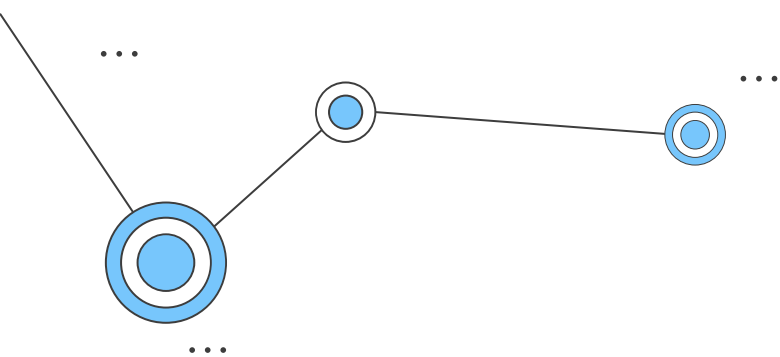


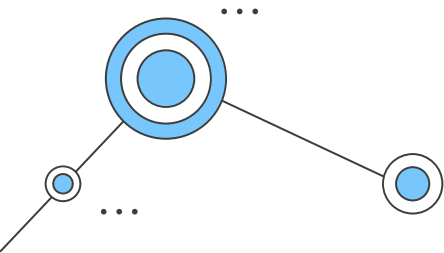
BASIS DATA LANJUT

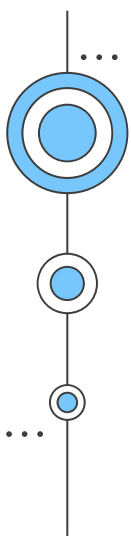
Pertemuan 8
Pivoting & Grouping Sets



OUTLINE

- PIVOT & UNPIVOT
- GROUPING SETS
- ROLLUP & CUBE





PIVOT & UNPIVOT





KONSEP PIVOT

- Secara umum, output SQL menghasilkan data secara vertical
- Dalam banyak kasus, menampilkan data secara horizontal akan lebih mudah dimengerti
- Operator PIVOT digunakan untuk memutar (rotate) data dari semula berorientasi row-based (vertical) menjadi berorientasi columns-based (horizontal)
- Nilai dari suatu kolom yang berbeda akan digunakan sebagai judul untuk kolom lainnya

KONSEP PIVOT

- Data secara vertikal

	Category	OrderYear	Quantity
1	Beverages	2006	1842
2	Beverages	2007	3996
3	Beverages	2008	3694
4	Condiments	2006	962
5	Condiments	2007	2895
6	Condiments	2008	1441
7	Confections	2006	1357
8	Confections	2007	4137
9	Confections	2008	2412
10	Dairy Products	2006	2086
11	Dairy Products	2007	4374
12	Dairy Products	2008	2689

- Data secara horizontal

	Category	2006	2007	2008
1	Beverages	1842	3996	3694
2	Condiments	962	2895	1441
3	Confections	1357	4137	2412
4	Dairy Products	2086	4374	2689
5	Grains/Cereals	549	2636	1377
6	Meat/Poultry	950	2189	1060
7	Produce	549	1583	858
8	Seafood	1286	3679	2716



MEKANISME PIVOTING

- Pilih base dataset/source untuk pivoting
- Buat temporary table dengan menggunakan derived table atau CTE (Common Table Expression)
- Gunakan PIVOT operator



SYNTAX PIVOT

```
SELECT <non-pivoted column>,  
    [first pivoted column] AS <column name>,  
    [second pivoted column] AS <column name>,  
    ...  
    [last pivoted column] AS <column name>  
FROM  
    (<SELECT query that produces the data>  
    AS <alias for the source query>  
PIVOT  
    (  
        <aggregation function>(<column being aggregated>  
FOR  
    [<column that contains the values that will become column headers>]  
        IN ( [first pivoted column], [second pivoted column],  
        ... [last pivoted column])  
    ) AS <alias for the pivot table>  
<optional ORDER BY clause>;
```



ELEMEN PIVOT

- GROUPING → menentukan kolom mana yang akan digunakan untuk pengelompokan data
- SPREADING → menentukan list nilai yang akan dijadikan judul kolom untuk hasil pivot
- AGGREGATION → menentukan fungsi agregasi (SUM dkk) yang akan digunakan pada baris data yang dikelompokkan

CONTOH

```
SELECT Category, [2006],[2007],[2008]
FROM
(
    SELECT Category, Quantity, OrderYear
    FROM Sales.CategoryQtyYear
) AS Ds
PIVOT
(
    SUM(Quantity) FOR OrderYear IN ([2006],[2007],[2008])
) AS Pvt;
```

GROUPING : Category
SPREADING : OrderYear IN ([2006],[2007],[2008])
AGGREGATION : SUM(Quantity)

	Category	2006	2007	2008
1	Beverages	1842	3996	3694
2	Condiments	962	2895	1441
3	Confections	1357	4137	2412
4	Dairy Products	2086	4374	2689
5	Grains/Cereals	549	2636	1377
6	Meat/Poultry	950	2189	1060
7	Produce	549	1583	858
8	Seafood	1286	3679	2716

CONTOH 2

	ID	SiswaID	Nama	MapelID	MataPelajaran	Nilai
1	1	1	Ani	1	Matematika	80
2	2	2	Budi	1	Matematika	85
3	3	1	Ani	2	Fisika	70
4	4	2	Budi	2	Fisika	90
5	5	1	Ani	3	Biologi	80
6	6	2	Budi	3	Biologi	75

	SiswaID	Nama	Matematika	Fisika	Biologi
1	1	Ani	80	70	80
2	2	Budi	85	90	75

GROUPING : SiswaID, Nama

SPREADING : MataPelajaran IN (Matematika, Fisika, Biologi)

AGGREGATION : SUM(Nilai)

CONTOH

	SiswaID	Nama	Matematika	Fisika	Biologi
1	1	Ani	80	70	80
2	2	Budi	85	90	75

```
SELECT SiswaID, Nama, Matematika, Fisika, Biologi
FROM (
    SELECT SiswaID, Nama, MataPelajaran, Nilai
    FROM NilaiSiswa
) AS Ds
PIVOT (
    AVG(Nilai) FOR MataPelajaran IN (Matematika, Fisika, Biologi)
) AS Pvt;
```

GROUPING : SiswaID, Nama

SPREADING : MataPelajaran IN (Matematika, Fisika, Biologi)

AGGREGATION : SUM(Nilai)

NOTE

- Semua kolom di data source yang tidak termasuk spreading dan aggregate akan dianggap sebagai grouping column


```
SELECT *  
FROM NilaiSiswa  
PIVOT (  
    AVG(Nilai) FOR MataPelajaran IN (Matematika, Fisika, Biologi)  
) AS PivotTable;
```

	ID	SiswaID	Nama	MapelID	Matematika	Fisika	Biologi
1	1	1	Ani	1	80	NULL	NULL
2	2	2	Budi	1	85	NULL	NULL
3	3	1	Ani	2	NULL	70	NULL
4	4	2	Budi	2	NULL	90	NULL
5	5	1	Ani	3	NULL	NULL	80
6	6	2	Budi	3	NULL	NULL	75



UNPIVOT

- Kebalikan dari logika PIVOT
- Mengembalikan data column-based menjadi row-based
- Untuk menggunakan operator UNPIVOT diperlukan:
 - ✓ kolom yang akan dilakukan UNPIVOT
 - ✓ nama untuk kolom baru yang akan menampilkan nilai UNPIVOT



```

SELECT Category, OrderYear, Quantity
FROM Sales.PivotCategoryYear
UNPIVOT (
    Quantity FOR OrderYear IN ([2006],[2007],[2008])
) AS P;

```

	Category	2006	2007	2008
1	Beverages	1842	3996	3694
2	Condiments	962	2895	1441
3	Confections	1357	4137	2412
4	Dairy Products	2086	4374	2689
5	Grains/Cereals	549	2636	1377
6	Meat/Poultry	950	2189	1060
7	Produce	549	1583	858
8	Seafood	1286	3679	2716

	Category	OrderYear	Quantity
1	Beverages	2006	1842
2	Beverages	2007	3996
3	Beverages	2008	3694
4	Condiments	2006	962
5	Condiments	2007	2895
6	Condiments	2008	1441
7	Confections	2006	1357
8	Confections	2007	4137
9	Confections	2008	2412
10	Dairy Products	2006	2086
11	Dairy Products	2007	4374
12	Dairy Products	2008	2689

```

SELECT SiswaID, Nama, MataPelajaran, Nilai
FROM (
    SELECT SiswaID, Nama, Matematika, Fisika, Biologi
    FROM PivotNilaiSiswa
) AS Ds
UNPIVOT (
    Nilai FOR MataPelajaran IN (Matematika, Fisika, Biologi)
) AS Upv;

```

	SiswaID	Nama	Matematika	Fisika	Biologi
1	1	Ani	80	70	80
2	2	Budi	85	90	75

	ID	SiswaID	Nama	MapelID	MataPelajaran	Nilai
1	1	1	Ani	1	Matematika	80
2	2	2	Budi	1	Matematika	85
3	3	1	Ani	2	Fisika	70
4	4	2	Budi	2	Fisika	90
5	5	1	Ani	3	Biologi	80
6	6	2	Budi	3	Biologi	75

GROUPING SETS, CUBE, & ROLLUP




GROUPING SETS

- Operator UNION digunakan untuk menggabungkan beberapa query
- GROUPING SET merupakan sub-clausa dari GROUP BY
- GROUPING SET memungkinkan banyak pengelompokkan untuk didefinisikan di dalam satu query



SYNTAX


```
SELECT <column list with aggregate(s)>
FROM <source>
GROUP BY
GROUPING SETS
(
    <column_name>,--one or more columns
    <column_name>,--one or more columns
    () -- empty parentheses if aggregating all rows
);
```





GROUPING SETS

ID	Kota	Produk	Jumlah
1	Jakarta	Laptop	300
2	Jakarta	Smartphone	200
3	Bandung	Laptop	150
4	Bandung	Tablet	100
5	Surabaya	Smartphone	250
6	Surabaya	Laptop	400





TANPA GROUPING SET

```
SELECT Kota, NULL AS Produk, SUM(Jumlah) AS Total
FROM Penjualan
GROUP BY Kota
```

```
UNION ALL
```

```
SELECT NULL, Produk, SUM(Jumlah) AS Total
FROM Penjualan
GROUP BY Produk
```

```
UNION ALL
```

```
SELECT NULL, NULL, SUM(Jumlah) AS Total
FROM Penjualan
```

Kota	Produk	Total
Jakarta	NULL	500
Bandung	NULL	250
Surabaya	NULL	650
NULL	Laptop	850
NULL	Smartphone	450
NULL	Tablet	100
NULL	NULL	1400



DENGAN GROUPING SET

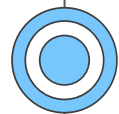
```
SELECT Kota, Produk, SUM(Jumlah) AS Total
FROM Penjualan
GROUP BY
GROUPING SETS (
    (Kota),
    (Produk),
    ()
)
ORDER BY Kota, Produk;
```

Kota	Produk	Total
Bandung	NULL	250
Jakarta	NULL	500
Surabaya	NULL	650
NULL	Laptop	850
NULL	Smartphone	450
NULL	Tablet	100
NULL	NULL	1400



ROLLUP VS CUBE

- ROLLUP dan CUBE merupakan sub-clausa dari GROUP BY
- Sama seperti GROUPING SETS, ROLLUP dan CUBE mendukung multiple grouping
- ROLLUP → menampilkan kombinasi dari set pengelompokkan dengan membentuk suatu hierarki
- CUBE → menampilkan semua kombinasi yang mungkin dari set pengelompokkan



ROLLUP Vs CUBE

- ROLLUP

```
(d1, d2, d3)
```

```
(d1, d2)
```

```
(d1)
```

```
()
```

- CUBE

```
(d1, d2, d3)
```

```
(d1, d2)
```

```
(d2, d3)
```

```
(d1, d3)
```

```
(d1)
```

```
(d2)
```

```
(d3)
```


```
()
```



ROLLUP

```
SELECT Kota, Produk, SUM(Jumlah) AS Total
FROM Penjualan
GROUP BY ROLLUP (Kota, Produk);
```

```
SELECT Kota, Produk, SUM(Jumlah) AS Total
FROM Penjualan
GROUP BY
GROUPING SETS (
    (Kota, Produk),
    (Kota),
    ()
);
```




Kota	Produk	Total
Jakarta	Laptop	300
Jakarta	Smartphone	200
Jakarta	NULL	500
Bandung	Laptop	150
Bandung	Tablet	100
Bandung	NULL	250
Surabaya	Smartphone	250
Surabaya	Laptop	400
Surabaya	NULL	650
NULL	NULL	1400



CUBE

```
SELECT Kota, Produk, SUM(Jumlah) AS Total
FROM Penjualan
GROUP BY CUBE (Kota, Produk);
```

```
SELECT Kota, Produk, SUM(Jumlah) AS Total
FROM Penjualan
GROUP BY
GROUPING SETS (
    (Kota, Produk),
    (Kota),
    (Produk),
    ()
);
```



Kota	Produk	Total
Jakarta	Laptop	300
Jakarta	Smartphone	200
Jakarta	NULL	500
Bandung	Laptop	150
Bandung	Tablet	100
Bandung	NULL	250
Surabaya	Smartphone	250
Surabaya	Laptop	400
Surabaya	NULL	650
NULL	Laptop	850
NULL	Smartphone	450
NULL	NULL	1300

Thanks!

Do you have any questions?



Team Teaching Matakuliah Basis Data Lanjut
JTI POLINEMA