

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

JOIN TABLE, GROUPING, AGGREGATE

- ✓ ALIAS
- ✓ GROUP BY dan Aggregate Function
- ✓ Criteria Having, Where
- ✓ INNER JOIN , LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN, CROSS JOIN
- ✓ SUB Query
- ✓ UNION
- ✓ DERIVED TABLE
- ✓ VIEW, Implementasi pada EXCEL

Tujuan Mata Kuliah

- ✓ Mahasiswa dapat membuat pengelompokan data dengan aggregate fungsi
- ✓ Mahasiswa mengetahui penggabungan table-table dengan JOIN
- ✓ Mahasiswa mampu membuat kondisi dengan sub query

Tools yang digunakan

- ✓ SQL Server Management Studio 2005 Express
- ✓ SQLCMD

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

ALIAS

Dalam penerapan SELECT statement terkadang field yang akan di seleksi membutuhkan nama yang jelas dan dimengerti oleh user, maka kita dapat menggunakannya. Alias juga dapat di gunakan pada clause GROUP BY. Alias dapat menggunakan keyword AS ataupun tidak:

Contoh:

```
SELECT MaterialNumber 'Kode Barang' FROM ORDERS
```

Atau

```
SELECT MaterialNumber [Kode Barang] FROM ORDERS
```

Atau

```
SELECT MaterialNumber AS 'Kode Barang' FROM ORDERS
```

Atau

```
SELECT MaterialNumber AS [Kode Barang] FROM ORDERS
```

ALIAS Pada Table

Biasanya Alias pada table digunakan pada operasi join beberapa table. Ini digunakan untuk meringkas nama table agar lebih pendek dan mudah di pahami pada beberapa statement.

```
SELECT m.MaterialNumber, m.MaterialDescription  
FROM Material m  
WHERE m.MaterialNumber = 300004
```

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

ALIAS calculated

Alias juga biasanya digunakan pada kalkulasi ataupun fungsi

```
SELECT Price*QtyOrder TOTAL
FROM ORDERS
```

GROUP BY

Pada seleksi data pengelompokan data biasanya di sertakan dengan aggregate function. Dalam penggunaannya aggregate function harus diikuti oleh GROUP BY. Berikut function-function yang di maksud :

MIN()

Fungsi ini digunakan untuk mencari nilai minimum dari sekumpulan data.

```
SELECT field1,field..n, MIN(nama_field)
FROM TABLE
GROUP BY field1,field..n
```

Contoh menampilkan minimum qty penjualan per material number

```
SELECT MaterialNumber, MIN(QtyOrder) AS Minimum_Jual
FROM ORDERS
GROUP BY MaterialNumber
```

MAX()

Kebalikan dari MIN(). Fungsi ini digunakan untuk mencari nilai maximum dari sekumpulan data.

```
SELECT field1,field..n, MAX(nama field)
FROM TABLE
GROUP BY field1,field..n
```

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

Contoh menampilkan maximum qty penjualan per material number

```
SELECT MaterialNumber, MAX(QtyOrder) AS Maximum_Jual
FROM ORDERS
GROUP BY MaterialNumber
```

AVG()

Fungsi ini digunakan untuk mencari nilai qty rata-rata dari sekumpulan data

```
SELECT field1,field..n, AVG(nama_field)
FROM TABLE
GROUP BY field1,field..n
```

Contoh menampilkan minimum qty penjualan per material number

```
SELECT MaterialNumber, AVG(QtyOrder) AS Ratarata_Jual
FROM ORDERS
GROUP BY MaterialNumber
```

SUM()

Fungsi ini digunakan untuk mencari jumlah nilai dari sekumpulan data

```
SELECT field1,field..n, SUM(nama_field)
FROM TABLE
GROUP BY field1,field..n
```

Contoh menampilkan total qty penjualan per material number

```
SELECT MaterialNumber,SUM (QtyOrder) AS Total_Jual
FROM ORDERS
GROUP BY MaterialNumber
```

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

COUNT()

Fungsi ini digunakan untuk mencari nilai banyaknya data

```
SELECT field1,field..n, COUNT(nama_field)
FROM TABLE
GROUP BY field1,field..n
```

Contoh menampilkan minimum qty penjualan per material number

```
SELECT OrderNumber, COUNT(MaterialNumber) AS Banyak_Material
FROM ORDERS
GROUP BY OrderNumber
```

Penggunaan fungsi aggregate juga dapat digunakan tanpa menggunakan group by dengan mengembalikan scalar single row.

Contoh :

```
SELECT MAX(QtyOrder) Maksimum FROM ORDERS
```

LATIHAN 1

BUATLAH SQL STATEMENT DENGAN KRITERIA DIBAWAH INI.

1. Tampilkan minimum pembelian per material Urutkan dari qty minimum terkecil
2. Tampilkan maximum pembelian per vendor
3. Tampilkan minimum penjualan per customer
4. Tampilkan jumlah maximum pembelian dari seluruh data yang ada
5. Tampilkan rata-rata pembelian per material
6. Tampilkan rata-rata penjualan per customer
7. Tampilkan total jumlah penjualan dari semua transaksi
8. Tampilkan total jumlah penjualan per material per customer
9. Tampilkan berapa kali pembelian per material
10. Tampilkan berapa kali pembelian beserta jumlah pembeliannya untuk tiap material
11. Tampilkan minimum, maksimum, rata-rata, jumlah barang serta berapa kali terjadi pembelian untuk tiap kode barang

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

HAVING CRITERIA

Kriteria dengan having adalah kriteria seleksi data dengan menggunakan kata **having**, dimana kata tersebut diikuti dengan aggregate function.

```
SELECT field1,field..n, .Aggregate_function
FROM TABLE
GROUP BY field1,field..n
HAVING kriteria_aggerage_function
```

Contoh menampilkan banyaknya jenis material yang terjual lebih dari 1 jenis per transaksi

```
SELECT OrderNumber, COUNT(MaterialNumber) AS Banyak_Material
FROM ORDERS
GROUP BY OrderNumber
HAVING COUNT(MaterialNumber)>1
```

WHERE CRITERIA

Kriteria ini dilakukan setelah keyword 'WHERE' dan biasanya diikuti dengan comparasion.

Comparison dengan Operator (=, <>, >, >=, <, <=)

Digunakan untuk seleksi dengan membandingkan 2 nilai yang hasilnya berupa nilai logic.

Contoh : Menampilkan barang yang diterima dengan criteria qty lebih besar dari 2

```
SELECT * FROM RECEIPT WHERE qtyReceipt > 2
```

Contoh : Menampilkan barang yang diterima dengan criteria qty = 1

```
SELECT * FROM RECEIPT WHERE qtyReceipt = 1
```

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

Contoh : Menampilkan barang yang diterima dengan criteria qty >= 1

```
SELECT * FROM RECEIPT WHERE qtyReceipt >= 2
```

Contoh : Menampilkan barang dengan stock <> 0

```
SELECT * FROM MATERIAL WHERE currentStock <> 0
```

BETWEEN

Digunakan untuk seleksi dengan range tertentu.

Contoh : Menampilkan barang penerimaan dengan stock 2 s/d 5

```
SELECT * FROM RECEIPT WHERE qtyReceipt BETWEEN 2 AND 5
```

Contoh : Menampilkan barang penerimaan dengan antara tanggal 1 Okt 2012 s/d 10 Okt 2012

```
SELECT * FROM RECEIPT WHERE  
convert(varchar, receiptDate, 103) BETWEEN '01/10/2012' AND '10/10/2012'
```

IN / NOT IN

IN digunakan untuk filter dengan array value yang termasuk

Contoh : Menampilkan barang BMX dan IBM series X220 pada master material

```
SELECT * FROM M_MATERIAL WHERE MaterialNumber IN ('300002', '300009')
```

NOT IN digunakan untuk filter dengan array value yang tidak termasuk

Contoh : Menampilkan barang yang tidak termasuk pada material group KOMPUTER & ELECTRONIC

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

```
SELECT * FROM MATERIAL WHERE materialGroup not in ('KOM', 'ELC')
```

LIKE / NOT LIKE

Seleksi Penggunaan LIKE biasa digunakan untuk filter dengan pola.

Contoh: Menampilkan dengan awalan 'C'

```
SELECT * FROM M MATERIAL WHERE MaterialDescription LIKE 'C%'
```

Menampilkan dengan akhiran 'C'

```
SELECT * FROM M MATERIAL WHERE MaterialDescription LIKE '%C'
```

Menampilkan yang mengandung kata/huruf 'C'

```
SELECT * FROM M MATERIAL WHERE MaterialDescription LIKE '%C%'
```

Menampilkan dengan range karakter awal

```
SELECT MaterialDescription FROM M MATERIAL  
WHERE MaterialDescription LIKE '[P-S]%'
```

Menampilkan dengan range karakter akhir

```
SELECT MaterialDescription FROM M MATERIAL  
WHERE MaterialDescription LIKE '%[P-S]'
```

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

Menampilkan dengan range karakter opposite gunakan tanda '^'

```
SELECT MaterialDescription FROM M_MATERIAL  
WHERE MaterialDescription LIKE '[P-S]'
```

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

LATIHAN 2

BUATLAH SQL STATEMENT DENGAN KRITERIA DIBAWAH INI.

1. Tampilkan vendor yang mengirim barang sebanyak 2 kali
2. Tampilkan customer yang total pembeliannya lebih dari 6
3. Tampilkan material dan vendor yang memiliki qty receipt 2
4. Tampilkan vendor yang beralamat bukan di Amerika
5. Tampilkan data material yang mengandung kata 'galaxy'
6. Tampilkan data material yang berawalan kata 'samsung s'
7. Tampilkan data receipt dari tanggal 1 oktober 2012 s/d 10 oktober 2012
8. Tampilkan variant material pada table orders
9. Tampilkan data pada table orders dengan qty dari 2 s/d 4
10. Tampilkan material number yang tidak di beli
11. Tampilkan material yang tidak terjual pada tanggal 5 oktober 2012
12. Tampilkan customer yang membeli paling banyak qty dan sebutkan materialnya
13. Tampilkan qty bonus untuk material 'ibm series x255' pada tanggal 11 oktober 2012

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

SELECT dengan beberapa Table

Pada seleksi ini anda dapat menggabungkan beberapa table, dan yang paling penting adalah ada kunci yang sama antara table yang satu dengan table yang lainnya.

```
SELECT r.ReceiptNumber,
       r.MaterialNumber,
       m.MaterialDescription,
       r.Price,
       r.QtyReceipt,
       r.Price * r.QtyReceipt Total
FROM RECEIPT r, MATERIAL m
WHERE r.MaterialNumber = m.MaterialNumber
```

SUB QUERY

Sub query adalah query di dalam query, yang berarti menyeleksi data dari hasil seleksi data yang telah ada, syntak nya sama seperti query biasa hanya kondisi setelah where diikuti dengan query baru atau sub query.

```
SELECT field1, field..n,
FROM TABLE
WHERE kriteria (
    SELECT field1, field..n,
    FROM TABLE
    WHERE kriteria )
```

Contoh menampilkan material yang qty paling banyak terjual beserta nama customernya

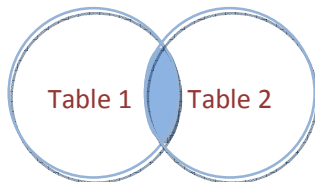
```
SELECT ORDERS.CUSTOMERNumber, CUSTOMER.CUSTOMERName, ORDERS.MaterialNumber,
       MATERIAL.MaterialDescription, ORDERS.QtyOrder
FROM ORDERS, MATERIAL, CUSTOMER
WHERE ORDERS.MaterialNumber = MATERIAL.MaterialNumber AND
       ORDERS.CUSTOMERNumber = CUSTOMER.CUSTOMERNumber AND
       QtyOrder=(
           SELECT MAX(QtyOrder) FROM ORDERS
       )
```

JOIN

JOIN adalah clause untuk menggabungkan antara satu table dengan table lainnya yang memiliki kondisi tertentu. Ada beberapa JOIN

- **JOIN/INNER JOIN**

JOIN/ INNER JOIN/ NORMAL JOIN/ EQUAL JOIN (=) bertujuan untuk mencari kesamaan data antara table yang satu dengan table lain.



```
SELECT Table1.field1, Table1.field...n
       Table1.field2, Table2.field...n

FROM   Table1,Table2, Table...n
WHERE  Table1.field_PK = Table2.fieldPK AND
       Table2.field_PK = TableN.fieldPK
```

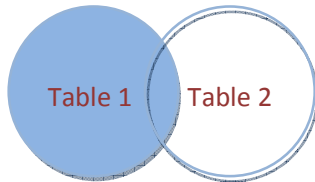
Atau

```
SELECT Table1.field1, Table1.field...n
       Table1.field2, Table2.field...n
       TableN.field1, TableN.field...n
FROM   Table1 INNER JOIN Table2 ON Table1.field PK = Table2.fieldPK
       INNER JOIN Tablen ON Table2.field PK = TableN.fieldPK
```

Contoh

```
SELECT MATERIAL.MaterialNumber, MATERIAL.MaterialDescription,
       MATERIAL_GROUP.Description
FROM   MATERIAL INNER JOIN MATERIAL_GROUP ON
       MATERIAL.MaterialGroup = MATERIAL_GROUP.MaterialGroup
```

- **LEFT OUTER JOIN**

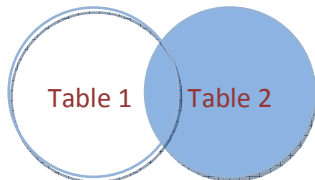


Digunakan untuk menampilkan semua data dari table1 (kiri), baik memiliki kesamaan seleksi kondisi ataupun tidak memiliki kesamaan pada table2 (kanan) . Jika tidak ada data yang sama pada kondisi di table2 (kanan) akan bernilai NULL

Contoh : Table 1 = MATERIAL, Table 2 = ORDERS.
menampilkan semua material yang terjual dan tidak terjual

```
SELECT MATERIAL.MaterialNumber, MATERIAL.MaterialDescription,  
        ORDERS.OrderDate,ORDERS.QtyOrder  
FROM    MATERIAL LEFT OUTER JOIN ORDERS ON  
        MATERIAL.MaterialNumber = ORDERS.MaterialNumber
```

- **RIGHT OUTER JOIN**



Digunakan untuk menampilkan semua data dari table2 (kanan), baik memiliki kesamaan seleksi kondisi ataupun tidak memiliki kesamaan pada table1 (kiri). Jika tidak ada data yang sama pada kondisi di table1 (kanan) akan bernilai NULL

Contoh : Table 1 = ORDERS ,Table 2 = MATERIAL.
menampilkan semua material yang terjual dan tidak terjual

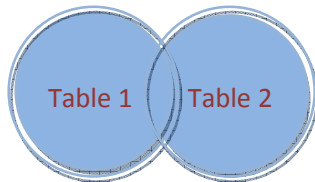
```
SELECT MATERIAL.MaterialNumber, MATERIAL.MaterialDescription,  
        ORDERS.OrderDate,ORDERS.QtyOrder  
FROM    ORDERS RIGHT OUTER JOIN MATERIAL ON  
        MATERIAL.MaterialNumber = ORDERS.MaterialNumber
```

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

- **FULL OUTER JOIN**



Kombinasi antara left outer join dengan right outer join NULL jika tidak ada kesamaan

- **CROSS JOIN**

Hasil dari cross join akan membuat hasil berdasarkan pada semua kemungkinan kombinasi baris data, maka jumlah baris data akan dikalikan. $N \times M$ dimana N adalah jumlah baris dalam kumpulan data pada table1 dan M adalah jumlah baris dalam kumpulan table2

	materialGroup	Description
1	ELC	ELECTRONIC
2	KOM	KOMPUTER
3	SPRT	SPORT

	MaterialGroup	MaterialNumber	MaterialDescription
1	SPRT	300001	SKATEBOARD
2	SPRT	300002	BMX
3	ELC	300003	IPHONE 5
4	ELC	300004	IPHONE 10
5	ELC	300005	IPHONE 15
6	ELC	300006	GALAXY TAB
7	KOM	300007	IBM series X350
8	KOM	300008	IBM series X255
9	KOM	300009	IBM series X220
10	KOM	300010	IBM series X360
11	KOM	300011	IBM series X370
12	ELC	300013	SAMSUNG GALAXY S2
13	ELC	300014	SAMSUNG GALAXY S3
14	ELC	300015	LG OPTIMUS 2X
15	ELC	300016	LG OPTIMUS ONE

The diagram illustrates a cross join between two tables. On the left, a table with 3 rows (materialGroup: ELC, KOM, SPRT) and 2 columns (materialGroup, Description) is shown. On the right, a table with 15 rows and 4 columns (MaterialGroup, MaterialNumber, MaterialDescription) is shown. Colored lines connect each row of the first table to every row of the second table, representing all possible combinations of rows from both tables.

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

	MaterialNumber	MaterialDescription	Description
1	300001	SKATEBOARD	ELECTRONIC
2	300002	BMX	ELECTRONIC
3	300003	IPHONE 5	ELECTRONIC
4	300004	IPHONE 10	ELECTRONIC
5	300005	IPHONE 15	ELECTRONIC
6	300006	GALAXY TAB	ELECTRONIC
7	300007	IBM series X350	ELECTRONIC
8	300008	IBM series X255	ELECTRONIC
9	300009	IBM series X220	ELECTRONIC
10	300010	IBM series X360	ELECTRONIC
11	300011	IBM series X370	ELECTRONIC
12	300013	SAMSUNG GALAXY S2	ELECTRONIC
13	300014	SAMSUNG GALAXY S3	ELECTRONIC
14	300015	LG OPTIMUS 2X	ELECTRONIC
15	300016	LG OPTIMUS ONE	ELECTRONIC
16	300001	SKATEBOARD	KOMPUTER
17	300002	BMX	KOMPUTER
18	300003	IPHONE 5	KOMPUTER
19	300004	IPHONE 10	KOMPUTER
20	300005	IPHONE 15	KOMPUTER

	MaterialNumber	MaterialDescription	Description
21	300006	GALAXY TAB	KOMPUTER
22	300007	IBM series X350	KOMPUTER
23	300008	IBM series X255	KOMPUTER
24	300009	IBM series X220	KOMPUTER
25	300010	IBM series X360	KOMPUTER
26	300011	IBM series X370	KOMPUTER
27	300013	SAMSUNG GALAXY S2	KOMPUTER
28	300014	SAMSUNG GALAXY S3	KOMPUTER
29	300015	LG OPTIMUS 2X	KOMPUTER
30	300016	LG OPTIMUS ONE	KOMPUTER
31	300001	SKATEBOARD	SPORT
32	300002	BMX	SPORT
33	300003	IPHONE 5	SPORT
34	300004	IPHONE 10	SPORT
35	300005	IPHONE 15	SPORT
36	300006	GALAXY TAB	SPORT
37	300007	IBM series X350	SPORT
38	300008	IBM series X255	SPORT
39	300009	IBM series X220	SPORT
40	300010	IBM series X360	SPORT
41	300011	IBM series X370	SPORT
42	300013	SAMSUNG GALAXY S2	SPORT
43	300014	SAMSUNG GALAXY S3	SPORT
44	300015	LG OPTIMUS 2X	SPORT
45	300016	LG OPTIMUS ONE	SPORT

```
SELECT MATERIAL.MaterialNumber, MATERIAL.MaterialDescription,  
        MATERIAL_GROUP.Description  
FROM    MATERIAL_GROUP CROSS JOIN MATERIAL
```


LATIHAN 3

BUATLAH SQL STATEMENT DENGAN KRITERIA DIBAWAH INI (gunakan ALIAS).

1. Tampilkan nama material dan tanggal berapa terjadinya dan berapa jumlahnya pada penerimaan barang
2. Tampilkan nama material, nama customer, tanggal berapa dan berapa jumlahnya pada transaksi penjualan
3. Tampilkan semua material beserta tanggal penjualan dan jumlahnya pada transaksi penjualan (material yang tidak terjual juga ditampilkan)
4. Tampilkan semua transaksi penjualan material, tampilkan juga material-material yang belum pernah dibeli oleh customer. Field yang ditampilkan tgl penjualan, material number, jumlahnya.

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

UNION

Union bertujuan untuk menggabungkan dua query atau lebih menjadi satu kesatuan dengan syarat harus memiliki kolom yang sama. Union tanpa diikuti keyword ALL, akan menghilangkan data yang sama (DISTINCT).

```
SELECT field1,field..n,  
FROM TABLE  
UNION [ALL]  
  
SELECT field1,field..n,  
FROM TABLE
```

Contoh menampilkan material yang dibeli dan yang di jual.

```
SELECT MaterialNumber,QtyReceipt  
FROM RECEIPT  
UNION ALL  
SELECT MaterialNumber,QtyOrder  
FROM ORDERS
```

UNION ALL		
	MaterialNumber	QtyReceipt
1	300001	5
2	300007	3
3	300008	2
4	300009	1
5	300010	7
6	300001	2
7	300004	9
8	300005	2
9	300015	8
10	300001	1
11	300007	2
12	300008	2
13	300009	1
14	300010	3
15	300001	4
16	300004	2
17	300005	4
18	300015	6

UNION		
	MaterialNumber	QtyReceipt
1	300001	1
2	300001	2
3	300001	4
4	300001	5
5	300004	2
6	300004	9
7	300005	2
8	300005	4
9	300007	2
10	300007	3
11	300008	2
12	300009	1
13	300010	3
14	300010	7
15	300015	6
16	300015	8

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

DERIVED TABLE (NESTED SELECT)

Derived table adalah nama internal dari sub query pada FROM clause. Deriver table juga diketahui sebagai sub query.

```
SELECT field1,field..n
FROM (
    SELECT field1,field..n
    FROM TABLE
)derived_table
```

Contoh:

```
SELECT KodeBarang,>NamaBarang
FROM (
    SELECT MaterialNumber KodeBarang,MaterialDescription>NamaBarang
    FROM MATERIAL
)material
```

Sesi1 : JOIN TABLE, GROUPING, AGGREGATE

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

LATIHAN 4

BUATLAH KARTU STOCK SEPerti TABLE DIBWAH DENGAN MENGGUNAKAN SQL STATEMENT

<u>Kode Barang</u>	<u>Nama Barang</u>	<u>Stock Awal</u>	<u>Masuk</u>	<u>Keluar</u>	<u>Stock Akhir</u>
--------------------	--------------------	-------------------	--------------	---------------	--------------------