MERDEKA BELAJAR Kampus Merdeka Belmawa

MODUL

Praktikum Basis Data

PROGRAM STRATA I
TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PGRI MAHADEWA INDONESIA

KATA PENGANTAR

Puji syukur ke hadirat Ida Sang Hyang Widhi Wasa/Tuhan Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Modul Pratikum Basis Data untuk mahasiswa/i Universitas PGRI Mahadewa Indonesia (UPMI Bali) Jurusan Teknik Informatika ini dapat diselesaikan dengan sebaik-baiknya.

Modul Pratikum Basis Data ini dibuat sebagai pedoman dalam melakukan kegiatan praktikum Basis Data yang merupakan kegiatan penunjang mata kuliah Basis Data pada Program Studi Teknik Informatika ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan tujuan pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Pratikum Basis Data ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Penyusun

DAFTAR ISI

MODUL 1
PENGENALAN MYSQL

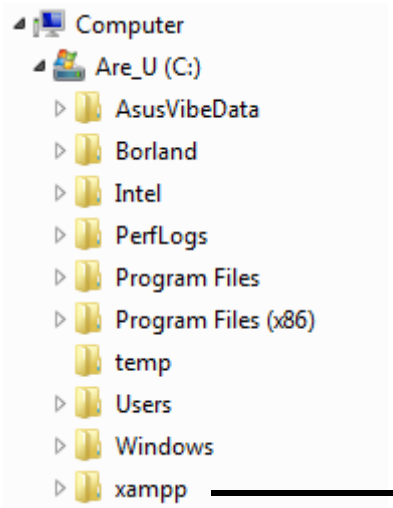
MySQL merupakan salah satu DBMS yang bersifat freeware, sehingga menjadi popular dikalangan pengguna database. Selain karena Free MySQL juga merupakan DBMS yang relatif ringan dan mudah digunakan. Ada beberapa software yang bisa digunakan untuk menjalankan service MySQL, diantaranya WAMP, PhpTriad dan XAMPP. Namun XAMPP merupakan yang paling popular saat ini karena masih terus dilakukan update pada software nya.

XAMPP merupakan program paket yang didalamnya terdapat beberapa software yang digabungkan menjadi satu. Dalam XAMPP sendiri didalamnya terdapat 5 software yang berbeda (MySql, Apache, FileZilla,Mercury, Tomcat). Jadi ketika menginstall XAMPP kita tidak perlu lagi menginstall Apache dan MySQL secara terpisah.

1. Install MySQL (XAMPP)

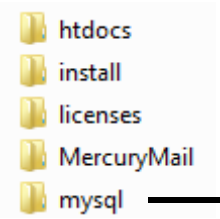
   Untuk master software XAMPP bisa didownload gratis di situ resminya www.apachefriends.org/en/**xampp**.html . Hal yang perlu diingat ketika menginstall XAMPP adalah direktori tempat meletakan hasil installan.

   Contoh hasil setelah diinstal :



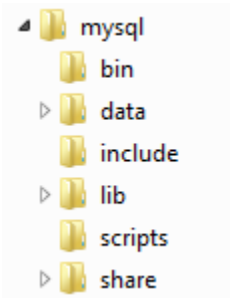
Direktori XAMPP setelah diinstal

2. Direktori MySQL



Direktori mysql

Semua proses yang berjalan untuk bekerja dalam MySQL terdapat dalam direktori mysql. Ada beberapa direktori yang perlu diketahui dalam direltori mysql :



a. Bin

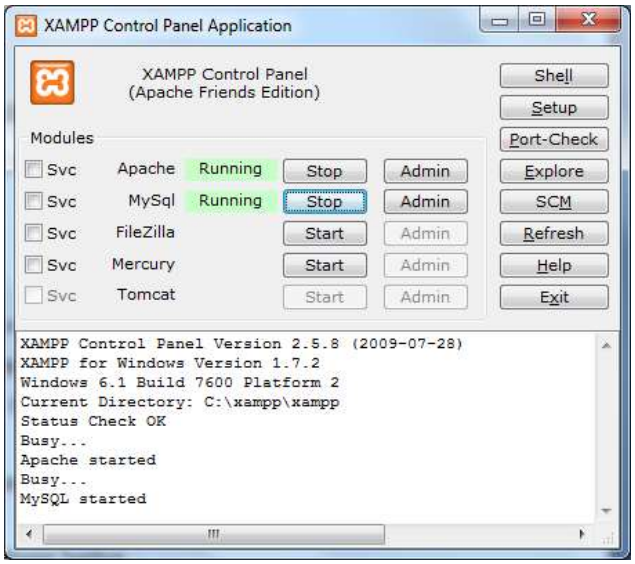
Direktori bin berisi service-service yang bisa dijalankan untuk mengakses mysql.

b. Data

Direktori ini berisi database dan tabel yang sudah dibuat.

3. Menggunakan MySQL

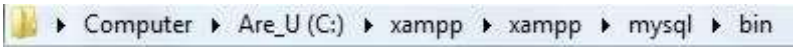
a. Nyalakan (START) Service MySQL



b. Masuk kedalam MySQL melalui CMD

- Masuk kedalam direktori mysql\bin dalamdirektori hasil installan xampp.

Contoh :



Sintaks :

```
cd : c\xampp\xampp\mysql\bin\
```

Contoh dalam CMD :

```
C:\Users\AyuMauLida>cd c:\xampp\xampp\mysql\bin
c:\xampp\xampp\mysql\bin>_
```

- Tuliskan sintaks sebagai berikut :

```
mysql.exe –user=(user) –password=(password)
```
atau
```
mysql –u user password
```

contoh untuk user = root dan password = (kosong) :

```
mysql.exe –user=root –password=
```
atau
```
mysql –u root
```

Contoh dalam CMD :

```
c:\xampp\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.37 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Tanda sudah bisa menuliskan query SQL.

MODUL 2
SHOW DAN CREATE

1. Show

   Berfungsi untuk melihat list/daftar dari database atau tabel yang sudah dibuat.

   a. Melihat daftar database

      Untuk melihat daftar database yang sudah dibuat.

      Query : `SHOW databases;`

   b. Melihat daftar tabel

      Untuk melihat daftar database yang ada digunakan fungsi show tables.

      Query : `SHOW tables;`

2. Use

   Perintah USE digunakan untuk memilih database yang ingin digunakan.

   Query : `USE nama_database;`

3. Create

   Berfungsi untuk membuat database atau tabel baru.

   a. Membuat database

      ```
      CREATE DATABASE nama_database;
      ```

   b. Membuat tabel

      ```
      CREATE TABLE nama_tabel (
      Field1 TipeData1,
      Field2 TipeData2,
      ……
      FieldN TipeDataN
      );
      ```

4. Desc

   Digunakan untuk melihat struktur (metadata) sebuah tabel.

   ```
   DESC nama_tabel;
   ```

Latihan

1. Buat sebuah database mahasiswa.

```
CREATE DATABASE mahasiswa;
```

2. Buat dua buah tabel siswa dan dosen di dalam database mahasiswa

```
CREATE TABLE siswa (
Nim int not null,
Nama char(20),
Wali int
);
CREATE TABLE dosen (
Nip int not null,
Nama char(20),
Alamat char(50)
);
```

Tugas

1. Buat satu buah database
2. Buat dua buah tabel pada database yang telah dibuat sebelumnya

MODUL 3
INSERT, SELECT, WHERE

1. Insert

   Berfungsi untuk mengisikan data kedalam sebuah tabel.

   Query :

   ```
   INSERT INTO nama_tabel VALUES ('nilaiField1', 'nilaiField2', ……,
   'nilaiFieldN');
   ```

2. Select

   Digunakan untuk menampilkan isi tabel. Untuk menampilkan sebuah tabel ada beberapa kondisi yang bisa digunakan, antara lain :

   a. SELECT ALL

      Kondisi ini digunakan untuk menampilkan semua record dan semua field nya dalam sebuah tabel.

      Query : `SELECT * FROM nama_tabel;`

   b. SELECT FIELD

      Kondisi ini digunakan untuk menampilkan remua record yang ada, tetapi hanya field tertentu saja.

      Query : `SELECT field1,field2,… FROM nama_tabel;`

   c. SELECT RECORD (WHERE)

      Kondisi ini digunakan untuk menampilkan isi dari record tertentu saja. WHERE bisa digunakan untuk SELECT ALL maupun SELECT FIELD.

      Query : `SELECT (ALL/FIELD) FROM nama_tabel WHERE nama_field = 'value';`

Latihan

1. Isikan data kedalam tabel siswa dan dosen yang telah dibuat sebelumnya

   a. Mengisi data pada tabel siswa

```
INSERT INTO siswa VALUES ('123040203', 'DWI MAHARANI', '19721201');
INSERT INTO siswa VALUES ('123040204', 'ANANTA WIJAYA', '19721204');
INSERT INTO siswa VALUES ('123040205', 'EKA INDRAWAN', '19721203');
INSERT INTO siswa VALUES ('123040206', 'PUTU ANI', '19721201'); INSERT
INTO siswa VALUES ('123040207', 'GDE IWAN', '19721202'); INSERT INTO
siswa VALUES ('123040208', 'WIRA SENTANU', '19721204');
```

b. Mengisi data pada tabel wali

```
INSERT INTO dosen VALUES ('19721201', 'KOMANG BAGUS', '19721201');
INSERT INTO dosen VALUES ('19721202', 'AYU APRILIA', '19730412');
INSERT INTO dosen VALUES ('19721203', 'KADEK JUNIARTA', '19721201');
INSERT INTO dosen VALUES ('19721204', 'PUTU MEGAYASA', '19721201');
```

2. Tampilkan isi data dari tabel siswa dan dosen
   a. Menampilkan isi tabel siswa

   ```
   SELECT * FROM siswa;
   ```

   b. Menampilkan isi tabel dosen

   ```
   SELECT * FROM dosen;
   ```

   c. Menampilkan nip dan nama dari tabel wali

   ```
   SELECT nip, nama FROM doen
   ```

   d. Menampilkan record dengan nim = 123040208 dari tabel siswa

   ```
   SELECT * FROM siswa WHERE nim = '123040208';
   ```

   e. Menampilkan nim dan wali dengan nim = 123040208 dari tabel siswa

   ```
   SELECT nim, dosen FROM siswa WHERE nim = '123040208';
   ```

Tugas

1. Tampilkan record dengan nim = 123040204 atau dosen = 19721204.
2. Isikan data pada tabel yang telah dibuat pada tugas sebelumnya.
3. Tampilkan isi data pada tabel yang telah dibuat pada tugas sebelumnya.

MODUL 4
KONDISI

Kondisi yang dibentuk pada perintah-perintah SQL dapat dalam kriteria yang sifatnya pembandingan dengan menggunakan tanda '=' untuk mencari suatu record maupun tidak. Beberapa kondisi dalam SQL :

1. LIKE

   Menampilkan isi record dengan menggunakan sebagian dari isi sebuah data. Secara fungsi LIKE berfungsi sama seperti WHERE (=).

   Query : `SELECT * FROM nama_tabel WHERE nama_field LIKE 'kondisi';`

   Ada terdapat 3 kondisi di dalam like :

   a. %data%

      Persen didepan dibelakang menunjukan bahwa record yang ingin ditampilkan adalah semua record yang mengandung "data" pada sebuah field. "data" bisa berada diawal, tengah ataupun akhir.

   b. %data

      Persen didepan menunjukan bahwa record yang ingin ditampilkan adalah semua record yang mengandung "data" pada sebuah field dan "data" harus berada diawal.

   c. data%

      Persen dibelakang menunjukan bahwa record yang ingin ditampilkan adalah semua record yang mengandung "data" pada sebuah field dan "data" harus berada akhir.

2. ORDER BY

   Jika sebuat SELECT tidak memiliki ORDER BY, maka record akan ditampilkan sesuai dengan urutan penginputan datanya. ORDER BY memungkinkan menampilkan hasil SELECT sesuai dengan urutan yang diinginkan.

   Query : `SELECT * FROM nama_tabel ORDER BY nama_field;`

3. ASC/DESC

   Dalam mengurutkan terdapat dua kondisi : Mengurutkan data dengan terurut naik (ASCENDING) atau terurut menurun (DESCENDING).

4. GROUP BY

   GROUP BY memungkinkan mengelompokan hasil dari SELECT berdasarkan persamaan isi datanya.

   Query : `SELECT * FROM nama_tabel ORDER BY nama_field;`

Latihan

Database : penjualan

Tabel : pelanggan

field	type
kode	int
nama	char
alamat	char
kota	char
notelp	char
tipe	char

Kode	Nama	Alamat	Kota	No Telpon	Tipe
123	INDRA	JL KLEDOKAN NO 21	DENPASAR	08123456789	GOLD
124	ANANTA	JL KAPAS N0 3	SINGARAJA	08123456788	PREMIUM
125	AGUNG	JL MERDEKA NO 14	BADUNG	08123456787	PREMIUM
126	MADE DEWI	JL PELANGI NO 2	JAKARTA	08123456786	PREMIUM
127	DEWA PUTRA	JL MACAN NO 34	SURABAYA	08123456785	GOLD
128	PUTRA BIMA	JL KELINCI NO 23	SURABAYA	08123456784	GOLD
129	ELYN GAURA	JL MELATI NO 12	YOGYAKARTA	08123456783	ECONOMY
130	LETISIA RURON	JL MERPATI NP 26	YOGYAKARTA	08123456782	ECONOMY
131	KARTINI	JL DANAU TOBA NO 12	SEMARANG	08123456781	PREMIUM
132	PUTRI DIANA	JL DANAU KERINCI NO 13	BEKASI	08123456799	GOLD
133	JULIA RAHMAN	JL ELANG NO 32	JAKARTA	08123456798	ECONOMY
134	PUTRA PERDANA	JL TULIP NO 12	NEGARA	08123456797	PREMIUM
135	PRANANDA GITA	JL SEDAP MALAM N0 27	KLUNGKUNG	08123456796	ECONOMY

1. Mengurutkan berdasarkan nama DESC

   Query : `SELECT * FROM pelanggan ORDER BY nama DESC;`

2. Mengelompokan record berdasarkan tipe

   Query : `SELECT * FROM pelanggan GROUP BY tipe;`

3. Menampilkan record dengan memiliki data pada field nama = ‘dewa’

Query : `SELECT * FROM pelanggan WHERE nama LIKE '%dewa%';`

Tugas

Gunakan fungsi LIKE :

1. Tampilkan record dengan kode 123 & 134. Urutkan berdasarkan alamat asc
2. Tampilkan record dengan kode 128 & 134.
3. Tampilkan record dengan nomor rumah 12, urutkan berdasarkan tipe desc.
4. Tampilkan record dengan kode 132, 133, 134, 135. Urutkan berdaskan nama asc.
5. Tampilkan record dengan kode 127, 128, 134. Urutkan berdasarkan alamat.

MODUL 5
UPDATE, DELETE, ALTER, DROP, DESC

1. UPDATE

   Fungsi UPDATE digunakan untuk merubah isi data field dari sebuah record.

   Query :

   ```
   UPDATE nama_tabel SET nama_field = 'value' WHERE nama_field = 'value'
   ```

2. DELETE

   Fungsi DELETE digunakan untuk menghapus sebuah record dalam sebuah database.

   Query : `DELETE nama_tabel WHERE nama_field = 'value'`

3. ALTER

   Perintah ALTER berfungsi untuk mengubah struktur dari sebuah tabel. Bisa berarti menambahkan field baru, merubah field yang sudah ada, maupun menghapus field yang sudah ada.

   Query :

   - `ALTER TABLE nama_tabel ADD new_field tipedata`
   - `ALTER TABLE nama_tabel CHANGE field_lama field_baru tipedata`
   - `ALTER TABLE nama_tabel DROP nama_field`

4. DROP

   Perintah DROP digunakan untuk menghapus struktur dalam database, termasuk menghapus database dan tabel.

   Query :

   ```
   DROP DATABASE nama_database;
   DROP TABLE nama_tabel;
   ```

5. DESC

   Perintah DESC digunakan untuk melihat struktur/metadata dari sebuah tabel.

   Query : `DESC nama_table;`

Latihan

Database :

penjualan

Tabel : barang

Field	Tipedata
ID	CHAR (PK)
NAMA_BARANG	CHAR
JENIS_BARANG	CHAR
HARGA_BELI	INT
HARGA_JUAL	INT
STOK	INT

Isi tabel BARANG

ID	NAMA_BARANG	JENIS_BARANG	HARGA_BELI	HARGA_JUAL	STOK
B01	RINSO	DETERGEN	10000	11000	10
B02	DAIA	DETERGEN	9000	10000	20
B03	LIFEBUOY	SABUN MANDI	2000	3000	30
B04	LIFEBUOY	SHAMPOO	12000	14000	20
B05	LUX	SABUN MANDI	2000	3000	12
B06	DETTOL	SABUN MANDI	3000	4000	15
B07	CITRA	BODY LOTION	12000	15000	16
B08	CITRA	SABUN MANDI	2000	2500	25
B09	PANTENE	SHAMPOO	16000	17000	30
B10	PEPSODENT	PASTA GIGI	9000	11000	19

- Mengubah stok baris pertama menjadi 15

```
UPDATE barang SET stok = '15' WHERE id = 'B01';
```

- Menghapus record pada baris pertama

```
DELETE FROM barang WHERE id = 'B01';
```

- Menambahkan sebuah kolom baru dengan nama "DISTRIBUTOR" bertipe CHAR

```
ALTER TABLE barang ADD distributor char(15);
```

- Mengubah nama kolom DISTRIBUTOR menjadi PENYALUR bertipe CHAR

```
ALTER TABLE barang CHANGE distributor penyalur char(15);
```

- Menghapus kolom PENYALUR

```
ALTER TABLE barang DROP penyalur;
```

SOAL

1. Ubah harga beli pada baris terakhir menjadi 10000
2. Ubah harga jual untuk LIFEBUOY jenis SABUN MANDI menjadi 4000
3. Ubah harga jual dan harga beli untuk SHAMPO PANTENE
4. Tambahkan sebuah kolom PRODUKSI bertipe CHAR
5. Ubah tipe data kolom PRODUKSI menjadi INT
6. Buat satu buah database : COBA_DROP
7. Buat satu buah tabel : COBA_DROP (coba int)
8. Hapus tabel COBA_DROP
9. Hapus database COBA_DROP

TUGAS

Gunakan data pada tugas sebelumnya :

1. Lakukan UPDATE isi sebuah kolom pada sebuah record.
2. Lakukan UPDATE isi 2 buah kolom pada sebuah record.
3. Hapus salah satu record
4. Tambahkan satu buah kolom baru pada tabel yang sudah ada
5. Isikan data pada kolom yang dibuat pada no.4
6. Buat satu buah database dan satu buah tabel di dalamnya
7. Lakukan penghapusa tabel dan database yang sudah dibuat pada no.6

MODUL 6
FUNGSI – FUNGSI DALAM MYSQL

Fungsi merupakan suatu rutin khusus yg disediakan oleh MySQL untuk melakukan manipulasi suatu data.

Bentuk Umum :

```
nama_fungsi([argumen1[, argumen2[, …]]])
```

Argumen1, argumen2, ... Adalah argumen/ parameter yang dibutuhkan oleh fungsi.

Fungsi digunakan sebagai bagian dari perintah select.

```
SELECT fungsi(ekspresi) [FROM namatabel];
```

FROM namatabel tidak harus diisi.

1. Fungsi STRING

- ASCII(x)

  Menghasilkan kode ASCII untuk karakter pertama dalam suatu string x.

  Contoh :

```
mysql> SELECT ASCII('a');mysql> SELECT ASCII('Teks');
+-------------+            +-----------------+
| ASCII('a') |             | ASCII('Teks') |
+-----------+              +-----------------+
|          97 |            |              84 |
+-------------+            +-----------------+
```

◉ CHAR(x,y,z,…)

Menghasilkan nilai string berdasarkan kode ASCII yang dituliskan dalam parameternya.

Contoh :

```
mysql> SELECT CHAR(65,78,84,65);
+-------------------+
| CHAR(65,78,84,65) |
+-------------------+
| ANTA              |
+-------------------+
```

◉ LENGTH(X)

Untuk mendapatkan panjang sebuah string X

Contoh :

```
mysql> SELECT LENGTH('DATABASE');
+--------------------+
| LENGTH('DATABASE') |
+--------------------+
|                  8 |
+--------------------+
```

◉ CONCAT()

Menggabungkan beberapa string dalam parameter menjadi satu string. Jika ada NULL maka hasil dari CONCAT adalah NULL.

```
mysql> SELECT CONCAT('SATU','DUA','TIGA');
+-----------------------------+
| CONCAT('SATU','DUA','TIGA') |
+-----------------------------+
| SATUDUATIGA                 |
+-----------------------------+
1 row in set (0.05 sec)
mysql> SELECT CONCAT('SATU','NULL','TIGA');
+-----------------------------+
| CONCAT('SATU','NULL','TIGA') |
+-----------------------------+
| SATUNULLTIGA                |
+-----------------------------+
```

◉ INSERT(X,Y,Z,J)

Menghasilkan string X yang telah diganti isinya dengan string J mulai dari posisi ke Y sebayak Z.

Contoh :

```
mysql> SELECT INSERT('DATABASE',5,4,'WARE');
+-------------------------------+
| INSERT('DATABASE',5,4,'WARE') |
+-------------------------------+
| DATAWARE                      |
+-------------------------------+
```

◉ INSTR(X,Y)

Menghasilkan nilai posisi Y di dalam string X

Contoh :

```
mysql> SELECT INSTR('NAMA KAMPUS INI BUMIGORA','KAMPUS');
+-----------------------------------------------+
| INSTR('NAMA KAMPUS INI BUMIGORA','KAMPUS') |
+-----------------------------------------------+
|                                             6 |
+-----------------------------------------------+
```

◉ LOCATE(X,Y,Z)

Memberikan posisi string X di dalam string Y mulai posisi ke Z.

Contoh :

```
mysql> SELECT LOCATE('D', 'DATABASE',3);
+-----------------------------+
| LOCATE('D', 'DATABASE',3) |
+-----------------------------+
|                           0 |
+-----------------------------+
1 row in set (0.00 sec)
mysql> SELECT LOCATE('D', 'DATABASE',1);
+-----------------------------+
| LOCATE('D', 'DATABASE',1) |
+-----------------------------+
|                           1 |
+-----------------------------+
```

◉ LEFT(X,Y)

Mengambil sejumlah Y karakter dari string X, mulai posisi pertama.

Contoh :

RIGHT(X,Y)

```
mysql> SELECT LEFT('DATABASE', 4);
+---------------------+
| LEFT('DATABASE', 4) |
+---------------------+
| DATA                |
+---------------------+
mysql> SELECT RIGHT('DATABASE', 4);
+----------------------+
| RIGHT('DATABASE', 4) |
+----------------------+
| BASE                 |
+----------------------+
```

Mengambil sejumlah Y karakter dari string X, mulai dari posisi paling akhir.

Contoh :

- MID(X,Y,Z)

  Mengambil data string X sejumlah Z karakter mulai dari posisi ke Y.

  Contoh :

```
mysql> SELECT MID('DATABASE', 3, 4);
+-----------------------+
| MID('DATABASE', 3, 4) |
+-----------------------+
| TABA                  |
+-----------------------+
```

- LTRIM(X)

  Membuang spasi di sebelah kiri string X.

  Contoh :

- RTRIM(X)

  Membuang spasi disebelah kanan string X.

  Contoh :

- REVERSE(X)

  Membalik urutan penulisan sebuah string X.

  Contoh :

```
mysql> SELECT REVERSE('ESABATAD');
+----------------------+
| REVERSE('ESABATAD') |
+----------------------+
| DATABASE             |
+----------------------+
```

2. Fungsi Tanggal

Format tanggal dalam MySQL :

yyyy-mm-dd HH:ii:ss

yyyy : Tahun

mm : Bulan

dd : Tanggal

HH : Jam

ii : Menit

ss : Detik

Simbol Format	Arti
%M	Nama bulan secara penuh
%m	Nomor bulan
%b	Nama bulan disingkat
%W	Nama hari secara lengkap
%D	Nomor hari dalam bulan
%Y	Tahun dalam format 4 digit
%y	Tahun dengan 2 digit
%j	Nomor hari dalam 1 tahun
%a	Nama hari disingkat
%d	Nomor hari dalam 1 bulan
%r	Waktu dalam format 12 jam
%T	Waktu dalam format 24 jam
%H	Jam dalam format 24 jam
%h	Jam dalam format 12 jam
%S	Detik

- CURDATE()

  Menampilkan tanggal sekarang dari sistem.

  Contoh :

```
mysql> SELECT CURDATE();
+------------+
| CURDATE()  |
+------------+
| 2012-04-19 |
+------------+
```

- CURTIME()

  Menampilkan waktu sekarang dari sistem.

  Contoh :

```
mysql> SELECT CURTIME();
+-----------+
| CURTIME() |
+-----------+
| 02:01:15  |
+-----------+
```

- CURRENT_TIMESTAMP() / NOW()

  Menampilkan waktu saat ini, tanggal dan jam.

Contoh :DATE_FORMAT(x, SimbolFormat)

Memgkonversi data tanggal sesuai dengan format yang diinginkan.

```
mysql> SELECT CURRENT_TIMESTAMP();
+---------------------+
| CURRENT_TIMESTAMP() |
mysql> SELECT DATE_FORMAT('2012-04-19', '%M %D %Y');
+-------------------------------------+
| DATE_FORMAT('2012-04-19', '%M %D %Y') |
+-------------------------------------+
| April 19th 2012                     |
+-------------------------------------+
```

Contoh :

- DAY_NAME(X)

  Menampilkan nama hari dari tanggal yang menjadi argumen dalam X. Contoh :

```
mysql> SELECT DAYNAME('2012-04-19');
+-----------------------+
| DAYNAME('2012-04-19') |
+-----------------------+
| Thursday              |
+-----------------------+
```

- DAYOFMONTH(DATE)

  Menampilkan nomor hari dari tanggal yang menjadi argumen.

  Contoh :

```
mysql> SELECT DAYOFMONTH('2012-04-19');
+--------------------------+
| DAYOFMONTH('2012-04-19') |
+--------------------------+
|                       19 |
+--------------------------+
```

- MONTH(DATE)

  Menampilkan bagian bulan dari tanggal yang menjadi argumen.

```
mysql> SELECT MONTH('2012-04-19');
+---------------------+
| MONTH('2012-04-19') |
+---------------------+
|                   4 |
+---------------------+
```

- YEAR(DATE)

  Menampilkan bagian tahun dari tanggal yang menjadi argumen.

```
mysql> SELECT YEAR('2012-04-19');
+--------------------+
| YEAR('2012-04-19') |
+--------------------+
|               2012 |
+--------------------+
```

- HOUR(TIME)

  Menghasilkan bagian jam dari waktu yang menjadi argumen.

Contoh :

MINUTE(TIME)

```
mysql> SELECT HOUR('10:20:11');
+------------------+
| HOUR('10:20:11') |
+------------------+
|               10 |
mysql> SELECT MINUTE('10:20:11');
+--------------------+
| MINUTE('10:20:11') |
+--------------------+
|                 20 |
+--------------------+
```

Menghasilkan bagian menit dari waktu yang menjadi argumen.

Contoh :

- SECOND(TIME)

  Menghasilkan bagian detik dari waktu yang menjadi argumen.

  Contoh :

```
mysql> SELECT SECOND('10:20:11');
+--------------------+
| SECOND('10:20:11') |
+--------------------+
|                 11 |
+--------------------+
```

3. Fungsi Agregat

- SUM (nama_field)

  Mendapatkan nilai total dari suatu kolom dalam sebuah tabel atau ekspresi.

```
mysql> SELECT SUM(HARGA_BELI) FROM BARANG;
+-----------------+
| SUM(HARGA_BELI) |
+-----------------+
|           69500 |
+-----------------+
```

- AVG (nama_field)

  Untuk mendapatkan nilai rata-rata sebuah kolom pada tabel atau ekspresi.

```
mysql> SELECT AVG(HARGA_BELI) FROM BARANG;
+-----------------+
| AVG(HARGA_BELI) |
+-----------------+
|       7722.2222 |
+-----------------+
```

- MAX (nama_field)

  Untuk mendapatkan nilai maksimum dari sebuah kolom dalam tabel atau ekspresi.

```
mysql> SELECT MAX(HARGA_BELI) FROM BARANG;
+-----------------+
| MAX(HARGA_BELI) |
+-----------------+
|           16000 |
+-----------------+
```

◉ MIN (nama_field)

Untuk mendapatkan nilai minimum sebuah kolom pada tabel atau ekspresi.

```
mysql> SELECT MIN(HARGA_BELI) FROM BARANG;
+-----------------+
| MIN(HARGA_BELI) |
+-----------------+
|            2000 |
+-----------------+
```

◉ COUNT (X)

Untuk menghitung jumlah record dari suatu kolom atau tabel X.

```
mysql> SELECT *FROM BARANG;
+-----+-------------+--------------+-------------+-------------+------+
| id  | nama_barang | jenis_barang | harga_beli  | harga_jual  | stok |
+-----+-------------+--------------+-------------+-------------+------+
| B01 | RINSO       | DETERGEN     |       11000 |       12000 |   10 |
| B02 | DAIA        | DETERGEN     |        9000 |       10000 |   20 |
| B03 | LIFEBUOY    | SABUN MANDI  |        2500 |        4000 |   30 |
| B04 | LIFEBUOY    | SHAMPOO      |       12000 |       14000 |   20 |
| B05 | LUX         | SABUN MANDI  |        2000 |        3000 |   12 |
| B06 | DETTOL      | SABUN MANDI  |        3000 |        4000 |   15 |
| B07 | CITRA       | BODY LOTION  |       12000 |       15000 |   16 |
| B08 | CITRA       | SABUN MANDI  |        2000 |        2500 |   25 |
| B09 | PANTENE     | SHAMPOO      |       16000 |       17000 |   30 |
| B10 | PEPSODENT   | PASTA GIGI   |        9000 |       11000 |   19 |
+-----+-------------+--------------+-------------+-------------+------+
10 rows in set (0.00 sec)

mysql> SELECT COUNT(*) FROM BARANG;
+----------+
| COUNT(*) |
+----------+
|       10 |
+----------+
```

4. Fungsi Aritmatika

◉ PENJUMLAHAN (+)

Untuk menjumlahkan dua buah data numerik.

◉ Pengurangan (-)

Untuk mengurangkan dua buah data numerik.

◉ Perkalian (x)

Untuk melakukan perkalian dua buah data numerik.

◉ Pembagian (/)

Untuk melakukan pembagian dua buah data numerik.

◉ Pembagian Sisa (%)

Untuk mendapatkan sisa pembagian dari suatu operasi pembagian bilangan numerik.

CONTOH :

Fungsi Sistem

```
mysql> SELECT 24+2 AS JUMLAH, 24-2 AS SELISIH, 24*2 AS PERKALIAN,
    -> 24/2 AS PEMBAGIAN, 24%5 AS MODULUS;
+--------+---------+-----------+-----------+---------+
| JUMLAH | SELISIH | PERKALIAN | PEMBAGIAN | MODULUS |
+--------+---------+-----------+-----------+---------+
|     26 |      22 |        48 |   12.0000 |       4 |
+--------+---------+-----------+-----------+---------+
```

- DATABASE()

  Mengetahui database yang sedang digunakan.

- LAST_INSERT_ID()

  Menampilkan info data terakhir yang digenerate oleh MySQL pada kolom bertipe AUTO INCREMENT (AI).

- SESSION_USER

  Menampilkan informasi pemakai yang sedang melakukan akses ke database.

LATIHAN

1. Tampilkan string "Praktikum Pemrograman" yang telah diubah menjadi string "Prakt Pemrog".
2. Tampilkan string "gara Bar" dalam string "Nusa Tenggara Barat".
3. Tampilkan posisi string "stmik" dalam string "Nama kampus stmik Bumigora".
4. Tampilkan Tanggal dan Waktu dengan format sbb :
   nm_hari, bln tgl thn jam:menit:detik AM/PM
   Contoh : Saturday, April 21st 12 02:15:40 PM
6. Tampilkan string "Sistem" dalam string "Sistem Basis Data"
7. Dari tabel barang yang sudah dibuat pada tugas sebelumnya, tampilkan jumlah stok yang paling sedikit (gunakan fungsi)

MODUL 7
JOIN WITHOUT JOIN STATEMENT

Dalam basis data relasional dimungkinkan untuk mengakses satu atau lebih tabel dalam waktu yang bersamaan. Penggunaan dua tabel atau lebih dalam satu buah baris query biasa disebut dengan JOIN.

Query :

```
SELECT <tabel1.field1>,<tabel2.field2> FROM <tabel1>, <tabel2> WHERE
<key.tabel1> = <key.tabel2>
```

LATIHAN

Database : KULIAH

Tabel : MAHASISWA, DOSEN

Tabel MAHASISWA

Field	Tipedata
NIM	CHAR (PK)
NAMA	CHAR
ALAMAT	CHAR
IPK	FLOAT (10,2)
ID_DOSEN	CHAR

NIM	NAMA	ALAMAT	IPK	ID_DOSEN
123070201	LALU HERMAN	JL MAWAR NO 11	3.01	12346
123070202	KURNIAWAN	JL AFFANDI NO 12	2.75	12344
123070203	INDRA KUSUMA	JL DEMANGAN NO 4	2.83	12345
123070204	KARMAN MAULANA	JL BABARSARI NO 23	2.91	12343
123070205	RIZAD RAHMAN	JL KAPAS NO 8	2.5	12344
123070206	WAWAN ADI PUTRA	JL KLEDOKAN NO 2	3.21	12341
123070207	M TAUFIK HIDAYAT	JL TAMBAKBAYAN NO 3	3.11	12341
123070208	FITRIADI BUDIMAN	JL MERPATI NO 24	3.41	12344
123070209	IDA KUSUMAWATI	JL BANTUL NO 15	3.32	12343
123070210	HIDAYAT NUGRAHA	JL PASIFIK NO 6	2.85	12346

Tabel DOSEN

Field	Tipedata
ID_DOSEN	CHAR (PK)
NAMA	CHAR
ALAMAT	CHAR
JABATAN	CHAR
NOTELP	CHAR

ID_DOSEN	NAMA	ALAMAT	JABATAN	NOTELP
12341	HERRY SOFIAN, M.KOM	JL ELANG NO 11	LEKTOR	08123456789
12342	HERU CAHYA, MT	JL AFFANDI NO 17	ASS AHLI	08123456788
12343	PAULUS INSAP, Phd	JL RINGROAD NO 89	LEKTOR	08123456787
12344	NOVRIDO, MT	JL ADI SUCIPTO NO 8	LEKTOR	08123456786
12345	AGUS SASMITO, ST	JL MERAPI NO 23	ASS AHLI	08123456785

- Menampilkan isi tabel DOSEN melalui tabel MAHASISWA.

```
SELECT * FROM dosen a, mahasiswa b WHERE a.id_dosen = b.id_dosen;
```

- Menampilkan isi tabel DOSEN dan MAHASISWA yang memiliki ID_DOSEN yang **sama.**

```
SELECT b.nim, b.nama, a.id_dosen, a.nama FROM dosen a, mahasiswa b WHERE
a.id_dosen = b.id_dosen;
```

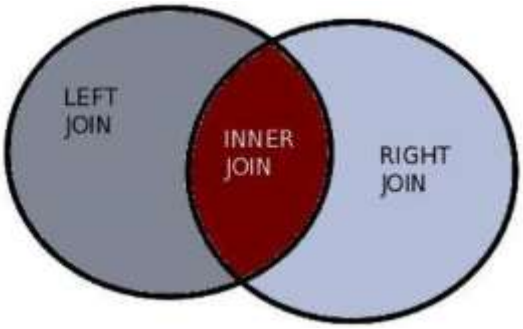
- Menampilkan isi tabel DOSEN dan MAHASISWA yang memiliki ID_DOSEN yang sama = ‘12344’.

```
SELECT b.nim, b.nama, a.id_dosen, a.nama, FROM dosen a, mahasiswa b
WHERE a.id_dosen = b.id_dosen AND a.id_dosen = ‘12344’;
```

MODUL 8
INNER JOIN DAN OUTER JOIN

Di dalam database, ada kalanya kita membutuhkan data dari beberapa tabel yang saling berhubungan. Untuk mendapatkan data dari beberapa tabel tersebut dapat digunakan perintah join pada perintah SQL.

Gambaran hasil join :



1. INNER JOIN

   Inner join merupakan jenis join yang paling umum yang dapat digunakan pada semua database. Jenis ini dapat digunakan bila ingin merelasikan dua set data yang ada di tabel, letak relasinya setelah pada perintah **ON** pada join. Hasil dari inner join adalah gabungan kedua tabel yang memiliki data join yang sama.

   Query :

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> INNER JOIN <tabel2>
ON <key.tabel1> = <key.tabel2>
```

2. OUTER JOIN

   Outer join merupakan join yang sedikit berbeda dengan inner join. Outer join akan menghasilkan record-record yang ada baik ada pasangan nya pada tabel lain atau tidak.

   Outer join dibedakan menjadi 2 jenis, yaitu :

   a. Left Outer Join

      Left join digunakan dalam situasi ketika ingin mengembalikan semua elemen data set A, terlepas dari apakah nilai kunci ada dalam data set B.

      Query :

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> LEFT JOIN <tabel2> ON
<key.tabel1> = <key.tabel2>
```

b. Right Outer Join

Right join digunakan dalam situasi ketika ingin mengembalikan semua elemen data set B, terlepas dari apakah nilai kunci ada dalam data set A.

Query :

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> RIGHT JOIN <tabel2>
ON <key.tabel1> = <key.tabel2>
```

Latihan

Dengan menggunakan data yang sama pada bab sebelumnya :

- Menampilkan isi tabel DOSEN melalui tabel MAHASISWA.

```
SELECT a.nama, a.id_dosen, b.id_dosen, b.nama FROM mahasiswa a INNER
JOIN dosen b ON a.id_dosen = b.id_dosen;
```

- Menampilkan isi tabel DOSEN dan MAHASISWA dengan LEFT JOIN.

```
SELECT a.nama, a.id_dosen, b.id_dosen, b.nama FROM mahasiswa a LEFT JOIN
dosen b ON a.id_dosen = b.id_dosen;
```

- Menampilkan isi tabel DOSEN dan MAHASISWA dengan RIGHT JOIN.

```
SELECT a.nama, a.id_dosen, b.id_dosen, b.nama FROM mahasiswa a RIGHT
JOIN dosen b ON a.id_dosen = b.id_dosen;
```

MODUL 9
CROSS JOIN DAN UNION JOIN

1. Cross Join

   Cross join kadangkala disebut juga sebagai **Cartesian Product**. Bila menggunakan cross join, maka hasil dari cross join akan menciptakan hasil yang didasarkan pada semua kemungkinan kombinasi baris dalam kedua set data. Dengan demikian, jumlah baris yang dikembalikan adalah N × M , dimana N adalah jumlah baris dalam kumpulan data A dan M jumlah baris dalam kumpulan data B. Jelas, jumlah baris dalam cross join dapat menjadi sampah.

   Query :

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> CROSS JOIN <tabel2>
```

   atau

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1>, <tabel2>
```

2. Union Join

   Union didukung oleh MySQL mulai dari versi 4.0. Pemakaian union dapat menyederhanakan perintah persyaratan **OR** yang bertingkat. Bila dalam sebuah query menghasilkan pemakaian perintah **OR** yang lebih dari satu sehingga dapat membuat bingung, sebagai gantinya digunakan perintah **UNION**. Union dapat dikatakan sebagai perintah untuk menggabungkan hasil query sql yang fungsinya sama dengan perintah OR.

   Query :

```
(SELECT <tabel1.field1>,<tabel2.field2> FROM <tabel1> JOIN <tabel2> ON
<key.tabel1> = <key.tabel2> WHERE key1 = 'Value1') UNION
(SELECT <tabel1.field1>,<tabel2.field2> FROM <tabel1> JOIN <tabel2> ON
<key.tabel1> = <key.tabel2> WHERE key2 = 'Value2')
```

Latihan

- Menampilkan tabel mahasiswa dan dosen dengan menggunakan CROSS JOIN

```
SELECT a.nama, a.id_dosen, b.id_dosen, b.nama FROM mahasiswa a CROSS JOIN
dosen b
```

- Menampilkan tabel mahasiswa dan dosen dengan menggunakan UNION JOIN

  Cara 1:

```
SELECT a.nama, a.id_dosen, b.id_dosen, b.nama FROM mahasiswa a JOIN dosen
b ON a.id_dosen = b.id_dosen WHERE a.id_dosen='12341' OR a.id_dosen
='12343';
```

  UNION :

```
(SELECT a.nama, a.id_dosen, b.id_dosen, b.nama FROM mahasiswa a JOIN
dosen b ON a.id_dosen = b.id_dosen WHERE a.id_dosen='12341')    UNION
(SELECT a.nama, a.id_dosen, b.id_dosen, b.nama FROM mahasiswa a JOIN
dosen b ON a.id_dosen = b.id_dosen WHERE a.id_dosen='12343');
```