



MODUL PRAKTIKUM BASIS DATA



Disusun oleh Tim Dosen Teknik Informatika



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS ESA UNGGUL**



Pertemuan I

Pengenalan Sql Server 2000

Tujuan

Mahasiswa memahami komponen database (khususnya Sql Server 2000),

Materi

1. Arsitektur Client/Server
2. Arsitektur Microsoft SQL Server
3. Komponen Dasar Microsoft SQL Server
4. Persyaratan Microsoft SQL Server
5. Instalasi Sql Server 2000

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

Pendahuluan

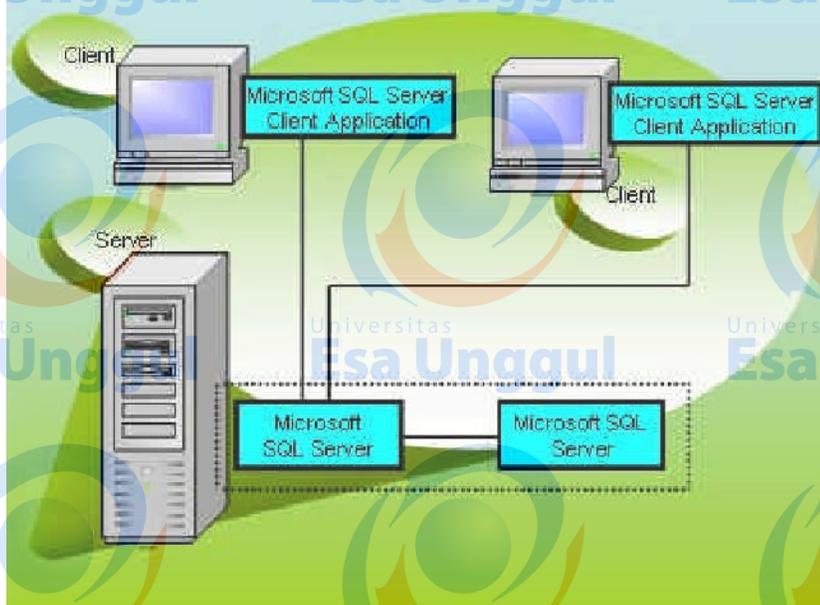
Microsoft SQL Server merupakan aplikasi database handal yang digunakan oleh sebagian besar perusahaan terkemuka di dunia termasuk di Indonesia. Microsoft SQL Server 2000 merupakan pendobrak dan inovasi database modern yang mengetengahkan kemudahan, kecepatan, ketepatan dan kecanggihan dalam mengelola sebuah database modern berskala kecil, menengah dan besar. Melihat kemampuan yang sangat hebat ini Microsoft SQL Server 2000 mendapat julukan **The Next Generation Database**. Dengan demikian Microsoft SQL Server 2000 merupakan solusi database modern yang mampu mengelola data Warehousing, komputer portable serta sektor e-Commerce.

Tidak heran saat ini masyarakat komputer di Indonesia termasuk mahasiswa mulai gandrung dan mencari solusi terbaik untuk mempelajari Microsoft SQL Server 2000 secara cepat, mudah dan tentu saja akurat.

1. Arsitektur Client/Server

Microsoft SQL adalah sebuah database relational yang dirancang untuk mendukung aplikasi dengan arsitektur Client/Server, di mana database terdapat pada komputer pusat yang disebut sebagai Server dan informasi digunakan bersama-sama oleh beberapa user yang menjalankan aplikasi di dalam komputer lokal yang disebut dengan Client. Dengan teknologi ini semua informasi bisa digunakan secara bersama dan tentu saja informasi yang digunakan juga sama. Sehingga tidak akan terjadi perbedaan informasi antara satu user dengan lainnya.

Dengan arsitektur Client/Server ini akan mengurangi padatnya lalu lintas data dalam suatu jaringan, karena apabila pemakai menginginkan suatu data dari ratusan ribu bahkan jutaan data, maka hanya data yang diminta saja yang akan dikirimkan melalui jaringan tersebut kepada Client. Hal ini sangat berbeda dengan sistem tradisional, di mana pada sistem ini semua data akan dikirim melalui jaringan sehingga lalu lintas data dalam suatu jaringan akan terasa padat tentu saja akan mengurangi kinerja jaringan tersebut secara menyeluruh. Sebagai gambaran berikut ini adalah beberapa tipe dan cara kerja Microsoft SQL Server.



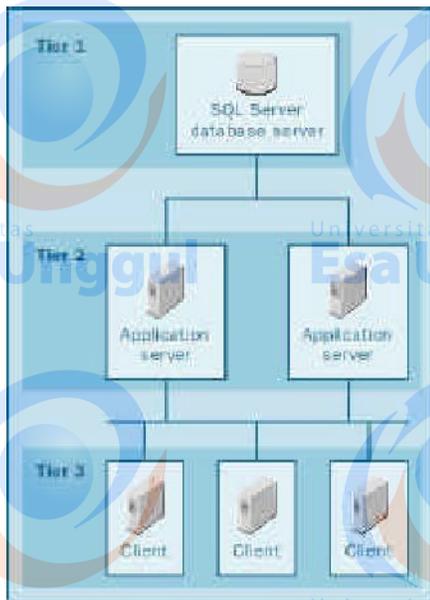
Gambar 1.1. Contoh sederhana Arsitektur Client/Server

2. Arsitektur Microsoft SQL Server

Database dalam Microsoft SQL Server dibagi ke dalam beberapa komponen logikal, antara lain, tabel, view dan elemen-elemen lain yang terlihat oleh user. Elemen-elemen ini secara fisik disimpan dalam dua atau lebih file dalam suatu media penyimpanan data. Sedangkan mengenai format file atau lokasi elemen-elemen logik ditulis tidak diketahui oleh user.



Gambar 1.2. Contoh system 2 Tier



Gambar 1.3. Contoh system 3 Tier

Microsoft SQL Server bisa mengandung beberapa database yang digunakan oleh beberapa user, misalnya sebuah perusahaan bisa memiliki sebuah database yang akan digunakan oleh banyak karyawannya pada beberapa departemen atau bagian. Bahkan tidak menutup kemungkinan dalam suatu instansi pemerintah atau swasta bisa saja memiliki beberapa database yang hanya digunakan khusus oleh user tertentu.

Sebagai gambaran berikut ini adalah contoh dalam tiga tabel dan implementasi fisik dari tabel-tabel tersebut melalui penggunaan file data, file indeks, dan file log.





Gambar 1.4. Contoh arsitektur SQL Server yang terdiri atas tabel dan file database

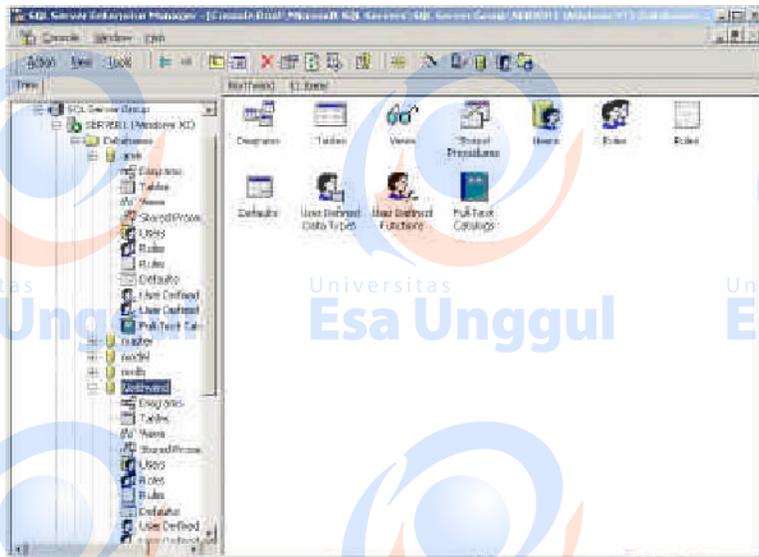
3. Komponen Dasar Microsoft SQL Server

Apabila Anda akan mempelajari atau bekerja menggunakan Microsoft SQL Server ada beberapa komponen penting yang perlu diketahui. Komponen-komponen ini merupakan inti dari sebuah database dalam Microsoft SQL Server.

3.1. Database

Mengandung objek-objek yang digunakan untuk mewakili, menyimpan dan mengakses data. Yang dimaksud dengan database dalam SQL Server adalah kumpulan Tabel, View, Indeks, Trigger, Procedur dan objek-objek lain yang terkandung di dalamnya. Yang perlu dicatat oleh semua developer dan calon developer adalah dengan mengimplementasikan Microsoft SQL Server semua file database Anda akan dipelihara dengan baik.

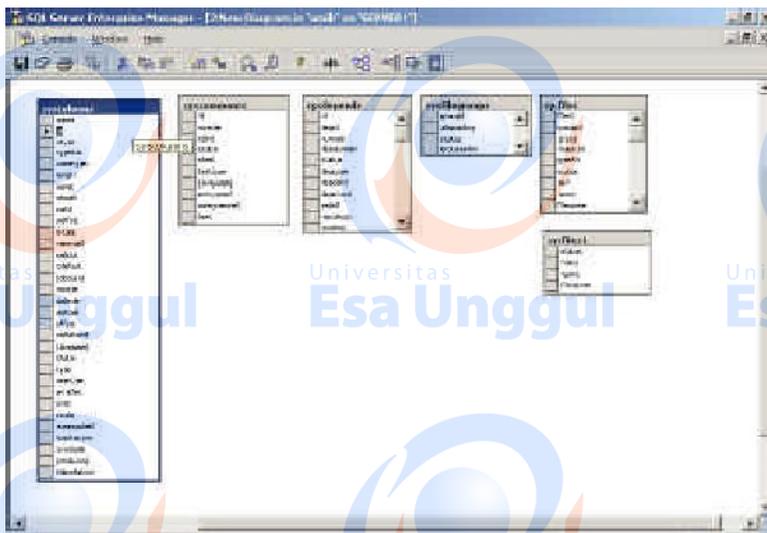
Berikut ini adalah gambaran mengenai komponen-komponen penting dalam Microsoft SQL Server 2000 yang di dalamnya mengandung beberapa komponen sebagaimana layaknya sebuah database modern.



Gambar 1.5. Menampilkan sebuah server bernama SERVER1 (Windows NT)

3.2. Tabel

Tabel sebenarnya merupakan sarana untuk menyimpan baris-baris atau record-record data dan hubungannya dengan tabel lain. Jadi yang dimaksud dengan tabel di sini adalah inti dari sebuah database. Tabel menyimpan data yang dikelompokkan di dalam bentuk baris dan kolom seperti layaknya lembar kerja. Setiap baris mewakili record dan setiap kolom adalah atribut atau field serta setiap field mengandung satu jenis informasi. Sebagai contoh field bernama Value memiliki tipe data Numeric, sementara field bernama Name memiliki informasi bertipe Teks. Field-field pada tabel ini bisa memiliki pembatasan mengenai isi yang akan disimpan ke dalam field tersebut. Misalnya sebuah field bernama Quantity tidak bias menyimpan teks dan hanya bisa menyimpan angka serta tidak bias menerima nilai di bawah 1 atau di atas 100.



Gambar 1.6. Menampilkan Database Diagram

3.3. Database Diagram

Secara grafis menampilkan objek database sehingga dapat dimanipulasi tanpa menggunakan bahasa Transact-SQL. Dengan bahasa Transact-SQL menjadi Microsoft SQL Server mampu menghasilkan diagram database yang canggih. Diagram database ini adalah representasi grafik dari Tabel, Indeks, dan View yang disimpan oleh database dan bisa dimanipulasi dengan teknik drag-and-drop dan interaksi dengan kotak dialog. Dengan demikian beberapa tugas bisa dilakukan tanpa perlu menggunakan bahasa Transact-SQL, seperti misalnya mengubah karakteristik fisik dari database atau tabel-tabelnya. Inilah kelebihan Microsoft SQL Server dalam menangani berbagai masalah dengan solusi yang cukup mudah dan cepat.

3.4. Indeks

Indeks adalah file-file tambahan yang meningkatkan kecepatan akses dari baris-baris tabel. Jadi Indeks adalah file jenis khusus yang bekerja sama dengan tabel. Tujuannya adalah untuk mempercepat proses pengaksesan record atau sekelompok record tertentu.

3.5. View

View adalah tabel virtual yang isinya ditentukan oleh Query ke dalam database. View ini bukanlah tabel fisik melainkan sekumpulan instruksi yang memberikan hasil berupa serangkaian data. Dengan demikian View ini bisa dikatakan cara untuk melihat data yang berbeda di dalam satu atau lebih tabel.

3.6. Stored Procedure

Data di dalam database bisa diakses hanya melalui eksekusi perintah Transact-SQL. Ketika para developer membuat aplikasi untuk berfungsi sebagai antar muka database, pengembang bisa memilih untuk membuat program SQL yang disimpan secara lokal dan dikirimkan ke server untuk dieksekusi, atau membuat dan memelihara program di dalam Server itu sendiri di dalam prosedur tersimpan yang bisa dipicu oleh program di dalam komputer client. Prosedur tersimpan bisa menerima parameter atau nilai yang dikirimkan kepada prosedur untuk memprosesnya. Tetapi dalam hal ini pekerjaan yang dilaksanakan tidak seperti fungsi, di mana prosedur tidak akan mengembalikan nilai apapun. Setelah prosedur dibuat, ia bisa digunakan oleh aplikasi apa saja yang mampu mengakses database. Prosedur dibuat dengan perintah Transact-SQL

CREATE PROCEDURE dan diubah dengan perintah **ALTER PROCEDURE**.

Kesimpulannya adalah program-program Transact-SQL yang disimpan di dalam Server adalah yang akan menjalankan tugas-tugas yang telah ditentukan.

3.7. Trigger

Trigger adalah prosedur tersimpan yang secara otomatis dijalankan apabila data di dalam tabel berubah karena eksekusi perintah Microsoft SQL seperti **INSERT**, **UPDATE**, atau **DELETE**. Salah satu dari kegunaannya yang paling umum adalah untuk menerapkan pembatasan yang lebih kompleks dari yang telah diizinkan melalui pembatasan CHECK yang berfungsi membatasi informasi yang disimpan ke dalam kolom. Trigger bisa dibuat bersama dengan perintah INSERT yang akan melakukan Query ke tabel lain dan mengembalikan nilai logik yang membantu membatasi data yang diberikan kepada kolom tertentu. Sebagai contoh, Trigger bisa dibuat untuk menjalankan replikasi, misalnya apabila ada sebuah baris disisipkan ke dalam database A, sebuah baris dengan informasi yang sama akan ditambahkan ke dalam database B. Jadi Trigger dibuat sebagai sebuah transaksi dan bisa dimundurkan apabila ada masalah yang terdeteksi. Dengan demikian Trigger adalah sebuah prosedur tersimpan khusus yang secara otomatis dijalankan apabila operasi tertentu dilakukan di dalam tabel.

3.8. Full-Text Index

Indeks khusus yang membuat pencarian menjadi lebih mudah di dalam kolom-kolom dengan tipe data Varchar dan Text.

4. Persyaratan Microsoft SQL Server 2000

Untuk menjalankan Microsoft SQL Server 2000 ada beberapa persyaratan teknis yang harus dipenuhi baik berupa hardware, sistem operasi maupun infrastrukturnya. Secara sederhana saya memberikan contoh Network dan beberapa hardware yang harus dipenuhi apabila Anda akan mengimplementasikan Microsoft SQL Server.

4.1. Persyaratan Hardware dan Sistem Operasi

Server yang digunakan minimal harus mampu bekerja dengan baik dan siap melayani data selama 24 jam kalau server tersebut digunakan sebagai Web server. Oleh karena itu untuk server ini harus benar-benar canggih sesuai kebutuhan untuk aplikasi database modern.

4.1.1. Persyaratan Server

Apabila Anda akan mengimplementasikan Microsoft SQL Server di kantor Anda, maka diperlukan suatu jaringan LAN agar Microsoft SQL Server bisa digunakan secara maksimal.

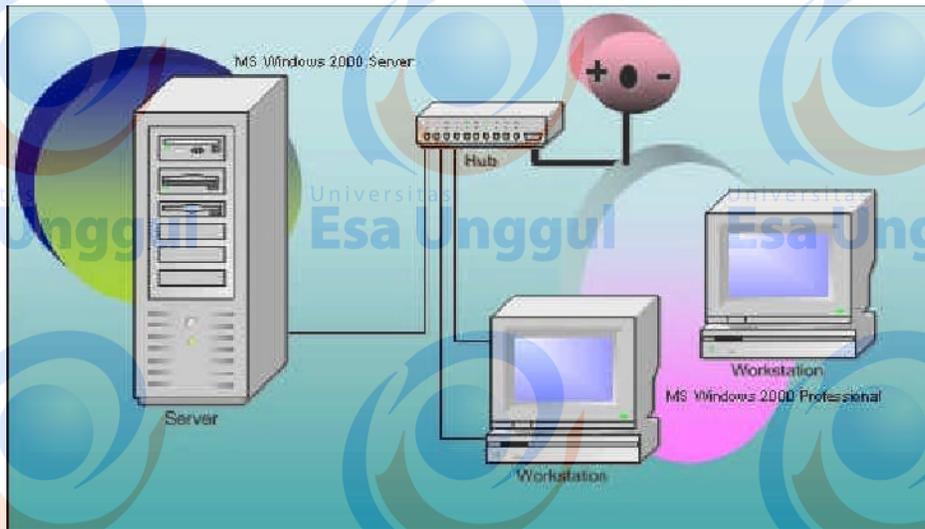
Pentium II, III atau Pentium IV dan yang setara dengan itu

RAM 256 MHz atau lebih

Hard disk 40 Gbyte atau lebih

NIC (Network Interface Card)

Sistem Operasi Microsoft Windows 2000 Server atau Microsoft Windows.NET



Gambar 1.7. LAN merupakan persyaratan sederhana untuk mengaplikasikan Microsoft SQL Server 2000

4.1.2. Persyaratan Client

Sistem komputerisasi berbasis Client/Server sudah bisa dikatakan demikian apabila sudah ada koneksi antara dua atau lebih komputer yang bekerja secara bersama, sehingga terjadi komunikasi dan tukar menukar data. Maka dalam suatu jaringan selain diperlukan suatu server juga diperlukan komputer sebagai Client.

Pentium II, III atau Pentium IV dan yang setara dengan itu

RAM 128 MHz atau lebih

Hard disk 10 Gbyte atau lebih

NIC (Network Interface Card)

Sistem Operasi Microsoft Windows 2000 professional atau

Microsoft Windows XP

4.2. Persyaratan Network

Seperti telah dijelaskan sebelumnya bahwa Microsoft SQL Server merupakan aplikasi database berbasis Client/Server. Maka apabila akan mengimplementasikan Microsoft SQL Server diperlukan suatu jaringan, baik LAN maupun lainnya. Sebagai gambaran jaringan sederhana berbasis LAN seperti ditampilkan Gambar 1.7. di atas.

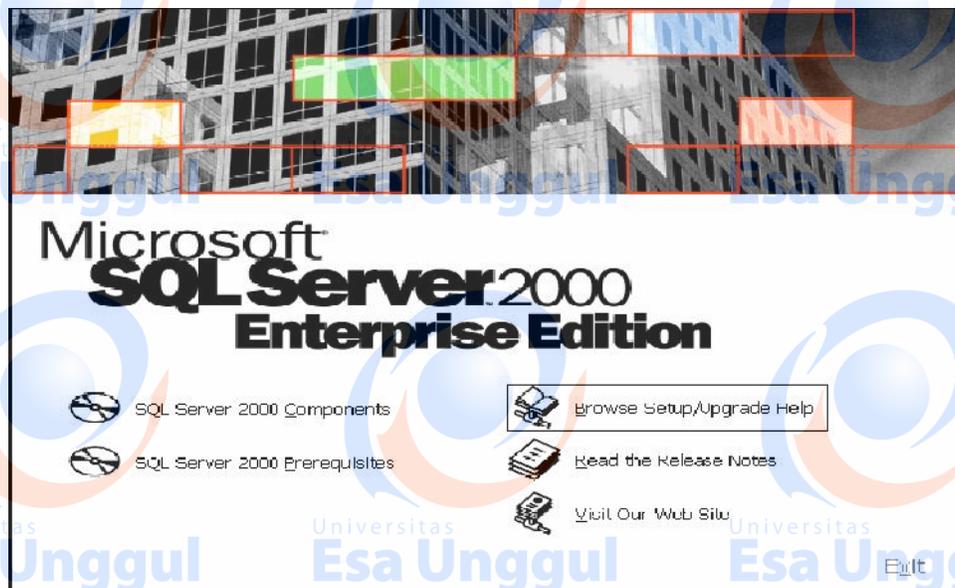
5. Instalasi

Sebelum Anda bisa menggunakan dan menjalankan aplikasi Ms SQL Server terlebih dahulu Anda harus menginstall Ms SQL Server 2000 ke dalam Komputer Anda.

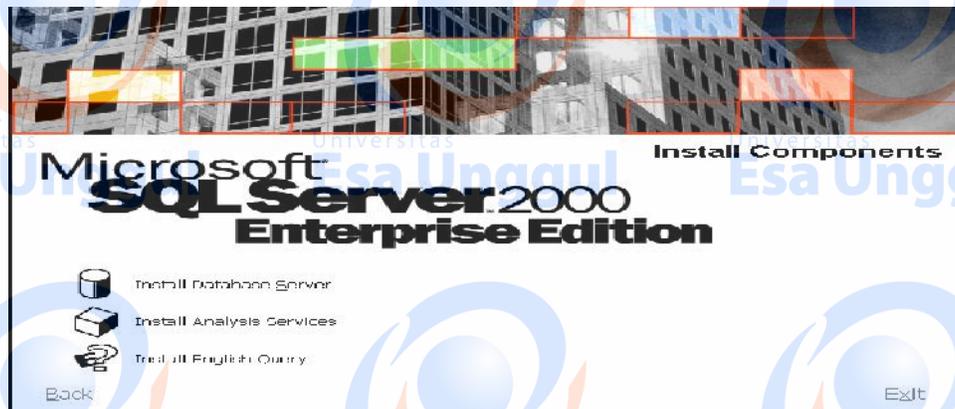
Adapun langkah-langkah yang harus dilakukan dalam menginstall Ms SQL Server 2000, dalam hal ini yang dicontohkan adalah menginstall Ms SQL Server 2000 Enterprise

Edition adalah sebagai berikut :

1. Masukkan CD Ms SQL Server 2000 Enterprise Edition tunggu sebentar hingga muncul tampilan seperti gambar 1, atau jika tidak muncul Anda bisa membuka windows explorer kemudian pilih cd room yang berisi CD Ms SQL Server 2000, kemudian klik file Autorun.exe.



2. Klik pada SQL Server 2000 Componen, sehingga muncul tampilan seperti



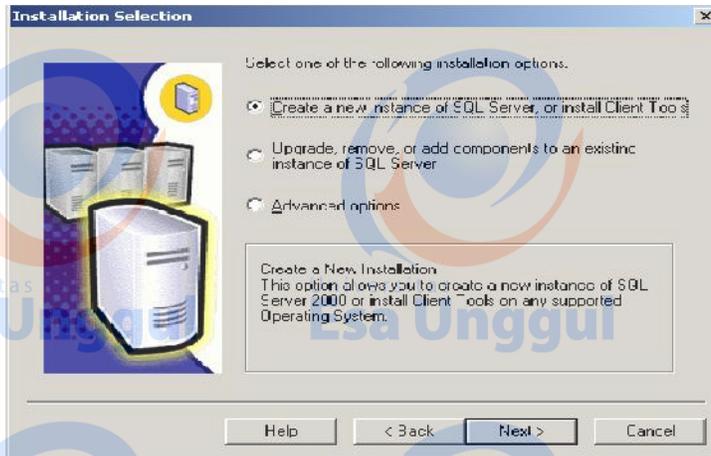
3. Klik Install Database Server kemudian akan muncul tampilan seperti gambar 3



4. Klik tombol Next sehingga muncul pilihan nama komputer



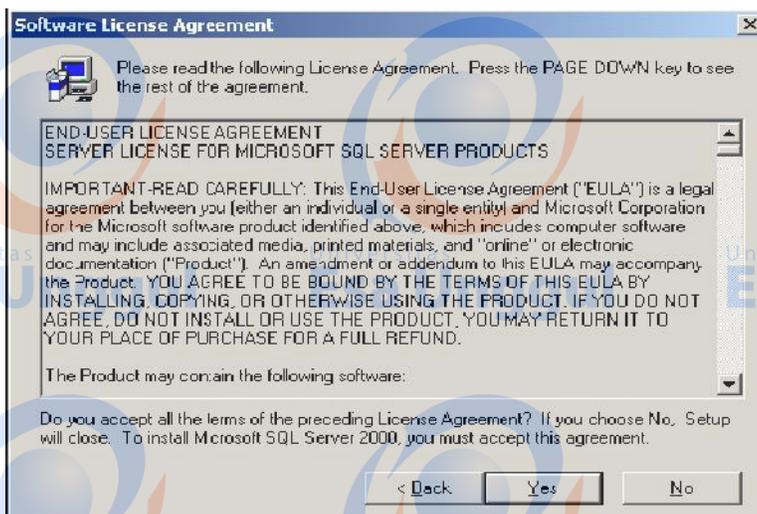
5. Klik Local Computer dan kemudian klik tombol Next



- Pilih Create a new instance of SQL Server, or install Client Tools kemudian klik tombol Next



- Masukkan Nama dan Perusahaan Anda kemudian klik tombol Next



- Jika Anda setuju dengan lembar kesepakatan yang di tawarkan seperti gambar 8

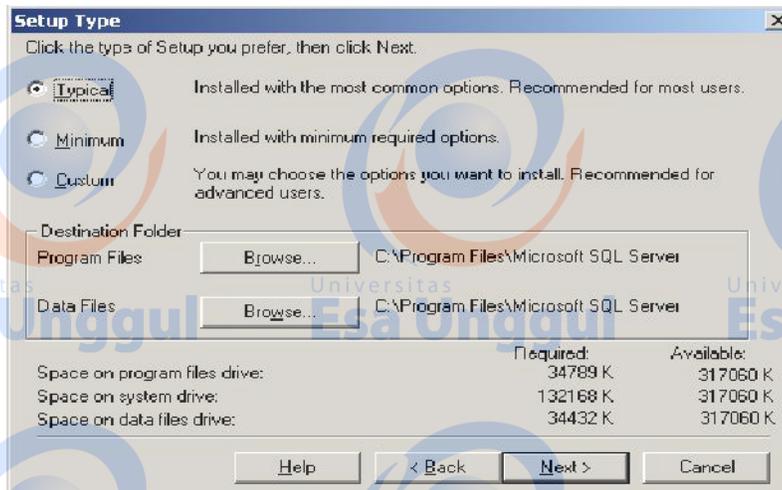
klik tombol Yes dan jika Anda tidak setuju maka klik tombol No dan itu berarti Anda membatalkan proses Instalasi.



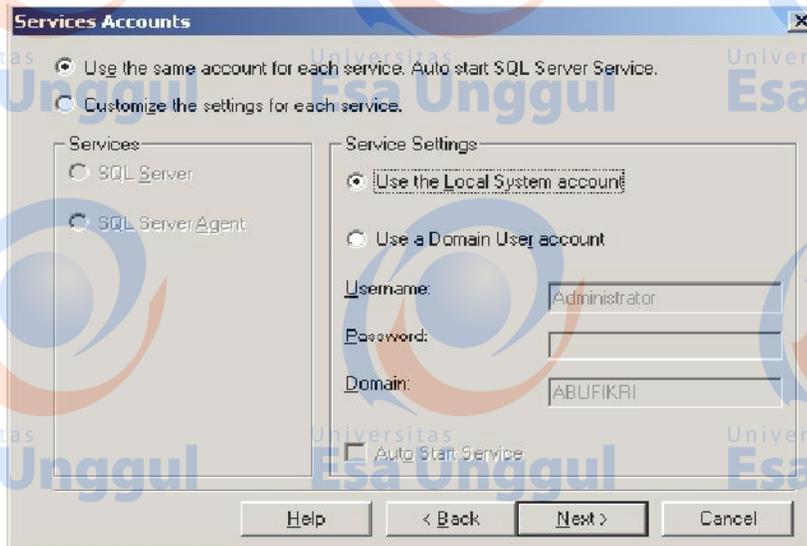
9. Pilih Server and Client Tools agar anda bisa mengelola database secara langsung dengan fasilitas Client Tools.



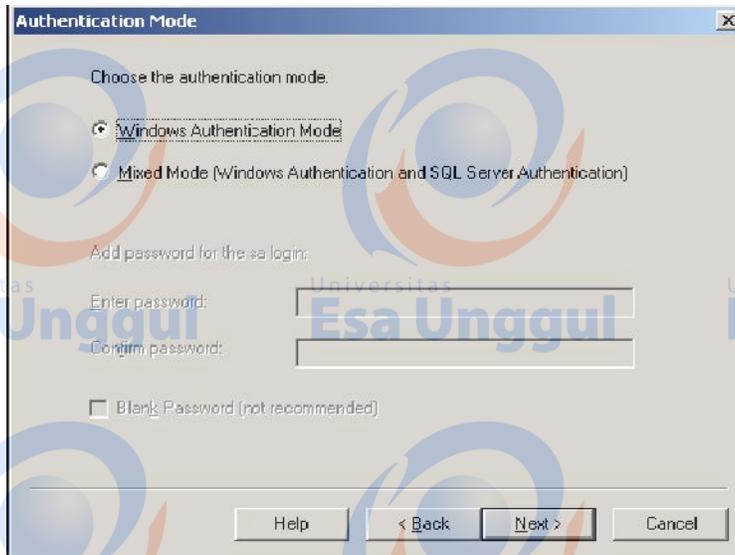
10. Insikan Nama Instansi Anda kemudian klik tombol Next



11. Tentukan lokasi file SQL server yang akan Anda install dengan mengklik tombol Browse atau klik Next untuk melanjutkan ke proses berikutnya



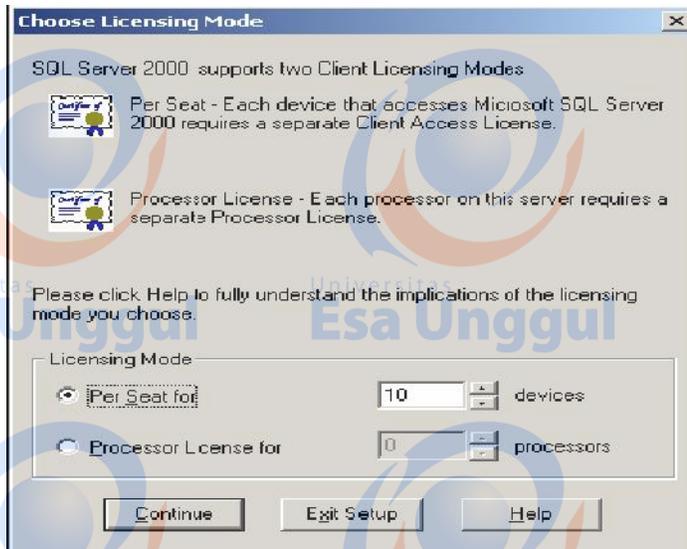
12. Pilih Use the same account for each service. Auto start SQL Server Service dan untuk Service Settings pilih Use the Local System account.
13. Untuk Authentication Mode sebaiknya pilih Windows Authentication Mode agar keamanan database server kita lebih terjaga karena dengan fasilitas ini hanya user yang terdaftar di windows yang di install SQL Server. Jika Anda memilih Mixed Mode berarti selain user yang terdaftar di Windows, user yang tidak terdaftar di Windows juga bisa mengakses server jika didaftarkan di SQL Server.



14. Jika pemilihan mode telah selesai klik tombol Next

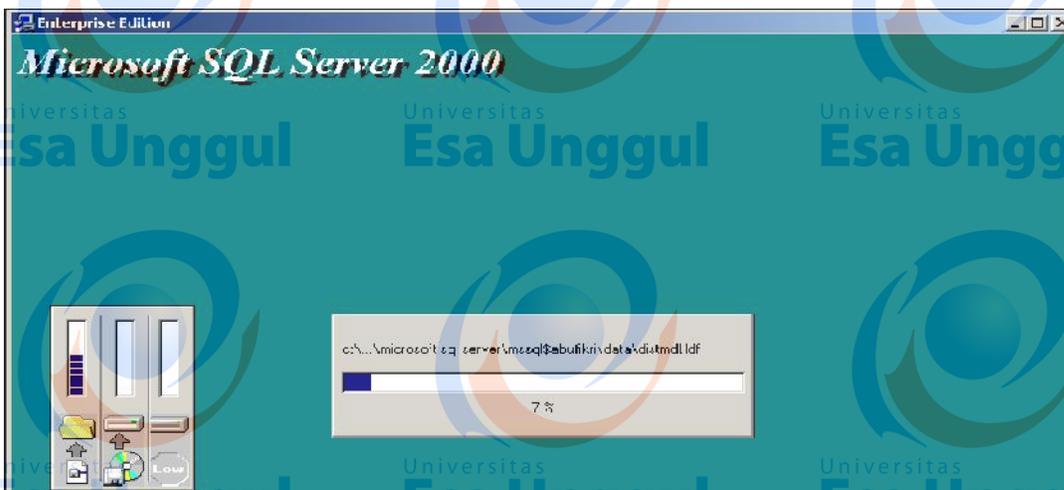


15. Klik tombol Next untuk melanjutkan ke proses berikutnya.



16. Pilihlah jenis lisensi apakah Per Seat For atau Processor License For. **Per Seat For lisensi** yang lebih ditekankan kepada berapa jumlah user/computer yang bisa mengakses server.

Processor License for adalah lisensi yang ditekankan kepada berapa jumlah prosesor yang bisa mengoperasikan SQL Server atau dengan kata lain lebih kepada berapa jumlah server yang bisa mengoperasikan SQL server. Jika jenis lisensi sudah di pilih dan jumlahnya maka selanjutnya klik tombol Continue untuk melanjutkan proses instalasi.



17. Tunggu sampai proses instalasi (seperti gambar 16) selesai



18. Proses instalasi SQL Server 2000 telah selesai dengan ditandai dengan munculnya jendela seperti gambar 17, klik tombol Finish

Pelaksanaan Praktikum

1. Memahami Arsitektur dan komponen-komponen Sql Server 2000.
2. Cara Menginstal Sql Server 2000

Tugas.

1. Apa pengertian DBMS
2. Sistem Database terbagi dari beberapa komponen? Sebutkan dan jelaskan.
3. Apa Manfaat/keunggulan DBMS dan Jelaskan
4. Jelaskan dan sebutkan Arsitektur DBMS
5. Sebutkan Komponen SQL Server 2000, dan Jelaskan





Pertemuan II

Interface SQL Server 2000, USER DAN DATABASE.



Tujuan

Mahasiswa memahami dan mengetahui interface SQL Server 2000, bagaimana menjalankan Sql Server dan cara membuat User dan database dengan Enterprise Manager.

Materi

- 
- 
- 
1. Interface SQL Server
 2. Menjalankan Sql Server
 3. Mengelola SQL Server, dengan membuat User dan database



Software

1. Windows 2003 Operating System.
2. Sql Server 2000



Landasan Teori

Interface SQL Server

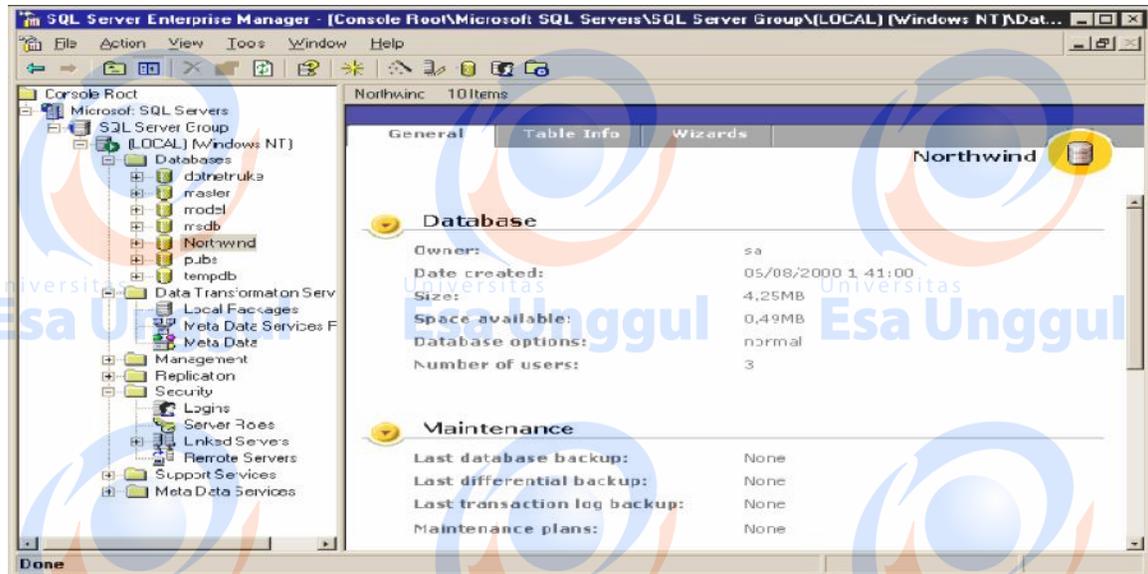
Ada 3 interface utama saat Anda bekerja dengan SQL Server:



A. Enterprise Manager

Merupakan interface utama dan paling sering digunakan oleh administrator database. Bagian ini mengandung sebagian besar fungsi-fungsi pokok dalam mengatur database.





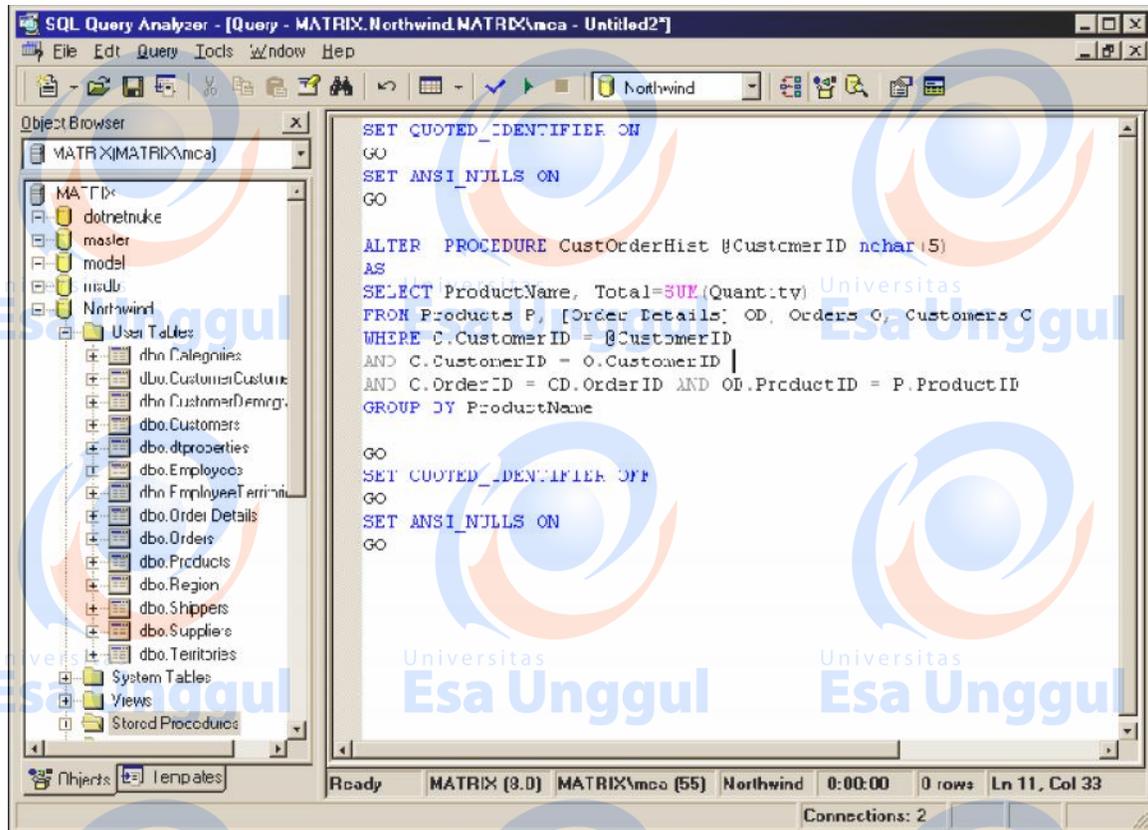
Di dalam folder database ditampilkan berbagai database yang ada. Database master, model, msdb, dan tempdb merupakan default system database yang diperlukan agar SQL Server dapat berfungsi baik. Keempat database ini tidak boleh dihapus ataupun dimodifikasi tanpa pengetahuan yang mencukupi tentang system SQL Server.

Sedangkan NorthWind dan pubs adalah database sampel yang dapat digunakan untuk berlatih perintah SQL maupun administration job. Dalam tutorial selanjutnya, kita akan banyak menggunakan NorthWind sebagai kasus. Di dalam folder Security terdapat tool Login yang berisi daftar user di dalam database. Di bagian ini semua manajemen menyangkut user account dilakukan. Fungsi-fungsi lain dalam Enterprise Manager akan dibahas pada artikel berikutnya tentang SQL Server Lanjutan.

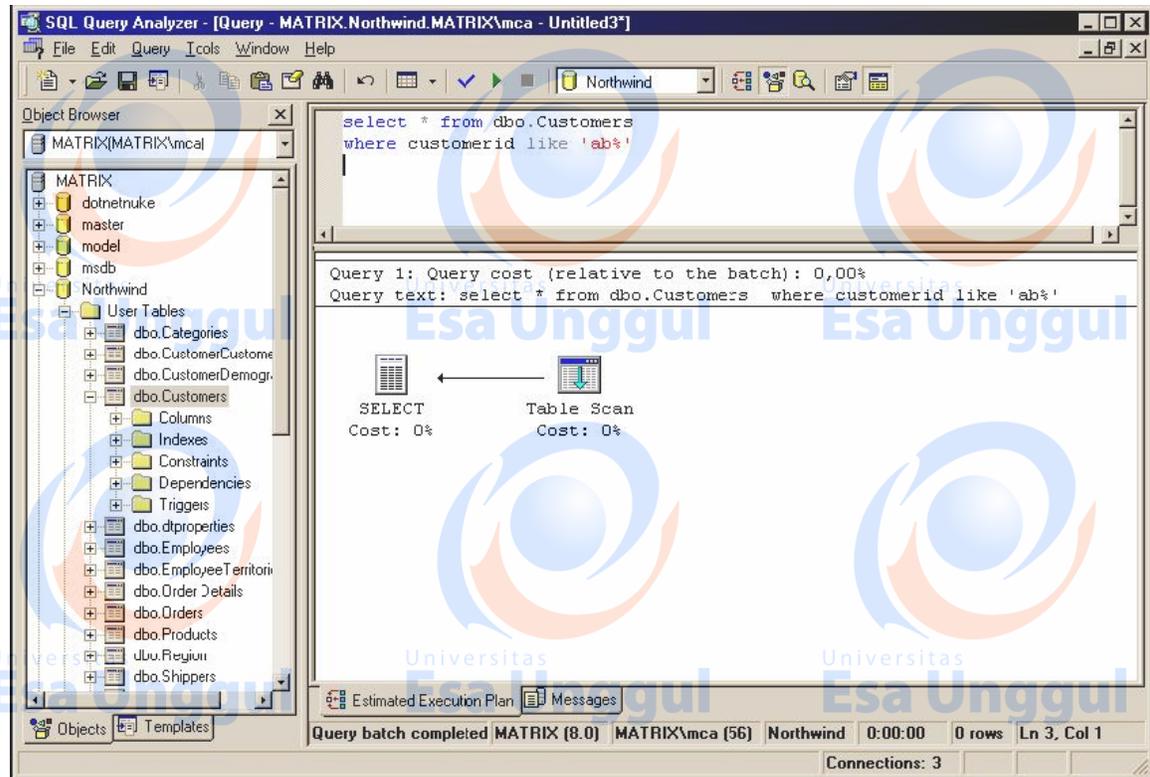
B. Query Analyser

Tool ini merupakan interface utama dalam melakukan pemrograman di SQL Server. Bahasa yang digunakan adalah Transact SQL (T-SQL). Anda dapat membuat perintah untuk mengambil data, sortir, manipulasi data serta melakukan perhitungan tertentu terhadap sekumpulan data dalam database.

Script yang telah dibuat dapat disimpan sebagai View ataupun Stored Procedure, sesuai dengan kebutuhan dalam pembuatan aplikasi. Tutorial



Di dalam Query Analyser juga tersedia tool lain, misalnya yang digunakan untuk menganalisis performa sebuah query dan mencari alternatif agar query tersebut dapat lebih dioptimalkan. Apabila toolbar Estimated Execution Plan diaktifkan maka ditampilkan estimasi waktu dan urutan eksekusi sebuah perintah.



Query Analyser juga dapat digunakan untuk membuat script sebuah database maupun obyek-obyek di dalamnya. Script ini selanjutnya dapat dijalankan di server lain untuk membuat database yang serupa, atau digunakan dalam sebuah aplikasi untuk mendukung pemrograman. Caranya adalah menklik kanan sebuah obyek, dan pilih Script to New Windows.

Berikut ini contoh sebuah script untuk membuat tabel Customers di dalam database NorthWind.

```
CREATE TABLE [Customers] (
[CustomerID] [nchar] (5) COLLATE Latin1_General_CI_AS NOT NULL ,
[CompanyName] [nvarchar] (40) COLLATE Latin1_General_CI_AS NOT NULL ,
[ContactName] [nvarchar] (30) COLLATE Latin1_General_CI_AS NULL ,
[ContactTitle] [nvarchar] (30) COLLATE Latin1_General_CI_AS NULL ,
[Address] [nvarchar] (60) COLLATE Latin1_General_CI_AS NULL ,
[City] [nvarchar] (15) COLLATE Latin1_General_CI_AS NULL ,
[Region] [nvarchar] (15) COLLATE Latin1_General_CI_AS NULL ,
[PostalCode] [nvarchar] (10) COLLATE Latin1_General_CI_AS NULL ,
[Country] [nvarchar] (15) COLLATE Latin1_General_CI_AS NULL ,
[Phone] [nvarchar] (24) COLLATE Latin1_General_CI_AS NULL ,
[Fax] [nvarchar] (24) COLLATE Latin1_General_CI_AS NULL
) ON [PRIMARY]
GO
```

C. Service Manager

Digunakan untuk mengatur service yang ada di SQL Server, apakah akan dijalankan atau dimatikan. Sebuah service juga dapat disetup agar berjalan otomatis sebagai Windows service, atau dijalankan secara manual.



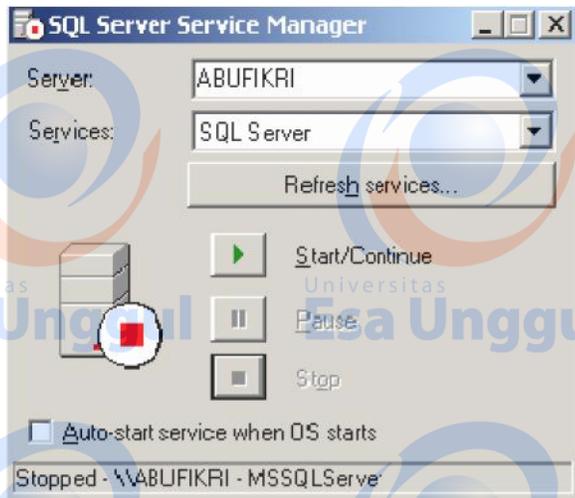
Ada 3 service standar dalam setiap instalasi default SQL Server:

- Distributed Transaction Coordinator
- SQL Server
- SQL Server Agent

II. Menjalankan SQL Server

Untuk menjalankan SQL Server lakukan langkah-langkah berikut :

1. Klik tombol Start -> All Programs -> Microsoft SQL Server -> Service Manager kemudian akan muncul tampilan seperti gambar seperti dibawah ini :



2. Jika Start/Continue nya aktif berarti posisi server masih belum aktif, klik tombol **START/CONTINUE** dan klik juga pada **AUTO SELECT SERVICE WHEN OS START** supaya begitu server nyala otomatis SQL Server akan di jalankan. Gambar adalah posisi jika SQL server sudah aktif



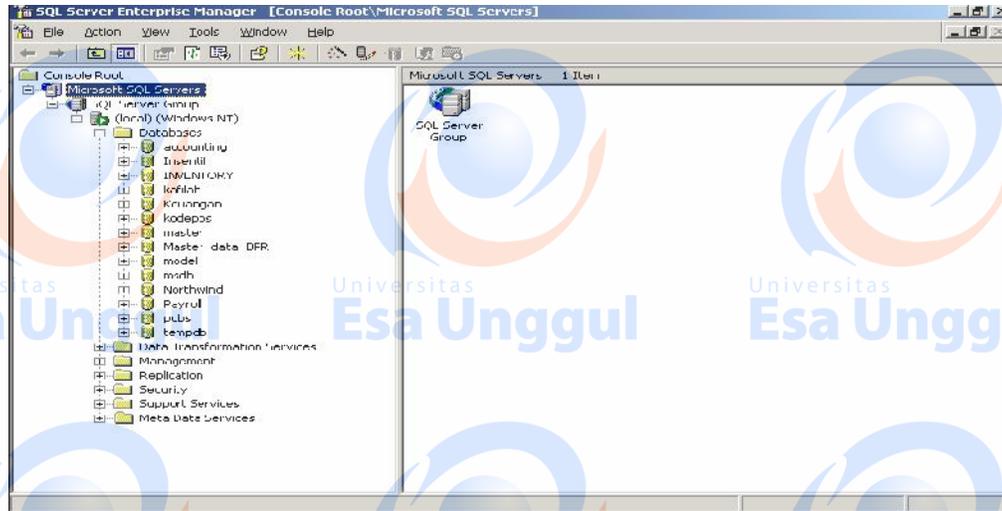
III. Mengelola SQL Server

1. Menggunakan SQL Server Enterprise Manager

SQL Enterprise Manager adalah fasilitas yang dimiliki oleh Microsoft SQL Server 2000 untuk mengelola SQL Server.

Adapun cara untuk menjalankannya klik

tombol Start -> All Programs -> Microsoft SQL Server -> Enterprise Manager kemudian akan muncul tampilan seperti gambar 20.



1.1 Mengelola User di SQL Server

1.1.a Membuat User baru

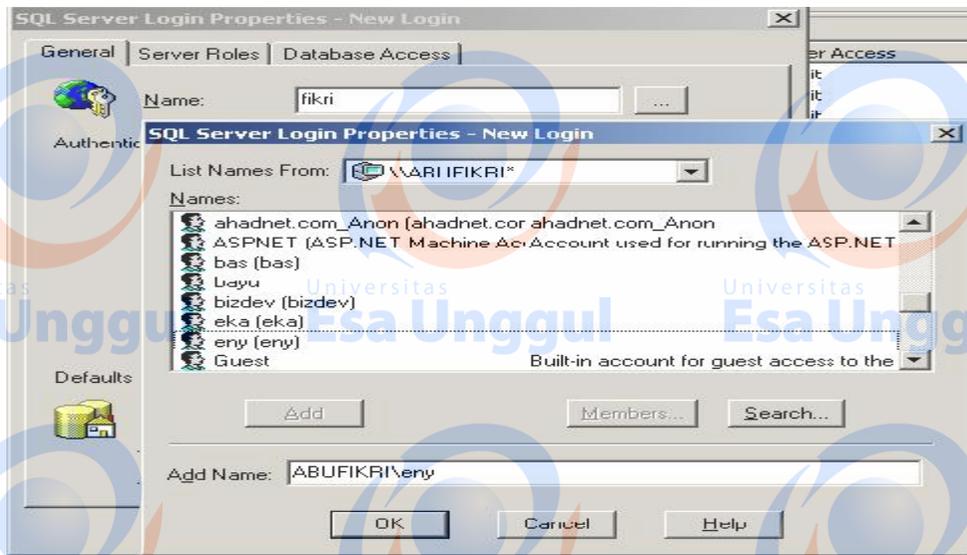
Untuk membuat user baru di SQL Server lakukan langkah-langkah sbb :

- klik tanda + pada Security Gambar 21



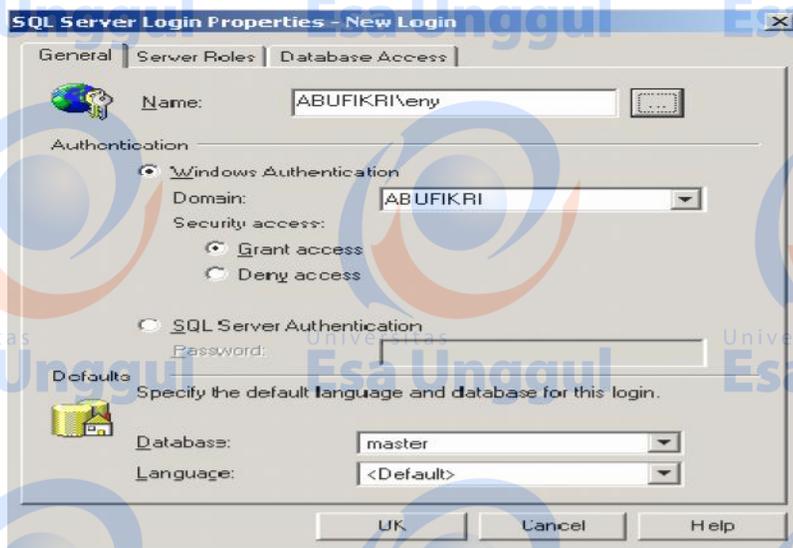
Gambar 21

- Klik kanan pada Logins kemudian pilih New Logon
- Tulis nama user yang akan dibuat atau klik tombol kemudian akan muncul jendela daftar user yang ada di Server Windows seperti gambar

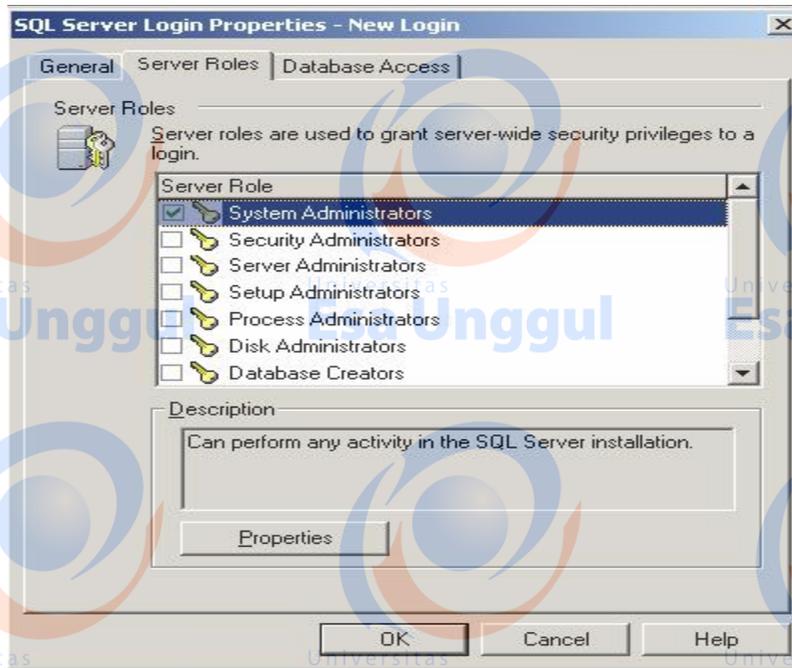


Gambar 22

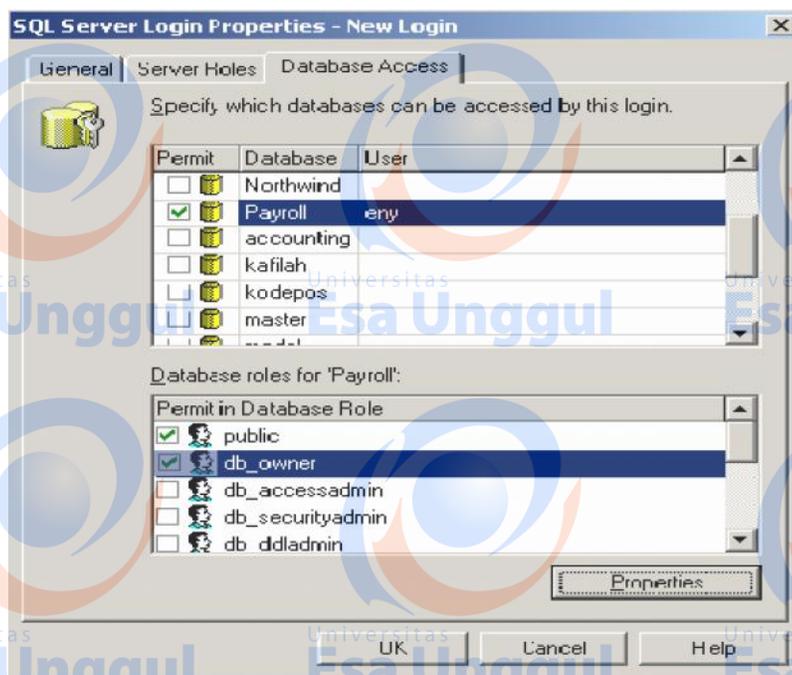
- Klik double pada user yang akan didaftarkan di SQL server kemudian klik tombol Ok dan kembali ke jendela Login Property seperti gambar 23
- Klik tab Server Roles dan pilih System Administrator seperti terlihat di gambar 24
- Kemudian klik tab Database untuk menentukan database mana yang dapat diakses dan hak akses terhadap database tersebut sebagai apa, contoh dalam hal ini databasenya Payroll dan hak aksesnya public dan db_owner Seperti yang nampak pada gambar 25



Gambar 23



Gambar 24



Gambar 25

- Jika setting user telah selesai klik tombol Ok.

1.2 Mengelola Database

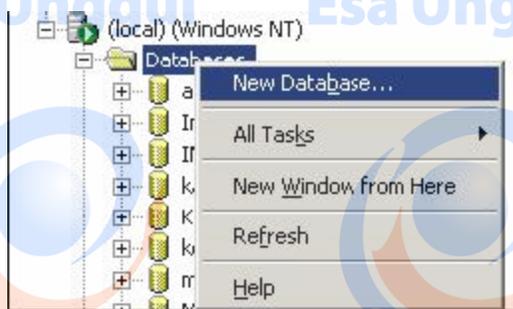
Salah satu yang paling penting dan menentukan dalam SQL Server adalah pengelolaan Database, sebab jika SQL server tidak dikelola dikhawatirkan nantinya akan tidak terasa

manfaat dan kelebihan database server SQL Server. Oleh sebab itu penting sekali kiranya Anda mempelajari dasar-dasar pengelolaan database.

Adapun langkah-langkah dasar pengelolaan database adalah sebagai berikut :

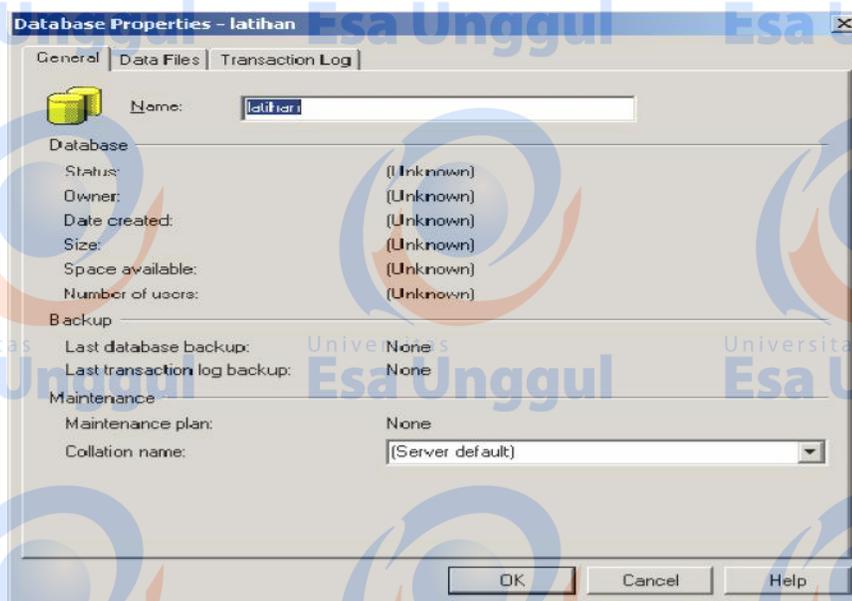
1.2.1 Membuat Database

- Untuk membuat database lakukan klik kanan pada databases seperti terlihat pada gambar 26

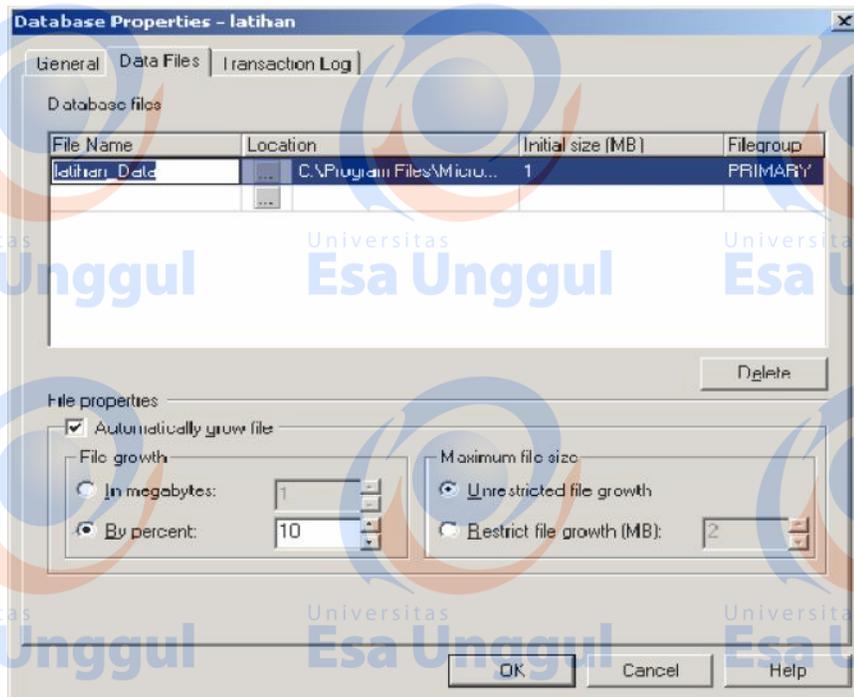


Gambar 26.

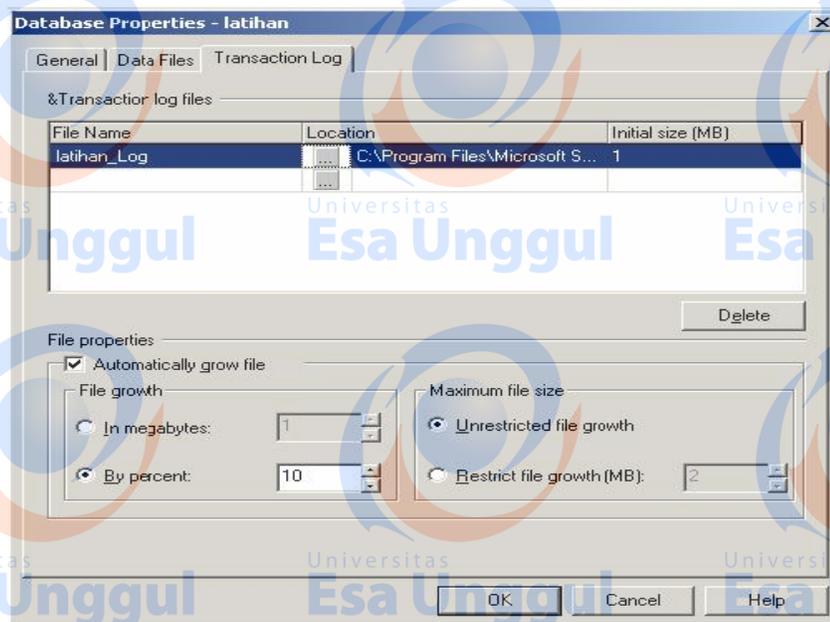
- Kemudian akan muncul jendela Database Property seperti yang nampak pada gambar 27, ketik nama database yang akan dibuat.
- Untuk mengubah letak posisi file database yang akan kita buat klik pada Data File, kemudian klik tombol untuk menentukan dimana database yang akan Anda buat disimpan. (gambar 28)
- Untuk mengubah letak posisi file log yang akan kita buat klik pada Transaction Log, kemudian klik tombol untuk menentukan dimana File Log yang akan Anda buat disimpan. (gambar 29)



Gambar 27



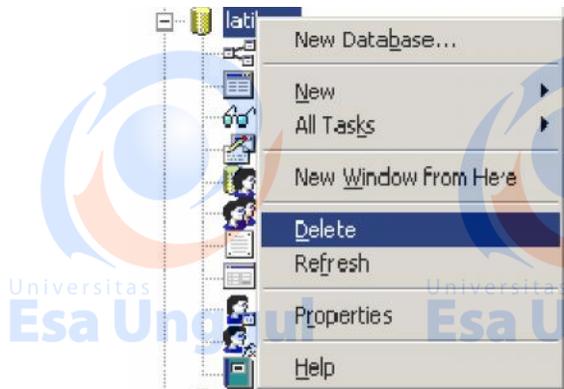
Gambar 28



Gambar 29

1.2.2 Menghapus Database

- Untuk menghapus database lakukan klik kanan pada database yang akan Anda hapus kemudian klik Delete



Gambar 30

- Kemudian Akan muncul Jendela pertanyaan (gambar 31) apakah database tersebut akan di hapus



Gambar 31

- Klik tombol yes jika database tersebut akan dihapus dan klik No jika database tersebut tidak jadi dihapus.

Pelaksanaan Praktikum

1. Mengetahui Interface SQL Server, menjalankannya dan pengelolaannya
2. Membuat User dan Database dengan menggunakan Enterprise manager

Tugas LAB

Buatlah user dan database dengan menggunakan Enterprise Manager dengan nama user = nama anda sendiri, kemudian buatlah Database dengan nama = DB_AKADEMIK

Pertemuan III

Tabel dan Diagram Database

Tujuan

Mahasiswa memahami cara membuat, memodifikasi dan menghapus Tabel dan bagaimana membuat relasi tabel dengan menggunakan Enterprise Manager.

Materi

1. Tipe Data
2. Pembuatan Tabel
3. Pengelolaan Tabel
4. Pembuatan Relasi Tabel

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

I. Tipe Data

Setiap kolom di dalam tabel harus memiliki Tipe Data tertentu. Dengan kata lain anda harus menentukan bahwa kolom tersebut akan menyimpan jenis data tertentu seperti misalnya teks, angka, data atau gambar.

Tipe data dalam SQL Server 2000 :

1. Binary : menyimpan deretan bit yang berisi bilangan hexadesimal

| Tipe Data | Ukuran Maksimal | Keterangan |
|-----------|-----------------|---------------------------|
| Binary | 8 Kb | Memiliki panjang yg tetap |
| Varbinary | 8 Kb | Panjang bervariasi |
| Image | 8 Kb | Data dpt bervariasi |

2. Character : berisi karakter alfanumerik dan symbol

| Tipe Data | Ukuran Maksimal | Keterangan |
|-----------|-----------------|-----------------------------|
| Char | 8 Kb | Memiliki panjang yg tetap |
| Varchar | 8 Kb | Panjang karakter bervariasi |

| | | |
|------|---|----------------------------|
| Text | $(2^{31})^{-1}$ atau 3.147.488.647 Byte | Data berupa karakter Ascii |
|------|---|----------------------------|

3. Date and Time : berisi kombinasi antara tanggal dan waktu

| Tipe Data | Ukuran Maksimal | Keterangan |
|---------------|-----------------|---|
| DateTime | 8 Byte | Dari tgl 1 januari 1753 sampai 31 desember 9999 |
| SmallDateTime | 4 Byte | Dari tgl 1 januari 1900 sampai 6 Juni 2079 |

4. Decimal : bilangan numeric yang disimpan dengan nilai tepat, tanpa pendekatan

| Tipe Data | Ukuran Maksimal | Keterangan |
|-----------|-----------------|---|
| Decimal | 38 digit | Semua digit dapat berupa digit disebelah kanan titik desimal. Data disimpan dengan representasi yg tepat tanpa pendekatan |
| Numeric | 38 Digit | Sama dengan desimal |

5. Floating Point : Bilangan numeric dengan nilai pendekatan

| Tipe Data | Ukuran Maksimal |
|-----------|-----------------------------|
| Float | -1.79E+308 sampai 1.79E+308 |
| Real | -3.40E+38 sampai 3.40E+38 |

6. Integer : Bilangan bulat negatif, nol dan positif

| Tipe Data | Ukuran Maksimal | Keterangan |
|-----------|-----------------|-------------------------------------|
| BigInt | 8 Byte | -2^{63} sampai 2^{63} |
| Int | 4 Byte | -2.147.483.648 sampai 2.147.483.648 |
| SmallInt | 2 Byte | -32.768 sampai 32.768 |
| TinyInt | 1 Byte | 0 sampai 32.767 |

7. Monetary : jumlah uang dalam bentuk positif dan negatif

| Tipe Data | Ukuran Maksimal | Keterangan |
|------------|-----------------|---|
| Money | 8 Byte | -922.337.203.685.477.5808 sampai 922.337.203.685.477.5807 |
| SmallMoney | 4 Byte | -214.748.3648 sampai 214.748.3648 |

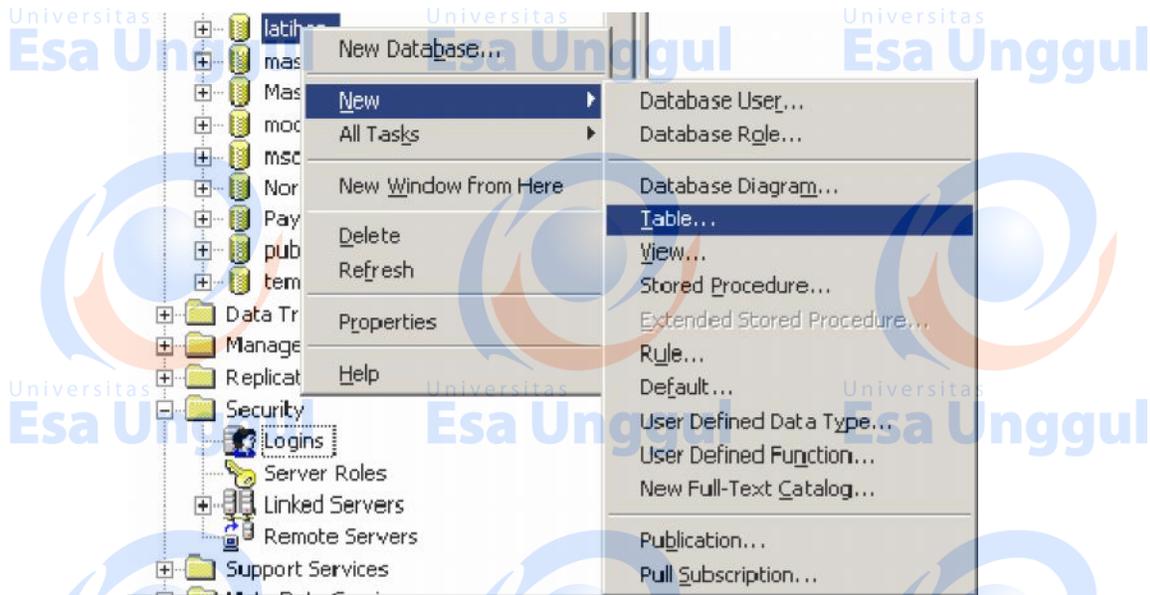
II. Tabel

2.1 Membuat Tabel

Tabel adalah bagian komponen dari database yang berfungsi untuk menyimpan data biasanya tabel terdiri dari baris dan kolom, baris dalam hal ini adalah sebagai record dan kolom sebagai field.

Cara Membuat Tabel dengan Enterprise Manager

- Klik kanan pada nama database yang akan dibuat tabelnya kemudian pilih Table seperti terlihat pada gambar 32.



Gambar 32

- Kemudian akan muncul jendela seperti gambar 33

Column Name adalah kolom untuk nama field

Data Type adalah kolom untuk tipe data dari field

Length adalah kolom untuk ukuran dari tipe data

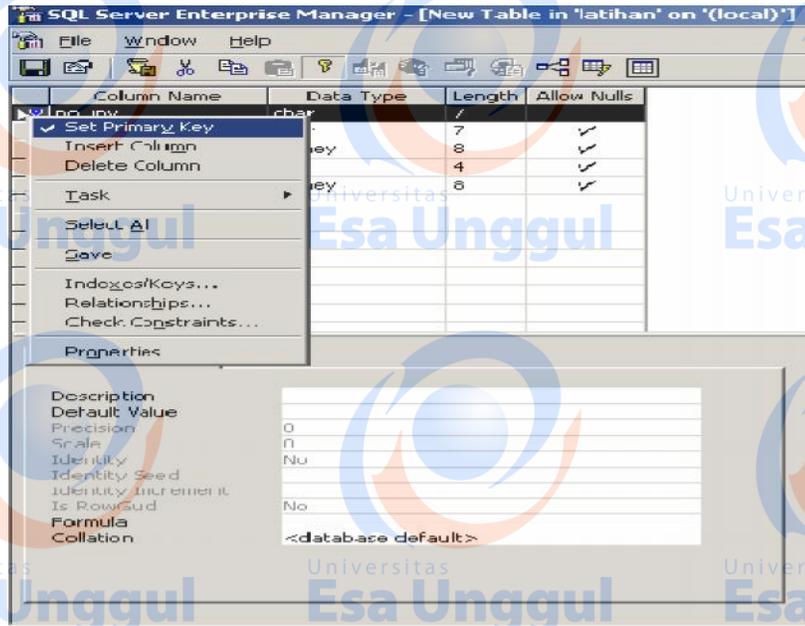
Allow Nulls adalah kolom untuk membolehkan null di field atau tidak

| Column Name | Data Type | Length | Allow Nulls |
|-------------|-----------|--------|-------------|
| no_inv | char | 7 | |
| kd_brg | char | 7 | ✓ |
| harga | money | 8 | ✓ |
| qty | int | 4 | ✓ |
| tot_hrg | money | 8 | ✓ |

Gambar 33

- Dalam sebuah tabel sebaiknya di buat Primary Key yang merupakan identitas unik dari setiap baris data. Untuk membuat Primary Key klik kanan pada field yang akan dijadikan

sebagai field kunci, misalnya dalam hal ini field kuncinya adalah no_inv kemudian klik Primari Key

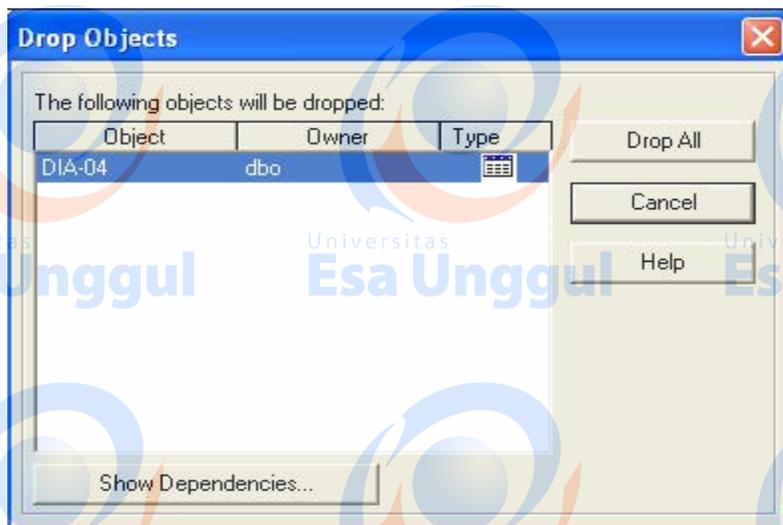


Gambar 34

- Untuk membuat field autonumber isi **Identity** dengan Yes, **Identity Seed** (nomor awal) diisi dengan 1 dan **Identity Increment** diisi dengan 1

2.2 Menghapus Tabel

Untuk menghapus tabel lakukan klik kanan pada tabel yang akan dihapus kemudian pilih Delete akan muncul jendela seperti gambar 35, klik tombol Drop All untuk menghapus tabel atau klik Cancel untuk membatalkan penghapusan tabel.



Gambar 35

III. Diagrams/Relasi Tabel

Diagrams adalah fasilitas untuk mengelola relasi antar tabel secara virtual. Relasi antar tabel sangat dibutuhkan untuk menjaga integritas dan konsistensi data antar tabel. Misalnya jangan sampai ada transaksi di tabel penjualan padahal kode barangnya belum dibuat.

Membuat Diagrams

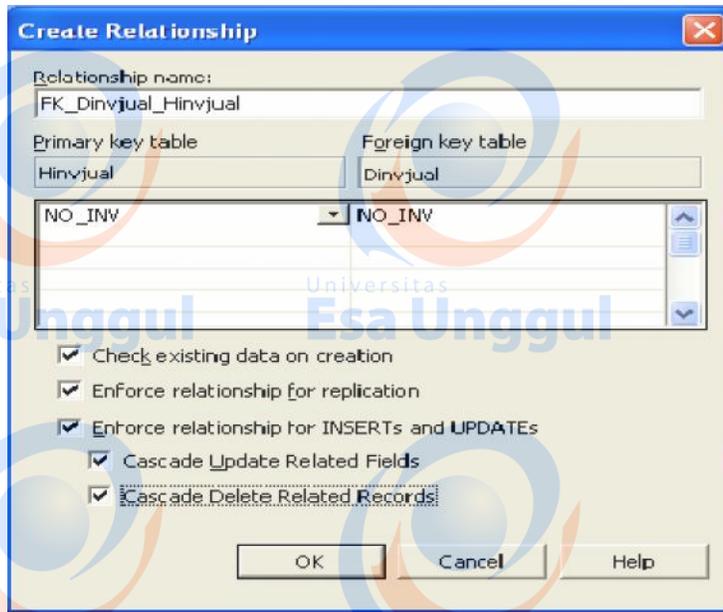
Tahap- tahap membuat Diagram

1. Untuk membuat Diagrams klik kanan pada Diagrams, kemudian pilih New Database Diagrams maka akan muncul tampilan seperti gambar 36. Jika Anda lebih suka membuat Diagrams dengan wizard tekan tombol Next namun dalam hal ini kita akan membuat Diagrams atau Relasi antar tabel tanpa bantuan wizard maka klik tombol Cancel maka akan muncul jendela kosong tempat pembuatan diagrams.



Gambar 36

2. Untuk memasukkan tabel-tabel yang akan di relasikan di dalam diagram tersebut lakukan klik kanan pada jendela tersebut dan pilih Add Table kemudian pilih tabel-tabel mana saja yang akan dibuat relasinya dalam hal ini Hinvjual dan Dinvjual dimana Hinvjual merupakan header dari transaksi penjualan dan Dinvjual merupakan data detil dari transaksi penjualan. Klik tombol Add untuk memasukkan tabel tersebut kedalam jendela diagrams.
3. Untuk melakukan relasi antar tabel klik pada field no_inv tabel Hinvjual kemudian drag ke field no_inv tabel Dinvjual maka akan muncul jendela seperti tampil pada gambar 37

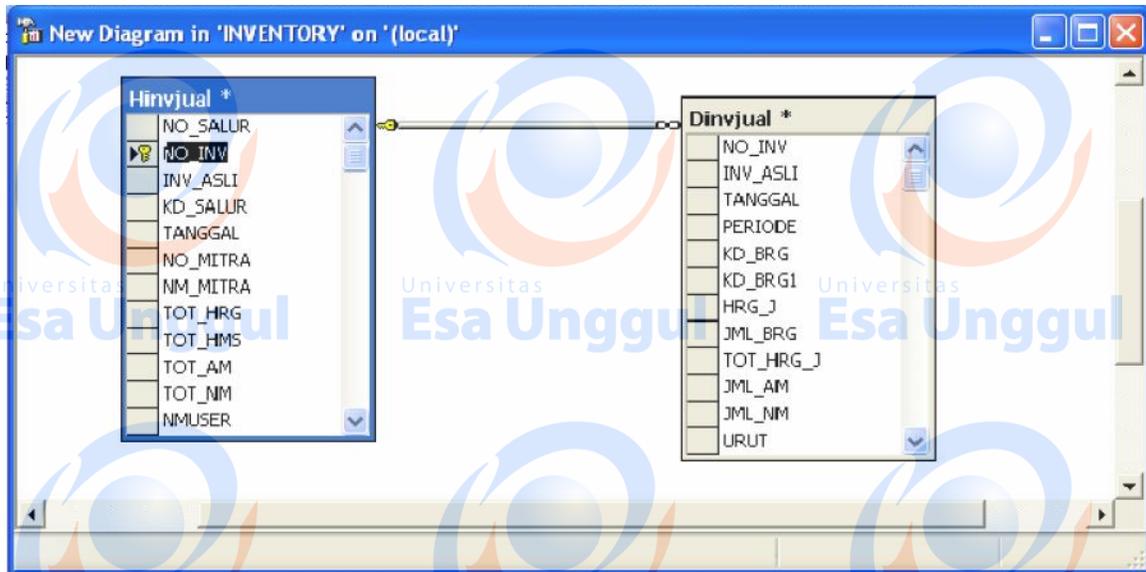


Gambar 37

Pada gambar 37 ada beberapa option yang fungsinya sebagai berikut:

- Check existing data on creation : Konsistensi data akan langsung dicek pada saat pembuatan relationship.
- Enforce relationship for replication : Pada saat replikasi antar database, pengcopian data akan dilakukan berdasarkan relasi antar tabel.
- Enforce relationship for INSERT s and UPDATES : Setiap ada proses perubahan atau penambahan data akan langsung dicek relasi antar tabelnya.
- Cascade Update Related Fields : Apabila terjadi perubahan data di tabel master maka otomatis di tabel relasinya akan langsung disesuaikan dengan tabel masternya.
- Cascade Delete Related Records : Apabila terjadi penghapusan data di tabel master maka data pada tabel relasinya otomatis akan dihapus juga.
- Jika pemilihan option telah selesai, klik tombol ok dan kemudian hasilnya akan tampak seperti gambar 38.

U4:versit Klik tombol untuk menyimpan diagrams dan kemudian isi nama diagramnya.



Gambar 38.

3.2 Menghapus Diagram

Untuk menghapus Diagram Klik kanan pada nama diagram yang akan dihapus kemudian pilih Delete.

Pelaksanaan Praktikum

1. Pembuatan tabel dan membuat Relasi Tabel

Tugas.

1. Apa pengertian konsep primary key dan foreign key.
2. Buatlah system database, nama database =akademik, yang memuat minimal 3 tabel, Matakuliah, jadwal dan dosen, dan buatlah relasi antar tabelnya Dengan enterprise manager.

Tabel Matakuliah

| Field | Tipe data | Allow null | keterangan |
|----------------|-------------|------------|-----------------------------|
| Kode_mk | Char(5) | No | Kode matakuliah/primary key |
| Nama_mk | Varchar(25) | No | Nama Matakuliah |
| Sks | Numeric | Yes | sks |
| Kode_prasyarat | Char(5) | Yes | Kode matakuliah prasyarat |

Tabel Dosen

| Field | Tipe data | Allow null | keterangan |
|----------|-------------|------------|------------------------|
| Kode_dsn | Char(5) | No | Kode dosen/primary key |
| Nama_dsn | Varchar(25) | No | Nama dosen |
| Jk | Char (5) | Yes | Jenis kelamin (L/P) |
| Agama | Varchar(25) | Yes | agama |

| | | | |
|------------|-------------|-----|----------|
| Nama_jalan | Varchar(50) | Yes | alamat |
| Kode_pos | Varchar(10) | Yes | Kode pos |

Tabel Jadwal

| Field | Tipe data | Allow null | keterangan |
|----------|-------------|------------|------------------------------|
| Kode_mk | Char(5) | No | Kode matakuliah /Foreign key |
| Kode_dsn | Char(5) | No | Kode dosen/Foreign Key |
| Hari | Varchar(15) | Yes | hari |
| Jam | dateTime | Yes | jam |
| Kelas | Varchar(10) | Yes | Nama kelas |



Pertemuan IV Perintah SQL

Tujuan

Mahasiswa memahami Perintah sql dan menerapkannya di Database SQL Server 2000

Materi

1. Konsep DDL
2. Konsep DML

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

SQL COMMAND

SQL singkatan dari **Structured Query Language**. SQL adalah bahasa komputer yang standart untuk mengakses dan memanipulasi database. Seluruh aplikasi database yang beredar di pasaran, baik yang gratis maupun yang berlisensi, mengadopsi bahasa SQL untuk mengolah databasenya.

Beberapa SQL command atau perintah SQL yang harus diketahui adalah :

- CREATE DATABASE, untuk membuat sebuah database.
- DROP DATABASE, untuk menghapus sebuah database.
- CREATE TABLE, untuk membuat sebuah table.
- ALTER TABLE, untuk memodifikasi sebuah table.
- DROP TABLE, untuk menghapus sebuah table.
- SELECT, untuk menampilkan data dari database.
- UPDATE, untuk memodifikasi data dari database.
- INSERT INTO, untuk menambah data di database.
- DELETE, untuk menghapus data dari database.

Sedangkan perintah SQL untuk tingkat lanjut ada cukup banyak. Command tingkat lanjut ini berguna untuk mengolah data dengan cara yang lebih rumit dan kompleks.

CREATE DATABASE COMMAND

Sebelum bekerja dengan database, anda harus membuat dulu sebuah database yang akan menjadi obyek pekerjaan anda.

Syntaxnya :

```
CREATE DATABASE database_name
```

Setelah membuat database dengan command seperti di atas, tidak secara otomatis database anda berisi tabel-tabel. Database anda dalam keadaan kosong. Anda harus membuat tabel-tabel seperti yang akan dijelaskan pada bagian selanjutnya.

DROP DATABASE COMMAND

Jika anda ingin menghapus sebuah database, gunakan perintah berikut ini :

```
DROP DATABASE database_name
```

Jika anda menggunakan perintah di atas, maka seluruh database dan tabel yang ada di dalamnya akan terhapus.

Catatan : untuk Access, anda tidak diperbolehkan menggunakan perintah DROP DATABASE.

Jika anda menggunakan perintah ini, aplikasi anda akan kacau. Jika anda ingin menghapus database dari Access, anda bisa menghapus file **mdb** yang telah dibuat oleh Access.

CREATE TABLE COMMAND

Selanjutnya anda dapat membuat table dengan perintah berikut :

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
.....
)
```

Contohnya :

```
CREATE TABLE Person
(
LastName varchar,
FirstName varchar,
Address varchar,
Age int
)
```

Anda juga bisa membatasi ukuran masing-masing field untuk menghasilkan file database yang lebih kecil :

```
CREATE TABLE Person
(
LastName varchar(30),
FirstName varchar,
Address varchar,
Age int(3)
)
```

Untuk tipe datanya, anda dapat melihat lagi di bagian pertama materi kuliah ini.

ALTER TABLE COMMAND

Perintah ALTER TABLE dapat anda gunakan jika anda ingin memodifikasi struktur table yang anda buat. Artinya, jika anda ingin menambahkan sebuah atau beberapa field baru dalam tabel yang sudah anda buat, atau anda ingin menghapus satu atau beberapa field yang sudah anda buat, anda dapat menggunakan perintah ini.

Syntaxnya :

Untuk menambah field baru :

```
ALTER TABLE table_name  
ADD column_name datatype (size)
```

Untuk menghapus field yang sudah ada :

```
ALTER TABLE table_name  
DROP COLUMN column_name
```

DROP TABLE COMMAND

Untuk menghapus sebuah table, perintahnya seperti berikut ini :

```
DROP TABLE table_name
```

Catatan : Perlu diingat, bahwa jika anda menghapus sebuah table, maka seluruh data yang ada di dalamnya akan ikut terhapus.

SELECT COMMAND

Untuk menampilkan data dari sebuah database, anda dapat menggunakan perintah SELECT. Bentuk baku dari perintah ini adalah :

```
SELECT column_name_1, column_name_2, dst
FROM table_name
```

Jika anda ingin menampilkan semua data dalam sebuah table tanpa memilih salah satu field, maka anda dapat menggunakan perintah berikut :

```
SELECT * FROM table_name
```

SELECT DISTINCT STATEMENT

SELECT DISTINCT Statement digunakan untuk menampilkan data dari satu atau beberapa field (column) dari sebuah tabel. Namun perbedaan antara SELECT dengan SELECT DISTINCT terletak pada hasilnya.

Perhatikan contoh table **ORDERS** berikut ini :

| Company | OrderNumber |
|-----------|-------------|
| Sega | 3412 |
| W3Schools | 2312 |
| Trio | 4678 |
| W3Schools | 6798 |

Jika anda menggunakan perintah SELECT seperti ini :

```
SELECT Company FROM Orders
```

maka hasil yang didapat akan seperti ini :

| Company |
|-----------|
| Sega |
| W3Schools |

| |
|-----------|
| Trio |
| W3Schools |

Tapi jika anda menggunakan perintah SELECT DISTINCT seperti ini :

```
SELECT DISTINCT Company FROM Orders
```

Maka hasilnya akan seperti ini :

| Company |
|-----------|
| Sega |
| W3Schools |
| Trio |

Perhatikan bahwa dengan perintah SELECT, data "W3Schools" akan ditampilkan 2 kali sesuai data yang ada, sedangkan dengan perintah SELECT DISTINCT, maka data "W3Schools" hanya ditampilkan 1 kali. Jadi perintah SELECT DISTINCT tidak akan menghasilkan data yang identik atau kembar.

WHERE CLAUSE

Anda juga dapat menambahkan sebuah kondisi pada perintah SELECT untuk menampilkan data yang lebih spesifik.

Syntax baku dari perintah WHERE adalah :

```
SELECT column FROM table  
WHERE column operator value
```

di mana operator yang dimaksud adalah :

| Operator | Description |
|------------|----------------------------|
| = | Equal |
| <> atau != | Not equal |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal |
| <= | Less than or equal |
| BETWEEN | Between an inclusive range |
| LIKE | Search for a pattern |

Perhatikan table **PERSONS** di bawah ini :

| LastName | FirstName | Address | City | Year |
|-----------|-----------|--------------|-----------|------|
| Hansen | Ola | Timoteivn 10 | Sandnes | 1951 |
| Svendson | Tove | Borgvn 23 | Sandnes | 1978 |
| Svendson | Stale | Kaivn 18 | Sandnes | 1980 |
| Pettersen | Kari | Storgt 20 | Stavanger | 1960 |

Jika anda menggunakan perintah SELECT dengan atribut WHERE seperti berikut ini :

```
SELECT * FROM Persons WHERE City='Sandnes'
```

Maka hasilnya akan ditampilkan seperti ini :

| LastName | FirstName | Address | City | Year |
|----------|-----------|--------------|---------|------|
| Hansen | Ola | Timoteivn 10 | Sandnes | 1951 |
| Svendson | Tove | Borgvn 23 | Sandnes | 1978 |
| Svendson | Stale | Kaivn 18 | Sandnes | 1980 |

Perhatikan, bahwa yang ditampilkan adalah data yang berasal dari kota "Sandnes" sesuai dengan kondisi yang diberikan pada perintah di atas. Perhatikan untuk tulisan tanda petik (") yang digunakan, berikut ini penulisan yang benar:

Untuk nilai berupa text :

Syntax yang benar :

```
SELECT * FROM Persons WHERE FirstName='Tove'
```

Syntax yang salah :

```
SELECT * FROM Persons WHERE FirstName=Tove
```

Untuk nilai berupa angka :

Syntax yang benar :

```
SELECT * FROM Persons WHERE Year>1965
```

Syntax yang salah :

```
SELECT * FROM Persons WHERE Year>'1965'
```

LIKE CLAUSE

Statemen LIKE dapat anda tambahkan pada perintah SELECT untuk memberikan suatu kondisi yang menyerupai sesuatu.

Syntax bakunya adalah :

```
SELECT column FROM table_name  
WHERE column LIKE conditions
```

Dalam perintah ini, digunakan tanda “%” untuk menunjukkan sesuatu yang belum diketahui.

Perhatikan contoh berikut ini :

```
SELECT * FROM Persons  
WHERE FirstName LIKE 'O%'
```

Perintah di atas ini akan menampilkan semua data yang FirstName-nya berawalan dengan huruf “O”.

```
SELECT * FROM Persons
WHERE FirstName LIKE '%a'
```

Sedangkan contoh di atas akan menghasilkan semua data yang FirstName-nya berakhiran dengan huruf "a".

```
SELECT * FROM Persons
WHERE FirstName LIKE '%la%'
```

Sedangkan perintah di atas ini akan menghasilkan semua data yang FirstName-nya memiliki huruf "la" di tengahnya.

ORDER BY STATEMENT

Statement ORDER BY digunakan untuk mengurutkan tampilan data berdasarkan satu atau beberapa field secara urut dari awal ke akhir (ascending) atau dari akhir ke awal (descending).

Syntaxnya adalah sebagai berikut :

```
SELECT column_1, column_2, dst FROM table_name
ORDER BY column_1, column_2, dst ASC/DESC
```

Atau anda juga dapat menggabungkannya dengan clause yang lain, seperti WHERE dan LIKE seperti berikut ini :

```
SELECT column_1, column_2, dst FROM table
WHERE column_2 LIKE conditions ORDER BY column_1 ASC
```

Perhatikan contoh table **ORDERS** berikut ini :

| Company | OrderNumber |
|----------|-------------|
| Sega | 3412 |
| ABC Shop | 5678 |

| | |
|-----------|------|
| W3Schools | 2312 |
| W3Schools | 6798 |

Misalnya anda punya perintah seperti ini :

```
SELECT Company, OrderNumber FROM Orders
ORDER BY Company
```

Maka hasilnya akan seperti ini :

| Company | OrderNumber |
|-----------|-------------|
| ABC Shop | 5678 |
| Sega | 3412 |
| W3Schools | 6798 |
| W3Schools | 2312 |

Perhatikan, bahwa jika anda tidak memberikan tanda **ASC** atau **DESC**, maka hasilnya akan ditampilkan secara **ASC** (urut dari awal ke akhir, atau dari kecil ke besar).

Anda juga dapat menggabungkan beberapa kondisi sekaligus. Perhatikan contoh berikut ini :

```
SELECT Company, OrderNumber FROM Orders
ORDER BY Company DESC, OrderNumber ASC
```

Hasilnya akan seperti ini :

| Company | OrderNumber |
|-----------|-------------|
| W3Schools | 2312 |
| W3Schools | 6798 |
| Sega | 3412 |
| ABC Shop | 5678 |

Perhatikan bahwa hasilnya menunjukkan bahwa di field company akan urut secara descending, dan di field ordernumber akan urut secara ascending.

AND dan OR STATEMENT

Statement AND atau OR dapat digunakan untuk menggabungkan 2 kondisi yang berbeda untuk dijalankan dalam perintah WHERE.

Perhatikan contoh table **PERSONS** berikut ini :

| LastName | FirstName | Address | City |
|----------|-----------|--------------|---------|
| Hansen | Ola | Timoteivn 10 | Sandnes |
| Svendson | Tove | Borgvn 23 | Sandnes |
| Svendson | Stephen | Kaivn 18 | Sandnes |

Misalnya anda mempunyai perintah seperti ini :

```
SELECT * FROM Persons  
WHERE FirstName='Tove'  
AND LastName='Svendson'
```

Maka hasilnya akan seperti ini :

| LastName | FirstName | Address | City |
|----------|-----------|-----------|---------|
| Svendson | Tove | Borgvn 23 | Sandnes |

Atau anda punya contoh perintah seperti ini :

```
SELECT * FROM Persons  
WHERE firstname='Tove'  
OR lastname='Svendson'
```

Maka hasil yang akan ditampilkan adalah seperti ini :

| LastName | FirstName | Address | City |
|----------|-----------|-----------|---------|
| Svendson | Tove | Borgvn 23 | Sandnes |

| | | | |
|----------|---------|----------|---------|
| Svendson | Stephen | Kaivn 18 | Sandnes |
|----------|---------|----------|---------|

Anda juga dapat menggabungkan statement AND dan OR secara bersamaan, misalnya :

```
SELECT * FROM Persons WHERE
(FirstName='Tove' OR FirstName='Stephen')
AND LastName='Svendson'
```

Maka hasilnya akan seperti ini :

| LastName | FirstName | Address | City |
|----------|-----------|-----------|---------|
| Svendson | Tove | Borgvn 23 | Sandnes |
| Svendson | Stephen | Kaivn 18 | Sandnes |

UPDATE COMMAND

Perintah UPDATE digunakan untuk memodifikasi data (record) yang ada dalam sebuah table.

Syntaxnya :

```
UPDATE table_name
SET column_name = new_value
WHERE column_name = old_value
```

Perhatikan contoh table **PERSONS** berikut ini :

| LastName | FirstName | Address | City |
|-----------|-----------|------------|-----------|
| Nilsen | Fred | Kirkegt 56 | Stavanger |
| Rasmussen | Ani | Storgt 67 | |

Misalnya anda punya perintah seperti ini :

```
UPDATE Person SET FirstName = 'Nina'
```

```
WHERE LastName = 'Rasmussen'
```

Maka hasilnya adalah seperti ini :

| LastName | FirstName | Address | City |
|-----------|-----------|------------|-----------|
| Nilsen | Fred | Kirkegt 56 | Stavanger |
| Rasmussen | Nina | Storgt 67 | |

Perhatikan bahwa record yang sebelumnya adalah "Ani" berubah menjadi "Nina" setelah perintah dijalankan.

INSERT INTO COMMAND

Perintah INSERT INTO digunakan untuk menambahkan data baru pada sebuah table, bukan untuk memodifikasi data dalam salah satu field saja.

Syntaxnya :

```
INSERT INTO table_name (column1, column2, dst)
VALUES (column1_value, column2_value, dst)
```

Perhatikan table **PERSONS** di bawah ini :

| LastName | FirstName | Address | City |
|-----------|-----------|-----------|-----------|
| Pettersen | Kari | Storgt 20 | Stavanger |

Untuk menambahkan data yang baru, anda dapat menggunakan perintah INSERT INTO. Berikut contohnya :

```
INSERT INTO Persons (LastName, FirstName, Address, City)
VALUES ('Hetland', 'Camilla', 'Hagabakka 24', 'Sandnes')
```

Atau perintah seperti ini :

```
INSERT INTO Persons
VALUES ('Hetland', 'Camilla', 'Hagabakka 24', 'Sandnes')
```

Perhatikan bahwa karena isi field berupa text, maka value yang ada diberi tanda petik (' ') yang menandakan text atau string. Hasilnya adalah sebagai berikut :

| LastName | FirstName | Address | City |
|-----------|-----------|--------------|-----------|
| Pettersen | Kari | Storgt 20 | Stavanger |
| Hetland | Camilla | Hagabakka 24 | Sandnes |

Anda juga dapat menambahkan data pada field tertentu saja, namun tetap membuat baris yang baru (bukan mengganti data yang lama).

Perhatikan table **PERSONS** berikut ini :

| LastName | FirstName | Address | City |
|-----------|-----------|--------------|-----------|
| Pettersen | Kari | Storgt 20 | Stavanger |
| Hetland | Camilla | Hagabakka 24 | Sandnes |

Misalnya anda punya perintah seperti ini :

```
INSERT INTO Persons (LastName, Address)
VALUES ('Rasmussen', 'Storgt 67')
```

Maka hasilnya adalah :

| LastName | FirstName | Address | City |
|-----------|-----------|--------------|-----------|
| Pettersen | Kari | Storgt 20 | Stavanger |
| Hetland | Camilla | Hagabakka 24 | Sandnes |
| Rasmussen | | Storgt 67 | |

Perhatikan bahwa data yang baru tidak menggantikan posisi data yang lama, melainkan menempati baris yang baru dengan 2 field yang dikosongkan isinya.

DELETE COMMAND

Perintah DELETE digunakan untuk menghapus data dari sebuah table. Fungsi menghapus ini akan dijalankan berdasarkan kondisi yang diberikan. Karena itu perlu anda ingat untuk selalu memberikan sebuah kondisi agar data yang terhapus benar-benar data yang ingin anda hapus.

Perintah ini tidak bisa menghapus hanya isi field tertentu saja. Seluruh field dalam satu rangkaian baris akan dihapus jika kondisi yang ditetapkan terpenuhi.

Syntaxnya :

```
DELETE FROM table_name  
WHERE column_name = value
```

Perhatikan contoh table **PERSONS** berikut ini :

| LastName | FirstName | Address | City |
|-----------|-----------|------------|-----------|
| Nilsen | Fred | Kirkegt 56 | Stavanger |
| Rasmussen | Nina | Stien 12 | Stavanger |

Misalnya anda punya perintah seperti ini :

```
DELETE FROM Person WHERE LastName = 'Rasmussen'
```

Maka hasilnya akan seperti ini :

| LastName | FirstName | Address | City |
|----------|-----------|------------|-----------|
| Nilsen | Fred | Kirkegt 56 | Stavanger |

Perhatikan bahwa seluruh data milik Rasmussen akan dihapus.

Jika anda ingin menghapus semua baris yang ada, anda dapat menggunakan perintah berikut ini :

```
DELETE FROM table_name
atau
DELETE * FROM table_name
```

Maka seluruh data dalam table itu akan terhapus semuanya.

Pelaksanaan Praktikum

Mempelajari sintak SQL mulai dari pembuatan database, penghapusan database, dan pembuatan table serta perintah manipulasi table baik itu insert, update, delete, dan select

Latihan DDL

1. Buatlah database dengan nama_DB = RUMAH SAKIT
2. Buatlah table table dengan menggunakan sintaks SQL di Query Analyzer, adapun table yang terdapat didalamnya adalah sebagai berikut :

Tabel Pemeriksaan

| No | Nama column | Type data | length | keterangan |
|----|-----------------|-----------|--------|-------------|
| 1 | No_pendaftaran | Char | 5 | Primary key |
| 2 | No_pasien | Char | 5 | Foreign key |
| 3 | Tgl_pendaftaran | Datetime | | |
| 4 | Tgl_periksa | datetime | | |

Tabel Dokter

| No | Nama column | Type data | length | keterangan |
|----|-------------|-----------|--------|-------------|
| 1 | Kode_dokter | Char | 5 | Primary key |
| 2 | Nama_dokter | Varchar | 25 | |

Tabel Pasien

| No | Nama column | Type data | length | keterangan |
|----|-------------|-----------|--------|-------------|
| 1 | No_pasien | Char | 5 | Primary key |
| 2 | Nama_pasien | Varchar | 25 | |
| 3 | Alamat | Varchar | 50 | |
| 4 | Telepon | Varchar | 12 | |

Tabel Resep

| No | Nama column | Type data | length | keterangan |
|----|----------------|-----------|--------|-------------|
| 1 | No_pendaftaran | Char | 5 | Foreign key |
| 2 | Kode_dokter | Char | 5 | Foreign key |
| 3 | Kode_sakit | Char | 5 | Foreign key |
| 4 | Kode_obat | Char | 5 | Foreign key |
| 5 | Dosis | Varchar | 25 | |

Tabel Obat

| No | Nama column | Type data | length | keterangan |
|----|-------------|-----------|--------|-------------|
| 1 | Kode_obat | Char | 5 | Primary key |
| 2 | Nama_obat | Varchar | 25 | |

Tabel Penyakit

| No | Nama column | Type data | length | keterangan |
|----|---------------|-----------|--------|-------------|
| 1 | Kode_sakit | Char | 5 | Primary key |
| 2 | Nama_penyakit | Varchar | 25 | |

3. Buatlah Relasi tabel dari tabel di atas

TUGAS DML

Buatlah database baru dengan nama = DB_BANK , kemudian buat tabel-tabel di bawah ini serta inputkan isi tabelnya dengan menggunakan perintah SQL.

Diketahui 2 tabel (nasabah dan cabang) beserta isi tabelnya,

Tabel Nasabah

| No | Nama column | Type data | length | keterangan |
|----|-------------|-----------|--------|-------------|
| 1 | Kd_nasabah | Char | 5 | Primary key |
| 2 | Nama | Varchar | 25 | |
| 3 | Alamat | Varchar | 50 | |
| 4 | Rekening | Varchar | 10 | |
| 5 | Kd_cabang | Char | 5 | Foreign key |
| 6 | Saldo | Float | | |

Tabel Cabang

| No | Nama column | Type data | length | keterangan |
|----|-------------|-----------|----------|-------------|
| 1 | Kd_cabang | Char | 5 | Primary key |
| 2 | Nama | Varchar | 25 | |
| 3 | Alamat | Varchar | 50 | |
| 4 | Jml_nasabah | Numeric | Not null | |

Inputan data seperti dlm tabel di bawah ini

DATA CABANG

| Kd_cabang | Nama | Alamat | Jml_nasabah |
|-----------|---------------|---------------|-------------|
| KC001 | KC AMIKOM | CONDONG CATUR | 8 |
| KC002 | KC JANTI | JANTI | 5 |
| KC003 | KC GEJAYAN | GEJAYAN | 6 |
| KC004 | KC WIROBRAJAN | WIROBRAJAN | 8 |
| KC005 | KC KAUMAN | KAUMAN | 7 |
| KC006 | KC IMOGIRI | IMOGIRI | 6 |
| KC007 | KC BABARSARI | BABARSARI | 3 |

DATA NASABAH

| Kd_nasabah | Nama | Alamat | Rekening | Kd_cabang | saldo |
|------------|--------|---------|----------|-----------|--------|
| N0001 | JANUAR | KALASAN | 12340001 | KC002 | 150000 |
| N0002 | FEBRI | KUNCEN | 12340002 | KC004 | 250000 |

| | | | | | |
|-------|----------|-----------|----------|-------|--------|
| N0003 | Maria | paris | 12340003 | KC006 | 300000 |
| N0004 | Aprillia | samirono | 12340004 | KC003 | 200000 |
| N0005 | Memey | seyegan | 12340005 | KC001 | 120000 |
| N0006 | Juno | gamping | 12340006 | KC004 | 280000 |
| N0007 | Julia | monjali | 12340007 | KC001 | 140000 |
| N0008 | Agusto | turi | 12340008 | KC001 | 220000 |
| N0009 | Septia | ngampilan | 12340009 | KC005 | 130000 |
| N0010 | Okta | sagan | 12340010 | KC003 | 270000 |
| N0011 | Nova | kotabaru | 12340011 | KC005 | 200000 |
| N0012 | Desi | terban | 12340012 | KC003 | 70000 |

Soal

1. Buatlah perintah untuk menampilkan semua nama nasabah, alamat dan saldonya
2. Buatlah perintah untuk menampilkan nama nasabah dengan saldo > 200000
3. Buatlah perintah untuk menampilkan nama nasabah dengan alamat gejayan
4. Buatlah perintah untuk menampilkan nama nasabah dengan kode cabang KC003
5. Buatlah perintah untuk menampilkan semua data kantor cabang
6. Buatlah perintah untuk menampilkan nama kantor cabang dengan alamat kauman
7. Buatlah perintah untuk menampilkan nama kantor cabang dengan jumlah nasabah > 6 orang
8. Buatlah perintah untuk mengupdate data kantor cabang dengan kode KC002 menjadi
Nama = KC RSJ pakem
Alamat = pakem
Jumlah nasabah = 7
9. Buatlah perintah untuk menghapus data semua kantor cabang dengan jumlah nasabah kurang dari 5
10. Buatlah perintah untuk mengupdate data nasabah dengan kode N0010 menjadi
Alamat = monjali
Saldo = 400000
11. Buatlah perintah untuk menghapus semua data nasabah dengan alamat terban





Pertemuan V

FUNGSI-FUNGSI DALAM

SQL SERVER 2000

Tujuan

Mahasiswa memahami fungsi– fungsi apa saja yang digunakan di SQL Server 2000.

Materi

1. Fungsi-fungsi Agregasi
2. Grouping
3. Fungsi Waktu
4. Fungsi Aritmatika
5. Fungsi Konversi
6. Fungsi Karakter

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

I. Fungsi Agregasi

Fungsi agregate sering digunakan untuk operasi perhitungan dari sebuah group data. Yang paling sering digunakan adalah Avg, SUM, MAX, MIN, DAN COUNT

1. Fungsi AVG

Perintah ini digunakan untuk mencari nilai rata-rata dari suatu nilai pada sebuah field dalam sebuah tabel yang telah didefinisikan dalam perintah SELECT

Sintaks : `SELECT AVG (NAMA_KOLOM) FROM NAMA_TABEL`

2. Fungsi SUM

Perintah ini digunakan untuk mencari hasil jumlah dari suatu nilai pada suatu field atau kolom tertentu dalam sebuah tabel yang telah didefinisikan dalam perintah SELECT

Sintaks : `SELECT SUM (NAMA_KOLOM) FROM NAMA_TABEL`

3. Fungsi MAX

Perintah ini digunakan untuk mencari nilai maksimum atau nilai terbesar pada suatu field atau kolom tertentu dalam sebuah tabel yang telah didefinisikan dalam perintah SELECT

Sintaks : `SELECT MAX (NAMA_KOLOM) FROM NAMA_TABEL`

4. Fungsi MIN

Perintah ini digunakan untuk mencari nilai minimum atau nilai terkecil pada suatu field atau kolom tertentu dalam sebuah tabel yang telah didefinisikan dalam perintah SELECT

Sintaks : `SELECT MIN (NAMA_KOLOM) FROM NAMA_TABEL`

5. Fungsi COUNT

Perintah ini digunakan untuk menghitung banyaknya record atau data dalam sebuah kolom pada sebuah tabel yang telah didefinisikan dalam perintah SELECT

Sintaks : `SELECT COUNT (NAMA_KOLOM) FROM NAMA_TABEL`

II. Grouping

Klausa GROUP BY digunakan untuk mengelompokkan data berdasarkan daftar kolom.

Contoh : menampilkan daftar asal daerah yang terdapat pada tabel atlet :

`SELECT ASAL_DAERAH FROM ATLET GROUP BY ASAL_DAERAH`

III. Fungsi Waktu

SQL Server menyediakan fungsi waktu untuk menghasilkan aritmatika tanggal.

1. Fungsi GETDATE

Digunakan untuk menampilkan tanggal sekarang

Sintaks : `SELECT GETDATE() [waktu saat ini]`

2. Fungsi DATEADD

Dengan menggunakan fungsi DATEADD kita dapat menambah jumlah tanggal, bulan dan tahun dari suatu tanggal.

Sintaks : menambah 1 bulan

`SELECT *,DATEADD (M,1,TGL) AS GAJI_LAGI FROM PENGAJIAN`

Sintaks menambah 31 hari

`SELECT *, DATEADD (D,31,TGL) AS GAJIAN_LAGI FROM PENGAJIAN`

Sintaks : menambah 12 bulan

`SELECT *, DATEADD (M,12,TGL) AS GAJIAN_THN_DEPAN FROM PENGAJIAN`

Sintaks : menambah 1 tahun

`SELECT *, DATEADD (Y,1,TGL) AS GAJIAN_THN_DEPAN FROM PENGAJIAN`

3. Fungsi DATEDIFF

Digunakan untuk menghitung jumlah hari, bulan dan tahun diantara rentang tanggal yang telah ditentukan.

Sintaks : `SELECT DATEDIFF (DATEPART, STARTDATE,ENDDATE)`

IV. Fungsi Aritmatika

Fungsi SQRT digunakan untuk mencari nilai akar dari suatu nilai

Fungsi CEILING digunakan untuk pembulatan ke atas dari suatu angka desimal

Fungsi FLOOR digunakan untuk pembulatan ke bawah dari suatu angka desimal

V. Fungsi Konversi

Untuk keperluan konversi data digunakan fungsi CONVERT, yaitu mengubah data dari suatu tipe menjadi tipe lain

VI. Fungsi Karakter

Fungsi Lower → mengubah sebuah string menjadi string kecil

Fungsi Upper → merubah sebuah string menjadi string kapital

Fungsi Len → menghitung banyaknya karakter dari sebuah deretan nilai string

Fungsi Substring -> untuk mengambil beberapa nilai karakter dari sebuah deretan nilai string

Fungsi Left -> mengambil beberapa karakter dimulai dari sebelah kiri deretan string

Fungsi Right → untuk mengambil beberapa karakter dimulai dari sebelah kanan deretan string

Pelaksanaan Praktikum

Mempelajari Fungsi-fungsi SQL Server

Latihan Fungsi SQL di LAB

Buatlah tabel ujian, Atlet dan isikan datanya di Database dengan nama = DB_FUNGSI

Isi Tabel Ujian

| No_peserta | Nama_peserta | Nilai_ujian |
|------------|--------------|-------------|
| 1 | Naufal | 80 |
| 2 | Farman | 60 |
| 3 | Esa | 55 |
| 4 | Nadif | 65 |
| 5 | Rama | 70 |
| 6 | Pandu | 50 |
| 7 | Adit | 75 |
| 8 | Arvi | 65 |

Dengan menggunakan tabel Ujian selesaikan soal berikut ini

1. Buatlah perintah untuk menampilkan nama peserta ujian yang nilainya paling tinggi
2. Buatlah perintah untuk menampilkan nama peserta ujian yang nilainya paling rendah
3. Buatlah perintah untuk menghitung banyaknya peserta ujian yang memperoleh nilai rata-rata
4. Buatlah perintah untuk menampilkan 2 peserta ujian yang nilainya rendah

Isi tabel Atlet

| Id_atlet | Nama_atlet | Asal daerah | gender |
|----------|------------------|-------------|--------|
| 1 | Budi Riandi | DIY | L |
| 2 | Susi susanti | DKI Jakarta | P |
| 3 | Febrianti | Surabaya | P |
| 4 | Perdana Anwar | Sumbar | L |
| 5 | Ani Wartiningsih | DIY | P |
| 6 | Firmansyah | Surabaya | L |
| 7 | Cahyo Hidayat | DIY | L |
| 8 | Adi Putra | DIY | L |
| 9 | Okta Ashriza | Sumbar | P |

Menggunakan tabel Atlet untuk menyelesaikan soal

1. Buatlah perintah untuk menampilkan nama daerah yang mempunyai atlet paling banyak
2. Buatlah perintah untuk menghitung banyaknya daerah yang merupakan asal atlet
3. Menghitung banyaknya peserta per daerah asal , khusus untuk yang banyaknya lebih dari satu
4. Buatlah perintah untuk menghitung banyaknya atlet laki-laki dan perempuan setiap daerah asal
5. Buatlah perintah untuk mencari atlet tersedikit setiap daerah asal, apakah laki-laki atau perempuan
6. Buatlah perintah untuk menampilkan nama daerah beserta banyaknya atlet masing-masing daerah, dimana hasilnya akan ditampilkan menurut urutan jumlah atlet
7. Buatlah perintah untuk mencari atlet tersedikit setiap daerah, apakah laki-laki atau perempuan
8. Buatlah perintah untuk menampilkan nama daerah beserta banyaknya atlet masing-masing daerah, dimana hasilnya akan ditampilkan menurut urutan jumlah atlet.

Tugas

- I. Dengan menggunakan tabel mata_kuliah untuk menyelesaikan soal berikut :
Tabel Matakuliah

| Kode_mk | Nama_mk | Sks | Kode_prasyarat |
|---------|-------------------------|-----|----------------|
| DT005 | Kalkulus | 4 | Null |
| DT006 | Logika dan Algoritma | 4 | Null |
| DT007 | Pemrograman Komputer I | 2 | Null |
| DT010 | Pengantar Ilmu Komputer | 2 | Null |
| DT015 | Struktur Data | 4 | DT007 |
| DT018 | Organisasi Komputer | 4 | DT010 |
| DT019 | Sistem Operasi | 4 | DT010 |
| DT022 | Pengolahan Basis Data | 4 | NULL |
| DT023 | Matematika Diskret | 2 | DT005 |
| DT028 | Jaringan Komputer I | 2 | DT018 |
| DT030 | Pemrograman Komputer II | 4 | DT015 |
| DT034 | Pemrograman Basis Data | 4 | DT022 |
| DT037 | Jaringan Komputer II | 4 | DT028 |
| DT041 | Pemrograman Client | 2 | DT037 |
| ST011 | Manajemen Umum | 2 | NULL |
| ST017 | Teknik Digital | 2 | NULL |
| ST025 | Mikroprosesor | 2 | ST017 |
| ST026 | Praktikum Hardware | 2 | ST017 |
| ST027 | Praktikum Elektronik | 2 | ST017 |
| ST031 | Kepemimpinan | 2 | ST011 |
| ST035 | Sistem Informasi mana | 4 | ST011 |

1. Buatlah perintah untuk menghitung banyaknya matakuliah per sks per kuliah prasyarat
2. Buatlah perintah untuk menghitung banyaknya matakuliah, masing-masing yang mempunyai kuliah prasyarat dan yang tidak mempunyai kuliah prasyarat
3. Buatlah perintah untuk menghitung banyaknya matakuliah berbobot 4 sks, masing-masing yg tidak mempunyai kuliah prasyarat
4. Buatlah perintah untuk menghitung banyaknya matakuliah per sks per kuliah prasyarat.
5. Buatlah perintah untuk menghitung banyaknya matakuliah yang memuat kata 'pemrograman'
6. Buatlah perintah untuk menghitung banyaknya matakuliah yang memuat kata 'komputer' dan 'basis data' serta mempunyai kuliah prasyarat.
7. Buatlah perintah untuk memperoleh sks terkecil untuk matakuliah yang mempunyai prasyarat
8. Buatlah perintah untuk menampilkan nama matakuliah yang menjadi huruf besar semua

II. Dengan menggunakan tabel pinjaman berikut ini, untuk menyelesaikan soal berikut ini :

| Tgl_pinjaman | Nama | Jml_pinjaman | Bunga (dlm %) |
|--------------|-------------------|--------------|---------------|
| 02/14/2005 | Naufal Rasendriya | 2000.000 | 4 |
| 06/26/2005 | Najwa Rashika | 2500.000 | 5 |
| 10/23/2005 | Ema Utami | 1000.000 | 2 |
| 11/19/2005 | Suwanto Raharjo | 3.500.000 | 7 |

1. Buatlah perintah untuk menghitung hutang yang harus dibayarkan untuk masing-masing anggota koperasi dengan ketentuan bunga seperti data di atas.
2. Buatlah perintah untuk menghitung keuntungan yang didapatkan koperasi
3. Jika setiap anggota dikenakan kewajiban membayar hutang tersebut dengan jangka waktu 60 hari, buatlah perintah untuk menghitung kapan hutang-hutang tersebut jatuh tempo

Universitas
Esa Unggul



Pertemuan VI

JOIN ANTAR TABEL

Tujuan

Mahasiswa memahami bagaimana perintah SQL dalam menggabungkan data dari berbagai

tabel.

Materi

1. Cross Join/Cartesian Join
2. Inner Join
3. Outer Join
4. Operasi Join Untuk Penginputan Data

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

Join merupakan sebuah operasi yang digunakan untuk mendapatkan data gabungan dari dua tabel atau lebih. Operasi ini digunakan dalam perintah SELECT dan biasanya dipakai untuk memperoleh detail data dari tabel-tabel yang saling terkait

I. CROSS JOIN/CARTESIAN JOIN

Menghasilkan Kombinasi Semua Baris Yang Terdapat Dalam tabel-tabel yang digabungkan baik yang tidak berpasangan maupun berpasangan.

Contoh:

T.Mhs

| Nim | Nama | Gender | Agama |
|------------|-------------------|--------|-----------|
| 04.01.1880 | Vita | P | Protestan |
| 04.01.1990 | Naufal Rasendriya | L | Islam |
| 04.11.5660 | Najwa | P | Islam |

T.mk

| Kode_Mk | Nama_MK | SKS |
|---------|---------------------------|-----|
| DKT1201 | Pemrograman Komp I | 2 |
| DKT5224 | Struktur Data | 4 |
| KKT3303 | Pengolahan Basis Data | 4 |
| KKT5431 | Pemrograman Client/Server | 2 |
| KKT5432 | Pemrograman Komp II | 4 |

Sehingga, jika kita lakukan proses Cross Join antar T.Mhs dengan T.Matakuliah

Select mhs.nama, mk.nama_mk from mhs,mk

Hasilnya

| No | Nama | Nama_mk |
|----|--------------------|---------------------------|
| 1 | vita | Pemrograman Komp I |
| 2 | vita | Struktur Data |
| 3 | vita | Pengolahan Basis Data |
| 4 | vita | Pemrograman Client/Server |
| 5 | vita | Pemrograman Komp II |
| 6 | Naufal Rasendriaya | Pemrograman Komp I |
| 7 | Naufal Rasendriaya | Struktur Data |
| 8 | Naufal Rasendriaya | Pengolahan Basis Data |
| 9 | Naufal Rasendriaya | Pemrograman Client/Server |
| 10 | Naufal Rasendriaya | Pemrograman Komp II |
| 11 | Najwa | Pemrograman Komp I |
| 12 | Najwa | Struktur Data |
| 13 | Najwa | Pengolahan Basis Data |
| 14 | Najwa | Pemrograman Client/Server |
| 15 | Najwa | Pemrograman Komp II |

II. INNER JOIN

Join ini menghasilkan output yang berupa kombinasi baris-baris yang memiliki pasangan saja, kombinasi baris yang bukan pasangan akan dieliminasi (baris-baris yang tidak memiliki pasangan pada table lainnya tidak dimunculkan).

Contoh:

T.krs

| Nim | Kode_mk | Semester | Thn_ajaran | Nilai |
|------------|---------|----------|------------|-------|
| 04.01.1990 | Kkt3303 | 3 | 2004/2005 | A |
| 04.01.1990 | Kkt5432 | 3 | 2004/2005 | A |
| 04.01.1990 | Dkt1201 | 2 | 2004/2005 | C |
| 04.01.1990 | Dkt5224 | 4 | 2005/2006 | B |
| 04.01.1990 | Kkt5431 | 4 | 2005/2006 | B |
| 04.11.5660 | Kkt3303 | 3 | 2004/2005 | A |
| 04.11.5660 | Kkt5432 | 4 | 2005/2006 | B |

Contoh → join antar table KRS dan matakuliah

| | | |
|----|--|---|
| At | <pre>Select * from krs,mk Where krs.kode_mk = mk.kode_mk</pre> | <pre>Select * from krs inner join mk On krs.kode_mk =mk.kode_mk</pre> |
|----|--|---|

III. OUTER JOIN

Hampir sama dengan inner join, hanya saja baris yang tidak memiliki pasangan tetap akan ditampilkan.

Join jenis ini dibagi lagi menjadi :

1. Left Outer Join

Bila baris pada tabel yang terletak di sebelah kiri operator LEFT OUTER JOIN ada yang tidak memiliki pasangan dengan tabel yang terletak di kanan operator Left Outer Join, baris ini tetap disertakan dalam hasil penggabungan

Contoh -> Left Outer Join T.Mhs dan T.krs

```
Select m.nim, m.nama, k.kode_mk, k.nilai
From mhs m, left join krs k
On m.nim = k.nim
```

2. Right Outer Join

Bila baris pada tabel yang terletak disebelah kanan operator RIGHT OUTER JOIN ada yang tidak memiliki pasangan dengan tabel yang terletak di kanan operator RIGHT OUTER JOIN, baris ini tetap disertakan dalam hasil penggabungan.

Contoh → join antara Tabel mhs dengan kelas

```
Select m.nim,m.nama, k.kode_kelas as kelas,
k.kode_jur as jurusan
From mahasiswa m right join kelas k
On m.kode_kelas = k.kode_kelas
```

IV. Operasi Join Untuk Penginputan data

Operasi join juga dapat digunakan untuk mengisikan data pada suatu kolom dari suatu tabel yang merupakan kolom hasil perhitungan dari kolom-kolom pada tabel yang berbeda

Contoh → mengisikan tabel penggajian

```
Insert into penggajian (no_kwitansi, kode_kar, tgl, gaji_kotor,
bonus, potongan)
Values ('00001','k0001','01/01/2006',375000,420000,50000)
```

Pelaksanaan Praktikum

Mempelajari perintah Join Antar Tabel

Tugas Praktikum

Berikut ini diberikan database market dealer



Relasi antar Tabel Market dealer

Berdasarkan Relasi Tabel di Atas, Buatlah perintah SQL untuk perintah dibawah ini:

1. Tampilkan data pembelian (kode pembelian, hari, tanggal, nama supplier, nama barang, jumlah)
2. Tampilkan data penjualan (kode penjualan, hari, tanggal, nama agen, nama barang, jumlah)
3. Tampilkan nama barang yang terjual dengan jumlah > 20
4. Tampilkan nama barang dan jumlah barang yang terjual pada tgl 1 september 2008, dikelompokkan berdasarkan nama barang
5. Tampilkan nama dan alamat supplier yang menyeluplai sabun
6. Tampilkan nama dan alamat agen yang membeli sampo
7. Tampilkan nama barang yang terbeli dengan agen yang beralamat Yogyakarta
8. Tampilkan nama supplier yang menyuplai barangnya pada tgl 2 september 2008

Pertemuan VII VIEW

Tujuan

Mahasiswa memahami cara membuat View baik dengan menggunakan enterprise Manager maupun dengan Query Analyzer, memodifikasi dan menghapus view dengan menggunakan Query Analyzer pada database SQL SERVER 2000.

Materi

1. Penjelasan singkat tentang View
2. Sintak Sql dalam membuat View
3. melakukan dan menjalankan perintah dalam memodifikasi View

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

View adalah sebuah tabel virtual yang keberadaanya tidak ada di dalam database dan isinya di definisikan oleh query. Fungsi View itu sendiri adalah untuk menyederhanakan proses manipulasi data.

I. Membuat View

Beberapa hal yang perlu diperhatikan dalam membuat View, diantaranya :

- Hanya dapat dibuat pada database yang sedang dipakai
- Nama view secara prinsip sama dengan nama tabel biasa namun, biasanya untuk membedakan dengan tabel, view sering diawali dengan huruf V atau awalan view_
- View dapat dibuat dari view lain
- Jumlah kolom yang tertera dalam view harus sama dengan jumlah kolom dalam query

Sintaks pembuatan View:

```
CREATE VIEW Nama_View [daftar kolom]
As
Ekspresi_Select
```

Contoh

```
CREATE VIEW VDataMHS
As
Select nim, nama from mahasiswa
```

Hasil dari pembuatan View dapat dilihat dengan menggunakan perintah :

```
Select * from VDataMHS
```

II. Mengubah View

Setelah view didefinisikan, kita dapat mengubah nama atau memodifikasi definisi view tersebut tanpa harus menghapus dan membuat ulang View tersebut

Sintak : untuk memodifikasi field2x di View

```
Alter view VDataMHS  
As  
Select nim as [NIM] , nama as [nama  
mhs] from mahasiswa  
Where nim = s001
```

III. Mengakses data Tabel asal menggunakan View

Kita dapat melakukan beberapa macam akses data melalui view, diantaranya :

1. Mengisi Tabel asal dari View

```
Insert VDataMHS  
Values ('s002','lia')
```

2. Mengganti Isi Tabel asal dari View

```
Update VDataMHS  
Set nama = 'sandra'  
Where nim ='s002'
```

3. Menghapus Isi Tabel asal dari View

```
Delete VDataMHS  
Where nim='s002'
```

IV. Menghapus View

Penghapusan View VDataMHS di atas dapat dilakukan dengan perintah,

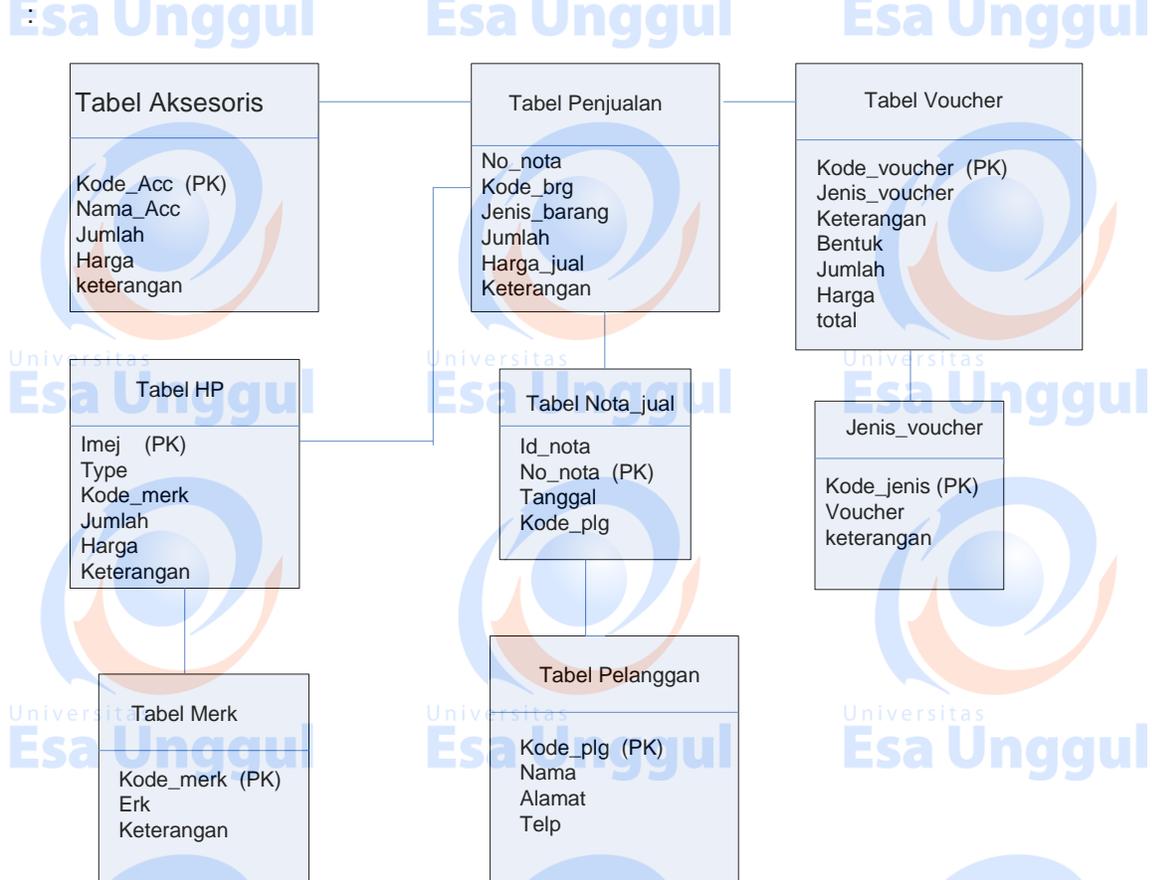
```
Drop View VDataMHS
```

Pelaksanaan Praktikum

Mempelajari cara pembuatan VIEW dan manipulasi VIEW

Latihan VIEW di LAB

Perhatikan database diagram counter untuk pengolahan data counter HP Sell Bandung berikut ini



Gambar : Database Diagram Counter

Isi Tabel Pelanggan

| Kode_plg | Nama | Alamat | Telp |
|----------|------|---------------|--------------|
| P001 | Dwi | Jl, Babarsari | 081328341367 |

| | | | |
|-------|----------|----------------------------|--------------|
| P002 | Uswan | Godean | 081328341368 |
| P003 | Sukrisno | Jl Tanjung 11 Concat YK | 081328341365 |
| P005 | Della | Slokan Mataram | 081328341369 |
| P006 | Rendra | Jl Magelang Km.7 | 081328341378 |
| P007 | Eko | Mancasan | 081328341314 |
| P0024 | Endik | Bantul | 081328341348 |

Isi Tabel Merk

| Kode_Merk | Merk | Keterangan |
|-----------|--------------|------------|
| MT | MOTOROLLA | |
| NK | NOKIA | LKP MLS |
| PL | PHILIPS | |
| PN | PANASONIC | |
| SG | SAGEM | |
| SM | SAMSUNG | |
| SMN | SIEMENS | |
| SN | SONY ERICSON | BAGUS |

Isi Tabel HP

| Imej | Type | Kode_merk | Jumlah | Harga | Keterangan |
|-------|-----------|-----------|--------|---------|------------|
| HP-1 | Null | Null | 0 | 5000000 | NULL |
| HP-2 | 8250 | Nk | 0 | 550000 | |
| HP-3 | N-Gage | Nk | 0 | 1250000 | |
| HP-4 | N-Gage 21 | Nk | 0 | 2350000 | |
| HP-5 | A | Nk | 1 | 100000 | |
| HP-6 | 3310 | Nk | 0 | 300000 | I KP |
| HP-7 | V | Nk | 0 | 2000000 | |
| HP-8 | C35 | SMN | 1 | 574000 | |
| HP-9 | C50 | SMN | 1 | 450000 | |
| HP-10 | C100 | NK | 1 | 560000 | |
| HP-11 | 9500 | NK | 0 | 7000000 | |

Isi Tabel Aksesories

| Kode_acc | Nama_Acc | JUmlah | HArga | Keterangan |
|-----------|-------------------|--------|--------|---------------|
| 12345 | Casing Nokia 8250 | 7 | 120000 | |
| SIM-IM3 | Im3 | 0 | 25000 | |
| SIM-MENT | Simcard Mentari | 0 | 12000 | 0815123456768 |
| SIM-SIMPT | Simcard Simpati | 0 | 12500 | |

Isi Tabel Jenis_Voucher

| Kode_jenis | Voucher | Keterangan |
|------------|----------|------------|
| AS | KArtu As | |
| IM3 | IM3 | |
| MT | MENTARI | |
| SMP | SIMPATI | |
| XL | PRO XL | |

Isi Tabel Voucher

| Kode_voucher | Jenis_voucher | Keterangan | Bentuk | JUmlah | Harga | Total |
|--------------|---------------|----------------------|------------|--------|-------|-------|
| IM3E20 | IM3 | IM3 Elektronik 20000 | Elektronik | 0 | 20000 | 0 |

| | | | | | | |
|--------|-----|--------------------------|------------|----|--------|--------|
| MTE20 | MT | Mintari Elektrik 20000 | Elektronik | 0 | 20000 | 0 |
| SME100 | SMP | Simpatielektrik 100000 | Elektronik | 0 | 100000 | 0 |
| SME20 | SMP | Simpatielektrik 20000 | Elektronik | 0 | 20000 | 0 |
| SME50 | SMP | Simpatielektrik 50000 | Elektronik | 0 | 50000 | 0 |
| SMF20 | SMP | Simpatifisik 20000 | fisik | 7 | 20000 | 140000 |
| SMF50 | SMP | Simpatifisik 50000 | fisik | 10 | 50000 | 500000 |
| XJE20 | XL | XI Jempol Elektrik 20000 | fisik | 0 | 20000 | 0 |

Isi Tabel Nota_jual

| Id_nota | No_nota | Tanggal | Kode_plg |
|---------|---------|------------|----------|
| 182 | 000001 | 2008-07-19 | Null |
| 184 | 000002 | 2008-07-23 | P003 |
| 185 | 000003 | 2008-07-23 | P003 |
| 186 | 000004 | 2008-07-23 | NULL |
| 187 | 000005 | 2008-07-23 | NULL |
| 188 | 000006 | 2008-07-27 | NULL |
| 189 | 000007 | 2008-07-28 | P003 |

Isi Tabel_penjualan

| No_nota | Kode_brg | Jenis_barang | Jumlah | Harga_jual | Keterangan |
|---------|----------|--------------|--------|------------|-------------|
| 000001 | 1234 | AC | 1 | 12000 | NULL |
| 000001 | SIM-MENT | AC | 1 | 13000 | NULL |
| 000002 | HP-11 | HP | 1 | 7650000 | |
| 000003 | HP-4 | HP | 1 | 550000 | |
| 000004 | IM3E20 | VC | 1 | 21500 | 08564300909 |
| 000005 | 12345 | AC | 2 | 12500 | NULL |
| 000005 | SIM-IM3 | AC | 1 | 15000 | NULL |
| 000006 | 12345 | AC | 1 | 20000 | NULL |
| 000007 | HP-4 | HP | 1 | 450000 | |

Dengan menggunakan relasi table di atas, buatlah view-view berikut ini :

1. Menampilkan data HP yang telah terjual, kolom yang ditampilkan adalah imej, type, merk dan harga jual
2. Menampilkan data HP yang masih tersedia yang merknya adalah 'NOKIA', kolom yang ditampilkan adalah imej, type, merk, jumlah dan harga
3. Menampilkan jumlah barang yang terjual dan total harga penjualan pada tanggal 19 july 2008
4. Menampilkan voucher yang paling banyak terjual , kolom yang ditampilkan adalah voucher dan total dari harga jual
5. Menampilkan stok voucher yang masih tersisa paling banyak , kolom yang ditampilkan adalah voucher dan jumlah total
6. Menampilkan harga aksesories yang paling mahal, kolom yang ditampilkan adalah semua kolom dalam tabel aksesories

Pertemuan VIII PROCEDURE

Tujuan

Mahasiswa memahami cara membuat, memodifikasi dan menghapus procedure dengan menggunakan Query Analyzer pada database SQL SERVER 2000.

Materi

1. Penjelasan singkat tentang procedure
2. Sintak Sql dalam membuat procedure
3. melakukan dan menjalankan perintah dalam memodifikasi procedure

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

I. Stored Procedure

Dalam Sql Server 2000, selain memiliki fungsi internal juga dapat dibuat fungsi yang didefinisikan secara mandiri. Fungsi tersebut akan dieksekusi oleh server dan berada dalam database itu sendiri sehingga disebut dengan Stored Procedure.

Stored procedure adalah sekumpulan perintah-perintah SQL yang tersimpan dengan nama tertentu dan diproses sebagai suatu kesatuan. Secara sederhana store procedure merupakan sub-program yang tersimpan dalam database.

Hal-hal yang dapat dilakukan oleh store procedure adalah

1. Menerima parameter sebagai input dan mengembalikan nilai dalam parameter output.
2. Mengandung perintah program yang melaksanakan operasi dalam database
3. Mengembalikan suatu nilai status pada pemanggilnya untuk mengindikasikan kesuksesan atau kegagalan procedure

Keuntungan menggunakan store procedure adalah

1. Pemrograman menjadi modular
Dimana kita hanya membuat store procedure sekali, menyimpannya dalam database, dan bisa memanggilnya berulang kali dari program kita.
2. Eksekusi program akan menjadi lebih cepat

Menjalankan program untuk mengakses database tentu saja akan lebih cepat jika dilakukan di dlm database itu sendiri.

3. Mengurangi lalu lintas jaringan

Mengirimkan nama procedure dalam aplikasi client server, akan lebih sedikit daripada mengirimkan baris-baris perintah yang ada di dlmnya.

4. Dapat digunakan untuk mekanisme keamanan

User dapat diberi hak untuk menjalankan store procedure walaupun dia tidak memiliki hak untuk menjalankan perintah yg ada di dalamnya.

II. Sintak SQL Membuat Store Procedure

Sintaks Umum

```
CREATE PROCEDURE nama_procedure  
Parameter_1 Tipedata_1,  
Parameter_1 Tipedata_1,  
AS
```

Contoh 1

Kita akan membuat procedure yang menampilkan kode matakuliah dan nama matakuliah

```
create procedure data_mk  
as  
select kode_mk, nama_mk from matakuliah
```

Maka perintah untuk menjalankannya, dengan mengetikkan data_mk

Contoh 2 → procedure pake parameter input

Bagaimana menambahkan parameter input dlm store procedure

```
create procedure data_mk1  
@kode char(5)  
as  
select kode_mk, nama_mk from matakuliah  
where kode_mk = @kode
```

Maka perintah untuk menjalankannya, dengan mengetikkan : data_mk1 'dt041'

Contoh 3 → procedure untuk insert

Membuat store procedure untuk menambah data matakuliah

```
create procedure proc_tambah_mk
@kode_mk char(5),
@nama_mk varchar(30),
@sks numeric
as
insert into matakuliah values (@kode_mk,@nama_mk,@sks)
```

Perintah untuk menjalankannya :

```
proc_tambah_mk 'dt045','pemrograman internet',4
```

Contoh 4 -> procedure untuk update

```
create procedure proc_ubah_sks
@kode char(5),
@sks numeric
as
update matakuliah
set sks = @sks
where kode_mk = @kode
```

Perintah untuk menjalankannya :

```
proc_ubah_sks 'dt041',3
```

Contoh 4 -> procedure untuk Hapus

```
create procedure proc_hapus_mk
@kode char(5)
as
delete from matakuliah
where kode_mk = @kode
```

Perintah untuk menjalankannya

```
proc_hapus_mk 'dt041'
```

Contoh 5 -> procedure untuk Pencarian

```
create procedure proc_cari_mhs
@nim char(10)
as
select m.nama, mk.nama_mk, k.semester, k.nilai
from mahasiswa m, krs k, matakuliah mk
where m.nim = k.nim and mk.kode_mk = k.kode_mk and m.nim = @nim
```

Perintah untuk menjalankannya

```
proc_cari_mhs '04011880'
```

Pelaksanaan Praktikum

Mempelajari cara pembuatan Procedure dan manipulasi Procedure

Latihan Procedure di LAB

- A. Jelaskan perbedaan cara kerja dari store procedure berparameter dan store procedure tanpa berparameter
- B. Gunakan tabel jabatan, karyawan, dan penggajian yang terdapat pada database Gaji

Isi Tabel Jabatan

| Kode_jabatan | Jabatan | Gaji_pokok | Tunjangan |
|--------------|-----------|------------|-----------|
| Dir | Direktur | 2500000 | 1250000 |
| Keu | Keuangan | 2200000 | 1100000 |
| Man | Manager | 2250000 | 1150000 |
| mar | Marketing | 1250000 | 800000 |

Isi Tabel Karyawan

| Kode_kar | Nama | Kode_jabatan | Gender | Tgl_lahir | Alamat | Ygl_mulai_kerja |
|----------|--------------|--------------|--------|------------|-----------------------|-----------------|
| K0001 | Ema Utami | Dir | P | 1979-02-21 | Jl Magelang 9 | 1997-01-10 |
| K0002 | Naufal | Man | L | 1980-08-28 | Jl Gejayan | 1998-09-10 |
| K0003 | Najwa | Keu | P | 1980-08-12 | Jl Taman Siswa 110 | 1999-12-08 |

Isi Tabel Penggajian

| Id | No_kwitansi | Kode_kar | Tgl | Gaji_kotor | Bonus | potongan |
|----|-------------|----------|------------|------------|--------|----------|
| 1 | 00001 | K0001 | 2006-01-01 | 3750000 | 250000 | 100000 |
| 2 | 00002 | K0002 | | 3750000 | 420000 | 50000 |

Untuk menyelesaikan soal berikut ini

1. Buatlah store procedure untuk menambah data jabatan
2. Buatlah store procedure untuk menambah data Karyawan
3. Buatlah store procedure untuk menambah data penggajian
4. Buatlah store procedure untuk menghapus data jabatan
5. Buatlah store procedure untuk menghapus data karyawan
6. Buatlah store procedure untuk menghapus data penggajian berdasarkan bulan
7. Buatlah store procedure untuk melihat data gaji perkaryawan
8. Buatlah store procedure untuk mengubah data jabatan
9. Buatlah store procedure untuk mengubah data karyawan
10. Buatlah store procedure untuk mengubah data penggajian
11. Buatlah store procedure untuk melihat karyawan yang ulang tahun pada bulan tertentu
12. Buatlah store procedure untuk memberikan potongan zakat (2.5 %) pada table penggajian



Pertemuan IX Transaction dan Trigger

Tujuan

Mahasiswa memahami cara membuat, memodifikasi dan menghapus procedure dengan menggunakan Query Analyzer pada database SQL SERVER 2000.

Materi

1. Penjelasan Tentang Trigger
2. Pembuatan trigger
3. Transaction
4. Penggunaan Trigger untuk Penginputan data

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

Transaction dalam pemrograman database digunakan untuk menangani terjadinya error dalam menangani terjadinya error dalam memanipulasi database. Sedangkan trigger merupakan tipe spesial dari Store Procedure yang berjalan secara Event Driven saat user melakukan modifikasi khusus terhadap tabel tertentu. SQL SERVER akan menjalankan trigger ketika ada perintah INSERT, UPDATE, DELETE pada tabel.

Trigger biasanya digunakan untuk menjaga integrasi data, menangani pergantian data serta dapat melakukan aksi lain saat tabel dimodifikasi dengan mengenai pernyataan INSERT, UPDATE, DELETE. Secara sederhana dapat dikatakan bahwa trigger adalah perkataan 'jalankan fungsi ini jika kejadian terjadi pada tabel ini'.

Pengaktifan trigger dapat memberikan manfaat yang sangat besar, seperti digunakan untuk :

1. Mencegah kesalahan pemasukan data
2. Mencegah kesalahan penghapusan data
3. Mencegah kesalahan pengupdatean data

I. Membuat Trigger

Perintah →

```
CREATE TRIGGER triggername
On tablename
For INSERT, UPDATE, DELETE
AS
DECLARE
Statement sql
```

Penjelasan:

- Statement CREATE TRIGGER diikuti dengan nama trigger yg akan dibuat, nama trigger harus mengikuti aturan identifier dan harus unik di dalam satu database.
- Keyword ON diikuti dengan nama tabel yang akan dihubungkan dengan Trigger. Tabel adalah tempat dimana trigger tersebut berada dan dieksekusi.
- Keyword FOR diikuti perintah yang akan mengaktifkan trigger, yaitu UPDATE, INSERT, atau DELETE.
- Keyword AS diikuti aksi yang dilakukan oleh trigger tersebut
- Statement SQL adalah kondisi dan aksi perintah SQL yang ada saat trigger itu dijalankan
- N merupakan variabel yang menyatakan bahwa anda bisa menyertakan banyak pernyataan Transact-SQL

Contoh → Trigger Update Terhadap Tabel Atlet

T.Atlet

| Id_atlet | Nama_atlet | Asal_daerah | Gender |
|----------|-----------------|-------------|--------|
| 1 | Budi Riadi | DIY | L |
| 2 | Susi Susanti | DKI Jakarta | P |
| 3 | Febrianti | Surabaya | P |
| 4 | Perdana Anwar | Sumbar | L |
| 5 | Ani Wartinarsih | DIY | P |
| 6 | Firmansyah | Surabaya | L |
| 7 | Cahyo Hidayat | DIY | L |
| 8 | Adi Putra | DIY | L |
| 9 | Okta Ashriza | Sumbar | P |

```
Create trigger T_updateAtlet
On atlet
For update
As
Print 'Tabel Atlet telah di Update'
```

Trigger di atas akan memberikan respon ketika tabel atlet tersebut dikenai perintah UPDATE. Respons yang akan terjadi adalah print 'tabel atlet telah di update'. Untuk melihat hasilnya kita harus melakukan perintah UPDATE terhadap tabel atlet.

```
Update atlet
Set nama_atlet = 'yuliati' where id_atlet = 3
```

Hasil dari eksekusi statement di atas adalah

Tabel atlet telah di Update

II. Transaction

Sintaks dasar dari Transaction

```
BEGIN TRANSACTION
    Perintah sql
If @@error = 0
    Commit transaction
Else
    Rollback transaction
```

{ masih belum komplet}

{ grant and Revoke belum selesai}

Pertemuan XII

Mem-Backup dan Merestorasi Data

Tujuan

Mahasiswa cara membuat Backup, Merestorasi data

Materi

1. Jenis Backup
2. Membuat Backup
3. Merestorasi Backup

Software

1. Windows 2003 Operating System.
2. Sql Server 2000

Landasan Teori

SQL Server 2000 memiliki dua buah sarana yang dirancang untuk menjalankan salinan Backup dari database dan komponen-komponennya. Sarana ini sangat penting untuk kesinambungan operasi database apabila ada kegagalan pada perangkat keras, penghapusan tabel secara tidak sengaja atau bahkan kehilangan data pada server.

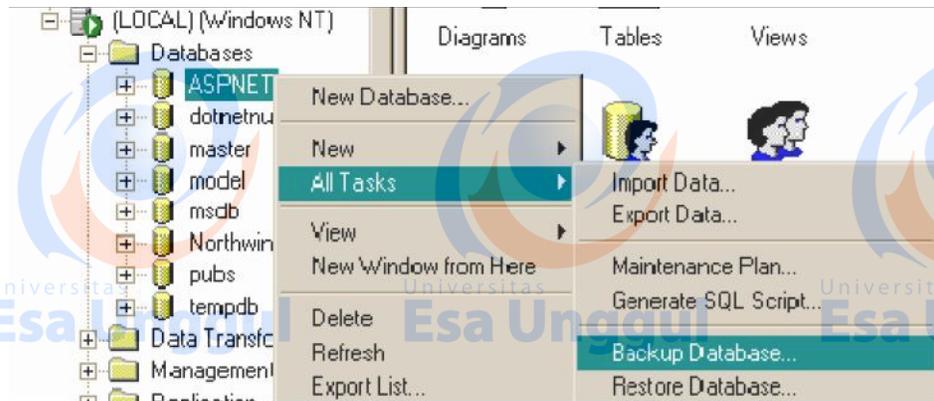
Membuat Backup meliputi meng-copy database ke lokasi yang aman. Salinan Backup harus dilakukan sebisa mungkin ke komputer lain didalam media network, tape, atau media magnetik lainnya. Proses backup menyalin semua yang terdapat di dalam database, termasuk transaksinya.

Catatan transaksi adalah sebuah record serial yang berisikan semua perubahan yang dilakukan terhadap database . log ini digunakan di dalam proses restorasi untuk mengulangi sebuah perubahan yang terjadi di dalam database semenjak proses Backup terakhir.

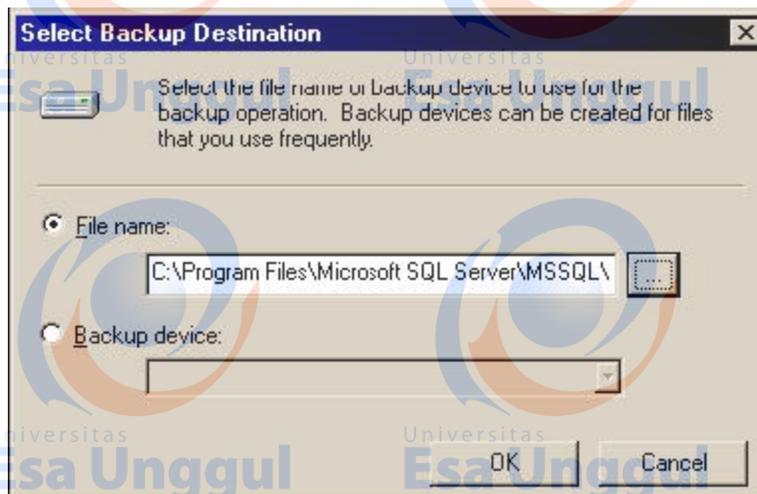
Database yang telah dibuat harus dibackup secara teratur dan disimpan di lokasi yang aman. File hasil backup dapat direstore ke server asal atau ke server lain.

Untuk melakukan backup database, ikuti langkah-langkah berikut:

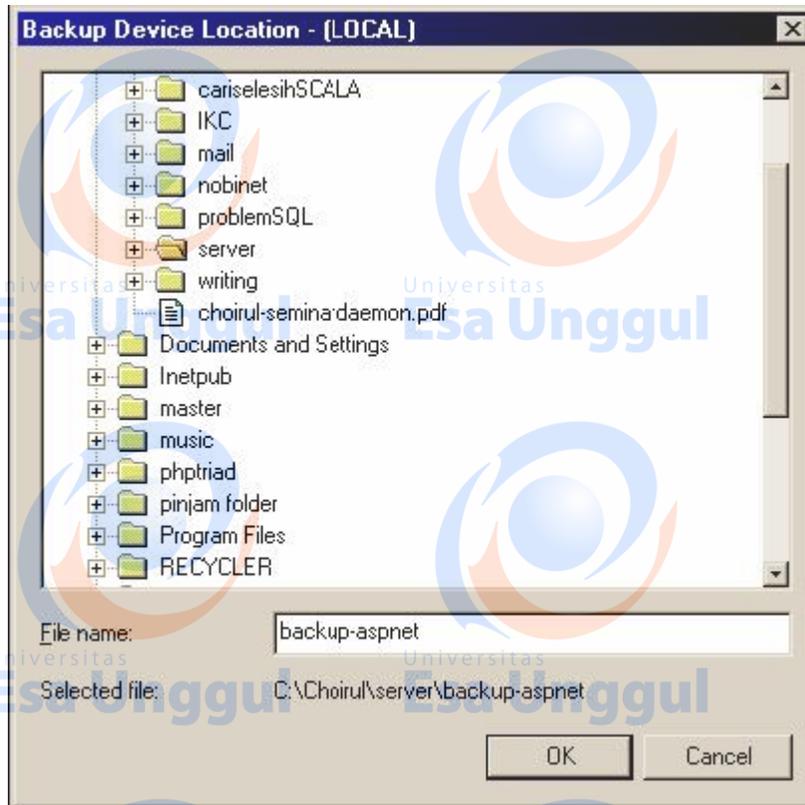
1. Klik kanan database yang akan dibackup, pilih All Task lalu Backup.



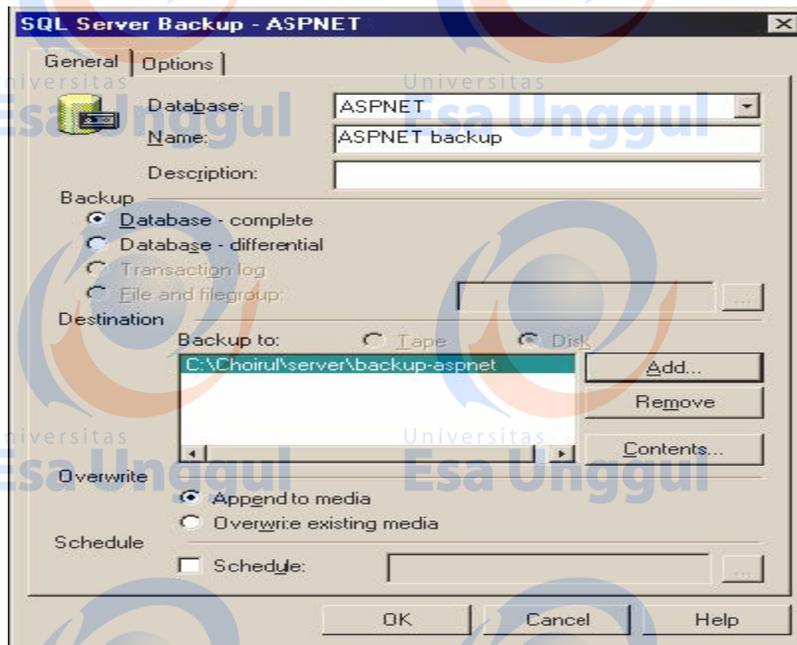
2. Terdapat 4 jenis tipe backup, pilih Database-Complete yang artinya seluruh isi database akan dibackup. Kemudian klik Add untuk menentukan lokasi backup. Klik tombol di sebelah kolom File name.



3. Pilih lokasi backup sesuai yang diinginkan dan berikan nama file hasil backup di kotak File name.



4. Klik OK setelah selesai, maka hasil pemilihan lokasi ditampilkan di property backup.



5. Klik OK, dan backup akan dilakukan di lokasi yang telah ditentukan.



Restore Database

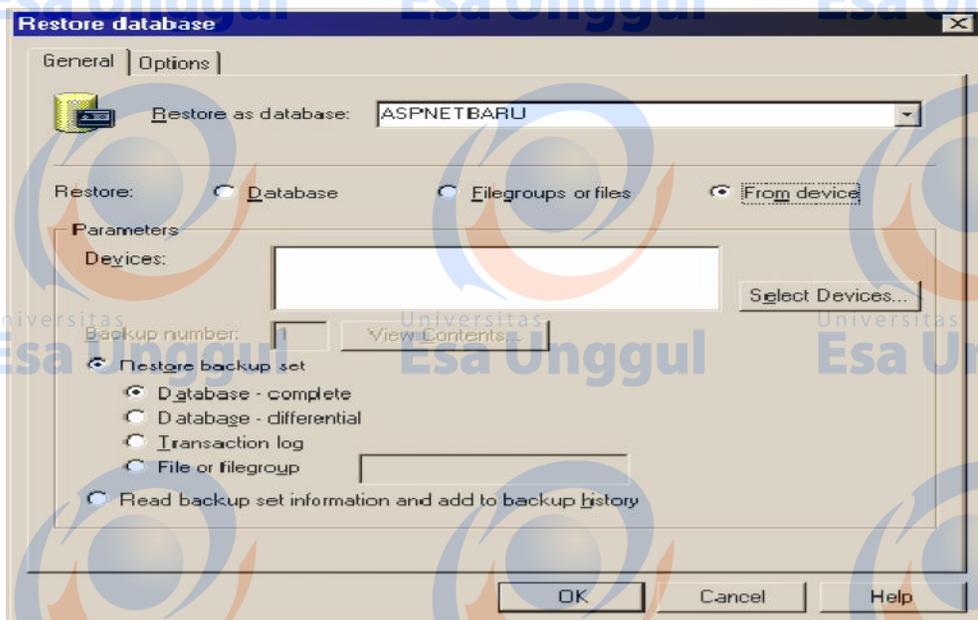
Proses restore adalah kebalikan dari backup. Dengan restore maka file hasil backup disimpan kembali ke SQL Server. Anda harus berhati-hati saat melakukan restore, agar tidak keliru dan menimpa database yang versi datanya lebih baru. Anda dapat menetapkan apakah suatu restore akan menimpa database lama, atau direstore sebagai database baru dengan nama berbeda.

Berikut langkah-langkah restore database:

1. Klik kanan folder database lalu pilih Restore. Setelah dialog restore ditampilkan, pada kotak Restore as database Anda dapat memilih apakah database lama akan ditimpa atau direstore sebagai database baru.

Apabila ingin menimpa database lama pilih nama database dari dropdown yang tersedia. Apabila ingin merestore sebagai database baru maka isikan nama database pada kolom tersebut.

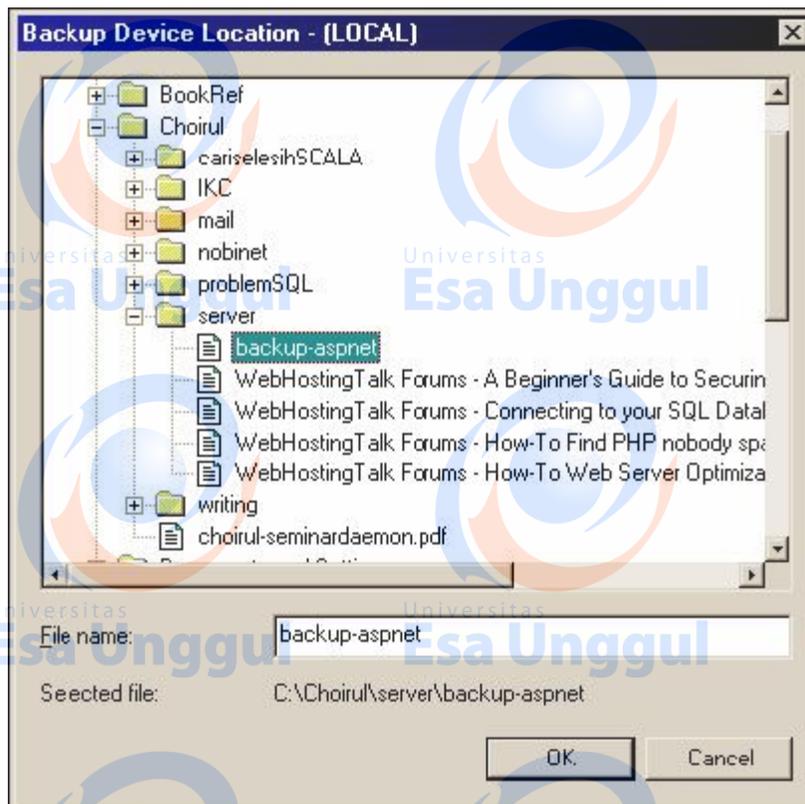
Pada contoh ini dibuat database baru dengan naama ASPNETBARU.



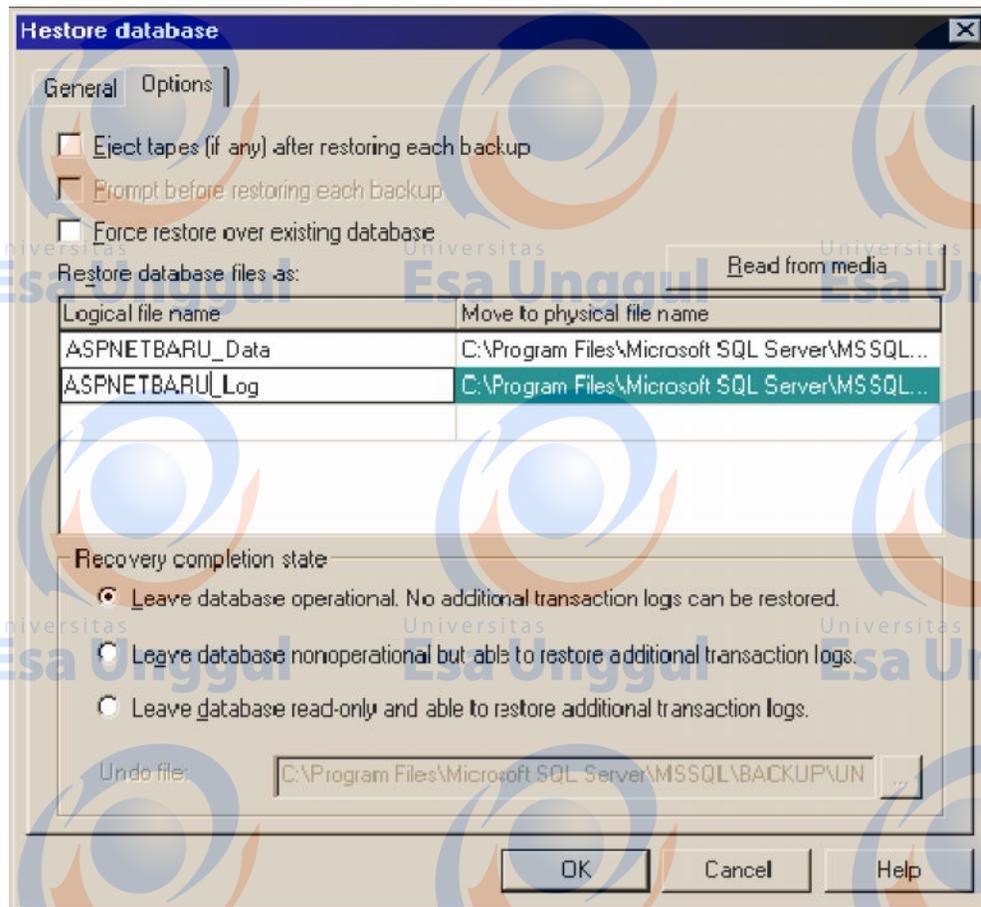
2. Klik from device untuk menentukan lokasi file backup. Kemudian klik Select Devices, pilih Disk dan klik Add untuk mencari lokasi file.



3. Pilih file hasil backup lalu klik OK sampai kembali ke menu konfigurasi restore.



- Setelah kembali ke menu restore database, klik option untuk menentukan nama dan lokasi file database yang akan dibuat dari hasil restore.
Secara default nama dan lokasi filenya sama dengan database ASPNET karena backupnya dibuat dari database ini. Karena akan dibuat restore ke database baru maka harus ditetapkan nama file database yang berbeda. Lokasi file boleh saja tetap sama dengan database ASPNET.
- Ganti nama file database dan log menjadi ASPNETBARU seperti berikut:



- Klik OK, maka restore dilakukan dan terbuat database baru bernama ASPNETBARU.

Note: Apabila Anda ingin menimpa database lama dengan file backup maka pada dialog Options harus dipilih: Force restore over existing database.

Pelaksanaan Praktikum

Mempelajari cara Mem-Backup dan Merestorasi Data

Latihan Procedure di LAB

A. Jelaskan Jenis Jenis Backup dan tahap-nya di SQL Server 2000

