

# Implementasi *Heterogenous Distributed Database System* Oracle Xe 10g dan MySQL Rekam Medis Poliklinik UIN Sunan Kalijaga

## *Heterogeneous Distributed Database System Polyclinic Medical Record UIN Sunan Kalijaga*

Valdi Adrian Abrar<sup>1</sup>, M. Didik R. Wahyudi<sup>2</sup>

<sup>1,2</sup>Teknik Informatika FST UIN Sunan Kalijaga

E-mail: \*<sup>1</sup>[valdiadrian@gmail.com](mailto:valdiadrian@gmail.com), <sup>2</sup>[m.didik@uin-suka.ac.id](mailto:m.didik@uin-suka.ac.id)

### **Abstrak**

*Infrastruktur yang biasa digunakan oleh sistem informasi yang ada di Indonesia kebanyakan mempunyai model yang terpusat. Sehingga, jika terjadi masalah pada server seperti server mati atau terjadi kerusakan pada basis data, maka sistem informasi tidak dapat digunakan sampai masalah pada server tersebut teratasi. Untuk mengatasi hal tersebut, sistem replikasi atau duplikasi data pada sistem basis data terdistribusi diharapkan dapat meminimalisir kehilangan data rekam medis sehingga walaupun ada server yang mengalami masalah, maka data tidak akan hilang. Dalam konteks sistem rekam medis di poliklinik UIN Sunan Kalijaga, hal ini bisa menjadi solusi untuk memenuhi aspek ketersediaan data. Sinkronisasi data antara server utama dan replika dapat dilakukan secara otomatis maupun secara manual. Sinkronisasi otomatis dilakukan dengan cara menjalankan baris program secara otomatis dan berkala dengan aturan tertentu. Sinkronisasi manual dijalankan oleh operator dengan menjalankan suatu perintah. Berdasarkan hasil dan pembahasan, diperoleh kesimpulan bahwa implementasi *Heterogenous Distributed Database System* pada sistem informasi poliklinik dapat mengatasi masalah jika terjadi pada beberapa server dengan cara mengolah dan mendistribusikan data pada server lain yang aktif. Proses replikasi dan sinkronisasi data rekam medis yang dilakukan, ternyata dapat meminimalisir kehilangan data.*

**Kata Kunci** — basis data, sistem basis data terdistribusi, database terdistribusi heterogen, replikasi, sinkronisasi.

### **Abstract**

*Infrastructure used by the existing information systems in Indonesia, mostly have a centralized model. Thus, if a problem occurs on the server as the server is dead or there is damage to the database, the system information can not be used until the problem is solved on the server. To overcome this, the system replication or duplication of data in a distributed database system is expected to minimize the loss of medical records so that even if there is a server that has the problem, then the data will not be lost. In the context of medical records system at the clinic UIN Sunan Kalijaga, this could be a solution to meet aspects of data availability. The data synchronization between the primary and replica servers can be done automatically or manually. Automatic synchronization is done by running the program line automatically and periodically with certain rules. Manual synchronization is run by an operator to execute a command. Based on the results and discussion, we concluded that the implementation of *heterogenous Distributed Database System* on clinic information systems can solve the problem if it occurs on multiple servers by processing and mendistribusikan data on another server that is active. The process of replication and synchronization of medical records that do, it can minimize data loss.*

**Keywords** — database, distributed database system, heterogenous distributed database, replication, synchronization.

---

## 1. PENDAHULUAN

Untuk mengolah suatu data bukan merupakan tugas yang mudah dikarenakan jika data tersebut merupakan data yang sangat penting dan apabila data tersebut hilang maka akan terjadi kerugian pada pihak-pihak tertentu. Data yang diolah oleh sistem informasi nantinya berguna untuk mendapatkan suatu informasi dari data-data yang sudah diolah.

Sistem informasi yang ada saat ini pada pelayanan kesehatan sering dilihat jika sistem informasi tersebut mendapatkan data hanya pada satu server. Jadi, ketika server yang menyimpan data-data rekam medis mengalami masalah maka seluruh sistem tidak akan berjalan dan mengakibatkan kehilangan data rekam medis yang penting itu.

Metode replikasi data berguna untuk mengatasi masalah jika sewaktu-waktu terjadi kerusakan pada basis data sehingga data rekam medis yang merupakan data yang sangat penting dapat diminimalisir kehilangannya. Lalu, proses sinkronisasi digunakan untuk menyamakan data antar server.

Setelah menganalisa penelitian-penelitian sebelumnya, penulis mengusulkan pengembangan sistem informasi rekam medis yang dimana nantinya data yang telah dimasukkan ke dalam basis data langsung secara realtime menduplikasi data tersebut ke beberapa server yang ada sehingga jika sewaktu-waktu server yang memiliki data tersebut mengalami suatu masalah, maka akan ada server lain yang bisa memberikan data yang dibutuhkan pada saat sistem sedang berjalan. Selain itu, proses sinkronisasi data secara manual membantu petugas untuk menyamakan data pada semua server yang sebelumnya telah mengalami masalah.

Penggunaan metode basis data terdistribusi untuk replikasi dan sinkronisasi itulah yang nantinya diharapkan bisa membantu dalam proses data rekam medis sehingga meminimalisir kehilangan data dan berhentinya penggunaan sistem dikarenakan salah satu server mengalami masalah.

## 2. METODE PENELITIAN

### 2.1. Database Management System (DBMS)

DBMS adalah koleksi terpadu dari database dan program-program komputer (utilitas) yang digunakan untuk mengakses dan memelihara database. Program-program tersebut menyediakan berbagai fasilitas operasi untuk memasukkan, melacak, dan memodifikasi data ke dalam database, mendefinisikan data baru, serta mengolah data menjadi informasi yang dibutuhkan (DBMS = Database + Program Utilitas) [1].

### 2.2. Basis Data Terdistribusi

Basis data terdistribusi adalah kumpulan data logic yang saling berhubungan, secara fisik terdistribusi dalam jaringan komputer yang tidak tergantung dari program aplikasi dan dapat digunakan oleh banyak aplikasi sekarang maupun yang akan datang [2].

Terdapat dua macam metode dari basis data terdistribusi yaitu Heterogenous dan Homogenous. Antara lain [3].

1. Homogenous, DBMS yang menangani data untuk semua server sejenis [4] atau DBMS yang sama digunakan di setiap simpul.
    - a. Otonomi, setiap DBMS bekerja secara mandiri. Saling mengirimkan pesan untuk berbagi pembaruan data.
    - b. Tidak otonomi, DBMS pusat, mengoordinasi akses basis data dan melakukan pembaruan melintasi simpul-simpul.
  2. Heterogenous, DBMS yang menangani data distribusi berbeda-beda mungkin digunakan di setiap simpul.
-

### 2.3. Replikasi

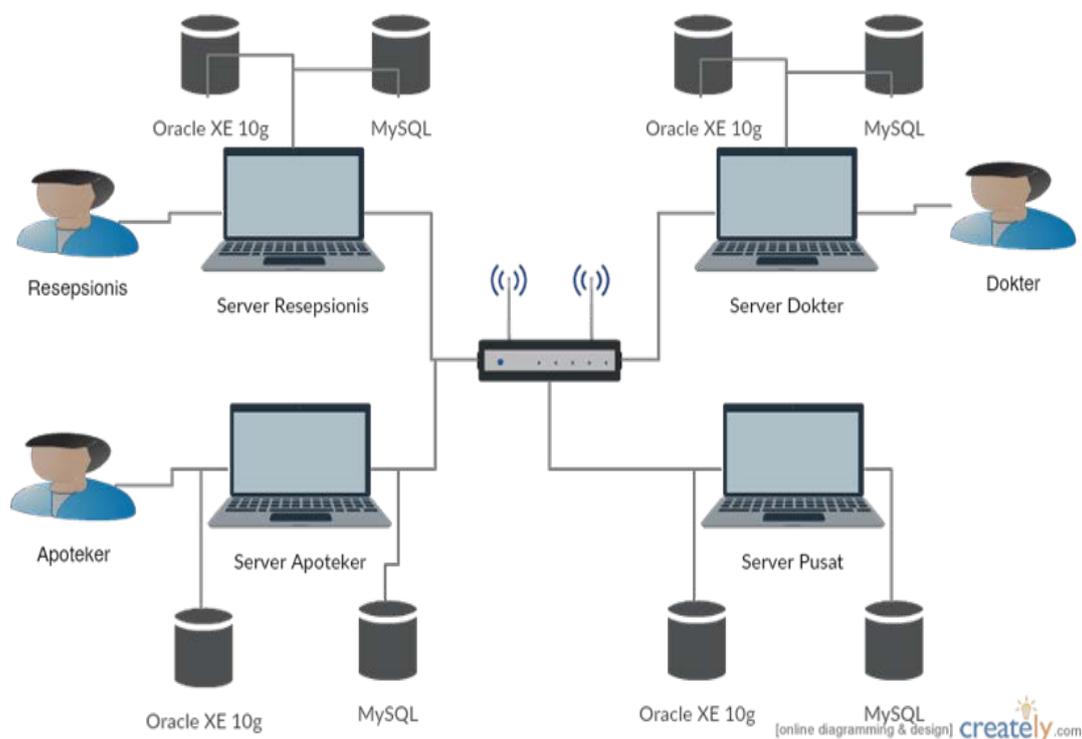
Replikasi adalah suatu teknik untuk melakukan copy dan pendistribusian data dan objek-objek database dari satu database ke database lain dan melaksanakan sinkronisasi antara database sehingga konsistensi data dapat terjamin. Dengan menggunakan teknik replikasi ini, data dapat didistribusikan ke lokasi 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 [5].

### 2.4. Sinkronisasi

Sinkronisasi data adalah suatu proses untuk menjaga konsistensi data yang ada pada suatu server dengan data yang berada pada server yang lain. Dalam proses sinkronisasi database, terdapat proses duplikasi data yang disimpan ke dalam suatu tabel dan skema yang berada pada database yang lain. Dengan adanya proses sinkronisasi memungkinkan suatu data yang berada pada database tertentu akan dapat terupdate secara langsung maupun berkala pada database yang lain. Teknik inilah yang merupakan dasar dari konsep replikasi yang ada pada database. Dalam suatu kasus database terdistribusi, mekanisme sinkronisasi ini diperlukan untuk melakukan update data yang terdapat pada global schema yang diakses oleh suatu user secara langsung. Data yang diupdate pada global schema merupakan data yang berada pada lokal schema dari database yang terdistribusi. Antara lokal schema dengan global schema bisa jadi mempunyai skema yang sama secara struktur datanya (homogeneous), ataupun berbeda (heterogeneous) [6].

### 2.5. Implementasi Heterogenous Distributed Database System

Metode penelitian basis data yang digunakan adalah metode *Heterogenous Distributed Database System* [7]. Gambaran umum tentang arsitektur metode *Heterogenous Distributed Database System* disajikan dalam Gambar 1 berikut ini.



Gambar 1. Gambaran Umum Implementasi Heterogenous Distributed Database System

Implementasi metode *Heterogenous* dalam penelitian ini menggunakan tiga perangkat keras yaitu dua buah laptop dan satu buah *access point* sebagai media penghubung jaringan komputer. Selain itu, penelitian ini juga menggunakan perangkat lunak seperti basis data Oracle XE 10g [8] basis data MySQL [9], dan virtualbox untuk membuat *virtual personal computer* pada setiap laptop yang akan dijadikan sebagai *server*.

Penelitian ini berfokus pada penerapan replikasi data secara *realtime* dan juga sinkronisasi data secara manual yang diilustrasikan pada prototype sistem informasi Poli Klinik. Tujuannya adalah membuktikan apakah replikasi dan sinkronisasi dapat meminimalisir kehilangan data jika sewaktu-waktu ada *server* yang sedang mengalami masalah.

#### 2.5.1. Konfigurasi Jaringan Komputer

Konfigurasi jaringan komputer dalam penelitian ini sangatlah penting karena untuk mengimplementasikan metode basis data terdistribusi yang mana dijadikan beberapa server dan setiap server memiliki tabel yang berbeda. Pada jaringan yang dikonfigurasi, alamat IP yang digunakan pada setiap server adalah IP Kelas C dengan range dari 192.168.1.3 sampai 192.168.1.6. IP Address yang dikonfigurasi ini statik sehingga IP yang digunakan tetap dan tidak akan berubah. IP Address pada laptop pertama menggunakan 192.168.1.3 untuk server pusat dan 192.168.1.4 untuk server resepsionis. Lalu, IP Address pada laptop kedua menggunakan 192.168.1.6 untuk server apoteker dan 192.168.1.5 untuk server dokter.

#### 2.5.2. Konfigurasi ODBC dan Database Link

Konfigurasi ODBC dan Database Link merupakan salah satu proses yang harus dilakukan untuk merancang sebuah sistem data yang terdistribusi. ODBC dalam hal ini berguna untuk menghubungkan dua basis data yaitu Oracle XE 10g dan MySQL sehingga penerapan metode *Heterogenous* dapat dilakukan. Lalu, untuk menghubungkan dua basis data tersebut dibutuhkan database link. Pada penelitian ini, ODBC dan database link digunakan untuk menghubungkan dua basis data tersebut pada server pusat, resepsionis, dokter, dan server apoteker.

#### 2.5.3. Implementasi Dengan Sistem Informasi Poli Klinik

Penerapan sistem informasi Poli Klinik UIN Sunan Kalijaga untuk membuktikan metode basis data terdistribusi, dan sinkronisasi manual untuk menyamakan data antar server. Pembuatan sistem informasi Poli Klinik ini menggunakan bahasa pemrograman web PHP, HTML, dan CSS. Lalu, dalam penelitian ini juga memanfaatkan ODBC dan database link untuk mewujudkan penelitian dengan metode *heterogenous* basis data terdistribusi.

---

### 3. HASIL DAN PEMBAHASAN

Dalam sistem informasi poliklinik UIN Sunan Kalijaga yang dipergunakan dalam penelitian ini, tabel database yang dipergunakan didistribusikan kedalam beberapa server yang berbeda yaitu Oracle XE 10g dan MySQL. Pembagian tabel dapat dilihat pada Tabel 1 berikut.

Tabel 1. Pembagian Tabel Ke Server

Server Pusat	Server Resepsionis	Server Dokter	Server Apoteker
Tabel_pasien	Tabel_pasien	Tabel_dokter	Tabel_obat
Tabel_pelayanan	Tabel_pelayanan	Tabel_users	Tabel_apoteker
Tabel_dokter	Tabel_perawat	Tabel_rekam_medis	Tabel_obat_keluar
Tabel_perawat	Tabel_rekam_medis		Tabel_beli_obat
Tabel_apoteker	Tabel_users		Tabel_rekam_medis
Tabel_obat			Tabel_users
Tabel_rekam_medis			
Tabel_beli_obat			
Tabel_obat_keluar			

Tabel basis data dirancang pada empat buah server yaitu server pusat, resepsionis, dokter, dan server apoteker. Komputer pertama berperan sebagai server pusat dan virtual personal computer pada komputer pertama digunakan sebagai server resepsionis. Komputer kedua berperan sebagai server apoteker dan virtual personal computer pada komputer kedua digunakan sebagai server dokter. Pada komputer pertama terdapat dua server yaitu server pusat dan server resepsionis. Pada server pusat DBMS Oracle XE 10g mempunyai tabel seperti tabel pasien, pelayanan, perawat, apoteker, dan tabel obat. Sedangkan di server resepsionis DBMS Oracle XE 10g memiliki beberapa tabel yang sama seperti di server pusat yaitu tabel pasien, pelayanan, perawat, serta tabel users. Lalu pada DBMS MySQL pada server pusat dan server resepsionis memiliki tabel yang sama yaitu tabel rekam\_medis. Alasan tabel rekam\_medis diletakkan pada DBMS di masing-masing server pusat dan server resepsionis adalah dikarenakan data rekam\_medis yang tumbuh sangat cepat (jumlahnya banyak) sehingga diletakkan di DBMS MySQL yang mempunya space unlimited (Oracle XE 10g hanya memiliki kapastias 4GB).

Pada komputer kedua terdapat dua server yaitu server dokter dan server apoteker. Pada server dokter memiliki beberapa tabel yang sudah dimiliki oleh server pusat. Tabel tersebut adalah users, dokter pada DBMS Oracle XE 10g dan Tabel rekam\_medis di DBMS MySQL. Untuk server apoteker mempunyai tabel apoteker, users, dan tabel obat di DBMS Oracle XE 10g dan juga tabel obat\_keluar, beli\_obat, dan tabel rekam\_medis di DBMS MySQL. Alasan tabel obat\_keluar dan tabel beli\_obat juga diletakkan pada DBMS MySQL adalah memiliki pertumbuhan data yang besar sama seperti data pada tabel rekam\_medis.

Tabel yang nantinya memiliki pertumbuhan data yaitu pada tabel pasien di DBMS Oracle XE 10g server pusat dan server resepsionis. Lalu, tabel di DBMS MySQL yang memiliki pertumbuhan data yaitu tabel rekam\_medis, obat\_keluar, dan beli\_obat. Setiap penambahan data, maka setiap server memiliki replikasi data yang sama sesuai dengan query yang telah ditentukan.

### 3.1. Proses Replikasi

Proses replikasi pada penelitian ini digunakan pada semua proses transaksi di sistem informasi Poli Klinik. Setiap proses transaksi, sistem akan melakukan duplikasi data secara *realtime* ke server-server yang sedang aktif. Sehingga, semua server yang aktif memiliki data yang sama. Contoh salah satu transaksi yaitu proses pembuatan data pasien baru apabila sebelumnya belum pernah melakukan perawatan atau berkunjung ke Poli Klinik UIN Sunan Kalijaga. Tampilan proses pembuatan data pasien baru dapat dilihat pada Gambar 2.

The image shows a web form titled "PENDAFTARAN - PASIEN BARU". It contains several input fields with labels and placeholder text:

- ID Pasien:** A text box containing the value "900015".
- Nama Pasien:** An empty text box with the placeholder "Nama lengkap pasien. Harus diisi."
- Nama Orang Tua/Wali:** An empty text box with the placeholder "Nama orang tua/wali pasien. Jika tidak ada isikan 'Tidak Ada'."
- Jenis Kelamin:** A dropdown menu showing "-- Pilih Kelamin --" with the placeholder "Jenis kelamin pasien. Harus diisi."
- Tempat Lahir:** An empty text box with the placeholder "Tempat lahir pasien. Harus diisi."
- Tanggal Lahir:** A text box with the placeholder "dd----yyyy" and the label "Tanggal lahir pasien. Harus diisi."

Gambar 2. Tampilan Halaman Pendaftaran Pasien Baru

Dari tampilan di atas, data pasien akan diduplikasi ke server pusat dan server resepsionis di DBMS Oracle XE 10g masing-masing server. Berikut adalah query replikasi data untuk penyimpanan data pasien baru.

```
“$query = oci_parse ($conn_lokal, "INSERT INTO pasien (id_pasien, nama_pasien,
nama_ortu, jenis_kelamin, tgl_lahir, tempat_lahir, umur, alamat_asal, alamat_domisili,
pekerjaan, telp, gol_darah) VALUES (:id_pasien, nama_pasien, nama_ortu, :jenis_kelamin,
to_date(:tgl_lahir, 'YYYY-MM-DD'), :tempat_lahir, :umur, :alamat_asal, :alamat_domisili,
:pekerjaan, :telp, :gol_darah)");
```

```
$query_pusat = oci_parse($conn_pusat, "INSERT INTO pasien (id_pasien,
nama_pasien, nama_ortu, jenis_kelamin, tgl_lahir, tempat_lahir, umur, alamat_asal,
alamat_domisili, pekerjaan, telp, gol_darah) VALUES (:id_pasien, :nama_pasien, :nama_ortu,
:jenis_kelamin, to_date(:tgl_lahir, 'YYYY-MM-DD'), :tempat_lahir, :umur, :alamat_asal,
:alamat_domisili, :pekerjaan, :telp, :gol_darah)");”
```

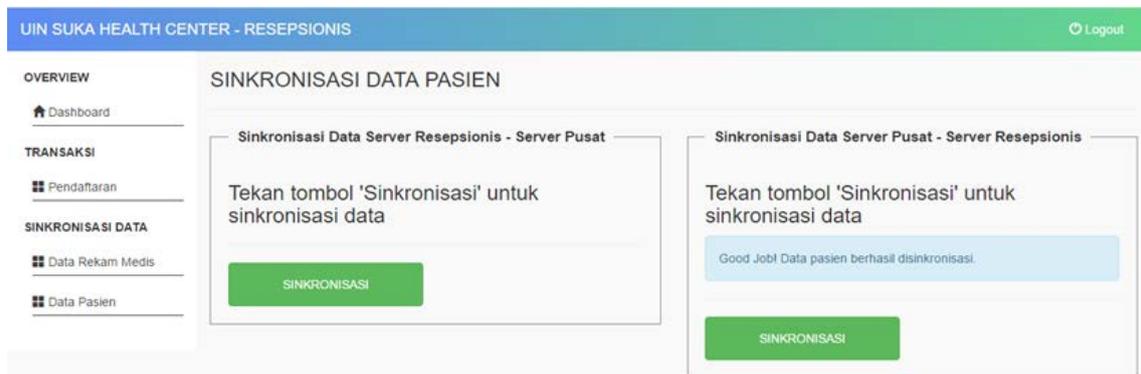
Dari perintah PHP di atas, dilakukan input data ke server resepsionis menggunakan *\$query\_lokal* dan input data ke server pusat dengan menggunakan *\$query\_pusat*. Gambar 3 adalah pembuktian jika data berhasil di replikasi.



Gambar 3. Bukti Sinkronisasi Berhasil

### 3.2. Proses Sinkronisasi

Proses sinkronisasi pada penelitian ini dilakukan secara manual. Proses sinkronisasi manual ini berguna untuk jika sewaktu-waktu ada server yang mengalami masalah. Secara otomatis jika ada transaksi maka server yang mengalami masalah tersebut tidak akan mendapatkan data transaksi yang baru. Sinkronisasi manual ini ada pada semua tabel yang digunakan untuk proses transaksi di semua sistem informasi Poli Klinik. Contoh salah satu proses sinkronisasi manual pada data pasien. Gambar 4 adalah tampilan dari fitur sinkronisasi manual data pasien.



Gambar 4. Tampilan Fitur Sinkronisasi Data Pasien

Pada gambar di atas, apabila ingin melakukan sinkronisasi manual ada pilih dari server mana data akan diambil lalu dilakukan sinkronisasi ke server yang memiliki data tidak lengkap. Contoh tersebut adalah proses sinkronisasi dengan sumber data pada server resepsionis lalu disinkronisasi ke server pusat. Gambar 5 adalah perbandingan data sebelum sinkronisasi.

server resepsionis	server pusat
<pre>SQL&gt; select id_pasien from pasien;</pre>	<pre>SQL&gt; select id_pasien from pasien@to_pusat;</pre>
<pre>ID_PASIEN ----- 900001 900002 900003 900004 900005 900006 900007 900008 900009 900010 900011  ID_PASIEN ----- 900012 900013 900014  14 rows selected.</pre>	<pre>ID_PASIEN ----- 900001 900002 900003 900004 900005 900006 900007 900008 900009 900010 900011  ID_PASIEN ----- 900012 900013 900014 900015  15 rows selected.</pre>

Gambar 5. Data Pasien Sebelum Sinkronisasi

Pada Gambar 5 dapat dilihat perbedaan jumlah data pasien di server resepsionis dan server pusat. Lalu pada Gambar 6 adalah bukti jika proses sinkronisasi berhasil dilakukan sehingga data pasien pada server resepsionis dan server pusat sama jumlahnya.

server resepsionis	server pusat
<pre>SQL&gt; select id_pasien from pasien;</pre>	<pre>SQL&gt; select id_pasien from pasien@to_pusat;</pre>
<pre>ID_PASIEN ----- 900001 900002 900003 900004 900005 900006 900007 900008 900009 900010 900011  ID_PASIEN ----- 900012 900013 900014 900015  15 rows selected.</pre>	<pre>ID_PASIEN ----- 900001 900002 900003 900004 900005 900006 900007 900008 900009 900010 900011  ID_PASIEN ----- 900012 900013 900014 900015  15 rows selected.</pre>

Gambar 6. Data Pasien Sesudah Sinkronisasi

#### 4. KESIMPULAN

Berdasarkan kegiatan yang telah dilaksanakan dalam penelitian ini, maka dapat diambil beberapa kesimpulan, diantaranya:

1. Sistem informasi masih dapat berjalan dengan baik walaupun salah satu atau lebih server mengalami masalah seperti down server atau kerusakan pada basis data. Data yang sedang diproses juga tidak akan hilang karena sudah dilakukan replikasi data secara horizontal ke DBMS server yang sedang aktif.
2. Sinkronisasi manual berguna untuk melakukan update data sehingga data pada server yang sedang melakukan sinkronisasi memiliki data yang sama dengan server lainnya.

#### 5. SARAN

Dalam penelitian ini tidak terlepas dari beberapa kekurangan. Oleh karena itu, penulis menyarankan beberapa hal, antara lain:

1. Melakukan implementasi sistem informasi menggunakan lebih dari dua DBMS yang berbeda. Karena dalam penelitian ini hanya menggunakan dua DBMS yang berbeda saja.
2. Membuat sistem informasi yang layak sehingga seperti keamanan dan fitur yang harus ada dalam sistem informasi terdapat dalam sistem. Sehingga dapat langsung diimplementasikan langsung dilapangan atau bisa disebut membuat sistem informasi yang sangat kompleks dan sempurna.
3. Membuat fitur edit dan delete untuk setiap data yang ingin dirubah.
4. Meminimalisir penggunaan user untuk melakukan sinkronisasi data.

#### DAFTAR PUSTAKA

- [1] B, Al-Bahra bin Ladjamuddin., 2004. *Konsep Sistem Basis Data dan Implementasinya*. In: G. Ilmu, ed. *Konsep Sistem Basis Data dan Implementasinya*. Yogyakarta: Graha Ilmu, p. 3.
- [2] Ceri, S. & Pelagatti, G., 1984. *Distributed Database: Principles and Systems*. New York: McGraw-Hill.
- [3] Nugroho, S., 2013. *IMPLEMENTASI KONSEP HOMOGENOUS DISTRIBUTED DATABASE SYSTEM*. Yogyakarta: UIN Sunan Kalijaga.
- [4] Basofi, A., 2012. *Sistem Basis Data Terdistribusi*. <http://ariv.lecturer.pens.ac.id/Database%20/T11.%20Basis%20Data%20Terdistribusi.pdf>. Diakses: 22 Mei 2016.
- [5] Gunadarma, 2007. *e-Learning Gunadarma*. [http://elearning.gunadarma.ac.id/docmodul/AS400/AS400\\_B1/04Replikasirev.pdf](http://elearning.gunadarma.ac.id/docmodul/AS400/AS400_B1/04Replikasirev.pdf). Diakses: 08 Juni 2016
- [6] Sani, Y., 2011. *SINKRONISASI DATA PADA HOMOGENEOUS SCHEME*. [http://www.academia.edu/12592052/SINKRONISASI\\_DATA\\_PADA\\_HOMOGENEOUS\\_SCHEME](http://www.academia.edu/12592052/SINKRONISASI_DATA_PADA_HOMOGENEOUS_SCHEME). Diakses: 08 Juni 2016.
- [7] Qalban, A. A., 2013. *Implementasi Heterogeneous Distributed Database System Untuk Meningkatkan Kapasitas Oracle XE 10g (Ekspress Edition) Studi Kasus Sistem Informasi Akademik*. Yogyakarta: UIN Sunan Kalijaga.
- [8] Oracle. Oracle. 2006. <http://www.oracle.com/technetwork/database/express-edition/overview/dbxe-datasheet-130365.pdf>. Diakses: 17 Mei 2016.
- [9] MySQL. dev.mysql.com. 2016. <http://dev.mysql.com/doc/refman/5.6/en/table-size-limit.html>. Diakses: 17 Mei 2016.