

Penerapan Replikasi Data pada Aplikasi Ticketing Menggunakan Slony PostgreSQL

Defriyanuar Dhining^{1*}, Yeni Rokhayati^{2**}, Dwi Ely Kurniawan^{3*}

* Teknik Informatika, Politeknik Negeri Batam

** Teknik Multimedia Jaringan, Politeknik Negeri Batam

defridhining@gmail.com¹, yeni@polibatam.ac.id², dwiialikhs@polibatam.ac.id³

Article Info

Article history:

Received 17-08-2017

Revised 09-09-2017

Accepted 18-10-2017

Keyword:

Replikasi Database

Internet Connection

PostgreSQL

ABSTRAK

Nowadays, almost every company use web application to do their business activity. Besides multi-platform it also easier for installation and maintenance. Other than web application need a fast and reliable internet connection. It is necessary to make a server in local area network with synchronize database from one server to another server. Synchronize database will run using replication server database. Database replication can use many tool, one of them is Slony that is specially create for PostgreSQL. Replication database can also to pretend stopping application when there is bad internet connection. Replication database is one of standby server technique because down server. Database synchronize will be build using PHP programming language and PostgreSQL database with Slony which is open source at all, so it will reduce the cost of installation and maintenance.

Copyright © 2017 Journal of Applied Informatics and Computing.
All rights reserved.

I. PENDAHULUAN

Saat ini hampir setiap perusahaan menggunakan internet untuk melakukan aktivitas bisnisnya. Baik menggunakan aplikasi dari perusahaan itu sendiri ataupun menggunakan aplikasi umum, hampir seluruhnya menggunakan internet. Penggunaan internet pasti akan memudahkan integrasi data antar proses bisnis perusahaan cabang sesuai dengan standar prosedur kerja. Namun, sering muncul kendala koneksi internet yang tidak stabil ataupun lambat, sehingga akan mengakibatkan kantor cabang tidak bisa membuka aplikasi yang berada di kantor pusat.

Diasumsikan terdapat PT. XYZ yang bergerak di bidang penjualan tiket kapal feri Batam - Singapore. PT. XYZ memiliki dua kantor penjualan tiket yang terletak di sebuah pulau Batam, sehingga membutuhkan aplikasi ticketing yang dapat saling terhubung pada kedua kantor tersebut. PT. XYZ menggunakan aplikasi berbasis web. PT. XYZ perlu menggunakan replikasi basisdata server atau dengan kata lain membuat basisdata perusahaan menjadi basisdata yang sinkron. Salah satu solusi permasalahan tersebut adalah dengan menggunakan fitur slony pada PostgreSQL. Alasan menggunakan slony adalah untuk mengantisipasi kendala aplikasi lambat apabila tiba-tiba koneksi internet melambat ataupun aplikasi malah tidak dapat diakses sama sekali ketika koneksi internet terputus. Jadi, fitur slony akan melakukan

replika basisdata pada setiap server perusahaan sehingga basisdata setiap server menjadi sinkron.

Penelitian yang terkait, merancang sistem informasi presensi perkuliahan menggunakan basis data terdistribusi dengan metode replikasi asynchronous [2], menerapkan replikasi data asynchronous untuk keamanan data sistem informasi akademik online [3], merancang sistem replikasi basisdata MySQL menggunakan VMWare dan ubuntu server [1], [11], melakukan replikasi basisdata saat pengiriman data secara terpusat pada pendataan inventaris menggunakan metode replikasi asynchronous [4], [11]. Beberapa penelitian tersebut menjadi dasar untuk mencoba mengimplementasikan replikasi basisdata pada PostgreSQL dengan memanfaatkan fitur slony dan mesin virtualisasi VirtualBox untuk pengujian sistem.

II. LANDASAN TEORI

A. Basis Data

Basis data adalah sebuah metode mendokumentasikan berbagai macam data yang kemudian dimanajemen dengan sebuah sistem untuk kemudian disimpan dalam sebuah media penyimpanan, seperti hardisk. Dengan hal ini, data-data dapat diakses dengan mudah dan cepat. Dalam basis data, data tidak hanya disimpan begitu saja dalam sebuah media penyimpanan, tetapi dikelola dengan sebuah sistem

pengaturan basisdata yang sering disebut dengan Database Management System. Bagian-bagian yang berperan penting dalam perancangan sebuah basisdata diantaranya; (1) kolom merupakan bentuk terkecil dalam basisdata yang digunakan untuk menyimpan data dan memisahkan basisdata dalam cakupan yang lebih kecil; (2) primary key merupakan bentuk kolom utama yang mewakili kolom lainnya dalam sebuah tabel untuk menghindari penggadaan data. Proses mewakili dalam tabel tersebut dapat disebut dengan tabel yang berelasi atau basisdata relasional; (3) foreign key merupakan kolom yang berasal dari tabel lain. Tabel memiliki satu kolom utama (primary key). Apabila primary key ini berelasi dengan tabel lain maka disebut foreign key dalam basisdata lawan tersebut.

B. PostgreSQL

PostgreSQL atau sering disebut Postgres merupakan salah satu dari sejumlah basisdata besar yang menawarkan skalabilitas, keluwesan, dan kinerja yang tinggi. Penggunaannya begitu meluas di berbagai platform dan didukung oleh banyak bahasa pemrograman. Bagi masyarakat TI (teknologi informasi) di Indonesia, Postgres sudah digunakan untuk berbagai aplikasi seperti web, billing system, dan sistem informasi besar lainnya.

Pemilihan basisdata PostgreSQL yang dilakukan penulis disini adalah dengan tujuan untuk menghemat waktu dalam pembuatan server yang menerapkan replikasi basisdata karena di dalam PostgreSQL itu sendiri terdapat pengembangan aplikasi slony yang berguna untuk memudahkan melakukan replikasi basisdata.

C. Slony

Slony-I adalah sebuah "master to multiple slaves" sistem replika untuk PostgreSQL yang bisa menangani kesalahan pada sistem jika tidak terkoneksi dengan master. Gambaran dari perkembangan Slony-I adalah replika sistem master-slave yang menyediakan semua fitur dan keunggulan yang dibutuhkan untuk mereplika basisdata dengan skala besar. Slony-I juga bisa digunakan sebagai data pusat and backup. Berikut perbandingan slony dengan solusi replikasi lain pada PostgreSQL.

	BDR	HOT STANDBY	LONDISTE	SLONY	BUCARDO
Multi-master	Yes	No	No ¹	No	Yes
Per DB Replication	Yes	No	Yes	Yes	Yes
Cascading	No ²	Yes	Yes	Yes	Unknown
DDL Replication	Yes	Yes	No ²	No ²	No
Needs External Daemon	No	No	Yes	Yes	Yes
New Table Added Automatically	Yes	Yes	No	No	No
Transparent Sequences	Yes	N/A	N/A	N/A	No
Uses triggers/double writes	No	No	Yes	Yes	Yes
Supports updates on PK columns	Yes	Yes	No	No	No
Selective Replication	Yes	No	Yes	Yes	Yes
Transactions applied individually	Yes	Yes	No	No	No

Gambar 1. Tabel Perbandingan slony dengan replikasi lain pada PostgreSQL.

D. Virtual Box

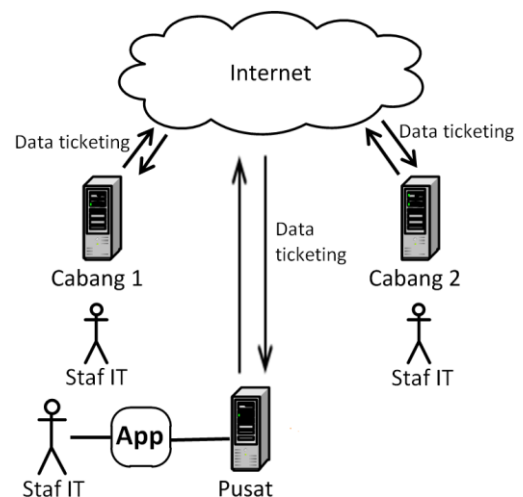
Oracle VM VirtualBox atau sering disebut dengan VirtualBox merupakan salah satu produk perangkat lunak yang sekarang dikembangkan oleh Oracle. Aplikasi ini pertama kali dikembangkan oleh perusahaan Jerman, Innotek GmbH. Pebruari 2008, Innotek GmbH diakusisi oleh Sun Microsystems. Sun Microsystem kemudian juga diakuisisi oleh Oracle. VirtualBox berfungsi untuk melakukan virtualisasi sistem operasi. VirtualBox juga dapat digunakan untuk membuat virtualisasi jaringan komputer sederhana. Penggunaan VirtualBox ditargetkan untuk Server, desktop dan penggunaan embedded. Berdasarkan jenis VMM yang ada, Virtualbox merupakan jenis hypervisor type 2.

Oracle VM VirtualBox adalah perangkat lunak virtualisasi, yang dapat digunakan untuk mengeksekusi sistem operasi "tambahan" di dalam sistem operasi "utama". Sebagai contoh, jika seseorang mempunyai sistem operasi MS Windows yang terpasang di komputernya, maka yang bersangkutan dapat pula menjalankan sistem operasi lain yang diinginkan di dalam sistem operasi MS Windows tersebut. Fungsi ini sangat penting jika seseorang ingin melakukan ujicoba dan simulasi instalasi suatu sistem tanpa harus kehilangan sistem yang ada.

III. ANALISIS DAN PERANCANGAN

A. Deskripsi Umum Sistem

PT. XYZ merupakan perusahaan yang bergerak di bidang penjualan tiket Kapal Feri Batam-Singapura. Perusahaan tersebut memiliki beberapa cabang penjualan tiket yang tersebar di seluruh pulau Batam. Sehingga perusahaan perlu menggunakan aplikasi Slony PostgreSQL, yang dapat mereplika server basisdata atau dengan kata lain membuat basisdata perusahaan menjadi basisdata terdistribusi. Penjualan tiket menggunakan aplikasi berbasis web yang terintegrasi pada setiap perusahaan cabang.

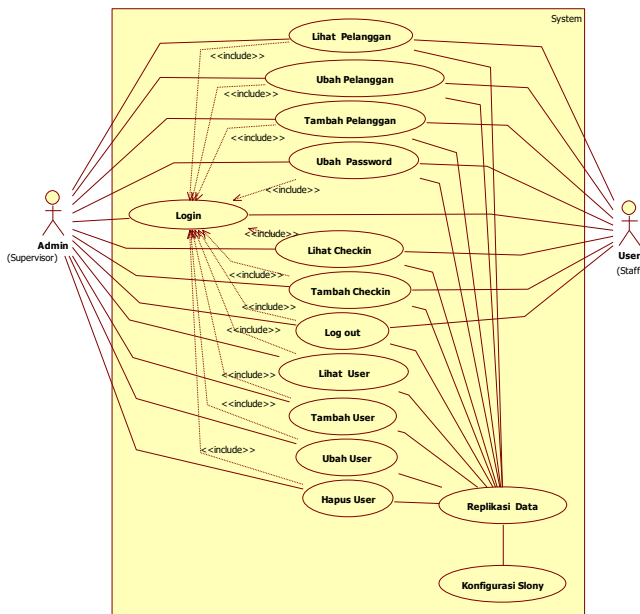


Gambar 2. Simulasi Replikasi Data

Perusahaan mempunyai server cabang dan server pusat, jadi ketika pengguna hendak membuka aplikasi web

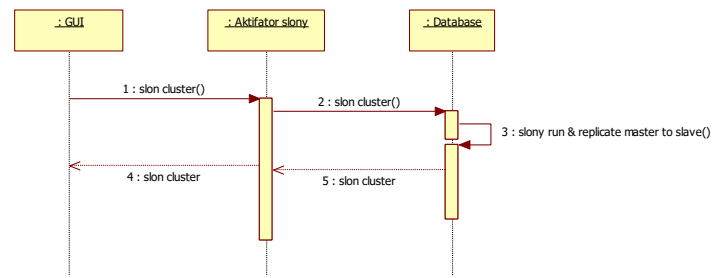
perusahaan, browser hanya perlu mengakses pada server cabang (jaringan lokal). Selanjutnya ketika ada penambahan data baru ataupun perubahan data di server cabang, maka slony mengirimkan atau mereplika data tersebut dari server cabang ke server pusat. Dalam hal ini yang dikirimkan hanya perubahan data pada basisdata sehingga tidak membutuhkan koneksi sistem yang tinggi, mengingat ukuran file basisdata yang relatif kecil. Begitu juga ketika kantor cabang tidak terkoneksi dengan sistem, aplikasi masih bisa digunakan dengan mengakses server lokal. Namun, kantor cabang yang tidak terkoneksi dengan sistem harus segera ditindaklanjuti. Aplikasi pada kantor cabang tersebut tidak boleh dibiarkan untuk tidak terkoneksi dengan sistem dalam kurun waktu yang lama karena akan mengakibatkan data tidak sinkron atau tidak terbaharui.

Pengguna yang terlibat dalam sistem adalah admin dan user (staf). Admin dan user harus login terlebih dahulu untuk menggunakan aplikasi ini. Setelah login, admin bisa melihat status koneksi internet server, mengubah data kantor, mengubah password, mengelola staf IT dan mengelola manager. Pengelolaan staf IT mencakup menambah, mengubah, menghapus, dan menampilkan user. Pengelolaan manager mencakup menambah, mengubah, menghapus, dan menampilkan manager. Untuk user hanya bisa melihat status koneksi internet server.



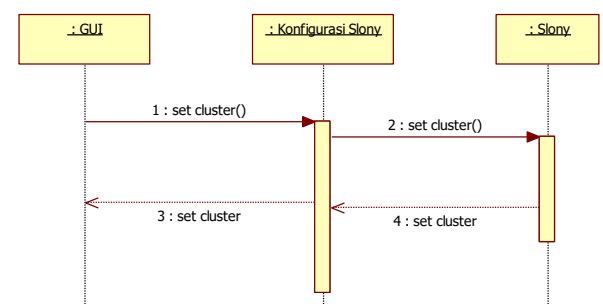
Gambar 3. Diagram Use Case Replikasi Basisdata.

Pengguna harus login terlebih dahulu untuk menggunakan aplikasi ini. Setelah login, aktor dapat melakukan lihat pelanggan, tambah pelanggan, ubah pelanggan, lihat checkin, buat checkin, ubah password, lihat user, tambah user, ubah user, hapus user, konfigurasi slony, replikasi data, dan logout.



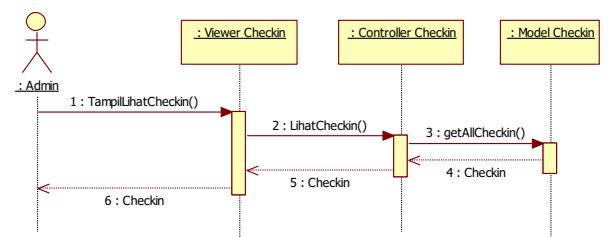
Gambar 4. Sequence diagram replikasi basisdata

Sequence diagram menjelaskan proses replikasi data pada aplikasi. Pertama aplikasi menampilkan halaman utama. Setelah itu aktor menjalankan slony untuk replikasi data. Selanjutnya slony terhubung dengan basisdata dan kemudian slony menampilkan tampilan running.



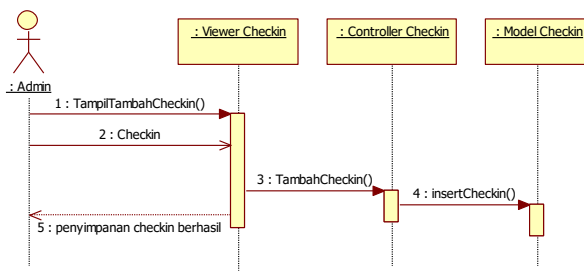
Gambar 5. Sequence diagram konfigurasi slony

Sequence diagram menjelaskan proses aktor melakukan konfigurasi pada slony. Pertama aktor harus membuat file konfigurasi untuk slony. Setelah itu aktor menjalankan file konfigurasi tersebut untuk menyimpan konfigurasi dalam slony. Selanjutnya slony menyimpan konfigurasi sesuai dengan parameter yang terdapat dalam file konfigurasi.



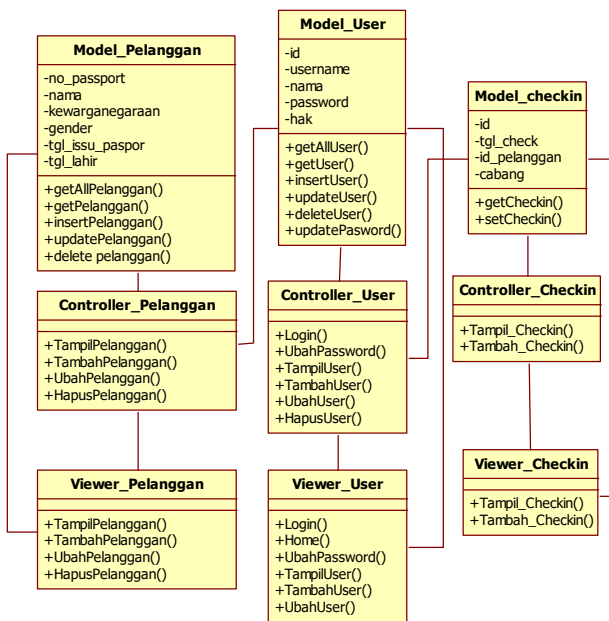
Gambar 6. Sequence Diagram Lihat Checkin

Sequence diagram ini menjelaskan proses aktor melakukan lihat checkin pada aplikasi. Aktor harus mengklik tombol checkin kemudian aplikasi menampilkan semua data checkin.



Gambar 7. Sequence Diagram Tambah Checkin

Sequence diagram ini menjelaskan proses aktor melakukan lihat checkin pada aplikasi. Aktor harus mengklik tombol checkin kemudian aplikasi menampilkan semua data checkin.

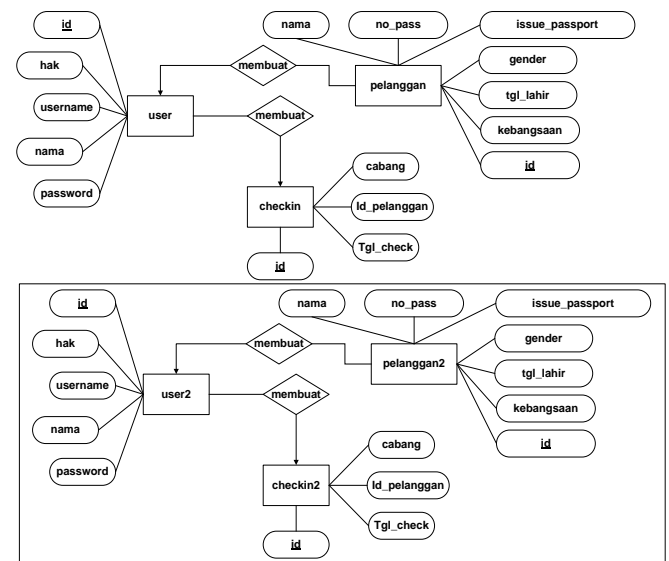


Gambar 8. Sequence diagram konfigurasi slony

Diagram kelas terdapat 6 kelas dan 3 model yang mempunyai kegunaan masing-masing antara lain.

- 1) Model_user merupakan kelas model yang merepresentasikan data dari tabel database user dan data tambahan untuk pengoperasiannya,
- 2) Kelas controller_user berguna untuk menyediakan fungsi-fungsi yang akan diteruskan oleh kelas viewer_user,
- 3) Kelas viewer_user berguna untuk menampilkan antarmuka login, home, dan yang berhubungan dengan olah user,
- 4) Model_pelanggan merupakan kelas model yang merepresentasikan data dari tabel database pelanggan dan data tambahan untuk pengoperasiannya,

- 5) Kelas controller_pelanggan berguna untuk menyediakan fungsi-fungsi yang akan diteruskan oleh kelas viewer_pelanggan,
- 6) Kelas viewer_pelanggan berguna untuk menampilkan antarmuka pengelolaan data pelanggan,
- 7) Model_checkin merupakan kelas model yang merepresentasikan data dari tabel database checkin dan data tambahan untuk pengoperasiannya,
- 8) Kelas controller_checkin berguna untuk menyediakan fungsi-fungsi yang akan diteruskan oleh kelas viewer_checkin,
- 9) Kelas viewer_checkin berguna untuk menampilkan antarmuka pengelolaan data checkin.



Gambar 9. ER Diagram Aplikasi Ticketing Cabang BTC

Pada dasarnya aplikasi ticketing mempunyai 3 tabel yang mempunyai kegunaan masing-masing dan tabel tersebut diduplikasi menjadi 6 tabel karena 3 tabel yang merupakan master server BTC (Batam Center) dan 3 tabel lagi yang berada dalam kotak sebagai slave dari server SPS (Sungai Panas). Deskripsi tabel-tabel tersebut antara lain:

ER diagram pertama merupakan master server BTC (Batam Center) dimana aplikasi dapat melakukan perubahan data, tiga tabel tersebut antara lain;

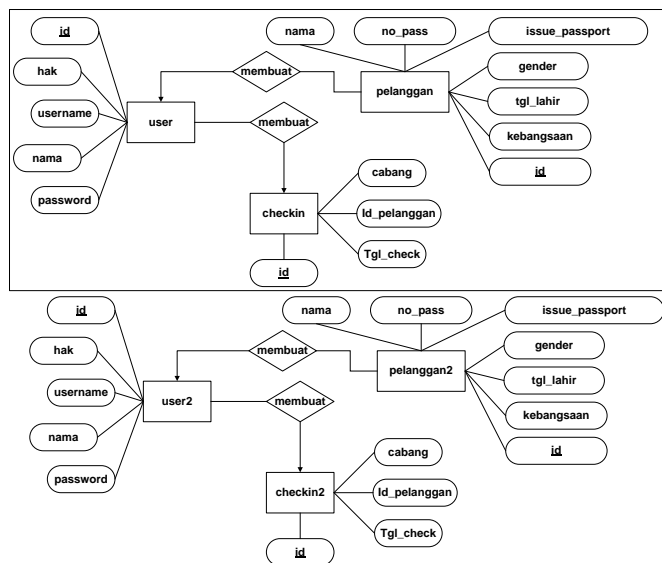
- 1) Tabel pelanggan mempunyai 7 atribut yang berkaitan yaitu nama yang berarti nama pelanggan, no_passport yang berarti no passport pelanggan, tanggal_issue_passport yang berarti tanggal pengeluaran passport pelanggan, gender yang berarti jenis kelamin pelanggan, id yang berarti id pelanggan, kebangsaan yang berarti kebangsaan pelanggan, tempat_lahir yang berarti tempat lahir pelanggan, tgl_lahir yang berarti tanggal lahir pelanggan

- 2) Tabel user mempunyai 4 atribut yang berkaitan yaitu Username yang berarti username yang digunakan untuk login, Password yang berarti password yang digunakan untuk login, Nama yang berarti nama panjang user, hak yang berarti hak akses untuk membatasi akses pengguna.
- 3) Tabel checkin mempunyai 4 atribut yang berkaitan yaitu id yang berarti id checkin, id_pelanggan, tgl_check, dan cabang.

ER diagram kedua merupakan slave dari server SPS (Sungai Panas) dimana aplikasi hanya dapat melakukan pembacaan data, tiga tabel tersebut antara lain;

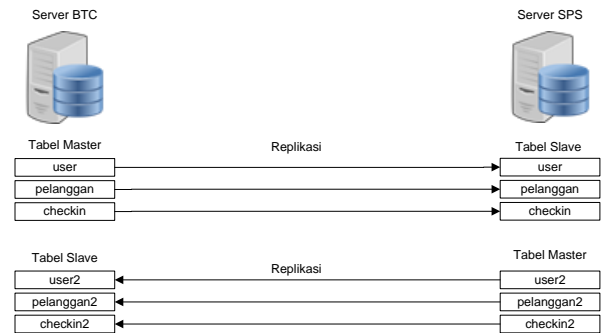
- 1) Tabel pelanggan2 mempunyai 7 atribut yang berkaitan yaitu nama yang berarti nama pelanggan, no_passport yang berarti no passport pelanggan, tanggal_issue_passport yang berarti tanggal pengeluaran passport pelanggan, gender yang berarti jenis kelamin pelanggan, id yang berarti id pelanggan, kebangsaan yang berarti kebangsaan pelanggan, tempat_lahir yang berarti tempat lahir pelanggan, tgl_lahir yang berarti tanggal lahir pelanggan
- 2) Tabel user2 mempunyai 4 atribut yang berkaitan yaitu Username yang berarti username yang digunakan untuk login, Password yang berarti password yang digunakan untuk login, Nama yang berarti nama panjang user, hak yang berarti hak akses untuk membatasi akses pengguna.
- 3) Tabel checkin2 mempunyai 4 atribut yang berkaitan yaitu id yang berarti id checkin, id_pelanggan, tgl_check, dan cabang.

Begitu sebaliknya untuk ER Diagram aplikasi ticketing pada cabang SPS (Sungai Panas). Pada dasarnya mempunyai 3 tabel dan diduplikasi menjadi 6 tabel karena 3 tabel yang merupakan master server SPS (Sungai Panas) dan 3 tabel, lagi yang berada dalam kotak sebagai slave dari server BTC (Batam Centre).



Gambar 10. ER Diagram Aplikasi Ticketing Cabang SPS

Berikut ilustrasi untuk menjelaskan perancangan replikasi tabel pada setiap server.

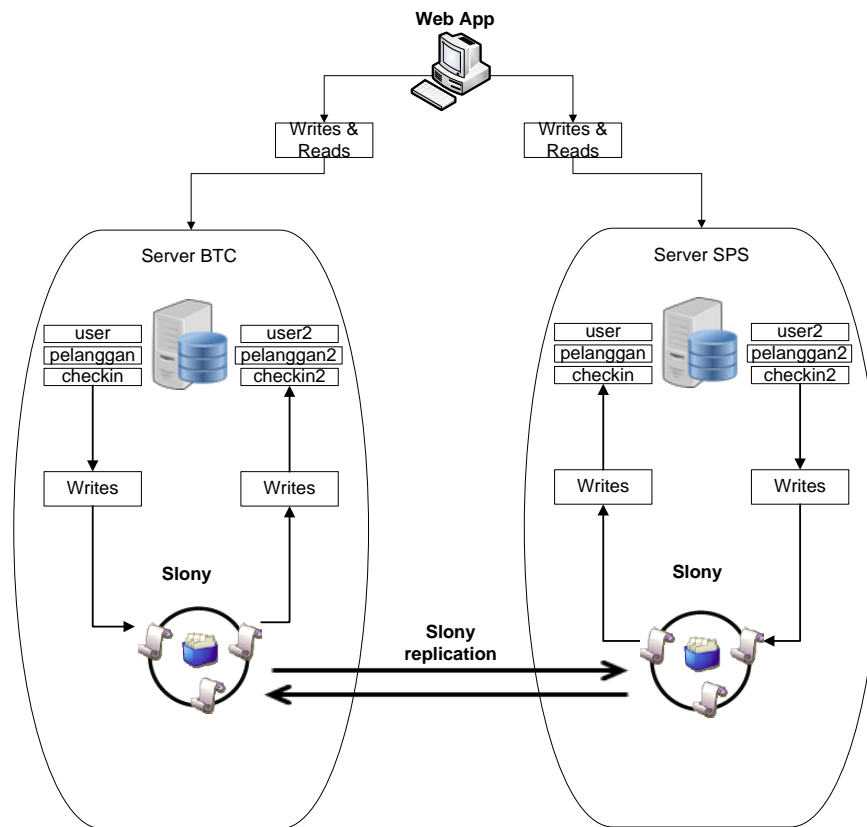


Gambar 11. Replikasi Data pada Server BTC dan SPS

Replikasi basisdata tidak boleh dibiarkan untuk tidak terkoneksi dengan sistem dalam kurun waktu yang lama karena akan mengakibatkan data tidak sinkron atau tidak terbaharui, data penjualan tiket akan tertahan pada server cabang dan akan mengakibatkan data tidak tersinkronisasi dengan baik.

IV. IMPLEMENTASI DAN HASIL

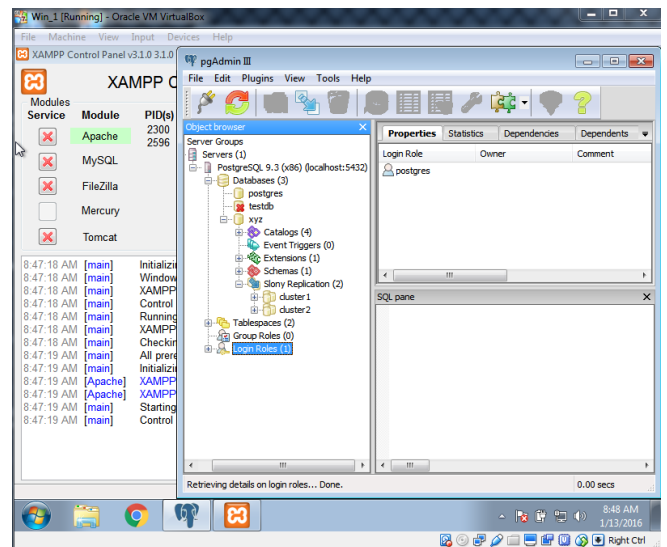
Agar dapat menggunakan fitur slony, PT. XYZ harus mempunyai server lokal pada kedua kantornya. Jadi ketika pengguna hendak membuka aplikasi web perusahaan, browser hanya perlu mengakses aplikasi web pada server lokal (jaringan lokal). Selanjutnya ketika ada penambahan data baru ataupun perubahan data di server cabang, maka aplikasi Slony akan mengirimkan atau mereplika data tersebut dari server cabang ke server pusat. Dalam hal ini yang dikirimkan hanya perubahan data pada database saja sehingga tidak membutuhkan koneksi internet yang tinggi mengingat ukuran file database yang relatif kecil. Begitu juga ketika perusahaan cabang tidak terkoneksi dengan internet, aplikasi masih bisa digunakan karena aplikasi diakses secara local pada server di masing-masing kantor.



Gambar 12. Sistem Replikasi Data pada Server BTC dan SPS

Server BTC dan SPS memiliki struktur tabel yang sama terintegrasi pada aplikasi web dan masing-masing server telah tersedia fitur slony. Server BTC memiliki akses untuk menulis data pada table user, pelanggan, dan checkin tetapi tidak dapat menulis pada table user2, pelanggan2, dan checkin2 karena table tersebut merupakan replikasi dari server SPS. Setiap perubahan data pada table user, pelanggan, dan checkin akan ditulis di slony yang kemudian slony akan mereplikasi data tersebut ke server SPS.

Selanjutnya berlaku kebalikan dari proses server BTC pada server SPS. Server SPS memiliki akses untuk menulis data pada table2 user2, pelanggan2, dan checkin tetapi tidak dapat menulis pada table user, pelanggan, dan checkin karena table tersebut merupakan replikasi dari server SPS. Setiap perubahan data pada table user2, pelanggan2, dan checkin2 akan ditulis di slony yang kemudian slony akan mereplikasi data tersebut ke server BTC. Proses replikasi berjalan dengan syarat kedua server tersebut saling terkoneksi. Jika koneksi terputus, perubahan data akan disimpan dahulu pada masing-masing server dan kemudian akan direplikasi jika kedua server tersebut kembali terkoneksi.



Gambar 13. Implementasi Server BTC pada Mesin Virtual

Pada gambar di atas terdapat dua server virtual yang menggunakan sistem operasi Windows 7 diantaranya yaitu, server 1 yang merupakan server BTC (Batam Center) dan server 2 yang merupakan server SPS (Sungai Panas). Server BTC berperan sebagai server pusat dan server SPS berperan sebagai server cabang. Pada kedua server tersebut sudah

terpasang dan terkonfigurasi Web Server, PostgreSQL, dan Slony sebagai sarana replikasi data. Konfigurasi slony mengarahkan database mana saja yang akan direplikasi. Berikut code untuk konfigurasi slony pada server 1 BTC.

```
cluster name = cluster1;

node 1 admin conninfo = 'dbname=xyz
host=10.10.3.165 user=postgres password=postgres
port=5432';
node 2 admin conninfo = 'dbname=xyz
host=10.10.3.166 user=postgres password=postgres
port=5432';

init cluster ( id=1, comment='Master Node');

create set (id=1, origin=1, comment='Daftar
Tabel');

set add table (set id=1, origin=1, id=1, fully
qualified name='public.checkin', comment='tabel
checkin');
set add table (set id=1, origin=1, id=2, fully
qualified name='public.pelanggan',
comment='tabel pelanggan');
set add table (set id=1, origin=1, id=3, fully
qualified name='public.user', comment='tabel
user');

store node (id=2, event node=1, comment='Slave
Node');

store path (server=1, client=2,
conninfo='dbname=xyz host=10.10.3.165
user=postgres password=postgres port=5432');
store path (server=2, client=1,
conninfo='dbname=xyz host=10.10.3.166
user=postgres password=postgres port=5432');
```

Konfigurasi slony di atas adalah untuk men-set cluster 1 sebagai cluster yang akan direplikasi. Server 1 BTC sebagai master dan Server 2 SPS sebagai slave. Selain itu juga terdapat create set untuk men-set table apa saja yang akan di-set. Terakhir store node untuk tujuan replikasi beserta path yang berupa alamat ip master dan slave. Konfigurasi di atas disimpan dalam sebuah file master.txt dan kemudian harus di eksekusi menggunakan command slonik yang didapat dari slony dengan cara “slonik master.txt”, maka konfigurasi di atas akan disimpan dalam slony. Selanjutnya slave SPS harus menjalankan konfigurasi subscribe untuk menerima konfigurasi dari master BTC. Berikut konfigurasi subscriber slave SPS.

```
cluster name = cluster1;

node 1 admin conninfo = 'dbname=xyz
host=10.10.3.165 user=postgres
password=postgres port=5432';
node 2 admin conninfo = 'dbname=xyz
host=10.10.3.166 user=postgres
password=postgres port=5432';
```

```
subscribe set (id=1,
provider=1, receiver=2, forward=yes);
```

Begitu juga sebaliknya untuk server 2 SPS perlu dikonfigurasi master.txt seperti berikut.

```
cluster name = cluster2;

node 1 admin conninfo = 'dbname=xyz
host=10.10.3.166 user=postgres password=postgres
port=5432';
node 2 admin conninfo = 'dbname=xyz
host=10.10.3.165 user=postgres password=postgres
port=5432';

init cluster ( id=1, comment='Master Node');

create set (id=1, origin=1, comment='Daftar
Tabel');

set add table (set id=1, origin=1, id=1, fully
qualified name='public.checkin2',
comment='tabel2 checkin2');
set add table (set id=1, origin=1, id=2, fully
qualified name='public.pelanggan2',
comment='tabel pelanggan2');
set add table (set id=1, origin=1, id=3, fully
qualified name='public.user2', comment='tabel
user2');

store node (id=2, event node=1, comment='Slave
Node');

store path (server=1, client=2,
conninfo='dbname=xyz host=10.10.3.166
user=postgres password=postgres port=5432');
store path (server=2, client=1,
conninfo='dbname=xyz host=10.10.3.165
user=postgres password=postgres port=5432');
```

Konfigurasi di atas adalah merupakan table master SPS, yang berisi tentang alamat masing-masing server serta table apa yang akan dijadikan master dan direplikasi ke server BTC.

```
cluster name = cluster1;

node 1 admin conninfo = 'dbname=xyz
host=10.10.3.165 user=postgres password=postgres
port=5432';
node 2 admin conninfo = 'dbname=xyz
host=10.10.3.166 user=postgres password=postgres
port=5432';

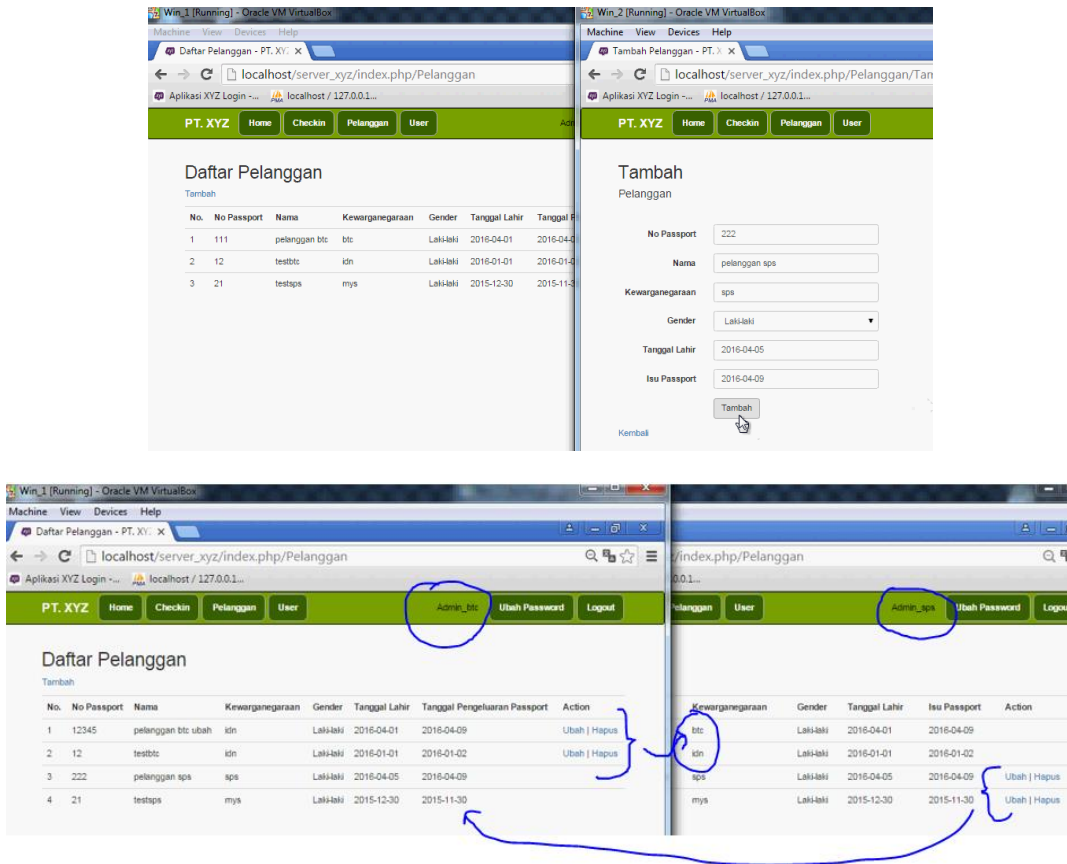
subscribe set (id=1,
provider=1, receiver=2, forward=yes);
```

Konfigurasi di atas adalah konfigurasi untuk tabel slave yang merupakan replikasi tabel dari server BTC. Setelah semua server dikonfigurasi seperti di atas selanjutnya perlu dijalankan slony untuk mereplikasi database kedua server.

Berikut command untuk menjalankan slony di masing-masing server:

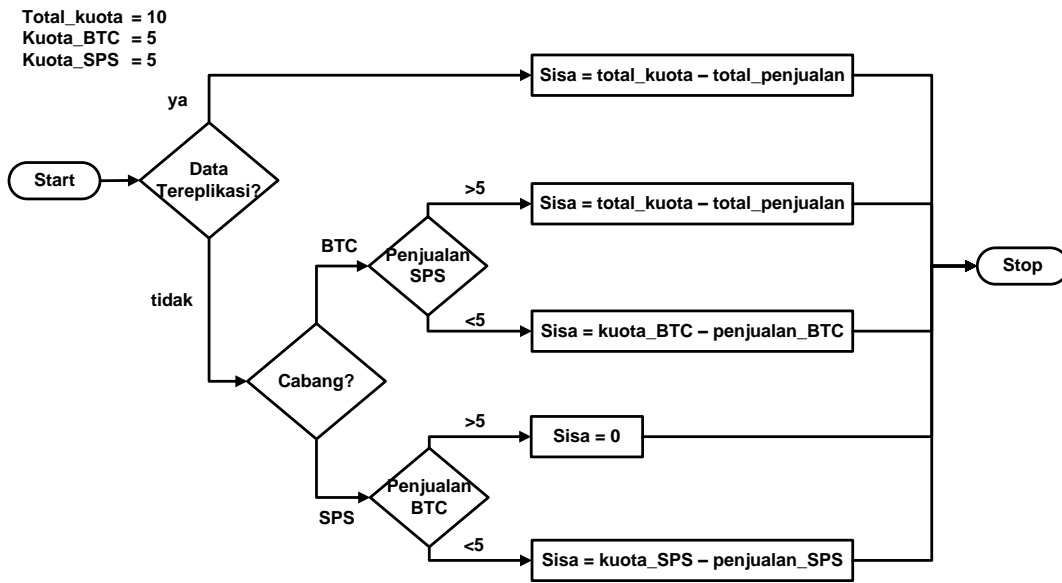
- Slon cluster1 "dbname=xyz user=postgres password=postgres"
- Slon cluster2 "dbname=xyz user=postgres password=postgres"

Kedua command diatas untuk menjalankan replikasi masing-masing server.



Gambar 14. Replikasi dengan Sinkronisasi Data Pelanggan Kedua Server

Kedua gambar di atas data checkin akan sama antara server 1 BTC dan server 2 SPS. Hal ini menandakan replikasi antara kedua server dapat berjalan. Namun jika koneksi kedua server terputus maka akan timbul masalah kelebihan checkin pada waktu yang sama. Dengan kata lain kedua server tidak tahu data checkin yang sudah masuk karena koneksi terputus. Untuk mencegah hal itu terjadi maka dibuat algoritma pembatasan untuk kuota checkin jika kedua server terputus.

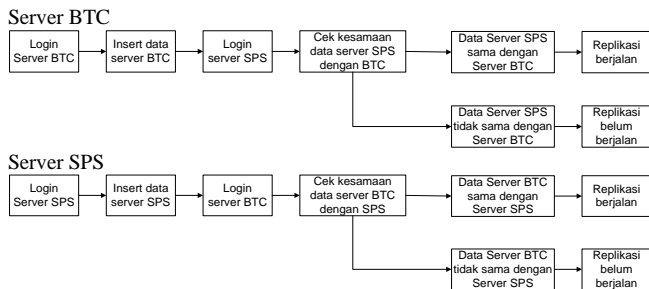


Gambar 15. Flowchart Pembatasan Kuota Checkin

Pengujian dilakukan dengan tabel berikut sesuai dengan use case dan scenario pengujian.

TABEL I
PENGUJIAN SISTEM

Pada gambar di atas diasumsikan kuota checkin di waktu yang sama yaitu 10. Jika proses replikasi berjalan maka sisa kuota = total_kuota - total_penjualan. Tetapi jika data tidak tereplikasi atau dengan kata lain kedua server tidak terkoneksi maka pembatasan kuota checkin berlaku seperti flowchart di atas. Pengujian sistem dilakukan dengan menggunakan black box testing sesuai dengan skenario pengujian berikut.



Gambar 16. Skenario Pengujian

No	Skenario Pengujian	Deskripsi	Input	Skenario keluaran
1	Username benar, Password benar	Skenario dasar	Username "admin" Password "123"	Validasi berhasil, Aplikasi menampilkan menu
2	Username salah, Password benar		Username "Vino" Password "123"	Validasi tidak berhasil, muncul pesan kesalahan
3	Username benar, Password salah		Username "admin" Password "qwerty"	Validasi tidak berhasil, muncul pesan kesalahan
4	Username salah, Password salah		Username "Vino" Password "qwerty"	Validasi tidak berhasil, muncul pesan kesalahan
5	Insert data pada salah satu table (tabel user) server BTC	Skenario dasar	Username : admin Nama : administrator Hak : Admin	Aplikasi menampilkan data yang telah diinsert pada database server BTC
6	Cek data pada server SPS			Aplikasi menampilkan data yang telah diinsert dari server BTC
7	Insert data pada salah satu table (tabel user) server SPS		Username : admin Nama : administrator Hak : Admin	Aplikasi menampilkan data yang telah diinsert pada database server SPS

8	Cek data pada server BTC			Aplikasi menampilkan data yang telah diinsert dari server SPS
---	--------------------------	--	--	---

V. KESIMPULAN

Agar dapat menggunakan fitur slony, PT. XYZ harus mempunyai server lokal pada kedua kantornya. Artinya ketika pengguna hendak membuka aplikasi web perusahaan, browser hanya perlu mengakses aplikasi web pada server lokal (jaringan lokal). Selanjutnya ketika ada penambahan data baru ataupun perubahan data di server cabang, maka slony akan mengirimkan atau mereplika data tersebut dari server cabang ke server pusat. Dalam hal ini yang dikirimkan hanya perubahan data pada database saja sehingga tidak membutuhkan koneksi internet yang tinggi mengingat ukuran file database yang relatif kecil. Begitu juga ketika perusahaan cabang tidak terkoneksi dengan internet, aplikasi masih bisa digunakan karena aplikasi diakses secara lokal pada server di masing-masing kantor.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Rendi, dkk mahasiswa Jurusan Teknik Informatika, Politeknik Negeri Batam dalam pengumpulan data dan pengujian sistem.

DAFTAR PUSTAKA

- [1]. H. Maulana, Analisis dan Perancangan Sistem Replikasi Database MySQL dengan menggunakan VMWare pada Sistem Operasi Opensource, Jurnal Nasional Informatika dan Teknologi Jaringan (InfoTekjar) Vol.1, No.1, September 2016.
- [2]. A.S. Ajinusa, A.B.M. Wijaya, A. Aziz, "Pembangunan Sistem Informasi Presensi Perkuliahan Menggunakan Basisdata Terdistribusi dengan Metode Replikasi-Asynchronous," Jurnal ITSMART Vol.4 No.2 Desember 2015
- [3]. E. Purnomo, A. Ashari, "Implementasi Replikasi Data Asynchronous untuk Keamanan Data Sistem Informasi Akademik Online", Berkala MIPA 24(2), Mei 2014
- [4]. F. Zaini, L. Atika, U. Ependi, "Implementasi Metode Asynchronous pada Replikasi Database Invenaris Universitas Bina Darma", Jurnal Informatika Universitas Bina Darma Palembang, Oktober 2015
- [5]. Nugl, 2014, Instalasi dan Konfigurasi Mysql Master to Master, <https://www.excellent.co.id/excellent/blog/instalasi-dan-konfigurasi-replikasi-mysql-master-to-master/>
- [6]. A.P. Basuki, 2010, Membangun Web Berbasis PHP dengan Framework Codeigniter, Lokomedia, Yogyakarta.
- [7]. K.R. Gufron, 2013, Mengenal Aplikasi Virtualisasi Oracle VM Virtual Box, <http://dosen.gufron.com/artikel/mengenal-aplikasi-virtualisasi-oracle-vm-virtualbo/10/>
- [8]. Nugroho, 2004, Database Relasional dengan MySQL, Andi, Yogyakarta.
- [9]. PostgreSQL, 2015, What is PostgreSQL?, <http://www.postgresql.org/docs/9.5/static/intro-what.html>
- [10]. Slony Development Group, 2010, Sloni-I Introduction, <http://slony.info/>
- [11]. Kurniawan, D. E., & Fajrianto, R. (2017, August). Sistem Pemantau Koneksi Internet Menggunakan Sistem Informasi Geografis dan Sms Gateway. In SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi (Vol. 5, No. 1).