# Programming Database

"Operator SET(Himpunan) dan Modifikasi data"

## Objective:

1. Memahami Konsep perintah Summary Select, window dalam Transaction (T-SQL) di SQL Server.
2. Memahami SET operator (UNION, EXCEPT, INTERSECT) dalam SQL Server.
3. Memahami Modifikasi data lanjutan.

## Review and Case Study:

*Bagian ini akan memberikan studi kasus dan review mengenai perintah yang terdapat dalam studi kasus yang diberikan.*

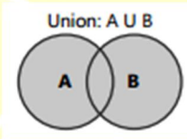
### 1. SET Operator

Set operator adalah operator yang menerapkan diantara dua set inputan atau menggunakan query yang lebih komplek, kemudian menghasilkan sebuah result set dalam dua inputan query. T-SQL mendukung tiga operator set: UNION, INTESECT, dan EXCEPT.

#### A. UNION

Dalam T-SQL, operator UNION menyatukan hasil dari dua inputan query, T-SQL mendukung kedua fungsi union, yakni: UNION ALL (Multisets) dan UNION (secara implisit akan distinct terhadap data).

UNION dalam teori SET, union mengabungkan dari dua SET (kita sebut A dan B)ialah set berisikan semua baik di A dan di B. Dengan kata lain, jika elemen milik salah satu set input, itu milik hasil yang ditetapkan. Gambar dibawah menunjukkan diagram set (juga dikenal sebagai diagram Venn) dengan gambaran grafis dari penyatuan dua set. Daerah yang diarsir merupakan hasil dari operator set.




GAMBAR.1UNION

**Case Study:**

Gunakan operator UNION ALL untuk menyatukan lokasi dari pegawai (employee) dan pelanggan (customer)

**Answer:**

```
SELECT country, region, city FROM HR.Employees
UNION ALL
SELECT country, region, city FROM Sales.Customers;
```

**Case Study:**

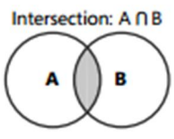
Gunakan operator UNION untuk menyatukan lokasi dari pegawai (employee) dan pelanggan (customer) yang berbeda-beda dari salah satu pada kedua data tersebut.

**Answer:**

```
SELECT country, region, city FROM HR.Employees
UNION
SELECT country, region, city FROM Sales.Customers;
```

#### B. INTERSECT

Dalam teori set, intersection dari dua set (kita sebut mereka a dan b) ialah set yang semua elemen yang dimiliki A dan juga milik B, dengan gambaran grafis dari intersection dua set. Daerah yang diarsir merupakan hasil dari operator set.




GAMBAR 2 INTERSECT

**Case Study:**

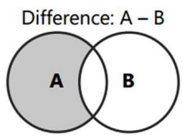
Munculkanlah lokasi-lokasi yang berbeda-beda tanpa ada duplikasi data untuk lokasi yang terdapat pada employee dan customer?

**Answer:**

```
SELECT country, region, city FROM HR.Employees
INERSECT
SELECT country, region, city FROM Sales.Customers;
```

## C. EXCEPT

Dalam teori set, EXCEPT dari dua set (kita sebut mereka a dan b) ialah set yang semua elemen yang dimiliki A dan bukan milik B, dengan gambaran grafis dari EXCEPT dua set. Daerah yang diarsir merupakan hasil dari operator set.




GAMBAR.3 EXCEPT

**Case Study:**

Munculkanlah lokasi-lokasi yang berbeda-beda tanpa ada duplikasi data untuk lokasi yang terdapat pada employee tetapi tidak terdapat di customer?

**Answer:**

```
SELECT country, region, city FROM HR.Employees
EXCEPT
SELECT country, region, city FROM Sales.Customers;
```

**Case Study:**

Munculkanlah lokasi-lokasi yang berbeda-beda tanpa ada duplikasi data untuk lokasi yang terdapat pada customer tetapi tidak terdapat di employee?

**Answer:**

```
SELECT country, region, city FROM HR. Customers
EXCEPT
SELECT country, region, city FROM Sales. Employees;
```

**Case Study:**

Munculkanlah lokasi-lokasi yang berbeda-beda tanpa ada duplikasi data untuk lokasi yang terdapat pada supplier tetapi tidak tedapat pada customer dan employee?

**Answer:**

```
SELECT country, region, city FROM Production.Suppliers
EXCEPT
SELECT country, region, city FROM HR.Employees
INTERSECT
SELECT country, region, city FROM Sales.Customers;
```

**Case Study:**

Munculkanlah data negara dan jumlah employee dan customer yang tunggal didalamnya?

**Answer:**

```
SELECT country, COUNT(*) AS numlocations
FROM (SELECT country, region, city FROM HR.Employees
        UNION
        SELECT country, region, city FROM Sales.Customers) AS U
GROUP BY country;
```

**Case Study:**

Munculkanlah dua order terbaru yang dilakukan oleh employee id 3 dan 5 ?

**Answer:**

```
SELECT empid, orderid, orderdate
FROM (SELECT TOP (2) empid, orderid, orderdate
        FROM Sales.Orders
WHERE empid = 3
ORDER BY orderdate DESC, orderid DESC) AS D1
UNION ALL
SELECT empid, orderid, orderdate
FROM (SELECT TOP (2) empid, orderid, orderdate
        FROM Sales.Orders
WHERE empid = 5
ORDER BY orderdate DESC, orderid DESC) AS D2;
```

## 2. Data Modification

Dalam melakukan Manipulasi data baris, biasanya kita menggunakan perintah berikut ini: INSERT, UPDATE dan DELETE. Dalam modul ini saya akan jelaskan data modfifikasi dalam tingkatan lanjut.

### A. Perintah INSERT

T-SQL menyediakan beberapa pernyataan untuk memasukkan data ke dalam tabel: INSERT VALUES (tidak akan dibahas karena dianggap bisa), INSERT SELECT, INSERT EXEC, SELECT INTO, dan BULK INSERT. Sebelum masuk ke materi buat dulu tabel dan data yang akan digunakan untuk proses Manipulasi penyisipan ini. Jalan perintah dibawah ini untuk pembuatan dan pengsisian datanya.

```
 --melakukan pengecekan jika tabel yang dicari ada maka table tersebut akan dihapus
IF OBJECT_ID('dbo.Orders', 'U') IS NOT NULL DROP TABLE dbo.Orders;

CREATE TABLE dbo.Orders
(
    orderid INT NOT NULL
    CONSTRAINT PK_Orders PRIMARY KEY,
    orderdate DATE NOT NULL
    CONSTRAINT DFT_orderdate DEFAULT(SYSDATETIME()),
    empid INT NOT NULL,
    custid VARCHAR(10) NOT NULL
)
```

Case Study:

Masukan data dari hasil query pada table sales.orders dimana table mengembalikan nilai berdasarkan pengapalan ke inggris (United Kingdom)?

Answer:

```
INSERT INTO dbo.Orders(ordered, orderdate, empid, custid)
SELECT orderid, orderdate, empid, custid
FROM Sales.Orders
WHERE shipcountry = 'UK'
```

Case Study:

Buatlah code untuk membuat table baru yang bernama dbo.orders, dan berisikan semua data pada table sales.orders?

Answer:

```
//melakukan pengcekan jika ada table bernama order maka akan dihapus dalam basis
data
IF OBJECT_ID('dbo.Orders', 'U') IS NOT NULL DROP TABLE dbo.Orders;

SELECT orderid, orderdate, empid, custid
INTO dbo.Orders
FROM Sales.Orders;
```

Case Study:

Masukanlah data yang beada dalam file c:\orders.txt kedalam table dbo.Orders dan menentukan jenis file data diatas adalah char, pembatas field pada data di file adalah koma, dan pembatas baris pada file adalah baris baru?

Answer:

```
BULK INSERT dbo.Orders FROM 'c:\orders.txt'
WITH
(
DATAFILETYPE = 'char',
FIELDTERMINATOR = ',',
ROWTERMINATOR = '\n'
);
```

### B. Perintah DELETE

T-SQL menyediakan dua pernyataan untuk menghapus baris dari tabel, yakni DELETE dan truncate. Sebelum masuk ke materi buat dulu tabel dan data yang akan digunakan untuk proses Manipulasi penghapusan ini. Jalan perintah dibawah ini untuk pembuatan dan pengsisian datanya.

```
--melakukan pengecekan jika tabel yang dicari ada maka table tersebut akan dihapus
IF OBJECT_ID('dbo.Orders', 'U') IS NOT NULL DROP TABLE dbo.Orders;
IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL DROP TABLE dbo.Customers;

--Membuat table customer dan orders
CREATE TABLE dbo.Customers
```

```
(
        custid INT NOT NULL,
        companyname NVARCHAR(40) NOT NULL,
        contactname NVARCHAR(30) NOT NULL,
        contacttitle NVARCHAR(30) NOT NULL,
        address NVARCHAR(60) NOT NULL,
        city NVARCHAR(15) NOT NULL,
        region NVARCHAR(15) NULL,
        postalcode NVARCHAR(10) NULL,
        country NVARCHAR(15) NOT NULL,
        phone NVARCHAR(24) NOT NULL,
        fax NVARCHAR(24) NULL,
        CONSTRAINT PK_Customers PRIMARY KEY(custid)
);

CREATE TABLE dbo.Orders
(
        orderid INT NOT NULL,
        custid INT NULL,
        empid INT NOT NULL,
        orderdate DATETIME NOT NULL,
        requireddate DATETIME NOT NULL,
        shippeddate DATETIME NULL,
        shipperid INT NOT NULL,
        freight MONEY NOT NULL
        CONSTRAINT DFT_Orders_freight DEFAULT(0),
        shipname NVARCHAR(40) NOT NULL,
        shipaddress NVARCHAR(60) NOT NULL,
        shipcity NVARCHAR(15) NOT NULL,
        shipregion NVARCHAR(15) NULL,
        shippostalcode NVARCHAR(10) NULL,
        shipcountry NVARCHAR(15) NOT NULL,
        CONSTRAINT PK_Orders PRIMARY KEY(orderid),
        CONSTRAINT FK_Orders_Customers FOREIGN KEY(custid)
        REFERENCES dbo.Customers(custid)
);
GO

--membuat penyisimpan data berdasarkan pada
--data tabel yang sudah ada, yakni: customer dan orders di skema sales
INSERT INTO dbo.Customers SELECT * FROM Sales.Customers;
INSERT INTO dbo.Orders SELECT * FROM Sales.Orders;
```

Case Study:

Hapuslah data orders yang pernah dilakukan pada saat tanggal 01/01/2007?

Answer:

```
DELETE FROM dbo.Orders
WHERE orderdate < '20070101';
```

Resultnya akan menghasilkan pesan sebagai berikut: (152 row(s) affected) yang mengartikan jumlah data yang difilter dan dilakukan penghapusan.

Case Study:

Hapuslah seluruh data customers?

Answer:

```
TRUNCATE FROM dbo.Customers;
```

Keuntungan dari penggunaan TRUCANTE ialah dari sisi minimal dalam proses logged dibandingkan dengan DELETE, sehingga perbedaaan kinerja yang sangat signifikan, sebagai contoh jika perintah TRUNCATE menghapus semua baris dari table dengan jutaan baris, operasi ini akan menyelesaiakannya dalam hitungan detik, sedangkan untuk perintah DELETE, pemprosesannya akan memakan waktu menitan bahkan sampai berjam-jam karena TRUNCATE minimal dalam login ke session dibandingkan DELETE yang full login setiap baris dilakukan penghapusan dan dalam

**Note:** Perhatikan bahwa pesan yang menunjukkan jumlah baris yang terjadi perubahan hanya muncul jika opsi session NOCOUNT MATI, dan secara default session NOCOUNT adalah OFF. Dan jika dalam penghapusan data skala besar ada akan menunggu beberapa saat karena selain menyimpan instruksi penghapusan SQL SERVER MANAGEMENT STUDIO terus-terusan login ke session dalam perhitungan, untuk menghemat waktu anda bisa matikan opsi session NOCOUNT tersebut dengan cara sebagai berikut: SET NOCOUNT ON

TRANCATION, TRUNCATE dapat dibatalkan(undo) saat menggunakan perintah ROLLBACK. Saat melakukan penghapusan TRUNCATE akan mereset keterurutan data pada kolom IDENTITY (AUTO_INCREMENT) baik SEED maupun FEED-nya pada kodisi awal.

Case Study:

seluruh data order yang dilakukan oleh customer yang tunggal di Amerika Serikat(United Stated)?

Answer:

```
DELETE FROM O
FROM dbo.Orders AS O
	JOIN dbo.Customers AS C
	ON O.custid = C.custid
WHERE C.country = N'USA';
```

## C. Perintah UPDATE

T-SQL mendukung pernyataan UPDATE standar yang memungkinkan Anda untuk memperbarui baris dalam sebuah tabel. T-SQL juga mendukung penggunaan yang tidak standar dari pernyataan UPDATE dengan bergabung dan denga Variable(kemudian hari saat belajar stored procedure). Bagian ini menjelaskan berbagai penggunaan pernyataan UPDATE. Sebelum membahas materi ini lebih lanjut eksekusilah perintah berikut ini:

```
IF OBJECT_ID('dbo.OrderDetails', 'U') IS NOT NULL DROP TABLE dbo.OrderDetails;
IF OBJECT_ID('dbo.Orders', 'U') IS NOT NULL DROP TABLE dbo.Orders;

CREATE TABLE dbo.Orders
(
    orderid INT NOT NULL,
    custid INT NULL,
    empid INT NOT NULL,
    orderdate DATETIME NOT NULL,
    requireddate DATETIME NOT NULL,
    shippeddate DATETIME NULL,
    shipperid INT NOT NULL,
    freight MONEY NOT NULL
    CONSTRAINT DFT_Orders_freight DEFAULT(0),
    shipname NVARCHAR(40) NOT NULL,
    shipaddress NVARCHAR(60) NOT NULL,
    shipcity NVARCHAR(15) NOT NULL,
    shipregion NVARCHAR(15) NULL,
    shippostalcode NVARCHAR(10) NULL,
    shipcountry NVARCHAR(15) NOT NULL,
    CONSTRAINT PK_Orders PRIMARY KEY(orderid)
);
CREATE TABLE dbo.OrderDetails
(
    orderid INT NOT NULL,
    productid INT NOT NULL,
    unitprice MONEY NOT NULL
    CONSTRAINT DFT_OrderDetails_unitprice DEFAULT(0),
    qty SMALLINT NOT NULL
    CONSTRAINT DFT_OrderDetails_qty DEFAULT(1),
    discount NUMERIC(4, 3) NOT NULL
    CONSTRAINT DFT_OrderDetails_discount DEFAULT(0),
    CONSTRAINT PK_OrderDetails PRIMARY KEY(orderid, productid),
    CONSTRAINT FK_OrderDetails_Orders FOREIGN KEY(orderid)
    REFERENCES dbo.Orders(orderid),
    CONSTRAINT CHK_discount CHECK (discount BETWEEN 0 AND 1),
    CONSTRAINT CHK_qty CHECK (qty > 0),
    CONSTRAINT CHK_unitprice CHECK (unitprice >= 0)
);
GO

INSERT INTO dbo.Orders SELECT * FROM Sales.Orders;
INSERT INTO dbo.OrderDetails SELECT * FROM Sales.OrderDetails;
```

Case Study:

Naikanlah diskon untuk semua pesanan yang dilakukan oleh customer dengan id 1 sebesar 5 persen ?

Answer:

```
UPDATE OD
SET discount += 0.05
FROM   dbo.OrderDetails AS OD
	JOIN dbo.Orders AS O
	ON OD.orderid = O.orderid
WHERE O.custid = 1;
```

Untuk melakukan hal yang sama seperti diatas kita juga bisa menggunakan konsep subquery untuk mendapatkan detail order mana saja yang dilakukan oleh customer.

```
UPDATE dbo.OrderDetails
SET discount += 0.05
WHERE EXISTS
(SELECT * FROM dbo.Orders AS O
WHERE O.orderid = OrderDetails.orderid
AND O.custid = 1);
```

## D. Perintah MERGE

SQL Server 2008 dan SQL Server 2012 mendukung pernyataan disebut MERGE yang memungkinkan Anda untuk memodifikasi data, menerapkan tindakan yang berbeda (INSERT, UPDATE, dan DELETE) berdasarkan logika kondisional secara bersamaan. Perintah MERGE adalah bagian dari standar SQL, meskipun versi T-SQL menambahkan beberapa ekstensi tidak standar pernyataan itu.

Sebuah tugas yang dicapai oleh perintah MERGE biasanya akan menerjemahkan ke kombinasi dari beberapa pernyataan lain DML (INSERT, UPDATE, dan DELETE). Keuntungan menggunakan MERGE sebagai alternatif adalah bahwa hal ini memungkinkan Anda untuk mengekspresikan permintaan dengan kode yang minimal dan menjalankannya lebih efisien karena memerlukan akses yang lebih sedikit untuk tabel yang terlibat. Sebelum membahas lebih lanjut, anda Eksekusi perintah berikut ini:



```
IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL DROP TABLE dbo.Customers;
GO
CREATE TABLE dbo.Customers
(
	custid INT NOT NULL,
	companyname VARCHAR(25) NOT NULL,
	phone VARCHAR(20) NOT NULL,
	address VARCHAR(50) NOT NULL,
	CONSTRAINT PK_Customers PRIMARY KEY(custid)
);

INSERT INTO dbo.Customers(custid, companyname, phone, address)
VALUES
(1, 'cust 1', '(111) 111-1111', 'address 1'),
(2, 'cust 2', '(222) 222-2222', 'address 2'),
(3, 'cust 3', '(333) 333-3333', 'address 3'),
(4, 'cust 4', '(444) 444-4444', 'address 4'),
(5, 'cust 5', '(555) 555-5555', 'address 5');

IF OBJECT_ID('dbo.CustomersStage', 'U') IS NOT NULL DROP TABLE dbo.
CustomersStage;
GO

CREATE TABLE dbo.CustomersStage
(
	custid INT NOT NULL,
	companyname VARCHAR(25) NOT NULL,
	phone VARCHAR(20) NOT NULL,
	address VARCHAR(50) NOT NULL,
	CONSTRAINT PK_CustomersStage PRIMARY KEY(custid)
);

INSERT INTO dbo.CustomersStage(custid, companyname, phone, address)
VALUES
(2, 'AAAAA', '(222) 222-2222', 'address 2'),
(3, 'cust 3', '(333) 333-3333', 'address 3'),
(5, 'BBBBB', 'CCCCC', 'DDDDD'),
(6, 'cust 6 (new)', '(666) 666-6666', 'address 6'),
(7, 'cust 7 (new)', '(777) 777-7777', 'address 7');
```

**Case Study:**

Tambahkanlah data seorang customer apabila pada table customer belum ada dan lakukanlah perubahan data jika data pada table customerstage sudah ada dalam table customer dan setiap data yang sudah ada dalam customer disesuaikan dengan customerstage?

**Answer:**

```
MERGE INTO dbo.Customers AS TGT
USING dbo.CustomersStage AS SRC
ON TGT.custid = SRC.custid
WHEN MATCHED THEN
```

```
        UPDATE SET
        TGT.companyname = SRC.companyname,
        TGT.phone = SRC.phone,
        TGT.address = SRC.address
WHEN NOT MATCHED THEN
        INSERT (custid, companyname, phone, address)
        VALUES (SRC.custid, SRC.companyname, SRC.phone, SRC.address)
WHEN NOT MATCHED BY SOURCE THEN
        DELETE;
```

**Note:** Pergunakanlah semicolon ; dalam perintah MERGE, dengan menggunakan semicolon ini, cara terbaik dala, mengentikan kerja dari MERGE.