Disaster Recovery using Oracle Data Guard

Redo Dwi Bagus Ferdiyanto[1], Rengga Asmara[2], Arif Basofi[2]
Mahasiswa[1], Dosen[2]
Jurusan Teknologi Informasi
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Kampus PENS-ITS Keputih Sukolilo Surabaya 60111
Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114
Email: redo.dbf@gmail.com

Abstrak

Kebutuhan sistem IT kini semakin meningkat terutama dalam hal reliabilitas dan availabilitas untuk menunjang kelangsungan bisnis perusahaan. Penting bagi perusahaan untuk mengimplementasikan konsep high availability (HA) guna melindungi data-data yang merupakan aset terpenting perusahaan. Tujuan proyek akhir ini adalah membangun sistem disaster recovery yang mudah digunakan, murah, komprehensif, efektif, efisien, dan dapat diandalkan serta menjamin ketersediaan data. Hasil dari proyek akhir ini berupa sistem Data Guard yang dapat diimplementasikan pada perusahaan, instansi pendidikan, atau instansi lainnya yang menggunakan Oracle Database, untuk melindungi data perusahaan tersebut dari kerusakan, bencana, failure, error, serta perawatan rutin yang menyebabkan database mengalami downtime. Sistem tersebut diharapkan dapat menjamin ketersediaan data selama 24 jam per hari, 7 hari per minggu. Dalam sistem Data Guard terdapat dua macam Database, yaitu Database utama (primary Database) dan satu atau lebih Database cadangan (standby Database) yang akan digunakan jika primary Database mengalami downtime.

Kata kunci: Oracle database, disaster recovery, Data Guard, primary database, standby database, high availability.

1. Pendahuluan

Di setiap sistem *database*, kemungkinan terjadinya *failure* terhadap sistem dan perangkat keras selalu ada. Sebelum terjadi *failure* yang mempengaruhi sistem *database* tersebut maka harus dipersiapkan sistem *backup* dari *database* tersebut. Tujuannya adalah untuk menjamin proses operasional harian yang penting bisa tetap berjalan, meskipun *primary database* sedang mengalami *failure*.

Berdasarkan survei yang dipublikasikan dalam *Disaster Recovery Journal* (DRJ), penyebab utama kerusakan data adalah *hardware and system errors*, dengan nilai prosentase 49%. Kemudian disusul oleh *human errors* 36%, lalu *computer viruses* 7%, *software corruption* 4%, dan yang terakhir *natural disaster* 3%. Setiap kejadian itu menyebabkan sistem mengalami *downtime* yang tidak direncanakan dan tidak dikehendaki. Ada pula *downtime* yang direncanakan, seperti perubahan data, perubahan sistem, perawatan rutin, serta pengembangan sistem.

Kedua jenis *downtime* tersebut sangat mungkin menyebabkan kerusakan data, dan kejadian tersebut sudah seringkali terjadi. Bagi perusahaan, data merupakan aset yang sangat penting terutama untuk kelangsungan bisnis perusahaan. Dengan rusaknya data, perusahaan dapat mengalami kerugian yang sangat besar. Hal inilah yang mendasari pengembangan proyek *disaster recovery* ini.

Tujuan utama dari proyek ini adalah menyediakan pengamanan yang efektif bagi sistem terhadap kerusakan data. Untuk membantu memaksimalkan availabilitas sistem Oracle *Database* ada banyak cara yang dapat diimplementasikan, salah satunya dapat dikatakan sangat efektif yaitu dengan mengimplementasikan sistem Oracle Data Guard. Sistem tersebut dapat memberikan proteksi data, *recovery* data, serta availabilitas data sehingga data dapat dipastikan ketersediaannya selama 24 jam per hari, 7 hari per minggu.

Dalam sistem Data Guard terdapat dua macam *Database*, yaitu *Database* utama (*primary Database*) dan satu atau lebih *Database* cadangan (*standby Database*) yang akan digunakan jika *primary Database* mengalami *downtime*.

2. Dasar Teori

2.1 *Business Continuity Plan* dan *Disaster Recovery Plan*

Business Continuity Plan (BCP) dan *Disaster Recovery Plan* (DRP) adalah dua hal yang sangat penting dalam proses bisnis, namun jarang menjadi prioritas karena alasan harganya mahal dan sulit penerapannya. Apalagi bencana adalah hal yang umumnya diyakini

karena faktor alam yang tak dapat diprediksi dan tak dapat dicegah ataupun dihindari, sehingga kalangan bisnis berkeyakinan bahwa pelanggan mereka akan memaklumi hal ini.

Namun demikian dengan perkembangan teknologi informasi, maka ditemukan teknologi yang dapat menjamin keberlanjutan bisnis dan pemulihan dari bencana, yang lebih murah dan mudah penerapannya. Bahkan BCP dan DRP telah menjadi standar tersendiri bagi kalangan bisnis terutama yang berhubungan dengan jalannya proses bisnis (aplikasi) dan penyimpanan data. Secara umum tujuan dari BCP dan DRP adalah menjaga bisnis tetap beroperasi meskipun ada gangguan dan menyelamatkan sistem informasi dari dampak bencana lebih lanjut.

BCP adalah proses otomatis atau pun manual yang dirancang untuk mengurangi ancaman terhadap fungsi-fungsi penting organisasi, sehingga menjamin kontinuitas layanan bagi operasi yang penting. DRP adalah prosedur yang dijalankan saat BCP berlangsung *(in action)* berupa langkah-langkah untuk penyelamatan dan pemulihan *(recovery)* khususnya terhadap fasilitas IT dan sistem informasi serta *database*.

Proses perencanaan suatu *disaster recovery plan* (DRP) akan melindungi organisasi dari kegagalan layanan komputer utama, meminimalisasi risiko organisasi terhadap penundaan *(delay)* dalam penyediaan layanan, menjamin keandalan sistem yang tersedia melalui pengujian dan simulasi, serta meminimalisasi proses pengambilan keputusan oleh manusia selama bencana.

2.2 Oracle *Database*

Basis data Oracle adalah basis data relasional yang terdiri dari kumpulan data dalam suatu sistem manajemen basis data RDBMS. Perusahaan perangkat lunak Oracle memasarkan jenis basis data ini untuk bermacam-macam aplikasi yang bisa berjalan pada banyak jenis dan merk perangkat keras komputer *(platform)*.

Basis data Oracle ini pertama kali dikembangkan oleh Larry Ellison, Bob Miner dan Ed Oates lewat perusahaan konsultasinya bernama Software Development Laboratories (SDL) pada tahun 1977.

Tahun 1979 SDL berubah nama menjadi RSI (Relational Software, Inc.) dan memperkenalkan produk Oracle Versi 2 sebagai awal produk komersial *relational database system*. Versi ini tidak mendukung transaksi tapi menerapkan basic SQL untuk *query* dan *join*. RSI tidak meluncurkan versi 1, sementara versi 2 dianggap sebagai trik marketing.

Pada tahun 1983, perusahaan ini berubah nama menjadi Oracle Corporation sampai sekarang. Oracle mendominasi pasar *database server*, hal ini mungkin didasarkan kepada banyak perusahaan berskala besar menggunakan Oracle dalam mengelola datanya.

2.3 Konsep Oracle Data Guard

Oracle Data Guard dapat menjamin *high availability*, *data protection*, dan *disaster recovery* bagi data perusahaan. Data Guard menyediakan layanan-layanan untuk membuat, mengelola, serta melakukan *monitoring* terhadap satu atau lebih *standby database* agar *production database* dapat bertahan dari bencana dan kerusakan data. *Standby database* adalah salinan dari *production database* yang konsisten secara transaksional.

Jika *production database* mengalami *downtime* karena sebab tertentu, maka Data Guard akan mengalihkan tugas dan fungsinya kepada *standby database*. *Downtime* pun dapat di minimalisir. Dengan Data Guard, administrator dapat dengan bebas meningkatkan *performance* dari *production database* dengan menyerahkan *backup* dan operasi *reporting* kepada *standby database*.

2.3.1 Susunan Data Guard

Susunan Data Guard terdiri atas satu *production database* dan satu atau lebih *standby database*. *Database-database* yang ada dalam sistem Data Guard disarankan terpisah secara geografis. *Production* dan *standby database* dapat dikelola dengan menggunakan SQL *command-line interfaces* atau Data Guard Broker *interfaces*, yaitu *command-line interface* (DGMGRL) dan *graphical user interface* yang terintegrasi dalam Oracle *Enterprise Manager Grid Control*.

Production database adalah *database* utama yang diakses oleh kebanyakan aplikasi. *Production database* dapat berupa *single-instance database* atau *Oracle Real Application Cluster Database*.

Standby database dapat dibuat hingga sembilan unit dan semuanya dapat digabungkan dalam sistem Data Guard. Seperti *production database*, *standby database* juga dapat berupa *single-instance database* atau *Real Application Cluster Database*.

2.3.2 Role Transition

Oracle *database* beroperasi pada salah satu dari dua role, yaitu *primary* atau *standby*. Dengan Data Guard, *role* sebuah *database* dapat diubah dengan melakukan operasi *switchover* atau *failover*. *Switchover* adalah operasi penukaran *role* antara *primary database* dengan salah satu dari *standby database*. Operasi *switchover* biasanya dilakukan untuk alasan perawatan rutin atau *downtime* lain yang telah direncanakan. Operasi ini menjamin tidak ada data yang hilang. Selama *switchover*, *primary database* beroperasi pada *standby role* dan *standby database* beroperasi pada *primary role*.

Operasi *failover* dilakukan hanya pada saat *primary database* mengalami *downtime* yang tidak direncanakan, misalnya terjadi *hardware failure* atau bencana alam. Operasi ini membuat *standby database* beroperasi pada *primary role*. *Database* administrator dapat melakukan konfigurasi agar Data Guard dapat menjamin tidak ada data yang hilang.

2.3.3 Data Guard *Broker*

Data Guard *broker* adalah manajemen *framework*

terdistribusi yang digunakan untuk mengotomatisasi pembuatan, pengelolaan, dan pengawasan sistem Data Guard. Data Guard *broker* secara logis mengelompokkan *primary* dan *standby database* dalam sebuah *broker configuration* sehingga keduanya dapat dikelola bersama sebagai unit yang terintegrasi. Manajemen *broker configuration* dapat dilakukan baik secara local maupun *remote* dengan Oracle *Enterprise Manager Grid Control graphical user interface* (GUI) atau Data Guard *command-line interface* (DGMGRL).

2.3.4 Mode proteksi Data Guard

Data merupakan aset penting bagi perusahaan. Pada situasi tertentu, data sangat dilindungi dan dijaga agar tidak rusak atau hilang. Pada situasi lain, ketersediaan *database* mungkin saja lebih penting daripada kehilangan data. Pada situasi yang lain lagi, beberapa aplikasi membutuhkan *performance database* yang maksimal dan hilangnya sedikit data dapat ditoleransi. Oracle Data Guard menyediakan tiga jenis mode proteksi yang dapat diterapkan sesuai dengan kondisi yang paling cocok dengan kriteria masing-masing mode.

Mode proteksi pertama adalah *maximum protection*. Mode ini menjamin tidak ada data yang hilang jika *primary database* mengalami *downtime*. Mode proteksi kedua adalah *maximum availability* yang mampu menyediakan perlindungan data level tinggi tanpa mengganggu atau membahayakan availabilitas dari *primary database*. Mode proteksi ketiga adalah *maximum performance* yang merupakan mode *default*, juga menyediakan perlindungan data level tinggi tanpa mempengaruhi *performance* dari *primary database*.

2.3.5 Perbedaan Metode Recovery Data Guard dengan Cara Konvensional

Recovery konvensional merupakan *file-based recovery*. Baik dengan menggunakan RMAN maupun secara manual, proses *recovery* konvensional pada dasarnya terdiri dari *backup*, *restoration*, serta *recovery*. Dibutuhkan langkah yang panjang untuk melakukan ketiga proses tersebut.

Oracle Data Guard menawarkan metode *recovery* yang lebih sederhana dan mudah. Ketiga proses *recovery* konvensional di atas dapat digantikan oleh satu kali konfigurasi yang dilakukan di awal, yaitu pada saat sistem Data Guard dibangun. Proses *backup* (dalam hal ini sinkronisasi antara *primary* dan *standby database*) dilakukan secara otomatis dengan pengiriman *redo data*. Ketika terjadi *failure*, tidak perlu melakukan proses restorasi dan *recovery* seperti cara konvensional. Data Guard hanya tinggal melakukan proses *switchover* atau *failover*.

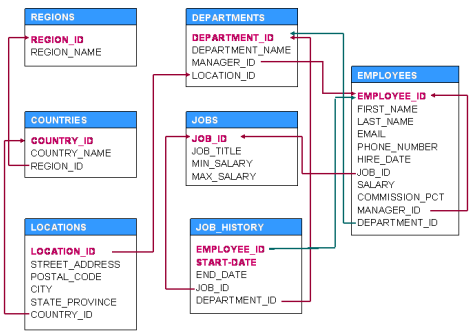
3. Perencanaan Sistem

3.1 Gambaran Umum dan Asumsi kerja

Dalam proyek akhir ini diasumsikan bahwa DBA dari PENS–ITS memutuskan untuk mengimplementasikan Oracle Data Guard guna melindungi Oracle *database* institusi tersebut. Manajemen sistem Data Guard dilakukan dengan menggunakan perintah-perintah SQL baik melalui *Enterprise Manager* Grid Control maupun melalui SQL*Plus.

Untuk merealisasikan asumsi tersebut akan dibuat satu *primary database* dan satu *standby database*. Setelah itu, dilakukan pemeriksaan dan pengujian terhadap konfigurasi yang telah dilakukan. Kemudian diberlakukan mode proteksi *maximum availability* yang mampu menyediakan proteksi data pada tingkat yang paling tinggi tanpa membahayakan *primary database*. Lalu hal yang terakhir dilakukan adalah menguji sistem dengan cara menambahkan *datafile* pada *primary database* serta memasukkan data baru pada salah satu tabel dalam skema HR. Apabila pada *standby database* terdapat data yang baru saja dimasukkan melalui *primary database*, maka sistem Data Guard dapat dikatakan berhasil dibangun.

Seperti yang telah disebutkan, dalam hal pengujian sistem Data Guard yang telah dibangun akan digunakan skema `HR` yang mempunyai struktur sebagai berikut.



Gambar 3 . 1 Skema `HR`

Selain itu, pengujian juga dilakukan dengan menambahkan *datafile* dari tablespace `EXAMPLE`. Penjelasan lebih rinci mengenai persiapan serta perencanaan sistem akan dibahas pada sub bab selanjutnya.

3.2 Kebutuhan *Hardware* dan *Software*

Berikut ini spesifikasi sistem komputer yang digunakan dalam proyek akhir ini.

Tabel 3 - 1 Spesifikasi Hardware dan Software Primary Database

No.	Deskripsi	Spesifikasi
1	CPU	Intel® Core™2 Duo CPU T6600 @ 2.20GHz
2	RAM	2 GB
3	*Platform*	Linux 32-bit

No.	Deskripsi	Spesifikasi
4	Sistem Operasi	Oracle *Enterprise* Linux *Release* 4 *Update* 7
5	Database	Oracle Database *Enterprise Edition* 10*g* R2
6	Sistem Koneksi	LAN 100Mbps

Tabel 3 - 2 Spesifikasi Hardware dan Software Standby Database

No.	Deskripsi	Spesifikasi
1	CPU	Intel® Pentium® 4 CPU @ 2.40GHz
2	RAM	2 GB
3	*Platform*	Linux 32-bit
4	Sistem Operasi	Oracle *Enterprise* Linux *Release* 4 *Update* 7
5	Database	Oracle Database *Enterprise Edition* 10*g* R2
6	Sistem Koneksi	LAN 100Mbps

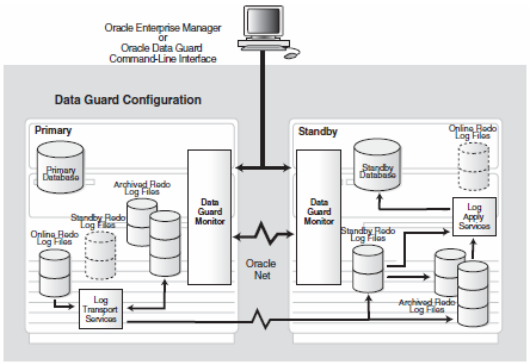
3.3 Skenario Pengerjaan

Rencana pengerjaan proyek akhir ini telah tersusun secara rinci dalam skenario berikut ini :

1. Menyiapkan *software* dan *hardware* serta sistem koneksi.
2. Membuat *primary database*.
3. Membuat *standby database*.
4. Memastikan bahwa *log transport service* dan *log apply service* berjalan.
5. Melakukan konfigurasi agar sistem Data Guard memberlakukan mode proteksi *maximum availability*.
6. Melakukan konfigurasi agar sistem Data Guard dapat melakukan *flashback*.
7. Mencoba menambahkan *data file* pada *primary database* kemudian memeriksa apakah pada *standby database* telah ditambahkan pula.
8. Mencoba menambahkan data baru pada tabel `REGIONS` dalam skema `HR` kemudian memeriksa apakah pada *standby database* telah ditambahkan pula.

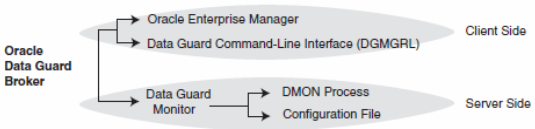
3.4 Skema Sistem

Di atas telah disinggung mengenai Data Guard *Broker* yang merupakan framework dan interface untuk mengelola sistem Data Guard, oleh karena itu penting untuk dimengerti mengenai konsep dan strukturnya. Sebelum lebih jauh membahas mengenai Data Guard *Broker*, berikut ini skema sistem Data Guard yang akan dibangun.



Gambar 3 . 2 Skema Sistem Data Guard

Pada gambar di atas, sistem Data Guard yang terdiri dari *primary* dan *standby database* dikendalikan dari suatu *host* melalui *Enterprise Manager* atau DGMGRL. Mekanisme tersebut berada dalam framework Data Guard *Broker* yang terbagi menjadi *client side* dan *server side*. Berikut ini skema komponen Data Guard *Broker*.



Gambar 3 . 3 Komponen Data Guard *Broker*

Framework Data Guard *Broker* terdiri dari *client side* dan *server side*. *Client side* merupakan *user interface* yang digunakan untuk membuat, mengelola, mengontrol, serta melakukan *monitoring* sistem. Dengan DGMGRL kita dapat megelola sistem langsung melalui *command-line, batch program*, atau *script*. *Server side* yaitu Data Guard Monitor merupakan komponen yang terintegrasi dengan Oracle *database* yang diperlukan untuk konfigurasi, pengelolaan, *control*, serta *monitoring* terhadap *Broker*. Data Guard Monitor terdiri dari proses DMON dan file konfigurasi.

3.5 Penyiapan Sistem Operasi dan *User Environment*

Sebelum memulai membangun sistem Data Guard ada beberapa hal yang harus dipersiapkan berkaitan dengan sistem operasi dan *user environment* untuk memenuhi *requirement* yang telah ditentukan.

Persiapan tersebut meliputi pembuatan *user* `oracle` serta grup terkait, penyesuaian profil *shell* untuk *user* `oracle`, pembuatan direktori untuk *software* yang akan diinstal, dan yang terakhir adalah konfigurasi kernel *parameters*.

4. Implementasi

4.1 Konfigurasi *Primary Database*

Database yang akan diinstal adalah Oracle *Database* 10*g Release* 2 *Enterprise Edition*. Hal ini dikarenakan versi *database* tersebut mendukung *grid computing* yang dapat menaungi sistem Data Guard. Informasi mengenai *primary database* dapat dilihat pada tabel berikut.

Tabel 4 - 1 *Primary Database*

Parameter	***Value***
Database Name	malang
Instance Name	malang
Database Unique Name	malang
Target Name	malang
Oracle Home	/u01/app/oracle/product/10.2.0/db_1
Host	oraserver.eepis-its.edu
IP Address	10.252.111.7

4.2 Konfigurasi *Broker*

Agar *database* pada masing-masing komputer dapat dikelola dengan mudah melalui *Enterprise Manager Grid Control*, maka kita perlu melakukan instalasi dan konfigurasi pada Oracle *Management Service*. Konfigurasi ini sangat menentukan kelancaran sistem Data Guard. Jika ada satu saja parameter yang tidak ditentukan dengan benar, maka akan mempengaruhi keseluruhan sistem. Apabila terjadi kegagalan sistem di kemudian waktu karena kesalahan pada konfigurasi *broker*, maka akan sangat sulit dicari sumber permasalahannya. Oleh karena itu, konfigurasi ini harus dilakukan dengan hati-hati dan teliti.

Broker, dalam hal ini *Enterprise Manager Grid Control* nantinya akan digunakan untuk mengelola semua *database* dalam sistem. Jadi, pengelolaan dilakukan secara terpusat melalui sebuah *interface* berupa *website*.

4.3 *Standby Database*

Standby database dibuat dari hasil *backup primary database*. Proses pembuatan *standby database* dilakukan melalui *Enterprise Manager* dengan cara menyalin *primary database*. Informasi mengenai *standby database* dapat dilihat pada tabel berikut.

Tabel 4 - 2 *Standby Dtabase*

Parameter	***Value***
Database Name	malang
Instance Name	surabaya
Database Unique Name	surabaya
Target Name	surabaya
Oracle Home	/u01/app/oracle/product/10.2.0/db_1
Host	Oraserver1.eepis-its.edu
IP Address	10.252.111.77

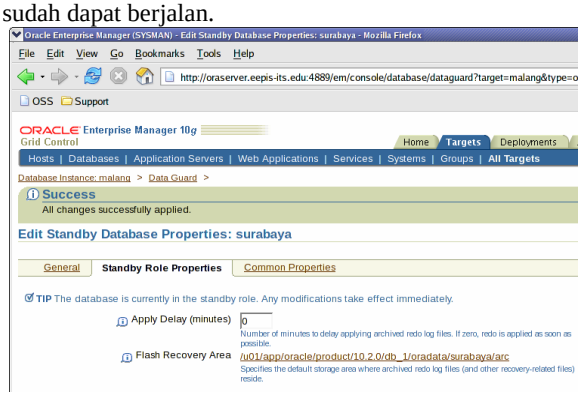
4.4 Pengaktifan Mode Proteksi *Maximum Availability*

Sebelum melakukan konfigurasi terhadap mode proteksi, nilai dari parameter log transport mode harus diatur terlebih dahulu. Untuk itu, parameter `LOG_ARCHIVE_DEST_`*n* harus diisi dengan atribut `LGWR`, `SYNC`, dan `AFFIRM`. Pengaktifan mode proteksi dapat dilakukan melalui *Enterprise Manager Grid Control*, DGMGRL, atau SQL*Plus. Berikut ini contoh jika menggunakan DGMGRL.

```
DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS
MAXAVAILABILITY;
```

4.5 Konfigurasi *Real-time Apply*

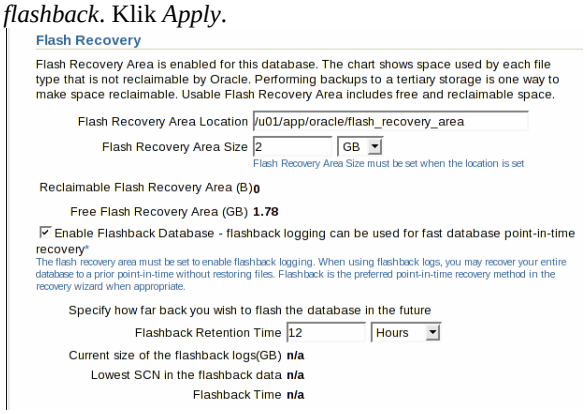
Dari halaman utama Data Guard, pilih *standby database* dan klik *Edit*. Pada bagian *Standby Role Properties* masukkan nilai nol pada *Apply Delay (minutes)*. Klik *Apply*, maka *real-time apply service* sudah dapat berjalan.



Gambar 4 . 1 Konfigurasi Real-time Apply Service

4.6 Konfigurasi *Flashback Database*

Dari halaman *Maintenance* pada *primary database* klik pada *Recovery Settings*. Pada halaman yang muncul tentukan seberapa jauh *database* akan melakukan *flashback*. Klik *Apply*.



Gambar 4 . 2 Konfigurasi *Flashback Database*

4.7 Pengujian dan Analisa Sistem

Pengujian sistem dilakukan melalui dua skenario, yaitu penambahan *datafile* untuk *tablespace* `EXAMPLE` dan penambahan data pada tabel `REGIONS` dalam skema `HR`. Semua pengujian dilakukan dari *primary database*.

Sistem Data Guard sudah dapat dikatakan berjalan dengan baik jika uji coba penambahan yang dilakukan juga berpengaruh terhadap *standby database*. Artinya, jika pada *standby database* terdapat *datafile* baru dalam *tablespace* `EXAMPLE` serta data baru pada tabel `REGIONS` dalam skema `HR`, maka dapat dipastikan bahwa semua konfigurasi yang dilakukan telah benar.

Untuk menambahkan *datafile*, masuk pada halaman *Administration* dari *primary database*. Klik pada *tablespaces*, akan ditampilkan halaman yang memuat informasi mengenai *tablespaces* yang ada pada sistem. Pilih *tablespace* `EXAMPLE` dan *Add Datafile* pada *Actions*, lalu klik *Go*.

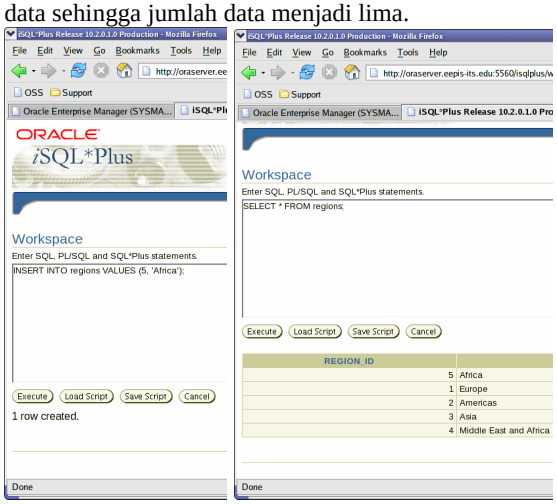
Pada gambar di bawah ini ditunjukkan bahwa *datafile* baru yang akan ditambahkan diberi nama `example02.dbf`. tampak pula bahwa letak *file* berada pada *primary database*. Klik *OK* maka akan ditampilkan informasi bahwa *datafile* telah berhasil ditambahkan.

Gambar 4 . 3 Penambahan *Datafile*

Untuk pengujian pada *standby database* dilakukan secara *remote* melalui koneksi `ssh` kemudian dijalankan *query* untuk menampilkan *datafile* yang ada pada *standby database*. Tampak pada gambar di bawah ini bahwa telah ada *datafile* dengan nama `example02.dbf`. Hal tersebut mengindikasikan bahwa sistem Data Guard telah berjalan lancar.

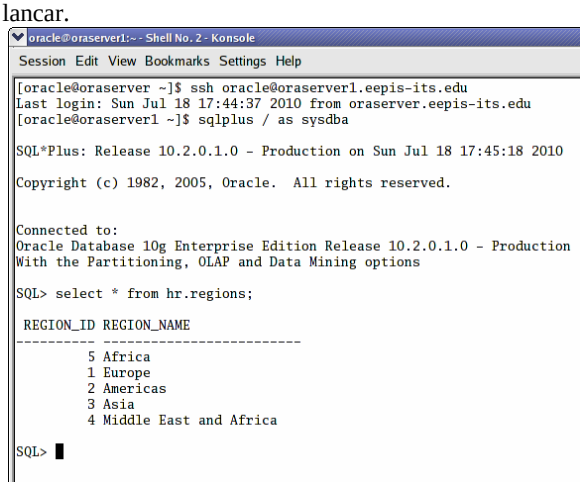
Gambar 4 . 4 Pengujian pada *Standby Database*

Pengujian selanjutnya adalah penambahan data pada tabel `REGIONS` dalam skema `HR`. Uji coba dilakukan melalui *i*SQL*Plus dari *primary database*. Untuk melakukan *querying* dan penambahan data digunakan user `HR`. Dalam uji coba ini akan ditambahkan satu lagi data sehingga jumlah data menjadi lima.



Gambar 4 . 5 Operasi Penambahan Data

Untuk pengujian pada *standby database* dilakukan secara *remote* melalui koneksi `ssh` kemudian dijalankan *query* untuk menampilkan data yang ada pada tabel `REGIONS`. Tampak pada gambar di bawah ini bahwa data pada tabel telah bertambah. Hal tersebut sekali lagi mengindikasikan bahwa sistem Data Guard telah berjalan lancar.



Gambar 4 . 6 Hasil pada *Standby Database*

Dari penelitian yang dilakukan, penyusun mengalami kesulitan pada saat konfigurasi, terutama saat mengintegrasikan *primary server* dengan *standby server*. Namun ketika kedua *server* sudah tersambung dan dapat berkomunikasi, konfigurasi menjadi lebih mudah. Pada waktu seluruh konfigurasi telah dilakukan dan sistem Data Guard berhasil dibangun, manajemen dan penggunaan sistem ternyata mudah, sederhana dan cukup *user-friendly*.

5. Penutup

5.1 Kesimpulan

Setelah dilakukan percobaan dan analisa terhadap kinerja sistem, dapat disimpulkan bahwa:

1. Implementasi sistem Data Guard terbagi menjadi dua bagian, yaitu pembuatan dan penggunaan. Proses pembuatan harus dilakukan dengan sangat cermat dan hati-hati karena cukup rumit. Kesalahan pada proses ini dapat berakibat sangat buruk pada kinerja sistem. Namun setelah pembuatan berhasil, penggunaan dan manajemen sistem sangat mudah dan sederhana terutama dalam hal *recovery* data.
2. Metode *recovery* Data Guard dapat menggantikan metode *recovery* konvensional yang menghabiskan banyak waktu. Dengan Data Guard, *downtime* dapat diminimalisir hingga kurang dari 10 detik.
3. Penggunaan Data Guard *broker* sangat membantu dalam mengelola sistem, hal ini karena sistem dapat dikontrol secara terpusat melalui satu *interface*.
4. Data Guard merupakan solusi yang murah namun tetap efektif dan dapat diandalkan untuk menjamin ketersediaan data.

5.2 Saran

Tugas akhir ini masih dapat terus dikembangkan karena teknologi Data Guard juga masih terus berkembang. Saran-saran yang dapat diberikan untuk pengembangan implementasi Data Guard antara lain:

1. Lebih baik membuat lebih dari satu *standby database* dengan lokasi yang berjauhan antar *database*.
2. Untuk kemudahan dan kelancaran manajemen sistem disarankan mengimplementasikan *framework* Data Guard *broker* dengan *Enterprise manager Grid Control* sebagai *interface*.
3. Selalu melakukan *update* dan *patching software* agar terhindar dari *bug* dan *error*.

Daftar Pustaka

[1] Ashish, R. & Kuhn, D. (1998). *Oracle Data Guard: Maximum Data Protection at Minimum Cost* [PowerPoint slides]. Retrieved from http://download.oracle.com/owsf_2003/40056_Ray.ppt

[2] Keesling, D. & Dyke, R.V. 2005. *Oracle Database 10g: Data Guard Administration Student Guide*. California: Oracle Corporation.

[3] Schupmann, V. 2008. *Oracle Data Guard Concepts and Administration, 10g Release 2 (10.2)*. California: Oracle Corporation.

[4] Schupmann, V. 2006. *Oracle Data Guard Broker, 10g Release 2 (10.2)*. California: Oracle Corporation.

[5] Polk, Jennifer. 2005. *Oracle Database Net Services Administrator's Guide, 10g Release 2 (10.2)*. California: Oracle Corporation.

[6] http://www.oracle.com