



**Putra Pandu Adikara, S.Kom**

**Backup & Recovery**

**Basis Data 2**



**Backup**



# Backup

- ❖ Backup → kopian data yang dapat digunakan untuk restore dan recovery
- ❖ Backup dapat digunakan untuk restore setelah failure
- ❖ Failure disebabkan:
  - Media failure.
  - User errors, misal, tidak sengaja drop table.
  - Hardware failures, misal disk-drive rusak atau permanent loss sebuah server.
  - Natural disasters.



# Tipe & Scope Backup

## ❖ Tipe Backup:

- Data backup → simple recovery model, full recovery model
  - Full Backup
  - Differential Backup
- Transaction Log Backup → Full recovery model; Bulk-logged recovery model

## ❖ Scope dari backup data dapat berupa:

- Keseluruhan database (**Database backups**)
- Sebagian database/partial database (**Partial backups**)
- Set of files or filegroups (**File backups**)



# Tipe Backup: Data Backup

## ❖ Full Backup

- Full backup berisi semua data pada database atau set of filegroups/file tertentu, dan berisi cukup log untuk recovery data



# Tipe Backup: Data Backup

## ❖ Differential Backup

- Differential backup berdasarkan dari full backup terakhir dari data (full backup utk differential backup=*differential base*)
- Differential backup berisi hanya data yang berubah sejak full backup (differential base)
  - Mempercepat proses backup
  - Ukuran backup lebih kecil
- Ketika database makin besar dan jumlah data pada differential backup bertambah
  - Proses pembuatan backup dan restore lambat
  - Buat **differential base baru** untuk differential backup



## ❖ Differential Backup

- Langkah backup:
  - Buat full backup (differential base)
  - Buat differential backup teratur
  - Bila differential backup sudah banyak, buat differential base baru
  
- Langkah restore:
  - Restore pertama kali diperlukan full backup
  - Diikuti restore differential backup paling baru



# Tipe Backup: Transaction Log Backup

## ❖ Transaction log (log backup)

- Meliputi bagian dari transaction log yang active ketika backup dibuat termasuk semua record log yang tidak dibackup log backup sebelumnya.
- Log backup dari urutan yang tidak terinterupsi berisi semua log chain dari database, disebut unbroken
- Dengan unbroken log chain, maka dapat merestore database suatu titik waktu kapan saja
- Langkah backup:
  - Buat full backup, misalnya database backup
  - Buat log backup pertama
  - Buat transaction log backup teratur





# Scope Backup

## ❖ Database backup

- Membackup database

## ❖ Partial backup

- Didesain untuk membackup database yang mengandung beberapa read-only filegroups

## ❖ File backup

- File pada database dapat dibackup dan direstore individual
- Misal database berisi banyak file yang tersimpan di banyak disk dan salah satu rusak, hanya file yang rusak yang direstore, tidak perlu seluruh database



# Ringkasan Tipe & Scope Backup

Scope Backup	Tipe Backup	Description
Database backup	<b>Full Database Backup</b>	Keseluruhan database Isi hingga waktu backup
	<b>Differential Database Backup</b>	Backup semua file di database Hanya berisi data yang dimodifikasi sejak fullvbackup database terbaru tiap file
Partial Backup	<b>Full Partial Backup</b>	Keseluruhan data pada primary filegroup, r/w filegroup, r filegroups
	<b>Differential Database Backup</b>	Backup semua data yang dimodifikasi sejak full partial backup paling baru dari set of filegroup yang sama
File Backup	<b>File Backup</b>	Backup full semua data di satu atau lebih file/filegroups
	<b>Differential File Backup</b>	Backup satu atau lebih file yang berisi data yang dimodifikasi sejak full backup tiap file



# Strategi Backup & Restore

- ❖ Strategi backup meliputi:
  - Tipe dan frekuensi backup,
  - Kecepatan hardware,
  - Bagaimana backup diuji
  - Dimana dan bagaimana media backup disimpan
- ❖ Strategi restore meliputi:
  - Siapa yang melakukan restore
- ❖ Disarankan untuk mendokumentasikan prosedur backup dan restore



# Desain Strategi Backup

- ❖ Berapa jam per hari aplikasi mengakses database?
  - Bila ada prediksi masa off-peak (benar-benar/puncak luang), jadwalkan full database backup pada masa itu
- ❖ Berapa sering perubahan dan update terjadi? bila sering:
  - **Simple recovery model**
    - Jadwalkan differential backup antara full database backup
    - Differential backup menangkap hanya perubahan sejak full database backup
  - **Full recovery model**
    - Jadwalkan log backup yang sering
    - Penjadwalan differential backup antara full backup dapat mengurangi waktu restore dengan mengurangi jumlah log backup untuk merestore



# Desain Strategi Backup

- ❖ Apakah perubahan terjadi di bagian kecil atau besar database?
  - Untuk database besar yang terkonsentrasi pada bagian files atau filegroups
    - Partial backup
    - File backup
- ❖ Berapa banyak ruang disk dibutuhkan untuk backup?
  - Perkirakan disk space terutama untuk full database backup
  - Backup berisi data aktual pada database, tidak termasuk space kosong/tidak digunakan
    - Seharusnya ukuran backup lebih kecil dibanding database itu sendiri
    - Gunakan system stored procedure `sp_spaceused`



# Penjadwalan Backup

- ❖ Buat maintenance plan
- ❖ Buat dan jadwalkan job schedule
  - SQL Server Agent harus started



# Restore & Recovery



# Restore

## ❖ Restore

- Proses mengkopi data dari backup dan mengaplikasikan logged transaction ke data untuk rolling-forward ke titik recovery yang diinginkan
- Contoh syntax untuk restore database:
  - `RESTORE DATABASE database`
  - `FROM device_name [WITH NORECOVERY]`

## ❖ Recovery

- Proses rolling-forward uncommitted transaction, bila ditentukan, mengembalikan database ke status online
- Contoh syntax
  - `RESTORE DATABASE database`
  - `FROM device_name WITH RECOVERY`





# Restore

- ❖ SQL Server mendukung restore beberapa level:
  - **Database (*a complete database restore*)**
    - Seluruh database direstore dan direcover,
    - Database offline ketika operasi restore & recovery
  - **Data File (*a file restore*)**
    - Data file atau sekumpulan file di restore dan direcover
    - Filegroups berisi file-file bersangkutan offline ketika restore & recover
  - **Data Page (*a page restore*)**
    - Pada model full recovery atau bull-logged recovery, dapat merestore individual database
    - Page restore dapat dilakukan pada database apapun, tidak tergantung jumlah dari filegroup



# Recovery Model

## ❖ Recovery model

- Menentukan tipe backup dan skenario restore
- Mengontrol bagaimana transaction log dikelola

## ❖ Biasanya database menggunakan model recovery

- **Simple recovery model**
- **Full recovery model**
  - Dapat dilengkapi dengan beralih ke **bulk-logged recovery model** sebelum bulk operation
- **Bulk-Logged Recovery Model**



# Simple Recovery Model

## ❖ Simple Recovery Model

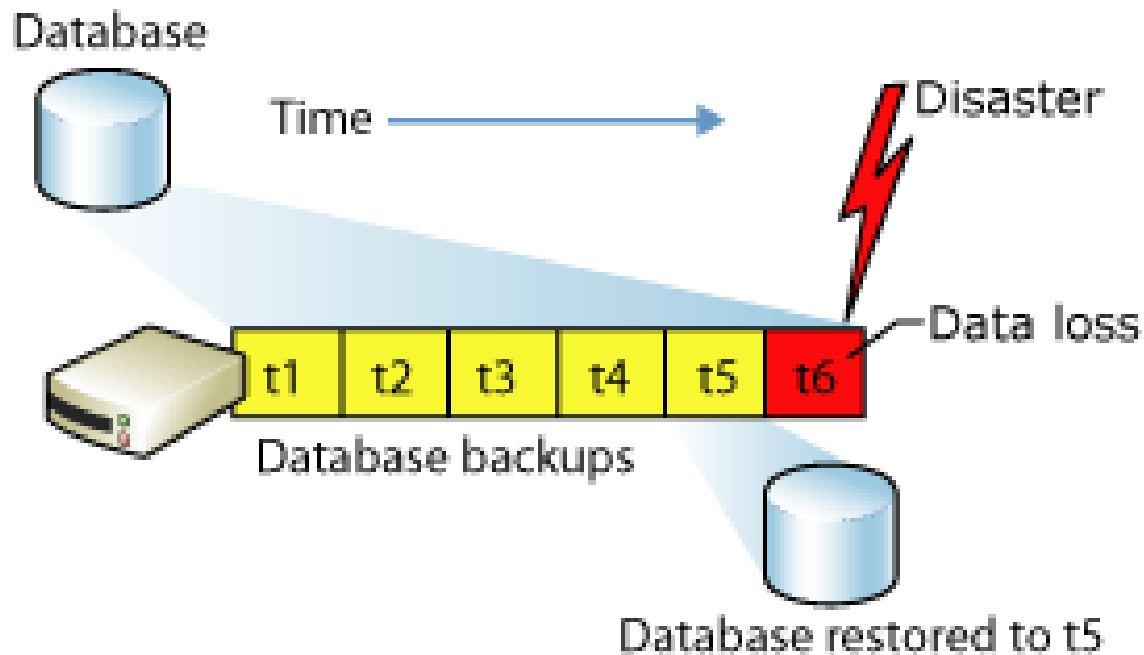
- Strategi ini hanya menggunakan bentuk sederhana untuk backup dan restore
- Support database backup, file backup
- Tidak support log backup
- Database hanya bisa direstore ke akhir dari backup paling baru
- Tidak cocok untuk sistem produksi dimana hilangnya perubahan terakhir tidak diperbolehkan (gunakan full recovery dalam kasus ini)



# Contoh Simple Recovery Model

## ❖ Contoh:

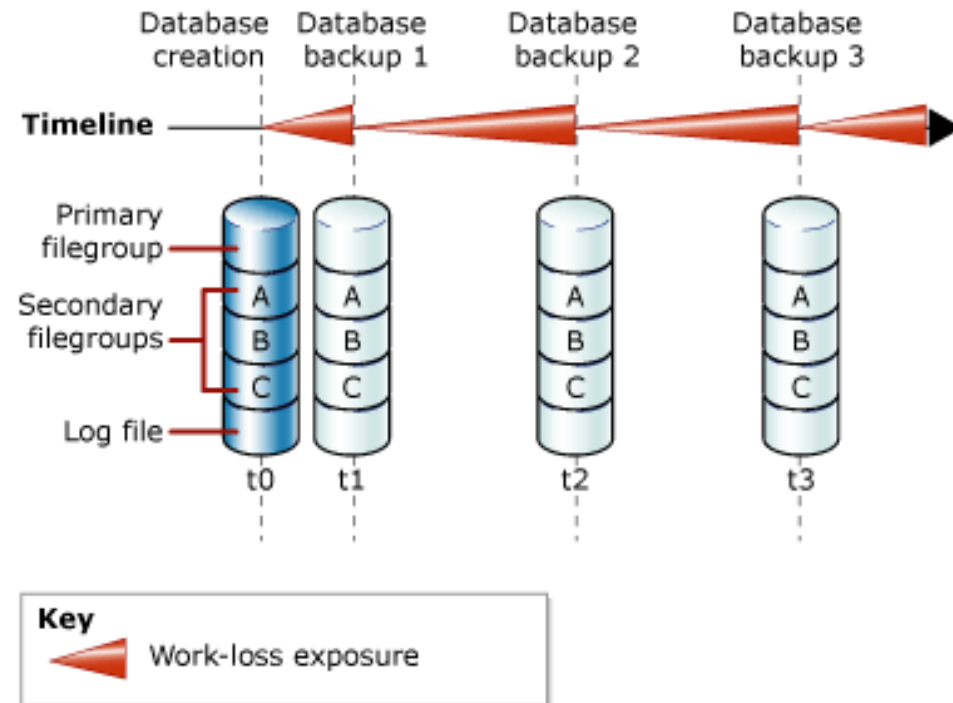
- Ada 5 backup database (hanya yang terbaru): t1-t5
- Dimisalkan harus direstore ke waktu t5 maka
  - Database kembali ke waktu t5
  - Semua update setelah t5 hilang





# Ilustrasi Simple Recovery Model

- **Full database backup**
  - Cocok untuk database kecil sehingga dapat sering dibackup



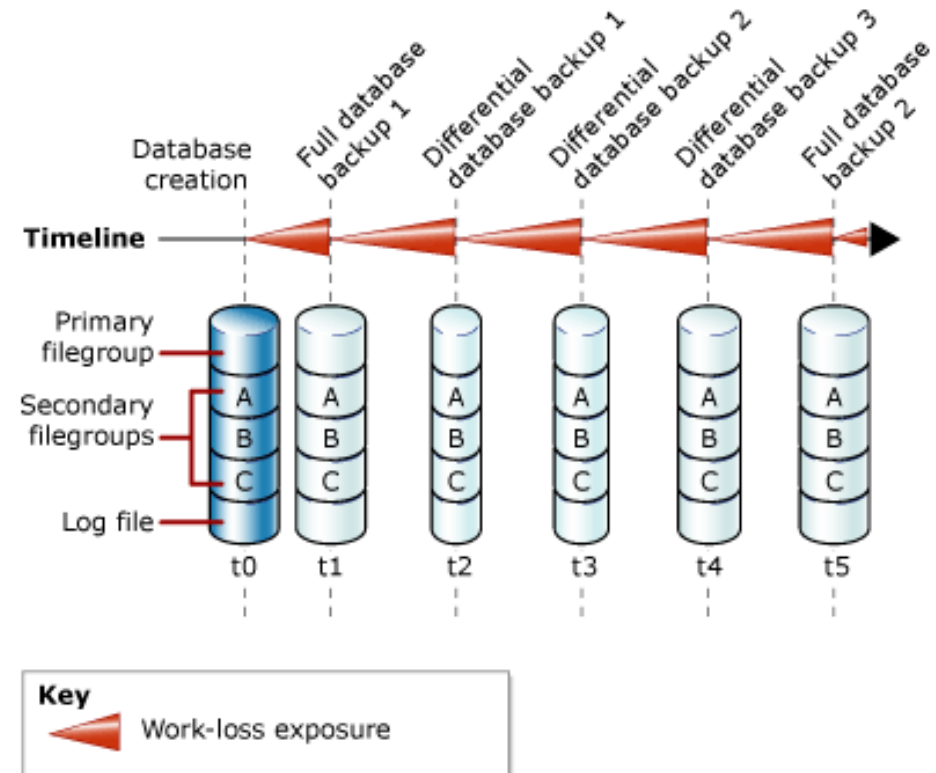


# Ilustrasi Simple Recovery Model

❖ Strategi backup mengurangi work-loss exposure dengan:

## ■ Differential database backup

- dibanding full database
- Setelah database backup pertama, sekumpulan differential backup dibuat (3 diff backup)
- Setelah diff backup ketiga cukup besar, backup berikutnya adalah database backup untuk membuat differential base baru





# Full Recovery Model

## ❖ Full Recovery Model

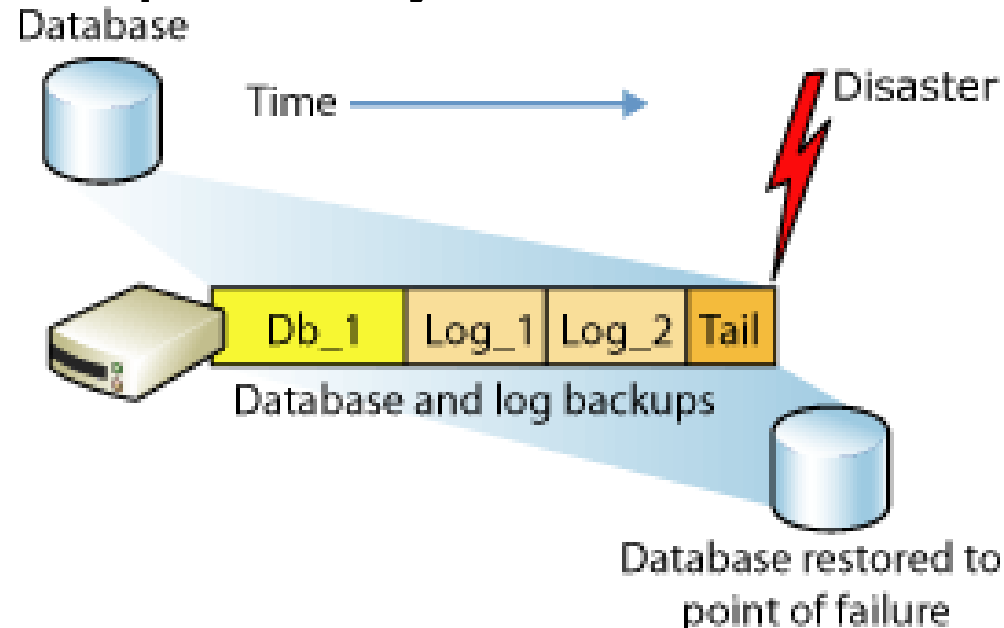
- Menggunakan log backup untuk mencegah kehilangan data karena berbagai macam skenario kegagalan
- Transaction log (log backup) diperlukan
- Dapat restore database ke suatu titik waktu yang terdapat dalam log backup (*point-in-time recovery*)
- Dapat menggunakan log-backup untuk roll-forward database ke suatu titik pada suatu log-backup
- Misal bisa membackup *active log (Tail)* setelah terjadi bencana, maka dapat me-restore database ke titik terjadi kegagalan tanpa kehilangan data
  - Kelemahannya membutuhkan media penyimpanan besar
  - Waktu restore dan kompleksitas meningkat



# Ilustrasi Full Recovery Model

## ❖ Full database backup +Log (yang paling mudah)

- Backup full database: Db\_1; Log backup: Log\_1, Log\_2
- Setelah Log\_2, hilangnya data terjadi
- **Sebelum** ketiga backup **direstore**, db admin **harus membackup active log (tail of the log / Tail)**
- **Restore** db\_1, Log\_1, Log\_2 **tanpa recovery database**
- Db admin **merestore dan merecover Tail**
- Database ter-recover ke titik kegagalan, merecover semua data



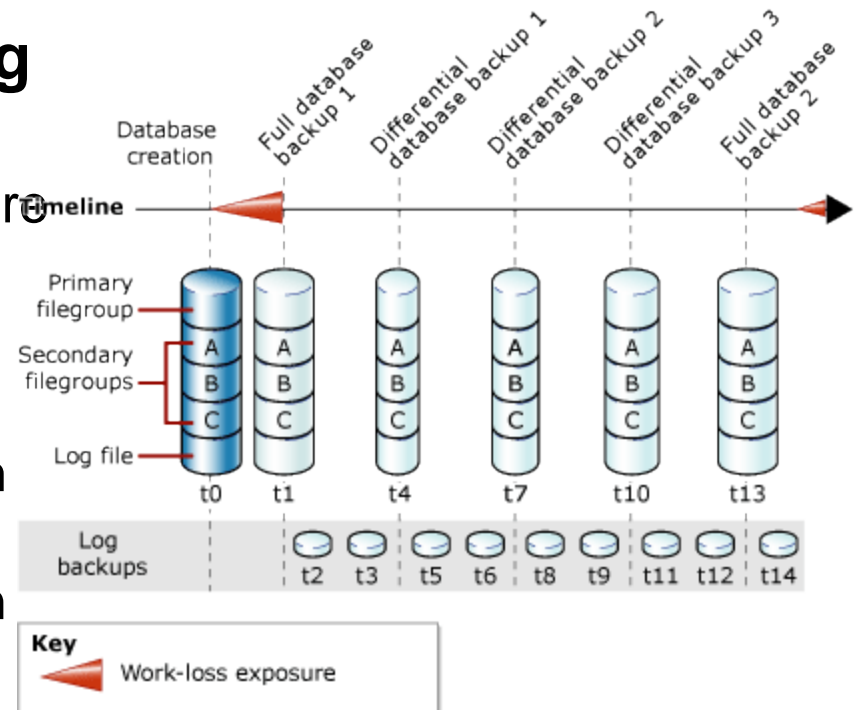




# Ilustrasi Full Recovery Model

❖ Strategi backup mengurangi work-loss exposure dengan:

- **Differential Backup+Log**
- Transaction log backup mengurangi work-less exposure potensial setelah log backup terbaru, t14
- Rangkaian 3 diff backup digunakan mengurangi jumlah transaction log that akan direstore kalau ada kegagalan
  - 3 diff backup cukup besar utk backup berikutnya sbg full database backup





- ❖ Sebelum backup database pertama, ada kemungkinan hilangnya data pada t0-t1
- ❖ Setelah itu log backup yg rutin mengurangi kemungkinan hilangnya data setelah log backup terakhir (pd gambar, stlh t14)
- ❖ Bila ada kegagalan, maka db admin membackup **tail of the log (Tail)** atau log yg belum di backup.
- ❖ Bila tail-log sukses di-backup, db admin dapat menghindari kehilangan data dgn me-restore ke titik kegagalan



# Skenario Restore

Restore scenario	Under simple recovery model	Under full/bulk-logged recovery models
<b>Complete database restore</b>	<p>Dasar strategi restore</p> <p>Restore database komplit dari :</p> <ul style="list-style-type: none"> <li>- Full database backup,</li> <li>- Differential backup (bila ada)</li> </ul>	<p>Dasar strategi restore</p> <p>Restore menggunakan:</p> <ul style="list-style-type: none"> <li>- Full database backup</li> <li>- Differential backup (bila ada)</li> <li>- Diikuti semua log backup berikutnya (sesuai urutan)</li> </ul> <p>Database restore selesai setelah recovery last log-backup dan restore (RESTORE WITH RECOVERY)</p>
<b>File restore *</b>	<p>Restore 1 atau lebih read-only file yang rusak, tanpa restore seluruh database.</p> <p>Hanya bisa dilakukan bila punya setidaknya satu filegroup yg read-only</p>	<p>Restore 1 atau lebih file yang rusak, tanpa restore seluruh database.</p> <p>Dapat dilakukan ketika database offline atau online (tergantung)</p> <p>Selama restore, filegroups berisi file direstore akan offline</p>
<b>Page restore</b>	Tidak bisa	<p>Restore 1 atau lebih page yang rusak.</p> <p>Dapat dilakukan ketika database offline atau online (tergantung)</p> <p>Selama restore, page-page yang direstore akan offline</p> <p>Unbroken chain dari log backup harus tersedia, hingga log file saat ini, dan diaplikasikan semua utk mengupdate page dengan log file saat ini</p>
<b>Piecemeal restore *</b>	<p>Restore dan recover database pada level filegroup, dimulai dari primary filegroup dan semua read/write, secondary filegroups</p>	<p>Restore dan recover database pada level filegroup, dimulai dari primary filegroup</p>



# Recovery Model dan Supported Restore Ops

Restore operation	Full recovery model	Bulk-logged recovery model	Simple recovery model
<b>Data recovery</b>	Complete recovery (if the log is available).	Some data-loss exposure.	Any data since last full or differential backup is lost.
<b>Point-in-time restore</b>	Any time covered by the log backups.	Disallowed if the log backup contains any bulk-logged changes.	Not supported.
<b>File restore *</b>	Full support.	Sometimes.**	Available only for read-only secondary files.
<b>Page restore *</b>	Full support.	Sometimes.**	None.
<b>Piecemeal (filegroup-level) restore *</b>	Full support.	Sometimes.**	Available only for read-only secondary files.

❖ \* SQL Server 2005 Enterprise Edition and later

❖ \*\* Baca Restriction under Simple Recovery Model



# Backup Database T-SQL



# Full Database Backup

## ❖ Syntax:

```
BACKUP DATABASE database  
TO backup_device [ ,...n]  
[WITH with_options [ ,...0]
```



# Full Database Backup: Contoh

## ❖ Contoh backup ke disk :

```
USE AdventureWorks2008R2;
GO
BACKUP DATABASE AdventureWorks2008R2
TO DISK =
'Z:\SQLServerBackups\AdventureWorks2008R2.Bak'
WITH FORMAT,
    MEDIANAME = 'Z_SQLServerBackups',
    NAME = 'Full Backup of AdventureWorks2008R2';
GO
```



# Differential Backup

## ❖ Syntax:

- **BACKUP DATABASE** database\_name **TO**  
<backup\_device> **WITH DIFFERENTIAL**





# Differential Backup: Contoh

## ❖ Contoh:

-- Create a full database backup first.

```
BACKUP DATABASE MyAdvWorks TO MyAdvWorks_1  
WITH INIT
```

```
GO
```

-- Time elapses.

-- Create a differential database backup, appending the backup  
-- to the backup device containing the full database backup.

```
BACKUP DATABASE MyAdvWorks TO MyAdvWorks_1  
WITH DIFFERENTIAL
```

```
GO
```





❖ Baca untuk backup dengan SQL Management Studio



# Restore Database T-SQL



# Simple Recovery Model

## ❖ Full database backup restore

### ❖ Syntax:

- **RESTORE DATABASE** database\_name **FROM** backup\_device [ **WITH NORECOVERY** ]

## ❖ Differential database backup restore

### ❖ Syntax:

- **RESTORE DATABASE** database\_name **FROM** backup\_device **WITH RECOVERY**



# Simple Recovery Model

## ❖ Full database backup restore

```
USE master;
```

```
--Make sure the database is using the simple recovery model.
```

```
ALTER DATABASE AdventureWorks2008R2 SET RECOVERY SIMPLE;
```

```
GO
```

```
-- Back up the full AdventureWorks2008R2 database.
```

```
BACKUP DATABASE AdventureWorks2008R2 TO DISK =
```

```
'Z:\SQLServerBackups\AdventureWorks2008R2.bak'
```

```
WITH FORMAT;
```

```
GO
```

```
--Create a differential database backup.
```

```
BACKUP DATABASE AdventureWorks2008R2 TO DISK = 'Z:\SQLServerBackups\AdventureWorks2008R2.bak'
```

```
WITH DIFFERENTIAL;
```

```
GO
```

```
--Restore the full database backup (from backup set 1).
```

```
RESTORE DATABASE AdventureWorks2008R2 FROM DISK='Z:\SQLServerBackups\AdventureWorks2008R2.bak'
```

```
WITH FILE=1, NORECOVERY;
```

```
--Restore the differential backup (from backup set 2).
```

```
RESTORE DATABASE AdventureWorks2008R2 FROM
```

```
DISK='Z:\SQLServerBackups\AdventureWorks2008R2.bak'
```

```
WITH FILE=2, RECOVERY;
```

```
GO
```



# Simple Recovery Model

## ❖ Differential backup restore

-- Assume the database is lost, and restore full database,  
-- specifying the original full database backup and NORECOVERY,  
-- which allows subsequent restore operations to proceed.

```
RESTORE DATABASE MyAdvWorks  
    FROM MyAdvWorks_1  
    WITH NORECOVERY
```

```
GO
```

-- Now restore the differential database backup, the second backup  
on  
-- the MyAdvWorks\_1 backup device.

```
RESTORE DATABASE MyAdvWorks  
    FROM MyAdvWorks_1  
    WITH FILE = 2,  
    RECOVERY
```

```
GO
```



# Full Recovery Model

## ❖ Basic restore

1. **RESTORE DATABASE** database **FROM** full database backup **WITH NORECOVERY;**
2. **RESTORE DATABASE** database **FROM** full\_differential\_backup **WITH NORECOVERY;**
3. **RESTORE LOG** database **FROM** log\_backup **WITH NORECOVERY;**
4. Repeat this restore-log step for each additional log backup.
5. **RESTORE DATABASE** database **WITH RECOVERY;**





# Full Recovery Model

## ❖ Full database restore

```
USE master;
--Create tail-log backup.
BACKUP LOG AdventureWorks2008R2 TO DISK = 'Z:\SQLServerBackups\AdventureWorks2008R2FullRM.bak'
    WITH NORECOVERY;
GO
--Restore the full database backup (from backup set 1).
RESTORE DATABASE AdventureWorks2008R2 FROM DISK='Z:\SQLServerBackups\AdventureWorks2008R2FullRM.bak'
    WITH FILE=1,
    NORECOVERY;
--Restore the regular log backup (from backup set 2).
RESTORE LOG AdventureWorks2008R2 FROM DISK = 'Z:\SQLServerBackups\AdventureWorks2008R2FullRM.bak'
    WITH FILE=2,
    NORECOVERY;
--Restore the tail-log backup (from backup set 3).
RESTORE LOG AdventureWorks2008R2 FROM DISK = 'Z:\SQLServerBackups\AdventureWorks2008R2FullRM.bak'
    WITH FILE=3,
    NORECOVERY;
GO
--recover the database:
RESTORE DATABASE AdventureWorks2008R2 WITH RECOVERY;
GO
```



# Full Recovery Model

## ❖ Differential backup restore

-- Assume the database is lost at this point. Now restore the full  
-- database. Specify the original full database backup and NORECOVERY.  
-- NORECOVERY allows subsequent restore operations to proceed.

```
RESTORE DATABASE MyAdvWorks FROM MyAdvWorks_1  
    WITH NORECOVERY
```

GO  
-- Now restore the differential database backup, the second backup on  
-- the MyAdvWorks\_1 backup device.

```
RESTORE DATABASE MyAdvWorks    FROM MyAdvWorks_1  
    WITH FILE = 2,  
    NORECOVERY
```

GO  
-- Now restore each transaction log backup created after  
-- the differential database backup.

```
RESTORE LOG MyAdvWorks    FROM MyAdvWorks_log1  
    WITH NORECOVERY
```

GO  
RESTORE LOG MyAdvWorks FROM MyAdvWorks\_log2  
 WITH RECOVERY

GO



# Full Recovery Model

## ❖ Transaction log backup restore

```
RESTORE LOG <database_name> FROM <backup_device> WITH  
NORECOVERY;
```

```
RESTORE DATABASE <database_name> WITH RECOVERY;
```

```
GO
```

## ❖ Contoh:

```
RESTORE DATABASE AdventureWorks2008R2  
FROM AdventureWorks2008R2_1  
WITH NORECOVERY
```

```
GO
```

```
RESTORE LOG AdventureWorks2008R2  
FROM AdventureWorks2008R2_log  
WITH FILE = 1,  
WITH NORECOVERY
```

```
GO
```

```
RESTORE DATABASE AdventureWorks2008R2  
WITH RECOVERY
```

```
GO
```



# Restore to Point-in-Time

## ❖ Restore dari log ke suatu titik waktu (point-in-time)

1. Execute the **RESTORE DATABASE** statement using the **NORECOVERY** option.
2. Execute the **RESTORE LOG** statement to apply each log backup, specifying:
  - The name of the database to which the transaction log is applied.
  - The backup device from where the transaction log backup is restored.
  - The **RECOVERY** and **STOPAT** options. If the transaction log backup does not contain the requested time (for example, if the time specified is beyond the end of the time covered by the transaction log), a warning is generated and the database remains unrecovered.



# Restore to Point-in-Time: Contoh

```
RESTORE DATABASE AdventureWorks2008R2  
FROM AdventureWorks2008R2Backups  
WITH FILE=3, NORECOVERY;
```

```
RESTORE LOG AdventureWorks2008R2  
FROM AdventureWorks2008R2Backups  
WITH FILE=4, NORECOVERY, STOPAT = 'Apr 15, 2020 12:00 AM';
```

```
RESTORE LOG AdventureWorks2008R2  
FROM AdventureWorks2008R2Backups  
WITH FILE=5, NORECOVERY, STOPAT = 'Apr 15, 2020 12:00 AM';  
RESTORE DATABASE AdventureWorks2008R2 WITH RECOVERY;  
GO
```



# Restore to the Point of Failure

1. Back up the tail of the log by running the following basic BACKUP statement:

```
BACKUP LOG <database_name> TO <backup_device>  
WITH NORECOVERY, NO_TRUNCATE;
```

2. Restore a full database backup by running the following basic RESTORE DATABASE statement:

```
RESTORE DATABASE <database_name> FROM <backup_device>  
WITH NORECOVERY;
```

3. Optionally, restore a differential database backup by running the following basic RESTORE DATABASE statement:

```
RESTORE DATABASE <database_name> FROM <backup_device>  
WITH NORECOVERY;
```

4. Apply each transaction log, including the tail-log backup you created in step 1, by specifying WITH NORECOVERY in the RESTORE LOG statement:

```
RESTORE LOG <database_name> FROM <backup_device>  
WITH NORECOVERY;
```

5. Recover the database by running the following RESTORE DATABASE statement:

```
RESTORE DATABASE <database_name>  
WITH RECOVERY;
```



# Restore to the Point of Failure: Contoh

```
/* Example of restoring a to the point of failure */
-- Step 1: Create a tail-log backup by using WITH NORECOVERY.
BACKUP LOG AdventureWorks2008R2 TO DISK='C:\AdventureWorks2008R2_Log.bck'
    WITH NORECOVERY;
GO
-- Step 2: Restore the full database backup.
RESTORE DATABASE AdventureWorks2008R2 FROM DISK='C:\AdventureWorks2008R2_Data.bck'
    WITH NORECOVERY;
GO
-- Step 3: Restore the first transaction log backup.
RESTORE LOG AdventureWorks2008R2 FROM DISK='C:\AdventureWorks2008R2_Log.bck'
    WITH NORECOVERY;
GO
-- Step 4: Restore the tail-log backup.
RESTORE LOG AdventureWorks2008R2 FROM DISK='C:\AdventureWorks2008R2_Log.bck'
    WITH NORECOVERY;
GO
-- Step 5: Recover the database.
RESTORE DATABASE AdventureWorks2008R2
    WITH RECOVERY;
GO
```



- ❖ Baca lebih jelas:
- ❖ Untuk restore file, page, piecema
- ❖ <http://msdn.microsoft.com/en-us/library/ms186858.aspx>