

# Materi SQL DML (Update)

## Pengertian

Perintah UPDATE digunakan untuk memperbarui nilai kolom dalam satu atau lebih baris data dalam sebuah tabel.

Format umum dari perintah INSERT di SQL adalah sebagai berikut:

```
UPDATE nama_tabel  
SET kolom1 = nilai1, kolom2 = nilai2, kolom3 = nilai3, ...  
WHERE kondisi;
```

## Penjelasan

- **UPDATE** : Pada bagian ini menunjukkan bahwa kita akan menggunakan perintah untuk memperbarui data dalam tabel tertentu.
- **nama\_tabel** : Pada bagian ini kita menentukan nama tabel dimana kita ingin memperbarui data.
- **SET** : Bagian ini menunjukan bahwa kita akan menggunakan perintah untuk menentukan kolom mana yang akan diperbarui dan nilainya.
- **kolom1 = nilai1, kolom2 = nilai2, kolom3 = nilai3, ...** : Bagian ini adalah daftar kolom yang akan diperbarui beserta nilai baru yang ingin diberikan.
- **WHERE** : Bagian ini digunakan untuk memfilter baris mana yang akan diperbarui berdasarkan kondisi yang diberikan. Jika kondisi tidak diberikan, maka semua baris dalam tabel yang sesuai akan diperbarui.

## Contoh Penggunaan

### Contoh penggunaan perintah UPDATE SQL :

Misalkan kita memiliki tabel "pegawai" dengan kolom-kolom "id", "nama", dan "usia". Kita ingin memperbarui usia pegawai dengan ID 1 menjadi 35. Perintah UPDATE SQL-nya akan terlihat seperti ini:

```
UPDATE pegawai
SET usia = 35
WHERE id = 1;
```

Dengan menggunakan perintah UPDATE SQL di atas, nilai usia pada baris data pegawai dengan ID 1 akan diperbarui menjadi 35. Perintah tersebut memperbarui kolom "usia" di dalam tabel "pegawai" hanya pada baris dengan kondisi ID sama dengan 1.

```
UPDATE pegawai
SET usia = 35;
```

Jika kondisi tidak diberikan seperti contoh diatas, maka perintah UPDATE akan memperbarui nilai kolom "usia" untuk semua baris dalam tabel "pegawai".

## Kesalahan Dalam Penggunaan Perintah Update

### 1. Kesalahan dalam Menyediakan Kondisi WHERE yang Tepat

Contoh :

```
UPDATE employees SET salary = 5000;
```

Kesalahan terjadi jika kita tidak menyediakan kondisi WHERE yang tepat. Dalam contoh ini, semua nilai gaji karyawan akan diubah tanpa memperhatikan karyawan mana yang seharusnya diperbarui.

Cara menghindarinya: Pastikan menyediakan kondisi WHERE yang tepat agar perintah UPDATE hanya mempengaruhi data yang diinginkan.

### 2. Kesalahan dalam Mengurutkan Klausula SET

Contoh :

```
UPDATE products SET price = '19.99', name = 'Product A' WHERE id = 1;
```

Kesalahan terjadi ketika kita mengurutkan klausa SET secara tidak tepat, menyebabkan nilai yang dimasukkan tidak sesuai dengan kolom yang diharapkan.

Cara menghindarinya: Pastikan mengurutkan klausa SET dengan benar, sesuai dengan kolom yang akan diperbarui.

### **3. Kesalahan dalam Menyediakan Nilai yang Tepat**

Contoh :

```
UPDATE customers SET age = '30' WHERE id = 1;
```

Kesalahan terjadi jika kita menggunakan tipe data yang tidak sesuai, seperti memasukkan nilai string ke dalam kolom yang seharusnya menerima tipe data numerik.

Cara menghindarinya: Pastikan nilai yang dimasukkan sesuai dengan tipe data yang ditentukan dalam skema tabel.

### **4. Kesalahan dalam Mengabaikan Kondisi WHERE**

Contoh :

```
UPDATE orders SET status = 'Completed';
```

Kesalahan terjadi jika kita tidak menyertakan kondisi WHERE, sehingga perintah UPDATE akan mempengaruhi semua data dalam tabel.

Cara menghindarinya: Selalu menyertakan kondisi WHERE yang tepat untuk memastikan perintah UPDATE hanya mempengaruhi data yang diinginkan.

### **5. Kesalahan dalam Mengubah Data yang Tidak Diperlukan**

Contoh :

```
UPDATE employees SET salary = salary + 500;
```

Kesalahan terjadi jika kita tidak memeriksa nilai saat memperbarui data. Dalam contoh ini, gaji semua karyawan akan dinaikkan, termasuk yang sudah memiliki gaji lebih tinggi.

Cara menghindarinya: Pastikan memeriksa kondisi atau batasan sebelum mengubah data, sehingga hanya data yang memenuhi kriteria yang diperbarui.

## Kesimpulan

Perintah UPDATE adalah operasi penting dalam manipulasi data pada basis data. Penting bagi pengguna untuk menghindari kesal