

Part 7 – SQL 2005 Trigger

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://suryodesign.asia>

www.suryodesign.wordpress.com

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

Pada pertemuan ke 7 ini saya akan membahas tentang Trigger pada SQL 2005 , masi banyak orang yang awam akan feature-feature yang dimiliki oleh SQL sendiri , kebanyakan orang menganggap bahwa database hanya sebagai tempat menyimpan data , dan mereka hanya menganggap database dengan 3 gambaran simple dalam penggunaanya seperti, create table , retrieve data (mengambil / menampilkan data) dan manipulasi data (insert , update delete) , padahal banyak sekali fitur yang perlu kita oleh pada SQL 2005 ini seperti pengaturan Constraint yang ada sesuai kebutuhan system , Function , Stored Procedure , View , Index , Statistic , Trigger , Locking Transaction dan banyak lainnya.

Untuk menjadi seorang DBA (Database Administrator) ini merupakan hal dasar yang harus dipelajari , setelah memahami konsep ini , kemudian baru kita menganalisis kebutuhan database kita dan menghitung estimasi dan baru mengambil keputusan database apa yang mesti digunakan di pertimbangkan dengan factor seperti size , cost , fitur , license , dan system yang digunakan untuk menjadi server database itu sendiri.

Isi

Ayo Teriaklah “SEMANGAT” sebelum anda memulai membaca isi dari database ini :) , Oke sekarang langsung saja , apa itu Trigger ? Trigger artinya adalah **pemicu** , atau bahasa kasarnya dapat kita bilang sebagai syntax SQL yang dijalankan ketika ada proses yang di picu , proses disini adalah proses pada Query database , baik untuk retrieve data (show data) atau manipulation data(insert , update , delete) atau kebanyakan orang IT menyebutnya sebagai proses yang di lakukan ketika ada DDL (Data Definition Language dan DML (Data Manipulation Language).

1

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2007 IlmuKomputer.Com

Trigger ini terbagi menjadi 3 yaitu **Before** , **After** , dan **Instead Of**.

Kebanyakan yang umum di pakai adalah Before dan After , maksudnya Trigger Before adalah Trigger yang di lakukan sebelum proses insert/update/delete dilakukan, sedangkan After Trigger adalah Trigger yang di lakukan setelah proses insert/update/delete dilakukan.

Perlu diketahui bahwa dalam sebuah tabel dapat mempunyai beberapa trigger.

Trigger sangat berguna karena dapat secara otomatis dilaksanakan dalam server tanpa perlu kita ketik kembali sehingga dapat menyederhanakan Query yang kita miliki, sekaligus dapat menjaga konsistensi informasi dalam database . Trigger ini pun sudah disupport pada beberapa DBMS yang ada seperti pada SQL Server, MySQL, PostgreSQL, ataupun Oracle ,

Berikut adalah beberapa keuntungan trigger :

- Konsistensi Database , maksudna trigger digunakan terhadap table yang menjadi reference , dimana semua user / admin yang melakukan proses pada table tersebut akan menghasilkan proses yang sama pada database dan dapat menghindari kesalahan yang terjadi karena perbedaan hasil yang ada.
- Mudah Untuk Di Update , maksudnya Trigger yang telah kita buat pada database yang ada dapat kita ubah sewaktu – waktu tanpa mengubah coding dalam aplikasi , dan dengan mengubah trigger ini , maka semua user / client yang ada akan menggunakan trigger yang ada
- Kecepatan. Tentu saja karena trigger yang tersimpan dalam sisi database dan bukan pada sisi client dapat mengurangi bandwitch atau lalu lintas yang ada pada jaringan dan hal ini menyebabkan system kita lebih efisien dalam digunakan.
- Mengurangi Pekerjaan Seorang programmer yang membuat aplikasi dalam mengatur proses yang ada karena proses ini sudah di miliki oleh trigger

Berikut Merupakan Syntax Trigger pada [MSDN](#) :

```
CREATE TRIGGER trigger_name
ON { table | view }
[ WITH ENCRYPTION ]
{
    { { FOR | AFTER | INSTEAD OF } { [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
      [ WITH APPEND ]
      [ NOT FOR REPLICATION ]
      AS
      [ { IF UPDATE ( column )
        [ { AND | OR } UPDATE ( column ) ]
        [ ...n ]
      | IF ( COLUMNS_UPDATED ( ) { bitwise_operator } updated_bitmask )
        { comparison_operator } column_bitmask [ ...n ]
      } ]
      sql_statement [ ...n ]
    }
}
```

Pertama Mari Kita membuat Table **MsBarang** dan **TrPenerimaan** Seperti berikut pada SQL2005 :

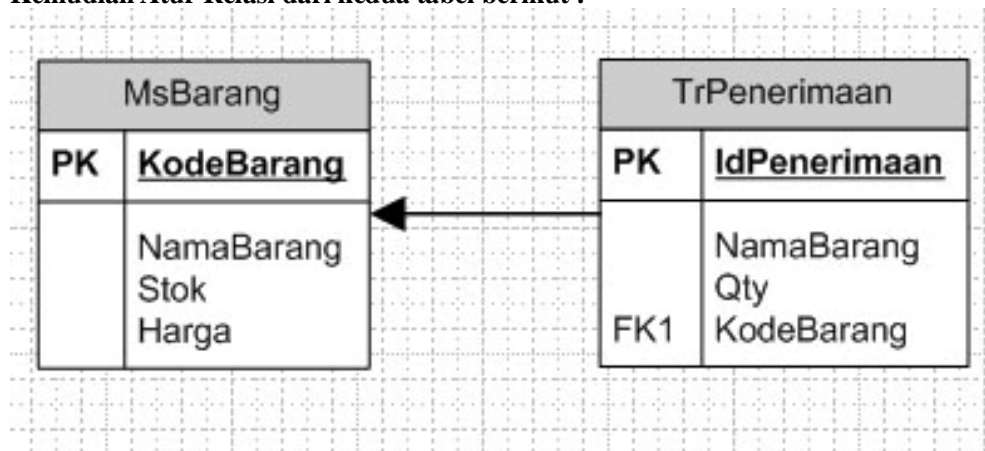
MSBARANG :

SURYODESIGN-PC\...bo.TrPenerimaan		SURYODESIGN-PC	
Column Name	Data Type	Allow Nulls	
KdBarang	int	<input type="checkbox"/>	
NamaBarang	varchar(30)	<input checked="" type="checkbox"/>	
Stok	int	<input checked="" type="checkbox"/>	
Harga	numeric(10, 0)	<input checked="" type="checkbox"/>	

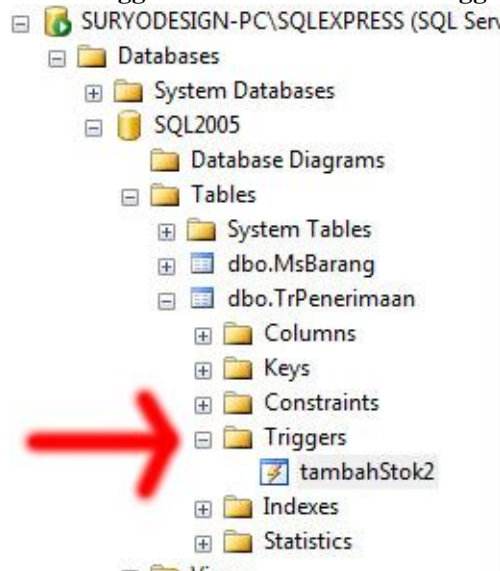
TRPENERIMAAN :

SURYODESIGN-PC\...bo.TrPenerimaan		SURYODESIGN-PC	
Column Name	Data Type	Allow Nulls	
IdPenerimaan	int	<input type="checkbox"/>	
NamaBarang	varchar(30)	<input checked="" type="checkbox"/>	
Qty	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Kemudian Atur Relasi dari kedua tabel berikut :



Oke Sekarang Saatnya kita membuat Trigger pada SQL 2005 , Pilih Tabel **MsBarang** dan Cari Menu **Trigger** Dan Klik Kanan **New Trigger**



Kemudian Kita Buat Proses Trigger Seperti Berikut :

```
USE [SQL2005]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create trigger [dbo].[tambahStok2]
on [dbo].[TrPenerimaan]
for INSERT
as
begin
SET NOCOUNT ON
declare @NamaBarang varchar(30) , @Qty int
select @NamaBarang = NamaBarang , @Qty = Qty from TrPenerimaan
update MsBarang set Stok = (STok + @Qty) where NamaBarang = @Namabarang
end
```

Saya Akan menjelaskan secara singkat pada syntax di atas

1. Gunakan Database SQL 2005
2. Create Trigger dengan Instance [dbo] dan nama trigger [tambahStok2]
3. Trigger dilakukan Ketika ada Proses insert
4. Deklarasikan variable dan type yang digunakan
5. Isi Variabel yang kita gunakan sesuai dengan variable yang kita input
6. Lakukan Proses Trigger Update pada table MsBarang untuk menambah jumlah Stok.

Setelah Proses Trigger Selesai maka kita mesti pastikan proses trigger kita jalan sebelum kita implementasikan pada aplikasi. Coba tambahkan Penerimaan dengan proses penerimaan barang dengan nama yang ada pada table MsBarang dan pastikan proses tersebut success untuk diinsert dan kemudian lihat table msBarang

```
insert into TrPenerimaan(NamaBarang,Qty) values('Gelas','500000')
```

Pastikan bahwa Stok pada MsBarang bertambah sesuai dengan penerimaan

```
select * from TrPenerimaan  
select * from MsBarang
```

Setelah Semua Proses Jalan maka sekarang kita akan mencoba meng-implementasikannya pada C#.Net , aplikasi ini merupakan aplikasi lanjutan dari Pertemuan 6 , bila anda belum memiliki aplikasi pada pertemuan 6 anda dapat mendownloadnya disini [http://www.suryodesign.asia/freedownload/C_Sharp_Part6%20-%20SQL2005%20Stored%20Procedure%20\(2\).zip](http://www.suryodesign.asia/freedownload/C_Sharp_Part6%20-%20SQL2005%20Stored%20Procedure%20(2).zip)

sekarang mari kita design aplikasi dengan Form Berikut , gunakan komponen DataGridView, ComboBox , TextBox , Button , Label , dan SplitContainer

Setelah Proses Design Aplikasi Selesai Sekarang kita akan mencoba untuk meload Nama Barang Ke Dalam Combobox , disini saya mempraktikkan 2 Cara untuk meload ke dalam Combo Box , ada banyak cara seperti binding , gunakan datasource , wizard dan manual additem. Tapi disini saya menerapkan 2 Cara yang simple saja , anda dapat berkreasi sendiri sesuai dengan logika anda untuk mengasah kemampuan anda sendiri :

Cara 1 Load Data ke dalam ComboBox

```
SqlCommand scom = new SqlCommand();
scom.Connection = con;
scom.CommandType = CommandType.Text;
scom.CommandText = "select NamaBarang from msbarang";

DataSet ds1 = new DataSet();
SqlDataAdapter sda = new SqlDataAdapter(scom);
sda.Fill(ds1, "barang");
cmbNamaBarang.DataSource = ds1.Tables["barang"] ;
cmbNamaBarang.DisplayMember = "NamaBarang";
```

Cara 2 Load Data ke Dalam ComboBox

```
SqlDataReader dr;
SqlCommand scom = new SqlCommand();
scom.Connection = con;
scom.CommandType = CommandType.Text;
scom.CommandText = "select * from msbarang";
con.Open();
DataSet ds1 = new DataSet();
SqlDataAdapter sda = new SqlDataAdapter(scom);
sda.Fill(ds1, "barang");
dr = scom.ExecuteReader();
while(dr.Read())
{
    cmbNamaBarang2.Items.Add(dr.GetValue(1));
}
cmbNamaBarang2.SelectedIndex = 0;
con.Close();
```

Setelah Proses Load data ke dalam Combo Box Selesai maka kita akan Load Data TrPenerimaan ke dalam DataGrid , berikut codingnya :

```
SqlCommand scom = new SqlCommand();
scom.Connection = con;
scom.CommandType = CommandType.Text;
scom.CommandText = "select * from trPenerimaan";

DataSet ds2 = new DataSet();
SqlDataAdapter sda = new SqlDataAdapter(scom);
sda.Fill(ds2, "transaksi");
dataGridTransaksi.DataSource = ds2;
dataGridTransaksi.DataMember = "transaksi";
```

Kemudian disini saya membuat dengan Split Container , dimana pada kolom kiri dan kanan ada sama2 tombol submit , anda dapat menggunakan yang mana saja untuk insert data , perbedaan pada cara insert data ke dalam combobox menyebabkan proses pengambilan data dari combobox itu sendiri dengan metode pengambilan yang berbeda.

Berikut Button Tambah data pada Left Container

```
string a = cmbNamaBarang.Text;
if (txtQty.Text == "")
//gunakan OR untuk argumen membandingkan
{
    MessageBox.Show("Qty mesti diisi");
    //Environment.Exit(-1);
    goto berhenti;
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
}

int num;
//buat variabel num
bool isNum = int.TryParse(txtQty.Text.ToString(), out num);
//membuat variabel is Num dan kemudian isi dari variabel isNum itu sendiri
//mengubah type data dan menyimpan hasilna pada variabel num
if (!isNum)
//mengecek nilai isNum apakah false
//( karena bukan number melainkan alfabet )
{
    MessageBox.Show("Qty mesti number");
    //bila false maka tampilkan pesan harga mesti number
    goto berhenti;
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
}

con.Open();
//open koneksi
SqlCommand scom = new SqlCommand();
//membuat SQLCOMMAND dengan alias scom
scom.Connection = con;
//include koneksi ke dalam sqlcommand
scom.CommandText = "insert into TrPenerimaan(NamaBarang,Qty) values('" +
a + "','" + txtQty.Text + "')";
scom.CommandType = CommandType.Text;
//menentukan jenis command , defaultnya adalah commandtype.text
scom.ExecuteNonQuery();
//jalankan storedProcedure
con.Close();
//akhiri koneksi ( tutup koneksi )
btnShowTr.PerformClick();
tabControl1.SelectedIndex = 0;
btnShow.PerformClick();
tabControl1.SelectedIndex = 1;
//jalankan btnshow untuk refresh data
berhenti:
    Console.WriteLine("STOP");
```

. Berikut Button Tambah data pada Right Container

```
string a = Convert.ToString(cmbNamaBarang2.SelectedItem);
if (txtQty2.Text == "")
//gunakan OR untuk argumen membandingkan
{
    MessageBox.Show("Qty mesti diisi");
    //Environment.Exit(-1);
    goto berhenti;
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
}
int num;
//buat variabel num
bool isNum = int.TryParse(txtQty2.Text.ToString(), out num);
//membuat variabel isNum dan kemudian isi dari variabel isNum itu sendiri
//mengubah type data dan menyimpan hasilna pada variabel num
if (!isNum)
//mengecek nilai isNum apakah false
//( karena bukan number melainkan alfabet )
{
    MessageBox.Show("Qty mesti number");
    //bila false maka tampilkan pesan harga mesti number
    goto berhenti;
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
}
con.Open();
//open koneksi
SqlCommand scom = new SqlCommand();
//membuat SQLCOMMAND dengan alias scom
scom.Connection = con;
//include koneksi ke dalam sqlcommand
scom.CommandText = "insert into TrPenerimaan(NamaBarang,Qty) values(' " +
a + "', '" + txtQty2.Text + "')";
scom.CommandType = CommandType.Text;
//menentukan jenis command , defaultnya adalah commandtype.text

scom.ExecuteNonQuery();
//jalankan storedProcedure
con.Close();
//akhiri koneksi ( tutup koneksi )
btnShowTr.PerformClick();
tabControl1.SelectedIndex = 0;
btnShow.PerformClick();
tabControl1.SelectedIndex = 1;
//jalankan btnshow untuk refresh data
berhenti:
Console.WriteLine("STOP");
```


Untuk Aplikasi Secara lengkap anda dapat mendownloadnya di : www.suryodesign.asia/freedownload , sampai jumpat pada artikel selanjutnya dengan materi pembuatan program yang efisien dengan menggunakan class dan locking transaction untuk menghindari kesalahan yang terjadi.

Penutup

Jangan Pernah menyerah apapun masalah yang anda hadapi , ingat pesan Thomas Edison yang berbunyi “Saya hanya menemukan 100000 cara yang tidak bekerja , dan bukan berarti saya gagal dalam menyelesaikannya” , terus berusaha dan keep improve dan jangan lupa sumbangkan ilmu anda pada komunitas yang anda , kaitkanlah ilmu selalu dengan membaca dan menuangkannya pada komunitas yang ada.

Referensi

- MSDN 2005
- <http://msdn.microsoft.com>
- www.gotdotnet.com
- <http://social.msdn.microsoft.com/>
- www.vbdotnetforum.com
- <http://www.c-sharpcorner.com>
- <http://www.dotnetspider.com>

Biografi Penulis



M.Suryo Pranoto – Alumni Mahasiswa Perguruan Tinggi Universitas Bina Nusantara , Aktif dalam beberapa komunitas komputer , dan beberapa project terutama berbasis aplikasi seperti VB.Net atau C#,

dan sedang berusaha keras untuk menabung dan berencana untuk melanjutkan Cisco CCNP setelah menyelesaikan CCNA , memiliki hobby untuk sharing mengenai komputer mulai dari software hingga jual beli hardware maupun modding & overclocking computer.

Ym : suryolovetyka

Email : suryodesign@yahoo.co.id

Website : <http://www.suryodesign.asia>

Blog : www.suryodesign.wordpress.com