# Create tabel daftar dosen dan tabel mata kuliah dengan ketentuan seperti tabel di bawah ini :

# create database universitas;

# use universitas;

# create table dosen (nip char(10), nama\_dosen char(25), no\_hp char(12), alamat char (25));

# insert into dosen (nip, nama\_dosen, no\_hp, alamat) values

# (‘0160636012’, ‘Sabrina Sari’, ‘0812349900’, ‘Pekanbaru’) ,

# (selanjutnya... );

**Tabel dosen**

|  |  |  |  |
| --- | --- | --- | --- |
| **NIP** | **Nama Dosen** | **No HP** | **Alamat** |
| 0160436012 | Sabrina Sari | 0812349900 | Pekanbaru |
| 0260432002 | Maya Ari Putri | 0812345234 | Palembang |
| 0275430005 | Susi Indriani | 0812656532 | Bogor |
| 0480432066 | Tia Santrini | 0812451177 | Padang |
| 0576431001 | M. Siddiq | 0812979005 | Jakarta |
| 0770435006 | Rubin Hadi | 0812567678 | Papua |
| 0869437003 | Mustalifah | 0812338877 | Aceh |
| 1080432007 | Arif Budiman | 0812456345 | Makasar |

**Tabel mata kuliah**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kode Mata Kuliah** | **Nama Matakuliah** | **Jumlah SKS** | **Semester** | **Nip Dosen** |
| MATDAS | Matematika Dasar | 4 | 1 | 0160436012 |
| FISDAS | Fisika Dasar | 2 | 1 | 0480432066 |
| TEKKOM | Teknik Kompilasi | 2 | 6 | 0480432066 |
| JARKOM | Jaringan Komputer | 3 | 3 | 0770435006 |
| DTBASE | Database | 4 | 4 | 0275430005 |
| SISOPR | Sistem Operasi | 2 | 4 | 0160436012 |
| MIKROP | Mikro Prosesor | 2 | 5 | 0480432066 |

# Cara Menggabungkan Tabel MySQL (INNER JOIN)

### **Cara Menggabungkan Tabel MySQL dengan query SELECT..INNER JOIN..ON**

Format dasar dari penulisan query **SELECT..INNER JOIN..ON** adalah:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *SELECT nama\_kolom\_tampil FROM nama\_tabel\_pertama INNER JOIN nama\_tabel\_kedua*  *ON nama\_kolom\_join\_tabel\_pertama = nama\_kolom\_join\_tabel\_kedua*  Syarat untuk **INNER JOIN** adalah kedua tabel ***harus memiliki sebuah kolom dengan nilai yang sama*** yang akan kita gunakan dalam proses **JOIN**. Dalam contoh kita, kolom tersebut adalah *kolom NIP* dari tabel *nama\_dosen*, dan kolom *NIP\_dosen* dari tabel *mata\_kuliah*. Perhatikan walaupun keduanya berisi NIP, namun nama kolomnya berbeda.  Query untuk **SELECT..INNER JOIN..ON** tabel *mata\_kuliah* dengan *daftar\_dosen* adalah :  *SELECT nama\_matkul,jumlah\_SKS,nama\_dosen FROM mata\_kuliah*  *INNER JOIN daftar\_dosen ON NIP\_dosen=NIP;*  Select nama\_matkul, jumlah\_sks, nama\_dosen from mata\_kuliah inner join dosen  ON NIP = nip\_dosen; **Cara Menggabungkan Tabel MySQL dengan SELECT..INNER JOIN..USING** Cara **JOIN** kedua adalah menggunakan **USING** sebagai pengganti **ON** untuk query **INNER** **JOIN**.  Format dasar dari penulisan query **SELECT..INNER JOIN..USING** adalah:   |  | | --- | | *SELECT nama\_kolom\_tampil FROM nama\_tabel\_pertama INNER JOIN*  *nama\_tabel\_kedua USING (nama\_kolom\_join)* |   Syarat untuk **INNER JOIN..USING** adalah ***kedua tabel harus memiliki nama kolom yang sama.*** Dalam contoh kita, kolom tersebut adalah kolom NIP. Namun karena nama kolom NIP untuk tabel *mata\_kuliah* adalah *NIP\_dosen*, sedangkan pada kolom *daftar\_dosen* kolom NIP hanya *NIP* saja, maka kita harus menyamakannya. Untuk hal ini saya memutuskan kedua kolom akan bernama **NIP\_dosen**.  *ALTER TABLE daftar\_dosen CHANGE NIP NIP\_dosen CHAR(10);*  *Desc daftar\_dosen;*  *Desc mata\_kuliah;*  *SELECT nama\_matkul, jumlah\_SKS, nama\_dosen*  *FROM mata\_kuliah INNER JOIN daftar\_dosen USING (NIP\_dosen);*  Di dalam **MySQL**, kata **INNER JOIN**  dapat diganti dengan **CROSS JOIN**, atau hanya **JOIN** saja. Ketiga kata kunci ini memiliki fungsi yang sama. MySQL menyediakannya agar dapat kompatible dengan SQL dari **RDBMS** lainnya seperti *ORACLE* dan *MS SQL*.  Misalnya untuk query kita diatas, dapat juga ditulis menjadi   |  | | --- | | *SELECT nama\_matkul, jumlah\_SKS, nama\_dosen*  *FROM mata\_kuliah CROSS JOIN daftar\_dosen USING (NIP\_dosen);* |  **Cara Penulisan Nama Kolom: Nama\_Tabel.Nama\_Kolom** Setelah kita merubah nama kolom NIP menjadi *NIP\_dosen* pada tabel *daftar\_dosen*, sekarang tiap tabel memiliki nama kolom yang sama, yakni **NIP\_dosen**. Jika kita menjalankan kembali query **SELECT..INNER JOIN..ON** dengan nama kolom ini, MySQL akan mengeluarkan error. Dari pesan error yang terjadi, sebenarnya MySQL ‘*bingung*’ dalam memilih kolom.  Query **ON NIP\_dosen=NIP\_dosen** menjadi ambigu karena kedua tabel kita memiliki nama kolom yang sama.  Untuk mengatasi ambiguitas ini, MySQL mengharuskan kita secara *eksplisit* menyebutkan nama tabelnya. Cara penulisan ini menggunakan tanda titik sebagai pemisah:   |  | | --- | | Nama\_Tabel.Nama\_Kolom |   Misalkan untuk merujuk pada kolom *NIP\_dosen* pada tabel *mata\_kuliah*, maka penulisannya menjadi:   |  | | --- | | Mata\_kuliah.NIP\_dosen |   Sehingga query **SELECT..INNER JOIN..ON** yang benar menjadi :  *SELECT nama\_matkul,jumlah\_SKS,nama\_dosen FROM mata\_kuliah*  *INNER JOIN daftar\_dosen ON mata\_kuliah.NIP\_dosen=daftar\_dosen.NIP\_dosen;* |

**Cara Mengganti Tampilan Nama Kolom & Tabel (Alias)**

### **Pengertian ALIAS (AS) dalam MySQL**

**MySQL** menyediakan perintah tambahan **AS** untuk *mengganti sementara* nama tabel atau kolom. Disebut sementara karena pada dasarnya nama tabel tersebut tidak berubah, hanya di ganti pada saat ditampilkan dengan query **SELECT.** Di dalam bahasa **SQL**, query **AS**  ini lebih dikenal sebagai ***alias*** dari nama tabel yang sebenarnya. **Alias** ditujukan untuk mempermudah penulisan query atau mempercantik tampilan hasil query.

### **Cara Menggunakan ALIAS untuk Mengganti Nama Tabel**

Perintah *alias* (dengan query **AS**) bukan merupakan query SQL yang dapat berdiri sendiri seperti **SELECT, UPDATE**, maupun **DELETE**. **AS** digunakan sebagai penambahan untuk query **SQL** lainnya.

Berikut adalah format dasar penulisan alias tabel:

|  |
| --- |
| *…nama\_tabel\_asli AS nama\_tabel\_alias…* |

Contoh sederhana penggunaan tabel ***alias***adalah sebagai berikut:

*SELECT ds.nama\_dosen, ds.no\_hp, ds.alamat FROM daftar\_dosen AS ds;*

Query diatas tidak terlalu berguna, karena kita bisa langsung menggunakannya **tanpa alias.** Perintah **AS** akan lebih berguna untuk query yang sedikit rumit, seperti **INNER JOIN.**

*SELECT nama\_matkul,jumlah\_SKS,nama\_dosen FROM mata\_kuliah*

*AS ma INNER JOIN daftar\_dosen AS da ON ma.NIP\_dosen=da.NIP\_dosen;*

### **Cara Menggunakan ALIAS untuk Mengganti Nama Kolom Tabel MySQL**

Selain untuk mengubah nama tabel, **MySQL** juga menyediakan ***ALIAS***untuk nama kolom. Salah satu kegunaan alias kolom ini adalah untuk merubah nama kolom agar lebih cantik.

Sama seperti ALIAS untuk nama tabel, alias untuk nama kolom juga merupakan perintah tambahan dari query inti seperti **SELECT**.

Format dasar penulisan alias kolom:

|  |
| --- |
| *…nama\_kolom\_asli AS nama\_kolom\_alias…* |

Contoh sederhana penggunaan kolom alias adalah :

SELECT nama\_dosen AS nama, no\_hp AS nomor\_handphone FROM daftar\_dosen;

Perhatikan hasil dari query tersebut. Pada bagian judul kolom, akan berubah seperti yang tertera pada bagian **AS**. Kolom **nama\_dosen** menjadi **nama**, dan kolom **no\_hp** menjadi **nomor\_handphone**.

Perintah **AS** sebagai *alias* dari nama kolom akan lebih berguna pada saat kita menggunakan fungsi untuk kolom. Contohnya, MySQL menyediakan fungsi **CONCAT** (singkatan dari *concatenates*) yang digunakan untuk menyambung kata:

*SELECT CONCAT(nama\_matkul, " jumlah SKS:", jumlah\_SKS) FROM mata\_kuliah;*

Hasil dari query tersebut, fungsi **CONCAT** kita gunakan untuk menggabungan kolom *nama\_matkul* dengan *jumlah\_SKS*. Perhatikan juga judul dari kolom tersebut.

Untuk membuat tampilan query lebih cantik, saya akan ubah judul kolom dengan menggunakan ALIAS: