



Pendahuluan



STANDAR KOMPETENSI :

Mahasiswa mampu memahami, merancang dan membuat basisdata menggunakan MySQL dan Aplikasi Basisdata menggunakan Visual Delphi.



Catatan :

Diharapkan hasil akhir setelah mengikuti perkuliahan ini selama 1 (satu) semester, mahasiswa mampu memahami, merancang dan membuat sendiri basisdata menggunakan MySQL dan membuat aplikasinya dengan menggunakan Borland Delphi 7. Selain menggunakan MySQL sebagai tool utama untuk membuat basisdata, mahasiswa juga akan diperkenalkan tool lain yang bisa digunakan untuk membuat basisdata misalnya Ms.Access dan Paradox (sebagai tambahan alternatif). Dimana pemberian materinya disesuaikan dengan kondisi waktu yang dimiliki.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

BAB I

DASAR PEMROGRAMAN DELPHI

(Pertemuan 1)



MATERI POKOK :

- 1) Mengingat (meriview) kembali materi yang telah disampaikan pada Mata Kuliah Pemrograman 1 (Delphi Dasar) yang meliputi :
 - Pengenalan Instalasi dan Interface pada Delphi
 - Pengenalan dan Penggunaan Komponen-komponen pada Delphi
 - Membuat dan menyimpan hasil project
 - Penggunaan Struktur Kondisi, Struktur Perulangan, array, fungsi dan procedure
- 2) Mendesain dan membuat aplikasi sederhana berdasarkan materi yang telah disampaikan pada Mata Kuliah Pemrograman 1 (Delphi Dasar)
- 3) Studi Kasus (Tugas Pengayaan)

1.1. Konsep Dasar

a) Mengetahui Sepintas Borland Delphi 7

Borland Delphi 7 adalah bahasa pemrograman yang bekerja pada lingkup sistem operasi windows, dan merupakan salah satu program yang berorientasi object (OOP). Kemampuannya dapat dipakai untuk merancang program aplikasi yang berpenampilan seperti program aplikasi lainnya yang berbasis Windows. Kemampuan Borland Delphi 7 secara umum adalah menyediakan komponen komponen yang memungkinkan anda membuat program aplikasi yang sesuai dengan tampilan dan cara kerja windows, diperkuat dengan bahasa pemrograman terstruktur yang sangat handal, yaitu bahasa pemrograman Object Pascal yang sangat terkenal. Khusus untuk pemrograman database Borland Delphi 7 menyediakan fasilitas object yang sangat kuat dan lengkap. Selain menyediakan format database Paradox dan dBase Borland Delphi 7 juga menangani berbagai macam format database seperti: MS-Access, ODBC, SyBASE, Oracle dan lain lain.

Untuk bisa menjalankan dan menggunakan Delphi, maka seorang user atau programmer harus terlebih dahulu memiliki file master program Delphi yang disukai. Misalnya Delphi 7, Delphi 2010 atau Delphi 2011. Misalnya selama perkuliahan ini kita akan menggunakan Borland Delphi 7, maka tahap awal yang mesti dilakukan adalah proses instalasi. Instalasi Delphi 7 tidak jauh berbeda dengan instalasi software lain. Silahkan anda coba sendiri...!!! (**Ingat waktu belajar delphi 1**).



Diktat Kuliah Pemrograman II

S1 Informatika

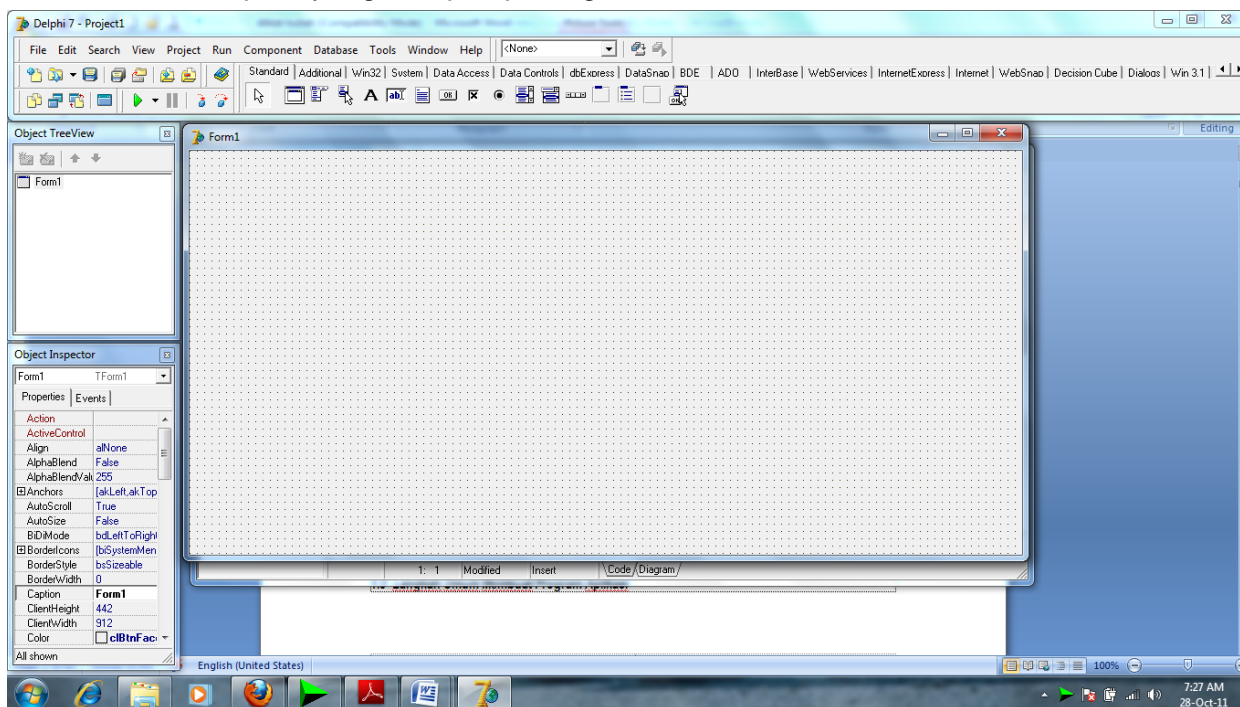
STMIK Bumigora Mataram

b) Memulai Program Borland Delphi 7

Setelah Delphi 7 sukses diinstalasi, maka selanjutnya kita sudah bisa menggunakannya untuk membuat program aplikasi yang kita inginkan.

Berikut adalah langkah-langkah untuk memulai menjalankan Borland Delphi 7 :

- ✓ Klik Icon Start pada sistem operasi Windows (pastikan program Borland Delphi 7 sudah di install)
- ✓ Tunjuk all program, lalu pilih Borland Delphi 7 kemudian klik Delphi 7, kemudian akan muncul program Borland Delphi 7 yang sudah siap dipakai, seperti yang nampak pada gambar dibawah ini :



Gambar 1.1 Tampilan Awal Lembar Kerja Delphi

c) Kegunaan Delphi

- ✓ Untuk membuat aplikasi windows
- ✓ Untuk merancang aplikasi program berbasis grafis
- ✓ Untuk membuat program berbasis jaringan (client/server)
- ✓ Untuk merancang program .Net (berbasis internet)

d) Keunggulan Delphi

- ✓ IDE (***Integrated Development Environment***) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program.
- ✓ Proses Kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada Delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
- ✓ Mudah digunakan, source kode delphi yang merupakan turunan dari pascal, sehingga tidak diperlukan suatu penyesuaian lagi.
- ✓ Bersifat multi purphase, artinya bahasa pemograman Delphi dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi.



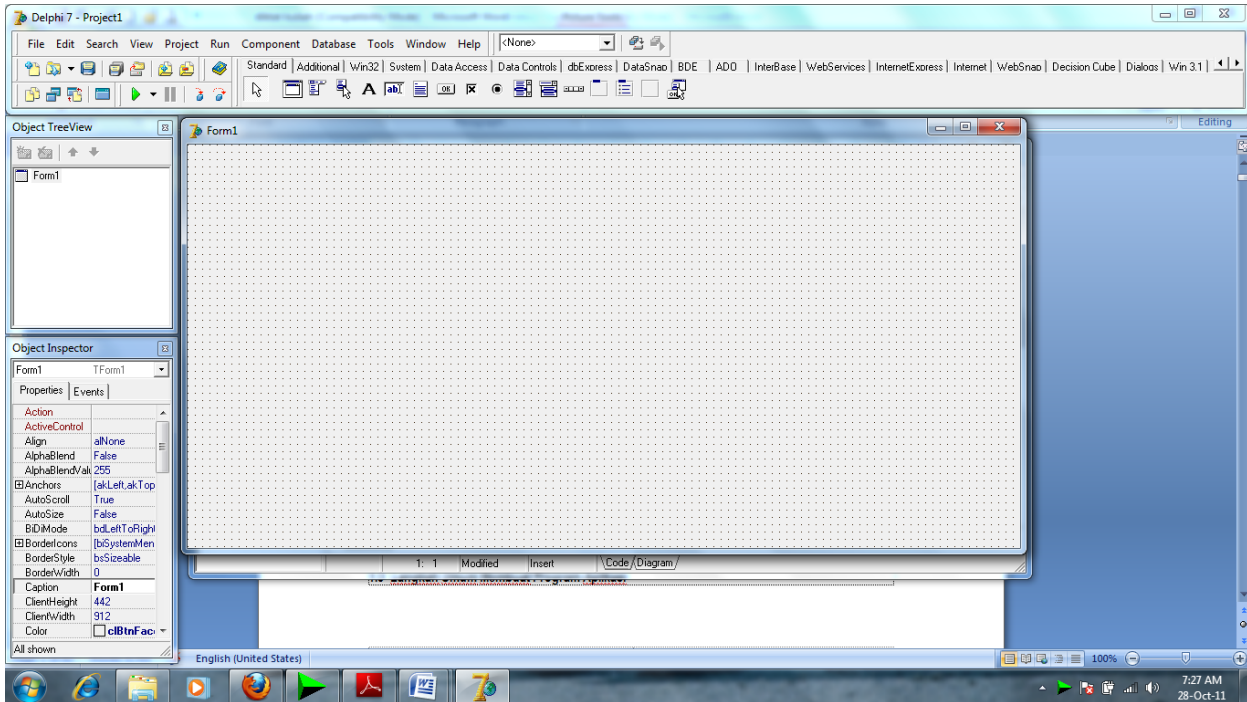
Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

e) Komponen/IDE Dalam Delphi

1. Jendela Utama

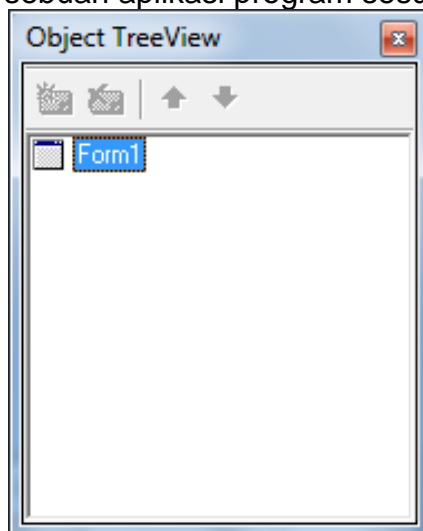


Gambar 1.2 Jendela Utama Delphi

2. Bagian-bagian dari IDE Delphi :

1). Oject Tree View

Merupakan sebuah diagram pohon yang menggambarkan hubungan logis, menghubungkan semua komponen yang terdapat dalam suatu proyek program. Komponen tersebut meliputi form, modul atau frame. Fungsinya digunakan untuk menampilkan seluruh daftar komponen program dalam sebuah aplikasi program sesuai dengan penempatannya.



Gambar 1.3 Jendela Object Tree View

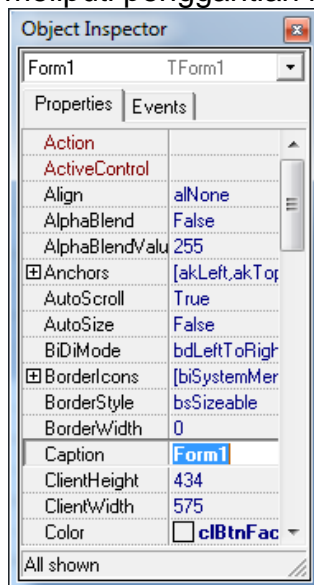
2). Object Inspector

Merupakan jendela yang digunakan untuk mengatur tampilan komponen pada form, misal bagaimana mengubah tulisan button pada command button menjadi Simpan atau menghapus tulisan pada label dan mengganti nama menjadi Nama Mahasiswa atau memberikan perintah tertentu pada sebuah komponen sehingga ada interaksi ketika program dijalankan.

Secara umum Object Inspector terbagi menjadi 2 yaitu :

a. Properties

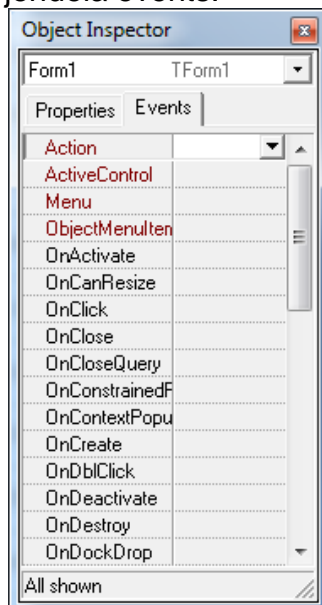
Digunakan untuk mengatur tampilan pada sebuah komponen baik itu meliputi penggantian nama, warna, jenis huruf, border dan lain-lain.



Gambar 1.4 Jendela Inspector (properties)

b. Events

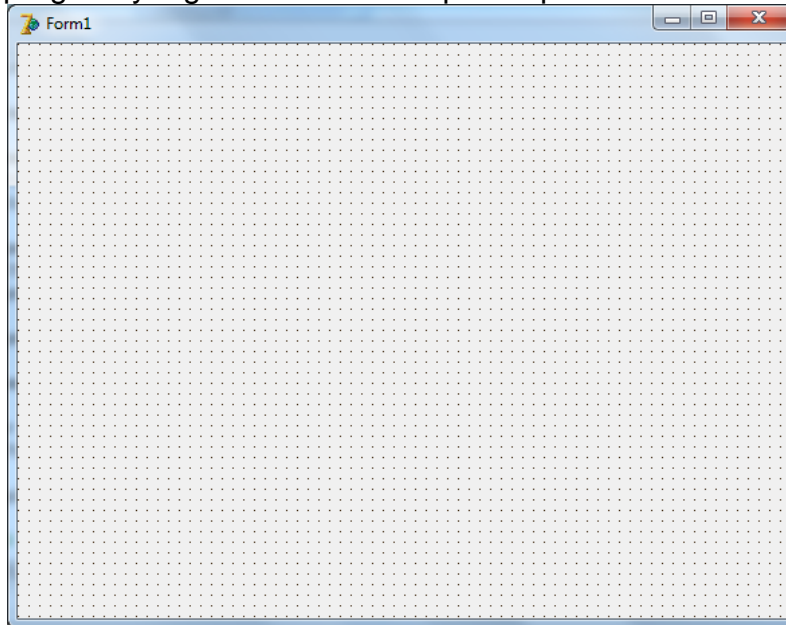
Merupakan jendela properties yang digunakan untuk memberikan fungsi yang lebih detail dari fungsi sebenarnya. Misalnya ketika tombol Simpan diklik maka program akan menjalankan perintah penyimpanan data. Dari kalimat tersebut ada event click untuk mengeksekusi sebuah tombol Simpan. Perintah event click tersebut dapat diberikan melalui jendela events.



Gambar 1.5 Jendela Events pada Object Inspector

3). Form Designer

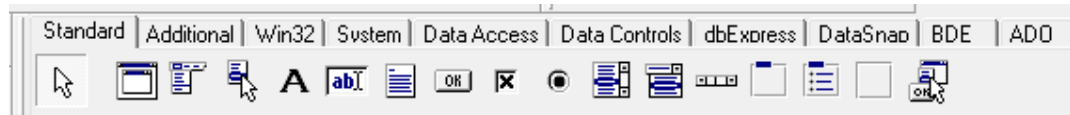
Merupakan tempat yang digunakan untuk merancang semua aplikasi program yang diambil dari komponen pallete.



Gambar 1.6 Jendela form designer

4). Component Pallete

Merupakan kumpulan icon yang digunakan untuk merancang suatu aplikasi untuk membentuk sebuah aplikasi yang user interface. Dalam komponen pallete semua icon dikelompokkan dalam berbagai komponen sesuai dengan fungsi dan kegunaannya.



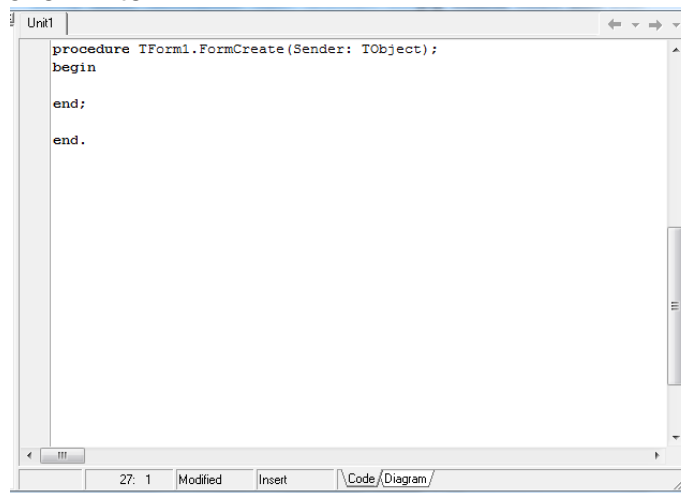
Gambar 1.7 Jendela Komponen Pallete

No	Icon	Name	Fungsi
1		Pointer	Mengembalikan fungsi mouse ke defaultnya
2		Frame	Membentuk suatu frame terhadap obyek yang ada didalamnya
3		Main menu	Membuat menu Utama
4		Popup Menus	
5		label	Hanya untuk menampilkan Teks
6		Edit	Untuk menampilkan dan input data (1 baris)
7		Memo	Sama seperti edit tetapi mempunyai kapasitas lebih besar (lebih dari 1 baris)
8		Button	Digunakan untuk melakukan eksekusi terhadap suatu proses

9		Checkbox	Digunakan untuk menentukan pilihan lebih dari satu
10		Radio Button	Digunakan untuk menentukan pilihan, tetapi hanya satu pilihan yang bisa digunakan
11		List Box	Menampilkan pilihan dalam bentuk list
12		Combo Box	Menampilkan pilihan dalam bentuk popup
13		Scroll Bar	Merupakan icon yang berupa baris status
14		Group Box	Digunakan untuk mengelompokkan suatu icon
15		Radio Group	Digunakan untuk mengelompokkan pilihan

5). Code Editor

Bagian dari Delphi yang digunakan untuk menuliskan kode program. Pada bagian code editor terdapat bagian utama yaitu bagian paling kiri yang berupa angka menunjukkan baris dan kolom. Keterangan modified menunjukkan bahwa telah terjadi modifikasi terhadap baris program. Dan paling kanan menunjukkan status keyboard tentang tombol insert atau over write.



Gambar 1.7 Jendela Code Editor

6). Code Explorer

Jendela yang digunakan untuk menampilkan sebuah variabel, tipe dan routine yang didefinisikan pada sebuah unit.



Gambar 1.8 Jendela Code Explorer



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

f) Proyek Delphi

1. File Proyek

File ini disimpan dengan ekstension **.dpr**. File ini berisi informasi mengenai seluruh proyek program.

2. File Unit

File ini merupakan kumpulan dari barisan kode program yang terdapat di jendela code editor, baik itu yang dituliskan oleh programmer maupun oleh system. Extension file ini adalah **.pas**.

File Unit dibagi menjadi 2 :

a. Bagian Interface

Barisan ini dimulai dari kata Interface (setelah nama unit), berisi seluruh deklarasi variabel, tipe data object maupun deklarasi tambahan.

b. Bagian Implementation

Dimulai dari kata kunci implementation dan diakhiri dengan kata end. Fungsinya digunakan untuk menuliskan kode program sebagai bagian dari interaksi antar komponen ataupun dengan user.

3. File form

Berisi tentang seluruh informasi yang ada kaitannya dengan form yang dibuat, meliputi tinggi, lebar, posisi form atau tentang komponen didalamnya. Penggunaan file ini tidak dianjurkan karena untuk pengaturan sudah disediakan **object inspector** sebagai media pengaturan semua komponen.

g) Menyimpan dan Menutup Proyek

Sebelum program dijalankan disarankan untuk disimpan dahulu. Untuk penyimpanan ada beberapa cara dan pilihan. Untuk menghindari kesalahan biasanya untuk memilih **Save All** atau **Save project as**. Setelah anda pilih pilihan itu maka anda akan disuruh menyimpan sebanyak dua kali yaitu menyimpan form/unitnya dan menyimpan projectnya dari unit tersebut.

Untuk menutup program aplikasi Borland Delphi 7 anda cukup memilih file kemudian pilih **Close all**, untuk keluar Klik File kemudian **Exit** atau tombol close yang terletak pada pojok kanan atas.

h) Struktur Kondisi

Dalam borland delphi terdapat operator atau struktur percabangan/kondisi yang digunakan untuk menentukan keputusan dari satu atau beberapa pilihan. Terdapat 2 (dua) macam struktur percabangan yaitu :

1. Struktur If

Digunakan untuk menentukan pilihan atas beberapa kondisi yang merupakan syarat terhadap pilihan yang sudah ditentukan.

Terdapat 2 (dua) macam struktur if yaitu :

a. If untuk kondisi pilihan tunggal

Digunakan untuk menentukan sebuah pilihan dengan kondisi tunggal.

Contoh :

```
If nilai >=80 then ket := "Lulus";
```

b. If untuk kondisi pilihan majemuk

Digunakan untuk menentukan pilihan dengan kondisi yang harus dipenuhi lebih dari satu.

Contoh :

```
If nilai >= 80 then
```

```
    Grade:="A" else
```

```
If nilai >= 70 then
```




Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
Grade := "B" else
```

```
.....  
.....
```

```
Grade := "E";
```

2. Struktur Case..Of

Berfungsi sama seperti if untuk melakukan seleksi atas beberapa pilihan dengan kondisi sebagai syarat yang harus dipenuhi. Secara fungsi, case dan if tidak ada perbedaan tetapi untuk penulisan fungsi case lebih mudah untuk diterapkan untuk pilihan atau kondisi lebih dari satu.

Contoh :

Case nilai of

```
90..100:Grade:="A";
```

```
80..89:Grade:="B+";
```

```
.....
```

```
0..29:Grade:="E";
```

```
end;
```

i) Struktur Perulangan

Merupakan control yang digunakan untuk suatu proses yang akan berjalan terus menerus. Kondisi perulangan merupakan proses berjalannya program secara terus menerus dan akan berhenti ketika proses mendapatkan kondisi yang sudah ditentukan.

Terdapat 3 (tiga) jenis struktur perulangan, yaitu :

1. Perulangan For..to..

Dibagi menjadi 2 bentuk yaitu :

a. For ... to .. do

Perulangan yang dilakukan secara ascending (dari kecil ke besar).

b. For ... downto... do

Perulangan yang dilakukan dari besar ke kecil.

2. Perulangan While do

Perulangan yang akan terus menerus dilakukan selama kondisinya bernilai benar.

Contoh :

```
i:=1;  
while i<=5 do  
begin  
write(i:2);  
i:=i+1;  
end;
```

3. Perulangan Repeat Until

Perulangan yang dilakukan terus menerus jika kondisinya bernilai salah.

Contoh :

```
i:=1;  
repeat  
write(i:2);  
i:=i+1;  
until i>5;
```

j) Array/Larik

Merupakan variabel berindeks yang bisa digunakan untuk menyimpan nilai suatu data yang berulang-ulang. Masing-masing data tersimpan pada indeks secara berurutan.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Contoh : (array 1 dimensi)

Larik Mhs :	Ana	Budi	Andi	Badun
indeks	[1]	[2]	[3]	[4]	[n]

Data juga dapat tersimpan dalam array 2 dimensi berupa matriks yang terdiri dari baris dan kolom. Misalnya terdapat matriks dengan ukuran M2x2 sebagai berikut :

Matriks M2x2

	K1	K2
B1	Basuki	Budiman
B2	Salman	Gusti

Catatan : Materi selengkapnya Lihat pada diktat Pemrograman 1

k) Procedure

Merupakan subrutin yang digunakan untuk membagi program berdasarkan modul-modul tertentu. Kelebihanannya adalah program menjadi lebih sederhana, bisa diakses kapan saja dan memudahkan dalam deteksi error source code program.

Contoh procedure dengan parameter :

```
Procedure segitiga(alas,tinggi:real; var luas:real);
Begin
  Luas:= alas*tinggi;
  L:= luas;
End;
```

l) Function

Sama dengan procedure, yaitu subrutin yang digunakan untuk membagi program berdasarkan modul-modul tertentu. Kelebihanannya adalah program menjadi lebih sederhana, bisa diakses kapan saja dan memudahkan dalam deteksi error source code program.

Contoh function dengan parameter :

```
Function segitiga(alas,tinggi:real):real;
Begin
  Segitiga:=alas*tinggi;
End;
```

1.2. Pembuatan Aplikasi Sederhana

Untuk lebih memahami materi yang telah disampaikan, silahkan anda merancang dan membuat sendiri aplikasi sederhana untuk menyelesaikan kasus berikut :

- Membuat aplikasi satu project untuk menghitung luas segitiga, luas lingkaran, luas balok dan luas persegi panjang. Selesaikan dengan menggunakan procedure dan function.
- Menghitung grade n mahasiswa

Ketentuan Tugas 1 :

- ✓ Dikumpulkan pada pertemuan selanjutnya, nilai dikurangi 3 setiap hari keterlambatan.
- ✓ Desain dan source code program disalin ulang ke kertas double folio dengan ditulis tangan.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram



Catatan :

Untuk memperjelas uraian materi pada BAB 1, silahkan buka kembali Diktat Pemrograman 1 (Delphi Dasar). File PDF nya bisa didownload pada link berikut :
<http://mhm658.files.wordpress.com/2012/01/diktat-pemrograman-1.pdf>



Tugas 2 Pertemuan 1 (Tulis Tangan) :

1. Sebut dan jelaskan perbedaan ketiga macam struktur perulangan. Jelaskan dengan contoh program dimana setiap baris program dibuatkan penjelasan.
2. Sebut dan jelaskan perbedaan procedure dengan function. Jelaskan dengan contoh potongan program dimana setiap baris program dibuatkan penjelasan.
3. Jelaskan arti istilah-istilah berikut :
 - a. Variabel lokal
 - b. Variabel global
 - c. Parameter formal
 - d. Parameter actual



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

BAB II

KONSEP DASAR DATABASE

Pertemuan 2,3,4



MATERI POKOK :

1. Konsep Dasar Basis Data
2. Pengenalan MySql
3. Dasar-dasar MySql
4. Struktur Tabel MySql
5. Operator dalam MySql
6. Database Relasi/Relational Database Management System (RDBMS)

2.1. Konsep Dasar Basis Data

2.1.1. Pengertian

Basisdata adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basisdata tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak.

2.1.2. Bahasa Pemrograman Basis Data

Bahasa pemrograman yang terdapat dalam basisdata disebut dengan perintah SQL (Structure Query Language). Semua perintah SQL tersebut secara katagori besar dibedakan menjadi 2 macam yaitu :

1. Data Definition Language (DDL)

Adalah perintah yang digunakan untuk mendefinisikan database maupun bagian dari database seperti tabel dan view. Perintah yang digunakan yaitu : CREATE (membuat database/tabel/indeks), DROP (menghapus database/tabel) dan ALTER (mengubah struktur database)

2. Data Manipulation Language (DML)

Adalah perintah yang digunakan untuk memanipulasi data dalam tabel dari suatu database. Perintah yang digunakan SELECT (mendapatkan data dari database), INSERT (memasukkan data ke database), UPDATE (mengubah data pada database) dan DELETE (menghapus data dalam database)



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

2.2. Pengenalan MySql

2.2.1. Pendahuluan

Seiring dengan waktu, banyak perkembangan yang terjadi pada dunia internet. Termasuk pesatnya perkembangan perangkat lunak Apache, MySql dan PHP yang selalu diupdate oleh produsennya masing-masing. Sebagai pertimbangan, versi terbaru dari Apache adalah apache 2.2.11 (per 14 Desember 2008), versi terbaru dari PHP adalah PHP 5.2.9 (per 26 Pebruari 2009) dan versi terbaru dari MySql adalah MySql 5.1.32 (per Maret 2009).

Awalnya ketiga perangkat lunak tersebut yaitu apache (sebagai web server), MySql (sebagai database relasi) dan PHP (bahasa pemrograman berbasis Web) dikembangkan secara terpisah-pisah. Sehingga ketiganya harus diinstall sendiri-sendiri. Namun kemudian oleh produsennya terus dilakukan pengembangan agar lebih memudahkan user dalam penggunaan. Misalnya telah dirilis versi PHPTriad yang didalamnya sekaligus terdapat apache sebagai web server dan mysql sebagai database relasi. Kemudian terakhir kita tahu sudah terdapat juga versi pengembangan terbaru dengan lahirnya XAMPP 1.8.1 (per Oktober 2012) sebagai penyedia web server apache dan mysql serta fasilitas-fasilitas lainnya yang mendukung untuk keperluan pembuatan dan pengembangan website.

XAMPP dirancang untuk bisa dijalankan dibanyak sistem operasi. Berikut adalah kepanjangan dari masing-masing huruf dalam XAMPP :

X : Program ini dapat dijalankan dibanyak sistem operasi, seperti Windows, Linux, Mac OS, dan Solaris.

A : Apache, merupakan aplikasi web server. Tugas utama Apache adalah menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. jika diperlukan juga berdasarkan kode PHP yang dituliskan, maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam MySQL) untuk mendukung halaman web yang dihasilkan.

M : MySQL, merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari Structured Query Language. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam database.

P : PHP, bahasa pemrograman web. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah MySQL. namun PHP juga mendukung sistem manajemen database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya.

P : Perl, bahasa pemrograman.

Selanjutnya selama perkuliahan kita akan banyak menggunakan PHPTriad dan XAMPP dalam pelaksanaan praktikkum maupun yang lainnya.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

2.2.2. Apa itu MySql ?

MySql adalah suatu perangkat lunak database relasi seperti halnya ORACLE, MS SQL dan sebagainya. Jangan disalah artikan MySql dengan SQL. SQL (Structure Query Language) sendiri adalah suatu sintaks perintah-perintah tertentu atau bahasa (pemrograman) yang digunakan untuk mengelola suatu database. Jadi, MySql dan SQL adalah dua makhluk yang berbeda. Mudah-mudahan, MySql adalah software-nya dan SQL adalah bahasa perintahnya.

2.2.3. Mengapa Memilih MySQL

Berikut adalah alasan kenapa harus memilih MySQL sebagai perangkat database :

a. Kecepatan

Banyak ahli berpendapat MySQL merupakan server tercepat

b. Kemudahan Penggunaan

MySQL punya performa tinggi namun merupakan database yang simple sehingga mudah disetup dan dikonfigurasi

c. Harga

MySQL cenderung gratis untuk penggunaan tertentu

d. Mendukung Query Language

MySQL mengerti bahasa SQL yang merupakan pilihan sistem database modern. Kita juga dapat mengakses MySQL lewat protokol ODBC (Open Database Connectivity) buatan microsoft.

e. Kapabilitas

Banyak klien dapat mengakses server dalam satu waktu. Mereka dapat menggunakan banyak database secara simultan.

f. Konektifitas dan sekuritas

Database MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu.

g. Pertabilitas

MySQL dapat berjalan dalam banyak varian UNIX dengan baik, sebaik seperti saat berjalan di system non-UNIX.

h. Distribusi yang terbuka

MySQL mudah didapatkan dan memiliki source code yang boleh disebarluaskan sehingga bisa dikembangkan lebih lanjut.

2.2.4. Instalasi MySQL

Sebelum dapat menggunakan MySQL terlebih dahulu kita harus memiliki file master program penyedia MySQL yang akan kita gunakan misalnya versi PHPTriad ataupun XAMPP, kemudian kita melakukan instalasi.

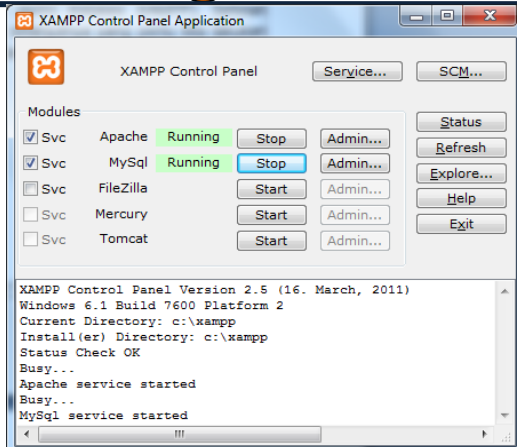
Langkah instalasi sama seperti kebanyakan menginstall software yang lain (coba sendiri/perhatikan saat dosen demo instalasi XAMPP). Setelah sukses melakukan instalasi XAMPP, maka selanjutnya yang perlu kita lakukan adalah mengaktifkan webserver (apache) dan MySQL sebagai berikut :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram



Gambar 1.9 Mengaktifkan apache dan MySQL pada XAMPP

Selanjutnya database MySQL siap untuk diciptakan. Untuk melakukan uji coba apakah berhasil atau tidak berikut caranya :

```
cmd. C:\Windows\system32\cmd.exe - mysql -u root
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Muhammad Yunus>cd/

C:\>cd xampp/

C:\xampp>cd mysql

C:\xampp\mysql>cd bin

C:\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.16 MySQL Community Server <GPL>

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

Gambar 1.10 Jendela Utama MySQL

Tugas 3 ?

Buatlah paper tentang sejarah MySQL lengkap dari versi awal sampai versi terbaru dan dikumpulkan pada pertemuan selanjutnya.

2.3. Dasar-dasar MySql

2.3.1. Konsep Dasar

Dalam bahasa SQL, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (row atau record) dan kolom (column atau field). Sedangkan dalam sebuah database dapat terdiri dari beberapa table.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Prompt `mysql>` menunjukkan bahwa database mysql telah aktif. Jika prompt ini telah aktif dapat langsung mengetikkan perintah-perintah dilingkungan MySQL.

Tabel 2.1 Perintah-perintah MySQL

Perintah	Perintah Singkat	Kegunaan
Help	<code>\h</code>	Menampilkan daftar perintah
Clear	<code>\c</code>	Menghapus (clear)
Connect	<code>\r</code>	Menghubungkan kembali database MySQL
Exit	<code>\q</code>	Keluar dari MySQL
Go	<code>\g</code>	Mengirimkan perintah kepada MySQL
Ego	<code>\G</code>	Mengirimkan perintah kepada MySQL dan menampilkan hasilnya secara vertical
Print	<code>\p</code>	Mencetak perintah saat ini
Use	<code>\u</code>	Membuat/mengganti koneksi kepada database

Ketentuan memberikan perintah :

- Perintah dalam MySQL mengenal case insensitive, perintah dapat ditulis dengan huruf besar (uppercase), ataupun dengan huruf kecil (lowercase).
- Setiap perintah diakhiri dengan ; (tanda titik koma) atau dengan memberikan perintah `\g` diakhir perintah
- Perintah dapat berupa perintah SQL atau perintah khusus MySQL
- Jika prompt `mysql>` berganti dengan `->` berarti prompt tersebut menunggu kelengkapan perintah dari baris sebelumnya atau menunggu diberikan tanda ; atau `\g`.

Contoh : perhatikan perintah dibawah ini ditulis tanpa tanda titik-koma “;”

```
mysql> create database latihan1
->
```

Sistem MySQL akan menampilkan tanda panah `'->'` yang menyatakan bahwa perintah MySQL tersebut dianggap belum selesai (karena belum diakhiri dengan tanda titik koma `';`).

Sekarang kita lengkapi perintah sebelumnya dengan tanda ;

```
mysql> create database latihan1
-> ;
Query OK, 1 row affected (0.02 sec)
```

2.3.2. Type Data pada MySQL

Pemilihan type data merupakan suatu hal yang cukup penting dalam mengelola server. Salah satu sebabnya adalah berkaitan dengan ruang di hardisk dan memori yang akan digunakan oleh data-data tersebut.

Berikut ini akan diberikan tipe-tipe data yang didukung oleh MySQL yang terambil dari dokumentasi MySQL. Tipe-tipe data ini diberikan dalam bentuk yang siap dituliskan pada sintaks-sintaks MySQL. Misalnya *create table*. Pada tipe-tipe data tersebut terdapat atribut yang memiliki arti sebagai berikut :

- M, menunjukkan lebar karakter maksimum. Nilai M maksimum adalah 255
- D, menunjukkan jumlah angka dibelakang koma. Nilai maksimum D adalah 30 tetapi dibatasi oleh nilai M, yaitu tidak boleh lebih besar daripada M-2
- Atribut yang diberi tanda [dan] berarti pemakaiannya adalah optional



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

- Jika atribut ZEROFILL disertakan, MySQL akan otomatis menambahkan atribut UNSIGNED
- UNSIGNED adalah bilangan tanpa tanda didepannya (misalnya tanda negatif).

Inilah tipe-tipe data tersebut :

No	Tipe data	Keterangan
1	TINYINT[(M)] [UNSIGNED] [ZEROFILL]	Integer yang sangat kecil jangkauannya, yaitu -128 hingga 127. Jangkauan unsigned adalah 0 hingga 255
2	SMALLINT[(M)] [UNSIGNED] [ZEROFILL]	Integer yang kecil jangkauannya, yaitu -32768 hingga 32767. Jangkauan unsigned adalah 0 hingga 65535
3	MEDIUM[(M)] [UNSIGNED] [ZEROFILL]	Integer tingkat menengah. Jangkauan nilainya adalah -8388608 hingga 8388607. Jangkauan unsigned adalah 0 hingga 16777215
4	INT[(M)] [UNSIGNED] [ZEROFILL]	Integer yang berukuran normal. Jangkauan nilainya adalah -2147483648 hingga 2147483647. Jangkauan unsigned adalah 0 hingga 4294967295
5	INTEGER[(M)] [UNSIGNED] [ZEROFILL]	Sama dengan INT
6	BIGINT[(M)] [UNSIGNED] [ZEROFILL]	Integer berukuran besar. Jangkauan nilainya adalah -9223372036854775808 hingga 9223372036854775807. Jangkauan unsigned adalah 0 hingga 18446744073709551615
7	FLOAT (precision) [ZEROFILL]	Bilangan floating point. Tidak dapat bersifat unsigned. Nilai atribut precision adalah ≤ 24 untuk bilangan floating point presisi tunggal dan diantara 25 dan 53 untuk bilangan floating point presisi ganda.
8	FLOAT[(M,D)] [ZEROFILL]	Bilangan floating point presisi tunggal. Tidak dapat bersifat unsigned. Nilai yang diizinkan adalah -3.402823466E+38 hingga -1.175494351E-38 untuk nilai negatif. 0, dan 1.175494351E-38 hingga 3.402823466E+38 untuk nilai



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

No	Tipe data	Keterangan
		positif
9	DOUBLE[(M,D)] [ZEROFILL]	Bilangan floating point presisi ganda. Tidak dapat bersifat unsigned. Nilai yang diizinkan adalah - 1.7976931348623157E+308 hingga - 2.2250738585072014E-308 untuk nilai negatif. 0, dan 2.2250738585072014E-308 hingga 1.7976931348623157E+308 untuk nilai positif
10	DOUBLE PRECISION[(M,D)] [ZEROFILL] dan REAL[(M,D)] [ZEROFILL]	Keduanya sama dengan DOUBLE
11	DECIMAL[(M[,D)] [ZEROFILL]	Bilangan floating point yang "unpacked". Tidak dapat bersifat unsigned. Memiliki sifat mirip dengan CHAR. Kata "unpacked" berarti bilangan disimpan sebagai string menggunakan satu karakter untuk setiap digitnya. Jangkauan nilai dari DECIMAL sama dengan DOUBLE, tetapi juga tergantung dari nilai atribut M dan D yang disertakan. Jika D tidak diisi akan dianggap 0. Jika M tidak diisi maka akan dianggap 10. Sejak MySQL 3.22 nilai M harus termasuk ruang yang ditempati oleh angka dibelakang koma dan tanda + atau -
12	NUMERIC(M,D) [ZEROFILL]	Sama dengan DECIMAL
13	DATE	Sebuah tanggal. MySQL menampilkan tanggal dalam format 'YYYY-MM-DD'. Jangkauan nilainya adalah '1000-01-01' hingga '9999-12-31'.
14	DATETIME	Sebuah kombinasi dari waktu (jam) dan tanggal. MySQL menampilkan waktu dan tanggal dalam format 'YYYY-MM-DD HH:MM:SS'. Jangkauan nilainya adalah '1000-01-01 00:00:00' hingga '9999-12-31 23:59:59'



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

No	Tipe data	Keterangan
15	TIMESTAMP[(M)]	Sebuah timestamp. Jangkauan nilainya adalah dari '1970-01-01 00:00:00' hingga suatu waktu di tahun 2037. MySQL menampilkan tipe data TIMESTAMP dalam format YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD, atau YYMMDD, tergantung dari nilai M. apakah 14 (atau tidak ditulis), 12, 8, atau 6.
16	TIME	Tipe data waktu. Jangkauannya adalah '-838:59:59' hingga '838:59:59'. MySQL menampilkan TIME dalam format 'HH:MM:SS'
17	YEAR[(2 4)]	Angka tahun, dalam format 2- atau 4-digit (default 4 digit). Nilai yang mungkin adalah 1901 hingga 2155.0000 pada format 4 digit dan 1970-2069 pada format 2 digit (70-69)
18	CHAR(M) [BINARY]	String yang memiliki lebar tetap. Nilai M adalah 1 hingga 255 karakter. Jika ada sisa, maka sisa tersebut diisi dengan spasi (misalnya nilai M adalah 10, tapi data yang disimpan hanya memiliki 7 karakter, maka 3 karakter sisanya diisi dengan spasi). Spasi ini akan dihilangkan apabila data dipanggil. Nilai dari CHAR akan disortir dan diperbandingkan secara case insensitive menurut default character set yang tersedia, kecuali bila atribut BINARY disertakan.
19	VARCHAR(M) [BINARY]	String dengan lebar bervariasi. Nilai M adalah dari 1 hingga 255 karakter. Jika nilai M adalah 10 sedangkan data yang disimpan hanya terdiri dari 5 karakter, maka lebar data tersebut hanya 5 karakter saja, tidak ada tambahan spasi.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

No	Type data	Keterangan
20	TINYBLOB dan TINYTEXT	Sebuah BLOB (semacam catatan) atau TEXT dengan lebar maksimum 255 (2^8-1) karakter
21	BLOB dan TEXT	Sebuah BLOB atau TEXT dengan lebar maksimum 65535 ($2^{16}-1$) karakter
22	MEDIUMBLOB dan MEDIUMTEXT	Sebuah BLOB atau TEXT dengan lebar maksimum 16777215 ($2^{24}-1$) karakter
23	LOB dan LONGTEXT	Sebuah BLOB atau TEXT dengan lebar maksimum 4294967295 ($2^{32}-1$) karakter
24	ENUM('value1','value2'),...	Sebuah enumerasi, yaitu objek string yang hanya dapat memiliki sebuah nilai, dipilih dari daftar nilai 'value1', 'value2',... NULL atau nilai special "" error. Sebuah ENUM maksimum dapat memiliki 65535 jenis nilai
25	SET('value1','value2',...)	Sebuah set, yaitu objek string yang dapat memiliki 0 nilai atau lebih yang harus dipilih dari daftar nilai 'value1', 'value2',... Sebuah SET maksimum dapat memiliki 64 anggota

2.3.3. Melihat User dan Versi MySQL

Untuk melihat user dan versi MySQL, kita menggunakan rumus :

```
Select User (),Version ();  
  
mysql> select user(),version();  
+-----+-----+  
| user()          | version()          |  
+-----+-----+  
| root@localhost | 5.1.34-community  |  
+-----+-----+  
1 row in set (0.05 sec)
```

2.3.4. Melihat Tanggal dan Waktu

Untuk melihat tanggal dalam MySQL dapat menggunakan rumus **curdate()** :

```
MySQL>select curdate();
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Sedangkan untuk melihat waktu dengan rumus `curtime()` :

```
MySQL>select curtime();
```

Untuk melihat waktu dan sekaligus tanggal, maka rumus yang dituliskan adalah `now()` :

```
MySQL>select now();
```

2.3.5. MySQL sebagai Kalkulator

Dengan MySQL, kita tidak usah bingung ketika suatu saat kita harus menggunakan alat bantu kalkulator, karena hal ini dapat ditangani langsung oleh MySQL tanpa harus membuat program terlebih dahulu. Rumus yang dituliskan adalah `select rumus_perhitungan` :

```
mysql> select 5+5;
+-----+
| 5+5 |
+-----+
| 10 |
+-----+
1 row in set (0.03 sec)
```

Operator perhitungan meliputi : penjumlahan (+), pengurangan (-), perkalian (*), dan pembagian (/).

2.3.6. Membuat Database Baru

Sudah ada 3 buah database dalam sistem MySQL. Sekarang kita akan membuat sebuah database untuk latihan kita. Gunakan perintah "**CREATE DATABASE**" untuk membuat sebuah database.

```
mysql> create database latihan2 ;
Query OK, 1 row affected (0.02 sec)
```

Kita periksa hasil dari perintah diatas dengan "**SHOW DATABASE**" :

```
mysql> show databases ;
+-----+
| Database |
+-----+
| information_schema |
| latihan1 |
| latihan2 |
| mysql |
| test |
+-----+
5 rows in set (0.00 sec)
```

2.3.7. Menghapus Database

Jika kita tidak memerlukan database *latihan2*, maka kita dapat menghapusnya dengan perintah **DROP DATABASE**. Hati-hati dalam menggunakan perintah **DROP DATABASE** ini, karena database beserta seluruh isinya akan lenyap dari muka bumi tanpa bisa kita kembalikan lagi !.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Parahnya lagi sistem MySQL tidak memberikan pertanyaan konfirmasi kepada anda sebelum melakukan proses penghapusan database ini !

```
mysql> drop database latihan2 ;  
Query OK, 0 row affected (0.02 sec)
```

Anda bisa memeriksanya lagi hasil dari perintah diatas dengan “SHOW DATABASE” :

```
mysql> show databases ;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| latihan1 |  
| mysql |  
| test |  
+-----+  
4 rows in set (0.00 sec)
```

Anda perhatikan, database *latihan2* sudah menghilang. Sekali lagi hati-hati dalam menggunakan perintah DROP DATABASE !

2.3.8. Memilih dan Membuka Sebuah Database

Untuk memilih database yang akan digunakan maka perintahnya adalah “USE”. Contohnya : misalnya kita akan menggunakan database *latihan1*:

```
mysql> use latihan1 ;  
Database change
```

2.3.9. Melihat Isi Sebuah Database

Sekarang kita pilih database “latihan1”, kita gunakan perintah SHOW TABLES.

```
mysql> show tables ;  
Empty set (0.00 sec)
```

Hasil dari perintah diatas adalah “Empty Set”, yang berarti belum ada tabel apapun didalam database *latihan1*.

2.3.10. Membuat Table Baru

Kita bisa membuat tabel baru dengan menggunakan perintah “CREATE TABLE”.

```
mysql> create table karyawan ;  
ERROR 1113 (42000): A table must have at least 1 column
```

Ternyata ada kesalahan yang terjadi. Untuk membuat sebuah table di MySQL, kita harus menentukan minimal satu buah field/kolom didalamnya. Sekarang kita ubah perintah diatas menjadi sebagai berikut :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
mysql> create table karyawan
-> (nopeg INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
-> nama VARCHAR(50) NOT NULL)
-> ;
Query OK, 0 rows affected (0.14 sec)
```

Cukup panjang ya perubahan perintahnya. Mungkin sintaks perintahnya agak membingungkan pada awalnya. Tidak apa-apa, nanti akan kita bahas artinya. Secara umum, kita akan membuat sebuah **tabel karyawan** dengan 2 buah kolom/field

- ✚ Kolom pertama adalah **NOPEG** dengan jenis data bilangan bulat (integer), tanpa tanda negatif (UNSIGNED), yang akan bertambah nilainya secara otomatis (AUTO_INCREMENT). Kolom NOPEG adalah kolom utama (PRIMARY KEY)
- ✚ Pada kolom kedua, **NAMA** akan menampung nama karyawan dengan jenis data VARIabel CHAracter, lebar datanya dapat menampung maksimal 50 karakter dan tidak boleh dikosongkan (NOT NULL).

Kita lihat kembali apa isi dari database *latihan1* :

```
mysql> show tables ;
+-----+
| Tables_in_latihan1 |
+-----+
| karyawan           |
+-----+
1 row in set (0.00 sec)
```

Dari hasil perintah diatas, kita lihat bahwa database latihan1 telah memiliki sebuah tabel yang bernama karyawan. Selanjutnya kita akan lihat apa struktur dari tabel karyawan tersebut.

2.3.11. Melihat Struktur Table

Untuk melihat struktur sebuah tabel dapat menggunakan perintah “**DESCRIBE**” atau bisa juga menggunakan “**SHOW COLUMNS FROM**”. Contohnya berikut ini :

```
mysql> describe karyawan ;
+-----+-----+-----+-----+-----+-----+
|Field |Type           | Null |Key |Default|Extra           |
+-----+-----+-----+-----+-----+-----+
|nopeg |int(10) unsigned| NO   |PRI |NULL   |auto_increment |
|nama  |varchar(50)     | NO   |    |       |               |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

Atau menggunakan perintah “**SHOW COLUMNS FROM**” :

```
mysql> show columns from karyawan ;
+-----+-----+-----+-----+-----+-----+
|Field|Type           |Null|Key|Default|Extra           |
+-----+-----+-----+-----+-----+-----+
|nopeg|int(10) unsigned|NO  |PRI|NULL   |auto_increment |
|nama |varchar(50)     |NO  |   |       |               |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Tidak ada perbedaan hasil dari dua perintah diatas, bukan? Sekarang kita buat sebuah tabel baru lagi. Kita namakan saja tabel *contoh1*.

```
mysql> create table contoh1
-> (noid INT)
-> ;
Query OK, 0 rows affected (0.13 sec)
```

Sekarang kita lihat berapa tabel yang ada didalam database *latihan1*

```
mysql> show tables ;
+-----+
| Tables_in_latihan1 |
+-----+
| contoh1             |
| karyawan            |
+-----+
2 rows in set (0.00 sec)
```

2.3.12. Menghapus Table

Tabel *contoh1* yang baru saja kita buat ini akan kita hapus kembali. Perintah untuk menghapus sebuah tabel dalam MySQL adalah “**DROP TABLE**”. Cukup mirip dengan perintah menghapus database, bukan? Kita harus menggunakan perintah “DROP” ini dengan **kehati-hatian yang tinggi**. Sistem MySQL tidak akan memberikan peringatan awal atau konfirmasi untuk proses penghapusan tabel. Dan bila sudah terhapus, maka tabel tersebut tidak bisa lagi kita kembalikan. **Maka, berhati-hatilah!!**

```
mysql> drop table contoh1 ;
Query OK, 0 rows affected (0.03 sec)
```

Kita lihat lagi tabel yang ada dalam database *latihan1*:

```
mysql> show tables ;
+-----+
| Tables_in_latihan1 |
+-----+
| karyawan            |
+-----+
1 rows in set (0.00 sec)
```

2.3.13. Constraint

Constraint adalah batasan atau aturan yang ada pada tabel. MySQL menyediakan beberapa tipe constraint berikut :

No	Constraint	Keterangan
1	NOT NULL	Suatu kolom yang didefinisikan dengan constraint NOT NULL tidak boleh berisi nilai NULL. Kolom yang berfungsi sebagai kunci primer (primary key) otomatis tidak boleh NULL
2	UNIQUE	Mendefinisikan suatu kolom menjadi bersifat unik, artinya antara satu data dengan data lainnya namanya tidak boleh sama, misal alamat email
3	PRIMARY KEY	Constraint PRIMARY KEY membentuk key yang unik untuk suatu tabel



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

4	FOREIGN KEY	FOREIGN KEY constraint didefinisikan pada suatu kolom yang ada pada suatu tabel, dimana kolom tersebut juga dimiliki oleh tabel yang lain sebagai suatu PRIMARY KEY, biasa dipakai untuk menghubungkan antara 2 tabel
---	-------------	---

Soal Latihan (Tugas 4) :

Buat database dengan nama dbKursus. Pilih dan buka database tersebut. Buat tabel dengan nama peserta untuk menyimpan data peserta meliputi : nomor, nama, email, alamat dan kota. Sedangkan strukturnya seperti tabel dibawah ini :

Kolom (<i>Field</i>)	Tipe Data (<i>Data Type</i>)
nomor	Char (8) Not Null Primary Key
nama	VarChar (20) Not Null
email	VarChar (30) Null
alamat	VarChar (20) Not Null
kota	VarChar (10) Not Null

Perintah-perintah SQL nya ditulis tangan dan dikumpulkan pertemuan berikutnya!!

2.4. Struktur Tabel MySql

2.4.1. Mengubah Struktur Sebuah Tabel

Ada saatnya kita perlu mengubah struktur tabel yang pernah kita buat sebelumnya. Perubahan struktur bisa dalam hal penambahan kolom (**ADD**), perubahan lebar dan jenis kolom (**MODIFY**), atau bisa saja penghapusan kolom dan indeks (**DROP**), penggantian nama kolom (**CHANGE**), penggantian nama tabel (**RENAME**), dan sebagainya. Apapun yang kita lakukan pada kolom tersebut tentu akan mempunyai dampak langsung pada data-data yang sudah ada. Nah, sekarang kita perlu menambahkan beberapa kolom baru, yaitu kolom jenis kelamin, kota, tanggal lahir dan kodepos pada tabel *karyawan*.

Perintah untuk mengubah struktur tabel adalah "**ALTER TABLE**". Mari kita coba :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
mysql> ALTER TABLE karyawan
-> ADD jenkelamin CHAR(2) NOT NULL,
-> ADD kota VARCHAR(25) NOT NULL,
-> ADD kodepos CHAR(5) NOT NULL,
-> ADD tgllahir DATE
-> ;
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Sekarang kita lihat hasilnya :

```
mysql> describe karyawan ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| nopeg      | int(10)       | NO   | PRI | NULL    | auto_increment|
| nama       | varchar(50)   | NO   |     |         |                |
| jenkelamin | char(2)       | YES  |     | NULL    |                |
| kota       | varchar(25)   | NO   |     |         |                |
| kodepos    | char(5)       | NO   |     |         |                |
| tgllahir   | date          | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Jenis kelamin hanya membutuhkan lebar 1, oleh karena itu kita bisa mengubah lebar data pada jenis kelamin tersebut dengan perintah MODIFY,

```
mysql> ALTER TABLE karyawan MODIFY jenkelamin Char(1);
Query OK, 0 rows affected (0.25 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Sekarang kita lihat hasilnya :

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| nopeg      | int(10) unsigned | NO   | PRI | NULL    | auto_increment|
| nama       | varchar(50)   | NO   |     |         |                |
| jenkelamin | char(1)       | YES  |     | NULL    |                |
| kota       | varchar(25)   | NO   |     |         |                |
| kodepos    | char(5)       | NO   |     |         |                |
| tgllahir   | date          | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.30 sec)
```

Mungkin ada baiknya kalau nama kolom *nopeg* kita ubah saja menjadi *noid*. Begitupun dengan nama kolom *jenkelamin*, kita ubah namanya menjadi *jenkel* saja. Dalam perubahan kolom sebaiknya sifat-sifat kolom asli tetap ditulis ulang. Misal bila kolom *nopeg* memiliki sifat "*auto_increment*", maka selama sifat itu tetap dipertahankan, maka dia (*auto_increment*) harus ditulis ulang. Begini caranya, mengubah kolom *nopeg* menjadi *noid*, tanpa mengubah jenis datanya (tetap INT(10), dan tetap *auto_increment*):

```
mysql> alter table karyawan
-> change nopeg noid int(10) auto_increment
-> ;
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Mengubah kolom *jenkelamin* menjadi *jenkel*,



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
mysql> alter table karyawan
-> change jenkelamin jenkel char(1) ;
Query OK, 0 rows affected (0.24 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Sekarang kita lihat struktur tabel setelah perubahan :

```
mysql> describe karyawan ;
+-----+-----+-----+-----+-----+-----+
|Field  |Type      |Null|Key  |Default|Extra  |
+-----+-----+-----+-----+-----+-----+
|noid   |int(10)   |NO  |PRI  |NULL   |auto_increment|
|nama   |varchar(50)|NO  |     |       |           |
|jenkel |char(1)   |YES  |     |NULL   |           |
|kota   |varchar(25)|NO  |     |       |           |
|kodepos|char(5)   |NO  |     |       |           |
|tgllahir|date     |YES  |     |NULL   |           |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Sekarang kita akan mencoba mengubah nama tabel karyawan menjadi pegawai, berikut caranya :

```
mysql> alter table karyawan
-> rename pegawai ;
Query OK, 0 rows affected (0.09 sec)
```

Kita lihat lagi hasilnya :

```
mysql> show tables ;
+-----+
| Tables_in_latihan1 |
+-----+
| pegawai             |
+-----+
1 row in set (0.00 sec)
```

Sekarang kita kembalikan lagi nama tabel pegawai menjadi karyawan tapi dengan menggunakan perintah yang berbeda, yaitu **“RENAME TABLE”** :

```
mysql> rename table pegawai
-> to karyawan
-> ;
Query OK, 0 rows affected (0.06 sec)
```

Jangan lupa untuk memeriksa hasilnya :

```
mysql> show tables ;
+-----+
| Tables_in_latihan1 |
+-----+
| karyawan            |
+-----+
1 row in set (0.00 sec)
```

2.4.2. Mengisi Data ke dalam Tabel

Kita akan mulai mengisi data karyawan ke dalam table. Perintah yang digunakan adalah **“INSERT INTO”**. Cara yang pertama sebagai berikut :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
mysql> insert into karyawan
-> (nama, jenkel, kota, kodepos, tgllahir)
-> values
-> ("Ahmad Zobari", "L", "Bandung", "41011", "1977-10-02")
-> ;
Query OK, 1 row affected (0.17 sec)
```

Anda perhatikan bahwa dalam pemasukan data yang berjenis karakter, selalu diapit dengan tanda kutip ganda (" "). Tetapi jangan dicampur dengan tanda kutip ganda dan tanda kutip tunggal, misal : "Ahmad Zobari". Perhatikan juga pada penulisan tanggal lahir, menggunakan format "tahun-bulan-tanggal". Memang agak janggal, tapi begitulah standar MySQL untuk format penulisan tanggal.

Kalau anda perhatikan lebih teliti, mengapa kita tidak memasukkan data untuk kolom "noid"? Ini karena sifat kolom noid yang auto_increment, sehingga dia akan secara otomatis berisi dengan angka 1, dan terus bertambah 1, seiring dengan penambahan data.

Nah, kita akan masukkan 3 buah record lagi dengan cara :

```
mysql> insert into karyawan
-> (nama, jenkel, kota, kodepos, tgllahir)
-> values
-> ("Sundariwati", "P", "Bandung", "40123", "1978-11-12"),
-> ("Ryan Cakep", "L", "Jakarta", "12111", "1981-03-21"),
-> ("Zukarman", "L", "Bekasi", "17211", "1978-08-10"),
-> ;
Query OK, 3 rows affected (0.05 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

Sekarang kita coba memasukkan data dengan cara yang kedua :

```
mysql> insert into karyawan
-> set nama="Yuliawati",
-> jenkel="P",
-> kota="Bogor",
-> kodepos="00000",
-> tgllahir="1982-06-09"
-> ;
Query OK, 1 row affected (0.05 sec)
```

Dan kita akan mencoba cara yang ketiga sebagai berikut :

```
mysql> insert into karyawan
-> values ("Mawar", "P", "Bogor", "12345", "1985-07-07")
-> ;
ERROR 1136: Column count doesn't match value count at row 1
```

Terdapat pesan error. Dimana disampaikan bahwa jumlah kolom tidak sesuai dengan jumlah masukan data yang ada. Kita ketahui bahwa jumlah kolom ada 6 yaitu noid, nama, jenkel, kota, kodepos dan tgllahir. Sedangkan data yang kita masukkan untuk 5 kolom saja yaitu nama, jenkel, kota, kodepos dan tgllahir. Jadi dengan demikian maka kolom noid menjadi tidak terisi. Itulah yang menyebabkan terjadinya error. Walaupun kolom noid ini sifatnya auto_increment, khusus untuk bentuk ketiga ini dia harus diisi juga dengan



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

nilai defaultnya yaitu "NULL". Sehingga perintah diatas kita ubah sedikit menjadi :

```
mysql> insert into karyawan
-> values ("NULL","Mawar","P","Bogor","12345","1985-07-07")
-> ;
Query OK, 1 row affected (0.03 sec)
```

2.4.3. Melihat Data pada Tabel

Kita bias melihat data yang ada didalam table dengan menggunakan perintah "SELECT". Perintah SELECT adalah perintah yang akan sering kita gunakan nantinya. Kita mulai dengan cara yang paling sederhana dulu yaitu :

```
mysql> select * from karyawan ;
+-----+-----+-----+-----+-----+
|noid|nama      |jenkel|kota  |kodepos|tgllahir |
+-----+-----+-----+-----+-----+
| 1|Ahmad Zobari|L     |Bandung|41011  |1977-10-02|
| 2|Sundariwati|P     |Bandung|40123  |1978-11-12|
| 3|Ryan Cakep |L     |Jakarta|12111  |1981-03-21|
| 4|Zukarman  |L     |Bekasi |17211  |1978-08-10|
| 5|Yuliahwati|P     |Bogor  |00000  |1982-06-09|
| 6|Mawar     |P     |Bogor  |12345  |1985-07-07|
+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)
```

Perintah diatas menampilkan seluruh data yang ada didalam table karyawan, karena menggunakan tanda asterisk "*" didalam perintah **SELECT**. Bagaimana kalau kita hanya mau menampilkan kolom nama dan jenis kelamin saja ?

```
mysql> select nama, jenkel from karyawan
-> ;
+-----+-----+
| nama      | jenkel |
+-----+-----+
| Ahmad Zobari | L     |
| Sundariwati | P     |
| Ryan Cakep  | L     |
| Zukarman    | L     |
| Yuliahwati  | P     |
| Mawar      | P     |
+-----+-----+
6 rows in set (0.00 sec)
```

Kalau kita hanya mau menampilkan data-data karyawan yang berjenis kelamin perempuan saja, caranya adalah cukup menambahkan perintah "**WHERE**" pada "**SELECT**" :

```
mysql> select nama, jenkel from karyawan
-> where jenkel="P"
-> ;
+-----+-----+
| nama      | jenkel |
+-----+-----+
| Sundariwati | P     |
| Yuliahwati  | P     |
| Mawar      | P     |
+-----+-----+
3 rows in set (0.00 sec)
```

Kita tampilkan data berdasarkan urutan nama karyawan dengan menambahkan perintah "**ORDER BY**" pada "**SELECT**" :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
mysql> select * from karyawan
-> order by nama ;
+-----+-----+-----+-----+-----+-----+
|noid|nama          |jenkel|kota   |kodepos|tgllahir|
+-----+-----+-----+-----+-----+-----+
| 1 |Ahmad Zobari |L     |Bandung|41011  |1977-10-02|
| 6 |Mawar       |P     |Bogor  |12345  |1985-07-07|
| 3 |Ryan Cakep  |L     |Jakarta|12111  |1981-03-21|
| 2 |Sundariwati|P     |Bandung|40123  |1978-11-12|
| 5 |Yulawati   |P     |Bogor  |00000  |1982-06-09|
| 4 |Zukarman   |L     |Bekasi |17211  |1978-08-10|
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Atau diurut berdasarkan **kota** :

```
mysql> select * from karyawan
-> order by kota ;
+-----+-----+-----+-----+-----+-----+
|noid|nama          |jenkel|kota   |kodepos|tgllahir|
+-----+-----+-----+-----+-----+-----+
| 1 |Ahmad Zobari |L     |Bandung|41011  |1977-10-02|
| 2 |Sundariwati |P     |Bandung|40123  |1978-11-12|
| 4 |Zukarman   |L     |Bekasi |17211  |1978-08-10|
| 5 |Yulawati   |P     |Bogor  |00000  |1982-06-09|
| 6 |Mawar       |P     |Bogor  |12345  |1985-07-07|
| 3 |Ryan Cakep  |L     |Jakarta|12111  |1981-03-21|
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Ternyata kita perlu menambahkan sebuah kolom lagi, yaitu kolom gaji. Kolom gaji merupakan kolom numerik yang menampung data gaji pokok karyawan per bulannya. Jadi, yang kita perlukan adalah jenis data **INT**eger dengan lebar data 12 digit.

```
mysql> alter table karyawan
-> ADD gaji INT(12) NOT NULL default 0
-> ;
Query OK, 6 rows affected (0.25 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

2.4.4. Meng-Update Data pada Tabel

Sekarang kita masukkan data gaji masing-masing karyawan dengan menggunakan perintah **UPDATE**. Kita mulai dari **Ahmad Zobari** dengan **noid=1**:

```
mysql> update karyawan
-> set gaji=1000000
-> where noid=1 ;
Query OK, 1 row affected (0.09 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Periksa dulu hasilnya :

```
mysql> select * from karyawan
-> where noid=1 ;
+-----+-----+-----+-----+-----+-----+
|noid|nama          |jenkel|kota   |kodepos|tgllahir|gaji  |
+-----+-----+-----+-----+-----+-----+-----+
| 1 |Ahmad Zobari |L     |Bandung|41011  |1977-10-02|1000000|
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Kita lanjutkan ke data karyawan berikutnya. Kita update satu per satu sehingga hasilnya sebagai berikut :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
mysql> select * from karyawan ;
+-----+-----+-----+-----+-----+-----+-----+
| noid | nama          | jenkel | kota    | kodepos | tgllahir | gaji    |
+-----+-----+-----+-----+-----+-----+-----+
| 1    | Ahmad Zobari  | L      | Bandung | 41011   | 1977-10-02 | 1000000 |
| 2    | Sundariwati  | P      | Bandung | 40123   | 1978-11-12 | 1250000 |
| 3    | Ryan Cakep   | L      | Jakarta | 12111   | 1981-03-21 | 1500000 |
| 4    | Zukarman     | L      | Bekasi  | 17211   | 1978-08-10 | 1750000 |
| 5    | Yuliawati    | P      | Bogor   | 00000   | 1982-06-09 | 2000000 |
| 6    | Mawar        | P      | Bogor   | 12345   | 1985-07-07 | 2250000 |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Cukup mudah kan? Nah, inilah dasar-dasar menggunakan perintah MySQL. Sekarang kita membutuhkan lebih banyak data untuk latihan kita. Ya minimal sekitar 30-an data lagi. Tapi apakah ada cara lain yang lebih mudah dibanding harus mengetikkan datanya satu per satu? Kan kalau kita ketik satu per satu, faktor resiko kesalahan ketik karena faktor kelelahan, dan sebagainya. Untungnya untuk pemasukan data masal kita bisa menggunakan cara yang lebih mudah.

2.4.5. Pemasukan Data Secara Masal

Untuk pemasukkan data secara masal, kita menggunakan data-data yang telah ditulis dalam sebuah file teks biasa. File ini kita namakan **tambahdata.txt**, dan untuk contoh ini kita simpan di dalam folder **C:\Data**. Perintah yang digunakan adalah sebagai berikut :

```
mysql> load data local infile 'C:\\data\\tambahdata.txt'
-> into table karyawan
-> fields terminated by ','
-> lines terminated by '\n'
-> ;
Query OK, 36 rows affected, 36 warnings (0.47 sec)
Records: 36 Deleted: 0 Skipped: 0 Warnings: 0
```

Catatan :

Perhatikan diatas bahwa digunakan 2 garis miring (\\) sebagai pembatas nama direktori, bukannya tunggal (\). Ini karena tanda \' dianggap sebagai karakter khusus oleh MySQL (disebut sebagai Escape Character).

Kelebihan menggunakan cara tersebut diatas adalah kita bisa “mengkawinkan” data-data dari program database apa saja (seperti Ms. Access, dBASE, FoxPro dan sebagainya) ke dalam MySQL dengan syarat diubah dulu ke dalam bentuk file teks.

Ada pesan dari sistem kalau perintah MySQL berhasil dilaksanakan. Tapi, tidak ada salahnya kalau kita periksa juga. Sekarang kita lihat hasilnya di tabel **karyawan** :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
mysql> select * from karyawan ;
+-----+-----+-----+-----+-----+-----+-----+
|noid | nama          |jenkel |kota      |kodepos |tgllahir |gaji   |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Ahmad Zobari |L      |Bandung   |41011   |1977-10-02|1000000|
| 2 | Sundariwati  |P      |Bandung   |40123   |1978-11-12|1250000|
| 3 | Ryan Cakep   |L      |Jakarta   |12111   |1981-03-21|1500000|
| 4 | Zukarman     |L      |Bekasi    |17211   |1978-08-10|1750000|
| 5 | Yulawati     |P      |Bogor     |00000   |1982-06-09|2000000|
| 6 | Mawar        |P      |Bogor     |12345   |1985-07-07|2250000|
| 7 | Zobari       |L      |Jakarta   |41011   |1976-10-02|1100000|
| 8 | Melia        |P      |Bandung   |40123   |1979-11-12|1200000|
| 9 | Zanda Cute   |L      |Jakarta   |12111   |1980-03-21|1300000|
| 10 | Maman        |L      |Bekasi    |17211   |1977-08-10|1400000|
| 11 | Yenny        |P      |Bogor     |00000   |1985-06-09|1150000|
| 12 | Rossa        |P      |Jakarta   |12345   |1987-07-07|1350000|
| 13 | Dadan        |L      |Bandung   |41011   |1975-10-02|1450000|
| 14 | Wawan        |P      |Semarang  |40123   |1971-11-12|1600000|
| 15 | The Cute     |L      |Jakarta   |12111   |1977-03-21|1700000|
| 16 | Marpaung     |L      |Surabaya  |17211   |1988-08-10|1800000|
| 17 | Yono         |P      |Bogor     |00000   |1989-06-09|1900000|
| 18 | Dian         |P      |Jakarta   |12345   |1980-07-07|1650000|
| 19 | Donno        |L      |Bandung   |41011   |1971-10-02|1850000|
| 20 | Ratu         |P      |Yogyakarta|40123   |1972-11-12|1950000|
| 21 | Bambang      |L      |Jakarta   |12111   |1982-03-21|2100000|
| 22 | Dadang       |L      |Surabaya  |17211   |1977-08-10|2200000|
| 23 | Yulawati     |P      |Bogor     |00000   |1974-06-09|2300000|
| 24 | Miranda      |P      |Bogor     |12345   |1980-07-07|2400000|
| 25 | Subur        |L      |Bandung   |41011   |1977-10-02|2150000|
| 26 | Banowati     |P      |Malang    |40123   |1978-11-12|2350000|
| 27 | Gungun       |L      |Jakarta   |12111   |1981-03-21|2450000|
| 28 | Gunadi       |L      |Bekasi    |17211   |1978-08-10|2125000|
| 29 | Yossy        |P      |Bogor     |00000   |1982-06-09|2225000|
| 30 | Melia        |P      |Malang    |12345   |1981-07-07|2325000|
| 31 | Anwar        |L      |Purwakarta|41011   |1972-10-02|2425000|
| 32 | Susilowati   |P      |Bandung   |40123   |1973-11-12|1125000|
| 33 | Rahmat       |L      |Jakarta   |12111   |1977-03-21|1225000|
+-----+-----+-----+-----+-----+-----+-----+
| 34 | Zamzam       |L      |Bekasi    |17211   |1974-08-10|1325000|
| 35 | Nenny        |P      |Medan     |00000   |1972-06-09|1425000|
| 36 | Mardiatun    |P      |Bogor     |12345   |1975-07-07|1625000|
| 37 | Andika       |L      |Bandung   |41011   |1978-10-02|1725000|
| 38 | Siti         |P      |Medan     |40123   |1988-11-12|1825000|
| 39 | Rohimat     |L      |Jakarta   |12111   |1980-03-21|1925000|
| 40 | Beno         |L      |Bekasi    |17211   |1978-08-10|1175000|
| 41 | Yanti        |P      |Jakarta   |00000   |1981-06-09|1275000|
| 42 | Miranti      |P      |Medan     |12345   |1975-07-07|1375000|
+-----+-----+-----+-----+-----+-----+-----+
42 rows in set (0.00 sec)
```

Soal Latihan (Tugas 5) :

Buka database dbKursus yang telah anda buat. Kemudian buka struktur table peserta.

- ✚ Ganti field nomor dengan idPeserta, tipe data sama
- ✚ Ganti lebar data field email menjadi 50
- ✚ Tambahkan field tempatlhr dan tgllhr
- ✚ Isikan data pada tabel tersebut, minimal 10 data



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

2.5. Operator dalam MySql

2.5.1. Operator Perbandingan dan Operator Logika

Sudah saatnya kita melangkah ke permainan data yang lebih mengasyikan lagi dengan menggunakan dua operator, yaitu **operator perbandingan** dan **operator logika**. Kedua jenis operator ini akan sering digunakan dalam proses “query” data.

a. Operator Perbandingan

Table Operator Perbandingan

Operator Perbandingan	Keterangan
Lebih besar	>
Lebih kecil	<=
Lebih besar atau sama dengan	>=
Lebih kecil atau sama dengan	<=
Sama dengan	=
Tidak sama dengan	<>

b. Operator Logika

Table Operator Logika

Operator Logika	Keterangan
Dan	AND atau &&
Atau	OR atau
Lebih besar atau sama dengan	NOT atau !
Lebih kecil atau sama dengan	<=
Tidak sama dengan	<>

Berikut ini adalah penerapan dari kedua operator diatas : kita tampilkan data karyawan yang tanggal lahirnya sebelum tanggal 1 Januari 1980, dan tampilan data diurut berdasarkan nama. Cukup kolom nama, jenis kelamin, tanggal lahir yang ditampilkan.

```
mysql> select nama, jenkel, tgllahir
-> from karyawan
-> where tgllahir < "1980-01-01"
-> order by nama ;
```

Berikut hasil yang ditampilkan dari perintah diatas :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
+-----+
| nama      | jenkel | tgllahir |
+-----+-----+
| Ahmad Zobari | L      | 1977-10-02 |
| Andika      | L      | 1978-10-02 |
| Anwar       | L      | 1972-10-02 |
| Banowati    | P      | 1978-11-12 |
| Beno        | L      | 1978-08-10 |
| Dadan       | L      | 1975-10-02 |
| Dadang      | L      | 1977-08-10 |
| Donno       | L      | 1971-10-02 |
| Gunadi      | L      | 1978-08-10 |
| Maman       | L      | 1977-08-10 |
| Mardiatun   | P      | 1975-07-07 |
| Melia       | P      | 1979-11-12 |
| Miranti     | P      | 1975-07-07 |
| Nenny       | P      | 1972-06-09 |
| Rahmat      | L      | 1977-03-21 |
| Ratu        | P      | 1972-11-12 |
| Zobari      | L      | 1976-10-02 |
| Subur       | L      | 1977-10-02 |
| Sundariwati | P      | 1978-11-12 |
| Susilowati  | P      | 1973-11-12 |
| The Cute    | L      | 1977-03-21 |
| Wawan       | P      | 1971-11-12 |
| Yuliawati   | P      | 1974-06-09 |
| Zamzam      | L      | 1974-08-10 |
| Zukarman    | L      | 1978-08-10 |
+-----+-----+
25 rows in set (0.00 sec)
```

MySQL memiliki kelonggaran penulisan tanggal selama formatnya mengikuti aturan “tahun-bulan-tanggal”. Misal “1971-11-12” dapat ditulis 1971-11-12, atau 1971#11#12, atau 19711112, atau 711112.

Kita lihat contohnya dibawah ini dimana tanggal “1980-01-01” ditulis dengan 19800101 :

```
mysql> select nama, jenkel, tgllahir
-> from karyawan
-> where tgllahir < 19800101
-> and jenkel="L"
-> order by nama ;
+-----+-----+
| nama      | jenkel | tgllahir |
+-----+-----+
| Ahmad Zobari | L      | 1977-10-02 |
| Andika      | L      | 1978-10-02 |
| Anwar       | L      | 1972-10-02 |
| Beno        | L      | 1978-08-10 |
| Dadan       | L      | 1975-10-02 |
| Dadang      | L      | 1977-08-10 |
| Donno       | L      | 1971-10-02 |
| Gunadi      | L      | 1978-08-10 |
| Maman       | L      | 1977-08-10 |
| Rahmat      | L      | 1977-03-21 |
| Zobari      | L      | 1976-10-02 |
| Subur       | L      | 1977-10-02 |
| The Cute    | L      | 1977-03-21 |
| Zamzam      | L      | 1974-08-10 |
| Zukarman    | L      | 1978-08-10 |
+-----+-----+
15 rows in set (0.00 sec)
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Silahkan dicoba sendiri dengan menggunakan penulisan tanggal yang lain. (Perhatikan bahwa hasilnya sama walaupun penulisan tanggalnya berbeda, asalkan mengikuti format aturan tanggal dalam MySQL).

Sekarang kita tampilkan data karyawan yang tanggal lahirnya antara tanggal 1 Januari 1980 dan 31 Desember 1985, dan tampilan data diurut berdasarkan nama. Cukup hanya kolom nama, jenis kelamin dan tanggal lahir saja yang ditampilkan.

```
mysql> select nama, jenkel, tgllahir
-> from karyawan
-> where tgllahir >= "1980-01-01"
-> and tgllahir <= "1985-12-31"
-> order by nama ;
```

Berikut hasilnya :

nama	jenkel	tgllahir
Bambang	L	1982-03-21
Dian	P	1980-07-07
Gungun	L	1981-03-21
Mawar	P	1985-07-07
Melia	P	1981-07-07
Miranda	P	1980-07-07
Rohimat	L	1980-03-21
Ryan Cakep	L	1981-03-21
Yanti	P	1981-06-09
Yenny	P	1985-06-09
Yossy	P	1982-06-09
Yuliawati	P	1982-06-09
Zanda Cute	L	1980-03-21

13 rows in set (0.00 sec)

Sekarang kita lanjutkan dengan menampilkan semua data karyawan dengan usianya ada saat ini. Untuk masalah ini memang cukup panjang solusinya. Tidak apa-apa, kita coba saja ya. Disini kita memerlukan bantuan beberapa fungsi-fungsi yang sudah disediakan oleh MySQL. Kita lihat dulu ya :

```
mysql> select nama, tgllahir,
-> current_date AS SEKARANG,
-> (year(current_date) - year(tgllahir))
-> - (right(current_date,5) < right(tgllahir,5)) AS USIA
-> from karyawan ;
```

Berikut hasil dari perintah diatas :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

nama	tgllahir	SEKARANG	USIA
Ahmad Zobari	1977-10-02	2007-08-30	29
Sundariwati	1978-11-12	2007-08-30	28
Ryan Cakep	1981-03-21	2007-08-30	26
Zukarman	1978-08-10	2007-08-30	29
Yulawati	1982-06-09	2007-08-30	25
Mawar	1985-07-07	2007-08-30	22
Zobari	1976-10-02	2007-08-30	30
Melia	1979-11-12	2007-08-30	27
Zanda Cute	1980-03-21	2007-08-30	27
Maman	1977-08-10	2007-08-30	30
Yenny	1985-06-09	2007-08-30	22
Rossa	1987-07-07	2007-08-30	20
Dadan	1975-10-02	2007-08-30	31
Wawan	1971-11-12	2007-08-30	35
The Cute	1977-03-21	2007-08-30	30
Marpaung	1988-08-10	2007-08-30	19
Yono	1989-06-09	2007-08-30	18
Dian	1980-07-07	2007-08-30	27
Donno	1971-10-02	2007-08-30	35
Ratu	1972-11-12	2007-08-30	34
Bambang	1982-03-21	2007-08-30	25
Dadang	1977-08-10	2007-08-30	30
Yulawati	1974-06-09	2007-08-30	33
Miranda	1980-07-07	2007-08-30	27
Subur	1977-10-02	2007-08-30	29
Banowati	1978-11-12	2007-08-30	28
Gungun	1981-03-21	2007-08-30	26
Gunadi	1978-08-10	2007-08-30	29
Yossy	1982-06-09	2007-08-30	25
Melia	1981-07-07	2007-08-30	26
Anwar	1972-10-02	2007-08-30	34
Susilowati	1973-11-12	2007-08-30	33
Rahmat	1977-03-21	2007-08-30	30
Zamzam	1974-08-10	2007-08-30	33
Nenny	1972-06-09	2007-08-30	35
Mardiatun	1975-07-07	2007-08-30	32
Andika	1978-10-02	2007-08-30	28
Siti	1988-11-12	2007-08-30	18
Rohimat	1980-03-21	2007-08-30	27
Beno	1978-08-10	2007-08-30	29
Yanti	1981-06-09	2007-08-30	26
Miranti	1975-07-07	2007-08-30	32

42 rows in set (0.00 sec)

Kita lanjutkan dengan menampilkan data karyawan yang usianya sama atau dibawah 25 tahun. Nah bagaimana caranya ?

```
mysql> select nama, tgllahir,  
-> current_date AS SEKARANG,  
-> (year(current_date) - year(tgllahir))  
-> - (right(current_date,5) < right(tgllahir,5))  
-> AS USIA  
-> from karyawan  
-> where ((year(current_date) - year(tgllahir))  
-> - (right(current_date,5) < right(tgllahir,5)))  
-> <= 25 ;
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Berikut hasil dari perintah diatas :

```
-----+-----+-----+-----+
| nama      | tgllahir  | SEKARANG | USIA  |
+-----+-----+-----+-----+
| Yuliawati | 1982-06-09 | 2007-08-30 | 25   |
| Mawar     | 1985-07-07 | 2007-08-30 | 22   |
| Yenny     | 1985-06-09 | 2007-08-30 | 22   |
| Rossa     | 1987-07-07 | 2007-08-30 | 20   |
| Marpaung  | 1988-08-10 | 2007-08-30 | 19   |
| Yono      | 1989-06-09 | 2007-08-30 | 18   |
| Bambang   | 1982-03-21 | 2007-08-30 | 25   |
| Yossy     | 1982-06-09 | 2007-08-30 | 25   |
| Siti      | 1988-11-12 | 2007-08-30 | 18   |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

Cukup panjang perintahnya ya. Disini kita menggunakan fungsi **CURRENT_DATE** yang mengambil nilai dari tanggal saat ini pada sistem komputer anda. **YEAR** adalah fungsi yang mengambil nilai tahun. Kemudian **AS** adalah singkatan dari **Alias**, yang seolah-olah memberikan nama lain (alias name) pada kolom atau hasil suatu proses. Sedangkan **RIGHT** adalah fungsi yang mengambil nilai dari sekian karakter dari sisi kanan sebuah target. Misal : **RIGHT('APRI',3)** maka akan menghasilkan karakter 'PRI'.

Sekarang kita akan mencoba menampilkan karyawan yang kota kelahirannya di "Bandung" :

```
mysql> select * from karyawan
-> where kota="Bandung" ;
+----+-----+-----+-----+-----+-----+
| noid|nama      |jenkel|kota   |kodepos|tgllahir |gaji  |
+----+-----+-----+-----+-----+-----+
| 1  |Ahmad Zobari|L     |Bandung|41011  |1977-10-02|1000000|
| 2  |Sundariwati|P     |Bandung|40123  |1978-11-12|1250000|
| 8  |Melia      |P     |Bandung|40123  |1979-11-12|1200000|
| 13 |Dadan     |L     |Bandung|41011  |1975-10-02|1450000|
| 19 |Donno     |L     |Bandung|41011  |1971-10-02|1850000|
| 25 |Subur     |L     |Bandung|41011  |1977-10-02|2150000|
| 32 |Susilowati|P     |Bandung|40123  |1973-11-12|1125000|
| 37 |Andika    |L     |Bandung|41011  |1978-10-02|1725000|
+----+-----+-----+-----+-----+-----+
8 rows in set (0.03 sec)
```

Kita tampilkan karyawan yang kota kelahirannya **bukan** di Bandung :

```
mysql> select * from karyawan
-> where kota != "bandung" ;
```

Berikut hasil perintah diatas :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
+-----+-----+-----+-----+-----+-----+-----+
|noid| nama      |jenkel|kota      |kodepos| tglahir |gaji  |
+-----+-----+-----+-----+-----+-----+-----+
| 3| Ryan Cakep|L      |Jakarta   |12111  | 1981-03-21|1500000|
| 4| Zukarman  |L      |Bekasi    |17211  | 1978-08-10|1750000|
| 5| Yuliawati |P      |Bogor     |00000  | 1982-06-09|2000000|
| 6| Mawar     |P      |Bogor     |12345  | 1985-07-07|2250000|
| 7| Zobari    |L      |Jakarta   |41011  | 1976-10-02|1100000|
| 9| Zanda Cute|L      |Jakarta   |12111  | 1980-03-21|1300000|
|10| Maman     |L      |Bekasi    |17211  | 1977-08-10|1400000|
|11| Yenny     |P      |Bogor     |00000  | 1985-06-09|1150000|
|12| Rossa     |P      |Jakarta   |12345  | 1987-07-07|1350000|
|14| Wawan     |P      |Semarang  |40123  | 1971-11-12|1600000|
|15| The Cute  |L      |Jakarta   |12111  | 1977-03-21|1700000|
|16| Marpaung  |L      |Surabaya  |17211  | 1988-08-10|1800000|
|17| Yono      |P      |Bogor     |00000  | 1989-06-09|1900000|
|18| Dian      |P      |Jakarta   |12345  | 1980-07-07|1650000|
|20| Ratu      |P      |Yogyakarta|40123  | 1972-11-12|1950000|
|21| Bambang   |L      |Jakarta   |12111  | 1982-03-21|2100000|
|22| Dadang    |L      |Surabaya  |17211  | 1977-08-10|2200000|
|23| Yuliawati |P      |Bogor     |00000  | 1974-06-09|2300000|
|24| Miranda   |P      |Bogor     |12345  | 1980-07-07|2400000|
|26| Banowati  |P      |Malang    |40123  | 1978-11-12|2350000|
|27| Gungun    |L      |Jakarta   |12111  | 1981-03-21|2450000|
|28| Gunadi    |L      |Bekasi    |17211  | 1978-08-10|2125000|
|29| Yossy     |P      |Bogor     |00000  | 1982-06-09|2225000|
|30| Melia     |P      |Malang    |12345  | 1981-07-07|2325000|
|31| Anwar     |L      |Purwakarta|41011  | 1972-10-02|2425000|
|33| Rahmat    |L      |Jakarta   |12111  | 1977-03-21|1225000|

| 34| Zamzam    |L      |Bekasi    |17211  | 1974-08-10|1325000|
| 35| Nenny     |P      |Medan     |00000  | 1972-06-09|1425000|
| 36| Mardiatun |P      |Bogor     |12345  | 1975-07-07|1625000|
| 38| Siti      |P      |Medan     |40123  | 1988-11-12|1825000|
| 39| Rohimat   |L      |Jakarta   |12111  | 1980-03-21|1925000|
| 40| Beno      |L      |Bekasi    |17211  | 1978-08-10|1175000|
| 41| Yanti     |P      |Jakarta   |00000  | 1981-06-09|1275000|
| 42| Miranti   |P      |Medan     |12345  | 1975-07-07|1375000|
+-----+-----+-----+-----+-----+-----+-----+
34 rows in set (0.00 sec)
```

Perintah diatas juga dapat menggunakan tanda "<>", dan hasilnya tetap sama dengan diatas : (cek sendiri hasilnya)

```
mysql> select * from karyawan
-> where kota <> "bandung" ;
```

Sebagai latihan, silahkan anda coba perintah-perintah berikut kemudian jelaskan perbedaannya serta jelaskan tujuan dari perintah tersebut :

Perintah Pertama :

```
mysql> select * from karyawan
-> where kota <> "bandung"
-> and kota <> "Jakarta"
-> and kota <> "Bekasi"
-> order by kota ;
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Perintah Kedua :

```
mysql> select * from karyawan
-> where kota <> "bandung"
-> and kota <> "Jakarta"
-> and kota <> "Bekasi"
-> order by kota and nama ;
```

Perintah Ketiga :

```
mysql> select * from karyawan
-> where kota <> "bandung"
-> and kota <> "Jakarta"
-> and kota <> "Bekasi"
-> order by (kota and nama) ;
```

Perintah Keempat :

```
mysql> select * from karyawan
-> where kota <> "bandung"
-> and kota <> "Jakarta"
-> and kota <> "Bekasi"
-> order by kota, nama ;
```

Simpulkan jawaban anda berdasarkan hasil uji coba diatas !!!

2.5.2. Fungsi Statistik Dasar

Misalnya kita akan menampilkan data gaji karyawan antara Rp 1.500.000 dan Rp 2.500.000, dan diurutkan berdasarkan kolom gaji dan nama karyawan maka berikut perintah yang bisa digunakan :

```
mysql> select * from karyawan
-> where gaji >= 1500000
-> and gaji <= 2500000
-> order by gaji, nama ;
```

Untuk melihat berapa orang karyawan yang gajinya dibawah Rp 2.000.000, berikut perintahnya :

```
mysql> select count(*) from karyawan
-> where gaji < 2000000 ;
+-----+
| count(*) |
+-----+
|         29 |
+-----+
1 row in set (0.01 sec)
```

Rata-rata gaji karyawan :

```
mysql> select avg(gaji) from karyawan ;
+-----+
| avg(gaji) |
+-----+
| 1719642.8571 |
+-----+
1 row in set (0.41 sec)
```

Nilai gaji terbesar karyawan :

```
mysql> select max(gaji) from karyawan ;
+-----+
| max(gaji) |
+-----+
| 2450000 |
+-----+
1 row in set (0.00 sec)
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Jumlah gaji seluruh karyawan :

```
mysql> select sum(gaji) from karyawan ;
+-----+
| sum(gaji) |
+-----+
| 72225000 |
+-----+
1 row in set (0.00 sec)
```

Soal Latihan (Tugas 6) :

1. Buka tabel **peserta** :
 - a. Tambahkan field `jnsKursus` varchar (30) Not Null dan field `Biaya` INT(12) Not Null default 0
 - b. Isikan `jnsKursus` dan `Biaya` pada masing-masing record
2. Tampilkan seluruh data peserta dimana tanggal lahirnya sebelum 01 Januari 1985urut berdasarkan nama
3. Tampilkan seluruh data peserta yang berasal dari Kota Solo dan jenis kursus yang diambil adalah Aplikasi Perkantoran
4. Tampilkan data peserta yang mengambil jenis kursus Aplikasi Perkantoran atau Multimedia dan berasal dari Kota Solo
5. Tampilkan berapa jumlah peserta yang mengambil kursus Aplikasi Perkantoran
6. Tampilkan total pendapatan yang diterima dari biaya kursus.

Catatan : Perintah-perintah dari soal diatas ditulis tangan dan dikumpulkan pada pertemuan selanjutnya.

2.5.3. Operator Precedence

Operator precedence adalah tingkatan hirarki dalam memproses serangkaian operator.

Berikut tabel Operator Precedence

Tingkatan hirarki	Jenis Operator
Paling Tinggi	BINARY
	NOT !
	- (unary minus)
	* / %
	+ -
	<< >>
	&
	< <= <> != <> >= > IN IS LIKE
	REGEXP RLIKE
	BETWEEN
	AND &&
	Paling Rendah



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Semakin keatas posisi operator, maka semakin tinggi tingkat hirarki operator tersebut. Begitu pula sebaliknya, semakin rendah posisinya maka akan semakin lemah hirarkinya. Untuk operator yang sama kuat, misal + dan – digabung dengan operator * / %, maka akan ditentukan hirarkinya tergantung dari posisi mana yang paling kiri paling awal ditemukan. Dan untungnya posisi hirarki ini dapat diubah dengan bantuan tanda kurung (“(“ dan “)”). Sekarang kita lihat penerapannya :

```
mysql> select 10+15-11*2, (10+15-11)*2,
-> 2*6-5, 2*(6-5) ;
+-----+-----+-----+-----+
| 10+15-11*2 | (10+15-11)*2 | 2*6-5 | 2*(6-5) |
+-----+-----+-----+-----+
|          3 |             28 |      7 |         2 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Anda bisa perhatikan bahwa walaupun angka dan operatornya sama, tapi hasilnya bisa berbeda. Dan itu karena adanya peranan dari tanda kurung (“(“ dan “)”) yang akan mengubah peta posisi hirarki operator.

2.5.4. Operator LIKE, NOT LIKE, REGEXP

Ketiga operator ini akan banyak kita gunakan dalam operasi karakter.

A. Operator LIKE

Operator **LIKE** digunakan untuk mencari data yang “**menyerupai**” atau “**hampir sama**” dengan kriteria tertentu. Biasanya untuk mencari data string teks/teks. Simbol “%” digunakan untuk membantu pelaksanaan operator LIKE. Posisi “%” sangat berpengaruh dalam menentukan kriteria.

Misalnya, kita akan menampilkan data karyawan yang namanya **berawalan** huruf “a” : (perhatikan posisi simbol persennya “%”)

```
mysql> select noid, nama
-> from karyawan
-> where nama LIKE "a%" ;
+-----+-----+
| noid | nama      |
+-----+-----+
|    1 | Ahmad Zobari |
|   31 | Anwar      |
|   37 | Andika     |
+-----+-----+
3 rows in set (0.00 sec)
```

Tampilkan data karyawan yang namanya **berawalan** huruf “d” :

```
mysql> select noid, nama
-> from karyawan
-> where nama LIKE "d%" ;
+-----+-----+
| noid | nama      |
+-----+-----+
|   13 | Dadan     |
|   18 | Dian      |
|   19 | Donno     |
|   22 | Dadang    |
+-----+-----+
4 rows in set (0.00 sec)
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Tampilkan data karyawan yang namanya berakhiran huruf "i". Perhatikan posisi penulisan tanda "%":

```
mysql> select noid, nama
-> from karyawan
-> where nama LIKE "%i" ;
+-----+-----+
| noid | nama          |
+-----+-----+
| 1    | Ahmad Zobari  |
| 2    | Sundariwati  |
| 5    | Yulawati     |
| 7    | Zobari       |
| 23   | Yulawati     |
| 26   | Banowati     |
| 28   | Gunadi       |
| 32   | Susilowati   |
| 38   | Siti         |
| 41   | Yanti        |
| 42   | Miranti      |
+-----+-----+
11 rows in set (0.00 sec)
```

Bagaimana caranya agar operator LIKE dapat membedakan huruf besar dan kecil? Caranya adalah dengan menambahkan kata BINARY setelah perintah LIKE (sehingga perintahnya menjadi **LIKE BINARY**). Kita perhatikan contoh berikut :

```
mysql> select noid, nama
-> from karyawan
-> where nama LIKE BINARY "a%" ;
Empty set (0.34 sec)
```

Kenapa hasilnya menjadi "Empty set"? Kita coba dengan mengubah perintah tadi menjadi :

```
mysql> select noid, nama
-> from karyawan
-> where nama LIKE BINARY "A%" ;
+-----+-----+
| noid | nama          |
+-----+-----+
| 1    | Ahmad Zobari  |
| 31   | Anwar         |
| 37   | Andika        |
+-----+-----+
3 rows in set (0.00 sec)
```

Ya, dengan menggunakan LIKE BINARY, penulisan huruf "a" akan dibedakan artinya dengan "A".

Sedangkan, untuk menampilkan data dengan kriteria bukan diawal atau diakhir kalimat, tapi berada diantara sebuah kata/kalimat dapat menggunakan perintah berikut ini :

```
mysql> select noid, nama
-> from karyawan
-> where nama LIKE BINARY "%lia%" ;
+-----+-----+
| noid | nama          |
+-----+-----+
| 5    | Yulawati     |
| 8    | Melia        |
| 23   | Yulawati     |
| 30   | Melia        |
+-----+-----+
4 rows in set (0.00 sec)
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

B. Operator NOT LIKE

Kebalikan dari LIKE

C. Operator REGEXP

Operator REGEXP (singkatan **REG**ular **EXP**ressions) merupakan bentuk lain dari operator LIKE, dengan fungsi yang lebih disempurnakan. Operator REGEXP biasanya ditemani juga dengan simbol-simbol tertentu dalam melaksanakan tugasnya, seperti :

Tabel Simbol Operator REGEXP

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

Contoh :

Kita akan menampilkan nama karyawan yang berawalan huruf 'a' :

```
mysql> select noid, nama
-> from karyawan
-> where nama REGEXP "^a" ;
+-----+-----+
| noid | nama      |
+-----+-----+
| 1    | Ahmad Zobari |
| 31   | Anwar      |
| 37   | Andika     |
+-----+-----+
3 rows in set (0.00 sec)
```

Menampilkan nama karyawan yang berawalan huruf 'a-d':

```
mysql> select noid, nama
-> from karyawan
-> where nama REGEXP "[a-d]"
-> order by nama ;
+-----+-----+
| noid | nama      |
+-----+-----+
| 1    | Ahmad Zobari |
| 37   | Andika     |
| 31   | Anwar      |
| 21   | Bambang    |
| 26   | Banowati   |
| 40   | Beno       |
| 13   | Dadan      |
| 22   | Dadang     |
| 18   | Dian       |
| 19   | Donno      |
+-----+-----+
10 rows in set (0.00 sec)
```



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Menampilkan nama karyawan yang panjang namanya 10 karakter :

```
mysql> select noid, nama
-> from karyawan
-> where nama REGEXP "^.....S" ;

+-----+-----+
| noid | nama      |
+-----+-----+
| 3    | Ryan Cakep |
| 9    | Zanda Cute |
| 32   | Susilowati |
+-----+-----+
2 rows in set (0.00 sec)
```

Atau dengan cara lain :

```
mysql> select noid, nama
-> from karyawan
-> where nama REGEXP "^.{10}$" ;

+-----+-----+
| noid | nama      |
+-----+-----+
| 3    | Ryan Cakep |
| 9    | Zanda Cute |
| 32   | Susilowati |
+-----+-----+
3 rows in set (0.00 sec)
```

Soal Latihan :

1. Buka tabel **peserta**
2. Tampilkan nama peserta yang diawali dengan huruf A
3. Tampilkan nama peserta yang diawali dengan huruf A dan diakhiri dengan huruf i
4. Tampilkan nama peserta yang berawalan d-g dan panjang karakternya 10

2.6. Database Relasi (RDBMS)

2.6.1. Model Database

Model database adalah suatu konsep yang terintegrasi dalam menggambarkan hubungan (relationship) antar data dan batasan-batasan (constraint) data dalam suatu sistem database. Model data yang paling umum berdasarkan pada bagaimana hubungan antar record dalam database (record base data models), terdapat 3 jenis yaitu :

- a. Model database hirarki (hierarchical database model)
- b. Model database jaringan (network database model)
- c. Model database relasi (relationship database model)

Model database hirarki dan jaringan merupakan model database yang tidak banyak dipakai saat ini karena adanya berbagai kelemahan dan hanya cocok untuk struktur hirarki dan jaringan saja. Artinya tidak mengakomodir



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

untuk berbagai macam jenis persoalan dalam suatu sistem database. Yang paling banyak dipakai saat ini model database relasi, karena mampu mengakomodir berbagai permasalahan dalam sistem database. Berikut keterangan tentang model database ini.

2.6.2. Model Database Relasi

Model database relasi merupakan model database yang paling banyak digunakan saat ini, karena paling sederhana dan mudah digunakan serta yang paling penting adalah kemampuannya dalam mengakomodasi berbagai kebutuhan pengolahan database. Sebuah database dalam model ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (record) dan kolom (field), pertemuan antara baris dengan kolom disebut item data (data value), tabel-tabel yang ada dihubungkan (relationship) sedemikian rupa menggunakan field-field kunci (primary key) sehingga dapat meminimalkan duplikasi data.

Model database relasi ini dikemukakan pertama kali oleh E.F. Codd, salah seorang pakar dalam bidang database. Sering juga model ini disebut Database Relasi.

2.6.3. Tingkatan Data dalam Database Relasi

Dalam suatu sistem database relasi, data yang tersimpan dalam DBMS mempunyai tingkatan-tingkatan sebagai berikut :

a. Karakter (Characters)

Merupakan bagian terkecil dalam database, dapat berupa karakter numerik (angka 0-9), huruf (A-Z, a-z) ataupun karakter-karakter khusus seperti “,&,%,# dan lain-lain.

b. Field atau Atribute

Merupakan bagian dari record yang menunjukkan suatu item data yang sejenis, misalnya : field nama, file NIM dan lain sebagainya. Setiap field harus mempunyai nama dan tipe data tertentu. Isi data field disebut data value. Dalam tabel database, field ini disebut juga kolom.

c. Record atau Tuple

Adalah kumpulan data value dari atribte yang berkaitan sehingga dapat menjelaskan sebuah entity secara lengkap. Misal : record entity mahasiswa adalah kumpulan data value dari field nim, nama, jurusan dan alamat per-barisnya. Dalam tabel database, record disebut juga baris.

d. Table/Entity

Entity merupakan sesuatu yang dapat diidentifikasi dari suatu sistem database, bisa berupa objek, orang, tempat, kejadian atau konsep yang informasinya akan disimpan di database. Misal pada sistem database akademik, yang menjadi entity adalah mahasiswa, dosen, mata kuliah, dan lain-lain.

e. Database

Kumpulan dari tabel-tabel yang saling berelasi, disusun secara logis, sehingga menghasilkan informasi yang bernilai guna dalam proses pengambilan keputusan.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

2.6.4. Sifat-sifat Yang Melekat Pada Suatu Tabel

- ✚ Tidak boleh ada record yang sama (kembar)
- ✚ Urutan record tidak terlalu penting, karena data dalam record dapat diurut sesuai dengan kebutuhan
- ✚ Setiap field harus mempunyai nama yang unik (tidak boleh ada yang sama)
- ✚ Setiap field mesti mempunyai tipe data dan karakteristik tertentu

2.6.5. Jenis Hubungan Antar Tabel

Jenis hubungan antar tabel dalam model database relasi, juga didefinisikan dengan :

1. Satu ke satu (one to one)

Relasi tabel one to one (satu-satu) adalah relasi 2 tabel dengan primary key (PK) dan foreign key (FK). Ini dilakukan dengan meletakkan kolom one to one ke tabel baru. Sebetulnya relasi ini jarang digunakan. Namun ada beberapa alasan relasi ini digunakan :

- a. Memindahkan data ke tabel lain memungkinkan untuk membuat query yang lebih cepat
- b. Mengisolasi dan menghindarkan nilai NULL pada tabel utama
- c. Membuat sebagian data susah diakses

Contoh tabel :

```
Tbl_Pegawai
- id_pegawai (PK)
- nama_depan
- nama_belakang

Tbl_Gaji
- kode_asuransi (PK)
- level_gaji
- id_pegawai (FK)
```

Tabel karyawan dan gaji didesain one to one untuk mendapatkan keuntungan :

- a. Query untuk mendapatkan kode_asuransi lebih cepat
- b. Data gaji lebih aman

2. Satu ke banyak (one to many)

Artinya satu record pada entity A berelasi dengan beberapa record ada entity B, tapi tidak sebaliknya, setiap record pada entity B berelasi paling banyak satu record dengan entity A. Dalam diagram E-R, relasi ini disimbolkan dengan angka 1 untuk menyatakan satu dan huruf M untuk menyatakan banyak.

Contoh : dalam proses belajar mengajar di sekolah dasar misalnya, satu orang guru mengajar beberapa (banyak) murid, tetapi satu kelas (beberapa murid) hanya diajar oleh satu guru.



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

3. Banyak ke banyak (many to many)

Artinya beberapa record pada entity A dapat berelasi dengan beberapa record juga pada entity B, begitu juga sebaliknya, beberapa record pada entity B dapat berelasi dengan beberapa record juga pada entity A. dalam diagram E-R, relasi ini disimbolkan dengan huruf M atau N untuk menyatakan banyak.

Contoh : dalam hubungan antara mahasiswa dengan dosen pada perguruan tinggi, yaitu seorang dosen mengajar banyak mahasiswa, sebaliknya seorang mahasiswa dapat diajar oleh beberapa dosen sehingga terjadi hubungan banyak ke banyak.

2.6.6. Relasi Database dengan MySQL

MySQL mempunyai kemampuan untuk menggabungkan dua tabel atau lebih guna mendapatkan informasi yang diinginkan. Proses yang dilakukan dengan nama JOIN. Dalam penggabungan dari beberapa (join) ada beberapa hal yang perlu diperhatikan antara lain :

1. Setiap kolom disebutkan dengan bentuk
2. Tabel-tabel yang dilibatkan dalam query perlu disebutkan dalam klausa FROM dengan antar tabel dipisah oleh koma
3. Kondisi dalam WHERE menentukan macam join yang terbentuk

Misalnya : kita akan membuat sistem penjualan barang untuk marketing feelance di perusahaan Oryn Textile. Barang berupa kaos kaki @ Rp 3.500

1. Kita buat database 'dbOryn'
2. Kita buat dua tabel sebagai berikut :
 - a. Tabel rbMarketing (untuk mencatat data marketing)
 - b. Tabel tbJual (untuk mencatat penjualan barang)
3. Berikut struktur tabelnya :

Tabel tbMarketing

Field	Type	Null	Key	Default	Extra
NoID	char(4)	NO	PRI	NULL	
Nama	varchar(20)	NO		NULL	
NoHP	varchar(12)	NO		NULL	
Alamat	varchar(20)	NO		NULL	

Tabel tbJual

Field	Type	Null	Key	Default	Extra
NoJual	char(4)	NO	PRI	NULL	
NoID	char(4)	NO		NULL	
TglJual	date	NO		NULL	
Quantity	smallint(3)	NO		NULL	

4. Isikan data pada tabel seperti berikut :



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Tabel tbMarketing

NoID	Nama	NoHP	Alamat
M001	Dwi Apri Setyorini	081888777666	Gumpang Kartasura
M002	Pipin Widyaningsih	081555666777	Gebyok Kartasura
M003	Haryanto	081222444666	Gedongan Solo
M004	Wijiyanto	081393837360	Tipes Solo
M005	Sri Sumarlinda	081999555222	Colomadu Solo

Tabel tbJual

NoJual	NoID	TglJual	Quantity
J001	M001	2010-01-13	25
J002	M002	2010-01-13	20
J003	M004	2010-01-13	30
J004	M006	2010-01-13	15

Jika kita perhatikan di tabel tbJual, tidak menginformasikan nama Marketing yang melakukan penjualan. Mari kita coba menuliskan perintah SQL untuk menampilkan NoJual, NoID, Nama, TglJual dan Quantity, tentunya dengan menggunakan relasi. Berikut perintah SQL yang digunakan :

```
mysql> SELECT NoJual, NoID, Nama, TglJual, Quantity
-> From tbMarketing, tbJual
-> Where tbMarketing.NoID=tbJual.NoID;
ERROR 1052 (23000): Column 'NoID' in field list is ambiguous
```

Perhatikan hasil diatas, terdapat ERROR 1052...dst. Kenapa terjadi seperti ini? Disebutkan bahwa terdapat field NoID ambiguous. Field NoID dimiliki oleh dua tabel yaitu tbMarketing dan tbJual, sehingga terdapat data ambigu karena kita tidak menjelaskan field NoID tersebut berasal dari tabel yang mana.

Solusinya

```
mysql> SELECT NoJual, tbMarketing.NoID, Nama, TglJual, Quantity
-> From tbMarketing, tbJual
-> Where tbMarketing.NoID=tbJual.NoID;
+-----+-----+-----+-----+-----+
| NoJual | NoID | Nama                | TglJual | Quantity |
+-----+-----+-----+-----+-----+
| J001   | M001 | Dwi Apri Setyorini | 2010-01-13 | 25 |
| J002   | M002 | Pipin Widyaningsih | 2010-01-13 | 20 |
| J003   | M004 | Wijiyanto           | 2010-01-13 | 30 |
+-----+-----+-----+-----+-----+
3 rows in set (0.08 sec)
```

Perhatikan hasil/tabel diatas pada judul kolom Nama. Jika anda menginginkan agar pada saat hasil ditampilkan judul kolom **Nama** diperjelas dengan diganti menjadi **Nama Marketing**, maka MySQL menyediakan perintah **AS** (kepanjangan **Alias**) untuk mengganti judul kolom pada saat ditampilkan (catatan : perintah AS tidak akan merubah struktur tabel, jadi nama field pada tbMarketing tetap **Nama**).



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

```
mysql> SELECT NoJual, tbMarketing.NoID, Nama AS Nama_Marketing,
-> TglJual, Quantity
-> From tbMarketing, tbJual
-> Where tbMarketing.NoID=tbJual.NoID;
```

NoJual	NoID	Nama_Marketing	TglJual	Quantity
J001	M001	Dwi Apri Setyorini	2010-01-13	25
J002	M002	Pipin Widyaningsih	2010-01-13	20
J003	M004	Wijiyanto	2010-01-13	30

3 rows in set (0.00 sec)

Dari data tersebut, dapat kita lihat bahwa pada tabel tbMarketing, tidak terdapat record dengan NoID M006, dan begitu pula tabel tbJual tidak terdapat record dengan NoID M003 dan M005. Apabila dilakukan join seperti perintah diatas maka beberapa record tidak akan tampil seperti yang kita mau.

```
mysql> SELECT NoJual, tbMarketing.NoID, Nama, TglJual, Quantity
-> From tbMarketing, tbJual
-> Where tbMarketing.NoID=tbJual.NoID;
```

Hanya akan menampilkan :

```
+-----+-----+-----+-----+-----+
| NoJual | NoID | Nama          | TglJual | Quantity |
+-----+-----+-----+-----+-----+
| J001   | M001 | Dwi Apri Setyorini | 2010-01-13 | 25 |
| J002   | M002 | Pipin Widyaningsih | 2010-01-13 | 20 |
| J003   | M004 | Wijiyanto      | 2010-01-13 | 30 |
+-----+-----+-----+-----+-----+
3 rows in set (0.08 sec)
```

Lalu dimana yang memiliki ID M003 dan M005? Untuk itulah ada beberapa koin khusus. Kondisi Where menentukan macam join yang terbentuk. Macam-macam bentuk penggabungan (Join) :

1. CROSS JOIN, Cross join merupakan bentuk penggabungan yang paling sederhana tanpa ada kondisi
Bentuk umum : SELECT field1,field2 FROM tabel1 CROSS JOIN tabel2;
2. INNER JOIN
Hampir sama dengan cross join tetapi diikuti dengan kondisi
Bentuk Umum : SELECT field FROM tabel1 INNER JOIN tabel2 ON kondisi
3. STRAIGHT JOIN
Straight Join identik dengan inner join tetapi tidak mengenal klausa where
Bentuk Umum : SELECT field FROM Tabel1 STRAIGHT JOIN tabel2
4. LEFT(OUTER) JOIN
Akan menampilkan tabel disebelah kanannya dengan NULL jika tidak terdapat hubungan antara tabel disebelah kiri
Bentuk Umum : SELECT field FROM tabel1 LEFT JOIN tabel2 ON kondisi
5. RIGHT (OUTER) JOIN
Kebalikan dari LEFT JOIN



Diktat Kuliah Pemrograman II

S1 Informatika

STMIK Bumigora Mataram

Bentuk Umum : SELECT field FROM tabel1 RIGHT JOIN tabel2 ON kondisi

Contoh :

Penggunaan LEFT JOIN

```
mysql> SELECT * FROM tbMarketing LEFT JOIN tbJual
-> ON tbMarketing.NoID=tbJual.NoID;
```

NoID	Nama	NoHP	Alamat	NoJual	NoID	TglJual	Quantity
M001	Dwi Apri Setyorini	081888777666	Gumpang Kartasura	J001	M001	2010-01-13	25
M002	Pipin Widyaningsih	081555666777	Gebyok Kartasura	J002	M002	2010-01-13	20
M003	Haryanto	081222444666	Gedongan Solo	NULL	NULL	NULL	NULL
M004	Wijiyanto	081393837360	Tipes Solo	J003	M004	2010-01-13	30
M005	Sri Sumarlinda	081999555222	Colomadu Solo	NULL	NULL	NULL	NULL

Penggunaan RIGHT JOIN

```
mysql> SELECT * FROM tbMarketing RIGHT JOIN tbJual
-> ON tbMarketing.NoID=tbJual.NoID;
```

NoID	Nama	NoHP	Alamat	NoJual	NoID	TglJual	Quantity
M001	Dwi Apri Setyorini	081888777666	Gumpang Kartasura	J001	M001	2010-01-13	25
M002	Pipin Widyaningsih	081555666777	Gebyok Kartasura	J002	M002	2010-01-13	20
M004	Wijiyanto	081393837360	Tipes Solo	J003	M004	2010-01-13	30
NULL	NULL	NULL	NULL	J004	M006	2010-01-13	15

Note : penggunaan SELECT * pada LEFT atau RIGHT join akan menampilkan column yang redundansi. Jadi ?? Kita lanjut...kan..

Mari kita bandingkan dengan penggunaan NATURAL JOIN

```
mysql> SELECT * FROM tbMarketing NATURAL JOIN tbJual;
```

NoID	Nama	NoHP	Alamat	NoJual	TglJual	Quantity
M001	Dwi Apri Setyorini	081888777666	Gumpang Kartasura	J001	2010-01-13	25
M002	Pipin Widyaningsih	081555666777	Gebyok Kartasura	J002	2010-01-13	20
M004	Wijiyanto	081393837360	Tipes Solo	J003	2010-01-13	30

Ternyata sama dengan query yang pertama. Jadi bisa dikatakan bahwa ini penyederhanaan query pertama, namun menghapus column yang redundansi.

Soal Latihan :

1. Buka database dbKursus
2. Tambahkan :
 - a. Tabel tutor untuk menyimpan data tutor yang mengajar
 - b. Tabel jenis untuk menyimpan jenis kursus yang ditawarkan
 - c. Tabel transaksi untuk menyimpan transaksi pembayaran kursus
 - d. Tabel jadwal untuk menyimpan jadwal kursus
3. Tampilkan data peserta beserta jenis kursus yang diambil
4. Tampilkan data tutor beserta jenis kursus yang diampu
5. Tampilkan jenis kursus apa saja yang ditawarkan pada hari Senin
6. Tampilkan total pendapatan yang diterima untuk jenis kursus Aplikasi Perkantoran selama bulan April 2009