOTICE: STANDBY FOLLOW successful
DETAIL: node 1 is now attached to node 2
-bash-4.2$ 1
-bash: 1: command not found
-bash-4.2$
-bash-4.2$ repmgrd -d -f /etc/repmgr/9.6/repmgr.conf --verbose >> $HOME/repmgr/repmgr.log 2>&1
-bash-4.2$
```

Gambar 4.10 proses register node1 menjadi slave

Setelah melakukan perintah di atas melakukan kembali pengecekan apakah node1 menjadi server standby masih dalam user postgres ketikan perintah

#repmgr -f /etc/repmgr/9.6/repmgr.conf cluster show

Akan tampak seperti gambar berikut:

```
-bash-4.2$ repmgr -f /etc/repmgr/9.6/repmgr.conf cluster show
ID | Name  | Role    | Status    | Upstream | Location | Connection string
---+-------+---------+-----------+----------+----------+-----------------------------------------------------
1  | node1 | standby |   running | node2    | default  | host=192.168.56.101 user=repmgr dbname=repmgr
2  | node2 | primary | * running |          | default  | host=192.168.56.102 user=repmgr dbname=repmgr
-bash-4.2$
```

Gambar 4.11 master baru di node2

Pengujian PGbouncer

sebelum kita melakukan benchmark kita harus generate beberapa table untuk yang nantinya akan digunakan untuk dalam proses benchmark, ikutin perintah dibawah untuk generate tablenya, jika diminta password masukan password yang kita buat untuk user test_user

pgbench -h 192.168.56.101 -p 5432 -i -s 10 -U test_user test_db

hasilnya akan seperti dibawah

Password:
creating tables...
100000 of 1000000 tuples (10%) done (elapsed 0.07 s, remaining 0.63 s)
200000 of 1000000 tuples (20%) done (elapsed 0.21 s, remaining 0.86 s)
300000 of 1000000 tuples (30%) done (elapsed 0.49 s, remaining 1.14 s)
400000 of 1000000 tuples (40%) done (elapsed 1.13 s, remaining 1.70 s)
500000 of 1000000 tuples (50%) done (elapsed 1.78 s, remaining 1.78 s)
600000 of 1000000 tuples (60%) done (elapsed 2.51 s, remaining 1.67 s)
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 10
query mode: simple
number of clients: 10
number of threads: 1
duration: 60 s
number of transactions actually processed: 5087
latency average = 118.006 ms
tps = 84.741672 (including connections establishing)
tps = 91.730603 (excluding connections establishing)

jadi dari hasil benchmark diatas pada jumlah koneksi 10 dan durasi 60 detik terlihat jelas bahwa jika kita menggunakan pgbouncer hasil tps nya lebih besar dan latency nya juga lebih kecil,

700000 of 1000000 tuples (70%) done (elapsed 3.36 s, remaining 1.44 s)
800000 of 1000000 tuples (80%) done (elapsed 3.94 s, remaining 0.98 s)
900000 of 1000000 tuples (90%) done (elapsed 4.80 s, remaining 0.53 s)
1000000 of 1000000 tuples (100%) done (elapsed 5.54 s, remaining 0.00 s)
vacuum...
set primary keys...
Done.

jika sudah selanjutkan coba kita benchmark dari node pgbouncer, pertama kita benchmark langsung ke server db postgresql tanpa melewati pgbouncer dengan perintah seperti dibawah, jika diminta password masukan password yang kita buat untuk user test_user

pgbench -h 192.168.56.101 -c 10 -C -T 60 -p 5432 -U test_user test_db

hasilnya akan seperti dibawah

Password:

bagaimana jika node1 repmgr down, yang harus kalian lakukan pertama kali adalah melakukan promote master pada node2 repmgr slave agar menjadi master untuk caranya rubah host pada pengaturan db di pgbouncer.ini jika sudah refresh rulenya dengan perintah dibawah ini

pgbouncer -d pgbouncer.ini -R

Kesimpulan

Dari simulasi yang di lakukan dapat ditarik kesimpulan tentang pembangunan sistem replikasi database dengan postgresql pada sistem operasi linux centOs 7 sebagai berikut:

1. Telah berhasil dibangun sebuah system replikasi database dengan menggunakan metode master-slave

dengan postgresql pada system operasi linux CentOs 7
2. System replikasi berjalan dengan baik pada system yang memiliki 2 buah server database yang masing-masing bertindak sebagai master server dan slave server.
3. System failover dengan repmgr dapat berjalan dengan baik.

Saran

Dari simulasi yang di lakukan terdapat beberapa kekurangan yang bisa di kembangkan oleh yang membaca laporan tugas akhir ini sebagai berikut:

1. Perlu dikembangkan kembali untuk server backup secara incremental
2. Dibuatkan sistem monintoring server database ketika terjadi down

Perlu dibuatkan otomasi pengarahan di sisi pgbouncer

DAFTAR PUSTAKA

1. A Ubaidillah, 2009. Perancangan dan Implementasi Sistem Database Terdistribusi Studi Kasus SIAKAD Universitas Trunojoyo, Skripsi Sarjana, Universitas Trunojoyo Madura.
2. Ashok,G., Randal, P.S. (2008). SQL Server Replication: Providing High Availability using Database Mirroring, Microsoft.
3. Ayyasammy, Sivannandam. 2010. A cluster Based Replication Architectur for Load Balancing in Peer to Peer Content Distribution. International Journal of Computer Network & Communication (IJCNC) Vol.2. No.5. September 210.
4. Cristian. Marius. 2010. Database Replication. Database System Journal vol. I no. 2/2010.
5. Kadir, Abdul. 2009. Dasar Perancangan dan Implementasi Database Relasional. Yogyakarta: ANDI.
6. Kemme, B. & Alonso, G. (2010). Database Replication: a Tale of
7. Research across Communities. Proceedings of the VLDB Endowment. Vol. 3. No. 1.
8. Mens, Jan Piet, 2009. Alternative DNS Servers. London. UIT Cambridge Ltd.
9. Mulyana, Deddy. 2008. Metodologi Penelitian Kualitatif. Bandung: Remaja Rosdakarya.
10. Ren, Kun., Thompson, Alexander., J.Abadi, Daniel. 2014. An Evalutation of the Advantages and Disadvantages of Determenistic Database System. Proceedings of the VLDB Endowment. Vol. 7. No. 10
11. Sugiyono. 2011. Metode Penelitian Pendidikan: Pendekatan Kuantitatif, Kualitatif, dan R&D. Bandung: Alfabeta.
12. Sugiyono. 2012. Metode Penelitian Pendidikan: Pendekatan Kuantitatif, Kualitatif, dan R&D. Bandung: Alfabeta.
13. Sukardi. 2013. Metodologi Penelitian Pendidikan Kompetensi dan Praktiknya, Jakarta: Penerbit Bumi Aksara.
14. Sukmadinata, Nana Syaodih. 2013. Metode Penelitian Pendidikan, Bandung: PPS UPI dan PT Remaja Rosdakarya.
15. Zuriah, Nurul. 2006. Metodologi Penelitian Sosial dan Pendidikan: Teori-Aplikasi. Jakarta: Bumi Aksara.

 ISSN : 1858 – 3938

16. Zuriah, Nurul. 2009. Metodologi Penelitian Sosial dan Pendidikan. Jakarta: Bumi Aksara.

.

Penulis adalah

* Dosen pada Sekolah Tinggi Teknologi Informasi NIIT I-Tech

** Alumni Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Informasi NIIT I-Tech