

# REPLIKASI UNIDIRECTIONAL PADA HETEROGEN DATABASE

**Hendro Nindito; Evaristus Didik Madyatmadja; Albert Verasius Dian Sano**

Information Systems Department, School of Information Systems, Binus University  
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480  
h\_nindito@yahoo.com; emadyatmadja@binus.edu; albert\_vds@yahoo.com

## ABSTRACT

*The use of diverse database technology in enterprise today can not be avoided. Thus, technology is needed to generate information in real time. The purpose of this research is to discuss a database replication technology that can be applied in heterogeneous database environments. In this study we use Windows-based MS SQL Server database to Linux-based Oracle database as the goal. The research method used is prototyping where development can be done quickly and testing of working models of the interaction process is done through repeated. From this research it is obtained that the database replication technology using Oracle Golden Gate can be applied in heterogeneous environments in real time as well.*

**Keywords:** replication, heterogen, real time, SQL Server, Oracle

## ABSTRAK

*Penggunaan perangkat lunak basis data yang beraneka ragam dalam suatu perusahaan dewasa ini tidak dapat dihindarkan. Sehingga dibutuhkan teknologi untuk menghasilkan informasi secara real time. Tujuan penelitian ini adalah membahas suatu teknologi replikasi basis data yang dapat diterapkan pada lingkungan database heterogen. Lingkungan heterogen yang digunakan sebagai sumber pada penelitian ini adalah database MS SQL Server berbasis Windows ke database Oracle berbasis sistem operasi Linux sebagai tujuannya. Metode penelitian yang digunakan adalah metoda prototyping di mana pengembangan dapat dilakukan secara cepat dan pengujian terhadap model kerja dilakukan melalui proses interaksi yang berulang-ulang. Hasil dari penelitian ini adalah pengujian terhadap teknologi replikasi basisdata menggunakan oracle golden gate dapat di terapkan pada lingkungan heterogen secara real time dengan baik.*

**Kata kunci:** replikasi, heterogen, real time, SQL Server, Oracle

## PENDAHULUAN

Dalam meraih sebuah kesuksesan di lingkungan yang sangat kompetitif saat ini, diperlukan adanya sebuah informasi yang *real-time*. Hal tersebut membutuhkan sebuah *platform* yang dapat menggabungkan semua informasi dari berbagai sistem tanpa adanya kendala ketersediaan data dan performansi sistem.

Menurut *Oracle white paper* (2009) tentang *Real Time Access to Realtime Information* terdapat empat aspek yang harus dilakukan pelaku bisnis untuk menjawab tantangan tersebut: (1) *availability*, yaitu ketersediaan data harus dapat diakses 24/7/365 yang memiliki arti 24 jam sehari 7 hari dalam seminggu 365 hari dalam setahun tanpa adanya gangguan layanan dan penurunan kinerja aplikasi; (2) *reduced latency*, yaitu data yang tersedia harus selalu baru, dan relevan dengan kebutuhan saat ini; (3) *heterogeneity and IT flexibility*, yaitu solusi replikasi dan integrasi yang fleksibel agar dapat dengan mudah untuk melakukan modifikasi dan distribusi di berbagai lingkungan sistem IT baik lingkungan sistem operasi maupun *platform* basis data; (4) *transaction integrity*, yaitu kelengkapan data dan akurasi harus dapat dijamin pada saat didistribusikan ke sistem lain.

Menurut Jhon Garmany *et all* (121) replikasi basis data adalah proses penggandaan sebagian atau keseluruhan objek basis data ke suatu atau beberapa basis data lainnya. Basis data tersebut dapat terletak di suatu tempat lain atau ke depan akan dijadikan *server* basis data utama. Karena kecepatan dan keandalan *internet*, banyak pengguna merasa replikasi ke depan tidak dibutuhkan. Jika hal tersebut benar, kegiatan transaksi data akan terjadi dari mana saja dan kapan saja ke satu penyedia data. Ketika basis data tersebut tidak dapat diakses karena kerusakan, semua kegiatan transaksi data tidak akan dapat dilakukan.

Kebutuhan akan replikasi terhadap seluruh atau sebahagian dari basis data menjadi meningkat dan umumnya dibutuhkan oleh perusahaan untuk dapat melakukan konsolidasi terhadap informasi yang mereka miliki. Beberapa perusahaan menggunakan beberapa metode replikasi untuk mengirimkan data ke *data warehouse* perusahaan. Replikasi juga dapat digunakan untuk menghasilkan bagian tertentu dari basis data yang dapat digunakan untuk pelaporan, sehingga dapat menghapus dampak dari agregasi data pada basis data utama.

Replikasi basis data pada setiap DMBS tidaklah sama, setiap DBMS memiliki metode replikasi yang berbeda. Berikut ini beberapa metode replikasi menurut Garmany *et all* (122): (1) replikasi manual, yaitu replikasi yang dilakukan secara manual dengan memanfaatkan fasilitas ekspor dan impor pada DBMS, atau menggunakan *distributed query* pada basis target; (2) basis data siaga (*standby database*), yaitu replikasi dengan cara membuat membuat sebuah proses yang akan mengirimkan semua perubahan pada basis data utama dan mengirimkan ke sebuah basis data siaga; (3) replikasi menggunakan *trigger*, yaitu replikasi yang dilakukan dengan cara membuat *trigger* pada tiap tabel yang ada. *Trigger* merupakan sebuah kejadian yang dapat digunakan untuk menangkap perubahan-perubahan pada objek-objek basis data dan segera mengirimkan perubahan pada baris data tersebut ke basis data target; (4) replikasi menggunakan *views*, secara umum *views* digunakan pada tabel-tabel untuk mempermudah perintah-perintah dalam melakukan kueri data. *Views* layaknya seperti tabel virtual. Pada DBMS Oracle disebut juga *materialized views*. Setiap perubahan yang terjadi terhadap tabel-tabel dijadikan *view*, secara otomatis *views* akan mengikuti semua perubahan data. Hasil dari *views* dapat digunakan untuk melakukan replikasi jika dibuat pada basis data target dan dibuat menjadi sebuah material data; (5) *one-way replication*, replikasi yang dilakukan dengan cara menggunakan pemicu atau kemampuan replikasi *built-in* pada DBMS yang mengirimkan semua atau sebahagian perubahan objek-objek pada basis data sumber ke basis data target. Pada metode ini basis data target bersifat *read only*; (6) *writeable replication*, yaitu metode replikasi seperti *one-way replication*, bedanya, perubahan pada objek-objek basis data pada lokal target tetap dapat dilakukan.

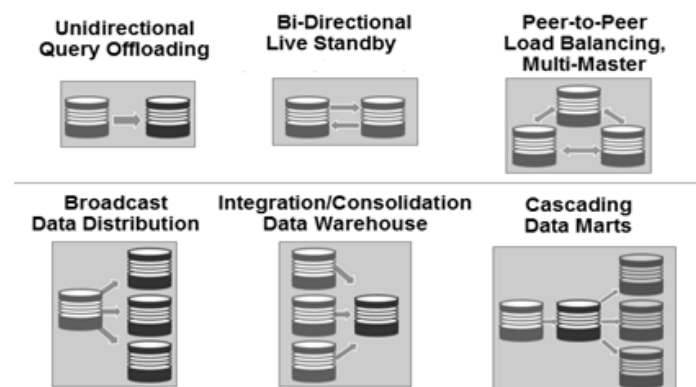
Perubahan yang terjadi pada basis data target tidak direplikasi ke basis data sumber; (7) *updateable replication*, yaitu metode replikasi di mana basis data sumber dan basis data target dapat diperbaharui secara bersamaan, pada metode ini merancang, membuat, dan memelihara jauh lebih sulit karena mempertimbangkan *collusions* dan konflik.

Berdasarkan penelitian pendahuluan yang telah dilakukan, ditemukan beberapa masalah pada perusahaan skala besar yang memiliki platform heterogen sebagai berikut: (1) kesulitan dalam mengakses dan mengintegrasikan data yang tersimpan pada platform heterogen; (2) perlunya effort yang cukup tinggi untuk menghasilkan informasi dari lingkungan *database* terdistribusi; (3) kemampuan memindahkan data dalam jumlah besar yang berpengaruh terhadap kinerja sumber data baik di sumber maupun tujuan; (4) pengamanan pengiriman data dengan panjang kunci enkripsi yang berbeda-beda.

Untuk mengatasi masalah itu, di usulkan untuk menggunakan teknologi replikasi dari Oracle GoldenGate. Menurut Jeffries (2011), teknologi ini memberikan beberapa keuntungan, di antaranya: (1) data dikirim secara *real time* dengan kecepatan *sub-second*; (2) mendukung lingkungan *database* dan platform yang heterogen, serta memelihara integritas data baik di data sumber ataupun tujuan; (3) arsitektur modular dan fleksibel; (4) mudah untuk *recovery*.

Tujuan penelitian ini adalah: (1) membuat solusi alternatif bagi perusahaan skala enterprise untuk menjaga ketersediaan data yang mereka miliki pada platform *database*, *server* maupun *operating system* yang heterogen; (2) menyediakan solusi replikasi *real time* yang handal, mudah di terapkan.

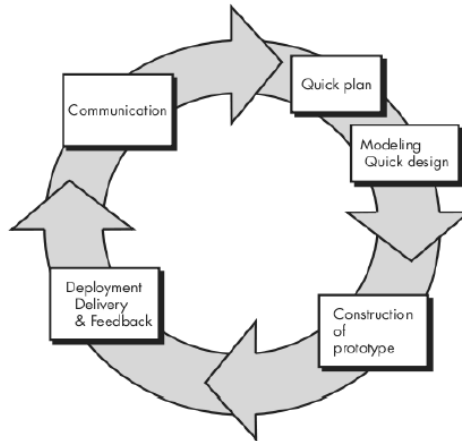
Arsitektur topologi teknologi ini dapat mengelola data dari satu sumber ke satu target lainnya, satu-ke-multi target, multi-ke-satu target, multi-ke-multi, bertingkat dan konfigurasi dua arah. Menurut Oracle® *GoldenGate Windows and UNIX Administrator's Guide* (2011) arsitektur solusi dapat dilihat pada Gambar 1 di bawah ini:



Gambar 1 Topology Support Oracle Golden Gate

## METODE

Metodologi penelitian yang digunakan dalam penelitian ini adalah metode *prototyping*. Metoda ini menjadi pilihan karena dalam proses replikasi ini akan di buat model sedekat mungkin dengan kondisi yang ada dalam dunia industri dan perlu adanya perulangan dari tahap-tahap dasar untuk mencapai kondisi yang optimal. Metoda *prototyping* digambarkan oleh Pressman (2010) seperti pada Gambar 2 berikut ini:



Gambar 2 Paradigma metoda *prototyping*

Terdapat perbedaan dari gambar diatas, bahwa dalam tahap *communication* diganti dengan tahap studi literatur mengenai berbagai kepustakaan yang berkaitan dengan replikasi *database*, *database* Oracle dan *database* MS SQL Server.

Adapun langkah-langkah yang dilakukan untuk mencapai hasil yang diharapkan adalah sebagai berikut: (1) studi literatur terhadap ilmu-ilmu yang terkait dengan penelitian; (2) analisis dan memodelkan konfigurasi sistem dengan teknologi perangkat lunak virtual; (3) pengujian. Skenario pengujian akan dilakukan dengan tiga buah tabel *sample real* yang direplikasikan dari *database* MS SQL Server ke *database* Oracle.

Pada tahapan analisis dari penelitian ini, dilakukan analisis terhadap kendala yang mungkin terjadi dari replikasi dua teknologi *database* terkait penanganan *constraint null*.

## HASIL DAN PEMBAHASAN

Permasalahan yang ada pada PT. XYZ adalah mereplikasi data data dari aplikasi *database* yang ada pada platform berbasis Windows serta aplikasi *database* yang berada pada platform Linux. Topologi yang sesuai adalah topologi *Uni Directional* seperti yang telah dibahas di atas.

Adapun *data definition language* tabel yang dijadikan penelitan adalah sebagai berikut:

```
CREATE TABLE [dbo].[SKGNAC](
  [SKINTX] [decimal](29, 0) NOT NULL,
  [SKCUCO] [nvarchar](15) NULL,
  [SKVNCO] [nvarchar](15) NULL,
  [SKBRCO] [nvarchar](30) NULL,
  [SKCNCO] [nvarchar](1) NULL,
  [SKITID] [nvarchar](9) NOT NULL,
  [SKITCD] [nvarchar](30) NULL,
  [SKLTID] [nvarchar](15) NULL,
  [SKMFPN] [nvarchar](30) NULL,
  [SKGCCD] [nvarchar](5) NOT NULL,
  [SKCOCD] [nvarchar](5) NOT NULL,
  [SKOUCD] [nvarchar](5) NOT NULL,
  [SKWHCD] [nvarchar](5) NOT NULL,
```

```

[SKSACD] [nvarchar](5) NOT NULL,
[SKLCCD] [nvarchar](5) NOT NULL,
[SKSFCD] [nvarchar](5) NOT NULL,
[SKBNCD] [nvarchar](5) NOT NULL,
[SKMSID] [decimal](18, 0) NOT NULL,
[SKSQTY] [decimal](18, 6) NOT NULL,
[SKSUOM] [nvarchar](5) NOT NULL,
[SKAQTY] [decimal](38, 0) NULL,
[SKAUOM] [varchar](5) NULL,
[SKFIYR] [varchar](5) NOT NULL,
[SKFIPT] [varchar](4) NOT NULL,
[SKTTID] [nvarchar](50) NULL,
[SKTRAC] [nvarchar](50) NOT NULL,
[SKATNU] [varchar](10) NULL,
[SKTXID] [decimal](18, 0) NOT NULL,
[SKCRBL] [decimal](18, 6) NOT NULL,
[SKBUOM] [varchar](5) NOT NULL,
[SKTXDT] [decimal](14, 0) NOT NULL,
[SKHEFG] [decimal](1, 0) NOT NULL,
[SKRCAC] [decimal](1, 0) NULL,
[SKMRPZ] [decimal](18, 6) NULL,
[SKMPDT] [decimal](8, 0) NULL,
[SKALST] [decimal](18, 6) NULL,
[SKAVST] [decimal](1, 0) NOT NULL,
[SKCRDT] [decimal](8, 0) NOT NULL,
[SKCRTM] [decimal](6, 0) NOT NULL,
[SKCHDT] [decimal](8, 0) NOT NULL,
[SKCHTM] [decimal](6, 0) NOT NULL,
[SKCHNO] [decimal](3, 0) NOT NULL,
[SKCRBY] [nvarchar](10) NOT NULL,
[SKCHBY] [nvarchar](10) NOT NULL,
[SKTMST] [decimal](14, 0) NOT NULL,
CONSTRAINT [PK_SKGNAC] PRIMARY KEY CLUSTERED
(
    [SKINTX] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
) ON [PRIMARY]

```

Demikian pula untuk struktur tabel SKGNLE dan SODLHD memiliki kemiripan dengan tabel SKGNAC. Perangkat keras, perangkat lunak serta konfigurasi yang digunakan pada penelitian ini adalah sebagai berikut (Tabel 1 dan 2):

Tabel 1 Spesifikasi Server

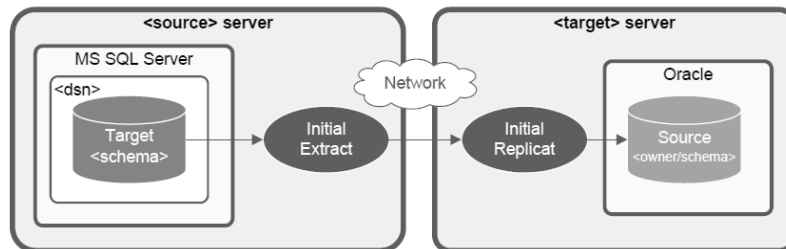
Komponen	Deskripsi
Server	SunFire X4270M2, 1 Proc 6 Core 2 Thread 3Ghz
Sistem Operasi	MS Windows 2008 R2
Memory	20 Gb
Software Replikasi	Oracle Golden Gate V.11.1.1

Tabel 2 Spesifikasi Perangkat Mesin Virtual

Komponen	Vmwhere 1	Vmwhere 2
Sistem Operasi	MS Windows 2003	Linux Redhat 4
Procesor	1 Proc 2 Core	1 Proc 2 Core

Memory	8 Gb	8 Gb
Database	MS SQL Server 2005	Oracle 11g R2
IP	192.168.128.20	192.168.128.30

Perangkat lunak yang digunakan untuk melakukan replikasi adalah Oracle GoldenGate. Arsitektur dari perangkat lunak ini adalah sebagai berikut (Gambar 3):

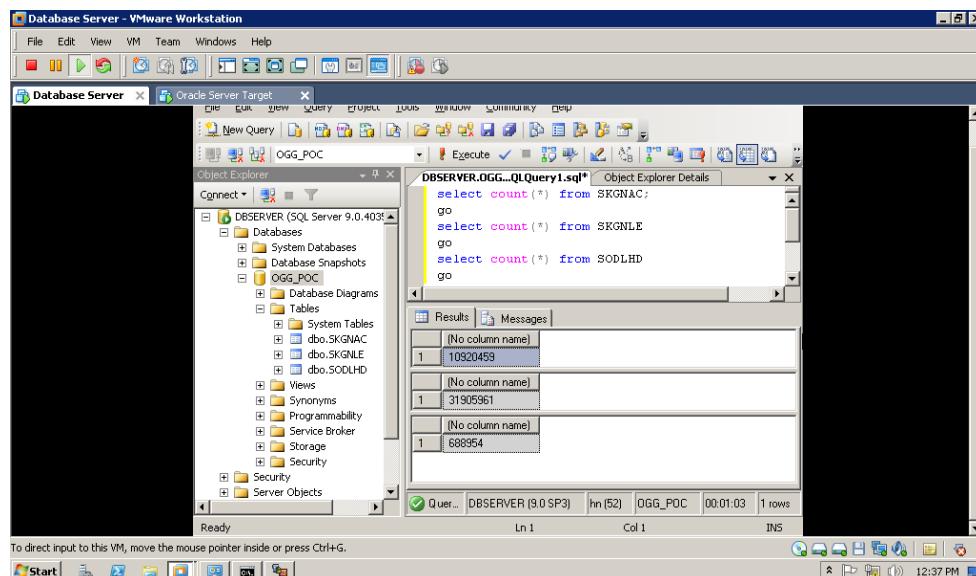


Gambar 3 Arsitektur replikasi dengan Oracle GoldenGate

Hal-hal awal yang dipersiapkan pada SQL Server *sourceserver* adalah:  
Pertama, instal manager sebagai berikut:

```
Shell> ggsci
GGSCI> EDIT PARAMS MGR
PORT 7809
GGSCI> START MANAGER
```

Selanjutnya, membuat *database* SQL Server; membuat ODBC *system data source* (Gambar 4); mempersiapkan tabel yang hendak di replikasi.



Gambar 4 Objek Tabel pada Server Source

Kemudian, membuat konfigurasi *source definitor* untuk membuat koneksi pada target server dengan cara sebagai berikut:

```
GGSCI> EDIT PARAM DEFGEN
DEFSFILE ./dirdef/source.def, PURGE
SOURCEDB dsn_trikonsel, USERID hn, PASSWORD ***
TABLE dbo.SKGNAC;
TABLE dbo.SKGNLE;
TABLE dbo.SODLHD;
```

Dilanjutkan dengan menjalankan source definition generator

```
Shell> defgen paramfile .\dirprm\defgen.prm
```

Transfer file *source definition* yang terbentuk ke target server. Setelah siap, lakukan konfigurasi *Initial load change capture* dengan langkah-langkah sebagai berikut:

Menambahkan initial dataload extract batch task group

```
Shell> ggsci
GGSCI> ADD EXTRACT EINITKS, SOURCEISTABLE
```

Membuat konfigurasi file EINITKS:

```
GGSCI> EDIT PARAMS EINITKS
EXTRACT EINITKS
SOURCEDB dsn_trikonsel, USERID hn, PASSWORD ***
NOCOMPRESSUPDATES
RMTHOST 192.168.128.30, MGRPORT 7809
RMTTASK REPLICAT, GROUP RINITKS
TABLE dbo.SKGNAC;
TABLE dbo.SKGNLE;
TABLE dbo.SODLHD;
```

Membuat konfigurasi pada *inital load replicat* parameter

```
GGSCI> EDIT PARAMS RINITKS
REPLICAT RINITKS
USERID demo, PASSWORD oracle
DISCARDFILE ./dirrpt/RINITKS.txt, PURGE
SOURCEDEFS ./dirdef/source.def
MAP dbo.SKGNAC, TARGET demo.SKGNAC;
MAP dbo.SKGNLE, TARGET demo.SKGNLE;
MAP dbo.SODLHD, TARGET demo.SODLHD;
```

Menjalankan proses replikasi dengan cara sebagai berikut:

```
GGSCI> START EXTRACT EINITKS
```

## Penanganan Constraint NULL pada Proses Replikasi

Menterjemahkan tipe data dari SQL Server ke Oracle tidak menjadi masalah. Namun penanganan *constraint* harus diperhatikan. SQL Server menganggap string kosong ("" ) berbeda dari nilai NULL, sehingga kolom char dapat didefinisikan sebagai NOT NULL, tapi masih termasuk string kosong dalam data. Oracle menganggap string sama dengan nilai NULL jika suatu kolom char didefinisikan. Berikut adalah *error* yang terjadi dikarenakan perbedaan penanganan *constraint* Null:

```

15:43:05 WARNING OGG-01194 Oracle GoldenGate Capture for ODBC, EINITKS.prm:
EXTRACT task RINITKS abended: OCI Error ORA-01400: cannot insert NULL into
("DEMO"."SKGNAC"."SKSFCD") (status = 1400), SQL <INSERT INTO "DEMO"."SKGNAC"
("SKINTX", "SKITID", "SKGCCD", "SKCOCD", "SKOUCD", "SKWHCD", "SKSACD", "SKLCCD", "SKSFCD",
"SKBNCD", "SKMSID", "SKSQTY", "SKSUOM", "SKAQTY", "SKAUOM@C.
15:43:05 ERROR OGG-01203 Oracle GoldenGate Capture for ODBC, EINITKS.prm:
EXTRACT abending.
15:43:05 ERROR OGG-01668 Oracle GoldenGate Capture for ODBC, EINITKS.prm:
PROCESS ABENDING.

```

Sebagai solusi yang diambil pada permasalahan ini adalah dengan cara menambahkan objek *database trigger* pada *database* tujuan yaitu Oracle sebagai berikut:

```

CREATE OR REPLACE TRIGGER col_abc
BEFORE INSERT
ON table xyz
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
DECLARE
BEGIN
    IF INSERTING AND:NEW.col_abc IS NULL THEN
        SELECT '-' INTO:NEW.col_abc FROM DUAL;
    END IF;
    EXCEPTION
        WHEN OTHERS THEN
            NULL;
END;
/

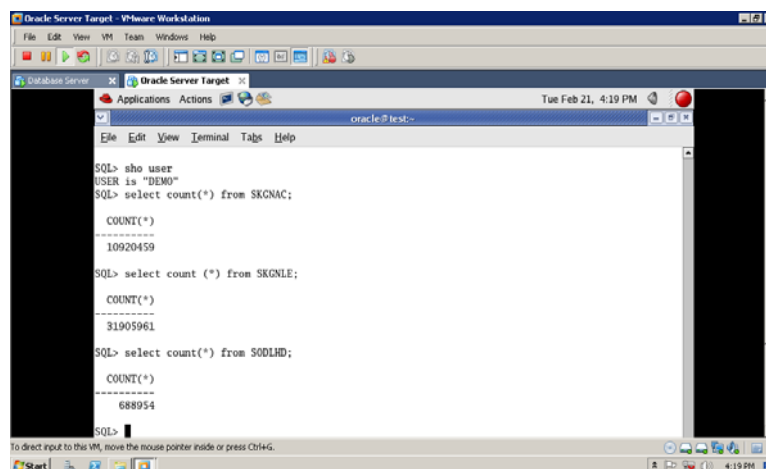
```

## Hasil Replikasi

Hasil replikasi adalah sebagai berikut (Tabel 3, Gambar 5).

Tabel 3 Hasil Replikasi Unidirectional

Parameter	SKGNAC	SKGNLE	SODLHD
Jumlah Record	10.920.459	31.905.961	688.954
Waktu	1 jam 2 mnt	5 jam 45 mnt	30 mnt



Gambar 5 Tabel pada server target.



## PENUTUP

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut: (1) dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa teknologi Oracle GoldenGate dapat diterapkan pada lingkungan sistem operasi dan *platform* DBMS yang heterogen, hal ini sudah dibuktikan dengan melakukan studi kasus dengan melakukan replikasi pada topologi *Uni-Directional* pada *platform* sistem operasi Windows dengan DBMS SQL Server dan Linux dengan DBMS Oracle; (2) dari pengujian yang telah dilakukan, Oracle GoldenGate dapat digunakan sebagai salah satu solusi korporasi untuk menjaga ketersediaan data secara *realtime* dengan *zero-down-time* tanpa mengganggu kegiatan transaksi; (3) penanganan *constraint* NULL pada DBMS yang berbeda dapat dilakukan dengan metoda yang berlainan. Pada penelitian ini, dibuat objek *database* trigger yang di jalankan sebelum melakukan *insert* data pada suatu kolom untuk melakukan pengujian apakah bernilai *null* atau tidak.

## DAFTAR PUSTAKA

- Garmany, J., Walker, J., Clark, T. (2005). *Logical Database Design Principles*. Boca Raton, Florida: Auerbach Publications.
- Jeffries, John P. (2011). *Oracle GoldenGate 11g Implementer's guide*. Birmingham: Packt Publishing.
- Oracle. (2009). *Oracle® GoldenGate 11g: Real Time Access to Real Time Information*. Diakses dari <http://www.oracle.com/us/products/middleware/data-integration/goldengate11g-realttime-wp-168153.pdf>.
- Oracle. (2011). *Oracle® GoldenGate: Windows and UNIX Administrator's Guide 11g Release 1 Patch Set 1 (11.1.1.1)*. Diakses dari [http://docs.oracle.com/cd/E22355\\_01/doc.11111/e22504.pdf](http://docs.oracle.com/cd/E22355_01/doc.11111/e22504.pdf)