

BAB - 4

Data Definition Language (DDL)

I. Data Definition Language (DDL)

DDL berfungsi menspesifikasikan skema atau struktur basisdata, hasil pernyataan DDL adalah himpunan definisi data yang disimpan secara khusus pada data dictionary (data directory).

Struktur/skema basis data yang menggambarkan/mewakili desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yaitu DDL. Dengan bahasa ini kita dapat membuat tabel (create table) baru, indeks, mengubah table, menentukan struktur penyimpanan table, dan lainnya. Hasil dari kompilasi perintah DDL, adalah kumpulan table yang disimpan dalam file khusus yang disebut kamus data (data dictionary).

Kamus data merupakan suatu metadata (superdata), yaitu data yang mendiskripsikan data sesungguhnya. Contoh perintah DDL dengan Foxpro adalah create matakuliah, modify report, modify structure, dan lainnya. Sedangkan perintah DDL dengan SQL Server 2005, contohnya adalah create new database kampus dan contoh lainnya adalah new table mahasiswa.

Perintah-perintah DDL terdiri dari CREATE, ALTER, dan DROP.

1. CREATE DATABASE

Perintah create database digunakan untuk membuat sebuah database.

Bentuk penulisan (Syntax) :

```
CREATE DATABASE nama_database
```

Contoh untuk membuat database dengan nama Latihan1

```
CREATE DATABASE Latihan1
```

```
CREATE DATABASE database_name
[ ON
{ [PRIMARY] (NAME=logical_file_name,
  FILENAME="os_file_name",
  [,SIZE =size]
  [,MAXSIZE=max_size]
  [ FILEGROWTH=growth_increment] ) } [,...n]
[ LOG ON
( NAME=logical_file_name,
  FILENAME='os_file_name',
  [,SIZE = max_size]
  [,FILEGROWTH=growth_increment]) } [ ,...n]
```

Database_name : nama untuk database baru dan harus bersifat unik.

ON : menspesifikasikan bahwa file-file disk yang digunakan untuk menyimpan porsi data dari database yang didefinisikan secara eksplisit.

Primary : menspesifikasikan relasi yang mendefinisikan primary file dan sebuah database hanya dapat mempunyai satu primary file.

FILEGROWTH=growth_increment : file data dapat mengembang apabila alokasi telah habis.

LOG ON : menspesifikasikan file-file yang digunakan untuk menyimpan database log (log files) yang didefinisikan secara eksplisit.

NAME=logical_file_name : untuk menspesifikasikan nama file logic dari database.

FILENAME='os_file_name' : tentukan nama file dan letak dimana file tersebut akan disimpan.

SIZE = max_size : memberikan inisialisasi ukuran/besar suatu file database.

2. DROP DATABASE

Perintah drop database digunakan untuk menghapus sebuah database, perintah tersebut akan menghapus database dan menghapus seluruh tabel yang berada didalamnya.

Bentuk penulisan (*Syntax*) :

```
DROP DATABASE nama_database
```

Contoh untuk menghapus database Latihan1

```
DROP DATABASE Latihan1
```

3. CREATE TABLE

Perintah create table digunakan untuk membuat sebuah table.

Bentuk penulisan (*Syntax*) :

```
CREATE TABLE nama_tabel  
(nama_kolom tipe_data [null|not null])
```

Untuk membatasi lebar data pada masing – masing field dapat ditambahkan ukuran dari field tersebut dan menghasilkan ukuran program yang sesuai dengan kebutuhan.

Contoh untuk membuat tabel karyawan beserta field-fieldnya :

```
CREATE TABLE Karyawan(  
    NIK nchar(10) Not Null,  
    Nama_Kar nchar(35) Not Null,  
    JK nchar(1) Not Null,  
    Alamat nchar(100) Null,  
    Kota nchar(26) Null)
```

Untuk membuat/menambahkan primary key pada tabel karyawan dapat menggunakan fungsi ALTER dan CONSTRAINT.

4. ALTER TABLE

Perintah alter table digunakan untuk memodifikasi sebuah table. Berfungsi untuk merubah, menambahkan, dan atau menghapus field yang ada didalam tabel.

Bentuk penulisan (*Syntax*) :

Perintah untuk membuat/menambahkan primary key pada tabel :

```
ALTER TABLE nama_tabel  
ADD CONSTRAINT nama_constraint tipe_constraint  
(nama_kolom)
```

Perintah untuk menambahkan field :

```
ALTER TABLE nama_tabel  
ADD nama_kolom datatype(size)
```

Perintah untuk menghapus field :

```
ALTER TABLE nama_tabel  
DROP COLUMN nama_kolom
```

Perintah untuk menghapus constraint :

```
ALTER TABLE nama_tabel  
DROP CONSTRAINT nama_constraint
```

Perintah untuk menambahkan check constraint :

```
ALTER TABLE nama_tabel  
ADD CONSTRAINT nama_check_constraint CHECK (kriteria  
yang akan dicheck)
```

5. DROP TABLE

Perintah drop table digunakan untuk menghapus sebuah table.

Bentuk penulisan (*Syntax*) :

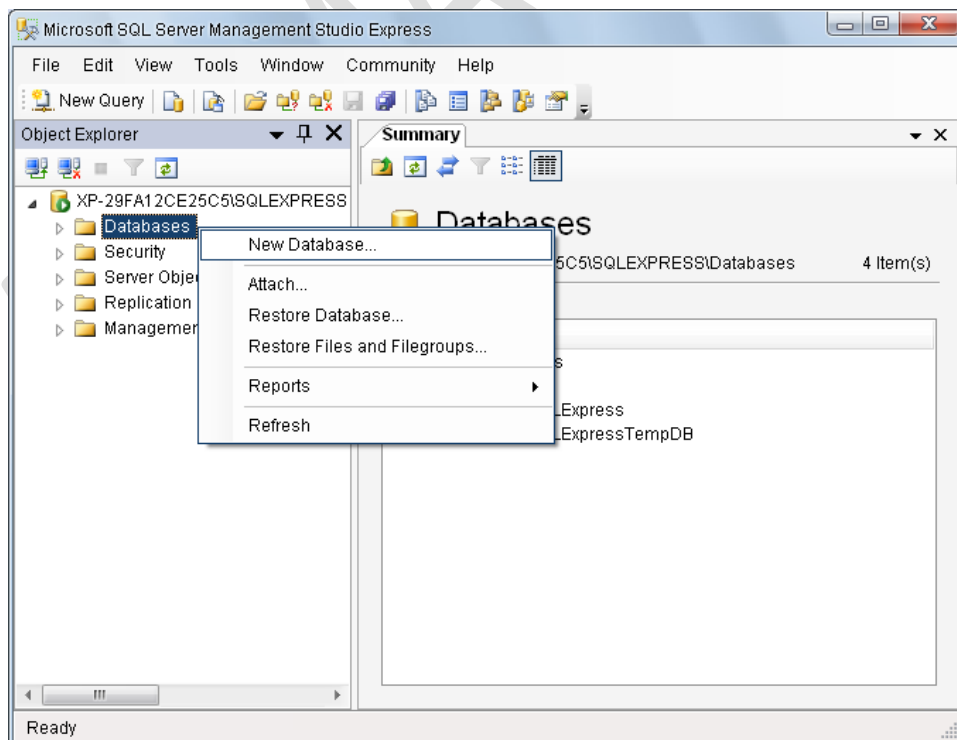
```
DROP TABLE nama_tabel
```

Contoh penggunaan fungsi DROP tabel :

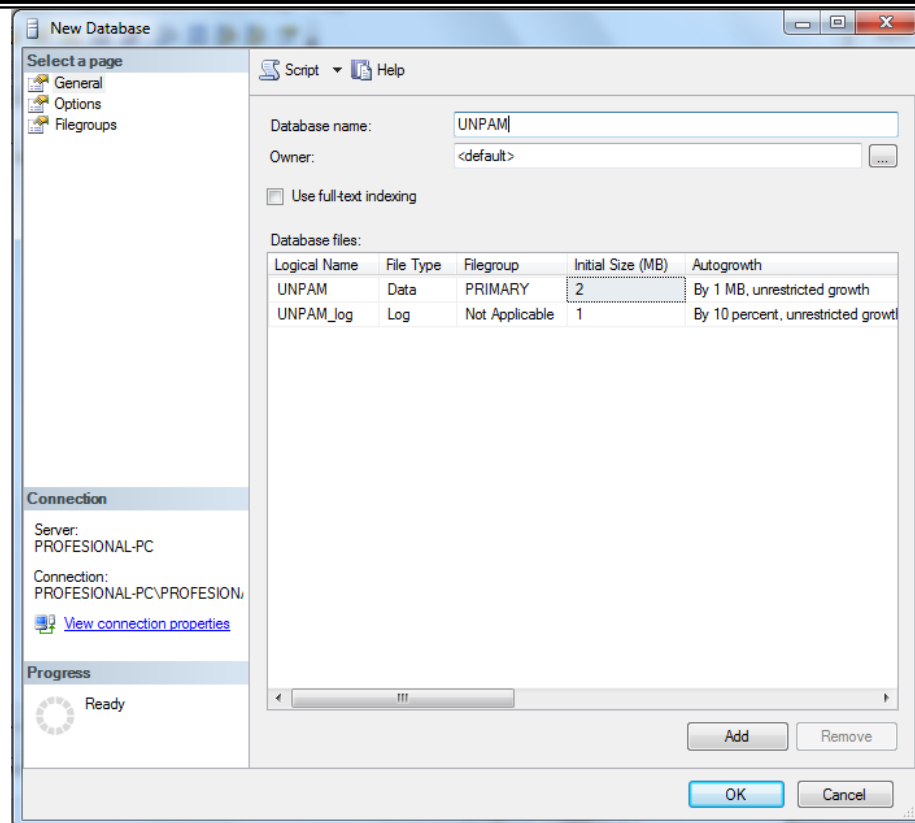
```
DROP TABLE Karyawan
```

II. Membuat Database Dengan SSMS

- Klik Start -> (All) Programs -> Microsoft SQL Server 2005 -> SQL Server Management Studio
- Pada Connect to Server, sesuaikan dengan koneksi pada komputer local atau server anda, kemudian klik Connect
- Pada Object Explorer, klik kanan pada Database, kemudian pilih New database



Gambar. Kotak Dialog Klik Kanan New Database



Gambar. Kotak Dialog New Database

Kemudian isikan nama database (UNPAM) dan konfigurasi file database tersebut setelah itu klik OK.

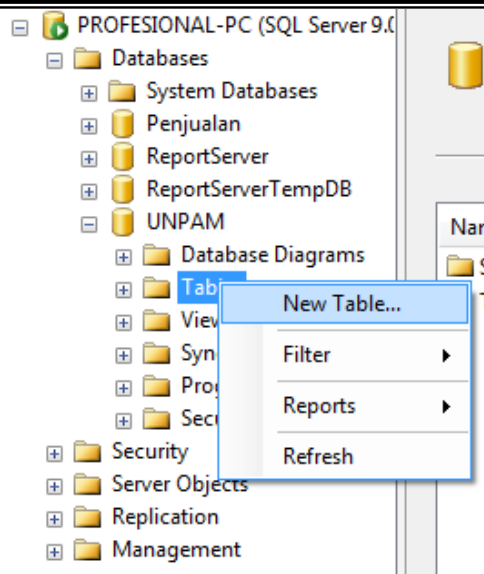
Perintah dengan query :

```
CREATE DATABASE UNPAM
```

III. Membuat Tabel

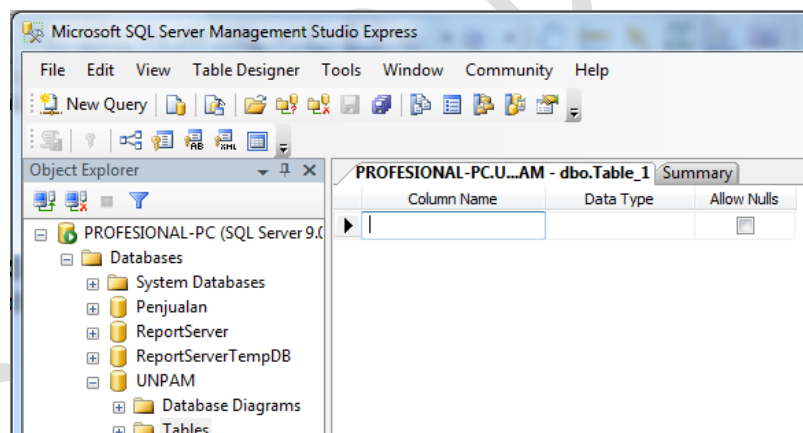
Berikut cara membuat tabel mahasiswa dengan menggunakan SQL Server Management Studio.

- Klik kanan pada tabel database UNPAM
- Klik New Table



Gambar. Kotak Dialog Klik New Table

- Kemudian tampil kotak dialog untuk mengisi field-field pada tabel



Gambar. Form isian tabel

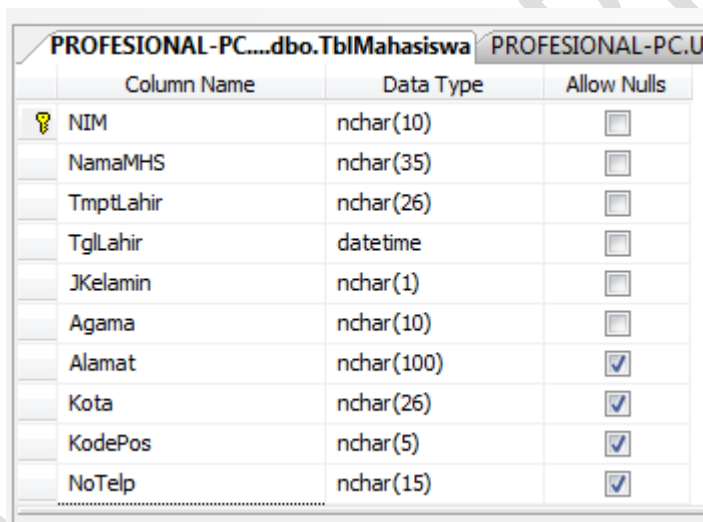
Perintah dengan query :

```
CREATE DATABASE UNPAM
```

- Langkah selanjutnya adalah mengetik nama field pada tabel mahasiswa sebagai berikut :

Tabel. Struktur Tabel Mahasiswa

Column Name	Data Type	Allow Nulls
NIM	nchar(10)	
NamaMHS	nchar(35)	
TmptLahir	nchar(26)	
TglLahir	Datetime	
Jkelamin	nchar(1)	
Agama	nchar(10)	
Alamat	nchar(100)	√
Kota	nchar(26)	√
KodePos	nchar(5)	√
NoTelp	nchar(15)	√



Gambar. Desain Tabel Mahasiswa

- Klik kanan pada field NIM kemudian pilih Set Primary Key untuk dijadikan sebagai kunci.
- Kemudian simpan tabel berikut dengan mengklik tombol **Save** dan ketikkan nama tabel tersebut dengan **TbIMahasiswa** lalu klik **OK**.

Perintah dengan query :

```
CREATE TABLE TbIMahasiswa (
    NIM nchar(10) NOT NULL,
    NamaMHS nchar(35) NOT NULL,
    TmptLahir nchar(26) NOT NULL,
    TglLahir datetime NOT NULL,
    Jkelamin nchar(1) NOT NULL,
    Agama nchar(10) NOT NULL,
    Alamat nchar(100) NULL,
    Kota nchar(26) NULL,
    KodePos nchar(5) NULL,
    NoTelp nchar(15) NULL
)
```

IV. Membuat Primary Key Constraint

Perintah dengan query :

```
ALTER TABLE TblMahasiswa  
ADD CONSTRAINT PK_Mahasiswa Primary key (NIM)
```

V. Menghapus Constraint

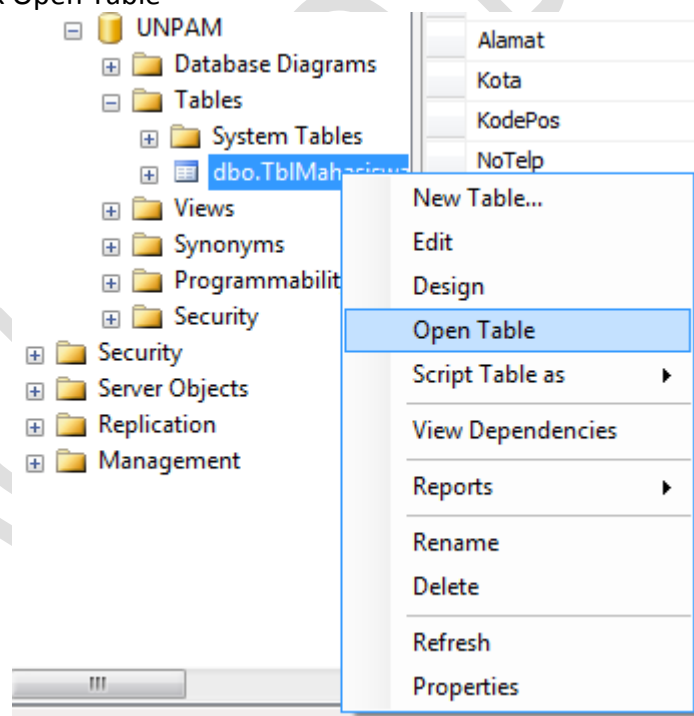
Perintah dengan query :

```
ALTER TABLE TblMahasiswa  
DROP CONSTRAINT PK_Mahasiswa
```

VI. Mengisi Data Pada Tabel Mahasiswa

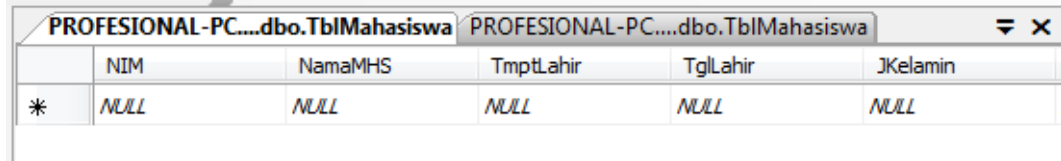
Untuk melakukan pengisian data pada tabel mahasiswa yang telah terbentuk, berikut langkah-langkahnya :

- Klik kanan pada TblMahasiswa
- Klik Open Table



Gambar. Kotak Dialog Open Tabbel Mahasiswa

- Kemudian tampil dialog untuk mengisi data pada tabel mahasiswa



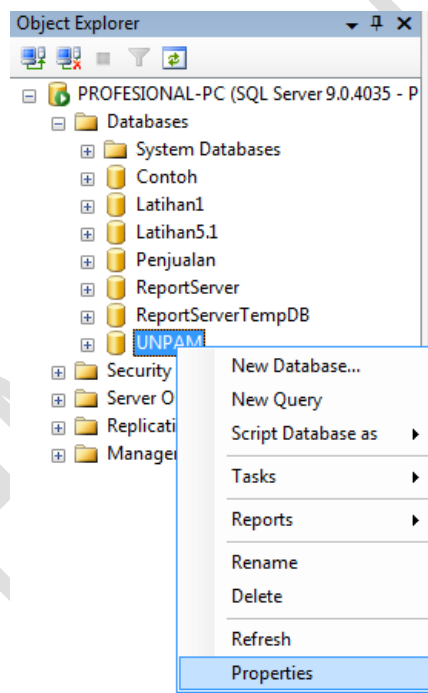
PROFESIONAL-PC....dbo.TbIMahasiswa					
	NIM	NamaMHS	TmptLahir	TglLahir	JKelamin
*	NULL	NULL	NULL	NULL	NULL

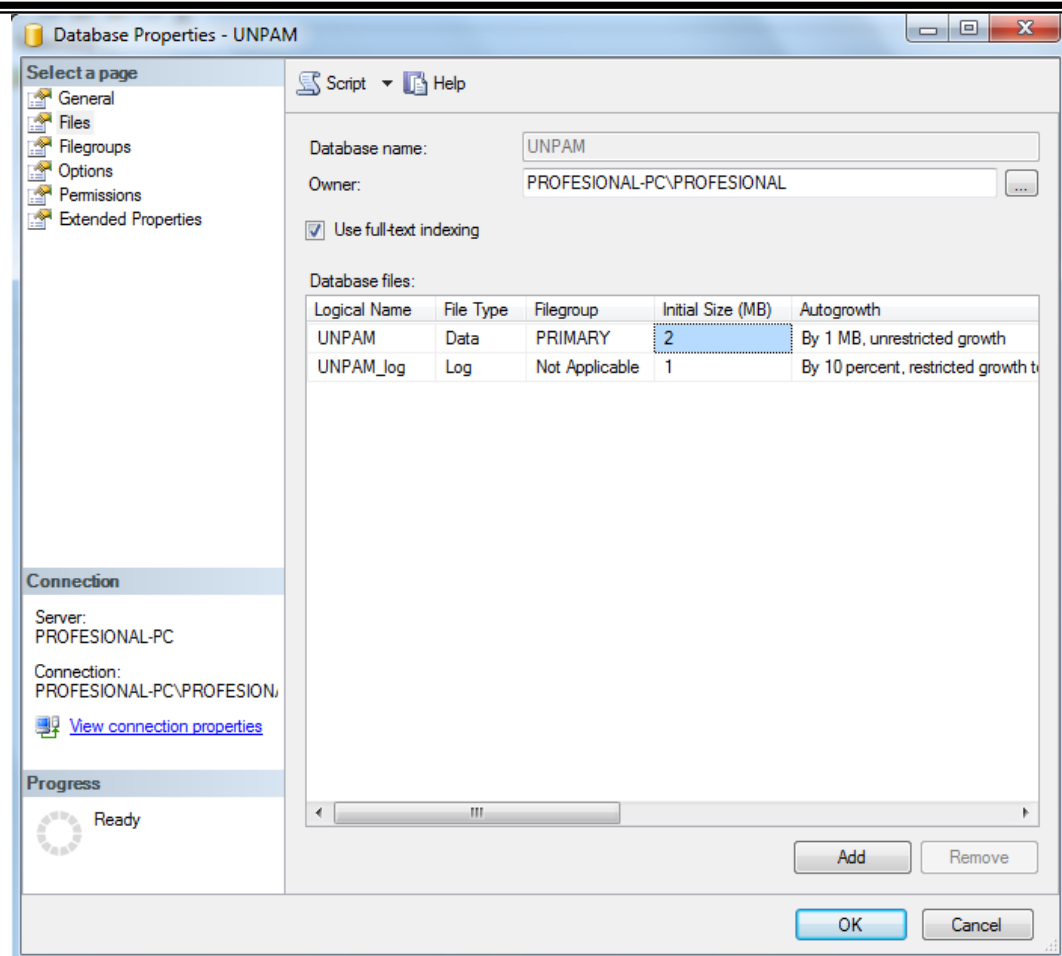
Gambar. Form Isian Data Pada Tabel Mahasiswa

VII. Modifikasi Database

Database yang telah dibuat dapat dimodifikasi kembali dengan cara :

- Klik kanan pada salah satu database
- Klik properties
- Kemudian anda dapat memodifikasi sesuai kebutuhan, hal ini hampir serupa dengan membuat database baru.

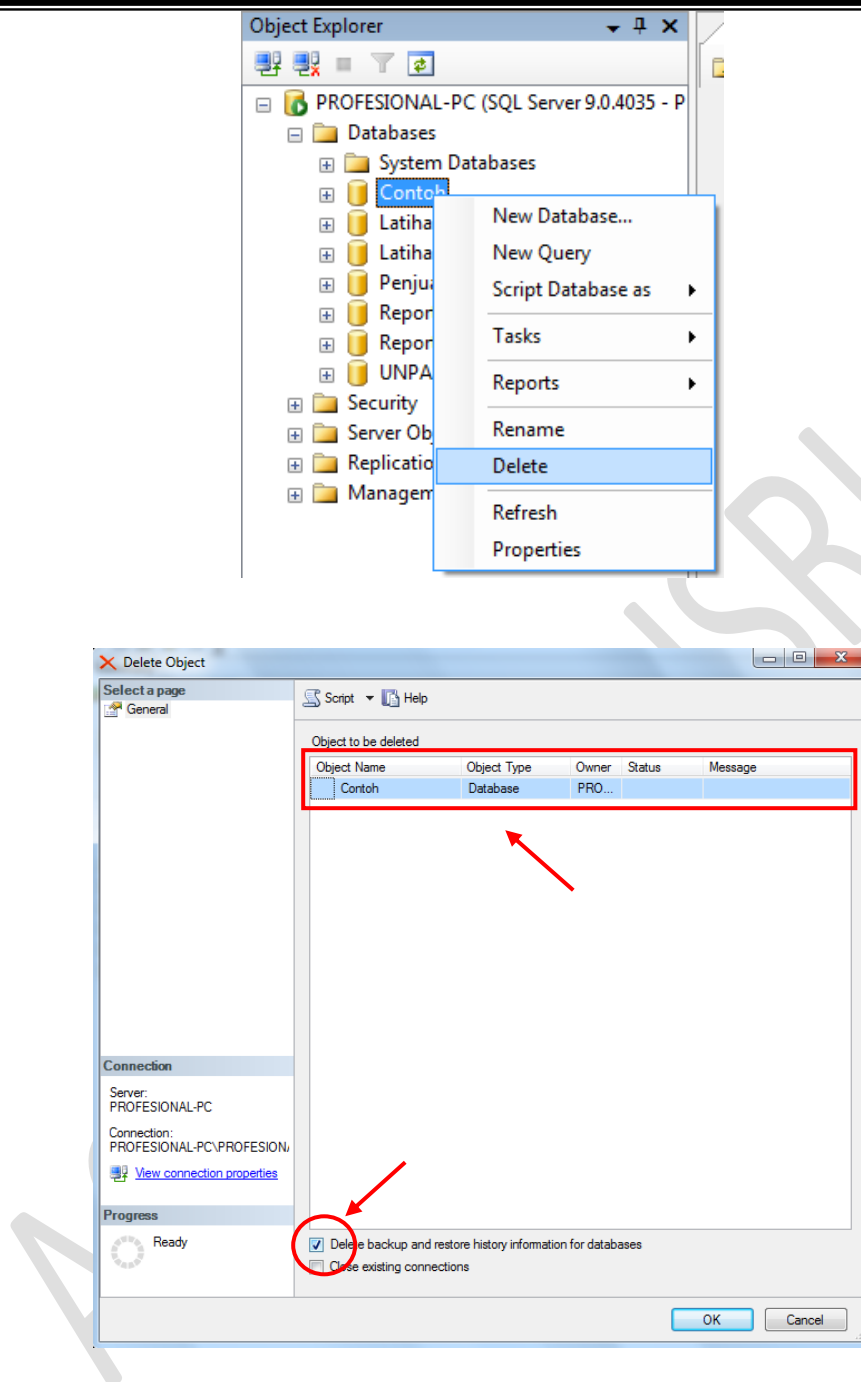




VIII. Menghapus Database

Selain modifikasi anda juga dapat menghapus database dengan cara :

- Klik kanan pada salah satu database
- Kemudian pilih delete
- Perhatikan jendela konfirmasi yang tampil
- Pastikan object name dan object type
- Checklist delete backup and restore history information for databases (untuk menghapus backup database msdb)
- Kemudian klik OK



Perintah dengan query :

```
DROP DATABASE Contoh
```

IX. Latihan 4.1

Untuk latihan berikut gunakanlah dengan bahasa query (DDL) :

- Buatlah sebuah database dengan nama **Latihan4**
- Kemudian buatlah tabel dengan nama **Buku** dengan field sbb :

Column Name	Data Type	Allow Nulls
KodeBuku	nchar(5)	
JudulBuku	nchar(100)	
Pengarang	nchar(35)	
Penerbit	nchar(35)	
TahunTerbit	nchar(4)	
JmlBuku	int	

- Buat primary key constraint dengan nama _constraint yaitu **PK_Buku**
- Tambahkan kolom/field baru dengan nama **ISBN** dengan tipe data nchar(30)
- Tambahkan kolom/field baru dengan nama **Status** dengan tipe data int (diisi 0 atau 1)
- Tambahkan kolom/field baru dengan nama