

# **JOBSHEET**

## **PRAKTIKUM BASIS DATA LANJUT**

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



## **PERTEMUAN 6**

**SQL SERVER – View, CTE, Derived Table, TVF**

### **Team Teaching:**

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

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

Annisa Puspa Kirana, S. Kom, M.Kom

Dika Rizky Yunianto, S.Kom, M.Kom

Muhammad Shulhan Khairy, S.Kom, M.Kom

Habibie Ed Dien, S.Kom., M.T.



Jurusan Teknologi Informasi Politeknik Negeri Malang

## **Jobsheet 6: Table Expression**

### **Mata Kuliah Basis Data Lanjut**

**Pengampu:** Tim Ajar Basis Data Lanjut

*September 2024*

---

#### **Topik**

1. Table Expressions

#### **Tujuan**

1. Mahasiswa memahami cara menggunakan VIEWS
2. Mahasiswa memahami cara menggunakan derived table/ tabel turunan
3. Mahasiswa memahami cara menggunakan common table-expression (CTE)
4. Mahasiswa memahami cara menggunakan inline table-valued function (TVF)

#### **Petunjuk Umum**

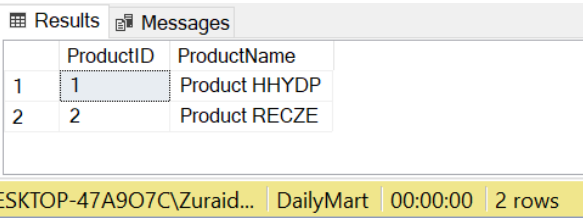
1. Ikuti langkah-langkah pada bagian-bagian praktikum sesuai dengan urutan yang diberikan.
2. Jawablah semua pertanyaan bertanda **[Soal-X]** yang terdapat pada langkah-langkah tertentu di setiap bagian praktikum.
3. Dalam setiap langkah pada praktikum terdapat penjelasan yang akan membantu Anda dalam menjawab pertanyaan-pertanyaan, maka baca dan kerjakanlah semua bagian praktikum dalam jobsheet ini.
4. Tulis jawaban dari soal-soal 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\_Kelas\_03\_NamaLengkapAnda.pdf**
  - 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: View – Membuat View

Langkah	Keterangan
1	<p><b>[Soal-1]</b> Tulislah sebuah query untuk menampilkan kolom ProductID, ProductName, SupplierID, UnitPrice, dan Discontinued dari tabel Productions.Product. Kemudian filter hasilnya agar hanya menampilkan produk yang ada di dalam kategori Beverages saja (categoryid = 1)</p>
2	<p>Eksekusi query pada langkah 2 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p>  <p>4.0 RTM) DESKTOP-47A9O7C\Zuraid... DailyMart 00:00:00 12 rows</p>
3	<p><b>[Soal-2]</b> Modifikasilah kode T-SQL dari langkah 1 di atas dengan menambahkan kode T-SQL berikut (letakkan sebelum query SELECT)</p> <pre>CREATE VIEW Production.BeverageProducts AS</pre>
4	<p>Eksekusilah query pada langkah 4 di atas, sehingga akan dibuat sebuah objek <b>VIEW</b> bernama <b>BeverageProducts</b> dengan skema <b>Production</b></p> <p>Views</p> <ul style="list-style-type: none"> <li>System Views</li> <li>Production.BeverageProducts</li> </ul>

## Praktikum – Bagian 2: View – SELECT dari View

Langkah	Keterangan
1	<p><b>[Soal-3]</b> Buatlah sebuah query untuk menampilkan kolom ProductID dan ProductName dari View <b>Production.BeverageProducts</b>. Filterlah hasilnya agar hanya menampilkan produk dengan SupplierID = 1.</p>

2	<p>Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> 
---	--

### Praktikum – Bagian 3: View - Menambahkan klausa ORDER BY pada VIEW

Langkah	Keterangan
1	<p>Perhatikan script T-SQL berikut:</p> <pre>ALTER VIEW Production.BeverageProducts AS SELECT     ProductID, ProductName, SupplierID, UnitPrice, Discontinued FROM Production.Products WHERE CategoryID = 1 ORDER BY ProductName;</pre>
2	<p><b>[Soal-4]</b> Setelah mengeksekusi T-SQL di atas, apakah yang terjadi? Tuliskan pesan error yang ada dan jelaskan penyebab mengapa error tersebut muncul!</p>
3	<p>Modifikasilah T-SQL pada langkah 1 di atas dengan menambahkan TOP(100) PERCENT sehingga sekarang query menjadi:</p> <pre>ALTER VIEW Production.BeverageProducts AS SELECT TOP 100 PERCENT     ProductID, ProductName, SupplierID, UnitPrice, Discontinued FROM Production.Products WHERE CategoryID = 1 ORDER BY ProductName;</pre>
4	<p>Eksekusi T-SQL pada langkah 3 di atas dan perhatikan bahwa query tersebut telah berhasil mengubah View <b>Production.BeverageProducts</b> meskipun masih terdapat klausa ORDER BY pada query tersebut.</p>
5	<p><b>[Soal-5]</b> Jika sebuah query dijalankan terhadap View <b>Production.BeverageProducts</b> yang telah dimodifikasi, apakah result set akan selaluurut berdasarkan ProductName? Jelaskan!</p>

**Praktikum – Bagian 4: View - Menambahkan kolom ke dalam VIEW**

Langkah	Keterangan
1	<p>Perhatikan statement T-SQL berikut ini yang menambahkan sebuah kolom ke dalam VIEW <b>Production.BeverageProducts</b> yang sudah dibuat di <u>Praktikum - Bagian 1</u> dengan perintah ALTER VIEW</p> <pre>ALTER VIEW Production.BeverageProducts AS SELECT TOP 100 PERCENT     ProductID, ProductName, SupplierID, UnitPrice, Discontinued,     CASE WHEN UnitPrice &gt; 100 THEN N'High' ELSE N'Normal' END FROM Production.Products WHERE CategoryID = 1 ORDER BY ProductName;</pre>
2	<b>[Soal-6]</b> Setelah mengeksekusi T-SQL di atas, apakah yang terjadi? Tuliskan pesan error yang ada dan jelaskan penyebab mengapa error tersebut muncul!
3	<b>[Soal-7]</b> Perbaiki sql script di atas sehingga view dapat diubah dengan tepat.

**Praktikum – Bagian 5: View - Menghapus View**

Langkah	Keterangan
1	<p>Untuk menghapus View <b>Production.BeverageProducts</b>, eksekusi perintah T-SQL berikut:</p> <pre>IF OBJECT_ID(N'Production.BeverageProducts', N'V') IS NOT NULL     DROP VIEW Production.BeverageProducts;</pre>

**Praktikum – Bagian 6: Derived Table**

Langkah	Keterangan
1	<p><b>[Soal-8]</b> Gunakan query SELECT yang ada pada <u>Praktikum Bagian 4 – Langkah 1</u> sebagai Derived Table. Beri nama alias Product.</p> <p>Buatlah query SELECT terhadap derived table tersebut untuk menampilkan kolom ProductID dan ProductName dari produk yang memiliki kategori harga 'High'</p>
2	Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:

Results

Messages

	ProductID	ProductName
1	38	Product QDOMO

DP-47A9O7C\Zuraid... | DailyMart | 00:00:00 | 1 rows

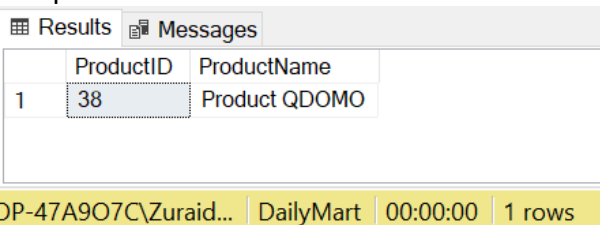
**Praktikum – Bagian 7: Derived Table - Membuat query SELECT untuk mengetahui total dan rata-rata jumlah order (nominal)**

Langkah	Keterangan																																				
1	<p><b>[Soal-9]</b> Tampilkan nilai TotalSalesAmount dan AvgSalesAmount untuk setiap customer.</p> <p>Pertama, buat sebuah derived table yang berisi query JOIN antara tabel <b>Sales.Orders</b> dan <b>Sales.OrderDetails</b>. SELECT kolom CustomerID dan OrderID dari tabel <b>Sales.Orders</b>, serta kolom SalesAmount (Quantity * UnitPrice dari tabel <b>Sales.OrderDetails</b>).</p> <p>Kemudian buat query terhadap derived table tersebut untuk menampilkan TotalSalesAmount dan AvgSalesAmount untuk setiap CustomerID dengan memanfaatkan fungsi agregasi.</p>																																				
2	<p>Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> <div><div>ResultsMessages</div><table><thead><tr><th></th><th>CustomerID</th><th>TotalSalesAmount</th><th>AvgSalesAmount</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>4596.20</td><td>766.0333</td></tr><tr><td>2</td><td>2</td><td>1402.95</td><td>350.7375</td></tr><tr><td>3</td><td>3</td><td>7515.35</td><td>1073.6214</td></tr><tr><td>4</td><td>4</td><td>13806.50</td><td>1062.0384</td></tr><tr><td>5</td><td>5</td><td>26968.15</td><td>1498.2305</td></tr><tr><td>6</td><td>6</td><td>3239.80</td><td>462.8285</td></tr><tr><td>7</td><td>7</td><td>19088.00</td><td>1735.2727</td></tr><tr><td>8</td><td>8</td><td>5297.80</td><td>1765.9333</td></tr></tbody></table><div>cal) (14.0 RTM)   DESKTOP-47A9O7C\Zuraid...   DailyMart   00:00:00   89 rows</div></div>		CustomerID	TotalSalesAmount	AvgSalesAmount	1	1	4596.20	766.0333	2	2	1402.95	350.7375	3	3	7515.35	1073.6214	4	4	13806.50	1062.0384	5	5	26968.15	1498.2305	6	6	3239.80	462.8285	7	7	19088.00	1735.2727	8	8	5297.80	1765.9333
	CustomerID	TotalSalesAmount	AvgSalesAmount																																		
1	1	4596.20	766.0333																																		
2	2	1402.95	350.7375																																		
3	3	7515.35	1073.6214																																		
4	4	13806.50	1062.0384																																		
5	5	26968.15	1498.2305																																		
6	6	3239.80	462.8285																																		
7	7	19088.00	1735.2727																																		
8	8	5297.80	1765.9333																																		

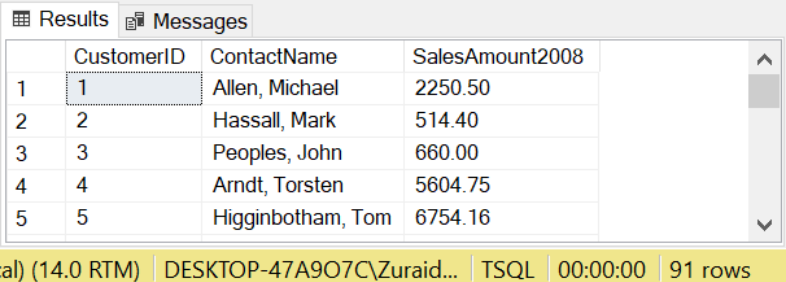
**Praktikum – Bagian 8: Derived Table - Membuat query SELECT untuk mendapatkan persentase kenaikan penjualan**

Langkah	Keterangan																				
1	<p>Buat view Sales.OrderValues dengan mengeksekusi script berikut:</p> <pre>CREATE VIEW Sales.OrderValues AS SELECT O.OrderID, O.CustomerID, O.EmployeeID, O.ShipperID,        O.OrderDate, O.RequiredDate, O.ShippedDate,        SUM(OD.Quantity) AS Quantity,        CAST(SUM(OD.Quantity * OD.unitprice * (1 - OD.discount))             AS NUMERIC(12, 2)) AS SalesAmount FROM Sales.Orders AS O JOIN Sales.OrderDetails AS OD ON O.orderid = OD.orderid GROUP BY O.orderid, O.CustomerID, O.EmployeeID, O.shipperid, O.orderdate, O.requireddate, O.shippeddate; GO</pre>																				
2	<p>Untuk mengecek result set dari view tersebut, eksekusi query berikut:</p> <pre>SELECT * FROM Sales.OrderValues</pre>																				
3	<p>[Soal-10] Tulislah sebuah query yang menampilkan kolom-kolom berikut ini:</p> <ul style="list-style-type: none"><li>- OrderYear: tahun dari tanggal order</li><li>- CurrTotalSales: total penjualan pada tahun tersebut</li><li>- PrevTotalSales: total penjualan pada tahun sebelumnya</li><li>- PercentGrowth: persentase kenaikan penjualan dari tahun yang sedang berjalan dibanding tahun sebelumnya</li></ul>																				
4	<ul style="list-style-type: none"><li>- Manfaatkan View <b>Sales.OrderValues</b>. Jumlah penjualan sudah dikalkulasi dari semua SalesOrderDetail pada kolom SalesAmount</li><li>- Buat derived table (subquery) untuk menghitung total penjualan untuk setiap tahun.</li><li>- Buat derived table kedua dengan query yang sama dengan derived table pertama.</li><li>- Lakukan JOIN untuk menggabungkan kedua derived table tersebut dengan kondisi OrderYear pada derived table pertama matching dengan (OrderYear + 1) pada derived table kedua.</li></ul>																				
5	<p>Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> <div><div>Results Messages</div><table><tr><th></th><th>OrderYear</th><th>CurrTotalSales</th><th>PrevTotalSales</th><th>PercentGrowth</th></tr><tr><td>1</td><td>2006</td><td>208083.99</td><td>NULL</td><td>NULL</td></tr><tr><td>2</td><td>2007</td><td>617085.30</td><td>208083.99</td><td>196.555800</td></tr><tr><td>3</td><td>2008</td><td>440623.93</td><td>617085.30</td><td>-28.595900</td></tr></table><div>(SQL Server 2008) (14.0 RTM)   DESKTOP-47A9O7C\Zuraid...   TSQL   00:00:00   3 rows</div></div>		OrderYear	CurrTotalSales	PrevTotalSales	PercentGrowth	1	2006	208083.99	NULL	NULL	2	2007	617085.30	208083.99	196.555800	3	2008	440623.93	617085.30	-28.595900
	OrderYear	CurrTotalSales	PrevTotalSales	PercentGrowth																	
1	2006	208083.99	NULL	NULL																	
2	2007	617085.30	208083.99	196.555800																	
3	2008	440623.93	617085.30	-28.595900																	

**Praktikum – Bagian 9: CTE - Membuat query menggunakan CTE**

Langkah	Keterangan
1	[Soal-11] Buatlah query untuk soal pada <u>Praktikum - Bagian 6</u> , akan tetapi dengan menggunakan Common Table Expressions (CTE). Beri nama alias query CTE tersebut sebagai <b>BeverageProducts</b> .
2	<p>Eksekusi query pada langkah 1 dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> 

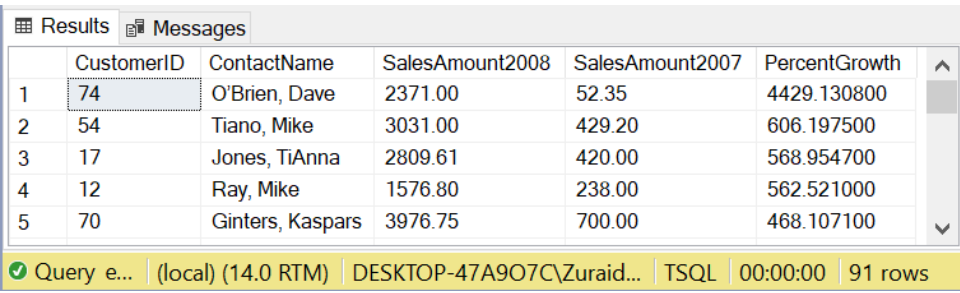
**Praktikum – Bagian 10: CTE**

Langkah	Keterangan
1	[Soal-12] Buatlah query untuk menampilkan sales amount untuk setiap customer pada tahun 2008. Tampilkan kolom CustomerID, ContactName, dan SalesAmount2008. <i>Gunakan Common Table Expression dan manfaatkan View <b>Sales.OrderValues</b>.</i>
2	<p>Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> 

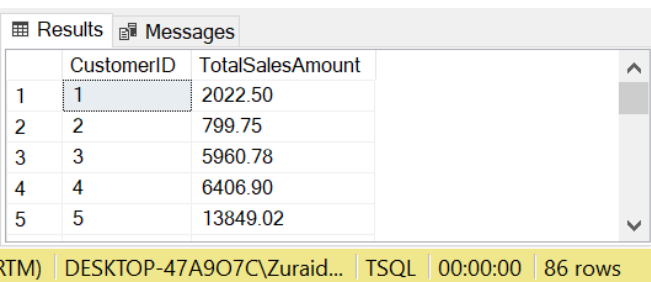
**Praktikum – Bagian 11: CTE - Membuat query SELECT untuk membandingkan jumlah total penjualan untuk setiap customer dengan tahun sebelumnya**

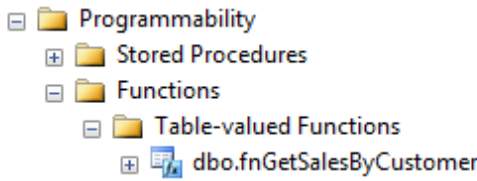
Langkah	Keterangan
1	<p>[Soal-13] Buatlah query untuk menampilkan data pembelian customer. Tampilkan kolom CustomerID, ContactName, dan 3 kolom berikut:</p> <ul style="list-style-type: none"> <li>- SalesAmount2008: total penjualan tahun 2008</li> <li>- SalesAmount2007: total penjualan tahun 2007</li> <li>- PercentGrowth: persentase kenaikan penjualan antara tahun 2007 dan 2008</li> </ul>



	<p>Apabila PercentGrowth menghasilkan NULL, tampilkan sebagai 0.</p> <p>Anda bisa gunakan CTE dari <u>Praktikum Bagian 10</u> dan buatlah satu CTE lagi untuk tahun 2007. Kemudian, lakukan operasi JOIN antara kedua CTE tersebut dengan tabel <b>Sales.Customers</b>. Urutkan hasilnya berdasarkan kolom PercentGrowth.</p>
2	<p>Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> 

### Praktikum – Bagian 12: Membuat Inline Table-Valued Function

Langkah	Keterangan
1	<p><b>[Soal-14]</b> Buat query terhadap View Sales.OrderValues untuk menampilkan total sales untuk masing-masing customer pada tahun 2017.</p>
2	<p>Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> 
3	<p><b>[Soal-15]</b> Buatlah sebuah inline TVF/ Table-Valued Function dengan menambahkan baris berikut ini dan diletakkan sebelum query SELECT pada Langkah 1 di atas</p> <pre>CREATE FUNCTION dbo.fnGetSalesByCustomer (@orderYear AS INT) RETURNS TABLE AS RETURN</pre>
4	<p><b>[Soal-16]</b> Modifikasilah query tersebut dengan mengganti nilai konstanta <b>2007</b> pada klausa WHERE, dengan parameter <b>@orderYear</b>.</p>

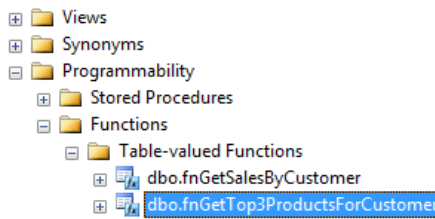
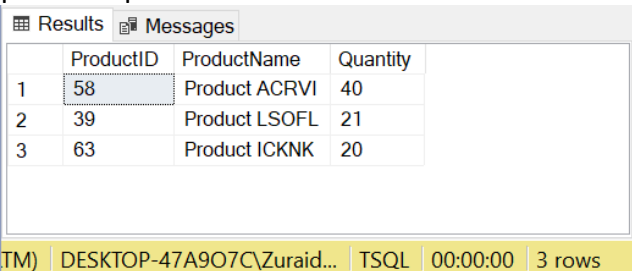
5	<p>Jalankan script pada langkah 4 di atas sehingga akan terbentuk sebuah inline TVF bernama <b>dbo.fnGetSalesByCustomer</b></p> 
---	---

**Praktikum – Bagian 12: Menggunakan Inline Table-Valued Function**

Langkah	Keterangan																				
1	[Soal-17] Buatlah sebuah query SELECT yang memanggil inline TVF <b>dbo.fnGetSalesByCustomer</b> dengan nilai <b>2007</b> sebagai argumennya																				
2	Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini: <div><div>ResultsMessages</div><table><thead><tr><th></th><th>CustomerID</th><th>TotalSalesAmount</th><th></th></tr></thead><tbody><tr><td>1</td><td>1</td><td>2022.50</td><td rowspan="5">^    v</td></tr><tr><td>2</td><td>2</td><td>799.75</td></tr><tr><td>3</td><td>3</td><td>5960.78</td></tr><tr><td>4</td><td>4</td><td>6406.90</td></tr><tr><td>5</td><td>5</td><td>13849.02</td></tr></tbody></table><div>(TM) DESKTOP-47A9O7C\Zuraid... TSQL 00:00:00 86 rows</div></div>		CustomerID	TotalSalesAmount		1	1	2022.50	^    v	2	2	799.75	3	3	5960.78	4	4	6406.90	5	5	13849.02
	CustomerID	TotalSalesAmount																			
1	1	2022.50	^    v																		
2	2	799.75																			
3	3	5960.78																			
4	4	6406.90																			
5	5	13849.02																			

**Praktikum – Bagian 13: Inline ITF - Membuat query SELECT untuk mendapatkan 3 produk terlaris untuk customer tertentu**

Langkah	Keterangan																
1	[Soal-18] Buatlah query untuk menampilkan 3 produk yang paling banyak dibeli oleh customer dengan ID = 1.																
2	<p>Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> <div><div>ResultsMessages</div><table><thead><tr><th></th><th>ProductID</th><th>ProductName</th><th>Quantity</th></tr></thead><tbody><tr><td>1</td><td>58</td><td>Product ACRVI</td><td>40</td></tr><tr><td>2</td><td>39</td><td>Product LSOFL</td><td>21</td></tr><tr><td>3</td><td>63</td><td>Product ICKNK</td><td>20</td></tr></tbody></table><div>TM) DESKTOP-47A9O7C\Zuraid... TSQL 00:00:00 3 rows</div></div>		ProductID	ProductName	Quantity	1	58	Product ACRVI	40	2	39	Product LSOFL	21	3	63	Product ICKNK	20
	ProductID	ProductName	Quantity														
1	58	Product ACRVI	40														
2	39	Product LSOFL	21														
3	63	Product ICKNK	20														

3	<p><b>[Soal-19]</b> Dengan menggunakan query pada langkah 1 di atas, buatlah sebuah inline TVF dengan menambahkan beberapa baris berikut sebelum query SELECT</p> <pre>CREATE FUNCTION dbo.fnGetTop3ProductsForCustomer (@customerID AS INT) RETURNS TABLE AS RETURN</pre> <p>Modifikasilah query di atas dengan mengganti nilai konstanta <b>1</b> pada klausa WHERE dengan parameter <b>@customerID</b>.</p>
4	<p>Jalankan skrip tersebut sehingga akan terbentuk sebuah inline TVF bernama <b>dbo.fnGetTop3ProductsForCustomer</b> yang memiliki parameter customer ID.</p> 
5	<p><b>[Soal-20]</b> Lakukan ujicoba dengan membuat sebuah query SELECT terhadap inline TVF tersebut dan gunakan nilai 1 sebagai argumen untuk parameter customerID.</p>
6	<p>Eksekusi query pada langkah 1 di atas dan bandingkan dengan hasil yang ditunjukkan pada tampilan berikut ini:</p> 

#### Praktikum – Bagian 14: Inline TVF - Menghapus inline Table-valued function

Langkah	Keterangan
1	<p>Hapuslah inline TVF yang sudah dibuat dengan menjalankan script berikut:</p> <pre>IF OBJECT_ID('dbo.fnGetSalesByCustomer') IS NOT NULL DROP FUNCTION dbo.fnGetSalesByCustomer; IF OBJECT_ID('dbo.fnGetTop3ProductsForCustomer') IS NOT NULL DROP FUNCTION dbo.fnGetTop3ProductsForCustomer;</pre>

--- Selamat Mengerjakan ---