

# **JOBSHEET**

## **PRAKTIKUM BASIS DATA LANJUT**

**Jurusan Teknologi Informasi**  
**POLITEKNIK NEGERI MALANG**



## **PERTEMUAN 7**

**SQL SERVER - Window Ranking, Offset, Fungsi Agregat**

### **Team Teaching:**

Dwi Puspitasari, S.Kom., M.Kom.

Yan Watequlis Syaifudin, ST., MMT., PhD.

Ariadi Retno Ririd, S.Kom., M.Kom

Eka Larasati Amalia, S.ST., MT.

Annisa Puspa Kirana, S. Kom, M.Kom

Milyun Ni'ma Shoumi, S.Kom., M.Kom

Dika Rizky Yunianto, S.Kom, M.Kom

Irsyad Arif Mashudi, S.Kom M.Kom

Muhammad Shulhan Khairy, S.Kom, M.Kom

Yoppy Yunhasnawa, S.ST., M.Sc.

Alvionitha Sari Agstringtyas, S.Kom.,M.Tr.T



### Topik

1. Membuat Window dengan OVER
2. Melakukan eksplorasi Fungsi Window

### Tujuan

1. Mahasiswa memahami cara menjelaskan komponen T-SQL yang digunakan untuk mendefinisikan window dan hubungan kedua hal tersebut
2. Mahasiswa memahami cara menuliskan query menggunakan klausa OVER dengan *partitioning*, *ordering*, dan *framing* untuk mendefinisikan window
3. Mahasiswa memahami cara menuliskan query menggunakan fungsi window agregat
4. Mahasiswa memahami cara menuliskan query menggunakan fungsi window ranking
5. Mahasiswa memahami cara menuliskan query menggunakan fungsi window offset

### Petunjuk Umum

1. Ikuti langkah-langkah pada bagian-bagian praktikum sesuai dengan urutan yang diberikan.
2. Anda dapat menggunakan SQL Server 2012 Standard Edition untuk mencoba praktikum pada jobsheet ini. Sesuaikan dengan kondisi komputer Anda.
3. Jawablah semua pertanyaan bertanda **[Soal-X]** yang terdapat pada langkah-langkah tertentu di setiap bagian praktikum.
4. Dalam setiap langkah pada praktikum terdapat penjelasan yang akan membantu Anda dalam menjawab pertanyaan-pertanyaan pada petunjuk nomor 3, maka baca dan kerjakanlah semua bagian praktikum dalam jobsheet ini.
5. Tulis jawaban dari soal-soal pada petunjuk nomor 3 pada sebuah laporan yang dikerjakan menggunakan aplikasi word processing (Word, OpenOffice, atau yang lain yang sejenis). Ekspor sebagai file **PDF** dengan format nama sebagai berikut:
  - **BDL\_09\_Kelas\_NamaLengkapAnda.pdf**
  - Contoh:
    - o **BDL\_09\_TI2U\_Mukiyo.pdf**
  - Perhatikan baik-baik format penamaanya.
  - Kumpulkan file PDF tersebut sebagai laporan praktikum kepada dosen pengampu.
  - Selain pada nama file, cantumkan juga identitas Anda pada halaman pertama laporan tersebut.

### Praktikum – Bagian 1: Menulis Query Menggunakan Fungsi RANKING

Langkah	Keterangan
1	<p>Skenario :</p> <p>Bagian penjualan ingin menentukan pesanan berdasarkan nilai masing-masing pelanggan. Untuk itu diperlukan laporan menggunakan fungsi RANK (termasuk kolom hasil perhitungan yang menambahkan kolom hasil perhitungan untuk menampilkan nomor baris dengan klausa SELECT).</p> <p>Untuk melakukan percobaan pada praktikum bagian 1 ini terlebih dahulu lakukan login pada SQL Server Management Studio (SSMS). Pastikan database terhubung dengan "TSQL".</p>

**[Soal-1]** Tulislah pernyataan SELECT untuk mengambil kolom orderid, orderdate, dan val serta kolom hasil perhitungan bernama rowno dari view Sales.OrderValues! Gunakan fungsi ROW\_NUMBER untuk mengembalikan rowno, urutkan nomor baris berdasarkan kolom orderdate!

```
SELECT
    orderid, orderdate, val,
    ROW_NUMBER() OVER(order by orderdate) AS rowno
FROM Sales.OrderValues;
```

orderid	orderdate	val	rowno
...	...	...	...
11075	2008-05-06 00:00:00.000	498.10	828
11076	2008-05-06 00:00:00.000	792.75	829
11077	2008-05-06 00:00:00.000	1255.72	830

(830 row(s) affected)

2

	orderid	orderdate	val	rowno
1	10248	2006-07-04 00:00:00.000	440.00	1
2	10249	2006-07-05 00:00:00.000	1863.40	2
3	10250	2006-07-08 00:00:00.000	1552.60	3
4	10251	2006-07-08 00:00:00.000	654.06	4
5	10252	2006-07-09 00:00:00.000	3597.90	5
6	10253	2006-07-10 00:00:00.000	1444.80	6
7	10254	2006-07-11 00:00:00.000	556.62	7
8	10255	2006-07-12 00:00:00.000	2490.50	8
9	10256	2006-07-15 00:00:00.000	517.80	9
10	10257	2006-07-16 00:00:00.000	1119.90	10
11	10258	2006-07-17 00:00:00.000	1614.88	11

Query executed successfully. | MENTARI-PC\MENTARI (11.0 SP2) | MENTARI-PC\TOSHIBA (52) | TSQL2012 | 00:00:00 | 830 rows

**[Soal-2]** Salin T-SQL pada soal no 1. Kemudian modifikasi dengan memasukkan kolom tambahan bernama rankno. Untuk membuat rankno gunakan fungsi RANK dengan urutan peringkat berdasarkan kolom orderdate!

53 - Lab Exercise 1 - Task 2 Result.txt

orderid	orderdate	val	rowno	rankno
...	...	...	...	...
10248	2006-07-04 00:00:00.000	440.00	1	1
10249	2006-07-05 00:00:00.000	1863.40	2	2
10250	2006-07-08 00:00:00.000	1552.60	3	3
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
11075	2008-05-06 00:00:00.000	498.10	828	827
11076	2008-05-06 00:00:00.000	792.75	829	827
11077	2008-05-06 00:00:00.000	1255.72	830	827

(830 row(s) affected)

3

orderid	orderdate	val	rowno	rankno
1	10248	2006-07-04 00:00:00.000	440.00	1
2	10249	2006-07-05 00:00:00.000	1863.40	2
3	10250	2006-07-08 00:00:00.000	1552.60	3
4	10251	2006-07-08 00:00:00.000	654.06	4
5	10252	2006-07-09 00:00:00.000	3597.90	5
6	10253	2006-07-10 00:00:00.000	1444.80	6
7	10254	2006-07-11 00:00:00.000	556.62	7
8	10255	2006-07-12 00:00:00.000	2490.50	8
9	10256	2006-07-15 00:00:00.000	517.80	9
10	10257	2006-07-16 00:00:00.000	1119.90	10
11	10258	2006-07-17 00:00:00.000	1614.88	11

Query executed successfully. MENTARI-PC\MENTARI (11.0 SP2) MENTARI-PC\TOSHIBA (52) TSQL2012 00:00:00 830 rows

```

SELECT
    orderid, orderdate, val,
    ROW_NUMBER() OVER(order by orderdate) AS rowno,
    RANK() OVER(order by orderdate) AS rankno
FROM Sales.OrderValues;

```

4 [Soal-3] Apakah perbedaan antara fungsi RANK dan fungsi ROW\_NUMBER?

Jika ROW\_NUMBER() akan menghitung secara urut perbarisnya, sedangkan jika RANK() apabila orderdate ada yang sama maka ranknya akan sama

[Soal-4] Tuliskan pernyataan SELECT untuk mengambil kolom orderid, orderdate, custid, dan val serta hitung kolom bernama orderrankno dari view Sales.OrderValues. Kolom orderrankno harus menampilkan rangking per pelanggan secara independen, berdasarkan pemesanan val dalam urutan menurun!

54 - Lab Exercise 1 - Task 3 Result.txt

orderid	orderdate	custid	val	orderrankno
11011	2008-04-09 00:00:00.000	1	933.50	1
10692	2007-10-03 00:00:00.000	1	878.00	2
10835	2008-01-15 00:00:00.000	1	845.80	3
...				
...				
...				
10906	2008-02-25 00:00:00.000	91	427.50	5
10792	2007-12-23 00:00:00.000	91	399.85	6
10870	2008-02-04 00:00:00.000	91	160.00	7

(830 row(s) affected)

5

	orderid	orderdate	custid	val	orderrankno
1	11011	2008-04-09 00:00:00.000	1	933.50	1
2	10692	2007-10-03 00:00:00.000	1	878.00	2
3	10835	2008-01-15 00:00:00.000	1	845.80	3
4	10643	2007-08-25 00:00:00.000	1	814.50	4
5	10952	2008-03-16 00:00:00.000	1	471.20	5
6	10702	2007-10-13 00:00:00.000	1	330.00	6
7	10926	2008-03-04 00:00:00.000	2	514.40	1
8	10625	2007-08-08 00:00:00.000	2	479.75	2
9	10759	2007-11-28 00:00:00.000	2	320.00	3
10	10308	2006-09-18 00:00:00.000	2	88.80	4
11	10573	2007-06-19 00:00:00.000	3	2082.00	1

Query executed successfully.

MENTARI-PC\MENTARI (11.0 SP2)

MENTARI-PC\TOSHIBA (52)

TSQL2012

00:00:00

830 rows

```
SELECT
   orderid, orderdate, custid, val,
    RANK() OVER(partition by custid order by val desc) AS orderrankno
FROM Sales.OrderValues;
```

6

[Soal-5] Tuliskan pernyataan SELECT untuk mengambil kolom custid dan val dari view Sales.OrderValues. Tambahkan dua kolom berikut:

- 1) orderyear sebagai tahun dari kolom orderdate
- 2) orderrankno sebagai nomor urut, dipartisi berdasarkan pelanggan dan tahun pesanan, dan diurutkan berdasarkan nilai pesanan dalam urutan menurun!

55 - Lab Exercise 1 - Task 4 Result.txt

custid	val	orderyear	orderrankno
1	878.00	2007	1
1	814.50	2007	2
1	330.00	2007	3
...			
...			
...			
91	591.60	2008	2
91	427.50	2008	3
91	160.00	2008	4

(830 row(s) affected)

Results		Messages	
custid	val	orderyear	orderrankno
1	878.00	2007	1
2	814.50	2007	2
3	330.00	2007	3
4	933.50	2008	1
5	845.80	2008	2
6	471.20	2008	3
7	88.80	2006	1
8	479.75	2007	1
9	320.00	2007	2
10	514.40	2008	1
11	403.20	2006	1

Query executed successfully. MENTARI-PC\MENTARI (11.0 SP2) MENTARI-PC\TOSHIBA (52) TSQL2012 00:00:00 830 rows

Results		Messages	
custid	val	orderyear	orderrankno
1	878.00	2007	1
2	814.50	2007	2
3	330.00	2007	3
4	933.50	2008	1
5	845.80	2008	2
6	471.20	2008	3
7	88.80	2006	1
8	479.75	2007	1
9	320.00	2007	2
10	514.40	2008	1
11	403.20	2006	1
12	2082.00	2007	1
13	1940.85	2007	2
14	813.37	2007	3
15	749.06	2007	4
16	375.50	2007	5
17	860.00	2008	1

Query executed successfully.

```
-- SOAL 5
SELECT
    custid,
    val,
    YEAR (orderdate) AS orderyear,
    RANK() OVER (PARTITION BY custid, YEAR(orderdate) ORDER BY val DESC) AS orderrankno
FROM Sales.OrderValues;
```

7

**[Soal-6]** Salin query jawaban soal nomor 6 dan modifikasi untuk memfilter hanya pesanan dengan dua peringkat paling awal berdasarkan kolom orderrankno!

56 - Lab Exercise 1 - Task 5 Result.txt

custid	orderyear	orderrankno	val
1	2007	1	878.00
1	2007	2	814.50
1	2008	1	933.50
...			
...			
...			
91	2007	2	399.85
91	2008	1	686.00
91	2008	2	591.60

(418 row(s) affected)

Results		Messages	
custid	orderyear	orderrankno	val
1	1	2007	1
2	1	2007	2
3	1	2008	1
4	1	2008	2
5	2	2006	1
6	2	2007	1
7	2	2007	2
8	2	2008	1
9	3	2006	1
10	3	2007	1
11	3	2007	2

Query executed successfully. MENTARI-PC\MENTARI (11.0 SP2) MENTARI-PC\TOSHIBA (52) TSQL2012 00:00:00 418 rows

Results		Messages	
custid	orderyear	orderrankno	val
1	1	2007	1
2	1	2007	2
3	1	2008	1
4	1	2008	2
5	2	2006	1
6	2	2007	1
7	2	2007	2
8	2	2008	1
9	3	2006	1
10	3	2007	1
11	3	2007	2
12	3	2008	1
13	4	2006	1
14	4	2006	2
15	4	2007	1
16	4	2007	2
17	4	2008	1

```
-- SOAL 6
WITH RankedOrders AS(
    SELECT
        custid,
        val,
        YEAR (orderdate) AS orderyear,
        RANK() OVER (PARTITION BY custid, YEAR(orderdate) ORDER BY val DESC) AS orderrankno
    FROM Sales.OrderValues
)
SELECT
    custid,
    orderyear,
    orderrankno,
    val
FROM RankedOrders
WHERE orderrankno <= 2;
```

8

**Kesimpulan :** Setelah menjalankan praktikum bagian ini, mahasiswa mengetahui bagaimana menggunakan fungsi ranking pada pernyataan T-SQL.

## Praktikum – Bagian 2: Menulis Query Menggunakan Fungsi OFFSET

Langkah	Keterangan																																																																																																																					
1	<p>Skenario :</p> <p>Laporan lainnya diperlukan untuk menganalisis perbedaan antara dua baris secara berturut-turut. Hal ini akan mempermudah <i>business user</i> untuk menganalisa pertumbuhan dan tren.</p> <p>Untuk melakukan percobaan pada praktikum bagian 2 ini, pastikan database terhubung dengan "TSQL".</p>																																																																																																																					
2	<p>[Soal-7] Buatlah (<i>common table expression</i>) CTE dengan nama OrderRows berdasarkan query yang mengambil kolom orderid, orderdate, and val dari view Sales.OrderValues. Tambahkan kolom hasil perhitungan dengan nama rowno menggunakan fungsi ROW_NUMBER yang diurutkan berdasarkan kolom orderdate dan orderid!</p> <pre>WITH OrderRows AS(     SELECT         orderid,         orderdate,         val,         ROW_NUMBER() OVER(ORDER BY orderdate, orderid) AS rowno     FROM Sales.OrderValues ) SELECT     orderid,     orderdate,     val,     rowno FROM OrderRows;</pre>																																																																																																																					
3	<p>[Soal-8] Tuliskan pernyataan SELECT terhadap CTE dan gunakan LEFT JOIN dengan CTE yang sama untuk mengambil baris saat ini (<i>current row</i>) dan baris sebelumnya (<i>previous row</i>) berdasarkan kolom rowno. Kembalikan kolom orderid, orderdate, and val untuk baris saat ini dan kolom val untuk baris sebelumnya sebagai prevval. Tambahkan kolom hasil perhitungan dengan nama diffprev yang menunjukkan perbedaan antara val saat ini dengan sebelumnya!</p> <div><div>62 - Lab Exercise 2 - Task 1 Result.txt</div><table><thead><tr><th>orderid</th><th>orderdate</th><th>val</th><th>prevval</th><th>diffprev</th></tr></thead><tbody><tr><td>10248</td><td>2006-07-04 00:00:00.000</td><td>440.00</td><td>NULL</td><td>NULL</td></tr><tr><td>10249</td><td>2006-07-05 00:00:00.000</td><td>1863.40</td><td>440.00</td><td>1423.40</td></tr><tr><td>10250</td><td>2006-07-08 00:00:00.000</td><td>1552.60</td><td>1863.40</td><td>-310.80</td></tr><tr><td>...</td><td></td><td></td><td></td><td></td></tr><tr><td>...</td><td></td><td></td><td></td><td></td></tr><tr><td>11075</td><td>2008-05-06 00:00:00.000</td><td>498.10</td><td>232.09</td><td>266.01</td></tr><tr><td>11076</td><td>2008-05-06 00:00:00.000</td><td>792.75</td><td>498.10</td><td>294.65</td></tr><tr><td>11077</td><td>2008-05-06 00:00:00.000</td><td>1255.72</td><td>792.75</td><td>462.97</td></tr></tbody></table><p>(830 row(s) affected)</p></div> <div><div>Results</div><table><thead><tr><th></th><th>orderid</th><th>orderdate</th><th>val</th><th>prevval</th><th>diffprev</th></tr></thead><tbody><tr><td>1</td><td>10248</td><td>2006-07-04 00:00:00.000</td><td>440.00</td><td>NULL</td><td>NULL</td></tr><tr><td>2</td><td>10249</td><td>2006-07-05 00:00:00.000</td><td>1863.40</td><td>440.00</td><td>1423.40</td></tr><tr><td>3</td><td>10250</td><td>2006-07-08 00:00:00.000</td><td>1552.60</td><td>1863.40</td><td>-310.80</td></tr><tr><td>4</td><td>10251</td><td>2006-07-08 00:00:00.000</td><td>654.06</td><td>1552.60</td><td>-898.54</td></tr><tr><td>5</td><td>10252</td><td>2006-07-09 00:00:00.000</td><td>3597.90</td><td>654.06</td><td>2943.84</td></tr><tr><td>6</td><td>10253</td><td>2006-07-10 00:00:00.000</td><td>1444.80</td><td>3597.90</td><td>-2153.10</td></tr><tr><td>7</td><td>10254</td><td>2006-07-11 00:00:00.000</td><td>556.62</td><td>1444.80</td><td>-888.18</td></tr><tr><td>8</td><td>10255</td><td>2006-07-12 00:00:00.000</td><td>2490.50</td><td>556.62</td><td>1933.88</td></tr><tr><td>9</td><td>10256</td><td>2006-07-15 00:00:00.000</td><td>517.80</td><td>2490.50</td><td>-1972.70</td></tr><tr><td>10</td><td>10257</td><td>2006-07-16 00:00:00.000</td><td>1119.90</td><td>517.80</td><td>602.10</td></tr><tr><td>11</td><td>10258</td><td>2006-07-17 00:00:00.000</td><td>1614.88</td><td>1119.90</td><td>494.98</td></tr></tbody></table><div>Query executed successfully.   MENTARI-PC\MENTARI (11.0 SP2)   MENTARI-PC\TOSHIBA (52)   TSQL2012   00:00:00   830 rows</div></div>	orderid	orderdate	val	prevval	diffprev	10248	2006-07-04 00:00:00.000	440.00	NULL	NULL	10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40	10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80	...					...					11075	2008-05-06 00:00:00.000	498.10	232.09	266.01	11076	2008-05-06 00:00:00.000	792.75	498.10	294.65	11077	2008-05-06 00:00:00.000	1255.72	792.75	462.97		orderid	orderdate	val	prevval	diffprev	1	10248	2006-07-04 00:00:00.000	440.00	NULL	NULL	2	10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40	3	10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80	4	10251	2006-07-08 00:00:00.000	654.06	1552.60	-898.54	5	10252	2006-07-09 00:00:00.000	3597.90	654.06	2943.84	6	10253	2006-07-10 00:00:00.000	1444.80	3597.90	-2153.10	7	10254	2006-07-11 00:00:00.000	556.62	1444.80	-888.18	8	10255	2006-07-12 00:00:00.000	2490.50	556.62	1933.88	9	10256	2006-07-15 00:00:00.000	517.80	2490.50	-1972.70	10	10257	2006-07-16 00:00:00.000	1119.90	517.80	602.10	11	10258	2006-07-17 00:00:00.000	1614.88	1119.90	494.98
orderid	orderdate	val	prevval	diffprev																																																																																																																		
10248	2006-07-04 00:00:00.000	440.00	NULL	NULL																																																																																																																		
10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40																																																																																																																		
10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80																																																																																																																		
...																																																																																																																						
...																																																																																																																						
11075	2008-05-06 00:00:00.000	498.10	232.09	266.01																																																																																																																		
11076	2008-05-06 00:00:00.000	792.75	498.10	294.65																																																																																																																		
11077	2008-05-06 00:00:00.000	1255.72	792.75	462.97																																																																																																																		
	orderid	orderdate	val	prevval	diffprev																																																																																																																	
1	10248	2006-07-04 00:00:00.000	440.00	NULL	NULL																																																																																																																	
2	10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40																																																																																																																	
3	10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80																																																																																																																	
4	10251	2006-07-08 00:00:00.000	654.06	1552.60	-898.54																																																																																																																	
5	10252	2006-07-09 00:00:00.000	3597.90	654.06	2943.84																																																																																																																	
6	10253	2006-07-10 00:00:00.000	1444.80	3597.90	-2153.10																																																																																																																	
7	10254	2006-07-11 00:00:00.000	556.62	1444.80	-888.18																																																																																																																	
8	10255	2006-07-12 00:00:00.000	2490.50	556.62	1933.88																																																																																																																	
9	10256	2006-07-15 00:00:00.000	517.80	2490.50	-1972.70																																																																																																																	
10	10257	2006-07-16 00:00:00.000	1119.90	517.80	602.10																																																																																																																	
11	10258	2006-07-17 00:00:00.000	1614.88	1119.90	494.98																																																																																																																	



Results Messages					
	orderid	orderdate	val	prevval	diffprev
1	10248	2006-07-04 00:00:00.000	440.00	NULL	NULL
2	10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40
3	10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80
4	10251	2006-07-08 00:00:00.000	654.06	1552.60	-898.54
5	10252	2006-07-09 00:00:00.000	3597.90	654.06	2943.84
6	10253	2006-07-10 00:00:00.000	1444.80	3597.90	-2153.10
7	10254	2006-07-11 00:00:00.000	556.62	1444.80	-888.18
8	10255	2006-07-12 00:00:00.000	2490.50	556.62	1933.88
9	10256	2006-07-15 00:00:00.000	517.80	2490.50	-1972.70
10	10257	2006-07-16 00:00:00.000	1119.90	517.80	602.10
11	10258	2006-07-17 00:00:00.000	1614.88	1119.90	494.98
12	10259	2006-07-18 00:00:00.000	100.80	1614.88	-1514.08
13	10260	2006-07-19 00:00:00.000	1504.65	100.80	1403.85
14	10261	2006-07-19 00:00:00.000	448.00	1504.65	-1056.65
15	10262	2006-07-22 00:00:00.000	584.00	448.00	136.00
16	10263	2006-07-23 00:00:00.000	1873.80	584.00	1289.80
17	10264	2006-07-24 00:00:00.000	605.62	1873.80	-1178.17

Query executed successfully.

```
-- SOAL 8
WITH orderrows as
(
    SELECT orderid, orderdate, val,
    ROW_NUMBER() OVER(order by orderdate, orderid) rowno
    from Sales.OrderValues
)
SELECT o.orderid, o.orderdate, o.val,
    o2.val as prevval,
    o.val - o2.val as diffprev
FROM orderrows AS o
LEFT OUTER JOIN orderrows as o2 ON o.rowno=o2.rowno+1;
```

4

[Soal-9] Tuliskan pernyataan SELECT menggunakan fungsi LAG untuk mendapat hasil yang sama dengan query pada soal no.2! Query tidak yang dibuat pada soal ini tidak menggunakan CTE.

63 - Lab Exercise 2 - Task 2 Result.txt\*

orderid	orderdate	val	prevval	diffprev
10248	2006-07-04 00:00:00.000	440.00	NULL	NULL
10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40
10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80
...				
...				
11075	2008-05-06 00:00:00.000	498.10	232.09	266.01
11076	2008-05-06 00:00:00.000	792.75	498.10	294.65
11077	2008-05-06 00:00:00.000	1255.72	792.75	462.97

(830 row(s) affected)

Results Messages					
	orderid	orderdate	val	prevval	diffprev
1	10248	2006-07-04 00:00:00.000	440.00	NULL	NULL
2	10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40
3	10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80
4	10251	2006-07-08 00:00:00.000	654.06	1552.60	-898.54
5	10252	2006-07-09 00:00:00.000	3597.90	654.06	2943.84
6	10253	2006-07-10 00:00:00.000	1444.80	3597.90	-2153.10
7	10254	2006-07-11 00:00:00.000	556.62	1444.80	-888.18
8	10255	2006-07-12 00:00:00.000	2490.50	556.62	1933.88
9	10256	2006-07-15 00:00:00.000	517.80	2490.50	-1972.70
10	10257	2006-07-16 00:00:00.000	1119.90	517.80	602.10
11	10258	2006-07-17 00:00:00.000	1614.88	1119.90	494.98

Query executed successfully.

MENTARI-PC\MENTARI (11.0 SP2) | MENTARI-PC\TOSHIBA (S2) | TSQL2012 | 00:00:00 | 830 rows

Results Messages					
	orderid	orderdate	val	prevval	diffprev
1	10248	2006-07-04 00:00:00.000	440.00	NULL	NULL
2	10249	2006-07-05 00:00:00.000	1863.40	440.00	1423.40
3	10250	2006-07-08 00:00:00.000	1552.60	1863.40	-310.80
4	10251	2006-07-08 00:00:00.000	654.06	1552.60	-898.54
5	10252	2006-07-09 00:00:00.000	3597.90	654.06	2943.84
6	10253	2006-07-10 00:00:00.000	1444.80	3597.90	-2153.10
7	10254	2006-07-11 00:00:00.000	556.62	1444.80	-888.18
8	10255	2006-07-12 00:00:00.000	2490.50	556.62	1933.88
9	10256	2006-07-15 00:00:00.000	517.80	2490.50	-1972.70
10	10257	2006-07-16 00:00:00.000	1119.90	517.80	602.10
11	10258	2006-07-17 00:00:00.000	1614.88	1119.90	494.98
12	10259	2006-07-18 00:00:00.000	100.80	1614.88	-1514.08
13	10260	2006-07-19 00:00:00.000	1504.65	100.80	1403.85
14	10261	2006-07-19 00:00:00.000	448.00	1504.65	-1056.65
15	10262	2006-07-22 00:00:00.000	584.00	448.00	136.00
16	10263	2006-07-23 00:00:00.000	1873.80	584.00	1289.80
17	10264	2006-07-24 00:00:00.000	605.62	1873.80	-1178.17

LAPTOP-CEEMFUHE (16.0 RTM) | LAPTOP-CEEMFUHE\JAG (S8) | TSQL | 00:00:00

```
SELECT
    orderid,
    orderdate,
    val,
    LAG(val) OVER (ORDER BY orderdate, orderid) AS prevval,
    val - LAG(val) OVER (ORDER BY orderdate, orderid) AS diffprev
FROM
    Sales.OrderValues;
```



5

**[Soal-10]** Buatlah sebuah CTE bernama SalesMonth2007 yang membuat dua kolom yaitu, monthno (jumlah bulan dari kolom orderdate) dan val (agregat dari kolom val)! Kemudian filter hasilnya hanya untuk tahun pesanan 2007 dan dikelompokkan berdasarkan monthno!

Results		Messages
	monthno	val
1	1	61258.08
2	2	38483.64
3	3	38547.23
4	4	53032.95
5	5	53781.30
6	6	36362.82
7	7	51020.86
8	8	47287.68
9	9	55629.27
10	10	66749.23
11	11	43533.80
12	12	71398.44

✓ LAPTOP-CEEMFUHE (16.0 RTM) | LAPTOP-CEEMFUHE\JAG (58) | TSQL

```

WITH SalesMonth2007 AS (
    SELECT
        MONTH(orderdate) AS monthno, SUM(val) AS val
    FROM
        Sales.OrderValues
    WHERE
        YEAR (orderdate) = 2007
    GROUP BY
        MONTH(orderdate)
)
SELECT
    monthno,
    val
FROM SalesMonth2007;
  
```

**[Soal-11]** Tuliskan pernyataan SELECT yang akan mengambil kolom monthno dan val dari CTE dan tambahkan 3 kolom untuk ditampilkan, yaitu :

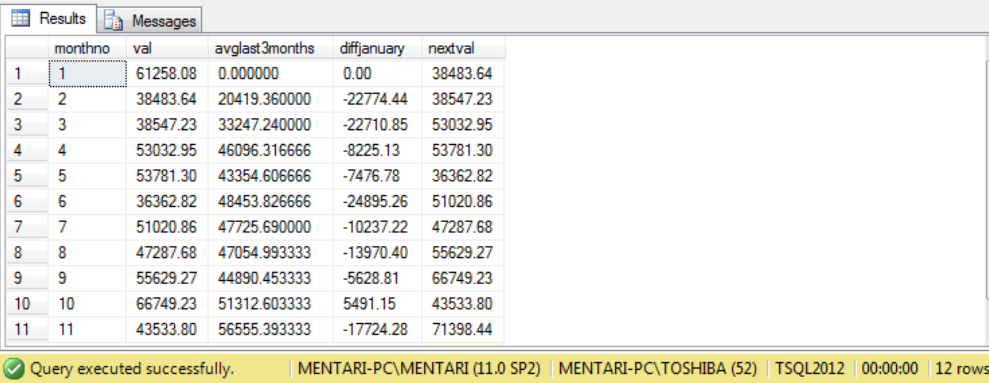
- 1) avglast3months (jumlah penjualan rata-rata tiga bulan terakhir)
- 2) diffjanuary (perbedaan antara val saat ini dengan val pada bulan januari, gunakan fungsi FIRST\_VALUE)
- 3) nextval (nilai dari kolom val pada bulan selanjutnya)

Informasi : Jumlah rata-rata untuk tiga bulan terakhir tidak dihitung dengan benar karena jumlah total 2 bulan pertama dibagi dengan 3.

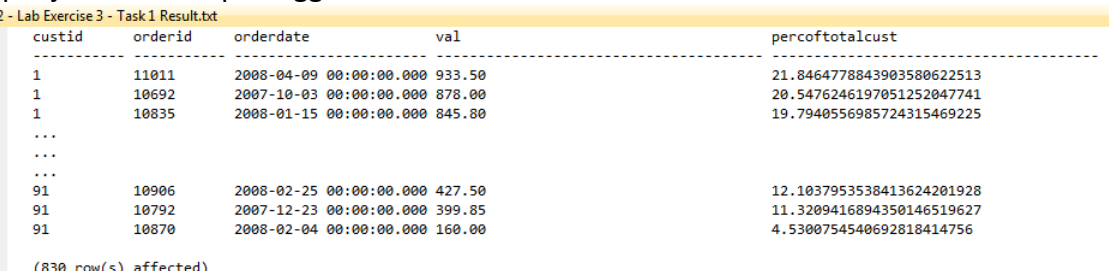
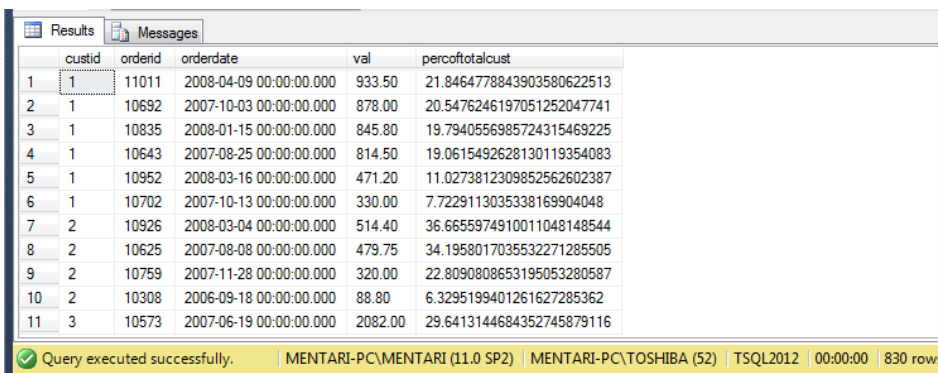
63 - Lab Exercise2 - Task 3 Result.txt				
monthno	val	avglast3months	diffjanuary	nextval
1	61258.08	0.000000	0.00	38483.64
2	38483.64	20419.360000	-22774.44	38547.23
3	38547.23	33247.240000	-22710.85	53032.95
4	53032.95	46096.316666	-8225.13	53781.30
5	53781.30	43354.606666	-7476.78	36362.82
6	36362.82	48453.826666	-24895.26	51020.86
7	51020.86	47725.690000	-10237.22	47287.68
8	47287.68	47054.993333	-13970.40	55629.27
9	55629.27	44890.453333	-5628.81	66749.23
10	66749.23	51312.603333	5491.15	43533.80
11	43533.80	56555.393333	-17724.28	71398.44
12	71398.44	55304.100000	10140.36	NULL

(12 row(s) affected)

6

7	
	<b>Kesimpulan:</b> Setelah menjalankan praktikum bagian ini, mahasiswa bisa menggunakan fungsi OFFSET pada pernyataan T-SQL.

### Praktikum – Bagian 3: Menulis Query Menggunakan Fungsi Agregasi Window

Langkah	Keterangan
1	<p>Skenario :</p> <p>Untuk lebih memahami nilai penjualan kumulatif pelanggan melalui waktu dan untuk menyediakan analisis penjualan dengan analisis selama setahun ini diperlukan pernyataan SELECT yang berbeda menggunakan fungsi agregasi window.</p> <p>Untuk melakukan percobaan pada praktikum bagian 3 ini, pastikan database terhubung dengan “TSQL”.</p>
2	<p><b>[Soal-12]</b> Tuliskan pernyataan SELECT untuk mengambil kolom custid, orderid, orderdate, dan val dari view Sales.OrderValues. Tambahkan kolom bernama percoftotalcust yang berisi persentase nilai masing-masing jumlah pesanan penjualan dibandingkan dengan jumlah penjualan untuk pelanggan tersebut!</p> 
	

**[Soal-13]** Salin pernyataan SELECT sebelumnya dan modifikasi dengan menambahkan kolom yang baru dihitung bernama runval! Kolom ini harus berisi total penjualan yang sedang terjadi untuk setiap pelanggan berdasarkan tanggal pemesanan, menggunakan orderid sebagai tiebreaker.

73 - Lab Exercise 3 - Task 2 Result.txt

custid	orderid	orderdate	val	percoftotalcust	runval
1	10643	2007-08-25 00:00:00.000	814.50	19.0615492628130119354083	814.50
1	10692	2007-10-03 00:00:00.000	878.00	20.5476246197051252047741	1692.50
1	10702	2007-10-13 00:00:00.000	330.00	7.7229113035338169904048	2022.50
...					
...					
91	10906	2008-02-25 00:00:00.000	427.50	12.1037953538413624201928	2254.35
91	10998	2008-04-03 00:00:00.000	686.00	19.4226985093220458953269	2940.35
91	11044	2008-04-23 00:00:00.000	591.60	16.7499539914211696088562	3531.95

(830 row(s) affected)

Results

Messages

	custid	orderid	orderdate	val	percoftotalcust	runval
1	1	10643	2007-08-25 00:00:00.000	814.50	19.0615492628130119354083	814.50
2	1	10692	2007-10-03 00:00:00.000	878.00	20.5476246197051252047741	1692.50
3	1	10702	2007-10-13 00:00:00.000	330.00	7.7229113035338169904048	2022.50
4	1	10835	2008-01-15 00:00:00.000	845.80	19.7940556985724315469225	2868.30
5	1	10952	2008-03-16 00:00:00.000	471.20	11.0273812309852562602387	3339.50
6	1	11011	2008-04-09 00:00:00.000	933.50	21.8464778843903580622513	4273.00
7	2	10308	2006-09-18 00:00:00.000	88.80	6.3295199401261627285362	88.80
8	2	10625	2007-08-08 00:00:00.000	479.75	34.1958017035532271285505	568.55
9	2	10759	2007-11-28 00:00:00.000	320.00	22.8090808653195053280587	888.55
10	2	10926	2008-03-04 00:00:00.000	514.40	36.6655974910011048148544	1402.95
11	3	10365	2006-11-27 00:00:00.000	403.20	5.7403352515240647040566	403.20

Query executed successfully.

MENTARI-PC\MENTARI (11.0 SP2)

MENTARI-PC\TOSHIBA (52)

TSQL2012

00:00:00

830 rows

Query executed successfully.

MENTARI-PC\MENTARI (11.0 SP2) | MENTARI-PC\TOSHIBA (52) | TSQL2012 | 00:00:00 | 830 rows

**[Soal-14]** Salin CTE SalesMonth2007 dalam percobaan 2. Tuliskan pernyataan SELECT untuk mengambil kolom monthno dan val. Tambahkan dua kolom yang dihitung:

- 1) avglast3months. Kolom ini harus berisi jumlah penjualan rata-rata untuk tiga bulan terakhir sebelum bulan saat ini menggunakan fungsi window agregat. Asumsikan bahwa tidak ada *missing months*.
- 2) ytdval Kolom ini harus berisi nilai penjualan kumulatif sampai dengan bulan saat ini.

74 - Lab Exercise 3 - Task 3 Result.txt

monthno	val	avglast3months	ytdval
1	61258.08	61258.080000	61258.08
2	38483.64	49870.860000	99741.72
3	38547.23	46096.316666	138288.95
4	53032.95	47830.475000	191321.90
5	53781.30	45961.280000	245103.20
6	36362.82	45431.075000	281466.02
7	51020.86	48549.482500	332486.88
8	47287.68	47113.165000	379774.56
9	55629.27	47575.157500	435403.83
10	66749.23	55171.760000	502153.06
11	43533.80	53299.995000	545686.86
12	71398.44	59327.685000	617085.30

(12 row(s) affected)

Results		Messages	
monthno	val	avglast3months	ytdval
1	61258.08	61258.080000	61258.08
2	38483.64	49870.860000	99741.72
3	38547.23	46096.316666	138288.95
4	53032.95	47830.475000	191321.90
5	53781.30	45961.280000	245103.20
6	36362.82	45431.075000	281466.02
7	51020.86	48549.482500	332486.88
8	47287.68	47113.165000	379774.56
9	55629.27	47575.157500	435403.83
10	66749.23	55171.760000	502153.06
11	43533.80	53299.995000	545686.86

Query executed successfully. | MENTARI-PC\MENTARI (11.0 SP2) | MENTARI-PC\TOSHIBA (52) | TSQL2012 | 00:00:00 | 12 rows

5

**Kesimpulan:** Setelah melakukan praktikum bagian ini mendapatkan dasar pemahaman bagaimana menggunakan fungsi agregasi window pada pernyataan T-SQL.

--- Selamat Mengerjakan ----