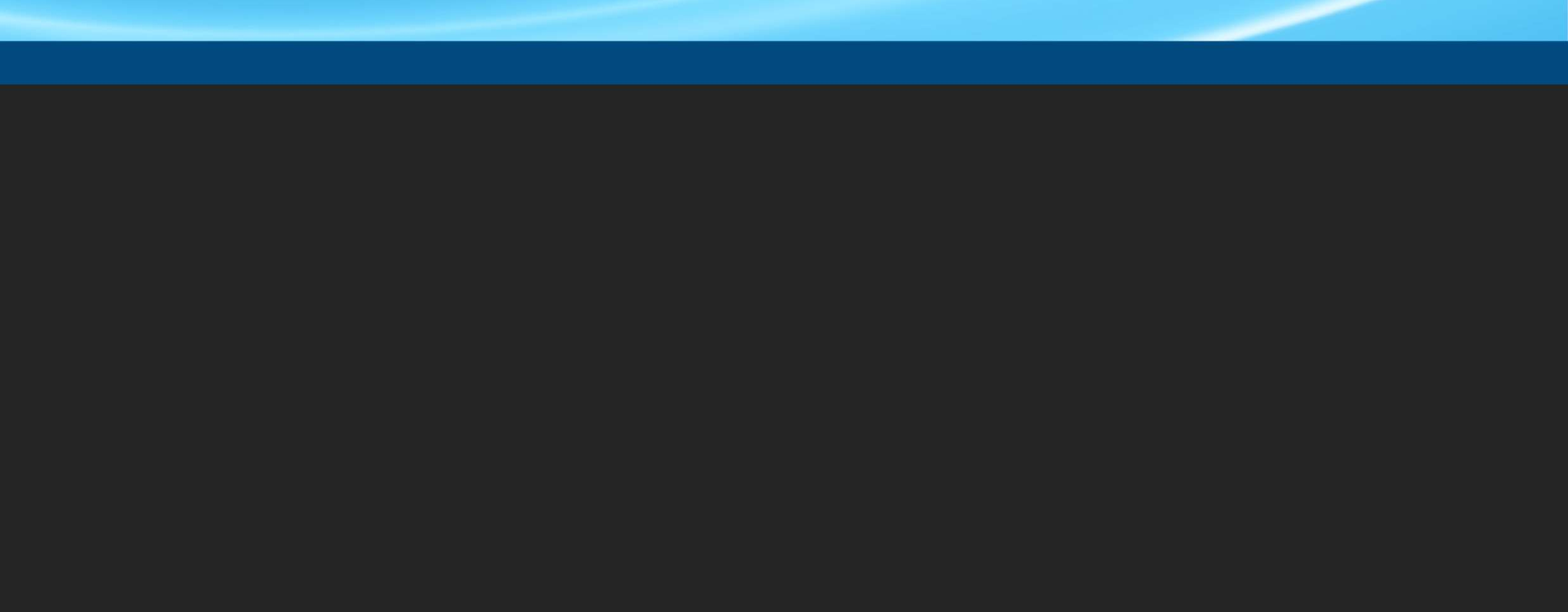
***JTI POLINEMA***



JOBSHEET

PRAKTIKUM BASIS DATA LANJUT

**Jurusan Teknologi Informasi POLITEKNIK NEGERI MALANG**

**PERTEMUAN 11**

# SQL SERVER – STORE PROCEDURE

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

***Team Teaching Basis Data Lanjut 1***

Jurusan Teknologi Informasi Politeknik Negeri Malang

**Jobsheet Minggu-14: Stored Procedure Mata Kuliah Basis Data Lanjut (BDL)** Pengampu: Tim Ajar Basis Data Lanjut *November 2023*

# Topik

Membuat Stored Procedure

# Tujuan

1. Mengembalikan hasil dengan Stored Procedure.
2. Melewatkan parameter ke Procedure.
3. Membuat stored procedure sederhana, yang menyertakan pernyataan SELECT.
4. Membangun dan menjalankan SQL dinamis dengan EXEC dan sp\_executesql.

# 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\_Tugas14\_Kelas\_2DigitNomorAbsen\_NamaLengkapAnda.pdf

* + Contoh:

# o BDL\_Tugas14\_TI2Q\_99\_JoeBiden.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: Menggunakan pernyataan EXECUTE Untuk Memanggil Stored Procedure

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| **1** | Skenario :  Departemen TI telah menyediakan kode T-SQL untuk membuat stored procedure untuk mengambil 10 pelanggan teratas berdasarkan jumlah penjualan. Percobaan bagian ini berisi tentang latihan stored procedure.  Untuk melakukan percobaan pada praktikum bagian 1 ini terlebih dahulu lakukan login pada SQL Server Management Studio (SSMS). Pastikan database terhubung dengan “TSQL2012”. |
| **2** | Jalankan kode T-SQL untuk membuat stored procedur Sales.GetTopCustomers berikut:  CREATE PROCEDURE Sales.GetTopCustomers AS  SELECT TOP(10) c.custid, c.contactname, SUM(o.val) AS salesvalue  FROM Sales.OrderValues AS o INNER JOIN Sales.Customers AS c ON c.custid = o.custid  GROUP BY c.custid, c.contactname ORDER BY salesvalue DESC; |
| **3** | [Soal-1] Tuliskan pernyataan T-SQL untuk menjalankan prosedur yang telah dibuat sebelumnya! |
| **4** | [Soal-2] Letakkan hasil eksekusi soal no. 1 pada bagian ini dan pastikan hasilnya sama dengan gambar di bawah ini. |
| **5** | Departemen TI telah merubah stored procedure dari percobaan 1 langkah 2 dengan kode T-SQL di bawah ini :  ALTER PROCEDURE Sales.GetTopCustomers AS  SELECT c.custid, c.contactname, SUM(o.val) AS salesvalue  FROM Sales.OrderValues AS o INNER JOIN Sales.Customers AS c ON c.custid = o.custid  GROUP BY c.custid, c.contactname ORDER BY salesvalue DESC  OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY; |
| **6** | [Soal-3] Tuliskan pernyataan T-SQL yang akan menjalankan prosedur pada percobaan 1 langkah 5 di atas! |
| **7** | [Soal-4] Letakkan hasil eksekusi soal no. 3 pada bagian ini dan pastikan hasilnya sama dengan gambar di bawah ini. |
| **8** | [Soal-5] Apa perbedaan kode T-SQL untuk prosedur pada percobaan 1 langkah 2 dan langkah 5?  **Jawaban: perbedaan terletak pada penggunaan klausa, pada langkah 2 menggunakan klausa top yg mana digunakan untuk mengembalikan nilai 10 rows paling atas, sedangkan pada langkah 5 menggunakan klausa offset dan fetch, yg mana offset digunakan untuk menentukan jumlah baris yg akan dilewati sebelum mengembalikan data sedangkan fetch digunakan untuk menentukan jumlah baris yang akan ditampilkan** |
| **9** | **Kesimpulan :** Setelah menjalankan praktikum bagian ini, mahasiswa mengetahui bagaimana memanggil stored procedure menggunakan pernyataan EXECUTE. |

**Praktikum – Bagian 2: Melewatkan Parameter ke Stored Procedure**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| **1** | Skenario :  Departemen TI memberikan tambahan modifikasi dari stored procedure di praktikum bagian 1. Stored procedure yang dimodifikasi akan melewatkan parameter yang menentukan tahun pesanan dan jumlah pelanggan yang akan diambil.  Untuk melakukan percobaan pada praktikum bagian 2 ini, pastikan database terhubung dengan “TSQL”. |
| **2** | Jalankan kode T-SQL berikut, untuk memodifikasi stored procedure Sales.GetTopCustomers untuk memasukkan parameter tahun pesanan (@orderyear) :  ALTER PROCEDURE Sales.GetTopCustomers @orderyear int  AS  SELECT c.custid, c.contactname, SUM(o.val) AS salesvalue  FROM Sales.OrderValues AS o INNER JOIN Sales.Customers AS c ON c.custid = o.custid  WHERE YEAR(o.orderdate) = @orderyear GROUP BY c.custid, c.contactname ORDER BY salesvalue DESC  OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY; |
| **3** | [Soal-6] Tuliskan pernyataan EXECUTE untuk memanggil stored procedure Sales.GetTopCustomers untuk tahun 2007! |
| **4** | [Soal-7] Letakkan hasil eksekusi soal no. 6 pada bagian ini dan pastikan hasilnya sama dengan gambar di bawah ini. |
| **5** | [Soal-8] Tuliskan pernyataan EXECUTE untuk memanggil stored procedure Sales.GetTopCustomers untuk tahun 2008! Pastikan hasilnya sesuai dengan gambar berikut : |
| **6** | [Soal-9] Tuliskan pernyataan EXECUTE untuk memanggil stored procedure Sales.GetTopCustomers tanpa parameter! |
| **7** | [Soal-10] Apakah hasil yang didaat dari soal no.9? Jika terdapat error, tuliskan pesan errornya!  **Jawaban: Stored procedure tersebut membutuhkan parameter bernama '@orderyear' yang wajib diisi, tetapi parameter tersebut tidak disertakan saat procedure dipanggil.** |
| **8** | Jalankan kode T-SQL berikut untuk memodifikasi stored procedure Sales.GetTopCustomers:  ALTER PROCEDURE Sales.GetTopCustomers @orderyear int = NULL  AS  SELECT c.custid, c.contactname, SUM(o.val) AS salesvalue  FROM Sales.OrderValues AS o INNER JOIN Sales.Customers AS c ON c.custid = o.custid  WHERE YEAR(o.orderdate) = @orderyear OR @orderyear IS NULL  GROUP BY c.custid, c.contactname ORDER BY salesvalue DESC  OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY; |
| **9** | [Soal-11] Tulis pernyataan EXECUTE untuk memanggil stored procedure Sales.GetTopCustomers tanpa parameter! Pastikan hasilnya sesuai dengan gambar berikut : |
| **10** | [Soal-12] Jika terdapat modifikasi pada stored procedure, apakah mempengaruhi aplikasi atau memerlukan perubahan pada desain aplikasi?  **Jawaban: Modifikasi stored procedure dapat mempengaruhi aplikasi terutama jika ada perubahan pada parameter wajib, nama parameter, tipe data, atau struktur output. Untuk perubahan internal seperti optimasi query atau penambahan parameter opsional dengan nilai default biasanya tidak memerlukan modifikasi aplikasi.** |
| **11** | Jalankan kode T-SQL yang disediakan untuk menambahkan parameter @n ke stored procedure Sales.GetTopCustomers. Parameter tersebut digunakan untuk menentukan berapa banyak pelanggan yang akan diambil. Nilai defaultnya adalah 10.  ALTER PROCEDURE Sales.GetTopCustomers @orderyear int = NULL,  @n int = 10 AS  SELECT c.custid, c.contactname, SUM(o.val) AS salesvalue  FROM Sales.OrderValues AS o INNER JOIN Sales.Customers AS c ON c.custid = o.custid  WHERE YEAR(o.orderdate) = @orderyear OR  @orderyear IS NULL  GROUP BY c.custid, c.contactname ORDER BY salesvalue DESC  OFFSET 0 ROWS FETCH NEXT @n ROWS ONLY; |
| **12** | [Soal-13] Tulis pernyataan EXECUTE untuk memanggil stored procedure Sales.GetTopCustomers tanpa parameter! Pastikan hasilnya sesuai dengan gambar berikut : |
| **13** | [Soal-14]. Tuliskan pernyataan EXECUTE untuk memanggil stored procedure Sales.GetTopCustomers untuk mengambil 20 orang pelanggan! Pastikan hasilnya sesuai dengan gambar berikut : |
| **14** | [Soal-15] Apakah aplikasi yang menggunakan stored procedure perlu diubah jika parameter lain ditambahkan?  **Jawaban: Jika parameter baru yang ditambahkan bersifat opsional (memiliki nilai default), maka aplikasi tidak perlu diubah. Namun jika parameter baru tersebut bersifat wajib (required), maka aplikasi perlu dimodifikasi untuk menyertakan nilai parameter tersebut saat memanggil stored procedure.** |
| **15** | Jalankan kode T-SQL berikut yang memodifikasi stored procedure Sales.GetTopCustomers untuk mengembalikan nama pelanggan/konsumen berdasarkan posisi tertentu dalam peringkat total penjualan yang disediakan oleh parameter @customerpos. Stored procedure ini juga menyediakan parameter baru bernama @customername yang memiliki opsi OUTPUT.  ALTER PROCEDURE Sales.GetTopCustomers @customerpos int = 1,  @customername nvarchar(30) OUTPUT AS  SET @customername = ( SELECT c.contactname FROM Sales.OrderValues AS o  INNER JOIN Sales.Customers AS c ON c.custid = o.custid  GROUP BY c.custid, c.contactname ORDER BY SUM(o.val) DESC  OFFSET @customerpos - 1 ROWS FETCH NEXT 1 ROW ONLY  ); |
| **16** | Departemen TI juga memberikan kode T-SQL untuk mendeklarasikan variabel baru @outcustomername. Variabel ini digunakan sebagai output parameter stored procedure yang dibuat.  DECLARE @outcustomername nvarchar(30); |
| **17** | **Kesimpulan**: Setelah menjalankan praktikum bagian ini, mahasiswa mengetahui bagaimana memanggil stored procedure yang memiliki parameter. |

# Praktikum – Bagian 3: Executing System Stored Procedures

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| **1** | Skenario :  Praktikum pada bagian ini akan mempelajari bagaimana menjalankan beberapa sistem yang paling sering digunakan untuk menyimpan informasi tentang tabel dan kolom.  Untuk melakukan percobaan pada praktikum bagian 3 ini, pastikan database terhubung dengan “TSQL”. |
| **2** | [Soal-16] Tuliskan pernyataan EXECUTE untuk memanggil stored procedure tersimpan sys.sp\_help tanpa parameter, dengan hasil yang sesuai tampilan di bawah ini! |
| **3** | [Soal-17] Tuliskan pernyataan EXECUTE untuk memanggil store prosedur sys.sp\_help untuk tabel tertentu dengan melewatkan parameter Sales.Customer, dengan hasil yang sesuai tampilan di bawah ini! |
|  |  |
| **4** | [Soal-18] Tuliskan pernyataan EXECUTE untuk memanggil stored prosedur sys.sp\_helptext, kemudian lewatkan stored prosedur Sales.GetTopCustomers yang disimpan sebagai parameter, dengan hasil yang sesuai tampilan di bawah ini! |
| **5** | [Soal-19] Tulislah pernyataan EXECUTE untuk memanggil stored procedure sys.sp\_columns untuk tabel Sales.Customers. Terdapat dua parameter yang harus dilewati, yaitu: @table\_name dan @table\_owner, dengan hasil yang sesuai tampilan di bawah ini! |
|  |  |
| **6** | Eksekusi pernyataan T-SQL berikut untuk menghapus stored procedure Sales.GetTopCustomers:  DROP PROCEDURE Sales.GetTopCustomers; |
| **7** | **Kesimpulan**: Setelah melakukan praktikum bagian ini, mahasiswa memiliki pengetahuan dasar cara memanggil berbagai sistem stored procedure yang berbeda-beda. |

## --- Selamat Mengerjakan ----