Bab 1

Arsitektur Oracle

Server Oracle merupakan sistem ORDBMS *(object-relational database management system)* yang yang dilengkapi dengan sarana manajement informasi yang terbuka, komprehensif dan terintegrasi.

Pada bab ini akan diberikan wawasan mengenai komponen-komponen yang membentuk arsitektur server Oracle serta beberapa hal mengenai :

- Struktur yang membentuk koneksi antara user dan server Oracle
- Tahapan yang dilakukan oleh server Oracle pada saat memproses query.
- Tahapan yang dilakukan oleh server Oracle pada saat memproses statement DML.
- Tahapan yang dilakukan oleh server Oracle pada saat memproses COMMIT.

1.1. Jenis Client

1. User

User yang akan menggunakan databse Oracle dapat connect ke server Oracle dengan melalui beberapa cara sebagai berikut :

- *Single-Tiered*

  Logging secara langsung menuju host yang merupakan mesin komputer yang menjalankan program server Oracle, sebagai contoh user connect ke mesin UNIX yang menjalankan Oracle, kemudian menggunakan program Server Manager untuk mengakses database pada host yang sedang di logging.
- *Two-Tiered*

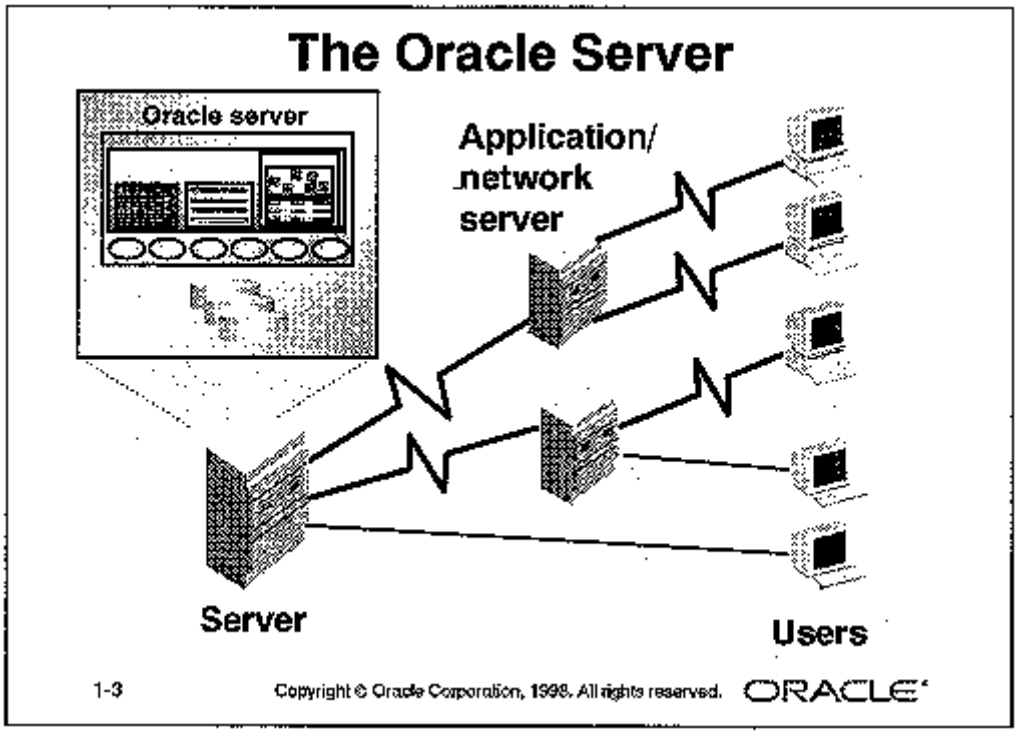
  Menggunakan koneksi secara *client-server*, dimana komputer yang digunakan user telah terhubung secara langsung dengan komputer yang menjalankan program server Oracle, sebagai contoh user menggunakan

program aplikasi Developer/2000 pada PC berbasis Windows98 untuk mengakses database Oracle pada server WindowsNT atau Windows2000.

- *Three-Tiered*

  Komputer yang dipergunakan user berkomunikasi dengan aplikasi atau server jaringan pada komputer yang menjalankan server Oracle, misalnya ketika user sedang menggunakan browser pada komputer jaringan untuk menggunakan aplikasi yang tersimpan pada server WindowsNT yang digunakan utnuk menampilkan data dari database Oracle pada host UNIX.

User dapat menampilkan data dari database menggunakan perintah SQL yang diberikan melalui SQL*Plus atau aplikasi yang berisi statement SQL, selanjutnya server Oracle memproses perintah tersebut dan mengirimkan hasilnya pada user.



2. Administrator

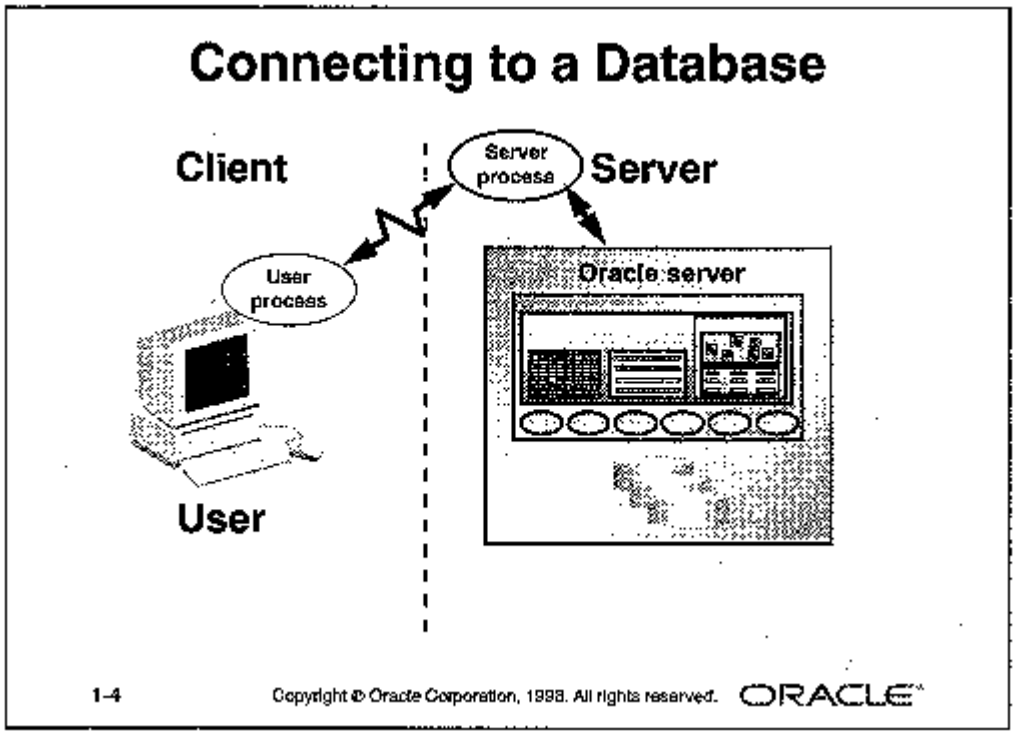
Administrator database bertanggung jawab pada perawatan server Oracle sehingga server dapat terus memproses semua permintaan user. Dengan mengerti

bentuk arsitektur server oracle maka diharapkan dapat melakukan perawatan secara efektiv.

Pada latihan yang akan dibahas pada diktat ini difokuskan untuk membuat dan merawat server Oracle dimana user dapat melakukannya dengan connect secara langsung pada PC yang menjalankan server Oracle atau menggunakan model *client-server*.

1.2. Menghubungkan ke Database

User yang berkeinginan untuk berinteraksi dengan server Oracle maka pertama kali perlu melakukan hubungan (koneksi) dengan database.



Pada langkah berikut ini adalah urut-urutan untuk melakukan kodeksi ke database, yaitu :

- User menjalankan tools seperti SQL*Plus atau aplikasi yang dibuat menggunakan tools seperti Developer/2000 Form. Dalam model *client-server* tools atau aplikasi dijalankan pada mesin PC client.

- Dalam konfigurasi yang banyak dipakai, user login menuju server Oracle menggunakan nama tertentu, password dan nama database. Jika berhasil maka process akan terbuat pada mesin PC yang menjalankan server Oracle. Process ini disebut dengan *server-process*, yang berguna untuk menghubungkan antara server Oracle dengan *user-process* yang berjalan sebagai client pada mesin PC user.

Komponen yang membentuk sistem *client-server* meliputi :

- *Koneksi*

  Koneksi merupakan jalur komunikasi antara proses pada user dan server Oracle. Jika user menjalankan tools atau aplikasi pada mesin PC yang sama sebagai server Oracle, maka jalur komunikasi yang dibuat menggunakan mekanisme komunikasi antar proses yang ada pada mesin PC tersebut. Jika user menjalankan tools pada mesin PC client, maka software jaringan yang mempergunakan jaringan untuk komunikasi antara user dan server Oracle.

- *Session*

  Session (sesi) merupakan koneksi khusus user terhadap server Oracle. Session akan mulai dibentuk pada saat user divalidasi pada server Oracle, dan akan berakhir ketika user logout atau terjadi proses yang berhenti secara tidak normal.

  Bagi user yang telah menggunakan database maka beberapa session dapat terbentuk secara concurent (bersama-sama) jika user tersebut login menggunakan beberapa tools, beberapa aplikasi atau beberapa terminal secara bersama-sama. Terkecuali untuk beberapa tools administrasi database yang khusus. Dengan menjalankan session database yang diperlukan maka server Oracle akan dapat siap untuk dipergunakan.

- *Troubleshooting*

  Pesan kesalahan Oracle “ORA-01034:Oracle not available” akan ditampilkan jika user atau aplikasinya mencoba melakukan koneksi ke server Oracle tetapi tidak dapat terlayani. Untuk mengatasi hal ini maka administrator database perlu untuk meng start-up ulang server Oracle.

1.3. User-process dan Server-process

Secara umum maka *user-process*, atau disebut dengan client, memiliki beberapa ciri-ciri sebagai berikut :

- Dibuat pada saat user menjalankan tools atau aplikasi seperti SQL*Plus, Server manager, atau aplikasi Developer/2000.
- Dijalankan pada sisi client, yaitu pada mesin PC dimana user login.
- *User-process* akan terbentuk pada saat tools mulai melakukan koneksi dan akan terhenti pada saat user exit (keluar0, atau dilakukan pemutusan hubungan.
- *User-process* memiliki *user-program-interface* (UPI)
- UPI berfungsi untuk membuat pemanggilan kepada server Oracle pada saat user melakukan request.

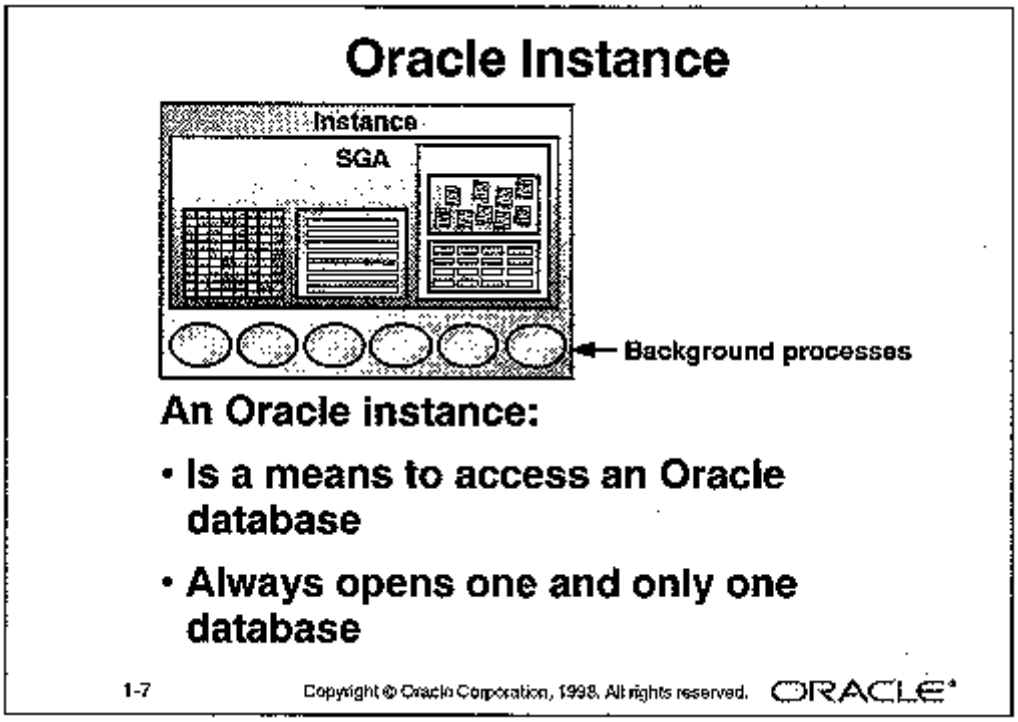
Secara umum maka *server-process*, memiliki beberapa ciri-ciri sebagai berikut :

- Dijalankan pada mesin yang sama dengan server Oracle
- Pada konfigurasi yang paling sederhana disebut dengan dedicated-server, setiap *server-process* melayani hanya satu proses user. *Server-process* terbentuk ketika user melakukan *request* untuk meminta koneksi dan umumnya server-process akan terhenti pada saat user memutuskan hubungan.
- Setiap *server-process* menggunakan suatu daerah memory yang disebut dengan *program-global-area* (PGA). PGA ini akan dijelaskan pada bagian berikutnya.
- *Server-process* menggunakan *oracle-program-interface* (OPI), yang digunakan untuk komunikasi dengan server Oracle, pada setiap *request* dari *user-process*.
- *Server-process* mencari informasi status untuk mengirimkan hasilnya kepada *user-process.*

Catatan : Dengan menggunakan konfigurasi multithreaded server (MTS), maka memungkinkan bagi beberapa user untuk menggunakan *server-process secara bersama.*

1.4. Oracle Instance

Server Oracle berisi Oracle Instance dan Oracle Database, dimana Oracel Instance berisi struktur memory yang disebut dengan *system-global-area* (SGA) dan *background-process* yang dipergunakan oleh server Oracle untuk mengatur database. Oracle Instance diindentifikasi dengan menggunakan setting pada sistem operasi sebagai ORACLE_SID, yang dapat dibuka dan dipergunakan pada hanya satu database saja pada setiap waktu.



1. System Global Area

Struktur memory dari Oracle Instance berada pada daerah memory yang disebut SGA, yang berisi data dan informasi pengontrol untuk server Oracle. SGA dialokasikan pada virtual memory komputer tempat server Oracle berada. SGA terdiri dari beberapa struktur memory yang meliputi :

- *Shared pool*

  Dipergunakan untuk menyimpan informasi seperti statement SQL yang baru saja dieksekusi dan data dari data dictionary yang baru saja dipergunakan.

- *Database buffer cache*

  Dipergunakan untuk menyimpan data yang baru saja dipergunakan.
- *Redo log buffer*

  Untuk meyimpan perubahan yang dibuat pada saat mengoperasikan database mnggunakan instance tersebut.

  Kegunaan dari struktur tersebut diatas akan didiskusikan secara detail pada bab berikutnya.

2. Background Processes

Background-processes yang berada didalam instance membentuk fungsi utama yang diperlukan untuk melayani request dari beberapa user yang aktif, tanpa memperngaruhi integritas dan unjuk kerja dari sistem secara keseluruhan. Setiap Oracle Instance akan menggunakan beberapa background-processes tergantung kepada konfigurasinya, akan tetapi setiap instance secara default akan terdiri dari lima buah beackground-processes yang meliputi :

- *Database Writer (DBWR)*

  Bertanggung jawab terhadap penulisan untuk melakukan perubahan data terhadap database.
- *Log Writer (LGWR)*

  Merekam perubahan yang terjadi dan merekamnya dalam redo log buffer dalam database.
- *System Monitor (SMON)*

  Memiliki fungsi utama sebagai pengecekan terhadap konsistensi maupun permulaan pembukaan database pada saat database dibuka.
- *Process Monitor (PMON)*

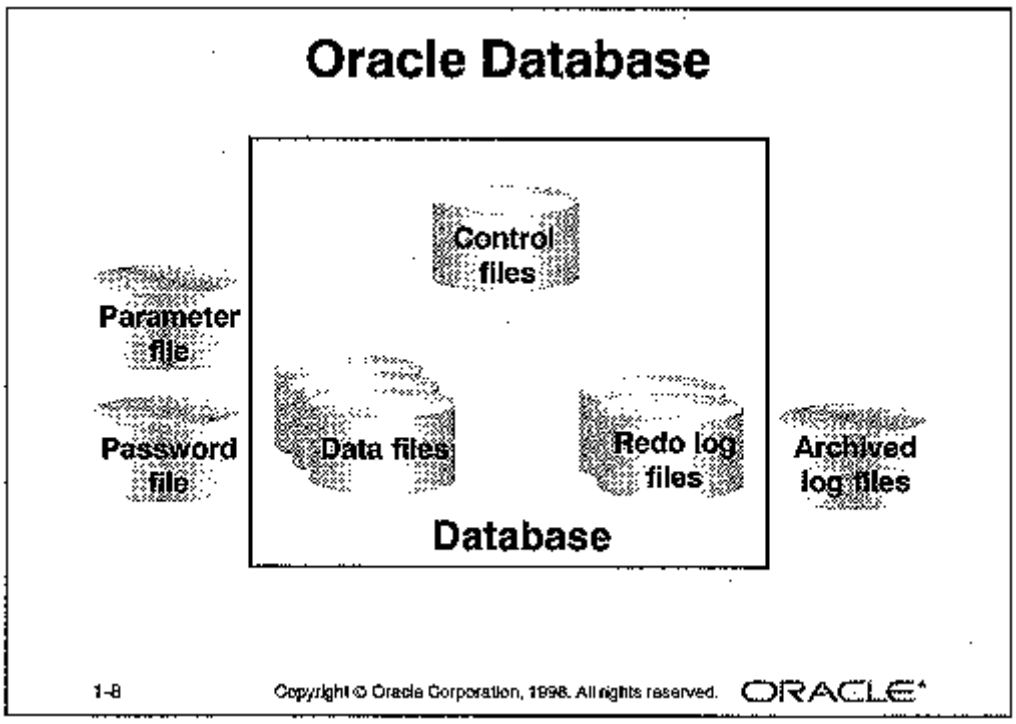
  Bertugas menutup (membersihkan) segala macam resource jika salah satu dari processes mengalami kegagalan.
- *Checkpoint Process (CKPT)*

  Bertanggung jawab untuk meng-update informasi status database jika terjadi perubahan pada buffer cache secara permanent tersimpan dalam database.

Kegunaan dari setiap processes tersebut akan didiskusikan secara detail pada bab berikutnya.

1.5. Oracle Database

Suatu database Oracle diidentifikasi menggunakan nama database (DB_NAME) yang mewakili struktur fisik dan dibentuk dari file sistem operasi. Walaupun memungkinkan untuk menggunakan nama database yang erbeda dengan nama instance, namun Oracle merekomendasikan untuk menggunakan nama yang sama agar mempermudah dalam langkah administrasi.



File database merupakan file-file dasar database yang berisi data tiap user dan informasi tambahan yang diperlukan untuk meyakinkan kebenaran operasi database yang dilakukan.

Database Oracle berisi beberapa tipe file berikut ini :

- *Data file*

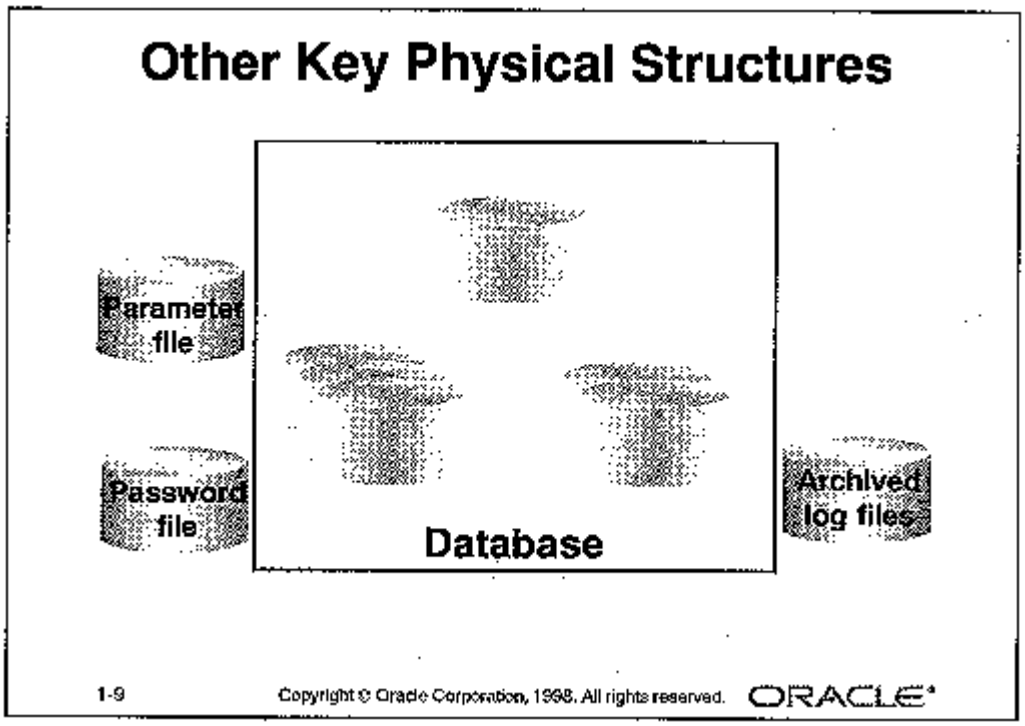
  Dipergunakan untuk menyimpan *data-dictionary*, obyek-obyek yang dimiliki user, serta *before-images* dari data yang dilakukan modifikasi oleh transaksi yang sedang berlangsung.

- *Redo log file*

  Berisi rekaman perubahan yang dilakukan pada database untuk meyakinkan jika terjasi rekonstruksi terhadap data jika mengalami kasus kegagalan. Database memerlukan sekurang-kurangnya dua redo log file.

- *Control file*

  Berisi informasi yang diperlukan untuk melakukan perawatan dan memriksa integritas database. Database memerlukan sekurang-kurangnya dua control file.



Disamping beberapa file database yang telah disebutkan diatas, maka Oracle juga menggunakan file-file yang lain, salah satunay adalah :

- *Parameter file*

  Dipergunakan untuk mendefinisikan karakteristik dari Oracle Instance.

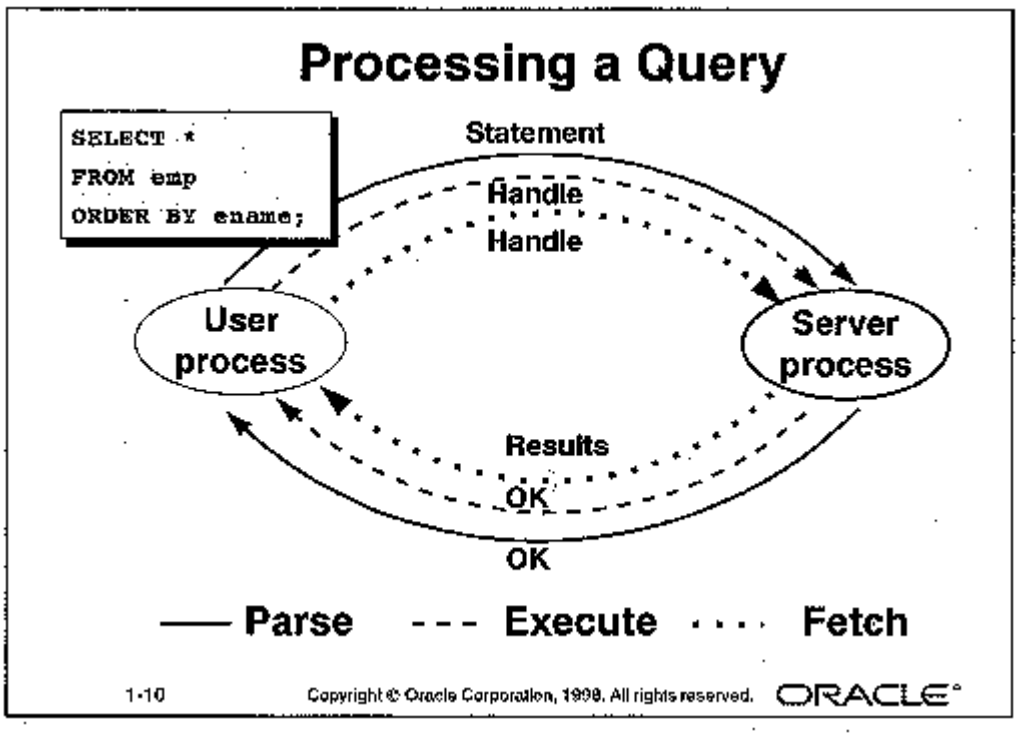
- *Password file*

  Dipergunakan untuk autentikasi privileged dari user yang akan menggunakan database.
- *Archived redo log file*

  Salinan dari file redo log yang dapat dipergunakan untuk keperluan recovery jika terjadi kegagalan media penyimpan.

1.6. Menjalankan Query

Beberapa langkah berikut ini adalah tahapan utama yang dilakukan server oracle dalam memproses Query yang diberikan oleh user.



Langkah-langkah tersebut yaitu :

- *Parse*

  Pada langkah ini *user-process* mengirimkan query kepada *server-process* beserta *request* untuk menguraikan kalimat query atau melakukan kompilasi terhadap query. *Server-process* akan melakukan pengecekan terhadap validasi perintah yang diberikan dan menggunakan daerah didalam SGA yang disebut *Shared-*

Pool untuk mengkompile perintah tersebut. Pada akhir tahap ini *server-process* akan memberikan status yang berisi informasi keberhasilan atau kegagalan dari fase *Parse* kepada *user-process.*

- *Execute*

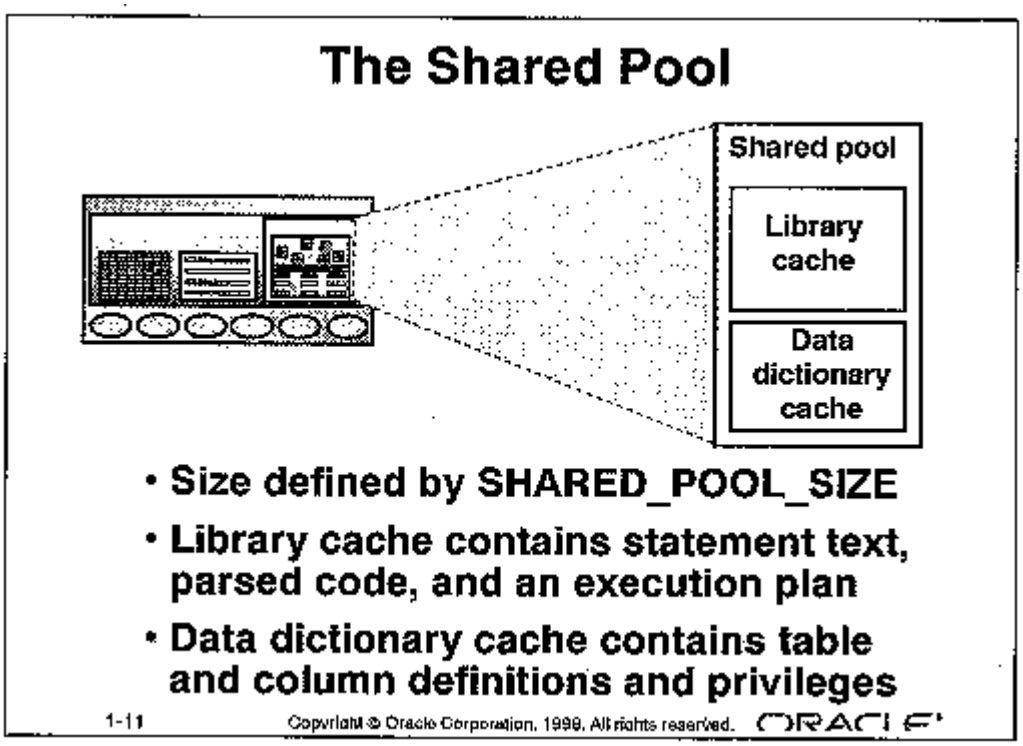
  Selama tahapan didalam memproses query ini maka *server-process* akan mencari dan menyiapkan untuk menampilkan data.

- *Fetch*

  Pada tahap ini baris-baris data yang dihasilkan oleh query dikirimkan dari *server* kepada *user*. Tergantung kepada jumlah memory yang dipergunakan untuk melakukan transfer, jika diperlukan akan dibutuhkan satu atau dua fetch untuk mentranfer hasil dari query yang diberikan oleh user.

1.7. Shared Pool

Shared Pool merupakan bagian dari SGA yang dipergunakan selama proses parse. Ukuran dari shared pool dapat dispesifikasikan dalam parameter inisialisasi SHARED_POOL_SIZE yang berada pada file parameter.



Beberapa komponen dari shared pool berikut ini dipergunakan dalam proses parse dari statement SQL yang diberikan oleh user.

1. Library Cache

Dipergunakan untuk meyimpan informasi tentang statement SQL yang saat ini diberikan, yang meliputi :

- Text dari statement SQL tersebut
- Parse tree, yang merupakan versi kompilasi dari statement SQL tersebut
- Execution plan, untuk mendefinisikan langkah-langkah yang akan dilakukan dalam menjalankan statement tersebut yang ditentukan oleh optimizer.

Pada saat library-cache menyimpan informasi ini, jika query dijalankan kembali sebelum execution plannya dihasilkan oleh beberapa statement yang lain, maka server-process tidak perlu untuk melakukan parse terhadap statement tersebut. Jadi library-cache membantu untuk meningkatkan performance dari aplikasi.

2. Data Dictionary Cache

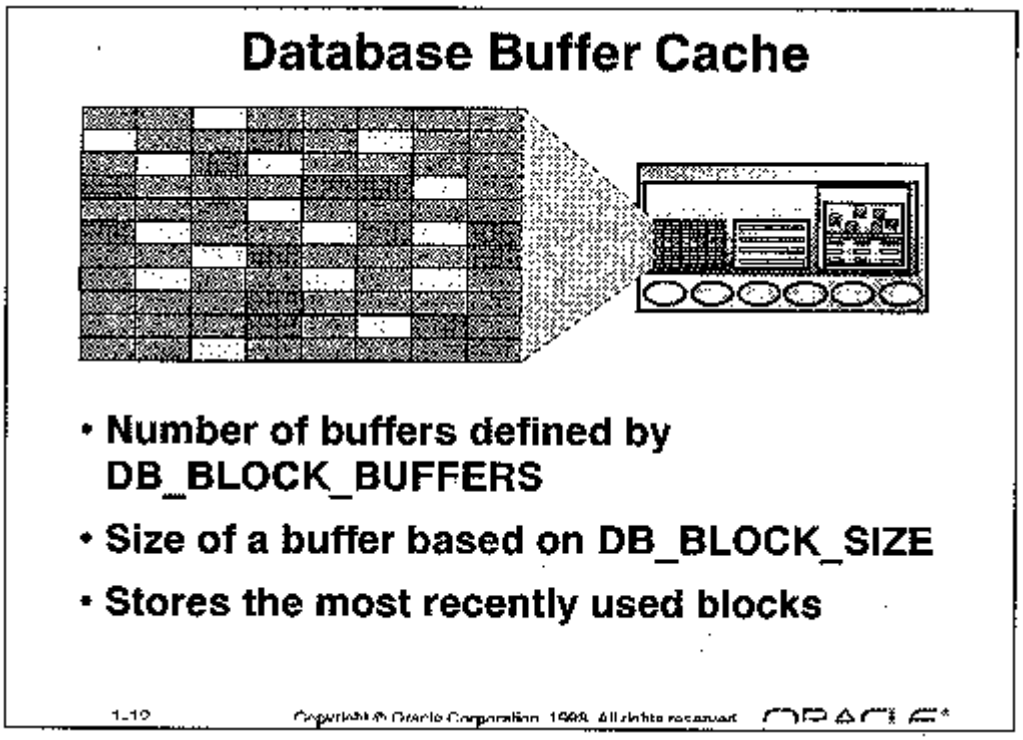
Data-dictionary-cache juga disebut dengan *dictionary-cache* atau *row-cache* adalah merupakan bagian dari share pool yang menyimpan informasi data dictionary yang baru saja dipergunakan oleh user, seperti contohnya definisi tabel dan kolom, nama user, password, serta privileges.

Pada saat fase *parse*, *server-process* melakukan penguncian bagi informasi didalam *dictionary-cache* untuk menemukan nama obyek yang ditentukan didalam statement SQL dan memvalidasi privilege akses yang dilakukan. Jika diperlukan *server-process* akan memulai untuk membaca informasi dari data file.

1.8. Database Buffer Cache

Pada saat query diproses, maka server-process akan melakukan penguncian bagi block-block yang diperlukan didalam *database-buffer-cache*. Jika block yang

dimaksud tersebut tidak ditemukan didalam *database-buffer-cache*, maka *server-process* akan membaca block yang dimaksud dari data file dan meletakkan duplikatnya didalam *buffer-cache*. Request berikutnya yang menggunakan block yang sama tinggal hanya menemukannya dalam memory dan tidak perlu lagi untuk melakukan pembacaan ke data file lagi.



Database-buffer-cache atau disingkat dengan *buffer-cache* adalah suatu daerah didalam SGA yang digunakan untuk menyimpan block-block data yang paling akhir digunakan.

Ukuran dari setiap buffer didalam *buffer-cache* sama dengan ukuran dari block data, yang dispesifikasi oleh parameter DB_BLOCK_SIZE. Jumlah buffer yang ditentukan sama dengan nilai parameter DB_BLOCK_BUFFERS.

Server Oracle menggunakan algoritma *least-recently-used* (LRU) untuk menyimpan sementara buffer yang belum diakses untuk membuat block baru dapat diakomodir didalam *buffer-cache*.

1.9. Program Global Area

Program-global-area (PGA) merupakan daerah di memory yang berisi data dan informasi pengontrol untuk server-process tunggal atau background-process tunggal. Dalam kenyataannya suatu SGA yang dipergunakan bersama dan dilakukan penulisan dan pembacaan oleh beberapa *process* yang disebut dengan PGA, dimana PGA merupakan area yang dipergunakan oleh satu *process* saja.

Pada saat menggunakan konfigurasi server Oracle secara *dedicated-server*, maka PGA berisi antara lain :

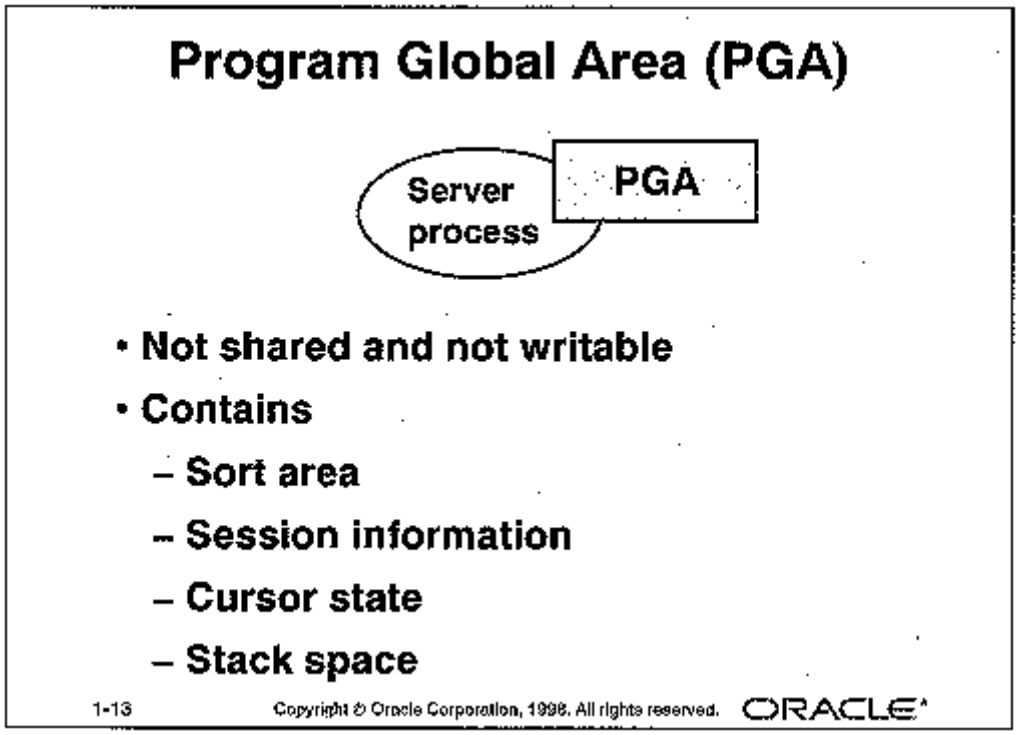
- *Sort area*

  Dipergunakan untuk keperluan melakukan pengurutan terhadap baris-baris sebelum diproses dan dikirimkan kembali kepada user.
- *Session information*

  Berisi informasi tentang privilege user yang berada pada session.
- *Cursor state*

  Untuk mengindikasikan tahapan didalam memproses berbagai cursor yang sedang dipergunakan didalam session.
- *Stack space*

  Suatu tempat untuk meletakkan variabel-variabel session.



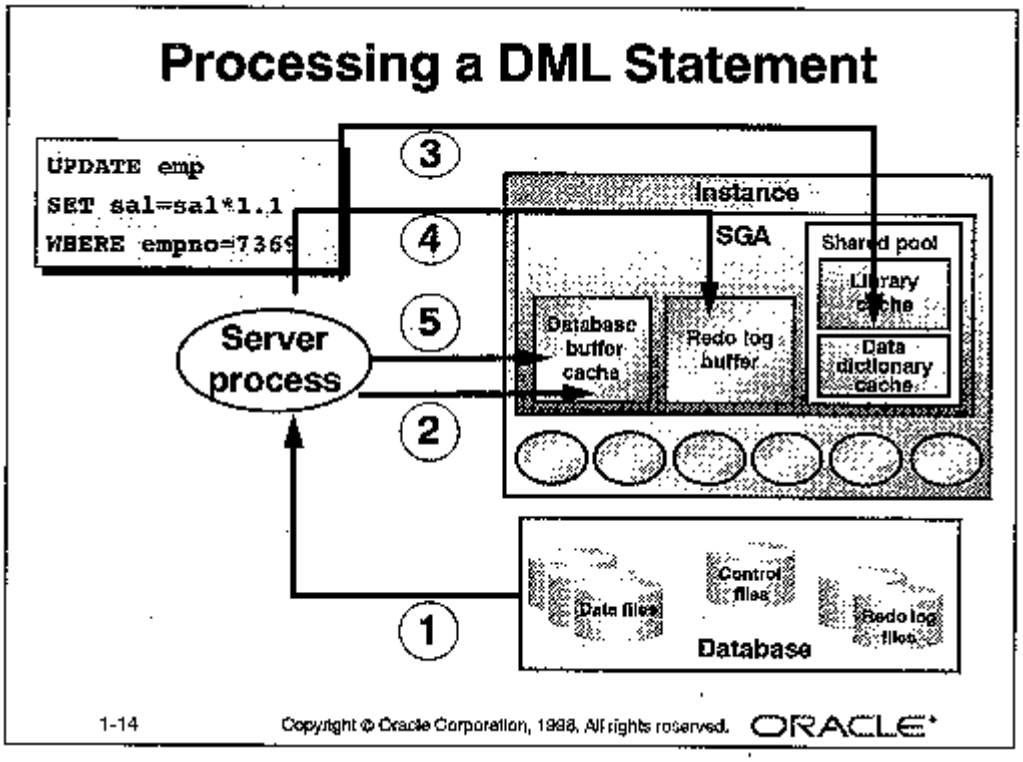
PGA dialokasikan pada saat *process* dibuat dan dihapus pada saat *process* dihentikan atau terjadi pemberhentian akibat kegagalan.

Catatan : Beberapa struktur yang telah disebutkan diatas akan tersimpan dalam SGA ketika sedang menggunakan konfigurasi MTS.

1.10. Menjalankan Statement DML

Statement *data-manipulation-language* (DML) memerlukan dia fasa proses, yaitu :

- *Parse*, yang mirip dengan fase parse pada saat memproses query.
- *Execute*



Untuk menjelaskan fase *Execute*, maka perhatikan contoh berikut yang menggambarkan user sedang mengeksekusi perintah UPDATE dengan bentuk sebagai berikut :

```
UPDATE emp SET sal=sal*1.1
WHERE empno=7369;
```

Pada uraian berikut merupakan urutan langkah yang dipergunakan dalam mengeksekusi statement UPDATE diatas, antara lain :

1. *Server-process* membaca data dan *rollback-block* dari data file, jika belum ada didalam *buffer-cache.*
2. Mengcopy block yang dibaca dan meletakkannya didalam *buffer-cache*.
3. *Server-process* meletakkan dan mengunci data.
4. *Server-process* merekam perubahan kedalam *rollback (before-image)* dan data (dengan nilai baru) didalam *redo-log-buffer*.
5. *Server-process* merekam *before-image* kedalam block *rollback* dan mengupdate block data, yang keduanya terletak didalam *database-buffer-cache*. Kedua perubahan block tadi didalam *buffer-cache* ditandai dengan *dirty-buffers* yang berarti *buffer* tersebut tidak sama jika dihubungkan dengan block yang berada pada disk.

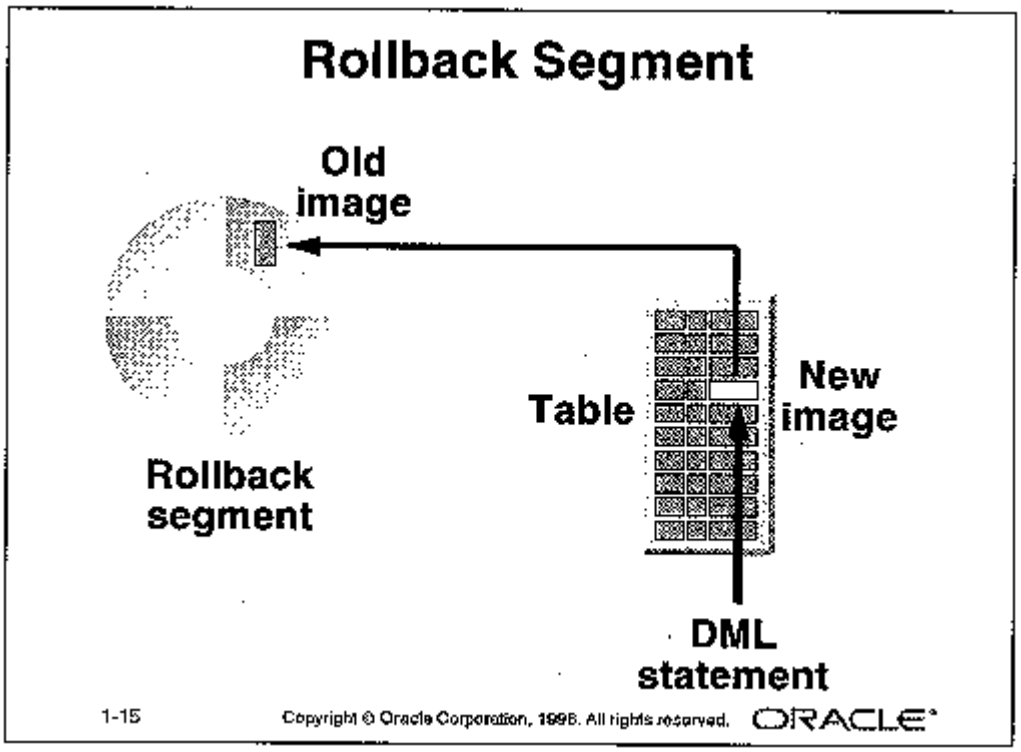
Catatan : Untuk memproses statement DELETE dan INSERT digunakan langkah yang sama. *Before-image* untuk statement DELETE berisi nilai colom pada baris yang dihapus, dan pada INSERT hanya berisi informasi lokasi baris yang akan disimpan dalam *rollback.*

1.11. Rollback Segment

Sebelum membuat perubahan suatu nilai, maka *server-process* akan menyimpan nilai yang lama kedalam *rollback-segment,* hal ini untuk keperluan :

- Membatalkan *(undo)* perubahan yang dilakukan jika transaksi tersebut hendak dikembalikan ke semula *(roller-back).*
- Untuk meyakinkan bahwa transaksi lain tidak melihat perubahan tersebut belum di-commit oleh statement DML *(read consistency).*

- Untuk menemukan kembali database pada kondisi semula jika terjadi kasus kegagalan.



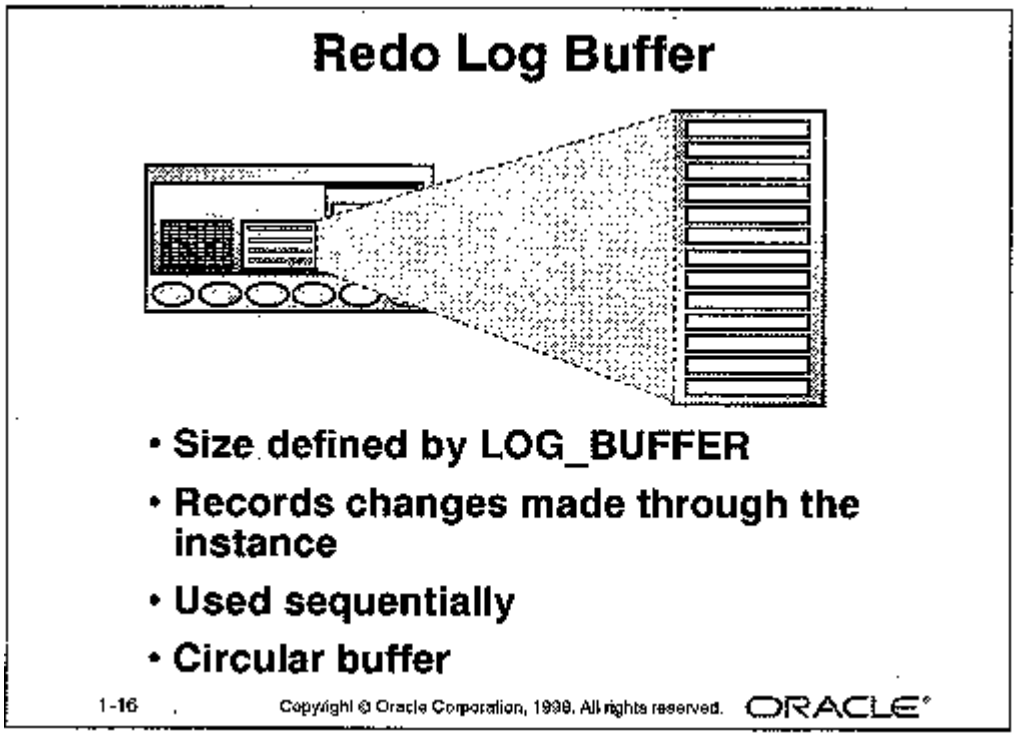
Rollback-segment, seperti halnya tabel dan index berada pada data file dan beberapa bagian darinya dipinjam oleh *database-buffer-cache* jika suatu saat diperlukan.

1.12. Redo Log Buffer

Server-process merekam perubahan yang dibuat oleh instance didalam *redo-log-buffer*, yang merupakan bagian dari SGA. *Redo-log-buffer* ini memiliki beberpaa karakteristik antara lain :

- Ukurannya dalam byte da didefinisi oleh parameter LOG_BUFFER.
- Meyimpan informasi *redo-record*, jika suatu record berubah maka block tersebut akan berubah, diikuti dengan perubahan lokasi serta nilai yang baru. *Redo* menyimpan perubahan, tetapi tidak ada perbedaan type block yang berubah, sehingga tidak dapat dibedakan antara merubah block data dari merubah index atau *rollback-block.*

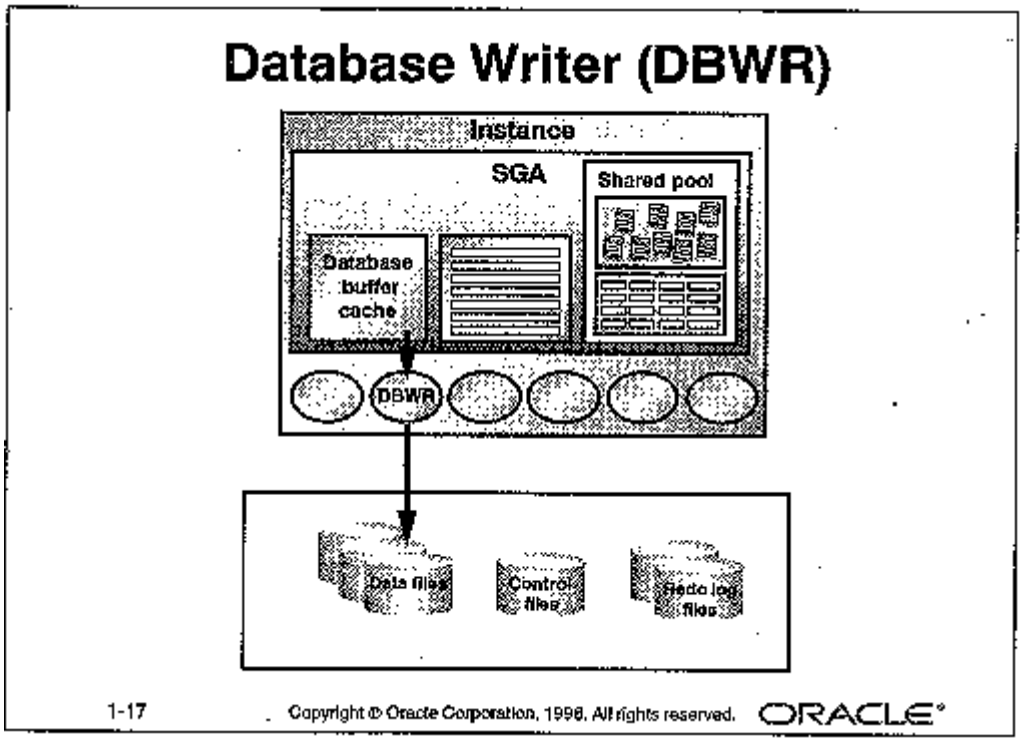
- *Redo-log-buffer* dipergunakan secar berurutan, dan perubahannya diakibatkan oleh satu transaksi, yang dapat berasal dari transaksi lain yang saling bersisipan.
- Merupakan buffer secara sircular yang berarti tempatnya akan digunakan kembali jika telah penuh, tetapi hanya sesudah semua isian *redo* yang lama telah terekam dalam *redo-log-files*.



Catatan : *Redo-log-files* akan dibahan secara lebih detail pada bab berikutnya.

1.13. Database Writer (DBWR)

Server-process merekam perubahan pada *rollback* dan block data pada *buffer-cache*. *Database-writer* (DBWM) menulis *dirty-buffer* dari *database-buffer-cache* kedalam data files. Hal ini untuk meyakinkan jumlah dari *free-buffers* mencukupi agar buffer tersebut dapat ditulisi ulang ketika *server-process* memerlukannya untuk menulis block kedalam *database-buffer-cache* dari data files.



Unjuk kerja database dapat ditingkatkan karena *server-process* hanya membuat perubahan pada *buffer-cache*, dan DBWR secara terpisah menulis kedalam data files sampai ditemukan keadaan sebagai berikut :

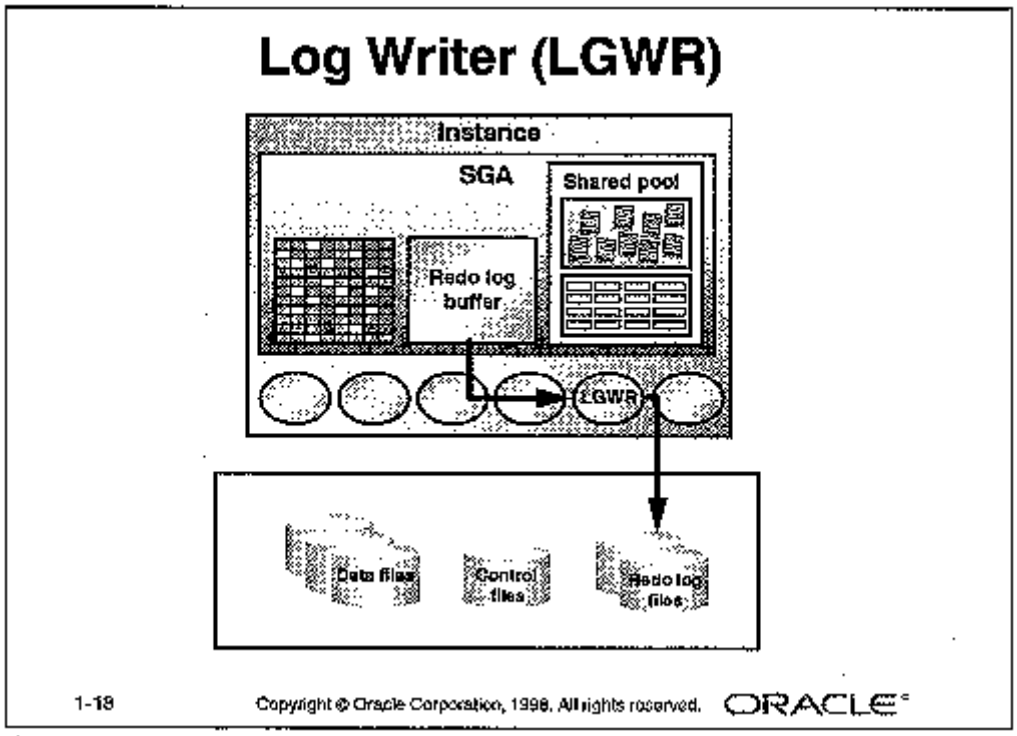
- Jumlah dari *dirty-buffer* mencapai nilai puncak.
- Process memeriksa secara spesifik jumlah block pada saat scanning untuk buffer yang kosong dan tidak menemukan apapun.
- Terjadi timeout
- Checkpoint DBWR dapat di-*trigger* oleh beberapa event seperti menutup database (checkpoint merupakan sinkronisasi antara *database-buffer-cache* dan data files).

1.14. Log Writer (LGWR)

Log-writer (LGWR) merupakan background-process yang bertugas untuk menulis isisan dari *redo-log-buffer* kedalam *redo-log files*.

LGWR melakukan penulisan yang berurutan kedalam *redo-log files* jika menemukan kondisi sebagai berikut ini:

- Jika *redo-log-buffer* dalam keadaan sepertiga penuh.
- Jika terjadi timeout (setiap tiga detik)
- Sebelum DBWR melakukan penulisan untuk memodifikasi block dalam *database-buffer-cache* pada data files.
- Jika transaksi di-commit.



1.15. COMMIT Processing

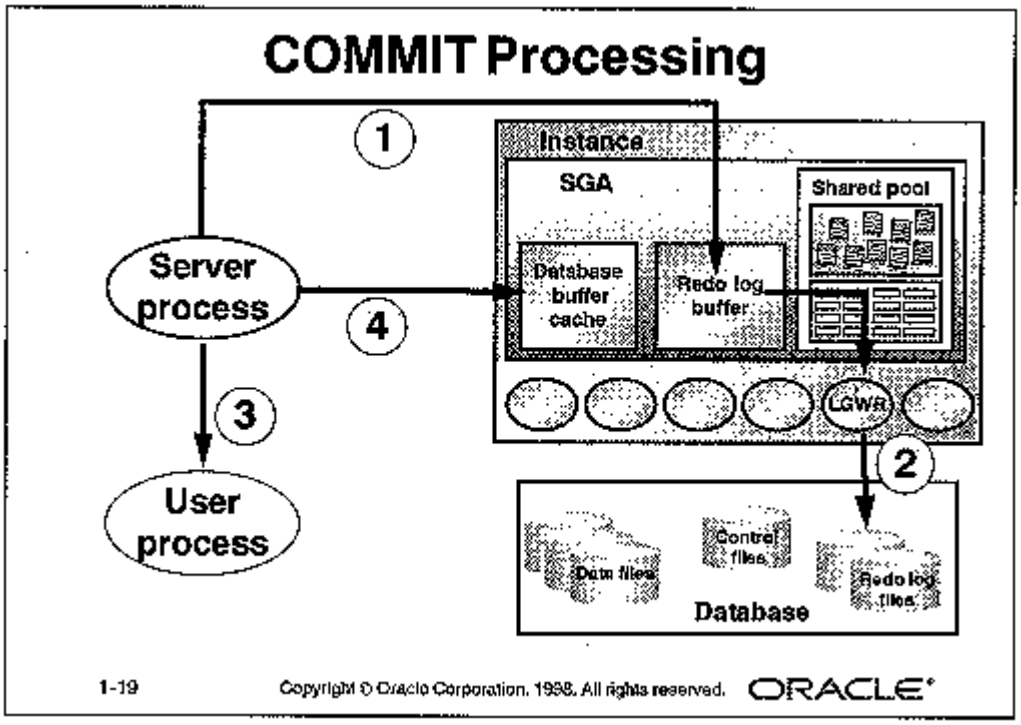
Oracle menggunakan mekanisme commit secara cepat yang dapat menjamin perubahan yang di-commit dapat kembalikan lagi keasalnya jika terjadi kesalahan.

1. System Change Number

Jika terjadi transaksi commit, maka server Oracle akan meminta *system-change-number* (SCN) untuk melakukan transaksi, yang akan menambah sendiri secara monoton dan merupakan penomoran tersendiri didalam database. SCN dipergunakan oleh server Oracle sebagai nomer stempel secara internaluntuk

mensinkronkan data dan untuk memperlengkapi sistem pembacaan secara konsisten jika data diambil dari data files.

Dengan menggunakan SCn maka server Oracle dapat membentuk sistem pengecekan secara konsisten tanpa tergantung kepada data tanggal dan waktu dari sistem operasi yang digunakan.



2. Tahapan Commit

Jika diberikan perintah commit, maka akan dilakukan langkah-langkah berikut ini :

Server-process akan meletakkan perekaman commit yang panjangnya sama dengan SCN didalam *redo-log-buffer*.

LGWR akan membentuk sistem penulisan secara berdampingan semua masukan pada *redo-log-buffer* hingga termasuk perekaman commit pada *redo-log-buffer*. Sesudah pada tahap ini maka server Oracle dapat menjamin bahwa perubahan yang telah dilakukan tidak akan hilang jika terjadi kegagalan sistem.

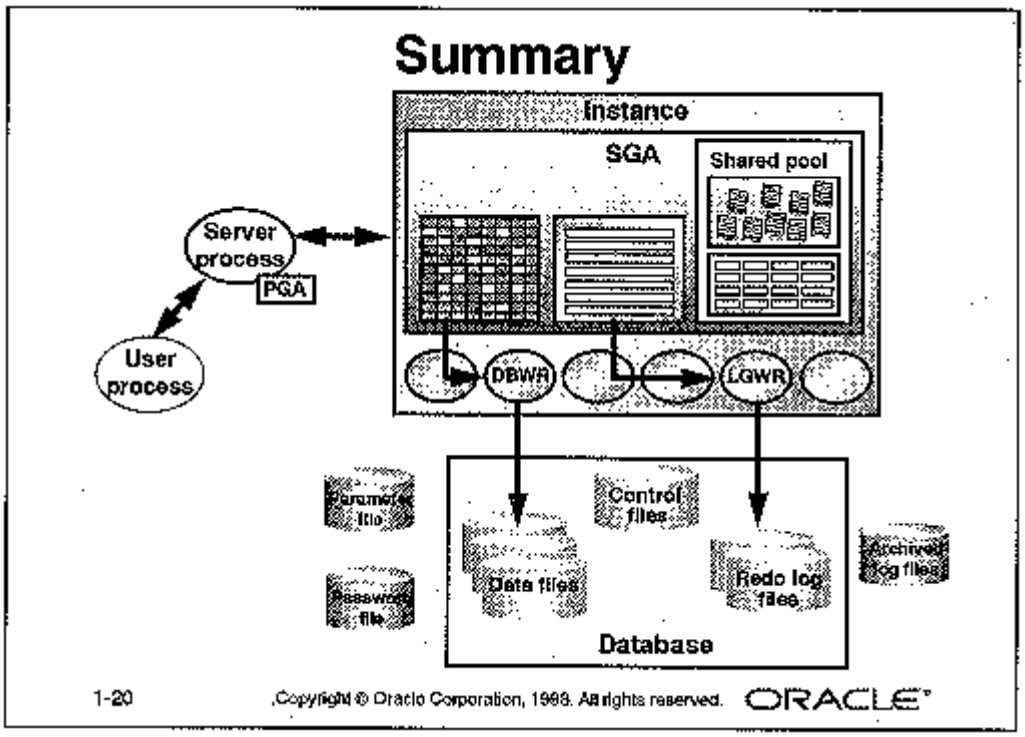
User akan diberikan informasi bahwa proses commit telah selesai.

Server-process akan merekam informasi untuk mengindikasikan bahwa transaksi telah selesai dengan lengkap dan penguncian terhadap *resource* dapat dibuka kembali.

Catatan : transaksi rolling-back tidak men-trigger LGWR untuk menulis pada disk. Server Oracle selalu melakukan rollback perubahan yang tidak di-commit ketika dilakukan proses penemuan data kembali (recovering) karena adanya duatu kegagalan. Jika terjadi kegagalan sesudah proses rollback, maka masukan sebelum rollback akan direkam pada disk, tidak adanya commit yang diberikan sudah cukup untuk meyakinkan bahwa perubahan data yang dilakukan oleh transaksi akan di rollback.

Pada beberapa langkah berikut ini akan memberikan keuntungan jika dilakukan commit, antara lain :

- Penulisan yang berurutan kepada log-files lebih cepat dari pada proses penulisan pada block berbeda didalam data file.
- Hanya ada informasi kecil yang perlu direkam dan ditulis pada log files, dan jika dilakukan penulisan kedalam data files akan memerlukan sejumlah block data yang ditulis.
- Commit pada database akan menyimpan sementara perekaman redo-log dari beebrapa transaksi yang dilakukan menjadi commit yang bersamaan dalam satu kali penulisan.
- Tidak adanya *redo-log-buffer* yang akan penuh, hanya satu penulisan yang secara sinkron diperlukan pada tiap transaksi.
- Ukuran transaksi tidak akan mempengaruhi jumlah waktu yang diperlukan untuk operasi commit yang sebenarnya.


Summary
Instance
SGA
Shared pool
Server process
PGA
User process
DBWR
LGWR
Control files
Data files
Redo log files
Database
1-20
Copyright © Oracle Corporation, 1998. All rights reserved.
ORACLE

Referensi Perintah

Kontek	Referensi
Processes	User process Server process Background process - DBWR - LGWR
Memory structures	SGA - Share pool - Database buffer cache - Redo log buffer PGA
Physical structures (files) Physical structures (files)	Database files - Data files - Control files - Redo log files Other files - Parameter files - Password file - Archieved redo log files
Parameters	SHARE_POOL_SIZE DB_BLOCK_SIZE DB_BLOCK_BUFFERS LOG_BUFFER