

Desain Table Database Menggunakan Excel

Eko Heri Susanto

ekoheri@gmail.com

<http://www.jawadwipa.com>

Lisensi Dokumen:

Copyright © 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.

Dalam tulisan kali ini saya mencoba membahas tips untuk mempermudah dalam melakukan desain tabel database melalui Microsoft Excel. Tulisan ini sudah pernah dimuat di <http://www.bengkelprogram.com>. Atas seijin web masternya dan tanpa mengurangi isi didalamnya saya mencoba untuk menuliskan ulang artikel ini. Untuk melihat artikel versi webnya silakan buka halaman web tersebut diatas.

Salah satu pekerjaan utama seorang system analyst ketika membangun sebuah sistem database adalah mendesain table. Dalam proses desain tersebut tidak jarang seorang system analyst menggunakan tools untuk mempermudah pekerjaannya. Banyak sekali tools yang bisa digunakan antara lain Visio, Rational Rose, Enterprise Architect dan lain sebagainya. Tapi tidak bisa dipungkiri untuk bisa menggunakan tools tersebut, kita harus merogoh kantong untuk membeli lincense-nya, dan jujur saja untuk sekelas saya harga tools tersebut banyak tidak terjangkau, kecuali kita main akal-akalan toolsnya di crack, tapi untuk nge-crack tools tersebut, sebenarnya bukan pekerjaan yang simple dan hukumnya haram. (lho kok malah lari ke masalah crack sih...). Kembali ke subject aslinya, jadi tujuan kita kali ini bagaimana caranya membuat tools yang powerfull untuk mendesain table dengan fasilitas yang kita punyai saat ini.

Saya yakin hampir semua orang IT pasti pernah menggunakan MS-Excel untuk hal-hal tertentu mulai dari hanya sekedar ngetik, sampai membuat laporan pekerjaan dan lain sebagainya. Disini saya ingin menunjukkan bagaimana cara membuat format table di excel sampai dengan membuat program macro sehingga desain table yang telah dibuat tersebut bisa di-auto generate menjadi script transact-SQL.

Langkah-langkah desain table dan pembuatan program macro adalah sebagai berikut :

Buka Excel anda, kemudian tentukan dahulu table-table yang akan anda buat. Untuk lebih jelasnya lihat gambar dibawah ini :

	A	B	C	D	E
1	No	Table Name	Description		
2	1	MSPRD001	Table Master : Kategori Product		
3	2	MSPRD002	Table Master : Product		
4	3	TRSL001	Table Transaction : Penjualan Header		
5	4	TRSL002	Table Transaction : Penjualan Detail		
6					

Pada gambar diatas "Sheet1" telah dirubah namanya menjadi "Master", sebenarnya tidak dirubah juga tidak apa-apa, tergantung selera anda. Hal yang sama juga dilakukan untuk Sheet2. Pada gambar diatas, Sheet2 telah dirubah namanya menjadi "Table". Selanjutnya sheet table tersebut digunakan untuk mendefinisikan semua struktur table mulai dari nama table, nama field, data type dan sebagainya. Untuk lebih jelasnya lihat gambar dibawah ini :

F13

OBJ

A	B	C	D	E	F	G	H	
1	Y	MSPRD001	Table Master : Kategori Product					
2		COLUMNS	TYPE	REQ'D	KEY	OBJ	REF	RCOL
3		CAT_ID	varchar(6)	NOT NULL	PK			
4		CAT_NAME	varchar(20)	NOT NULL				
5								
6	Y	MSPRD002	Table Master : Product					
7		COLUMNS	TYPE	REQ'D	KEY	OBJ	REF	RCOL
8		PRODUCT_ID	varchar(6)	NOT NULL	PK			
9		PRODUCT_NAME	varchar(20)	NOT NULL				
10		CAT_ID	varchar(6)	NOT NULL	FK	FK_MSPRD002_MSPRD001_CAT_ID	MSPRD001	CAT_ID
11								
12	Y	TRSL001	Table Transaction : Penjualan Header					
13		COLUMNS	TYPE	REQ'D	KEY	OBJ	REF	RCOL
14		NO_NOTA	varchar(6)	NOT NULL	PK			
15		TGL_JUAL	varchar(20)	NOT NULL				
16		KETERANGAN	text	NULL				
17								
18	Y	TRSL002	Table Transaction : Penjualan Detail					
19		COLUMNS	TYPE	REQ'D	KEY	OBJ	REF	RCOL
20		NO_NOTA	varchar(6)	NOT NULL	PF	PF_TRSL002_TRSL001_NO_NOTA	TRSL001	NO_NOTA
21		PRODUCT_ID	varchar(20)	NOT NULL	PF	PF_TRSL002_MSPRD002_PRODUCT_ID	MSPRD002	PRODUCT_ID
22		QTY	int	NOT NULL				
23		PRICE	money	NOT NULL				
24								

MasterTableSheet3

Pada gambar diatas dapat anda lihat ada beberapa kolom yang digunakan mulai dari kolom A sampai dengan H.

Silahkan anda amati kolom A1, B1 dan C1. Pada Kolom A1 terdapat huruf "Y". di kolom itu sebenarnya hanya sebagai flag (tanda) dimana jika terdapat huruf "Y" maka table yang bersangkutan akan di generate menjadi script. Flag ini nantinya akan sangat berguna di program macro. Kolom B1 terdapat nama table, dimana nama table tersebut harus sama persis dengan nama table di sheet master. Sedangkan kolom C1 berisi keterangan dari nama table yang bersangkutan.

Selanjutnya anda lihat di kolom B2 sampai kolom H2. Disitu ada tulisan COLUMNS sampai dengan RCOL, berikut ini keterangan masing-masing fungsinya :

Kolom	Fungsi
COLUMNS	Berfungsi untuk mendefinisikan field-field apa saja yang terdapat dalam table yang bersangkutan.
TYPE	Berfungsi untuk mendefinisikan type data apa yang digunakan untuk masing-masing field.
REQ'D	Berfungsi untuk mendefinisikan field tersebut harus terisi (NOT NULL) atau boleh tidak diisi (NULL)
KEY	Key disini terdapat jenis 3 flag yaitu "PK" yang melambangkan Primary Key, "FK" yang melambangkan "Foreign Key" atau "PF" Primary Foreign Key. Anda bisa menambahkan flag yang lain misalkan "DF" = default "TR"= trigger dan lain sebagainya sesuai dengan kebutuhan anda.
OBJ	Pada SQL Server atau di Database engine lainnya seperti Oracle atau MySql, sebuah constraint diwajibkan diberi nama yang unik. Penamaan constraint tersebut disimpan pada kolom OBJ, misalkan jika terdapat constraint foreign key untuk CAT_ID, maka foreign key tersebut diberi nama FK_MSPRD002_MSPRD001_CAT_ID dan sebagainya. Aturan penamaan ini saya mengikuti aturan penamaan constraint di SQL Server.

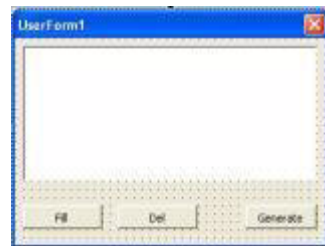
REF	Kolom ini hanya berfungsi untuk menjelaskan key tersebut direferensikan ke table mana. Jadi misalkan pada table product (MSPRD002) terdapat foreign key yang mengarah ke table kategori product (MSPRD001), maka kolom ini berisi nama table yang direferensi oleh key yang bersangkutan.
RCOL	Hampir sama dengan kolom REF, yang membedakan adalah jika kolom REF berisi nama table yang direferensi, maka pada kolom ini berisi nama field yang direferensi.

Sebenarnya format tersebut tidaklah harus seperti itu, format tersebut masih bisa anda modifikasi sesuai dengan selera dan kebutuhan anda.

Setelah anda selesai membuat formatnya, maka langkah selanjutnya adalah membuat macro yang berfungsi untuk men-generate desain table tersebut menjadi script transact-SQL. Berikut ini langkah-langkah membuat macronya :

Step 1. Pilih menu Tools-Macro-Visual Basic Editor

Step 2. Desain interfacenya seperti gambar disamping :



Step 3. Susun property component-componentnya seperti dibawah ini

Component	Setting Property
Form	Name = frmMain
ListBox	Name = Lst
Command1	Name = CmdFill, Caption = Fill
Command2	Name = CmdDel, Caption = Del
Command3	Name = CmdGen, Caption = Generate

Step 4. Click frmMain, kemudian tekan F7 untuk mnampilkan window Code, selanjutnya silahkan copy paste code programnya dibawah ini :

```
Private Sub CmdFill_Click()
    Dim O As Object
    For Each O In Excel.Sheets
        Lst.AddItem O.Name
    Next
End Sub

Private Sub CmdDel_Click()
    Lst.RemoveItem Lst.ListIndex
End Sub

Private Sub CmdGen_Click()
    Call CreateSQL
End Sub

'-----
Private Sub setdef()
    Dim i As Long
    For i = 1 To 5000
        If Sheet13.Cells(i, 8) = "(GETDATE())" Then
            Sheet13.Cells(i, 5) = "DF"
        End If
    Next i
End Sub

Private Sub CreateSQL()
    Dim SQL As String
    Dim FileName As String
    FileName = "C:\Generated.sql"
```

```
Dim i As Integer
Dim ketemu As Boolean
Dim O As Object
For Each O In Excel.Sheets
    ketemu = False
    i = 0
    While (i <= Lst.ListCount - 1) And Not ketemu
        'ListBox1.AddItem o.Name
        If O.Name = Lst.List(i) Then
            GenerateSQL SQL, O, "table" 'ITEM 1
            GenerateSQL SQL, O, "keys" 'ITEM 1
            GenerateSQL SQL, O, "update_triggers" 'ITEM 1
            GenerateSQL SQL, O, "triggers"

            ketemu = True
        Else
            i = i + 1
        End If
    Wend
Next
'GenerateSQL SQL, Sheet3, "insert"
'GenerateSQL SQL, Sheet17, "insert"

SQL = "Begin tran " & vbCrLf & SQL & vbCrLf & "rollback tran"

Open FileName For Output As #1
Print #1, , SQL
Close #1

If MsgBox("Scrip sudah disimpan dalam file " & FileName & ", anda ingin membuka file tsb?", vbYesNo, "Konfirmasi") = vbYes Then
    Shell "notepad.exe " & FileName, vbNormalFocus
End If
End Sub

Private Sub GenerateSQL(ByRef SQL As String, Sheet As Object, Part As String)

    Dim i As Long
    Dim RowReff As String
    Dim TableN() As String
    Dim StartFrom() As Long
    Dim StartTo As Long
    Dim PKString As String
    Dim IsString As Boolean
    Dim TrgString As String

    Dim LastRow As Long
    Dim ObjName As String
    Dim PF As String
    Dim PK As String

    Dim ColString As String
    Dim ValString As String

    ReDim TableN(0)
    ReDim StartFrom(0)

    'cari lastrow

    SQL = SQL & vbCrLf & "/* === " & Sheet.Name & " : " & Part & " ===== */" & vbCrLf
    J = 0
    Do Until i > 2
        J = J + 1
        If Sheet.Cells(J, 2) <> "" Then
            i = 0
        Else
            i = i + 1
        End If
    Loop

    LastRow = J - 3
```

```
For i = 2 To LastRow
RowReff = Sheet.Cells(i, 2)

If (RowReff = "TABLE" Or RowReff = "COLUMNS" Or RowReff = "TRIGGER") Then
ReDim Preserve TableN(UBound(TableN) + 1)
ReDim Preserve StartFrom(UBound(StartFrom) + 1)

TableN(UBound(TableN)) = Sheet.Cells(i - 1, 2)

Select Case RowReff
Case "COLUMNS"

    StartFrom(UBound(StartFrom)) = i + 1
Case "TABLE"
    StartFrom(UBound(StartFrom)) = i + 3
Case "TRIGGER"
    StartFrom(UBound(StartFrom)) = i + 1

End Select

End If
Next i

Select Case Part
Case "table"
    'CREATE TABLE
    For i = 1 To UBound(TableN)

    If i < UBound(TableN) Then
        StartTo = StartFrom(i + 1) - 5
    Else
        StartTo = LastRow
    End If

    If Sheet.Cells(StartFrom(i) - 2, 1) = "Y" Then

        SQL = SQL & vbCrLf & "CREATE TABLE [" & TableN(i) & "] ("

        For J = StartFrom(i) To StartTo

            SQL = SQL & vbCrLf & Sheet.Cells(J, 2) & " " & Sheet.Cells(J, 3) & " " &
Sheet.Cells(J, 4)
            If J <> StartTo Then
                SQL = SQL & ","
            End If
        Next J

        SQL = SQL & vbCrLf & ") ON [PRIMARY]"
        SQL = SQL & vbCrLf & "GO" & vbCrLf

    End If

Next i

    'PRIMARY KEY
    For i = 1 To UBound(TableN)

    If i < UBound(TableN) Then
        StartTo = StartFrom(i + 1) - 5
    Else
        StartTo = LastRow
    End If

    If Sheet.Cells(StartFrom(i) - 2, 1) = "Y" Then

        SQL = SQL & vbCrLf & "ALTER TABLE [dbo].[" & TableN(i) & "] WITH NOCHECK ADD"

        'DEFAULT VALUE
        For J = StartFrom(i) To StartTo
            RowReff = Sheet.Cells(J, 5)
```

```

        If RowReff = "DF" Or RowReff = "PD" Then
            SQL = SQL & vbCrLf & "CONSTRAINT [DF__" & TableN(i) & "__" & Sheet.Cells(J, 2) & "]"
            DEFAULT (" & Sheet.Cells(J, 8) & ") FOR [" & Sheet.Cells(J, 2) & "], "

        End If
    Next J

    SQL = SQL & vbCrLf & "CONSTRAINT [PK__" & TableN(i) & "]" PRIMARY KEY CLUSTERED"
    SQL = SQL & vbCrLf & "("

    PKString = ""
    For J = StartFrom(i) To StartTo
        RowReff = Sheet.Cells(J, 5)

        If RowReff = "PK" Or RowReff = "PF" Or RowReff = "PD" Then
            PKString = PKString & vbCrLf & Sheet.Cells(J, 2) & ", "

        End If
    Next J

    PKString = Left(PKString, Len(PKString) - 1)
    SQL = SQL & PKString
    SQL = SQL & vbCrLf & ") ON [PRIMARY]"
    SQL = SQL & vbCrLf & "GO" & vbCrLf

    End If
Next i

Case "keys"

'FOREIGN KEY
For i = 1 To UBound(TableN)

    If i < UBound(TableN) Then
        StartTo = StartFrom(i + 1) - 5
    Else
        StartTo = LastRow
    End If

    If Sheet.Cells(StartFrom(i) - 2, 1) = "Y" Then
        PF = ""
        PK = ""

        For J = StartFrom(i) To StartTo
            RowReff = Sheet.Cells(J, 5)

            If Sheet.Cells(J, 1) <> "O" And (RowReff = "FK" Or RowReff = "PF") Then
                ObjName = Sheet.Cells(J, 6)
                If Trim(ObjName) <> "" Then
                    PF = PF & IIf(PF = "", "", ", " & vbCrLf) & Sheet.Cells(J, 2)
                    PK = PK & IIf(PK = "", "", ", " & vbCrLf) & Sheet.Cells(J, 8)

                    SQL = SQL & vbCrLf & "ALTER TABLE [dbo].[" & TableN(i) & "] ADD"

                    SQL = SQL & vbCrLf & "CONSTRAINT [" & ObjName & "] FOREIGN KEY"
                    SQL = SQL & vbCrLf & "("

                    SQL = SQL & vbCrLf & PF
                    SQL = SQL & vbCrLf & ") REFERENCES [dbo].[" & Sheet.Cells(J, 7) & "] ("

                    SQL = SQL & vbCrLf & PK
                    SQL = SQL & vbCrLf & ") "
                    SQL = SQL & vbCrLf & "GO" & vbCrLf
                    PK = ""

                    PF = ""
                    'MsgBox SQL
                Else
                    PF = PF & IIf(PF = "", "", ", " & vbCrLf) & Sheet.Cells(J, 2)
                    PK = PK & IIf(PK = "", "", ", " & vbCrLf) & Sheet.Cells(J, 8)
                End If
            End If
        Next J
    End If
Next i

```

```
End If
Next i

Case "update_triggers"

'CREATE UPDATE TRIGGER
For i = 1 To UBound(TableN)

If i < UBound(TableN) Then
    StartTo = StartFrom(i + 1) - 5
Else
    StartTo = LastRow
End If

If Sheet.Cells(StartFrom(i) - 2, 1) = "Y" Then

    PKString = ""

    For J = StartFrom(i) To StartTo
        RowReff = Sheet.Cells(J, 5)
        If RowReff = "PK" Or RowReff = "PF" Then
            PKString = PKString & "+"/'+' & Sheet.Cells(J, 2)
        End If
    Next J

    PKString = Right(PKString, Len(PKString) - 5)

    SQL = SQL & vbCrLf & "CREATE TRIGGER [TR_" & TableN(i) & "_MODIFY_DATE] ON dbo." &
    TableN(i) & " FOR UPDATE AS " & vbCrLf & "update " & TableN(i) & " set
    MODIFY_DATE=getdate() where " & PKString & " in (select " & PKString & " from
    inserted)"

    SQL = SQL & vbCrLf & "GO" & vbCrLf

End If

Next i
Case "triggers"
'CREATE TRIGGER
For i = 1 To UBound(TableN)
If i < UBound(TableN) Then
    StartTo = StartFrom(i + 1) - 3
Else
    StartTo = LastRow
End If
If Sheet.Cells(StartFrom(i) - 1, 1) = "Y" Then
    TrgString = ""

    For J = StartFrom(i) To StartTo
        PKString = Sheet.Cells(J, 2)
        TrgString = TrgString & IIf(TrgString <> "", vbCrLf, "") & Sheet.Cells(J, 2)
    Next J
    SQL = SQL & vbCrLf & TrgString
    SQL = SQL & vbCrLf & "GO" & vbCrLf

End If
Next i
End Select
End Sub
```

Step 5. Masih di Visual Basic editor, silahkan anda lihat project explorer, disitu terdapat terdapat dua folder yaitu Microsoft Excel objects dan Forms. Pada folder Microsoft Excel objects terdapat object Sheet1(Master), Sheet2(Table), Sheet3(Sheet3) dan ThisWorkbook. Sekarang silahkan click ThisWorkbook, kemudian tekan F7. Setelah muncul window code anda copy-paste code program dibawah ini :

```
Sub main()
    Dim f As frmMain
    Set f = New frmMain
    f.Show
End Sub
```

Sampai step ke 5, proses desain table dan pembuatan macro sudah selesai, langkah selanjutnya tinggal menjalankan macro untuk proses generate script transact SQL. Untuk menjalankan macro ada dua cara yang pertama di window Visual Basic editor, silahkan tekan tombol F5. Cara yang kedua di window excel anda pilih menu Tools > Macro > Macros (Alt+F8). Setelah muncul window Run, click ThisWorkbook.main kemudian click Run.

Setelah macro anda dalam kondisi running, maka selanjutnya adalah mengoperasikan program macronya. Pertama tekan tombol Fill. Tombol Fill ini berfungsi untuk menampilkan semua Sheet mulai dari Sheet Master sampai dengan Sheet3 pada Listbox yang telah anda buat tadi. Dikarenakan Sheet yang dipakai hanya Sheet Table, maka hapus Sheet Master dan Sheet3 di Listbox tersebut dengan cara meng-click tombol Del. Terakhir click tombol Generate, tombol generate ini nantinya akan menghasilkan sebuah file yang bernama Generate.sql dimana file ini tersimpan di Drive C.

Demikian artikel sederhana ini dibuat, semoga bermanfaat bagi kita semua. Penulis sangat menyadari bahwa artikel ini masih sangat jauh dari kata sempurna oleh karena itu penulis sangat mengharapkan kritik dan saran dari pembaca yang budiman demi kesempurnaan artikel ini maupun artikel-artikel penulis selanjutnya.

REFERENCES

1. <http://www.bengkelprogram.com>

BIOGRAFI PENULIS



Eko Heri Susanto. Putra daerah Pacitan Jawa Timur ini lahir pada tanggal 5 Maret 1979. Menamatkan pendidikan sekolah dasar di SDN Tahunan II Tegalombo Pacitan Jatim pada tahun 1991. Menamatkan sekolah menengah tingkat pertama di SLTPN I Ponorogo Jatim pada tahun 1994. Menamatkan sekolah menengah tingkat atas di SMUN II Ponorogo Jatim pada tahun 1997. Menyelesaikan study S1 di Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang Jatim, pada tahun 2002.

Semasa kuliah, penulis berkecimpung dalam sebuah organisasi Unit Kegiatan Mahasiswa (UKM) STIKI Computer Networking Club (SCeN) “Kampus ELANG” STIKI Malang.

Dalam kesehariannya penulis berprofesi sebagai programmer, adapun bahasa pemrograman maupun tools lain yang digeluti penulis saat ini adalah Visual Basic 6, Microsoft .Net, ASP, PHP, Java, SQL Server, MySql.

Saat ini penulis bekerja sebagai programmer di PT. Modulo Lima Belas, Cikarang Bekasi, Jawa Barat.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:

URL: <http://www.jawadwipa.com>

Email: ekoheri@gmail.com