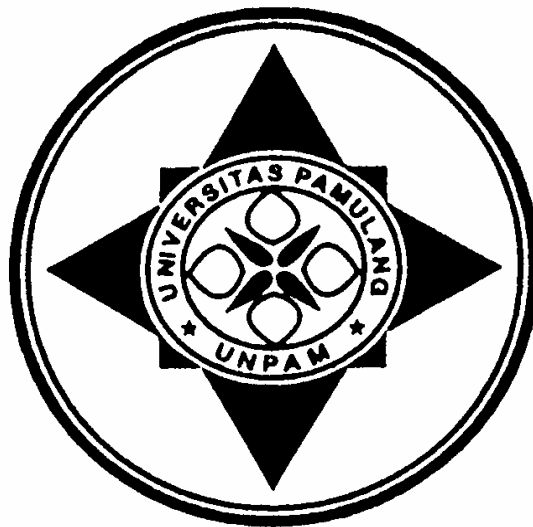


# **MODUL PRAKTIKUM BASIS DATA**

**REVISI KEDUA**



**Disusun Oleh:**

**Shandi Noris, M.Kom**

**Emi Sita Eriana**

**TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PAMULANG**

**Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566  
Tangerang Selatan – Banten**

# **TATA TERTIB PRAKTIKUM**

- 1) Peserta praktikum sudah terdaftar sebagai mahasiswa Teknik Informatika Universitas Pamulang pada semester bersangkutan dengan nama sudah tertera pada lembar kehadiran dan bersedia menjalankan tata tertib yang sudah ditentukan.
- 2) Praktikum dilaksanakan sesuai dengan jadwal yang telah ditentukan.
- 3) Peserta praktikum wajib hadir tepat pada waktunya.
- 4) Peserta praktikum diharuskan menandatangani lembar kehadiran.
- 5) Semua peserta praktikum harus memiliki modul praktikum.
- 6) Selama praktikum berlangsung diwajibkan :
  - Memelihara suasana agar nyaman dan tenang
  - Tidak Membawa makanan dan minuman
  - Tidak Merokok dan mengotori ruangan
  - Tidak Bersuara dengan keras
  - Tidak Hilir mudik yang tidak perlu
  - Tidak Bertindak atau berbicara yang tidak ada hubungannya dengan kegiatan praktikum
  - Tidak diperkenankan bermain Game
  - Tidak mencoret-coret sarana yang ada
  - Tidak merubah, merusak, atau mengambil peralatan di Laboratorium
  - Meletakan tas ditempat yang ditentukan oleh Pengajar
- 7) Pakaian yang dikenakan peserta praktikum sebagai berikut:
  - Pria
    - Celana Panjang, tidak robek
    - Sepatu Tertutup (bukan sepatu sandal)
    - Kemeja (Bukan kaos berkerah atau almamater atau sejenisnya)
  - Wanita
    - Kemeja (Bukan kaos berkerah atau almamater atau sejenisnya)
    - Sepatu tertutup (bukan Sepatu sandal)
    - Celana Panjang/Rok

## **Sanksi-Sanksi :**

- 1) Jika Keterlambatan lebih dari 20 menit tanpa alasan yang dapat diterima, maka peserta praktikum tidak diijinkan mengikuti praktikum.
- 2) Jika peserta praktikum merusak peralatan praktikum akibat melakukan perbuatan yang tidak berkaitan dengan praktikum/prosedur praktikum maka yang bersangkutan harus mengganti kerugian akibat perbuatannya.
- 3) Jika Tata tertib tidak ditaati maka peserta praktikum dapat diberi sanksi tegas serta dapat diberi nilai E (Tidak Lulus).

# **PERTEMUAN I**

## **PENGANTAR**

### **TUJUAN PRAKTIKUM**

- a) Peserta dapat melaksanakan praktikum, membuat laporan praktikum dengan benar dan mengikuti tata tertib dalam melaksanakan praktikum.

### **ATURAN PRAKTIKUM**

#### **a) Mengikuti Tata tertib praktikum**

#### **b) Penilaian :**

- 1) Kehadiran : 30%
  - Minimal kehadiran adalah 11 kali (ujian praktikum termasuk kehadiran). Jika kurang dari 11 kali maka peserta praktikum dinyatakan tidak lulus.
  - Absen Jalan diperbolehkan jika disertai alasan yang kuat (seperti surat dokter, surat dinas, atau sejenisnya) dengan batas maksimal 3 kali. Lembar Absen Jalan diberikan kepada pengajar pada saat Ujian Praktikum.
- 2) Tugas/Laporan Praktikum : 35%

Nilai Tugas/Laporan terdiri dari: tugas/quiz, laporan awal, dan laporan akhir.
- 3) Ujian Praktikum: 35%

Ujian praktikum dilaksanakan pada pertemuan ke-14 atau pertemuan akhir.
- 4) Salah satu atau lebih dari komponen nilai di atas tidak ada, maka peserta praktikum dinyatakan tidak lulus.

#### **c) Membuat Laporan Praktikum.**

- 1) Laporan Praktikum terdiri dari Laporan Awal dan Laporan Akhir.
- 2) Laporan Awal
  - Isi dari laporan awal adalah menulis ulang setiap Modul sesuai dengan pertemuan berikutnya, dan menjawab soal Tugas Pendahuluan.
- 3) Laporan Akhir
  - Isi dari laporan akhir adalah Menulis kesimpulan pada Modul yang telah dipraktekkan dan menjawab soal Tugas Akhir.
- 4) Tata Cara Mengumpulkan Laporan
  - Laporan dikumpulkan mulai pertemuan ke-2.

- Pada pertemuan ke-2, hanya laporan awal yang dikumpulkan.
- Pada pertemuan ke-3 sampai dengan pertemuan ke-13, laporan yang dikumpulkan adalah laporan awal dan laporan akhir.
- Pada pertemuan ke-14 atau pertemuan akhir atau pada saat ujian praktikum, hanya laporan akhir yang dikumpulkan.

Contoh:

- Pada Pertemuan ke-2, peserta praktikum harus membawa laporan awal yang isinya adalah penulisan ulang modul pertemuan ke-2 dan menjawab soal tugas pendahuluan pertemuan ke-2.
- Pada Pertemuan ke-3, peserta praktikum harus membawa laporan awal yang isinya adalah penulisan ulang modul pertemuan ke-3 dan menjawab soal tugas pendahuluan pertemuan ke-3, serta membawa laporan akhir pertemuan ke-2 dan menjawab soal tugas akhir pertemuan ke-2.
- Pada pertemuan ke-4 sampai dengan pertemuan ke-13, cara mengumpulkan laporannya sama dengan pada pertemuan ke-3.
- Pada pertemuan ke-14 atau pada saat ujian praktikum, peserta praktikum hanya membawa Laporan akhir pertemuan ke-13 dan menjawab soal tugas akhir pertemuan ke-13.
- Contoh jika peserta praktikum Tidak Hadir:
  - Jika pada pertemuan ke-4 peserta praktikum tidak hadir dikarenakan sakit dan pada pertemuan ke-5 peserta praktikum hadir, maka peserta praktikum harus membawa laporan awal yang isinya adalah penulisan ulang modul pertemuan ke-5 dan menjawab soal tugas pendahuluan pertemuan ke-5, serta membawa laporan akhir pertemuan ke-3 dan menjawab soal tugas akhir pertemuan ke-3.

##### 5) Format laporan praktikum :

- Isi Laporan ditulis tangan pada kertas A4.
- Halaman depan Laporan Awal dan Laporan Akhir boleh diprint atau ditulis tangan, contoh seperti gambar di bawah ini.

Contoh halaman Depan (diPrint) :

# **LAPORAN AWAL BASIS DATA**

**LAPORAN KE-1**



**Disusun Oleh :**

Nama : Cecep Gorbacep

NIM : 123456789

Kelas : V-A Malam

**TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PAMULANG**

Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566  
Tangerang Selatan – Banten

Contoh halaman Depan (diPrint) :

# LAPORAN AKHIR BASIS DATA

## LAPORAN KE-1



Disusun Oleh :

Nama : Cecep Gorbacep

NIM : 123456789

Kelas : V-A Malam

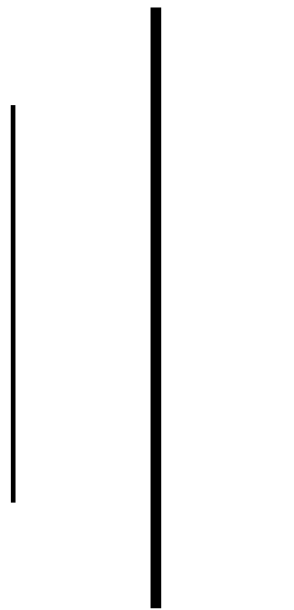
**TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PAMULANG**

Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566  
Tangerang Selatan – Banten

Contoh halaman Depan (diTulis) :

# **LAPORAN AWAL BASIS DATA**

## **LAPORAN KE-1**



**Disusun Oleh :**

Nama : Cecep Gorbacep

NIM : 123456789

Kelas : V-A Malam

**TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PAMULANG**

Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566  
Tangerang Selatan – Banten

## PERTEMUAN II

### ADMINISTRASI BASIS DATA MYSQL

#### TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan dan menggunakan Software Basis Data yaitu MySQL.
- b) Mahasiswa dapat membuat dan menghapus user, memberi dan mencabut privileges user, mengubah data user (seperti mengganti password dan namanya), mengatur dari mana user dapat mengakses MySQL Server.

#### a) Pendahuluan

MySQL merupakan salah satu aplikasi standar dalam pengembangan web, bersama dengan Web Server Apache, dan bahasa pemrograman PHP. MySQL adalah salah satu aplikasi RDBMS (Relational Database Management System). Keunggulan MySQL dibandingkan dengan RDBMS lainnya diantaranya MySQL memiliki kecepatan, dapat memproses data yang sangat besar dan kompleks, tanpa ada penurunan performa instalasi.

#### b) Pengaksesan dan Privileges user

Untuk mengatur hak akses di MySQL, pada dasarnya kita menggunakan bentuk perintah Grant Dan Revoke. Bentuk umum perintah GRANT dan REVOKE secara sederhana :

```
GRANT priv_type
ON {tbl_name | * | *.* | db_name.*}
TO user_name [IDENTIFIED BY 'password'] [WITH GRANT OPTION]

REVOKE priv_type
ON {tbl_name | * | *.* | db_name.*}
FROM user_name
```

Perintah Grant Dan Revoke dapat digunakan untuk membuat user baru maupun mengatur hak akses user yang sudah ada dengan hak akses (privileges) tertentu. Tingkatan hak akses user dapat terbagi menjadi tingkatan global (tersimpan di table mysql.user), database (tersimpan di table mysql.host dan mysql.db), table (tersimpan di table mysql.tables\_priv) dan kolom (tersimpan di table mysql.columns\_priv). Setiap perubahan hak akses di MySQL, termasuk menambahkan user baru, tidak akan berlaku sebelum



diakhiri dengan perintah Flush Privileges. Pernyataan privileges digunakan agar MySQL membaca ulang database mysql dan semua perubahan yang ada di dalamnya tanpa me-restart ulang server MySQL, sebelum kita menggunakan perintah Flus privileges atau me-restart Server MySQL.

Hak hak MySQL yang dapat diberikan kepada user, yaitu:

<b>Hak</b>	<b>Keterangan</b>
ALL	Memberikak akses penuh kepada seorang user kecuali perintah GRANT
ALTER	Mengizinkan user untuk menggunakan perintah ALTER TABLE
ALTER ROUTINE	Mengizinkan user untuk mengubah struktur (schema) dari sebuah fungsi tersimpan dan prosedur tersimpan.
CREATE	Mengizinkan user untuk menggunakan perintah CREATE TABLE
CREATE USER	Mengizinkan user untuk membuat fungsi tersimpan perintah CREATE USER,DROP USER,RENAME USER,dan REVOKE ALL PRIVILEGES
CREATE TEMPORARY TABLES	Mengizinkan user untuk menggnakan perintah CREATE TEMPORARY TABLES(HEAP)
CREATE VIEW	Mengizinkan User Untuk menggunakan perintah CREATE VIEW
DELETE	Mengizinkan User Untuk menggunakan perintah DELETE
DROP	Mengizinkan User Untuk menggunakan perintah DROP TABLE
EVENT	Mengizinkan User Untuk membuat penjadwalan perintah SQL.Hak ini terdapat pada MySQL versi 5.1.6 keatas.
EXECUTE	Mengizinkan User Untuk menjalankan fungsi tersimpan dan prosedur tersimpan.
FILE	Mengizinkan user untuk menggunakan perintah SELECT....INTO OUTFILE,LOAD DATA INFILE dan proses replikasi.
INDEX	Mengizinkan User untuk menggunakan perintah CREATE INDEX dan DROP INDEX.
INSERT	Mengizinkan User untuk mennggunakan perintah INSERT.
LOCK TABLES	Mengizinkan user untuk mengunakan perintah LOCK TABLES

PROCESS	Mengizinkan user untuk menggunakan perintah SHOW FULL PROCESSLIST.
RELOAD	Mengizinkan user untuk menggunakan perintah FLUSH.
REPLICATION CLIENT	Mengizinkan user untuk mendapatkan informasi komputer Master atau Slave.
REPLICATION SLAVE	Mengizinkan user dari komputer Slave untuk membaca file binary Log pada computer Master untuk proses replikasi.
SELECT	Mengizinkan user untuk menggunakan perintah SELECT.
SHOW DATABASES	Mengizinkan user untuk menggunakan perintah SHOW DATABASES.
SHOW VIEW	Mengizinkan user untuk menggunakan perintah SHOW CREATE VIEW.
SHUTDOWN	Mengizinkan user untuk menghentikan server MySQL melalui aplikasi mysqladmin
SUPER	Mengizinkan user untuk menggunakan perintah CHANGE MASTER,KILL,PURGE MASTER LOGS,dan SET GLOBAL,perintah dari aplikasi mysqladmin debug,dan mengizinkan untuk melakukan koneksi pada Server MySQL meskipun nilai dari variable max_connections dilampaui.
TRIGGER	Mengizinkan user untuk membuat dan menghapus sebuah objek trigger. Hak ini terdapat pada MySQL versi 5.1.6 keatas.
UPDATE	Mengizinkan user untuk menggunakan perintah UPDATE
USAGE	Mengizinkan user untuk login dalam server MySQL tetapi tanpa hak apapun (hanya login)
GRANT OPTION	Mengizinkan user untuk menggunakan perintah GRANT

## TUGAS PRAKTIKUM

### a) Mengakses MySQL

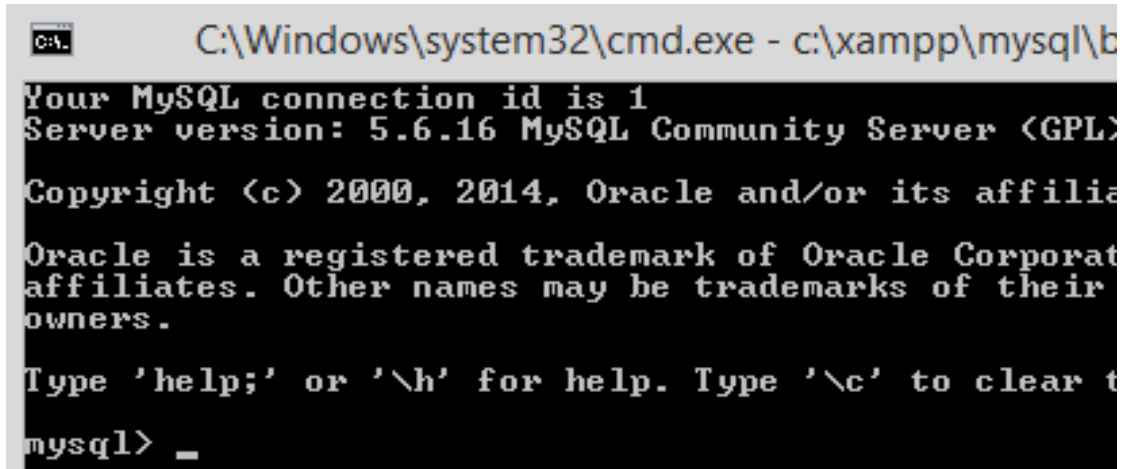
- 1) Akses masuk Ke Directory pada Command Prompt

Klik Start → Run → ketik cmd <enter>

Ketik perintah di bawah ini

```
C:> cd c:\xampp\mysql\bin <enter>
C:\xampp\mysql\bin> mysql -u root
```

Kemudian akan terlihat gambar di bawah ini



```
C:\Windows\system32\cmd.exe - c:\xampp\mysql\bin
Your MySQL connection id is 1
Server version: 5.6.16 MySQL Community Server (GPL)
Copyright (c) 2000, 2014, Oracle and/or its affiliates
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\\h' for help. Type '\\c' to clear the current display.
mysql> _
```

- 2) Untuk Keluar dari mysql

Gunakan perintah exit atau quit

```
mysql> exit
atau
mysql> quit
```

## b) Membuat Dan Menghapus User

- 1) Membuat user

Ketik perintah di bawah ini. (membuat user dengan nama praktikan, password 12345678)

```
mysql> create user praktikan@localhost identified by '12345678';
```

- 2) Menampilkan user yang telah dibuat

Ketik perintah di bawah ini.

```
mysql> select user from mysql.user;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select user from mysql.user;
+-----+
| user      |
+-----+
| root      |
| root      |
| pma       |
| praktikan |
| root      |
+-----+
6 rows in set (0.00 sec)
```

3) Menghapus user

Ketik perintah di bawah ini. (menghapus user dengan nama praktikan)

```
mysql> drop user praktikan@localhost;
```

4) Menampilkan user yang telah dihapus

Gunakan perintah select user from mysql.user;

c) Hak akses User

1) Buat kembali user dengan nama praktikan, password 12345678

2) Membuat hak akses untuk user praktikan (hanya insert dan delete)

Ketik perintah di bawah ini

```
mysql> grant insert,delete on *.* to praktikan identified by '12345678';
```

3) Membuat hak akses untuk user praktikan (hak penuh)

Ketik perintah di bawah ini

```
mysql> grant all on *.* to praktikan identified by '12345678';
```

4) Menampilkan hak akses untuk user

Ketik perintah di bawah ini

```
mysql> show grants for praktikan
```

Kemudian akan terlihat gambar di bawah ini

```

mysql> show grants for praktikan;
+-----+
| Grants for praktikan@%                |
+-----+
| GRANT INSERT, DELETE ON *.* TO 'praktikan'@'%' IDENTIFIED BY '2F54AB666ECFC2A83C676908C8BBC381B1' |
+-----+
1 row in set (0.00 sec)

```

5) Menghapus hak akses untuk user

Ketik perintah di bawah ini:

```

mysql> revoke all on *.* from praktikan;
mysql> flush privileges;

```

#### d) Mengganti Password User

Ketik perintah di bawah ini. (mengganti password user praktikan)

```

mysql> set password for 'praktikan'@'localhost'=password ('qwerty');
mysql> flush privileges;

```

#### e) Buatlah User dan password berdasarkan table di bawah ini

NO	User	Password
1	unpam	mencoba11
2	mahasiswa	password12
3	kuliah	semangat33
4	universitas	myunpam24
5	informatika	jurusan123

#### f) Dari point e) di atas, ubah password berdasarkan table di bawah ini

NO	User	Password Lama	Password Baru
1	unpam	mencoba11	latihan99
2	mahasiswa	password12	unpam12
3	kuliah	semangat33	latihan56
4	universitas	myunpam24	Pamulang67
5	informatika	jurusan123	Teknik123

**g) Dari point f) di atas, ubah hak akses berdasarkan table di bawah ini**

<b>NO</b>	<b>User</b>	<b>Hak Akses</b>
1	unpam	Create
2	mahasiswa	Alter
3	kuliah	Insert
4	universitas	Update
5	informatika	Usage

### **TUGAS PENDAHULUAN**

1. Jelaskan perbedaan dari RDBMS dan DBMS !
2. Bagaimana user melakukan pengaksesan dalam MySQL?
3. Apa yang dimaksud dengan Privileges user, Grant dan Revoke ?
4. Jelaskan fungsi dari perintah flush Priveleges !

### **TUGAS AKHIR**

1. Buatlah kesimpulan praktikum di atas !

## **PERTEMUAN III**

### **DATA DEFINITION LANGUAGE (Bagian-1)**

#### **TUJUAN PRAKTIKUM**

- a) Mahasiswa dapat mengenal dan memahami Data Definition Language pada bahasa pemrograman SQL.
- b) Mahasiswa dapat membuat, menampilkan, dan menghapus struktur yang ada di dalam basis data pada MySQL.

#### **TEORI DASAR**

##### **a) Pendahuluan**

DDL merupakan perintah-perintah yang biasa digunakan administrator database untuk mendefinisikan skema dan subskema database. Data Definition Language (DDL) mempunyai fungsi utama untuk mendefinisikan data dalam database secara logika, diantaranya yaitu: Digunakan untuk mendefinisikan karakteristik dari record (meliputi nama, tipe dan lebar dari field), untuk menentukan kunci field, menyediakan cara untuk menentukan hubungan dengan data di file lain, untuk mengubah struktur dari record, untuk menampilkan struktur dari record. DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya table. Perintah yang termasuk DDL:

- 1) CREATE : untuk membuat, termasuk diantaranya membuat database dan table baru.
- 2) ALTER: untuk mengubah struktur table yang telah dibuat.
- 3) RENAME: Perintah Rename digunakan untuk mengubah nama table atau objek database.
- 4) DROP : untuk menghapus database dan table.

##### **b) Database**

Basis data (database) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. Database merupakan

salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau user. Penyusunan basis data meliputi proses memasukkan data ke dalam media penyimpanan data dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (Database Management System DBMS). Manipulasi basis data meliputi pembuatan pernyataan (query) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (update) data, serta pembuatan report data.

**c) Table**

Kumpulan record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.

**d) Field**

Merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari field membentuk suatu record. Record adalah kumpulan dari field membentuk suatu record. Record menggambarkan suatu unit data individu yang tertentu. Kumpulan dari record membentuk suatu file. Misalnya file personalia, tiap record dapat mewakili data tiap karyawan. Field name harus diberi nama untuk membedakan field yang satu dengan lainnya dan memiliki isi dari field untuk masing-masing (field value). Sedangkan field representation adalah tipe field (karakter, teks, tanggal, angka, dsb), lebar field (ruang maksimum yang dapat diisi dengan karakter-karakter data).

**e) Primary Key dan Foreign Key**

Primary key merupakan suatu field atau kombinasi field yang secara unik mengidentifikasi sebuah record sekaligus membedakannya dengan record yang lain, digunakan untuk membedakan 1 record dengan record lainnya. Syaratnya tidak boleh sama dan harus diisi. atau juga Primary Key adalah field kunci / utama dari suatu table yang menunjukkan bahwa field yang menjadi kunci tersebut tidak bisa diisi dengan data yang sama, / dengan kata lain Primary key menjadikan tiap record memiliki identitas sendiri-sendiri yang membedakan satu sama lainnya (unik). Tujuan utama dari adanya kedua istilah tersebut adalah pengidentifikasian tiap table dan kejelasan hubungan antara duani atau lebih table. Primary key adalah satu atau beberapa kolom pada table yang mengidentifikasikan tiap kolom dan baris pada table tersebut. Sedangkan Foreign Key



adalah satu atau beberapa kolom pada table yang merupakan primary key pada table satu namun diletakan pada table dimana tablenya berelasi dengan table dirinya.

#### f) Tipe-tipe Data

##### 1) Tipe data Numeric

Tipe data ini digunakan untuk menyimpan data berupa numeric (angka). Tipe ini dibagi menjadi 2, yaitu :

##### a. Bilangan Bulat (Integer)

<b>Tipe data</b>	<b>Keterangan</b>
TINYINT	Digunakan untuk menyimpan data bilangan bulat positif dan negatif. Ukuran : 1 bytes Signed : Minimum value = -128 , Maximum value =127 Unsigned : Minimum value = 0 , Maximum value =255
SMALLINT	Digunakan untuk menyimpan data bilangan bulat positif dan negatif. Ukuran : 2 byte Signed : Minimum value = -32.768, Maximum value =32.767 Unsigned : Minimum value = 0 , Maximum value = 65.535
MEDIUMINT	Digunakan untuk menyimpan data bilangan bulat positif dan negatif Ukuran : 3 bytes Signed : Minimum value = -8.388.608, Maximum value =8.388.607 Unsigned : Minimum value = 0 , Maximum value = 16.777.215
INT	Digunakan untuk menyimpan data bilangan bulat positif dan negatif. Ukuran : 4 bytes Signed : Minimum value = -2.147.483.648, Maximum value = 2.147.483.647 Unsigned : Minimum value = 0 , Maximum value = 4.294.967.295

b. Bilangan Pecahan (Integer)

<b>Tipedata</b>	<b>keterangan</b>
BIGINT	<p>Digunakan untuk menyimpan data bilangan bulat positif dan negatif.</p> <p>Ukuran : 8 bytes</p> <p>Signed : Minimum value = -9.223.372.036.854.775.808, Maximum value = 9.223.372.036.854.775.807</p> <p>Unsigned : Minimum value = 0 , Maximum value 184.467.440.737.079.551.615</p>
FLOAT	<p>Digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi tunggal.</p> <p>Ukuran : 4 bytes</p> <p>Jangkauan = -3.402823466E+38 s/d -1.175494351E-38, 0 dan 1.175494351E-38 s/d 3.402823466E+38.</p>
DOUBLE	<p>Digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi ganda.</p> <p>Ukuran : 8 bytes</p> <p>Jangkauan = -1.7976931348623157E+308 s/d - 2.2250738585072014E-308, 0, dan 2.2250738585072014E-308 s/d 1.7976931348623157E+308.</p>
DECIMAL	<p>Digunakan untuk menyimpan data bilangan pecahan positif dan negatif.</p> <p>Ukuran : 8 Bytes</p> <p>Jangkauan = -1.7976931348623157E+308 s/d - 2.2250738585072014E-308, 0,dan 2.2250738585072014E-308 s/d 1.7976931348623157E+308</p>

2) Tipe Data String (Text) dan Tipe Blob(Biner)

Digunakan untuk menyimpan data berupa tulisan atau sebuah artikel

<b>Tipe Data</b>	<b>Keterangan</b>
CHAR	Penggunaan : digunakan untuk menyimpan data string ukuran tetap Jangkauan : 0 s/d 255 karakter
VARCHAR	Penggunaan : digunakan untuk menyimpan data string ukuran dinamis. Jangkauan : 0 s/d 255 karakter (versi 4.1), 0 s/d 65.535 (versi 5.0.3)
TINYTEXT	Penggunaan : digunakan untuk menyimpan data text. jangkauan : 0 s/d 255 karakter (versi 4.1), 0 s/d 65.535 (versi 5.0.3)
TEXT	Penggunaan : digunakan untuk menyimpan data text. Jangkauan : 0 s/d 65.535 (216 – 1) karakter
LONGTEXT	Penggunaan : digunakan untuk menyimpan data text. Jangkauan : 0 s/d 232 – 1 karakter
BIT (sejak versi 5.0.3)	Penggunaan : digunakan untuk menyimpan data biner. Jangkauan : 64 digit biner
TINYBLOB	Penggunaan : digunakan untuk menyimpan data biner. Jangkauan : 255 byte
MEDIUMBLOB	Penggunaan : digunakan untuk menyimpan data biner. Jangkauan : 224 – 1 byte
LOB	Penggunaan : digunakan untuk menyimpan data biner. Jangkauan : 232 – 1 byte

### 3) Tipe Data Date and Time

Digunakan untuk menyimpan data tanggal dan waktu.

<b>Tipe Data</b>	<b>Keterangan</b>
DATE	Penggunaan : digunakan untuk menyimpan data tanggal.

	Jangkauan : 1000-01-01 s/d 9999-12-31 (YYYY-MM-DD). Ukuran : 3 byte.
TIME	Penggunaan : digunakan untuk menyimpan data waktu. jangkauan : -838:59:59 s/d +838:59:59 (HH:MM:SS) Ukuran : 3 byte
DATETIME	Penggunaan : digunakan untuk menyimpan data tanggal dan waktu. Jangkauan : '1000-01-01 00:00:00' s/d '9999-12-31 (YYYY-MM-DDHH:MM:SS) Ukuran : 8 byte
YEAR	Penggunaan : digunakan untuk menyimpan data tahun dari tanggal. Jangkauan : 1900 s/d 2155 (YYYY) Ukuran : 1 byte
TIMESTAMP	Kombinasi tanggal dan jam jangkauan dari '1970-01-01' sampai dengan '2037 ' Ukuran 4 byte.

#### 4) Tipe data Himpunan (Set dan Enum)

Tipe Data	Keterangan
ENUM	Penggunaan : enumerasi (kumpulan data). Jangkauan : sampai dengan 65535 string.
SET	Penggunaan : combination (himpunan data). Jangkauan : sampai dengan 255 string anggota

## TUGAS PRAKTIKUM

### a) Membuat, Menampilkan, dan Memilih, dan Menghapus Database

#### 1) Membuat database

Ketik perintah di bawah ini. (membuat database dengan nama unpam)

```
mysql> create database unpam;
```

2) Menampilkan database

Ketik perintah di bawah ini

```
mysql> show databases;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| mysql      |
| performance_schema |
| phpmyadmin |
| test       |
| unpam      |
| webauth    |
+-----+
```

3) Memilih database

Ketik perintah di bawah ini

```
mysql> use unpam;
```

4) Menghapus database

Ketik perintah di bawah ini

```
mysql> drop database unpam;
```

**b) Membuat, Menampilkan, Merubah Nama, dan Menghapus Table**

1) Buat database terlebih dahulu dengan nama kampus

```
mysql> create database kampus;
mysql> use kampus;
```

2) Membuat table

Ketik perintah di bawah ini

```
mysql> create table dosen(nip int(12),nama char(20),telepon varchar(12),primary
key(nip));
```

3) Menampilkan table

Ketik perintah di bawah ini

```
mysql>show tables;
```

Kemudian akan terlihat gambar di bawah ini:

```
mysql> show tables;
+-----+
| Tables_in_kampus |
+-----+
| dosen             |
+-----+
1 row in set (0.02 sec)

mysql>
```

4) Menampilkan field dalam table

Ketik perintah di bawah ini

```
mysql>desc dosen;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> desc dosen;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nip   | int(12) | NO   | PRI | 0       |       |
| nama  | char(20) | YES  |     | NULL    |       |
| telepon | varchar(12) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.08 sec)
```

5) Merubah Nama table

Ketik perintah di bawah ini

```
mysql>rename table dosen to data_dosen;
```

6) Menghapus table

Ketik perintah di bawah ini

```
mysql>drop table data_dosen;
```

**c) Buatlah Struktur Database, dengan mengetik perintah di bawah ini**

```
mysql>create database koperasi;
mysql>use koperasi;
mysql>create table anggota(nip int(10),nama char(20) not null,tempattinggal
varchar(30),kota char(10),primary key(nip));
mysql>create table pengurus(id int(10),nama varchar (20),alamat
varchar(20),tlp varchar(10),primary key(id));
mysql>create table nasabah(kode char(10),nama char(20),alamat
varchar(20),tlp varchar(10),primary key(kode));
mysql>show tables;
mysql>desc anggota;
mysql>desc pengurus;
mysql>desc nasabah;
```

**d) Buatlah Struktur Database berdasarkan table di bawah ini**

nama database : Universitas dan nama table : Mahasiswa

No	Nama Field	Tipe	Panjang
1	Nim	Int	12
2	Nama	Varchar	15
3	Alamat	Text	-
4	Email	Varchar	15
5	Telepon	Varchar	12

**TUGAS PENDAHULUAN**

1. Apa yang dimaksud dengan Auto Increment ?
2. Apa yang dimaksud dengan field dalam sebuah table ?
3. Apa perbedaan primary key dan foreign key ?
4. Apa perbedaan tipe data himpunan ENUM dan SET ?

**TUGAS AKHIR**

1. Buatlah kesimpulan praktikum di atas !

## PERTEMUAN IV

### DATA DEFINITION LANGUAGE (Bagian-2)

#### TUJUAN PRAKTIKUM

- a) Mahasiswa dapat mengenal dan memahami Data Definition Language pada bahasa pemrograman SQL.
- b) Mahasiswa dapat membuat, menampilkan, menghapus, dan memanipulasi struktur yang ada di dalam basis data pada MySQL.

#### TEORI DASAR

##### a) Pendahuluan

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya table, view, user, dan sebagainya. Secara umum, DDL yang digunakan adalah CREATE untuk membuat objek baru, USE untuk menggunakan objek, ALTER untuk mengubah objek yang sudah ada, dan DROP untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data. jadi gampangnya DDL digunakan ketika kita ingin membuat, mengubah, dan menghapus object pada database. oleh karena itu DDL lebih berhubungan pada object bukan pada isi atau data.

##### b) Perintah Alter

ALTER digunakan untuk memodifikasi definisi (struktur) dari table dengan memodifikasi definisi kolom-kolomnya. Perintah ALTER digunakan untuk melakukan fungsi-fungsi berikut.

- 1) Tambah, drop, memodifikasi kolom table
- 2) Tambah dan drop kendala
- 3) Aktifkan dan Nonaktifkan kendala

Sintaks untuk menambahkan kolom

```
ALTER TABLE table_name ADD column_name datatype;
```



Sintaks untuk menambahkan primary key

```
ALTER TABLE table_name ADD PRIMARY KEY column_name;
```

Sintaks untuk menambahkan foreign key

```
ALTER TABLE table_name2 ADD FOREIGN KEY (primary key table_name1 )  
REFERENCES table_name1 (primary key table_name1) ;
```

Sintaks untuk menjatuhkan kolom

```
ALTER TABLE table_name DROP column_name;
```

Sintaks untuk memodifikasi kolom

```
ALTER TABLE table_name MODIFY column_name datatype;
```

## TUGAS PRAKTIKUM

a) **Buatlah Struktur Database, dengan mengetik perintah di bawah ini**

```
mysql> create database toko;  
mysql> use toko;  
mysql>create table karyawan(nik int(10),nama char(20) not null,tempattinggal  
varchar(30),kota char(10),primary key(nik));  
mysql>create table pelanggan(kode_plg int(10),nama varchar (20),alamat  
varchar(20),tlp varchar(20]);  
mysql>create table barang(kode_brg int(10),nama_brg char(15),stok  
varchar(10),harga varchar(10));
```

b) **Dari point a) di atas, ubah struktur database sebagai berikut:**

1) Menambah field(kolom) pendidikan pada table karyawan

Ketik perintah di bawah ini

```
mysql> ALTER TABLE karyawan ADD pendidikan char(10) after nama;  
mysql> desc karyawan;
```

- 2) Mengubah nama sebuah kolom pada table karyawan.

Ketik perintah di bawah ini

```
mysql> ALTER TABLE karyawan CHANGE tempattinggal alamat varchar(25);
```

- 3) Mengganti nama sebuah table pada table pelanggan

Ketik perintah di bawah ini

```
mysql> ALTER TABLE pelanggan RENAME to pembeli;
```

- 4) Menghapus sebuah kolom pada table karyawan

Ketik perintah di bawah ini

```
mysql> ALTER TABLE karyawan DROP kota;
```

- 5) Memodifikasi kolom pada table barang

Ketik perintah di bawah ini:

```
mysql> ALTER TABLE barang MODIFY nama_brg char(20);
```

- 6) Memodifikasi dan menambah auto increment pada table barang

Ketik perintah di bawah ini:

```
mysql> ALTER TABLE barang MODIFY nama_brg char(20)not null  
auto_increment;
```

- 7) Memodifikasi kolom untuk menentukan awal auto increment

Ketik perintah di bawah ini:

```
mysql> ALTER TABLE barang auto_increment=1000;
```

**c) Buatlah Struktur database berdasarkan table di bawah ini**

Nama database : rumah\_sakit

Nama table 1 : pasien

Field	Tipedata	Jangkauan
kode_pasien	Char(primary key)	20
nama	Varchar	15
penyakit	Varchar	20
kode_dokter	varchar	13
biaya	char	15

Nama table 2 : jam\_praktek

Field	Tipe data	Jangkauan
kode_dokter	Char	10
nama	char	20
spesialis	char	20
waktu_praktek	date	-

- d) Dari point c) di atas, dengan menggunakan perintah Alter, ubah menjadi seperti table di bawah ini

Nama table 1 : pasien

Field	Tipe data	Jangkauan
id_pasien	Char(primary key)	20
nama_pasien	Varchar	15
penyakit	Varchar	20
kd_dokter	char(foreign key)	10
kode_obat	char	10
biaya	char	15

Nama table 2 : jam\_praktek

Field	Tipe data	Jangkauan
kd_dokter	char(primary key)	10
nama_dokter	char	20
spesialis	varchar	15
jam_praktek	datetime	-

**Ketentuan:**

- 1) Mengubah kode\_pasien dengan id\_pasien pada table 1
- 2) Mengubah nama menjadi nama\_pasien pada table 1
- 3) Mengubah kode\_dokter menjadi kd\_dokter pada table 1 dan table 2
- 4) Menggunakan auto increment kd\_dokter dimulai dari 20 pada table 2
- 5) Menambah kolom kode\_obat char (10) setelah kd\_dokter pada table 1
- 6) Tambahkan foreign key pada kd\_dokter pada table 1
- 7) Tambahkan primery key pada kd\_dokter pada table 2

- 8) Mengubah nama dengan nama\_dokter pada table 2
- 9) Mengubah spesialis char menjadi varchar (15) table 2
- 10) Mengubah waktu\_praktek menjadi jam\_praktek tipe data-nya datetime pada table 2

### **TUGAS PENDAHULUAN**

1. Jelaskan yang dimaksud dengan perintah alter !
2. Tuliskan dan jelaskan deklarasi sintaks penggunaan perintah alter !
3. Mengapa dalam pembuatan database seringkali memerlukan perintah alter ?
4. Buatlah sebuah struktur database sederhana yang dapat menggunakan perintah alter !

### **TUGAS AKHIR**

1. Buatlah kesimpulan pada praktikum di atas !

## **PERTEMUAN V**

### **DATA MANIPULATION LANGUAGE (Bagian-1)**

#### **TUJUAN PRAKTIKUM**

- a) Mahasiswa dapat mengenal dan memahami Data Manipulation Language pada bahasa pemrograman SQL.
- b) Mahasiswa dapat melakukan penambahan, pengubahan, dan penghapusan suatu data/record pada table dalam basis data.

#### **TEORI DASAR**

##### **a) Pendahuluan**

Data Manipulation Language (DML) adalah perintah-perintah yang berfungsi untuk melakukan manipulasi data ataupun objek-objek yang ada di dalam table. Definisi lainnya tentang DML adalah perintah-perintah untuk memanipulasi data pada basis data, misalnya perintah untuk memilih data (query), menyisipkan, mengubah dan menghapus data dalam basis data. Bentuk manipulasi yang dapat dilakukan oleh DML diantaranya adalah untuk keperluan pencarian kembali data lama, penyisipan data baru, penghapusan data, pengubahan data. Perintah-perintah yang ada di DML adalah INSERT, UPDATE, SELECT, DELETE

Kegunaan DML adalah:

- 1) Pengambilan informasi yang disimpan dari basis data.
- 2) Penyisipan informasi baru ke basis data.
- 3) Penghapusan informasi dari basis data.
- 4) Modifikasi informasi yang disimpan dalam basis data.
- 5) Query adalah perintah yang ditulis untuk mengambil informasi. Bagian dari DML yang menangani pengambilan informasi ini disebut bahasa query.

##### **b) Penambahan data pada table dengan insert**

INSERT dan INSERT INTO adalah sebuah perintah yang berfungsi untuk menambahkan data dalam suatu table yang sudah dibuat. ada tiga cara yaitu:

Cara pertama :

**INSERT INTO** nama\_table **values** ('nilai1','nilai2','nilai3',.....);

Cara kedua :

```
INSERT INTO nama_table (field1,field2,field3,...)values ('nilai1','nilai2','nilai3',....);
```

Cara ketiga :

```
INSERT INTO nama_table SET field1='nilai1',field2='nilai2',field3='nilai3',field4='nilai4',...;
```

### c) Mengubah data pada table dengan update

Perintah UPDATE berfungsi untuk merubah/memperbaharui data yang telah ada di dalam table.

```
UPDATE nama_table SET nama_field = 'nilai_baru' WHERE nama_field = 'kondisi';
```

### d) Menghapus data pada table dengan delete

Perintah DELETE berfungsi untuk menghapus record/data yang ada pada sebuah table.

```
DELETE FROM nama_table WHERE nama_field = 'option';
```

## TUGAS PRAKTIKUM

### a) Buatlah Struktur Database, dengan mengetik perintah di bawah ini

```
mysql> create database universitas;  
mysql> use universitas;  
mysql> create table mahasiswa(nim int(10),nama char(20),alamat varchar(30), email  
char(20),no_tlp char(10),primary key(nim));  
mysql> show tables;  
mysql> desc mahasiswa;
```

**b) Dari point a) di atas, tambahkan data sebagai berikut:**

Menambah data pada table mahasiswa dengan insert into

Ketik perintah di bawah ini

```
mysql>INSERT INTO mahasiswa (nim,nama,alamat,email,no_tlp)
values ('080911002','Sari','Pamulang','sari@gmail.com','08561009999');
mysql>INSERT INTO mahasiswa (nim,nama,alamat,email,no_tlp)
values ('080911001','lala','jakarta','lala@gmail.com','08561009124');
mysql>INSERT INTO mahasiswa values
('080911003','Tono','Parung','tono@gmail.com','08561009888');
mysql>INSERT INTO mahasiswa values
('080911006','Toti','ciputat','toti@gmail.com','08561009555');
mysql> INSERT INTO mahasiswa set nim='080911014',nama='Ida',
alamat='Serpong',email='ida@gmail.com',no_tlp='08561009777';
mysql> INSERT INTO mahasiswa set nim='080911004',nama='Iya',
alamat='Semarang',email='iya@gmail.com',no_tlp='08561009999';
```

Hasil Penambahan, di lihat menggunakan perintah berikut:

```
mysql>select * from mahasiswa;
```

**c) Dari point a) dan b) di atas, ubah data sebagai berikut:**

Mengubah data pada table mahasiswa dengan update

Ketik perintah di bawah ini

```
mysql>UPDATE mahasiswa set alamat ='ciputat' where nim=080911002;
mysql>UPDATE mahasiswa set email ='jaya@yahoo.com' where nim=080911002;
mysql>UPDATE mahasiswa set nama ='Bambang'where nim=0809110003;
mysql>UPDATE mahasiswa set alamat ='ciputat' where nim=0809110003;
mysql>UPDATE mahasiswa set email ='yaho@yahoo.com' where
nim=0809110004;
mysql>UPDATE mahasiswa set email ='Ida@yahoo.com'where nim=0809110004;
```

d) Dari point a), b) dan c) di atas, hapus data sebagai berikut:

Menghapus data pada table mahasiswa dengan delete

Ketik perintah di bawah ini

```
mysql>delete from mahasiswa where nim=0809110004;  
mysql>delete from mahasiswa where nim=080911014;
```

e) Buat Database dengan nama Perusahaan, kemudian table dengan nama karyawan, kemudian isi table tersebut dengan data seperti di bawah ini

NIK	Nama	Alamat	Email	No_Telepon
080911001	Susi	Serpong	susi@yahoo.com	0856777777
080911002	Nuri	Jakarta	nuri@gmail.com	0856888888
080911003	Santi	Ciputat	santi@gmail.com	0812455555
080911004	Nunu	Kebayoran	nunu@yahoo.com	0817677776

f) Dari point e), kemudian ubah data pada table tersebut sehingga menjadi seperti table di bawah ini

NIK	Nama	Alamat	Email	No_Telepon
080911001	Susi	Muncul	susi@yahoo.com	0851236789
080911002	Sari	Jakarta	sari@ymail.com	0856888888
080911003	Sindy	Cinere	sindy@yahoo.com	0812454564
080911004	Sandra	Kebayoran	sandra@yahoo.com	0817677776

## TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan DML ?
2. Tuliskan dan jelaskan deklarasi sintaks perintah-perintah pada DML !
3. Apa perbedaan perintah drop dan delete dalam MySQL, jelaskan !
4. Buatlah sebuah struktur database sederhana dengan menggunakan perintah pada DML !

## TUGAS AKHIR

1. Buatlah kesimpulan dari praktikum diatas !



## PERTEMUAN VI

### DATA MANIPULATION LANGUAGE (Bagian-2)

#### TUJUAN PRAKTIKUM

- a) Mahasiswa dapat mengenal dan memahami Data Manipulation Language pada bahasa pemrograman SQL.
- b) Mahasiswa dapat mencari dan menampilkan data/record pada table dalam basis data.

#### TEORI DASAR

##### a) Pendahuluan

Menampilkan data dari sebuah table merupakan hal paling sering dilakukan dalam menangani database. Menampilkannya kembali dalam format yang diinginkan merupakan rutinitas selama database masih digunakan. Cara menampilkan data dari table MySQL dengan query **SELECT**, yaitu perintah yang digunakan untuk menampilkan data dari satu atau lebih table, biasanya dalam sebuah basis data yang sama.

##### b) Mencari dan Menampilkan Data pada Table

Mencari dan menampilkan data pada Table diperlukan perintah **Select**. Parameter-parameter perintah **select** bermacam-macam sesuai kebutuhan yang ingin dicari dan ditampilkan user. **Select** untuk menampilkan keseluruhan kolom pada table, menggunakan sintaks seperti di bawah ini.

```
SELECT * FROM nama_table;
```

Untuk memilih hanya kolom-kolom tertentu yang ingin ditampilkan ganti karakter **\*** dengan nama kolom yang ingin ditampilkan sebagai berikut.

```
SELECT kolom1,kolom2 ,...FROM nama_table;
```

Untuk mengganti judul hasil tampilan output perintah **Select**, digunakan keyword **AS** seperti sintaks berikut.

```
SELECT kolom1 AS 'judul yang akan ditampilkan',kolom2... FROM nama_table;
```

## TUGAS PRAKTIKUM

- a) **Buatlah Struktur Database dengan pasien sebagai table, dengan mengetik perintah di bawah ini**

```
mysql>create database apotek;
mysql>use apotek;
mysql>create table pasien(id_pasien int(10),nama char(20),alamat varchar(30),
penyakit char(20),no_tlp char(15),primary key(id_pasien));
mysql>desc pasien;
```

- b) **Dari point a) di atas, tambahkan data sebagai berikut:**

```
mysql>INSERT INTO pasien values
('10001','Tatang','ciputat','amandel','08561009111'),('10002','Tuti','Tangerang','An
imia','08561009134'),('10003','Tono','jakarta','Alergi','08561009566'),
('10004','Budi','Bekasi','Demam','08561009177');
```

- c) **Dari point a) dan b) di atas, cari dan tampilkan data**

Menampilkan keseluruhan data pada table pasien

Ketik perintah di bawah ini

```
mysql>select * from pasien;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select * from pasien;
+-----+-----+-----+-----+-----+
| id_pasien | nama   | alamat   | penyakit | no_tlp   |
+-----+-----+-----+-----+-----+
| 10001     | Tatang | Ciputat  | Amandel  | 08561009111 |
| 10002     | Tuti   | Tangerang | Animia   | 08561009134 |
| 10003     | Tono   | Jakarta  | Alergi   | 08561009566 |
| 10004     | Budi   | Bekasi   | Demam    | 08561009177 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

**d) Dari point a), b) dan c) di atas, manampilkan field-field tertentu**

Mencari dan menampilkan data pada field tertentu di table pasien

Ketik perintah di bawah ini:

```
mysql>select nama from pasien;  
mysql>select id_pasien,penyakit from pasien;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select  nama from pasien;  
+-----+  
|  nama  |  
+-----+  
|  Tatang  |  
|  Tuti   |  
|  Tono    |  
|  Budi    |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql> select  id_pasien,penyakit from pasien;  
+-----+-----+  
| id_pasien | penyakit |  
+-----+-----+  
| 10001    | Amandel  |  
| 10002    | Anemia   |  
| 10003    | Alergi   |  
| 10004    | Demam    |  
+-----+-----+
```

**e) Dari point a), b),c) dan d) di atas, mengganti nama field**

Menampilkan nama field pada table pasien

Ketik perintah di bawah ini:

```
mysql>select id_pasien AS 'Kode Pasien',nama,penyakit from pasien;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select id_pasien As'kode pasien',nama,penyakit from pasien;  
+-----+-----+-----+  
| kode pasien | nama  | penyakit |  
+-----+-----+-----+  
| 10001      | Tatang | Amandel  |  
| 10002      | Tuti   | Anemia   |  
| 10003      | Tono    | Alergi   |  
| 10004      | Budi    | Demam    |  
+-----+-----+-----+  
4 rows in set (0.00 sec)
```

- f) Buatlah Database dengan nama penjualan, kemudian table dengan nama barang, kemudian isi table tersebut dengan data seperti di bawah ini

Kode_barang	Nama_barang	Satuan	Harga_satuan	Stok
11001	Sabun	Bungkus	3000	1010
11002	Detergen	Kg	10000	1044
11003	Shampo	Botol	7000	578
11004	kopi	Bungkus	5000	466
11005	Gula	Kg	12000	500
11006	Teh	Bungkus	5000	600
11007	Tepung	Kg	7000	100

- g) Dari point f) di atas, tampilkan seluruh data yang telah di buat dengan perintah select
- h) Dari point f) di atas, tampilkan isi data field pada table tersebut sehingga menjadi seperti gambar di bawah ini

nama_barang	harga_satuan
sabun	3000
detergen	10000
shampo	7000
kopi	5000
gula	12000
teh	5000
tepung	7000

- i) Dari point f) di atas, ubah nama field pada table tersebut sehingga menjadi seperti gambar di bawah ini

id barang	nama_barang	harga_satuan
11001	sabun	3000
11002	detergen	10000
11003	shampo	7000
11004	kopi	5000
11005	gula	12000
11006	teh	5000
11007	tepung	7000

### **TUGAS PENDAHULUAN**

1. Jelaskan kegunaan perintah select !
2. Bagaimana cara menampilkan seluruh data pada sebuah table dalam sebuah database ?
3. Jelaskan fungsi karakter bintang (\*) pada perintah select !
4. Apa yang menjadi alasan sebuah data perlu ditampilkan dalam sebuah database ?

### **TUGAS AKHIR**

1. Buatlah kesimpulan dari praktikum diatas !

## PERTEMUAN VII

### DATA MANIPULATION LANGUAGE (Bagian-3)

#### TUJUAN PRAKTIKUM

- a) Mahasiswa dapat mengenal dan memahami Data Manipulation Language pada bahasa pemrograman SQL.
- b) Mahasiswa dapat mencari dan menampilkan data/record dengan menggabungkan beberapa perintah seperti: where, operator matematika, operator logika pada table dalam basis data.

#### TEORI DASAR

**a) Mencari dan Menampilkan Data dengan Where pada Table dalam Basis Data.**

Untuk menyeleksi seluruh data yang ingin ditampilkan dengan kriteria tertentu dapat menggunakan where

```
SELECT * FROM nama_table WHERE nama_kolom='kriteria yang ditampilkan';
```

**b) Mencari dan Menampilkan Data dengan Operator Matematika pada Table dalam Basis Data.**

Operasi matematika dalam mysql sering dijadikan alternatif yang paling efektif dalam menghasilkan sebuah laporan yang detail karena dapat meminimalkan penggunaan memori serta mempermudah penyusunan coding program. Operasi matematika dalam MySQL biasanya diterapkan dengan menggunakan operator aritmatika. Untuk menyeleksi seluruh data yang ingin ditampilkan dengan operator matematika dengan kriteria tertentu dengan sintaks sebagai berikut:

OPERATOR	KETERANGAN
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
% atau mod	Sisa Hasil Bagi
div	Pembagian integer

Dengan adanya operasi matematika yang disisipkan dalam query maka tidak perlu lagi melakukan perhitungan melalui script program, selain itu juga meminimalkan penggunaan memori karena tidak perlu menambahkan field untuk menyimpan hasil perhitungan masing-masing field yang ada.

**c) Mencari dan Menampilkan Data dengan Operator pembandingan pada Table dalam BasisData.**

Operator ini digunakan untuk melakukan pembandingan apakah beberapa elemen sama, identik, kurang dari atau lebih besar dari yang lain. Operator perbandingan digunakan untuk membandingkan data kolom dengan nilai-nilai tertentu dalam suatu kondisi. Operator Perbandingan digunakan bersama dengan pernyataan SELECT untuk menyaring data berdasarkan kondisi tertentu.

OPERATOR	PENJELASAN
<	Kurang dari
>	Lebih dari
<=	Kurang dari sama dengan
>=	Lebih dari sama dengan
=	Sama Dengan
< > atau !=	Tidak sama dengan

**d) Mencari dan Menampilkan Data dengan Operator Logika pada Table dalam BasisData.**

**1) Operator AND**

Operator AND digunakan untuk mengambil data dari satu atau beberapa table menggunakan syarat atau lebih, data akan ditampilkan bila kedua syarat terpenuhi, begitu juga sebaliknya jika salah satu diantara kedua syarat tersebut tidak terpenuhi maka data tidak akan ditampilkan

**SELECT \* FROM** nama\_table **WHERE** nama\_kolom [kriteria yang ingin ditampilkan] **AND** nama\_kolom [kriteria yang ingin ditampilkan];

## 2) Operator OR

Berbeda dengan operator AND, operator OR digunakan untuk mengambil data dari satu atau lebih, tetapi memiliki syarat tidak harus kedua-duanya bernilai benar. Jika salah satu bernilai benar maka data akan tetap ditampilkan, kemudian apabila kedua kondisi tidak bernilai benar atau tidak terpenuhi maka data tersebut tidak akan ditampilkan.

```
SELECT * FROM nama_table WHERE nama_kolom [kriteria yang ingin  
ditampilkan] OR nama_kolom [kriteria yang ingin ditampilkan];
```

## TUGAS PRAKTIKUM

- a) **Buatlah Struktur Database dengan nama perusahaan dan nama karyawan sebagai table.** ketik perintah di bawah ini

```
mysql>create database perusahaan;  
mysql>use perusahaan;  
mysql>create table karyawan(nik char(10),nama char(20),alamat varchar(30), email  
char(20),no_tlp char(15),primary key(nik));  
mysql>desc karyawan;
```

- b) **Dari point a) di atas, tambahkan data sebagai berikut:**

ketik perintah di bawah ini

```
mysql> INSERT INTO karyawan values  
(‘10011’,’Rini’,’Ciputat’,’rini@yahoo.co.id’,’08561009111’),(‘10012’,’Agung’,’Tang  
erang’,’agung@yahoo.com’,’08561009134’),(‘10008’,’Cici’,’Jakarta’,’cici@gmail.co  
m’,’08561009566’), (‘10004’,’Hendra’,’Tangerang’,’hendra@gmail.com  
,’08561009177’), (‘10002’,’Geri’,’Tangerang’,’geri@yahoo.com’,’08561009187’);
```

- c) **Dari point a) dan b) di atas, cari dan tampilkan data**

Menampilkan data dengan perintah where untuk menampilkan data karyawan yang berasal dari Tangerang.

Ketik perintah di bawah ini

```
mysql>select * from karyawan where alamat =’tangerang’;
```



Kemudian akan terlihat gambar di bawah ini

```
mysql> select *from karyawan where alamat='Tangerang';
+-----+-----+-----+-----+-----+
| nik   | nama  | alamat | email          | no_tlp   |
+-----+-----+-----+-----+-----+
| 10002 | Geri   | Tangerang | geri@yahoo.com | 08561009187 |
| 10004 | Hendra | Tangerang | hendra@gmail.com | 0856009177  |
| 10012 | Agung  | Tangerang | agung@yahoo.com | 08561009134 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

- d) Dari point a) dan b) di atas, cari dan tampilkan data

Menampilkan data dengan perintah where dengan menggunakan operator matematika

Ketik perintah di bawah ini

```
mysql>select * from karyawan where nik >= 10008;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select *from karyawan where nik >= 10008;
+-----+-----+-----+-----+-----+
| nik   | nama  | alamat | email          | no_tlp   |
+-----+-----+-----+-----+-----+
| 10008 | Cici   | Jakarta | cici@gmail.com | 08561009566 |
| 10011 | Rini   | Ciputat | rini@yahoo.co.id | 08561009111 |
| 10012 | Agung  | Tangerang | agung@yahoo.com | 08561009134 |
+-----+-----+-----+-----+-----+
3 rows in set (0.06 sec)
```

- e) Buat Database dengan nama Universitas, kemudian table dengan nama Dosen, kemudian isi table tersebut dengan data seperti di bawah ini

NIK	Nama	Alamat	Email	No_Telepon
0809111	Anang	Serpong	anang@yahoo.com	0856777777
0809112	Baudiono	Jakarta	budi@gmail.com	0856888888
0809113	Sariyani	Ciputat	sari@gmail.com	0812455555
0809114	Wahyudi	jakarta	wahyudi@yahoo.com	0817676666
0809115	Suharta	Tangerang	harta@yahoo.com	0817674444

- f) Dari point e) di atas, tampilkan data yang hanya beralamat Jakarta, sehingga menjadi seperti gambar di bawah ini

Nik	Nama	Alamat	Email	No_Telepon
0809112	Budiono	Jakarta	budi@gmail.com	0856888888
0809114	Wahyudi	Jakarta	wahyudi@yahoo.com	0817676666

- g) Dari point e) di atas, tampilkan data yang memiliki nik diatas 0809113 atau yang beralamat Ciputat, sehingga menjadi seperti gambar di bawah ini

Nik	Nama	Alamat	Email	No_Telepon
0809113	Sariyani	Ciputat	sari@gmail.com	0812455555
0809114	Wahyudi	Jakarta	wahyudi@yahoo.com	0817676666
0809115	Suharta	Tangerang	harta@yahoo.com	0817674444

- h) Dari point e) di atas, tampilkan data yang memiliki No telepon 0856777777, Nik 0809113, sehingga menjadi seperti table di bawah ini

Nik	Nama	Alamat	Email	No_Telepon
0809111	Anang	Serpong	anang@yahoo.com	0856777777
0809113	Sariyani	Ciputat	sari@gmail.com	0812455555

- i) Dari point e) di atas, tampilkan data nama dosen dengan order by yang hanya menampilkan 4 data dengan limit, sehingga menjadi seperti di bawah ini

Nik	Nama	Alamat	Email	No_Telepon
0809111	Anang	Serpong	anang@yahoo.com	0856777777
0809112	Budiono	Jakarta	budi@gmail.com	0856888888
0809113	Sariyani	Ciputat	sari@gmail.com	0812455555
0809115	Suharta	Tangerang	harta@yahoo.com	0817674444

### **TUGAS PENDAHULUAN**

1. Tuliskan dan jelaskan Deklarasi sintaks perintah where pada MySQL !
2. Sebutkan dan Jelaskan operator-operator matematika yang dipakai dalam MySQL!
3. Sebutkan dan Jelaskan operator-operator logika yang dipakai dalam MySQL!
4. Perintah apa saja yang digunakan untuk menampilkan dua kondisi yang akan ditampilkan, jelaskan !

### **TUGAS AKHIR**

1. Buatlah kesimpulan dari praktikum diatas !

## PERTEMUAN VIII

### DATA MANIPULATION LANGUAGE (Bagian-4)

#### TUJUAN PRAKTIKUM

- a) Mahasiswa dapat mengenal dan memahami Data Manipulation Language pada bahasa pemrograman SQL.
- b) Mahasiswa dapat mencari dan menampilkan data/record dengan menggabungkan beberapa perintah seperti: between, like, in, order by pada table dalam basis data.

#### TEORI DASAR

##### a) Mencari dan Menampilkan Data dengan Between pada Table dalam Basis Data.

Between terdiri dari dua jenis yaitu Between dan Not Between. Between digunakan untuk mengolah data suatu nilai dalam *range* tertentu. Not Between Merupakan negasinya, yaitu mengolah suatu nilai di luar dari *range* yang telah ditentukan.

```
SELECT * FROM nama_table WHERE nama_kolom BETWEEN nilai_awal AND nilai_akhir;
```

##### b) Mencari dan Menampilkan Data dengan Like, in Pada Table dalam Basis Data.

###### 1) Operator Like

Operator Like biasa dipergunakan untuk mencari sebuah karakter atau kata tertentu dalam sebuah data. Operator ini biasa digunakan untuk mencari data bertipe String/Teks. Penempatan simbol '%' sangat berpengaruh dalam pengoperasian operator ini. Contoh penggunaan simbol '%' pada LIKE dalam pencarian data menurut awalan, akhiran, atau penggalan suatu kata atau suku kata.

Karakter	Keterangan
'A%'	Berawalan A
'%A'	Berakhiran A
'%S%'	Mengandung S
'%i%n%'	mengandung i dan n

Sintaks **LIKE** dapat ditulis sebagai berikut:

```
SELECT * FROM nama_table WHERE nama_kolom LIKE keyword_pencarian;
```

## 2) Operator NOT LIKE

Operator NOT LIKE berfungsi hampir sama dengan operator LIKE, namun Outputnya akan berbeda. Jika operator Like akan memunculkan data yang mengandung karakter tertentu yang ingin ditampilkan, maka Not Like tidak akan menampilkan data yang mengandung karakter tersebut.

Sintaks **NOT LIKE** dapat ditulis sebagai berikut :

```
SELECT * FROM nama_table WHERE nama_kolom NOT LIKE  
keyword_pencarian;
```

## 3) IN dan NOT IN

IN ini biasa digunakan untuk membandingkan dengan sebuah kumpulan nilai-nilai yang diisikan seperti query tunggal dan query yg hasilnya hanya terdapat sebuah kolom saja. Biasanya IN digunakan untuk pencarian data menggunakan lebih dari satu filter pada perintah Where.

Sintaks IN dapat ditulis sebagai berikut:

```
SELECT * FROM nama_table WHERE nama_kolom IN ('values1','values2'...);
```

Sedangkan NOT IN adalah kebalikan dari nilai yang ditampilkan oleh IN

Sintaks NOT IN dapat ditulis sebagai berikut:

```
SELECT * FROM nama_table WHERE nama_kolom NOT IN  
( 'values1','values2'...);
```

## 4) Mencari dan Menampilkan Data dengan Order by pada Table dalam Basis Data.

Order by digunakan untuk mengurutkan data yang akan ditampilkan dari sebuah kolom, secara default untuk ORDER BY adalah ascending(ASC), yaitu diurutkan berdasarkan dari kecil ke besar. Jika ingin mengurutkan dari besar ke kecil dapat digunakan descending (DESC).

Sintaks ORDER BY ascending dapat ditulis sebagai berikut:

```
SELECT * FROM nama_table ORDER BY nama_kolom ;
```

Sintaks ORDER BY descending dapat ditulis sebagai berikut:

```
SELECT * FROM nama_table ORDER BY nama_kolom DESC ;
```

#### 5) Mencari dan Menampilkan Data dengan Group by pada Table dalam Basis Data.

Group by Digunakan untuk mengelompokkan sekumpulan record berdasarkan kolom tertentu sesuai data yang ingin akan ditampilkan.

```
SELECT nama_kolom FROM nama_table GROUP BY nama_kolom ;
```

Perintah having hanya dapat digunakan jika group by di gunakan semua data akan ditampilkan kecuali data yang di having.

```
SELECT nama_kolom FROM nama_table GROUP BY nama_kolom  
HAVING nama_kolom [operator pembandingan] 'kriteria yang ditampilkan';
```

### TUGAS PRAKTIKUM

#### a) Buatlah Struktur Database penjualan dengan suplier sebagai table

Ketik perintah di bawah ini

```
mysql>create database penjualan;  
mysql>use penjualan;  
mysql>create table suplier(id_suplier varchar(15) , nama_suplier char(20),alamat  
varchar(30), no_tlp char(15),primary key(nik));  
mysql>desc suplier;
```

#### b) Dari point a) di atas, tambahkan data sebagai berikut:

Menambah data pada table penjualan dengan insert into

Ketik perintah di bawah ini

```
mysql>INSERT INTO suplier values ('11234','Elex Media','PalMerah',
,'0215455454');
mysql>INSERT INTO suplier values ('11236','BIP','Cakung','0215455234');
mysql>INSERT INTO suplier values ('11235','AgroMedia','Ciganjur','0215455776');
mysql>INSERT INTO suplier values ('11232','Erlangga','Ciracas','0215454242');
mysql>INSERT INTO suplier values ('11231','Mizan','Cilandak','0215455899');
mysql>INSERT INTO suplier values ('11230','KompasMedia','PalMerah',
,'0215455433');
```

**c) Dari point a) dan b) di atas, cari dan tampilkan data**

Memampilkan data dengan perintah between untuk menampilkan data suplier yang mempunya id\_suplier 11231 dan 11235

Ketik perintah di bawah ini:

```
mysql>select * from suplier where id_suplier between 11231 and 11234;
```

Kemudian akan terlihat gambar di bawah ini

id_suplier	nama_suplier	alamat	no_tlp
11231	Mizan	Cilandak	0215455899
11232	Erlangga	Ciracas	0215454242
11234	ElexMedia	PalMerah	0215455454

**d) Masih di dalam Database penjualan, buat table dengan nama buku, kemudian isi table tersebut dengan data seperti di bawah ini**

Isbn	Judul	Pengarang	Penerbit	Harga
2001145	IPS Terpadu	Tim Guru	Erlangga	54.000
2005666	Cerdas Berbahasa	Srikanti	KompasMedia	60.000
2007575	Teknik Industri	Suryanto	Penerbit Andi	50.000
2000698	Akuntansi Lanjutan	Tonikurnia	Graha Ilmu	40.000
2000543	Good English	Michael R	ElexMedia	45.000
2054449	Kimia Dasar	Michael Purba	Erlangga	60.000

- e) Dari point d) di atas, cari dan tampilkan data yang mempunyai kisaran harga 50.000 sampai 60.000, sehingga seperti gambar di bawah ini

ISBN	Judul	Pengarang	Penerbit	Harga
2001145	IPS Terpadu	Tim Guru	Erlangga	54000
2005666	Cerdas Berbahas	Srikanti	KompasMedia	60000
2007575	Teknik Industri	Suryanto	Penerbit And	50000
2054449	Kimia Dasar	Michael Purba	Erlangga	60000

- f) Dari point d) di atas, cari dan tampilkan data menggunakan LIKE dengan menampilkan nama penerbit yang berawalan dengan huruf E, sehingga seperti gambar di bawah ini

ISBN	Judul	Pengarang	Penerbit	Harga
2000543	Good English	Michael R	Elex Media	45000
2001145	IPS Terpadu	Tim Guru	Erlangga	54000
2054449	Kimia Dasar	Michael Purba	Erlangga	60000

- g) Dari point d) di atas, cari dan tampilkan data menggunakan NOT LIKE dengan tidak menampilkan nama penerbit yang berawalan dengan huruf E, sehingga seperti gambar di bawah ini

ISBN	Judul	Pengarang	Penerbit	Harga
2000698	Akuntansi Lanju	Tonikurnia	Graha Ilmu	40000
2005666	Cerdas Berbahas	Srikanti	KompasMedia	60000
2007575	Teknik Industri	Suryanto	Penerbit And	50000

- h) Dari point d) di atas, cari dan tampilkan data menggunakan IN dan NOT IN untuk menampilkan pengarang Srikanti, Suryanto, Tim Guru
- i) Dari point d) di atas, cari dan tampilkan data nama suplier saja dengan perintah Group by dan Having
- j) Dari point d) di atas, cari dan tampilkan data ISBN dengan order by asc dan desc



### **TUGAS PENDAHULUAN**

1. Tuliskan dan jelaskan Deklarasi sintaks perintah between pada MySQL?
2. Apakah fungsi perintah having dalam menampilkan data ?
3. Sebutkan dan Jelaskan pengurutan data dalam Order by ?
4. Apa perbedaan dari like dan Not like ?

### **TUGAS AKHIR**

1. Buatlah kesimpulan praktikum di atas !

## PERTEMUAN IX

### DATA MANIPULATION LANGUAGE (Bagian-5)

#### TUJUAN PRAKTIKUM

- a) Mahasiswa dapat mengenal dan memahami Data Manipulation Language pada bahasa pemrograman SQL.
- b) Mahasiswa dapat mencari dan menampilkan data/record dengan menggabungkan beberapa perintah seperti: distinct, operator aritmatika, agregat pada table dalam basis data.

#### TEORI DASAR

**a) Mencari dan Menampilkan Data dengan Distinct pada Table dalam Basis Data.**

DISTINCT digunakan hanya untuk memanipulasi tampilan hasil dari table. Duplikasi yang dihapus adalah untuk tampilan data, bukan table asli dari MySQL. DISTINCT hanya mengeleminasi query yang unik, atau tidak sama dilihat secara baris per baris (per record). Dengan penambahan perintah DISTINCT di awal query SELECT, maka hanya data yang unik saja (data yang tidak sama) yang akan ditampilkan. Seandainya hasil query terdapat data yang sama lebih dari 1 kali kemunculan, perintah DISTINCT hanya akan menampilkannya 1 kali saja.

Sintaks query **DISTINCT** adalah sebagai berikut:

```
SELECT DISTINCT nama_kolom FROM nama_table;
```

**b) Mencari dan Menampilkan Data dengan Operator Aritmatika pada Table dalam Basis Data.**

Fungsi Operator Aritmatika dasar yang dimiliki oleh MySQL adalah; penjumlahan, pengurangan, perkalian, dan pembagian. Operator aritmetika tersebut disertakan dalam perintah select yang bertujuan untuk memanipulasi data yang sifatnya sederhana. Untuk kebutuhan yang lebih kompleks guna memanipulasi data secara aritmetika, MySQL juga menyediakan fungsi-fungsi built-in. Sintaks query **DISTINCT** adalah sebagai berikut

Sintaks query menggunakan operator aritmatika adalah sebagai berikut:

```
SELECT Operator Aritmatika(data yang akan dicari);
```

Adapun fungsi aritematika [numerik] built-in yang telah disediakan oleh MySQL.

<b>FUNGSI</b>	<b>KETERANGAN</b>
ABS(x)	Digunakan untuk mendapatkan nilai absolut dari suatu data numerik.
ACOS(x)	Fungsi ini dapat menghasilkan nilai arc cosine x, apabila x tidak bernilai antara -1 dan 1 maka akan menghasilkan NULL.
ASIN(x)	Dapat menghasilkan arc sinus x, juga akan menghasilkan NULL jika nilai x tidak bernilai antara -1 dan 1.
ATAN(x)	Menghasilkan arc tangent x.
ATAN2(x, y)	Fungsi ini menghasilkan arc tangent x berdasarkan parameter y untuk menentukan kuadrant.
BIN(x)	Fungsi ini digunakan untuk mendapatkan nilai biner dari bilangan x dengan tipe data BIGINT. Hasil dari fungsi ini memiliki tipe data STRING.
CEILING(x)	Nilai yang dihasilkan dari fungsi ini dikonversikan menjadi BIGINT berupa nilai bilangan bulat terbesar yang tidak lebih kecil dari x.
CONV(x, daribasis, kebasis)	Digunakan untuk melakukan konversi suatu bilangan dari suatu basis bilangan ke basis bilangan lainnya. Hasil dari fungsi ini adalah STRING.
COS(x)	Digunakan untuk menghitung nilai cosinus x dalam radian.
COT(x)	Menghitung cotangent x.
DEGREES(x)	Dapat menghasilkan besar sudut dalam derajat, dari suatu x dalam radian.
EXP(x)	Menghitung e pangkat x
FLOOR(x)	Membulatkan bilangan ke integer terdekat yang tidak lebih besar dari x.
FORMAT(num,dec)	Membuat format desimal "x.xxx.xxx.xx" dengan membulatkan angka yang didefinisikan dengan dec.
GREATEST(x, y, ...)	Fungsi ini digunakan untuk mendapatkan nilai terbesar dari daftar bilangan.
HEX(x)	Untuk menghasilkan bilangan hexadesimal dengan tipe data

	STRING dari x dengan tipe data BIGINT.
LEAST(x, y, z, ...)	Fungsi ini digunakan untuk mendapatkan nilai terkecil dari daftar bilangan
LOG(x)	Untuk mendapatkan nilai logaritma natural dari x.
LOG10(x)	Menghasilkan logaritma dasar 10.
MOD(x, y)	Fungsi yang sama ketika menggunakan operator % yaitu untuk mendapatkan sisa bagi.
OCT(x)	Memberikan bilangan oktal dari suatu bilangan x dengan tipe data BIGINT. Nilai yang dihasilkan bertipe STRING
PI()	Tentu saja digunakan untuk mendapatkan nilai konstanta PI.
RADIANS(x)	Digunakan untuk menghasilkan nilai radian dari x dalam derajat. Merupakan kebalikan dari fungsi DEGREES(x).
RAND() dan RAND(seed)	Untuk mendapatkan bilangan random pada pecahan 0 sampai 1. Jika diberikan parameter seed maka jika seed sama bilangan randomnya juga sama dan bisa diprediksi
ROUND(x)	Membulatkan bilangan x menjadi bilangan bulat integer.
ROUND(x, y)	Membulatkan angka termasuk dibelakang koma jika y didefinisikan.
SIGN(x)	Digunakan untuk menentukan apakah nilai x bernilai positif, negatif atau nol. Fungsi ini akan memberikan nilai -1 untuk data negatif, 1 untuk data positif dan 0 untuk data nol
SIN(x)	Digunakan untuk menghitung Sinus x dalam radian.
SQRT(x)	Untuk mencari akar kuadrat bilangan x
TAN(x)	Digunakan untuk menghitung tangent x dalam radian.
TRUNCATE(x, y)	Memotong bilangan desimal x hingga sejumlah angka dibelakang koma y.

**c) Mencari dan Menampilkan Data dengan Agregat pada Table dalam Basis Data.**

Fungsi agregat (aggregate function) adalah fungsi di dalam MYSQL yang digunakan untuk melakukan perhitungan pada query. Contohnya yakni fungsi untuk menampilkan banyaknya jumlah data, mencari total nilai suatu atribut, mencari rata-rata atribut serta

mencari nilai maksimum dan nilai minimum dari suatu atribut. Macam-Macam Fungsi Agregat

KEYWORD	FUNGSI
COUNT	Berfungsi untuk mendapatkan nilai banyaknya record dari suatu hasil Query.
SUM	Berfungsi untuk menjumlahkan nilai dari suatu kolom numerik dalam table tertentu
AVG	Fungsi ini digunakan untuk mengitung nilai rata-rata dari suatu kolom yang dihasilkan oleh query tertentu.
MAX	Fungsi MAX merupakan fungsi yang berguna untuk memperoleh nilai maksimal atau nilai terbesar dari suatu kolom yang terdapat dalam table tertentu. Pada umumnya fungsi MAX digunakan dalam data yang bertipe numerik dan mengembalikan nilai terbesar dari data tersebut. Namun, jika fungsi MIN diterapkan pada kolom yang bertipe string maka akan mengembalikan nilai terbesar dari string (berdasarkan urutan alfabet).
MIN	Fungsi ini digunakan untuk mengembalikan nilai minimal atau terkecil dari suatu kolom pada table tertentu. Pada umumnya fungsi MIN digunakan dalam data yang bertipe numerik dan mengembalikan nilai terkecil dari data tersebut. Namun, jika fungsi MIN diterapkan pada kolom yang bertipe string maka akan mengembalikan nilai terkecil dari string (berdasarkan urutan alfabet).

## TUGAS PRAKTIKUM

### a) Buatlah Struktur Database penjualan dan buku sebagai table

Ketik perintah di bawah ini

```
mysql>create database Penjualan;  
mysql>use Penjualan;  
mysql>create table buku(isbn varchar(15) , judul char(20),pengarang char(30), harga  
varchar (15),stok char (10) ,primary key(isbn));  
mysql>desc buku;
```

**b) Dari point a) di atas, tambahkan data sebagai berikut:**

Ketik perintah di bawah ini

```
mysql>INSERT INTO buku values ('11231','Matematika Diskrit','Hanafi',
'60000','25');
mysql>INSERT INTO buku values ('11232','Pintar Java','Median','50000','20');
mysql>INSERT INTO buku values ('11233','Struktur Data','Andrianto','70000','15');
mysql>INSERT INTO buku values ('11234','Algoritma','SintaSari','45000','16');
mysql>INSERT INTO buku values ('11235','Kewarganegaraan','Ramdani','64000'
,'22');
mysql>INSERT INTO buku values ('11236','Basisdata','Suginanto','46000','33');
mysql>INSERT INTO buku values ('11237','Sistem Berkas','Suginanto','60000','20');
mysql>INSERT INTO buku values ('11238','Web PHP','Median','50000','25');
```

**c) Dari point a) dan b) di atas, cari dan tampilkan data dengan perintah distinct**

Ketik perintah di bawah ini

```
mysql>select distinct pengarang from buku;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select distinct pengarang from buku;
+-----+
| pengarang |
+-----+
| Hanafi    |
| Median    |
| Andrianto |
| SintaSari |
| Ramdani   |
| Suginanto |
+-----+
```

**d) Dari point a) ,b) dan c) di atas, cari dan tampilkan data dengan perintah distinct untuk menampilkan data harga**

e) Untuk menampilkan data dengan Operator Aritmatika

Ketik perintah di bawah ini

```
mysql>select ABS(-90);  
mysql>select ACOS(-0.90);  
mysql>select ASIN(-0.90);
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select ABS(-90);  
+-----+  
| ABS(-90) |  
+-----+  
|      90 |  
+-----+  
1 row in set (0.00 sec)  
  
mysql> select Acos(-0.90);  
+-----+  
| Acos(-0.90) |  
+-----+  
| 2.6905658417935308 |  
+-----+  
1 row in set (0.05 sec)  
  
mysql> select Asin(-0.90);  
+-----+  
| Asin(-0.90) |  
+-----+  
| -1.1197695149986342 |  
+-----+  
1 row in set (0.05 sec)
```

f) Dari point e) di atas, cari dan tampilkan data dengan perintah operator aritmatika dan data sebagai berikut:

ATAN(10)	EXP(9)
ATAN2(5, 4)	FLOOR(8.98)
BIN(10)	FORMAT(12345.67, 4)
CEILING(6.123)	GREATEST(9, 200, 344, 4, 3, 1)
CONV(5, 15, 4)	HEX(15);
COS(9)	LEAST(15, 3, 0, 100, 355)
COT(9)	LOG(10)
DEGREES(5)	LOG10(15)
MOD(15, 5)	OCT(18)

PI();	POW(5, 4)
RADIANS(150)	RAND(180)
ROUND(6.43)	RAND(10)
ROUND(5.4315, 3)	SIGN(-5.5637)
SIN(300)	SQRT(50)
TAN(150)	TRUNCATE(1234.56789, 2)

**g) Dari point a) dan b) di atas, cari dan tampilkan data**

Menampilkan data dengan fungsi Agregat untuk menampilkan jumlah data record pada table buku.

Ketik perintah di bawah ini

```
mysql>select count(*) from buku;
mysql>select count(*) from buku where harga= 60000;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select count(*) from buku;
+-----+
| count(*) |
+-----+
|         8 |
+-----+
1 row in set (0.04 sec)

mysql> select count(*) from buku where harga= 60000;
+-----+
| count(*) |
+-----+
|         2 |
+-----+
```

**h) Dari point a) dan b) di atas, cari dan tampilkan dengan fungsi Agregat untuk menampilkan nilai tertinggi, nilai terendah dan rata-rata dari seluruh harga buku**

**i) Dari point a) dan b) di atas, cari dan tampilkan dengan fungsi Agregat untuk menampilkan total seluruh harga**

Kemudian akan terlihat gambar di bawah ini

```
+-----+
| 445000 |
+-----+
```



- j) Dari point a) dan b) di atas, cari dan tampilkan dengan fungsi Agregat untuk menampilkan total keseluruhan harga setelah harga buku dikalikan dengan stok buku terlebih dahulu.

Kemudian akan terlihat gambar di bawah ini



### **TUGAS PENDAHULUAN**

1. Jelaskan perbedaan operator aritmatika dengan operator pembandingan !
2. Jelaskan apa yang dimaksud dengan distinct pada Mysql ! berikan contohnya !
3. Mengapa user memerlukan tampilan data dengan fungsi distinct ?
4. Jelaskan apa yang dimaksud dengan Fungsi Agregat pada MySql !

### **TUGAS AKHIR**

1. Buatlah kesimpulan praktikum di atas !

## **PERTEMUAN X**

### **OPERASI JOIN**

#### **TUJUAN PRAKTIKUM**

- a) Mahasiswa dapat mengenal dan memahami mengenai Operasi Join pada bahasa pemrograman SQL.
- b) Mahasiswa dapat menjelaskan jenis-jenis Operasi Join.

#### **TEORI DASAR**

##### **a) Pendahuluan**

Pada bahasa SQL, operasi join atau penggabungan antar table adalah operasi dasar database relasional yang sangat penting. Untuk mendukung perancangan database relasional yang baik. Secara singkat join adalah operasi menghubungkan antar beberapa table sehingga terbentuk satu set data yang baru. Join adalah penggabungan table yang dilakukan melalui kolom / key tertentu yang memiliki nilai terkait untuk mendapatkan satu set data dengan informasi lengkap. Lengkap disini artinya kolom data didapatkan dari kolom-kolom hasil join antar table tersebut. Join diperlukan karena perancangan table pada sistem transaksional kebanyakan di-normalisasi, salah satu alasannya untuk menghindari redundansi

##### **b) Jenis-jenis Operasi Join**

###### **1) Inner Join**

Merupakan operasi join yang digunakan untuk menghasilkan kumpulan record dari dua buah table atau lebih dengan nilai yang sama, nilai tersebut merupakan irisan yang terdapat pada table-table tersebut.

```
SELECT *from table_1 INNER JOIN table_2 on table_1.nama=table_2.nama;
```

###### **2) Left Outer Join**

Merupakan operasi join yang digunakan untuk menghasilkan kumpulan record dari dua table atau lebih, dengan asumsi bahwa table sebelah kiri sebagai rujukan. Anggotanya adalah semua yang memiliki nilai sama dengan asumsi table sebelah kiri tersebut.

```
SELECT * from table_1 LEFT OUTER JOIN table_2 on table_1.nama=
table_2.nama;
```

### 3) Right Outer Join

Merupakan operasi Join yang digunakan untuk menghasilkan kumpulan record dari dua buah table atau lebih, dengan asumsi bahwa table sebelah kanan sebagai rujukan, sehingga anggotanya adalah semua yang memiliki nilai sama dengan asumsi table sebelah kanan tersebut.

```
SELECT*from table_1 RIGHT OUTER JOIN table_2 on table_1.nama=
table_2.nama;
```

### 4) Full Outer Join

Merupakan operasi join yang digunakan untuk menghasilkan kumpulan record dari dua buah table atau lebih, anggotanya adalah semua yang memiliki nilai sama ataupun tidak sama.

```
SELECT *from table_1 FULL OUTER JOIN table_2 on table_1.nama=
table_2.nama;
```

### 5) Cross Join

Cross Join merupakan bentuk penggabungan yang paling sederhana, tanpa ada kondisi.

```
SELECT field1,field2 FROM table1 CROSS JOIN table2;
```

### 6) Union

Union digunakan untuk menggabungkan beberapa perintah select hingga menghasilkan sebuah hasil yang merupakan gabungan dari beberapa perintah select tersebut.

```
SELECT field1,field2...FROM table1 UNION SELECT field1,field2...FROM
table2;
```

## 7) Subquery

Subquery adalah sebuah perintah yang memproses nilai dari hasil perintah yang lainnya.

```
SELECT * FROM table1 WHERE field = (SELECT perintah yang akan diproses  
dari table1/table2);
```

## TUGAS PRAKTIKUM

### a) Buatlah Struktur Database koperasi dan member sebagai table

Ketik perintah di bawah ini

```
mysql>create database koperasi ;  
mysql>use koperasi ;  
mysql>create table member (Id_member varchar(15) , Nama char(20) ,Alamat char  
(20) ,primary key(Id_member));  
mysql>desc member;
```

### b) Dari point a) di atas, tambahkan data sebagai berikut:

Menambah data pada table member dengan insert into

Ketik perintah di bawah ini

```
mysql>INSERT INTO member values ('1111','Anang','Cipondoh');  
mysql>INSERT INTO member values ('1112','Budi','Ciledug');  
mysql>INSERT INTO member values ('1113','Cici','Cinangka');  
mysql>INSERT INTO member values ('1114','Darma','Cikupa');  
mysql>INSERT INTO member values ('1115','Endang','Cipondoh');
```

### c) Buatlah table kedua dengan nama pinjam

Ketik perintah di bawah ini

```
mysql>create table pinjam(Nota varchar(15) , Id_member varchar(20) ,Tanggal date,  
Jumlah char (10) ,primary key(Nota));  
mysql>desc pinjam;
```

d) Tambahkan data-data di bawah ini pada table pinjam

Nota	Id_member	Tanggal	Jumlah
A100	1111	2014-02-04	500000
B100	1113	2014-02-06	700000
C100	1112	2014-02-07	400000
D100	1114	2014-02-09	900000

e) Dari point a), b), c) dan d) di atas, cari dan tampilkan data menggunakan inner join using

Ketik perintah di bawah ini

```
mysql> select * from member Inner Join pinjam using(id_member);
```

Kemudian akan terlihat gambar di bawah ini

id_member	nama	alamat	nota	tanggal	jumlah
1111	Anang	Cipondoh	A100	2014-02-04	500000
1112	Budi	Ciledug	C100	2014-02-07	400000
1113	Cici	Cinangka	B100	2014-02-06	700000
1114	Darma	Cikupa	D100	2014-02-09	900000

f) Dari point a), b), c) dan d) di atas, cari dan tampilkan data menggunakan inner join

Ketik perintah di bawah ini

```
mysql> select a.id_member,a.nama,a.alamat,b.nota,b.tanggal,b.jumlah from member a
inner join pinjam b on(a.id_member=b.id_member);
```

Kemudian akan terlihat gambar di bawah ini

id_member	nama	alamat	nota	tanggal	jumlah
1111	Anang	Cipondoh	A100	2014-02-04	500000
1112	Budi	Ciledug	C100	2014-02-07	400000
1113	Cici	Cinangka	B100	2014-02-06	700000
1114	Darma	Cikupa	D100	2014-02-09	900000

- g) Dari point a), b), c) dan d) di atas, cari dan tampilkan data menggunakan Left Outer join

Ketik perintah di bawah ini

```
mysql> select a.id_member,a.nama,a.alamat,b.nota,b.tanggal,b.jumlah from member a
left outer join pinjam b on(a.id_member=b.id_member);
```

Kemudian akan terlihat gambar di bawah ini

id_member	nama	alamat	nota	tanggal	jumlah
1111	Anang	Cipondoh	A100	2014-02-04	500000
1113	Cici	Cinangka	B100	2014-02-06	700000
1112	Budi	Ciledug	C100	2014-02-07	400000
1114	Darma	Cikupa	D100	2014-02-09	900000
1115	Endang	Cipondoh	NULL	NULL	NULL

- h) Dari point a), b), c) dan d) di atas, cari dan tampilkan data menggunakan Left Outer Join memakai where is not null

Hasil seperti gambar di bawah ini

id_member	nama	alamat	nota	tanggal	jumlah
1111	Anang	Cipondoh	A100	2014-02-04	500000
1112	Budi	Ciledug	C100	2014-02-07	400000
1113	Cici	Cinangka	B100	2014-02-06	700000
1114	Darma	Cikupa	D100	2014-02-09	900000

- i) Dari point a), b), c) dan d) di atas, cari dan tampilkan data menggunakan Right Outer Join

Hasil seperti gambar di bawah ini

id_member	nama	alamat	nota	tanggal	jumlah
1111	Anang	Cipondoh	A100	2014-02-04	500000
1112	Budi	Ciledug	C100	2014-02-07	400000
1113	Cici	Cinangka	B100	2014-02-06	700000
1114	Darma	Cikupa	D100	2014-02-09	900000

j) Dari point a), b), c) dan d) di atas, cari dan tampilkan data menggunakan Cross Join

Ketik perintah di bawah ini

```
mysql> select nama, jumlah from member cross join pinjam;
```

Kemudian akan terlihat gambar di bawah ini

nama	jumlah
Anang	5000
Anang	7000
Anang	4000
Anang	9000
Budi	5000
Budi	7000
Budi	4000
Budi	9000
Cici	5000
Cici	7000
Cici	4000
Cici	9000
Darma	5000
Darma	7000
Darma	4000
Darma	9000
Endang	5000

k) Dari point a), b), c) dan d) di atas, cari dan tampilkan data menggunakan Union

Ketik perintah di bawah ini

```
mysql> select id_member,nama,alamat from member union select  
nota,tanggal,jumlah from pinjam;
```

Kemudian akan terlihat gambar di bawah ini

id_member	nama	alamat
1111	Anang	Cipondoh
1112	Budi	Ciledug
1113	Cici	Cinangka
1114	Darma	Cikupa
1115	Endang	Cipondoh
A100	2014-02-04	500000
B100	2014-02-06	700000
C100	2014-02-07	400000
D100	2014-02-09	900000

l) Dari point a), b), c) dan d) di atas, cari dan tampilkan data menggunakan SubQuery

Ketik perintah di bawah ini

```
mysql> select *from pinjam where nota=(select max(nota)from pinjam);
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select *from pinjam where nota=(select max(nota) from pinjam);
+-----+-----+-----+-----+
| nota | id_member | tanggal | jumlah |
+-----+-----+-----+-----+
| D100 | 1114      | 2014-02-09 | 9000000 |
+-----+-----+-----+-----+
```

### TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Join dalam MySQL ?
2. Jelaskan perbedaan dari Operasi Join INNER JOIN dan OUTER JOIN !
3. Jelaskan perbedaan dari Operasi Join UNION dan Subquery !
4. Jelaskan perbedaan dari Operasi Join !

### TUGAS AKHIR

1. Buatlah kesimpulan praktikum di atas !



## **PERTEMUAN XI**

### **ROUTINE**

#### **TUJUAN PRAKTIKUM**

- a) Mahasiswa dapat mengenal dan memahami Routine pada MySQL.
- b) Mahasiswa dapat menjelaskan dan menggunakan fasilitas Routine pada MySQL.

#### **TEORI DASAR**

##### **a) Pendahuluan**

MySQL memiliki dua buah objek yang sangat berguna, yaitu fungsi tersimpan dan prosedur tersimpan. Kedua objek ini biasa disebut dengan Routine. Routine dapat meringkas beberapa perintah atau statement SQL menjadi sebuah perintah. Semua perintah yang ada di dalamnya telah diringkas menjadi sebuah fungsi. Alasan mengapa perlu meringkas sebuah perintah SQL, yaitu untuk mempermudah penggunaannya dan lebih aman dengan meringkas. Hal ini memberikan keamanan dan memperkecil pemberian hak akses kepada masing-masing user, sehingga hanya perlu memberikan hak untuk menjalankan atau mengeksekusi objek prosedur tersimpan dan fungsi tersimpan tanpa harus memberikan hak untuk memanipulasi table aslinya.

##### **b) Menggunakan Fasilitas Routine Pada Mysql**

Fasilitas Stored routines terdiri dari Procedure dan Function. Stored Procedure adalah kumpulan dari perintah SQL yang dapat disimpan di dalam server layaknya routine coding pada program aplikasi. Dengan stored procedure, setiap client tidak perlu menyimpan routine code sendiri yang berhubungan dengan database tetapi hanya dengan mengaksesnya di server. Ini akan sangat membantu memperkecil jalannya lalu lintas data pada sebuah jaringan. Sedangkan Stored Function bermanfaat agar dapat membuat fungsi.

Sebelum membuat sebuah obyek prosedur tersimpan dan fungsi tersimpan, harus dipastikan User mempunyai hak-hak berikut :

- 1) Create routines adalah hak yang diperlukan untuk membuat sebuah obyek fungsi tersimpan dan prosedur tersimpan.

- 2) Alter routines adalah hak yang diperlukan untuk menghapus atau mengubah isi dari sebuah objek prosedur tersimpan dan fungsi tersimpan.
- 3) Execute adalah hak yang digunakan untuk mengeksekusi obyek prosedur tersimpan dan fungsi tersimpan.
- 4) Fungsi tersimpan (stored function) untuk membuat sebuah obyek fungsi tersimpan menggunakan perintah

## TUGAS PRAKTIKUM

- a) **Buatlah database dengan nama Routine.**
- b) **Dari point a) buatlah objek fungsi tersimpan dengan nama fungsi tersimpan fungsi1, yang mendeklarasikan variabel parameter a tipe data smallint dengan create function.**

Ketik perintah di bawah ini

```
mysql> create function fungsi1(a smallint) returns int return(a+a);
mysql> select fungsi1(60);
```

Kemudian akan terlihat gambar di bawah ini

```
+-----+
| fungsi1 (60) |
+-----+
|           120 |
+-----+
```

- c) **Dari point a) buatlah objek fungsi tersimpan dengan nama fungsi tersimpan fungsi2, yang mendeklarasikan variabel kar tipe data char dengan create function.**

Ketik perintah di bawah ini

```
mysql> create function fungsi2(kar char(50)) returns int return length(kar);
mysql> select fungsi2('MySQL');
```

Kemudian akan terlihat gambar di bawah ini

```
+-----+
| fungsi2('MySQL') |
+-----+
|                5 |
+-----+
```

- d) Dari point a), b) dan c) di atas, buat dan tampilkan data tersimpan fungsi3 yang mendeklarasikan variabel parameter a tipe data smallint dengan create function

Hasil seperti gambar di bawah ini

```
mysql> select fungsi3 (10);
+-----+
| fungsi3 (10) |
+-----+
|          110 |
+-----+
1 row in set (0.00 sec)
```

- e) Dari point a) dan b) di atas, tampilkan informasi mengenai objek fungsi tersimpan fungsi1 yg telah dibuat sebelumnya dengan show create

Ketik perintah di bawah ini

```
mysql> show create function fungsi1;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> show create function fungsi1;
+-----+
| Function | sql_mode | Create Function | character_set_client |
+-----+
| fungsi1 | NO_ENGINE_SUBSTITUTION | CREATE DEFINER='root'@'localhost' FU | latin1_swedish_ci |
+-----+
| 'fungsi1' | (a SMALLINT) RETURNS int(11) | return a+a | cp850 | cp850_general_ci | latin1_swedish_ci |
+-----+
```

- f) Dari point a), b), c) dan d) di atas, tampilkan informasi mengenai objek fungsi tersimpan fungsi2 dan fungsi3 yg telah dibuat sebelumnya dengan show create
- g) Dari point a), b), c) dan d) di atas, hapuslah fungsi tersimpan fungsi3 yang telah dibuat sebelumnya dengan drop function

Ketik perintah di bawah ini

```
mysql> drop function fungsi3;  
mysql> show create function fungsi3;
```

**h) Buatlah Struktur Database sekolah dan siswa sebagai table**

Ketik perintah di bawah ini

```
mysql> create database sekolah;  
mysql> use sekolah;  
mysql> create table siswa(nis varchar(15) , nama char(20), angkatan  
varchar(30), primary key(nis));
```

**i) Dari point h) di atas, tambahkan data sebagai berikut:**

Ketik perintah di bawah ini

```
mysql> INSERT INTO siswa values ('11234', 'ana', '2008/2009',);  
mysql> INSERT INTO siswa values ('11235', 'bayu', '2009/2010',);  
mysql> INSERT INTO siswa values ('11236', 'canda', '2010/2011',);  
mysql> INSERT INTO siswa values ('11237', 'dirga', '2012/2013',);  
mysql> INSERT INTO siswa values ('11238', 'endang', '2013/2014',);
```

**j) Dari point h) dan i) di atas, pembuatan procedure menggunakan create**

Ketik perintah di bawah ini

```
mysql> delimiter //  
mysql> create procedure jumlahsiswa (out parameter1 int)  
> begin  
> select count(*) into parameter1 from siswa;  
> end;  
> //  
mysql> delimiter ;
```

**k) Dari point h), i) dan j) di atas, pemanggilan procedure menggunakan perintah call**

Ketik perintah di bawah ini:

```
mysql> call jumlahsiswa (@a);  
mysql> select (@a);
```

Kemudian akan terlihat seperti gambar di bawah ini

```

+-----+
| 2a    |
+-----+
|      5 |
+-----+
1 row in set (0.00 sec)

```

- l) Buatlah database dengan nama toko dengan table dengan nama barang, seperti dari point h), i), j) dan k) di atas, buatlah procedure dan Pemanggilan dari database yang dibuat dengan data seperti di bawah ini

NamaBarang	Satuan	Harga	Jumlah
Sabun	Bungkus	3000	20
Sikat gigi	Bungkus	4000	50
Sampho	Botol	10000	30
Kopi	Kg	30000	20
Teh	Bungkus	5000	40
Beras	Kg	10000	10

## TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Routine ?
2. Jelaskan perbedaan antara fungsi tersimpan dan prosedur tersimpan !
3. Mengapa dengan menggunakan Routine dapat mempermudah penggunaannya bagi user ?
4. Mengapa dengan menggunakan Routine dapat lebih menjaga keamanan suatu database ?

## TUGAS AKHIR

1. Buatlah kesimpulan praktikum di atas !

## **PERTEMUAN XII**

### **TRIGGER & VIEW**

#### **TUJUAN PRAKTIKUM**

- a) Mahasiswa dapat mengenal dan memahami Trigger dan View pada MySQL.
- b) Mahasiswa dapat menjelaskan dan menggunakan fasilitas Trigger dan View pada MySQL.

#### **TEORI DASAR**

##### **a) Pendahuluan**

Trigger, Stored Procedure/Function, dan View merupakan komponen dan fitur database, yang dengan keunikan fungsi masing-masing dapat dimanfaatkan untuk menjaga, mengelola, dan membantu kinerja database engineer dalam upaya terjaminnya integritas sebuah database.

##### **b) Fasilitas Trigger Pada Mysql**

Trigger adalah sebuah objek yang bekerja bersama sebuah table dan digunakan untuk memeriksa sebuah operasi atau kejadian tertentu pada sebuah table. Objek ini akan diaktifkan atau dijalankan oleh MySQL apabila terjadi operasi atau kejadian tertentu pada sebuah table. Trigger digunakan untuk memeriksa sebuah operasi terhadap table sebelum operasi di jalankan MySQL. Operasi yang dapat menyebabkan sebuah trigger diaktifkan terdiri atas tiga,yaitu:

- 1) Insert adalah operasi ketika terjadi penambahan data pada sebuah table.
- 2) Delete adalah operasi ketika terjadi penghapusan data pada sebuah table.
- 3) Update adalah operasi ketika terjadi pengubahan isi data yang telah ada pada sebuah table.

Sintaks CREATE TRIGGER dapat ditulis sebagai berikut:

**CREATE TRIGGER** nama\_trigger on nama\_table **FOR EACH ROW** operasi;

### c) Fasilitas View Pada Mysql

View atau Table View adalah table yang bisa menampilkan informasi dari satu table atau lebih. View berisi definisi suatu perintah untuk melihat satu table atau lebih dalam sebuah table baru yang merupakan maya, karena data yang di ambil dan simpan tetap dalam table aslinya. Setelah sebuah view di buat, selanjutnya View bisa dipakai untuk mengakses table. Dengan demikian satu atau gabungan beberapa table dapat dibuat berbagai macam view sesuai informasi yang diinginkan. Masing-masing view akan dilihat sebagai sebuah table tersendiri. Sintaks CREATE VIEW dapat ditulis sebagai berikut:

```
CREATE [OR REPLACE] VIEW view_nama AS perintah_select;
```

Beberapa keuntungan penggunaan View antara lain adalah:

- 1) Tidak memperumit pengolahan dan penyimpanan data, meskipun kita membuat berbagai macam tampilan data dengan View. Karena data tetap tersimpan dalam table aslinya sehingga pemasukan data, editing, penghapusan dan pengolahan data lainnya dikerjakan seperti biasanya.
- 2) Dengan membuat view, suatu tampilan data tertentu yang sering dibutuhkan menjadi sangat mudah dilaksanakan, karena tinggal menampilkan view saja.

## TUGAS PRAKTIKUM

### a) Buatlah Struktur Database rumahsakit dengan pasien sebagai table

Ketik perintah di bawah ini

```
mysql>create database rumahsakit ;  
mysql>use rumahsakit ;  
mysql>create table pasien (id_pasien varchar(5) , nama char(20), alamat char(20),  
no_antrian varchar(5), tgl_masuk date, primary key(id_pasien));
```

### b) Dari point a) di atas, tambahkan data sebagai berikut:

Ketik perintah di bawah ini

```
mysql>insert into pasien values ('a1','yahya','pondok cabe','a11','2014-05-21');  
mysql>insert into pasien values ('a2','yanto','cinere','a12','2014-05-23');  
mysql>insert into pasien values ('a4','wandi','ciputat','a14','2014-05-24');  
mysql>select *from pasien;
```

c) Dari point a) dan b) di atas, buatlah objek trigger dengan operasi insert

Ketik perintah di bawah ini

```
mysql>set @jmlpenambahan=0;
mysql>create trigger trigger1 before insert on pasien for each row set
@jmlpenambahan=@jmlpenambahan +1;
mysql>insert into pasien values ('a3','lulu','cinere','a15','2014-05-24');
mysql>insert into pasien values ('a5','sari','cilandak','a16','2014-05-25');
mysql>insert into pasien values ('a6','bari','cinere','a16','2014-05-25');
mysql>select @jmlpenambahan;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select @jmlpenambahan;
+-----+
| @jmlpenambahan |
+-----+
|                3 |
+-----+
1 row in set (0.00 sec)
```

d) Buatlah Struktur Database bioskop dan JadwalFilm sebagai table

Ketik perintah di bawah ini

```
mysql>create database bioskop ;
mysql>use bioskop ;
mysql>create table JadwalFilm (Id_film varchar(15) , Judul char(20),Waktu
datetime, primary key(Id_Film));
mysql>desc JadwalFilm;
```

e) Dari point d) di atas, tambahkan data sebagai berikut:

Ketik perintah di bawah ini

```
mysql>insert into JadwalFilm values ('D11','In Fear','2014-03-07 18:30:00');
mysql>insert into JadwalFilm values ('H12','Haunt','2014-03-07 19:00:00');
mysql>insert into JadwalFilm values ('C13','Bad Words','2014-03-07 19:30:00');
mysql>insert into JadwalFilm values ('A14','Divergent','2014-03-07 20:00:00');
mysql>insert into JadwalFilm values ('E15','Enemy','2014-03-07 20:30:00');
```



f) Dari point d) di atas, buatlah table kedua dengan nama stdio

Ketik perintah di bawah ini

```
mysql>create table stdio(kode_stdio varchar(15) , namaStdio char(20) ,Id_Film
varchar(10),judul char(20),primary key(kode_stdio));
mysql>desc stdio;
```

g) Tambahkan data pada table stdio dengan data sebagai berikut:

Kode_stdio	NamaStdio	Id_film	Judul
STD4	Stdio 4	E15	Enemy
STD3	Stdio 3	D11	In Fear
STD2	Stdio 2	C13	Bad Words
STD5	Stdio 5	A14	Divergent
STD1	Stdio 1	H12	Haunt

h) Dari point d), e), f) dan g) di atas, buatlah objek view dengan perintah create view

Ketik perintah di bawah ini

```
mysql> create view tblview as select jadwalfilm.judul,jadwalfilm.waktu,namaStdio
from jadwalfilm,stdio where jadwalfilm.id_film=stdio.id_film;
mysql>select *from tblview;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> select *from tblview;
+-----+-----+-----+
| judul      | waktu                | namaStdio |
+-----+-----+-----+
| Divergent  | 2014-03-07 20:00:00 | stdio 5   |
| Bad Words  | 2014-03-07 19:30:00 | stdio 2   |
| In Fear    | 2014-03-07 18:30:00 | stdio 3   |
| Enemy      | 2014-03-07 20:30:00 | stdio 4   |
| Haunt      | 2014-03-07 19:00:00 | stdio 1   |
+-----+-----+-----+
```

- i) Dari point d), e), f) dan g) di atas, tampilkan data-data dari point h) dengan perintah dengan tblview, select dan where sehingga menjadi seperti gambar di bawah ini

judul	waktu	namaStudio
Haunt	2014-03-07 19:00:00	studio 1

### **TUGAS PENDAHULUAN**

1. Jelaskan apa yang dimaksud dengan trigger !
2. Jelaskan apa yang dimaksud dengan view !
3. Jelaskan kelebihan dari penggunaan trigger dan view !
4. Jelaskan operasi-operasi apa saja yang menyebabkan sebuah trigger diaktifkan !

### **TUGAS AKHIR**

1. Buatlah kesimpulan praktikum di atas !

## **PERTEMUAN XIII**

### **BACKUP & RESTORE**

#### **TUJUAN PRAKTIKUM**

- a) Mahasiswa dapat mengenal dan memahami Backup dan Restore pada MySQL.
- b) Mahasiswa dapat menjelaskan dan menggunakan fasilitas Backup dan Restore pada MySQL.

#### **TEORI DASAR**

##### **a) Pedahuluan**

Database merupakan suatu sistem yang harus dapat diandalkan kinerjanya. Akan tetapi, suatu sistem database tidak selamanya dapat berjalan dengan lancar. Adakalanya suatu database mengalami gangguan. Gangguan-gangguan itu dapat menyebabkan kerusakan data pada sistem tersebut. Kerusakan data pada sistem database dapat dicegah dengan berbagai macam teknik. Untuk pencegahan kerusakan data tersebut dapat dilakukan dengan menggunakan metode backup dan restore serta replikasi server MySQL. Metode backup dan restore merupakan metode yang sudah lama digunakan untuk mencegah kerusakan data. Metode ini dijalankan dengan membuat salinan dari data yang ada pada server secara berkala. Jika ada kerusakan pada data yang berada pada server MySQL maka harus diusahakan untuk memperbaiki data tersebut dengan mengambil salinannya.

##### **b) Fasilitas Backup**

Proses backup data merupakan hal yang sangat penting dilakukan. Hal ini diperlukan untuk mengantisipasi hal-hal yang tidak diinginkan di database kita, misalnya hilangnya data, rusaknya database dsb. Sebaiknya proses backup dilakukan secara rutin dan terus-menerus. Backup di MySQL sebenarnya ada 2 jenis, yaitu secara otomatis dan manual. Secara otomatis kita dapat menggunakan konsep replication, dimana server database kita secara real-time di-backup dengan server lain. Jika terdapat perubahan di server utama kita, maka secara otomatis perubahannya akan direplikasike server kedua.

```
mysqldump nama_database > lokasi_backup -u root
```

**c) Fasilitas Restore**

Restore merupakan prosedur yang dilaksanakan untuk mengembalikan file yang dibackup ke database MySQL. Proses restore juga dapat dilakukan melalui perintah SQL dengan memanfaatkan mysql dan juga bisa menggunakan software bantuan.

```
mysql nama_database < lokasi_backup -u root
```

**TUGAS PRAKTIKUM**

**a) Buatlah Struktur Database elektronik dan Televisi sebagai table**

ketik perintah di bawah ini

```
mysql>create database Elektronik ;
mysql>use Elektronik ;
mysql>create table Televisi (kode_tv varchar(15) , merk char(20) ,Tipe char (20) ,unit
char (5) ,primary key(kode_tv));
```

**b) Dari point a) di atas, tambahkan data sebagai berikut:**

Ketik perintah di bawah ini

```
mysql>insert into Televisi values ('a1','Samsung','AF2','21');
mysql>insert into Televisi values ('a2','Toshiba','AL2','15');
mysql>insert into Televisi values ('a3','Politron','AC1','13');
mysql>insert into Televisi values ('a4','Sharp','AF1','10');
mysql>select *from Televisi;
mysql>show tables;
mysql>exit
```

**c) Dari point a) dan b) di atas, buatlah backup data dengan mysqldump**

Buatlah folder di D (bukan di C) dengan nama **semester5** dan namai simpan backup1.sql dalam folder tersebut.ketik perintah di bawah ini

```
C:\xampp\mysql\bin>cd\
C:\>mysqldump elektronik > D:\semester5\backup1.sql -u root
```

- d) Dari point a) dan b) di atas, lakukan penghapusan database Elektronik dan buat kembali database dengan nama yang sama

ketik perintah di bawah ini

```
mysql> drop database Elektronik;  
mysql> create database Elektronik;  
mysql> use Elektronik;  
mysql> show tables;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> drop database Elektronik;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> create database Elektronik;  
Query OK, 1 row affected (0.01 sec)  
  
mysql> use Elektronik;  
Database changed  
mysql> show tables;  
Empty set (0.00 sec)
```

- e) Dari point a), b), c) dan d) di atas, buatlah restore data, dengan keluar mysql

ketik perintah di bawah ini

```
mysql> exit  
C:\xampp\mysql\bin>mysql elektronik < "D:\semester5\backup1.sql" -u root  
(file "backup1.sql" yang disimpan di "D:\semester5\" telah direstore ke database elektronik)  
C:\xampp\mysql\bin>mysql-u root
```

- f) Dari point a), b), c), d) dan e) di atas, cek struktur database yang telah direstore

ketik perintah di bawah ini

```
mysql> use Elektronik;  
mysql> show tables;  
mysql> select *from Televisi;
```

Kemudian akan terlihat gambar di bawah ini

```
mysql> use elektronik;
Database changed
mysql> show tables;
+-----+
| Tables_in_elektronik |
+-----+
| televisi              |
+-----+
1 row in set (0.00 sec)

mysql> select *from televisi;
+-----+-----+-----+-----+
| kode_tv | merk      | tipe  | unit |
+-----+-----+-----+-----+
| a1      | Samsung  | AF2   | 21   |
| a2      | Toshiba  | AL2   | 15   |
| a3      | Politron | AC1   | 13   |
| a4      | Sharp    | AF1   | 10   |
+-----+-----+-----+-----+
```

e) **Buatlah Struktur Database berdasarkan table di bawah ini**

nama database : Perpustakaan dan nama table : Buku

Kode	Judul	Pengarang	stok	jumlah_terpinjam
A11	Pemograman Java	Sutendi	5	5
A12	MySQL dan ERD	Yudi Priyadi	6	4
A13	PHP mudah	RolanSyah	4	6
A14	Algoritma	DianaAndre	2	5

f) **Dari point e) di atas, buatlah backup data dengan mysqldump dan disimpan dengan nama backup2.sql pada direktori D:/Semester5**

g) **Dari point e) dan f) di atas, hapus database perpustakaan dan buat kembali database dengan nama perpustakaan\_baru**

h) **Dari point e), f) dan g) di atas, buatlah proses restore data**

- i) Dari point e), f), g) dan h) di atas, cek struktur database yang telah direstore

Hasil seperti gambar di bawah ini

```
mysql> use perpustakaan_baru;
Database changed
mysql> show tables;
+-----+
| Tables_in_perpustakaan_baru |
+-----+
| buku                          |
+-----+
1 row in set (0.00 sec)

mysql> select *from buku;
+-----+-----+-----+-----+-----+
| kode | Judul          | Pengarang | stok | jumlah_terpinjam |
+-----+-----+-----+-----+-----+
| A11  | Pemograman Java | Sutendi   | 5    | 5                |
| A12  | MySQL dan ERD   | Yudi Priyadi | 6    | 4                |
| A13  | PHP mudah       | RolanSyah | 4    | 6                |
| A14  | Algoritma       | DianaAndre | 2    | 5                |
+-----+-----+-----+-----+-----+
```

## TUGAS PENDAHULUAN

1. Jelaskan apa yang dimaksud dengan backup ?
2. Jelaskan apa yang dimaksud dengan restore ?
3. Jelaskan secara singkat tahapan-tahapan mem-backup sebuah data pada MySQL !
4. Jelaskan secara singkat tahapan-tahapan me-restore sebuah data pada MySQL !

## TUGAS AKHIR

1. Buatlah kesimpulan praktikum di atas !

## **PERTEMUAN XIV**

### **MYSQL GUI TOOLS**

#### **TUJUAN PRAKTIKUM**

- a) Mahasiswa dapat mengenal dan memahami aplikasi-aplikasi MySQL Client.
- b) Mahasiswa dapat menjelaskan dan menggunakan phpMyAdmin sebagai aplikasi MySQL Client berbasis GUI.

#### **TEORI DASAR**

##### **a) Pendahuluan**

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya secara cepat, multi user serta menggunakan perintah dasar SQL (Structured Query Language). Bekerja dengan Server MySQL memerlukan Client MySQL sebagai pehubungnya. Jenis Client MySQL ada yang berupa Command Line atau GUI. Berikut ini adalah beberapa daftar software client MySQL.

- 1) SQLYog adalah aplikasi client desktop yang sangat terkenal terutama di kalangan pengguna Indonesia.
- 2) phpMyAdmin merupakan aplikasi web yang dapat digunakan untuk manajemen dan administrasi server dan database serta objek-objek yang terdapat di dalamnya.
- 3) MySQL GUI Tools merupakan paket aplikasi yang terdiri dari MySQL Administrator, MySQL Query Browser, MySQL Migration.
- 4) Maatkit berbagai utilitas berbasis PERL yang berguna untuk melakukan tugas-tugas rumit dengan mudah.

##### **b) phpMyAdmin**

phpMyAdmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyadmin, dapat membuat database, membuat table, menginsert, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual. phpMyAdmin menawarkan fasilitas yang mencakup pengelolaan keseluruhan server MySQL dan juga basis data tunggal. Karena berbasis web, maka phpMyAdmin dapat di jalankan di banyak Sistem Operasi, selama dapat menjalankan webserver dan Mysql. phpMyAdmin juga mempunyai



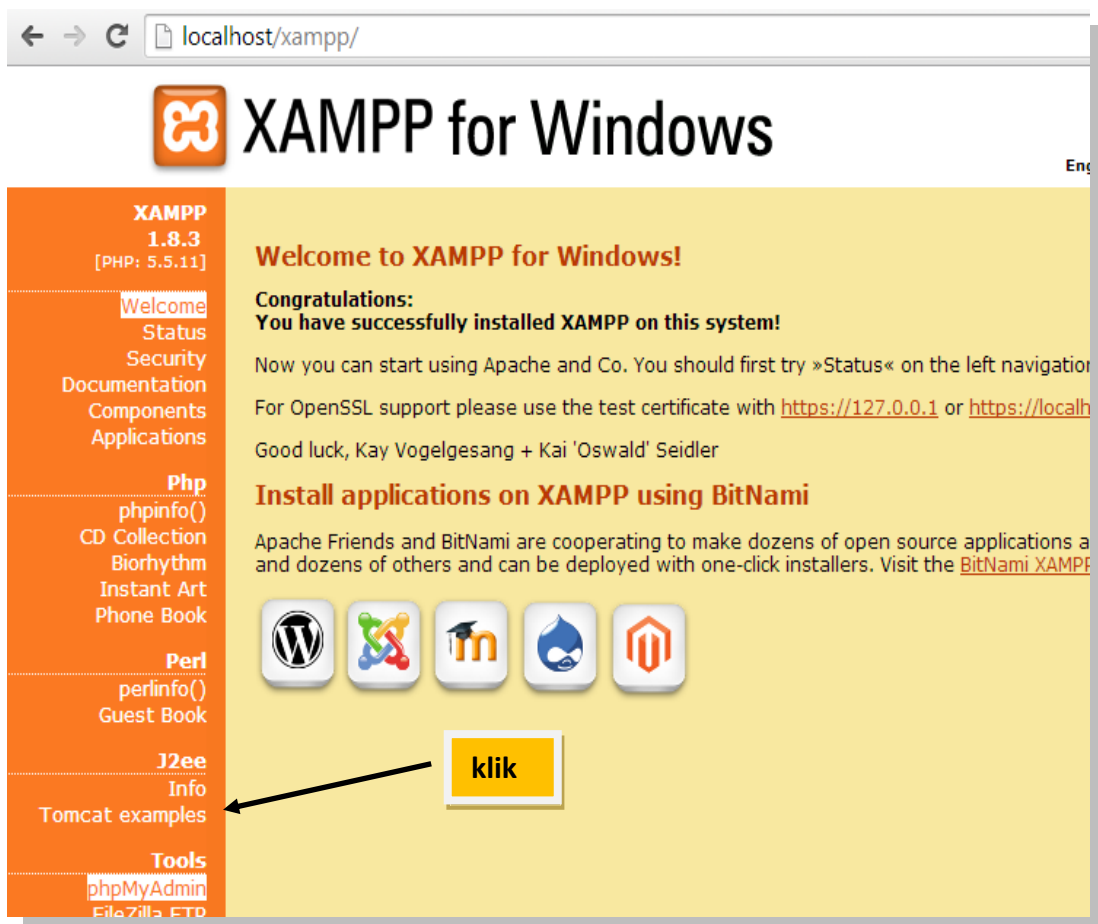
sistem internal yang digunakan untuk mengelola metadata dan mendukung fitur-fitur untuk operasi tingkat lanjut. Melalui sistem administrator, phpMyAdmin juga dapat mengelola users dan sekaligus hak aksesnya (privilage). Biasanya aplikasi phpMyAdmin sudah ada dalam software XAMPP atau WAMP.

## TUGAS PRAKTIKUM

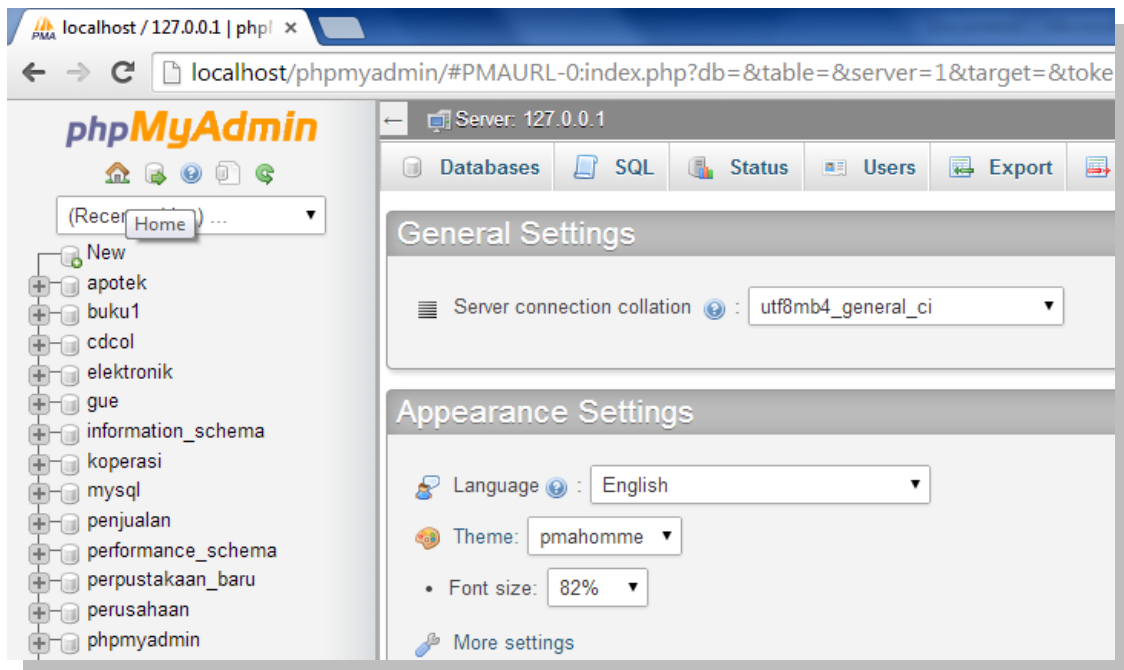
### a) Mengakses phpMyAdmin

- 1) Install XAMPP
- 2) Buka Web Browser (Mozilla Firefox atau Internet Explorer)
- 3) Ketik `http://localhost/` <enter>

Kemudian akan terlihat gambar di bawah ini

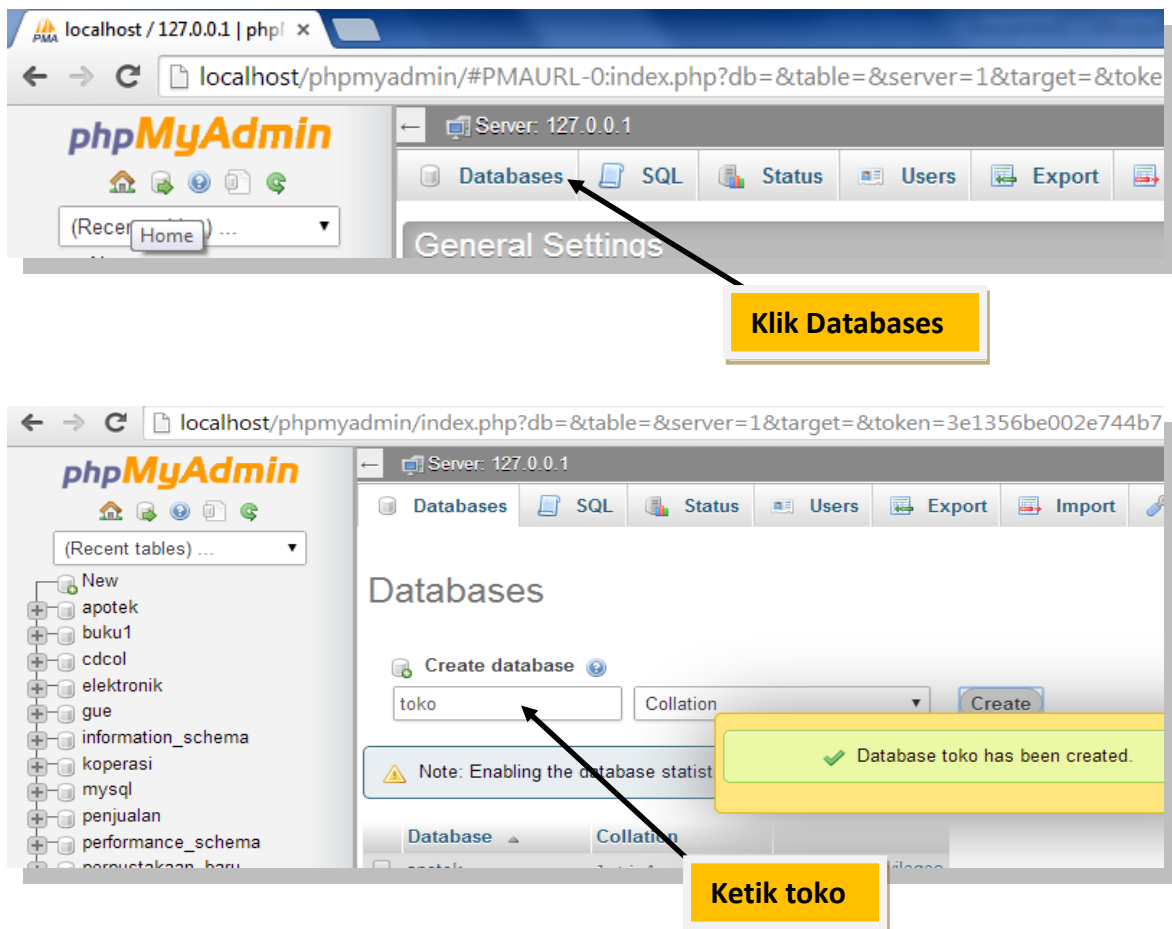


klik phpMyAdmin, kemudian akan terlihat gambar di bawah ini



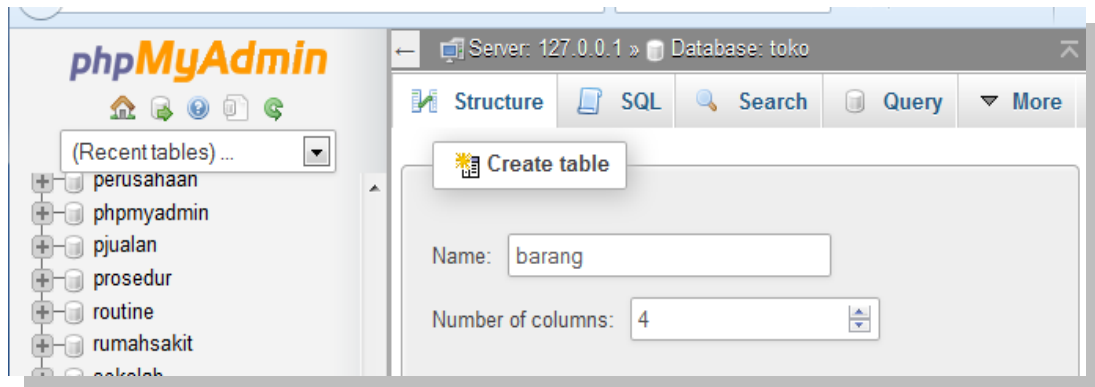
**b) Buat database toko**

Seperti gambar di bawah ini



1) buat table barang

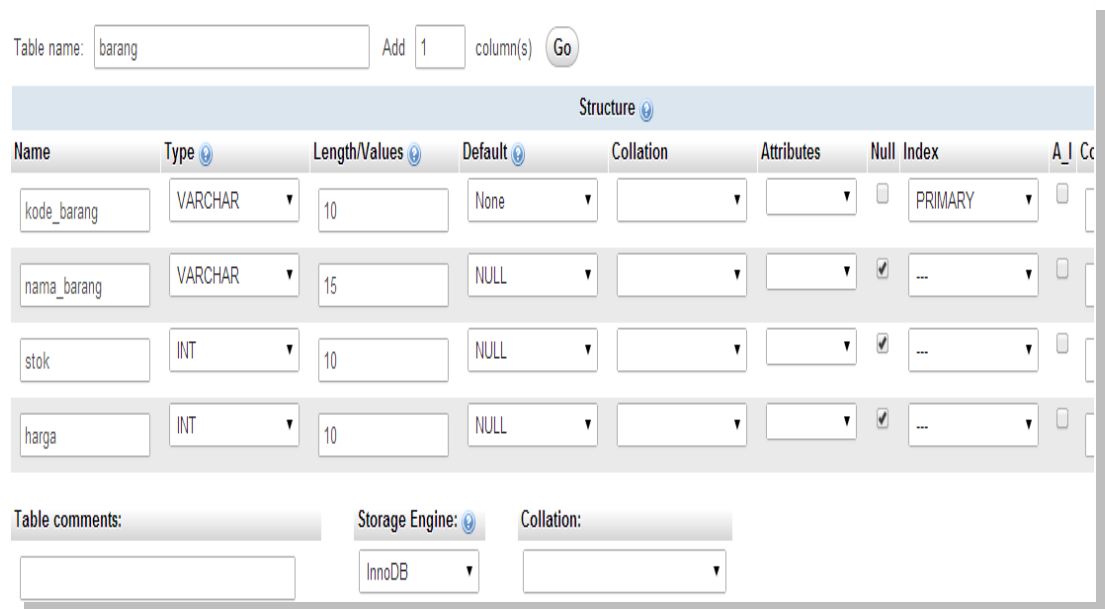
Kemudian akan terlihat gambar di bawah ini



2) Buat field pada table barang

Nama	Tipe data	Length	Key
kode_barang	varchar	10	Primary
nama_barang	varchar	15	-
stok	Int	10	-
harga	Int	10	-

Seperti gambar di bawah ini



c) Dari point b) di atas, tambahkan data sebagai berikut:

kode_barang	nama_barang	stok	harga
a1	buku tulis	15	5000
a2	Pulpen	20	3000
a3	Penggaris	10	2000
a4	tipeX	15	5000

Seperti gambar di bawah ini

Server: 127.0.0.1 » Database: toko » Table: barang

Column Type Function Null Value

kode\_barang varchar(15) a4

nama\_barang varchar(10) tipeX

stok int(15) 15

harga int(10) 5000

Go

Klik Insert (tambah data)

Kemudian klik “Go”

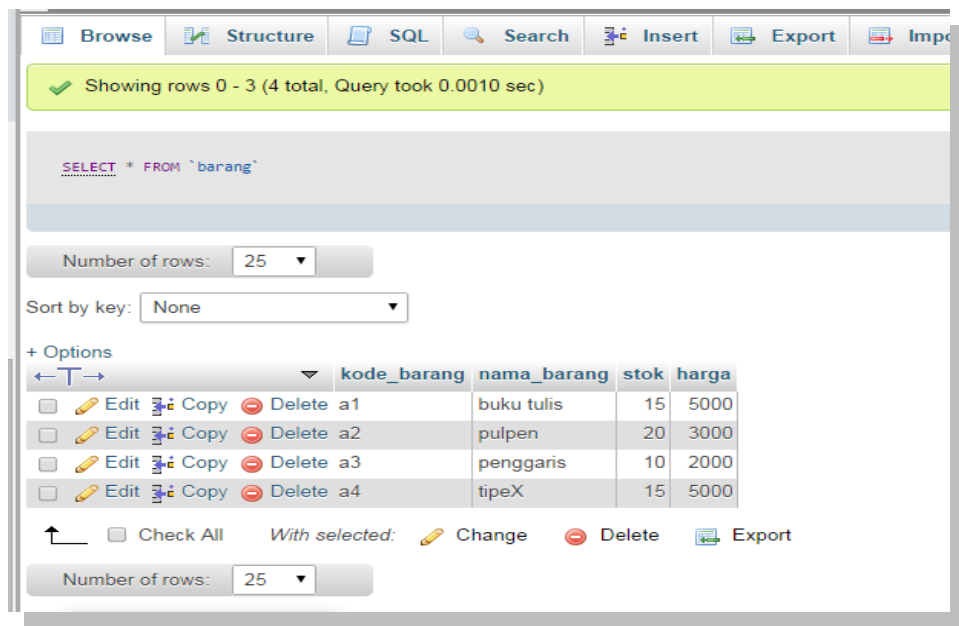
d) Dari point b) dan c) di atas, tampilkan data pada table barang

Server: 127.0.0.1 » Database: toko » Table: barang

Browse Structure SQL Search Insert Export

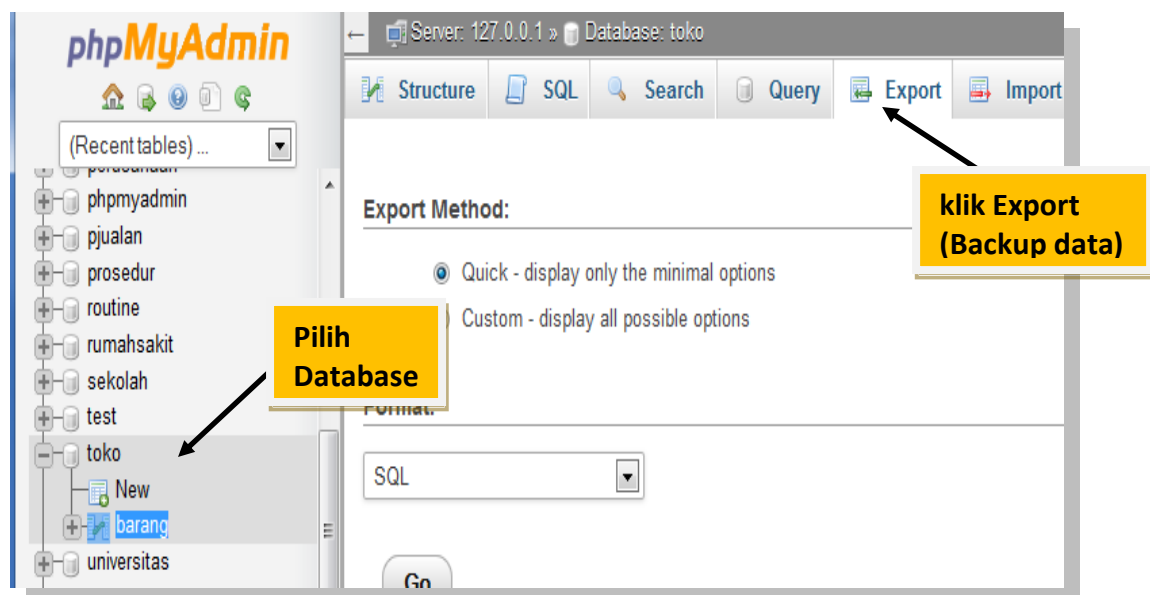
klik Browse

Kemudian akan terlihat gambar di bawah ini



**e) Dari point b), c) dan d) di atas, buatlah backup data**

- 1) Jika hanya satu table yang dibackup, pilih database, kemudian pilih table, lalu klik Export
- 2) Jika semua table yang dibackup, pilih database, lalu klik Export



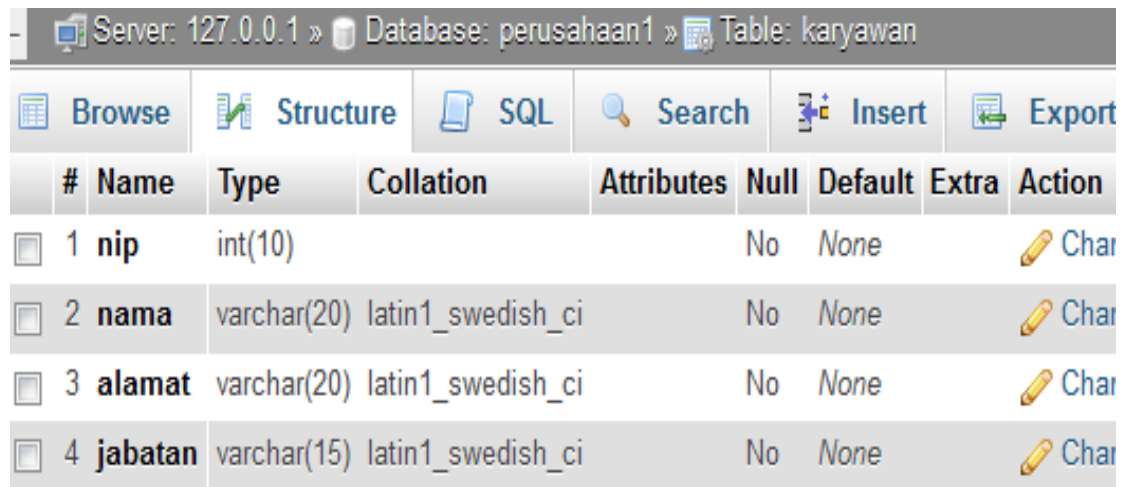
Setelah itu klik Go → save (simpan di d:\Semester5\backupGUI.sql) → Ok

**f) Dari point b), c), d) dan e) di atas, buatlah restore data**

- 1) Buat database dengan nama toko\_baru
- 2) Kemudian pilih database toko\_baru, lalu klik import
- 3) Klik browse → cari file backup → ok → klik Go

**g) Buat database perusahaan1 dan karyawan sebagai table**

Seperti gambar di bawah ini



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	nip	int(10)			No	None		Char
2	nama	varchar(20)	latin1_swedish_ci		No	None		Char
3	alamat	varchar(20)	latin1_swedish_ci		No	None		Char
4	jabatan	varchar(15)	latin1_swedish_ci		No	None		Char

**h) Dari point g) di atas, tambahkan data sebagai berikut:**

Kemudian akan terlihat gambar di bawah ini

nip	nama	alamat	jabatan
20111	Toni	Ciledug	Teknik
20112	Vina	Depok	HRD

**i) Dari point g) dan h) di atas, buatlah backup data dengan nama “backupGUI2.sql”**

**j) Dari point g), h) dan i) di atas, buatlah proses restore data backupGUI2.sql ke database yang baru yaitu perusahaan2. Kemudian cek struktur database perusahaan2.**

### **TUGAS PENDAHULUAN**

1. Jelaskan apa yang dimaksud dengan Webr Server dan phpMyAdmin !
2. Jelaskan apa yang dimaksud dengan aplikasi client-server !
3. Jelaskan perbedaan dari aplikasi berbasis GUI dan Command Line !
4. Jelaskan perbedaan dari database yang bersifat Stand Alone dan Server !

### **TUGAS AKHIR**

1. Buatlah kesimpulan praktikum di atas!