

# ALTER Statement

## Definisi ALTER Statement

ALTER Statement adalah perintah DDL yang digunakan untuk **memodifikasi struktur tabel** yang telah ada di dalam basis data. Melalui ALTER Statement, kita bisa **menambahkan, mengubah,** atau **menghapus kolom** dan **constraint** dalam tabel. Penggunaan ALTER Statement sangat penting dalam pengembangan basis data, karena struktur tabel seringkali perlu diubah seiring dengan perubahan kebutuhan dan pertumbuhan aplikasi.

## Contoh Alter Statement

### 1. Menambahkan Kolom Baru

```
ALTER TABLE nama_tabel  
  
ADD COLUMN nama_kolom tipe_data;
```

Query di atas digunakan untuk menambahkan kolom baru ke dalam tabel yang sudah ada. Kamu perlu mengganti nama\_tabel dengan nama tabel yang ingin kamu ubah, nama\_kolom dengan nama kolom yang ingin kamu tambahkan, dan tipe\_data dengan tipe data untuk kolom baru tersebut. Contoh :

```
ALTER TABLE anggota  
  
ADD COLUMN email VARCHAR(255);
```

### 2. Menghapus Kolom

```
ALTER TABLE nama_tabel  
  
DROP COLUMN nama_kolom;
```

Query di atas digunakan untuk menghapus kolom dari tabel yang sudah ada. Kamu perlu mengganti nama\_tabel dengan nama tabel yang ingin kamu ubah dan nama\_kolom dengan nama kolom yang ingin kamu hapus.



```
ALTER TABLE anggota
```

```
DROP COLUMN email;
```

### 3. Mengubah Tipe Data Kolom

```
ALTER TABLE nama_tabel
```

```
MODIFY nama_kolom tipe_data_baru;
```

Query di atas digunakan untuk mengubah tipe data kolom dalam tabel yang sudah ada. Kamu perlu mengganti nama\_tabel dengan nama tabel yang ingin kamu ubah, nama\_kolom dengan nama kolom yang ingin kamu ubah tipe datanya, dan tipe\_data\_baru dengan tipe data baru yang ingin kamu terapkan pada kolom tersebut.

```
ALTER TABLE buku
```

```
MODIFY judul VARCHAR(200);
```

### 4. Mengganti Nama Kolom

```
ALTER TABLE nama_tabel
```

```
RENAME COLUMN nama_kolom_lama TO nama_kolom_baru;
```

Query di atas digunakan untuk mengganti nama kolom dalam tabel yang sudah ada. Kamu perlu mengganti nama\_tabel dengan nama tabel yang ingin kamu ubah, nama\_kolom\_lama dengan nama kolom yang sekarang, dan nama\_kolom\_baru dengan nama kolom baru yang ingin kamu gunakan.

```
ALTER TABLE anggota
```

```
RENAME COLUMN alamat TO alamat_lengkap;
```

### 5. Menambahkan Primary Key

```
ALTER TABLE nama_tabel
```

```
ADD PRIMARY KEY (nama_kolom);
```

Contoh :

```
ALTER TABLE buku
```

```
ADD PRIMARY KEY (id_buku);
```

## 6. Menghapus Primary Key

```
ALTER TABLE nama_tabel
```

```
DROP PRIMARY KEY;
```

Contoh :

```
ALTER TABLE buku
```

```
DROP PRIMARY KEY;
```

## 7. Menambahkan Foreign Key

```
ALTER TABLE nama_tabel
```

```
ADD CONSTRAINT nama_constraint FOREIGN KEY (nama_kolom)
REFERENCES tabel_target(kolom_target);
```

Query di atas digunakan untuk menambahkan **FOREIGN KEY** ke kolom dalam tabel yang sudah ada. Kamu perlu mengganti nama\_tabel dengan nama tabel yang ingin kamu ubah, nama\_constraint dengan nama untuk constraint baru ini, nama\_kolom dengan nama kolom yang ingin kamu jadikan FOREIGN KEY, tabel\_target dengan nama tabel yang kolomnya akan dijadikan referensi, dan kolom\_target dengan nama kolom di tabel target yang akan dijadikan referensi.

```
ALTER TABLE peminjaman
```

```
ADD CONSTRAINT fk_id_buku FOREIGN KEY (id_buku) REFERENCES
buku(id_buku);
```

## 8. Menghapus Foreign Key Dengan Nama Constraint

```
ALTER TABLE nama_tabel
```

```
DROP FOREIGN KEY nama_constraint;
```

Contoh :

```
ALTER TABLE peminjaman
```

```
DROP FOREIGN KEY fk_id_buku;
```

—