

**BASIS DATA LANJUT
JOBSHEET 8**



**OLEH :
MUHAMMAD ALIYUL MURTADLO (17 / 1741720031)**

**KELAS TI - 2F
PROGRAM STUDI D - IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2018**

PRAKTIKUM

1. Soal-1

```
(SELECT
    productid,
    productname
FROM
    Production.Products
WHERE
    categoryid = 4)

UNION

(SELECT
    P.productid,
    P.productname
FROM
    Production.Products P INNER JOIN Sales.OrderDetails OD
ON
    P.productid = OD.productid
GROUP BY
    P.productid, P.productname
HAVING
    SUM(OD.qty * OD.unitprice) > 50000);
```

	123 productid	ABC productname
1	11	Product QMVUN
2	12	Product OSFNS
3	29	Product VJXYN
4	31	Product XWOXC
5	32	Product NUNAW
6	33	Product ASTMN
7	38	Product QDOMO
8	59	Product UKXRI
9	60	Product WHBYK
10	69	Product COAXA
11	71	Product MYMOI
12	72	Product GEEEO

2. Soal-2

```

(SELECT
    productid,
    productname
FROM
    Production.Products
WHERE
    categoryid = 4)

UNION ALL

(SELECT
    P.productid,
    P.productname
FROM
    Production.Products P INNER JOIN Sales.OrderDetails OD
ON
    P.productid = OD.productid
GROUP BY
    P.productid, P.productname
HAVING
    SUM(OD.qty * OD.unitprice) > 50000);

```

	123 productid T↑	ABC productname T↑
1	11	Product QMVUN
2	12	Product OSFNS
3	31	Product XWOXC
4	32	Product NUNAW
5	33	Product ASTMN
6	59	Product UKXRI
7	60	Product WHBYK
8	69	Product COAXA
9	71	Product MYMOI
10	72	Product GEEOO
11	29	Product VJXYN
12	38	Product QDOMO
13	59	Product UKXRI
14	60	Product WHBYK

3. Soal-3

Perbedaan UNION dan UNION ALL

Jawab : UNION digunakan untuk menggabungkan dua table atau lebih namun setelah digabungkan akan diseleksi data yang sama, data yang sama tersebut hanya akan ditampilkan satu kali. Sedangkan UNION ALL digunakan untuk menggabungkan dua table atau lebih dan akan ditampilkan apa adanya yaitu semua data setelah digabungkan akan ditampilkan semua. Jadi, inti perbedaan dari UNION dan UNION ALL adalah jika

UNION maka akan ditampilkan data yang UNIQUE, sedangkan UNION ALL akan menampilkan semua data yang ada.

4. Soal-4


```
(SELECT
    c1.custid, c1.contactname, c1.orderdate, c1.val
FROM
    (
        SELECT TOP (10)
            o.custid, c.contactname, o.orderdate, o.val
        FROM Sales.OrderValues AS o
        INNER JOIN Sales.Customers AS c ON c.custid = o.custid
        WHERE o.orderdate >= '20080101' AND o.orderdate < '20080201'
        GROUP BY o.custid, c.contactname, o.orderdate, o.val
        ORDER BY o.val DESC
    ) AS c1)

UNION





















(SELECT c2.custid, c2.contactname, c2.orderdate, c2.val
FROM
    (
        SELECT TOP (10)
            o.custid, c.contactname, o.orderdate, o.val
        FROM Sales.OrderValues AS o
        INNER JOIN Sales.Customers AS c ON c.custid = o.custid
        WHERE o.orderdate >= '20080201' AND o.orderdate < '20080301'
        GROUP BY o.custid, c.contactname, o.orderdate, o.val
        ORDER BY o.val DESC
    ) AS c2)
ORDER BY orderdate ASC, val DESC;
```

	123 custid T	ABC contactname T	orderdate T	123 val T
1	39	Song, Lolan	2008-01-06 00:00:00	10,952.85
2	32	Krishnan, Venky	2008-01-06 00:00:00	8,446.45
3	46	Dressler, Marlies	2008-01-09 00:00:00	2,826.00
4	20	Kane, John	2008-01-16 00:00:00	4,705.50
5	76	Gulbis, Katrin	2008-01-20 00:00:00	4,581.00
6	63	Veronesi, Giorgio	2008-01-21 00:00:00	3,812.70
7	71	Navarro, Tomás	2008-01-22 00:00:00	4,931.92
8	65	Moore, Michael	2008-01-26 00:00:00	2,984.00
9	20	Kane, John	2008-01-27 00:00:00	2,966.50
10	89	Smith Jr., Ronaldo	2008-01-30 00:00:00	3,523.40
11	63	Veronesi, Giorgio	2008-02-02 00:00:00	16,387.50
12	30	Shabalin, Rostislav	2008-02-05 00:00:00	2,058.46
13	34	Cohen, Shy	2008-02-13 00:00:00	3,127.50
14	65	Moore, Michael	2008-02-16 00:00:00	11,380.00
15	50	Mace, Donald	2008-02-17 00:00:00	2,090.00
16	20	Kane, John	2008-02-18 00:00:00	6,379.40
17	39	Song, Lolan	2008-02-18 00:00:00	5,502.11
18	71	Navarro, Tomás	2008-02-18 00:00:00	2,753.10
19	37	Crăciun, Ovidiu V.	2008-02-19 00:00:00	10,835.24
20	37	Crăciun, Ovidiu V.	2008-02-26 00:00:00	6,200.55


5. Soal-5

	SELECT c.custid, c.contactname, p.productid, p.productname, p.totalsalesamount FROM Sales.Customers AS c OUTER APPLY dbo.fnGetTop3ProductsForCustomer (c.custid) AS p ORDER BY c.custid;
-------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Result					
SELECT c.custid, c.contactname, p.productid Enter a SQL expression to filter results (use Ctrl+Space)					
	123 custid T	ABC contactname T	123 productid T	ABC productname T	123 totalsalesamount T
1	1	Allen, Michael	63	Product ICKNK	878.0000
2	1	Allen, Michael	59	Product UKXRI	825.0000
3	1	Allen, Michael	28	Product OFBNT	775.2000
4	2	Hassall, Mark	72	Product GEEOO	348.0000
5	2	Hassall, Mark	60	Product WHBYK	340.0000

Save Cancel Script                    

6. Soal-6

		<pre> SELECT c.custid, c.contactname, p.productid, p.productname, p.totalsalesamount FROM Sales.Customers AS c OUTER APPLY dbo.fnGetTop3ProductsForCustomer (c.custid) AS p WHERE p.productid IS NULL; </pre>			
	123 custid	ABC contactname	123 productid	ABC productname	123 totalsalesamount
1	22	Bueno, Janaina Burdan, Neville	[NULL]	[NULL]	[NULL]
2	57	Tollefsen, Bjørn	[NULL]	[NULL]	[NULL]

7. Soal-7

<pre> SELECT custid FROM Sales.Customers WHERE country = 'USA' EXCEPT SELECT o.custid FROM Sales.Orders AS o INNER JOIN Sales.OrderDetails AS d ON d.orderid = o.orderid GROUP BY o.custid HAVING COUNT(DISTINCT d.productid) > 20; </pre>	
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

	123 custid
1	32
2	36
3	43
4	45
5	48
6	55
7	75
8	77
9	78
10	82

8. Soal-8

```

SELECT
    c.custid
FROM Sales.Customers AS c

EXCEPT

SELECT
    o.custid
FROM Sales.Orders AS o
INNER JOIN Sales.OrderDetails AS d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING COUNT(DISTINCT d.productid) > 20

INTERSECT

SELECT
    o.custid
FROM Sales.Orders AS o
INNER JOIN Sales.OrderDetails AS d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING SUM(d.qty * d.unitprice) > 10000;

```

	123 custid
1	1
2	2
3	3
4	6
5	8
6	11
7	12
8	13
9	14
10	15
11	16
12	17

☒ Save
 ☐ Cancel

 59 row(s) fetched - 36ms

 59

9. Soal-9

Jawab: Customer yang tampil pada hasil Soal-8 adalah customer yang membeli produk kurang dari 20 dan juga customer dengan total harga pembelian lebih dari 10000.

10. Soal-10

```

(
SELECT
    c.custid
FROM Sales.Customers AS c

EXCEPT

SELECT
    o.custid
FROM Sales.Orders AS o
INNER JOIN Sales.OrderDetails AS d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING COUNT(DISTINCT d.productid) > 20
)

INTERSECT

SELECT
    o.custid
FROM Sales.Orders AS o
INNER JOIN Sales.OrderDetails AS d ON d.orderid = o.orderid
GROUP BY o.custid
HAVING SUM(d.qty * d.unitprice) > 10000;

```

	123 custid
1	14
2	19
3	23
4	32
5	50
6	55
7	59
8	73
9	75

Save Cancel Script 9 row(s) fetched - 30ms

11. Soal-11

Jawab: Hasil soal-10 berbeda dengan soal-8 karena pada query soal-10 dicari terlebih dahulu customer yang membeli barang dengan jumlah kurang dari 20 kemudian data yang diperoleh tersebut dicari irisannya dengan query setelah intersect yaitu yang total harga pembeliannya lebih dari 10000.

12. Soal-12


```

IF OBJECT_ID('Sales.trgAutoProductDiscontinue') IS NOT NULL
    DROP TRIGGER Sales.trgAutoProductDiscontinue;
GO

CREATE TRIGGER trgAutoProductDiscontinue ON Sales.OrderDetails
AFTER DELETE
AS
    PRINT 'TRIGGER trgAutoProductDiscontinue dipanggil';
    DECLARE @productid int = (select TOP (1) productid from deleted);

    UPDATE Production.Products SET discontinued='true'
    WHERE productid=@productid;
    PRINT 'Men-discontinue product dengan id: ' + CAST(@productid AS VARCHAR);
GO

```

```

IF OBJECT_ID('Production.trgAutoUpdateOrderDetailsUnitPrice') IS NOT NULL
    DROP TRIGGER Production.trgAutoUpdateOrderDetailsUnitPrice;
GO

CREATE TRIGGER trgAutoUpdateOrderDetailsUnitPrice ON Production.Products
AFTER UPDATE
AS
    PRINT 'TRIGGER trgAutoUpdateOrderDetailsUnitPrice dipanggil';
    DECLARE @productid int = (select productid from inserted);
    DECLARE @unitprice MONEY = COALESCE((SELECT unitprice FROM inserted), 0.0);

    UPDATE Sales.OrderDetails SET unitprice = @unitprice
    WHERE productid=@productid;
    PRINT 'Harga di tabel Sales.OrderDetails secara otomatis disesuaikan';
GO

```

```

delete from Sales.OrderDetails where productid = 14;
select * from Production.Products where productid = 14;

```

Results	Messages
	TRIGGER trgAutoProductDiscontinue dipanggil
	TRIGGER trgAutoUpdateOrderDetailsUnitPrice dipanggil
	(0 row(s) affected)
	Harga di tabel Sales.OrderDetails secara otomatis disesuaikan
	(1 row(s) affected)
	Men-discontinue product dengan id: 14
	(22 row(s) affected)

Results		Messages				
	productid	productname	supplierid	categoryid	unitprice	discontinued
1	14	Product PWCJB	6	7	23.25	1

13. Soal-13

```

IF OBJECT_ID('HR.trgDivertUpdateEmployeeToBackup') IS NOT NULL
    DROP TABLE HR.trgDivertUpdateEmployeeToBackup
Go

CREATE TRIGGER trgDivertUpdateEmployeeToBackup ON HR.Employees
INSTEAD OF UPDATE
AS
    PRINT 'TRIGGER trgDivertUpdateEmployeeToBackup DIPANGGIL!';
    DECLARE @firstname VARCHAR(15)=(SELECT firstname from inserted);
    DECLARE @lastname VARCHAR(15)=(SELECT lastname from inserted);
    DECLARE @title VARCHAR(5)=(SELECT title from inserted);
    DECLARE @titleofcourtesy VARCHAR(5)=(SELECT titleofcourtesy from inserted);
    DECLARE @birthdate DATETIME=(SELECT birthdate from inserted);
    DECLARE @hiredate DATETIME=(SELECT hiredate from inserted);
    DECLARE @address VARCHAR(100)=(SELECT [address] from inserted);
    DECLARE @city VARCHAR(15)=(SELECT city from inserted);
    DECLARE @region VARCHAR(7)=(SELECT region from inserted);
    DECLARE @postalcode VARCHAR(6)=(SELECT postalcode from inserted);
    DECLARE @country VARCHAR(10)=(SELECT country from inserted);
    DECLARE @phone VARCHAR(15)=(SELECT phone from inserted);
    DECLARE @mgrid VARCHAR(15)=(SELECT mgrid from inserted);
    DECLARE @empid INT = (SELECT empid FROM inserted);

    UPDATE HR.EmployeesBackup SET firstname=@firstname, lastname=@lastname, title=@title,
    titleofcourtesy=@titleofcourtesy, birthdate=@birthdate, hiredate=@hiredate,
    [address]=@address, city=@city, region=@region, postalcode=@postalcode,
    country=@country, phone=@phone, mgrid=@mgrid WHERE empid=@empid;

    PRINT 'Karyawan dengan empid: '+CAST(@empid AS VARCHAR)+' yang ada di
    HR.EmployeesBackup yang diupdate.';

```

```

UPDATE HR.Employees SET firstname='DEPAN', lastname='BELAKANG'
WHERE firstname='Adi';

```

Messages

TRIGGER trgDivertUpdateEmployeeToBackup DIPANGGIL!

(0 row(s) affected)

(0 row(s) affected)

14. Soal-14

```

IF OBJECT_ID('HR.trgDivertDeleteEmployeeToBackup') IS NOT NULL
    DROP TABLE HR.trgDivertDeleteEmployeeToBackup
Go

CREATE TRIGGER trgDivertDeleteEmployeeToBackup ON HR.Employees
INSTEAD OF DELETE
AS
    PRINT 'TRIGGER trgDivertDeleteEmployeeToBackup DIPANGGIL!';
    DECLARE @empid INT = (SELECT empid FROM deleted);
    DECLARE @firstname VARCHAR(15) = (SELECT firstname FROM deleted);
    DECLARE @lastname VARCHAR(15) = (SELECT lastname FROM deleted);

    DELETE FROM HR.EmployeesBackup WHERE empid=@empid;

    PRINT 'Karyawan dengan nama '+@firstname+' '+@lastname+' dihapus
    di HR.EmployeesBackup saja. Di tabel aslinya tetap.';

```

```

DELETE FROM HR.Employees WHERE firstname='Maria';
select * from HR.EmployeesBackup;

```

Results	Messages
TRIGGER trgDivertDeleteEmployeeToBackup DIPANGGIL!	
(1 row(s) affected)	
Karyawan dengan nama Maria Cameron dihapus di HR.EmployeesBackup saja. Di tabel aslinya tetap.	
(1 row(s) affected)	

#Software : Saya menggunakan software DBeaver karena saya menggunakan linux sebagai sistem operasi. Namun pada saat penggunaan TRIGGER, DBeaver tidak dapat menampilkan hasil triggernya. Jadi, khusus untuk trigger saya menggunakan SQL SERVER Management Studio di laptop yang saya pinjam dari teman saya.