

JOBSHEET 10

AUTOMATIC SQL SERVER MANAGEMENT

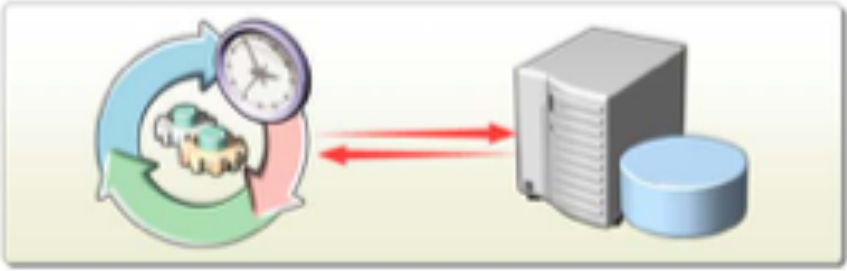
1.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Melakukan otomasi SQL Server Management.
2. Menggunakan SQL Server Agent.
3. Mengatur pekerjaan SQL Server Agent.

Lakukan praktikum di bawah dan buat laporan proses dan hasil berupa screenshot disertai penjelasannya. Kumpulkan dengan format file: **SMBD_KELAS_10_NAMA.pdf**

1.2 Otomasi SQL Server Management

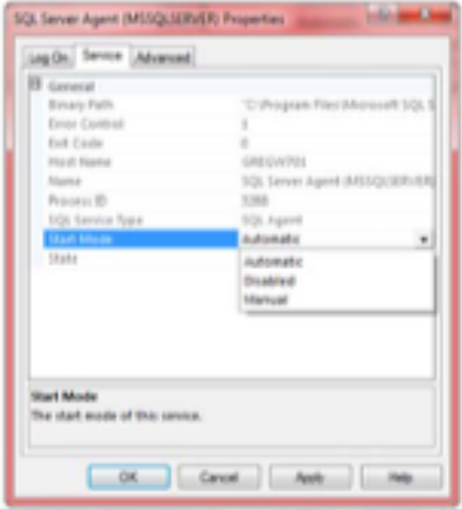


- Reduced administrative workload
 - Automate and Schedule regular tasks
- Proactive management
 - Monitor Performance
 - Recognize and respond to potential problems

Otomatisasi SQL Server Management memiliki keuntungan dalam mengurangi beban kerja administratif serta kemudahan dalam mengatur secara manajerial, salah satunya dapat dengan mudah untuk mengidentifikasi jika ada kesalahan dalam eksekusi perintah perintah pada SQL Server.

SQL Server Agent is the component of SQL Server that is responsible for automation

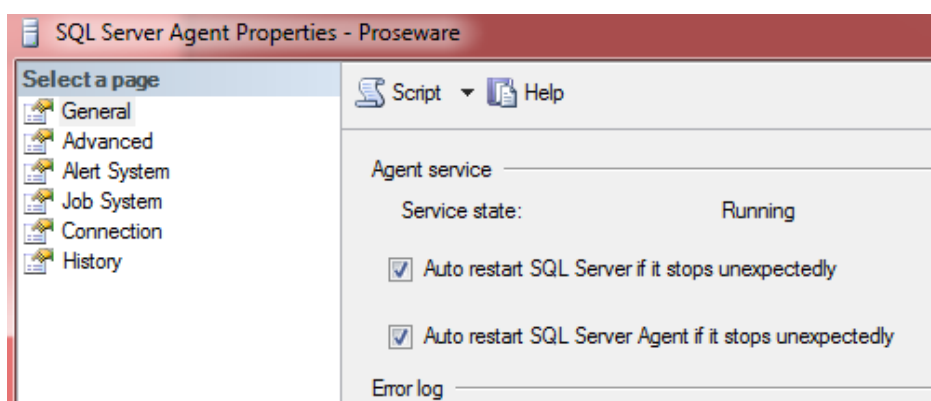
- Runs as a Windows service
- Must be running to
 - Execute jobs
 - Fire alerts
 - Contact operators
- Start Mode should be set to Automatic



SQL Server Agent adalah sebuah modul atau komponen yang terdapat pada SQL Server yang bertugas untuk melakukan otomasi. SQL Server Agent harus dijalankan secara konstan agar dapat melakukan tugas utamanya dalam mengeksekusi pekerjaan dan memberikan peringatan. Konfigurasi SQL Server Agent ada pada menu 'properties' pada servis SQL Server Agent yang dapat ditemukan pada SQL Server Configuration Manager. Terdapat 3 macam mode start SQL Server Agent, yaitu:

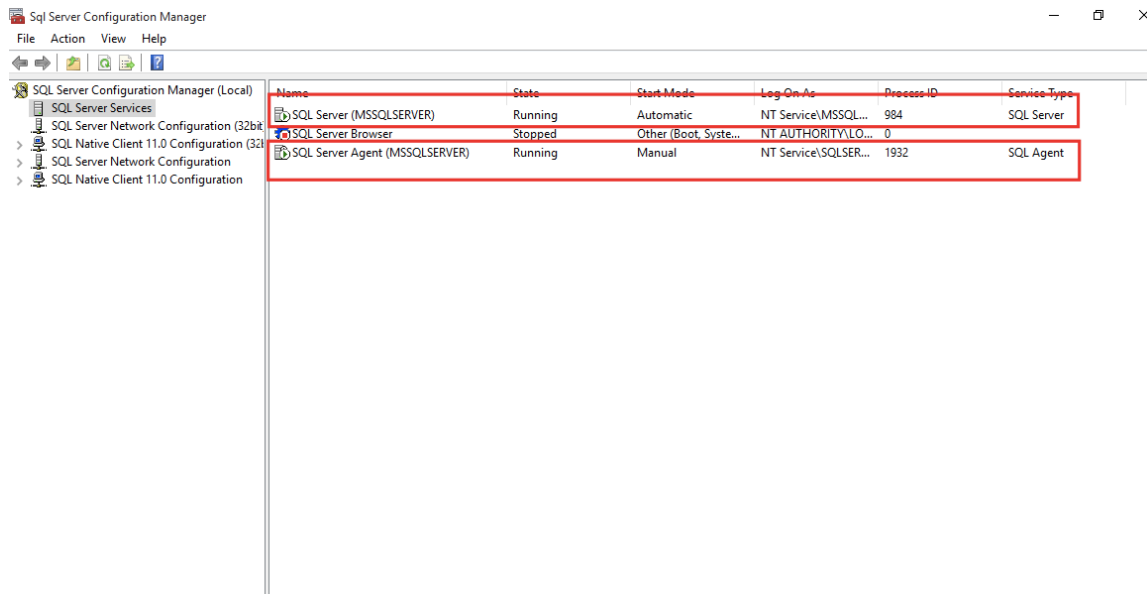
- Automatic : Servis otomatis berjalan ketika sistem operasi menyala.
- Disabled : Servis tidak berjalan walaupun dilakukan start secara manual.
- Manual : Servis dimulai secara manual oleh user

Kita dapat melakukan konfigurasi servis SQL Server Agent untuk melakukan restart secara otomatis ketika servis berhenti tiba-tiba. Pilihan auto restart dapat ditemukan pada menu SQL Server Agent Properties.

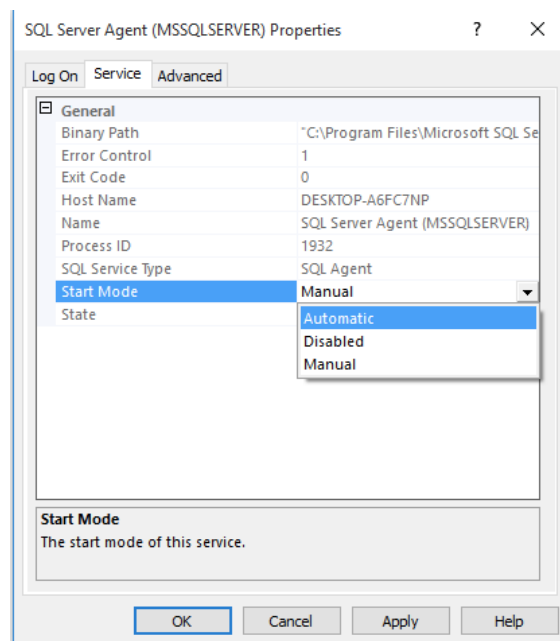


1.2.1 Langkah-langkah Percobaan

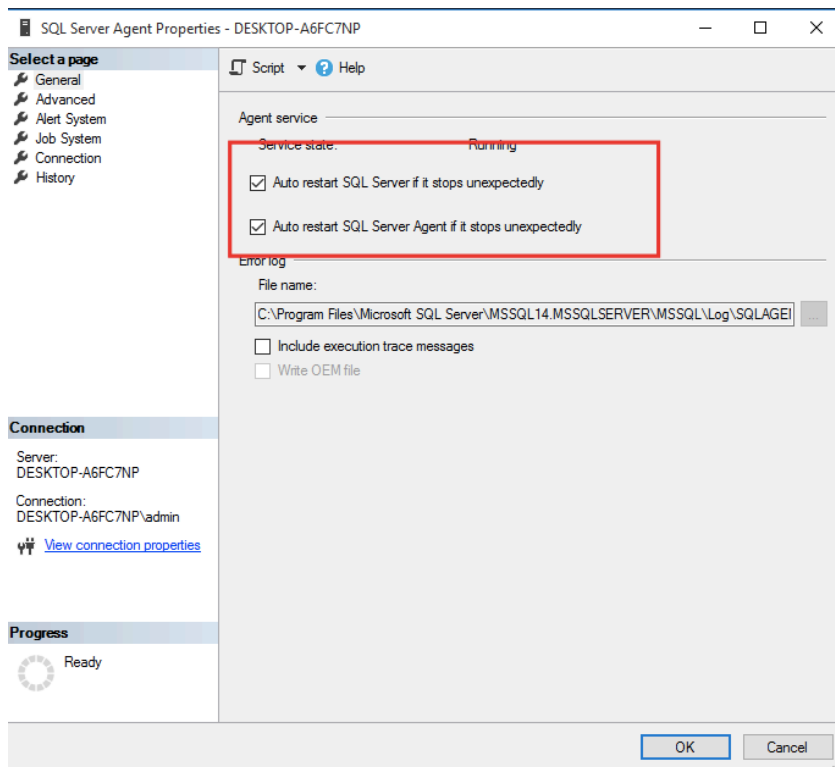
1. Buka SQL Server Management Studio dan melakukan Connect to Server sesuai Authentication masing-masing. Kemudian buka dan eksekusi file **00-Setup.sql**.
2. Buka SQL Server Configuration Manager. Pada panel sebelah kiri, klik SQL Server Services. Pastikan servis SQL Server Agent sudah menyala dan secara otomatis start.



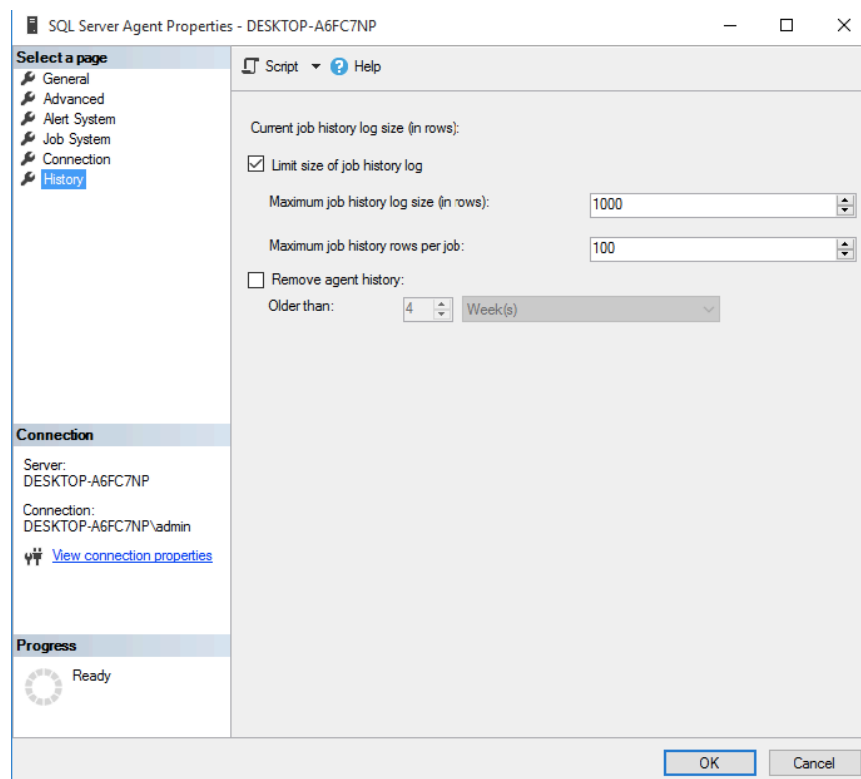
3. Klik kanan pada SQL Server Agent dan klik properties. Pilih tab Service, klik 'automatic in start mode', pilih automatic pada start mode, dan klik ok. Kemudian tutup SQL Server Configuration Manager.



4. Pada SSMS, klik kanan pada SQL Server Agent yang tertera pada object explorer, kemudian pastikan opsi auto restart dipilih semua.

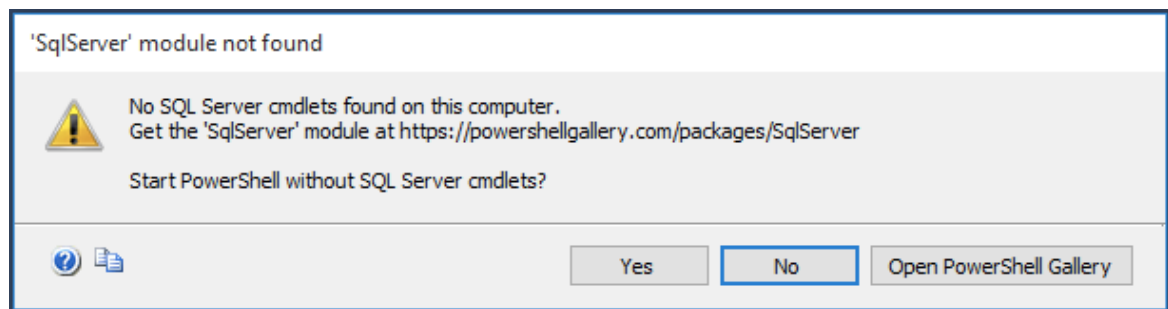


5. Buka pada menu history, perhatikan dan catat setting log size, kemudian tutup window tersebut.



6. Buka Powershell dari SQL Server Agent. Klik kanan pada SQL Server Agent dan pilih start powershell.

NB: Jika terdapat pesan seperti dibawah ini, klik no, kemudian buka windows powershell dengan run as administrator.



Kemudian masukkan perintah berikut dan eksekusi. Ketika selesai, tutup window powershell dan buka kembali powershell dari SQL Server Agent.

- Gunakan perintah 'dir' untuk menampilkan semua objek.

8. Eksekusi perintah 'cd jobs', kemudian eksekusi 'dir', maka list dari jobs akan tampil.

```
PS SQLSERVER:\SQL\DESKTOP-A6FC7NP\DEFAULT\JobServer> cd Jobs
PS SQLSERVER:\SQL\DESKTOP-A6FC7NP\DEFAULT\JobServer\Jobs> dir
```

Name	Owner	Category	Enabled	CurrentRunStatus	DateCreated
-----	-----	-----	-----	-----	-----
syspolicy_purge_history	sa	[Uncategorized (Local)]	True	Idle	4/5/2019 4:...

```
PS SQLSERVER:\SQL\DESKTOP-A6FC7NP\DEFAULT\JobServer\Jobs>
```

9. Eksekusi perintah 'dir | Select Name, NextRunDate, LastRunDate'.

```
PS SQLSERVER:\SQL\DESKTOP-A6FC7NP\DEFAULT\JobServer\Jobs> dir | Select Name, NextRunDate, LastRunDate
```

Name	NextRunDate	LastRunDate
-----	-----	-----
syspolicy_purge_history	4/8/2019 2:00:00 AM	1/1/0001 12:00:00 AM

```
PS SQLSERVER:\SQL\DESKTOP-A6FC7NP\DEFAULT\JobServer\Jobs> _
```

10. Tutup powershell dengan mengeksekusi perintah 'exit'

1.2.2 Soal

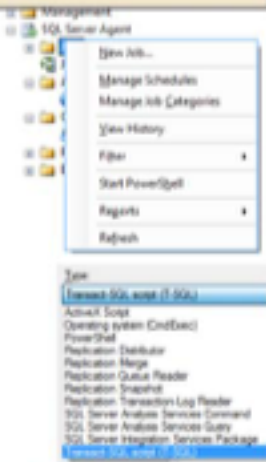
1. Apa fungsi auto restart SQL Server Agent?
2. Data apa yang ditampilkan pada langkah praktikum nomor 9?

1.3 Menggunakan SQL Server Agent

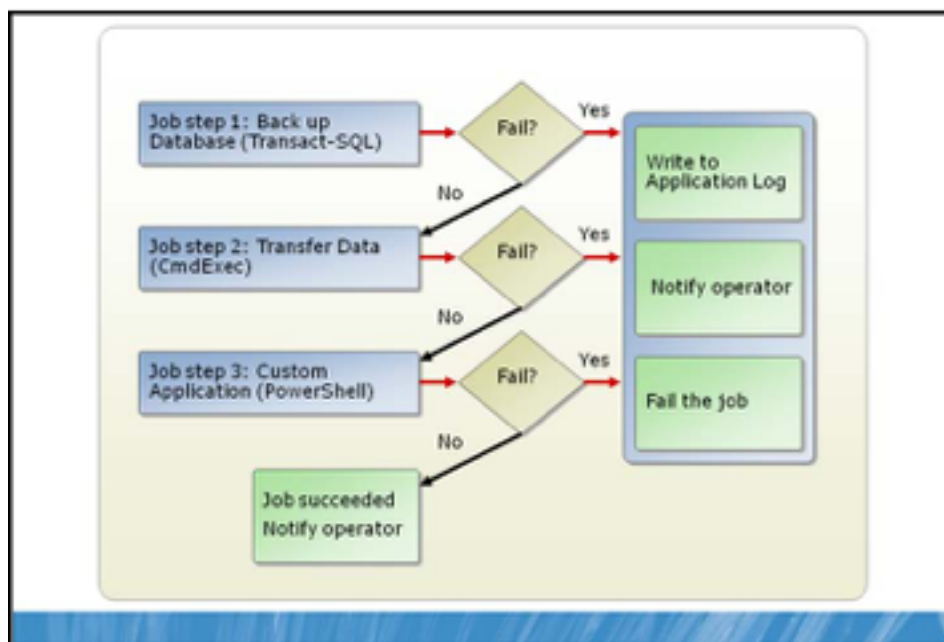
Pekerjaan/job pada SQL Server Agent adalah operasi yang dijalankan SQL Server Agent secara berurutan. Contoh dari pekerjaan tersebut adalah eksekusi perintah T-SQL, eksekusi script pada powershell, eksekusi SQL Server Integration Services Package, dan eksekusi perintah analisis servis.

SQL Server Agent Jobs are a specified series of operations performed sequentially by SQL Server Agent

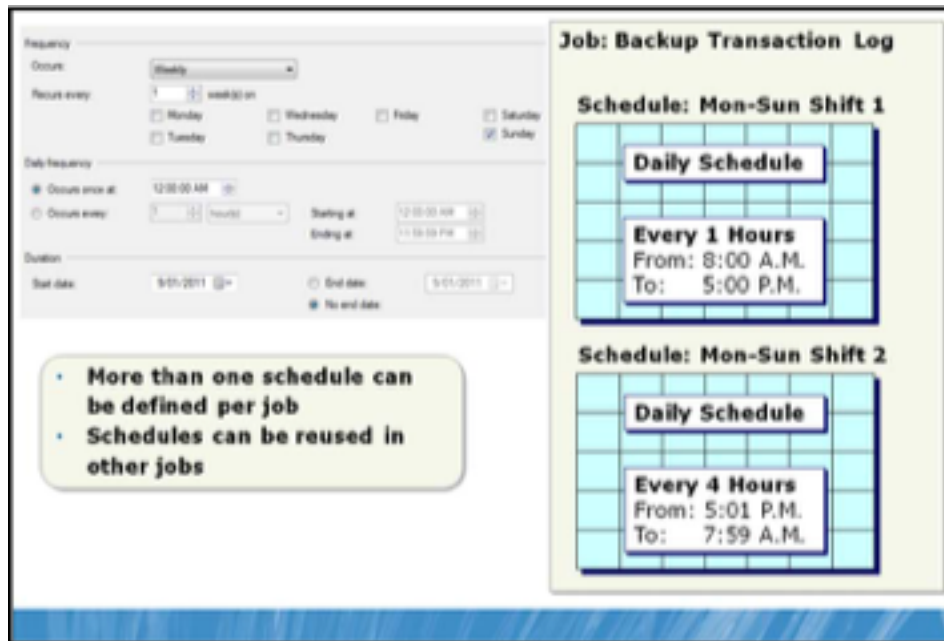
- Jobs support many types including:
 - Transact-SQL
 - Command line script or application execution
 - PowerShell script execution
- Jobs can be configured to:
 - Run once or repeatedly
 - Start at SQL Server Agent start-up or manually
- Jobs can be assigned to a Category



Secara detil, langkah-langkah SQL Server Agent melakukan pekerjaan tergambar pada bagan berikut.



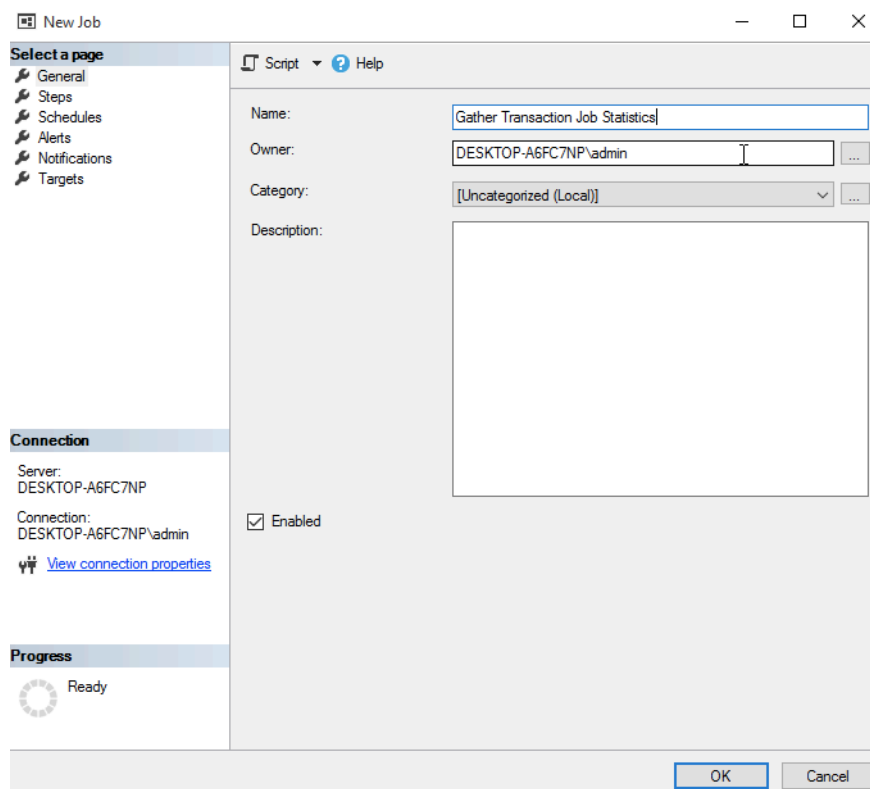
Job atau pekerjaan dari SQL Server Agent senantiasa dilakukan penjadwalan atau scheduling. Penjadwalan tersebut dapat diberi identifier kemudian diapsangkan pada beberapa pekerjaan.



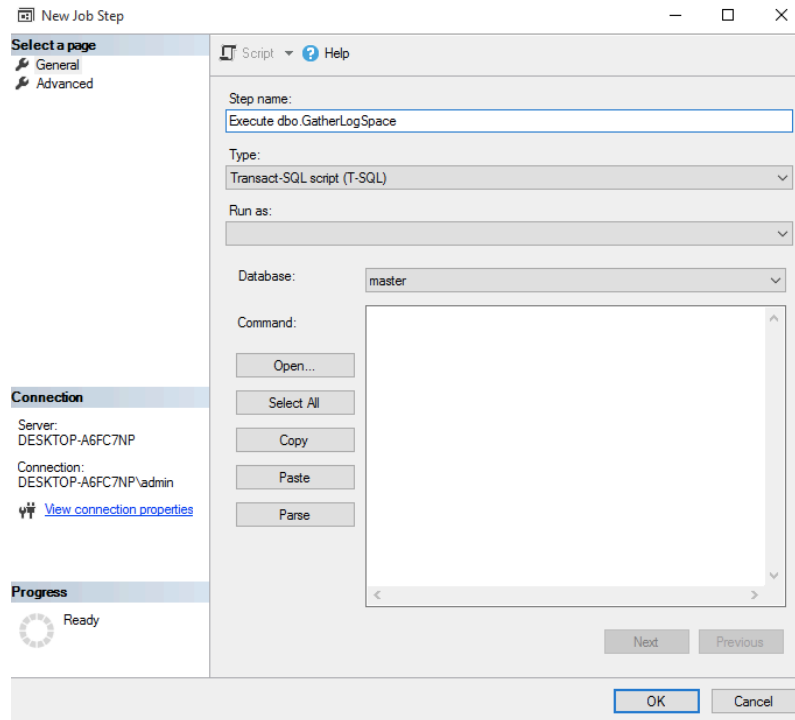
Scripting pada pekerjaan atau job SQL Server Agent dilakukan agar mempermudah dalam pendokumentasian serta memudahkan penggunaannya ketika terjadi kegagalan eksekusi.

1.2.1 Langkah-langkah Percobaan

1. Buat job baru dengan nama “Gather Transaction Log Statistics”. Buka SQL Server Agent, klik kanan Jobs, kemudian klik kanan pilih new job. Masukkan nama job sesuai perintah.



2. Eksekusi Procedure GatherLogSpace. Pilih menu Steps, kemudian New, masukkan 'Execute dbo.GatherLogSpace'. Pada dropdown database, pilih AdminDB. Pada textbox Command, ketikkan 'EXEC dbo.GatherLogSpace', kemudian klik Parse dan OK. Setelah itu buka menu Advanced, perhatikan opsi yang tertera, kemudian klik OK.



3. Jadwalkan pekerjaan SQL Server Agent tiap 10 menit. Pilih Menu Schedules, klik Pick. Pada window pick schedule, perhatikan jadwal yang tampil, klik cancel. Klik New, isi nama schedule dengan 'Every 10 Minutes', pilih opsi 'Daily' pada dropdown, pilih opsi Occurs Every, ubah nilainya menjadi 10 Menit, kemudian klik OK.

New Job Schedule

Name: Jobs in Schedule

Schedule type: ☒ Enabled

One-time occurrence

Date: Time:

Frequency

Occurs:

Recurs every: day(s)

Daily frequency

☐ Occurs once at:

☒ Occurs every: minute(s) Starting at: Ending at:

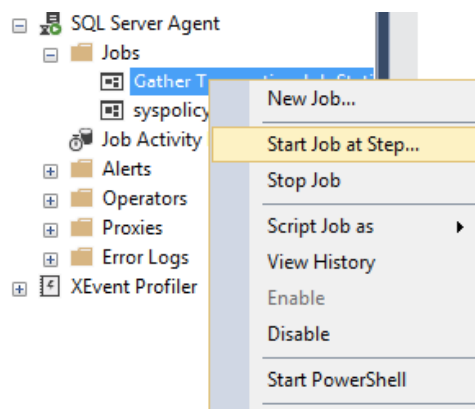
Duration

Start date: ☐ End date: ☒ No end date:

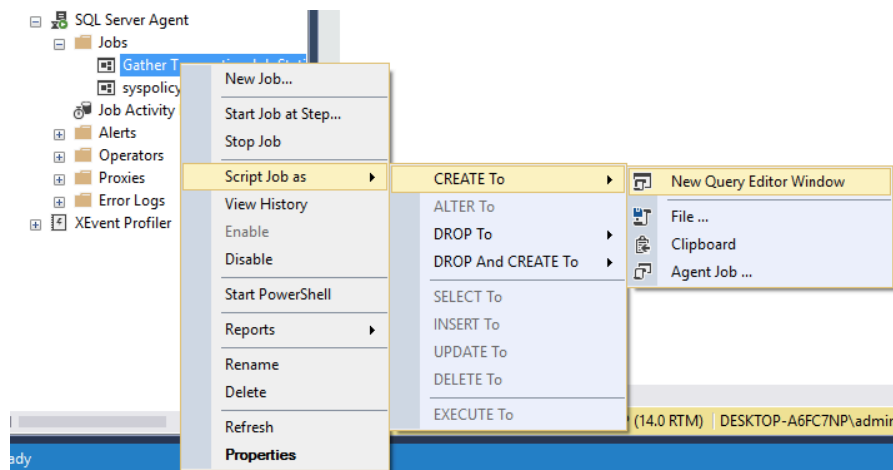
Summary

Description: Occurs every day every 10 minute(s) between 12:00:00 AM and 11:59:59 PM. Schedule will be used starting on 4/8/2019.

4. Klik OK untuk membuat job baru.
5. Eksekusi job secara manual. Pada Object Explorer, expand Jobs, klik kanan Gather Transaction Log Statistics, dan klik Start Job at Step.



6. Review output dari job yang dihasilkan, yaitu akan menampilkan statistik log transaksi pada tabel dbo.LogSpace. Tampilkan data dari tabel tersebut!
7. Lakukan Job Scripting pada panel query. Klik kanan pada Gather Transaction Log Statistics, klik Script Job as, klik CREATE To New Query Editor Window. Perhatikan pada Script yang di-generate.



1.2.2 Soal

1. Setelah melakukan percobaan, apa yang anda pahami tentang Job pada SQL Server Agent?
2. Apa fungsi scheduler? Apa yang terjadi jika sebuah Job tidak dibuat schedule-nya?

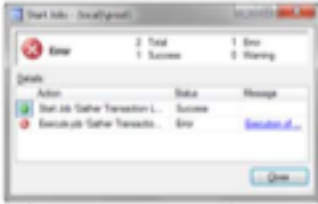
1.3 Mengatur Job pada SQL Server Agent

Dalam melakukan pengaturan SQL Server Agent, kita dapat melihat history Job yang telah dilakukan oleh SQL Server Agent serta melihat data Tabel dan View yang memiliki relasi dengan SQL Server Agent. Berdasarkan dari history yang telah dilakukan, seorang Database Administrator dapat melakukan review dan penanganan pekerjaan yang gagal dilakukan.

Job ID	Step ID	Server	Job Name	Step Name	Step Outcome
1000000001	1	SQLSRV001	Gather Transaction Log Statistics	SQLSRV001	Succeeded
1000000001	1	SQLSRV001	Gather Transaction Log Statistics	SQLSRV001	Succeeded
1000000001	1	SQLSRV001	Gather Transaction Log Statistics	SQLSRV001	Succeeded
1000000001	1	SQLSRV001	Gather Transaction Log Statistics	SQLSRV001	Succeeded
1000000001	1	SQLSRV001	Gather Transaction Log Statistics	SQLSRV001	Succeeded

- SQL Server Agent keeps history information in msdb
- Job history can be queried directly or viewed through SSMS
- Job Activity monitor shows current live information
- Retention of history can be configured based on time or size

- If SQL Server Agent is not running
 - Check settings of the service
 - Check the msdb database
- Review job history
 - Check job outcome to identify the last step that was executed
 - Check job step outcome to identify why the step failed
- If the job did not start, verify:
 - Job is enabled
 - Job is scheduled
 - Schedule is enabled
- Verify that all dependent objects such as databases, files, procedures are available
 - Check that security settings allow access to dependent objects



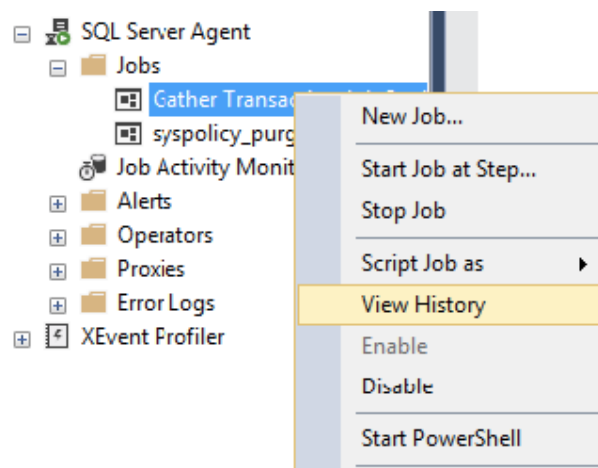
1.3.1 Langkah-langkah Percobaan

1. Ubah stored procedure dbo.GatherLogSpace dengan mengeksekusi script dibawah.

```
USE AdminDB;
GO

ALTER PROCEDURE dbo.GatherLogSpace
AS
INSERT INTO dbo.LogSpace (DatabaseName, LogSizeMB, LogUsedPercent, Status)
EXEC ('DBCC SQLPERF(LOG_SPACE)');
GO
```

2. Eksekusi kembali job Gather Transaction Log Statistics seperti pada praktikum sebelumnya. Percobaan ini akan menghasilkan error.
3. Buka job history pada job tersebut. Perhatikan history-nya, kemudian tampilkan detail masing-masing Log error yang terjadi.



4. Eksekusi perintah T-SQL dibawah ini, tampilkan hasilnya!

```
SELECT sj.name, sjh.run_date, sjh.run_time, sjh.message,
       sjh.run_status, sjh.run_duration
FROM msdb.dbo.sysjobhistory AS sjh
INNER JOIN msdb.dbo.sysjobs AS sj
ON sjh.job_id = sj.job_id
WHERE sjh.step_id = 0
ORDER BY sjh.run_date DESC;
GO
```

5. Eksekusi perintah T-SQL dibawah ini dan tampilkan hasilnya!

```
WITH jobhistory AS
( SELECT sj.name,
  (CASE sjh.run_date
   WHEN 0 THEN NULL
   ELSE CONVERT(datetime,
    STUFF(STUFF(CAST(sjh.run_date AS nchar(8)),
      7,0,'-'),
      5,0,'-')
    + N' '
    + STUFF(STUFF(SUBSTRING(CAST(1000000 + sjh.run_time AS nchar(7)),
      2, 6),
      5, 0, ':'),
      3, 0, ':'),
      120) END) AS RunDate,
    sjh.message,
    sjh.run_status,
    sjh.run_duration
  FROM msdb.dbo.sysjobhistory AS sjh
  INNER JOIN msdb.dbo.sysjobs AS sj
  ON sjh.job_id = sj.job_id
  WHERE sjh.step_id = 0
)
SELECT *
FROM jobhistory
WHERE RunDate > DATEADD(d,-1,SYSDATETIME());
GO
```

6. Eksekusi perintah T-SQL dibawah ini dan tampilkan hasilnya!

```
SELECT sj.name, sjh.run_date, sjh.run_time, sjh.message,
       sjh.run_status, sjh.run_duration
FROM msdb.dbo.sysjobhistory AS sjh
INNER JOIN msdb.dbo.sysjobs AS sj
ON sjh.job_id = sj.job_id
WHERE sjh.step_id > 0
AND sjh.run_status <> 1
ORDER BY sjh.run_date DESC;
GO
```

1.3.2 Soal

1. Pada langkah nomor 4, data apa yang dihasilkan? Berikan penjelasan dan analisis tentang hasil eksekusi langkah tersebut!
2. Pada langkah nomor 5, data apa yang dihasilkan? Berikan penjelasan dan analisis tentang hasil eksekusi langkah tersebut!
3. Pada langkah nomor 6, data apa yang dihasilkan? Berikan penjelasan dan analisis tentang hasil eksekusi langkah tersebut!