



# MENJADI AHLI ADMINISTRATOR DATABASE **ORACLE**



# **Menjadi Ahli**

# **Database Administrator Oracle**

**Radiant Victor Imbar**

Penerbit Garudhawaca

# **Menjadi Ahli Database Administrator Oracle**

**Oleh** : Radian Victor Imbar  
Hak Cipta ©2017 pada Penulis

**ISBN** : 978-602-6581-11-2

**Editor** : Bernard Renaldy S.

**Desain Cover** : Gisela Kurniawati

**Penerbit** : Garudhawaca  
Jl. Gabus Raya I No.14, Minomartani, Sleman,  
Daerah Istimewa Yogyakarta 55581, Indonesia.

**Pemasaran** : Paragon Book - myparagonbook@gmail.com

Cetakan Pertama, Agustus 2017

*Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronis maupun mekanis, memfotocopy, merekam, atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis.*

---

---

## **Katalog Dalam Terbitan (KDT)**

**Imbar, Radian V.**

Menjadi Ahli Administrator Database Oracle / Penyusun Radian  
Victor Imbar.--Ed.1, Cet.1--Yogyakarta: Garudhawaca, Agustus 2017.

xiv, 141 hlm ; Uk: 14,8x20 cm

**ISBN 978-602-6581-11-2**

1. Basisdata

I. Judul

005.74

---

---

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Pengasih atas limpahan rahmatNya sehingga Buku Menjadi Ahli Database Administrator Oracle dapat diselesaikan dengan baik.

Dalam Buku ini dijelaskan secara lengkap bagaimana melakukan administrasi basis data Oracle dimulai dari instalasi Oracle, membuat *Database Oracle*, mengatur *Database Oracle* dan melakukan *backup* serta *recovery Database Oracle*.

Dalam Pembuatan buku ini penulis mengucapkan terima kasih kepada Universitas Kristen Maranatha, rekan – rekan dosen di Fakultas Teknologi Informasi UK. Maranatha dan semua pihak yang membantu sehingga Buku ini dapat diselesaikan.

Akhirnya penulis mohon maaf yang tulus dari pembaca atas segala kekurangan yang ada didalam buku ini. Besar harapan penulis agar Buku ini dapat bermanfaat bagi seluruh masyarakat luas.

Bandung,

Radiant Victor Imbar



# DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>iii</b>
<b>DAFTAR ISI .....</b>	<b>v</b>
<b>DAFTAR GAMBAR .....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>xiii</b>
<b>BAB I MEMBUAT DATABASE ORACLE.....</b>	<b>1</b>
1.1. Tujuan .....	1
1.2. Arsitektur Database.....	1
1.3. Menjelajah Struktur Penyimpanan.....	3
1.4. Control File .....	3
1.5. Redo Log File.....	4
1.6. Tablespace dan Datafile .....	5
1.7. Segment, Extent, dan Block.....	6
1.8. Pengaturan Perangkat Oracle .....	8
1.9. Struktur Memori Oracle .....	9
1.10. Proses – proses Oracle .....	11
1.11. Kamus Data.....	13
1.12. Database Control.....	14
1.13. Grid Control .....	15
1.14. Database Configuration Assistant (DBCA) ....	16
1.15. Membuat Database.....	17
1.16. Identifikasi Database.....	18
1.17. Management Options .....	19
1.18. Kata Kunci dan Penyimpanan.....	21

1.19. Pilihan Lokasi File dan Recovery Backup .....	22
1.20. File Location Variables .....	24
1.21. Parameter Isi dan Inisialisasi.....	24
1.22. Database Storage .....	26
1.22.1. Pilihan Creation dan Create .....	27
1.22.2. Aksi Lain dengan DBCA .....	28
<b>BAB II MENGENDALIKAN DATABASE .....</b>	<b>31</b>
2.1. Tujuan .....	31
2.2. Memulai dan Mengakhiri iSQL*Plus .....	31
2.3. Kerangka Pengaturan .....	32
2.4. Memulai dan Mengakhiri Database Control...	33
2.5. Mengakses Database Control .....	34
2.6. SYSOPER dan SYSDBA.....	36
2.7. Database Home Page .....	36
2.8. Mengubah Status Listener.....	37
2.9. STARTUP dan SHUTDOWN .....	39
2.10. Memulai Sebuah Database.....	39
2.11. Mematikan Database.....	43
2.12. File Parameter Inisialisasi .....	47
2.13. Menampilkan Parameter Inisialisasi .....	48
2.14. Menampilkan Alert Log .....	49
<b>BAB III STRUKTUR PENYIMPANAN.....</b>	<b>51</b>
3.1. Tujuan .....	51
3.2. Tablespace dan File Data .....	51
3.3. Pengaturan Ruangan dalam Tablespace.....	52
3.4. Membuat Tablespace Baru.....	53
3.5. Penyimpanan Pengaturan Tablespace Lokal...	56
3.6. Tablespace dalam Prekonfigurasi Database....	59
3.7. Mengubah Tablespace.....	61
3.8. Aksi dengan Tablespace.....	64
3.9. Menghilangkan Tablespace.....	67
3.10. Menampilkan Informasi Tablespace .....	68

<b>BAB IV ADMINISTRASI PENGGUNA .....</b>	<b>71</b>
4.1. Tujuan .....	71
4.2. Account Pengguna Database.....	71
4.3. Menciptakan Pengguna .....	72
4.4. Profil dan Pengguna .....	73
4.5. Pengguna Otentik .....	76
4.6. Tablespace dan Penguncian Default dan Tidak Tetap .....	79
4.7. Pengguna Database dan Skema.....	80
4.8. Penanda untuk Pengguna .....	80
4.9. Memberikan Kuota untuk Pengguna.....	85
4.10. Memberikan Roles untuk Pengguna .....	86
4.11. Roles.....	87
4.12. Keuntungan dari Roles.....	88
4.13. Pendefinisian Awal Roles .....	88
4.14. Mengamankan Roles.....	90
<b>BAB V MENGATUR OBYEK SKEMA .....</b>	<b>93</b>
5.1. Tujuan .....	93
5.2. Apa Itu Skema?.....	93
5.3. Mengakses Obyek Skema .....	96
5.4. Menamakan Obyek Database.....	97
5.5. Ruang Nama Obyek Skema .....	98
5.6. Menspesifikasi Tipe Data dalam Tabel.....	99
5.7. Tipe Data Lain .....	100
5.8. Membuat dan Memodifikasi Tabel .....	103
5.9. Memahami Kesatuan Data .....	105
5.10. Mendefinisikan Konstrain.....	107
5.11. Menampilkan Atribut dari Sebuah Tabel.....	108
5.12. Menampilkan Isi dari Sebuah Tabel .....	109
5.13. Aksi dengan Tabel .....	110
5.14. Membuat Indeks.....	113

<b>BAB VI BACKUP DATABASE .....</b>	<b>115</b>
6.1. Tujuan .....	115
6.2. Terminologi.....	115
6.3. Recovery Manager (RMAN) .....	117
6.4. Menkonfigurasi Pengaturan Backup .....	118
6.5. Menjadwalkan Backup: Strategy .....	121
6.6. Menjadwalkan Backup: Option .....	122
6.7. Menjadwalkan Backup: Settings.....	124
6.8. Menjadwalkan Backup: Schedule .....	124
6.9. Menjadwalkan Backup: Review .....	125
6.10. Control File Backup to Trace.....	126
6.11. Manage Current Backups.....	127
6.12. Flash Recovery Area .....	129
<b>BAB VII RECOVERY DATABASE .....</b>	<b>131</b>
7.1. Tujuan .....	131
7.2. Membuka Sebuah Database .....	131
7.3. Mengubah Status Perangkat.....	133
7.4. Menjaga Agar Sebuah Database Tetap Terbuka .....	134
7.5. Kehilangan Sebuah Control File .....	135
7.6. Kehilangan Sebuah Redo Log File .....	136
7.7. Kehilangan Sebuah File Data pada Mode NOARCHIVELOG .....	137
7.8. Kehilangan Sebuah File Data Noncritical pada Mode ARCHIVELOG.....	138
7.9. Kehilangan Sebuah File Data SYSTEM- CRITICAL pada Mode ARCHIVELOG .....	139
<b>DAFTAR PUSTAKA .....</b>	<b>141</b>

# Daftar Gambar

Gambar 1.	Arsitektur Database .....	2
Gambar 2.	Oracle Enterprise Manager .....	3
Gambar 3.	Redolog .....	5
Gambar 4.	Tablespace.....	6
Gambar 5.	Struktur Penyimpanan Logik Oracle.....	7
Gambar 6.	Struktur Oracle .....	8
Gambar 7.	Struktur Memori Oracle .....	11
Gambar 8.	Struktur Background Proses Oracle .....	12
Gambar 9.	Contoh Query .....	14
Gambar 10.	Enterprise Manager Database Control .....	15
Gambar 11.	Grid Control .....	16
Gambar 12.	Instalasi Oracle (1) .....	17
Gambar 13.	Instalasi Oracle (2) .....	18
Gambar 14.	Instalasi Oracle (3) .....	19
Gambar 15.	Instalasi Oracle (4) .....	20
Gambar 16.	Instalasi Oracle (5) .....	22
Gambar 17.	Instalasi Oracle (6) .....	23
Gambar 18.	Instalasi Oracle (7) .....	24
Gambar 19.	Instalasi Oracle (8) .....	26
Gambar 20.	Instalasi Oracle (9) .....	27
Gambar 21.	Instalasi Oracle (10) .....	28
Gambar 22.	Instalasi Oracle (11) .....	29
Gambar 23.	ISQLplus .....	32
Gambar 24.	Listener .....	33
Gambar 25.	Database Control .....	34
Gambar 26.	Login Enterprise Manager (1) .....	35
Gambar 27.	Login Enterprise Manager (1) .....	36
Gambar 28.	Database Homepage.....	37

Gambar 29. Status Listener.....	38
Gambar 30. Startup Database .....	39
Gambar 31. Status Database .....	40
Gambar 32. Status Database (2) .....	41
Gambar 33. Status Database (3) .....	42
Gambar 34. Shutdown Database.....	43
Gambar 35. Status Database Shutdown .....	46
Gambar 36. Status Database Startup.....	46
Gambar 37. Parameter Inisiasi .....	48
Gambar 38. Parameter Inisiasi (2) .....	49
Gambar 39. Alert Log .....	50
Gambar 40. Datafile.....	52
Gambar 41. Extent Managemet .....	53
Gambar 42. Pembuatan tablespace (1) .....	55
Gambar 43. Pembuatan Tablespace (2) .....	59
Gambar 44. Pembuatan Tablespace (3) .....	61
Gambar 45. Pembuatan Tablespace (4) .....	64
Gambar 46. Pembuatan Tablespace (5) .....	66
Gambar 47. Pembuatan Tablespace(6) .....	68
Gambar 48. Pembuatan Tablespace(7) .....	69
Gambar 49. Pembuatan User (1) .....	73
Gambar 50. Pembuatan Profile .....	76
Gambar 51. Autentikasi User.....	78
Gambar 52. Pilihan Tablespace untuk user.....	79
Gambar 53. Hak Akses User.....	83
Gambar 54. Memberikan Hak Akses Objek .....	84
Gambar 55. Memberikan Kuota .....	86
Gambar 56. Memodifikasi Roles .....	87
Gambar 57. Role .....	91
Gambar 58. Create Role.....	91
Gambar 59. Objek Schema .....	95
Gambar 60. Pembuatan Objek Schema .....	96
Gambar 61. Create table .....	104
Gambar 62. Struktur Tabel user HR .....	107
Gambar 63. Membuat Konstrain .....	108
Gambar 64. Atribut Tabel .....	109
Gambar 65. Isi Tabel .....	110

Gambar 66. Contoh aksi terhadap tabel .....	113
Gambar 67. Create Index .....	114
Gambar 68. Backup (1) .....	121
Gambar 69. Backup (2) .....	122
Gambar 70. Backup (3) .....	123
Gambar 71. Backup (4) .....	124
Gambar 72. Backup (5) .....	125
Gambar 73. Backup menggunakan RMAN Script .....	126
Gambar 74. Backup Control File .....	127
Gambar 75. Manage Backup .....	129
Gambar 76. Manage Flash Recovery Area .....	130
Gambar 77. Konfirmasi Shutdown/Startup .....	134
Gambar 78. Archive Log .....	137
Gambar 79. Recovery Datafile .....	139



## **Daftar Tabel**

Tabel 1. Tabel Hak Akses ..... 89



# BAB I

## MEMBUAT DATABASE ORACLE

### 1.1. Tujuan

Setelah menyelesaikan pelajaran ini, Anda diharapkan dapat melakukan hal – hal sebagai berikut:

- ✓ Menjelaskan arsitektur database Oracle.
- ✓ Mengerti arsitektur contoh.
- ✓ Menggunakan kerangka kerja manajemen.
- ✓ Menggunakan DBCA untuk:
  - Membuat sebuah database
  - Mengkonfigurasi sebuah database
  - Menurunkan sebuah database
  - Mengatur templet

### 1.2. Arsitektur Database

File – file yang disusun oleh database Oracle diatur sebagai berikut:

- ✓ Control files:

Terdiri atas metadata, yaitu data tentang database itu sendiri. File ini penting bagi database. Tanpa bagian tersebut, Anda tidak dapat membuka file – file data untuk mengakses data di dalam database.
- ✓ Data files:

Terdiri atas data dari database.
- ✓ Online redo log files:

Memungkinkan recovery instan database. Jika database bermasalah dan tidak kehilangan file – file datanya, recovery instan ini dapat menampilkan ulang database dengan informasi yang ada di dalamnya.

Ada beberapa file yang bukan merupakan bagian tetap dari database tetapi penting untuk kelancaran menjalankan database, yaitu:

- ✓ File parameter:

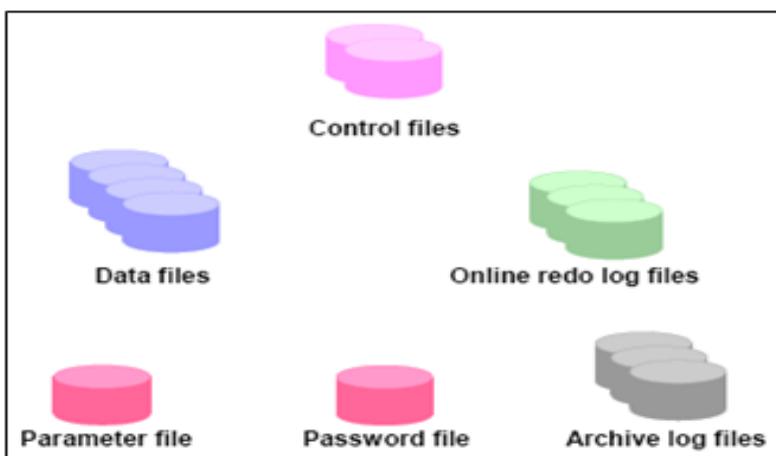
Digunakan untuk menjelaskan bagaimana contoh akan tersusun ketika Oracle dijalankan.

- ✓ File password:

Memungkinkan pengguna untuk masuk dalam database dan menjalankan tugas – tugas administrasinya.

- ✓ File archive log:

Terdiri atas sejarah pengulangan yang terjadi. File ini memungkinkan penggunaan recovery database. Dengan menggunakan file ini dan sebuah backup database, maka Anda dapat menampilkan file data yang hilang.



Gambar 1. Arsitektur Database

### 1.3. Menjelajah Struktur Penyimpanan

Struktur data logika disimpan di dalam file – file fisik database. Anda dapat dengan mudah menampilkan struktur logika dari database Anda melalui **Enterprise Manager**. Informasi selengkapnya dari tiap struktur dapat diperoleh dengan cara mengklik link – link yang ada dalam daerah **Storage** pada halaman **Administration**.



Gambar 2. Oracle Enterprise Manager

### 1.4. Control File

Ketika Anda mulai memasuki database, maka control file akan terbaca. Entri – entri yang ada dalam control file menspesifikasikan file – file fisik dalam database.

Ketika Anda menambahkan file pada database, control file secara otomatis akan ter-update. Adapun lokasi dari control file terspesifikasi dalam sebuah parameter inisialisasi.

Untuk melindungi kegagalan database akibat kehilangan control file, maka Anda harus menggandakan control file pada sedikitnya tiga peralatan fisik yang berbeda. Dengan menspesifikasi file – file ganda melalui parameter inisialisasi maka Anda memungkinkan server database Oracle untuk memperoleh penggandaan dari control file.

Anda dapat mengakses informasi tentang control file dalam database Anda dengan cara mengklik link **Control Files** pada daerah **Storage** pada halaman **Administration** dalam **Enterprise Manager**. Halaman **Control Files General** menampilkan nama dan lokasi dari control file dalam database Anda. Halaman **Advanced** menyediakan informasi tentang pembuatan control file dan identifikasi database. Halaman **Record Section** menampilkan informasi tentang entri – entri di dalam control file. Intinya adalah:

- ✓ Control file terdiri atas informasi struktur fisik database.
- ✓ Lakukan penggandaan control file untuk menghindari kegagalan database.
- ✓ Control file dibutuhkan ketika mulai memasuki database.

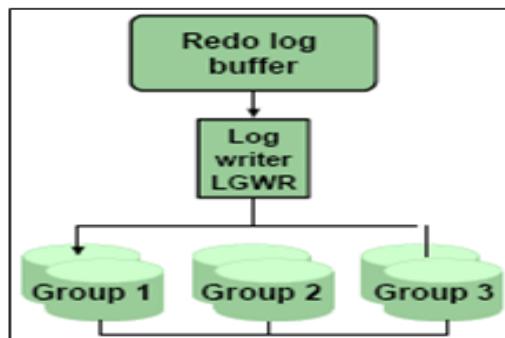
## 1.5. Redo Log File

Redo log file digunakan untuk menyimpan perubahan – perubahan pada database sebagai suatu hasil transaksi dan kegiatan internal server database Oracle. Bagian tersebut melindungi database dari kegagalan akibat kerusakan sistem karena mati listrik, kerusakan piringan, dan lain-lain. Redo log file harus digandakan untuk memastikan bahwa informasi yang tersimpan di dalamnya tidak akan hilang akibat kegagalan piringan.

Redo log file terdiri atas sekumpulan file redo log. Sebuah grup terdiri atas sebuah redo log file dan penggandaannya. Setiap penggandaan akan menjadi anggota grup dan setiap grup teridentifikasi oleh sebuah nomor. Proses penulisan log (LGWR) menulis rekaman – rekaman redo dari buffer redo log ke sebuah grup redo log sampai file terisi atau sebuah operasi pemindahan log diminta. Kemudian akan berpindah dan tertulis pada file di grup selanjutnya. Grup – grup redo log digunakan dalam sebuah siklus.

Anda dapat mengakses informasi tentang redo log file pada database Anda dengan cara mengklik link **Redo Log Groups** pada daerah **Storage** pada halaman **Administration**. Anda dapat menampilkan informasi selengkapnya, termasuk nama – nama dari redo log file, dengan cara menyeleksi sebuah grup dan mengklik **View**. Intinya adalah:

- ✓ Redo log file menyimpan perubahan database.
- ✓ Lakukan penggandaan redo log file untuk menghindari kegagalan.



Gambar 3. Redolog

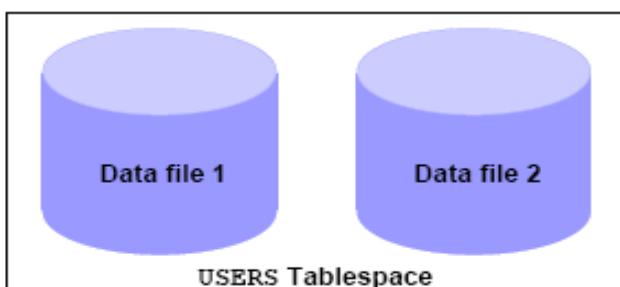
## 1.6. Tablespace dan Datafile

Sebuah database dibagi ke dalam unit – unit penyimpanan yang dinamakan tablespace, yang dapat digunakan oleh grup

yang berhubungan dengan struktur – struktur logika. Setiap database logikanya dibagi ke dalam satu atau lebih tablespace. Satu atau lebih file data secara jelas dibuat untuk setiap tablespace ke penyimpanan fisik data dari semua struktur logika dalam sebuah tablespace.

Anda juga dapat membuat tablespace *bigfile* dimana terdapat tablespace tunggal tetapi kapasitas file datanya besar (lebih dari 4G blok). Tradisional tablespace *smallfile* (untuk defaultnya) dapat memuat file – file data yang banyak, tetapi file – file tersebut tidak bisa besar kapasitasnya. Untuk informasi selanjutnya tentang tablespace *bigfile*, Anda dapat melihat pada *Database Administrator's Guide*. Intinya adalah:

- ✓ Tablespace terdiri dari satu atau lebih file data.
- ✓ File data termasuk dalam satu tablespace.



Gambar 4. Tablespace

### 1.7. Segment, Extent, dan Block

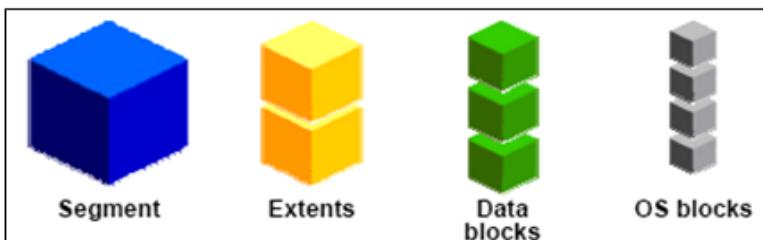
Obyek – obyek database seperti tabel dan indeks disimpan dalam tablespace sebagai segment. Setiap segment terdiri atas satu atau lebih extent. Sebuah extent terdiri atas blok – blok data yang saling berdampingan, dimana berarti bahwa setiap extent dapat ada hanya dalam satu file data. Blok data adalah unit masukan/keluaran terkecil dalam database.

Ketika database meminta sebuah pengaturan blok data dari sistem operasi, maka sistem operasi memetakannya kepada sebuah blok sistem operasi yang sedang berlangsung pada peralatan penyimpanan. Karenanya, Anda tidak perlu mengkuatirkan alamat fisik dari data dalam database Anda. Hal ini juga berarti bahwa sebuah file data dapat ditandai atau dicerminkan pada piringan yang lain.

Ukuran dari blok data dapat diatur pada waktu pembuatan database. Ukuran default dari 8K layak untuk kebanyakan database. Jika database Anda mendukung sebuah aplikasi gudang data yang mana mempunyai tabel dan indeks yang besar, maka sebuah blok besar akan sangat menguntungkan. Jika database Anda mendukung sebuah aplikasi transaksi dimana baca dan tulis terjadi sangat acak, maka menggunakan sebuah blok kecil akan sangat menguntungkan. Ukuran blok maksimal akan sangat tergantung pada sistem operasi Anda. Ukuran blok minimal adalah 2K dan biasanya jarang digunakan.

Anda dapat memiliki tablespace dengan ukuran – ukuran blok yang berbeda. Biasanya digunakan hanya untuk mendukung pemindahan tablespace. Untuk selengkapnya silahkan lihat **Database Administrator's Guide**. Intinya adalah:

- ✓ Segment ada dalam sebuah tablespace.
- ✓ Segment dibuat dari sekumpulan extent.
- ✓ Extent adalah sekumpulan blok data.
- ✓ Blok data dipetakan pada blok sistem operasi.



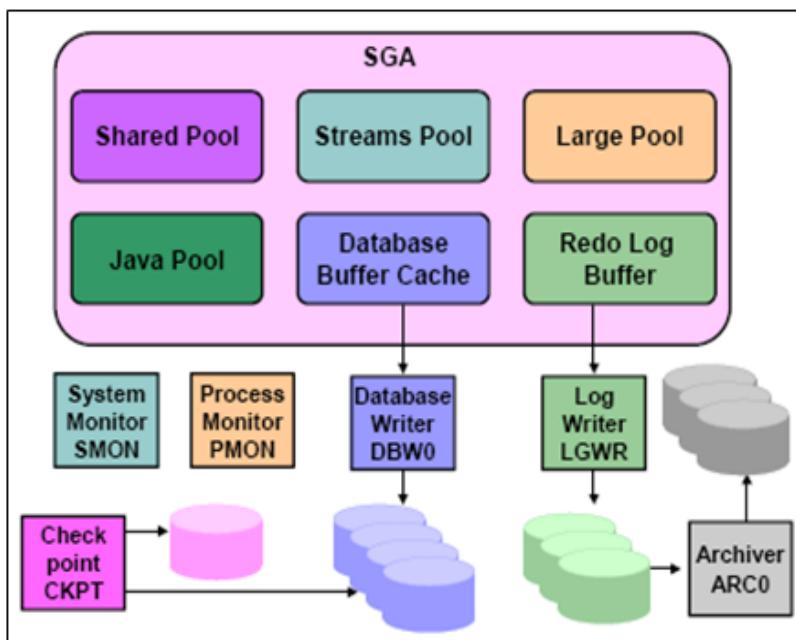
Gambar 5. Struktur Penyimpanan Logik Oracle

## 1.8. Pengaturan Perangkat Oracle

Sebuah server database Oracle terdiri atas sebuah database Oracle dan sebuah perangkat Oracle. Sebuah perangkat Oracle terbuat dari buffer – buffer memori yang dikenal sebagai **System Global Area (SGA)** dan proses yang terjadi di belakangnya yang memegang banyak peranan termasuk juga dalam menjalankan perangkat tersebut.

Perangkat tersebut menganggur sampai dijalankan. Ketika perangkat tersebut dijalankan, sebuah file parameter inisialisasi akan terbaca dan perangkat akan tersusun sesuai dengan perintah yang terkandung di dalam file parameter.

Setelah perangkat dimulai dan database terbuka, pengguna dapat mengakses database.



Gambar 6 Struktur Oracle

## 1.9. Struktur Memori Oracle

Struktur memori dasar yang berhubungan dengan perangkat Oracle termasuk:

- ✓ **System Global Area (SGA):**

Dihubungkan oleh semua server dan proses yang terjadi di belakangnya.

- ✓ **Program Global Area (PGA):**

Khusus untuk setiap server dan proses yang terjadi di belakangnya. Hanya ada satu PGA untuk setiap proses.

SGA adalah sebuah daerah memori bersama yang terdiri atas data dan informasi pengendalian untuk perangkat Oracle. Struktur data yang ada dalam SGA:

- ✓ Database buffer cache:

Menyimpan blok – blok data yang diperoleh dari database.

- ✓ Redo log buffer:

Menyimpan informasi redo (digunakan untuk recovery) sampai dapat dituliskan pada file fisik redo log yang tersimpan pada piringan.

- ✓ Shared pool:

Menyimpan berbagai macam perintah yang dapat dibagikan antar pengguna.

- ✓ Large pool:

Daerah pilihan yang digunakan untuk menahan permintaan masukan/keluaran yang besar.

- ✓ Java pool:

Digunakan untuk semua kode spesifik Java dan data di dalam **Java Virtual Machine (JVM)**.

- ✓ Streams pool:

Digunakan untuk aliran Oracle.

Dengan infrastruktur dinamis SGA, ukuran dari database buffer cache, shared pool, large pool, Java pool, dan streams pool dapat berubah tanpa harus mematikan perangkat Oracle.

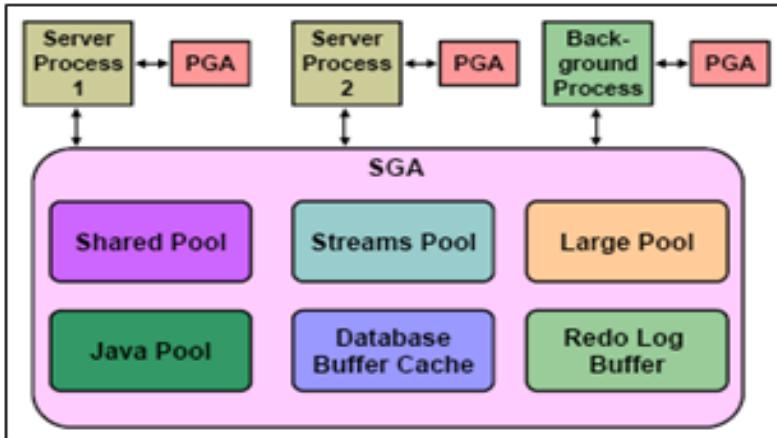
Konfigurasi awal database akan diatur dengan pengaturan yang layak untuk parameter – parameter memori. Meskipun demikian, untuk perluasan kegunaan database, Anda mungkin dapat menemukannya jika perlu untuk mengubah pengaturan parameter – parameter memori.

Oracle menyediakan peringatan dan saran untuk mengidentifikasi masalah pengukuran memori dan untuk membantu Anda menentukan nilai yang layak bagi parameter – parameter memori.

Sebuah **Program Global Area (PGA)** adalah sebuah daerah memori yang terdiri atas data dan informasi pengendalian untuk setiap proses server. Sebuah server proses adalah sebuah proses yang melayani permintaan klien. Setiap proses server mempunyai daerah khusus PGA sendiri yang dibuat ketika proses server dimulai. Aksesnya eksklusif untuk proses server, dan dibaca serta ditulis hanya oleh kode Oracle yang beraksi untuk keperluan tersebut.

Jumlah penggunaan memori PGA dan isinya tergantung dari perangkat tersusun dalam model pembagian server. Biasanya, PGA terdiri atas:

- ✓ Daerah khusus SQL:  
Terdiri atas data seperti informasi terikat dan memori terbatas. Setiap bagian menunjuk pada sebuah pernyataan SQL yang mempunyai daerah khusus SQL.
- ✓ Memori bagian:  
Memori ditempatkan untuk menahan bagian variabel – variabel dan informasi lainnya yang berhubungan dengan bagian tersebut.



Gambar 7. Struktur Memori Oracle

### 1.10. Proses – proses Oracle

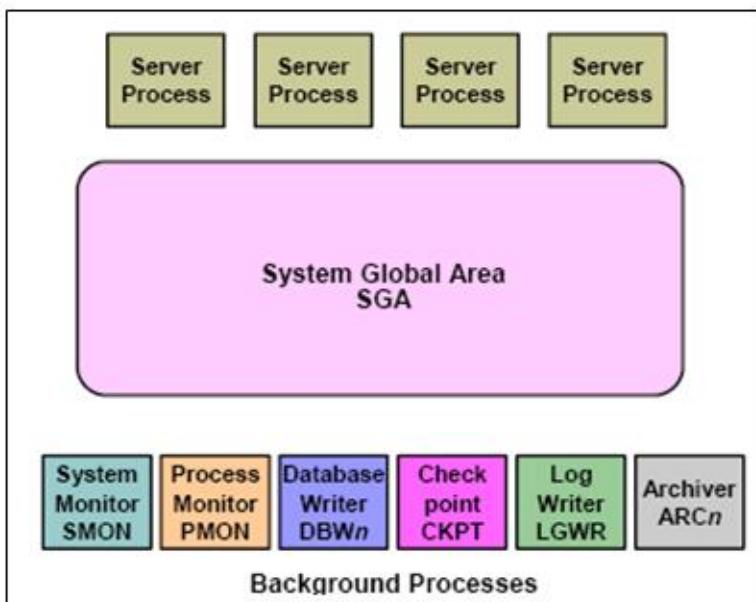
Ketika Anda memanggil sebuah program aplikasi atau sebuah peralatan Oracle seperti ***Enterprise Manager***, server Oracle akan membuat sebuah proses server untuk menjalankan perintah yang dibawa oleh aplikasi tersebut.

Oracle juga membuat sebuah proses belakang untuk perangkat yang berinteraksi satu sama lain dan dengan sistem operasi untuk mengatur struktur memori, terutama menampilkan masukan/keluaran untuk menulis data ke piringan, dan melakukan pekerjaan rumah seperti biasa.

Proses – proses yang terjadi di belakangnya tergantung dari tampilan yang digunakan dalam database. Proses yang sering terjadi di belakang adalah sebagai berikut:

- ✓ ***System Monitor (SMON)***:  
Menampilkan recovery kerusakan ketika perangkat dijalankan diikuti kegagalan.
- ✓ ***Process Monitor (PMON)***:  
Menampilkan pembersihan proses ketika terjadi kegagalan proses pengguna.

- ✓ **Database Writer (DBWn):**  
Menuliskan blok – blok modifikasi dari database buffer cache ke file – file dalam piringan.
- ✓ **Checkpoint (CKPT):**  
Sinyal – sinyal DBWn pada checkpoint dan memperbaharui semua file data dan control file dari database untuk mengindikasikan checkpoint terbaru.
- ✓ **Log Writer (LGWR):**  
Menuliskan entri – entri redo log ke piringan.
- ✓ **Archiver (ARCn):**  
Menggandakan file – file redo log ke penyimpanan tertentu ketika file – file log penuh atau terjadi sebuah pemindahan log.



Gambar 8. Struktur Background Proses Oracle

## 1.11. Kamus Data

Kamus data adalah pusat pengaturan tabel dan tampilan yang digunakan sebagai referensi read-only mengenai sebuah database terutama. Sebuah kamus data menyediakan informasi seperti:

- ✓ Struktur fisik dan logika dari database.
- ✓ Para pengguna database.
- ✓ Informasi tentang konstrain.
- ✓ Seberapa besar ruang yang dialokasikan untuk sebuah obyek skema dan berapa banyak yang akan digunakan.

Sebuah kamus data dibuat ketika sebuah database dibuat dan akan terupdate secara otomatis ketika struktur database terupdate.

Kamus data adalah tempat dimana *Enterprise Manager* menampilkan informasi tentang obyek – obyek dalam database. Anda dapat memilih informasi dari tabel kamus. *Enterprise Manager* akan melakukannya untuk Anda dan menampilkan informasi dalam sebuah format yang amat mudah digunakan. Tampilan **DICTIONARY** terdiri atas penjelasan dari tabel dan tampilan kamus data. Tabel dan tampilan tersebut biasanya mempunyai salah satu dari tiga awalan berikut ini:

- ✓ **USER:**  
Informasi yang berhubungan dengan obyek – obyek yang dimiliki oleh pengguna sekarang.
- ✓ **ALL:**  
Informasi yang berhubungan dengan obyek – obyek yang diakses oleh pengguna sekarang.
- ✓ **DBA:**  
Informasi yang berhubungan dengan semua obyek dalam database.

The screenshot shows the Oracle Database workspace interface. At the top, there is a text input field containing the SQL statement: "select \* from dictionary;". To the right of the input field is a "Clear" button. Below the input field are four buttons: "Execute", "Load Script", "Save Script", and "Cancel". The main area below the buttons displays a results grid with two columns: "TABLE\_NAME" and "COMMENTS". The grid contains four rows of data:

TABLE_NAME	COMMENTS
USER_RESOURCE_LIMITS	Display resource limit of the user.
USER_PASSWORD_LIMITS	Display password limits of the user.
USER_CATALOG	Tables, Views, Synonyms and Sequences owned by the user.
ALL_CATALOG	All tables, views, synonyms, sequences accessible to the user.

Gambar 9. Contoh Query

## 1.12. DATABASE CONTROL

Dalam Oracle Database tersedia ***Oracle Enterprise Manager's Database Control***. Database control adalah sebuah konsol web yang memungkinkan administrator database dapat menggunakannya untuk:

- ✓ Meningkatkan pengawasan.
- ✓ Mengatur peringatan – peringatan waspada.
- ✓ Mengendalikan wizard pemeliharaan dan saran.
- ✓ Administrasi obyek pengguna dan database.
- ✓ Backup database dan recoverynya.
- ✓ Pengaturan penyimpanan.
- ✓ Dan lain-lain.

Setiap Oracle Database yang Anda buat pasti akan mempunyai Database Control sendiri. Anda dapat menggunakan ***Enterprise Manager Database Control*** untuk mengatur database Anda.



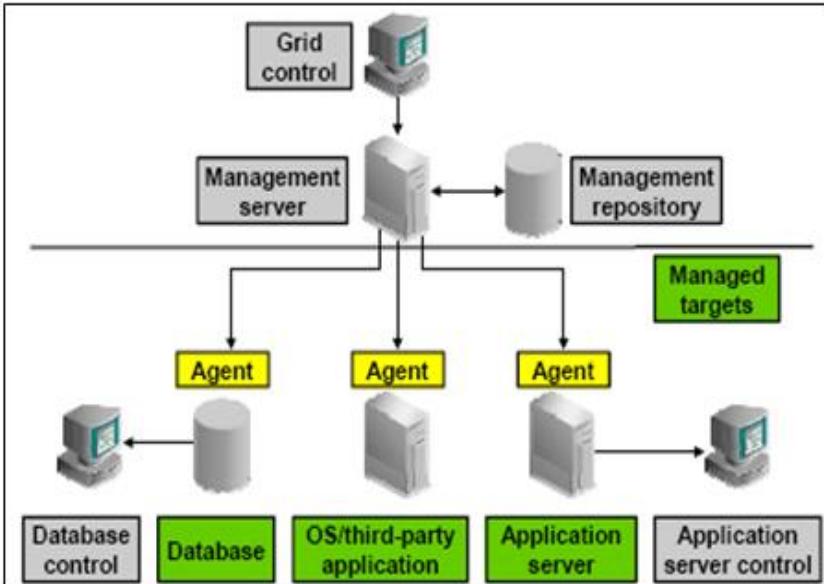
Gambar 10. Enterprise Manager Database Control

### 1.13. Grid Control

Kemampuan database control dapat diperluas dan diintegrasikan dengan sisa sistem Anda dengan menggunakan ***Oracle Enterprise Manager's Grid Control***. Arsitektur dari kerangka grid control menyediakan suatu tingkat fleksibilitas dan kegunaan yang tinggi. Anda dapat dengan mudah menggunakan ***Enterprise Manager*** untuk memenuhi kebutuhan pengawasan dan administrasi lingkungan Anda.

Kerangka konfigurasi ***Enterprise Manager*** terdiri atas daerah fungsi sebagai berikut:

- ✓ Target diatur.
- ✓ Pelayanan manajemen.
- ✓ Web berdasarkan grid control.
- ✓ Database control.
- ✓ Application server control.



Gambar 11. Grid Control

### 1.14. Database Configuration Assistant (DBCA)

DBCA memungkinkan Anda untuk membuat, mengubah konfigurasi database atau menghapus database. Anda juga dapat membuat sebuah database dari sebuah daftar templet yang ada atau menggunakan sebuah database yang sudah ada sebagai contoh untuk membuat sebuah database atau templet yang baru. Sebuah templet menunjukkan database yang Anda gunakan sebagai langkah awal untuk sebuah database yang baru.

Beberapa keterangan yang perlu Anda ketahui:

- ✓ ***Create a database:***

Jika Anda memilih ini, maka Anda dapat membuat sebuah database atau templet yang baru.

- ✓ ***Configure options in a database:***

Jika Anda memilih ini, maka Anda dapat mengkonfigurasi pilihan – pilihan terinstalasi yang sebelumnya tidak pernah

terkonfigurasi untuk penggunaan database Anda. Anda juga dapat membolehkan atau melarang shared server support.

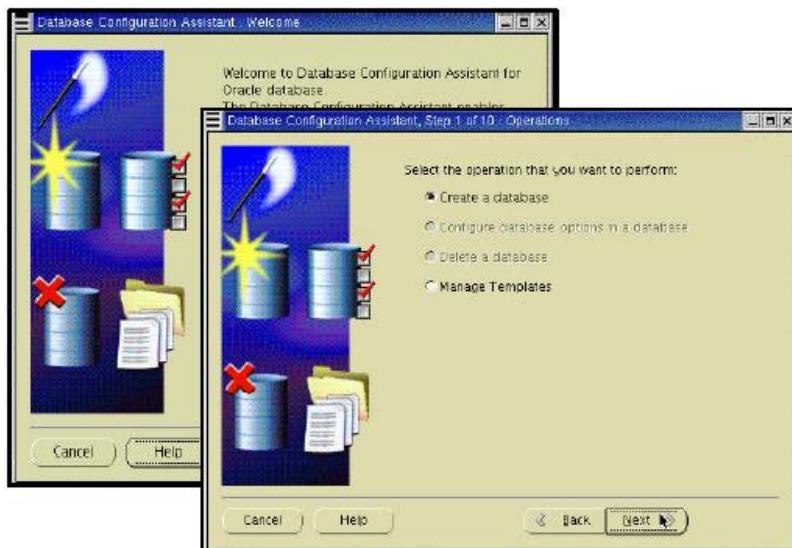
✓ **Delete a database:**

Jika Anda memilih ini, maka Anda dapat menghapus semua file database.

✓ **Manage templates:**

Jika Anda memilih ini, maka Anda mempunyai tiga cara untuk membuat sebuah templet:

- Dari templet yang sudah ada.
- Dari database yang sudah ada (hanya strukturnya saja).
- Dari database yang sudah ada (struktur seperti halnya data).



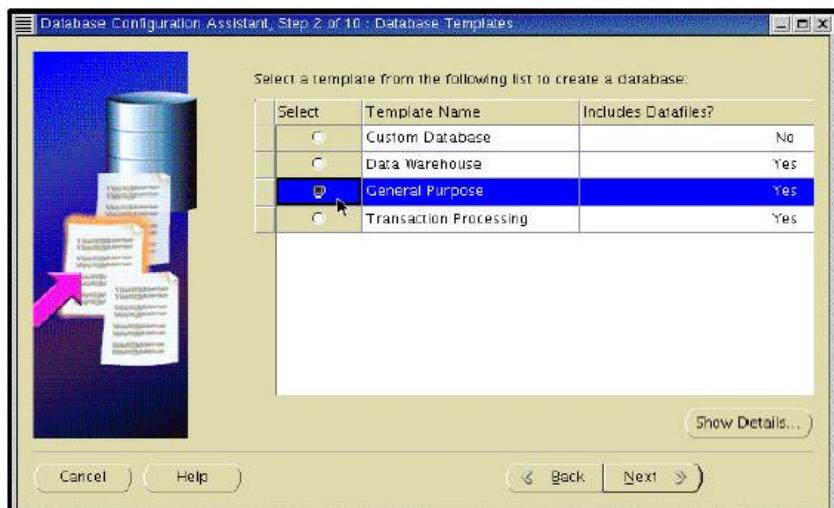
Gambar 12. Instalasi Oracle (1)

### 1.15. Membuat Database

Ketika Anda membuat database dengan DBCA maka Anda dapat memilih salah satu dari tiga database yang telah ditentukan, atau membuat sebuah custom database. Oracle

sendiri menyediakan templet yang terbatas. Ada templet untuk data warehouse, kepeluan umum, dan database proses transaksi. Templet – templet tersebut sudah berisikan pengaturan yang optimal untuk beban kerja. Klik **Show Detail** untuk melihat konfigurasi untuk setiap tipe database. Pilihlah templet yang sesuai untuk tipe beban kerja database Anda. Jika anda tidak yakin, pilihlah templet default **General Purpose**.

Untuk lingkungan yang lebih kompleks, Anda dapat memilihnya dari pilihan **Custom Database**. Hasilnya dalam interview yang lebih luas dan butuh waktu yang lama untuk membuat database Anda, karena pembuatan skrip database harus berjalan.

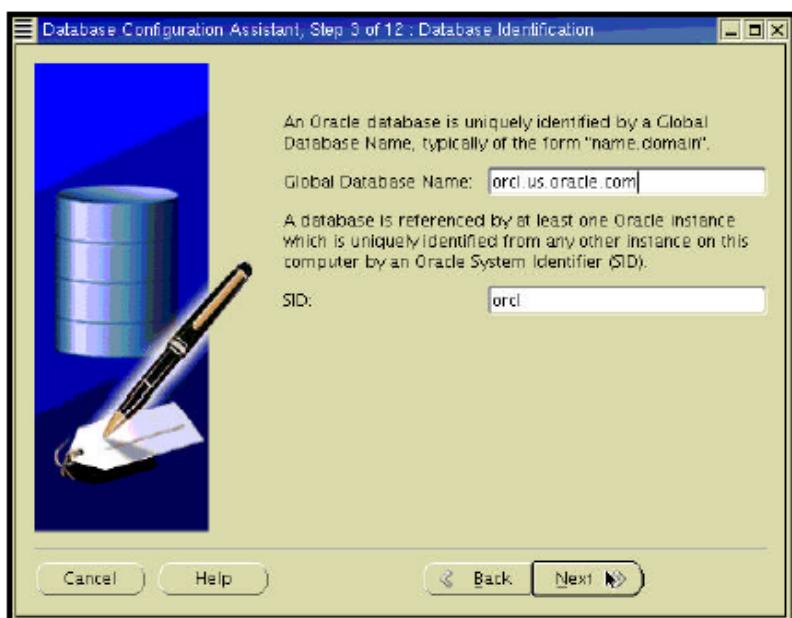


Gambar 13. Instalasi Oracle (2)

### 1.16. Identifikasi Database

Masukkan dalam **Global Database Name**, dalam form **database\_name.domain\_name** dan SID (Oracle System Identifier). SID secara langsung akan menunjukkan nama

database dan secara unik akan mengidentifikasi perangkat yang menjalankan database. Penting untuk mengerti bahwa SID adalah nama dari perangkat yang tersambung ke database dan bukanlah nama dari database itu sendiri. Perangkat dan database yang terkoneksi tidak perlu punya nama yang sama, meskipun hal itu sangat memudahkan. Dengan ***Real Application Cluster***, perangkat majemuk membuka database yang sama dan juga SID untuk setiap perangkat akan juga berbeda.



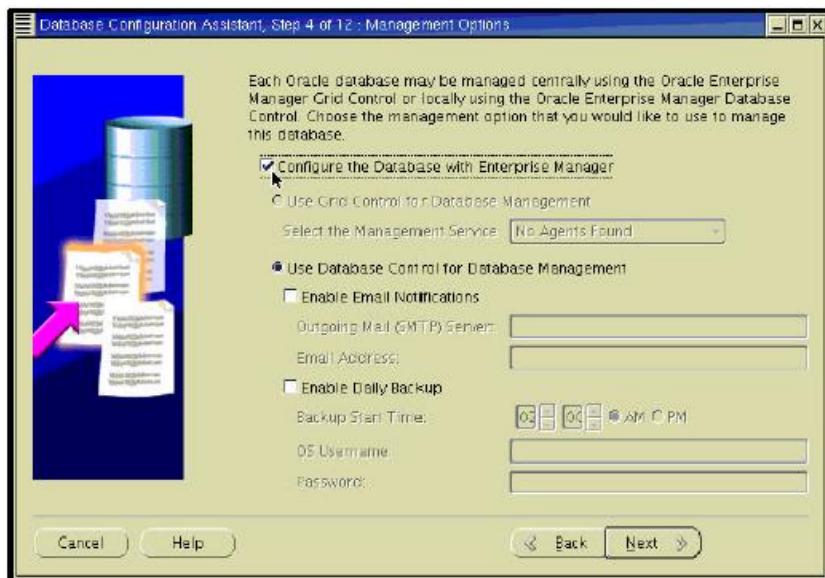
Gambar 14. Instalasi Oracle (3)

### 1.17. Management Options

Gunakan halaman ini untuk mengatur database Anda jadi database Anda dapat diatur dengan ***Oracle Enterprise Manager***, yang menyediakan peralatan pengaturan web untuk database individual, sebaik peralatan pengaturan terpusat untuk mengatur lingkungan Oracle Anda. Untuk menggunakan ***Enterprise***

**Manager**, pilihlah **Configure the Database with Enterprise Manager**.

Jika **Oracle Management Agent** sudah terinstalasi pada komputer host, maka Anda akan diberi pilihan dalam menyeleksi pengaturan terpusat dengan cara menyeleksi **Use Grid Control** untuk **Database Management**. Jika Anda menyeleksi tipe pengaturan, maka anda juga harus mengindikasikan pelayanan pengaturan yang manakah yang akan dipakai dalam menu drop-down. Selain itu, pilihlah **Use Database Control for Database Management** untuk mengatur database Anda secara individual. Jika Anda memilih pilihan ini, Anda dapat menambahkan **Enable Email Notifications** dan **Enable Daily Backup**. Klik **Help** untuk keterangan selengkapnya tentang pilihan ini.



Gambar 15. Instalasi Oracle (4)

## **1.18. Kata Kunci dan Penyimpanan**

Kata kunci skema database: menyediakan kata kunci untuk pengguna administrasi, **SYS**, **SYSTEM**, **SYSMAN** dan **DBSNMP**. Anda dapat menyediakan sebuah kata kunci untuk setiap bagian terpisah atau menyediakan satu kata kunci untuk semuanya. *File Storage Options* terdiri atas beberapa:

✓ ***File System:***

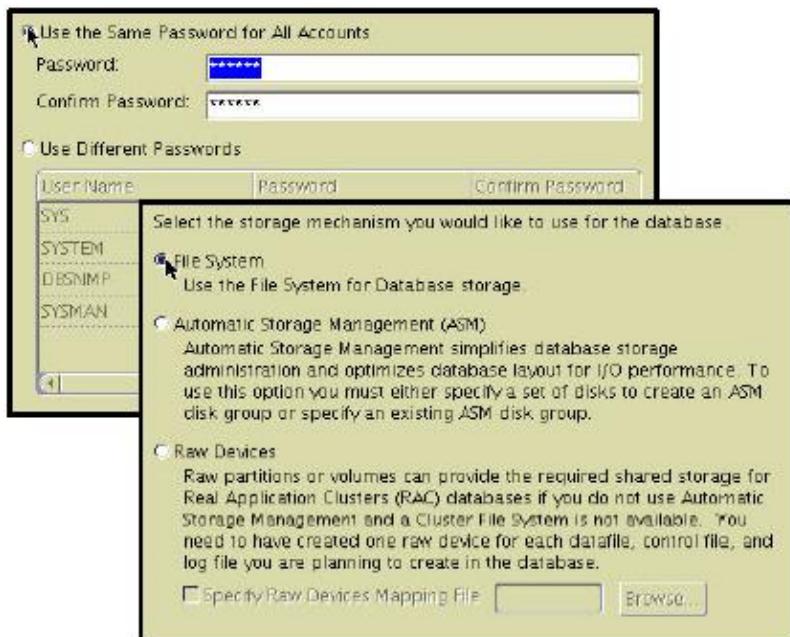
Menyimpan file dalam sistem file sistem operasi terkonfigurasi Anda.

✓ ***Automatic Storage Management (ASM):***

File ASM dibuat dan diatur secara otomatis, dan Anda mendapatkan keuntungan tampilan – tampilan tambahan seperti pencerminan dan penandaan. Untuk selengkapnya dalam bagaimana cara mengatur ASM, lihatlah *Oracle Database Administrator's Guide*.

✓ ***Raw Devices* (partisi):**

Merupakan partisi piringan tanpa sebuah sistem file di dalamnya. Biasanya Anda dapat menggunakan hanya jika Anda sudah sangat familiar dengan penggunaan partisi mentah. Periksa dokumentasi sistem operasi Anda untuk pengaturan dan pemeliharaan partisi mentah selengkapnya.



Gambar 16. Instalasi Oracle (5)

## 1.19. Pilihan Lokasi File dan Recovery Backup

Beberapa keterangan dalam pilihan lokasi file:

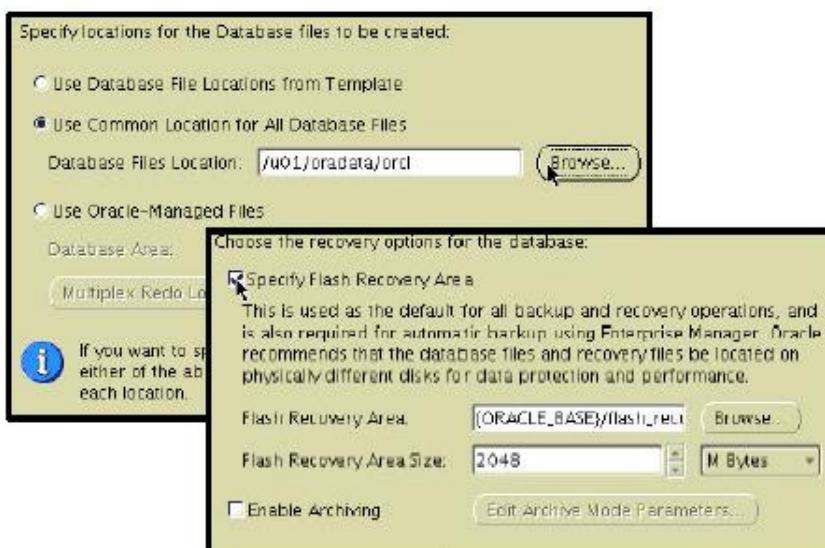
- ✓ ***Use Database File Locations from Template:***  
Memilih pilihan ini akan mengintruksikan DBCA untuk menggunakan informasi direktori seperti yang dispesifikasikan oleh templet. Anda akan berkesempatan nantinya untuk membuat modifikasi ke nama – nama file database dan lokasinya.
- ✓ ***Use Common Location for All Database Files:***  
Pilihan ini meminta Anda untuk menspesifikasikan sebuah daerah untuk semua file database Anda. Anda nantinya akan berkesempatan untuk membuat modifikasi untuk nama – nama file database dan lokasinya dalam halaman *Storage*.

✓ **Use Oracle-Managed Files:**

Pilihlah pilihan ini untuk menyinjirkan kebutuhan Anda, DBA, untuk secara langsung mengatur file sistem operasi yang dibandingkan database Oracle. Anda harus myediakan lintasan untuk daerah database. Untuk keterangan lengkap mengenai **Oracle Managed Files** lihatlah **Database Administrator's Guide**.

Beberapa keterangan dalam pilihan recovery backup:

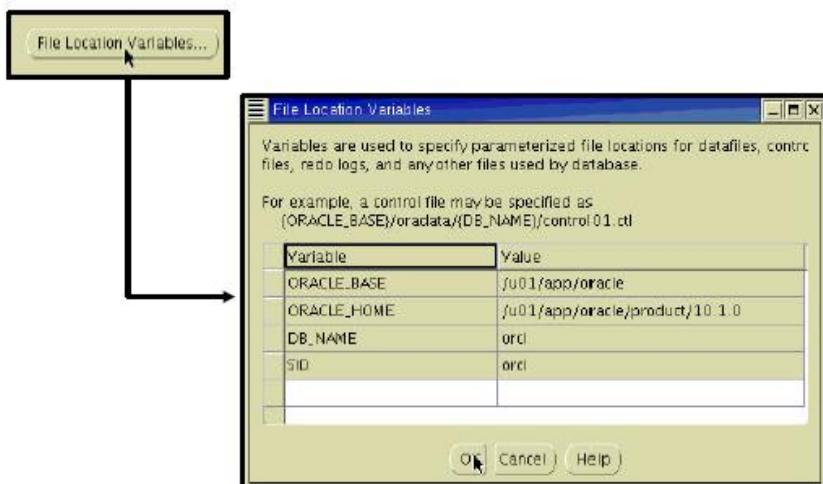
- ✓ Sebuah **Flash Recovery Area** adalah sebuah lokasi dimana Oracle dapat menyimpan dan mengatur file – file yang berhubungan dengan backup dan recovery. Untuk keterangan lengkapnya lihatlah panduan **Oracle Database Backup and Recovery Basics**.
- ✓ **Enable Archiving** meletakkan database dalam mode archive log pada waktu pembuatan.



Gambar 17. Instalasi Oracle (6)

## 1.20. File Location Variables

Pada beberapa halaman Anda dapat mengklik **File Location Variables** untuk membuka sebuah halaman yang akan menunjukkan kepada Anda mengenai definisi dari variabel yang ada. Variabel – variabel tersebut digunakan dalam pendefinisian lintasan untuk file – file database. Anda tidak dapat mengubah nilai pada saat di DBCA. Jika Anda perlu memodifikasi nilai tersebut, maka Anda harus keluar dari DBCA, mengubah bagian tersebut dalam lingkungan OS, dan kemudian merestart DBCA.



Gambar 18. Instalasi Oracle (7)

## 1.21. Parameter Isi dan Inisialisasi

*Sampel Schemas* merupakan sebuah daerah pengaturan skema untuk demonstrasi dan training. *Custom Scripts*, disini Anda dapat menspesifikasi skrip – skrip yang ingin Anda jalankan pada waktu pembuatan.

Pada parameter inisialisasi terdapat 4 tab yang dapat digunakan untuk mengatur parameter – parameter yang biasa digunakan, dan dengan mengklik **All Initialization Parameters**

maka Anda dapat menampilkan dan mengatur semua parameter seperti:

✓ **Memory:**

Mengalokasikan memori yang digunakan oleh SGA dan setiap PGA dari proses pengguna.

✓ **Sizing:**

Disini Anda dapat mengatur ukuran blok, tetapi jika menggunakan sebuah templet maka ukuran blok tidak bisa diubah. Anda juga dapat mengatur nomor maksimal dari proses OS yang diijinkan terkoneksi pada perangkat.

✓ **Character Sets:**

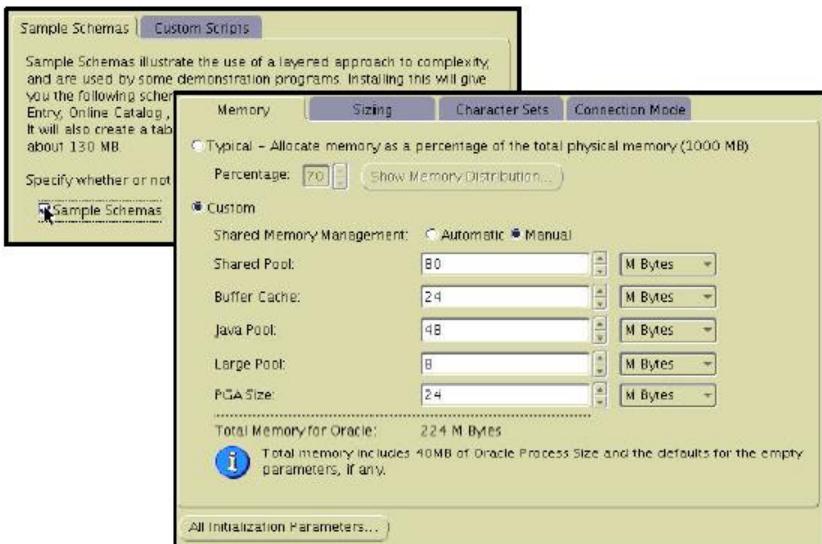
Anda dapat mengatur pengaturan karakter default untuk database dan pengaturan karakter nasional. Pengaturan karakter default digunakan untuk kebanyakan tipe data dalam database. Tipe – tipe data **NCHAR**, **NVARCHAR2**, dan **NCLOB** hanya mendukung data **Unicode**, dimana terdapat pilihan pengaturan karakter nasional. Anda dapat menggunakan antara pengaturan karakter **AL32UTF8** atau **AL16UTF16**. Untuk informasi selengkapnya lihatlah *Globalization Support Guide*.

✓ **Connection Mode:**

Database Oracle membuat proses server untuk menangani permintaan proses pengguna yang terhubung pada perangkat. Sebuah server proses dapat melakukan hal – hal sebagai berikut:

- Sebuah proses berdedikasi, dimana satu proses server hanya melayani satu proses pengguna.
- Proses server bersama, dimana satu proses server dapat melayani banyak proses pengguna.

Database Anda selalu dapat mengijinkan proses server berdedikasi, tetapi Anda harus secara khusus mengkonfigurasi dan memungkinkan pembagian server dengan mengatur satu atau lebih parameter inisialisasi.



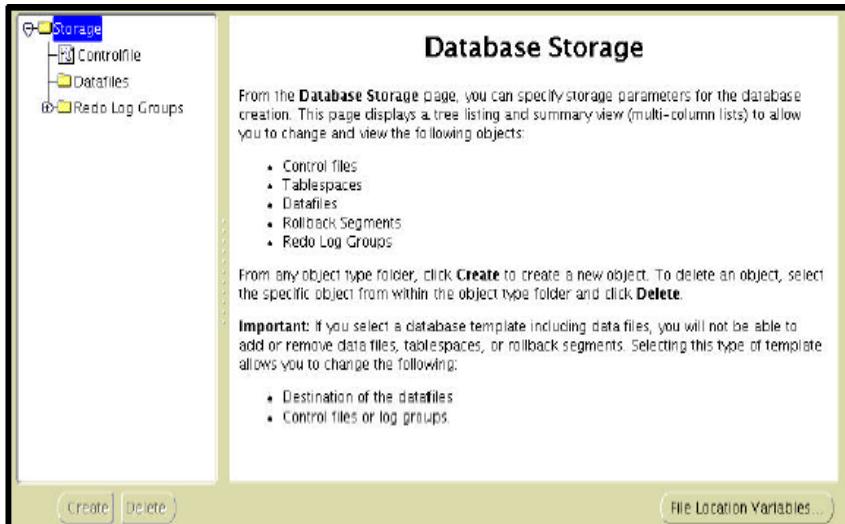
Gambar 19. Instalasi Oracle (8)

## 1.22. Database Storage

Pada halaman ini Anda dapat melihat pengaturan penyimpanan untuk control file, data file, dan online redo log file. Ketika menggunakan sebuah templet maka Anda tidak dapat menambahkan file data apapun ke database, tetapi Anda dapat menambahkan lebih banyak control file dan online redo log file.

DBCA secara otomatis akan menggandakan control file melalui tiga file. Anda dapat menambahkan lebih banyak lagi jika mau.

Redo log bekerja dalam grup dan bisa digandakan. DBCA tidak secara otomatis menggandakan online redo log file. Setiap file dalam sebuah grup log adalah penggandaan dari anggota lain dari grup. Anda dapat menambah lebih banyak anggota dalam grup sekarang atau menambahkan bagian tersebut suatu saat nanti setelah pembuatan database.



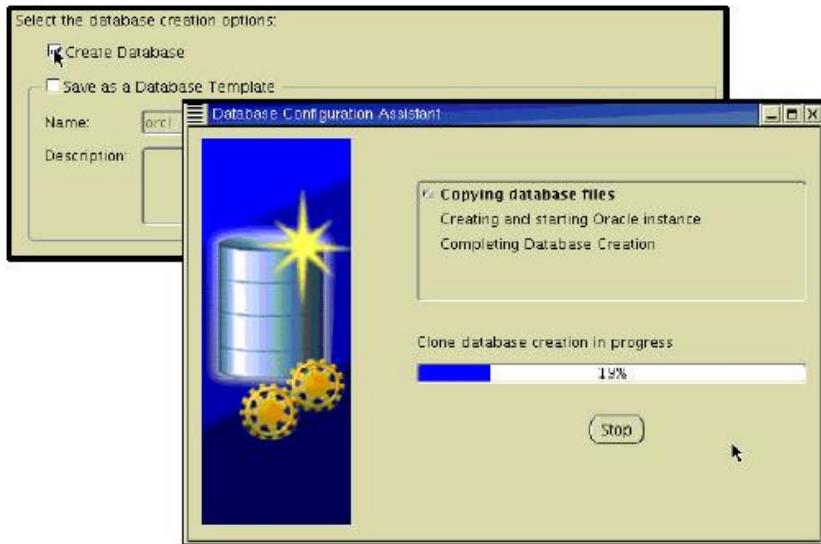
Gambar 20. Instalasi Oracle (9)

### 1.22.1. Pilihan Creation dan Create

Anda mempunyai pilihan menyimpan database yang Anda definisikan sebagai sebuah templet. Templet ini dapat digunakan nantinya dengan berbagai macam pilihan yang sudah Anda definisikan untuk pembuatan database.

Setelah Anda mengklik **Next** pada halaman ***Creation Options***, sebuah halaman konfirmasi muncul dimana Anda dapat mengulang semua pilihan yang telah diambil. Ini merupakan kesempatan terakhir untuk membuat perubahan sebelum proses pembuatan dimulai. Anda juga dapat menyimpan halaman konfirmasi ini sebagai file HTML untuk mengulangnya kembali nanti. Setelah Anda mengklik **OK** pada halaman konfirmasi, maka pembuatan database dimulai.

Pada akhir instalasi Anda akan melihat sebuah halaman yang menampilkan kesempatan untuk membuka pembuatan account dan mengubah kata kunci jika Anda menginginkannya. Klik **Password Management** untuk membuka account dan ubahlah kata kunci. Klik **OK** ketika tidak mengatur account, kemudian klik **Exit** pada DBCA.



Gambar 21. Instalasi Oracle (10)

### 1.22.2. Aksi Lain dengan DBCA

Pilihan database konfigurasi dalam sebuah database: mengijinkan Anda untuk menambah pilihan pada database yang ada.

**Delete a database:** secara permanen akan menyengkirkan sebuah database dari sistem Anda.

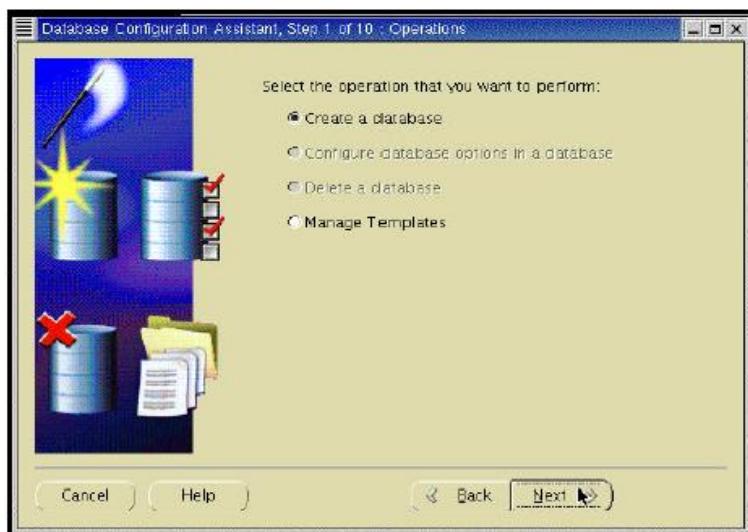
**Manage Templates:** mengijinkan Anda untuk menampilkan templet – templet berikut ini:

✓ ***Create a Template:***

- Dari sebuah templet yang sudah ada:  
Menggunakan templet yang sudah ada untuk membangun templet baru. Banyak halaman yang sama dengannya dalam membuat proses database.
- Dari sebuah database yang sudah ada (hanya struktur):  
Meminta logging dalam database yang sudah ada dan menggunakan strukturnya.
- Dari sebuah database yang sudah ada (struktur seperti halnya data):  
Meminta logging dalam database yang sudah ada dan menggunakan strukturnya sebagai dasar, meskipun demikian dapat menangkap file – file data dengan baik.

✓ ***Delete a database template:***

Secara permanen menyingkirkan sebuah templet database dari sistem Anda.



Gambar 22. Instalasi Oracle (11)



## BAB II

# MENGENDALIKAN DATABASE

### 2.1. Tujuan

Setelah menyelesaikan pelajaran ini, Anda harus dapat melakukan hal – hal sebagai berikut:

- ✓ Memulai dan mengakhiri **iSQL\*Plus**.
- ✓ Memulai dan mengakhiri **Enterprise Manager Database Control**.
- ✓ Memulai dan mengakhiri **Oracle Listener**.
- ✓ Memulai dan mematikan **Oracle Database 10g**.

### 2.2. Memulai dan Mengakhiri iSQL\*Plus

**iSQL\*Plus Application Server** harus berjalan sebelum Anda dapat memulai sebuah sesi **iSQL\*Plus**. Sebuah **command-line utility** dan sebuah **Windows Service** akan tersedia untuk memulai dan mengakhiri **iSQL\*Plus** pada Windows. **iSQL\*Plus Application Server** dimulai secara default selama instalasi database Oracle.

Untuk memulai atau mengakhiri **iSQL\*Plus Application Server** pada Unix (Linux) dilakukan:

- ✓ Memulai sebuah sesi terminal.
- ✓ Memasuki hal berikut ini untuk memulai: **isqlplusctl start**
- ✓ Memasuki hal berikut ini untuk mengakhiri: **isqlplusctl stop**

Anda mungkin harus mengarahkan ke dalam direktori **\$ORACLE\_HOME/bin** jika direktori ini tidak ada dalam lintasan OS Anda.

Untuk memulai atau mengakhiri **iSQL\*Plus Application Server** pada Windows dilakukan:

- ✓ Pilih *Services* dari menu *Start > Programs > Administrative Tools*.
- ✓ Tempatkan **iSQL\*Plus Windows Service, OracleHomeName iSQL\*Plus**.
- ✓ Memulai **Windows Service** untuk memulai **iSQL\*Plus Application Server**, atau mengakhiri **Windows Service** untuk mengakhiri **iSQL\*Plus Application Server**.
- ✓ Alternatif lain, Anda dapat memulai **iSQL\*Plus** dari sebuah **command prompt**. Gunakan perintah yang sama pada Unix (Linux).

```
$ isqlplusctl start
iSQL*Plus 10.1.0.2.0
Copyright (c) 2004 Oracle. All rights reserved.
Starting iSQL*Plus ...
iSQL*Plus started.
```

```
$ isqlplusctl stop
iSQL*Plus 10.1.0.2.0
Copyright (c) 2004 Oracle. All rights reserved.
Stopping iSQL*Plus ...
iSQL*Plus stopped.
```

Gambar 23. iSQLplus

### 2.3. Kerangka Pengaturan

Tiga komponen dalam *Oracle Database 10g* adalah:

- ✓ Perangkat database.
- ✓ **Listener**.

- ✓ Antarmuka pengaturan.

Merupakan agen pengaturan yang menghubungkan server kepada *Oracle Enterprise Manager Grid Control*, atau *Oracle Enterprise Manager Database Control*.

Setiap komponen harus dimulai dengan jelas sebelum Anda dapat menggunakan pelayanan komponen tersebut dan harus dimatikan dengan bersih ketika mematikan server hosting *Oracle Database 10g*.

Komponen pertama yang harus dimulai adalah antarmuka pengaturan. Setelah ini berjalan, antarmuka pengaturan dapat digunakan untuk memulai komponen yang lainnya.



Gambar 24. Listener

## 2.4. Memulai dan Mengakhiri Database Control

Untuk database yang tidak terkoneksi pada kerangka *Grid Control*, Oracle menyediakan konsol pengaturan mandiri yang dinamakan *Database Control*. Setiap database diatur dengan *Database Control* yang mempunyai sebuah instalasi *Database Control* yang terpisah, dan dari *Database Control* manapun Anda dapat mengatur hanya satu database saja. *Database Control* meminta proses **dbconsole** yang sedang dijalankan diprioritaskan untuk digunakan. Untuk memulai proses **dbconsole**:

**emctl start dbconsole**

Untuk mengakhiri proses **dbconsole**:

**emctl stop dbconsole**

Untuk menampilkan status dari proses **dbconsole**:

**emctl status dbconsole**

Anda mungkin harus mengarahkan ke dalam direktori **\$ORACLE\_HOME/bin** jika direktori ini tidak ada dalam lintasan OS Anda.

**Database Control** menggunakan sebuah proses agen **server-side**. Proses agen ini secara otomatis memulai dan mengakhiri ketika proses **dbconsole** dimulai atau diakhiri.

```
$ emctl start dbconsole
TS set to US/Pacific
Oracle Enterprise Manager 10g Database Control Release 10.1.0.2.0
Copyright (c) 1996, 2004 Oracle Corporation. All rights reserved.
http://edrsr9pl.us.oracle.com:5500/em/console/aboutApplication
Starting Oracle Enterprise Manager 10g Database Control
..... started.

-----
Logs are generated in directory
/u01/app/oracle/product/10.1.0/db_1/edrsr9pl.us.oracle.com_orcl/sy
sman/log
```

```
$ emctl stop dbconsole
TS set to US/Pacific
Oracle Enterprise Manager 10g Database Control Release 10.1.0.2.0
Copyright (c) 1996, 2004 Oracle Corporation. All rights reserved.
http://edrsr9pl.us.oracle.com:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 10g Database Control ...
... Stopped.
```

Gambar 25. Database Control

## 2.5. Mengakses Database Control

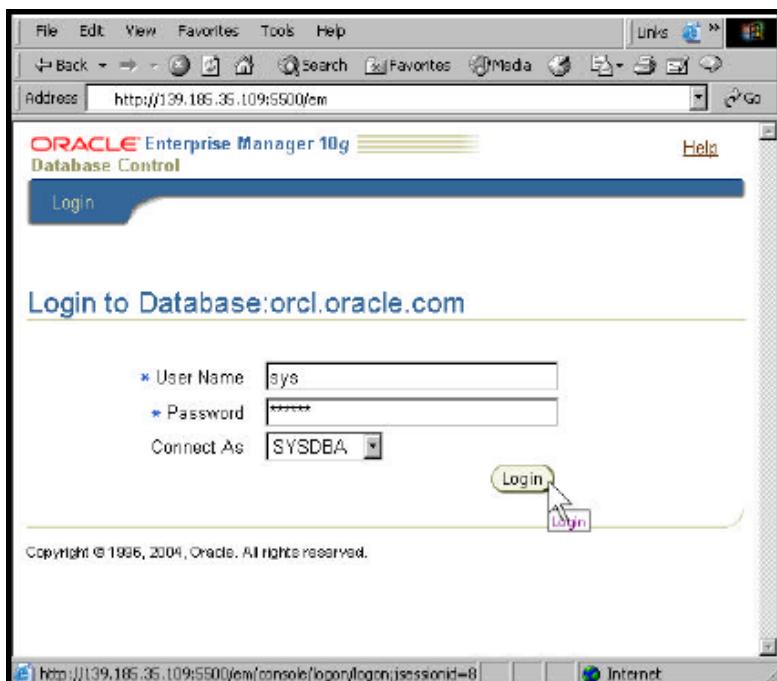
Bukalah browser Web Anda dan masuki URL berikut ini (port defaultnya adalah 5500):

**http://hostname:portnumber/em**

Jika database dibuka, maka **Enterprise Manager** akan menampilkan halaman **Database Control Login**. Masuklah ke dalam database dengan menggunakan sebuah nama pengguna

yang telah diijinkan untuk mengakses **Database Control**. Inisialnya akan menjadi **SYS**, **SYSMAN** atau **SYSTEM**. Gunakan kata kunci yang telah Anda persiapkan untuk account tersebut selama instalasi database.

Jika database turun dan perlu untuk dimulai, maka **Enterprise Manager** akan menampilkan halaman **Startup/Shutdown and Perform Recovery**. Jika terjadi kasus ini, klik tombol **Startup/Shutdown**. Anda kemudian akan dipromted untuk host dan target nama pengguna dan kata kunci login database, dimana Anda harus masuki. Untuk pengguna database dan kata kunci, gunakan **SYS** dan kata kunci yang Anda persiapkan selama instalasi. Klik **OK** untuk memulai database. Dalam halaman konfirmasi, klik **YES** untuk memulai database dalam mode terbuka.

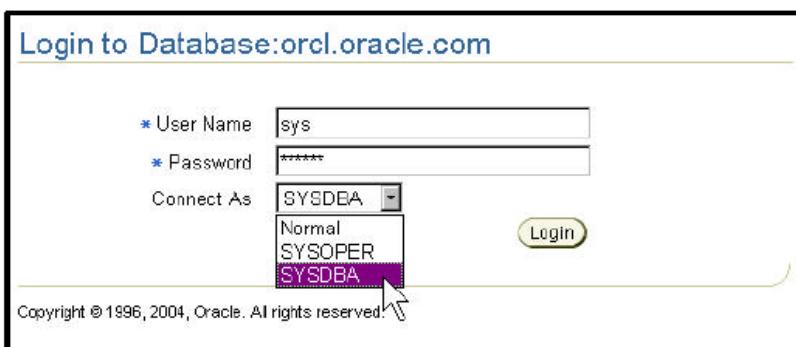


Gambar 26. Login Enterprise Manager (1)

## 2.6. SYSOPER dan SYSDBA

**SYSOPER** adalah peraturan database spesial yang mengijinkan administrator database untuk melakukan **STARTUP**, **SHUTDOWN**, **ALTER**, **DATABASE OPEN/MOUNT**, **ALTER DATABASE BACKUP**, **ARCHIVE LOG**, dan **RECOVER**, dan termasuk ijin **RESTRICTED SESSION**. Ketika Anda terhubungkan dengan **SYSDBA**, Anda ada dalam skema yang dimiliki oleh **SYS**.

**SYSDBA** adalah peraturan database spesial yang terdiri dari semua ijin masuk sistem dengan **ADMIN OPTION** dan ijin masuk **SYSOPER**. **SYSDBA** juga mengijinkan aksi **CREATE DATABASE** dan recovery yang tidak lengkap. Ketika Anda terkoneksi sebagai **SYSOPER**, maka Anda berada dalam skema publik.



Gambar 27. Login Enterprise Manager (1)

## 2.7. Database Home Page

Halaman ini memungkinkan Anda untuk mengakses performansi, administrasi, dan pemeliharaan halaman untuk mengatur database Anda. Bagian yang beragam dalam database home page, dan link yang saling berhubungan, menyediakan banyak informasi berharga tentang lingkungan dan kesehatan database.

Untuk dapat mengaksesnya ke pengguna yang lain ikutilah prosedur berikut ini:

1. Mulai dengan Web browser dan log dalam **Database Control** sebagai **SYS** atau pengguna database **SYSTEM**.
2. Klik **Setup** di atas database homepage.
3. Klik **Administrators** di kotak navigasi kiri.
4. Klik **Create** untuk membuat pengguna baru **Enterprise Manager** dengan menugaskan ijin masuk manajemen ke pengguna database yang sudah ada.
5. Klik ikon berkedip di depan **Name** dan pilih pengguna database yang sudah ada dari **pop-up window**.
6. Masukkan kata kunci untuk pengguna terpilih dan klik **Finish**.



Gambar 28. Database Homepage

## 2.8. Mengubah Status Listener

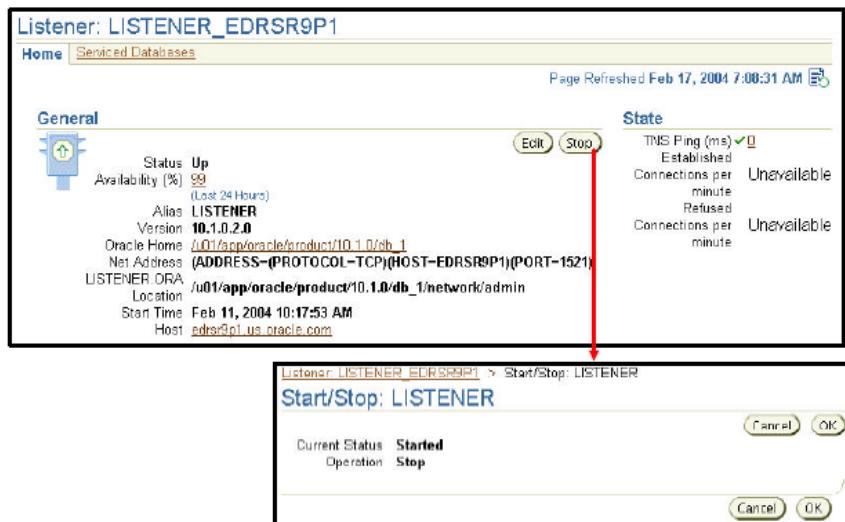
Dari Database home page klik nama **Listener** untuk membuka Listener home page. Klik **Stop** untuk menghentikan

listener jika sedang berjalan atau **Start** untuk memulai listener jika berhenti. Anda mesti log sebagai host sebagai pengguna **OS** jika Anda belum melakukannya. Ini akan menjadi pengguna **OS** ketika memulai atau mengakhiri listener.

Perintah untuk memulai, mengakhiri, dan melihat status dari listener adalah:

```
1snrct1 START [listener_name]  
1snrct1 STOP [listener_name]  
1snrct1 STATUS [listener_name]
```

Dimana *listener\_name* adalah merupakan nama yang didefinisikan dalam file **listener.ora**. Hal ini tidak perlu dilakukan untuk mengidentifikasi listener jika Anda menggunakan listener default yang dinamakan **LISTENER**. Perintah **COMMAND** menyediakan informasi status dasar tentang listener, termasuk ringkasan dari pengaturan konfigurasi listener, alamat protokol listener, dan ringkasan pelayanan yang terregistrasi dengan listener.

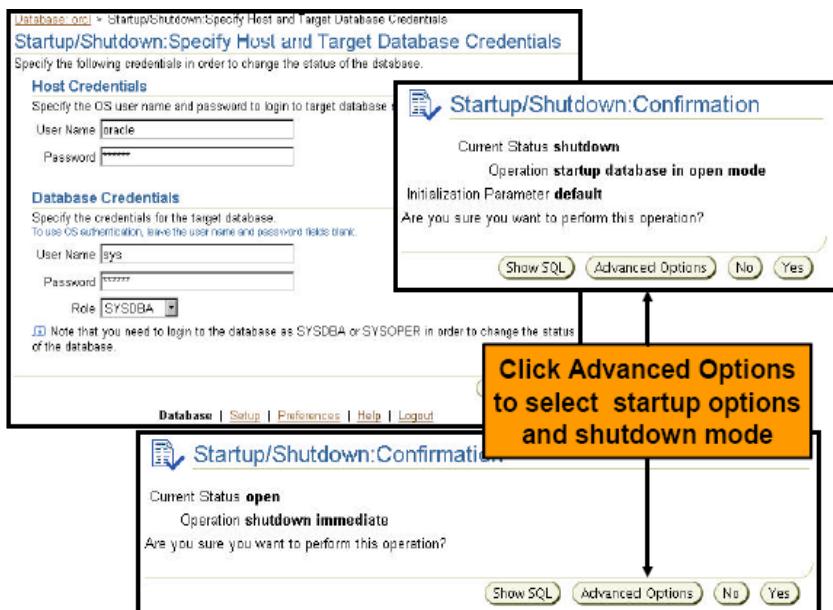


Gambar 29. Status Listener

## 2.9. STARTUP dan SHUTDOWN

Ketika Anda mengklik *startup* atau *shutdown* Anda mungkin akan diprompted untuk dipercayakan akan digunakan untuk logging ke dalam host (letak database komputer) dan database itu sendiri.

Anda kemudian dapat mengklik *Advance Options* untuk mengubah pilihan *startup* manapun atau mode *shutdown* ketika diperlukan. Anda juga dapat mengklik *Show SQL* untuk melihat pernyataan **SQL** yang akan digunakan untuk *startup* atau *shutdown*.



Gambar 30. Startup Database

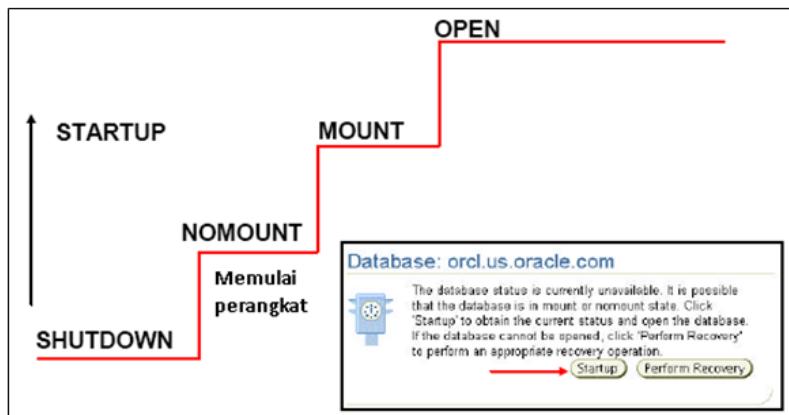
## 2.10. Memulai Sebuah Database

Ketika memulai database, Anda memilih kedudukan dimana akan dimulainya. Skenario berikut ini menjelaskan bagian – bagian yang berbeda dalam menjalankan perangkat:

- ✓ Memulai perangkat (**NOMOUNT**):

Perangkat dimulai dari bagian **NOMOUNT** hanya selama pembuatan database atau pembuatan ulang control file. Dalam memulainya ikutilah instruksi berikut ini:

- Membaca file inisialisasi dari **\$ORACLE\_HOME/ dbs** sesuai aturan:
  - Pertama adalah **spfileSID.ora**
  - Jika tidak ketemu maka selanjutnya adalah **spfile.ora**
  - Jika tidak ketemu maka selanjutnya adalah **initSID.ora**
- Menspesifikasi parameter **PFILE** dengan **STARTUP** memacu kelakuan default.
- Mengalokasikan SGA.
- Memulai proses background.
- Membuka file **alertSID.log** dan trace file – file.



Gambar 31. Status Database

- ✓ Memasuki database (**MOUNT**):

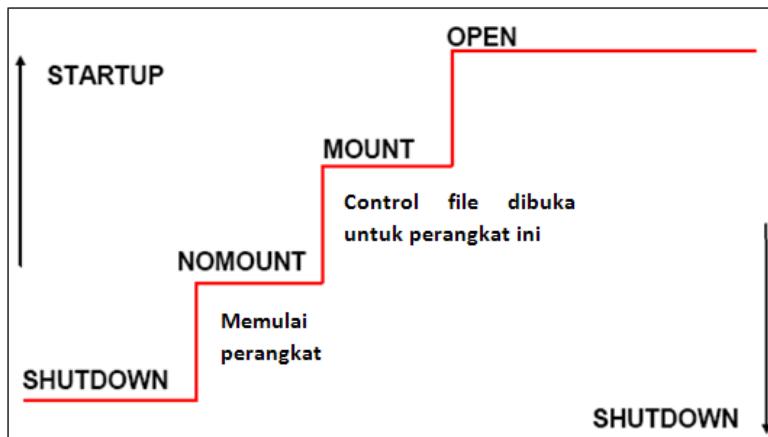
Untuk melakukan operasi – operasi pemeliharaan spesifik, Anda memulai perangkat dan memasuki database tetapi

tidak membukanya. Maksudnya dapat dilihat pada contoh – contoh berikut ini:

- Menamai kembali file – file data (file data untuk tablespace offline dapat dinamai kembali ketika database dibuka).
- Memungkinkan dan melarang pilihan ***online redo log file archiving***.
- Menampilkan recovery penuh database.

Memasuki database termasuk dalam tugas–tugas berikut ini:

- Menghubungkan database dengan perangkat yang sebelumnya telah dimulai.
- Menempatkan dan membuka control file yang dispesifikasikan dalam file parameter.
- Membaca control file untuk memperoleh nama dan status dari file data dan file online redo log. Meskipun demikian, tidak ada pemeriksaan untuk memverifikasi keberadaan file – file data dan file – file online redo log pada waktu itu.



Gambar 32. Status Database (2)

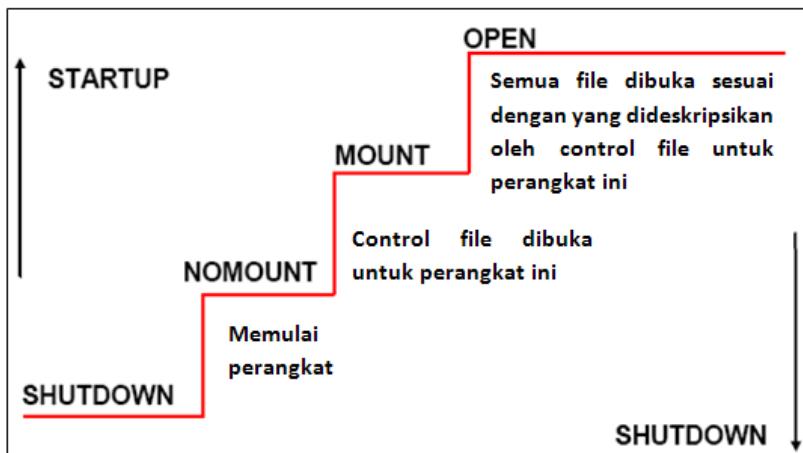
✓ Membuka database (OPEN):

Operasi normal database berarti bahwa perangkat sudah dijalankan dan database sudah dimasuki dan terbuka. Dengan operasi normal database, semua pengguna terdaftar dapat terkoneksi ke database dan melakukan operasi akses data tertentu. Membuka database termasuk dalam tugas – tugas berikut ini:

- Membuka file data online.
- Membuka file online redo log.

Jika semua file data online dan file online redo log tidak terbuka ketika Anda mencoba untuk membukanya, itu berarti server Oracle Anda bermasalah.

Selama bagian akhir ini, server Oracle memverifikasi semua file data online dan file online redo log yang dapat dibuka dan memeriksa konsistensi database. Jika perlu, proses background **SMON** akan menginisiasi recovery perangkat.



Gambar 33. Status Database (3)

## 2.11. Mematikan Database

Mematikan database untuk membuat backup sistem operasi offline dari semua struktur fisik dan untuk memodifikasi parameter inisialisasi statik akan menimbulkan efek ketika direstart.

Untuk mematikan perangkat Anda harus terkoneksi sebagai **SYSOPER** atau **SYSDBA** dan menggunakan perintah berikut ini:

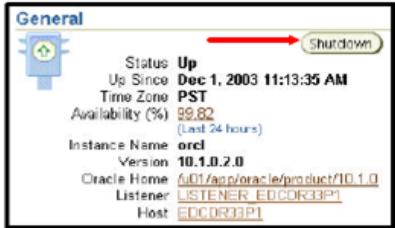
**SHUTDOWN [NORMAL|TRANSACTIONAL|IMMEDIATE|ABORT]**

Ini merupakan perintah yang akan dijalankan ketika Anda mengklik **Shutdown** dan mengkonfirmasikan perhatian Anda pada halaman berikutnya.

Model Shutdown	A	I	T	N
Mengijinkan koneksi baru	No	No	No	No
Menunggu sampai sesi yang sedang berlangsung berakhir	No	No	No	Yes
Menunggu sampai transaksi yang sedang berlangsung berakhir	No	No	Yes	Yes
Mewajibkan pemeriksaan dan menutup file	No	Yes	Yes	Yes

**Model Shutdown :**

- **A = ABORT**
- **I = IMMEDIATE**
- **T = TRANSACTIONAL**
- **N = NORMAL**



Gambar 34. Shutdown Database

Terdapat empat pilihan **SHUTDOWN**, yaitu:

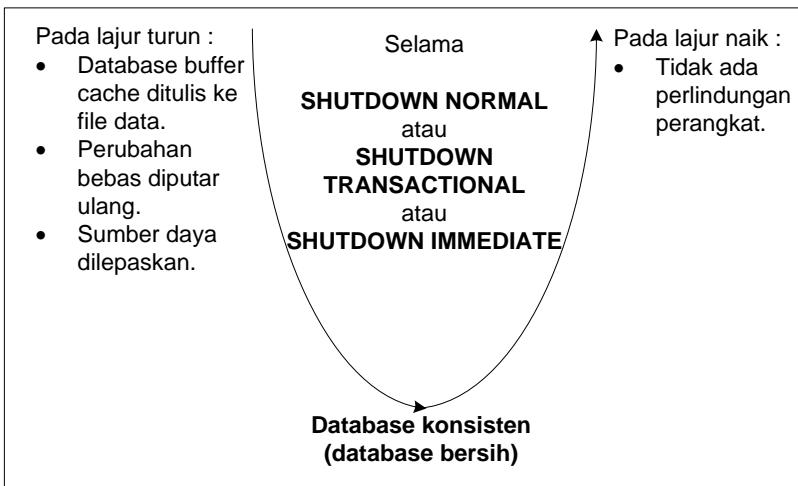
### 1. **SHUTDOWN NORMAL**

Normal adalah model default dari **SHUTDOWN**.

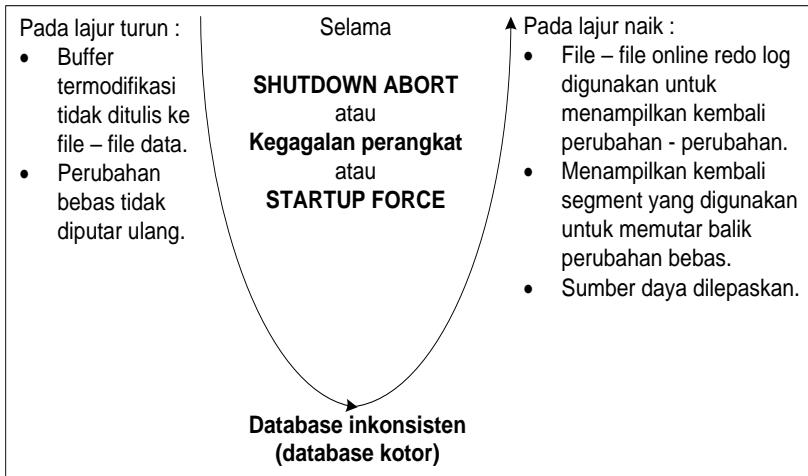
**SHUTDOWN NORMAL** database akan dijalankan dengan kondisi – kondisi berikut ini:

- Tidak ada koneksi baru yang dibuat.
  - Server Oracle menunggu semua pengguna untuk memutuskan hubungan sebelum menyelesaikan shutdown.
  - Database dan buffer redo ditulis di piringan.
  - Proses background berakhir dan SGA disingkirkan dari memori.
  - Oracle tertutup dan keluar dari database sebelum mematikan perangkat.
  - *Startup* berikutnya tidak meminta recovery perangkat.
2. **SHUTDOWN TRANSACTIONAL**  
**SHUTDOWN TRANSACTIONAL** mencegah pengguna dari kehilangan pekerjaan. **SHUTDOWN TRANSACTIONAL** dijalankan dengan kondisi sebagai berikut:
- Tidak ada klien yang dapat memulai transaksi baru pada perangkat utama.
  - Seorang klien akan terputus ketika klien mengakhiri transaksi yang sedang berlangsung.
  - Ketika semua transaksi sudah selesai, shutdown akan terjadi secepatnya.
  - *Startup* berikutnya tidak meminta recovery perangkat.
3. **SHUTDOWN IMMEDIATE**  
**SHUTDOWN IMMEDIATE** dijalankan dengan kondisi sebagai berikut:
- Pernyataan SQL tidak dapat diproses oleh Oracle dengan lengkap.
  - Server Oracle tidak menunggu pengguna, yang sedang terkoneksi pada database, untuk memutuskan hubungan.

- Oracle memutar balik transaksi aktif dan memutuskan hubungan semua pengguna yang sedang terkoneksi.
  - Oracle tertutup dan keluar dari database sebelum mematikan perangkat.
  - *Startup* berikutnya tidak meminta recovery perangkat.
4. **SHUTDOWN ABORT**  
Jika **SHUTDOWN NORMAL** dan **SHUTDOWN IMMEDIATE** tidak bekerja, Anda dapat membatalkan perangkat database yang ada. Membatalkan perangkat yang sedang berjalan dengan kondisi berikut ini:
- Pernyataan SQL yang sedang dijalankan oleh server Oracle tiba – tiba berhenti.
  - Oracle tidak menunggu pengguna yang terkoneksi pada database untuk memutuskan hubungan.
  - Database dan buffer redo tidak ditulis ke piringan.
  - Transaksi bebas tidak diputar ulang.
  - Perangkat berhenti tanpa menutup file.
  - Database tidak tertutup atau keluar.
  - *Startup* berikutnya meminta recovery perangkat, yang akan terjadi secara otomatis.
- Tidak disarankan untuk membackup database karena berada dalam kondisi yang inkonsisten.



Gambar 35. Status Database Shutdown



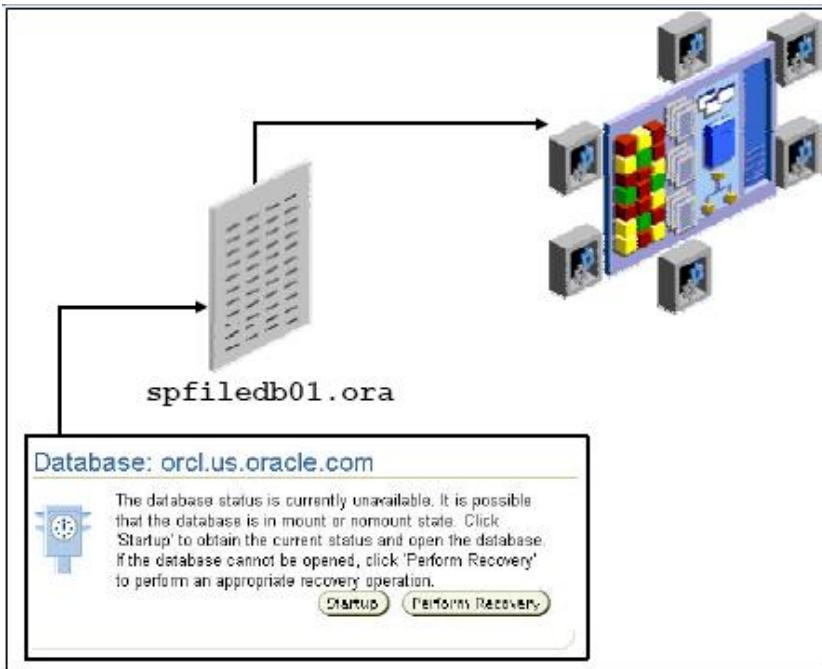
Gambar 36. Status Database Startup

## **2.12. File Parameter Inisialisasi**

Untuk memulai perangkat, Oracle harus membaca file parameter inisialisasi atau file parameter server. File – file ini terdiri atas sebuah daftar konfigurasi parameter untuk perangkat dan database. Oracle biasanya menyimpan parameter inisialisasi ke dalam sebuah file parameter inisialisasi teks. Anda juga dapat memilih untuk memelihara parameter – parameter inisialisasi ke dalam sebuah file parameter server biner (SPFILE).

Parameter inisialisasi disimpan dalam sebuah file parameter server secara tetap, dengan tujuan bahwa perubahan – perubahan yang dibuat pada parameter ketika perangkat sedang berjalan dapat tetap ada sewaktu perangkat dimatikan atau dinyalakan. Parameter inisialisasi terbagi dalam 2 grup: dasar dan ahli. Dalam kebanyakan kasus, perlu untuk mengatur dan menyetel hanya parameter dasar saja untuk mendapatkan kinerja yang masuk akal. Dalam kondisi jarang, modifikasi untuk parameter ahli mungkin diperlukan untuk kinerja yang optimal.

Oracle menyarankan Anda untuk membuat file parameter server (SPFILE) sebagai sesuatu yang dinamis dalam memelihara parameter inisialisasi. Dengan menggunakan sebuah file parameter server Anda dapat menyimpan dan mengatur parameter inisialisasi Anda secara tetap dalam sebuah file piringan sisi server.



Gambar 37. Parameter Inisiasi

### 2.13. Menampilkan Parameter Inisialisasi

Database Oracle menyediakan sejumlah parameter inisialisasi untuk mengoptimalkan operasinya dalam berbagai macam lingkungan. Hanya beberapa dari parameter yang harus diatur dengan jelas karena nilai default digunakan dalam kebanyakan kasus. Ada 28 buah parameter dasar.

Parameter ahli disediakan untuk mengijinkan DBAs ahli untuk mengadaptasi kebiasaan database Oracle untuk dapat memenuhi persyaratan tanpa menutupi yang lainnya yang tidak mempunyai persyaratan.

The screenshot shows the Oracle Enterprise Manager interface for a database named 'orcl.us.oracle.com'. The top navigation bar includes 'Home', 'Performance', 'Administration' (which is selected), and 'Maintenance'. Under the 'Administration' tab, there are two main sections: 'Instance' and 'Storage'. The 'Instance' section contains links for 'Memory Parameters', 'Undo Management', and 'All Initialization Parameters'. The 'Storage' section contains links for 'Controlfiles', 'Tablespaces', 'Datafiles', 'Rollback Segments', 'Redo Log Groups', 'Archive Logs', and 'Temporary Tablespace Groups'. Below these sections is a table titled 'Initialization Parameters'. The table has columns for 'Name', 'Help', 'Revisions', 'Value', 'Type', 'Basic', 'Default', 'Dynamic', and 'Category'. It lists parameters like 'cluster\_database' (Value: FALSE, Type: Boolean) and 'compatible' (Value: 10.1.0.1.0, Type: String). There are also entries for control files with multiple values. At the bottom of the table, there are buttons for 'Save to File', 'Show All', and navigation links for 'Previous' and 'Next 25'.

Gambar 38. Parameter Inisiasi (2)

## 2.14. Menampilkan Alert Log

Setiap database juga mempunyai **alert\_sid.log**. file ini berada dalam server dengan database dan disimpan dalam direktori yang dispesifikasikan dengan parameter inisialisasi **background\_dump\_dest**. File peringatan dari database adalah sebuah log kronologis dari pesan dan error, termasuk:

- ✓ Semua error internal (ORA-600), error blok korupsi (ORA-1578), dan error deadlock (ORA-60) yang terjadi.
- ✓ Operasi administrasi, seperti pernyataan SQL **CREATE**, **ALTER**, **DROP DATABASE**, **TABLESPACE**, **ROLLBACK SEGMENT** dan *Enterprise Manager* atau pernyataan **SQL\*Plus** **STARTUP**, **SHUTDOWN**, **ARCHIVE LOG**, dan **RECOVER**.
- ✓ Beberapa pesan dan error yang berhubungan dengan fungsi dari server bersama dan kecepatan proses.
- ✓ Error selama penyegaran otomatis dari tampilan materi.

**EM** mengawasi file alert log dan memperingatkan Anda tentang error kritis. Anda juga dapat menampilkan log untuk melihat error nonkritis dan pesan informasi. File juga dapat berkembang menjadi ukuran yang tidak teratur jika ditinggalkan sendirian. Anda harus membuat backup dari file peringatan sesering mungkin dan menghapus file peringatan yang terjadi. Ketika database mencoba untuk menulis file peringatan kembali, database kemudian akan membuat ulang sebuah file peringatan yang baru.

The screenshot shows the Oracle Enterprise Manager interface. At the top, there's a navigation bar with 'Related Links' containing 'Advisor Central', 'All Metrics', 'Jobs', and 'Monitoring Configuration'. To its right are links for 'Alert History', 'Blackouts', 'Manage Metrics', and 'User-Defined Metrics'. A red arrow points from the 'Alert History' link to the 'Alert Log Entries' link in the 'Related Links' bar. Below this is another red arrow pointing down to the 'Alert Log Entries' page. The main content area has a header 'Database: orcl > Most Recent Alert Log Entries' and a sub-header 'Most Recent Alert Log Entries'. It displays the last 100,000 bytes of the alert log, which is constantly growing. The log entries show database operations like renaming databases and tablespaces, and setting default datafile formats. The page footer includes 'Copyright © 1998, 2004, Oracle. All rights reserved.' and 'About Oracle Database Manager 10g Database Control'.

```
alter database rename global_name to "orcl"
Completed: alter database rename global_name to "orcl"
Mon Jan 5 12:16:53 2004
ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracle/oradata/orcl/temp01.dbf' SIZE 20480K R
Mon Jan 5 12:16:53 2004
Setting default datafile format ID for platter 0
Mon Jan 5 12:16:53 2004
Completed: ALTER TABLESPACE TEMP ADD TEMPFILE '/u01/app/oracl
Mon Jan 5 12:16:53 2004
ALTER DATABASE DEFAULT TABLESPACE TBS001"
```

Gambar 39. Alert Log

## BAB III

# STRUKTUR PENYIMPANAN

### 3.1. Tujuan

Setelah menyelesaikan pelajaran ini, Anda diharapkan dapat melakukan hal – hal sebagai berikut:

- ✓ Mendefinisikan kegunaan dari tablespace dan file data.
- ✓ Membuat tablespace.
- ✓ Mengatur tablespace.
- ✓ Memperoleh informasi tablespace.

### 3.2. Tablespace dan File Data

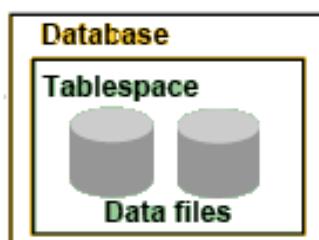
Database, tablespace dan file data saling berhubungan erat, tetapi bagian tersebut mempunyai perbedaan yang penting, yaitu:

- ✓ Database Oracle terdiri atas satu atau lebih unit – unit penyimpanan yang dinamakan tablespace, yang secara kolektif menyimpan semua data database.
- ✓ Setiap tablespace dalam database Oracle terdiri atas satu atau lebih file yang dinamakan file data, yang merupakan struktur fisik yang berkaitan dengan sistem operasi ketika Oracle sedang berjalan.
- ✓ Sebuah data database secara kolektif disimpan dalam file data yang mengatur setiap tablespace dari database. Sebagai contoh, database Oracle termudah akan mempunyai satu tablespace dan satu file data. Database yang lain dapat

mempunyai tiga tablespace, yang masing – masingnya terdiri atas dua file data (untuk totalnya ada enam file data). Sebuah database tunggal potensialnya mempunyai 65.535 buah file data.

Intinya adalah Oracle menyimpan data logika dalam tablespace dan data fisik dalam file data dimana:

- ✓ Tablespace:
  - Dapat hanya termasuk dalam satu database pada waktu tersebut.
  - Terdiri dari satu atau lebih file data.
  - Dibagi dalam unit – unit penyimpanan logika.
- ✓ File data:
  - Dapat hanya termasuk dalam satu database dan satu tablespace.
  - Merupakan gudang penyimpanan untuk skema data obyek.



Gambar 40. Datafile

### 3.3. Pengaturan Ruangan dalam Tablespace

Tablespace mengalokasikan ruangan yang luas. Tablespace dapat dibuat dengan menggunakan salah satu dari dua metode yang berbeda berikut ini:

- ✓ Pengaturan tablespace lokal.  
Pengaturannya dilakukan dalam tablespace melalui bitmap. Setiap bit dalam bitmap berhubungan dengan sebuah blok

atau sekelompok blok. Ketika sebuah ruang teralokasi atau bebas digunakan, server Oracle mengubah nilai bitmap untuk menunjukkan status baru dari blok.

- ✓ Pengaturan tablespace kamus.

Pengaturannya dilakukan oleh kamus data. Server Oracle mengupdate tabel – tabel dalam kamus data ketika sebuah ruang teralokasi atau terdisalokasi. Hal ini untuk cadangan saja, Anda harus menggunakan pengaturan lokal untuk semua tablespace.



Gambar 41. Extent Management

### 3.4. MEMBUAT TABLESPACE BARU

Untuk membuat sebuah tablespace, ikuti langkah – langkah berikut ini:

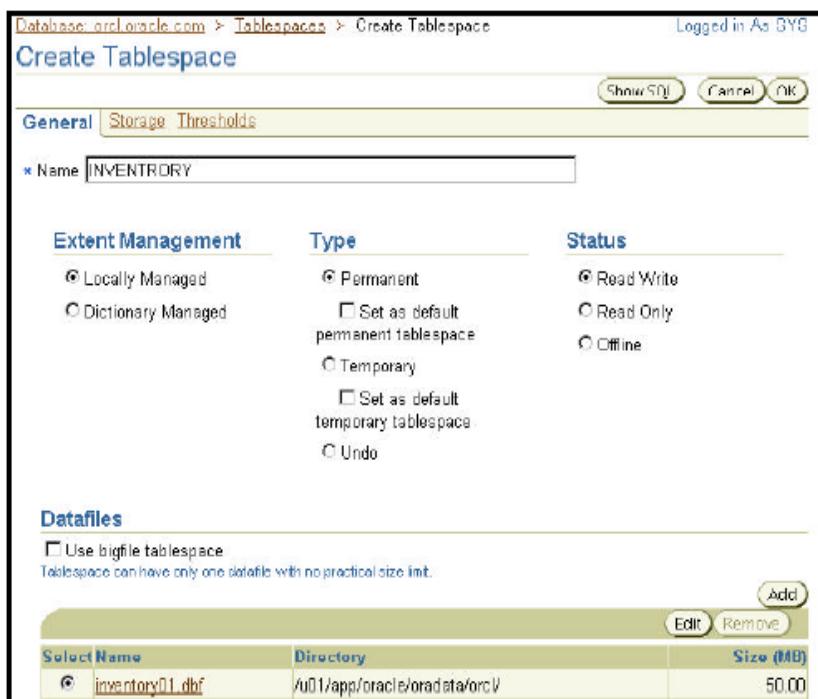
1. Masuklah pada halaman **Tablespace**. Pergi ke tab **Administration**, kemudian klik **Tablespaces** di bawah kepala **Storage**.
2. Klik tombol **Create**.  
**CATATAN:** jika Anda ingin membuat sebuah tablespace seperti sebuah tablespace yang sudah ada, maka pilih sebuah tablespace yang sudah ada dan pilih **Create Like** dari menu **Actions**. Klik **Go**. Halaman **The Create Tablespace General** akan muncul.
3. Masukkan sebuah nama untuk tablespace.
4. Di bawah kepala **Extent Management**, pilih **Locally Managed**. Ruangan dari sebuah pengaturan lokal tablespace diatur secara efisien dalam tablespace oleh server

database Oracle. Untuk sebuah pengaturan tablespace kamus Anda harus lebih aktif dalam mengatur ruangan dan akses kamus data yang diminta untuk membukanya. Pengaturan tablespace kamus menjadi tidak berarti. Oracle tidak menyarankan penggunaannya.

5. Di bawah kepala **Type**, pilih **Permanent**. Tablespace **Permanent** menyimpan obyek database secara permanen yang dibuat oleh sistem atau pengguna.
6. Di bawah kepala **Status**, pilih **Read Write**. Status baca / tulis berarti pengguna dapat membaca dan menulis pada tablespace setelah dibuat. Hal ini merupakan default.
7. Di area **Datafiles** pada halaman tersebut klik **Add** untuk menambah file data pada tablespace, sebuah tablespace harus mempunyai paling sedikit satu file. File besar tablespace digunakan dengan database ultra besar pada **Oracle's Automatic Storage Management** atau pendukung pengaturan volume logika yang lainnya ada atau RAID, dan volume logika ruangan yang dinamis.
8. Dalam halaman **Add Datafiles**, masukkan sebuah nama file. Terima default untuk **File Directory** dan **File Size**.
9. Di bawah area **Storage**, pilih “**Automatically extend datafile when full (AUTOEXTEND)**” dan spesifikasikan sebuah bilangan dalam daerah **Increment** yang Anda inginkan untuk ruangan file data setiap waktu bila akan diisi. Tinggalkan pengaturan **Maximum File Size** ke **Unlimited**. Klik **OK**. Anda kembali ke halaman **Create Tablespace General**.
10. Klik tab **Storage**. Halaman **Create Tablespace Storage** akan muncul.
11. Terima semua default dalam halaman **Storage**.
12. Klik tab **Tresholds** untuk membuka halaman **Tresholds**. Halaman ini memungkinkan Anda untuk mengatur pengawasan tresholds untuk penggunaan ruang. Anda

- menerima saran dan sebuah pilihan untuk aksi ketika treshold tercapai.
13. Setelah menspesifikasi treshold klik **OK** untuk menambah tablespace. Anda kembali pada halaman **Tablespace** dimana Anda akan menerima sebuah konfirmasi dari pembuatan tablespace. Anda dapat menampilkan tablespace baru Anda dalam bagian **Result**.

**Catatan:** langkah – langkah tersebut dijelaskan pada Anda agar Anda dapat mengerti betapa cepatnya membuat sebuah tablespace dalam kebanyakan situasi. Anda mungkin membutuhkan beberapa perubahan pilihan tergantung pada permintaan sistem Anda.



Gambar 42. Pembuatan tablespace (1)

### 3.5. Penyimpanan Pengaturan Tablespace Lokal

Ruangan dalam sebuah pengaturan tablespace lokal dapat dialokasikan dalam salah satu cara berikut ini:

✓ **Automatic:**

Biasa juga disebut **autoallocate**, menspesifikasikan ukuran ruang dalam tablespace yang diatur sistem. Anda tidak dapat menspesifikasikan sebuah ukuran ruang. Anda tidak dapat menspesifikasikan otomatis untuk sebuah tablespace tidak tetap.

✓ **Uniform:**

Menspesifikasikan tablespace yang diatur dengan ruang uniform dari sebuah ukuran yang Anda spesifikasikan. Ukuran defaultnya adalah 1 MB. Semua ruang pada tablespace tidak tetap adalah berukuran uniform, jadi hal ini optional untuk sebuah tablespace tidak tetap. Anda tidak dapat menspesifikasi uniform untuk sebuah pengulangan kembali tablespace.

Pengaturan ruang segment dalam sebuah pengaturan tablespace lokal:

✓ **Automatic:**

Oracle menggunakan bitmap untuk mengatur ruang kosong dalam segment. Sebuah bitmap, dalam kasus ini, merupakan sebuah peta yang menggambarkan status dari setiap blok data dalam sebuah segment dengan melihat pada jumlah ruang dalam blok yang tersedia untuk menyelipkan baris. Baik itu banyak atau pun sedikit ruang yang tersedia dalam sebuah blok data, akan terefleksikan dalam bitmap. Bitmap mengijinkan Oracle untuk mengatur ruang bebas lebih otomatis lagi, dan selain itu, bentuk dari pengaturan ruang tersebut dinamakan pengaturan ruang segment otomatis.

Penyimpanan untuk pengaturan tablespace lokal:

✓ **Manual:**

Memberitahukan Oracle bahwa Anda ingin menggunakan daftar bebas untuk mengatur ruang bebas dalam segment.

Daftar bebas adalah daftar blok data yang mempunyai ruang yang tersedia untuk menyelipkan baris. Bentuk ini dinamakan pengaturan ruang segment manual karena kebutuhan untuk menspesifikasikan dan menyalakan parameter penyimpanan **PCTUSED**, **FREELISTS**, dan **FREELIST GROUPS** untuk obyek skema yang dibuat dalam tablespace. Hal ini merupakan pendukung untuk pengadaan cadangan.

Keuntungan dari pengaturan tablespace lokal:

- ✓ Pengaturan lokal menghindari operasi pengaturan ruang rekursif. Hal ini dapat terjadi pada pengaturan tablespace kamus jika mengkonsumsi atau melepaskan ruang pada sebuah ruang hasil pada operasi yang lain yang mengkonsumsi atau melepaskan ruang pada sebuah segment undo atau tabel kamus data.
- ✓ Karena pengaturan tablespace lokal tidak menyimpan ruang kosong pada tabel kamus data, bagian tersebut mengurangi isi pada tabel.
- ✓ Pengaturan lokal dari jalur ruang otomatis berdekatan dengan ruang bebas, membatasi kebutuhan untuk penggabungan ruang bebas.
- ✓ Ukuran ruang yang diatur secara lokal dapat dibatasi secara otomatis oleh sistem.
- ✓ Perubahan pada ruang bitmap tidak menghasilkan informasi undo karena bagian tersebut tidak mengupdate tabel pada kamus data (kecuali untuk kasus tertentu seperti informasi kuota tablespace).

**CATATAN:** jika Anda mengatur sebuah database yang mempunyai pengaturan tablespace kamus dan Anda ingin untuk mengubah bagian tersebut pada pengaturan lokal gunakanlah prosedur

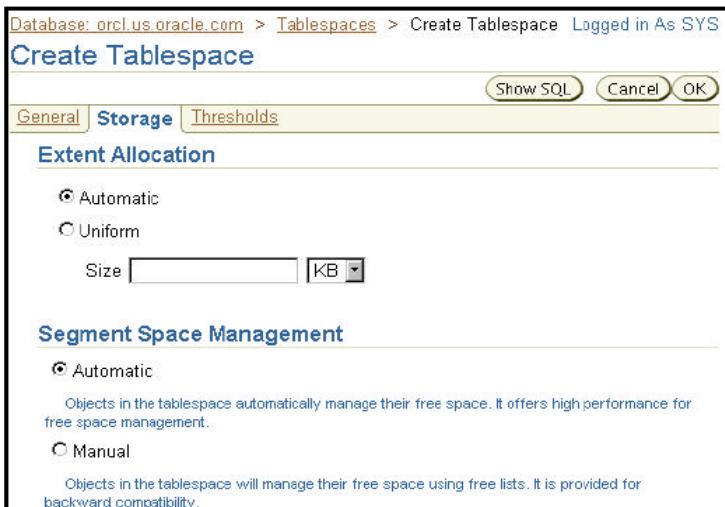
**DBMS\_SPACE\_ADMIN.TABLESPACE\_MIGRATE\_TO\_LOCAL** untuk melakukannya.

***Logging:***

Ketika perubahan dibuat pada obyek dalam tablespace, perubahan tersebut akan terlog dalam apa yang disebut sebagai redo stream. Redo stream dimulai dalam memori, ditulis pada file online redo log dan mungkin ditulis untuk mendapatkan file log. Anda dapat mengubah proses ini menjadi off. Jika anda mematikan logging, obyek dalam tablespace akan tidak terlindungi jika terjadi kegagalan.

***Tresholds:***

Gunakan tab **Tablespace Tresholds** untuk mengatur ruang yang digunakan tresholds untuk tablespace database yang ada. Anda dapat memilih untuk menggunakan penggunaan tresholds default untuk tablespace untuk database atau Anda dapat menspesifikasikan treshold untuk persentase penggunaan ruang yang digunakan untuk tablespace yang ada dengan memasukkan nilai untuk daerah **Warning and Critical**. Anda juga dapat menolak ruang yang menggunakan tresholds untuk tablespace.



Gambar 43. Pembuatan Tablespace (2)

### 3.6. Tablespace dalam Prekonfigurasi Database

Tablespace berikut ini dibuat untuk Anda dalam database prekonfigurasi:

✓ **SYSTEM:**

Tablespace **SYSTEM** digunakan oleh server database Oracle untuk mengatur database, mengandung kamus data dan tabel yang terdiri atas informasi administrasi tentang database. Semuanya terkandung dalam skema **SYS**, dan dapat diakses hanya oleh pengguna **SYS**, atau pengguna administrasi lainnya dengan ijin tertentu.

✓ **SYSAUX:**

Merupakan tablespace pembantu untuk tablespace **SYSTEM**. Beberapa komponen dan produk yang menggunakan tablespace **SYSTEM** atau tablespace itu sendiri dalam pelepasan utama Oracle, sekarang menggunakan tablespace **SYSAUX**. Setiap Oracle Database 10g atau tingkatan database yang tertinggi sekalipun harus mempunyai sebuah sebuah tablespace **SYSAUX**.

✓ **TEMP:**

Tablespace ini digunakan untuk menyimpan tabel dan indeks tidak tetap ketika memproses pernyataan SQL. Jika dapat, untuk contoh, dapat digunakan untuk ruang kerja pendek. Setiap database harus mempunyai sebuah tablespace tidak tetap yang ditugaskan untuk pengguna sebagai tablespace tidak tetap bagian tersebut. Dalam prekonfigurasi database, tablespace **TEMP** dispesifikasikan sebagai tablespace tidak tetap default. Hal ini berarti jika tidak ada tablespace tidak tetap yang dispesifikasikan ketika account pengguna dibuat, kemudian Oracle akan menugaskan tablespace tersebut kepada pengguna.

✓ **UNDOTBS1:**

Merupakan undo tablespace yang digunakan oleh server database untuk menyimpan informasi undo. Setiap database harus mempunyai sebuah undo tablespace yang dibuat selama pembuatan database.

✓ **USERS:**

Tablespace ini digunakan untuk menyimpan obyek pengguna tetap dan data. Dalam database prekonfigurasi, tablespace **USERS** merupakan tablespace default untuk semua obyek yang dibuat oleh pengguna non sistem. Untuk pengguna **SYS** dan **SYSTEM** (pengguna sistem), tablespace default tetap mengacu pada **SYSTEM**.

✓ **EXAMPLE:**

Tablespace ini mengandung skema contoh yang dapat diinstalasi ketika Anda membuat database. Skema contoh menyediakan sebuah platform umum untuk contoh. Dokumentasi Oracle dan kursusnya mengandung contoh yang berdasarkan pada skema contoh.

Select	Name	Type	Extent Management	Segment Management	Status	Size (MB)	Used (MB)	Used (%)
<input checked="" type="radio"/>	EXAMPLE	PERMANENT	LOCAL	AUTO	ONLINE	150.000	66.875	44.55
<input checked="" type="radio"/>	SYSAUX	PERMANENT	LOCAL	AUTO	ONLINE	230.000	222.688	96.82
<input checked="" type="radio"/>	SYSTEM	PERMANENT	LOCAL	MANUAL	ONLINE	440.000	434.375	98.72
<input checked="" type="radio"/>	TEMP	TEMPORARY	LOCAL	MANUAL	ONLINE	26.000	25.000	96.15
<input checked="" type="radio"/>	UNDOTBS1	UNDO	LOCAL	MANUAL	ONLINE	25.000	11.750	47.00
<input checked="" type="radio"/>	USERS	PERMANENT	LOCAL	AUTO	ONLINE	5.000	2.750	55.00

Gambar 44. Pembuatan Tablespace (3)

### 3.7. Mengubah Tablespace

Setelah Anda membuat sebuah tablespace, Anda kemudian dapat mengubahnya dalam beberapa cara sebagai perubahan kebutuhan dari sistem Anda.

Menamakan kembali: Masukkan sebuah nama baru untuk tablespace dan klik **Apply**.

Mengubah **Status**: Sebuah tablespace dapat berada pada satu atau tiga status atau kedudukan yang berbeda. Berdasarkan jenis dari tablespace, tidak semua kedudukan mungkin tersedia:

- ✓ **Read Write:**

Tablespace online dan dapat dibaca dari dan ditulis kepada.

- ✓ **Read Only:**

Menspesifikasikan read – only untuk menempatkan tablespace dalam model transisi read – only. Dalam kedudukan ini, transaksi yang ada dapat dilengkapi (tetap atau memutar ulang), tetapi tidak semua operasi DML diijinkan untuk tablespace kecuali untuk memutar ulang transaksi yang sudah ada yang sebelumnya dimodifikasi blok dalam tablespace. Tablespace online ketika berada dalam kedudukan read – only. Anda tidak dapat membuat tablespace read – only **SYSTEM** atau **SYSAUX**.

✓ **Offline:**

Anda dapat membuat tablespace online menjadi offline jadi porsi dari database tersebut secara titik tetap dapat tidak tersedia untuk penggunaan secara umum. Sisa database terbuka dan tersedia bagi pengguna untuk mengakses data. Ketika Anda membuatnya menjadi offline maka Anda mempunyai kesempatan untuk melakukan hal – hal berikut ini:

- **Normal:**

Normalnya sebuah tablespace dapat dibuat offline jika terjadi kondisi error untuk file data dari tablespace. Oracle membuat sebuah checkpoint untuk semua file data dari tablespace seperti halnya membuat bagian tersebut offline.

- **Temporary:**

Sebuah tablespace dapat dibuat offline secara tidak tetap, meskipun terdapat kondisi error untuk satu atau lebih file dari tablespace. Oracle membuat offline file data yang tidak sedang offline, menandai bagian tersebut seperti biasanya. Jika tidak ada file yang offline, tetapi Anda menggunakan klausa tidak tetap, media recovery tidak diperlukan untuk membawa tablespace kembali online. Meskipun demikian, jika satu atau lebih file dari tablespace offline karena penulisan error, dan Anda mengambil tablespace offline tidak tetap, tablespace meminta recovery sebelum Anda dapat membawanya kembali online.

- **Immediate:**

Sebuah tablespace dapat dibuat offline secepatnya, tanpa Oracle membuat sebuah checkpoint pada file data. Ketika Anda menspesifikasi **Immediate**, media recovery untuk tablespace diminta sebelum tablespace dapat kembali online. Anda tidak dapat membuat

- tablespace offline secepatnya jika database aktif dalam model **NOARCHIVELOG**.
- **For Recover:**  
Pengaturan **FOR RECOVER** ditolak. Sintaks hanya untuk mendukung kompatibilitas cadangan.

Mengubah ukuran: Anda dapat menambahkan spasi untuk tablespace yang ada dengan menambah file data pada tablespace atau Anda dapat mengubah ukuran dari file data yang ada:

- ✓ Untuk menambah sebuah file data baru pada tablespace klik **Add**, dan isi dalam informasi tentang file data pada halaman **Add Data File**. Catatlah bahwa nama tablespace sudah tetap.
- ✓ Untuk mengubah ukuran dari file data yang sudah ada, pilih file data dalam area **Datafiles** dari halaman **Edit Tablespace** dengan mengklik nama dari file data, atau pilih file data dan klik **Edit**. Kemudian pada halaman **Edit Datafile** Anda dapat mengubah ukuran file data. Anda dapat membuat tablespace lebih besar atau lebih kecil. Meskipun demikian Anda tidak dapat membuat sebuah file data lebih kecil dari ruang yang digunakan dalam file, jika Anda tetap mencobanya maka Anda akan mendapatkan error sebagai berikut:

**ORA-03297: file contains used data beyond requested RESIZE value**

Pilihan penyimpanan: Klik **Storage** untuk mengubah kebiasaan logging dari tablespace.

Thresholds: Klik **Thresholds** untuk dapat mengubah peringatan dan kritik perubahan penggunaan ruang untuk tablespace. Anda mempunyai tiga pilihan:

- ✓ **Use Default Thresholds:**  
Menggunakan default preset, dan Anda mempunyai pengaturan default.

- ✓ **Specify Thresholds:**  
Mengijinkan Anda untuk mengatur thresholds untuk bagian tablespace tersebut.
- ✓ **Disable Thresholds:**  
Mematikan perubahan penggunaan ruang untuk tablespace tersebut.



Gambar 45. Pembuatan Tablespace (4)

### 3.8. Aksi dengan Tablespace

Dengan **Actions Menu** Anda dapat melakukan berbagai macam tugas dengan tablespaces Anda. Pilih sebuah tablespace dan kemudian Anda dapat melakukan aksi – aksi sebagai berikut:

- ✓ **Add Datafile:**

Menambah sebuah file data ke tablespace, yang mana membuat tablespace bertambah besar.

✓ **Create Like:**

Membuat tablespace yang lain menggunakan tablespace sebagai templet.

✓ **Generate DDL:**

Mengerjakan pernyataan DDL yang membuat tablespace. Dapat di-copy paste ke dalam sebuah file teks untuk digunakan sebagai sebuah skrip atau untuk penggunaan dokumentasi.

✓ **Make Locally Managed:**

Jika tablespace sudah diatur secara kamus, maka akan mengubah tablespace ke dalam pengaturan lokal.

✓ **Make Readonly:**

Menghentikan semua penulisan ke tablespace. Transaksi yang ada diijinkan untuk diselesaikan tetapi tidak ada DML baru atau kegiatan penulisan yang lain yang diijinkan untuk dimulai pada tablespace.

✓ **Make Writeable:**

Mengijinkan DML dan kegiatan penulisan yang lainnya dapat diinisiasi pada obyek dalam tablespace.

✓ **Place Online:**

Jika tablespace dalam keadaan offline, maka akan membawanya kembali online.

✓ **Reorganize:**

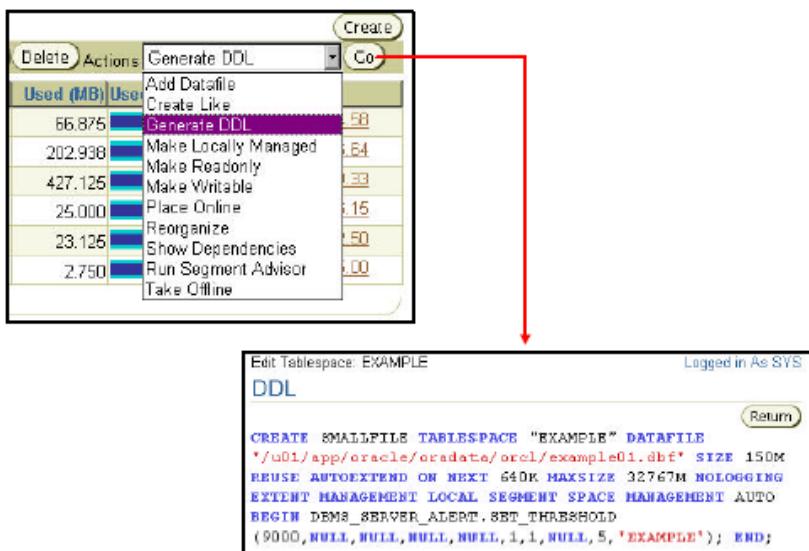
Memulai **Reorganization Wizard**, yang mana Anda dapat gunakan untuk memindahkan obyek di dalam tablespace untuk mengklaim kembali ruang yang mungkin tidak digunakan. Ini merupakan tugas yang harus dikerjakan selama penggunaan yang tidak dalam kondisi puncak dari obyek dalam tablespace.

✓ **Show Dependencies:**

Menunjukkan obyek dimana tablespace bergantung pada atau obyek yang bergantung pada tablespace.

- ✓ **Run Segment Advisor:**  
Segment Advisor membantu Anda untuk memutuskan apakah sebuah obyek mempunyai ketersediaan ruang untuk reklamasi berdasarkan tingkatan dari fragmentasi ruang dalam obyek. Pada tingkatan tablespace, saran dilakukan untuk setiap segment dalam tablespace.
- ✓ **Take Offline:**  
Jika tablespace dalam keadaan online maka akan membuat tablespace menjadi tidak tersedia. Tablespace tidak dihapus atau hilang, hanya tidak tersedia.

**CATATAN:** Tidak semua aksi tersedia untuk setiap tablespace. Tergantung pada jenis tablespace yang terpilih, beberapa aksi tidak dapat dilakukan. Sebagai contoh, Anda tidak dapat membuat tablespace **SYSTEM** offline, atau pun membuat sebuah tablespace undo – only.



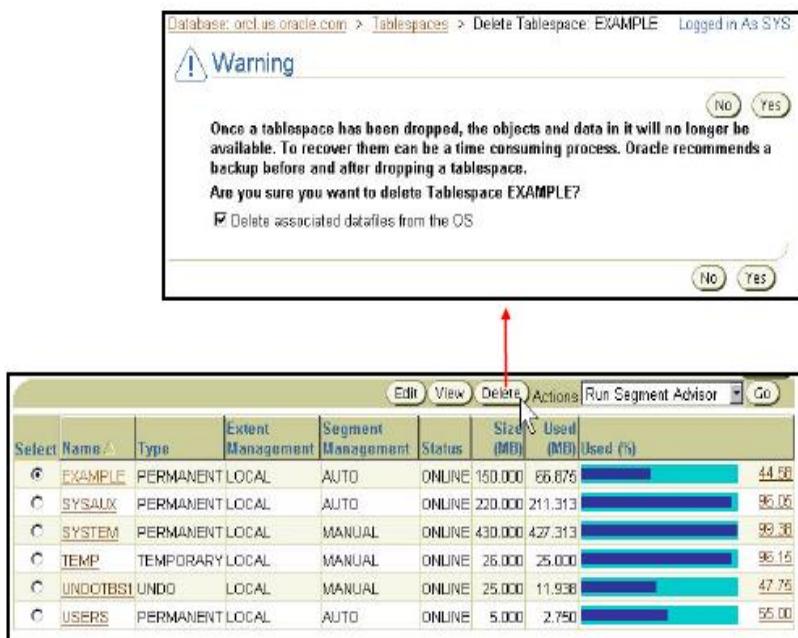
Gambar 46. Pembuatan Tablespace (5)

### **3.9. Menghilangkan Tablespace**

Anda dapat menghilangkan sebuah tablespace dan isinya (segment – segment yang ada dalam tablespace) dari database jika tablespace dan isinya tidak lagi diperlukan. Anda harus memiliki ijin sistem **DROP TABLESPACE** untuk menghilangkan sebuah tablespace.

Ketika Anda menghilangkan sebuah tablespace, penunjuk file dalam file kontrol dari sekumpulan database juga dihilangkan. Anda dapat secara optional mengarahkan Oracle untuk menghapus file sistem operasi (file data) yang memerintahkan penghilangan tablespace. Jika anda tidak mengarahkan Oracle untuk menghapus file data pada waktu yang bersamaan maka akan menghapus tablespace, Anda kemudian nantinya harus menggunakan perintah yang sesuai dari sistem operasi Anda untuk menghapus bagian tersebut.

Anda tidak dapat menghilangkan sebuah tablespace yang mengandung segment yang aktif. Sebagai contoh, jika sebuah tabel dalam tablespace sedang digunakan atau tablespace mengandung data undo yang diperlukan untuk memutar kembali transaksi bebas, maka Anda tidak dapat menghilangkan tablespace. Tablespace dapat menjadi online atau offline, tetapi adalah lebih baik untuk membuat tablespace offline sebelum menghilangkannya.



Gambar 47. Pembuatan Tablespace(6)

### 3.10. Menampilkan Informasi Tablespace

Klik **View** untuk melihat informasi tentang tablespace terpilih. Dalam halaman **View Tablespace**, Anda dapat juga mengklik **edit** untuk mengubah tablespace.

Memperoleh tablespace dan informasi file data dapat juga diperoleh dengan melakukan hal – hal berikut ini:

- ✓ Informasi tablespace:
  - **DBA\_TABLESPACES**
  - **V\$TABLESPACE**
- ✓ Informasi file data:
  - **DBA\_DATA\_FILES**
  - **V\$DATAFILE**

- ✓ Informasi file temp:
  - DBA\_TEMP\_FILES
  - V\$TEMPFILE

Database: orcl.oracle.com > Tablespaces >  
View Tablespace: EXAMPLE      Logged in As SYS

**View Tablespace: EXAMPLE**

Name **EXAMPLE**  
Bigfile tablespace **No**  
Status **ReadWrite**  
Type **Permanent**  
Extent Management **Local**

**Storage**

Allocation Type **Automatic**  
Segment Space Management **Automatic**  
Enable logging **No**  
Block Size (B) **8192**

**Datafiles**

Name	Directory	Size (MB)	Used (MB)
example01.dbf	/u01/app/oracle/oradata/orcl/	150.00	80.25

**Thresholds**

Use Default Thresholds  
Warning (% used) **85**  
Critical (% used) **95**

**Actions**

Select	Name ▾	Type	Extent Management	Segment Management	Status	Size (MB)	Used (MB)	User Actions
<input checked="" type="radio"/>	EXAMPLE	PERMANENT LOCAL	AUTO	AUTO	ONLINE	150.000	80.250	<a href="#">Edit</a> <a href="#">View</a> <a href="#">Delete</a> <a href="#">Actions</a>

Gambar 48. Pembuatan Tablespace(7)



## BAB IV

# ADMINISTRASI PENGGUNA

### 4.1. Tujuan

Setelah menyelesaikan pelajaran ini, Anda diharapkan dapat melakukan hal – hal sebagai berikut:

- ✓ Membuat dan mengatur account pengguna database.
- ✓ Membuat dan mengatur peran.
- ✓ Memberikan dan membatalkan ijin.
- ✓ Mengendalikan penggunaan sumber daya oleh pengguna.

### 4.2. Account Pengguna Database

Untuk mengakses database seorang pengguna harus menspesifikasi sebuah account pengguna database tertentu dan secara sukses dikenali sebagai account pengguna yang otentik.

Dalam beberapa sistem, setiap pengguna database mempunyai account databasenya sendiri. Yang lainnya, banyak pengguna saling berbagi account database. Apapun juga sistem Anda, setiap account pengguna akan memiliki:

- ✓ Sebuah nama pengguna yang unik.  
Nama pengguna tidak boleh lebih dari 30 karakter, tidak bisa mengandung karakter spesial, dan harus diawali dengan sebuah huruf.
- ✓ Sebuah metode otentik.  
Metode otentik pada umumnya berupa kata kunci, tetapi Oracle Database 10g mendukung beberapa metode lainnya termasuk biometrik, sertifikat, dan lain-lain.

- ✓ Sebuah tablespace default.  
Sebuah tempat dimana pengguna akan membuat obyek jika pengguna tidak menspesifikasi beberapa tablespace lainnya. Ingatlah bahwa mempunyai sebuah tablespace default tidak berarti pengguna mempunyai ijin untuk membuat obyek dalam tablespace atau pun juga sebuah kuota ruang dalam tablespace untuk maksud membuat obyek. Keduanya diijinkan secara terpisah.
- ✓ Sebuah tablespace tidak tetap.  
Sebuah tempat dimana pengguna dapat membuat obyek tidak tetap seperti tabel pendek dan tidak tetap.
- ✓ Sebuah profil pengguna.  
Pengaturan pembatasan sumber daya dan kata kunci yang ditugaskan kepada pengguna.
- ✓ Mengendalikan penggunaan sumber daya oleh pengguna.

### 4.3. Menciptakan Pengguna

Menciptakan seorang pengguna:

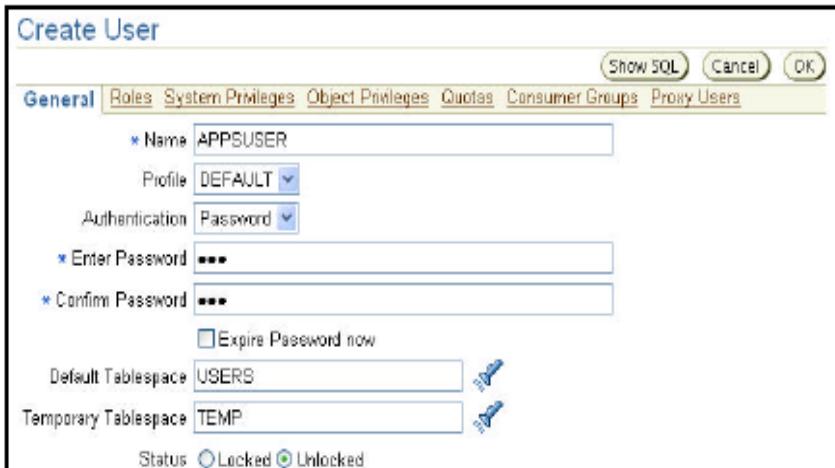
- ✓ Dalam **Oracle Enterprise Manager** Anda dapat mengatur daftar dari pengguna database yang diijinkan untuk mengakses database yang ada melalui lembar properti **Users**. Anda dapat menggunakan lembar properti untuk membuat, menghapus, dan memodifikasi pengaturan dari seorang pengguna.
- ✓ Lembar properti **Users** terdiri atas beberapa halaman. Anda dapat membuat atau mengedit pengaturan keamanan untuk seorang pengguna database dengan cara memindahkan halaman lain yang berkaitan dengan lembar properti **Users**. Klik **Help** untuk informasi tentang halaman tertentu.

Untuk menciptakan pengguna database:

1. Arahkan pada home page **Database**.
2. Pada halaman properti **Administration** di bawah **Security**, klik **Users**. Halaman **Users Search** muncul

mengandung sebuah tabel hasil yang mendaftarkan semua pengguna yang ada untuk database.

3. Klik **Create** untuk membuat pengguna baru.



Gambar 49. Pembuatan User (1)

#### 4.4. Profil dan Pengguna

Profil menggunakan sebuah nama pengaturan batas sumber daya pada penggunaan database dan sumber daya perangkat. Profil juga menempatkan pembatasan kata kunci pengguna (panjang, waktu kadaluarsa, dan lain-lain). Semua pengguna ditugaskan pada profil dan mungkin termasuk pada satu profil pada suatu waktu.

Profil default disediakan sebagai basis dari semua profil yang lainnya. Pembatasan untuk sebuah profil dapat secara implisit dispesifikasikan (seperti di CPU / Session), tidak terbatas (seperti di CPU / Call) atau referensi pengaturan apapun pada profil default (seperti di Connect Time).

Profil tidak dapat menggunakan pembatasan sumber daya pada pengguna jika inisialisasi parameter **RESOURCE\_LIMIT**

tidak diatur ke **TRUE**. Dengan **RESOURCE\_LIMIT** sebagai nilai default dari **FALSE**, pembatasan limit diabaikan.

Profil mengijinkan administrator untuk mengendalikan sumber daya sistem berikut ini:

✓ **CPU:**

Sumber daya **CPU** mungkin terbatas pada basis per – sesi atau per – panggilan. Sebuah **CPU / Session** terbatas pada 1000 yang berarti bahwa jika setiap sesi individual yang menggunakan profil ini memakai lebih dari 10 detik dari waktu **CPU** (waktu **CPU** terbatas pada 100 detik), kemudian sesi tersebut menerima sebuah error dan ter-log off:

**ORA-02392: exceeded session limit on CPU usage, you are being logged off**

Pembatasan per – panggilan merupakan hal yang sama, tetapi disamping dari pembatasan semua sesi pengguna, hal ini mencegah setiap perintah tunggal dari memakai terlalu banyak **CPU**. Jika **CPU / Call** terbatas dan pengguna melampaui batas, perintah akan dibatalkan dan pengguna mendapatkan sebuah pesan error seperti berikut ini:

**ORA-02393: exceeded call limit on CPU usage**

✓ **Jaringan / memori:**

Setiap sesi database memakai sumber daya memori sistem dan (jika sesi berasal dari pengguna bukan lokal dari server) sumber daya jaringan.

- **Connect Time:**

Seberapa lama seorang pengguna dapat terkoneksi sebelum secara otomatis ter-log off.

- **Idle Time:**

Seberapa lama sebuah sesi pengguna dapat menganggur sebelum secara otomatis ter-log off. Waktu menganggur terkalkulasi hanya untuk proses server. Tidak diambil ke dalam kegiatan aplikasi account.

Batas **IDLE\_TIME** tidak terpengaruh oleh query long-running dan operasi lainnya.

- **Concurrent Sessions:**  
Berapa banyak sesi serempak dapat dibuat menggunakan sebuah account pengguna database.
  - **Private SGA:**  
Batas jumlah dari ruang yang dipakai di dalam **SGA** untuk sorting, merging bitmaps, dan lain-lain. Larangannya akan berefek hanya jika sesi menggunakan server bersama.
- ✓ **Disk I/O:**  
Batas jumlah dari data seorang pengguna dapat dibaca pada saat tingkatan per – sesi atau per – panggilan. **Reads / Session** dan **Reads / Call** menempatkan sebuah pembatasan pada sejumlah total bacaan diantara memori dan disk. Hal ini dapat dilakukan untuk memastikan bahwa tidak ada pernyataan intensif **I/O** menimbun memori dan mengikat disk.

Profil juga mengijinkan sebuah batas majemuk. Batas majemuk berdasarkan pada kombinasi berat dari **CPU / Session**, **reads / session**, **connect time**, dan **SGA** khusus.

**Create Profile**

Show SQL Cancel OK

**General** Password

Name: LIMITED\_USER

**Details**

CPU/Session (Sec./100)	1000	edit
CPU/Call (Sec./100)	UNLIMITED	edit
Connect Time (Minutes)	DEFAULT	edit
Idle Time (Minutes)	60	edit

**Database Services**

Concurrent Sessions (Per User)	DEFAULT	edit
Reads/Session (Blocks)	DEFAULT	edit
Reads/Call (Blocks)	DEFAULT	edit
Private SGA (kBytes)	DEFAULT	edit
Composite Limit (Service Units)	DEFAULT	edit

Gambar 50. Pembuatan Profile

#### 4.5. Pengguna Otentik

Otentik berarti memverifikasi identitas seseorang (pengguna, alat, atau entitas lainnya) yang ingin menggunakan data, sumber daya, atau aplikasi. Memvalidasi identitas menetapkan sebuah hubungan kepercayaan untuk interaksi selanjutnya. Otentik juga memungkinkan akuntabilitas dengan membuatnya mungkin untuk berhubungan akses dan aksi untuk spesifik identitas. Setelah otentik, proses otorisasi bisa diijinkan atau batas tingkatan akses dan aksi diijinkan untuk entitas tersebut.

Ketika Anda membuat pengguna Anda harus memutuskan teknik otentik yang digunakan, yang nantinya dapat dimodifikasi.

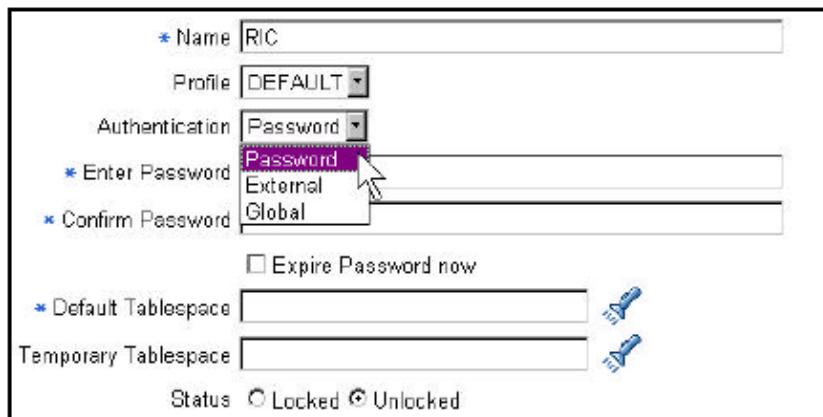
- ✓ **Password:**  
Juga direferensikan sebagai otentik oleh database Oracle, membuat setiap pengguna dengan sebuah kata kunci yang harus dimasukkan ketika pengguna mencoba untuk masuk ke koneksi. Ketika mengatur kata kunci Anda dapat mengkadaluarsakan kata kunci tersebut secepatnya, yang mana memaksa pengguna untuk mengubah kata kunci setelah logging in pertama. Jika Anda merencanakan kata kunci terkadaluarsa, pastikan bahwa pengguna mampu mengubah kata kunci. Beberapa aplikasi tidak mempunyai fungsi ini.
- ✓ **External:**  
Juga direferensikan sebagai otentik oleh sistem operasi, pengguna tidak dapat terkoneksi ke Oracle secara tradisional, tanpa menspesifikasi sebuah nama pengguna atau kata kunci. Dengan otentik eksternal, database Anda terbentang di bawah sistem operasi atau servis otentik jaringan untuk melarang akses ke account database. Sebuah kata kunci database tidak digunakan untuk jenis login ini. Jika sistem operasi Anda atau servis jaringan mengijinkan, Anda dapat mempunyai pengguna otentik. Jika Anda melakukannya juga, aturlah parameter inisialisasi **OS\_AUTHENT\_PREFIX** dan gunakan prefiks ini dalam nama pengguna Oracle. Parameter **OS\_AUTHENT\_PREFIX** mendefinisikan sebuah prefiks dimana Oracle menambahkan ke awal semua nama account sistem operasi pengguna. Nilai default dari parameter ini adalah **OPS\$** untuk kompatibilitas cadangan dengan versi Oracle sebelumnya. Oracle membandingkan prefiks nama pengguna dengan nama pengguna Oracle dalam database ketika seorang pengguna mencoba untuk terkoneksi. Sebagai contoh asumsikan bahwa **OS\_AUTHENT\_PREFIX** diatur sebagai berikut:  
**OS\_AUTHENT\_PREFIX=OPS\$**

Jika seorang pengguna dengan sebuah nama account sistem operasi **tsmith** terkoneksi ke database Oracle dan terotentik oleh sistem operasi, Oracle memeriksa apakah ada korespondensi pengguna database **OPS\$tsmith** dan, jika ada, mengijinkan pengguna untuk terkoneksi. Semua referensi pengguna yang terotentikasi oleh sistem operasi harus termasuk prefik, seperti dapat dilihat pada **OPS\$tsmith**.

**CATATAN:** Teks dari parameter inisialisasi **OS\_AUTHENT\_PREFIX** merupakan hal yang sensitif pada beberapa sistem operasi. Lihat spesifik sistem operasi Oracle Anda untuk informasi lebih lanjut mengenai parameter inisialisasi ini.

✓ **Global:**

Otentik kuat melalui pilihan **Oracle Advanced Security**, otentik global mengijinkan pengguna untuk teridentifikasi melalui penggunaan biometrik, sertifikat x509, peralatan tertentu, dan **Oracle Internet Directory**.



Gambar 51. Autentikasi User

#### 4.6. Tablespace dan Penguncian Default dan Tidak Tetap

Sebuah tablespace default merupakan tablespace dimana obyek dibuat jika sebuah tablespace tidak terspesifikasi dalam pembuatan obyek. Jika Anda tidak memilih sebuah default tablespace, kemudian sistem mendefinisikan default maka tablespace tetap akan digunakan.

Tablespace tidak tetap merupakan tablespace dimana semua aksi sorting terjadi. Ada beberapa cara untuk mensortir, sebagai contoh ketika membuat sebuah indeks atau menggunakan sebuah klausa **ORDER BY** pada pernyataan **SELECT**. Jika Anda tidak menspesifikasikan satu, maka sistem akan mendefinisikan penggunaan tablespace tidak tetap.

Ketika pengguna tercipta, maka pengguna memiliki status terkunci atau tidak terkunci. Jika pengguna terkunci, hal ini berarti bahwa tidak ada seorang pun yang dapat log sebagai penggunanya. Account pengguna baru tidak terkunci secara default.

**CATATAN:** Hal ini mungkin untuk mendefinisikan sebuah default dan tablespace tidak tetap pada tingkat database. Jika Anda mendefinisikan bagian tersebut pada tingkat database, maka pengguna linnya tidak ditugaskan pada tablespace spesifik untuk default dan tidak tetap ketika dibuat, dibuat sebagai default database dan tablespace tidak tetap.



Gambar 52. Pilihan Tablespace untuk user

#### **4.7. Pengguna Database dan Skema**

Sebuah skema adalah sekumpulan obyek database yang dimiliki oleh pengguna database. Sebuah skema mempunyai nama yang sama dengan pengguna yang memilikinya. Obyek skema termasuk struktur seperti tabel, view, indeks, dan Java dan kode penyimpanan PL/SQL. Tidak ada hubungan antara tablespace dengan skema.

Obyek pada skema yang sama dapat berada dalam tablespace yang berbeda, dan sebuah tablespace dapat menahan obyek dari skema yang berbeda. Ketika seorang pengguna database terbentuk, skema korespondensi dengan nama yang sama diciptakan untuk pengguna tersebut. Seorang pengguna hanya bisa terhubung dengan sebuah skema dari nama yang sama, dan kemudian **username** dan **schema** akan sering digunakan untuk pergantian.

Obyek skema yang memiliki ruang dalam database (seperti tabel dan indeks) dibuat dalam tablespace default pengguna yang jika tidak ditempatkan di suatu tempat. Pemilik skema mempunyai kontrol penuh atas obyek skema dan dapat memberikan ijin untuk pengguna lainnya untuk menggunakan obyek tersebut.

Beberapa account pengguna dibuat khusus untuk menyediakan sebuah skema dan tidak disediakan untuk mengijinkan pengguna untuk log in ke database. Account tersebut harus terkunci, yang berarti bahwa tidak ada seorang pun yang bisa log in pada account tersebut. Untuk mengunci account, pilih **Locked** dalam bagian status dari halaman **Create** atau **Edit User**.

#### **4.8. Penanda untuk Pengguna**

Untuk membuat pengguna, administrator harus menugaskan sebuah profil, memilih sebuah teknik otentik, dan menugaskan tablespace. Dengan cara default maka pengguna yang terbuat tidak akan diberi ijin apapun juga, yang berarti

tidak akan dapat melakukan apa – apa pada database. **Database Control** secara otomatis memberikan sedikit ijin yang mengijinkan pengguna untuk terkoneksi pada database dan membuat beberapa obyek dalam database. Meskipun demikian jika pengguna tidak mempunyai kuota dalam tablespace manapun juga, maka pengguna akan tetap tidak bisa membuat obyek.

## A. IJIN

Ijin adalah hak untuk mengeksekusi tipe bagian dari pernyataan SQL atau untuk mengakses obyek pengguna yang lain. Oracle mengijinkan kontrol yang sangat baik atas pengguna apakah bisa atau tidak bisa bekerja dalam database. Ijin dibagi dalam dua kategori, yaitu:

- ✓ Ijin sistem:  
Setiap ijin sistem mengijinkan pengguna untuk melakukan operasi bagian database atau kelas dari operasi database; sebagai contoh, ijin untuk membuat tablespace adalah merupakan ijin sistem. Ijin sistem dapat juga diberikan oleh administrator atau oleh seseorang yang secara jelas memberikan ijin untuk mendaftar ijin tersebut. Ada lebih dari 100 buah ijin sistem yang berbeda.
- ✓ Ijin obyek:  
Ijin obyek mengijinkan pengguna untuk melakukan aksi bagian dalam sebuah obyek yang spesifik, seperti tabel, view, sequence, prosedur, fungsi, atau paket. Tanpa ijin yang spesifik, pengguna hanya dapat mengakses obyek sendiri. Ijin obyek juga dapat diberikan oleh pemilik obyek, oleh administrator, atau oleh seseorang yang secara jelas memberikan ijin untuk memperoleh ijin obyek tersebut.

## B. IJIN SISTEM

Untuk memperoleh ijin sistem, klik link **Systems Privileges**, pilih ijin yang layak dari daftar ijin yang tersedia,

dan gerakkan bagian tersebut ke kotak daftar **Selected System Privileges** dengan mengklik panah **Move**.

Memperoleh ijin dengan klausa ANY berarti bahwa ijin melintasi garis skema. Sebagai contoh, ijin **CREATE TABLE** mengijinkan pengguna untuk membuat tabel, tetapi hanya dalam skema sendiri. Ijin **SELECT ANY TABLE** mengijinkan pengguna untuk memilih dari tabel yang dimiliki oleh pengguna yang lain.

Memilih kotak periksa **Admin Option** mengijinkan pengguna untuk mendaftar ijin dan memberikan ijin sistem ke pengguna yang lain.

Berhati – hatilah dalam memutuskan kebutuhan keamanan sebelum memberikan ijin sistem. Beberapa ijin sistem biasanya hanya bisa diberikan oleh administrator:

✓ **RESTRICTED SESSION:**

Ijin tersebut mengijinkan pengguna untuk log in meskipun jika database sudah dibuka dalam mode terlarang.

✓ **SYSDBA dan SYSOPER:**

Ijin tersebut mengijinkan pengguna untuk shut down, start up, melakukan recovery, dan tugas administratif lainnya dalam database.

✓ **DROP ANY object:**

Ijin tersebut mengijinkan pengguna untuk menghapus obyek yang tidak dimiliki.

✓ **CREATE, MANAGE, DROP, ALTER TABLESPACE:**

Selain administrator biasanya tidak bisa mengontrol tablespace.

✓ **CREATE ANY DIRECTORY:**

Oracle mengijinkan pengembang untuk memanggil kode eksternal (sebagai contoh, sebuah perpustakaan C) dari dalam PL/SQL. Sebagai sebuah ukuran keamanan, direktori sistem operasi dimana kode ditempatkan harus tersambung ke sebuah obyek direktori virtual Oracle. Dengan ijin

**CREATE ANY DIRECTORY**, seorang pengguna dapat secara potensial memanggil obyek kode yang tidak aman.

✓ **EXEMPT ACCESS POLICY:**

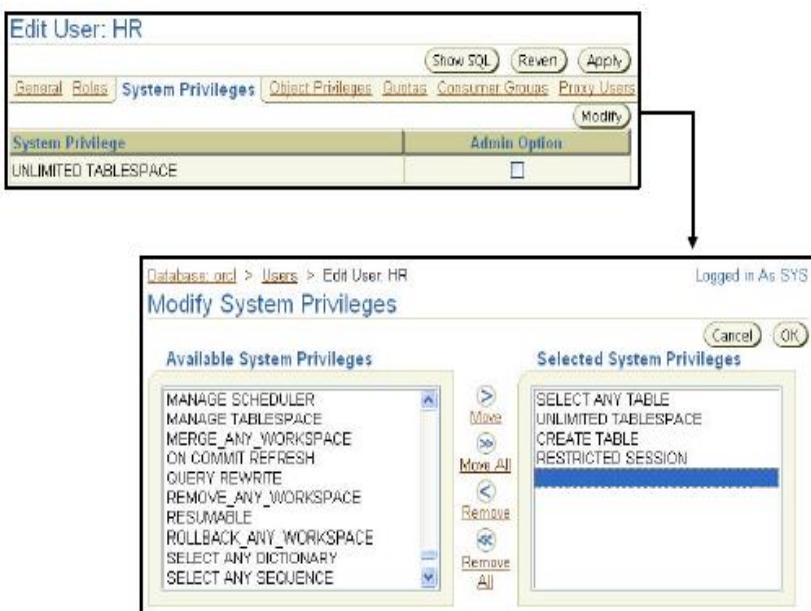
Ijin tersebut mengijinkan seorang pengguna untuk memotong jalur fungsi keamanan yang ditempatkan dalam tabel atau view.

✓ **GRANT ANY OBJECT PRIVILEGE:**

Ijin tersebut mengijinkan seorang pengguna untuk memberikan ijin obyek pada obyek yang tidak dimiliki.

✓ **ALTER DATABASE dan ALTER SYSTEM:**

Selain administrator biasanya tidak diijinkan untuk mengubah database atau perangkat.



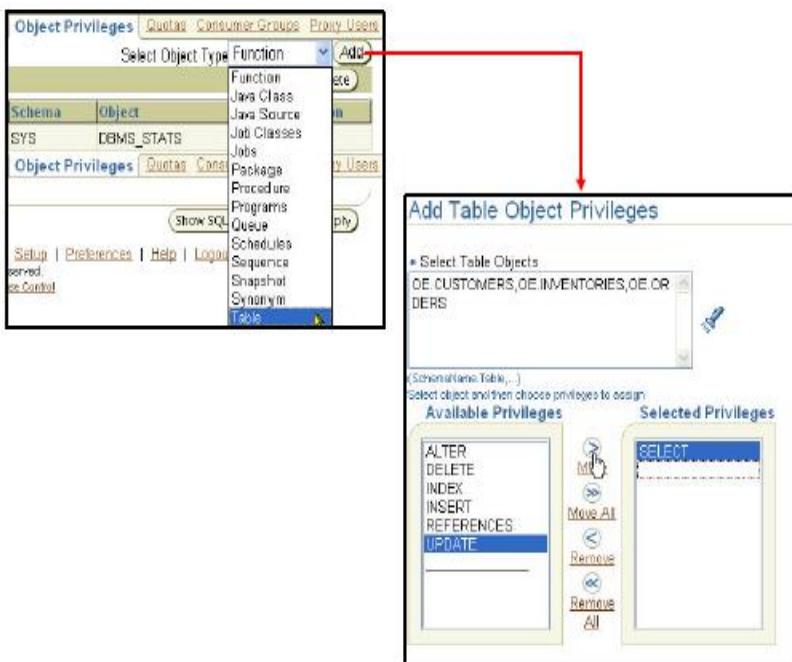
Gambar 53. Hak Akses User

## C. IJIN OBYEK

Untuk memperoleh ijin obyek, klik link **Object Privileges**, pilih tipe obyek yang Anda inginkan untuk dimasuki, dan klik tombol **Add**. Pilih obyek yang Anda ingin masuki dengan memasukkan dalam **username.object name** atau dengan memilih bagian tersebut dari daftar.

Selanjutnya, pilih ijin layak dari kotak daftar **Available Privileges** dan klik tombol **Move**.

Pilih kotak periksa **Grant** dari daftar ijin obyek untuk mengijinkan pengguna untuk memberikan ijinnya pada pengguna yang lain pada akses yang sama.



Gambar 54. Memberikan Hak Akses Objek

#### 4.9. Memberikan Kuota untuk Pengguna

Kuota adalah sebuah ruang yang diijinkan dalam sebuah tablespace yang diberikan. Secara default pengguna tidak mempunyai kuota pada tablespace. Anda mempunyai tiga pilihan untuk menyediakan seorang pengguna kuota pada sebuah tablespace.

✓ **Unlimited:**

Mengijinkan pengguna untuk menggunakan ruang sebanyak mungkin yang tersedia dalam tablespace.

✓ **Value:**

Merupakan sejumlah kilobytes atau megabytes yang pengguna dapat pakai. Hal ini tidak menjamin ruang diatur untuk pengguna. Nilai ini dapat menjadi lebih besar atau kecil daripada ruang yang ada yang tersedia dalam tablespace.

✓ Ijin sistem **UNLIMITED TABLESPACE:**

Ijin sistem ini menolak semua kuota tablespace individual dan memberikan pengguna kuota tidak terbatas pada semua tablespace, termasuk **SYSTEM** dan **SYS AUX**. Ijin ini harus diberikan dengan peringatan.

Anda seharusnya tidak memberikan kuota untuk pengguna dalam tablespace **SYSTEM** atau **SYS AUX**. Biasanya, hanya pengguna **SYS** dan **SYSTEM** yang dapat menciptakan obyek dalam tablespace **SYSTEM** atau **SYS AUX**. Pengguna tidak membutuhkan kuota pada tablespace penugasan tidak tetap atau tablespace undo apapun juga.

**Edit User: HR**

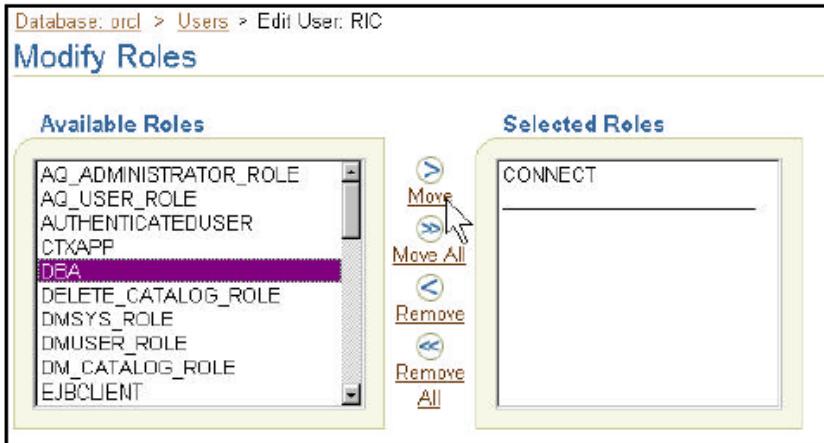
Tablespace	Quota	Value	Unit
EXAMPLE	Value	250	MBytes
SYSAUX	None	0	MBytes
SYSTEM	None	0	MBytes
TEMP	None	0	MBytes
UNDOTBS1	None	0	MBytes
USERS (Default)	Unlimited	0	MBytes

Gambar 55. Memberikan Kuota

#### 4.10. Memberikan Roles untuk Pengguna

Sebuah role adalah pengaturan ijin yang dapat diberikan kepada pengguna atau kepada role yang lainnya. Anda dapat menggunakan role untuk mendaftarkan ijin database. Anda dapat menambah ijin ke sebuah role dan kemudian memberikan role kepada seorang pengguna. Pengguna kemudian dapat menjalankan role dan mengerjakan ijin yang diberikan oleh role. Sebuah role terdiri atas semua ijin yang diberikan untuk role tersebut dan semua ijin dari role lain yang diberikan kepadanya.

Secara default, **Enterprise Manager** secara otomatis memberikan role koneksi untuk pengguna baru. Mengijinkan pengguna untuk terkoneksi pada database dan membuat obyek database dalam skema sendiri.



Gambar 56. Memodifikasi Roles

#### 4.11. Roles

Dalam kebanyakan sistem terlalu banyak waktu yang dipakai untuk memberikan ijin tertentu kepada setiap pengguna secara individual, dan terlalu banyak juga kesempatan untuk terjadi error. Oracle menyediakan ijin pengaturan yang mudah dan terkontrol melalui roles. Roles merupakan nama kelompok dari ijin terelasi yang diberikan kepada pengguna atau roles yang lainnya. Bagian tersebut didesain untuk memudahkan administrasi ijin dalam database, dan meningkatkan perbaikan keamanan.

Karakteristik role:

- ✓ Ijin diberikan atau ditolak dari roles jika role tersebut adalah pengguna.
- ✓ Ijin diberikan atau ditolak dari pengguna atau role yang lain jika bagian tersebut adalah ijin sistem.
- ✓ Role dapat terdiri diantara sistem dan ijin obyek.
- ✓ Role dapat dijalankan atau tidak dapat dijalankan untuk setiap pengguna yang diberi role.

- ✓ Role dapat membutuhkan kata kunci untuk dapat dijalankan.
- ✓ Role tidak dimiliki oleh siapa pun, dan bagian tersebut tidak ada dalam skema mana pun juga.

#### 4.12. Keuntungan dari Roles

Keuntungan – keuntungan roles, yaitu:

- ✓ Ijin pengaturan yang lebih mudah:  
Gunakan roles untuk menyederhanakan ijin pengaturan. Daripada memberikan pengaturan ijin yang sama untuk beberapa pengguna, Anda dapat memberikan ijin ke sebuah role, dan kemudian memberikan role tersebut kepada setiap pengguna.
- ✓ Ijin pengaturan yang dinamis:  
Jika ijin terasosiasikan dengan sebuah role yang termodifikasi, semua pengguna yang memperoleh role mendapatkan ijin termodifikasi secara otomatis dan secepatnya.
- ✓ Ijin ketersediaan yang selektif:  
Roles dapat dijalankan dan juga tidak dapat dijalankan untuk mengubah ijin hidup dan mati secara tidak tetap. Menjalankan role dapat juga digunakan untuk memverifikasi bahwa seorang pengguna sudah diberi role tersebut.
- ✓ Diberikan melalui sistem operasi:  
Perintah sistem operasi atau kegunaan dapat digunakan untuk menugaskan roles kepada pengguna dalam database.

#### 4.13. Pendefinisian Awal Roles

Ada beberapa roles yang didefinisikan secara otomatis untuk database Oracle ketika Anda memulai pembuatan skrip database. **CONNECT** diberikan secara otomatis untuk semua pengguna yang bekerja dengan **Enterprise Manager**.

**SELECT\_CATALOG\_ROLE** disediakan untuk mengakses view kamus data dan paket (role ini ditolak bekerja dengan ijin sistem **SELECT\_ANY\_DICTIONARY**). Role DBA termasuk dalamnya adalah semua ijin dan tidak bisa diberikan kepada yang bukan administrator.

Tabel 1. Tabel Hak Akses

<b>CONNECT</b>	CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE, CREATE DATABASE LINK, CREATE CLUSTER, ALTER SESSION
<b>RESOURCE</b>	CREATE TABLE, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TRIGGER, CREATE TYPE, CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR
<b>SCHEDULER_ADMIN</b>	CREATE ANY JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
<b>DBA</b>	Ijin sistem yang paling banyak digunakan, beberapa roles yang lain. Tidak diberikan kepada non administrator
<b>SELECT_CATALOG_ROLE</b>	Bukan ijin sistem tetapi lebih dari 1600 ijin obyek dalam kamus data

Roles yang lainnya yang mengotorisasi Anda untuk pendaftaran fungsi spesial dibuat ketika fungsi tersebut diinstall. Sebagai contoh, **XDBADMIN** mengandung ijin yang

dibutuhkan untuk mendaftar database XML jika tampilan tersebut diinstall. **AQ\_ADMINISTRATOR\_ROLE** menyediakan ijin untuk mendaftar antrian ahli. **HS\_ADMIN\_ROLE** termasuk ijin yang dibutuhkan untuk mendaftar pelayanan yang bervariasi. Anda tidak boleh mengubah ijin yang diberikan ke role fungsional tanpa panduan dari **Oracle Support** karena Anda mungkin secara tidak sengaja akan membuat tidak jalan fungsi – fungsi yang dibutuhkan tersebut.

#### 4.14. Mengamankan Roles

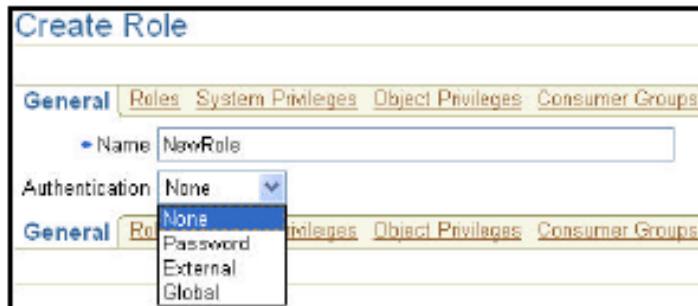
Role biasanya dapat dijalankan secara default, yang berarti bahwa jika role diberikan kepada seorang pengguna, maka pengguna tersebut dapat bekerja dengan ijin yang diberikan ke role tersebut. Hal ini memungkinkan untuk:

- ✓ Membuat sebuah role menjadi non default. Ketika role diberikan kepada seorang pengguna, bersihkan kotak periksa **DEFAULT**. Sekarang pengguna harus secara jelas menjalankan role sebelum ijin role dapat dilakukan.
- ✓ Mempunyai sebuah role yang dibutuhkan oleh otentikasi tambahan. Otentik default untuk sebuah role adalah **NONE** tetapi ini mungkin untuk mempunyai role yang dibutuhkan oleh otentikasi tambahan sebelum dapat diatur.
- ✓ Membuat role aplikasi keamanan yang hanya dapat dijalankan dengan cara mengeksekusi sebuah prosedur PL/SQL dengan sukses. Prosedur PL/SQL dapat memeriksa hal – hal seperti alamat jaringan pengguna, yang mana program pengguna sedang dijalankan. Waktu hari, atau hal – hal lainnya yang dibutuhkan untuk mengamankan sekelompok ijin.

Role dapat berupa non default:

```
SET ROLE vacationdba;
```

Role dapat dilindungi melalui otentikasi:



Gambar 57. Role

Role dapat diamankan secara program:

```
CREATE ROLE secure_application_role  
IDENTIFIED USING <security_procedure_name>;
```

Gambar 58. Create Role



## BAB V

# MENGATUR OBYEK SKEMA

### 5.1. Tujuan

Setelah menyelesaikan pelajaran ini, Anda diharapkan dapat melakukan hal – hal sebagai berikut:

- ✓ Membuat dan memodifikasi tabel.
- ✓ Mendefinisikan konstrain.
- ✓ Menampilkan atribut tabel.
- ✓ Menampilkan isi tabel.
- ✓ Membuat indeks dan tampilan.

### 5.2. Apa Itu Skema?

Sebuah skema adalah sekumpulan obyek database yang dimiliki oleh sebagian pengguna. Sebuah skema mempunyai nama yang sama dengan pengguna yang milikinya. Obyek skema adalah struktur logika yang secara langsung menunjuk kepada data dari database. Obyek skema termasuk struktur seperti tabel, view, dan indeks.

**CATATAN:** Tidak ada hubungan antara sebuah tablespace dan sebuah skema. Obyek dalam skema yang sama boleh jadi merupakan tablespace yang berbeda, dan sebuah tablespace dapat memegang obyek dari skema yang berbeda.

Anda dapat membuat dan memanipulasi obyek skema dengan menggunakan SQL atau **Enterprise Manager**. Ketika

Anda menggunakan **Enterprise Manager**, SQL akan dibuka juga untuk Anda.

Ketika Anda membuat database, sejumlah skema akan dibuat untuk Anda. Dua bagian yang terpenting adalah:

✓ **Skema SYS:**

Semua tabel dasar dan tampilan yang mewajibkan kamus data database terbuat dalam skema **SYS**. Kamus data adalah sekumpulan tabel yang menjelaskan database Oracle. Kamus data terbuat dalam tablespace **SYSTEM** ketika database terbuat dan diperbaharui oleh server database Oracle ketika sebuah pernyataan bahasa pendefinisian data (DDL) dieksekusi. Kamus data mengandung informasi tentang pengguna, obyek skema, dan struktur penyimpanan. Anda dapat menggunakan kamus data sebagai referensi read – only untuk informasi tentang database. Ketika Anda menggunakan Enterprise Manager Anda mengakses tabel kamus data melalui tampilan.

Obyek dalam skema **SYS** harus tidak boleh dimodifikasi oleh pengguna lainnya atau administrator database, dan tidak seorang pun boleh membuat tabel apapun juga dalam skema dari pengguna **SYS**.

✓ **Skema SYSTEM:**

Mengandung tabel tambahan dan tampilan yang menyimpan informasi administrasi, dan tabel serta tampilan internal yang digunakan oleh berbagai macam pilihan dan peralatan Oracle. Anda tidak boleh membuat obyek tambahan apapun juga dalam skema **SYSTEM**.

Selama instalasi lengkap dari database Oracle, skema contoh secara otomatis juga diinstal dengan benih database. Skema contoh menyediakan tujuan dari penyediaan sebuah platform umum untuk contoh dalam dokumentasi dan kurikulum Oracle. Bagian tersebut adalah satu set bantuan skema yang

terhubung satu sama lain pada penyediaan sebuah pencapaian berlapis untuk kompleksitas dan termasuk hal – hal berikut ini:

✓ **HR:**

Skema Human Resource (HR) adalah sebuah skema sederhana untuk mengenalkan topik – topik dasar. Sebuah ekstensi untuk skema ini mendukung demonstrasi Oracle Internet Directory.

✓ **OE:**

Skema Order Entry (OE) adalah untuk pesetujuan dengan hal – hal dari kompleksitas intermediate. Sejumlah tipe data tersedia dalam skema OE. Subskema **OC** (Online Catalog) adalah sekumpulan obyek database obyek – relasional yang terbentuk di dalam skema **OE**.

✓ **PM:**

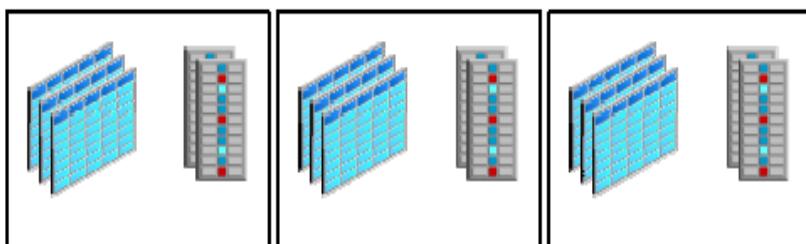
Skema Product Media (PM) ditujukan untuk tipe data multimedia.

✓ **QS:**

Skema Queued Shipping (QS) mengandung sekumpulan skema yang digunakan untuk mendemonstrasikan kemampuan Oracle Advanced Queuing.

✓ **SH:**

Skema Sales History (SH) dirancang untuk mengijinkan demonstrasi dengan sejumlah besar data. Ekstensi untuk skema ini menyediakan dukungan untuk proses analisis berpengalaman.

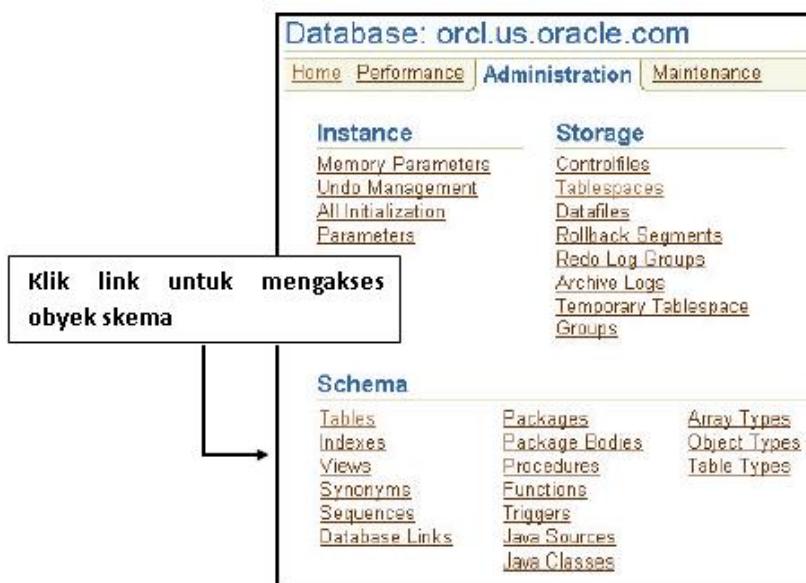


Gambar 59. Objek Schema

### 5.3. Mengakses Obyek Skema

Anda dapat dengan cepat mengakses sejumlah tipe dari obyek skema dari wilayah **Schema** dari halaman **Database Administration**.

Setelah mengklik salah satu dari link tersebut, halaman **Result** akan ditampilkan. Dalam wilayah **Search** dari halaman tersebut Anda dapat memasukkan sebuah nama skema dan nama obyek untuk mencari sebuah obyek tertentu. Sebagai tambahan, Anda dapat mencari tipe – tipe obyek yang lainnya dari wilayah **Search** dengan memilih tipe obyek dari menu **drop down**. Menu **drop down** termasuk tipe obyek tambahan yang tidak ditampilkan sebagai link pada halaman **Database Administration**.



Gambar 60. Pembuatan Objek Schema

## 5.4. Menamakan Obyek Database

Ketika Anda menamai sebuah obyek dalam database maka Anda mempunyai pilihan untuk memberi tanda kurung pada nama dengan tanda kurung dobel (“”). Jika Anda melakukannya maka Anda dapat memecahkan beberapa peraturan penamaan yang disebutkan dalam slide. Meskipun demikian hal ini tidak direkomendasikan, karena jika Anda melakukan hal ini maka Anda harus selalu menunjukkannya dengan kuota di sekitar nama. Sebagai contoh, jika Anda menamai tabel dengan “Local Temp” maka Anda harus melakukan hal – hal berikut ini:

```
SQL> select * from "Local Temp";
TEMP_DATE LO_TEMP HI_TEMP
-----
01 - DEC - 03      30      41
```

Jika Anda lupa maka Anda akan mendapatkan:

```
SQL> select * from "Local temp";
select * from "Local Temp"
*
ERROR at line 1:
ORA-00942: table or view does not exist
```

Nama nonkuota disimpan dalam huruf besar dan bukanlah kasus yang sensitif. Ketika sebuah pernyataan SQL diproses, nama nonkuota diubah menjadi huruf besar semua. Identifikasi nonkuota dapat terdiri dari hanya karakter alfanumerik dari pengaturan karakter database Anda dan tanda garis bawah (\_), tanda dollar (\$), dan tanda pound (#). Link database juga dapat terdiri dari titik(.) dan tanda “at” (@). Anda sebaiknya tidak menggunakan \$ dan # dalam identifikasi nonkuota.

Identifikasi kuota dapat terdiri dari karakter dan tanda baca seperti spasi. Meskipun demikian, baik itu identifikasi kuota atau pun non kuota dapat terdiri dari tanda petik dobel.

Intinya adalah:

- ✓ Panjang nama adalah 1 – 30 bytes dengan perkecualian sbb:
  - Nama database dibatasi hanya 8 bytes.
  - Nama link database bisa sepanjang 128 bytes.
- ✓ Nama nonkuota tidak disediakan oleh kata – kata dalam Oracle.
- ✓ Nama nonkuota harus dimulai dengan sebuah karakter alfabetik dari pengaturan karakter database Anda.
- ✓ Nama nonkuota hanya bisa terdiri dari:
  - Karakter alfanumerik dari pengaturan karakter database Anda.
  - Tanda garis bawah (\_).
  - Tanda dollar (\$).
  - Tanda pound (#).
- ✓ Tidak ada dua obyek yang akan mempunyai nama yang sama di dalam namespace yang sama.

## 5.5. Ruang Nama Obyek Skema

Database Oracle menggunakan namespace untuk memecahkan referensi obyek skema. Ketika Anda menunjuk sebuah obyek dalam sebuah pernyataan SQL, Oracle mempertimbangkan konteks dari pernyataan SQL dan menempatkan obyek dalam namespace yang tepat. Setelah menempatkan obyek, Oracle melakukan operasi yang ditentukan oleh pernyataan dalam obyek. Jika penamaan obyek tidak dapat ditemukan dalam namespace yang tepat, maka Oracle akan error.

Karena tabel dan tampilan ada dalam namespace yang sama, sebuah tabel dan sebuah tampilan dalam skema yang sama tidak bisa mempunyai nama yang sama. Meskipun

demikian, tabel dan indeks ada dalam namespace yang berbeda. Kemudian, sebuah tabel dan sebuah indeks dalam skema yang sama dapat mempunyai nama yang sama.

Setiap skema dalam database mempunyai namespace-nya sendiri untuk obyek yang terkandung di dalamnya. Artinya, untuk contoh, dua tabel dalam skema yang berbeda berada dalam namespace yang berbeda dan dapat mempunyai nama yang sama.

Hal – hal berikut ini berada dalam namespace yang sama :	Hal – hal berikut ini mempunyai namespace mereka sendiri :
<ul style="list-style-type: none"><li>• Tables</li><li>• Views</li><li>• Sequences</li><li>• Private synonyms</li><li>• Stand-alone procedures</li><li>• Stand-alone stored functions</li><li>• Packages</li><li>• Materialized views</li><li>• User-defined types</li></ul>	<ul style="list-style-type: none"><li>• Indexes</li><li>• Constraints</li><li>• Clusters</li><li>• Database triggers</li><li>• Private database links</li><li>• Dimensions</li></ul>

## 5.6. Menspesifikasi Tipe Data dalam Tabel

Ketika Anda membuat tabel, Anda harus menspesifikasi sebuah tipe data untuk setiap kolomnya. Ketika Anda membuat sebuah prosedur atau fungsi, Anda harus menspesifikasi sebuah tipe data untuk setiap argumennya. Tipe data tersebut mendefinisikan domain nilai yang mana tiap kolom dapat mengandungnya atau setiap argumen dapat memilikiinya.

Database Oracle mengembangkan tipe data termasuk berikut ini:

✓ **CHAR:**

Data karakter dengan panjang tetap dari ukuran panjang bytes. Ukuran maksimum adalah 2000 bytes. Ukuran minimum dan default adalah 1 byte.

✓ **VARCHAR2:**

String karakter panjang variabel mempunyai ukuran panjang maksimum bytes atau karakter. Ukuran maksimum adalah 4000 bytes. Anda harus menspesifikasi ukuran untuk **VARCHAR2**.

✓ **DATE:**

Jangkauan data valid dari 1 Januari 4712 SM sampai dengan 31 Desember 9999 M; juga tersedia waktu mundur hingga ke detik.

✓ **NUMBER:**

Number mempunyai presisi p dan skala s. Jangkauan presisi dari 1 sampai dengan 38. Jangkauan skala dari -84 sampai dengan 127.

## 5.7. Tipe Data Lain

Tipe data yang lain adalah:

✓ **FLOAT (p):**

Merupakan tipe data ANSI. Tipe data FLOAT adalah nomor floating-point dengan sebuah biner presisi p. Presisi default untuk tipe data ini adalah 126 binary, atau 38 desimal.

✓ **INTEGER:**

Ekivalen dengan NUMBER (p , 0).

✓ **NCHAR (*length*):**

Tipe data NCHAR adalah sebuah tipe data Unicode-only. Ketika Anda membuat sebuah tabel dengan sebuah kolom NCHAR, Anda mendefinisikan panjang kolom dalam

karakter. Anda mendefinisikan pengaturan karakter nasional ketika Anda membuat database Anda. Panjang maksimum dari sebuah kolom ditentukan oleh pendefinisian pengaturan karakter nasional. Spesifikasi lebar dari tipe data karakter NCHAR menunjuk pada sejumlah karakter. Ukuran kolom maksimum yang diijinkan adalah 2000 bytes. Jika Anda menyelipkan sebuah nilai yang lebih pendek dari panjang kolom, maka Oracle akan menghilangkan nilai pada panjang kolom. Anda tidak dapat menyelipkan sebuah nilai CHAR ke dalam sebuah kolom NCHAR, tapi Anda dapat menyelipkan sebuah nilai NCHAR ke dalam sebuah kolom CHAR.

✓ **NVARCHAR2 (*length*):**

Merupakan tipe data Unicode-only. Seperti halnya NCHAR yaitu panjang maksimumnya adalah 4000 bytes dan tidak akan dihilangkan untuk panjang terspesifikasi.

✓ **LONG:**

Data karakter dari variabel panjangnya di atas 2 GB, atau  $2^{31}-1$  bytes.

✓ **LONG RAW:**

Data biner mentah dari variabel panjangnya di atas 2 GB.

✓ **RAW (*size*):**

Data biner mentah dari ukuran panjang bytes. Ukuran maksimumnya adalah 2000 bytes. Anda harus menspesifikasikan ukuran untuk sebuah nilai RAW.

✓ **ROWID:**

Dasar 64 string mewakili alamat unik dari sebuah baris dalam tabelnya. Tipe data ini diutamakan untuk nilai yang dikembalikan oleh kolom semu ROWID.

✓ **UROWID:**

Dasar 64 string mewakili alamat logika dari sebuah baris dari sebuah tabel indeks yang terorganisir. Ukuran opsional adalah ukuran dari sebuah kolom dari tipe UROWID. Ukuran maksimum dan default adalah 4000 bytes.

- ✓ **BLOB:**  
Sebuah obyek biner yang besar.
- ✓ **CLOB:**  
Sebuah obyek karakter yang besar mengandung byte tunggal atau karakter multibyte. Karakter lebar tetap dan lebar variabel tersedia, dan keduanya menggunakan pengaturan karakter database CHAR.
- ✓ **NCLOB:**
  - Sebuah obyek karakter yang besar mengandung karakter Unicode. Karakter lebar tetap dan lebar variabel tersedia, dan keduanya menggunakan pengaturan karakter database NCHAR.
  - Menyediakan data pengaturan karakter nasional.
- ✓ **Catatan:**  
Ukuran maksimum untuk semua tipe data LOB (BLOB, CLOB dan NCLOB) adalah: (4 GB – 1) \* (nilai CHUNK). CHUNK adalah sebuah atribut opsional yang Anda dapat atur ketika mendefinisikan sebuah LOB. CHUNK menspesifikasi sejumlah bytes yang dialokasikan untuk manipulasi LOB. Jika ukuran bukanlah merupakan sejumlah ukuran blok database, maka database tertangkap dalam bytes ke penggandaan selanjutnya. Untuk contoh, jika ukuran blok database adalah 2048 dan ukuran CHUNK adalah 2050, maka database menempatkan 4096 bytes (2 blok). Nilai maksimum adalah 32768 (32K), yang merupakan ukuran blok database Oracle terbesar yang diijinkan. Ukuran default CHUNK adalah salah satu blok database Oracle.
- ✓ **BFILE:**  
Mengandung sebuah lokator untuk sebuah file biner besar yang disimpan di luar database. Memungkinkan akses aliran byte I/O ke LOB eksternal yang ada dalam server database. Ukuran maksimum adalah 4 GB.

- ✓ **TIMESTAMP (*fractional\_seconds\_precision*):**  
Tahun, bulan, dan nilai hari dari tanggal, seperti jam, menit, dan detik nilai waktu, dimana *fractional\_seconds\_precision* adalah sejumlah digit dalam bagian fraksional dari area waktu tanggal SECOND (sebagai fraksi detik). Nilai yang diterima dalam *fractional\_seconds\_precision* adalah 0 – 9. Defaultnya adalah 6.

## 5.8. Membuat dan Memodifikasi Tabel

Tabel adalah unit dasar dari penyimpanan data dalam database Oracle. Bagian tersebut memegang semua akses data pengguna. Setiap tabel mempunyai kolom dan baris.

Anda dapat membuat sebuah tabel menggunakan Enterprise Manager dengan mengikuti langkah – langkah sbb:

1. Klik **Tables** dalam wilayah **Schema** pada halaman **Administration**. Halaman **Tables** akan muncul.
2. Jika Anda mengetahui nama skema, masukkan semua atau sebagian dalam area **Schema** dalam wilayah **Search**. Jika Anda tidak tahu nama skema, klik ikon **flashlight** di sebelah area **Schema**. **Search** dan **Select**: jendela **Schema** akan ditampilkan. Anda dapat melihat semua nama skema dan pilih salah satu yang ingin Anda lihat.
3. Klik **Create**. **Create Table**: halaman **Table Organization** akan dimunculkan.
4. Terima default dari **Standard**, **Heap Organized** dengan mengklik **Continue**. Halaman **Create Table** akan ditampilkan.
5. Masukkan nama tabel dalam area **Name**.
6. Masukkan nama skema dalam area **Schema** atau klik ikon **flashlight** untuk memanggil fungsi pencarian.
7. Masukkan nama tablespace dalam area **Tablespace** atau klik ikon **flashlight** untuk memanggil fungsi pencarian.

- Dalam bagian **Columns**, masukkan nama kolom dan tipe data.
- Klik **OK**. Sebuah pesan update akan muncul mengindikasikan tabel telah sukses dibuat.

Anda dapat memodifikasi sebuah tabel menggunakan Enterprise Manager dengan mengikuti langkah – langkah sbb:

- Dalam halaman **Tables**, pilih tabel dalam daftar hasil dan klik **Edit**.
- Dalam halaman **Edit Table**, klik tombol **Add 5 Table Columns**. Sebuah daftar kolom edit akan muncul.
- Masukkan nama kolom yang baru, tipe data, dan ukuran. Klik **Apply**.
- Sebuah pesan update akan muncul mengindikasikan tabel telah sukses dimodifikasi.



Gambar 61. Create table

## 5.9. Memahami Kesatuan Data

Anda dapat menggunakan kesatuan konstrain berikut ini untuk memaksa larangan pada input nilai kolom:

✓ **NOT NULL:**

Secara default, semua kolom dalam sebuah tabel mengijinkan nilai null. Null berarti ketiadaan nilai. Sebuah konstrain **NOT NULL** meminta sebuah kolom mengandung nilai no null. Untuk contoh, Anda dapat mendefinisikan sebuah konstrain **NOT NULL** untuk meminta sebuah nilai yang akan diinput dalam kolom **last\_name** untuk semua baris pada tabel **employees**.

✓ **UNIQUE KEY:**

Meminta semua nilai dalam sebuah kolom atau sekumpulan kolom (kunci) menjadi unik yang berarti tidak ada dua baris dari sebuah tabel mempunyai nilai duplikat dalam sebuah kolom tertentu atau sekumpulan kolom. Untuk contoh, sebuah konstrain kunci **UNIQUE** didefinisikan dalam kolom **department\_name** dari tabel **department** untuk melarang baris dengan duplikat nama **department**.

✓ **PRIMARY KEY:**

Setiap tabel dalam database dapat mempunyai lebih dari satu konstrain **PRIMARY KEY**. Nilai dalam kelompok dari satu atau lebih subyek kolom ke aturan konstrain tersebut untuk identifikasi unik dari baris. Efeknya, setiap baris dinamai oleh nilai kunci utamanya.

Implementasi Oracle dari kesatuan konstrain **PRIMARY KEY** menjamin bahwa dua hal berikut ini adalah benar:

- Tidak ada dua baris dari sebuah tabel mempunyai nilai duplikat dalam kolom tertentu atau sekumpulan kolom.
- Kolom kunci utama tidak mengijinkan null. Berarti, sebuah nilai harus ada untuk kolom kunci utama dalam setiap baris.

Oracle memaksa semua konstrain PRIMARY KEY menggunakan indeks. Konstrain kunci utama yang dibuat untuk kolom **department\_id** dalam tabel **departments** dipaksa oleh pembuatan implisit dari:

- Sebuah indeks unik pada kolom tersebut.
- Konstrain NOT NULL untuk kolom tersebut.

✓ Referensi kesatuan konstrain:

Tabel yang berbeda dalam sebuah database relasional dapat direlasikan oleh kolom umum, dan aturan yang mengatur hubungan dari kolom harus dipelihara. Aturan kesatuan referensi menjamin bahwa hubungan tersebut tersedia. Sebuah konstrain kesatuan referensi meminta untuk setiap baris dari sebuah tabel, nilai dalam kunci asing sesuai dengan sebuah nilai dalam sebuah kunci orang tua.

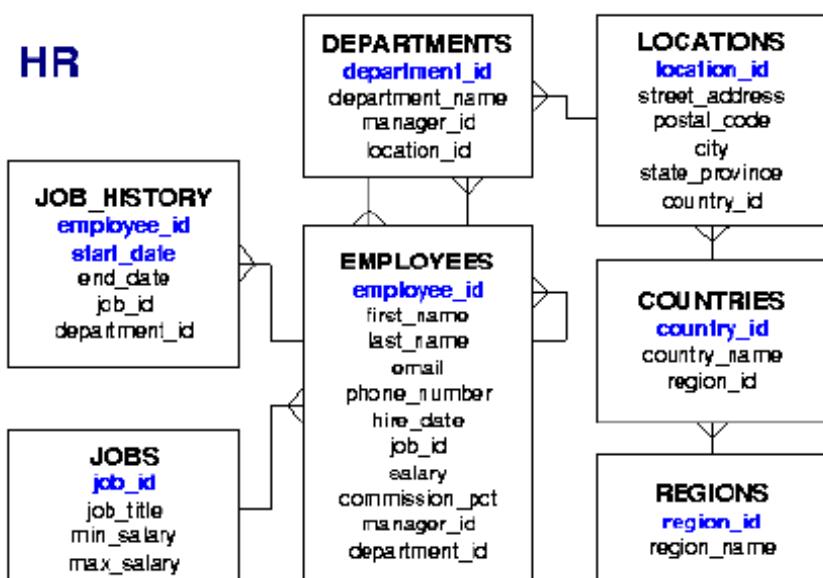
Sebagai contoh, sebuah kunci asing didefinisikan dalam kolom **department\_id** pada tabel **departments**. Hal ini menjamin bahwa semua nilai dalam kolom harus sesuai dengan sebuah nilai dalam kunci utama dari tabel **departments** (juga kolom **department\_id**). Kemudian, tidak ada nomor departemen yang salah yang ada dalam kolom **department\_id** dari tabel **departments**.

Tipe lain dari konstrain kesatuan referensi dinamakan konstrain kesauan referensi diri sendiri. Tipe ini dari kunci asing mereferensikan sebuah kunci orang tua dalam tabel yang sama.

✓ Konstrain **CHECK**:

Meminta kondisi spesifik benar atau tidak dikenal untuk semua baris dari tabel. Jika sebuah pernyataan DML menghasilkan kondisi konstrain CHECK mengevaluasi salah, maka pernyataan dapat diulang.

## HR



Gambar 62. Struktur Tabel user HR

### 5.10. Mendefinisikan Konstrain

Anda dapat menambahkan sebuah konstrain ke sebuah tabel sbb:

1. Pilih tabel pada halaman **Tables** dan klik **Edit**.
2. Klik **Constraints**. Halaman **Constraints** ditampilkan menampilkan semua konstrain yang telah didefinisikan pada tabel.
3. Pilih tipe konstrain yang Anda ingin tambahkan dari menu **drop-down** dan klik **Add**.
4. Masukkan informasi yang sesuai untuk tipe konstrain yang Anda definisikan. Klik **OK**.

Database: orcl.us.oracle.com > Tables > Edit Table: HR.COUNTRIES

## Add UNIQUE Constraint

Up to 32 columns can make up a UNIQUE key constraint. The unique key columns constitute a unique definition.

Name: <System Assigned 3>

**Table Columns**

Available Columns	Selected Columns
COUNTRY_ID REGION_ID	COUNTRY_NAME

**Available Columns:**

- COUNTRY\_ID
- REGION\_ID

**Selected Columns:**

- COUNTRY\_NAME

Buttons: Move, Move All, Remove, Remove All.

Gambar 63. Membuat Konstrain

### 5.11. Menampilkan Atribut dari Sebuah Tabel

Anda dapat menggunakan Enterprise Manager untuk menampilkan atribut dari sebuah tabel dengan mengikuti langkah – langkah sbb:

1. Klik link **Tables** dalam bagian **Schema** dari halaman **Database Administration**.
2. Pilih sebuah tabel dari daftar **Result** dan klik tombol **View** untuk melihat atribut dari tabel tersebut.

**Edit Table: HR.DEPARTMENTS**

**General** [Constraints](#) [Segments](#) [Storage](#) [Options](#)

* Name	DEPARTMENTS
Schema	HR
Tablespace	EXAMPLE
Organization <b>Standard, Heap Organized</b>	

**Columns**

Select Name	Data Type	Size
<input checked="" type="radio"/> DEPARTMENT_ID	NUMBER	4
<input type="radio"/> DEPARTMENT_NAME	VARCHAR2	30
<input type="radio"/> MANAGER_ID	NUMBER	6
<input type="radio"/> LOCATION_ID	NUMBER	4
<a href="#">Add 5 Table Columns</a>		

Gambar 64. Atribut Tabel

### 5.12. Menampilkan Isi dari Sebuah Tabel

Anda dapat dengan mudah menampilkan baris dalam sebuah tabel menggunakan Enterprise Manager dengan langkah – langkah sbb:

1. Pilih tabel pada halaman **Tables**.
2. Pilih **View Data** dari menu **Actions** dan klik **Go**.

Halaman **View Data For Table** muncul. Data baris untuk tabel ditampilkan dalam bagian **Result**. Kotak **Query** menampilkan kueri SQL yang sudah dieksekusi untuk membuat hasil. Pada halaman ini Anda dapat mengklik nama kolom apapun dan mengatur data dalam kolom pada urutan naik atau turun. Jika Anda ingin mengubah kueri tersebut, klik tombol **Refine Query**. Pada **Refine Query** untuk halaman **Table**, Anda

dapat memilih kolom yang Anda ingin tampilkan dan spesifikasikan sebuah klausula WHERE untuk pernyataan SQL untuk membatasi hasilnya.

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

Gambar 65. Isi Tabel

### 5.13. Aksi dengan Tabel

Anda dapat memilih sebuah tabel dan kemudian melakukan aksi – aksi pada tabel tersebut seperti:

✓ **Create Like:**

Dengan ini Anda dapat membuat sebuah tabel yang mempunyai struktur yang sama sebagai tabel terpilih. Anda harus mengubah nama tabel dan nama konstrain. Anda juga dapat menambah atau menghapus kolom dan membuat perubahan lainnya pada struktur tabel. Ketika dibuat, tabel akan menjadi kosong.

✓ **Create Index:**

Gunakan pilihan ini untuk membuat indeks pada sebuah tabel. Indeks hanya bisa dibuat pada kolom yang

menggunakan kondisi WHERE dari pilihan dan pernyataan DML.

✓ **Create Synonym:**

Sebuah sinonim adalah sebuah nama yang digunakan untuk menggantikan nama lengkap tabel. Sebuah sinonim dapat menjadi privat atau publik.

✓ **Create Trigger:**

Sebuah tabel-level trigger adalah sebuah blok PL/SQL yang dieksekusi karena sebuah kejadian yang terjadi pada tabel. Sebagai contoh Anda mungkin ingin menyimpan sebuah salinan data pada tabel yang lain seperti halnya dihapuskan dari tabel **employees**. Anda kemudian akan membuat sebuah trigger “**before delete**” yang akan menyelipkan data yang dihapus dari tabel **employees** ke dalam tabel yang lainnya.

✓ **Gather Statistics:**

**Gather Statistics Wizard** membimbing Anda melalui proses penggerjaan dan modifikasi statistik teroptimalkasi. Statistik disimpan dalam kamus data dan digunakan oleh optimiser berdasarkan biaya. Statistik up to date dapat memperbaiki dengan sangat baik kinerja dari kueri SQL melawan obyek Anda. Oracle menyarankan Anda untuk menggunakan tugas otomatis untuk mengerjakan statistik harian dengan jendela pemeliharaan.

✓ **Generate DDL:**

Mengerjakan perintah DDL untuk membuat tabel terpilih. Perintah ini kemudian dapat tercopy paste ke dalam sebuah file teks untuk digunakan sebagai sebuah skrip atau untuk tujuan dokumentasi.

✓ **Grant Privileges:**

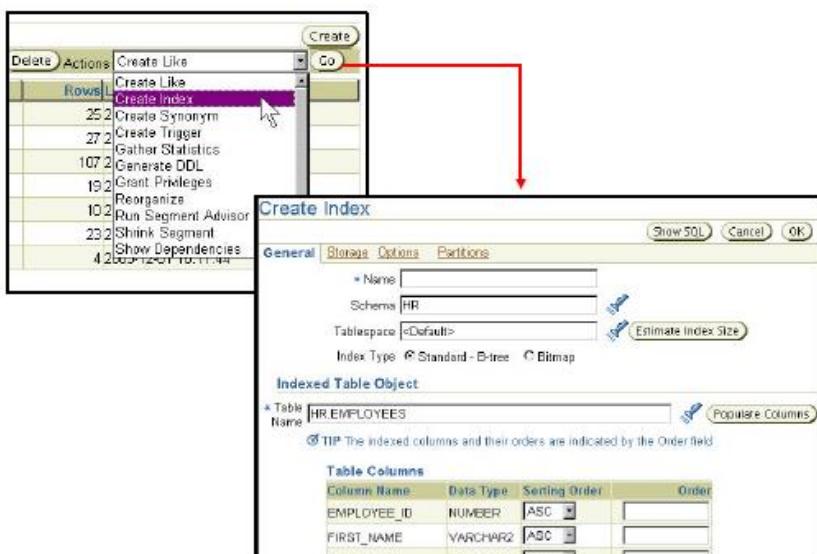
Secara default ketika sebuah tabel terbuat hanya pemilik saja yang dapat melakukan apa saja dengannya. Pemilik harus mengijinkan pengguna lainnya untuk dapat melakukan DML dan beberapa DDL pada tabel.

- ✓ **Reorganize:**  
Anda dapat menggunakan **Reorg Wizard** untuk membangun kembali indeks atau tabel berfragmen, memindahkan obyek ke sebuah tablespace yang berbeda, atau mengoptimalkan atribut penyimpanan dari obyek terspesifikasi.
- ✓ **Run Segment Advisor:**  
**Segment Advisor** menentukan apakah obyek mempunyai ruang yang tidak terpakai yang dapat dilepaskan, mengambil ruang masa depan terestimasi ke dalam pertimbangan. Kalkulasi dari ruang masa depan terestimasi adalah berdasarkan trend historik.
- ✓ **Shrink Segment:**  
Operasi ini memadatkan ruang terfragmen dan, biasanya, membekukan ruang.
- ✓ **Show Dependencies:**  
Menunjukkan tabel yang bergantung pada atau obyek yang bergantung pada tabel.
- ✓ **View Data:**  
Menjalankan sebuah pernyataan terpilih pada tabel, Anda menyempurnakan pernyataan terpilih. Anda tidak dapat mengubah data.
- ✓ **Flashback Table:**  
Fasilitas ini mengijinkan Anda untuk melindungi sebuah tabel ke sebuah waktu sebelumnya. Menyediakan sebuah metode recovery tabel yang secara tiba – tiba dimodifikasi atau dihapus oleh seorang pengguna atau aplikasi. Anda dapat memindahkan data tabel dengan semua atribut yang terhubung dengannya, seperti indeks, trigger, dan lain-lain. Hal ini dapat dilakukan ketika database online dengan memutar balik hanya perubahan pada tabel yang diberikan. Anda dapat mengembalikan sebuah tabel dan isinya ke suatu waktu tertentu atau **user-specified system change number (SCN)**. Gunakan tabel kembali dengan kueri

kembali dan versi baris untuk menemukan sebuah waktu dimana tabel harusnya sudah dipindahkan.

✓ **Flashback by Row Versions:**

Mengijinkan Anda untuk kueri metadata dan data historik di dalam sebuah interval waktu. Anda dapat mengerjakan kueri pada tabel seperti halnya waktu tertentu jam dinding. Pilih kondisi filter yang mengijinkan Anda untuk memperbaiki versi yang berbeda dari baris dalam sebuah tabel yang ada dalam sebuah interval waktu tertentu.



Gambar 66. Contoh aksi terhadap tabel

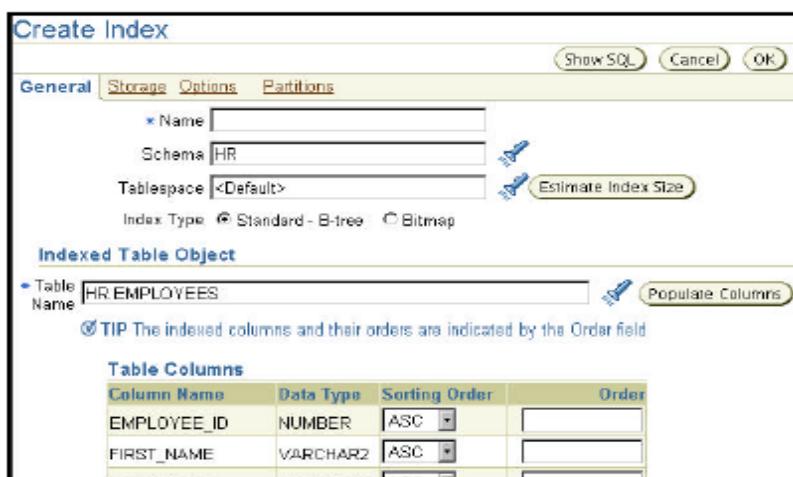
## 5.14. Membuat Indeks

Indeks adalah struktur opsional yang terhubung dengan tabel. Bagian tersebut dapat dibuat untuk menambah kinerja perbaikan data. Sebuah indeks Oracle menyediakan sebuah jalur akses langsung ke data tabel.

Indeks dapat dibuat pada satu atau lebih kolom dari sebuah tabel. Setelah sebuah indeks dibuat, maka secara otomatis akan terpelihara dan digunakan oleh server Oracle. Update ke sebuah data tabel, seperti menambah baris baru, update baris, atau menghapus baris, secara otomatis disebarluaskan ke semua indeks yang relevan dengan transparansi lengkap untuk pengguna membuat perubahan.

Anda dapat mengklik link **Indexes** di bawah kepala **Schema** dari halaman **Administration** untuk menampilkan halaman **Indexes**. Anda dapat menampilkan atribut indeks, atau menggunakan menu **Actions** ke **View Dependencies** untuk sebuah indeks.

Indeks dapat dibuat dengan jelas, atau secara jelas melalui konstrain yang mengantikan tabel.



Gambar 67. Create Index

# BAB VI

## BACKUP DATABASE

### 6.1. Tujuan

Setelah menyelesaikan pelajaran ini, Anda diharapkan dapat melakukan hal – hal sebagai berikut:

- ✓ Membuat backup database yang konsisten.
- ✓ Membackup database Anda tanpa mematikannya.
- ✓ Membuat backup tambahan.
- ✓ Mengotomatisasi backup database.
- ✓ Mengawasi flash recovery area.

### 6.2. Terminologi

Keseluruhan backup database termasuk semua file data dan setidaknya satu control file (ingat bahwa semua control file dalam sebuah database adalah identik).

Sebagian backup database mungkin juga termasuk kosong atau lebih tablespace, kosong atau lebih file data, dan mungkin atau juga tidak mungkin termasuk sebuah control file.

Backup penuh membuat sebuah salinan dari semua blok data dalam file yang sedang dibackup yang mengandung data.

Backup tambahan membuat sebuah salinan dari semua blok data yang sudah berubah sejak pada beberapa backup sebelumnya. Oracle Database 10g mendukung dua level dari backup tambahan (0 – 1). Sebuah level 0 atau backup dasar adalah sama dengan sebuah backup penuh dan mengandung semua blok data. Sebuah level 1 tambahan membackup semua

blok database yang berubah sejak pada backup level 0. Untuk mengembalikan menggunakan backup tambahan, backup dasar harus dikembalikan dulu pertama kali, dan kemudian tambahannya.

Backup offline (juga dikenal sebagai backup konsisten) akan difungsikan ketika database tidak dibuka. Bagian tersebut konsisten karena pada waktu backup, kepala file data SCN menyesuaikan dengan SCN dalam control file.

Backup online (juga dikenal sebagai backup panas atau inkonsisten) akan difungsikan ketika database terbuka. Backup inkonsisten karena dengan database yang terbuka maka tidak ada jaminan bahwa file data akan sinkronisasi dengan control file. Backup inkonsisten memerlukan recovery dengan tujuan akan dipakai.

Salinan image merupakan duplikat dari data atau archived log files (serupa dengan salinan sederhana file menggunakan perintah sistem operasi).

Backup sets merupakan salinan dari satu atau lebih data atau archived log files. Dengan backup sets, data kosong tidak disimpan, dengan demikian menyebabkan backup sets menggunakan sedikit ruang pada disk atau tape. Backup sets dapat dikecilkan untuk mengurangi permintaan ruang dari backup.

Salinan image harus dibackup ke disk. Backup sets dapat dikirim ke disk atau secara langsung ke tape.

Keuntungan dari membuat sebuah backup sebagai sebuah salinan image adalah perbaikan satu per satu dari operasi pengembalian. Dengan sebuah salinan image hanya file yang diperlukan akan diperbaiki dari tape. Dengan backup sets backup set tertentu akan diperbaiki dari tape sebelum Anda mengekstrak file yang diperlukan.

Keuntungan dari membuat sebuah backup sebagai sebuah backup sets adalah penggunaan ruang yang lebih baik. Kebanyakan database mengandung 20% atau lebih blok kosong.

Salinan image mengembalikan semua blok data atas tunggal, meskipun jika blok data tersebut kosong. Backup sets secara signifikan mengurangi kebutuhan ruang backup. Dalam kebanyakan sistem, keuntungan dari backup sets jauh lebih besar daripada keuntungan dari salinan image.

Database dalam mode **NOARCHIVELOG** harus berjalan offline, penuh, keseluruhan backup database.

Database dalam mode **ARCHIVE LOG** mempunyai akses ke jangkauan penuh dari pilihan backup.

Intinya adalah:

- ✓ Strategi backup dapat termasuk:
  - Database tertentu (keseluruhan).
  - Seporsi database (sebagian).
- ✓ Tipe backup dapat merupakan:
  - Semua informasi dari semua file data (penuh).
  - Hanya informasi yang telah berubah sejak pada beberapa backup sebelumnya (tambahan).
- ✓ Mode backup dapat merupakan:
  - Offline (konsisten, dingin).
  - Online (inkonsisten, panas).
- ✓ Backup dapat juga disimpan sebagai:
  - Salinan image.
  - Backup sets.

### 6.3. Recovery Manager (RMAN)

Recovery Manager merupakan komponen dari Oracle Database 10g yang digunakan untuk melakukan backup dan operasi recovery. RMAN dapat menjadi backup konsisten dan inkonsisten, mengerjakan backup tambahan atau penuh, dan membackup database keseluruhan yang lainnya atau seporsi darinya.

RMAN menggunakan pengendalian kerjanya sendiri yang powerful dan menskrip bahasa, sebaik yang dipublikasikan API

yang mengantarmuka RMAN dengan banyak solusi software backup populer.

RMAN dapat menyimpan backup pada disk untuk recovery cepat, atau meletakkan bagian tersebut pada tape untuk penyimpanan jangka panjang. Dengan tujuan untuk RMAN untuk menyimpan backup pada tape, sebuah antarmuka ke peralatan tape dikenal sebagai Media Management Layer (MML) yang harus dikonfigurasi.

Enterprise Manager menyediakan sebuah antarmuka grafik pada kebanyakan penggunaan umum fungsi RMAN. Backup ahli dan operasi recovery dapat diakses melalui klien garis perintah RMAN.

Intinya adalah Enterprise Manager menggunakan RMAN untuk melakukan backup dan operasi recovery:

- ✓ Klien garis perintah untuk fungsi ahli.
- ✓ Pengendalian powerful dan skrip bahasa.
- ✓ Mempublikasikan API yang mengijinkan antarmuka dengan kebanyakan software backup populer.
- ✓ Membackup data, control, archive log, dan file parameter server.
- ✓ Membackup file ke disk atau tape.

#### 6.4. Menkonfigurasi Pengaturan Backup

Caranya adalah dengan menavigasikan ke halaman **Maintenance** dan klik **Configure Backup settings**. Dari halaman properti Anda mengatur setting backup yang ada yang akan digunakan untuk membuat backup. Ada setting yang terpisah antara disk dan tape. Setting tape bergantung pada kemampuan MML. Setting disk termasuk:

##### ✓ **Parallelism:**

Berapa banyak aliran terpisah dari informasi database yang Anda ingin buat. Setting terbaik untuk **Parallelism** tergantung pada hardware Anda. Sebuah CPU tunggal, disk controller tunggal, atau disk server tunggal tidak akan

menguntungkan dari pengkonduksian backup parallel. Sebagai penambahan sumber daya hardware, derajat yang layak dari **Parallelism** juga bertambah.

✓ **Disk Backup Location:**

Dimana seharusnya backup disimpan ? Defaultnya adalah pada Flash Recovery Area. Jika Anda mengubahnya, klik **“Test Disk Backup”** untuk memverifikasi bahwa RMAN dapat menulis pada lokasi baru.

✓ **Disk Backup Type:**

Pilih **Image Copy**, **Backup Set**, atau **Compressed Backup Set**.

Klik **Backup Set** untuk mengatur ukuran maksimum dari file backup set (backup set dapat dibagi nantinya jika diperlukan untuk pencapaian yang lebih mudah).

Host credential dibutuhkan untuk Enterprise Manager untuk menyimpan semua perubahan pada setting backup.

Klik tab **Policy** untuk:

- ✓ Mengotomatisasi backup control dan SPFILE dengan setiap backup. Anda juga diberi kesempatan untuk menspesifikasikan sebuah lokasi untuk backup tersebut jika Anda tidak ingin bagian tersebut pergi ke flash recovery area.
- ✓ Mengoptimalkan backup dengan cara tidak membackup file – file yang sudah sesuai dengan sebuah file yang merupakan bagian dari backup yang ditahan. Setting ini mengijinkan Anda untuk menskip file data read – only dan offline.
- ✓ Memungkinkan menahan perubahan jalur dan menspesifikasi sebuah lokasi untuk file tracking. Jika Anda ingin membuat backup tambahan maka setting ini dapat mengurangi permintaan waktu untuk memilih blok mana yang dimasukkan dalam backup tambahan.
- ✓ Mengecualikan tablespace dari keseluruhan backup database. Beberapa administrator memilih untuk tidak membackup tablespace yang mengandung data atau obyek

yang dapat dengan mudah dibuat ulang (seperti indeks atau data yang merupakan frekuensi beban batch).

- ✓ Kebijakan penyimpanan: Seberapa lama RMAN harus menyimpan backup Anda ? Jika Anda menggunakan flash recovery area untuk menyimpan backup, RMAN secara otomatis menghapus backup lama untuk menyiapkan ruang untuk yang baru (jika kebijakan penyimpanan mengijinkannya). Secara default hanya backup terakhir yang akan disimpan. Kebijakan penyimpanan ini dapat dispesifikasikan sebagai sejumlah backup atau sejumlah hari.

**Configure Backup Settings**

Device [Backup Set](#) Policy

**Disk Settings**

Parallelism  Concurrent streams to disk drives [Test Disk Backup](#)

Disk Backup Location

An existing directory or diskgroup name where database files will be backed up. If you do not specify a location, database files will be backed up to the flash recovery area location.

Disk Backup Type  **Backup Set**  
An Oracle proprietary format which has to be restored before use.

**Compressed Backup Set**  
An Oracle proprietary format in compressed format which has to be restored before use.

**Image Copy**  
A bit-for-bit copy of database files that can be used as-is to perform recovery.

**Host Credentials**

To save the backup settings, supply operating system login credentials.

\* Username

\* Password

Save as Preferred Credential

**Backup Policy**

Automatically backup the control file and server parameter file (SPFILE) with every backup and database structural change.

Autobackup Disk Location  An existing directory or diskgroup name where the control file and server parameter file will be backed up. If you do not specify a location, the files will be backed up to the flash recovery area location.

Optimize the whole database backup by skipping unchanged files such as read-only and offline datafiles that have been backed up.

Enable block change tracking for faster incremental backups.

Block Change Tracking File  Specify a location and file, otherwise an Oracle managed file will be created in the database area.

**Tablespaces Excluded From Whole Database Backup**

Populate this table with the tablespaces you want to exclude from a whole database backup. Use the Add button to add tablespaces to this table.

Select Tablespace Name	Tablespace Number	Status	Contents
No items Selected			

 **TIP** These tablespaces can be backed up separately using tablespace backup.

**Retention Policy**

Retain All Backups  
You must manually delete any backups.

Retain backups that are necessary for a recovery to any time within the specified number of days (point-in-time recovery) Days  Recovery Window

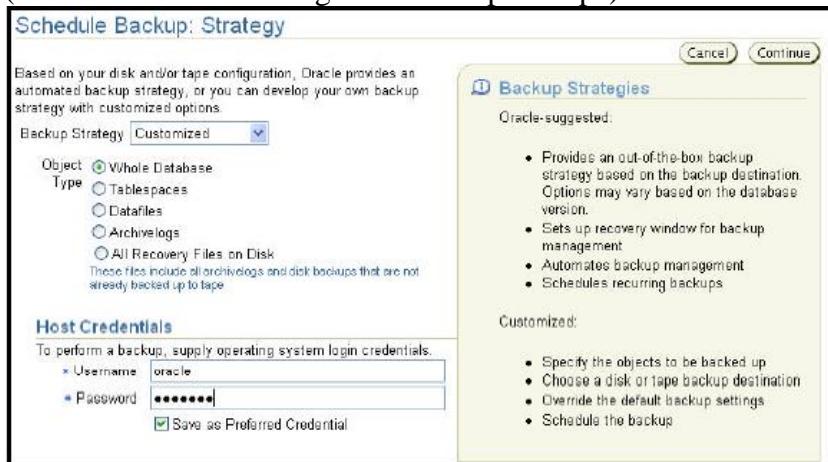
Retain at least the specified number of full backups for each datafile Backups  Redundancy

Gambar 68. Backup (1)

## 6.5. Menjadwalkan Backup: Strategy

Klik **Schedule Backup** dari bagian **Backup/Recovery** dari halaman properti **Maintenance**. Pilih antara **Oracle-suggested backup strategy** atau **customized strategy** Anda sendiri. **Oracle-suggested backup strategy** membuat sebuah keseluruhan database lengkap sekaligus, tambahan dasar level 0, backup online, dan kemudian menjadwalkan backup tambahan level 1 untuk setiap hari berikutnya.

Dengan memilih **Customized**, Anda memperoleh akses ke jangkauan yang lebih luas dari pilihan konfigurasi. Pilihlah obyek mana yang Anda ingin backup – database keseluruhan (default) atau tablespace individual, file data, archivelog, atau backup Oracle apapun yang ditempatkan kembali pada disk (untuk memindahkan bagian tersebut pada tape).



Gambar 69. Backup (2)

## 6.6. Menjadwalkan Backup: Option

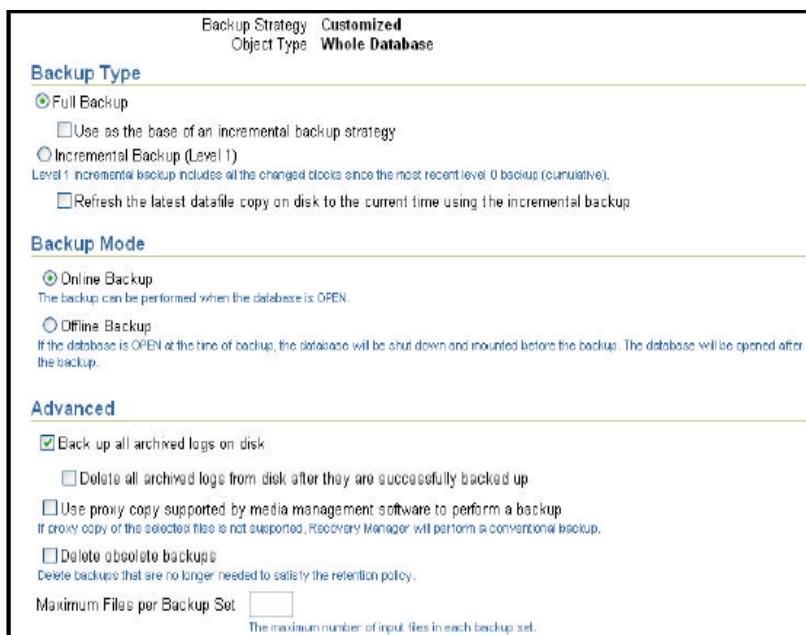
Pilih level backup penuh atau tambahan. Jika mengerjakan backup database penuh, Anda dapat memilih “**Use as the base of an incremental backup strategy**” untuk membuat backup database penuh sebuah level 0 tambahan. Jika menggunakan salinan image, pilihlah kotak cek “**Refresh the latest datafile copy on disk to the current time using the incremental backup**” yang mengupdate backup yang ada lebih daripada membuat sebuah salinan image baru.

Pilih mode backup online atau offline. Ingatlah bahwa mode database **NOARCHIVELOG** harus mengerjakan backup offline.

Anda juga dapat memilih untuk membackup file archive log dan menghapus bagian tersebut dari disk ketika bagian tersebut sudah sukses dibackup.

Klik **Delete obsolete backups** untuk menghilangkan semua backup yang terjatuh di luar kebijakan penyimpanan yang Anda konfigurasi sebelumnya. RMAN akan secara otomatis menyinkirkan **obsolete backups** jika Anda membackup ke flash recovery area.

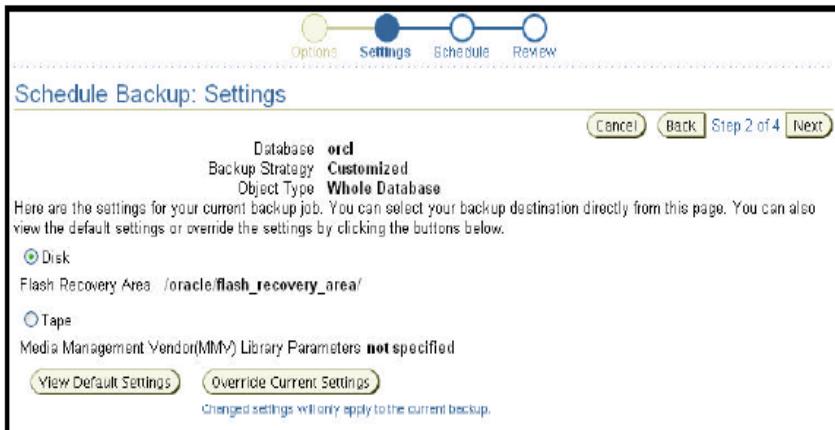
Pilih jumlah file maksimum yang ingin Anda tempatkan pada setiap pengaturan backup. Membatasi jumlah file per pengaturannya dapat mempercepat proses pengembalian karena ukuran dari pengaturan backup yang harus dikembalikan dari tape akan menjadi lebih kecil daripada jika sebuah backup tunggal yang mengandung semua file data untuk database yang telah dibuat.



Gambar 70. Backup (3)

## 6.7. Menjadwalkan Backup: Settings

Pengaturan konfigurasi backup yang tetap dapat ditolak untuk backup ini dengan cara mengklik **Override Current Settings**.

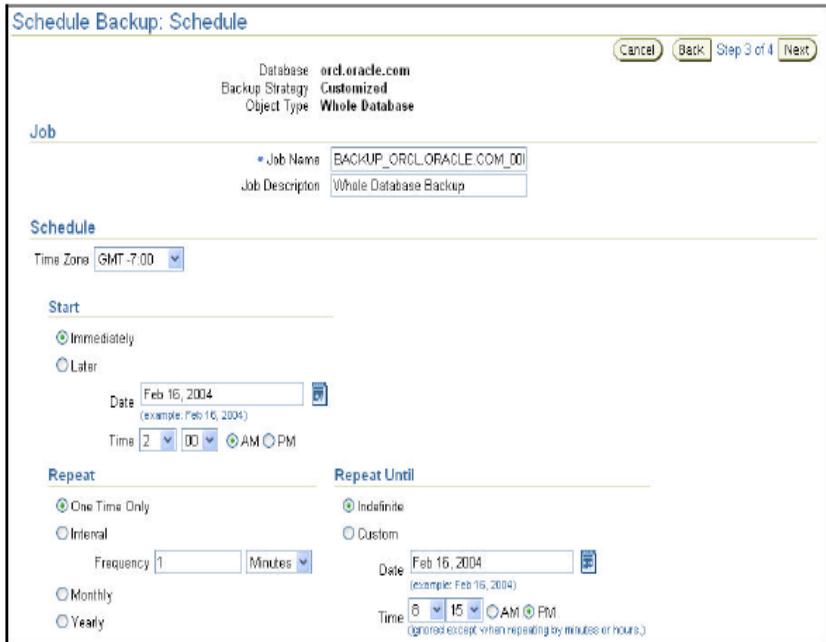


Gambar 71. Backup (4)

## 6.8. Menjadwalkan Backup: Schedule

Pilihlah bagaimana Anda ingin backup dijadwalkan – bisa saja merupakan pekerjaan saat itu atau sebagai sebuah pekerjaan otomatis, proses yang berulang.

Untuk mengkonfigurasi sebuah database untuk recovery maksimum, Oracle menyarankan untuk secara reguler menjadwalkan backup. Mengotomatisasi backup dapat meringankan beban dari administrator.



Gambar 72. Backup (5)

## 6.9. Menjadwalkan Backup: Review

RMAN menggunakan sintaks perintah dan bahasa skripnya sendiri. Tombol **Edit RMAN Script** memberi Anda pandangan perintah penjadwalan backup yang telah dilakukan berdasarkan masukan Anda.

Dari halaman ini Anda dapat meng-customize skrip RMAN jika diperlukan, atau menyalin bagian tersebut untuk keperluan perekaman.

Klik **Edit RMAN Script** untuk menampilkan perintah – perintah RMAN.



Gambar 73. Backup menggunakan RMAN Script

## 6.10. Control File Backup to Trace

Klik **Controlfiles** pada bagian **Storage** dari halaman properti **Administration** untuk mengatur control file database Anda. Control file mempunyai sebuah pilihan backup tambahan; bagian tersebut mungkin dapat dibackup ke trace. Sebuah backup trace control file mengandung pernyataan SQL yang diperlukan untuk membuat ulang contro file dalam kejadian bahwa semua control file hilang.

Meskipun sangat tidak menyenangkan dimana sebuah database terkonfigurasi yang cocok, dengan salinan jamak dari control file yang ditempatkan pada disk dan pengendali yang terpisah, akan menghilangkan semua control file pada saat yang sama, mungkin dapat terjadi dan kemudian administrator harus mengembalikan control file di atas trace setelah setiap perubahan ke struktur fisik dari database (menambah tablespace atau file data, menambah kelompok redo log tambahan).

Salinan trace dari control file dapat dibuat menggunakan Enterprise Manager dengan cara mengklik **Controlfiles** dari

halaman propertis **Administrator**, atau dengan perintah SQL:

```
SQL> ALTER DATABASE BACKUP
CONTROLFILE TO TRACE ;
```

Backup trace dibuat pada lokasi yang dispesifikasi oleh parameter inisialisasi `USER_DUMP_DEST` dengan sebuah nama file seperti `sid_ora_pid.trc`.

File trace mengandung informasi tentang tujuan archive log yang diikuti oleh perintah yang membuat pemindahan control file dan kemudian melindungi database:



Gambar 74. Backup Control File

## 6.11. Manage Current Backups

Klik **Manage Current Backups** dari halaman propertis **Administration** untuk mengatur backup Anda yang sudah ada. Dari halaman ini Anda dapat melihat ketika sebuah backup lengkap, dimana backup ini dibuat (disk atau tape), dan apakah masih ada.

Pada atas halaman **Manage Current Backups** Anda dapat melihat empat tombol yang mengijinkan Anda untuk bekerja dengan backup yang telah ada:

- ✓ **Catalog Additional Files:**  
Meskipun RMAN (bekerja melalui Enterprise Manager) merupakan cara yang direkomendasikan untuk membuat backup, hal ini memungkinkan untuk membuat salinan image pada cara yang lain (sebagai contoh, dengan cara mematikan database dan menyalin semua file). RMAN dan Enterprise Manager tidak akan tahu tentang backup tersebut kecuali jika Anda menambahkan bagian tersebut ke katalog.
- ✓ **Crosscheck All:**  
RMAN dapat secara otomatis menghapus obsolete backups, tetapi Anda juga dapat menghapus bagian tersebut dengan menggunakan perintah sistem operasi. Jika Anda menghapus sebuah backup di luar RMAN, maka katalog tidak akan tahu backup tersebut hilang sampai Anda mengerjakan sebuah crosscheck di antara katalog dan apa yang sebenarnya ada di sana.
- ✓ **Delete All Obsolete:**  
Menghapus backup yang lama daripada kebijakan penyimpanan.
- ✓ **Delete All Expired:**  
Menghapus daftar katalog untuk semua backup yang tidak ditemukan ketika crosscheck dikerjakan.

Klik **Edit RMAN Script** untuk menampilkan perintah – perintah RMAN.

Select	Key/Tag	Completion Time	Contents	Type	Status	Obsolete	Keep	Pieces
<input type="checkbox"/>	3 BACKUP_ORCL_000006_120303103223	Dec 3, 2008 10:48:48 AM	ARCHIVED LOG	DISK	AVAILABLE	NO	NO	1
<input type="checkbox"/>	2 BACKUP_ORCL_000006_120303103223	Dec 3, 2008 10:41:41 AM	DATAFILE SPFILE CONTROLFILE	DISK	AVAILABLE	NO	NO	1

Gambar 75. Manage Backup

## 6.12. Flash Recovery Area

Flash recovery area adalah sebuah ruang yang ditempatkan pada disk untuk mengandung archived log, backup, dan flashback log.

Jika Anda mengkonfigurasi archived log Anda untuk ditulis pada lokasi ini (dengan bendera USE\_DB\_RECOVERY\_AREA dalam satu lokasi), maka penting untuk memonitor ruang ini untuk memastikan bahwa kapasitasnya tidak tercapai. Jika perangkat tidak memungkinkan untuk membuat sebuah archive log karena kehabisan ruang, maka akan terdiam sampai administrator membetulkan situasi tersebut.

Klik **Configure Recovery Settings** dari halaman properti **Maintenance** yang akan membawa Anda pada setting **Flash Recovery Area**. Pada halaman ini Anda dapat:

- ✓ Menspesifikasi lokasi dari flash recovery area.
- ✓ Menspesifikasi ukuran dari flash recovery area (Oracle menyarankan paling tidak dua kali ukuran dari database jadi dapat menampung sebuah backup dan beberapa archived log).
- ✓ Memverifikasi berapa banyak flash recovery area yang telah dikonsumsi.
- ✓ Mengkonfigurasi database flashback.



Gambar 76. Manage Flash Recovery Area

## BAB VII

# RECOVERY DATABASE

### 7.1. Tujuan

Setelah menyelesaikan pelajaran ini, Anda diharapkan dapat melakukan perlindungan dari kehilangan hal – hal sebagai berikut:

- ✓ Control file.
- ✓ Redo log file.
- ✓ File data.

### 7.2. Membuka Sebuah Database

Apabila sebuah database dipindahkan dari keadaan mati ke keadaan hidup seutuhnya, maka akan melakukan pengecekan konsistensi internal pada setiap keadaan.

#### ✓ **NOMOUNT:**

Agar perangkat dapat mencapai status **NOMOUNT** (juga dikenal sebagai **STARTED**), perangkat harus membaca file parameter inisialisasi. Tidak ada file database yang diperiksa pada saat mencapai **NOMOUNT**.

#### ✓ **MOUNT:**

Ketika perangkat bergerak pada status **MOUNT**, memeriksa semua control file yang terdaftar dalam file parameter inisialisasi yang ada dan disinkronisasi. Walaupun jika satu control file hilang atau korup, perangkat akan mengembalikan error tersebut kepada administrator

mengenai kehilangan control file dan akan kembali pada keadaan **NOMOUNT**.

✓ **OPEN:**

Ketika perangkat bergerak dari keadaan **MOUNT** ke **OPEN** maka:

- Memeriksa semua kelompok redo log yang dikenali control file yang paling tidak memiliki satu anggota yang ada. Anggota lainnya yang hilang akan dicatatkan pada log peringatan.
- Memverifikasi semua file data yang dikenali control file yang ada setidaknya ketika bagian tersebut dalam kondisi offline. File offline tidak diperiksa sampai administrator mencoba untuk membawa bagian tersebut online. Administrator mungkin mengambil sebuah file data offline dan membuka perangkat sepanjang file data tidak termasuk pada tablespace SYSTEM atau UNDO. Jika ada file yang hilang maka sebuah error akan dicatatkan file hilang pertama yang dikembalikan ke administrator dan perangkat akan kembali pada keadaan **MOUNT**. Ketika perangkat menemukan file yang hilang, hanya file pertama penyebab masalah timbul dalam pesan error. Untuk menemukan semua file yang memerlukan recovery, administrator dapat memeriksa tampilan kinerja dinamis v\$recover\_file untuk mendapatkan sebuah daftar lengkap file yang memerlukan perhatian.
- Memverifikasi semua file data yang tidak offline atau read-only yang disinkronisasi dengan control file. Jika perlu, perangkat recovery akan dilakukan secara otomatis. Meskipun jika sebuah file keluar dari sinkronisasi lebih dari yang bisa dilindungi menggunakan kelompok redo log online, kemudian administrator harus mengerjakan media recovery. Jika ada file yang membutuhkan media recovery, maka

sebuah pesan error akan memunculkan file pertama yang membutuhkan recovery yang akan dikembalikan pada administrator dan perangkat akan kembali pada keadaan **MOUNT**.

ORA-01113: file 4 needs media recovery

ORA-01110: data file 4: ‘ /oracle /oradata /orcl /users01 . dbf ‘

Kembali, v\$recover\_file memberikan sebuah daftar lengkap file yang memerlukan perhatian. File ditampilkan tetapi butuh media recovery akan didaftarkan, tetapi tidak akan mempunyai sebuah pesan error.

### 7.3. Mengubah Status Perangkat

Ketika memulai perangkat, mode start defaultnya adalah **OPEN**. Anda dapat memilih untuk memulai perangkat dalam beberapa mode yang lainnya, atau masalah dengan database mungkin memaksa Anda untuk memulai pada mode yang lainnya. Layar properti **Advanced Startup Options** mengijinkan Anda untuk memilih sebuah keadaan lain daripada **OPEN** ketika memulai perangkat dan untuk mempertahankan keadaan jika perangkat sudah siap dijalankan pada mode yang lain. Anda juga dapat menggunakan perintah SQL untuk memodifikasi status perangkat:

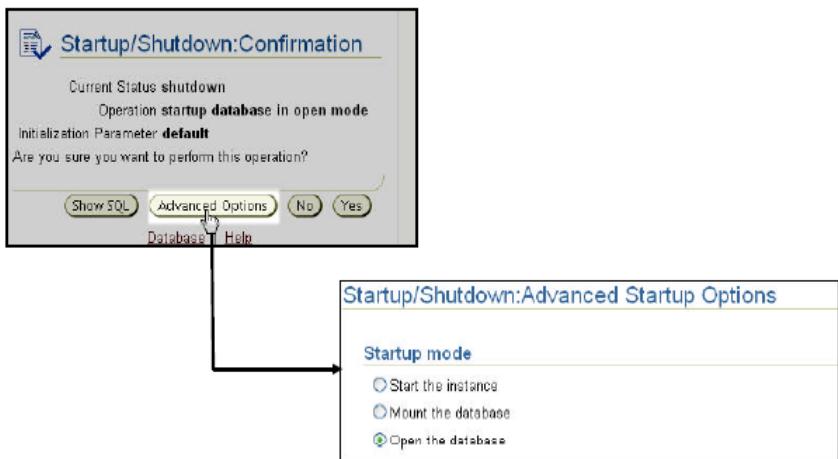
```
SQL> STARTUP NOMOUNT  
ORACLE instance started .
```

bytes	Total System Global Area	188743680
bytes	Fixed Size	778036
bytes	Variable Size	162537676

Database Buffers	25165824
bytes	
Redo Buffers	262144
bytes	

SQL> ALTER DATABASE MOUNT  
DATABASE altered .

SQL> ALTER DATABASE OPEN



Gambar 77. Konfirmasi Shutdown/Startup

#### 7.4. Menjaga Agar Sebuah Database Tetap Terbuka

Setelah sebuah perangkat terbuka, kegagalan media menyebabkan kehilangan sebuah control file, kehilangan sejumlah anggota kelompok redo log, atau kehilangan sebuah file data yang termasuk pada tablespace SYSTEM atau UNDO yang akan menyebabkan perangkat gagal.

Dalam banyak kasus kegagalan perangkat tidak akan sepenuhnya mati, tetapi akan tidak dapat melanjutkan pekerjaan.

Melindungi dari berbagai macam kegagalan media harus dilakukan dengan database turun, jadi administrator harus menggunakan perintah **SHUTDOWN ABORT** sebelum memulai usaha recovery.

Kehilangan file data termasuk pada tablespace lainnya tidak menyebabkan kegagalan perangkat, dan database akan direcover ketika terbuka dengan melanjutkan pekerjaan pada tablespace yang lainnya.

Intinya, sekali terbuka maka perangkat akan gagal dengan:

- ✓ Kehilangan control file.
- ✓ Kehilangan sebuah file data yang termasuk pada tablespace **SYSTEM** atau **UNDO**.
- ✓ Kehilangan sebuah kelompok redo log tertentu. Sepanjang setidaknya satu anggota kelompok ada, perangkat akan kembali terbuka.

## 7.5. Kehilangan Sebuah Control File

Recover dari kehilangan sebuah control file (selama setidaknya ada satu control file) dapat dicapai sebagai berikut:

1. Jika perangkat tidak gagal, matikanlah dengan menggunakan **SHUTDOWN ABORT**.
2. Salin satu dari control file yang ada ke lokasi kehilangan file. Jika kegagalan media dikarenakan kehilangan sebuah disk drive atau pengendali, salinlah satu dari control file yang ada ke beberapa lokasi lainnya dan update file parameter perangkat untuk menunjukkan lokasi baru. Alternatif lainnya, Anda dapat menghapus referensi ke kehilangan control file dari file parameter inisialisasi. Ingat bahwa Oracle menyarankan mempunyai setidaknya dua control file pada semua waktu.
3. Jalankan perangkat.

Intinya jika sebuah control file hilang atau korup maka:

1. Perangkat biasanya akan digugurkan. Jika masih terbuka, matikanlah.
2. Pindahkan control file yang hilang dengan cara menyalin sebuah control file yang ada.
3. Jalankan perangkat.

## 7.6. Kehilangan Sebuah Redo Log File

Recover dari kehilangan anggota kelompok log tunggal seharusnya tidak berakibat pada perangkat yang sedang berjalan.

1. Pastikan bahwa ada sebuah file log hilang dengan cara menguji log peringatan.
2. Pindahkan file hilang dengan cara menyalin satu dari file yang ada dari kelompok yang sama.
3. Jika kegagalan media dikarenakan kehilangan disk drive atau pengendali, namai kembali file yang hilang tersebut.
4. Jika kelompok sudah diarsip, atau jika Anda berada pada mode noarchivelog, Anda dapat memilih untuk memecahkan masalah dengan cara membersihkan kelompok log untuk membuat ulang file yang hilang. Pilih kelompok yang sesuai dan pilih aksi **Clear Logfile**. Anda juga dapat membersihkan kelompok yang terpengaruh secara manual dengan perintah:  
`SQL> ALTER DATABASE CLEAR LOGFILE GROUP # ;`

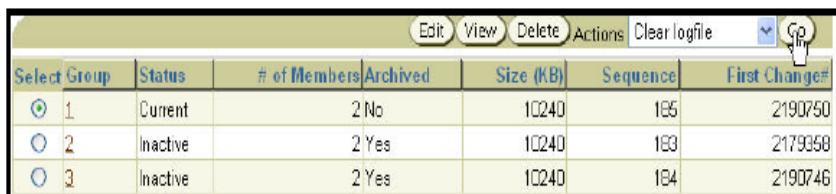
Database Control tidak akan mengijinkan Anda untuk membersihkan sekelompok log yang belum pernah diarsipkan sebelumnya. Lakukan beberapa pemecahan rantai dari informasi redo. Jika Anda harus membersihkan sekelompok log yang tidak diarsipkan, Anda harus dengan segera mengambil backup penuh dari keseluruhan database. Kegagalan untuk melakukannya akan menyebabkan kehilangan data jika kegagalan lainnya terjadi.

Untuk membersihkan kelompok log tidak terarsip gunakan perintah ini:

```
SQL> ALTER DATABASE CLEAR UNARCHIVED  
LOGFILE GROUP #;
```

Intinya jika sejumlah anggota dari sekelompok file log hilang, setidaknya kelompok masih mempunyai satu anggota:

1. Operasi normal perangkat tidak akan terpengaruh.
2. Anda akan menerima pesan dalam log peringatan yang memperingati Anda bahwa sebuah anggota tidak dapat ditemukan.
3. Pindahkan file log yang hilang dengan cara menyalin satu dari file yang ada dari kelompok yang sama.



Select	Group	Status	# of Members	Archived	Size (KB)	Sequence	First Change#
<input checked="" type="radio"/>	1	Current	2	No	10240	185	2190750
<input type="radio"/>	2	Inactive	2	Yes	10240	183	2179358
<input type="radio"/>	3	Inactive	2	Yes	10240	184	2190746

Gambar 78. Archive Log

## 7.7. Kehilangan Sebuah File Data pada Mode NOARCHIVELOG

Kehilangan file data dari sebuah database pada mode NOARCHIVELOG memerlukan pemindahan lengkap database termasuk control file dan semua file data.

Dengan database pada mode NOARCHIVELOG, recovery hanya mungkin terjadi sampai waktu backup terakhir, jadi pengguna harus memasukkan kembali semua perubahan yang dibuat sejak backup tersebut dilakukan.

1. Matikan perangkat jika belum mati.
2. Klik **Perform Recovery** dari halaman properti **Maintenance**.

3. Pilih “**Whole Database**” sebagai tipe recovery.

Intinya jika database dalam keadaan mode NOARCHIVELOG, dan ada file data yang hilang maka:

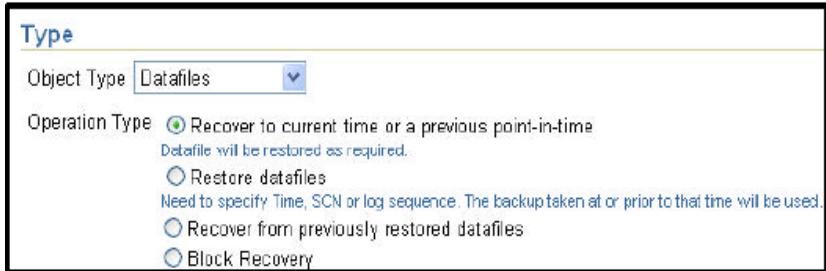
1. Matikan perangkat jika belum mati.
2. Kembalikan database yang ada, termasuk semua data dan control file, dari backup.
3. Buka database.
4. Pastikan pengguna memasukkan kembali semua perubahan yang dibuat sejak backup terakhir.

## **7.8. Kehilangan Sebuah File Data Noncritical Pada Mode ARCHIVELOG**

Dengan database pada mode ARCHIVELOG, kehilangan file data tidak termasuk pada tablespace SYSTEM atau UNDO hanya mempengaruhi obyek yang ada pada file yang hilang. Sisa database kembali ada untuk pengguna untuk melanjutkan kerja. Untuk file data yang hilang:

1. Klik **Perform Recovery** dari halaman properti **Maintenance**.
2. Pilih “**Datafiles**” sebagai tipe recovery dan pilih “**Restore to current time**”.
3. Tambahkan semua file data yang memerlukan recovery.
4. Pastikan apakah Anda ingin memindahkan file ke lokasi default atau (jika kehilangan sebuah disk atau pengendali) ke lokasi yang baru.
5. Masukkan pekerjaan RMAN untuk pemindahan dan recover file yang hilang.

Karena database berada pada mode ARCHIVELOG, memungkinkan recovery terjadi sampai waktu dari peletakan terakhir dan pengguna tidak perlu memasukkan kembali data apapun juga.



Gambar 79. Recovery Datafile

## 7.9. Kehilangan Sebuah File Data SYSTEM-CRITICAL pada Mode ARCHIVELOG

File data yang termasuk pada tablespace SYSTEM atau mengandung data UNDO dianggap sebagai system-critical. Kehilangan salah satu dari file tersebut membuat database harus dipindahkan dari keadaan MOUNT (tidak seperti file data lain yang mungkin dipindahkan ketika database terbuka).

1. Matikan perangkat jika belum mati.
2. Naikkan database.
3. Klik **Perform Recovery** dari halaman properti **Maintenance**.
4. Pilih “**Datafiles**” sebagai tipe recovery dan pilih “**Restore to current time**”.
5. Tambahkan semua file data yang memerlukan recovery.
6. Pastikan apakah Anda ingin memindahkan file ke lokasi default atau ( jika kehilangan sebuah disk atau pengendali) ke lokasi yang baru.
7. Masukkan pekerjaan RMAN untuk pemindahan dan recover file yang hilang.
8. Buka database. Pengguna tidak perlu memasukkan kembali data apapun juga karena recovery akan berlangsung sampai saat pemasukkan terakhir.

Intinya jika sebuah file data hilang atau korup, dan file tersebut termasuk pada tablespace SYSTEM atau UNDO maka:

1. Perangkat mungkin atau mungkin juga tidak mati secara otomatis. Jika tidak mati gunakan **SHUTDOWN ABORT** untuk mematikannya.
2. Naikkan database.
3. Pindah dan recover file data yang hilang.
4. Buka database.

## Daftar Pustaka

- Alapati, S. (2009). *Expert Oracle Database 11g Administration*. Apress.
- Oracle University. (2013). *Oracle Database 11g: Administration Workshop I*. Oracle University.
- Oracle University. (2013). *Oracle Database 11g: Administration Workshop II*. Oracle University.





**Radiant Victor Imbar, S. Kom, M.T., OCP., CISO.**

Penulis adalah Dosen Tetap Fakultas Teknologi Informasi yang menitikberatkan penelitiannya di bidang *Database*, *Enterprise System* dan *Security*.

Buku ini akan menuntun anda langkah per langkah untuk menjadi seorang DBA (Database Administrator) ORACLE yang ahli. Buku ini akan dimulai instalasi Oracle dan membuat database Oracle dan bagaimana pengaturannya, mengatur pengguna, mengatur *schema*, melakukan *backup* dan *recovery* Database.

### **Buku ini membahas :**

- Instalasi Oracle
- Arsitektur Database (*Data File, Control File, Redolog File, Online Redolog File*)
- Struktur Memori Oracle
- Proses-proses di Oracle
- Database Control
- *Grid Control*,
- *Startup* dan *Shutdown* Database
- File Parameter Inisialisasi
- *Alert Log*
- Struktur Penyimpanan Oracle
- Membuat *User, Profile, Roles*
- Mengatur skema Objek
- *Backup* dengan Rman
- Macam-macam *Recovery*

KATEGORI : BASISDATA