

ANALISIS DAN IMPLEMENTASI STANDBY DATABASE DENGAN ORACLE (STUDI KASUS: BACKUP STANDBY DATABASE PADA PERUSAHAAN JASA KOMUNIKASI)

Astianto Wibowo¹, Dhinta Darmantoro², Dana Sulistyio Kusumo³

Jurusan Teknik Informatika, Sekolah Tinggi Teknologi Telkom, Bandung

e-mail: ¹astianto.wibowo@gmail.com, ²ddd@stttelkom.ac.id, ³dana@stttelkom.ac.id

ABSTRAKSI

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

Standby database adalah salinan dari database regular yang terinstal pada system yang terpisah. Jika failure terjadi pada system primer, system standby itu diaktifkan dan mengambil alih proses, sehingga meminimalisasi efek dari failure yang terjadi terhadap faktor availability dan downtime. Oracle menjaga standby database agar terupdate dengan menerapkan archived redo log yang dikirimkan dari database primer secara konstan.

Penelitian ini mengangkat analisis dan implementasi dari standby database dengan menggunakan dbms oracle 9i. Akan dibuat juga suatu perangkat lunak pencatatan archived redo log dari primary database untuk dibandingkan dengan archived redo log yang terdapat pada standby database serta simulasi transaksi berupa proses DML.

Kata kunci: standby database, primary database, archived redo log.

1. PENDAHULUAN

1.1 Latar Belakang

Di setiap sistem database, kemungkinan terjadinya failure terhadap sistem dan hardware 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 kritis bisa tetap berjalan, meskipun primary sistem sedang mengalami failure.

Sistem komputer, seperti hal peralatan lainnya, adalah sebuah alat yang dapat mengalami failure yang disebabkan beberapa hal: disk crash, power outage, software error, human error, bahkan sampai sabotase. Dalam failure apapun, terdapat ancaman hilangnya informasi. Oleh karenanya, sistem database harus mengambil langkah-langkah antisipasi untuk menjamin atomicity, yaitu jika transaksi sukses maka semua perubahannya tercatat dalam database begitupun sebaliknya, dan durability, yaitu jika transaksi sukses maka perubahannya tercatat dalam database dan bertahan terhadap failure apapun. transaksi yang ada. Sebuah bagian utuh dari sistem database adalah strategi backup yang menjamin keutuhan data. Strategi backup juga harus menyediakan tingkat availability yang tinggi; yaitu, harus menjamin tidak ada data yang hilang dan meminimalisasi data yang tidak bisa digunakan setelah terjadinya crash.

Untuk menentukan bagaimana sistem harus di-recover dari failure, maka harus diidentifikasi failure-failure tersebut. Langkah selanjutnya, kita harus memperkirakan efek dari failure tersebut

terhadap isi dari database. Setelah itu maka bisa disusun strategi backup yang dapat menjamin konsistensi database dan atomicity transaksi.

Oracle menyediakan beberapa solusi backup dan recovery sebagai counter terhadap failure yang mungkin terjadi. Oracle juga memungkinkan kita untuk membuat sebuah standby database sebagai backup dari production database atau primary database.

Standby database adalah sebuah duplikasi, atau standby, salinan dari sebuah database pada suatu remote site untuk menyediakan kontinuitas availability database primer saat terjadi kondisi failure. Standby database dibuat dengan tujuan untuk meminimalisir waktu downtime yang terjadi ketika primary sistem mengalami failure. Standby database diciptakan dengan menggunakan salinan spesial control file dari database primer. Standby database terjaga sinkronisasinya dengan database primer dengan menerapkan file archived log database primer ke standby database. Oleh karena itu diperlukan untuk mengoperasikan database primer pada ArchiveLog Mode untuk mendapatkan keuntungan dari standby database.

1.2 Perumusan Masalah

Dari penjelasan di atas maka dapat dirumuskan permasalahan pokok diantaranya adalah:

- Server yang berjalan 24 jam memerlukan sistem backup dengan downtime rate yang sekecil mungkin.

- Sistem backup yang dibuat diperlukan untuk meminimalkan *data lost* yang terjadi ketika *failure*.
- Perancangan implementasi *standby* database sebagai *backup* sistem.
- Pencatatan *archived log* untuk mengupdate *standby* database.

1.3 Tujuan Penulisan

Penelitian ini bertujuan untuk:

1. Membuat sistem backup guna menggantikan peran dari primary sistem ketika terjadi failure untuk meminimalisir downtime rate.
2. Membuat sistem backup dengan physical standby database.
3. Mensimulasikan terciptanya redo log pada primary database, dan
4. Membuat perangkat lunak untuk pencatatan archive log guna membandingkan archive log pada primary database dengan archive log pada standby database.

1.4 Metodologi Penyelesaian Masalah

Metodologi penyelesaian masalah dalam penelitian ini adalah:

1. **Studi Literatur;** tahap pendalaman materi, identifikasi masalah dan metodologi pemecahan serta metode perancangan pemodelan sistem.
2. **Pencarian dan pengumpulan data;** dari internet, jurnal, laporan-laporan dari lembaga lembaga terkait.
3. **Perancangan dan implementasi;** tahap penerapan dan transformasi dari representasi teoritik ke dalam pemodelan sistem.
4. **Analisa dan evaluasi;** tahap pengujian, analisa dan perbaikan.

2. LANDASAN TEORI

2.1 Primary Database

Primary database mengacu pada pengertian production database yang memiliki tingkat availabilitas tinggi. Primary database adalah database yang sering diakses oleh aplikasi kebanyakan.

2.2 Standby Database

Standby database adalah salinan konsisten secara transaksional dari primary database. Standby database diciptakan dari salinan backup dari primary database. Setelah tercipta maka standby database dijaga dengan menyalurkan redo data dari primary database ke system standby lalu menerapkan redo log tersebut ke standby database.

Serupa dengan primary database, standby database dapat berupa single-instance Oracle Database atau sebuah Oracle Real Application Clusters database.

Sebuah standby database dapat berupa sebuah physical standby database atau sebuah logical standby database.

2.2.1 Physical Standby Database

Physical standby database adalah database yang secara fisik identik dengan primary database, dengan struktur database on-disk identik dengan primary database pada basis block-for-block. Skema database termasuk indeks, adalah sama. Physical standby database terupdate dengan melakukan recovery, ini juga bisa sekedar recovering data atau dibuka untuk pelaporan read-only. Physical standby database tetap tersinkronisasi dengan primary database dengan melakukan recover terhadap redo data yang diterima dari primary database.

Oracle menjaga physical standby database dengan melakukan operasi managed recovery. Saat physical standby database tidak melakukan operasi recovery, maka dia dapat dibuka untuk operasi read-only.

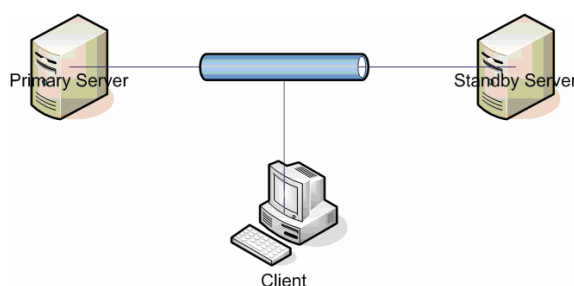
- **Managed Recovery**
Physical standby database dijaga dengan menerapkan archived redo logs ke standby system menggunakan mekanisme recovery Oracle. Operasi recovery menerapkan perubahan-perubahan blok-demi-blok menggunakan row ID fisik. Database tidak bisa dibuka untuk operasi-operasi read atau read/write saat redo data sedang diterapkan.
- **Open read only**
Physical standby database dapat dibuka untuk operasi read only sehingga dapat mengeksekusi query-query pada database itu. Saat dibuka untuk operasi read only, standby database dapat terus menerima redo log tetapi aplikasi data dari log tersebut ditunda sampai database melanjutkan operasi managed recovery.

Meskipun physical standby database tidak dapat menampilkan operasi read only dan managed recovery pada waktu yang bersamaan, tetapi bisa di-switch over.

3. ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

3.1 Konfigurasi Standby System

Konfigurasi standby yang akan dibangun adalah sebagai berikut:



Gambar 1. Konfigurasi Sistem

3.1.1 Konfigurasi Primary Database Untuk Pembuatan Standby Database

- Mempersiapkan file parameter inialisasi
SQL> create pfile='c:\pfilename.ora'
from spfile;
File created
- Memodifikasi file parameter inialisasi file tersebut ditambahkan dengan parameter-parameter di bawah ini:
*.log_archive_dest_1='LOCATION=/oradata /log/archivename'
*.log_archive_format='name%t_%s.arc'
*.log_archive_start=true
- Create Control File untuk standby database
SQL> alter database create standby controlfile as 'c:\control_standby.ctl';
- Membuat copy dari primary database
Pada primary database dilakukan langkah-langkah berikut untuk membuat backup copy dari primary database:
 - Identifikasi datafile dari primary database
Pada primary database dilakukan query terhadap datafile-datafile yang akan digunakan untuk membuat standby database.
SQL> select name from v\$datafile;
 - Shutdown primary database
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
 - Copy datafile ke lokasi sementara
 - Restart primary database
 - Untuk mengaktifkan mode archiving dilakukan langkah-langkah sebagai berikut:
 1. Startup database dengan status mounting menggunakan file parameter yang telah dibuat tadi.
SQL> startup mount
pfile='c:\pfileanalisa.ora';
 2. Ubah mode archivelog dari database.
SQL> alter database archivelog;
Database altered.
 3. Startup penuh database.
SQL> alter database open;
 4. Lihat status dari archivelog dan pastikan sudah pada mode archivelog menggunakan script berikut.
SQL> archive log list;

3.1.2 Konfigurasi Standby Database

Standby database harus memiliki struktur konfigurasi database yang identik dengan primary database, sehingga susunan direktori pun identik, dalam hal ini baik primary sistem dan standby sistem telah terinstall DBMS Oracle 9i untuk general purpose.

- Menyalin datafile, control file, PWD file, parameter inialisasi pada direktori yang sama di standby dengan direktori pada primary
- Restart dan mount standby database

```
SQL> startup nomount  
pfile='d:\pfileanalisa.ora';  
SQL> alter database mount standby  
database;
```

- Sinkronisasi dan apply archivelog pada standby database
SQL> recover standby database until
cancel;
Specify log: {<RET>=suggested |
filename | AUTO | CANCEL}
Pilih auto, maka oracle akan segera meng-
apply semua archivelog yang ada.

3.1.3 Aktivasi Standby Database

Langkah aktivasinya adalah sebagai berikut:

1. Aktifkan standby database dengan script berikut
SQL> alter database activate standby
database;
2. Mount database
SQL> alter database mount;
3. Start penuh database
SQL> alter database open;

Standby database yang telah diaktifkan telah berubah sifat menjadi primary database.

3.1.4 Skenario Uji

1. SU 1: Media failure yang menghilangkan datafile

Tujuan SU 1:

SU 1 bertujuan untuk menganalisis kemampuan standby database dalam menggantikan peran dari primary database yang mengalami media failure yang mengakibatkan kehilangan datafile.

2. SU 2: Media failure yang menghilangkan logfile.

Tujuan SU 2:

SU 2 memiliki tujuan yang sama dengan SU 1 hanya saja kali ini primary database mengalami kehilangan logfile akibat media failure.

3. SU 3: User error yang menghapus record dari tabel.

Tujuan SU 3:

SU 3 bertujuan menangani kehilangan data yang diakibatkan user error, meskipun pada kenyataannya selama studi kasus di PT PSN belum pernah terjadi.

4. SU 4: power outage

Tujuan SU 4:

Dari simulasi ini akan dilihat bagaimana standby database akan segera menggantikan primary database ketika mengalami power outage.

5. SU 5: Instance Failure

Tujuan SU 5:

SU 5 akan mensimulasikan software error yang mengakibatkan instance failure pada primary database.

tinggi karena dapat langsung diaktifkan dan beroperasi begitu terjadi failure terhadap primary database.

4. ANALISIS HASIL UJI COBA

Tabel 2. Summary Hasil Uji Standby Database

Point of Analysis	Media Failure Datafile	Media Failure Logfile	User Error	Power Outage	Instance Failure
waktu terjadinya failure	17:19:08	21:36:11	11:02:53	0:27:18	13:18:26
waktu terakhir backup archive log sebelum failure	17:18:33	21:35:58	11:02:38	0:25:57	13:17:59
waktu archive log terakhir terbentuk sebelum failure	17:18:29	21:35:40	10:51:22	0:25:54	13:17:59
waktu diaktifkannya standby database	17:21:29	21:38:50	11:03:25	0:29:33	13:20:34
jumlah record pada primary sebelum terjadinya failure	238	231	123	262	263
jumlah record yang ada pada standby setelah terjadinya failure	236	230	102	262	262
total transaksi yang terjadi pada masa simulasi	145	142	141	172	145
total transaksi yang ditangani primary	136	129	141	160	161
total transaksi yang ditangani standby	7	8	0	9	6
total transaksi yang gagal	2	5	0	3	3
gap record antara primary dan standby	0,84%	0,40%	0%	0%	0,38%
downtime rate	0:02:21	0:02:39	0:00:32	0:02:15	0:02:08

Recovery tanpa standby database memiliki prinsip yang sama dengan standby database yaitu mengembalikan database ke state point terakhir menggunakan archive log yang ada. Namun yang perlu dipertimbangkan dari recovery tanpa standby database adalah berapa lama waktu yang dibutuhkan untuk melakukan recovery yang akan mengakibatkan downtime rate terhadap transaksi.

Dari uji skenario diatas maka dapat disimpulkan dengan adanya standby database maka tingkat availability dapat terjamin karena dapat segera diaktifkan kapan pun sehingga meminimalisir downtime rate. Rata-rata downtime rate terhadap proses transaksi dengan adanya standby database hanya terjadi selama 2-3 menit.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil percobaan, kesimpulan yang dapat diambil setelah melakukan percobaan dengan standby database dalam menangani beberapa kondisi failure:

- Dengan adanya standby database maka dapat meminimalisir downtime yang terjadi pada proses bisnis dari suatu perusahaan
- Dari skenario uji yang telah dilakukan standby database menunjukkan tingkat availabilitas yang

5.2 Saran

Salah satu yang mungkin perlu dilakukan analisis lebih lanjut adalah bagaimana cara menghilangkan gap data yang mungkin terjadi hingga 0%, namun cara itu juga harus dapat melindungi database dari user error.

PUSTAKA

- Hunter, Jeff, Sr. 2004, *Creating an Oracle Standby Database*; http://www.idevelopment.info/data/Oracle/DB_A_tips/Standby_Database/SD_1.shtml
- Oracle Documentation. 2005, *Make Recovery Logical*; http://www.oracle.com/technology/oramag/oracle/04-jul/044tech_avail.html
- Hutabarat, Bernaridho Msc,ocp, Ahmadsyah Algozhi Nugroho, Ocp. 2004. *Oracle 8i/9i Backup & Recovery*. Penerbit Andi:Yogyakarta.
- Oracle Documentation. 2002, *Oracle Data Guard Concepts and Administration Release 2* ; <http://otn.oracle.com/docs/index.htm>
- Silberschatz, Korth, Sudarshan. 2002, *Database System Concepts 4th Edition*. McGraw Hill.
- Ashdown, Lance. 2002. *Oracle 9i Recovery Manager User's Guide*; <http://www.oracle.com/database/recovery>
- Cyran, Michele. 2002. *Oracle 9i Database Concepts*. Oracle Documentation.
- Ashdown, Lance. 2002. *Oracle 9i Backup and Recovery Concepts Release 2 (9.2)*. Oracle Documentation.
- Petroustos, Evangelos. 1996. *Mastering Visual Basic 6*. Sybex
- Kumar, Vijay, Meichun Hsu. 1998. *Recovery Mechanisms in Database Systems*. Prentice Hall PTR.
- Oracle Documentation. 2002. *SQL*Plus User's Guide and Reference, Release 9.2*
- Pressman, Roger S., Ph. D.1997. *Software Engineering: A Practitioner's Approach*. McGraw Hill.
- Oracle Documentation. 2004, *Oracle Data Guard Concepts and Administration-Creating a Physical Standby Database*; <https://cwisdbe.cc.kuleuven.ac.be/ora10doc/server.101/b10823/create-ps.htm>