

## PENDAHULUAN

1. Mata pelajaran : Praktikum Basis Data Dengan Database Server MySQL
3. Program Studi : RPL
4. Tujuan Inst. Umum : Setelah mengikuti mata pelajaran ini peserta didik mampu menggunakan perangkat lunak database jaringan MySQL untuk pengelolaan database yang kompleks

No.	Pokok Bahasan	Tujuan Instruksional Khusus	Praktikum Ke
1	Mengenal MySQL	Setelah mengikuti materi pembelajaran ini peserta didik dapat menjelaskan mengenai MySQL, Cara Mendapatkan Softwarena dan perbedaannya dengan DBMS yang lain.	I
2	Memulai Menggunakan MySQL	Setelah mengikuti materi pembelajaran ini peserta didik dapat memulai menggunakan MySQL, menjelaskan tipe data di MySQL dan menggunakan perintah-perintah dasar MySQL.	II
3	Kekangan nilai Pada Data	Setelah mengikuti materi pembelajaran ini peserta didik dapat Membuat Tabel dengan MySql beserta batasanbatasannya.	III
4	Perintah Manipulasi Data	Setelah mengikuti materi pembelajaran ini peserta didik dapat menggunakan perintah-perintah MySql untuk memanipulasi Data	IV
5	Perintah Manipulasi Data Lanjutan	Setelah mengikuti materi pembelajaran ini peserta didik dapat menggunakan perintah-perintah manipulasi data lanjutan seperti pengelompokan, pengurutan, fungsi agregasi dan kombinasinya.	V
6	Fungsi-Fungsi SQL	Setelah mengikuti materi pembelajaran ini peserta didik dapat menggunakan fungsifungsi SQL, ekspresi waktu dan tanggal serta kombinasinya.	VI dan VII

7	Query Antar Tabel	Setelah mengikuti materi pembelajaran ini peserta didik dapat merelasikan beberapa tabel untuk penyajian informasi	VIII dan IX
8	Query Antar Table Lanjutan	Setelah mengikuti materi pembelajaran ini peserta didik dapat pengelolaan tabel dengan menggunakan beberapa jenis join dan sub query sehingga penyajian informasi lebih mudah	X
9	View & Indeks	Setelah mengikuti materi pembelajaran ini peserta didik dapat membuat view agar pengelolaan tabel-tabel yang rumit dapat lebih mudah dan membuat indeks sehingga mempercepat pengelolaan data	XI
10	Manajemen User	Setelah mengikuti materi pembelajaran ini peserta didik dapat mengaktifkan keamanan standar, membuat user baru dan mengatur hak akses terhadap data sehingga data lebih terjamin keamanannya.	XII
11	Sistem Administrasi Database	Setelah mengikuti materi pembelajaran ini peserta didik dapat melaksanakan tugas tugas administrator yaitu backup dan pemeriksaan.	XIII
12	Studi Kasus Database	Setelah mengikuti materi pembelajaran ini peserta didik dapat melakukan sinergi terhadap materi-materi sebelumnya untuk menyelesaikan masalah pengelolaan database untuk aplikasi yang sesungguhnya.	XIV

5. Persiapan : Peserta didik harus sudah dapat masuk ke jaringan komputer

laboratorium agar dapat bekerja dan menggunakan MySQL.

4. Pengetahuan Prasyarat : Teori Basis Data

5. Teori Dasar :

## KONSEP DATABASE SERVER

Client / server secara sederhana dapat diartikan suatu system yang membagi fungsi dan proses system antara server (back end) yang mengolah database dengan client (front end) yang menjalankan aplikasi dengan tujuan mengurangi beban server, client dan lalu lintas data di jaringan. Istilah client / server yang diterima secara umum dimaksudkan sebagai multiuser client server. Dalam hal ini terdapat beberapa terminal yang berfungsi sebagai front end (client) yang berhubungan dengan satu atau lebih back end (server).

Penekanan konsep client / server lebih pada model software dan bukan pada hardware. Sehingga dari konfigurasi perangkat keras sepintas kita lihat bahwa komputer di LAN tidak ada bedanya dengan client / server. Namun dengan perkembangan teknologi jaringan dan middleware serta konsep client / server yang memisahkan database dengan aplikasi yang akan mengakses database, memungkinkan konfigurasi perangkat keras client / server lebih kompleks.

Untuk mencapai hasil yang optimal dalam penerapan system client / server, terdapat beberapa factor yang harus diperhatikan seperti implementasi business rules, teknik pemrograman, hardware, databases dan application tools.

### Komponen Client

Client merupakan sisi yang langsung berinteraksi dengan user yang terdiri dari hardware dan software.

Komponen secara hardware :

Dalam pengelolaan database, personal computer yang aa sekarang ini telah cukup memadai untuk digunakan sebagai client.

Komponen secara software : a.

Sistem Operasi

b. Development tools

c. Query / reporting Application

d. Data integration & Analysis Application

### Komponen Server

Agar suatu server dapat memberikan hasil yang lebih optimal selain pemilihan perangkat keras dan perangkat lunak / database yang tepat, dapat dipertimbangkan penggunaan distributed processing / distributed server, yaitu suatu system yang memisahkan proses dilebih dari satu server. Kebalikannya disebut centralized server.

Komponen secara hardware :

Karena hardware di sisi server berfungsi sebagai pusat proses database dan tempat penyimpanan data, maka hal yang harus dipertimbangkan secara baik saat pengadaan ialah kecepatan, keamanan, kemudahan untuk ekspansi dan adanya dukungan software khususnya database. Komponen secara Software :

a. Sistem Operasi  
Sistem operasi di sisi server dapat terdiri dari Windows NT, Windows XP, Novell Netware maupun Unix.

b. Database

Perangkat lunak pengolah database yang ada cukup banyak, seperti : Oracle, SQL Server, MySQL dan lain-lain.

### Komponen Jaringan

Dengan memahami dasar-dasar network, diharapkan dapat memberikan fondasi dasar yang lebih kokoh untuk mendesain infrastruktur client / server karena client / server menggunakan network untuk berkomunikasi diantara client dan server.

- a. Model Network
- b. Skema pengiriman data
  - c. Hardware
  - d. Software
  - e. Komunikasi antar network

## **PRAKTIKUM KE – I**

1. Judul Materi : Pengantar MySQL
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat menjelaskan mengenai MySQL, Cara Mendapatkan Software-nya dan perbedaannya dengan DBMS yang lain.
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan : Teori Basis Data  
Prasyarat
5. Teori Dasar :

### **PENGANTAR MySQL**

#### **Apa itu MySQL**

MySQL adalah sebuah system manajemen database relasi (RDBMS) yang bersifat terbuka (open source), yaitu MySQL dapat di download oleh siapa saja, baik versi kode program aslinya (source code program) maupun versi binernya (executable program) dan bisa digunakan secara (relatif) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer.

Bila kita seorang programer dan ikut berpartisipasi dalam memodifikasi program asli MySQL untuk keperluan sendiri atau perusahaan sendiri, maka kitadiwajibkan secara moral untuk memberikan hasil modifikasi tersebut kepada MySQL dan komunitas open source. Hal ini merupakan konsekuensi dari konsep open source. Tetapi bila kita ingin merahasiakan hasil modifikasi tersebut, maka kita wajib membayar sejumlah dana kepada MySQL AB sebagai biaya lisensi.

MySQL menggunakan bahasa standar SQL (Structure Query Language) sebagai bahasa interaktif dalam mengelola data (sering disebut : query).

Sejarah MySQL yang merupakan hasil buah pikiran dari Michael “Monty Widenius, David Axmark, dan Allan Larson dimulai tahun 1995. Pada bulan juni 2000, MySQL AB mengumumkan bahwa mulai MySQL versi 3.23.19 diterapkan sebagai General Public License (GPL). Beberapa penghargaan pun diperoleh oleh MySQL, terutama dari majalah-majalah dengan topik Linux. Hal ini disebabkan pada awalnya MySQL hanya berjalan di atas platform Sistem Operasi Linux. Bahkan hampir semua distro Linux yang beredar saat ini telah memasukkan program MySQL sebagai database standarnya. Tetapi, saat ini MySQL dapat pula berjalan di atas system operasi lainnya seperti Microsoft

Windows, Solaris, FreeBSD, IBM’s AIX, Mac OS X dan lain-lain.

### **Benarkah MySQL Gratis ?**

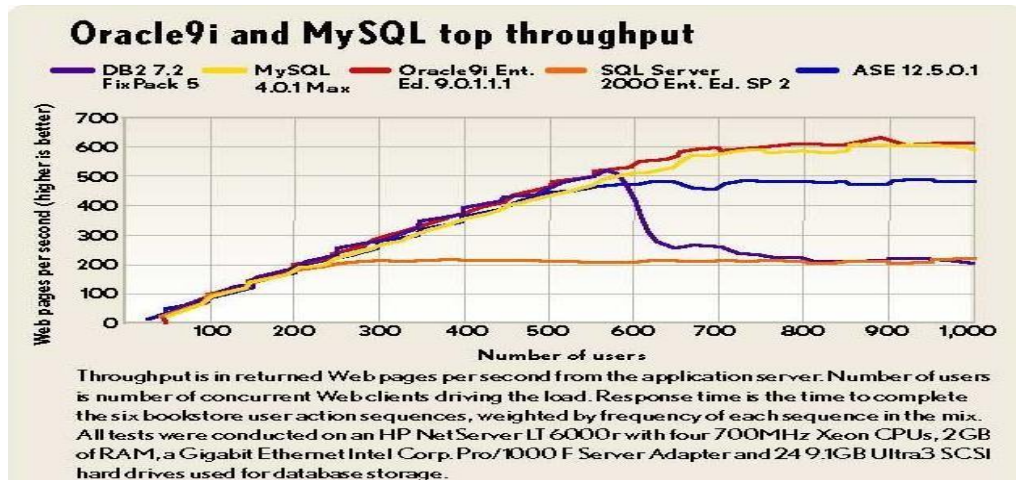
Ada dua jenis lisensi :

- a. Lisensi gratis (free software / open source GNU general public license (GPL)). Seperti yang telah dijelaskan di atas, dan MySQL tidak menjamin adanya layanan dukungan dari MySQL AB.
- b. Lisensi komersial (Non GPL commercial license). Kita harus membayar sejumlah dana kepada MySQL AB sesuai dengan jenis layanan tertentu yang disediakan oleh MySQL AB, atau bila :
  - ✓ Mendistribusikan MySQL dan aplikasinya dengan menggunakan program-program bukan open source.
  - ✓ Menghendaki adanya jaminan dari MySQL AB tentang kinerja MySQL.
  - ✓ Menghendaki dukungan teknis dan non-teknis dari MySQL AB.

### **Mengapa Memilih MySQL ?**

Ada beberapa pertimbangan mengapa memilih MySQL :

- a. Kecepatan. MySQL mempunyai kecepatan paling baik dibanding database server lainnya (lihat <http://www.mysql.com/it-resource/benchmarks/eweek.html>)



- b. Mudah Digunakan.
- c. Open Source. Siapapun dapat berpartisipasi untuk mengembangkan MySQL dan hasil pengembangan itu diserahkan kepada umum atau komunitas Open Source.
- d. Kapabilitas. MySQL telah digunakan untuk mengelola database dengan jumlah 50 juta record. Bahkan sanggup untuk mengelola 60.000 tabel dengan jumlah record 5 juta. Mendukung penggunaan index hingga 32 buah index pertabelnya.
- e. Biaya Rendah.
- f. Konektifitas dan Keamanan. MySQL mendukung dan menerapkan system keamanan dan ijin akses tingkat lanjut (advanced permissions and security system), termasuk dukungan pengamanan data dengan cara pengacakan lapisan data.
- g. Fleksibilitas/Portabilitas. MySQL mendukung perintah-perintah ANSI SQL 99 dan beberapa perintah database alternatif lainnya sehingga memudahkan untuk beralih dari dan ke MySQL.
- h. Lintas Platform system operasi. MySQL dpat dijalankan pada beberapa system operasi yang berbeda, seperti Linux, Microsoft Windows, Sun Solaris, Novell Netware, IBM's AIX, Mac OS X dll.

### Dimana mendapatkan MySQL ?

MySQL dapat di download langsung dari situs MySQL AB

([www.mysql.com/downloads/](http://www.mysql.com/downloads/)). Sebaiknya kita memilih rilis terbaru yang stabil (biasanya diberi keterangan “recommended”). Rilis terakhir yang disarankan per bulan Agustus 2003 adalah rilis 4.0.14. Secara umum, MySQL terdiri dari dua versi yaitu standar dan max. Versi standar disarankan bagi para pengguna secara umum, sedangkan versi max memiliki beberapa fitur tambahan dibanding versi standar.

### **Program dan Data di MySQL.**

Posisi penempatan data dan program di MySQL tergantung dari jenis system operasi yang digunakan.

- a. Pada system operasi Windows, direktori data biasanya berada di C:\MYSQL\DATA\ dan direktori program berada di C:\MYSQL\BIN.
- b. Pada system operasi Linux Mandrake, direktori data biasanya berada di /var/lib/mysql/ dan direktori program berada di /usr/sbin/.
- c. Pada system operasi Linux Redhat, direktori data biasanya berada di /var/lib/mysql dan direktori program berada di /var/lib/.

Informasi mengenai posisi direktori tersebut dapat dilihat dengan membuka file my.cnf, yaitu sebuah file yang berisi konfigurasi dasar dari MySQL, seperti datadir, socket, user, basedir dsb. File my.cnf biasanya berada di direktori /etc/ atau /var/lib/mysql/ pada Linux. Sedangkan pada Windows ada di direktori C:\Windows\System32 dengan nama file my.ini. Keunikan MySQL adalah sebuah database akan memiliki satu direktori data yang berdiri sendiri, tidak bercampur dengan database-database lainnya yang ada didalam server tersebut. Nama sebuah database akan menjadi nama dari direktori data tersebut.

### **Bahasa Basis Data**

Merupakan bahasa yang secara khusus digunakan untuk mengelola suatu basis data. Dalam implementasi penyusunan basis data pada umumnya digunakan bahasa basis data ini sebagai bahasa standar yang dikenal dengan sebutan SQL (*Structured Query Language*). SQL paling tidak mempunyai dua macam perintah yang digunakan untuk mengelola dan mengorganisasikan basis data, yaitu :



- a. Bahasa Definisi Data (DDL / Data Definition Language)
- b. Bahasa Manipulasi Data (DML / Data Manipulation Language)

Selain kedua macam perintah di atas, pada kebanyakan DBMS (Database Management System) terdapat perintah yang tergolong DCL (Data Control Language) yang berkaitan dengan pengaturan sekuritas basis data.

#### Bahasa Definisi Data (DDL)

Adalah perintah-perintah yang biasa digunakan oleh administrator basis data (DBA) untuk mendefinisikan skema ke dalam DBMS. Skema adalah deskripsi lengkap tentang struktur table, rekaman dan hubungan data pada basis data.

DDL juga digunakan untuk mendefinisikan subskema. Subskema adalah pandangan (view) bagi pengguna basis data yang merupakan himpunan bagian dari skema. Bila suatu item tidak tercantum dalam skema seseorang pengguna, maka item tersebut tidak tersedia bagi pengguna bersangkutan. Subskema dapat menjadi mekanisme pengamanan system basis data, yakni dengan mengatur hak pengaksesan item-item dalam basis data.

DDL juga digunakan untuk menciptakan, mengubah dan menghapus basis data.

#### Bahasa Manipulasi Data (DML)

Adalah perintah-perintah yang digunakan untuk mengubah, memanipulasi dan mengambil data pada basis data. Tindakan seperti menghapus, mengubah dan mengambil data menjadi bagian dari DML. DML pada dasarnya dibagi menjadi dua :

- ✓ Prosedural, yang menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkannya.
- ✓ Nonprosedural, yang menuntut pengguna menentukan data apa saja yang diperlukan, tetapi tidak perlu menyebutkan cara mendapatkannya.

6. Tugas Praktikum Ke-I :

Lakukan kunjungan ke situsnya MySQL ([www.mysql.com](http://www.mysql.com)), carilah informasi sebanyak-banyaknya mengenai MySQL. Setelah itu buatlah laporan mengenai hal-hal berikut :

- a. The difference between MySQL and MySQL AB
- b. How MySQL AB Operates
- c. MySQL Core Values
- d. MySQL Dual Licensing
- e. Organization and structure of the MySQL Reference Manual
- f. The MySQL Mailing List Tugas dikumpulkan paling lambat pada Pertemuan ke II mata pelajaran ini.

## PRAKTIKUM KE – II

1. Judul Materi : Memulai Menggunakan MySQL
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat memulai menggunakan MySQL, menjelaskan tipe data di MySQL dan menggunakan perintah-perintah dasar MySQL.
3. Jumlah Pertemuan : 1 Kali
4. Pengetahuan : Teori Basis Data
- Prasyarat
5. Teori Dasar :

### MEMULAI MENGGUNAKAN MySQL

#### Memanggil MySQL

MySQL bekerja dibawah sistem operasi Linux, sehingga untuk dapat menggunakan MySQL kita harus masuk dahulu ke jaringan Linux. Urutan langkah untuk dapat menggunakan MySQL adalah sebagai berikut :

- Klik START kemudian Klik pilihan RUN, pada isian Open, tuliskan perintah :

Putty 192.168.10.253 (*IP address server lab. Komputer*)

Atau

Telnet 192.168.10.253

- Setelah muncul pertanyaan untuk memasukkan nama login

Login : .....

Isikan nama login anda.

- Tampilan yang muncul selanjutnya adalah prompt sebagai berikut :

[namalogin@gw namalogin]\_

tampilan prompt diatas sebagai tanda bahwa kita sudah masuk kedalam system operasi Linux.

- Dan untuk masuk ke MySQL ketikkan : mysql setelah prompt diatas :

[namalogin@gw namalogin] mysql

Setelah anda enter, maka prompt yang muncul selanjutnya adalah :

[mysql] \_

Prompt sebagai tanda bahwa kita sudah berada di MySQL.

### **Tipe Data dan Macam-Macam Operator di MySQL**

Sebelum menuliskan operasi-operasi di MySQL, maka ada beberapa hal yang harus diperhatikan, yaitu tentang tipe data, operator logika, operator relasi dan operator aritmatika yang berlaku di MySQL.

#### Tipe DataMySQL

Setiap table di dalam database dibuat, maka table tersebut dapat terdiri dari satu atau banyak field. Masing-masing field tersebut harus didefinisikan tipe datanya. Selain menentukan tipe data, kita juga dapat menentukan berapa lebar data, field tersebut dapat berisi null atau tidak, field tersebut merupakan kunci atau bukan dan lain-lain.

*Tipe Data Numerik adalah :*

Tipe Data	Keterangan	Ukuran
Integer atau Int[(m)]	Signed value : -2147683648 to 2147683647 Unsigned values : 0 to 4294967295	4 byte
Decimal atau Dec(M,D)	Bilangan desimal dengan nilai tergantung pada besaran M dan D	M byte
Tinyint[(m)]	Signed value : -128 to 127 Unsigned value : 0 to 255	1 byte
Smallint[(m)]	Signed value : -32768 to 32767 Unsigned value : 0 to 65535	2 byte

Mediumint[(m)]	Signed value : -8388608 to 838860 Unsigned value : 0 to 16777215	3 byte
Bigint[(m)]	Signed value : -922337203685477808 to 9223372036854775807 Unsigned value : 0 to 18446744073709551615	8 byte
Float(m,d)	Bilangan pecahan presisi tunggal	4 byte
Double[(m,d)]	Bilangan pecahan presisi ganda	8 byte

*Tipe Data String adalah :*

Tipe Data	Keterangan
Char atau Character	A Fixed-length character string : Menyatakan deretan karakter (string) yang lebarnya tetap yaitu maksimum adalah 255 karakter
Varchar	A variable-length character string : Data string dengan lebar data yang bervariasi (M), Maksimum lebar adalah 255 karakter
Tinyblob	A very small BLOB (binary large object)
BLOB	A small BLOB
Mediumblob	A medium-sized BLOB
Longblob	A large BLOB
Tinytext	A very small text string
Text	A small text string
Mediumtext	A medium-sized text string
Longtext	A large text string
Enum	An enumeration

*Tipe Data String adalah :*

Tipe Data	Keterangan	Ukuran
Date	Digunakan untuk tanggal dengan format “YYYY-MMDD” Range nilai : “1000-01-01” s.d “9999-12-31”	3 byte
Time	Digunakan untuk waktu dengan format “hh:mm:ss” Range nilai : -838:59:59” s.d “838:59:59”	3 byte
Datetime	Digunakan untuk tanggal dan waktu dengan format “YYYY-MM-DD hh:mm:ss” Range nilai : “1000-01-01 00:00:00” s.d “9999-12-31 23:59:59”	8 byte
Timestamp	Digunakan untuk penulisan tanggal dan waktu dengan format “YYYYMMDDhhmmss”	4 byte

Year	Digunakan untuk penulisan tahun dengan format "YYYY" Range nilai : 1901 s.d 2155	1 byte
------	---	--------

### Operator Aritmatika

Operator aritmatika pada MySql berdasarkan fungsinya dapat dikategorikan menjadi :

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Mencari sisa hasil bagi (modulus)

### Operator Relasional (Pembandingan)

Ada beberapa macam operator relasional pada MySql, yaitu :

Operator	Keterangan
=	Sama dengan
>	Lebih besar dari
<	Lebih Kecil dari
>=	Lebih besar dari atau sama dengan
<=	Lebih kecil dari atau sama dengan
<>	Tidak sama dengan
!=	Tidak sama dengan
Is Null	Apakah sebuah nilai adalah null
In	Apakah sebuah nilai berada didalam pilihan yang ada
Not In	Apakah sebuah nilai tidak berada didalam pilihan yang ada
Regexp	Singkatan dari Reguler Expression. Nilai menjadi benar (true) bila nilai a sama dengan nilai b

### Operator Logika

Ada beberapa macam operator logika pada MySql, yaitu :

Operator	Keterangan
Not atau !	Sebagai negasi atau pembalik nilai
Or atau	Atau
And atau &&	Dan

## MySQL Naming Rules

Ada beberapa aturan pemberian nama di MySQL :

1. Karakter yang diperbolehkan untuk penamaan. Nama terdiri dari kumpulan alphanumeric (angka dan huruf) ditambah dengan karakter ‘\_’ (underscore) dan ‘\$’.
2. Panjang Nama. Nama dari database, table, field dan indexes panjang maksimal adalah 64 karakter.
3. Nama database dan table sifatnya not case-sensitive (tidak membedakan huruf kecil dan besar).

Contoh :

`select * from Mhs;` sama dengan `select * from mhs;`

4. Statements dan option MySQL sifatnya adalah not case-sensitive (tidak membedakan huruf kecil dan besar) Contoh :

`SELECT NOW()`

`Select Now()` `select now()` adalah tiga perintah yang artinya sama.

5. Nama field dan Indexes sifatnya not case-sensitive (tidak membedakan huruf kecil dan besar)
6. Nama yang digunakan untuk alias sifatnya case-sensitive

## Perintah Dasar MySQL :

Beberapa perintah-perintah dasar MySQL yang akan dijelaskan adalah sebagai berikut :

### Memanggil MySQL

Untuk menjalankan MySQL pertama kali cukup dengan mengetikkan mysql pada system prompt :

[system prompt] `mysql database_name -h host_name -u user_name -p password` Keterangan :

Database\_name adalah nama database yang akan dituju

-h host\_name adalah nama host server yang akan dituju

-u user\_name adalah nama user yang akan dituju. Jika kita menggunakan UNIX, maka nama user mysql adalah sama dengan nama login, dan kita dapat menganggap ini sebagai option yang nantinya mysql akan menggunakan nama login sebagai nama user mysql.

### Menampilkan Bantuan

MySQL memiliki fasilitas bantuan sederhana (HELP) yang dapat diaktifkan dengan cara mengetikkan **mysql --help** atau **mysql --h** pada siste prompt.

Contoh :

```
[namalagin@gw namalagin] mysql --help
```

### Mengetahui Nama User yang sedang Aktif

Untuk mengetahui nama user yang sedang aktif pada saat itu, kita dapat menggunakan perintah `user()`.

### Menampilkan Tanggal dan Waktu Sekarang

Untuk menampilkan data tanggal dan waktu sekarang, kita dapat menggunakan perintah `now()` atau `current_date` atau `current_timestamp` (penjelasan detil di bab berikutnya).

### Menampilkan Informasi Status dari Tabel

Untuk menampilkan status dari masing-masing table yang telah kita buat, gunakan perintah :

```
Mysql> Show table Status;
```

### Mengetahui Versi MySQL



Kita dapat mengetahui versi MySQL yang digunakan perintah berikut untuk mengetahui versi MySQL yang digunakan :

Contoh:

```
mysql> select user(), now(), version();
```

user()	now()	version()
arifin@localhost	2005-08-08 10:26:07	3.23.54

1 row in set (0.00 sec)

; -> mengakhiri perintah

### Membatalkan Perintah

Untuk membatalkan perintah yang telah terlanjur dituliskan (tapi belum ditekan tombol ENTER), tambahkan saja “\c” (singkatan dari Cancel) diakhir perintah tersebut dan diikuti menekan tombol ENTER.

Contoh :

```
Mysql> Select Now( ),
      → version( ),
      → \c
```

### Keluar dari MySQL

Untuk mengakhiri penggunaan MySQL, kita dapat menggunakan perintah EXIT atau Quit.

Contoh :

```
Mysql> exit;
```

### Memberi Catatan atau Komentar

Memberikan catatan atau komentar pada MySQL dilakukan dengan memberikan awalan /\* dan akhiran \*/ atau awalan # pada baris komentar. Perbedaannya : /\* .... \*/ dapat digunakan untuk lebih dari satu baris komentar, sedangkan # hanya untuk satu baris komentar saja.

Contoh :

```
Mysql> /* ini hanya komentar
      → Jadi tidak akan diproses oleh MySQL */
      → Show Databases;
```

Membuat Basis Data

Untuk membuat sebuah database baru, maka perintah adalah :

Create Database <namabasisdata>;

Contoh :

Akan dibuat sebuah database baru dengan nama latihan, maka :

```
mysql> create database latihan;
Query OK, 1 row affected (0.02 sec)
```

Mengetahui NamaBasisdata Yang Sedang Aktif

Untuk mengetahui nama basis data yang sedang kita masuki (gunakan), tuliskan perintah database( ).

Contoh :

```
mysql> select database();
+-----+
| database() |
+-----+
| latihan    |
+-----+
1 row in set (0.00 sec)
```

Menampilkan nama-nama Basisdata

Database yang telah ada di dalam system dapat ditampilkan menggunakan perintah :

```
mysql> show databases;
+-----+
| Database |
+-----+
| latihan  |
| mysql   |
| test    |
+-----+
3 rows in set (0.00 sec)
```

Database mysql dan test adalah dua buah database yang sudah ada secara default pada saat pertama kali kita menggunakan MySQL. Database mysql berisi data-data tentang user, jenis hak akses user (*user privileges*), password, dan sebagainya. Sedangkan database test hanya sebuah database yang dapat kita lakukan untuk uji coba atau latihan.

Memilih Database yang akan Digunakan

Untuk menggunakan atau memilih suatu database digunakan perintah :

Use <namabasisdata>;

Contoh :

Mysql> use latihan;

Melihat nama-nama table dalam suatu Basisdata

Untuk melihat table yang ada di dalam sebuah database digunakan perintah :

```
mysql> show tables;
+-----+
| Tables_in_latihan |
+-----+
| mhs                |
+-----+
1 row in set (0.00 sec)
```

Membuat Tabel Baru

Untuk membuat table baru perintahnya adalah :

Create Table <namatable> (nama\_field1 spesifikasi\_field1, ..... );

Contoh :

Akan dibuat table mhs yang berisi data nim, nama dan tanggal lahir :

```
mysql> create table mhs (nim char(4) not null primary key,
-> nama char(20), tgl_lhr date);
Query OK, 0 rows affected (0.03 sec)
```

Not null dan primary key adalah option yang digunakan oleh field nim. Option not null berarti field nim tidak boleh diisi dengan null, sedangkan primary key menunjukkan bahwa field nim digunakan sebagai kunci utama. Field yang ditunjuk sebagai kunci utama (primary key) ini nantinya harus berisi data yang unik. Apabila kita mencoba mengisi data yang sama seperti data yang sudah tersimpan, maka secara otomatis MySQL akan melakukan validasi terhadap data tersebut sehingga data akan ditolak.

Menampilkan Struktur Suatu Table

Struktur dari sebuah table dapat ditampilkan dengan menggunakan perintah :

Describe <namatable>;

Contoh :

```
mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim    | char(4) |      | PRI |          |       |
| nama   | char(20) | YES  |     | NULL    |       |
| tgl_lhr | date    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

mysql> describe mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim    | char(4) |      | PRI |          |       |
| nama   | char(20) | YES  |     | NULL    |       |
| tgl_lhr | date    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Penulisan perintah describe dapat disingkat dengan menuliskan 4 karakter pertama, sehingga perintah yang dituliskan menjadi desc.

Mengisikan Data

Tabel yang telah terbentuk dapat diisi data dengan perintah *insert*

Insert into <namatable> Values (data\_field1,data\_field2,data\_field3, dst.....);

Contoh :

```
mysql> insert into mhs values
-> ('1111','Wahyu Ayu Tenan','1997-03-11'),
-> ('2222','Joko Mudo','1975-10-09'),
-> ('3333','Hari Mulyono','1976-11-12');
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

Mengisikan Data Secara Masal

Ada tiga cara yang dapat dilakukan untuk memasukkan data secara masal.

1. Menggunakan Script SQL dalam file Teks

Dengan cara ini kita tuliskan dulu perintah MySQL dalam sebuah file teks biasa yang disimpan dengan nama file berekstensi .sql. Penulisan bentuk file teks ini dapat dilakukan dengan menggunakan teks editor Notepad, Textpad maupun yang lainnya. Untuk pemasukan datanya, kita harus keluar dahulu dari MySQL kemudian dituliskan perintah dengan format sebagai berikut :

```
[system prompt] mysql -h localhost -u root -p database < namafile.sql
```

Misalnya, kita membuat file teks yang berisi data tersebut dengan nama file isidata.sql pada direktori kerja yang isinya :

```
Insert into mhs values
('1111','Wahyu Ayu Tenan','1997-03-11'),
('2222','Joko Mudo','1975-10-09'),
('3333','Hari Mulyono','1976-11-12')
```

Kemudian tuliskan perintah berikut :

```
[system prompt] mysql -h localhost -u root -p latihan < "/usr/data/isidata.sql" Apabila
file teks tersebut tersimpan didirektori /usr/data pada linux.
```

Atau

```
[system prompt] mysql -h localhost -u root -p latihan < "c:\data\isidata.sql" Apabila
file teks tersebut tersimpan didirektori c:\data pada Windows.
```

## 2. Menggunakan Perintah Load Data

Cara ini menggunakan prinsip yang hampir sama dengan cara pertama. Perintah MySQL untuk pemasukan data disimpan ke dalam sebuah file teks biasa dengan ekstensi txt. Perbedaannya, setiap kolom data harus diberikan pemisah berupa tabs (secara default tabs delimited) dan satu baris perintah hanya untuk satu record saja.

Kemudian proses pemasukan datanya dilakukan dengan perintah LOAD DATA dari MySQL prompt sebagai berikut :

```
Load Data [local] infile 'lokasi dan nama file teks' into table nama_table;
```

Penggunaan option local berarti file teks tersebut berada di dalam komputer kita sendiri dan bila tanpa option local, berarti file teks tersebut berada di dalam server MySQL. Apabila komputer yang kita gunakan merangkap sebagai server, maka perintah yang digunakan tanpa menggunakan local.

Sebagai contoh data yang akan dimasukan disimpan dalam file yang diberi nama isidata1.sql dan isinya adalah :

1111	Wahyu Ayu Tenan	1997-03-11
2222	Joko Mudo	1975-10-09
3333	Hari Mulyono	1976-11-12

Kemudian dilanjutkan dengan memberikan perintah :

mysql> Load Data Local Infile 'isidata1.sql' into table mhs;

### 3. Menggunakan Program Bantu MYSQLIMPORT

Perintah ini dilakukan dari system prompt. Nama file teks sebaiknya sama dengan nama table yang akan diisikan (hanya ditambahkan dengan ekstensi .txt).

Bentuk umum perintah MYSQLIMPORT adalah :

[system prompt] mysqlimport [pilihan] database file\_teks.txt

File teks yang dibuat diberi nama mhs.txt dan berisi :

1111	Wahyu Ayu Tenan	1997-03-11
2222	Joko Mudo	1975-10-09
3333	Hari Mulyono	1976-11-12

Dilanjutkan dengan menuliskan perintah MYSQLIMPORT :

[system prompt] mysqlimport --local latihan "/usr/data/mhs.txt"

*Apabila file teks tersebut tersimpan didirektori /usr/data pada linux.*

Atau

[system prompt] mysqlimport –local latihan “c:\data\mhs.txt”

*Apabila file teks tersebut tersimpan didirektori c:\data pada Windows.*

### Menambahkan sebuah field ke dalam Tabel

Apabila ingin menambahkan field baru ke dalam sebuah table, maka perintahnya :

Alter Table <namatable> Add <namafield> tipe\_data (ukuran);

Contoh :

Menambahkan field Sex dengan tipe karakter yang lebarnya 1 :

Mysql> Alter Table Mhs Add sex char(1);

Perintah penambahan field baru tersebut dapat disertakan dengan option FIRST ataupun AFTER. Option *first* digunakan untuk menempatkan field yang baru pada posisi field pertama. Sedangkan option *after* digunakan apabila kita menginginkan field yang baru ditempatkan pada posisi tertentu.

Contoh :

Akan ditambahkan field baru dengan nama *noid int(4)* pada posisi pertama :

Mysql> alter table mhs add noid int(4) first;

Contoh :

Akan ditambahkan field baru dengan nama *alamat char(30)* yang ditempatkan setelah field nama :

Mysql> alter table mhs add alamat char(30) after nama;

### Menghapus Field

Untuk menghapus suatu field, maka perintah yang digunakan :

Alter Table <namatable> Drop <namafield>;

Contoh :

Akan menghapus field yang bernama sex pada table mhs :

Mysql> Alter Table Mhs Drop Sex;

### Menghapus Database

Untuk menghapus suatu database, maka perintah yang digunakan :

```
Drop Database [if exists] <namadatabase>;
```

Contoh :

Akan menghapus database yang namanya latihan:

```
Mysql> Drop database if exists latihan;
```

Keterangan :

Apabila option IF EXISTS diberikan, maka perintah tersebut tidak akan memunculkan pesan kesalahan walaupun nama databasenya tidak ada.

### Menghapus Table

Untuk menghapus suatu table, maka perintah yang digunakan :

```
Drop table [if exists] <namatable>;
```

Contoh :

Akan menghapus table yang namanya mhs :

```
Mysql> Drop table if exists mhs;
```

### Mengganti Nama Tabel

Apabila kita ingin mengganti nama suatu field, maka dapat digunakan perintah rename dengan format :

```
Alter Table <namatable> Rename <namafield_baru>;
```

Contoh :

Akan mengganti nama tabel emp menjadi table pegawai :

```
Mysql> Alter Table mhs Rename pegawai;
```

### Mengganti nama field serta ukuran



Sedangkan untuk mengganti nama serta ukuran suatu field, gunakanlah perintah change dengan format seperti berikut :

```
Alter Table <Namatable> Change <namafield_lama> <namafield_baru> tipe_data_baru;
```

Contoh :

Mengganti nama field NIM menjadi NOINDUK yang panjangnya 14 menjadi 15 :

```
Mysql> Alter Table Mhs Change nim noinduk char(15);
```

### Mengganti Tipe Data(ukuran) Field

Untuk mengganti tipe data (ukuran) field saja, kita dapat menggunakan perintah modify dengan format perintahnya adalah :

```
Alter Table <Namatable> Modify <namafield> <tipe_data_baru>;
```

Contoh :

Akan diganti tipe data dari field sal, yang semula int menjadi decimal(8,2) :

```
Mysql> Alter Table emp modify sal decimal(8,2);
```

## PRAKTIKUM KE – III

1. Judul Materi : Kekangan Nilai Pada Data
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat Membuat Tabel dengan MySql beserta batasanbatasannya. 3. Jumlah Pertemuan : 1 Kali
4. Teori Dasar :

### KEKANGAN NILAI PADA DATA

Pada bab ini yang akan dipelajari adalah bagaimana caranya agar data tidak bisa berisi null (tidak kosong), mencegah data yang kembar, mencegah nilai yang tidak valid, mengatur nilai bawaan.

#### Mengatur agar data selalu diisi :

Untuk mengatur agar data tertentu harus diisi (tidak kosong), dapat dilakukan dengan menambahkan kata kunci NOT NULL pada pendefinisian field yang bersangkutan.

Contoh :

```
Mysql> Create table mhs (nim char(15) not null, nama char(25) not null,  
→ alamat char(30));
```

Keterangan :

Field nim dan alamat tidak boleh kosong.

### Menentukan Kunci Primer

Untuk menyebutkan kunci primer terhadap suatu field maka sertakan PRIMARY KEY pada pendefinisian kolom.

Contoh :

```
Mysql> Create table mhs (nim char(15) not null primary key,  
□ nama char(25) not null, alamat char(30));
```

Keterangan :

Field nim dan alamat tidak boleh kosong dan field nim dipilih sebagai primary key.

### Membuat Kunci Komposit

Untuk membuat kunci primer yang terdiri dari lebih satu field, maka field-field tersebut ditulis setelah PRIMARY KEY sebagai berikut : PRIMARY KEY (field1, field2, ...) yang ditulis pada akhir deklarasi struktur tabel.

Contoh :

```
Mysql> Create table mhs (nim char(15) not null,  
□ nama char(25) not null, alamat char(30), primary key (nim, nama));
```

Keterangan :

Field nim dan alamat tidak boleh kosong dan sebagai primary key adalah field nim dan nama.

### Membuat Nilai Yang Unik

Sebenarnya kunci primer sudah bertindak sebagai penjamin bahwa nilai suatu field adalah unik (tidak kembar). Tetapi operasi ini digunakan untuk membuat suatu field yang bukan primer menjadi unik, yaitu dengan menambahkan UNIQUE pada deklarasi field yang bersangkutan.

Contoh :

```
Mysql> Create table mhs (nim char(15) not null primary key,  
□ nama char(25) not null unique, alamat char(30));
```

Keterangan :

Field nama tidak boleh kosong dan nilai juga harus unique.

Memberi Nilai Bawaan

Apabila kita tidak memberikan suatu nilai pada field tertentu maka field tersebut akan terisi dengan null. Kita dapat mengatur agar kalau suatu field tidak diberi nilai eksplisit maka nilai bawaan tertentu akan diisikan ke field tersebut. Yaitu dengan menambahkan kata kunci default pada field bersangkutan sewaktu tabel didefinisikan.

Contoh :

```
Mysql> Create table mhs (nim char(15) not null primary key,  
□ nama char(25) not null unique, alamat char(30), sex char(1) default 'W');
```

Keterangan :

Pada saat sex tidak diisikan dengan suatu nilai, maka nilai bawaan 'W' yang akan disimpan.

Memvalidasi Nilai

Untuk mengatur nilai tertentu saja yang dapat diberikan ke suatu field, maka dapat dilakukan dengan memberikan kata kunci CHECK diikuti dengan suatu kondisi pada field tertentu.

Contoh:

```
Mysql > Create table mhs (nim char(15) not null primary key,  
□ nama char(25) not null unique, alamat char(30), → sex char(1) default 'W' check (sex = 'W'  
or sex = 'P'));
```

Keterangan :

Pada saat sex tidak diisikan dengan suatu nilai, maka nilai bawaan 'W' yang akan disimpan.

Nilai yang dapat diterima oleh field sex adalah W atau P.

Option lain yang dapat digunakan untuk melakukan validasi data seperti di atas adalah

ENUM. Bentuk umum perintahnya : Enum (nilai1, nilai2, ..... nilaiN).

Sehingga apabila kita akan memvalidasi isian data pada field sex, agar dapat diisikan huruf 'W' atau 'P' maka perintah menjadi :

```
Mysql> Create table mhs (nim char(15) not null primary key,
```

□ nama char(25) not null unique, alamat char(30), sex enum('W','P','w','p');

Catatan : option enum hanya dapat diisi dengan data-data yang bertipe string dan hanya dapat menampung 1 data saja dari pilihan data yang ada atau null, dengan jumlah maksimum 65535 pilihan data.

### Option khusus pada field numeric

Option adalah suatu perintah tambahan yang bersifat optional dan hanya diterapkan pada field numeric saja. Option-option tersebut antara lain :

#### ✓ Auto\_increment

Digunakan untuk menaikkan nilai sebuah field numeric secara otomatis dengan pertambahan nilai 1. Nilai awal yang digunakan pada field ini adalah 1. Ada beberapa hal yang harus diperhatikan untuk dapat menggunakan option auto\_increment yaitu :

- a. hanya dapat digunakan pada jenis field numeric
- b. Dalam satu table hanya boleh ada satu field yang menggunakan option ini
- c. Field tersebut harus bersifat unik atau berstatus primary key
- d. Field ini tidak boleh bersifat null

Contoh :

```

Mysql> create table buku
→ (nourur int unsigned auto_increment primary key, judul varchar(50));
  
```

#### ✓ Signed dan Unsigned

Signed digunakan apabila bilangan bulat tersebut dapat memiliki nilai negatif karena menggunakan tanda minus (-) maupun positif (+). Sedangkan unsigned digunakan apabila bilangan tersebut tidak memiliki nilai negatif, hanya positif saja (tanda (+) tidak perlu dicantumkan).

#### ✓ Zerofill

Digunakan untuk menempatkan angka nol (0) didepan bilangan pada suatu kolom.

### 5. Latihan Praktikum III :

1. Buatlah sebuah tabel dengan nama mhs dengan struktur tabel sebagai berikut :

Nama Field	Type	Keterangan
Nim	Char(14)	Primary key, unik dan tidak boleh kosong
Nama	Char(25)	Tidak boleh kosong dan unik
Kelamin	Char(1)	Hanya dapat diisikan P ( pria) atau W (wanita)
Agama	Char(1)	Hanya dapat diisikan angka 1 s.d 5 (1=islam, 2=katholik, 3=kristen, 4=hindu, 5=budha) dan apabila tidak diisikan maka nilai yang tersimpan adalah 1.
Tgl_lhr	Date	

2. Tambahkan field alamat varchar(30) yang diletakkan setelah field nama
3. Tambahkan field kota char(20) yang diletakkan setelah field alamat
4. Gantilah nama field kelamin menjadi sex (tipe datanya masih tetap)

5. Masukkan record-record berikut :

'A21.2001.00234','Dedy Sutanta','Jl. A. Yani 20 ','Solo','P','1','1979-10-20');

‘A21.2001.00214’, ‘Budi Satria’, ‘Jl. Dago 10’, ‘Yogyakarta’, ‘P’, ‘1’, ‘1978-11-22’);  
‘A21.2001.00224’, ‘Sulis’, ‘Jl. Nakula I 25’, ‘Semarang’, ‘W’, ‘2’, ‘1975-08-16’);  
‘A21.2001.00344’, ‘Suci Andriani’, ‘Jl. Seroja Raya’, ‘Semarang’, ‘W’, ‘3’, ‘1977-09-21’);

## **PRAKTIKUM KE – IV**

1. Judul Materi : Perintah Manipulasi Data
  2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat menggunakan perintah-perintah MySql untuk memanipulasi Data 3.
- Jumlah Pertemuan : 1 Kali
4. Teori Dasar :

### **PERINTAH MANIPULASI DATA**

#### *Menampilkan Informasi (Retrieving Information)*

Tabel yang telah dibuat dan diisi dengan data dapat ditampilkan isinya (informasi) dengan menggunakan perintah SELECT. Bentuk umum perintah select adalah :

SELECT *what to select*

FROM *tables or tables*

WHERE *conditions that data must satisfy*

Perintah select diikuti dengan nama field (kolom) yang akan ditampilkan. Jika field yang akan ditampilkan jumlahnya banyak, maka antara field yang satu dengan yang lain dipisahkan dengan koma (.). Apabila kita ingin menampilkan seluruh field dari suatu table, maka kita dapat menggunakan tanda \* (mewakili seluruh field). Perintah from diikuti dengan nama table yang akan digunakan. Apabila table yang akan digunakan ternyata lebih dari satu, maka nama-nama table tersebut dituliskan dengan pemisah tanda koma(.). Perintah where diikuti dengan (kondisi) syarat tertentu yang diinginkan. Kondisi yang dituliskan ini akan digunakan sebagai dasar seleksi data per record (seleksi).

Contoh 1 :

Akan menampilkan seluruh field dan record dari tabel mhs, maka perintahnya adalah :

```
mysql> select * from mhs;
```

nim	nama	alamat	kota	sex	agama	tgl_lhr
A21.2001.00234	Dedy Sutanta	Jl. A. Yani 20	Solo	P	1	1979-10-20
A21.2001.00214	Budi Satria	Jl. Dago 10	Yogyakarta	P	1	1978-11-22
A21.2001.00224	Sulis	Jl. Nakula I 25	Senarang	W	2	1975-08-16
A21.2001.00344	Suci Andriani	Jl. Seroja Raya	Senarang	W	3	1977-09-21

```
4 rows in set (0.00 sec)
```

Contoh 2 :

Menampilkan field nim dan nama peserta didik yang sex = 'P' :

```
mysql> select nim, nama from mhs where sex='P';
```

nim	nama
A21.2001.00234	Dedy Sutanta
A21.2001.00214	Budi Satria

```
2 rows in set (0.00 sec)
```

### Menghapus Table

Tabel yang sudah ada dapat dihapus dengan perintah :

Drop Table <namatable>;

Contoh :

Akan dihapus table yang bernama mhs :



mysql> Drop Table mhs;

### Memasukkan Nilai NULL(The Null Value)

Apabila kita bermaksud untuk tidak mengisi suatu field, maka kita dapat memberikan nilai NULL kedalam field tersebut. Nilai null tidak dapat digunakan untuk operasi perbandingan aritmatika.

Contoh 1 :

```
mysql> select null < 0, null = 0, null != 0, null > 0;
+-----+-----+-----+-----+
| null < 0 | null = 0 | null != 0 | null > 0 |
+-----+-----+-----+-----+
|      NULL |      NULL |      NULL |      NULL |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Untuk proses pengecekan terhadap nilai null, kita harus menggunakan cara khusus yaitu menggunakan IS NULL atau IS NOT NULL, bukan = atau !=.

Contoh 2 :

```
mysql> Insert into mhs (nim, nama, alamat, sex) →
values ('A21.2001.02123','Andy', NULL, 'P');
```

### Mengubah Data dengan UPDATE

Untuk mengubah data yang terdapat pada record tertentu, digunakan pernyataan UPDATE dengan syntax sebagai berikut :

UPDATE nama\_tabel

SET kolom\_1 = nilai\_baru\_1,  
kolom\_2 = nilai\_baru\_2, .... [Where kondisi];

Contoh 1:

Akan diubah nama peserta didik yang NIMnya = A21.2001.02123 menjadi Andi Azis :

```
mysql> Update mhs set nama = 'Andi Azis'
→ Where nim = 'A21.2001.02123';
```

Contoh 2 :

Akan diubah nama dan alamat peserta didik yang NIMnya = A21.2001.02123 menjadi Andi Achmad, Jl. A. Yani 20. Semarang :

```
Mysql> Update mhs
  □ set nama = 'Andi Achmad' ,
  □ alamat = 'Jl. A. Yani 20. Semarang' →
  Where nim = 'A21.2001.02123';
```

Contoh 3 :

Akan diubah field sex semua peserta didik menjadi NULL :

```
Mysql> Update mhs set sex = null;
```

### Menghapus Data dengan DELETE

Untuk menghapus record, maka digunakan pernyataan DELETE, dengan syntax sebagai berikut :

```
DELETE FROM nama_table [WHERE kondisi];
```

Contoh 1 :

Akan dihapus data peserta didik yang nimnya = A21.2001.02123 :

```
Mysql> Delete from mhs Where nim = 'A21.2001.02123';
```

Contoh 2 :

Akan dihapus data peserta didik yang sex = 'P' :

```
Mysql> Delete from mhs Where sex = 'P';
```

Contoh 3 :

Akan dihapus semua record dari tabel mahasiswa :

```
Mysql> Delete from mhs;
```

### Memperoleh Data Yang Unik

Untuk memperoleh data yang unik (tidak ada yang kembar), maka digunakan pernyataan DISTINCT pada saat menuliskan perintah Select.

Bandingkan perintah berikut :

```
mysql> select sex from mhs;
+-----+
| sex   |
+-----+
| P     |
| P     |
| W     |
| W     |
+-----+
4 rows in set (0.03 sec)
```

```
mysql> select distinct(sex) from mhs;
+-----+
| sex   |
+-----+
| P     |
| W     |
+-----+
2 rows in set (0.02 sec)
```

### Menggunakan Operator BETWEEN

Operator between digunakan untuk menangani operasi jangkauan.

Contoh :

Akan menampilkan nama peserta didik yang nimnya antara A21.2001.00100 s.d A21.2001.00300 :

MySQL> Select nama From mhs

→ Where nim between 'A21.2001.00100' and 'A21.2001.00300';

### Menggunakan Operator LIKE

Operator LIKE digunakan untuk mencari data tertentu yang mempunyai sifat kemiripan.

Penggunaan operator LIKE dibarengi dengan simbol tertentu (wildcard) yaitu garis bawah ( \_ ) atau persen ( % ).

Tanda garis bawah ( \_ ) untuk mewakili sebuah karakter sedangkan (%) mewakili beberapa karakter.

Contoh 1 :

Menampilkan nama peserta didik yang namanya diawali dengan huruf 'S' :

```
mysql> select nama from mhs where nama like 'S%';
+-----+
| nama          |
+-----+
| Suci Andriani |
| Sulis         |
+-----+
2 rows in set (0.00 sec)
```

Contoh 2 :

Menampilkan nama peserta didik yang namanya diawali dengan huruf 'B' dan huruf ke empatnya adalah 'i' :

```
mysql> select nama from mhs where nama like 'B__i%';
+-----+
| nama      |
+-----+
| Budi Satria |
+-----+
1 row in set (0.00 sec)
```

Contoh 3 :

Menampilkan nama dan alamat peserta didik yang namanya mengandung huruf 's'

```
mysql> select nama, alamat from mhs where nama like '%s%';
+-----+-----+
| nama      | alamat      |
+-----+-----+
| Dedy Sutanta | Jl. A. Yani 20 |
| Budi Satria  | Jl. Dago 10   |
| Sulis        | Jl. Nakula I 25 |
| Suci Andriani | Jl. Seroja Raya |
+-----+-----+
4 rows in set (0.01 sec)
```

### Menggunakan Operator Regexp(Regular Expressions)

Operator ini memiliki kesamaan fungsi dengan like. Ada beberapa symbol khusus yang biasa digunakan bersama-sama dengan operator Regexp, seperti :

Simbol	Keterangan
.	Satu tanda titik (.) untuk mewakili satu karakter
[...]	Untuk mewakili beberapa karakter atau range yang ditentukan
^	Untuk posisi awal dari sebuah criteria yang ditentukan
\$	Untuk posisi akhir dari sebuah criteria yang ditentukan

Contoh 1 :

Akan ditampilkan nama dan alamat peserta didik yang berakhiran dengan huruf 'a' :

```
mysql> select nama, alamat from mhs where nama regexp 'a$';
+-----+-----+
| nama      | alamat      |
+-----+-----+
| Dedy Sutanta | Jl. A. Yani 20 |
| Budi Satria  | Jl. Dago 10   |
+-----+-----+
2 rows in set (0.00 sec)
```

Contoh 2 :

Akan ditampilkan nama dan alamat peserta didik yang namanya berakiran dengan huruf 'a' atau huruf 'o' :

```
mysql> select nama,alamat from mhs where nama regexp '[ai]$';
```

nama	alamat
Dedy Sutanta	Jl. A. Yani 20
Budi Satria	Jl. Dago 10
Suci Andriani	Jl. Seroja Raya

```
3 rows in set (0.00 sec)
```

Contoh 3 :

Akan ditampilkan nim dan nama peserta didik yang namanya berawalan dari huruf 'a' s.d huruf 'd' :

```
mysql> select nama,alamat from mhs where nama regexp '[ai]$';
```

nama	alamat
Dedy Sutanta	Jl. A. Yani 20
Budi Satria	Jl. Dago 10
Suci Andriani	Jl. Seroja Raya

```
3 rows in set (0.00 sec)
```

### Menggunakan Operator IN

Operator IN digunakan untuk mencocokkan suatu nilai dengan salah satu nilai yang ada pada suatu daftar nilai.

Contoh :

Akan ditampilkan nama dan alamat peserta didik yang namanya 'Dedy Sutanta' atau 'Budi Satria' atau 'Sulid' atau Suci Andriani' :

```
mysql> select nama,alamat from mhs where nama='Dedy Sutanta' or
-> nama='Budi Satria' or nama='Sulis' or nama='Suci Andriani';
+-----+-----+
| nama      | alamat      |
+-----+-----+
| Dedy Sutanta | Jl. A. Yani 20 |
| Budi Satria  | Jl. Dago 10    |
| Sulis        | Jl. Nakula I 25 |
| Suci Andriani | Jl. Seroja Raya |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select nama,alamat from mhs
-> where nama in ('Dedy Sutanta','Budi Satria','Sulis','Suci Andriani');
+-----+-----+
| nama      | alamat      |
+-----+-----+
| Dedy Sutanta | Jl. A. Yani 20 |
| Budi Satria  | Jl. Dago 10    |
| Sulis        | Jl. Nakula I 25 |
| Suci Andriani | Jl. Seroja Raya |
+-----+-----+
4 rows in set (0.00 sec)
```

Dari contoh di atas, dapat dilihat bahwa penggunaan operator IN untuk memilih salah satu alternatif lebih praktis.

### Menggunakan Operator Boolean AND, OR dan NOT

Operator boolean OR dan AND digunakan untuk query dengan kondisi yang majemuk.

Hasil ekspresi operator boolean AND, OR dan NOT dapat dilihat seperti tabel berikut :

Kondisi_1	Kondisi_2	AND	OR	NOT (kondisi_1)
Salah	Salah	Salah	Salah	Benar
Salah	Benar	Salah	Benar	Benar
Benar	Salah	Salah	Benar	Salah
Benar	Benar	Benar	Benar	Salah

Contoh 1 :

Akan menampilkan nama peserta didik yang tinggalnya di kota 'Semarang' dan jenis kelaminnya adalah Pria :

```
mysql> select nama from mhs
-> where kota='Semarang' and sex='P';
Empty set (0.01 sec)
```

Pesan 'Empty set' muncul disebabkan data yang diminta tidak terdapat di table mhs (table mhs masih kosong).

Contoh 2 :

Akan Menampilkan nama peserta didik yang namanya mengandung huruf 'i' dan jenis kelaminnya selain Wanita' :

```
mysql> select nama from mhs
-> where nama like '%i%' and sex != 'W';
+-----+
| nama      |
+-----+
| Budi Satria |
+-----+
1 row in set (0.01 sec)
```

#### 5. Latihan Pertemuan IV :

Untuk mempratekkan perintah-perintah dibawah ini, gunakanlah tabel mhs yang telah anda miliki.

Tuliskan perintah - perintah untuk menyelesaikan masalah-maslah dibawah ini dikertas, kemudian dikumpulkan.

1. Tambahkan record berikut kedalam tabel mhs : 'A12.2001.00001','Agus Wahyu','Jl. Nakula I No. 5','Semarang','P','2','1976-08-12'  
'A11.2001.00002','Sari Wijaya','A. Yani. 20','Solo','W','1','1977-10-11'  
'A22.2001.00003','Lestari','Nakula raya 20','Semarang','W','3','1975-11-15'
2. Tampilkan seluruh field dan record peserta didik.
3. Tampilkan nama peserta didik yang sex = 'P'
4. Tampilkan nama dan alamat peserta didik yang nim antara A21.2001.00001 s.d A21.2001.00100'
5. Tampilkan nama peserta didik yang nama depannya adalah 'Sari'
6. Tampilkan nama dan alamat peserta didik yang namanya mengandung huruf 'a' atau huruf 'i'.
7. Tampilkan nama, alamat dan kota peserta didik yang kota asalnya 'Semarang' atau 'Solo'.
8. Tampilkan nim, nama dan alamat peserta didik yang sex nya selain 'W'
9. Gantilah nama peserta didik yang nimnya 'A21.2001.00003' menjadi 'Lestari Handayani'

10. Gantilah alamat dan tanggal lahir peserta didik yang nimnya 'A21.2001.00002' menjadi 'Sari Wijayanti', '1977-10-17'
11. Hapuslah record peserta didik yang sex = 'P' atau agamanya = '3'
12. Hapuslah record-record peserta didik yang nimnya antara A21.2001.00100 s.d A21.2001.00300'



## PRAKTIKUM KE – V

1. Judul Materi : Perintah Manipulasi Data Lanjutan
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat menggunakan perintah-perintah manipulasi data lanjutan seperti pengelompokan, pengurutan, fungsi agregasi dan kombinasinya.
3. Jumlah Pertemuan : 1 Kali
4. Persiapan :

Untuk mempraktekan perintah-perintah pada pertemuan ini, terlebih dahulu masuklah ke database **latihan**. Di dalam database latihan ini sudah disiapkan sebuah table dengan nama **emp** yang isinya adalah sebagai berikut :

```
mysql> select * from emp;
```

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	1000	0	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02	2975	0	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7698	BLAKE	MANAGER	7839	1981-03-01	2850	0	30
7782	SCOTT	MANAGER	7839	1981-06-09	2450	0	10
7788	CLARK	ANALYST	7566	1981-04-19	3000	0	20
7839	KING	PRESIDENT	NULL	1981-10-17	5000	0	10
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1987-05-23	1100	0	20
7900	JAMES	CLERK	7698	1981-12-03	950	0	30
7902	FORD	ANALYST	7566	1981-12-03	3000	0	20
7934	MILLER	CLERK	7782	1982-01-23	1300	0	10

14 rows in set (0.02 sec)

5. Teori Dasar :

### PERINTAH MANIPULASI DATA (lanjutan)

#### Mengurutkan Data dengan ORDER BY

Untuk mengurutkan data, maka kita dapat menggunakan ORDER BY. Pengurutan dapat dilakukan secara ascending (urut baik) maupun urut menurun (descending).

Contoh 1 :

Mengurutkan berdasarkan sebuah field (field yang dijadikan field kunci adalah field nama berdasarkan urutan naik) :

```
mysql> Select * From emp Order by ename;
```

Contoh 2 :

Mengurutkan dengan urutan menurun (descending) :

```
mysql> Select * From emp Order by ename Desc;
```

Keterangan : Untuk menghasilkan urutan yang menaik (ascending) maupun menurun (descending), maka ditambahkan option ASC untuk ascending atau DESC untuk descending pada field yang dijadikan sebagai kunci pengurutan. Option ASC sifatnya optional artinya apabila kita ingin mengurutkan berdasarkan urutan ascending, maka option ASC dapat tidak dituliskan.

Contoh 3:

Mengurutkan berdasarkan beberapa field (field yang dijadikan field kunci adalah field job dan ename) :

```
mysql> select empno, ename, job from emp order by job, ename;
```

empno	ename	job
7788	CLARK	ANALYST
7902	FORD	ANALYST
7876	ADAMS	CLERK
7900	JAMES	CLERK
7934	MILLER	CLERK
7369	SMITH	CLERK
7698	BLAKE	MANAGER
7566	JONES	MANAGER
7782	SCOTT	MANAGER
7839	KING	PRESIDENT
7499	ALLEN	SALESMAN
7654	MARTIN	SALESMAN
7844	TURNER	SALESMAN
7521	WARDS	SALESMAN

```
14 rows in set (0.09 sec)
```

Contoh 4:

Mengurutkan berdasarkan beberapa field (field yang dijadikan field kunci adalah field job secara ascending dan ename secara descending) :

```
mysql> select empno, ename, job from emp order by job asc, ename desc;
+-----+-----+-----+
| empno | ename  | job      |
+-----+-----+-----+
| 7902   | FORD   | ANALYST  |
| 7788   | CLARK  | ANALYST  |
| 7369   | SMITH  | CLERK    |
| 7934   | MILLER | CLERK    |
| 7900   | JAMES  | CLERK    |
| 7876   | ADAMS  | CLERK    |
| 7782   | SCOTT  | MANAGER  |
| 7566   | JONES  | MANAGER  |
| 7698   | BLAKE  | MANAGER  |
| 7839   | KING   | PRESIDENT|
| 7521   | WARD   | SALESMAN |
| 7844   | TURNER | SALESMAN |
| 7654   | MARTIN | SALESMAN |
| 7499   | ALLEN  | SALESMAN |
+-----+-----+-----+
14 rows in set (0.00 sec)
```

Contoh 5:

Mengurutkan berdasarkan field ename secara descending (field kedua dari field yang disebutkan dalam Select) :

mysql> Select empno, ename, job From emp Order by 2 Desc;

### Mengelompokkan Data dengan GROUP BY

Untuk mengelompokkan data dapat digunakan perintah GROUP BY. Yang perlu diperhatikan pada saat melakukan pengelompokan data adalah proses pengelompokan hanya dapat dilakukan jika terdapat data yang sejenis.

Contoh 1 :

Akan dikelompokkan data pegawai (employee) berdasarkan job :

```
mysql> select job from emp group by job;
+-----+
| job      |
+-----+
| ANALYST  |
| CLERK    |
| MANAGER  |
| PRESIDENT|
| SALESMAN |
+-----+
5 rows in set (0.14 sec)
```

### Menentukan Kondisi pada GROUP BY

Untuk memberikan kondisi pada saat mengelompokkan data, kita dapat menggunakan HAVING. Perintah having hanya dapat digunakan jika group by digunakan.

Contoh 1 :

Akan dikelompokkan data pegawai berdasarkan job, kecuali pegawai yang job-nya 'CLERK' :

```
mysql> select job from emp group by job having job<>'CLERK';
+-----+
| job   |
+-----+
| ANALYST |
| MANAGER |
| PRESIDENT |
| SALESMAN |
+-----+
4 rows in set (0.03 sec)
```

### Menggunakan Fungsi Agregasi

Fungsi agregasi adalah fungsi yang disediakan oleh MySQL untuk menghasilkan sebuah nilai berdasarkan sejumlah data. Adapun beberapa fungsi agregasi pada MySQL adalah :

Fungsi Agregasi	Keterangan
AVG	Memperoleh nilai rata – rata
Count	Menghitung jumlah data
Max	Menghasilkan nilai terbesar
Min	Menghasilkan nilai terkecil
Sum	Menjumlahkan suatu nilai

### Fungsi AVG

Digunakan untuk menghitung nilai rata-rata dari seluruh field yang bertipe numerik.

Contoh 1 :

Menghitung rata-rata gaji pegawai :

```
mysql> select avg(sal) from emp;
+-----+
| avg(sal) |
+-----+
| 2087.5000 |
+-----+
1 row in set (0.04 sec)
```

Contoh 2 :

Menghitung rata-rata pegawai per job :

```
mysql> select job,avg(sal) from emp group by job;
```

job	avg(sal)
ANALYST	3000.0000
CLERK	1087.5000
MANAGER	2758.3333
PRESIDENT	5000.0000
SALESMAN	1400.0000

```
5 rows in set (0.01 sec)
```

Contoh 3 :

Menghitung rata-rata gaji pegawai yang job-nya 'SALESMAN' :

```
mysql> select job,avg(sal) from emp
-> group by job having job='SALESMAN';
```

job	avg(sal)
SALESMAN	1400.0000

```
1 row in set (0.02 sec)
```

### Fungsi COUNT

Digunakan untuk melakukan pencacahan terhadap data.

Contoh 1 :

Menghitung jumlah data pegawai di table emp:

```
mysql> select count(*) from emp;
```

count(*)
14

```
1 row in set (0.00 sec)
```

Contoh 2 :

Menghitung jumlah pegawai per job :

```
mysql> select job, count(*) from emp
-> group by job;
```

job	count(*)
ANALYST	2
CLERK	4
MANAGER	3
PRESIDENT	1
SALESMAN	4

```
5 rows in set (0.00 sec)
```

### Fungsi MAX

Digunakan untuk mencari nilai yang terbesar pada seluruh field yang bertipe numeric.

Contoh 1 :

Menampilkan informasi gaji terbesar :

```
mysql> select max(sal) from emp;
+-----+
| max(sal) |
+-----+
|      5000 |
+-----+
1 row in set (0.01 sec)
```

Contoh 2 :

Menampilkan informasi gaji terbesar pegawai per job :

```
mysql> select job,max(sal) from emp
-> group by job;
+-----+-----+
| job      | max(sal) |
+-----+-----+
| ANALYST  |      3000 |
| CLERK    |      1300 |
| MANAGER  |      2975 |
| PRESIDENT|      5000 |
| SALESMAN |      1600 |
+-----+-----+
5 rows in set (0.00 sec)
```

### Fungsi MIN

Digunakan untuk memperoleh nilai terkecil.

Contoh 1:

Menampilkan informasi gaji terkecil :

```
mysql> select min(sal) from emp;
+-----+
| min(sal) |
+-----+
|       950 |
+-----+
1 row in set (0.00 sec)
```

Contoh 2 :

Menampilkan informasi gaji terkecil pegawai per job :

```
mysql> select job,min(sal) from emp
-> group by job;
```

job	min(sal)
ANALYST	3000
CLERK	950
MANAGER	2450
PRESIDENT	5000
SALESMAN	1250

5 rows in set (0.02 sec)

### Fungsi SUM

Digunakan untuk melakukan penjumlahan isi dari suatu field numerik.

Contoh 1 :

Menghitung total gaji pegawai :

```
mysql> select sum(sal) from emp;
```

sum(sal)
29225

1 row in set (0.00 sec)

Contoh 2:

Menghitung jumlah gaji pegawai per job dengan tampilanurut naik berdasarkan job :

```
mysql> select job, sum(sal) from emp
-> group by job order by job;
```

job	sum(sal)
ANALYST	6000
CLERK	4350
MANAGER	8275
PRESIDENT	5000
SALESMAN	5600

5 rows in set (0.00 sec)

Contoh 3:

Menghitung jumlah gaji pegawai per job dengan tampilanurut menurun berdasarkan jumlah gaji :

```
mysql> select job, sum(sal) as gaji from emp
-> group by job order by gaji desc;
```

job	gaji
MANAGER	8275
ANALYST	6000
SALESMAN	5600
PRESIDENT	5000
CLERK	4350

```
5 rows in set (0.02 sec)
```

## 5. Latihan Praktikum V :

1. Buatlah sebuah tabel pada database yang sudah anda miliki dengan nama tabel mkuliah dan struktur tabelnya adalah sebagai berikut :

Nama Field	Type Data	Keterangan
Kode_kul	Char(7)	Primary key, tidak boleh, unik
Nama_kul	Char(20)	Tidak boleh kosong
SKS	Decimal(2)	Tidak boleh kosong
Smt_tawar	Char(1)	Diisikan 1,2,3,4 dst
Klp_kul	Char(1)	Dapat diisikan angka 1-3 : 1 = MPK 2 = MKK 3 = MKB default adalah 1
Progdi	Char(1)	Dapat diisikan angka 1-2 : 1 = MI-D3 2 = SI-S1 3 = TI-S1 4 = TI-D3 5 = KA-D3

2. Isikan data – data berikut :  
 'A21-101','Pancasila', 2, '1', '1','1'  
 'A21-102','Logika Algoritma', 4, '1', '2','1'  
 'A21-103','Pemrograman Basic', 2, '1', '2','1'  
 'A21-201','Sistem Basis Data I', 2, '2', '2','1'



'A21-202', 'Pemrograman Pascal', 4, '2', '2', '1'  
'A21-203', 'Ilmu Sosial Dasar', 2, '2', '1', '1'  
'A21-501', 'Bimbingan Karier', 2, '5', '3', '1' 'A21-601', 'Proyek  
Akhir', 4, '6', '3', '1'

3. Tampilkan seluruh data dari tabel mkuliah.
4. Tampilkan Nama mata pelajaran yang sksnya 2 dan ditawarkan disemester 6
5. Tampilkan nama mata pelajaran yang sksnya lebih besar dari 2 sks
6. Tampilkan nama mata pelajaran yang huruf depannya dimulai dengan 'P'
7. Gantilah nama mata pelajaran 'Pancasila' menjadi 'Pendidikan Pancasila'
8. Gantilah nama mata pelajaran 'Pemrograman Basic' menjadi 'Pemrograman (Basic)'
9. Tambahkan record-record berikut :  
'A12-202', 'Pemrograman Pascal', 4, '2', '2', '2'  
'A22-203', 'Kewiraan', 2, '2', '1', '2'  
'A11-501', 'Bimbingan Karier', 2, '5', '3', '2'  
'A12-601', 'Tugas Akhir', 6, '6', '3', '2'
10. Gantilah nama mata pelajaran 'Kewiraan' menjadi 'Pendidikan Kewarganegaraan'
11. Hitunglah seluruh data yang telah tersimpan pada tabel mkuliah tersebut.
12. Tampilkan nama mata pelajaran dan sks yang telah diurutkan berdasarkan kunci nama mata pelajaran secara ascending
13. Tampilkan nama mata pelajaran, sks, semester tawar dan program studi yang telah diurutkan berdasarkan field kunci semester tawar dengan urutan ascending dan field kunci program studi dengan urutan descending
14. Kelompokkan data mata pelajaran berdasarkan kelompok mata pelajaran.
15. Tampilkan kelompok mata pelajaran dan hitung jumlah sks berdasarkan kelompok mata pelajaran
16. Tampilkan kode program studi dan hitung jumlah sks berdasarkan kelompok program studi
17. Tampilkan jumlah sks terkecil untuk program studi MI-D3
18. Tampilkan jumlah sks terbesar untuk program studi MI-D3
19. Tampilkan jumlah sks terkecil untuk program studi SI-S1

20. Tampilkan sks rata-rata untuk mata pelajaran kelompok MKP
21. Tampilkan sks rata-rata untuk mata pelajaran kelompok MKB
22. Hitung jumlah sks untuk mata pelajaran kelompok MPK
23. Hitung jumlah sks untuk mata pelajaran kelompok MKK
24. Berapa jumlah mata pelajaran yang termasuk dalam kelompok MPK

## **PRAKTIKUM KE – VI dan VII**

1. Judul Materi : Fungsi – Fungsi SQL
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat menggunakan fungsi-fungsi SQL, ekspresi waktu dan tanggal serta kombinasinya.
3. Jumlah Pertemuan : 2 Kali
4. Teori Dasar :

### **FUNGSI-FUNGSI SQL**

Ada dua kategori fungsi-fungsi dalam Oracle :

1. Single-Row : Menerima satu baris sebagai operand-nya
2. Multiple-Row : Menerima lebih dari satu sebagai operand-nya

Fungsi-fungsi seperti Group By, Sum, Max, Min, Count dan Avg merupakan contoh fungsi yang termasuk dalam multiple-row.

Sedangkan fungsi-fungsi Single-Row terbagi menjadi : a.

Fungsi Character / String

- b. Fungsi Date (Tanggal dan waktu)
- c. Fungsi Numeric
- d. Fungsi General

### Fungsi Karakter / String

Merupakan fungsi-fungsi yang berorientasi ke string, yaitu :

Function	Result
Lower('MySQL Database Server')	Mysql database server
Upper('MySQL Database Server')	MYSQL DATABASE SERVER
Initcap('MySQL DATABASE SERVER')	Mysql Database Server
Concat('MySQL', ' Database Server')	MySQL Database Server
Substr('MySQL Database Server',3,3)	SQL
Left('MySQL Database Server',5)	MySQL
Right(MySQL Database Server',6)	Server
Length('MySQL Database Server')	22
Instr('MySQL Database Server', 'Q')	4
Ascii('a')	97
Ascii('b')	98
Ascii(null)	Null
Char(65)	A
Conv('11111111',2,8)	377
Conv('e',16,2)	1110
Format(1234.56789,3)	1,234.568
Format(999999.99,2)	999,999.99
Format(1000000.00,0)	1,000,000
Lpad(800,8,'*')	*****800
Rpad(800,8,'*')	800*****
Ltrim(' UDINUS ')	UDINUS
Rtrim(' UDINUS ')	UDINUS
Repeat('x',10)	Xxxxxxxxxx
Repeat('abc',3)	Abcabcab

Reverse('UDINUS')	SUNIDU
Reverse('Fasilkom')	MoklisaF
Replace('fik','i','Ilmu')	FIlmuk
Replace('fik','k','Komputer')	fiKomputer
Insert('nighttime',6,4,'fall')	Nightfall
Insert('sunshine',1,3,'rain or')	Rain or shine
Trim(' Fasilkom ')	Fasilkom

### Fungsi Numerik

Merupakan fungsi-fungsi yang beoreintasi pada nilai numerik, yaitu :

Fungsi	Hasil	Keterangan
Truncate (45.923, 2)	45.92	
Truncate (45.923)	45	
Truncate(45.923, -1)	40	
Truncate(451.923, -2)	400	
Round(45.923, 2)	45.92	
Round(45.927, 2)	45.93	
Round(45.927)	46	
Round(45.927, -1)	50	
Round(45.927, -2)	0	
Mod(10, 3)	1	
Sqrt(625)	25	
Sign(15.803)	1	
Sign(0)	0	
Sign(-98)	-1	
Floor(3.8)	3	Return the smallest integer
Floor(-3.8)	-4	Return the smallest integer
Ceiling(3.8)	4	Return the largest integer
Ceiling(-3.8)	-3	Return the largest integer
Abs(13.5)	13.5	Absolute value
Abs(-13.5)	13.5	Absolute value
Acos(1)	0.000000	Arccosine of
Cos(0)	1.000000	Cosine of
Cot(PI()/2)	0.000000	Cotangent of
Asin(1)	1.570796	Arcsine of
Sin(0)	0.000000	Sine of
Tan(0)	0.000000	Tangent of
Atan(1)	0.785398	Arctangent of

Log(1)	0.000000	Logaritm of
Log10(10)	1.000000	Logaritm of
PI( )	3.141593	The value of $\pi$

Contoh :

```
mysql> select truncate(45.928,2), round(45.928,2), mod(10,3);
+-----+-----+-----+
| truncate(45.928,2) | round(45.928,2) | mod(10,3) |
+-----+-----+-----+
| 45.92 | 45.93 | 1 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Buatlah sebuah kesimpulan mengenai fungsi truncate, round dan floor.

### Fungsi General

Fungsi General merupakan fungsi-fungsi yang tidak termasuk dalam fungsi string, numeric dan tanggal. Fungsi-fungsi tersebut antara lain adalah :

1. Fungsi Encode.

Digunakan untuk mengkonversikan string tertentu ke binary string.

Bentuk umum : Encode(str,  
password) Dimana :

Str adalah string yang akan dikonversi

Password adalah kata kunci untuk perintah konversi tersebut Contoh

```
mysql> select encode('udinus','fik');
+-----+
| encode('udinus','fik') |
+-----+
| Ȳŧŧ°▼Ä                 |
+-----+
1 row in set (0.00 sec)
```

## 2. Fungsi Decode.

Digunakan untuk mengembalikan nilai ke string yang telah dilakukan proses encode.

Bentuk Umum :

Decode (str, password)

Keterangan :

Str adalah string yang nilainya akan dikembalikan Password adalah kata kunci untuk perintah konversi tersebut Contoh :

```
mysql> select decode(encode('udinus','fik'),'fik');
+-----+
| decode(encode('udinus','fik'),'fik') |
+-----+
| udinus                                |
+-----+
1 row in set (0.00 sec)
```

### Fungsi Tanggal & Waktu

SQL92 menyediakan ekspresi DATE, TIME, NOW( ), SYSDATE( ) dan TIMESTAMP.

Date menyatakan sebuah tanggal dengan susunan : yyyy-bb-tt, misalnya 1964-9-1.

Time menyatakan waktu, dengan format : jj:mm:dd

Timestamp menyatakan sebuah tanggal dan waktu, misalnya 1964-9-1 5:32:24

Now() akan menghasilkan sebuah tanggal dan waktu seperti Timestamp Contoh

1 :

Menampilkan nama peserta didik dan tanggal lahir yang tanggal lahirnya diatas tanggal 1964/9/1 :

Mysql> Select nama, tgl\_lhr From mhs Where tgl\_lhr > Date '1977/01/01'; atau

Mysql> Select nama, tgl\_lhr From mhs Where tgl\_lhr > '1977/01/01';  
atau

Mysql> Select nama, tgl\_lhr From mhs Where tgl\_lhr > '1977-01-01';

Sql menyediakan sejumlah fungsi tanggal sebagai berikut :

Fungsi	Keterangan
Current_date atau Curdate( )	menghasilkan tanggal sekarang
Current_time atau Curtime( )	menghasilkan waktu sekarang
Current_timestamp	menghasilkan tanggal dan waktu sekarang
Month( )	Menghasilkan angka nomor urut bulan
Monthname( )	Menghasilkan nama bulan
Dayofmonth( )	Menghasilkan angka nomor urut tanggal
To_days( )	Menghasilkan jumlah hari dari suatu tanggal
Year( )	Menghasilkan angka tahun

Dayname( )	Menghasilkan nama hari
Hour( )	Menghasilkan angka jam
Minute( )	Menghasilkan angka menit
Second( )	Menghasilkan angka detik
Weekday( )	Menghasilkan angka urut dalam satu minggu

Contoh 1 :

```
mysql> select current_date,current_time, current_timestamp;
+-----+-----+-----+
| current_date | current_time | current_timestamp |
+-----+-----+-----+
| 2005-08-09   | 07:10:42    | 2005-08-09 07:10:42 |
+-----+-----+-----+
1 row in set (0.02 sec)
```

Contoh 2 :

```
mysql> select monthname(current_date),month(current_date),year(current_date);
+-----+-----+-----+
| monthname(current_date) | month(current_date) | year(current_date) |
+-----+-----+-----+
| August                  | 8                   | 2005               |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Contoh 3 :

Perhatikan dan analisislah contoh berikut :

```
mysql> select nama,floor((to_days(current_date)-to_days(tgl_lhr))/365) as umur from mhs;
+-----+-----+
| nama      | umur |
+-----+-----+
| Dedy Sutanta | 25   |
| Budi Satria  | 26   |
| Sulis       | 30   |
| Suci Andriani | 27   |
+-----+-----+
4 rows in set (0.01 sec)

mysql> select nama,(to_days(current_date)-to_days(tgl_lhr))/365 as umur from mhs;
+-----+-----+
| nama      | umur |
+-----+-----+
| Dedy Sutanta | 25.82 |
| Budi Satria  | 26.73 |
| Sulis       | 30.00 |
| Suci Andriani | 27.90 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select nama,year(current_date)-year(tgl_lhr) as umur from mhs;
+-----+-----+
| nama      | umur |
+-----+-----+
| Dedy Sutanta | 26   |
| Budi Satria  | 27   |
| Sulis       | 30   |
| Suci Andriani | 28   |
+-----+-----+
4 rows in set (0.00 sec)
```

apa kesimpulan mengenai contoh di atas ?

Contoh 4 :

Akan menampilkan nama peserta didik yang usianya lebih dari atau sama dengan 20 tahun

```
Mysql> Select nama_mhs From Mhs
→ Where year(current_date) – year(tgl_lahir ) >= 20;
```

### **Lebih lanjut dengan SELECT**

Sejauh ini nilai yang ditampilkan pada select selalu berupa nama field. Tetapi sebetulnya kita juga dapat menggunakan ekspresi pada kolom-kolom yang ditentukan pada perintah Select.

#### □ Menggunakan perintah Limit

Pada saat perintah menampilkan data dengan select diberikan, maka data yang sesuai dengan persyaratan (jika ada) akan ditampilkan berapun jumlah recordnya. Apabila ternyata jumlah record yang ditampilkan sangat banyak, maka record-record tersebut akan ditampilkan begitu saja sehingga sulit untuk membaca setiap datanya. Sebagai jalan keluar gunakanlah perintah limit. Perintah ini akan membatasi jumlah record yang diinginkan berdasarkan nilai limitnya.

Contoh 1 :

Untuk menampilkan data 10 record pertama dari table mhs, maka perintahnya :

```
Mysql> select nim, nama, alamat, tgl_lhr from mhs Limit 10;
```

Contoh 2 :

Sedangkan Untuk menampilkan data 5 record table mhs dihitung dari posisi record ke 15 adalah :

```
Mysql> select nim, nama, alamat, tgl_lhr from mhs Limit 15,5;
```

Contoh 3 :

Apabila yang ingin ditampilkan adalah 5 record terakhir table mhs, maka kita harus mengetahui jumlah seluruh record pada table mhs (gunakan perintah : select count(\*) from Mhs), misalnya jumlahnya adalah 40 berarti untuk lima data terakhir dimulai pada posisi recor ke 35 :



```
Mysql> select nim, nama, alamat, tgl_lhr from mhs Limit 35,5;
```

- Menggunakan option BINARY agar data Case-Sensitive Perhatikan penulisan perintah berikut :

```
mysql> select nim,nama,alamat from mhs
-> where kota='solo';
+-----+-----+-----+
| nim          | nama      | alamat      |
+-----+-----+-----+
| A21.2001.00234 | Dedy Sutanta | Jl. A. Yani 20 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select nim,nama,alamat from mhs
-> where binary kota='solo';
Empty set (0.00 sec)

mysql> select nim,nama,alamat from mhs
-> where kota= binary 'solo';
Empty set (0.00 sec)
```

Dari contoh di atas dapat dilihat bahwa dengan menambahkan option BINARY maka data yang ditulis untuk pencarian harus sama dengan data yang tersimpan ditabel

(case-sensitive). Sehingga kondisi seperti di atas : *where binary kota = 'solo'* akan menghasilkan himpunan yang kosong (data yang dicari tidak ada), karena yang tersimpan di table adalah 'Solo' bukan 'solo'.

- Menuliskan ekspresi pada kolom Contoh :

Akan menampilkan seluruh nama dan gaji pegawai setelah gaji dinaikkan 5% dari gaji semula :

```
Mysql> Select ename, sal + (0.05*sal) From emp;
```

- Memberi nama lain terhadap kolom Contoh :

Akan menampilkan seluruh nama dan gaji pegawai setelah gaji dinaikkan 5% dari gaji semula dengan perubahan nama kolom :

```
Mysql> Select ename, sal + (0.05*sal) as gjbbaru From emp;
```

Atau

```
Mysql> Select nama, sal + (0.05*sal) as 'Gaji Baru' From emp;
```

#### ❑ Menambahkan Teks dalam baris keluaran

Pernyataan select memungkinkan pemakai menambahkan suatu teks.

Contoh :

Akan menampilkan nama dan gaji seluruh pegawai dengan menambahkan teks rupiah setelah angka gaji :

```
mysql> Select ename, sal ,'Rupiah' From emp;
```

#### ❑ Pemakaian Select tanpa klausa FROM

Cara ini digunakan untuk mengevaluasi ekspresi tertentu yang tidak ada kaitannya dengan tabel.

Contoh 1 :

Akan dihitung nilai perkalian 2 \* 1 :

```
mysql> select 2 * 1;
+-----+
| 2 * 1 |
+-----+
|      2 |
+-----+
1 row in set (0.04 sec)
```

Contoh 2 :

Akan dicek nilai kondisi angka 2 dan 1 :

```
mysql> select 2 > 1, 2 < 1, 2 = 1;
+-----+-----+-----+
| 2 > 1 | 2 < 1 | 2 = 1 |
+-----+-----+-----+
|      1 |      0 |      0 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

#### ❑ Ekspresi Berkondisi

Ekspresi dimanfaatkan untuk mengubah keluaran yang berbeda dengan data asalnya.

Syntax :

Case nilai\_ekspresi

When nilai\_ekspresi Then nilai\_ekspresi

[Else nilai\_ekspresi]0

End

Contoh 1 :

Akan ditampilkan kota 'Yogya' menjadi 'Jogjakarta' :

```
Mysql> Select nama, Case kota when 'Yogya'
→ Then 'Jogjakarta' End From Pegawai;
```

Contoh 2 :

Akan ditampilkan nama, nilai dan skor nilai peserta didik :

```
Mysql> Select nama_mhs, Nilai,
□ Case
□ When nilai >= 80 Then 'A'
□ When nilai >= 70 Then 'B'
□ When nilai >= 60 Then 'C'
□ When nilai >= 50 Then 'D'
□ Else 'E'
□ End as Skor
□ From Ujian;
```

## 5. Latihan Praktikum VI dan VII :

- Buatlah tabel dengan nama Dosen pada databases masing-masing dan struktur tabelnya adalah sebagai berikut :

Nama Fied	Type Data	Keterangan
Kode_dos	Int(4)	Primary key, tidak boleh kosong, unik, auto_increment, unsigned
Nama_dos	Varchar(30)	tidak boleh kosong
Alamat_dos	Varhar(50)	
Tgl_masuk	Date	

- Tambahkan field kota\_dos char(20) yang ditempatkan setelah field alamat\_dos.

- Isikan record-record ke table dosen :

Kode_dos	Nama_dos	Alamat_dos	Kota_dos	Tgl_masuk
1	Arief Budiman, Ir.	Jl. A. Yani 20	Semarang	19-02-1990
2	Husni, Prof.	Jl. Merdeka Raya	Semarang	13-10-1988
3	Candawati, Dra.	Perum Korpri I	Solo	20-09-1994
4	Irawan S., DR.	Jl. Kemuning Barat 11	Semarang	15-04-1992
5	Sumiati, SE., M.Si.	Jl. Seroja Indah 100	Kudus	11-08-1995

- Editlah data alamat, kota dan tanggal masuk dosen yang bernama Candawati, Dra.

Menjadi :

Alamat\_dos = Perum Pondok Indah C-10

Kota\_dos = Semarang

Tgl\_masuk = 21-09-1994

- Tampilkan nama\_dos, kota\_dos dan lama\_kerja seluruh dosen dengan tampilan :

Nama_dos	Alamat_dos	Lama_kerja

- Buatlah Tabel Krs dengan struktur sebagai berikut :

Nama Fied	Type Data	Keterangan
Kode_kul	Char(7)	tidak boleh kosong
Nim	Char(14)	tidak boleh kosong
Kode_dos	Int(4)	Tidak boleh kosong
Nilai	Tinyint	Diisi angka 0 .. 100

Keterangan : kunci primer adalah Kode\_kul, Nim dan Kode\_dos

- Isikan record-record berikut pada table krs :

'A21-101','A21.2001.00234',2,80

'A21-201','A21.2001.00214',4,75

'A21-201','A21.2001.00234',1,89

'A21-601','A21.2001.00214',3,90

'A21-501','A21.2001.00214',5,85

'A21-501','A21.2001.00234',5,77

'A21-101','A21.2001.00224',2,95

- Dari table krs tampilkan data nilai angka dan nilai huruf dengan format sebagai berikut :

Kode_kul	NIM	Kode_dos	Nilai Angka	Nilai Huruf

ketentuan nilai huruf adalah :

Nilai Angka 0..40 → huruf 'E'

Nilai Angka 41..55 → huruf 'D'

Nilai Angka 56..70 → huruf 'C'

Nilai Angka 71..85 → huruf 'B'

Nilai Angka 86..100 → huruf 'A'

## **PRAKTIKUM KE – VIII dan IX**

1. Judul Materi : Queri Antar Tabel
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat merelasikan beberapa tabel untuk penyajian informasi
3. Jumlah Pertemuan : 2 Kali

4. Persiapan :

Untuk mempraktekan materi query antar table ini, gunakan database **latihan**. Di dalam database ini telah ditambahkan sebuah table **dept** yang isinya sebagai berikut :

```
mysql> select * from dept;
```

deptno	dname	loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
4 rows in set (0.00 sec)
```

Sehingga di database latihan, sekarang telah tersedia dua buah table yang dapat digunakan untuk mempraktekan relasi antar table yaitu table emp dan table dept.

## 5. Teori Dasar :

### QUERY ANTAR TABEL

#### Jenis-jenis table Berdasarkan Fungsinya

##### Tabel Data

Adalah sebuah table yang menampung data-data yang digunakan sebagai informasi. Table data merupakan table yang sangat umum digunakan dalam database relasi. Biasanya sifat data pada table ini adalah dinamis karena sering diubah, dihapus dan diperbaharui.

##### Tabel Validasi

Adalah sebuah table yang menampung data-data yang memiliki criteria tertentu dan biasanya digunakan oleh data pada table lainnya sebagai validasi. Tabel validasi disebut juga table referensi (reference table) dan table pencarian (look-up table). Oleh karena itu biasanya table validasi harus lebih dulu ada struktur data dan isinya, sebelum table-table lainnya.

##### Tabel Penghubung

Adalah sebuah table yang berfungsi sebagai ‘jembatan’ antara dua table yang mengalami relasi many-to-many. Dengan adanya table penghubung, maka hubungan relasi yang terjadi akan berubah dari many-to-many menjadi one-to-many.

#### Menggabungkan Tabel

Sql mempunyai kemampuan untuk menggabungkan dua atau lebih tabel untuk membentuk sebuah informasi. Sedangkan prosesnya disebut join.

Contoh :

Akan menampilkan nama pegawai yang lokasin departemennya di DALLAS.

```
Mysql> Select ename
□      From emp, dept
□      where emp.deptno = dept.deptno and
□      loc = 'DALLAS';
```

Keterangan :

Garis bawah : digunakan untuk menggambarkan bentuk relasi tabel emp dengan dept dengan kunci relasi deptno.

Emp.deptno menunjukan field deptno dari table emp. Hal ini dilakukan karena field deptno terdapat ditabel emp dan dept. Untuk field yang namanya unik (hanya terdapat pada satu table) maka penulisannya tidak perlu menyebutkan nama table pemiliknya.

### Menggunakan Alias

Nama tabel yang digunakan dalam query antartabel dapat diganti dengan menyebutkan nama aliasnya :

```
Mysql> Select ename
□      From emp x, dept y
□      where x.deptno = y.deptno and
□      loc = 'DALLAS';
```

### Operator Cross Join / Perkalian Kartesian

Operator Cross Join digunakan untuk melakukan operasi perkalian kartesian.

Contoh :

Akan dilakukan operasi cross join untuk dua buah table :

sales

Sales_id	Nama_sales
101	Adi
102	Reni

pelanggan

Nama_pel	Alamat_pel	Sales_id
Agus	Jl. Sempit	102
Ayu	Jl. Longgar	101

Dengan perintah :

mysql> Select \* From sales, pelanggan; *atau*  
*dapat ditulis* mysql> Select \* From sales cross  
 join pelanggan;

Maka table dihasilkan adalah :

```
mysql> select * from sales, pelanggan;
```

sales_id	nama_sales	nama_pel	alamat_pel	sales_id
101	Adi	Agus	Jl. Sempit	102
102	Reni	Agus	Jl. Sempit	102
101	Adi	Ayu	Jl. Longgar	101
102	Reni	Ayu	Jl. Longgar	101

4 rows in set (0.03 sec)

Keterangan : Tabel yang dihasilkan mempunyai jumlah record yang merupakan perkalian antara jumlah record table sales dan jumlah record table pelanggan. Record-record pada table hasil cross join merupakan kombinasi dari record pada table sales dengan record-record pada table pelanggan, demikian seterusnya sehingga semua record pada table sales telah terkombinasikan.

Apabila kita ingin melakukan relasi antara kedua table tersebut, maka kita hubungkan data yang sama antara table sales dan pelanggan sebagai kunci relasi. Jadi, relasi akan terjadi apabila di kedua table tersebut terdapat data yang sama (kunci relasi). Sehingga Operasi cross join di atas diperbaiki menjadi :

```
mysql> select * from sales, pelanggan
-> where sales.sales_id = pelanggan.sales_id;
```

sales_id	nama_sales	nama_pel	alamat_pel	sales_id
102	Reni	Agus	Jl. Sempit	102
101	Adi	Ayu	Jl. Longgar	101

2 rows in set (0.01 sec)

Keterangan : Dari operasi di atas, yang dihasilkan hanya record yang sales.sales\_id = pelanggan.sales\_id.

### Equajoin

Adalah penggabungan antar tabel yang menggunakan operator = pada kondisi Where. (sebagai contoh perhatian penggabungan antar tabel seperti diatas)



Selfjoin

Adalah penggabungan antar tabel yang sama :

Contoh :

Akan ditampilkan nama pegawai yang job-nya 'salesman' dan pegawai yang job-nya 'manager' :

```

Mysql> Select x.ename, y.ename
        From emp x, emp y
        Where x.job = 'salesman' and y.job = 'manager';

```

Operator Union

Digunakan untuk menggabungkan hasil dari dua buah query.

Contoh :

Pinjam

Bname	Loan#	Cname	Amount
BCA Pemuda	17	Jony	1000000
BCA Majapahit	23	Seno	2000000

Tabungan

Bname	account#	Cname	Balance
BCA Pemuda	102	Hari	400000
BCA Pemuda	204	Seno	700000

Akan menampilkan semua nama nasabah yang mempunyai tabungan, pinjaman atau keduanya :

```

Mysql> (Select Cname From Pinjam)
        Union
        (Select Cname From Tabungan);

```

Operator Intersect

Digunakan untuk memperoleh baris-baris yang terdapat pada kedua tabel.

Contoh :

Akan menampilkan semua nama nasabah yang mempunyai pinjaman dan pinjaman :

```

Mysql> (Select Cname From Pinjam)
        Intersect
        (Select Cname From Tabungan);

```

### Operator Except / Minus

Jika terdapat tabel A dan B, operasi except akan menghasilkan semua yang ada pada tabel A tetapi tidak terdapat pada tabel B :

Contoh :

Akan menampilkan semua nama nasabah yang mempunyai simpanan tetapi tidak mempunyai pinjaman.

Mysql> (Select Cname From Pinjam)

- Except
- (Select Cname From Tabungan);

Keterangan : Perintah-perintah seperti UNION, INTERSECT dan EXCEPT / MINUS hanya dapat dipoperasikan pada MySQL mulai versi 3.24. Sehingga apabila MySQL yang kita gunakan dibawah versi tersebut, maka perintah-perintah tersebut tidak dapat berfungsi.

### Operator Natural Join

Operator natural join akan melakukan operasi equijoin dengan memperlakukan nama-nama field yang sama sebagai field penghubung antar tabel.

Contoh :

Akan direlasikan dua buah table yaitu table emp dan dept dengan kunci relasinya adalah deptno.

```
mysql> select ename, dname, loc from emp, dept
-> where emp.deptno = dept.deptno
-> order by ename;
```

ename	dname	loc
ADAMS	RESEARCH	DALLAS
ALLEN	SALES	CHICAGO
BLAKE	SALES	CHICAGO
CLARK	RESEARCH	DALLAS
FORD	RESEARCH	DALLAS
JAMES	SALES	CHICAGO
JONES	RESEARCH	DALLAS
KING	ACCOUNTING	NEW YORK
MARTIN	SALES	CHICAGO
MILLER	ACCOUNTING	NEW YORK
SCOTT	ACCOUNTING	NEW YORK
SMITH	RESEARCH	DALLAS
TURNER	SALES	CHICAGO
WARDS	SALES	CHICAGO

14 rows in set (0.02 sec)

Atau dapat ditulis :

```
mysql> select ename, dname, loc from emp natural join dept
-> order by ename;
```

ename	dname	loc
ADAMS	RESEARCH	DALLAS
ALLEN	SALES	CHICAGO
BLAKE	SALES	CHICAGO
CLARK	RESEARCH	DALLAS
FORD	RESEARCH	DALLAS
JAMES	SALES	CHICAGO
JONES	RESEARCH	DALLAS
KING	ACCOUNTING	NEW YORK
MARTIN	SALES	CHICAGO
MILLER	ACCOUNTING	NEW YORK
SCOTT	ACCOUNTING	NEW YORK
SMITH	RESEARCH	DALLAS
TURNER	SALES	CHICAGO
WARDS	SALES	CHICAGO

14 rows in set (0.00 sec)

##### 5. Latihan Praktikum VIII dan IX :

Dengan menggunakan table-tabel yang sudah dibuat sebelumnya seperti table mhs, mkuliah, dosen dan krs, maka tuliskan query untuk menampilkan informasi-informasi yang diinginkan seperti dibawah ini :

3. Tampilkan nama peserta didik, nama mata pelajaran dan nilai untuk setiap peserta didik yang telah mengambil mata pelajaran yang sudah ada nilainya
4. Tampilkan nama peserta didik yang sudah mengambil mata pelajaran 'Bimbingan Karier'

5. Tampilkan nama peserta didik yang sudah mengambil mata pelajaran dan nilainya adalah 'A'
6. Tampilkan nama peserta didik yang belum mengambil mata pelajaran 'Proyek Akhir'
7. Berapa jumlah sks kumulatif dari peserta didik yang nimnya adalah 'A21.2001.00214'
8. Berapa jumlah mata pelajaran yang sudah diambil dari peserta didik yang nimnya adalah 'A21.2001.00214'
9. Tampilkan nama peserta didik yang telah mengambil mata pelajaran 'Proyek Akhir' dan nilainya adalah 'A'
10. Tampilkan daftar nama peserta didik yang mengambil mata pelajaran 'Bimbingan Karier' dan 'Proyek Akhir'.
11. Tampilkan daftar nama peserta didik yang sudah mengambil mata pelajaran 'Bimbingan Karier' tetapi belum mengambil 'Proyek Akhir'.

## PRAKTIKUM KE – X

1. Judul Materi : Query Antar Tabel Lanjutan dan Sub Query
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat pengelolaan tabel dengan menggunakan beberapa jenis join dan sub query sehingga penyajian informasi lebih mudah
3. Jumlah Pertemuan : 1 Kali
4. Teori Dasar :

### QUERY ANTAR TABEL (Lanjutan)

Menghubungkan dua atau lebih table disebut dengan join. Sebenarnya kita telah menggunakan join pada beberapa contoh-contoh sebelumnya, misalnya :

```
Mysql> Select ename  
□ From emp, dept  
□ where emp.deptno = dept.deptno ;
```

Pada contoh tersebut, kita dapat menampilkan nama pegawai dan nama departemennya. Tetapi tidak semua pegawai tersebut memiliki (berelasi) dengan departemen. Jadi konsep join sebenarnya adalah menampilkan data tepat memiliki pasangan dikedua table yang saling berkait. Tetapi bila tidak ada pasangannya, data tersebut tidak ditampilkan. Jenis join seperti ini disebut seperti equajoin. Bagaimana kalau kita ingin menampilkan data data yang tidak memiliki pasangan data ? Agar hal ini dapat dilakukan, maka kita dapat menggunakan LEFT JOIN atau RIGHT JOIN.

#### Left Join

Perintah ini digunakan untuk menampilkan data-data seolah-olah dari sudut pandang table pertama.

Perhatikan contoh berikut :

```
mysql> select * from t1;
+-----+-----+
| i1    | c1    |
+-----+-----+
|      1 | a     |
|      2 | b     |
|      3 | c     |
+-----+-----+
3 rows in set (0.03 sec)
```

```
mysql> select * from t2;
+-----+-----+
| i2    | c2    |
+-----+-----+
|      2 | c     |
|      3 | b     |
|      4 | a     |
+-----+-----+
3 rows in set (0.00 sec)
```

Dengan perintah LEFT JOIN menjadi :

```
mysql> select t1.* from t1 left join t2 on t1.i1 = t2.i2
-> where t2.i2 is null;
+-----+-----+
| i1    | c1    |
+-----+-----+
|      1 | a     |
+-----+-----+
1 row in set (0.00 sec)
```

### Right Join

Perintah ini akan menghasilkan tampilan yang sama seperti left join, hanya saja posisi peletakan nama tabelnya sedikit berbeda.

Perhatikan contoh berikut :

```
mysql> select t2.* from t1 right join t2 on t1.i1 = t2.i2
-> where t1.i1 is null;
+-----+-----+
| i2    | c2    |
+-----+-----+
|      4 | a     |
+-----+-----+
1 row in set (0.00 sec)
```

## SUB QUERY (SUBSELECT)

### Pengertian Sub Query

Merupakan query yang terletak dalam query yang lain.

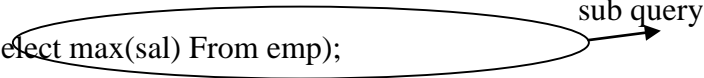
Contoh :

Akan ditampilkan nama pegawai yang mempunyai gaji paling besar :

Mysql> Select ename

☐ From emp

☐ Where sal = (select max(sal) From emp);



### Operator Exists

Merupakan jenis operator boolean, yang menghasilkan nilai benar (true) atau salah (false).

Operator Exists akan memberikan nilai benar (true) kalau sub query menghasilkan paling tidak sebuah baris / record.

Contoh 1 :

Akan ditampilkan daftar nama pegawai yang sudah mempunyai departemen :

Mysql> Select ename From emp

☐ Where Exists (select \* From dept Where deptno = emp.deptno);

*Keterangan :*

*Perintah pada sub query di atas berisi pengecekan nim peserta didik pada tabel krs*

Contoh 2 :

Akan ditampilkan daftar nama pegawai yang tidak mempunyai departemen :

Mysql> Select ename From emp

☐ Where not Exists (select \* From dept Where deptno = emp.deptno);

*Keterangan :*

*Perintah pada sub query di atas sama dengan contoh 1, hanya ditambahkan dengan Not sebelum Exists*

### Operator Any (Some)

Operator Any (some) hampir sama penggunaannya seperti Exists. Tetapi operator relasi yang digunakan biasanya selain = (sama dengan), hal tersebut disebabkan apabila operator relasi = yang digunakan, maka sebetulnya fungsi operator Any (some) sama seperti

operator IN, sehingga kondisi seperti itu tidak dianjurkan karena lebih mudah pemahamannya apabila menggunakan operator IN.

Contoh :

Akan menampilkan daftar nama pegawai dan gaji yang gajinya tidak paling sedikit :

```
Mysql> Select ename, sal
        □      From emp
        □      Where sal > any (select sal from emp);
```

### Operator All

Operator all digunakan untuk melakukan perbandingan dengan sub query. Kondisi dengan all menghasilkan nilai benar jika perbandingan menghasilkan benar untuk setiap nilai dalam sub query.

Contoh :

Akan menampilkan nama dan gaji pegawai yang gajinya lebih rendah daripada semua pegawai yang job-nya 'SALESMAN' :

```
Mysql> Select ename, sal
        □      From emp
        □      Where sal < all (select sal From emp Where job = 'SALESMAN');
```

Sub query (subselect) hanya dapat digunakan untuk MySQL versi 3.24, Sehingga perintah-perintah di atas tidak dapat dijalankan di MySQL ini (versi 3.23). Tetapi, dengan menggunakan option LEFT JOIN, masalah-masalah di atas dapat diselesaikan.

Contoh 1:

```
Mysql> select * from score
        □      where even_id in (select even_id from even where type = 'T');
```

apabila dituliskan dengan cara lain menjadi :

```
Mysql> select score.* from score, event
        □      where score.even_id = event.even_id and type = 'T';
```

Contoh 2:

```
Mysql> select ename from emp
        □      where sal = (select max(sal) from emp);
```



apabila dituliskan dengan cara lain menjadi :

MySQL> select max(sal) from emp;

```
mysql> select max(sal) from emp;
+-----+
| max(sal) |
+-----+
|      5000 |
+-----+
1 row in set (0.24 sec)
```

MySQL> select ename from emp where sal = 5000;

```
mysql> select ename from emp where sal=5000;
+-----+
| ename |
+-----+
| KING  |
+-----+
1 row in set (0.03 sec)
```

Contoh 3:

MySQL> select \* from student

□ where student\_id not in (select student\_id from absence);

apabila dituliskan dengan cara LEFT JOIN menjadi :

MySQL> select student.\*

□ from student left join absence on student.student\_id = absence.student\_id

□ where absence.student\_id is null;

## 5. Latihan Praktikum ke IX :

Untuk mengerjakan latihan dibawah ini, kita masih menggunakan table-table yang telah kita miliki yaitu table Mhs, mkuliah, dosen dan krs. Dari table-table tersebut tuliskan query-nya untuk menampilkan informasi-informasi dibawah ini : a. Tampilkan nama peserta didik yang tanggal lahirnya paling tua;

b. Tampilkan nama-nama peserta didik yang melakukan pengisian krs (nim-nya terdapat di table krs).

c. Tambahkan field berikut pada tabel krs :

Nama Field	Type Data	Keterangan
------------	-----------	------------

Kode_hari	Char(1)	Kode_hari : 1 = Senin, 2 = Selasa, 3 = Rabu, 4 = Kamis, 5 = Jumat
Kode_waktu	Char(1)	Kode_waktu : 1 = 08.00-10.00 2 = 10.00-12.00 3 = 12.00-14.00 4 = 14.00-16.00 5 = 16.00-18.00 6 = 18.00-20.00

Sehingga isi table krs menjadi :

```
'A21-101','A21.2001.00234',2,80,'1','1'
'A21-201','A21.2001.00214',4,75,'1','3'
'A21-201','A21.2001.00234',1,89,'3','1'
'A21-601','A21.2001.00214',3,90,'2','2'
'A21-501','A21.2001.00214',5,85,'2','3'
'A21-501','A21.2001.00234',5,77,'5','1'
'A21-101','A21.2001.00224',2,95,'4','3'
```

- d. Tampilkan jadwal kuliah untuk peserta didik yang mempunyai NIM = A21.2001.00214 dengan contoh bentuk tampilan adalah :

Mata pelajaran	Hari	Jam	Dosen
Bimbingan Karier	Selasa	12.00-14.00	Sumiati, SE., M.Si.
.....			

5. Tampilkan jadwal mengajar untuk seorang dosen yang mempunyai kode = 102 dengan contoh bentuk tampilan adalah :

Mata pelajaran	SKS	Hari	Jam
Pendidikan Pancasila	2	Selasa	08.00 – 10.00
.....			

6. Tampilkan jumlah sks kumulatif untuk setiap mahasiswa dengan contoh tampilan :

Nama Peserta didik	Jumlah_SKS_kum
--------------------	----------------

Agus Wahyu .....	10
---------------------	----

7. Tampilkan beban sks mengajar untuk setiap dosen dengan contoh tampilan :

Nama Dosen	Jumlah_SKS_kum
Sumiati, SE., M.Si. .....	6

8. Tampilkan daftar nama dosen yang mengajar jam 08.00-10.00 (apabila terdapat data double, maka ditampilkan sekali)
9. Tampilkan daftar nama peserta didik yang kuliah jam 08.00-10.00 (apabila terdapat data double, maka ditampilkan sekali)

## PRAKTIKUM KE – XI

- |                           |   |
|---------------------------|---|
| 1. Judul Materi           | : View & Indeks   |
| 2. Tujuan Inst.<br>Khusus | : Setelah mengikuti materi pembelajaran ini peserta didik dapat membuat view agar pengelolaan tabel-tabel yang rumit dapat lebih mudah dan membuat indeks sehingga mempercepat pengelolaan data |
| 3. Jumlah Pertemuan       | : 1 Kali  |
| 4. Teori Dasar            | :   |

### MENGENAL PANDANGAN (VIEW)

#### Pengertian Pandangan

Pandangan (view) adalah tabel semu karena secara fisik tidak ada.

Keuntungan view adalah :

- ✓ Query yang kompleks dapat di sembunyikan sehingga informasi dapat diakses dengan cepat

- ✓ Untuk mengatur data tertentu menjadi tersembunyi bagi pemakai
- ✓ Untuk melindungi data yang bersifat rahasia

### Cara menciptakan Pandangan

Untuk membuat pandangan / view digunakan perintah CREATE VIEW.

Contoh :

```
Mysql> Create View mhs_view as
      □      Select nim, nama, alamat From Mhs;
```

Keterangan :

Mhs\_view : nama view (pandangan)

### Memperoleh Informasi pada View

Setelah suatu view dibuat, maka view tersebut dapat diakses sebagaimana tabel.

Contoh :

```
Mysql> Select * From mhs_view;
```

### Membuat View dari BerepaTabel

Sebuah view juga dapat dibuat dari relasi beberapa tabel.

Contoh :

```
Mysql> Create View hasil_view as
      □      Select a.nim, a.nama, a.alamat      → From mhs a, mkuliah b, krs c
      □      Where  a.nim = c.nim and b.kode_kul = c.kode_kul
      □      and  b.nama_kul = 'Pancasila';
```

### Membuat View Pengelompokan

View yang di dasarkan pada pengelompokkan juga dapat dibuat.

Contoh :

```
Mysql> Create View klp_view as
      □      Select depart, count(*)
      □      From Pegawai Group by depart;
```

### Memberi nama lain terhadap kolom

Judul kolom / field yang akan ditampilkan dapat diubah namanya dengan cara meletakkan nama-nama judul kolom diantara nama view dan as.

Contoh :

```
Mysql> Create View klp_view
      (nama_depart, jumlah) as
      Select depart, count(*)
      From Pegawai Group by depart;
```

## MENGENAL INDEKS

### Pengertian Indeks

Indeks dapat dibayangkan sebagai indeks buku, sehingga melalui indeks buku tersebut dapat dicari letak item tertentu dalam buku dengan mudah. Keberadaan indeks dalam basis data antara lain adalah untuk mempercepat pencarian data berdasarkan kolom tertentu, misalnya diberikan perintah :

```
Mysql> Select * from pegawai where nip = '12345';
```

Jika nip tidak dijadikan sebagai indeks, pencarian data akan dilakukan terhadap seluruh tabel, sama seperti kalau akan mencari sesuatu dalam buku tetapi buku tersebut tidak dilengkapi dengan indeks. Namun sekiranya indeks yang berkaitan dengan nip ada, maka sistem akan menemukannya dengan cepat.

Beberapa hal yang sangat terbantu dengan adanya indeks adalah : a.

Proses penggabungan sejumlah tabel

b. Proses dengan ORDER BY

c. Proses fungsi agregat seperti MIN dan MAX

### Menciptakan Indeks

Untuk membuat file indeks digunakan perintah CREATE INDEX.

Misalnya terdapat sebuah tabel dengan perintah pembuatannya adalah sebagai berikut :

```
Mysql> Create Table Pelamar (nama char(20) not null, no_tes char(3) not null);
```

Maka kita dapat membuat indeks dari tabel tersebut, seperti :

```
Mysql> Create Index no_tes_idx on pelamar (no_tes);
```

Keterangan :

No\_tes\_idx adalah nama indeks yang dibuat

Pelamar adalah nama tabel yang diindeks

No\_tes yang berada di dalam tanda kurung menyatakan kolom / field / atribut yang digunakan untuk mengindeks.

### **Menciptakan Indeks yang Unik**

Untuk menciptakan indeks yang unik maka harus ditambahkan kata UNIQUE diantara create dan index.

Contoh :

```
Mysql> Create unique index no_tes_idx on pelamar (no_tes);
```

### **Menciptakan Indeks Untuk Beberapa Field**

Untuk menciptakan indeks beberapa field, maka perlu penyebutan nama-nama field yang dimaksud sebagai berikut :

```
Mysql> Create Table nilai (no_mhs Char(5) not null,  
→ kode_mkuliah char950 not null, nilai char);
```

Selanjutnya untuk menciptakan indeks berdasarkan tabel tersebut :

```
Mysql> Create unique index nilai)idx on nilai (no_mhs, kode_mkuliah);
```

5. Latihan Praktikum X :

1. Indekslah tabel mhs yang telah anda miliki berdasatrkan kunci field NIM.
2. Indeks tabel mkuliah yang telah anda miliki berdasatrkan kunci field KODE\_KUL
3. Indeks tabel krs yang telah anda miliki berdasatrkan kunci field KODE\_KUL dan NIM

## **PRAKTIKUM KE – XII**

1. Judul Materi : Manajemen User
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat mengaktifkan keamanan standar, membuat user baru dan mengatur hak akses terhadap data sehingga data lebih terjamin keamanannya.
3. Jumlah Pertemuan : 1 Kali
4. Teori Dasar :

### **MENGAKTIFKAN KEAMANAN STANDAR**

Saat pertama kali kita menggunakan MySQL setelah proses instalasi selesai, MySQL dapat digunakan oleh siapa saja tanpa harus menggunakan password dan nama user. Hal ini akan

sangat berbahaya karena pintu menjadi terbuka dan sangat mudah untuk dimasuki oleh orang-orang yang tidak bertanggung jawab. Maka hal pertama yang harus dilakukan setelah berhasil menginstal MySQL adalah mengaktifkan password untuk root dan menghapus anonym user. Anonym user adalah user tanpa identitas dan tanpa password. Masuklah ke dalam system sebagai root dan jalankan MySQL :

[system prompt] mysql

Kemudian masuklah ke database mysql (database default yang sudah otomatis tersedia adalah mysql dan test) :

```
mysql> use mysql;
Database changed
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| host            |
| tables_priv     |
| user            |
+-----+
5 rows in set (0.00 sec)
```

Tabel-tabel di atas merupakan table standar yang terdapat pada database mysql yang digunakan untuk mengatur user dan izin akses masing-masing user (user privileges). Oleh karena itu, table-tabel tersebut disebut juga grant tables. Sedangkan fungsi masing-masing table adalah :

Nama table	Fungsi	Tingkatan Akses
User	Berisi data user yang mendapatkan izin akses MySQL, asal koneksi dari mana saja, izin akses apa yang diberikan ke user tersebut	Global
Db	Mengatur database apa saja yang dapat diakses oleh seorang user, berikut jenis izin aksesnya	Database
Host	Mengatur asal host yang diperkenankan bagi seorang user untuk mengakses MySQL	Database
Tables_priv	Mengatur table apa saja yang dapat diakses oleh seorang user, berikut jenis izin aksesnya	Table
Columns_priv	Mengatur field apa saja yang dapat diakses oleh seorang user, berikut izin aksesnya	Field



### Memberi Password untuk root

Pemberian password untuk root dilakukan dengan perintah Update sebagai berikut :

```
Mysql> update user set password = password ('xxxxxxxxxxxxxx')  
→ where user = 'root';
```

Kemudian dilanjutkan dengan perintah FLUSH sebagai berikut :

```
Mysql> Flush Privileges;
```

Apabila diperiksa maka password telah berisi kode dari password yang telah dimasukkan.

### Membuat User Baru

Sekarang kita coba untuk membuat user baru. Untuk melakukan perintah ini kita harus login sebagai root.

Contoh :

Kiat akan membuat user baru dengan nama latihan yang dapat mengakses semua database yang ada di dalam server. Dan user baru ini dapat memberikan izin akses (grant) kepada user-user lainnya :

```
Mysql> Grant all privileges on *.* to Latihan → identified by  
'nama_password' with grant option;
```

## **MENGATUR HAK AKSES**

### Untuk Apa Hak Akses Perlu Diatur

Mengontrol basis data merupakan hal yang sangat perlu diperhatikan dalam rangka menjaga keamanan dari orang-orang yang tidak berhak. Data yang bersifat rahasia tidak dapat disembunyikan dan bahkan kemungkinan data menjadi sampah karena setiap orang bisa melakukan apa saja sesuka hati.

### Cara Mengatur Hak Akses

SQL menyediakan perintah Grant untuk mengatur hak akses seseorang terhadap suatu tabel. Bentuk perintahnya adalah :

Grant hak\_akses On nama\_tabel\_atau\_view To pemakai;

Keterangan :

- Hak akses merupakan hak akses yang akan diberikan (misalnya Select atau insert atau gabungannya). Jika hak akses lebih dari satu, antar hak akses perlu dipisahkan dengan koma.
- Nama\_tabel\_atau\_view menyatakan tabel atau pandangan (view) yang hak aksesnya akan diatur
- Pemakai adalah nama pemakai yang telah didaftarkan pada sistem basis data. Sejumlah pemakai dapat disebutkan dengan antar pemakai dipisah dengan koma.

Macam-macam hak akses

Hak Akses	Keterangan
Select	Hak akses yang memungkinkan operasi Select
Insert	Hak akses yang memungkinkan operasi Insert
Update	Hak akses yang memungkinkan operasi Update
Delete	Hak akses yang memungkinkan operasi Delete
Index	Hak akses yang memungkinkan operasi membuat indeks
Drop	Hak akses yang memungkinkan operasi Drop

Contoh 1 :

```
MySQL> Grant Select on emp To Terra;
```

Contoh 2 :

```
MySQL> Grant Select, Insert on emp To Terra;
```

### **Membatasi Hak Akses pada Field tertentu**

SQL memungkinkan pemberian hak akses hanya terbatas pada kolom tertentu.

Contoh :

```
MySQL> Grant Select, Update (empno, ename, job) on emp to Dewi;
```

### **Memberikan Segala Hak**

Jika semua hak akan diberikan kepada seorang pemakai, maka dapat dituliskan ALL PRIVILEGES pada bagian hak akses.

Contoh :

Mysql> Grant All Privileges On emp To Dewi;

## **PRAKTIKUM KE – XIII**

1. Judul Materi : Sistem Administrasi MySQL
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat melaksanakan tugas-tugas administrator yaitu backup dan pemeriksaan.
3. Jumlah Pertemuan : 1 Kali
4. Teori Dasar :

## **SISTEM ADMINISTRASI MYSQL**

Backup Database dengan mysqldump

Mysqldump adalah suatu program Bantu yang digunakan untuk membuat backup sebuah database, beberapa database, beberapa table sekaligus bahkan dapat juga untuk memindahkan data ke server yang lain. Cara kerja mysqldump adalah menuliskan kembali isi dari suatu database tertentu dalam bentuk perintah-perintah MySQL kedalam sebuah file teks. Sehingga dapat digunakan juga pada server database lain yang tidak sama system operasinya dan bahkan tidak sama program databasenya.

Bentuk umum perintah mysqldump adalah :

[system prompt] mysqldump [pilihan] nama\_ss [nama\_table] atau

[system prompt] mysqldump [pilihan] --nama\_database [pilihan] DB1 [DB2 DB3 ...] atau

[system prompt] mysqldump [pilihan] --all -databases [pilihan]

Pilihan yang dapat disertakan banyak, di bawah ini dijelaskan beberapa pilihan yang sering digunakan :

✓ -?, -- help

Untuk menampilkan layar bantuan yang berisi pilihan yang tersedia pada mysqldump

✓ -A, --all-databases

Akan membackup seluruh databases yang ada didalam server tersebut.

✓ -B, --databases

Akan membackup beberapa databases tertentu.

✓ -d, --no-data

Tidak menyertakan data-data yang ada didalam table. Hanya struktur tablenya saja yang dibackup.

✓ -l, --lock -nama\_table

Melakukan penguncian pada seluruh table ebelum melakukan proses pem-backup-an.

✓ Dan lain-lain

Backup Database dengan mysqlhotcopy

Adalah sebuah program Bantu yang dibuat dengan scrip PERL untuk melakukan penyalinan seluruh file-file data. Program mysqlhotcopy ini tidak tersedia pada MySQL versi Windows. Untuk dapat menjalankan perintah ini kita harus login sebagai root atau setidaknya memiliki izin akses (privileges).

Bentuk umum perintah ini adalah :

mysqlhotcopy [pilihan] nama\_databases /nama\_direk\_tuj;

Contoh :

Untuk membackup database latihan kemudian di simpan di direktori /home/hary/.

[system prompt] mysqlhotcopy Latihan /home/hary/ -- user='Joni' password = 'xxxxx';

#### Pemeriksaan dan Perbaikan Struktur Database

Beban kerja yang berat, sibuknya permintaan query maupun ketidakstabilan kinerja listrik dan perangkat dapat mengakibatkan kerusakan pada database. Untungnya kita dapat memeriksa kondisi fisik database tersebut dengan menggunakan beberapa alat Bantu yang telah disediakan oleh MySQL;

#### Pemeriksaan dengan MYSQLCHECK

mysqlcheck adalah sebuah program Bantu (utility) yang disediakan oleh MySQL sejak versi 3.23.38. Fungsi mysqlcheck meliputi memeriksa (check), memperbaiki (repair), menganalisis (ANALyze), dan mengoptimasi (optimize) kondisi fisik suatu database dari table.

#### Pemeriksaan dengan MYISAMCHK dan ISAMCHK

Adalah sebuah program Bantu untuk mengetahui kondisi table, memeriksa dan memperbaiki table yang rusak, maupun mengoptimasi table dalam format MYISAM. Sedangkan untuk formattabel ISAM program bantu adalah isamchk. Format MyISAM merupakan perkembangan dari format ISAM dan format MySQL menjadi standar format table pada MySQL.

## **PRAKTIKUM KE – XIV**

1. Judul Materi : Studi Kasus
2. Tujuan Inst. Khusus : Setelah mengikuti materi pembelajaran ini peserta didik dapat melakukan sinergi terhadap materi-materi sebelumnya untuk menyelesaikan masalah pengelolaan database untuk aplikasi yang sesungguhnya.
3. Jumlah Pertemuan : 1 Kali
4. Teori Dasar : -
5. Tugas :

Pilihlah salah satu topik dari studi kasus yang telah disediakan di halaman lampiran. Buatlah seluruh table dari masing-masing studi kasus tersebut kemudian data untuk masing-masing table dimasukkan. Apabila semua table dan isinya sudah siap, buatlah query untuk menampilkan informasi-informasi yang diinginkan.

## **PRAKTIKUM KE – XIV**

### **TUGAS MANDIRI**

Setiap peserta didik diharapkan dapat mengaplikasikan materi-materi MySQL dalam pengelolaan sebuah basis data . Basis data yang ada di dalam suatu system tidak terbentuk dengan begitu saja, tetapi melalui proses perancangan database yang cukup panjang. Oleh karena itu, untuk melatih peserta didik membangun / merancang sebuah basis data dan mengaplikasikannya pada sebuah perangkat lunak pengolah basis data (DBMS) seperti

MySQL, maka setiap peserta didik diwajibkan untuk merancang sebuah basis data mengenai suatu kasus dan mengaplikasikannya dengan menggunakan MySQL. Tugas ini merupakan kompilasi dari materi kuliah System Basis Data dan Praktikum Basis Data. Adapun ketentuan penyusunan karya ilmiah ini adalah sebagai berikut :

✓ Sistematika Penulisan

Laporan karya ilmiah yang disusun, minimal mencakup hal-hal sebagai berikut : a.

Latar belakang masalah

- b. Deskripsi prosedur / urutan kerja / system yang sedang diteliti
- c. Perancangan database dengan menggunakan ERD, Normalisasi ataupun keduanya sehingga didapatkan table-tabel yang normal.
- d. Diagram Hubungan antar Tabel
- e. Struktur masing-masing tabel
- f. Implementasi table-tabel didalam MySQL.
- g. Pemasukan data masing-masing table tersebut
- h. Query-query yang dapat menghasilkan informasi-informasi yang kompleks dan bermanfaat
- i. Kesimpulan
- j. Daftar Pustaka

✓ Ketentuan lain

- a. Setiap peserta didik membuat satu laporan karya ilmiah
- b. Topik dan obyek penelitian tidak boleh ada yang sama antara satu peserta didik dengan peserta didik lainnya.
- c. Laporan karya ilmiah dikumpulkan paling lambat 1 hari sebelumnya Ujian Akhir Semester matakuliah ini dilaksanakan.
- d. Laporan karya ilmiah ini digunakan sebagai syarat kelulusan mata pelajaran ini.



## **DAFTAR PUSTAKA**

Abdul Kadir, Ir (1999), Konsep dan Tuntuan Praktis Basis Data, Andi Offset, Yogyakarta

Fatansyah, Ir, Basis Data

Arbie (2004), Manajemen Database dengan MySQL, Andi Offset, Yogyakarta

Paul DuBois, MySQL, Indianapolis, Indiana

Waljiyanto (200), Sistem Basis Data, Analisis dan Pemodelan Data, J&J Learning,  
Yogyakarta

[WWW.MYSQL.COM](http://WWW.MYSQL.COM), MySQL Manual Reference

## Lampiran I

### 1. Studi Kasus : Database Kesiswaan

Database kesiswaan digunakan untuk menyimpan data-data pada system pengolahan data siswa yang terdiri dari table jenis\_tes, pelajaran, jadwal\_tes, siswa, absensi dan nilai dengan struktur table dan datanya sebagai berikut :

Struktur Tabel :

Tabel Jenis\_tes

Nama Field	Type Data	Lebar Data	Keterangan
Jenis_tes_id	Karakter	2	Primary key
Nama	Karakter	20	Tidak Boleh Kosong

Tabel Pelajaran

Nama Field	Type Data	Lebar Data	Keterangan
Pelajaran_id	Karakter	3	Primary key, Auto_increment
Nama_pel	Karakter	40	Tidak boleh kosong

Tabel Jadwal\_tes

Nama Field	Type Data	Lebar Data	Keterangan
Jadwal_id	Numerik	4	Primary key, Auto_increment
Tgl_tes	Tanggal		Default '0000-00-00'
Jam_mulai	Jam		
Jam_selesai	Jam		
Ruang	Karakter		
Jenis_tes_id	Karakter	2	
Pelajaran_id	Karakter	3	

Tabel Siswa

Nama Field	Type Data	Lebar Data	Keterangan
Siswa_id	Karakter	10	Primary key
Nama_siswa	Karakter	50	Tidak boleh kosong

Kelamin	Karakter	1	Tidak boleh kosong, Default 'L'
Tempat_lahir	Karakter	50	Tidak boleh kosong
Tanggal_lahir	Tanggal		Tidak boleh kosong, default '0000-00-00'
Nama_ortu	Karakter	50	
Alamat1	Karakter	40	
Alamat2	Karakter	40	
Kota	Karakter	40	
Kode_pos	Karakter	5	
Telp_rumah	Karakter	25	

Tabel Absensi

Nama Field	Type Data	Lebar Data	Keterangan
Siswa_id	Karakter	10	Primary key
Tanggal	Tanggal		Primary key, Tidak boleh kosong, Default '0000-00-00'
Jenis_absen	Karakter	5	Diisi dengan : 'Alpa', 'Ijin' atau 'Sakit' Default 'Alpa'

Tabel Nilai

Nama Field	Type Data	Lebar Data	Keterangan
Siswa_id	Karakter	10	Primary key
Jadwal_tes	Numerik	4	Primary key
Nilai	Numerik		

Isi table-table tersebut adalah :

Tabel Pelajaran

Pelajaran_id	Nama_pel
MTK	Matematika
FIS	Fisika
KMA	Kimia
BIO	Biologi
ING	Bahasa inggris
AGM	Agama

Table Jenis\_tes

Jenis_tes_id	nama
HR	Tes Harian

TS	Tes Tengah Semester
AS	Tes Akhir Semester
PB	Tes Perbaikan
SL	Tes Susulan

Tabel Absensi

Siswa_id	Tanggal	Jenis_absen
1999010001	2002-01-10	Alpa
1999010030	2002-01-10	Alpa
1999010017	2002-02-10	Alpa
1999010019	2002-02-22	Alpa
1999010002	2002-03-19	Alpa
1999010012	2002-04-11	Alpa
1999010016	2002-05-05	Alpa
1999010011	2002-01-11	Ijin
1999010005	2002-02-13	Ijin
1999010015	2002-02-13	Ijin
1999010012	2002-02-17	Ijin
1999010029	2002-03-10	Ijin
1999010021	2002-03-11	Ijin
1999010004	2002-03-31	Ijin
1999010009	2002-04-01	Ijin
1999010001	2002-04-07	Ijin
1999010008	2002-04-09	Ijin
1999010009	2002-05-04	Ijin
1999010004	2002-02-01	Sakit
1999010006	2002-04-23	Sakit
1999010017	2002-04-17	Sakit
1999010019	2002-01-20	Sakit
1999010022	2002-02-24	Sakit
1999010022	2002-02-25	Sakit
1999010026	2002-01-21	Sakit

Tabel Siswa

Siswa_id	Nama	Kelamin	Tempat_lahir	Tgl_lahir	
1999010001	Ahmad Wijaya	L	Jakarta	1986-02-04	
1999010002	Andriansyah	L	Jakarta	1986-01-11	
1999010003	Bagus S	L	Bogor	1986-06-12	
1999010004	Bambang Laksana	L	Depok	1986-05-04	
1999010005	Cintiawati	P	Jakarta	1986-04-17	
1999010006	Charlie Soen	L	Jakarta	1986-07-22	
1999010007	Ayu W	P	Jakarta	1986-08-19	

1999010008	Deni P	L	Jakarta	1986-08-04	
1999010009	Dion Paprika	L	Jakarta	1986-11-26	
1999010010	Dewi Bestari	P	Semarang	1986-06-06	
1999010011	Dewi Anggraini	P	Belawan	1986-09-04	
1999010012	Dudung S	L	Jakarta	1986-02-22	
1999010013	Endang Subakti	L	Jakarta	1986-04-26	
1999010014	Endang Wartini	P	Jakarta	1986-07-12	
1999010015	Rama	L	Jakarta	1986-10-21	
1999010016	Friski Catty	P	Washington	1986-02-04	
1999010017	Gunawan	L	Jakarta	1986-03-03	
1999010018	Gina G	P	Jakarta	1986-05-04	
1999010019	Hendro K	L	Jakarta	1986-06-04	
1999010020	Herman K	L	Bandung	1986-11-05	
1999010021	Hendriana Jati	L	Jakarta	1986-03-06	
1999010022	Irmandis	L	Jakarta	1986-02-09	
1999010023	Julian	L	Tegal	1986-07-11	
1999010024	Suroso	L	Pemalang	1986-06-14	
1999010025	Lusi Wijaya	L	Jakarta	1986-10-13	
1999010026	Loe Kam Sie	P	Jakarta	1986-05-04	
1999010027	Loe Sia Pa	L	Medan	1986-12-18	
1999010028	Maman S	L	Cimahi	1986-09-04	
1999010029	Momon S	L	Bandung	1986-11-23	
1999010030	Nunung	P	Jakarta	1986-02-25	

Tabel Jadwal\_tes

Jadwal_id	Tgl_tes	Jam_mulai	Jam_selesai	Ruang	Jenis_tes-id	Pelajaran_id
1	2002-02-01	08:00:00	09:40:00	R.201	HR	MTK
2	2002-02-01	09:40:00	12:10:00	R.203	HR	FIS
3	2002-02-02	09:00:00	10:40:00	R.205	HR	KMA
4	2002-02-03	09:00:00	10:40:00	R.201	HR	BIO
5	2002-02-04	08:00:00	09:40:00	R.203	HR	ING

6	2002-02-05	09:00:00	10:40:00	R.201	HR	AGM
7	2002-02-15	08:00:00	09:40:00	R.201	TS	MTK
8	2002-02-15	10:30:00	12:10:00	R.201	TS	FIS
9	2002-02-16	08:00:00	10:40:00	R.203	TS	KMA
10	2002-02-16	10:30:00	12:10:00	R.203	TS	BIO
11	2002-02-17	08:00:00	09:40:00	R.201	TS	ING
12	2002-02-17	10:30:00	12:10:00	R.201	TS	AGM
13	2002-05-01	08:00:00	09:40:00	R.205	AS	MTK
14	2002-05-01	09:40:00	12:10:00	R.205	AS	FIS
15	2002-05-03	08:00:00	10:40:00	R.201	AS	KMA
16	2002-05-03	12:10:00	12:10:00	R.201	AS	BIO
17	2002-05-05	08:00:00	09:40:00	R.203	AS	ING
18	2002-05-05	09:40:00	12:10:00	R.203	AS	AGM
19	2002-02-15	08:00:00	09:40:00	R.203	PB	MTK
20	2002-02-15	09:40:00	12:10:00	R.203	PB	FIS
21	2002-02-16	09:00:00	10:40:00	R.201	PB	KMA
22	2002-02-16	09:00:00	10:40:00	R.201	PB	BIO
23	2002-02-17	08:00:00	09:40:00	R.203	PB	ING
24	2002-02-17	09:00:00	10:40:00	R.203	PB	AGM

Table Nilai

Siswa_id	Jadwal_id	Nilai
----------	-----------	-------

1999010001	1	78
1999010002	1	88
1999010003	1	85
.....		
1999010001	2	88
1999010002	2	87
1999010003	2	67
.....		
1999010001	5	78
1999010002	5	75
1999010003	5	65
.....		

Dari table-table diatas, tuliskan query untuk menghasilkan informasi-informasi berikut :

1. a. Tampilkan data absensi untuk masing-masing siswa dengan output :

Nama Siswa	Jumlah Absensi
.....	
.....	

- b. Tampilkan pula data siswa yang tidak pernah absen

Nama Siswa	Junis Absen
.....	
.....	

2. Berapakah nilai maksimum yang diperoleh siswa pada tes harian matematika

Nilai Maksimum
.....

3. tampilkan nama siswa yang mendapatkan nilai maksimum pada tes harian matematika

Siswa_id	Nama Siswa	Jenis_tes	Pelajaran	Nilai
----------	------------	-----------	-----------	-------



.....				
-------	--	--	--	--

4. Tampilkan nama-nama siswa yang termasuk lima besar dalam mendapatkan nilai tes harian untuk pelajaran matematika

Siswa_id	Nama Siswa	Jenis_tes	Pelajaran	Nilai
.....				

5. Tampilkan nama-nama siswa yang mendapatkan nilai di atas nilai rata-rata

Siswa_id	Nama Siswa	Jenis_tes	Pelajaran	Nilai
.....				

6. Berapakah nilai maksimum yang diperoleh Andriansyah dengan nomor siswa\_id = '1999010002' untuk seluruh tes harian yang dia ikuti

Nilai Maksimum
.....

## 2. Studi Kasus : Database Koleksi Buku

Informasi awal tentang buku biasanya sudah tertera pada sampul buku atau di halaman awal pada buku tersebut. Kita tinggal memilih data mana yang diperlukan. Biasanya, data yang paling sering digunakan adalah judul buku, subjudul buku, nomor seri buku (ISBN), nama penulis, nama penerbit, tahun cetakan, edisi cetakan, jumlah halaman, jenis sampul buku, jenis kertas, harga beli buku, tanggal pembelian dan sebaianya. Dengan melalui tahap perancangan database, maka didapatkan table-table yang digunakan pada database koleksi buku sebagai berikut :

Struktur Tabel :

Tabel Buku

```
mysql> desc buku;
```

Field	Type	Null	Key	Default	Extra
buku_id	int(5) unsigned		PRI	NULL	auto_increment
judul	varchar(50)				
tema_id	char(3)	YES		NULL	
bahasa	varchar(15)	YES		NULL	
isbn	varchar(20)	YES		NULL	
tahun_cetak	year(4)			0000	
harga_beli	int(11)	YES		0	
tanggal_beli	date			0000-00-00	
jumlah_hal	int(4) unsigned	YES		0	
jenis_sampul	enum('HC','SC')			HC	
jenis_kertas	enum('B','H')			B	
sinopsis	varchar(255)	YES		NULL	
keterangan	varchar(255)	YES		NULL	

```
13 rows in set (0.01 sec)
```

### Tabel Negara

```
mysql> desc negara;
```

Field	Type	Null	Key	Default	Extra
negara_id	char(2)		PRI		
nama_neg_pembuat	varchar(30)				

```
2 rows in set (0.05 sec)
```

### Tabel Tema

```
mysql> desc tema;
```

Field	Type	Null	Key	Default	Extra
tema_id	char(3)		PRI		
nama	varchar(20)				

```
2 rows in set (0.01 sec)
```

### Tabel Penulis

```
mysql> desc penulis;
```

Field	Type	Null	Key	Default	Extra
penulis_id	int(4) unsigned		PRI	NULL	auto_increment
nama	varchar(50)				
kelamin	enum('P','L')			P	
alamat	varchar(50)	YES		NULL	
kota	varchar(40)	YES		NULL	
kodepos	varchar(5)	YES		NULL	
propinsi	varchar(20)	YES		NULL	
negara_id	char(2)	YES		NULL	
website	varchar(25)	YES		NULL	
email	varchar(25)	YES		NULL	
telepon	varchar(20)	YES		NULL	
keterangan	varchar(255)	YES		NULL	

```
12 rows in set (0.00 sec)
```

## Tabel Penerbit

```
mysql> desc penerbit;
```

Field	Type	Null	Key	Default	Extra
penerbit_id	int(4)		PRI	NULL	auto_increment
nama	varchar(50)	YES		NULL	
alamat	varchar(50)	YES		NULL	
kota	varchar(40)	YES		NULL	
kodepos	varchar(5)	YES		NULL	
negara_id	char(2)	YES		NULL	
website	varchar(25)	YES		NULL	
email	varchar(25)	YES		NULL	
telepon	varchar(20)	YES		NULL	
keterangan	varchar(255)	YES		NULL	

10 rows in set (0.00 sec)

## Tabel Buku\_penerbit

```
mysql> desc buku_penerbit;
```

Field	Type	Null	Key	Default	Extra
buku_id	int(5) unsigned		PRI	0	
penerbit_id	int(4) unsigned		PRI	0	

2 rows in set (0.02 sec)

## Tabel Buku\_penulis

```
mysql> desc buku_penulis;
```

Field	Type	Null	Key	Default	Extra
buku_id	int(5) unsigned		PRI	0	
penulis_id	int(4) unsigned		PRI	0	

2 rows in set (0.01 sec)

Isi table-table di atas adalah :

## Tabel Buku

```
mysql> select buku_id,judul,tema_id,bahasa from buku;
```

buku_id	judul	tema_id	bahasa
1	Butir-butir Waktu	AGM	Indonesia
2	Sabar	AGM	Indonesia
3	Berkah	AGM	Indonesia
4	Langit Runtuh	MTR	Indonesia
5	Ceritakan Mimpi	FKS	Indonesia
6	Ikhlās	AGM	Indonesia
7	Menghadapi Kesulitan	PUM	Indonesia
8	Malaikat Keadilan	MJL	Indonesia
9	Mencari Ridho Ilahi	AGM	Indonesia
10	Bersih Pangkal Sehat Jasmani dan Rohani	KES	Indonesia
11	The Testament	KEU	Inggris
12	The Brethen	PUM	Inggris
13	Menggapai Rahmat Tuhan	PDR	Indonesia
14	Mengendalikan Hawa Nafsu	FKS	Indonesia
15	Manajemen Rumah Tangga	MAN	Indonesia
16	Ikhtiar-Pikir-Zikir	AGM	Indonesia
17	Adab Bergaul dalam Bertetangga	PUM	Indonesia
18	Rich Dad Poor Dad	MAN	Inggris
19	Rich Dads Cashflow	MAN	Inggris
20	The Partner	MJL	Inggris
21	Beginning Visual Basic 6.0	KOM	Indonesia
22	The Firm	MAN	Inggris
23	Pelican Brief	LLN	Inggris

```
23 rows in set (0.02 sec)
```

Tabel Buku\_penerbit

```
mysql> select * from buku_penerbit;
```

buku_id	penerbit_id
1	19
2	19
3	19
4	19
5	19
6	1
7	2
8	3
9	3
10	4
11	4
12	4
13	25
14	25
15	24
16	23
17	20
18	15
19	9
20	9
21	18
22	26
23	18
24	17

```
24 rows in set (0.00 sec)
```

Tabel Buku\_penulis

```
mysql> select * from buku_penulis;
```

buku_id	penulis_id
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	2
19	2
20	2
21	3
22	3
23	4
24	4
25	4
26	5

```
26 rows in set (0.00 sec)
```

Tabel Negara

```
mysql> select * from negara;
```

negara_id	nama_neg_pembuat
ID	Indonesia
JP	Jepang
SG	Singapura
MY	Malaysia
KR	Korea
IG	Inggris
US	Amerika Serikat

```
7 rows in set (0.01 sec)
```

Tabel Penerbit

```
mysql> select penerbit_id,nama,kota,negara_id from penerbit;
```

penerbit_id	nama	kota	negara_id
1	Gramedia Pustaka Utama, PT.	Jakarta	ID
2	Pustaka Sinar Harapan	Jakarta	ID
3	Indira	Jakarta	ID
4	Kepustakaan Populer Gramedia	Jakarta	ID
5	Wrox Press Inc	Illinois	US
6	Sams Publishing	Indiana	US
7	Elex Media Komputindo, PT.	Jakarta	ID
8	Computer Step	NULL	IG
9	John Wiley	New York	US
10	Osborne	NULL	US
11	Addision Wesley	Boston	US
12	No Starch Press	San Fransisco	US
13	New Riders	Indiana	US
14	Jossey Bass	New York	US
15	Lowell House	NULL	US
16	Contemporary Books	Chicago	US
17	NTC	NULL	US
18	McGraw Hill	New York	US
19	Mizan	Bandung	ID
20	Gema Insani Press	Bandung	ID
21	Remaja Rosdakarya	Bandung	ID
22	Truedee Books	Jakarta	ID
23	Penerbit Andi	Yogyakarta	ID
24	Bhuana Ilmu Populer	Jakarta	ID
25	Salemba Empat	Jakarta	ID
26	Sybex	California	US

```
26 rows in set (0.00 sec)
```

Tabel Penulis

```
mysql> select penulis_id,nama,kelamin,negara_id from penulis;
```

penulis_id	nama	kelamin	negara_id
1	Abdullah Gymnastiar	L	ID
2	Robert T. Kiyosaki	L	JP
3	John Connel	L	MY
4	John Grisham	L	SG
5	Sharon L. Lechter	P	US

```
5 rows in set (0.02 sec)
```

Tabel Tema

```
mysql> select * from tema;
```

tema_id	nama
AGM	Agama
AKU	Akuntansi
BHS	Bahasa
BIO	Biografi
FKS	Fiksi Ilmiah
HOB	Hobi
HKM	Hukum
KES	Kesehatan
KEU	Keuangan
KMK	Komik
KOM	Komputer
KRI	Kriminalitas
LLN	Lain-lain
MJL	Majalah
MAN	Manajemen
MRK	Marketing
MSK	Memasak
MTR	Misteri
MUS	Musik
NRM	Novel Remaja
NDW	Novel Dewasa
OLR	Olah Raga
PPM	Peraturan Pemerintah
PJK	Perpajakan
PDR	Pengembangan Diri
PUM	Pengetahuan Umum
PTN	Pertanian
PSI	Psikologi
REF	Referensi
SEK	Seksologi

```
30 rows in set (0.02 sec)
```

Dari table-tabel diatas, tuliskan query untuk menghasilkan informasi-informasi berikut :

1. Menampilkan jumlah terbitan dari masing-masing penerbit dengan tampilanurut berdasarkan nama penerbit secara descending.

nama	Jumlah_buku
Sybex	1
Salemba Empat	2
Pustaka Sinar Harapan	1
Penerbit Andi	1
NTC	1
Mizan	5
McGraw Hill	2
Lowell House	1
Kepustakaan Popular Gramedia	3
John Wiley	2
Indira	2
Gramedia Pustaka Utama, PT.	1
Gema Insani Press	1
Bhuana Ilmu Popular	1

```
14 rows in set (0.04 sec)
```

2. Menampilkan nama-nama penulis yang bukunya paling banyak dikoleksi (minimal 3 judul buku) dengan tampilanurut berdasarkan jumlah buku secara descending.

nama	Jumlah_buku
Abdullah Gymnastiar	17
Robert T. Kiyosaki	3

2 rows in set (0.00 sec)

3. Menampilkan judul-judul buku, nama penerbit dan asal negara dengan tampilanurut berdasarkan negara secara ascending.

judul	Nama Penerbit	Negara
Beginning Visual Basic 6.0	McGraw Hill	Amerika Serikat
Rich Dad Poor Dad	Lowell House	Amerika Serikat
Pelican Brief	McGraw Hill	Amerika Serikat
The Partner	John Wiley	Amerika Serikat
The Firm	Sybex	Amerika Serikat
Rich Dads Cashflow	John Wiley	Amerika Serikat
Bersih Pangkal Sehat Jasmani dan Rohani	Kepustakaan Populer Gramedia	Indonesia
Menghadapi Kesulitan	Pustaka Sinar Harapan	Indonesia
Langit Runtuh	Mizan	Indonesia
Manajemen Rumah Tangga	Bhuana Ilmu Populer	Indonesia
Butir-butir Waktu	Mizan	Indonesia
The Brethen	Kepustakaan Populer Gramedia	Indonesia
Mencari Ridho Ilahi	Indira	Indonesia
Ikhlash	Gramedia Pustaka Utama, PT.	Indonesia
Adab Bergaul dalam Bertetangga	Gema Insani Press	Indonesia
Berkah	Mizan	Indonesia
Mengendalikan Hawa Nafsu	Salemba Empat	Indonesia
The Testament	Kepustakaan Populer Gramedia	Indonesia
Malaikat Keadilan	Indira	Indonesia
Ceritakan Mimpi	Mizan	Indonesia
Ikhtiar-Pikir-Zikir	Penerbit Andi	Indonesia
Sabar	Mizan	Indonesia
Menggapai Rahmat Tuhan	Salemba Empat	Indonesia

23 rows in set (0.00 sec)

4. Menampilkan judul-judul buku yang ditulis oleh penulis dengan nama awalnya adalah John.

nama	judul	bahasa
John Connel	Beginning Visual Basic 6.0	Indonesia
John Connel	The Firm	Inggris
John Grisham	Pelican Brief	Inggris

3 rows in set (0.02 sec)

5. Menampilkan nama-nama asal negara penerbit yang belum menerbitkan buku (jumlah buku terbitannya adalah 0), ditampilkanurut berdasarkan nama negara secara ascending



```

+-----+
| nama_neg_pembuat |
+-----+
| Jepang           |
| Korea            |
| Malaysia         |
| Singapura        |
+-----+
4 rows in set (0.00 sec)

```

6. Menampilkan nama negara dan nama penerbit (nama negara yang belum memiliki penerbit juga ditampilkan).

```

+-----+-----+
| nama_neg_pembuat | nama_pernerbit |
+-----+-----+
| Indonesia        | Gramedia Pustaka Utama, PT. |
| Indonesia        | Pustaka Sinar Harapan       |
| Indonesia        | Indira                      |
| Indonesia        | Kepustakaan Popular Gramedia |
| Indonesia        | Elex Media Komputindo, PT.   |
| Indonesia        | Mizan                       |
| Indonesia        | Gema Insani Press           |
| Indonesia        | Remaja Rosdakarya           |
| Indonesia        | Truedee Books               |
| Indonesia        | Penerbit Andi               |
| Indonesia        | Bhuana Ilmu Populer         |
| Indonesia        | Salemba Empat               |
| Jepang           | NULL                        |
| Singapura        | NULL                        |
| Malaysia         | NULL                        |
| Korea            | NULL                        |
| Inggris          | Computer Step               |
| Amerika Serikat  | Wrox Press Inc              |
| Amerika Serikat  | Sams Publishing              |
| Amerika Serikat  | John Wiley                  |
| Amerika Serikat  | Osborne                     |
| Amerika Serikat  | Addison Wesley              |
| Amerika Serikat  | No Starch Press             |
| Amerika Serikat  | New Riders                  |
| Amerika Serikat  | Jossey Bass                 |
| Amerika Serikat  | Lowell House                |
| Amerika Serikat  | Contemporary Books          |
| Amerika Serikat  | NTC                         |
| Amerika Serikat  | McGraw Hill                 |
| Amerika Serikat  | Sybex                       |
+-----+-----+
30 rows in set (0.00 sec)

```



## Lampiran II

### Exam Contents – MySQL Core Certification

*As of February 1<sup>st</sup>, 2005, questions on the certification exam will pertain to MySQL 4.1*

- MySQL and MySQL AB (5%)
  - a. The difference between MySQL and MySQL AB
  - b. How MySQL AB Operates
  - c. MySQL Core Values
  - d. MySQL Dual Licensing
  - e. Organization and structure of the MySQL Reference Manual
  - f. The MySQL Mailing List
- MySQL Software (10%)
  - a. Major program components used in MySQL
  - b. Major operating system families supported by MySQL
  - c. Differences between major MySQL distributions
  - d. Available MySQL client interfaces
- Using MySQL Client Programs (10%)
  - a. Invoking command-line client programs
  - b. Specifying commend-line options
  - c. The MySQL client
    - ✓ Using MySQL interactively
    - ✓ Using script files with MySQL
    - ✓ MySQL client commands and SQL statements
    - ✓ Using the `–safe—updates` option
  - d. Using `mysqlimport`
  - e. Using `mysqldump` and reloading the dump
  - f. Checking tables with `mysqlcehck` and `myisamchk`
  - g. Using MySQL Connector / ODBC and MySQL Connector
- Data Definition Language (20%)
  - a. General database and table properties

- b. Storage engines and table types
- c. Limits on number and size of database components
- d. Identifier syntax
- e. Create Database, Drop Database
- f. Create Table, Alter Table, Drop Table
- g. Create Index, Drop Index, Specifying indexes at table-creation time
- h. Creating and using primary keys
- i. Column types
- j. Using Auto\_increment
- k. String and number formats
- l. Using Show and describe to review table structures
- Select Statement (15%)
  - a. Selecting which columns to display
  - b. Restricting a selection using where
  - c. Using order by to sort query results
  - d. Limiting a selection using limit
  - e. Aggregate functions, group by and having
  - f. Using distinct to eliminate duplicates
  - g. Concatenating Select results with union
- Basic SQL (10%)
  - a. Using SQL expressions and functions
  - b. Using like for pattern matching
  - c. Using in () to test membership
  - d. Case sensitivity in string comparisons
  - e. Case sensitivity in database, table, column and function names
  - f. Using reserved words as identifiers
  - g. Null values in select statements
  - h. Comments in SQL statements
- Update Statements (10%)
  - a. Insert and replace
  - b. Update

- c. Delete and Truncate
- d. Handling duplicate key values
- e. Using order by and limit with update and delete statements
- Join (15%)
  - a. Writing inner joins using inner join and the comma (',') operator
  - b. Writing outer joins using left join and right join
  - c. Converting subqueries to inner and outer joins
  - d. Resolving name clashes using qualifiers and aliases
  - e. Multiple-table update and delete statements
- Importing and Exporting Data (5%)
  - a. Load data infile
  - b. Using files on the server and the client host
  - c. Limiting the columns and rows being imported
  - d. Selected into outfile
  - e. Privileges needed for load data infile and select into outfile

