

Pelatihan Oracle untuk mahasiswa



Introduction to Oracle



**Oleh :
Eru Puspita
Edi Satriyanto**

**Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
2003**

Teori Database

Database bisa dianggap sebagai kumpulan dari berbagai informasi yang diatur sedemikian rupa dan disimpan pada media tertentu agar informasi tersebut dapat dimanfaatkan/diambil kembali.

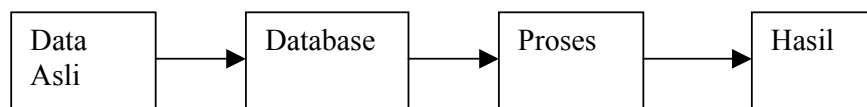
Ada tiga macam pembagian pada sistem Database

- Operasi kerja database
- Bagian-bagian atau komponen sistem database
- Konfigurasi sistem database

Pembagian berdasarkan operasi kerja meliputi

- Penyimpanan
- Pengolahan
- Pembacaan kembali

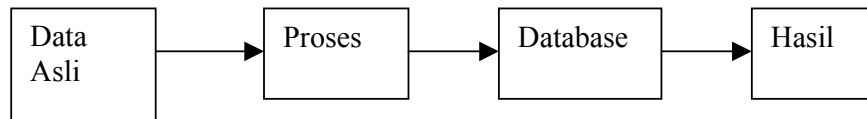
Penyimpanan digunakan untuk memasukkan data-data tertentu ke dalam database. Penyimpanan ini bisa berarti memasukkan data baru atau mengubah data yang lama. Data yang disimpan ke dalam database bisa berupa data mentah (data-data yang belum mengalami pengolahan), atau data-data yang siap digunakan (sudah mengalami pengolahan terlebih dahulu). Bentuk yang pertama memungkinkan proses penyimpanan menjadi sederhana, karena dari data asli langsung dimasukkan ke database. Selain itu, seluruh data asli masih kelihatan (tersedia) yang memungkinkan dilakukan pengolahan sesuai dengan kebutuhan. Bahkan jika kebutuhannya berubah, cukup dilakukan perubahan pada prosesnya, bukan pada datanya. Namun ini membawa konsekuensi, setiap kali diinginkan hasil akhirnya, harus selalu dilakukan pengolahan data.



Gambar Penyimpanan data asli secara langsung

Bentuk yang kedua adalah dari data aslinya, sebelum dilakukan penyimpanan ke database, data asli tersebut diolah terlebih dahulu untuk mendapatkan hasil akhir. Setelah itu baru disimpan ke database. Cara ini mempercepat proses secara keseluruhan, karena pada saat memasukkan data, meskipun dilakukan pengolahan awal, datanya biasanya sedikit-sedikit sehingga tidak terasa.

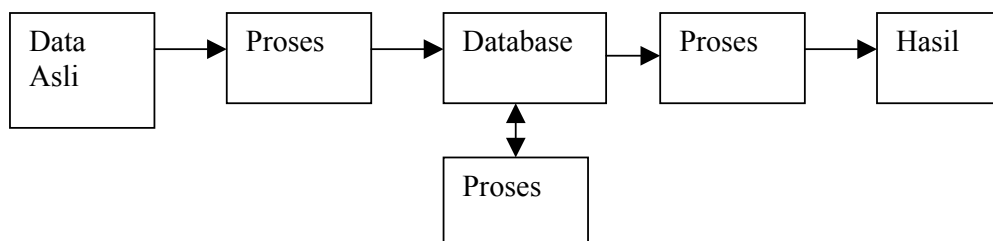
Sedangkan saat membaca hasil akhirnya, cukup dilakukan pembacaan langsung dari tabel yang jelas mempercepat proses. Namun cara ini akan banyak menghilangkan data-data aslinya, dan setiap terjadi perubahan sistem, akan banyak mengalami kesulitan.



Gambar Dilakukan pengolahan data asli sebelum disimpan

Untuk penerapan yang sesungguhnya, dapat dipertimbangkan untuk menggabungkan antara kedua cara tersebut. Misalkan jika ada data aslinya yang memang suatu saat tidak akan pernah digunakan lagi, dapat langsung diolah. Kemudian jika dianggap proses yang akan dilakukan tidak memerlukan waktu yang cukup panjang, sebaiknya data disimpan apa adanya, dengan proses dilakukan di belakang. Bisa juga digunakan data perantara, artinya dari data asli diolah sedikit untuk mendapatkan data pertengahan baru disimpan ke database. Ini akan sangat menghemat waktu.

Bagian Pengolahan digunakan untuk mendapatkan suatu informasi sesuai dengan yang diinginkan dari data aslinya. Ambil contoh data asli mengenai mahasiswa, pengolahan/proses yang bisa dilakukan misalnya menghitung jumlah total mahasiswa, atau berdasar jurusan, kelas dan sebagainya. Proses pengolahan ini bisa dilakukan di awal sebelum data disimpan, bisa pada saat membaca hasil akhir, dapat juga dari data yang sudah disimpan di dalam database diambil kembali untuk diolah kemudian disimpan kembali.

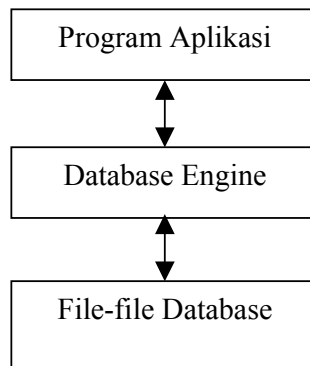


Gambar Proses dalam sistem database

Pembacaan kembali digunakan untuk mengambil data yang ada di dalam data base untuk diolah, kemudian disimpan kembali atau dibuat laporannya. Ada beberapa hal yang bisa dilakukan pada saat pengambilan, antara lain, mengambil seluruh isi dari sebuah tabel database, atau gabungan dari beberapa tabel database. Ini biasanya untuk ditampilkan dalam bentuk tabel. Bisa juga mengambil sebagian dari isi tabel (dengan difilter/diambil dengan kriteria tertentu), atau hanya mengambil satu baris data saja. Semua itu tergantung dari proses atau hasil yang ingin dilakukan.

Bagian-bagian atau komponen dari suatu sistem database secara umum meliputi

- File database/media simpan (DF)
- Program/mesin pengolah database (Database Engine, DE)
- Program aplikasi (AP)



Gambar Konfigurasi sistem database

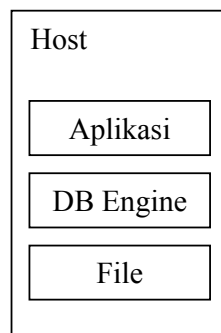
Suatu program aplikasi, saat melakukan operasi dengan database, harus melewati/menggunakan database engine. Perintah-perintah dari program aplikasi seperti membuat tabel, membuka tabel, membaca isi tabel dan sebagainya. Perintah-perintah ini diteruskan ke database engine untuk diolah lebih lanjut.

Database engine bertanggung jawab untuk melakukan operasi pada file database secara fisik, misalkan suatu tabel/isi tabel disimpan pada disk yang mana dan di mana lokasinya. Suatu data pada saat disimpan ke disk, format datanya seperti apa. Dan seterusnya.

Bahasa yang digunakan dari program aplikasi ke database engine hampir sama untuk setiap jenis database, terutama jika menggunakan perintah SQL. Sedangkan database engine harus selalu sesuai untuk setiap jenis database yang digunakan.

Konfigurasi dari sistem database meliputi

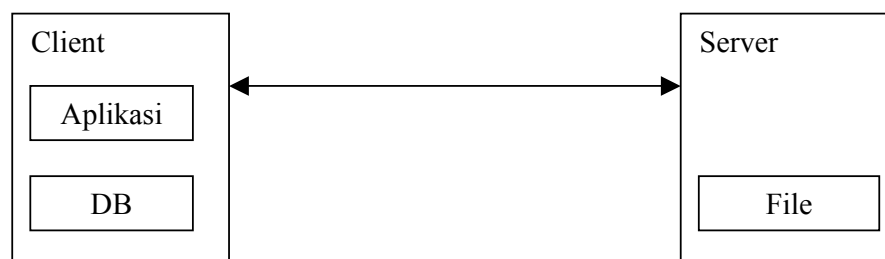
- Database lokal
- Database file server
- Database client server



Gambar Database Lokal

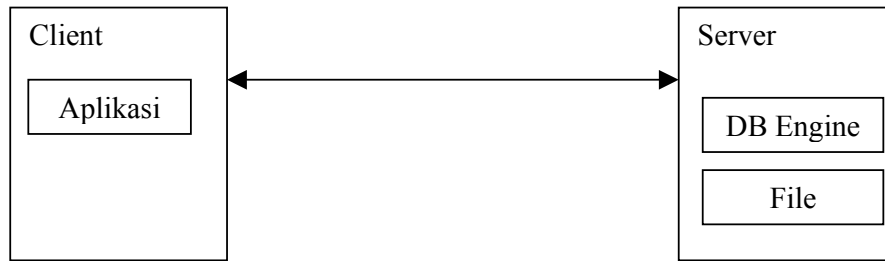
Jika file database, database engine dan program aplikasi terletak pada komputer yang sama, maka konfigurasi ini disebut dengan database lokal. Jika program aplikasi dan database engine terletak pada satu komputer, tetapi file database terletak pada komputer lain, dan terhubung dalam jaringan, maka konfigurasi ini disebut dengan database file server. Sedangkan jika file database dan database engine terletak pada satu komputer dan program aplikasi terletak pada komputer lainnya, maka ini disebut dengan database client server.

Konfigurasi database lokal memiliki konfigurasi yang paling sederhana, paling murah dan paling cepat. Tetapi jelas, jenis ini tidak dapat digunakan dalam mode multi-user atau dalam suatu jaringan.



Gambar Database file server

Database file server memperbaiki fungsi database lokal dengan memungkinkan adanya mode multi-user atau kerja dalam suatu jaringan. Selain itu, server yang digunakan tidak terlalu berat, karena hanya digunakan untuk menampung file database saja. Karena database engine terpisah jauh dengan file database, menyebabkan lalu-lintas jaringan menjadi relatif padat dan ujung-ujungnya sistem menjadi cukup lambat.



Gambar Database Client Server

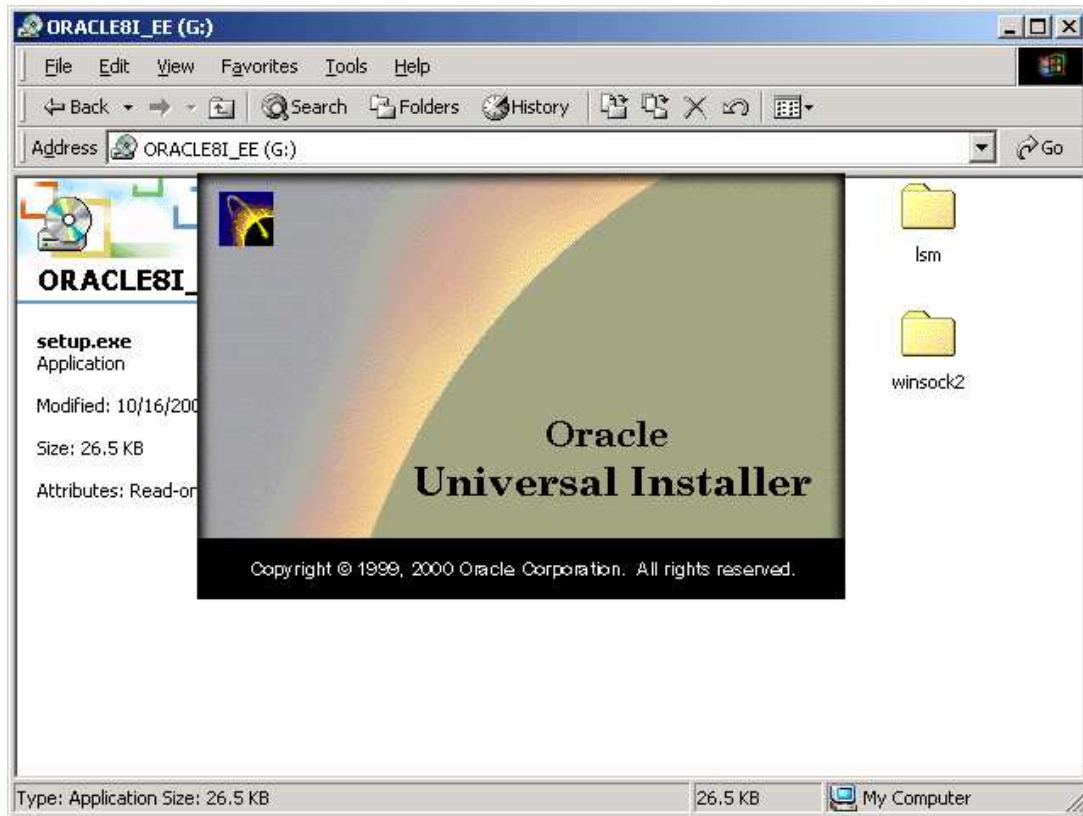
Database client server memperbaiki keduanya, yaitu dengan menawarkan mode multi-user, kerja jaringan dan akses yang cepat. Berhubung server harus menyimpan file database dan database engine, maka kerja server menjadi sangat berat. Karena itu, kinerja sistem database client server sangat ditentukan oleh kemampuan dari komputer server dan oleh kompleksitas dari database engine.

Dilihat dari cara penyimpanan/representasi data dalam database, dapat dibagi menjadi dua struktur database, database berbasis relasi (**relational database**) dan database berbasis obyek (**object oriented database**). Khusus untuk Oracle, struktur yang digunakan adalah berbasis relasi dan juga obyek. Artinya, Oracle dapat dipandang sebagai database relasi biasa, tetapi dapat juga dapat diperlakukan sebagai database obyek. Dalam hal ini, data-data yang disimpan ke dalam database Oracle bisa dalam bentuk object. Misalkan data-data object untuk penyimpanan multimedia.

Secara gampang, database relasi adalah database yang bekerja berdasarkan penyimpanan data dalam bentuk tabel dengan susunan baris dan kolom, dan antara satu tabel dengan tabel lainnya dapat saling dihubungkan/direlasikan. Sedangkan database yang berbasis obyek adalah, setiap data yang ada dianggap sebagai benda, sedangkan setiap benda sangat terkait dengan benda-benda lainnya yang terhubung seperti hirarki suatu keluarga (ada induk/orang tua/leluhur, dan ada anak/keturunan).

Instalasi Oracle pada Sistem Operasi Windows

Langkah Instalasi Gunakan CD installer Oracle Enterprise. saat CD dimasukkan ke CD Drive, program instalasi akan langsung dijalankan.



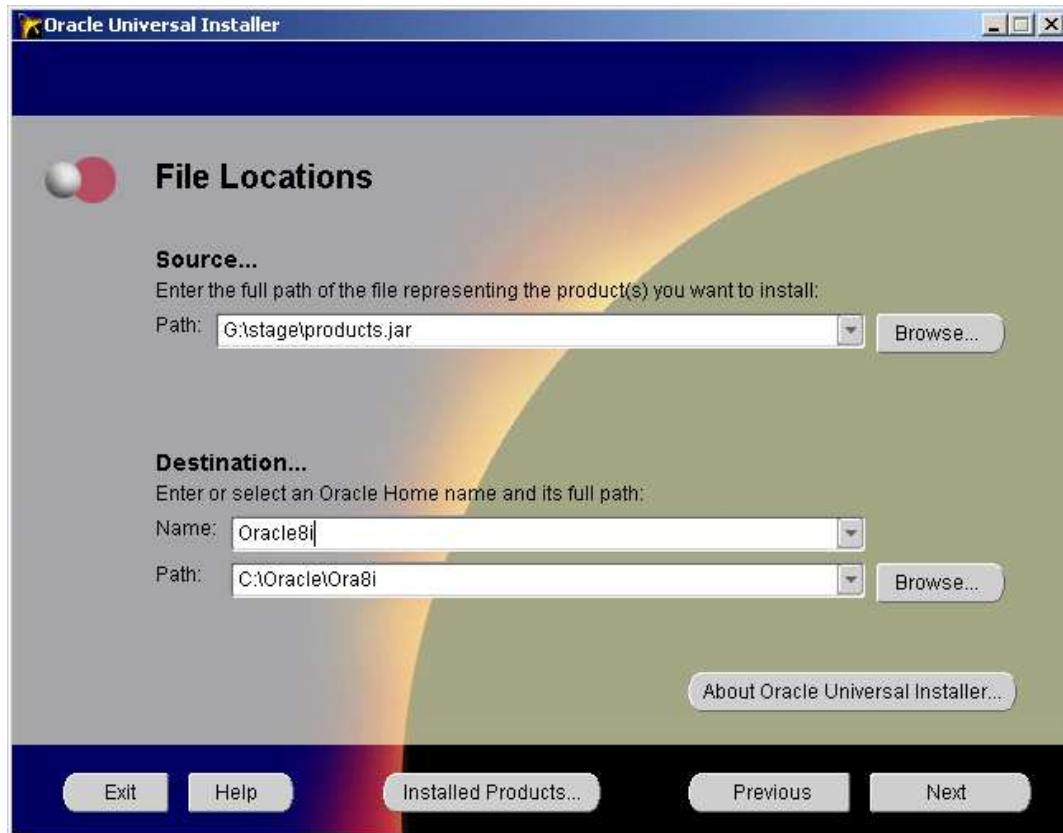
Gambar Saat pertama kali Oracle Installer dijalankan

Beberapa saat kemudian akan muncul menu utama dari program installer. Meliputi melihat komponen oracle yang sudah terinstall di komputer, dan untuk menghapus instalasi sebelumnya (deinstall). Yang perlu diingat, oracle pada saat melakukan instalasi, oracle tidak mendaftarkan produknya yang telah dinstall ke registry uninstall dari window, tetapi menangani sendiri. Karena itu, setiap kali akan melakukan instalasi atau deinstall, silakan masuk ke program ini.



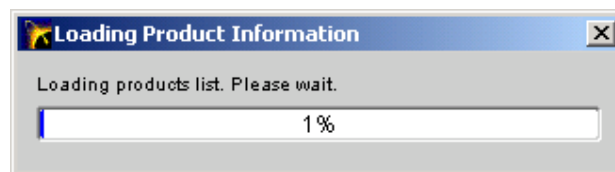
Gambar Menu utama Program Instalasi

Tekan NEXT. Berikutnya akan muncul setting yang meliputi memasukkan nama direktori tujuan, direktori asal program setup, dan nama dari Oracle Home. Ubah sesuai dengan keinginan. Dan tekan NEXT.



Gambar Setting Lokasi instalasi dan nama OracleHome

Sesaat setelah tombol NEXT ditekan, oracle melakukan pembacaan seluruh produk yang tersedia. Ini akan memerlukan waktu yang cukup lama. Tunggu beberapa saat.



Gambar Oracle membaca informasi produk yang tersedia

Jika CD installer yang digunakan adalah oracle server enterprise, maka akan muncul tiga produk utama instalasi oracle, instalasi server, instalasi client dan oracle management server. Dalam hal ini pilih instalasi server.



Gambar Tiga Produk yang disediakan oleh Oracle Inteprise

Berikutnya akan muncul jenis instalasi, typical, integrated, atau custom. Pilihan typical akan melakukan instalasi komponen-komponen oracle yang sesuai dengan kebutuhan oracle server. Ini memerlukan sekitar 779 MB. Sebagai catatan, ini adalah total besar dari seluruh ukuran file yang akan disimpan ke harddisk. Tetapi besar kenyataannya saat disimpan dapat lebih besar (ini disebabkan faktor ukuran sektor/cluster dari media penyimpanan). Pilihan minimal akan melakukan instalasi untuk kebutuhan paling sedikit dari oracle server. Gunakan ini jika ruang disk tidak besar atau memang tidak diperlukan program-program tambahan kecuali oracle servernya sendiri. Pilihan custom digunakan untuk memilih sendiri komponen-komponen yang akan diinstal. Pilihan ini sebaiknya tidak dipilih jika belum mengenal benar konfigurasi dasar dari oracle. Sebaiknya pilih typical. Tekan NEXT.



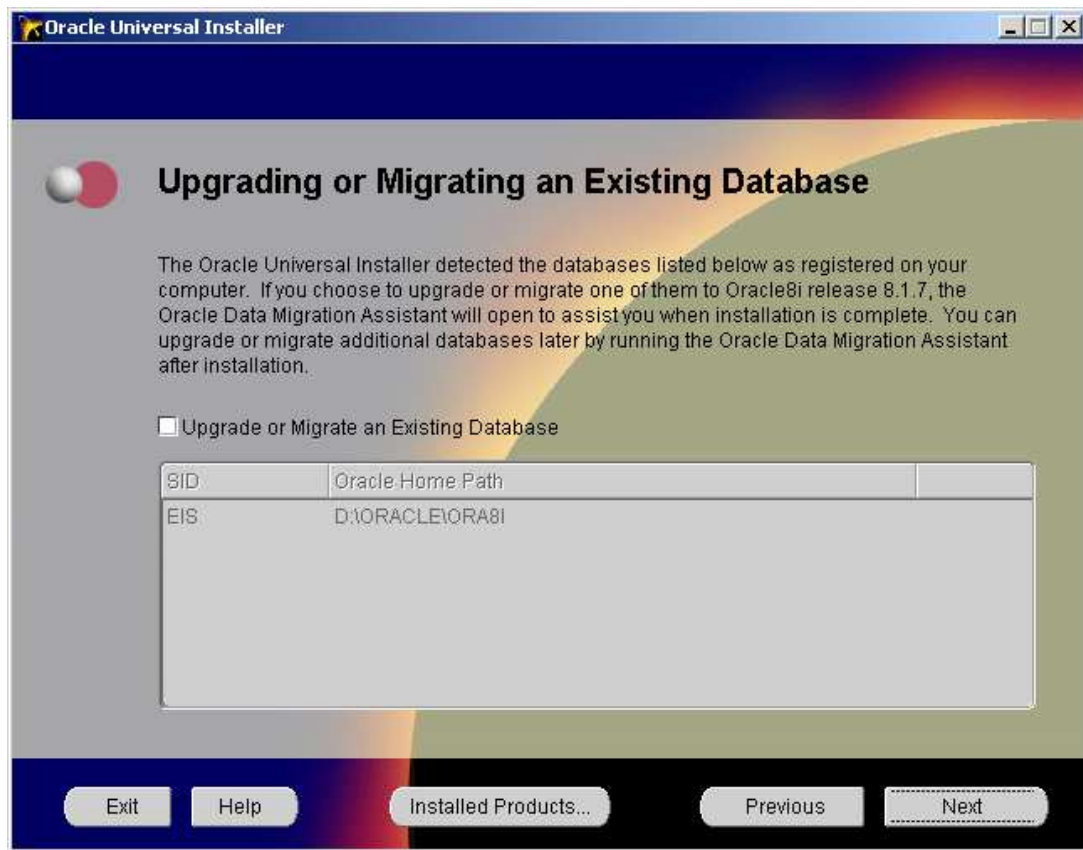
Gambar Pilihan Instalasi Oracle Server Enterprise

Pada tampilan berikutnya, oracle akan melakukan pengujian pada komponen/produk yang akan diinstal dan produk/komponen yang sudah ada di dalam komputer. Ini akan memerlukan waktu yang cukup lama. Pada tahap ini sering terjadi masalah. Oracle 8i adalah aplikasi yang dibangun menggunakan Java. Java adalah salah satu jenis bahasa pemrograman yang sangat haus akan sumber daya komputer. Jika ada sumber daya komputer yang tidak sesuai dengan kebutuhan Java oracle ini, maka akan terjadi kesalahan pada saat program dijalankan. Pada saat membaca produk ini, jika komputer tidak memenuhi kriteria oracle, program bisa macet atau terjadi kesalahan.



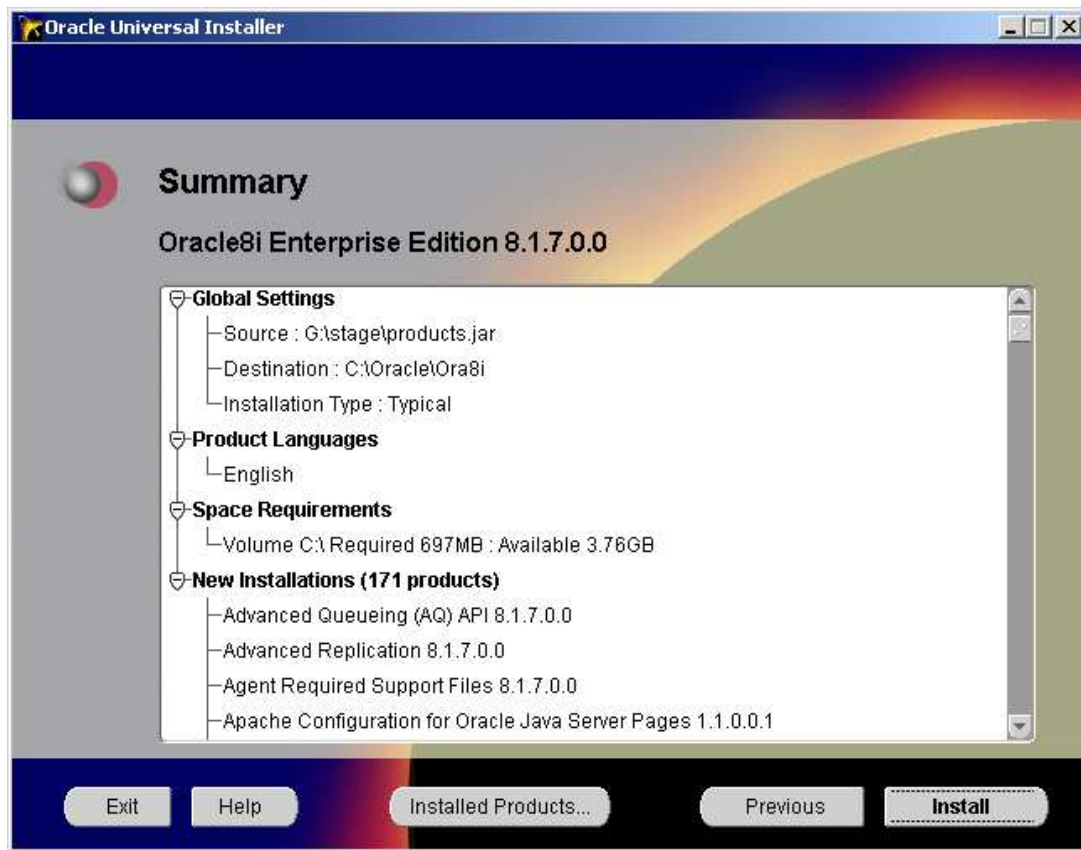
Gambar Oracle melakukan pembacaan produk yang akan diinstall/telah terinstall

Jika pada saat instalasi, di dalam server terdapat service database Oracle, maka service ini akan terdeteksi dan Oracle menyarankan untuk melakukan "Update" (jika versi database-nya sama) atau "Migrate" (jika versi database-nya tidak sama). Silakan pilih untuk mengambil service yang ada atau mengabaikan.



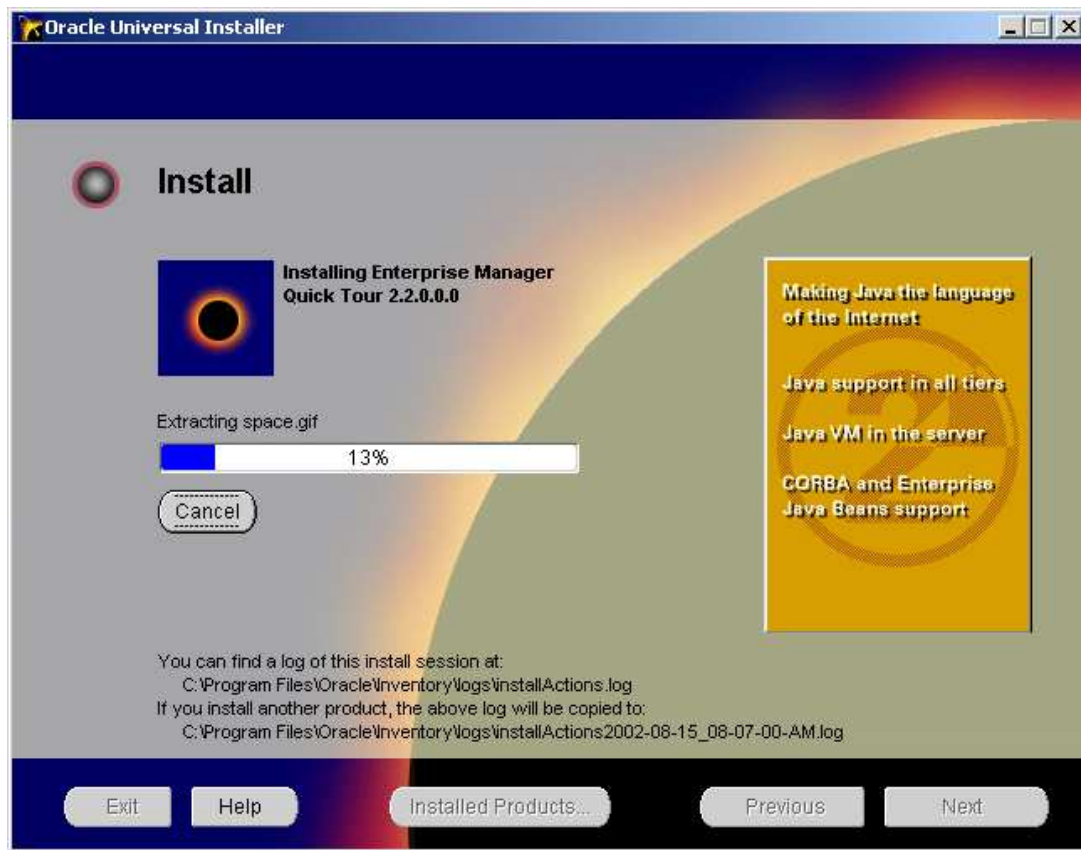
Gambar Pilihan untuk melakukan Update atau Migrate saat instalasi

Tampilan berikutnya hanya informasi mengenai instalasi yang akan dilakukan (komponen-komponen yang akan diinstal).



Gambar Tampilan Informasi mengenai produk yang akan diinstall

Tekan “Install” untuk memulai instalasi, dan berikutnya akan ditampilkan informasi mengenai instalasi yang sedang dilakukan. Instalasi ini memakan waktu cukup lama. Bisa antara ½ jam sampai 2 jam, tergantung dari kecepatan komputer dan jenis instalasi yang digunakan.

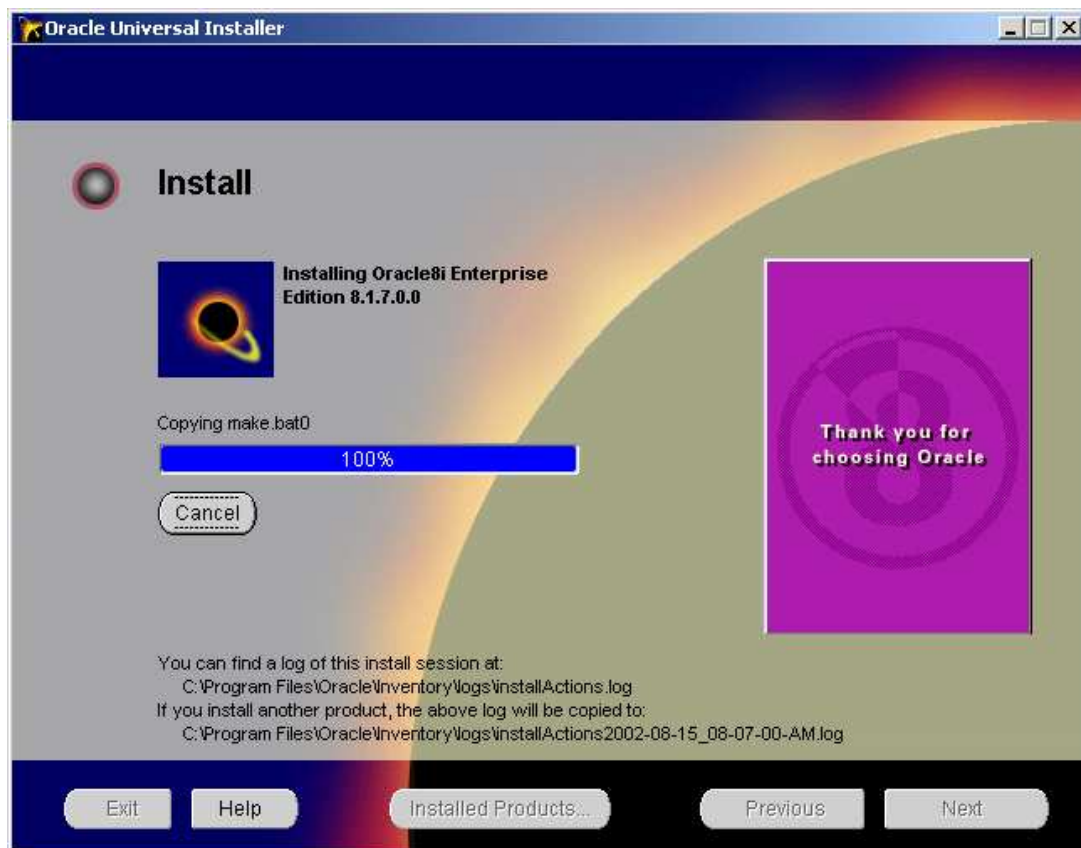


Gambar Tampilan saat instalasi sedang berlangsung

Sekali lagi yang perlu diperhatikan, proses instalasi ini sangat memakan resource dari komputer yang digunakan. Jika ada sedikit kriteria dari komputer yang digunakan tidak sesuai dengan yang diperlukan (ini barangkali tidak akan tampak pada program aplikasi lainnya), proses instalasi akan mengalami cukup banyak gangguan (bisa macet, atau kegagalan saat melakukan penyalinan file ke disk). Apabila ini terjadi, coba ulangi instalasi kembali. Dan jika tetap gagal, coba lakukan penghapusan isi disk yang tidak digunakan. Lakukan scandisk/chkdsk dan disk defragmentasi untuk menghilangkan gangguan yang disebabkan oleh disk. Lakukan uninstall pada program-program tertentu, terutama yang bersifat resident untuk menghindari konflik antar program.

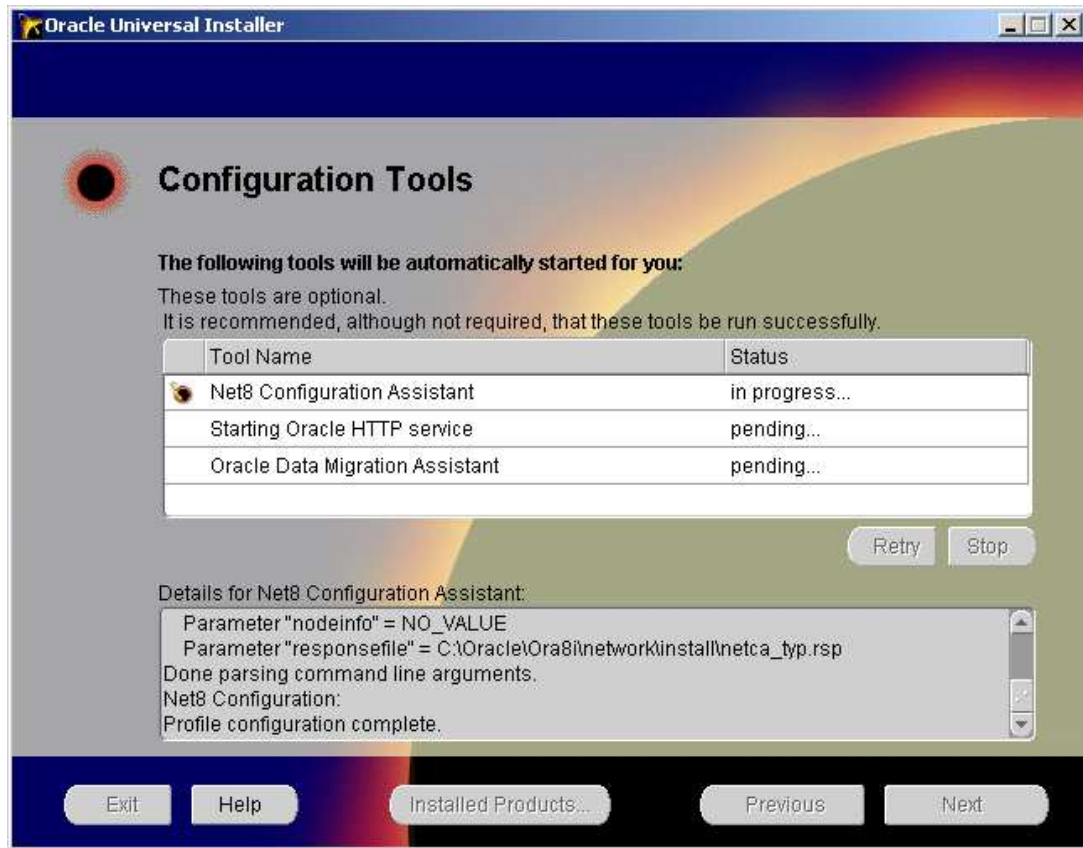
Jika tetap terjadi kemacetan/kesalahan, masalah bisa terjadi pada perangkat keras komputer yang kurang memenuhi syarat (terkadang permasalahan perangkat keras komputer tidak hanya berhubungan dengan prosesor, RAM, harddisk, tetapi juga berhubungan dengan mutu/kinerja dari motherboard-nya) atau ada masalah pada software/instalasi operating system-nya (windows

NT-nya). Jika memungkinkan, coba install ulang windows NT-nya, dan akan lebih baik jika sebelumnya dilakukan partisi ulang harddisk-nya. Lakukan sekali lagi instalasi Oracle-nya. Jika tetap terjadi kesalahan, maka kemungkinan besar perangkat kerasnya tidak sesuai untuk oracle server.



Gambar Tampilan setelah instalasi selesai 100%

Akhir dari instalasi ditandai dengan munculnya tampilan yang menyatakan status dari program yang sedang dijalankan. Dimulai dengan setting konfigurasi sambungan dengan Oracle server, instalasi Apache web server, dan proses migrate/instalasi global database.



Gambar Tampilan Konfigurasi pasca instalasi Oracle Server

Jika semua berhasil, langkah terakhir adalah melakukan instalasi/konfigurasi database global. Database global ini adalah tempat seluruh database akan disimpan. Mulai dari program database, tabel-tabel, counter dan lainnya.



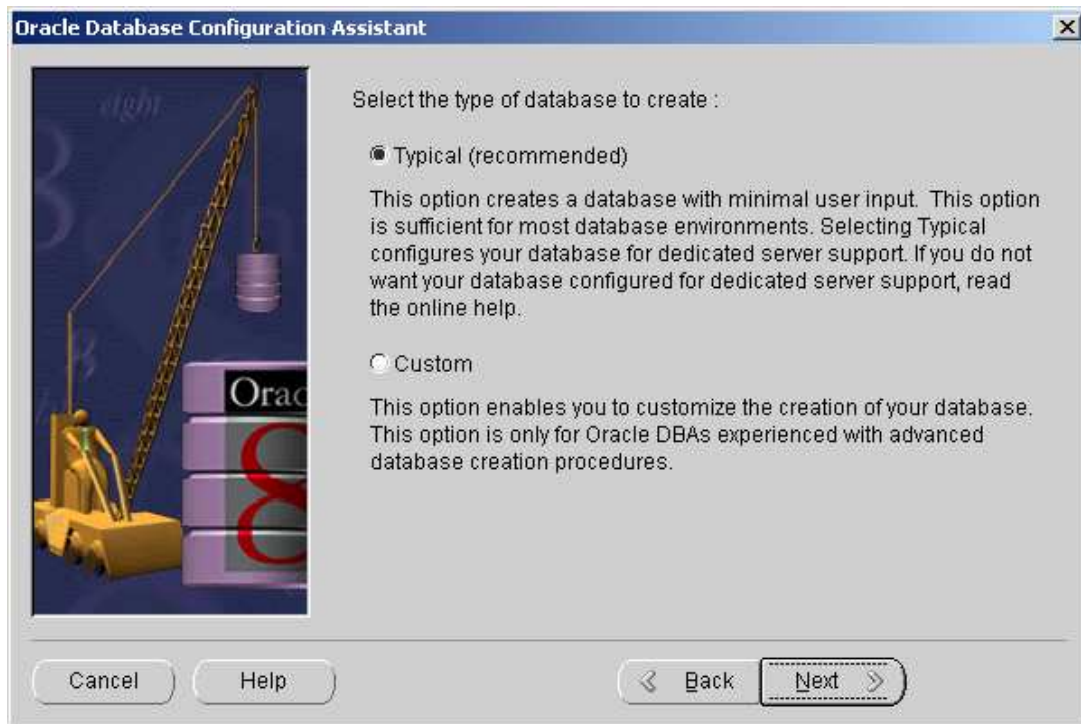
Gambar Setting/instalasi global database sedang dimulai

Jika ini merupakan instalasi global database yang pertama, tampilan berikutnya memerintahkan untuk membuat database.



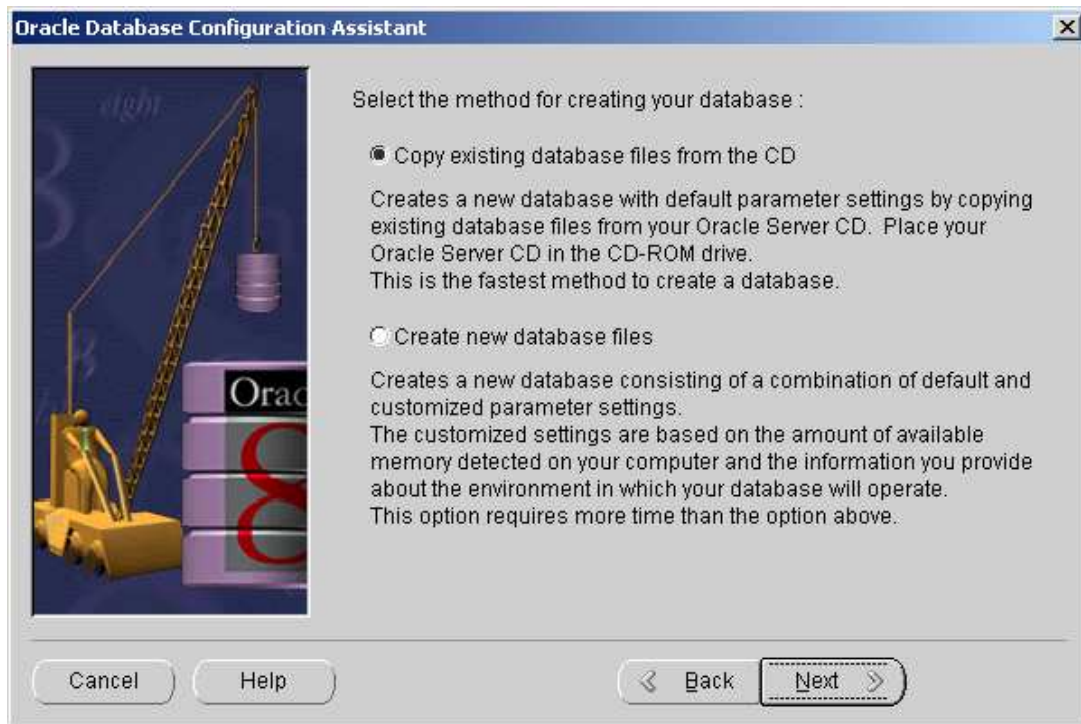
Gambar Pilihan perintah setting global database

Akan muncul pilihan untuk melakukan instalasi secara typical atau custom. Pilih typical, karena akan memilih konfigurasi standard database global. Pemilihan custom akan cukup menyulitkan karena cukup banyak setting yang harus dilakukan.



Gambar Pilihan jenis instalasi global database

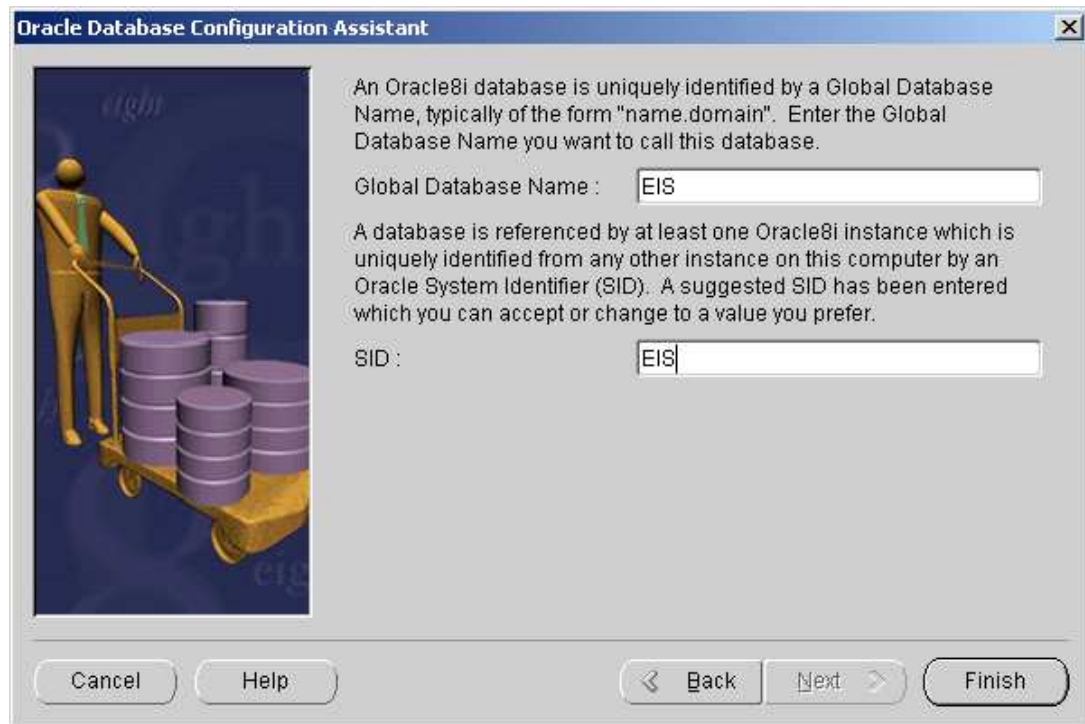
Cara pembuatan database Oracle bisa dilakukan dengan melakukan pengkopian dari contoh database yang sudah ada (dalam hal ini dari CD), atau benar-benar membuat baru. Sebaiknya melakukan pengkopian dari database yang sudah ada, karena akan lebih cepat (memerlukan waktu sekitar ½ jam). Sedangkan jika melakukan pembuatan database baru, akan memakan waktu sampai 2 jam.



Gambar Pilihan pembuatan database baru

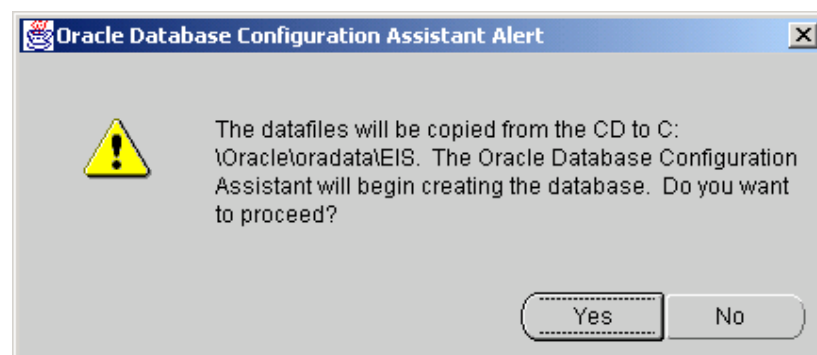
Setelah itu akan muncul tampilan setting yang meminta masukan nama dari Global Database, atau SID (service Identification). Pilih nama yang sesuai, pilih yang sederhana dan pendek. Yang perlu diperhatikan, nama ini harus diingat, karena selalu digunakan saat akan membuat koneksi/hubungan antara client ke server. Nama yang sering digunakan pada Oracle versi sebelumnya adalah ORCL. Dapat juga diganti dengan nama misalnya LATIHAN, EEPIS, ORACLE dan seterusnya.

Global Database Name adalah nama yang digunakan pada Oracle mulai 8i ke atas, sedangkan SID adalah nama yang digunakan pada Oracle sebelum 8i. Dengan asumsi oracle akan digunakan mulai versi 8i ke atas, maka pilih penamaan Global Database Name dan SID yang sama.

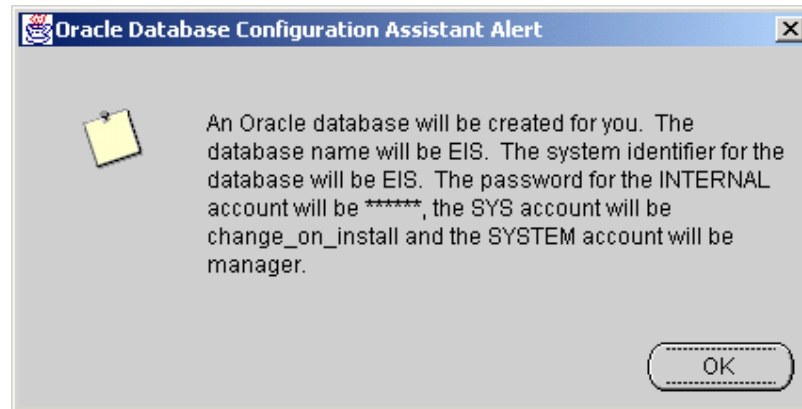


Gambar Pemberian nama database global

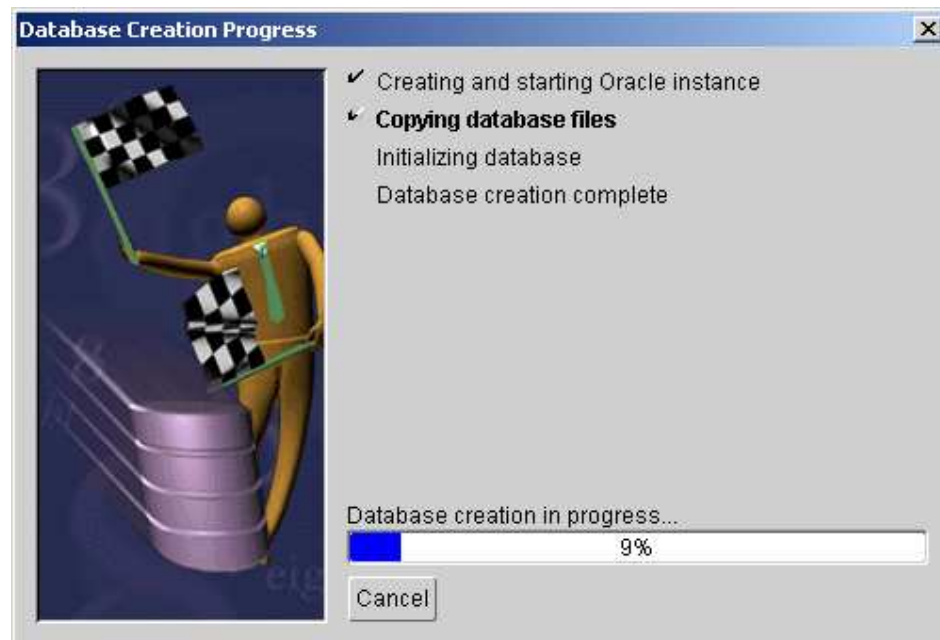
Tekan “Finish” untuk memulai pembuatan database global.



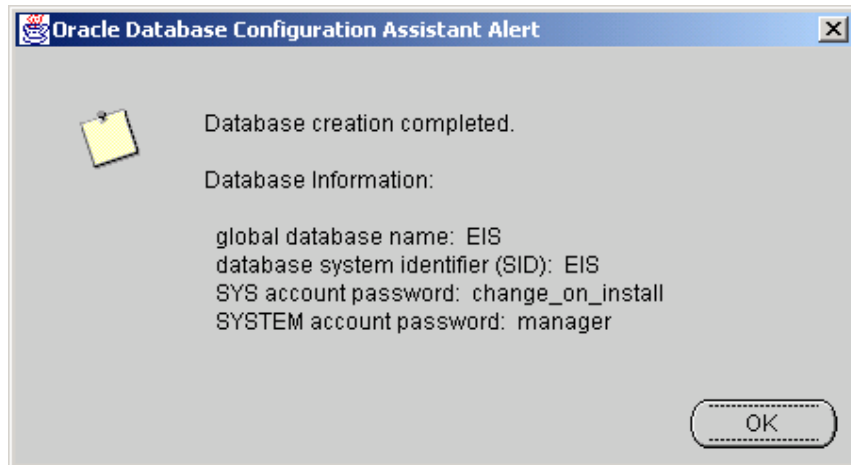
Gambar Pesan akan melakukan pengkopian database



Gambar Pesan setting User Default



Gambar Proses pengkopian sedang berjalan



Gambar Pesan Pembuatan database selesai dilakukan

Sampai tahap ini, seluruh proses instalasi server Oracle selesai dilakukan. Setting user default adalah,

- user **SCOTT** dengan password **TIGER**
- user **SYSTEM** dengan password **MANAGER**
- user **SYS** dengan password **CHANGE_ON_INSTALL**
- user **INTERNAL** dengan password **ORACLE**.

Catatan, nama-nama di dalam oracle, jika tidak dinyatakan dengan cara khusus, dianggap huruf besar semua. Karena itu, user name dan password di dalam oracle tidak pengaruh diketik dengan huruf besar atau kecil (incase sensitive).

Client

Instalasi Oracle Client memerlukan instalasi yang tidak lebih gampang dari pada instalasi oracle server. Namun memerlukan persyaratan perangkat yang lebih sederhana. Oracle client dapat diinstal pada windows 95, 98, Me, NT, 2000 atau XP.

Berikut ini contoh perangkat yang sesuai:

- Pentium 233 MMX, sebaiknya PII.
- RAM 32 MB, sebaiknya 128 MB.
- Harddisk 1 GB, sebaiknya 4,3 GB.
- Hardware komunikasi (modem, Ethernet atau lainnya).

Langkah instalasinya adalah sebagai berikut:

Gunakan CD installer Oracle Enterprise, atau Oracle Client. Di dalam Oracle server juga terdapat oracle client, dan jika instalasi dilakukan pada windows 95, 98 atau Me, maka oracle hanya akan melakukan instalasi oracle client saat CD dimasukkan ke CD Drive, program instalasi akan langsung dijalankan. Akan muncul setting yang meliputi memasukkan nama direktori tujuan, direktori asal program setup, dan nama dari Oracle Home. Ubah sesuai dengan keinginan. Dan tekan NEXT.

Berikutnya akan muncul jenis instalasi, Administrator, Programming, Application atau Custom. Administrator digunakan jika client ingin dapat akses penuh ke server (sebagai Administrator). Programming digunakan jika client hanya digunakan untuk keperluan pembuatan program aplikasi saja. Pilihan Application digunakan jika client hanya digunakan untuk menjalankan program aplikasi saja. Sedangkan Custom digunakan sesuai dengan keinginan dari User.

Pilih saja Administrator. Tekan NEXT.

Tampilan berikutnya hanya informasi mengenai instalasi yang akan dilakukan. Tekan NEXT untuk memulai instalasi dan berikutnya akan dilakukan instalasi yang memerlukan waktu yang cukup panjang (antara 1 jam sampai lebih dari 2 jam tergantung dari kecepatan komputer dan RAM yang digunakan). Tunggu sampai seluruh instalasi selesai.

Untuk kasus tertentu, ditengah-tengah instalasi, proses dapat berhenti dalam waktu yang relatif lama, tetapi kemudian berjalan lagi. Bisa jadi macet terus dan tidak jalan lagi. Atau terkadang saat instalasi terjadi kesalahan yang berulang-ulang. Jika ini terjadi, coba ulangi instalasi dari awal dan perhatikan, apakah keadaan yang sama muncul.

Jika tetap terjadi kemacetan/kesalahan, masalah bisa terjadi pada perangkat keras komputer yang kurang memenuhi syarat (terkadang permasalahan perangkat keras komputer tidak hanya berhubungan dengan prosesor, RAM, harddisk, tetapi juga berhubungan dengan mutu/kinerja dari motherboard-nya) atau ada masalah pada software/instalasi operasinya (windows-nya). Jika memungkinkan, coba install ulang windows-nya, dan akan lebih baik jika sebelumnya dilakukan partisi ulang harddisk-nya atau dilakukan defragmentasi. Lakukan sekali lagi instalasi Oracle-nya. Jika tetap terjadi kesalahan, maka kemungkinan besar perangkat kerasnya tidak sesuai untuk oracle client.

Di akhir instalasi, akan dilakukan setting koneksi antara client dengan server.

Personal Oracle

Personal Oracle merupakan salah satu alternatif lain untuk membuat sistem database tanpa menggunakan Oracle Server. Oracle server memerlukan beberapa persyaratan yang tidak mudah (meskipun saat ini komputer-komputer dengan persyaratan untuk instalasi Oracle Server sudah tidak asing lagi). Selain itu, dengan beberapa alasan tertentu, hanya diperlukan database Oracle untuk keperluan sendiri, dan bersifat lokal. Untuk itu disediakan konfigurasi Oracle Personal, yaitu Oracle dengan fasilitas database yang hanya bisa digunakan untuk satu komputer.

Database Connection

Untuk dapat menggunakan database oracle pada aplikasi development lain, seperti Visual Basic, Cold Fusion, ASP dll. Perlu untuk terlebih dahulu dilakukan konfigurasi koneksi database, berikut adalah langkah-langkah untuk melakukan konfigurasi database oracle :

Langkah 1 : Start (klik) Icon Net Manager

Klik pada ikon Net Manager, yang akan memunculkan window konfigurasi koneksi, Sebelum membuat koneksi isikan dulu Listener Addresses dengan cara ;

- Add Address (klik Button +)
- Isikan protocol, hostname dan port yang digunakan. Nilai default port = 1521 dengan protocol TCP/IP.

Kemudian **Konfigurasi Service name** :

- Klik + pada **Local**
- Pilih Local → **Services Naming**
- Add Service Name.
- Beri **Nama Net Service Name**
- Pilih Protocol, Pilih **TCP/IP Protocol**
- Isikan **Nama Hostname**.

Hostname adalah nama computer server, dimana dikomputer itu telah terdapat database oracle, misalkan nama computer server ELISTA.

Hostname bisa diisi dengan nomor IP.

- Isikan nama **service name database**.

Service name database, adalah nama koneksi yang dibuka oleh database server untuk koneksi dari luar.

Nama ini dibuat saat instalasi awal oracle database server.

- Lakukan **test** pada koneksi yang dibuat. Klik **Finish**.
- Jika proses test koneksi gagal, berarti ada yang salah dengan konfigurasi yang kita buat. Ulangi lagi konfigurasi.

Nama service name yang kita buat tadi, akan menjadi penghubung aplikasi yang kita jalankan dengan database oracle.

Langkah 2 : Menggunakan Oracle Manager Console

Untuk dapat membuat tabel, user dan segala administrasi pada database oracle, gunakan oracle manager console dengan menggunakan username system dengan password Manager. Berikut adalah langkah-langkah penggunaan Oracle Manager Console (OMC).

- Di window OMC anda akan menemukan *tree list Network* → **Database**.
Klik kanan pada database, pilih add **database to tree**.
- Muncul Window baru, pilihlah database pada window tersebut dengan memilih pilihan auto. Pilihan ini akan menambahkan koneksi ke database yang telah kita buat di langkah 1 tadi.
- Klik Ok.
- Setelah koneksi database ada, klik tanda + pada *tree list database*, selanjutnya anda akan diminta untuk memasukkan username dan password.
- Setelah terkoneksi akan muncul *tree list administrasi database oracle*.

Langkah 3 : Membuat User Baru

Pilih *tree list* untuk menkonfigurasi user baru, :

- Klik tanda + pada **Security**.
- Klik kanan pada **user** → Pilih **create**
- Akan Muncul window baru, yaitu window untuk pilihan user baru.
- Masukkan nama user baru beserta passwordnya
- Pilih windows tab **Role** , tambahkan role database ke user baru.
- Anda juga akan dapat menambah **privileges, quota, proxy** dan **group** pada user baru, dengan memilih window tab yang ada.

Langkah 4 : Membuat Tablespaces

Tablespaces adalah ruang penyimpanan untuk penggunaan database. Dengan kebebasan untuk mengatur kapasitas dan lokasi penyimpanan database, maka kan dengan mudah kita mengadministrasi database. Langkah-langkah pembuatan tablespaces :

- Pilih *tree list Storage* → Klik kanan pada *tree list tablespaces* → Pilih **Create**.
- Pada Window baru, isikan nama tablespaces, lokasi direktori penyimpanan, dan kapasitasnya.
- Klik Create.

Langkah 5 : Membuat Table

Lakukan koneksi ulang pada database dengan menggunakan username yang baru dibuat. Kemudian ikuti langkah berikut :

- Pilih *tree list* **Schema** → Klik kanan *Tree list* **Table** → Pilih **create**.
- Pada window baru, isikan nama table, pilih schema, dan tablespace.
Pilih schema user baru dan tablespaces yang telah kita buat pada langkah sebelumnya.
- Isikan nama kolom-kolom yang akan kita buat beserta atributnya yaitu tipe data dan jumlahnya, nilai default dan atribut NULL.
- Klik Create.

Catatan .

Jika kita ingin memanipulasi table dengan menggunakan OMC, seperti menambah kolom, edit kolom atau menambah data baru, dapat dilakukan dengan menklik kanan pada nama table tersebut.

SQL*Plus

SQL Plus merupakan salah satu aplikasi yang bisa kita gunakan untuk melakukan administrasi database. Namun ada syarat yang musti dimiliki yaitu kemampuan untuk menggunakan query SQL. Berikut penulis jelaskan beberapa penggunaan query SQL.

SQL (*Structured Query Language*) merupakan bahasa *query* yang digunakan untuk mengakses *database* relasional. SQL sekarang sudah menjadi bahasa database standard dan hampir semua sistem *database* memahaminya. SQL terdiri dari berbagai jenis statemen. Semuanya didesain agar dia memungkinkan untuk dapat secara interaktif berhubungan dengan *database*.

Penggunaan SQL pada DBMS (*Database Management System*) sudah cukup luas. SQL dapat dipakai oleh berbagai kalangan, misalnya DBA (*Database Administrator*), *programmer* ataupun pengguna. Hal ini disebabkan karena :

- SQL sebagai bahasa administrasi *database*
Dalam hal ini SQL dipakai oleh DBA untuk menciptakan serta mengendalikan pengaksesan *database*.
- SQL sebagai bahasa *query* interaktif
Pengguna dapat memberikan perintah-perintah untuk mengakses *database* yang sesuai dengan kebutuhannya.
- SQL sebagai bahasa pemrograman *database*
Pemrogram dapat menggunakan perintah-perintah SQL dalam program aplikasi yang dibuat.
- SQL sebagai bahasa klien/server
SQL juga dipakai sebagai untuk mengimplementasikan sistem klien/server. Sebuah klien dapat menjalankan suatu aplikasi yang mengakses *database*. Dalam hal ini sistem operasi antara server dan klien bisa berbeda.

Di samping hal tersebut diatas SQL juga diterapkan pada internet atau intranet untuk mengakses *database* melalui halaman-halaman web untuk mendukung konsep web dinamis.

Kelompok Pernyataan SQL

Pernyataan SQL dapat dikelompokkan menjadi 5 kelompok DDL, DML, DCL, pengendali transaksi dan pengendali programatik.

- **DDL (*Data Definition Language*)**

DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut *database*, table, atribut (kolom), batasan-batasan terhadap suatu atribut serta hubungan antar table. Yang termasuk kelompok DDL ini adalah

CREATE	untuk menciptakan table ataupun indeks
ALTER	untuk mengubah struktur table
DROP	untuk menghapus table ataupun indeks

- **DML (*Data Manipulation Language*)**

Adalah kelompok perintah yang berfungsi untuk memanipulasi data, misalnya untuk pengambilan, penyisipan pengubahan dan penghapusan data. Yang termasuk DML adalah

SELECT	memilih data
INSERT	menambah data
DELETE	menghapus data
UPDATE	mengubah data

- **DCL (*Data Control Language*)**

Berisi perintah-perintah untuk mengendalikan pengaksesan data. Yang termasuk DCL adalah :

GRANT	memberikan kendali pada pengaksesan data.
REVOKE	mencabut kemampuan pengaksesan data
LOCK TABLE	mengunci table

- **Pengendali transaksi**

Adalah perintah-perintah yang berfungsi untuk mengendalikan pengeksekusian transaksi. Yang termasuk kelompok ini adalah :

COMMIT	menyetujui rangkaian perintah yang berhubungan erat yang telah berhasil dilakukan
ROLLBACK	membatalkan transaksi yang dilakukan karena adanya kesalahan atau kegagalan pada salah satu rangkaian perintah.

Menjalankan Aplikasi SQL Plus

Untuk menjalankan ikon SQL Plus berikut langkah-langkahnya :

- Jalankan SQL Plus dengan klik 2 kali pada ikon SQL Plus
- Masukkan Username,password dan hostname
- Tunggu sampai keluar prompt SQL>.
- Ketikkan perintah SQL anda.
- Setiap selesai menjalankan perintah SQL akhiri dengan perintah
Sql> Commit

Penggunaan Perintah SQL

Berikut ini akan dibahas lebih mendetail mengenai perintah SQL yang telah dijelaskan sebelumnya.

DDL

DDL kependekan dari Data Definition Language, yaitu perintah yang digunakan untuk menyatakan/mendefinisikan sesuatu. Meliputi perintah untuk membuat, menghapus maupun mengubah. Berikut ini daftar perintah yang termasuk dalam DDL:

- Membuat, mengubah dan menghapus obyek: [CREATE](#), [ALTER](#), [DROP](#)
- Mengubah nama skema: [RENAME](#)
- Menghapus semua data dalam obyek skema tanpa menghapus struktur obyek skema: [TRUNCATE](#)
- Melakukan analisa statistik dari suatu obyek: [ANALYZE](#)
- Memberikan atau menghapus role dan privilege: [GRANT](#), [REVOKE](#)
- Mengubah nilai audit on atau off: [AUDIT](#), [NOAUDIT](#)
- Menambahkan komentar pada kamus database: [COMMENT](#)

Perintah DDL ini berlaku untuk semua jenis object database, seperti tablespace, table, view, procedure, function dan sebagainya. Namun masing-masing memiliki sedikit perbedaan pada saat menyatakan beberapa perintah.

Create

Create digunakan untuk membuat suatu object database. Format untuk membuat tiap object berbeda. Beberapa contoh format untuk membuat tablespace, tabel, view, fungsi, prosedur, sekuen,

Membuat Tablespace

```
CREATE          TABLESPACE          "NAMA_TABLESPACE"          LOGGING          DATAFILE  
'C:\ORACLE\ORADATA\EIS\NAMA_FILE.ora' SIZE 200M REUSE AUTOEXTEND ON NEXT  
10240K MAXSIZE 32767M DEFAULT STORAGE ( INITIAL 64K NEXT 64K MINEXTENTS 1  
MAXEXTENTS 2147483645 PCTINCREASE 50 );
```

Membuat User dan hak user

```
CREATE USER "NAMA_USER" PROFILE "DEFAULT" IDENTIFIED BY "PASSWORD" DEFAULT  
TABLESPACE "NAMA_TABLESPACE" TEMPORARY TABLESPACE "NAMA_TABLESPACE" ACCOUNT  
UNLOCK;
```

```
GRANT UNLIMITED TABLESPACE TO "NAMA_USER";  
GRANT "CONNECT" TO "NAMA_USER";  
GRANT "RESOURCE" TO "NAMA_USER";
```

Membuat Tabel

```
CREATE TABLE "NAMA_TABEL" (  
    "NOMOR" NUMBER(2) NOT NULL,  
    "AGAMA" VARCHAR2(20),  
    CONSTRAINT "Agama_Nomor_1" PRIMARY KEY("NOMOR") USING INDEX);
```

Membuat Index

```
CREATE UNIQUE INDEX "NAMA_INDEX" ON "ABSENSI_MAHASISWA" ("BULAN");
```

Membuat View

```
CREATE OR REPLACE VIEW "NAMA_VIEW" AS Select * from emp where sal>1000;
```

Membuat Sekuen

```
CREATE SEQUENCE "NAMA_SEKUEN" INCREMENT BY 1 START WITH 1 MAXVALUE 1.0E27  
MINVALUE 1 NOCYCLE CACHE 20 NOORDER
```


Membuat Fungsi

```
CREATE OR REPLACE FUNCTION "NAMA_FUNGSI" (nrp varchar)
    return varchar
is
tahun number(3,0);
begin
    tahun:=to_number(substr(nrp,3,2));
    if(tahun<80) then tahun:=100+tahun; end if;
    tahun:=200-tahun;
    return substr(nrp,1,2)||trim(to_char(tahun,'000'))||substr(nrp,5);
end;
```

Membuat Prosedur

```
CREATE OR REPLACE PROCEDURE "NAMA_PROSEDUR" (noid_
    varchar,tanggal_ date,jam_ varchar) is
pegawai_ number(8);
n integer;
t date;

cursor cp is select nomor from pegawai where noid=noid_;
cursor ca is select pegawai,masuk,pulang,libur from absensikaryawan
    where pegawai=pegawai_ and tanggal=tanggal_
    for update of masuk, pulang, terlambat1, terlambat2, pulangawal,
tidakabsen, tidakmasuk, libur, lembur;
cursor ct is select tanggal from tanggal;

m char(8);
p char(8);
am char(8);
ap char(8);
pgw number(8);
aL number(1);
pm char(8);
t1 char(8);
t2 char(8);
xt1 number(1);
xt2 number(1);
xa number(1);
xp number(1);
```

```

nm number(2);
nn varchar(2);
nnn varchar(10);
L number(4,2);
j char(8);
H number(1);
Libur number(1);

begin
    open cp;
    fetch cp into pegawai_;
    close cp;
    if pegawai_ is not null then
        open ca;
        fetch ca into pgw,am,ap,aL;
        -- pegawai ada di database ?
        if pgw is null then
            open ct;
            fetch ct into t;
            close ct;
            -- data tanggal sesuai dengan bulan ini ?
            if t is null or to_char(t,'mm-yyyy')<>to_char(tanggal_,'mm-yyyy') then
                --t:=trunc(tanggal_,'month');
                delete from tanggal;
                nm:=to_number(to_char(last_day(tanggal_), 'dd'));
                n:=1;
                -- data tanggal untuk bulan ini
                while n<=nm loop
                    nn:=to_char(n);
                    nnn:=to_char(tanggal_,'mm-yyyy');
                    t:=to_date(nn||'-'||nnn,'dd-mm-yyyy');
                    select count(libur) into Libur from harilibur where tanggal=t;
                    if Libur=0 then
                        H:=to_char(t,'d');
                        select libur into Libur from harikerja where hari=H;
                    end if;
                    insert into tanggal values(t,Libur);
                    n:=n+1;
                end loop;
            end if;
            -- data awal setiap pegawai setiap bulan

```

```

close ca;
insert                                into                                absensikaryawan
(pegawai,tanggal,tidakabsen,tidakmasuk,libur,lembur)
      select  pegawai_,tanggal,0,decode(libur,1,0,1),libur,0  from  tanggal
where tanggal not in
      (select tanggal from absensikaryawan where pegawai=pegawai_);
commit;
open ca;
fetch ca into pgw,am,ap,aL;
end if;
select masuk,pulang,pulang_max,terlambat_1,terlambat_2 into m,p,pm,t1,t2
from setting;
j:=jam_;
--if j<m then j:=m; end if;
--if j>pm then j:=pm; end if;
if j<am or am is null then am:=j; end if;
if j>ap or ap is null then ap:=j; end if;
-- jika hari libur, semua status tidak ada
if aL='1' then
  L:=0.0; -- tidak ada lembur
  xa:=0; -- tidak ada yang absen hanya sekali
  xt1:=0; -- tidak ada yang terlambat I
  xt2:=0; -- tidak ada yang terlambat II
  xp:=0; -- tidak ada yang pulang terlambat
else
  -- sehari absen satu kali
  if am=ap then
    L:=0.0;
    xa:=1;
  else
    -- nilai awal lembur, hari libur=0.0 jam, jumat=1.5 jam,
    -- hari lain 1.0 jam
    if to_char(tanggal_,'d')='6' then
      L:=1.50;
    else
      L:=1.00;
    end if;
    xa:=0;
  end if;
  -- terlambat I
  if am>=t1 and am<=t2 then xt1:=1; L:=L-0.5; else xt1:=0; end if;

```

```

-- terlambat II
if am>t2 then xt2:=1; L:=L-1.0; else xt2:=0; end if;
-- pulang awal
if ap<p then xp:=1; L:=L-0.5; else xp:=0; end if;
if L<0 then L:=0; end if;
end if;
update                                absensikaryawan                set
masuk=am,pulang=ap,terlambat1=xt1,terlambat2=xt2,
    pulangawal=xp,tidakmasuk=0,tidakabsen=xa,lembur=L
    where current of ca;
close ca;
else
    raise_application_error(-20000,'Pegawai    dengan    NoID:'''||noid_||'    tidak
dikenal ...');
end if;
commit;
end;

```

Drop

Digunakan untuk menghapus suatu object database tertentu. Karena hanya untuk menghapus, maka perintahnya hampir sama semua. Format umumnya,

```
DROP OBJECT NAMA_OBJECT;
```

Contoh,

```

Drop tablespace NAMA_TABLESPACE;
Drop user NAMA_USER;
Drop table NAMA_TABLE;
Drop index NAMA_INDEX;
Drop view NAMA_VIEW;
Drop sequence NAMA_SEQUENCE;
Drop function NAMA_FUNCTION;
Drop procedure NAMA_PROCEDURE;

```

Alter

Alter merupakan perintah yang digunakan untuk mengubah suatu object database. Yang perlu diperhatikan, Oracle tidak banyak memberikan kemudahan untuk mengubah suatu object. Cukup banyak batasan yang ada. Misalkan tidak dapat mengganti nama suatu object. Tidak dapat mengubah suatu jenis data dalam suatu tabel yang sudah ada isinya. Dan masih banyak lagi.

Mengubah Nama Tablespace

```
ALTER DATABASE  
    RENAME FILE 'C:\ORACLE\ORADATA\EIS\NAMA1.ORA' TO  
    'C:\ORACLE\ORADATA\EIS\NAMA_BARU.DBF'
```

Mengubah Nama Direktori Tablespace

```
ALTER DATABASE  
    RENAME FILE 'C:\ORACLE\ORADATA\EIS\NAMA1.ORA' TO  
    'C:\DIR_BARU\NAMA1.ORA'
```

Mengubah Ukuran Tablespace

```
ALTER DATABASE  
    DATAFILE 'C:\ORACLE\ORADATA\EIS\NAMA1.ORA' RESIZE 20M
```

Mengubah Password User

```
ALTER USER "NAMA" IDENTIFIED BY "PASSWORD_BARU"
```

Mengubah Default Tablespace untuk suatu User

```
ALTER USER "NAMA" DEFAULT  
    TABLESPACE "COBA" TEMPORARY  
    TABLESPACE "COBA"
```

Mematikan User

```
ALTER USER "NAMA" ACCOUNT LOCK
```

Membuang Grant suatu User

```
REVOKE "CONNECT"  
    FROM "NAMA";  
ALTER USER "NAMA" DEFAULT ROLE ALL
```

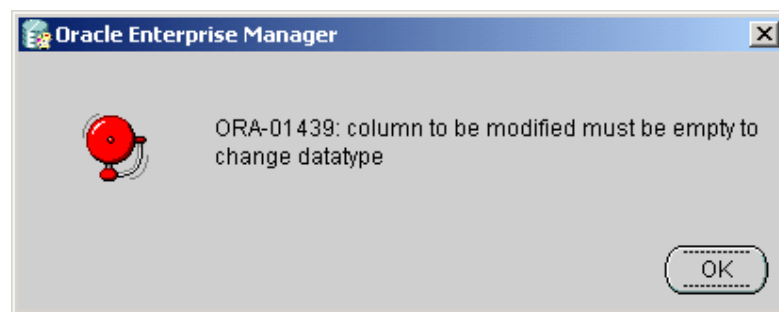
Menambah Grant untuk suatu User

```
GRANT "DBA" TO "NAMA";  
ALTER USER "NAMA" DEFAULT ROLE ALL
```

Mengubah jenis data atau ukuran data pada suatu kolom

```
ALTER TABLE "EIS"."AGAMA"  
MODIFY ("NOMOR" CHAR(10))
```

Seperti yang telah disebutkan sebelumnya, Oracle tidak banyak memberikan keleluasaan untuk mengubah suatu database, contohnya kolom dari suatu tabel. Jika pengubahan tersebut di atas dilakukan pada tabel yang sudah ada isinya, maka akan muncul pesan kesalahan.



Gambar Pesan kesalahan karena mengubah kolom dari suatu tabel yang tidak kosong

Khusus untuk mengubah ukuran data, jika ukurannya lebih besar dari yang sudah ada, maka perubahan tersebut dapat diterima. Namun jika perubahannya lebih kecil, perubahan akan diterima jika tabel yang bersangkutan dalam keadaan kosong.

Mengubah Kolom Tabel menjadi boleh tidak diisi

```
ALTER TABLE "EIS"."AGAMA"  
MODIFY ("NOMOR" NULL)
```

Menambahkan kolom baru

```
ALTER TABLE "EIS"."AGAMA"  
ADD ("KOLOM" VARCHAR2(10) NOT NULL)
```

Menghapus kolom baru

```
ALTER TABLE "EIS"."AGAMA"
```

```
SET UNUSED ("KOLOM") CASCADE CONSTRAINTS
```

Menambahkan constraint

```
ALTER TABLE "EIS"."AGAMA"  
ADD (CONSTRAINT "AGAMA_UK31031673027967" UNIQUE ("AGAMA"))
```

Menghapus constraint

```
ALTER TABLE "EIS"."AGAMA"  
DROP CONSTRAINT "Agama_Nomor" CASCADE
```

Mengubah suatu sequence

```
ALTER SEQUENCE "EIS"."NBUKU" MINVALUE 10  
ALTER SEQUENCE "EIS"."NBUKU" MAXVALUE 1.0E2  
ALTER SEQUENCE "EIS"."NBUKU" INCREMENT BY 10
```

DML

DML kependekan dari Data Modification Language, yaitu perintah yang digunakan untuk melakukan manipulasi data. Meliputi menyisipkan data baru, mengubah data, menghapus data, membaca data. Yang termasuk dalam DML adalah:

- Mengambil data dari satu atau lebih tabel atau view: [SELECT](#)
- Menambahkan baris baru ke dalam tabel atau view: [INSERT](#)
- Mengubah isi data dari suatu baris pada tabel atau view: [UPDATE](#)
- Menghapus suatu baris dari tabel atau view: [DELETE](#)
- Melihat rencana pelaksanaan perintah SQL: [EXPLAIN PLAN](#)
- Mengunci suatu tabel atau view, membatasi sementara akses dari user lain: [LOCK TABLE](#)

Select

Perintah **SELECT** digunakan untuk membaca atau mengambil suatu nilai dari suatu tabel atau suatu view. Hasil dari suatu perintah select bisa berupa baris-baris data dari tabel atau view, bisa juga merupakan kelompok dari hasil (grup). Format umum dari perintah select adalah:

SELECT KOLOM,FUNGSI(KOLOM),SEKUEN,... FROM NAMATABEL (atau NAMAVIEW) [WHERE KRITERIA] [ORDER BY KOLOM [DESC]] [GROUP BY KOLOM]

dimana,

KOLOM, FUNGSI(KOLOM),SEKUEN,... adalah daftar dari kolom data atau hasil fungsi atau dari sekuen yang akan ditampilkan. Jika ingin menampilkan semua yang ada, gunakan tanda asterik (*).

NAMATABEL atau NAMAVIEW adalah nama dari tabel atau view atau kombinasi dari beberapa tabel atau/dan view yang datanya akan diambil.

WHERE KRITERIA (bisa ditulis atau tidak) adalah daftar kriteria/filter, yang menyatakan data dengan jenis yang bagaimana akan ditampilkan.

ORDER BY KOLOM (bisa ditulis atau tidak) adalah menyatakan apakah data hasil yang ditampilkan akan diurutkan mememuni kriteria tertentu. DESC disertakan jika urutannya ke bawah (besar ke kecil).

Contoh, di dalam user **SCOTT/TIGER**, terdapat beberapa tabel yang salah satunya adalah tabel pegawai (emp). Untuk melihat isi dari tabel tersebut, digunakan perintah:

```
connect scott/tiger@eis.elista;  
select * from emp;
```

maka akan muncul tabel seperti berikut:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/1980	800	(null)	20
7499	ALLEN	SALESMAN	7698	20/02/1981	1600	300	30
7521	WARD	SALESMAN	7698	22/02/1981	1250	500	30

7566	JONES	MANAGER	7839	02/04/1981	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28/09/1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01/05/1981	2850	(null)	30
7782	CLARK	MANAGER	7839	09/06/1981	2450	(null)	10
7788	SCOTT	ANALYST	7566	19/04/1987	3000	(null)	20
7839	KING	PRESIDENT	(null)	17/11/1981	5000	(null)	10
7844	TURNER	SALESMAN	7698	08/09/1981	1500	0	30
7876	ADAMS	CLERK	7788	23/05/1987	1100	(null)	20
7900	JAMES	CLERK	7698	03/12/1981	950	(null)	30
7902	FORD	ANALYST	7566	03/12/1981	3000	(null)	20
7934	MILLER	CLERK	7782	23/01/1982	1300	(null)	10

Untuk menampilkan **ENAME** dan **JOB** saja, data diurutkan berdasarkan **ENAME** dan dipilih yang pekerjaannya sebagai **SALESMAN**, digunakan perintah:

```
select ename,job from emp where job='SALESMAN' order by ename;
```

maka akan muncul hasil:

ENAME	JOB
ALLEN	SALESMAN
MARTIN	SALESMAN
TURNER	SALESMAN
WARD	SALESMAN

Suatu lebih dari tabel dan/atau view yang ditampilkan menjadi satu tabel hasil disebut dengan joint. Jika lebih dari satu tabel/view digabungkan, maka jumlah baris yang dihasilkan merupakan perkalian dari jumlah baris masing-masing tabel/view. Untuk menghindari hal demikian, digunakan suatu relasi, yaitu baris-baris yang akan ditampilkan dari suatu tabel, nilai/data dari suatu kolom-kolomnya harus sesuai dengan nilai/data dari kolom tabel lainnya.

Misalkan, ingin mengetahui nama department dari setiap pegawai. Dari tabel sebelumnya dapat dilihat, setiap pegawai hanya memiliki nomor/kode department (**DEPTNO**), tetapi nama department-nya sendiri tidak diketahui. Nama department dapat dilihat dari tabel department (**DEPT**). Coba lihat isi dari tabel **DEPT**,

```
select * from dept;
```

Gabung antara tabel **EMP** dengan tabel **DEPT** tanpa menggunakan relasi,

```
select * from emp,dept;
```

Sekarang gunakan relasi. Dalam hal ini, relasinya adalah **DEPTNO**, dimana **DEPTNO** dari tabel **EMP** harus sama dengan **DEPTNO** dari tabel **DEPT**, dan hanya mengambil informasi **ENAME, JOP, DEPTNO,DNAME,LOC**,

```
select      ename, job, emp.deptno, dname, loc      from      emp, dept      where  
emp.deptno=dept.deptno;
```

atau dapat juga dituliskan dengan,

```
select ename, job, e.deptno, dname, loc from emp e, dept d where e.deptno=d.deptno
```

Pernyataan emp e berarti nama emp telah dialiaskan (namanya diganti) dengan e. Ini bertujuan untuk penyederhanaan penulisan. Dalam kasus ini mungkin tidak begitu terlihat, namun dalam kasus dimana jumlah tabel yang direlasikan cukup banyak dengan nama yang cukup panjang, dan banyak sekali kolom dari masing-masing tabel yang memiliki nama sama (jika ada kolom dari suatu tabel dengan nama yang sama, maka nama tabelnya harus disebutkan, seperti pada contoh di atas, deptno dimiliki oleh kedua tabel, sehingga harus disebutkan tabel yang mana).

Sekarang coba hitung, berapa total gaji (SAL) yang harus diberikan untuk setiap jenis pekerjaan (JOB). Misalkan, untuk satu jenis pekerjaan SALESMAN, total gaji yang dibayarkan adalah,

```
select sum(sal) "Total Gaji" from emp where job='SALESMAN'
```

Untuk menghitung seluruh jenis pekerjaan, digunakan pengelompokan (GROUP BY),

```
select job Pekerjaan, sum(sal) "Total Gaji" from emp group by job
```

PEKERJAAN merupakan nama pengganti (alias) dari nama **JOB**, dan **"Total Gaji"** merupakan alias dari **SUM(SAL)**. Tanda dalam petik ganda menyatakan penamaannya (dalam oracle) apa adanya seperti yang tertulis, termasuk huruf besar atau kecil, spasi dan sebagainya. Tanpa tanda petik ganda ini, oracle akan selalu menganggap nama menggunakan huruf besar dan dalam satu kata tunggal.

Ada satu perintah select yang bersifat khusus, yaitu untuk mengetahui tabel atau view yang dimiliki oleh user. Dengan menggunakan perintah,

```
SELECT * FROM TAB;
```

Jika ingin khusus mengetahui tabel saja, dapat digunakan perintah,

```
SELECT * FROM TAB WHERE TABTYPE='TABLE' ;
```

Insert

Perintah insert digunakan untuk memasukkan satu atau sejumlah baris ke dalam tabel. Data yang dimasukkan dapat terdiri dari semua kolom yang ada dalam tabel atau hanya beberapa saja, bahkan mungkin hanya satu kolom yang diisi. Format umum dari perintah insert adalah,

Jika susunan kolom tidak sesuai dengan susunan kolom pada tabel atau jumlah kolom lebih sedikit dibandingkan jumlah kolom dalam tabel, digunakan perintah,

```
INSERT INTO NAMA_TABLE (KOLOM1, KOLOM2, KOLOM3, ..., KOLOMn) VALUES (DATA1,  
DATA2, DATA3, ..., DATAn)
```

Dimana n adalah jumlah kolom yang akan diisi (tidak harus sama dengan jumlah kolom dalam tabel). KOLOM1 sampai KOLOMn adalah nama kolom yang akan diisi (disebutkan satu persatu). DATA1 sampai DATAn adalah data yang harus diisikan ke kolom yang sesuai urutan KOLOM1 sampai KOLOMn. Sedangkan untuk pengisian dengan syarat jumlah kolom dan urutan kolom sama persis dengan tabel, maka digunakan perintah,

```
INSERT INTO NAMA_TABLE VALUES (DATA1, DATA2, DATA3, ..., DATAn)
```

Dengan n adalah jumlah kolom sesuai dengan jumlah kolom dalam tabel. DATA1 sampai DATAn adalah data yang akan dimasukkan untuk setiap kolom dan berurutan sesuai dengan urutan kolom dalam tabel.

Jika pengisian ke dalam tabel ingin didapatkan dari hasil query perintah select, maka dapat digunakan perintah sebagai berikut,

```
INSERT INTO NAMA_TABLE SELECT KOLOM1, KOLOM2, ..., KOLOMn FROM  
NAMA_TABLE_SUMBER WHERE KRITERIA
```

Pada bentuk ini, segala jenis perintah select dapat digunakan, asalkan nama kolom yang dihasilkan sama dengan nama kolom dalam tabel yang akan diisi. Susunan atau jumlahnya tidak perlu sama, yang penting namanya harus sama. Dengan cara ini, dapat memasukkan lebih dari satu baris data sekaligus sesuai dengan hasil query select-nya.

Perlu diperhatikan, setiap data yang memiliki jenis CHAR, VARCHAR, DATE, dan lainnya yang tidak bernilai bilangan, harus ditutup dengan tanda petik tunggal (tanda kutip). Kecuali jika data tersebut berupa angka, meskipun kolomnya berjenis, misalkan VARCHAR, dapat langsung diisi dengan tanpa tanda petik. Oracle akan otomatis melakukan konversi ke jenis bilangan yang sesuai. Termasuk jika kolomnya berjenis NUMBER, diberikan data angka yang diapit tanda petik tunggal, Oracle langsung mengenalinya sebagai bilangan biasa (akan melakukan konversi seperlunya).

Misalkan terdapat kolom dengan nama ALAMAT dengan jenis VARCHAR, UMUR dengan jenis NUMBER, TELP dengan jenis VARCHAR, berikut ini adalah benar,

ALAMAT ← 'Ngagel baru DKA I/56 Surabaya'

ALAMAT ← 123456

ALAMAT ← '7654321'

UMUR ← 12

UMUR ← '32'

TELP ← '+62315947280'

TELP ← 5947280

Berikut ini adalah pengisian data yang salah,

ALAMAT ← Bratang Gede Surabaya

UMUR ← 12 th

UMUR ← 'sepuluh'

UMUR ← sepuluh

TELP ← (+62)(031)5947280

Khusus untuk jenis DATE, Oracle memiliki format system khusus, yaitu dalam bentuk DD-
MMM-YYYY. Contoh 12-JAN-2002, 31-MAR-2005. Setiap data tanggal yang dimasukkan,
harus mengikuti format tersebut, misalkan,

TANGGAL ← '12-DEC-2001'

TANGGAL ← '31-jan-2003'

TANGGAL ← '1-jun-03'

Berikut ini cara memasukkan data yang salah,

TANGGAL ← '1-12-2001'

TANGGAL ← '12/3/2001'

Jika dikehendaki untuk memasukkan data tanggal dengan format lainnya, maka harus dilakukan
pengubahan format, dengan menggunakan perintah,

TO_DATE(TANGGAL,FORMAT)

Contoh, untuk memasukkan tanggal dengan bentuk '12-03-2001', digunakan perintah,

TANGGAL ← TO_DATE('12-03-2001','DD-MM-YYYY')

Delete

Perintah delete digunakan untuk menghapus isi dari tabel, satu baris, beberapa baris atau semua
isi tabel, sesuai dengan yang diinginkan. Perintah ini tidak menghapus, dalam arti
menghilangkan tabel, tetapi hanya menghapus isi tabel, sedangkan tabelnya sendiri masih ada.
Untuk menghilangkan tabelnya sendiri, digunakan perintah DROP.

Format umum untuk menghapus tabel adalah,

DELETE FROM **NAMA_TABLE** WHERE **KRITERIA**;

Contoh untuk menghapus nama pegawai yang memiliki gaji lebih kecil dari 500000 digunakan perintah,

```
Delete from pegawai where gaji<500000;
```

Update

Perintah update digunakan untuk mengubah isi dari suatu tabel. Satu baris, beberapa baris atau semua isi tabel, sesuai dengan kriteria yang diberikan. Perintah ini dapat mengubah hanya satu kolom, beberapa kolom atau semua kolom dari tabel dan urutannya tidak perlu sesuai dengan urutan kolom dalam tabel. Ini karena setiap kolom harus disebutkan namanya satu per satu.

Format umum untuk mengubah isi tabel adalah,

```
UPDATE NAMA_TABEL SET KOLOM1=DATA1, KOLOM2=DATA2, ..., KOLOMn=DATAAn WHERE  
KRITERIA;
```

Contoh untuk menaikkan semua gaji pegawai yang statusnya sebagai honorer dengan kenaikan 10 persen, dapat digunakan perintah sebagai berikut,

```
Update pegawai set gaji=gaji*1.1 where status='HONORER' ;
```

DCL

DCL kependekan dari Data Control Language, yaitu perintah SQL untuk keperluan pengendalian database. Ada cukup banyak perintah dalam kategori ini. Di sini hanya diberikan dua contoh yang banyak digunakan, yaitu commit dan rollback.

Commit

Secara default, Oracle menyediakan fasilitas untuk menampung hasil dari perintah-perintah yang masuk, dimana hasil-hasil tersebut tidak langsung berakibat pada suatu object database.

Misalkan perintah untuk menyisipkan suatu data ke tabel, maka data yang disisipkan tersebut tidak benar-benar masuk ke tabel yang dikehendaki. Ini bertujuan agar dapat dilakukan

pembatalan perintah sebelumnya yang sudah dilakukan. Misalkan dalam kasus tadi, membatalkan data yang sudah disisipkan.

Jika ingin hasil dari perintah SQL yang telah dilakukan benar-benar berakibat pada object yang dimaksud, dapat diketikkan perintah commit. Artinya perintah-perintah yang telah dilakukan sebelumnya benar-benar dilaksanakan pada object yang dimaksud.

Rollback

Rollback digunakan untuk menghapus kembali perintah-perintah SQL yang sebelumnya telah dilakukan, sebelum perintah commit dilakukan. Jika setelah menjalankan suatu perintah tertentu, kemudian diberikan perintah commit, perintah setelah commit tersebut tidak dapat dibatalkan kembali.

Mekanisme rollback ini sangat membantu dalam transaksi-transaksi yang cukup rumit yang memerlukan berbagai konfirmasi untuk melakukan. Jika suatu transaksi sudah terlanjur dilakukan, dengan perintah batal (rollback), transaksi tersebut dapat dibatalkan kembali. Namun demikian, dalam kasus tertentu, cara ini menyebabkan operasi database menjadi sedikit lebih lama, terutama jika dilakukan pada data yang besar dan dilakukan secara berulang-ulang dalam jumlah yang besar.

Biasanya dalam aplikasi-aplikasi tertentu, mekanisme rollback ini ditiadakan. Yaitu dengan memasang atau mengatur sebagai auto commit, artinya setiap perintah yang masuk langsung di-commit. Mekanisme ini dapat juga ditiadakan dengan mengatur pada tablespace-nya agar tidak dapat di-rollback.

Contoh,

```
select * from dept;
Insert into emp
insert into dept values (55, 'EDUCATION', 'PENS') ;
select * from dept;
rollback;
select * from dept;
insert into dept values (55, 'EDUCATION', 'PENS') ;
```

```
select * from dept;
commit;
rollback;
select * from dept;
```

Fungsi

Oracle menyediakan berbagai macam fungsi dasar yang dapat digunakan dalam operasi query, misalkan pada perintah select. Ada dua macam fungsi yang didukung Oracle, Fungsi SQL (built-in) dan user define function. Untuk fungsi SQL, dapat dikategorikan dalam fungsi dengan satu hasil satu baris data dan fungsi dengan satu hasil untuk banyak data.

Fungsi dengan Satu Data Satu Hasil

Yang dimaksud dengan fungsi dengan satu hasil adalah fungsi yang hanya menghasilkan satu baris hasil untuk setiap baris data. Fungsi ini dapat digunakan pada WHERE, START WITH, CONNECT BY. Ada kelompok yang masuk dalam fungsi ini.

Bilangan

Fungsi bilangan digunakan untuk menerima data dalam bentuk bilangan dan mengeluarkan hasil juga dalam bentuk bilangan. Kebanyakan dari fungsi ini memiliki ketelitian sampai 38 digit.

Fungsi transcendental COS, COSH, EXP, LN, LOG, SIN, SINH, SQRT, TAN, and TANH memiliki ketelitian sampai 36 digit. Fungsi transcendental ACOS, ASIN, ATAN, and ATAN2 memiliki ketelitian sampai 30 digit.

ABS(n)	, nilai absolute
ACOS(n)	, nilai arc cosinus
ADD_MONTHS (d, n)	, tanggal d ditambah bulan n
ATAN (n)	, nilai arc tangen
BITAND (n,m)	, perhitungan and dari bit
CEIL (n)	, integer terkecil yang lebih besar atau sama dengan n
COS (n)	, fungsi cos
COSH (n)	, cos hiperbola
EXP (n)	, eksponensial
FLOOR (n)	, integer terbesar yang lebih kecil atau sama dengan n
LN (n)	, logaritma natural
LOG (m,n)	, logaritma basis m
MOD (m,n)	, sisa hasil bagi
POWER (m,n)	, m pangkat n

ROUND (m,n)	, pembulan ke atas n digit desimal
SIGN (n)	, mencari tanda bilangan
SIN (n)	, fungsi sinus
SINH (n)	, fungsi hiperbolik sinus
SQRT (n)	, fungsi akar
TAN (n)	, tangensial
TANH (n)	, hiperbolik tangensial
TRUNC (m,n)	, pembulatan ke bawan n digit

Contoh,

```
Select sin(1,24) from dual;
Select add_month(sysdate,2) dua_bulan_lagi from dual;
Select round(123.456) from dual;
```

String

Nilai string mengembalikan string

CHR(n)	, mengeluarkan kode char dari bilangan n
CONCAT(m,n)	, menggabungkan m dengan n
INITCAP(n)	, mengubah huruf awal menjadi huruf besar
LOWER (n)	, mengubah menjadi huruf kecil semua
LPAD (n,m,x)	, menngeluarkan string n rapat kanan sejumlah m, mengisi spasi dengan x
LTRIM (n)	, menghilangkan spasi pada sisi kiri
NLS_INITCAP	, huruf awal besar, dan dilakukan transfer bentuk
NLS_LOWER	, huruf kecil semua dan dilakukan transfer bentuk
NLSSORT	
NLS_UPPER	, huruf besar semua dan dilakukan transfer bentuk
REPLACE (n,m,x)	, mengubah string m dalam n menjadi x
RPAD (m,n,x)	, rapat kiri sejumlah n dan mengisi spasi dengan x
RTRIM (n)	, membuang spasi sebelah kanan
SOUNDEX	
SUBSTR (m,n,x)	, mengambil sebagian string m, mulai n sebanyak x
SUBSTRB	
TRANSLATE (n,m,x)	, menterjemahkan string n sesuai tabel konversi m ke x
TRIM (n)	, menghilangkan spasi di kiri dan kanan
UPPER(n)	, mengubah ke huruf besar

Contoh,

```
Select upper('ini huruf kecil') huruf_besar from dual;
Select substr('ini suatu kalimat',5,4) from dual;
```

String mengembalikan nilai

ASCII (char) , mengembalikan nilai ascii
INSTR (n,m,x) , mencari posisi string m dalam string n mulai x
INSTRB
LENGTH , menghitung panjang string
LENGTHB

Contoh

```
Select length('ini sebuah kalimat') panjang from dual;  
Select instr('saudara sekalian','dara',1) posisi from dual;
```

Fungsi Tanggal

ADD_MONTHS
LAST_DAY
MONTHS_BETWEEN
NEW_TIME
NEXT_DAY
ROUND (date function)
SYSDATE
TRUNC (date function)

Contoh,

```
Select sysdate waktu_sekarang from dual;  
Select next_day(sysdate) besok from dual;
```

Fungsi Konversi

CHARTOROWID
CONVERT
HEXTORAW
NUMTODSINTERVAL
NUMTOYMINTERVAL
RAWTOHEX
ROWIDTOCHAR
TO_CHAR (date conversion)
TO_CHAR (number conversion)
TO_DATE
TO_LOB
TO_MULTI_BYTE
TO_NUMBER
TO_SINGLE_BYTE
TRANSLATE ... USING

Contoh

```
Select to_date('12-01-2002','dd-mm-yyyy') from dual;
```

```
Select to_char(sysdate,'dd') tanggal, to_char(sysdate,'mm') bulan,  
to_char(sysdate,'yyyy') tahun from dual;
```

Aggregate

Fungsi aggregate adalah fungsi yang mengolah lebih dari satu baris data menjadi satu hasil. Hasil dari perhitungan dipengaruhi oleh jumlah pengelompokan (GROUP BY) perhitungan. Kalau dikelompokkan dan terdapat lebih dari satu kelompok, maka jumlah hasilnya sesuai dengan jumlah kelompok yang ada.

AVG	, menghitung rata-rata
CORR	
COUNT	, menghitung jumlah baris
COVAR_POP	
COVAR_SAMP	
GROUPING	
MAX	, mencari nilai tertinggi
MIN	, mencari nilai terendah
REGR_ (linear regression) functions	
STDDEV	
STDDEV_POP	
STDDEV_SAMP	
SUM	, menghitung jumlah nilai
VAR_POP	
VAR_SAMP	
VARIANCE	

Contoh,

```
Select sum(sal) total_gaji from emp;  
Select count(empno) total_pegawai from emp;  
Select count(sal) jumlah_pegawai ,deptno from emp group by deptno;
```