

## 3. MANAGE TABLES

---

### Obyektif :

Setelah menyelesaikan bahasan ini, diharapkan dapat melakukan hal berikut:


1. Membuat Tabel melalui Enterprise Manager
  2. Menyebutkan dan menjelaskan tipe-tipe data dalam SQL Server
  3. Menjelaskan tentang Database Diagram
  4. Membuat hubungan antar table
- 

### 3.1. Definisi Tabel

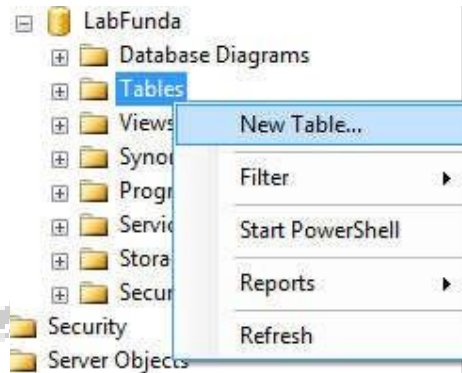
Tabel merupakan blok dasar yang paling umum digunakan dalam sistem basis data, atau disebut juga dengan relasi. Tabel adalah unit dasar penyimpanan yang terdiri dari baris dan kolom. Komponen tabel terdiri dari beberapa kolom yang ditandai dengan **jenis atribut**. Perpotongan antara baris dan kolom disebut **nilai atribut**. Tujuan penggunaan tabel adalah untuk menyederhanakan logika pandangan terhadap data.

### 3.2. Proses Pembuatan Tabel melalui Management Studio

Berikut ini langkah-langkah pembuatan tabel pada sql server 2008

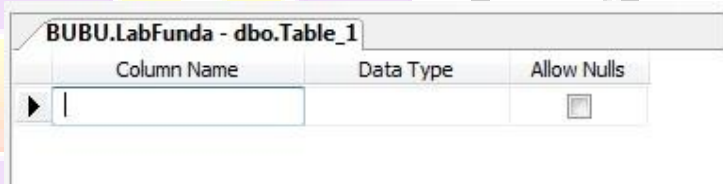
1. Klik  pada menu bar sql server 2008.
2. Kemudian akan muncul Query Editor. Query Editor adalah tempat untuk mengetikkan query.
3. Sebelum membuat tabel, buat database terlebih dahulu dengan nama LabFunda.
4. Gunakan Database LabFunda dengan perintah:  
**use [nama database].**

5. Klik Subfolder LabFunda →Klik Subfolder Table.



**Gambar 3.1 Tampilan Isi Database LabFunda**

6. Klik Kanan Mouse → Pilih New Table  
7. Kemudian akan muncul tampilan seperti gambar 3.2.



**Gambar 3.2. Jendela Pengisian Kolom**

Berikut adalah penjelasan setiap kolom dan jendela tersebut:

Column Name	Menunjukkan nama untuk kolom atau field Data
Type	Menunjukkan tipe data yang disimpan oleh kolom
Allow Nulls	Mengizinkan nilai Null dalam Field.

8. Masukkan kolom-kolom tersebut kemudian simpan dengan cara klik tombol *save table* (pada icon disket) dan ketikkan nama untuk tabel tersebut.

### 3.3. Menggunakan Tabel

Operasi-operasi yang dilakukan pada tabel :

- a. Mengisi atau Melihat Tabel
- Setelah membuat tabel, selanjutnya kita masukkan data-datanya.
  - Ketikkan query sebagai berikut:  
Insert into [nama tabel] values [isi data kolom 1, isi data kolom 2, ....]

- Setelah masukkan data-data ke dalam tabel, lihat isi data tersebut dengan cara :

`select * from [nama tabel]`

Ketikkan query tersebut pada Query Editor.

b. Mengganti Nama Tabel

- Sebelum mengganti nama tabel, pastikan tabel yang akan di rename sedang tidak digunakan.
- Buka Sub Folder Tables → Klik kanan pada tabelnya → pilih Rename.

c. Menghapus Tabel

- Sebelumnya menghapus tabel, pastikan tabel yang akan di hapus sedang tidak digunakan.
- Buka Sub Folder Tables → Klik kanan pada tabelnya → pilih Delete.
- Kemudian akan muncul kotak dialog Delete Object → Klik OK.

- **Kolom Pengenal**

Kolom jenis ini hanya mengandung satu nilai yang mengenali satu baris dari table. Kolom jenis ini ideal untuk pembuatan record secara otomatis. Kemudian tipe data dari kolom ini adalah Int atau Smallint, dan tidak mengizinkan nilai null.

- **Nilai Null**

Menentukan bahwa kolom tidak dapat menerima nilai null. Namun, nilai null bukan nilai 0, tetapi nilai null adalah yang tidak diisi atau tidak terdefinisi.

- **Primary Key**

Satu atau lebih kolom pada tabel yang memiliki baris yang eksklusif di dalam tabel akan membentuk primary key. Spesifikasi dari primary key memastikan integritas dari tabel. Kolom yang membentuk primary key tidak bisa mengandung nilai null. Apabila menentukan nilai sebagai primary, SQL Server akan membuat indeks tertentu untuk memastikan bahwa kunci tersebut bersifat

eksklusif. Tabel yang merupakan primary key ditandai dengan adanya tanda kunci disamping kolom.

- **Foregin Key**

Menentukan hubungan antara tabel-tabel. Foreign Key adalah field di table aktif yang menunjukan ke field kunci pada tabel lain. Penggunaan foreign key mampu mencegah penghapusan baris-baris pada tabel yang aktif, apabila ada referensi field-field kunci dari tabel eksternal. Batasan semacam ini mencegah nilai yang dimasukkan ke dalam tabel yang tidak ditentukan pada field kunci dari tabel eksternal yang digabungkan.

1. **Batasan Unique**

Memaksa kolom-kolom memiliki nilai yang eksklusif. Dengan kata lain kolom yang di beri batasan unique tidak boleh memiliki nilai yang sama.

- **Batasan Check**

Membatasi nilai yang bisa diletakkan ke dalam kolom dengan menguji suatu kondisi.

### 3.4. Menyunting Data di Dalam Tabel

- a. Menambah Kolom Baru

- Buka Folder Databases → Buka LabFunda.
- Ketikkan use LabFunda pada Query Editor.
- Buka Sub Folder Tables → Pilih tabel yang akan digunakan
- Klik kanan folder Columns → Pilih New Column

- b. Mengganti Tipe Data Kolom

- Buka Folder Databases → Buka LabFunda.
- Ketikkan use LabFunda pada Query Editor.
- Buka Sub Folder Tables → Pilih tabel yang akan digunakan
- Buka Folder Columns → Klik kanan pada kolom yang akan diubah tipe datanya → Pilih Modify.

- c. Menghapus Kolom dari Tabel

- Buka Folder Databases → Buka LabFunda. □ Ketikkan use LabFunda pada Query Editor.
- Buka Sub Folder Tables → Pilih tabel yang akan digunakan.
- Buka Folder Columns → Klik kanan pada kolom yang akan diubah tipe datanya → Pilih Delete.

### 3.5. Tipe Data

Setiap kolom di dalam tabel harus memiliki tipe data tertentu. Dengan kata lain, jenis data tertentu harus disimpan seperti misalnya teks atau angka. SQL Server bisa menggunakan beberapa tipe data dan juga mengizinkan user membuat tipe data sendiri.

Pada SQL Server 2008 terdapat beberapa tipe data baru yang tidak ada pada versi SQL Server sebelumnya. SQL Server 2008 memiliki lebih dari 35 tipe data yang dikelompokkan ke dalam 7 kategori yaitu: Exact Numbers, Approximate Numbers, Strings, Date and Times, Unicode Character, Binary Stings, and Other data types. Kategori tipe data SQL Server 2008 adalah sebagai berikut :

#### Exact Number

Exact Number	Keterangan
Decimal	-10 <sup>38</sup> s/d 10 <sup>38</sup> -1
Numeric	Sama dengan decimal. Fungsinya adalah untuk mendefinisikan angka pecahan baik fixed desimal ataupun floating point
Integer	Keterangan
Bit	Integer dengan nilai 0 atau 1
Int	-2.147.483.648 s/d 2.147.483.648
BigInt	Mirip dengan Int. Hanya saja nilai yang diterima lebih besar dari pada tipe data INT. Tipe data ini dapat menerima nilai mulai dari 263 hingga 263-1
SmallInt	-32.768 s/d 32.767
TinyInt	0 s/d 255

Money	Keterangan
Money	-922.377.203.685.477,5808 s/d 922.337.203.685.477,5807
Small Money	-214.748,3648 s/d 214.748,3647

#### Approximate Number

Approximate Number	Keterangan
Float	-214.748,368 sampai 1.79E+308
Real	-3.40E+308 sampai 3.40E+38

#### Strings

String	Keterangan
Char	Jumlah karakter tetap dengan ukuran maksimal 8000 karakter
Varchar	Jumlah karakter variabel dengan ukuran maksimal 8000 karakter
Text	Jumlah karakter variabel dengan ukuran maksimal 2.147.483.647 karakter

#### Dates and Times

Dates and Times	Keterangan
DateTime	1 Januari 1753 s/d 31 Desember 9999
SmallDateTime	1 Januari 1900 s/d 6 juni 2079

#### Unicode Character

Unicode String	Keterangan
Nchar	Jumlah karakter tetap dengan ukuran maksimal 4000 karakter
Nvarchar	Jumlah karakter variabel dengan ukuran maksimal 4000 karakter
Ntext	Jumlah karakter variabel dengan ukuran maksimal 1.073.741.823 karakter

#### Binary String

Binary String	Keterangan
Binary	Jumlah biner tetap dengan ukuran maksimal 8000 bytes
Varbinary	Jumlah biner variabel dengan ukuran maksimal 8000 bytes
Image	Jumlah karakter variabel dengan ukuran maksimal 2.147.483.647 bytes

#### Other Data Types

Tipe Data Lainnya	Keterangan
Cursor	Referensi ke kursor
HierarchyID	Sebuah length variable, tipe data system. Tipe data baru di SQL Server 2008
SQL Variant	Digunakan ketika tidak tahu jenis data yang diharapkan

### Other Data Types

Tipe Data Lainnya	Keterangan
TimeStamp	Angka unik yang dikenal database
UniqueIdentifier	Pengenal global eksklusif (unik)
XML	Kumpulan atau koleksi dari String

### 3.6. Unicode Char Strings

Satu masalah dengan tipe data yang menggunakan 1 byte untuk menulis setiap karakter, tipe data tersebut hanya dapat menampilkan 256 karakter berbeda. Ini bisa mencakup penulisan spesifik untuk alphabets yang berbeda seperti alphabets eropa, yang relatif kecil. Mustahil untuk menangani sistem seperti alphabets Kanji Jepang atau Hangul Korea yang mempunyai ribuan karakter.

Unicode adalah karakter internasional yang menampung 16 bit per karakter. SQL Server mendukung tipe data Unicode yang menghilangkan masalah konversi karakter. SQL Server menyediakan tipe data string yang support unicode, yaitu dengan menambahkan huruf 'n' pada tipe-tipe data string. Contohnya : nchar, nvarchar, ntext.

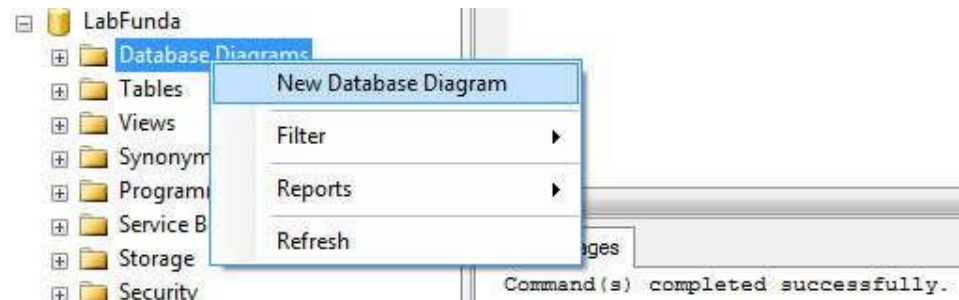
Karena semua sistem unicode secara konsisten menggunakan pola bit yang sama untuk merepresentasikan semua karakter. Tidak ada masalah dengan karakter yang dikonversikan secara salah ketika berpindah dari satu sistem kesistem lainnya. Akan tetapi dengan menggunakan unicode ini maka disk space yang dibutuhkan semakin tinggi. Untuk melihat semua type data yang ada dalam Ms.SQL Server adalah dengan menggunakan perintah stored procedure **sp\_datatype\_info**.

### 3.7. Database Diagram

Diagram database adalah representasi dari sebuah database. Dengan diagram database bisa mengelola database dengan menggunakan antarmuka grafis serta menggeser mouse untuk melakukan tugas-tugas yang biasanya dilakukan oleh perintah bahasa SQL. Tugas-tugas ini meliputi menambah atau menghapus tabel, trigger prosedur, batasan kolom tabel, serta hubungan antar tabel.

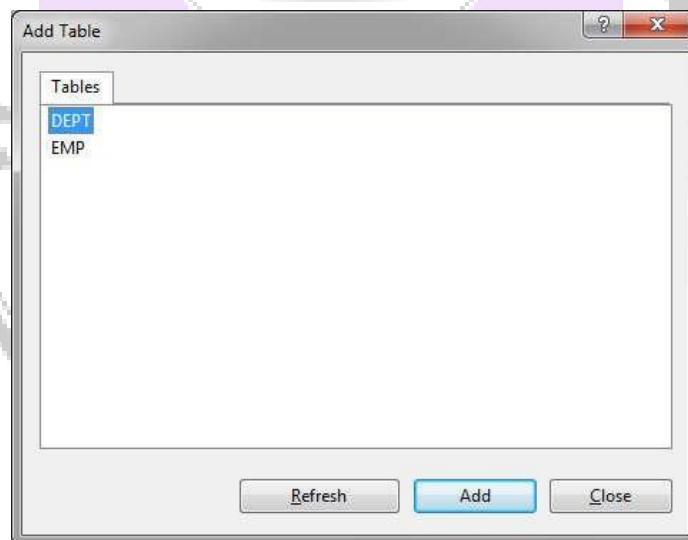
- **Membuat Diagram**

Untuk membuat diagram, buka database LabFunda terlebih dahulu kemudian use LabFunda. Buka database LabFunda → klik kanan Database Diagram → Pilih New Database Diagram.



**Gambar 3.3. Tampilan Pembuatan Database Diagram**

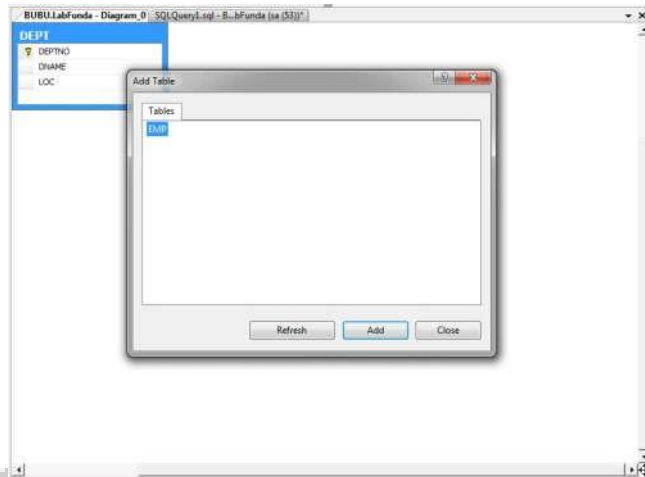
Kemudian akan muncul kotak dialog Add Table → Pilih tabel DEPT → Klik Add.



**Gambar 3.4. Kotak Dialog Add Table**

Jangan ditutup dulu kotak dialognya, kita Add Tabel EMP.





**Gambar 3.5. Kotak Dialog Add Table**

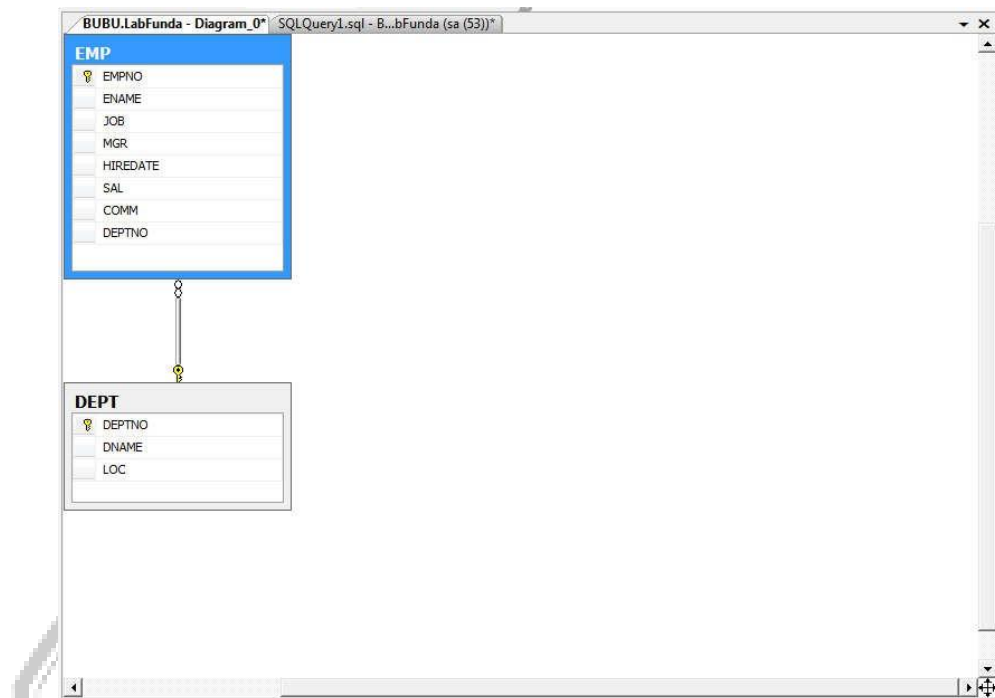
Setelah Add semua tabel, diagram akan ditampilkan dalam daerah kerja. Untuk menyimpan diagram, klik tombol save (pada icon disket) dan ketikkan nama untuk diagram tersebut.

- **Membuat Hubungan Antartabel**

Setelah membuat tabel, hubungan tabel dapat dibuat dengan cara mendrag cursor dari tabel master yang memiliki *primary key* ke tabel tujuan yang memiliki atribut sebagai *foreign key*. Pada saat membuat relasi atau ketika melakukan drag dari tabel parent ke tabel child akan ditampilkan dialog Foreign Key Relationship dan dialog Tables and Columns sekaligus. Secara default dialog Tables and Columns akan terisi yaitu :

- Relationship name, akan terisi dengan nama relasi tabel.
- Primary key table, menunjukkan tabel parent dengan atribut yang menjadi *primary key*.
- Foreign key table, menunjukkan tabel child dengan atribut yang menjadi *foreign key*.

Setelah dipastikan dialog Tables and Columns tersebut terisi, selanjutnya klik tombol OK.



**Gambar 3.6. Hubungan Antar Tabel**