

MUKADDIMAH

Awalnya saya diminta untuk melakukan delete terhadap record yang redundan (duplikat) pada sebuah tabel dalam database. Ini terjadi pada aplikasi yang saya buat dengan Visual basic 6 dan menggunakan Microsoft Access sebagai databasenya. Namun yang menjadi masalah adalah data yang tampil merupakan hasil SELECT yang tertampung pada FlexGrid dan kemudian si user dapat menghapus data yang kebetulan sama satu persatu. Saya coba googling ke beberapa situs dan menemukan sedikit pencerahan. Pada tulisan ini hanya akan dibahas mengenai cara mendelete / hapus record yang sama pada sebuah tabel yang tidak memiliki sebuah indeks unik atau kolom identitas / Key. Saya menggunakan Microsoft SQL Server 2000 untuk melakukan simulasi tulisan ini.

TEORI

Menghapus record (delete) pada sebuah database adalah suatu hal yang biasa dilakukan oleh setiap orang. Dengan perintah DELETE tentu akan selesai masalahnya. DELETE dapat dikelompokkan menjadi perintah Data Manipulation Language (DML) dalam SQL. Ketika sebuah tabel tidak mempunyai Key / Index dan memiliki sejumlah record yang identik / sama, maka tentu tidak semudah itu kita mendelete record yang sama kemudian hanya satu record saja yang tetap ada. Menghilangkan redudansi record pada tabel yang tidak memiliki primary key yang akan kita lakukan pada tulisan ini. Berikut ini ada beberapa kemungkinan yang dapat dilakukan :

1. Menggunakan perintah ROWCOUNT
2. Menggunakan perintah TOP
3. Menggunakan tabel bantu sebagai perantara
4. Menambahkan sebuah kolom sebagai penanda
5. Dan Cari Sendiri :P

Untuk melakukan simulasi, maka kita akan menggunakan data-data seperti nampak pada script di bawah ini.

ASUMSI TABEL

```
CREATE TABLE contoh  
(  
  kode CHAR(5),  
  nama VARCHAR(50)  
)
```

```
INSERT INTO contoh (kode, nama) VALUES ('OS','OPERATING SYSTEM')  
INSERT INTO contoh (kode, nama) VALUES ('HD','HARD DISK')  
INSERT INTO contoh (kode, nama) VALUES ('OD','OPTICAL DRIVE')  
INSERT INTO contoh (kode, nama) VALUES ('OS','OPERATING SYSTEM')  
INSERT INTO contoh (kode, nama) VALUES ('MO','MONITOR')  
INSERT INTO contoh (kode, nama) VALUES ('OS','OPERATING SYSTEM')  
INSERT INTO contoh (kode, nama) VALUES ('HD ','HARD DISK')  
INSERT INTO contoh (kode, nama) VALUES ('MO','MONITOR')  
INSERT INTO contoh (kode, nama) VALUES ('OD','OPTICAL DRIVE')  
INSERT INTO contoh (kode, nama) VALUES ('OD','OPTICAL DRIVE')
```

```
SELECT * FROM contoh
```

	kode	nama
1	OS	OPERATING SYSTEM
2	HD	HARD DISK
3	OD	OPTICAL DRIVE
4	OS	OPERATING SYSTEM
5	MO	MONITOR
6	OS	OPERATING SYSTEM
7	HD	HARD DISK
8	MO	MONITOR
9	OD	OPTICAL DRIVE
10	OD	OPTICAL DRIVE



	kode	nama
1	OS	OPERATING SYSTEM
2	HD	HARD DISK
3	MO	MONITOR
4	OD	OPTICAL DRIVE

Dengan kondisi tabel yang pertama, terlihat beberapa record yang redundan. Pada awal tadi telah disebutkan bahwa dengan kondisi seperti ini, kita tidak dapat melakukan perintah DELETE FROM... seperti contohnya :

```
DELETE FROM contoh
WHERE kode = 'OS' AND nama = 'OPERATING SYSTEM'
```

Perintah di atas akan menyebabkan seluruh record yang kodenya OS dan namanya OPERATING SYSTEM akan terhapus. Oleh karena itu kita akan menjabarkan kemungkinan yang dapat dilakukan untuk menyelesaikan masalah ini sesuai dengan apa yang telah diuraikan pada teori.

ROWCOUNT

Perintah SET ROWCOUNT akan membatasi engine SQL Server untuk memproses sejumlah record secara spesifik. Sehingga proses akan berhenti setelah jumlah limit didefinisikan telah tercapai. Nilai standar yang ada pada SET ROWCOUNT adalah 0 yang berarti tidak ada pembatasan jumlah record yang akan diproses. Setelah perintah ROWCOUNT berjalan dan proses selesai, maka kita dapat mengembalikan nilai default dari ROWCOUNT yakni 0. Berikut sintaks penulisan perintah ROWCOUNT adalah :

```
SET ROWCOUNT { number | @number_var }
```

Sekarang akan dicontohkan penggunaan ROWCOUNT pada tabel contoh, yakni untuk mendelete record yang berulang tiga kali, maka kita akan mengubah nilai ROWCOUNT menjadi 2.

```
SET ROWCOUNT 2
DELETE FROM contoh
WHERE kode='OS' AND nama='OPERATING SYSTEM'
SET ROWCOUNT 0
(2 row(s) affected)
```

Setelah menjalankan perintah DELETE di atas dengan nilai ROWCOUNT adalah dua, maka record dengan kode OS dan nama OPERATING SYSTEM akan ditunjukkan pada tabel berikut.

	kode	nama
1	HD	HARD DISK
2	OD	OPTICAL DRIVE
3	MO	MONITOR
4	OS	OPERATING SYSTEM
5	HD	HARD DISK
6	MO	MONITOR
7	OD	OPTICAL DRIVE
8	OD	OPTICAL DRIVE

Untuk mendelete record yang redundan secara otomatis, kita dapat menggunakan Cursor. Yang perlu menjadi catatan adalah ketika kita mendeklarasikan Cursor pada record yang redundan, maka kita perlu menghitung (COUNT) jumlah record yang redundan dikurangi satu (1), karena kita ingin menyisakan satu record saat proses DELETE terjadi. Berikut ini adalah contoh penggunaan Cursor menggunakan perintah ROWCOUNT.

```

DECLARE @Hitung int
DECLARE @Kode char(5)
DECLARE @Nama varchar(50)

DECLARE kursor_duplikasi CURSOR FAST_FORWARD FOR
SELECT Kode, Nama, COUNT(*) - 1
FROM contoh
GROUP BY Kode, Nama
HAVING COUNT(*) > 1

OPEN kursor_duplikasi

FETCH NEXT FROM kursor_duplikasi INTO @Kode, @Nama, @Hitung

WHILE @@FETCH_STATUS = 0
BEGIN
    SET ROWCOUNT @Hitung
    DELETE FROM contoh WHERE Kode = @Kode AND Nama = @Nama
    SET ROWCOUNT 0
    FETCH NEXT FROM kursor_duplikasi INTO @Kode, @Nama, @Hitung
END

CLOSE kursor_duplikasi
DEALLOCATE kursor_duplikasi

```

Dengan menjalankan perintah di atas, maka tujuan kita telah tercapai, yakni record-record yang redundan pada tabel contoh akan di delete secara otomatis sehingga nampak pada tabel di bawah ini :

	kode	nama
1	OS	OPERATING SYSTEM
2	HD	HARD DISK
3	MO	MONITOR
4	OD	OPTICAL DRIVE

TOP

Metode berikutnya yang dapat kita gunakan untuk menghilangkan redudansi record pada tabel adalah dengan perintah TOP untuk menunjang penggunaan perintah DELETE. Berikut ini sintaks penggunaan TOP :

```
TOP n [PERCENT]
```

Berikut ini contoh penggunaan :

```
DELETE TOP (2) FROM contoh  
WHERE kode = 'OD' AND nama = 'OPTICAL DRIVE'
```

Error Message yang ditampilkan :

```
Server: Msg 156, Level 15, State 1, Line 1  
Incorrect syntax near the keyword 'TOP'.
```

Dengan perintah TOP, kita juga dapat mendelete record yang redundan secara otomatis menggunakan Cursor. Berikut ini adalah contoh penggunaan Cursor menggunakan perintah TOP

```
DECLARE @Hitung int  
DECLARE @Kode char(5)  
DECLARE @Nama varchar(50)  
  
DECLARE kursor_duplikasi CURSOR FAST_FORWARD FOR  
SELECT Kode, Nama, COUNT(*) - 1  
FROM contoh  
GROUP BY Kode, Nama  
HAVING COUNT(*) > 1  
  
OPEN kursor_duplikasi  
  
FETCH NEXT FROM kursor_duplikasi INTO @Kode, @Nama, @Hitung  
  
WHILE @@FETCH_STATUS = 0  
BEGIN  
    DELETE TOP(@Hitung) FROM contoh WHERE Kode = @Kode AND Nama = @Nama  
    FETCH NEXT FROM kursor_duplikasi INTO @Kode, @Nama, @Hitung  
END  
  
CLOSE kursor_duplikasi  
DEALLOCATE kursor_duplikasi
```

Analisa :

Dari hasil menjalankan perintah di atas, keduanya menunjukkan error yang sama. Saya sudah baca manual book SQL Server, kemudian sintaks juga sudah saya coma ganti beberapa kali, namun hasilnya nihil. ??? ☹ Kemungkinan perintah tersebut hanya berlaku pada SQL Server versi 2005 ke atas >>> saya juga belum coba lagi.

Mohon pencerahan dari para Master SQL sekalian...

TABEL BANTU

Metode berikutnya yang dapat digunakan dalam mendelete record yang redundan pada sebuah tabel adalah dengan memanfaatkan table bantu (Temporary Table) yang digunakan untuk menampung hasil SELECT Distinct dari tabel yang redundan. Apabila selesai maka tabel bantu tersebut akan di DROP. Berikut ini contoh penggunaan Temporary Table :

```
SELECT DISTINCT *
  INTO #temp
  FROM contoh

TRUNCATE TABLE contoh

INSERT contoh
  SELECT *
  FROM #temp

DROP TABLE #temp
```

Ada empat tahap yang dijalankan dari perintah di atas :

1. Menyeleksi record yang redundan (SELECT DISTINCT *), kemudian memasukkan hasilnya kedalam tabel bantu (#temp)
2. Mengosongkan / Delete semua record dari tabel contoh
3. Memasukkan data ke tabel contoh hasil seleksi dari tabel #temp
4. Menghapus tabel bantu #temp

Metode ini dapat dikatakan paling sederhana, namun efektif. Namun tidak cocok diterapkan untuk aplikasi yang menggunakan database Microsoft Access.

KOLOM IDENTITAS

Metode selanjutnya adalah dengan menambahkan sebuah kolom baru sebagai penanda untuk membedakan record yang satu dengan record yang lainnya. Dengan asumsi, kita (user) mempunyai hak akses untuk melakukan alter terhadap tabel, jika tidak memiliki hak akses maka metode ini tidak dapat diimplementasikan.

Untuk melakukan cara ini maka kita akan menggunakan perintah ALTER Table untuk menambah sebuah kolom baru dengan tipe data Integer.

```
ALTER TABLE contoh ADD penanda INT IDENTITY(1,1)
```

Setelah menjalankan perintah di atas, maka akan di dapat struktur baru yang nampak pada tabel di bawah ini :

```
SP_HELP contoh
```

	Identity	Seed	Increment	Not For Replication
1	penanda	1	1	0

	kode	nama	penanda
1	OS	OPERATING SYSTEM	1
2	HD	HARD DISK	2
3	OD	OPTICAL DRIVE	3
4	OS	OPERATING SYSTEM	4
5	MO	MONITOR	5
6	OS	OPERATING SYSTEM	6
7	HD	HARD DISK	7
8	MO	MONITOR	8
9	OD	OPTICAL DRIVE	9
10	OD	OPTICAL DRIVE	10

Sekarang kita dapat dengan mudah mendelete record yang redundan pada tabel, misalnya kita akan mendelete record yang kodenya = OD yang ditunjukkan pada record ke tiga.

```
DELETE FROM contoh WHERE penanda IN (9,10)
```

Dengan perintah di atas, maka record dengan kode OD pada baris ke 9 dan 10 akan di delete. Untuk SQL Server versi 2005 ke atas (2000 ke bawah tidak akan diimplementasikan), kita dapat menggunakan Common Table Expression (CTE). Berikut ini contoh penggunaannya :

```
WITH duplikasiCTE(kode, nama, penanda)
AS
(
    SELECT kode, nama, MIN(penanda) penanda
    FROM contoh
    GROUP BY kode, nama
    HAVING COUNT(*) > 1
)
DELETE FROM contoh
WHERE penanda IN (
    SELECT contoh.penanda
    FROM contoh
    INNER JOIN duplikasiCTE ON
    contoh.kode = duplikasiCTE.kode
    AND contoh.nama = duplikasiCTE.nama
    AND contoh.penanda <> duplikasiCTE.penanda
)
```

Setelah selesai bekerja dengan kolom penanda, maka kita dapat mendrop kolom penanda pada tabel contoh dengan perintah :

```
ALTER TABLE contoh DROP COLUMN penanda
```

KODE TIDAK TERDOKUMENTASI

```
/*KODE 1*/
SET ROWCOUNT 1

DELETE contoh
FROM contoh a
WHERE (SELECT COUNT(*) FROM contoh b WHERE b.kode = a.kode) > 1
WHILE @@rowcount > 0
DELETE contoh
FROM contoh a
WHERE (SELECT COUNT(*) FROM contoh b WHERE b.kode = a.kode) > 1

SET ROWCOUNT 0

/*KODE 2*/

SET ROWCOUNT 1

BEGIN TRANSACTION duplikasi
SELECT * FROM contoh
WHILE @@rowcount > 0 DELETE a FROM contoh a WHERE (SELECT COUNT(*) FROM contoh b
WHERE a.kode = b.kode) > 1

SET ROWCOUNT 0
```

MICROSOFT SQL SERVER RELEASE VERSION

Version	Year	Release Name	Codename
4.21	1993	SQL Server 4.21	-
6.0	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	Hydra
7.0	1999	SQL Server 7.0	Sphinx
-	1999	SQL Server 7.0 OLAP Tools	Plato
8.0	2000	SQL Server 2000	Shiloh
8.0	2003	SQL Server 2000 64-bit Edition	Liberty
9.0	2005	SQL Server 2005	Yukon
10.0	2008	SQL Server 2008	Katmai

PENUTUP

Kiranya sampai disini saja tulisan yang tidak jelas ini. Tulisan ini jelas terinspirasi dari hasil Googling di internet dan Manual Books SQL Server 2000. Kritik dan saran silahkan disampaikan ke : <http://prihastomo.wordpress.com>

Wassalam

Yoga Prihastomo