

# MODEL RELASIONAL

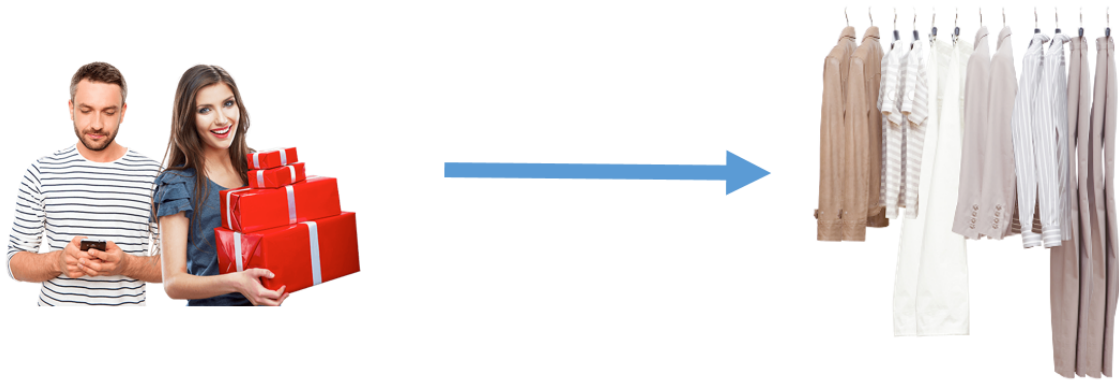
---

## OBJEKTIF :

1. Mahasiswa dapat memahami pengertian model relasional dan diagram relasional.
  2. Mahasiswa mampu menggunakan *constraint* untuk membatasi nilai pada sebuah kolom.
- 

## 2.1 PENGERTIAN MODEL RELASIONAL

**Model relasional** diperkenalkan oleh E.F.Codd. **Model relasional** sering disebut sebagai Model basis data relasional. Model relasional adalah suatu *database* yang memiliki hubungan (relasi) antara tabel satu dengan tabel yang lainnya. Model relasional merupakan bentuk tabel yang alami atau natural. Hal tersebut dapat dilihat dari fakta atau data yang sering digunakan. Contohnya adalah saat pengguna memesan suatu barang. Data tersebut akan tersimpan pada tabel pemesanan.



Tabel pemesanan merupakan tabel relasi antara tabel pengguna dengan tabel barang. Pada tabel pemesanan pasti terdapat data pengguna sebagai informasi siapakah pemesan barang tersebut. Lalu pada tabel pemesanan juga terdapat informasi dari data barang untuk menunjukkan barang apa yang dipesan oleh pengguna.

Dengan menggunakan model relasional maka struktur *database* akan terlihat lebih konsisten. Hal tersebut, dapat dilihat dengan adanya kolom *foreign key* dalam tabel relasi. *Foreign key* merupakan kunci tamu atau kunci asing pada tabel relasi, namun perlu di ketahui bahwa *foreign key* merupakan *primary key* atau kunci utama pada tabel asal. Hubungan menggunakan *foreign key* membuat tabel-tabel relasi memiliki data yang konsisten atau sama dengan tabel asal.

## 2.2 DIAGRAM RELASIONAL

Diagram Relasional adalah model yang digunakan untuk mendesain keterhubungan data pada sebuah basis data (*Database*). Biasanya sebelum membuat sebuah tabel maka akan dibuat model diagram relasional. Diagram relasional memiliki beberapa jenis relasi atau hubungan, yaitu :

- a. Satu-ke-Satu (*One-to-One*) : Setiap baris data hanya berhubungan maksimal dengan 1 baris data pada tabel lain.

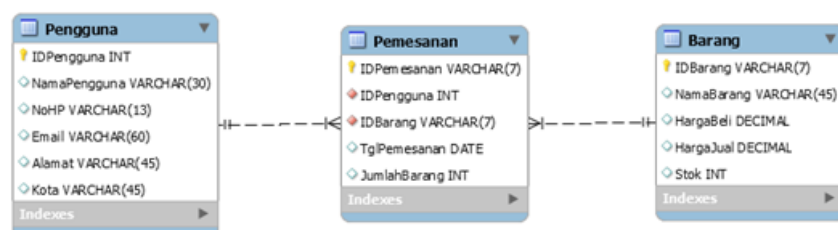
b. Satu-ke-Banyak (*One-to-Many*) : Setiap satu baris data dapat berhubungan dengan beberapa baris pada tabel lain.



c. Banyak-ke-Banyak (*Many-to-Many*) : Setiap beberapa baris data dapat berhubungan dengan beberapa baris pada tabel lain.



Berikut merupakan contoh dari diagram relasional yang dibuat melalui MySQL Workbench. Gambar di bawah menjelaskan bahwa tabel pemesanan merupakan tabel relasi. Dapat dilihat dari masing-masing tabel tersebut memiliki *primary key*. *Primary key* dari tabel pengguna adalah kolom IDPengguna, *primary key* dari tabel barang adalah kolom IDBarang, dan *primary key* dari tabel pemesanan adalah kolom IDPemesanan. Tabel pemesanan memiliki *foreign key* atau kunci tamu dari tabel pengguna dan tabel barang. *Foreign key* tersebut adalah IDPengguna dan IDBarang. *Foreign key* membuat tabel pemesanan memiliki hubungan data dari tabel pengguna dan tabel barang. Diagram di bawah ini memiliki hubungan atau relasi *one-to-many* antara tabel pengguna dengan tabel pemesanan. Arti dari hubungan tersebut adalah satu pengguna dapat berhubungan dengan beberapa data pada tabel pemesanan. Hubungan *one-to-many* juga berlaku untuk tabel barang dan pemesanan karena satu barang dapat berhubungan dengan beberapa data pada tabel pemesanan.



## 2.3 CONSTRAINT

*Constraint* adalah sebuah batasan pada nilai-nilai yang diperbolehkan untuk diisikan pada sebuah kolom. Dengan menggunakan *constraint* dipastikan bahwa data-data tersebut sesuai dengan batasan yang ditentukan pada kolom. Terdapat beberapa tipe pada *constraint* yaitu :

### 2.3.1 PRIMARY KEY

*Primary key* atau biasa disebut dengan kunci utama, merupakan kunci yang membatasi pengisian data pada sebuah tabel agar tidak duplikat atau dapat dibedakan antara baris satu dengan yang lainnya. *Primary key* juga dapat dikatakan sebagai kolom yang secara unik mengidentifikasi setiap baris dalam tabel. Contoh kolom yang biasa di deklarasikan menggunakan *primary key* adalah id, NPM, atau NIK. Penentuan *primary key constraint* dapat dilakukan saat pembuatan sebuah table, berikut bentuk umum dari pembuatan tabel yang memiliki *primary key* :

```
CREATE TABLE table_name (primarykey_column data_type PRIMARY KEY, column_2 data_type, column_3 data_type,...);
```

Penjelasan :

- `table_name` : Nama tabel
- `primarykey_column` : Merupakan nama kolom yang akan di deklarasikan menjadi *primary key*
- `data_type` : Merupakan tipe data dari kolom
- `column_2, column_3, ..` : Merupakan nama kolom yang lainnya.

Berikut merupakan contoh pembuatan tabel Pengguna dengan IDPengguna sebagai *primary key* dan tabel Barang dengan IDBarang sebagai *primary key* :

- ```
CREATE TABLE Pengguna (IDPengguna INT AUTO_INCREMENT PRIMARY KEY, NamaPengguna VARCHAR(30), NoHP VARCHAR(13), Email VARCHAR(60), Alamat VARCHAR(40), Kota VARCHAR(40));
```
- ```
CREATE TABLE Barang (IDBarang VARCHAR(7) PRIMARY KEY, NamaBarang VARCHAR(30), HargaBeli DECIMAL, HargaJual DECIMAL, Stok INT);
```

### 2.3.2 FOREIGN KEY

*Foreign key* merupakan kunci asing yang berfungsi untuk menghubungkan dua tabel. *Foreign key* merupakan kunci utama pada tabel asal, dan menjadi kunci asing pada tabel lain. Penentuan *foreign key* dapat dilakukan saat pembuatan tabel, berikut bentuk umum pembuatan tabel menggunakan *foreign key constraint* :

```
CREATE TABLE table_name (primarykey_column data_type PRIMARY KEY, column_2 data_type, column_3 data_type,..., FOREIGN KEY(foreignkey_column) REFERENCES originaltable_name(foreignkey_column);
```

Penjelasan :

- `foreignkey_column` : Merupakan nama kolom yang ditentukan sebagai *foreign key*
- `originaltable_name` : Merupakan nama tabel asal dari kolom *foreign key*

Berikut merupakan contoh pembuatan tabel Pemesanan dengan *foreign key*:

- ```
CREATE TABLE Pemesanan (IDPemesanan VARCHAR(7) PRIMARY KEY, IDPengguna INT, IDBarang VARCHAR(7), tglPemesanan DATE, JumlahBarang INT, FOREIGN KEY(IDPengguna) REFERENCES Pengguna(IDPengguna), FOREIGN KEY(IDBarang) REFERENCES Barang(IDBarang));
```

Terlihat dari query diatas bahwa *foreign key* pada tabel Pemesanan adalah IDPengguna yang berasal dari tabel Pengguna dan IDBarang yang berasal dari tabel Barang. Struktur yang akan terlihat akan seperti tampilan gambar di bawah ini :

## Tabel Pengguna

|  | IDPengguna | NamaPengguna | NoHP | Email | Alamat | Kota |
|--|------------|--------------|------|-------|--------|------|
|--|------------|--------------|------|-------|--------|------|

## Tabel Pemesanan

|  | IDPemesanan | IDPengguna | IDBarang | TglPemesanan | JumlahBarang |
|--|-------------|------------|----------|--------------|--------------|
|--|-------------|------------|----------|--------------|--------------|

## Tabel Barang

|  | IDBarang | NamaBarang | HargaBeli | HargaJual | Stok |
|--|----------|------------|-----------|-----------|------|
|--|----------|------------|-----------|-----------|------|

### 2.3.3 UNIQUE

*Unique Constraint* merupakan kunci unik yang membatasi sebuah nilai pada kolom. *Constraint* ini biasanya digunakan pada kolom email, alamat, atau nomor telepon. *Unique constraint* digunakan pada kolom-kolom tersebut karena kolom tersebut memiliki data yang pasti berbeda dan unik. *Unique constraint* dapat bernilai kosong atau NULL. *Constraint* ini berbeda dengan *primary key*, jika *primary key* merupakan kolom yang mewakili satu baris atau record tetapi *unique constraint* hanya membatasi untuk pengisian data pada kolom yang ditentukan. Berikut bentuk umum yang digunakan untuk menentukan kolom yang unik :

```
CREATE TABLE table_name (... , column_name data_type UNIQUE,...);
```

Penjelasan :

- **column\_name** : Merupakan nama kolom yang akan ditentukan sebagai *unique constraint*
- **data\_type** : Merupakan tipe data dari kolom

Berikut contoh pembuatan tabel pengguna\_bogor dengan kolom NoHP dan Email ditentukan sebagai *unique constraint* :

- ```
CREATE TABLE Pengguna_Bogor (IDPengguna INT PRIMARY KEY, NamaPengguna VARCHAR(50), NoHP VARCHAR(13) UNIQUE, Email VARCHAR(90) UNIQUE, Alamat VARCHAR(90), Kota VARCHAR(30) CHECK(kota='Bogor'));
```

### 2.3.4 CHECK

*Check constraint* adalah batasan untuk menyeleksi nilai yang sudah dideklarasikan sebelumnya. *Check constraint* ini akan menyeleksi apakah data yang dimasukkan ke dalam kolom tersebut sesuai dengan kondisi di dalam *check constraint*. Contohnya pada tabel pengguna hanya diinginkan pengguna yang berada di kota Bogor maka pada pembuatan tabel dapat di sesuaikan kota='Bogor'. Berikut bentuk umum pembuatan tabel yang kolomnya ditentukan sebagai *check constraint*:

```
CREATE TABLE table_name (... , column_name data_type CHECK,...);
```

Penjelasan :

- `column_name` : Merupakan nama kolom yang akan ditentukan menggunakan *check constraint*
- `data_type` : Merupakan tipe data dari kolom

Berikut contoh pembuatan tabel pengguna\_bogor menggunakan *check constraint* pada kolom kota :

- ```
CREATE TABLE Pengguna_Bogor (IDPengguna INT PRIMARY KEY, NamaPengguna VARCHAR(50), NoHP VARCHAR(13) UNIQUE, Email VARCHAR(90) UNIQUE, Alamat VARCHAR(90), Kota VARCHAR(30) CHECK(kota='Bogor'));
```

## REFERENSI

- [1] Fathansyah. 2018. *Basis Data Revisi Ketiga*. Bandung: Bandung Informatika.
- [2] Udemy. 2020. "SQL for Beginner : Learn SQL using MySQL and Database Design" , Diakses dari <https://www.udemy.com/course/sql-for-beginners-course/> pada 31 Agustus 2020.
- [3] Mysqltutorial. "MySQL Primary key", Diakses dari <https://www.mysqltutorial.org/mysql-primary-key/> pada 31 Agustus 2020.
- [4] Mysqltutorial. "MySQL Foreign key",Diakses dari <https://www.mysqltutorial.org/mysql-foreign-key/> pada 31 Agustus 2020.
- [5] Mysqltutorial. "MySQL Unique Constraint", Diakses dari <https://www.mysqltutorial.org/mysql-unique-constraint/> pada 31 Agustus 2020.
- [6] Mysqltutorial. "MySQL Check Constraint" , Diakses dari <https://www.mysqltutorial.org/mysql-check-constraint/> pada 31 Agustus 2020.