

# Materi VIII

Nama : Roy Agus Martin Marbun

Kelas : XII – RPL

Mata Pelajaran : Basis Data

---

## BAB II

### Perintah SQL dan Basis Data pada RDBMS

#### A. Key (Kunci)

Tabel dalam database secara sederhana bisa diartikan sebagai tempat untuk menyimpan data di dalam *database* dari sebuah informasi. Tabel itu terdiri dari *field* dan *record*. Tanpa tabel maka *database* tidak bisa menyimpan, memanggil dan mengolah data-data yang dibutuhkan. Terkadang tabel yang dibuat lebih dari satu dan tabel yang satu berelasi dengan tabel yang lain. Untuk melakukan relasi dibutuhkan yang namanya Kunci. Kunci adalah satu gabungan dari beberapa atribut yang dapat membedakan semua basis data (*row*) dalam tabel secara unik. Kunci di dalam database berfungsi sebagai suatu cara untuk mengidentifikasi dan menghubungkan satu tabel dengan tabel yang lain.

#### B. Jenis-jenis Key

*Key* atau kunci yang digunakan memiliki fungsi dan kegunaan masing-masing dalam mengidentifikasi yang berbeda-beda. Berikut adalah macam-macam *key* pada sebuah tabel :

##### a. *Primary Key*

*Primary Key* merupakan kunci utama dalam suatu tabel di dalam *database*. Keberadaannya memang sangat diperlukan sebagai identitas yang mewakili seluruh isi dari sebuah tabel. Dengan adanya *primary key* maka memudahkan dalam proses pengolahan data di dalam tabel. *Primary key* juga berperan penting ketika ingin merelasikan dua tabel atau lebih. Ketika merelasikan sebuah tabel, maka atribut yang menjadi perwakilan untuk masuk ke dalam tabel lain adalah atribut yang memiliki *primary key*. Selain itu, syarat utama dari *primary key* adalah tidak boleh sama. Untuk lebih jelasnya berikut adalah syarat=syarat dari *primary key*.

- Hanya ada satu *primary key* dalam satu tabel dan tidak boleh lebih.
- Kolom yang dijadikan *primary key* harus bersifat *NOT NULL* yang artinya tidak boleh kosong.
- Data yang dimasukkan ke kolom yang di jadikan *primary key* tidak boleh sama.
- Tidak boleh menggunakan tipe data *BLOB* untuk kolom yang dijadikn *primary key*.

b. *Foreign Key*

*Foreign key* disebut juga kunci asing atau kunci tamu yaitu kunci utama suatu tabel yang berada pada tabel lain sebagai tanda pereliasian antar tabel. Disebut kunci tamu karena memang menjadi tamu pada tabel lain. Syarat menjadi *foreign key* adalah ia harus merupakan kunci utama dari suatu tabel yang direlasikan. Tanpa *foreign key*, maka tidak ada relasi yang sebenarnya.

c. *Alternate Key*

*Alternate Key* merupakan *key* yang tidak dipilih. Jadi jika diibaratkan dalam sebuah pemilihan. Misalnya ada 2 kandidat yang akan menjadi kunci utama, setelah di pertimbangkan maka terpilihlah satu untuk menjadi kunci utama. Kandidat yang tidak terpilih maka kandidat tersebut yang disebut dengan *alternate key*.

d. *Candidate Key*

*Candidate Key* adalah kunci yang mengidentifikasi suatu kejadian spesial pada tabel. Syarat untuk menjadi *candidate key* adalah harus unik dan tidak boleh memiliki kesamaan data.

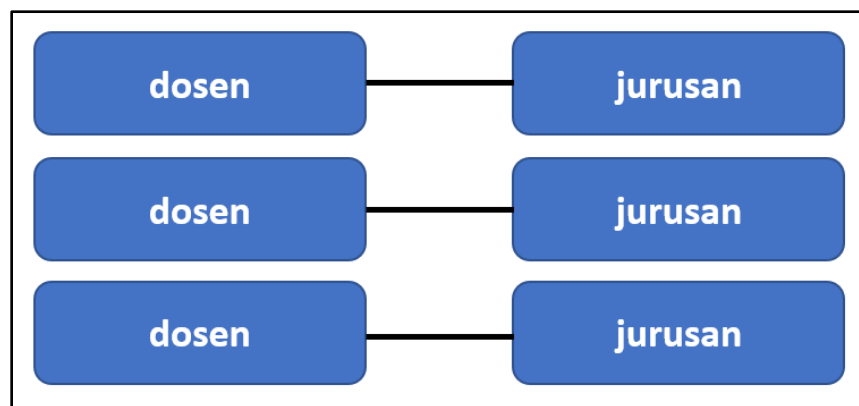
### C. Relasi dalam Basis Data

Relasi adalah hubungan antara tabel yang mempresentasikan hubungan antar objek di dunia nyata. Relasi merupakan hubungan yang terjadi pada suatu tabel dengan lainnya yang merepresentasikan hubungan antar objek di dunia nyata dan berfungsi untuk mengatur operasi suatu *database*.

Hubungan yang dapat dibentuk mencakup 3 macam hubungan yaitu :

a. *One-To-One*

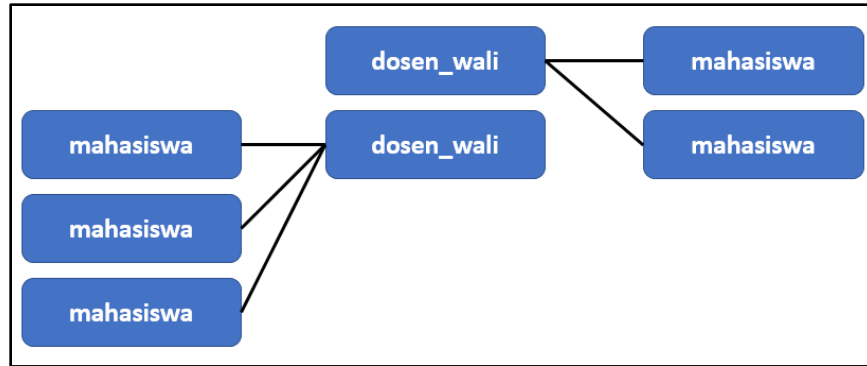
Relasi *One to One* adalah relasi yang mana setiap satu baris data pada tabel pertama hanya berhubungan dengan satu baris pada tabel kedua. Berikut adalah contoh relasi *One-to-One* dalam visualisasi gambar.



Bentuk Relasi *One-To-One*

b. *One-To-Many*

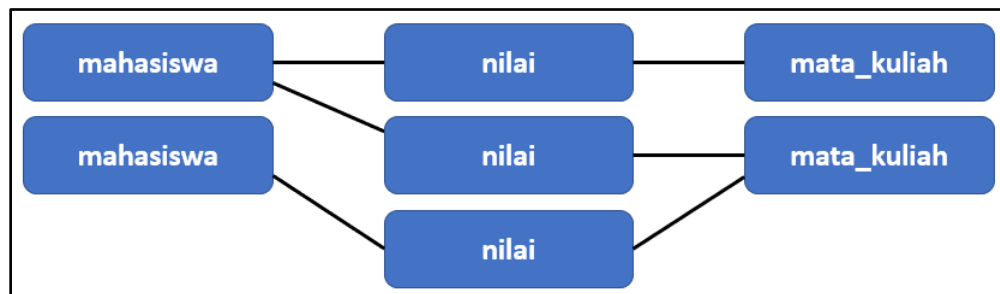
Relasi *One to Many* adalah relasi yang mana setiap satu baris data pada tabel pertama berhubungan dengan lebih dari satu baris pada tabel kedua. Berikut adalah contoh relasi *One-To-Many* dalam visualisasi gambar.



Bentuk Relasi *One-To-Many*

c. *Many-To-Many*

Relasi *Many to Many* adalah relasi yang mana setiap lebih dari satu baris data dari tabel pertama berhubungan dengan lebih dari satu baris data pada tabel kedua. Artinya, kedua tabel masing-masing dapat mengakses banyak data dari tabel yang direlasikan. Dalam hal ini, relasi *Many to Many* akan menghasilkan tabel ketiga sebagai perantara tabel pertama dan kedua sebagai tempat untuk menyimpan foreign key dari masing-masing tabel. Berikut adalah contoh relasi *Many-To-Many* dalam visualisasi gambar.



Bentuk Relasi *Many-To-Many*