

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Analisis Sistem**

Menurut Whitten (2004), analisis sistem adalah sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi bagian – bagian komponen dengan tujuan mempelajari seberapa bagus bagian – bagian komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka.

Menurut Jogiyanto (1999), Analisis sistem dapat didefinisikan sebagai penguraian dari suatu system informasi yang utuh ke dalam bagian – bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan – permasalahan, kesempatan – kesempatan, hambatan – hambatan yang terjadi dan kebutuhan – kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan – perbaikannya. Tujuan utamanya adalah untuk memahami sistem dan masalah yang ada, untuk menguraikan kebutuhan sistem dan untuk menetapkan prioritas pekerjaan sistem selanjutnya. Terdapat empat tahap atau langkah umum dalam analisis sistem :

1. Survei sistem berjalan.
2. Mengidentifikasi kebutuhan sistem pemakai.
3. Mengidentifikasi kebutuhan sistem yang perlu untuk memenuhi kebutuhan sistem pemakai.
4. Penyajian laporan analisis sistem.

## 2.2. *High Availability*

*Availability* adalah sejauh mana sebuah sistem, aplikasi, layanan atau fungsi dapat diakses berdasarkan kebutuhan (Schupmann, 2009). *High Availability* yang juga sering disebut *Failover Cluster* pada umumnya diimplementasikan untuk tujuan meningkatkan ketersediaan layanan yang disediakan oleh *cluster* tersebut. Elemen *cluster* akan bekerja dengan memiliki node-node redundan, yang kemudian digunakan untuk menyediakan layanan saat salah satu elemen *cluster* mengalami kegagalan. Kemampuan meningkatkan ketersediaan layanan meliputi antisipasi / persiapan dari hal-hal yang tidak terduga seperti aliran listrik padam sampai dengan bencana baik itu yang disebabkan oleh manusia ataupun alam. Data *high availability* terdiri dari 4 komponen :

### 1. *Reliability*

Kemampuan memberikan layanan secara berkelanjutan walaupun terdapat gangguan pada sistem yang menyediakan layanan. Tingkat reliabilitas suatu sistem dapat ditingkatkan melalui fitur yang disediakan oleh sistem tersebut dan sistem yang *reliable* tidak hanya memberikan informasi kerusakan ataupun kegagalan pada sistem tetapi sebaliknya, mendeteksi dan jika mungkin menkoreksi kegagalan ataupun kerusakan yang terjadi.

### 2. *Recoverability*

Kemampuan membangun kembali layanan jika terjadi kesalahan ataupun kegagalan komponen ke titik *Point-in-time Recovery*.

### 3. *Timely Error Detection*

Penanganan responsif dan *fast detection* terhadap error jika terjadi kegagalan ataupun kesalahan yang tidak terduga terhadap sistem.

### 4. *Continous Operation*

Kemampuan menyediakan *access* terhadap data secara berkelanjutan.

## 2.3. **Database Management System (DBMS)**

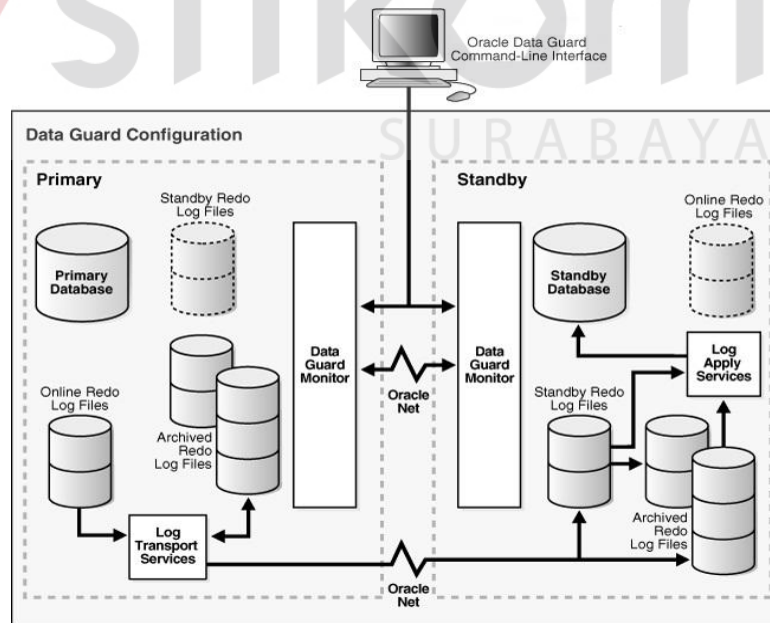
*Database* adalah sekumpulan data/informasi yang tersimpan secara teratur berdasarkan kriteria/kelompok tertentu, dimana kriteria tersebut saling berhubungan satu sama lain. Sedangkan *Database Management System (DBMS)* adalah kumpulan program perangkat lunak (*software*) yang memperbolehkan *user* untuk membuat dan memelihara *database*. Semua *DBMS* tersebut disatukan oleh *Relational Database Management System (RDBMS)*.

## 2.4. **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 computer (*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.

## 2.5. Oracle *Dataguard*

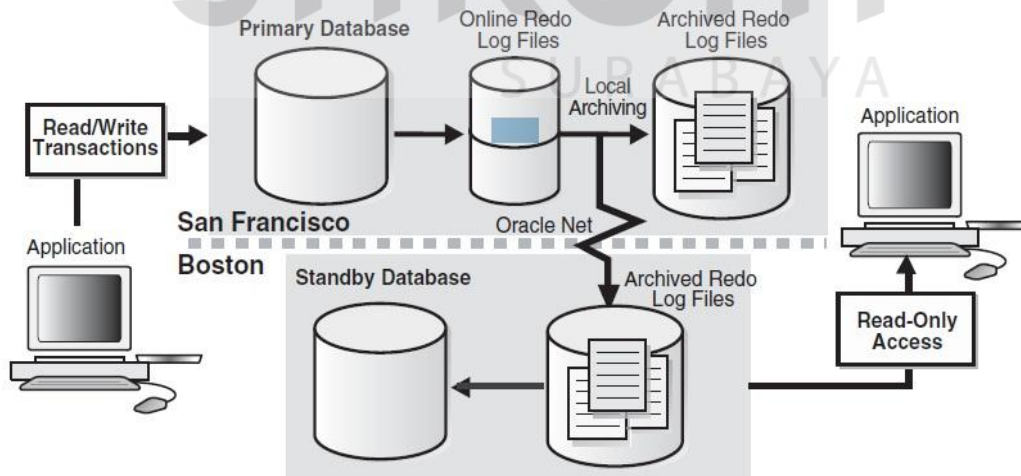
*Dataguard* adalah solusi *high availability* dari Oracle, tujuannya adalah untuk membuat *database standby* agar bila sewaktu-waktu *database production (primary)* mengalami kegagalan maka *database standby* akan menggantikan posisi menjadi *production* (Rohmad, 2008b). *Dataguard* menjaga *standby database* sebagai salinan data transaksi yang konsisten dari *primary database* untuk melindungi data perusahaan dari kegagalan, bencana, kesalahan dan kerusakan-kerusakan yang mengancam. *Standby database* ini dapat ditempatkan pada lokasi yang berada beribu-ribu mil dari *data center* utama, atau dapat ditempatkan di lokasi yang sama. Jika *primary database* menjadi tidak tersedia baik direncanakan maupun tidak direncanakan. *Dataguard* dapat mengganti *standby database* untuk mengambil peran sebagai *primary*, sehingga dapat mempersingkat *downtime*. Pada gambar 2.1 berikut adalah skema konfigurasi *Dataguard* secara umum.



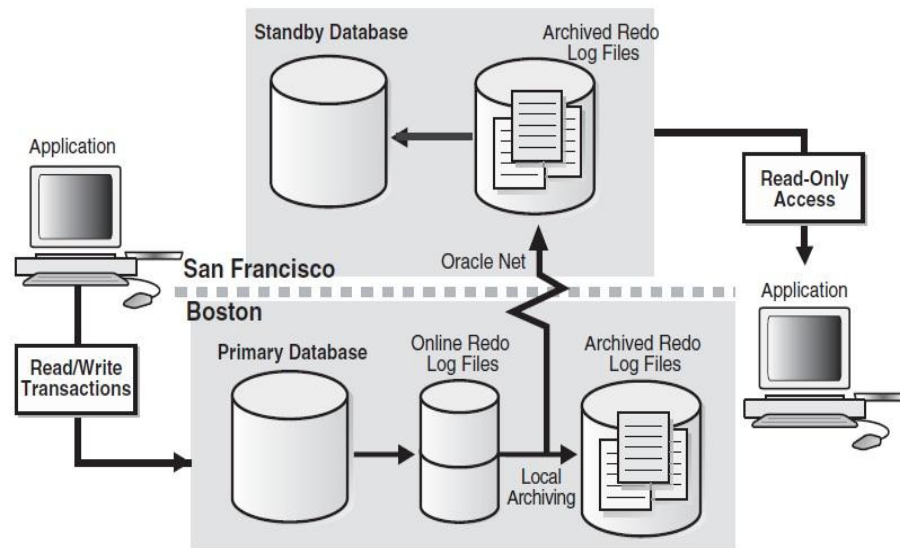
Gambar 2.1 Skema *Dataguard*. (Sumber : Schupmann hlm. 39)

### 2.5.1 Role Transition

*Role Transition* merupakan perubahan fungsi dari *primary* ke *standby database* ataupun sebaliknya (Rohmad, 2008a). Oracle *dataguard* beroperasi pada salah satu dari dua *role*, yaitu *primary* dan *standby*. Dengan *dataguard*, *role* sebuah *database* dapat diubah dengan melakukan operasi *switchover* dan *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. Sebelum *switchover*, *user* melakukan transaksi pada *primary database* dan *standby database* dapat digunakan untuk akses *read-only* yang bisa digunakan untuk keperluan *reporting* namun setelah *switchover*, *primary database* beroperasi pada *standby role* dan *standby database* beroperasi pada *primary role* seperti gambar berikut.

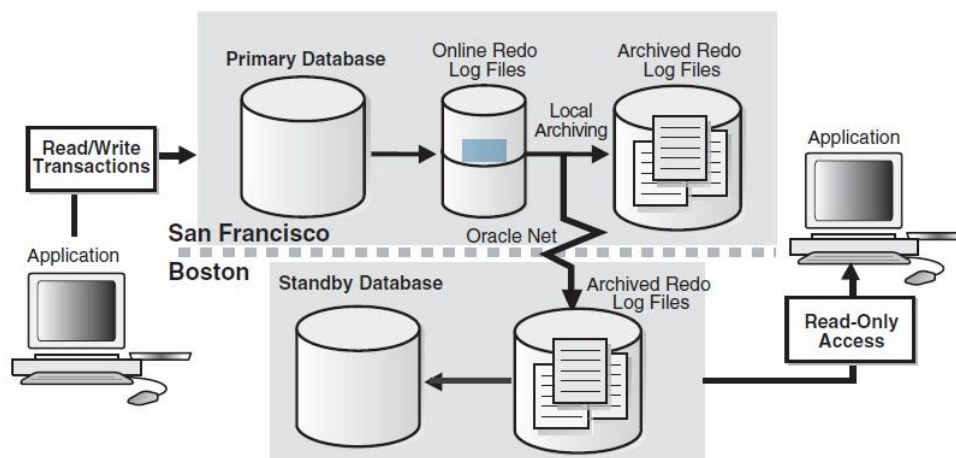


Gambar 2.2 *Dataguard* sebelum *Switchover* (Sumber : Schupmann hlm. 110)

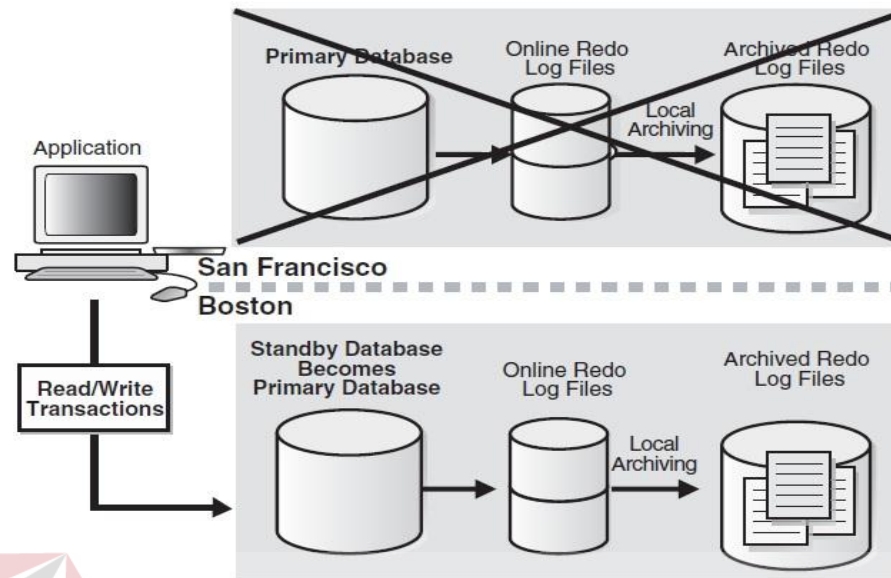


Gambar 2.3 *DataGuard* setelah *Switchover* (Sumber : Schupmann hlm. 111)

Sedangkan 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*. Setelah operasi *failover*, *primary database* yang mengalami kegagalan tidak dapat diakses dan memerlukan intervensi *Database Administrator* untuk menkonfigurasi ulang agar dapat digunakan kembali.



Gambar 2.4 *DataGuard* sebelum *Failover* (Sumber : Schupmann hlm. 112)



Gambar 2.5 *DataGuard* setelah *Failover* (Sumber : Schupmann hlm. 112)

### 2.5.2 Mode Proteksi

Data merupakan asset 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 kinerja *database* yang maksimal dan hilangnya sedikit data dapat ditoleransi. Oracle *DataGuard* menyediakan tiga jenis mode proteksi yang berperan agar data pada *standby database* tetap sinkron dengan *primary* yang juga berfungsi agar tidak terjadi *gap* pada *redolog* (Oracle Corporation, 2012).

#### 1. *Maximum Protection*

Mode ini menjamin tidak ada data yang hilang jika *primary database* mengalami *downtime*, transaksi tidak akan di *commit* sampai *log* terkirim ke *standby database*, tetapi ketika terjadi kegagalan pada *standby database* sehingga *log* dari

transaksi tidak dapat di *transmit* oleh *primary*, maka secara otomatis *primary* akan *shutdown* dengan sendirinya.

## 2. *Maximum Availability*

Mode proteksi ini menyediakan perlindungan data level tinggi tanpa mengganggu atau membahayakan ketersediaan dari *primary database*, mode ini memiliki kesamaan dengan *maximum protection*, bedanya ketika terjadi kegagalan pada *standby database* sehingga *log* dari transaksi tidak dapat di *transmit* oleh *primary*, maka *instance* dari *primary* tetap berjalan dan merubah status *protection* menjadi *maximum performance* sehingga transaksi tetap bisa berjalan.

## 3. *Maximum Performance*

Mode proteksi ini merupakan mode *default* dari instalasi Oracle *database*, juga menyediakan perlindungan data level tinggi tanpa mempengaruhi *performance* dari *primary database*, biasanya mode ini digunakan jika jarak *Datacenter* dengan *Disaster Recovery Center* (DRC) berjauhan. Dengan *setting* seperti ini, maka setiap transaksi yang dilakukan dapat segera di *commit* tanpa perlu menunggu pengiriman redo data dari *primary* ke *standby database*. Kelemahan pada *setting* ini adalah jika transaksi sedang berjalan dan *primary* mengalami gangguan secara tak terduga sehingga data-data transaksi yang belum sempat diproses oleh *standby database* akan hilang jika dilakukan *failover*.

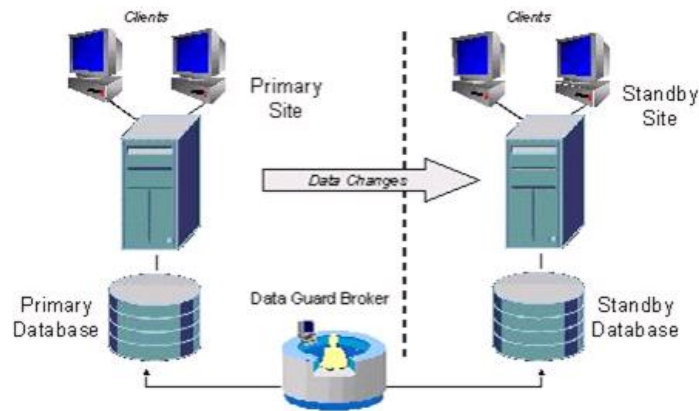


Tabel 2.1 Syarat Mode Proteksi Oracle

	Maximum Protection	Maximum Availability	Maximum Performance
Proses Redo Archival	LGWR	LGWR	LGWR atau ARCH
Tipe Transmisi Network	SYNC	SYNC	SYNC atau ASYNC.
Tipe Penulisan Disk	AFFIRM	AFFIRM	AFFIRM atau NOAFFIRM
Standby Redolog	Ya	Ya	Tidak, namun disarankan

### 2.5.3 Dataguard Broker

*Dataguard broker* adalah manajemen *framework* terdistribusi yang digunakan untuk mengotomatisasi pembuatan, pengelolaan dan pengawasan sistem *dataguard* (Samsiyar, 2006). *Dataguard 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 lokal maupun *remote* dengan Oracle *Enterprise Manager Grid Control graphical user interface* atau *Dataguard command-line interface* (DGMGRL).



Gambar 2.6 Skema *DataGuard Broker* (Sumber : Schupmann 122)

#### 2.5.4 Fast-Start Failover

*Fast-start failover* adalah salah satu fitur yang terdapat pada Oracle Database 11g yang bekerja secara otomatis dengan mengaktifkan peran *standby database* sebagai *primary* ketika *broker* serta *standby* kehilangan kontak dengan *primary* yang berarti mengindikasikan terjadinya *failure* pada *database primary*, proses ini tidak memerlukan intervensi dari seorang DBA untuk melakukan manual *failover*. *Fast-start failover* dapat digunakan pada mode proteksi *maximum availability* ataupun *maximum performance*, pada *maximum availability* proses *failover* menjamin tidak ada data yang hilang sedangkan pada *maximum performance* menjamin tidak ada data yang hilang melebihi waktu yang ditentukan oleh parameter *FastStartFailoverLagLimit*, parameter ini mengindikasikan banyaknya data hilang yang dapat ditoleransi ketika proses *failover* terjadi dan parameter ini hanya berlaku ketika *database* dalam mode *maximum performance*.

## 2.6. Oracle Solaris

Oracle Solaris, sebelumnya dikenal sebagai Sun Solaris merupakan sebuah sistem operasi keluarga Unix yang sebelumnya dikembangkan oleh Sun Microsystems Inc, seperti halnya HP-UX dari HP dan AIX dari IBM. Sun Microsystems menggantikan Sun OS sebelumnya yang telah ada pada tahun 1999 sedangkan Solaris adalah gabungan antara *operating system* SunOS, *interface* GUI CDE (Common Desktop Environment), dan *software* ONC+ (*Open Network Computing*) (Masruri, 2008). Setelah proses akuisisi Sun oleh Oracle pada bulan Januari 2010, Solaris menjadi lebih dikenal sebagai Oracle Solaris. Dari berbagai macam versi Unix yang ada saat ini tidak ada yang menyamai solaris dari segi penggunaan yang luas dalam segala industry (Masruri, 2008), hal ini disebabkan karena sejak berdirinya pada tahun 1982 Sun Microsystems fokus pada pengembangan Unix dan tidak pernah berpindah dari Unix *Operating System* seperti HP dan IBM yang juga mengadopsi Windows sebagai OS Server mereka. Sistem operasi Solaris dikenal secara luas karena skalabilitasnya yang dimilikinya, utamanya pada sistem komputer berbasis SPARC, dan sejumlah fitur-fitur inovatif yang dibawanya seperti :

1. DTrace
2. ZFS (Zettabyte File System)
3. *Time Slider*.

Sistem operasi ini dapat dijalankan di atas prosesor x86 baik 32bit atau 64bit (berbasis instruksi Amd64), serta prosesor SPARC baik yang diproduksi oleh Sun

ataupun Fujitsu. Solaris terdaftar sebagai sistem operasi yang kompatibel dan memenuhi spesifikasi *Single Unix Specification*.

