

Restore Database & Querying, Sorting, Filtering

Virtual Internship Experience

id/x partners

Disclaimer

“Dokumen ini memiliki hak cipta. Barang siapa yang menyebarluaskan atau menduplikasi tanpa izin dari instansi terkait dapat diproses sesuai dengan ketentuan hukum yang berlaku.”



Outline

- 1** *SQL Database Restore*
- 2** *Querying, Sorting and Filtering*
- 3** *Case Study*



1 **SQL Restore Database**

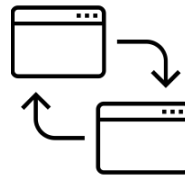
Pemulihan *Database* atau *Restore Database* adalah proses menyalin data dari file cadangan ke database untuk memulihkan data yang hilang atau rusak. Kenapa harus melakukan *Restore Database*?



DATA LOSS



CORRUPTED FILE



Testing



Upgrades



1 **SQL Restore Database**

Untuk melakukan *restore database*, berikut adalah langkah-langkah umum yang dapat dilakukan:

1. Siapkan *file backup* database yang akan di-*restore*.
2. Buka aplikasi yang digunakan untuk mengelola database (misal: phpMyAdmin, SQL Server Management Studio, dll).
3. Pilih database tujuan untuk di -*restore*, atau buat database baru jika belum ada.
4. Pilih menu "*Restore*" atau "*Import*" dan pilih *file backup* yang akan di-*restore*.
5. Ikuti proses *restore* sampai selesai.



2 Querying, Sorting and Filtering

1 SQL QUERYING

Basic Syntax:

```
SELECT column_1, column_2, ..., column_n  
FROM table_name  
WHERE conditions
```

Contoh Query:

```
SELECT first_name, last_name, salary  
FROM employee  
WHERE salary > 75000;
```





2 Querying, Sorting and Filtering

2 SORTING

Basic Syntax:

```
SELECT column_1, column_2, ..., column_n  
FROM table_name  
ORDER BY column_name ASC/DESC;
```

Contoh Query:

```
SELECT first_name, last_name, salary  
FROM employee  
ORDER BY salary ASC;
```





2 Querying, Sorting and Filtering

3 FILTERING

Basic Syntax (IN operator):

```
SELECT column_1, column_2, ..., column_n  
FROM table_name  
WHERE column_name IN (value1, value2, ..., value_n);
```

Contoh Query (IN operator):

```
SELECT first_name, last_name, department  
FROM employee  
WHERE department IN ('Marketing', 'Sales');
```





2 Querying, Sorting and Filtering

3 FILTERING

Basic Syntax (BETWEEN operator):

```
SELECT column_1, column_2, ..., column_n  
FROM table_name  
WHERE column_name BETWEEN value1 AND value2;
```

Contoh Query (BETWEEN operator):

```
SELECT first_name, last_name, salary  
FROM employee  
WHERE salary BETWEEN 75000 AND 100000;
```





2 Querying, Sorting and Filtering

3 FILTERING

Basic Syntax (LIKE operator):

```
SELECT column1, column2, ...  
FROM table_name  
WHERE column_name LIKE pattern;
```

Contoh Query (LIKE operator):

```
SELECT *  
FROM employee  
WHERE first_name LIKE 'A%';
```





2 *Querying, Sorting and Filtering*

3 *FILTERING*

Tambahan:

Operator Perbandingan

- Menggunakan operator sama dengan (=).
- Menggunakan operator tidak sama (<>).
- Menggunakan operator kurang dari (<) dan lebih besar dari (>).
- Menggunakan operator kurang dari atau sama dengan (<=) dan lebih besar dari atau sama dengan (>=).

Operator Logika

- Menggunakan operator AND
- Menggunakan operator OR
- Menggunakan operator NOT



3 **Case Study: Problem Statement**

Bagaimana cara menampilkan nama, jenis kelamin, dan usia dari semua pelanggan yang berusia di atas 25 tahun dan mengurutkan data berdasarkan usia dari yang terkecil?

id_pelanggan	nama	jenis_kelamin	usia
1	John Doe	Laki-laki	30
2	Jane Doe	Perempuan	28
3	Bob Smith	Laki-laki	31
4	Alice Johnson	Perempuan	22

Gambar 1 Tabel *Customer*

3 Case Study: *Solution*

SQL Query:

```
SELECT nama, jenis_kelamin, usia  
FROM customer  
WHERE usia > 25  
ORDER BY usia ASC;
```

nama	jenis_kelamin	usia
Jane Doe	Perempuan	28
John Doe	Laki-laki	30
Bob Smith	Laki-laki	31

Gambar 2 Tabel Hasil Query



Thank You!



id/x partners