**Modul 6**

**Menambahkan data dari FILE (Load Data Infile)**

Pada modul sebelumnya [cara Menambahkan data ke dalam Tabel (INSERT)](http://www.duniailkom.com/tutorial-mysql-menambahkan-data-ke-dalam-tabel-insert/), kita menambahkan data langsung pada **cmd MySQL**. Untuk keperluan menambahkan data dalam jumlah besar, akan lebih efisien dan mudah jika data tersebut kita siapkan dalam bentuk file terlebih dahulu, dan kemudian diload ke dalam tabel. Untuk keperluan ini **MySQL** menyediakan [query LOAD DATA INFILE](http://www.duniailkom.com/tutorial-mysql-menambahkan-data-dari-file-load-data-infile/).

mysql> SHOW DATABASES;

+--------------------+

| Database           |

+--------------------+

| information\_schema |

| universitas          |

| mysql              |

| performance\_schema |

| test               |

+--------------------+

5 rows in set (0.08 sec)

mysql> USE universitas;

Database changed

mysql> CREATE TABLE daftar\_dosen (NIP CHAR(10) PRIMARY KEY,

nama\_dosen VARCHAR(50) NOT NULL, no\_hp CHAR(13),

alamat VARCHAR(100));

Query OK, 0 rows affected (0.16 sec)

mysql> DESC daftar\_dosen;

+------------+--------------+------+-----+---------+-------+

| Field      | Type         | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+-------+

| NIP        | char(10)     | NO   | PRI | NULL    |       |

| nama\_dosen | varchar(50)  | NO   |     | NULL    |       |

| no\_hp      | char(13)     | YES  |     | NULL    |       |

| alamat     | varchar(100) | YES  |     | NULL    |       |

+------------+--------------+------+-----+---------+-------+

4 rows in set (0.14 sec)

**Cara Penulisan dasar query LOAD DATA INFILE**

**LOAD DATA INFILE** digunakan untuk menginput data kedalam tabel dimana data yang diinput berasal dari sebuah file. Query lengkap dari **LOAD DATA INFILE** sesuai dari manual **MySQL** adalah sebagai berikut :

LOAD DATA [LOW\_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file\_name'

    [REPLACE | IGNORE]

    INTO TABLE tbl\_name

    [CHARACTER SET charset\_name]

    [{FIELDS | COLUMNS}

        [TERMINATED BY 'string']

        [[OPTIONALLY] ENCLOSED BY 'char']

        [ESCAPED BY 'char']

    ]

    [LINES

        [STARTING BY 'string']

        [TERMINATED BY 'string']

    ]

    [IGNORE number {LINES | ROWS}]

    [(col\_name\_or\_user\_var,...)]

    [SET col\_name = expr,...]

**Latihan :**

Buka **notepad**, lalu tuliskan data berikut ke dalam file text tersebut, dan save sebagai **input\_data.txt**:

|  |
| --- |
| "0576431001","M. Siddiq","0812979005","Jakarta""0770435006","Siswanto","0812567765","Medan""0869437003","Andi Mulia","0812332564","Padang""0260432002","Maya Ari Putri","0812345234","Palembang""1080432007","Arif Budiman","0812456345","Makasar""0275430005","Susi Indriani","0812656532","Bogor" |

Pada saat membuat file, pastikan menekan **Enter** di akhir baris untuk membuat baris baru. Selanjutnya buka **MySQL Client command prompt** dan jalankan query berikut :

mysql> LOAD DATA INFILE 'D:\\MySQL\\input\_data.txt'

INTO TABLE daftar\_dosen FIELDS TERMINATED BY ','

ENCLOSED BY '"' LINES TERMINATED BY '\r\n';

Query OK, 6 rows affected (0.05 sec)

Records: 6  Deleted: 0  Skipped: 0  Warnings: 0

mysql> SELECT \* FROM daftar\_dosen;

+------------+----------------+------------+-----------+

| NIP        | nama\_dosen     | no\_hp      | alamat    |

+------------+----------------+------------+-----------+

| 0260432002 | Maya Ari Putri | 0812345234 | Palembang |

| 0275430005 | Susi Indriani  | 0812656532 | Bogor     |

| 0576431001 | M. Siddiq      | 0812979005 | Jakarta   |

| 0770435006 | Siswanto       | 0812567765 | Medan     |

| 0869437003 | Andi Mulia     | 0812332564 | Padang    |

| 1080432007 | Arif Budiman   | 0812456345 | Makasar   |

+------------+----------------+------------+-----------+

6 rows in set (0.00 sec)

Keterangan :

* Hal pertama yang kita tulis setelah perintah **LOAD DATA INFILE**adalah lokasi tempat file **input\_data.txt** berada. Dalam query diatas, file tersebut berada pada **D:\MySQL\input\_data.txt**. Tetapi karena didalam MySQL karakter ‘**\**’merupakan karakter khusus, maka penulisannya harus di-*double*, sehingga menjadi **D:\\MySQL\\input\_data.txt**. Jika lokasi file anda berbeda, tinggal menyesuaikan saja.
* Setelah lokasi file, selanjutnya kita menambahkan perintah ke tabel mana file tersebut akan diinput. Pada contoh kita, tabel tersebut adalah **daftar\_dosen**, sehingga ditulis **…INTO TABLE daftar\_dosen…**
* Perintah **FIELDS TERMINATED BY ‘,’**  dimaksudkan sebagai instruksi pada **MySQL** bahwa setiap kolom pada **input\_data.txt** dipisahkan oleh karakter koma ‘,’. Namun tidak harus tanda koma, tanda lain seperti titik, tanda ‘|’, maupun karakter tab bisa kita gunakan sebagai tanda pemisah, dan instruksikan **MySQL**untuk menggunakan tanda tersebut sebagai penanda kolom.
* Instruksi **ENCLOSED BY ‘”‘** memberitahu MySQL agar menghapus penanda kutip dua (“) dari tiap kolom. Namun jika pada **daftar\_dosen.txt** kita tidak menambahkan tanda kutip dua diawal data, **MySQL** tetap akan menerima data tersebut.
* Pada akhir query, perintah **…LINES TERMINATED BY ‘\r\n’** menginstruksikan **MySQL** agar tiap baris diakhiri dengan karakter **new line** dan **carriage return**. Karakter khusus ini terbentuk ketika kita menekan **Enter** pada keyboard untuk pindah baris pada notepad. ‘**\r**‘ merupakan cara penulisan untuk karakter **carriage return**, sedangkan ‘**\n**‘ adalah karakter **new line**.

**Modul 7**

### Cara Membatasi Hasil query SELECT (SELECT…LIMIT)

### Buatlah terlebih dahulu tabel daftar\_dosen dan tabel mata\_kuliah seperti di bawah ini :

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

**MySQL** menyediakan pilihan opsional **LIMIT** untuk membatasi hasil query **SELECT**, format dasar query **SELECT…LIMIT** adalah sebagai berikut:

|  |
| --- |
| SELECT nama\_kolom1 FROM nama\_tabel WHERE kondisi LIMIT baris\_awal, jumlah\_baris |

Dimana **baris\_awal** adalah awal nomor baris yang ingin ditampilkan, dan **jumlah\_baris** adalah jumlah baris yang diurutkan dari **baris\_awal**. Nomor baris pada MySQL diawali dengan nomor **0**.

Misalkan kita ingin menampilkan data 3 nama dosen paling atas yang dirutkan berdasarkan nama, maka querynya adalah:

|  |  |
| --- | --- |
|  | mysql> SELECT \* FROM daftar\_dosen ORDER BY nama\_dosen ASC LIMIT 0,3;+------------+----------------+------------+-----------+| NIP        | nama\_dosen     | no\_hp      | alamat    |+------------+----------------+------------+-----------+| 1080432007 | Arif Budiman   | 0812456345 | Makasar   || 0576431001 | M. Siddiq      | 0812979005 | Jakarta   || 0260432002 | Maya Ari Putri | 0812345234 | Palembang |+------------+----------------+------------+-----------+3 rows in set (0.00 sec) |

Jika kita hanya memberikan 1 angka saja di belakang instruksi **LIMIT**, maka MySQL menganggap bahwa angka dihitung dari baris teratas (baris ke 0). Contohnya jika kita ingin menampilkan 5 baris teratas mata kuliah yang diurutkan berdasarkan nama, maka querynya:

|  |  |
| --- | --- |
|  | mysql> SELECT \* FROM mata\_kuliah ORDER BY nama\_matkul LIMIT 5;+-------------+-------------------+------------+----------+------------+| kode\_matkul | nama\_matkul       | jumlah\_SKS | semester | NIP\_dosen  |+-------------+-------------------+------------+----------+------------+| DTBASE      | Database          |          4 |        4 | 0275430005 || FISDAS      | Fisika Dasar      |          2 |        1 | 0480432066 || JARKOM      | Jaringan Komputer |          3 |        3 | 0770435006 || MATDAS      | Matematika Dasar  |          4 |        1 | 0160436012 || MIKROP      | Mikro Prosesor    |          2 |        5 | 0480432066 |+-------------+-------------------+------------+----------+------------+5 rows in set (0.00 sec) |
|  |  |

**Modul 8**

# **Pencarian Data MySQL dengan Regular Expression (REGEXP)**

### Pengertian Regular Expression

Bagi yang sudah pernah mengenal bahasa pemrograman lain, mungkin sudah mengenal **Reguler Expression** yang kadang disingkat dengan **RexExp**, atau hanya **RE**. Dalam bahasa sederhananya, **Regular Expression** *adalah kumpulan huruf atau karakter yang digunakan untuk pencocokan pola (****pattern matching****)*. Pola disini contohnya ‘*pola untuk kata yang diawali dengan huruf a dan diakhiri dengan huruf j*’, atau yang sedikit rumit seperti ‘*pola untuk kata yang diawali huruf a,b, c dengan panjang maksimal 5 huruf, mengandung minimal sebuah angka*’.

Cakupan **Regular Expression** cukup luas, dan tidak terbatas hanya pada MySQL saja. Hampir semua bahasa pemrograman komputer menyediakan fungsi khusus untuk ***regular expression***. Dalam tutorial ini saya hanya membahas sebagian kecil dari **RegExp**.

Jika pada query **SELECT..LIKE** kita menggunakan pola ‘**s%**’ sebagai kata kunci yang berarti ‘*kata yang diawali dengan huruf s dan memiliki banyak huruf 1 atau lebih*’ dalam regular expression, penulisannya menjadi **‘^S.\***’. Berikut ini adalah aturan penulisan **RegExp**:

* **.** : tanda titik dalam RegExp berarti sebuah karakter apa saja
* **[ … ]**: tanda kurung siku ini berarti kumpulan karakter. Misalkan [abc] akan cocok dengan ‘a’, ‘b’, atau ‘c’. kita bisa juga menggunakan jangkauan (range), contohnya [a-z] akan cocok dengan seluruh huruf, [0-9] akan cocok dengan seluruh angka.
* **\***: tanda bintang ini akan cocok dengan 0 atau lebih karakter sebelumnya. Misalkan ‘a\*’ berarti akan cocok dengan seluruh kata yang mengandung 0 atau lebih a.
* **^**: tanda pangkat atau topi ini menandakan berada di awal kata.
* **$**: tanda dollar ini berarti bahwa pola berada di akhir kata.

Contoh penggunakan pola **RegExp**:

* ‘**ab\***’: Pola ini berarti akan cocok dengan seluruh kata yang mengandung a dan diikuti oleh b atau tidak sama sekali. Contohnya: ‘a’, ‘ab’, ‘abbbbbb’, dan juga ‘kebab’, karena untuk RegExp, kita harus menyatakan dimana karakter itu muncul.
* ‘**^ab\***’: Pola ini sama artinya dengan ‘ab\*’ seperti diatas, namun tanda ^ menyatakan bahwa pola ini harus berada di awal kata, sehingga ‘kebab’ tidak akan cocok.
* ‘**^s..i$**’: Pola ini akan cocok dengan seluruh kata yang diawali dengan s, dan diakhiri dengan i, terdiri dari 4 huruf. Contohnya: susi, sapi, dan siti.

| 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 |

mysql> SELECT \* FROM daftar\_dosen WHERE nama\_dosen REGEXP '^s.\*';

+------------+---------------+------------+-----------+

| NIP        | nama\_dosen    | no\_hp      | alamat    |

+------------+---------------+------------+-----------+

| 0160436012 | Sabrina Sari  | 0812349900 | Pekanbaru |

| 0275430005 | Susi Indriani | 0812656532 | Bogor     |

+------------+---------------+------------+-----------+

2 rows in set (0.00 sec)

Penggunaan **Regular Expression** akan diperlukan jika kita butuh pencarian yang lebih rumit. Contohnya kita ingin mencari *nama\_dosen* yang diawali dengan huruf ‘**m**’ atau ’**s**’, dan diakhiri dengan huruf vocal. Pencarian seperti ini bisa saja kita lakukan dengan **SELECT..LIKE**, namun akan menggunakan operator **OR** yang banyak, karena untuk huruf vocal berarti **nama\_dosen** dapat berakhir dengan huruf a, i, u, e dan o. Dalam Regular Expression, pola itu akan berbentuk: ‘**^[ms].\*[aiueo]$**‘

mysql> SELECT \* FROM daftar\_dosen WHERE nama\_dosen REGEXP '^[ms].\*[aiueo]$';

+------------+----------------+------------+-----------+

| NIP        | nama\_dosen     | no\_hp      | alamat    |

+------------+----------------+------------+-----------+

| 0160436012 | Sabrina Sari   | 0812349900 | Pekanbaru |

| 0260432002 | Maya Ari Putri | 0812345234 | Palembang |

| 0275430005 | Susi Indriani  | 0812656532 | Bogor     |

+------------+----------------+------------+-----------+

3 rows in set (0.00 sec)

**Modul 9**

# **Cara Menggabungkan Tabel MySQL (INNER JOIN)**

| 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 |

 **Tabel mata\_kuliah**

### Cara Menggabungkan Tabel MySQL dengan INNER JOIN

Tujuan dari menggabungkan tabel adalah untuk menyajikan informasi secara lebih detail. Contohnya dari tabel **daftar\_dosen** dan tabel **mata\_kuliah** diatas, tujuan kita adalah menyajikan informasi mata kuliah sekaligus nama dosen yang mengajar mata kuliah tersebut. Kita ingin menyajikan informasi yang berisi nama **mata kuliah, jumlah SKS**, dan **nama dosen** yang mengajar.

Query **INNER JOIN** memiliki beberapa variasi, yaitu **SELECT..INNER JOIN..ON** dan **SELECT..INNER JOIN..USING**. Kita akan membahasnya satu persatu.

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 |

* *nama\_kolom\_tampil* adalah nama dari kolom yang akan kita tampilkan, bisa semua kolom dalam tabel, atau hanya kolom tertentu saja.
* *nama\_tabel\_pertama* adalah nama tabel pertama yang akan digabung.
* *nama\_tabel\_kedua* adalah nama tabel kedua yang akan digabung.
* *nama\_kolom\_join\_tabel\_pertama* adalah nama kolom yang akan digunakan sebagai join dari tabel pertama.
* *nama\_kolom\_join\_tabel\_kedua* adalah nama kolom yang akan digunakan sebagai join dari 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;

+-------------------+------------+---------------+

| nama\_matkul       | jumlah\_SKS | nama\_dosen    |

+-------------------+------------+---------------+

| Database          |          4 | Susi Indriani |

| Fisika Dasar      |          2 | Tia Santrini  |

| Jaringan Komputer |          3 | Rubin Hadi    |

| Matematika Dasar  |          4 | Sabrina Sari  |

| Mikro Prosesor    |          2 | Tia Santrini  |

| Sistem Operasi    |          2 | Sabrina Sari  |

| Teknik Kompilasi  |          2 | Tia Santrini  |

+-------------------+------------+---------------+

7 rows in set (0.00 sec)

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 JOINnama\_tabel\_kedua USING (nama\_kolom\_join) |

* *nama\_kolom\_tampil* adalah nama dari kolom yang akan kita tampilkan, bisa semua kolom dalam tabel, atau hanya kolom tertentu saja.
* *nama\_tabel\_pertama* adalah nama tabel pertama yang akan digabung.
* *nama\_tabel\_kedua* adalah nama tabel kedua yang akan digabung.
* *nama\_kolom\_join* adalah nama kolom yang akan digunakan sebagai 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);

Query OK, 8 rows affected (0.08 sec)

Records: 8  Duplicates: 0  Warnings: 0

Setelah kedua kolom NIP bernama sama, saatnya kita mengabungkan kedua tabel ini

|  |  |
| --- | --- |
|  | mysql> SELECT nama\_matkul, jumlah\_SKS, nama\_dosenFROM mata\_kuliah INNER JOIN daftar\_dosen USING (NIP\_dosen);+-------------------+------------+---------------+| nama\_matkul       | jumlah\_SKS | nama\_dosen    |+-------------------+------------+---------------+| Database          |          4 | Susi Indriani || Fisika Dasar      |          2 | Tia Santrini  || Jaringan Komputer |          3 | Rubin Hadi    || Matematika Dasar  |          4 | Sabrina Sari  || Mikro Prosesor    |          2 | Tia Santrini  || Sistem Operasi    |          2 | Sabrina Sari  || Teknik Kompilasi  |          2 | Tia Santrini  |+-------------------+------------+---------------+7 rows in set (0.00 sec) |

Hasilnya sama persis dengan **SELECT..INNER JOIN..ON**, hanya berbeda cara penulisan.

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

|  |
| --- |
| mysql> SELECT nama\_matkul, jumlah\_SKS, nama\_dosenFROM 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:

|  |  |
| --- | --- |
|  | mysql> SELECT nama\_matkul,jumlah\_SKS,nama\_dosen FROM mata\_kuliah INNER JOIN daftar\_dosen ON NIP\_dosen=NIP\_dosen;ERROR 1052 (23000): Column 'NIP\_dosen' in on clause is ambiguous |

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:

|  |  |
| --- | --- |
|  | mysql> SELECT nama\_matkul,jumlah\_SKS,nama\_dosen FROM mata\_kuliahINNER JOIN daftar\_dosen ON mata\_kuliah.NIP\_dosen=daftar\_dosen.NIP\_dosen;+-------------------+------------+---------------+| nama\_matkul       | jumlah\_SKS | nama\_dosen    |+-------------------+------------+---------------+| Database          |          4 | Susi Indriani || Fisika Dasar      |          2 | Tia Santrini  || Jaringan Komputer |          3 | Rubin Hadi    || Matematika Dasar  |          4 | Sabrina Sari  || Mikro Prosesor    |          2 | Tia Santrini  || Sistem Operasi    |          2 | Sabrina Sari  || Teknik Kompilasi  |          2 | Tia Santrini  |+-------------------+------------+---------------+7 rows in set (0.00 sec) |

[Query SELECT..INNER JOIN](http://www.duniailkom.com/tutorial-belajar-menggabungkan-tabel-mysql-dengan-inner-join/) ini akan sering kita gunakan untuk menampilkan data dari beberapa tabel. Dalam mendesain database, sedapat mungkin kita mengeleminasi data yang redundan (sama) dari satu tabel besar menjadi beberapa tabel kecil, lalu menggabungkannya pada saat menampilkan data.