

TRIGGER

- ✓ Trigger
 - Create Trigger
 - Alter Trigger
 - Drop Trigger
 - *Update()*

Tujuan Mata Kuliah

- ✓ Mahasiswa dapat membuat Trigger dan mengimplementasikannya pada transaksi

Tools yang digunakan

- ✓ SqlCmd
- ✓ SQL Server Management Studio SQL Query Editor

TRIGGER

adalah T-SQL yang disimpan pada database server dan akan diaktivasi melalui event ketika melakukan statement-statement SQL (INSERT, DELETE, UPDATE) pada sebuah table. Trigger dapat membantu dalam menjaga konsistensi dan integritas data.

Create Trigger

```
CREATE TRIGGER <nama_trigger>
ON <nama_table>|<nama_view>
FOR | AFTER | INSTEAD OF
[INSERT] [UPDATE] [DELETE]

AS
T-SQL /SQL STATEMENT
```

Alter Trigger

```
ALTER TRIGGER <nama_trigger>
ON <nama_table>|<nama_view>
FOR | AFTER | INSTEAD OF
[INSERT] [UPDATE] [DELETE]

AS
T-SQL /SQL STATEMENT
```

DROP Trigger

```
DROP TRIGGER <nama_trigger>
```

Trigger Event

<u>NAMA EVENT</u>	<u>KETERANGAN</u>
AFTER / FOR INSERT	Terjadi setelah statement Insert
AFTER / FOR UPDATE	Terjadi setelah statement Update
AFTER / FOR DELETE	Terjadi setelah statement Delete

Untuk me-nonaktifkan trigger

```
DISABLE TRIGGER <nama_trigger> on Table | View
```

Untuk me-nonaktifkan trigger

```
ENABLE TRIGGER <nama_trigger> on Table | View
```

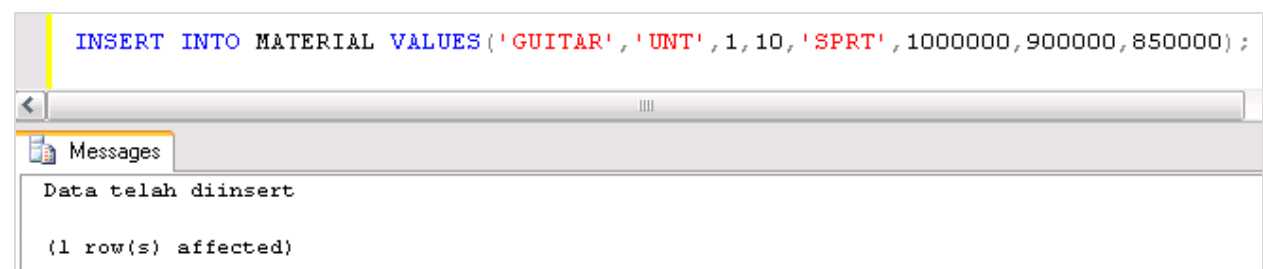
Contoh Trigger Insert

Insert Trigger event pada table MATERIAL

```
CREATE TRIGGER trg_material_after_insert  
ON Material AFTER INSERT  
AS  
BEGIN  
    PRINT 'Data telah diinsert'  
END
```

Test Trigger Event

```
INSERT INTO MATERIAL  
VALUES ('GUITAR', 'UNT', 1, 10, 'SPRT', 1000000, 900000, 850000);
```



The screenshot shows a database client window with a SQL editor at the top containing the following code:

```
INSERT INTO MATERIAL VALUES ('GUITAR', 'UNT', 1, 10, 'SPRT', 1000000, 900000, 850000);
```

Below the editor is a 'Messages' pane that displays the output of the query:

```
Data telah diinsert  
(1 row(s) affected)
```

Alias in Trigger

Dalam Trigger dikenal istilah alias atau referensi, yaitu sejenis variable resultset yang menyimpan nilai dari suatu kolom didalam table.

INSERTED

Table virtual untuk menampung nilai pada trigger event INSERT dan UPDATE.

DELETED

Table virtual untuk menampung nilai pada trigger event DELETE dan UPDATE.

Jika dilihat dari table

<u>EVENT</u>	<u>INSERTED</u>	<u>DELETED</u>
<u>Insert</u>	<u>Berisi inserted rows</u>	<u>Empty</u>
<u>Update</u>	<u>Berisi rows setelah di update</u>	<u>Berisi rows sebelum di update</u>
<u>Delete</u>	<u>Empty</u>	<u>Berisi rows sebelum di delete</u>

Contoh**Insert Trigger pada table MATERIAL**

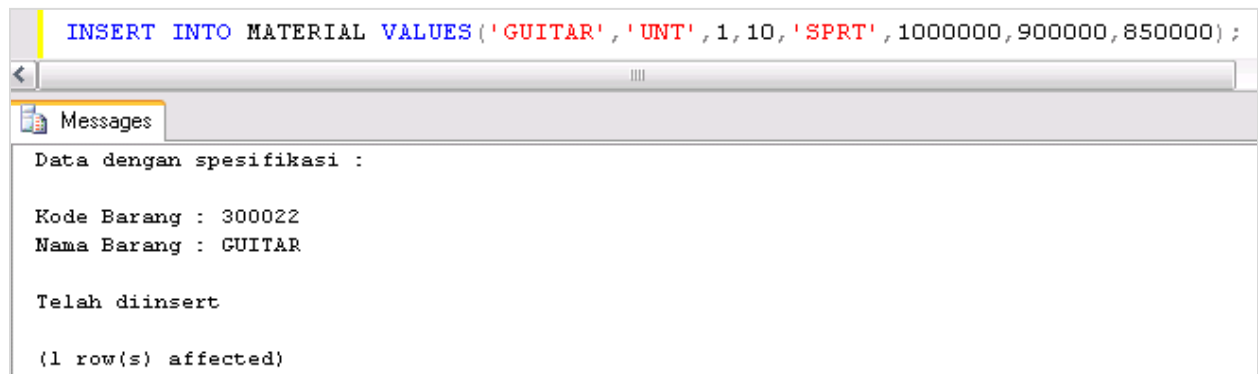
```

ALTER TRIGGER trg_material_after_insert
ON Material AFTER INSERT
AS
BEGIN
    DECLARE @kodebarang int,
            @namabarang varchar(35)

    SELECT @kodebarang = MaterialNumber,
           @namabarang = MaterialDescription
    FROM INSERTED

    PRINT 'Data dengan spesifikasi : ' + CHAR(13) + CHAR(13) +
          'Kode Barang : ' + LTRIM(str(@kodebarang)) + CHAR(13) +
          'Nama Barang : ' + @namabarang + CHAR(13) + CHAR(13) + 'Telah diinsert'
END

```



The screenshot shows a SQL query execution window with the following SQL statement:

```
INSERT INTO MATERIAL VALUES ('GUITAR', 'UNT', 1, 10, 'SPRT', 1000000, 900000, 850000);
```

Below the query, a 'Messages' window displays the output of the trigger:

```

Data dengan spesifikasi :

Kode Barang : 300022
Nama Barang : GUITAR

Telah diinsert

(1 row(s) affected)

```

Contoh Trigger Delete**Delete Trigger event pada table MATERIAL**

```

CREATE TRIGGER trg_material_after_delete
ON Material AFTER DELETE
AS
DECLARE @namabarang varchar(35)

SELECT @namabarang = MaterialDescription
FROM DELETED

PRINT 'Data dengan Nama Barang : ' + @namabarang + ' telah di hapus'

```

```
DELETE FROM MATERIAL WHERE MaterialNumber = 300022
```

Messages

Data dengan Nama Barang : GUITAR telah di hapus

(1 row(s) affected)

Trigger Update

Tidak seperti event trigger insert dan delete, trigger update memiliki 2 table virtual yaitu inserted dan deleted. (Old dan New di Oracle dan Mysql). Dimana deleted = old (data sebelum di update) dan inserted=new (data setelah di update).

Contoh

Trigger Update Nama Barang pada table material

```
CREATE TRIGGER trg_material_after_update
ON Material AFTER UPDATE
AS
DECLARE @namabarangOld varchar(35),
        @namabarangNew varchar(35)

SELECT @namabarangOld = MaterialDescription FROM DELETED
SELECT @namabarangNew = MaterialDescription FROM INSERTED

PRINT 'Nama Barang telah dirubah dari '+@namabarangOld+' menjadi '+@namabarangNew
```

```
UPDATE MATERIAL SET MaterialDescription = 'BMW' WHERE MaterialNumber = 300002
```

Messages

Nama Barang telah dirubah dari BMX menjadi BMW

(1 row(s) affected)

Memeriksa column yang di update dengan *UPDATE(<nama field>)*

Contoh:

```
ALTER TRIGGER trg_material_after_update
ON Material AFTER UPDATE
AS
BEGIN
    DECLARE @namabarangOld varchar(35),
            @namabarangNew varchar(35),
            @satuanOld varchar(3),
            @satuanNew varchar(3),
            @stokAwalOld int,
            @stokAwalNew int

    SELECT @namabarangOld = MaterialDescription,
           @satuanOld = UoM ,
           @stokAwalOld = BeginningStock
    FROM DELETED

    SELECT @namabarangNew = MaterialDescription,
           @satuanNew = UoM ,
           @stokAwalNew = BeginningStock
    FROM INSERTED

    if UPDATE(MaterialDescription)
        PRINT 'Nama Barang telah dirubah dari ' + @namabarangOld + ' menjadi ' +
@namabarangNew

    if UPDATE(UoM)
        PRINT 'Satuan telah dirubah dari ' + @satuanOld + ' menjadi ' +
@satuanNew

    if UPDATE(BeginningStock)
        PRINT 'Stock Awal telah dirubah dari ' + LTRIM(STR(@stokAwalOld)) + '
menjadi ' + LTRIM(STR(@stokAwalNew))

END
```

Kombinasi Trigger Event

Event trigger INSERT, DELETE, UPDATE dapat dibuat sekaligus

```
CREATE TRIGGER TrgMaterial
ON Material
AFTER INSERT,DELETE,UPDATE
AS

declare @operasi varchar

SET @operasi = ''

IF EXISTS(SELECT * FROM DELETED)
    BEGIN
        IF EXISTS(SELECT * FROM INSERTED)
            SET @operasi = 'U'
        ELSE
            SET @operasi = 'D'
    END
ELSE
    SET @operasi = 'I'

IF @operasi = 'I'
    PRINT 'T-SQL INSERT...'
ELSE IF @operasi = 'U'
    PRINT 'T-SQL UPDATE...'
ELSE IF @operasi = 'D'
    PRINT 'T-SQL DELETE...'
```


INSTEAD OF

Instead Of trigger hanya di panggil tetapi tidak dieksekusi. Tidak seperti AFTER/FOR Instead of dapat digunakan juga pada view. Instead of tidak dapat dijalankan pada table yang memiliki relasi ON DELETE dan ON UPDATE dan WITH CHECK OPTION.

Contoh

```
CREATE TABLE Mahasiswa(
  nrp int PRIMARY KEY NOT NULL,
  nama varchar(35),
)
```

```
CREATE TRIGGER TrgMhs
ON Mahasiswa
INSTEAD OF INSERT
AS
  SELECT * FROM MAHASISWA
  WHERE nrp = (SELECT nrp FROM INSERTED)
```

```
INSERT INTO Mahasiswa VALUES('3302019', 'Iman')
```

nrp	nama
-----	------

Rubah dengan after

```
ALTER TRIGGER TrgMhs
ON Mahasiswa
AFTER INSERT
AS
  SELECT * FROM MAHASISWA
  WHERE nrp = (SELECT nrp FROM INSERTED)
```

```
INSERT INTO Mahasiswa VALUES('3302019', 'Iman')
```

nrp	nama
1	3302019 Iman

INSTEAD OF OVERRIDE VALUE

Tambahkan field nilai pada table mahasiswa

```
ALTER TABLE Mahasiswa ADD nilai int
```

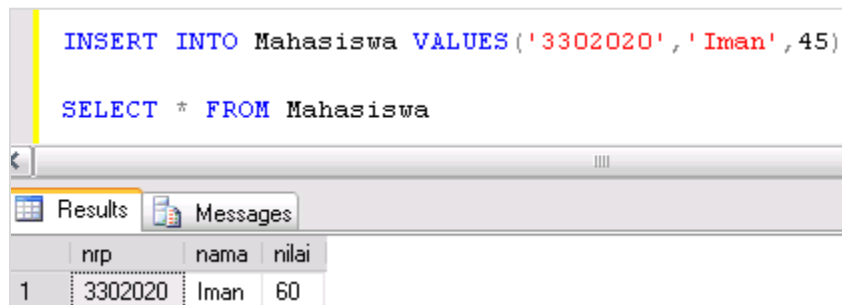
Alter Trigger dengan case force nilai dengan 60 jika nilai <60

```
ALTER TRIGGER TrgMhs
ON Mahasiswa
INSTEAD OF INSERT
AS
DECLARE @xnilai int

SET NOCOUNT ON
SELECT @xnilai=nilai FROM INSERTED
if @xnilai<60
    INSERT INTO Mahasiswa SELECT nrp,nama,60 FROM INSERTED
else
    INSERT INTO Mahasiswa SELECT * FROM INSERTED
```

```
INSERT INTO Mahasiswa VALUES ('3302020', 'Iman', 45)

SELECT * FROM Mahasiswa
```



The screenshot shows a SQL query window with the following text:

```
INSERT INTO Mahasiswa VALUES ('3302020', 'Iman', 45)

SELECT * FROM Mahasiswa
```

Below the query window, there are two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with the following data:

	nrp	nama	nilai
1	3302020	Iman	60

Melihat Trigger yang terpasang pada table

```
Sp_helptrigger View | Table
```

Contoh :

```
Sp_helptrigger Mahasiswa
```

LATIHAN

1. Buatlah trigger untuk mengupdate NRP mahasiswa jika kode jurusan nya berubah.
2. Buatlah trigger untuk meng-update table material pada field currentstock setiap ada penambahan(RECEIPT) dan pengurangan(ORDERS) stock.
3. Buatlah Trigger untuk meng-counter auto number (table counter dengan field fromNumber, ToNumber, LastNumber) pada Table Adjust
4. Buatlah Trigger INSERT,DELETE,UPDATE untuk Log Audit Trail pada table (table tentukan sendiri) dan kapan waktunya serta siapa user nya.

~~EOF~~

#roots