

Rangkuman Materi Basis Data

1. Pengertian Basis Data

Basis data (database) adalah kumpulan data yang terorganisir yang disimpan secara elektronik dalam sebuah sistem komputer. Basis data memungkinkan pengelolaan data yang efisien dan terstruktur, sehingga memudahkan penyimpanan, akses, dan manipulasi data.

2. Model Basis Data

- **Model Relasional:** Menggunakan tabel (relasi) untuk menyimpan data. Setiap tabel terdiri dari baris (rekord) dan kolom (atribut).
 - Contoh: MySQL, PostgreSQL
- **Model NoSQL:** Dirancang untuk menyimpan data yang tidak terstruktur atau semi-terstruktur.
 - Contoh: MongoDB (dokumen), Cassandra (wide-column), Redis (key-value)

3. SQL (Structured Query Language)

SQL adalah bahasa standar untuk mengelola dan memanipulasi basis data relasional. Beberapa perintah dasar dalam SQL meliputi:

3.1. DDL (Data Definition Language)

Digunakan untuk mendefinisikan struktur basis data.

```
`CREATE TABLE siswa (  
  id INT PRIMARY KEY,  
  nama VARCHAR(100),  
  jurusan VARCHAR(50)  
);  
  
ALTER TABLE siswa ADD COLUMN umur INT;  
DROP TABLE siswa;`
```

- **Penjelasan:**
 - `CREATE TABLE` : Membuat tabel baru.
 - `ALTER TABLE` : Mengubah struktur tabel yang ada.
 - `DROP TABLE` : Menghapus tabel dari basis data.

3.2. DML (Data Manipulation Language)

Digunakan untuk manipulasi data dalam tabel.

```
INSERT INTO Mahasiswa (id, nama, jurusan, umur) VALUES (1, 'Hasiza', 'RPL', 17);

SELECT * FROM Mahasiswa WHERE jurusan = 'RPL';

UPDATE siswa SET umur = 22 WHERE id = 1;

DELETE FROM siswa WHERE id = 1;
```

- **Penjelasan:**

- **INSERT INTO** : Menambahkan data baru ke dalam tabel.
- **SELECT** : Mengambil data dari tabel.
- **UPDATE** : Memperbarui data yang ada dalam tabel.
- **DELETE** : Menghapus data dari tabel.

5. ERD (Entity-Relationship Diagram)

ERD adalah alat visual yang digunakan untuk memodelkan data dan hubungan antar entitas dalam basis data. Komponen utama ERD meliputi:

- **Entitas**: Objek yang memiliki keberadaan nyata atau konseptual.
- **Atribut**: Properti atau karakteristik dari entitas.
- **Relasi**: Hubungan antara dua atau lebih entitas.

6. SELECT Statement

Perintah SELECT digunakan untuk mengambil data dari tabel dalam basis data.

```
SELECT kolom1, kolom2, ...
FROM nama_tabel;
```

- **Penjelasan:**

- **SELECT** : Memilih kolom-kolom tertentu dari tabel.
- **kolom1, kolom2, ...** : Nama kolom yang ingin ditampilkan.
- **FROM** : Menunjukkan tabel mana yang akan digunakan.

7. WHERE Clause

WHERE digunakan untuk menyaring baris yang sesuai dengan kondisi tertentu.

```
SELECT *  
FROM Mahasiswa  
WHERE jurusan = 'Informatika';
```

- **Penjelasan:**

- Perintah ini mengambil semua data dari tabel `Mahasiswa` di mana nilai kolom `jurusan` adalah 'Informatika'.

8. Operator Logika (AND, OR, NOT)

Operator logika digunakan untuk menggabungkan atau membalikkan kondisi dalam WHERE.

```
SELECT *  
FROM Mahasiswa  
WHERE jurusan = 'Informatika' AND umur > 20;
```

- **Penjelasan:**

- `AND` : Mengembalikan baris yang memenuhi kedua kondisi.
- `OR` : Mengembalikan baris yang memenuhi salah satu kondisi.
- `NOT` : Membalikkan hasil kondisi (misal: NOT LIKE).

9. Operator Comparison (BETWEEN, NOT BETWEEN, <=, >=)

Operator perbandingan digunakan untuk membandingkan nilai.

```
SELECT *  
FROM Mahasiswa  
WHERE umur BETWEEN 18 AND 25;
```

- **Penjelasan:**

- `BETWEEN` : Mengembalikan baris jika nilai di antara dua nilai tertentu.
- `NOT BETWEEN` : Membalikkan hasil dari `BETWEEN`.
- `<=` : Kurang dari atau sama dengan.
- `>=` : Lebih besar dari atau sama dengan.

10. LIKE dan NOT LIKE

LIKE digunakan untuk pencarian pola berdasarkan string.

```
SELECT *  
FROM Mahasiswa  
WHERE nama LIKE 'B%';
```

- **Penjelasan:**

- `LIKE 'B%'` : Mengembalikan baris dengan `nama` yang dimulai dengan 'B'.
- `%` : Wildcard untuk nol atau lebih karakter.

11. ORDER BY

ORDER BY digunakan untuk mengurutkan hasil query.

```
SELECT *  
FROM Mahasiswa  
ORDER BY nama ASC;
```

- **Penjelasan:**

- `ORDER BY nama ASC` : Mengurutkan hasil berdasarkan kolom `nama` secara ascending (ASC) atau descending (DESC).

Kesimpulan

Basis data adalah komponen penting dalam sistem informasi yang memungkinkan pengelolaan data yang terstruktur dan efisien. Model basis data relasional dan NoSQL memiliki keunggulan masing-masing sesuai dengan kebutuhan spesifik. Penggunaan SQL dalam basis data relasional sangat penting untuk manipulasi dan pengelolaan data. Normalisasi dan ERD membantu dalam perancangan basis data yang efektif dan efisien. Menguasai konsep dasar SQL seperti SELECT, WHERE, operator logika (AND, OR, NOT), operator perbandingan (BETWEEN, NOT BETWEEN, <=, >=), LIKE, NOT LIKE, dan ORDER BY adalah langkah penting untuk memulai memahami cara mengambil, menyaring, dan mengurutkan data dalam basis data relasional. Dengan latihan dan pemahaman yang lebih dalam, Anda dapat membangun query SQL yang lebih kompleks dan efisien.