

# Type / Jenis Join di MySQL

Private Training Web Development

9/14/2014

[www.phpmu.com](http://www.phpmu.com)

## INNER JOIN

Tipe join ini akan mengambil semua row dari table asal dan table tujuan dengan kondisi nilai key yang terkait saja - jika ada, dan jika tidak maka row tersebut tidak akan muncul.

### INNER JOIN Antar Table dengan Kondisi ("ms\_cabang" dan "ms\_kota" dengan key "kode\_kota")

```
SELECT * FROM ms_cabang INNER JOIN ms_kota ON ms_cabang.kode_kota = ms_kota.kode_kota
```

Table sumber adalah ms\_cabang (left) mencari referensi row lain dari table ms\_kota (right) dengan kondisi nilai **kode\_kota** diantara kedua table tersebut sama. Kondisi ini menggunakan keyword **ON**.

Keterangan :

- untuk row pertama kita memiliki kode\_kota dengan nilai "KOTA-003", ini akan dicari referensinya ke table ms\_kota untuk nilai yang sama dan kita dapatkan row dengan nilai nama\_kota "Lhokseumawe" adalah padanannya.
- untuk row kedua kita memiliki kode\_kota dengan nilai "KOTA-083", ini akan dicari referensinya ke table ms\_kota untuk nilai yang sama dan kita dapatkan row dengan nilai nama\_kota "Bau-bau" adalah padanannya.
- demikian seterusnya.

Sekarang mari kita coba hapus referensi untuk "Lhokseumawe" dari table ms\_kota :

```
DELETE FROM ms_kota WHERE nama_kota = 'Lhokseumawe';
```

Setelah itu coba jalankan kembali perintah JOIN di atas, Anda akan mendapatkan hasil JOIN tanpa referensi row "Lhokseumawe". Jadi INNER JOIN dengan kondisi mengharuskan row dari tiap table memiliki nilai yang sama untuk column referensinya (dalam hal ini **kode\_kota**).

Kembalikan lagi row referensi yang kita hapus tadi dengan perintah INSERT berikut ini :

```
INSERT INTO `ms_kota` VALUES ('KOTA-003','Lhokseumawe','P33')
```

### INNER JOIN Antar Table Tanpa Kondisi ("ms\_cabang" dengan "ms\_kota")

```
SELECT * FROM ms_cabang INNER JOIN ms_kota
```

Hasilnya adalah untuk **tiap row** dari ms\_cabang akan dikombinasikan dengan **semua row** dari ms\_kota - contoh hasilnya terlihat seperti pada gambar di bawah ini.

### INNER JOIN antar table "ms\_cabang", "ms\_kota" dan "ms\_propinsi"

```
SELECT ms_cabang.nama_cabang,  
       ms_kota.nama_kota,  
       ms_propinsi.nama_propinsi  
FROM ms_cabang  
INNER JOIN ms_kota ON ms_cabang.kode_kota = ms_kota.kode_kota  
INNER JOIN ms_propinsi ON ms_kota.kode_propinsi = ms_propinsi.kode_propinsi
```

Hasil eksekusi terlihat seperti gambar berikut. Dengan penggabungan ketiga table ini sekarang kita dapatkan setiap cabang memiliki informasi nama kota dan nama propinsi dimana cabang tersebut berada.

### Implisit INNER JOIN dengan Koma

INNER JOIN antar table secara implisit dapat menggunakan daftar table yang dipisah dengan tanda koma (,). Pengkondisian menggunakan klausa `where`.

```
SELECT ms_cabang.nama_cabang,  
       ms_kota.nama_kota ,  
       ms_propinsi.nama_propinsi  
FROM   ms_cabang  
       , ms_kota  
       , ms_propinsi  
WHERE  ms_cabang.kode_kota = ms_kota.kode_kota  
AND    ms_kota.kode_propinsi = ms_propinsi.kode_propinsi
```

## **CROSS JOIN**

CROSS JOIN identik dengan INNER JOIN pada MySQL 5.0. Pembahasannya sama dengan INNER JOIN sehingga tidak diulangi lagi disini.

Contoh Penggunaan :

```
SELECT ms_cabang.nama_cabang,  
       ms_kota.nama_kota,  
       ms_propinsi.nama_propinsi  
FROM  
       ms_cabang  
CROSS JOIN  
       ms_kota ON ms_cabang.kode_kota = ms_kota.kode_kota  
CROSS JOIN  
       ms_propinsi ON ms_kota.kode_propinsi = ms_propinsi.kode_propinsi
```

## OUTER JOIN

**OUTER JOIN** merupakan tipe join yang mencari referensi data dari suatu table sumber ke table lain dengan tidak menghilangkan data sumber apabila referensi tidak ditemukan.

Untuk menggunakan tipe OUTER JOIN maka perlu memperhatikan beberapa hal berikut :

- perlu dibedakan antara table sumber dan table referensi, ini ditentukan dengan cara menspesifikasikan kedudukan table sumber apakah di kiri (**LEFT**) atau di kanan (**RIGHT**).
- jika tidak ada data dari table referensi yang cocok dengan kondisi join maka hanya data dari table sumber yang ditampilkan tetapi kolom-kolom table referensi akan berisi null.

Contoh Penggunaan :

- Hapus data master "Lhokseumawe" dari table **ms\_kota**

```
DELETE FROM ms_kota WHERE nama_kota = 'Lhokseumawe'
```

- Lakukan join seperti perintah berikut ini, dan perhatikan hasilnya seperti pada gambar.

```
SELECT ms_cabang.*,  
       ms_kota.nama_kota  
FROM ms_cabang  
LEFT OUTER JOIN ms_kota  
ON ms_cabang.kode_kota = ms_kota.kode_kota
```

Tambahkan kembali data "Lhokseumawe" ke table **ms\_kota**

```
INSERT INTO ms_kota(kode_kota, nama_kota, kode_propinsi)  
VALUES ('KOTA-003', 'Lhokseumawe', 'P33');
```

## STRAIGHT\_JOIN

**STRAIGHT\_JOIN** merupakan pengganti keyword **JOIN** pada MySQL yang digunakan untuk "memaksa" proses join table dari kiri (LEFT) ke kanan (RIGHT).

Contoh Penggunaan :

```
SELECT ms_cabang.*,  
       ms_kota.nama_kota  
FROM ms_cabang  
STRAIGHT_JOIN ms_kota  
ON ms_cabang.kode_kota = ms_kota.kode_kota
```



## **Oleh : Robby Prihandaya**

Designer Developer & Freelance, Padang , Indonesia

Robby is a student in Universitas Putra Indonesia 'YPTK' Padang and web developer for design from Padang. He founded PHPmu in june 2013 and publishes new tutorials.