

## MODUL 7

### BACKUP AND RECOVERY SYSTEM

#### A. Tujuan Praktikum

- a. Memahami materi tentang Backup dan Restore di MySQL
- b. Mampu mengimplementasikan Backup dan Restore di MySQL

#### B. Dasar Teori

Sistem basis data rentan terhadap berbagai kegagalan. Baik itu akibat pemadaman listrik mendadak, bug perangkat lunak, maupun kerusakan perangkat keras, Database Management System (DBMS) harus memiliki mekanisme untuk dapat beroperasi kembali tanpa kehilangan data atau meninggalkannya dalam kondisi rusak. Proses untuk mengembalikan basis data ke keadaan yang benar dan konsisten ini disebut recovery (pemulihan). Tujuan utama dari pemulihan adalah untuk menjaga "atomicity" dan "durability", yang merupakan dua komponen penting dari sifat ACID dalam basis data:

1. Atomicity (Atomisitas): Sebuah transaksi harus diselesaikan sepenuhnya atau tidak dijalankan sama sekali.
2. Durability (Daya Tahan): Setelah transaksi dinyatakan *commit*, perubahan yang terjadi harus tetap tersimpan di basis data, bahkan jika terjadi kegagalan setelahnya.

Beberapa teknik pemulihan utama dalam sistem basis data adalah sebagai berikut:

##### 1. Log-Based Recovery (Pemulihan Berbasis Log)

Ini adalah salah satu metode pemulihan yang paling umum digunakan dalam basis data modern. Sistem menjaga berkas log (log file atau jurnal) di penyimpanan stabil (*stable storage*) yang mencatat setiap perubahan baik insert, update, maupun delete sebelum diterapkan ke basis data. Jika terjadi kegagalan, DBMS membaca log tersebut untuk menentukan langkah yang harus diambil:

- Undo: Membatalkan perubahan dari transaksi yang belum selesai (untuk menjaga atomicity).
- Redo: Menerapkan kembali perubahan dari transaksi yang sudah *commit* (untuk menjaga durability).

Log-based recovery umumnya diterapkan dalam dua cara:

- Immediate Update (Undo/Redo):  
Basis data dapat diperbarui bahkan sebelum transaksi di-*commit*. Jika terjadi kegagalan, sistem akan melakukan *undo* terhadap transaksi yang belum selesai dan *redo* terhadap transaksi yang sudah *commit*.
- Deferred Update (No-Undo/Redo):  
Perubahan hanya diterapkan setelah transaksi di-*commit*. Jika terjadi kegagalan, tidak perlu melakukan *undo* — cukup *redo* terhadap transaksi yang sudah *commit*.

## 2. Shadow Paging

Shadow Paging adalah teknik pemulihan alternatif yang tidak memerlukan log. Prinsipnya adalah menjaga dua versi halaman basis data selama transfusi berlangsung: *current page table* dan *shadow page table*. *Shadow page table* menunjuk ke halaman asli (belum dimodifikasi) dari basis data sebelum transaksi dimulai. Ini berfungsi sebagai “bayangan” dari keadaan basis data yang konsisten. Ketika transaksi mulai memodifikasi data, sistem membuat salinan baru dari halaman yang dimodifikasi. *Current page table* kemudian diperbarui untuk menunjuk ke halaman baru ini, sementara *shadow page table* tetap tidak berubah. Jika transaksi *commit*, maka *current page table* akan menjadi *shadow* yang baru. Jika transaksi gagal, sistem cukup menghapus halaman yang telah dimodifikasi dan kembali menggunakan *shadow* tanpa perlu melakukan *undo* atau *redo*.

## 3. Checkpointing – Mempercepat Proses Pemulihan

Checkpointing bukanlah teknik pemulihan yang berdiri sendiri, melainkan optimisasi penting yang bekerja bersama Log-Based Recovery. Tanpa checkpoint, proses pemulihan dari kegagalan akan memerlukan pemrosesan seluruh isi log, yang bisa memakan waktu lama. Checkpoint berfungsi seperti penanda atau “bookmark” di dalam log. Secara berkala, sistem melakukan operasi checkpoint, yang meliputi:

1. Menyimpan semua catatan log dari memori ke penyimpanan permanen.
2. Menyimpan semua halaman basis data yang telah dimodifikasi ke penyimpanan permanen.
3. Menulis catatan khusus *checkpoint record* ke dalam log file.

Ketika proses pemulihan dibutuhkan, DBMS akan mencari checkpoint terakhir. Sistem tahu bahwa setiap transaksi yang sudah *commit* sebelum checkpoint tersebut sudah aman tersimpan secara permanen. Oleh karena itu, DBMS hanya perlu membaca log setelah checkpoint untuk menentukan transaksi mana yang perlu di-*undo* atau di-*redo*, sehingga mempercepat proses recovery secara signifikan, cukup menghapus halaman yang telah dimodifikasi dan kembali menggunakan *shadow* tanpa perlu melakukan *undo* atau *redo*.

## 4. Backup dan Restore

Teknik ini berfungsi sebagai perlindungan terakhir terhadap kegagalan berat, seperti kerusakan total pada media penyimpanan (disk crash), dengan cara menyimpan salinan cadangan (backup) dari berkas basis data. Terdapat beberapa jenis backup:

- Full Backup yaitu menyalin seluruh isi basis data secara lengkap.
- Differential Backup yaitu menyalin hanya data yang berubah sejak *full backup* terakhir dilakukan.
- Transaction Log Backup yaitu menyalin isi dari log transaksi.

Jenis backup ini sangat penting karena memungkinkan pemulihan basis data hingga ke titik waktu tertentu (*point-in-time recovery*). Misalnya, dapat memulihkan backup penuh dari Minggu malam, lalu menerapkan log transaksi untuk memulihkan basis data hingga beberapa saat sebelum terjadi kegagalan pada Selasa pagi.

Backup dan recovery terjadi karena adanya kerusakan atau kegagalan operasi pada basis data, beberapa sebab kerusakan atau kegagalan operasi tersebut antara lain:

- a. Aliran Listrik terputus dapat mengakibatkan hilangnya informasi yang ada di memori utama dan register.
- b. Kesalahan operator (human error), database administrator melakukan kesalahan operasi yang disengaja maupun tidak disengaja
- c. Kesalahan perangkat lunak dapat mengakibatkan hasil pengolahan tidak benar, informasi yang disajikan salah dan database menjadi tidak konsisten.
- d. Disk rusak mengakibatkan hilangnya informasi atau rusaknya database yang ada di dalam disk.

Dari penyebab diatas dapat terjadi beberapa jenis kerusakan, diantaranya Adalah:

- a. Kegagalan transaksi (transaction failure)  
Kegagalan transaksi dapat terjadi karena kesalahan logika (logical error) Dimana program tidak dapat melanjutkan eksekusi program karena adanya kondisi tertentu seperti masukan yang salah/rusa, data yang tidak tersedia, logika program tidak tepat. Selain itu, dapat juga terjadi karena kesalahan system (system error) dimana system telah memasuki kondisi yang tidak diharapkan (deadlock) sebagai hasil dari tidak dieksekusinya program secara normal.
- b. Kerusakan system (system crash)  
Salah satu penyebabnya adalah hardware macet (hang) yang mengakibatkan penyimpanan sementara hilang.
- c. Kegagalan/kerusakan disk (disk failure)  
Terjadinya bad sector atau disk macet pada saat berlangsungnya operasi I/O ke disk.
- d. Kerusakan terhadap disk (media penyimpanan), kerusakan data karena aktivitas pemakai ataupun kerusakan data oleh aplikasi eksternal dapat diantisipasi dengan melakukan operasi backup secara periodik. Berdasarkan waktu pelaksanaan atau strategi terdapat dua jenis operasi backup yaitu:
  - a. Backup statis, dimana backup dilakukan dengan lebih dulu menonaktifkan basis data secara keseluruhan.
  - b. Backup dinamis, dimana backup dilakukan tanpa penonaktifan basis data.

Terdapat tiga pilihan skema untuk menjalankan mekanisme recovery secara otomatis begitu kerusakan atau kegagalan sistem telah terjadi, skema tersebut adalah:

1. File Log dengan Penundaan Pengubahan (Incremental Log with Deferred Update)
2. File Log dengan Pengubahan Langsung (Incremental Log with Immediate Updates)
3. Page Bayangan (Shadow Paging), yang memerlukan akses ke disk yang lebih sedikit.

Dalam MySQL terdapat beberapa cara yang dapat digunakan untuk melakukan proses backup dan restore, diantaranya adalah sebagai berikut:

1. Untuk proses Backup
  - a. Menggunakan perintah BACKUP TABLE, OUTLIFE, SELECT INTO, dll
  - b. Melakukan penyalinan file secara langsung
2. Untuk proses Restore
  - a. Menggunakan perintah RESTORE
  - b. Menggunakan statement LOAD DATA INFILE

MySQL menyediakan fasilitas untuk memudahkan dalam mendokumentasikan perintah perintah yang digunakan untuk membuat database dan tabel-tabel yang ada di dalamnya.

### C. Latihan Praktikum

#### **Backup dan Restore di PHPMyAdmin**

PHPMyAdmin Adalah salah satu alat bantu dalam mengelola database MySQL. Salah satu fasilitas yang ada pada PHPMyAdmin yaitu menggunakan GUI sehingga mempermudah melakukan manipulasi data, dari pembuatan database sampai pada manipulasi dan organisasi data di dalamnya. Langkah-langkah dalam melakukan backup dan recovery pada PHPMyAdmin Adalah sebagai berikut :

1. Jalankan PHPMyAdmin
2. Buka salah satu database yang ada pada kolom sebelah kiri
3. Klik export yang ada pada menu bar
4. Pilihlah salah satu format backup yang akan disimpan, kemudian klik kirim. Secara otomatis file backup akan di download, lalu buka file download.
5. Buka Kembali PHPMyAdmin untuk melakukan proses recovery.
6. Masuk kedalam database yang ingin di recovery. Klik menu import pada menu bar untuk memulai proses recovery.
7. Klik button pilih file untuk memilih file backup. Tampil kotak dialog untuk memilih Lokasi file backup berada dan memilih file backup. Setelah file backup ditemukan, klik open dan proses recovery akan berlangsung
8. Tampilan akan Kembali ke browser. Klik kirim yang berada di sisi bawah.
9. Jika proses import berhasil maka akan muncul pesan bahwa import database berhasil dilakukan.

## Backup – Menggunakan perintah BACKUP (MySQL)

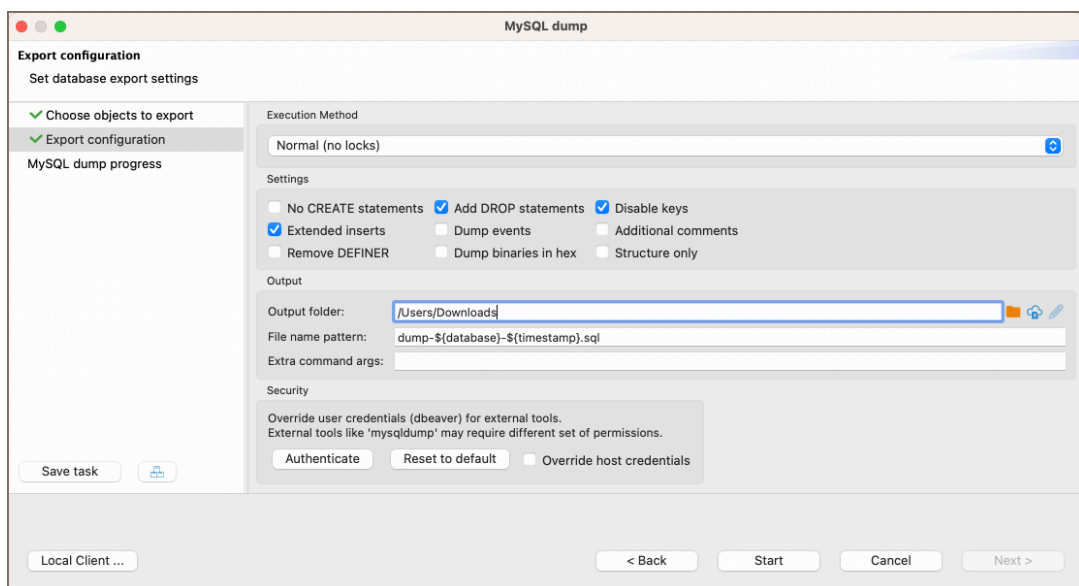
PostgreSQL menyediakan dua utilitas utama untuk manajemen backup :

1. `pg_dump` adalah perintah yang digunakan untuk membuat backup database.
2. `pg_restore` adalah perintah yang digunakan untuk mengembalikan database dari hasil backup format custom.

DBeaver menggunakan kedua tool tersebut di GUI nya, sehingga pengguna dapat melakukan backup dan restore tanpa mengetik perintah terminal.

Untuk memulai Backup:

1. Pilih basis data yang diinginkan.
2. Klik kanan pada basis data dan pilih **Tools -> Backup**
3. Kotak dialog **Dump** akan muncul. Pilih objek yang diperlukan dan klik **Next** .
4. Tab **Export configuration** akan ditampilkan. Pili checkboxes dan fields seperti pada gambar 1

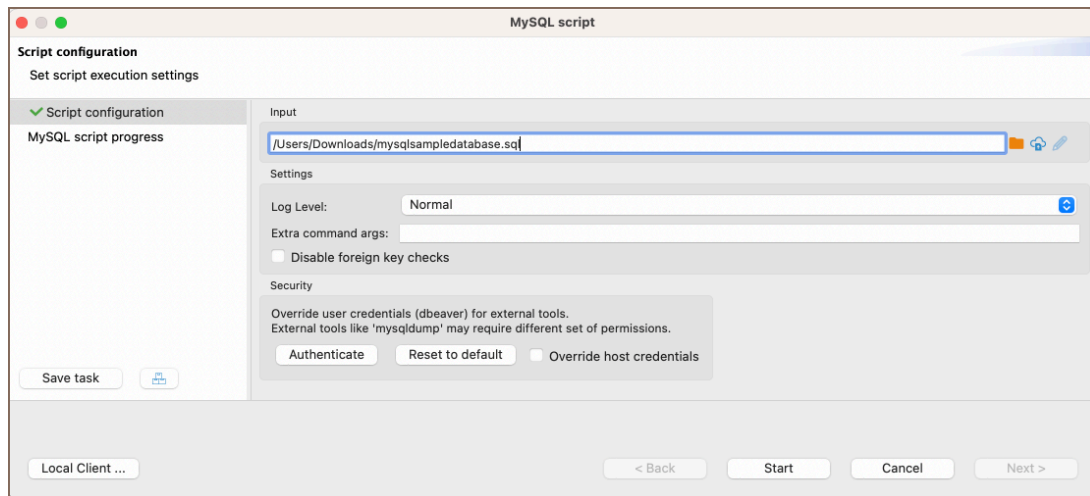


5. Setelah mengatur konfigurasi, klik **start** .
6. Setelah berhasil, pemberitahuan akan muncul berisi informasi tentang prosesnya.
7. Backup file dapat ditemukan dalam folder yang ditentukan selama langkah **Export configuration**.

## Restore – Menggunakan perintah RESTORE (MySQL)

1. Pilih basis data yang diinginkan.
2. Klik kanan pada basis data dan pilih **Tools -> Restore database**.

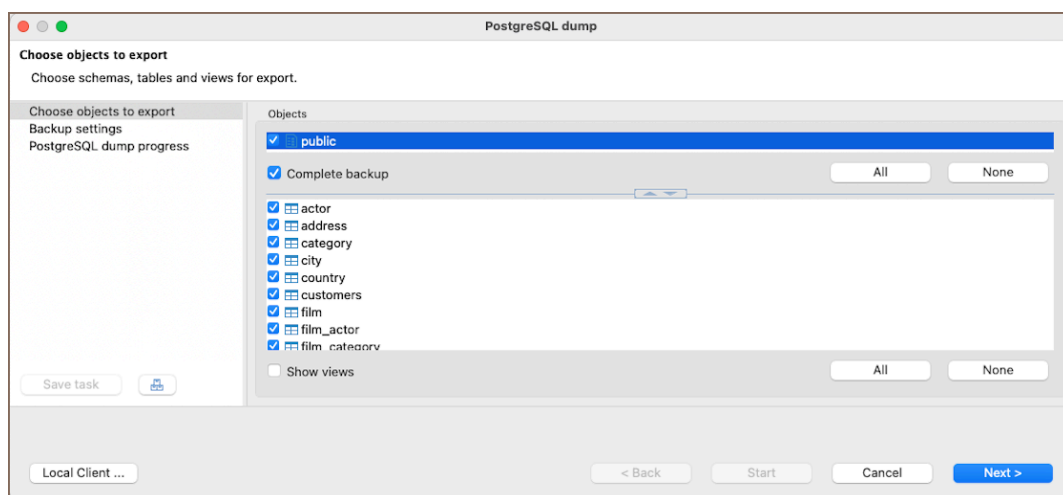
3. Kotak dialog **Restore** akan muncul. Masukkan path ke berkas SQL yang ingin dipulihkan di kolom yang ditentukan. Contoh: /path/to/file.sql



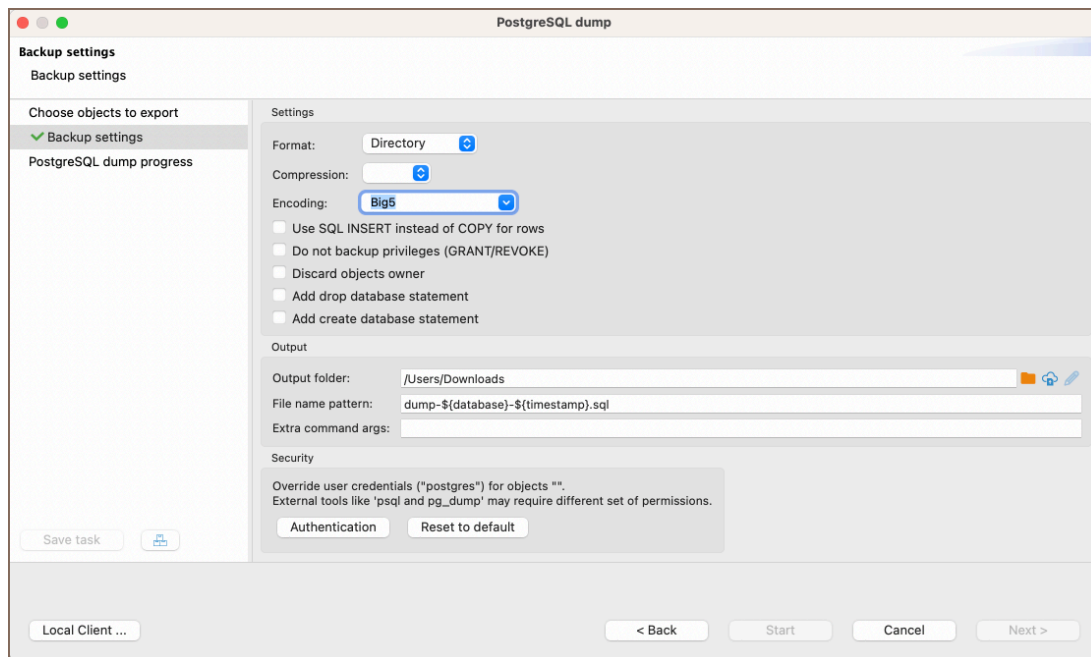
4. Setelah mengatur konfigurasi, klik **start**.
5. Setelah berhasil, pemberitahuan akan muncul berisi informasi tentang prosesnya.

## Backup – Menggunakan perintah BACKUP (PostgreSQL)

1. Pilih basis data yang diinginkan.
2. Klik kanan pada basis data dan pilih **Tools -> Backup database**.
3. Kotak dialog **Dump** akan muncul. Pilih objek yang diperlukan dan klik **Next**.



4. Tab **Pengaturan Cadangan** akan ditampilkan. Pada tahap ini akan menemukan kotak centang dan kolom berikut:



5. Setelah mengkonfigurasi pengaturan, klik **Mulai** .
6. Setelah berhasil diselesaikan, pemberitahuan akan muncul berisi informasi tentang prosesnya.
7. Berkas cadangan dapat ditemukan dalam folder yang ditentukan selama langkah **Pengaturan pencadangan** .

## Restore – Menggunakan perintah RESTORE (PostgreSQL)

1. Pilih basis data yang diinginkan.
2. Klik kanan pada basis data dan pilih **Alat** -> **Pulihkan** .

