

# Modul

# 1

## PENGENALAN DAN LINGKUNGAN KERJA SQL

### KOMPETENSI

Setelah menyelesaikan modul ini, mahasiswa mengenal, mengetahui dan mampu melakukan instalasi tools pemrograman SQL

#### 1. TUJUAN

Setelah menyelesaikan modul ini, anda diharapkan :

- a. Mengenal, mengetahui lingkungan kerja perangkat lunak sistem manajemen basis data
- b. Mampu melakukan instalasi tools pemrograman Basis Data MySQL/SQL Yog/XAMPP\*
- c. Mengenal perintah – perintah sederhana di MySQL/SQL Yog/XAMPP\*

#### 2. TUGAS PENDAHULUAN

- a. Apa yang anda ketahui tentang MySQL?
- b. Apa perbedaan instalasi MySQL pada operating system Windows dan Linux?
- c. Sebutkan operator dan fungsi pada MySQL!
- d. Apa yang anda ketahui tentang XAMPP, SQL Yog dan MySQL?jelaskan!

#### 3. DASAR TEORI

##### *Pengenalan Basis Data, DBMS, RDBMS*

**Basis data** adalah kumpulan data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga dapat diolah menjadi informasi yang disimpan pada media elektronik. Sedangkan untuk mengelola basis data diperlukan perangkat lunak **DBMS** (Database Management System) yang merupakan suatu sistem perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses basis data secara praktis dan efisien.

Salah satu DBMS yang mendukung adanya relationship antar tabel suatu basis data adalah **RDBMS**. Selain RDBMS, terdapat pula jenis DBMS lain antara lain Hierarchy DBMS, Object Oriented DBMS, dsb. Beberapa software atau perangkat lunak DBMS yang sering digunakan dalam aplikasi program antara lain :

- DB2 - <http://www-306.ibm.com/software/data/db2/>
- Microsoft SQL Server - <http://www.microsoft.com/sql/>
- Oracle - <http://www.oracle.com>
- Sybase - <http://www.sybase.com/>
- Interbase - <http://www.borland.com/interbase>
- Teradata - <http://www.teradata.com/>
- Firebird - <http://www.firebirdsql.org/>
- MySQL – <http://www.mysql.com>
- PostgreSQL - <http://www.postgresql.org/>

### ***MySQL***

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang perangkat lunak dan konsultan basis data bernama MySQL AB yang berada di Swedia. MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. Fitur-fitur MySQL antara lain :

- Relational Database System. Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.
- Arsitektur Client-Server. MySQL memiliki arsitektur client-server dimana server database MySQL terinstal di server. Client MySQL dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet.
- Mengenal perintah SQL standar. SQL (Structured Query Language) merupakan suatu bahasa standar yang berlaku di hampir semua software database. MySQL mendukung SQL versi SQL:2003.
- Mendukung Sub Select. Mulai versi 4.1 MySQL telah mendukung select dalam select (sub select).
- Mendukung Views. MySQL mendukung views sejak versi 5.0

- Mendukung Stored Prosedured (SP). MySQL mendukung SP sejak versi 5.0
- Mendukung Triggers. MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1.
- Mendukung replication.
- Mendukung transaksi.
- Mendukung foreign key.

#### Format Perintah MySQL

Berikut merupakan beberapa ketentuan untuk memberi perintah pada MySQL antara lain :

- Setiap perintah harus diakhiri dengan tanda titik koma, kecuali untuk perintah tertentu misal exit.
- Setiap perintah akan disimpan dalam buffer (memori sementara) untuk menyimpan histori perintah – perintah yang pernah diberikan.
- Perintah dapat berupa perintah SQL atau perintah khusus MySQL.
- Perintah – perintah dalam lingkungan MySQL tidak menerapkan aturan case sensitive, tetapi insensitive yaitu perintah bisa dituliskan dalam huruf besar ataupun huruf kecil.
- Aturan case sensitive diterapkan pada penamaan objek – objek dalam basis data seperti nama basis data atau tabel.

Beberapa tanda yang sering muncul pada prompt :

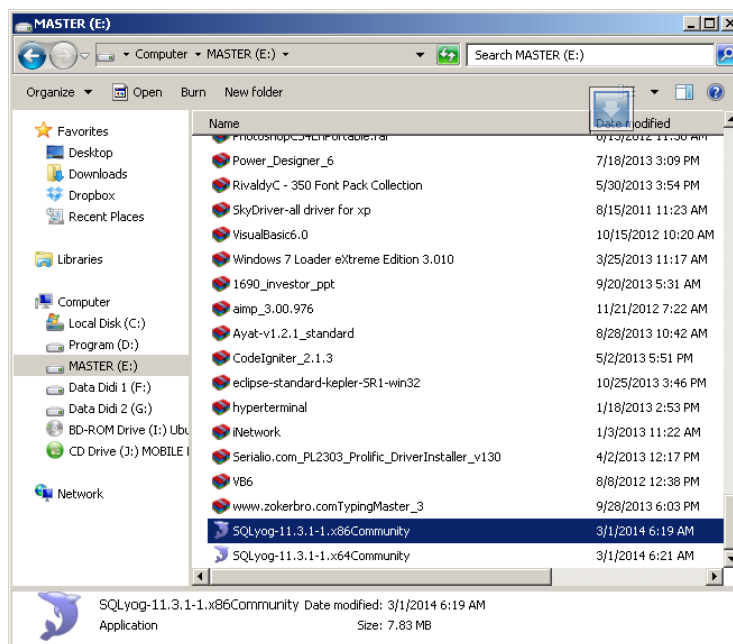
Prompt	Arti
mysql>	Siap menerima perintah baru
->	Menunggu baris berikut untuk perintah yang lebih dari satu baris
'>	Menunggu baris berikut, menunggu penutup string yang dimulai dengan tanda kutip satu (~'~)
">	Menunggu baris berikut, menunggu penutup string yang dimulai dengan tanda kutip dua (~"~)
`>	Menunggu baris berikutnya, menunggu penutup identifier yang dimulai dengan tanda backtick ("`~)

## SQLYog dan XAMPP

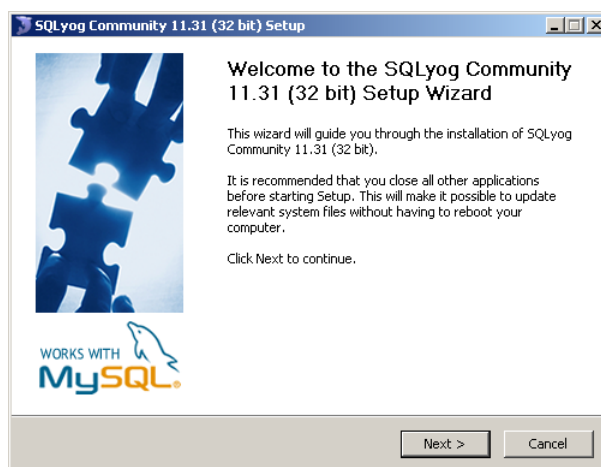
SQLYog adalah aplikasi client MySQL yang sangat populer digunakan di Indonesia. Aplikasi ini memiliki banyak fitur yang memudahkan pengguna melakukan administrasi maupun melakukan pengolahan data MySQL. (<http://mysql.phy-integration.com/mysql-client/sqlyog>).

### Instalasi SQLYog

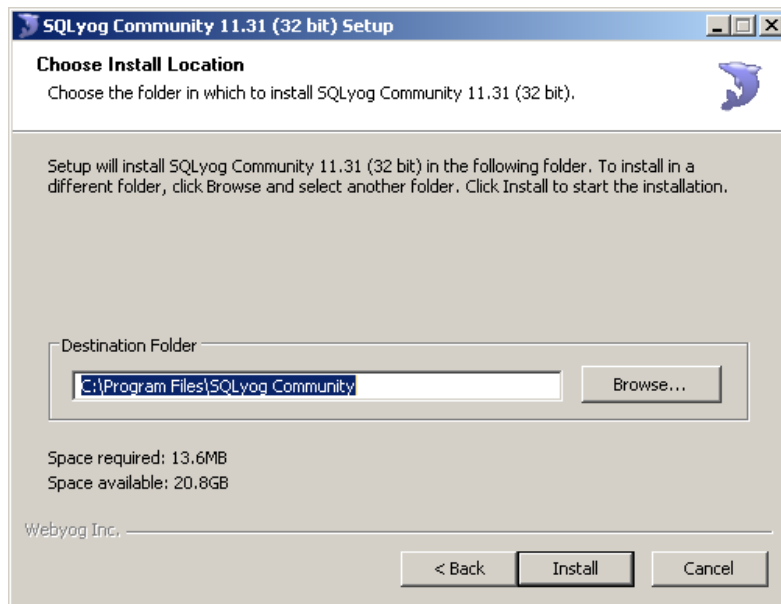
- Double klik file installer SQLYog (sesuaikan dengan spesifikasi laptop anda)



- Ikuti langkah petunjuk instalasi



- Pilih lokasi hardisk untuk menempatkan program installer SQLYog



d. Tunggu hingga proses instalasi selesai

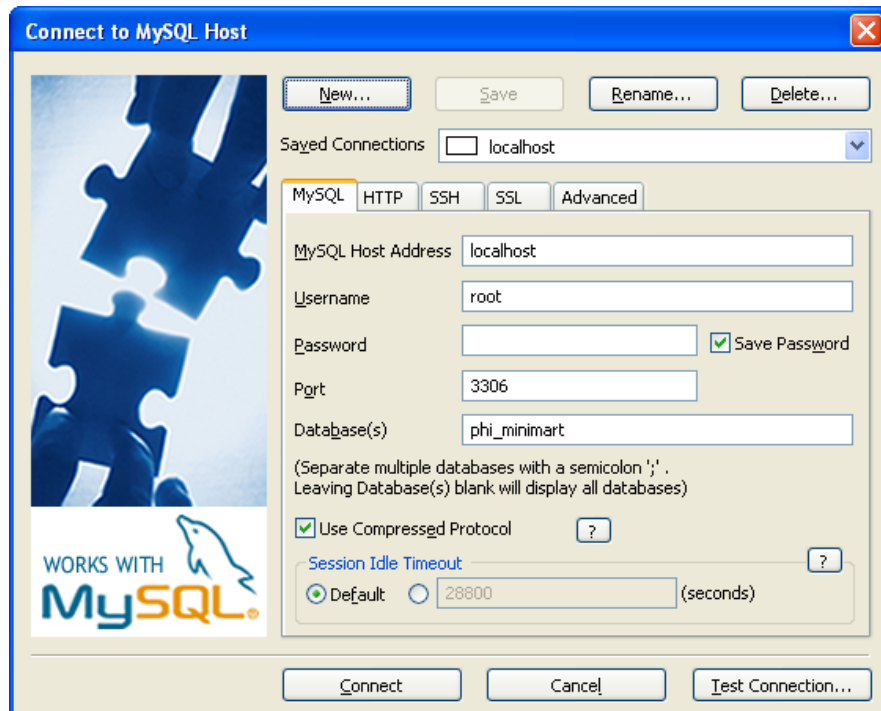


e. Selesai

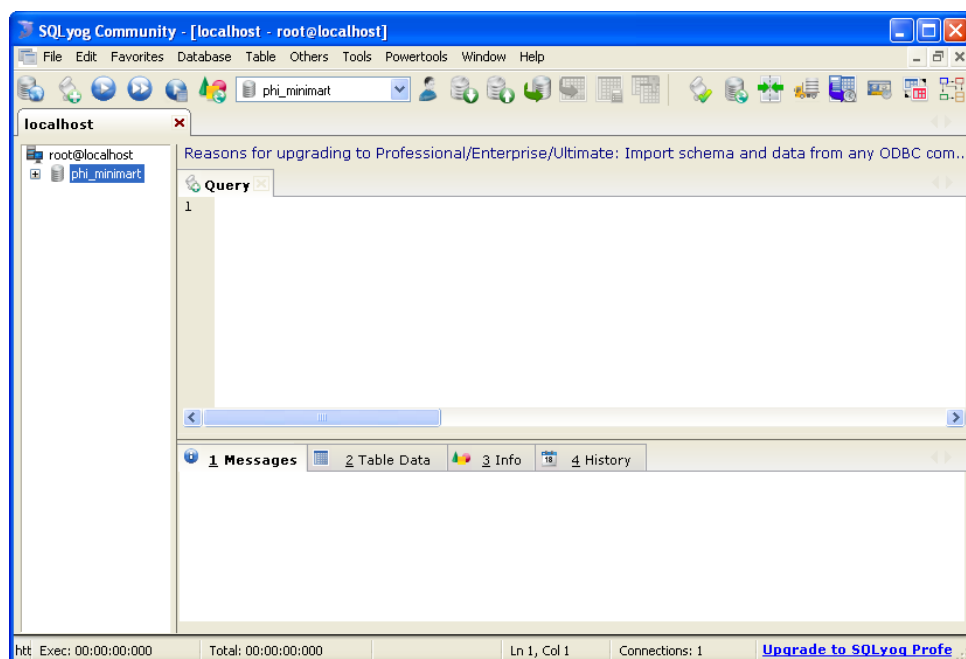
Berikut adalah contoh penggunaan koneksi ke host MySQL dan melihat objek-objek database yang ada dengan mudah :

- Jalankan SQLyog.
- Jika ada komentar "Upgrade to Professional/Enterprise/Ultimate edition now", klik tombol "Continue..." saja.

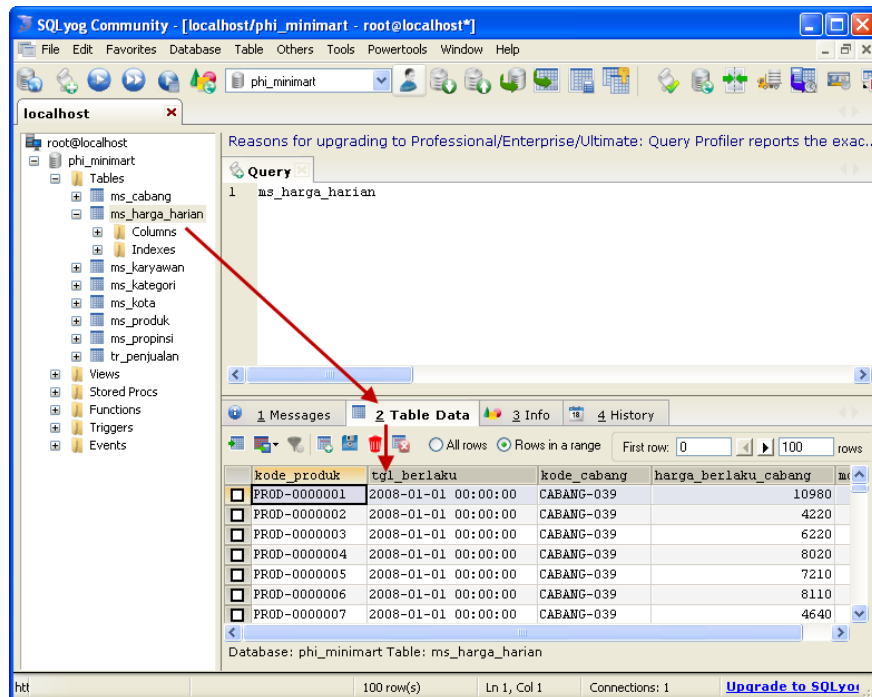
- Pada dialog "**Connect to MySQL host**" klik tombol "**New**", namakan koneksi tersebut misalkan dengan "**localhost**"
- Masukkan rincian koneksi dan klik tombol "**Connect**"



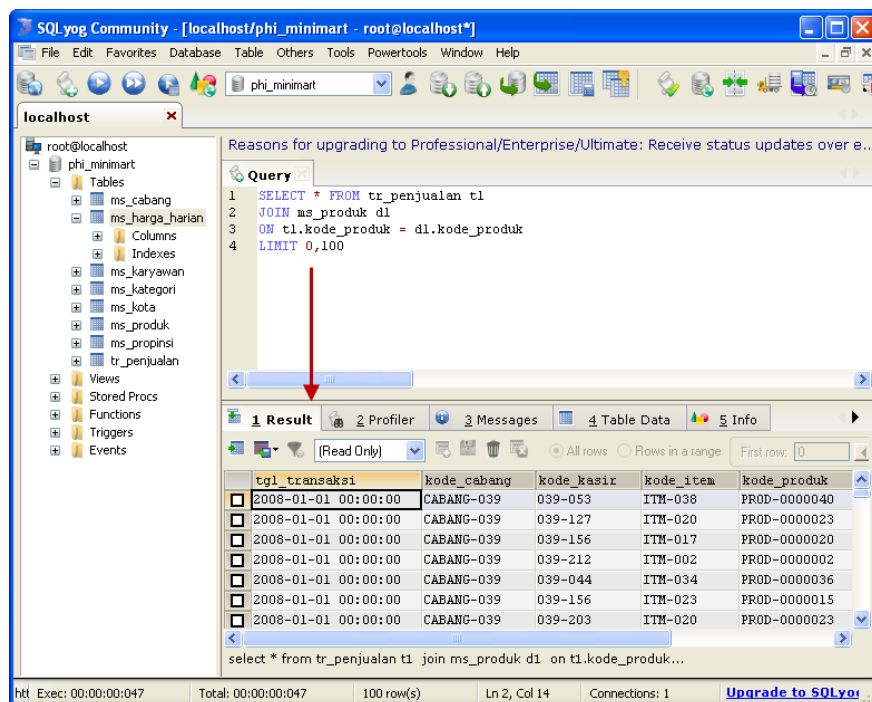
- Jika koneksi berhasil akan muncul layar kerja seperti gambar di bawah ini dengan panel kiri adalah bagian "**Object Browser**" yang berisi daftar informasi dari semua objek database kita.



- Klik salah satu table dari database kita, kemudian klik tab "Table Data" pada panel kanan bawah. Row-row data dari table tersebut akan segera terbaca.



- Eksekusi query di panel kanan atas, tekan F9 (Execute) dan hasilnya dapat dilihat di panel kanan bawah.



XAMPP adalah merupakan perangkat lunak web server apache yang didalamnya sudah menyediakan paket perangkat lunak lain ke dalam satu buah paket yaitu database server MySQL dan mendukung pemrograman PHP. XAMPP merupakan software yang mudah digunakan, dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5) dan beberapa module lainnya. Dulu XAMPP untuk Linux dinamakan LAMP, sekarang diganti namanya menjadi XAMPP FOR LINUX.

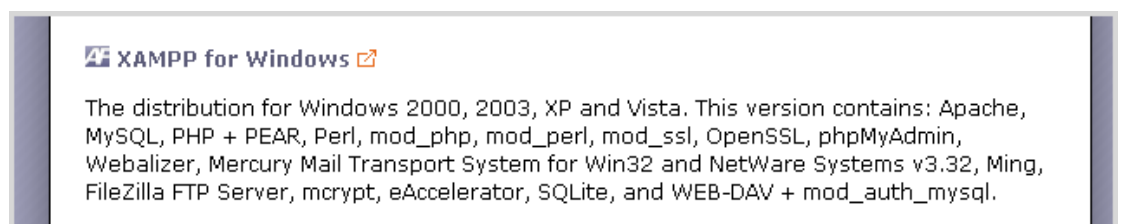


Gambar 1. Logo XAMPP

Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstallasi dan mengkonfigurasikannya secara otomatis.

#### **Download dan Instalasi Paket XAMPP**

- Kunjungi halaman situs XAMPP di <http://www.apachefriends.org/en/xampp.html>
- Cari dan klik link paket "XAMPP for Windows"



- Cari distribusi "XAMPP Lite", distribusi ini tidak terlalu banyak aplikasi dan sudah cukup untuk kebutuhan belajar MySQL. Pada saat penulisan dibuat versi terakhir adalah 1.7.3 (atau pilih versi terbarunya). Ada dua pilihan disana, yaitu zip dan exe, pilih file zip walaupun agak besar filenya.

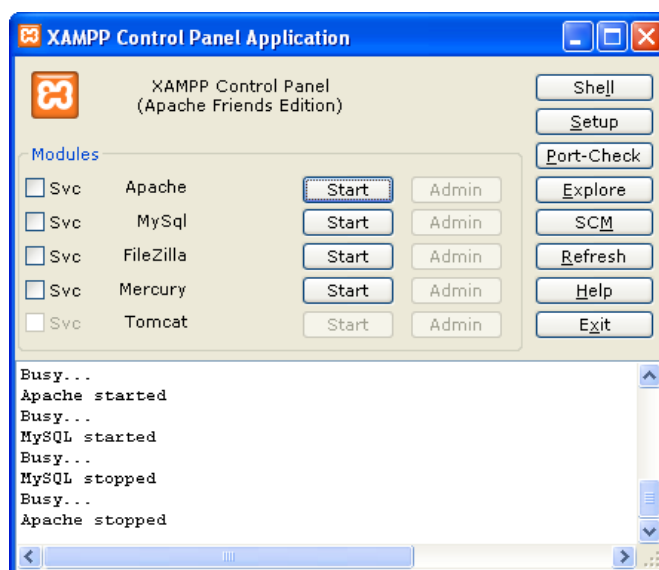


XAMPP Lite		
Version	Size	Content
<b>XAMPP Lite 1.7.3</b>		Apache 2.2.14 (IPv6 enabled), MySQL 5.1.41 + PBXT engine, PHP 5.3.1, OpenSSL 0.9.8l, phpMyAdmin 3.2.4, XAMPP Control Panel 2.5.8, XAMPP CLI Bundle 1.6, Webalizer 2.21-02, msmtp 1.4.19, SQLite 2.8.17, SQLite 3.6.20, Ming 0.4.3 For Windows 2000, XP, Vista, 7. See <a href="#">README</a>
EXE	28 MB	Self-extracting RAR archiv MD5 checksum: ce02838d8e92407926429e3829d113a5
ZIP	61 MB	ZIP archiv MD5 checksum: af1dfef84d1f14be81b772ca885cb7af

- Dalam beberapa saat file akan ter-download. Simpan di tempat yang Anda inginkan.
- Ekstrak paket zip ke folder yang diinginkan dengan dengan aplikasi *compression* yang Anda miliki, misalkan [7zip](#) (utilitas kompresi *free dan open source*).
- Masuk ke folder hasil ekstraksi tersebut.
- Cari dan jalankan file xampp-control.exe.

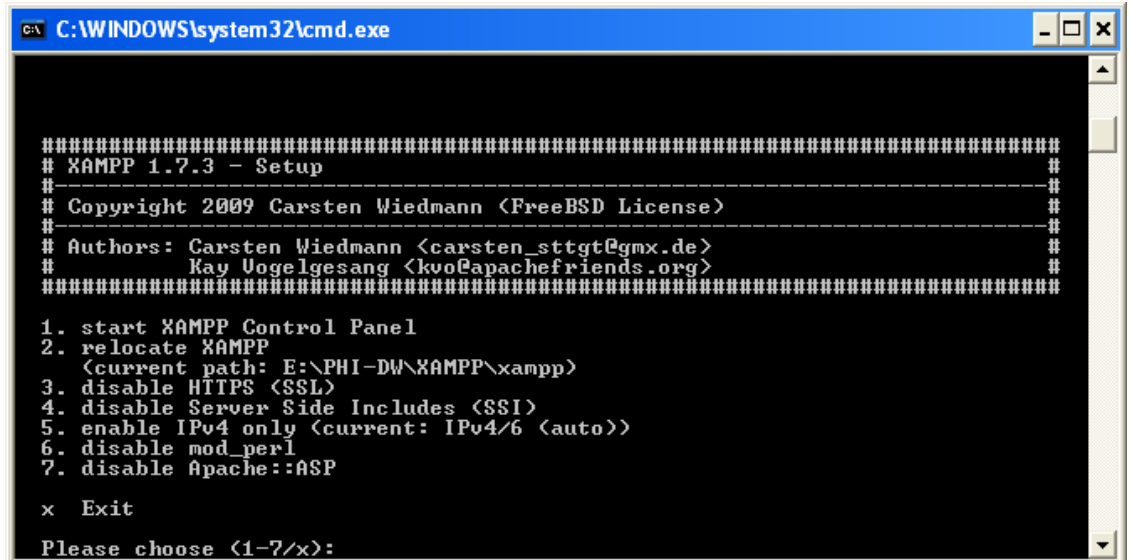
xampp_start.exe	93 KB	Application
xampp_stop.exe	93 KB	Application
xampp-changes.txt	3 KB	Text Document
xampp-control.exe	145 KB	Application
xampp-portcheck.exe	213 KB	Application

- Dalam beberapa saat akan muncul dialog "XAMPP Control Panel Application".



## Konfigurasi XAMPP Control Panel

- Klik tombol "Setup", dalam beberapa saat akan muncul command line screen "XAMPP 1.7.3 - Setup" dengan beberapa pilihan (1-7).



```
C:\WINDOWS\system32\cmd.exe

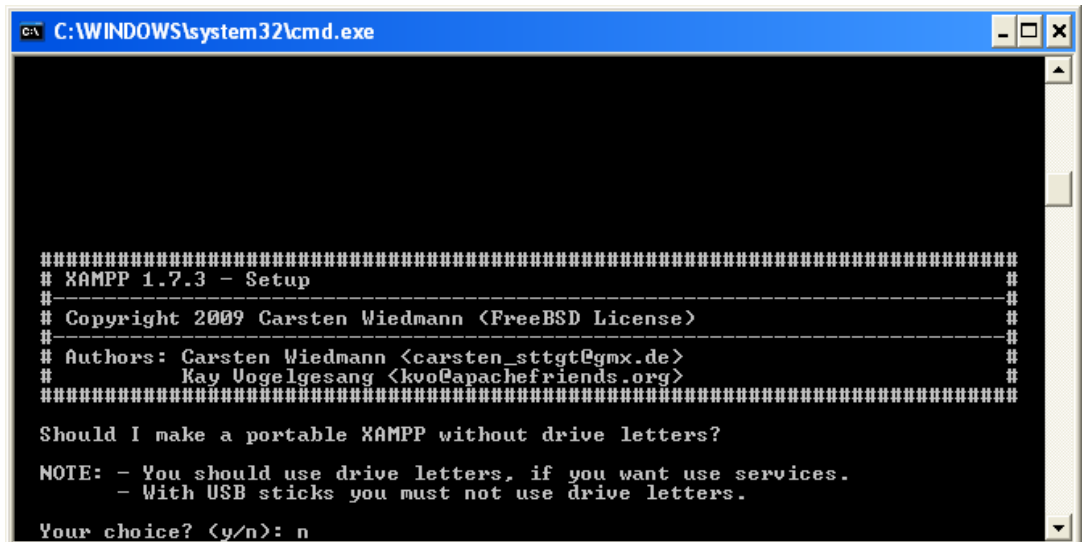
#####
# XAMPP 1.7.3 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)     #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>      #
#           Kay Uogelgesang <kvo@apachefriends.org>      #
#####

1. start XAMPP Control Panel
2. relocate XAMPP
   <current path: E:\PHI-DW\XAMPP\xampp>
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 (auto))
6. disable mod_perl
7. disable apache::ASP

x Exit

Please choose (1-7/x):
```

- Pilih pilihan no. 2 yaitu "relocate XAMPP". Pada dialog "Should I make a portable XAMPP without drive letters ?", pilih "n".



```
C:\WINDOWS\system32\cmd.exe

#####
# XAMPP 1.7.3 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)     #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>      #
#           Kay Uogelgesang <kvo@apachefriends.org>      #
#####

Should I make a portable XAMPP without drive letters?

NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.

Your choice? (y/n): n
```

- Proses relokasi ke folder instalasi akan dimulai, sampai muncul tampilan "XAMPP is ready to use" seperti pada gambar di bawah ini. Tekan tombol "Enter" jika sudah selesai.

```
C:\WINDOWS\system32\cmd.exe
Should I make a portable XAMPP without drive letters?
NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.
Your choice? (y/n): n
relocating XAMPP...
relocate XAMPP base package
relocate Apache
relocate FileZilla FTP Server
relocate Mercury
relocate MySQL
relocate OpenSSL
relocate Perl
relocate PHP
relocate phpMyAdmin
relocate Sendmail
relocate Webalizer
relocate XAMPP Demopage
relocating XAMPP successful.
XAMPP is ready to use.
Press <Return> to continue:
```

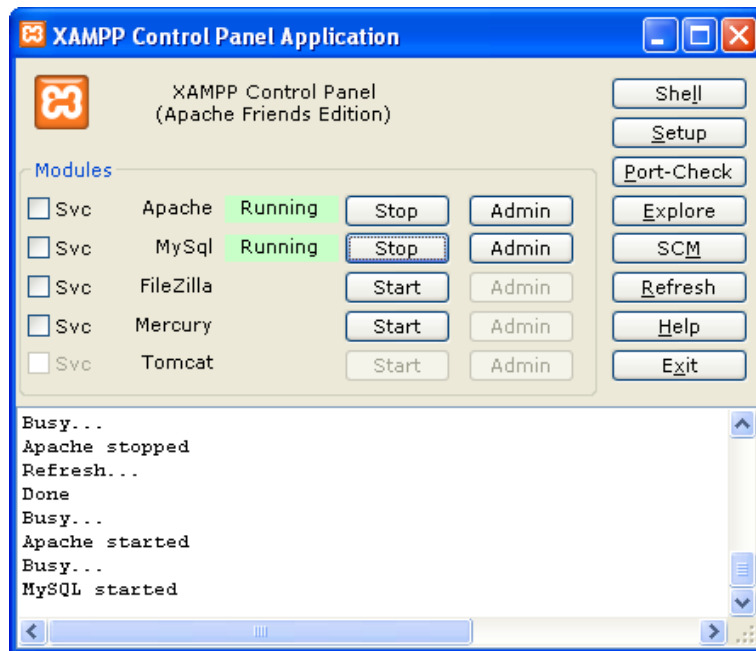
- Setelah kembali ke menu utama, tekan tombol "x" dan Enter untuk menutup dialog command line.

```
C:\WINDOWS\system32\cmd.exe
#####
# XAMPP 1.7.3 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)    #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>     #
#          Kay Vogelgesang <kvo@apachefriends.org>      #
#####

1. start XAMPP Control Panel
2. relocate XAMPP
   (current path: E:\PHI-DW\XAMPP\xampp)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 (auto))
6. disable mod_perl
7. disable Apache::ASP

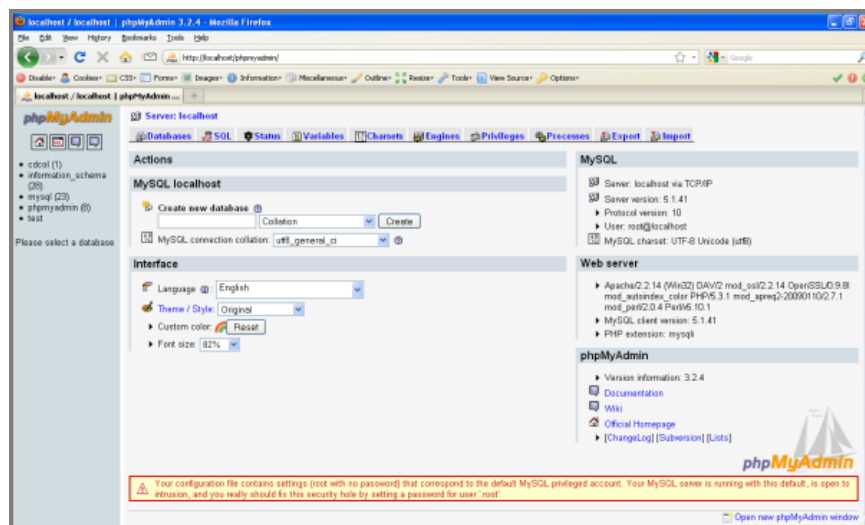
x Exit
Please choose (1-7/x): x
```

- Kembali ke dialog "XAMPP Control Panel Application", jalankan Apache dan MySQL dengan cara klik tombol "Start". Jika semuanya berjalan dengan lancar maka notifikasi "Running" dengan latar belakang hijau akan tampil untuk masing-masing aplikasi.



## Aplikasi web PHPMyAdmin

- Pada paket XAMPP sudah disertakan aplikasi manajemen web MySQL, yaitu PHPMyAdmin. Kita akan mencoba membuka aplikasi ini sekaligus memastikan instalasi telah berjalan dengan baik.
- Buka browser Anda, misalkan Firefox.
- Ketik alamat <http://localhost/phpmyadmin>.
- Aplikasi web PHPMyAdmin dengan daftar database MySQL saat ini akan tampil seperti pada gambar di bawah ini.



#### 4. PRAKTIKUM

##### a. Peralatan dan Bahan

- Perangkat komputer / PC / Laptop / Notebook / Netbook
- Sistem operasi Windows / Linux
- MySQL versi 5.0 atau di atasnya (sebagai engine basis data)

##### b. Keselamatan Kerja

- Baca dan pahami semua langkah kerja dari praktikum ini dengan cermat
- Pastikan diri anda terlindungi dari efek kejutan listrik yang dapat dikarenakan oleh *grounding* instalasi listrik yang tidak sempurna.

##### c. Langkah Kerja

- Siapkan alat dan bahan praktek.
- Selalu memperhatikan keselamatan kerja.
- Jalankan perangkat lunak MySQL dengan menggunakan tools XAMPP/SQL Yog atau menggunakan command line (cmd).
- Organisasikan dengan baik folder pribadi anda untuk masing – masing modul praktikum.

##### d. Kegiatan praktek

1. Lakukan instalasi MySQL client menggunakan salah satu perangkat lunak yang telah dijelaskan sebelumnya (SQLYog ataupun XAMPP)

2. Instalasi Koneksi ke MySQL

Buka command prompt, masuk ke direktori \AppServ\MySQL\bin, kemudian ketik:

```
mysql -u namauser -p (enter)
```

```
password : (ketikkan password) (enter)
```

3. Diskoneksi dari MySQL

Untuk diskoneksi, ketik:

```
QUIT atau \q
```

No.	Ketikkan perintah	Hasil
3.	SELECT VERSION (), CURRENT_DATE;	

4.	<p>Ketik perintah di atas (nomor 3):</p> <ul style="list-style-type: none"> <li>• Dengan huruf kecil semua</li> <li>• Dengan huruf besar semua</li> </ul>	
5.	<pre>SELECT SIN (PI ()/4), (4+1)*5;</pre>	
6.	<pre>SELECT VERSION (); SELECT NOW();</pre>	
7.	<pre>MySQL&gt;SELECT -&gt; USER () -&gt; , -&gt; CURRENT_DATE;</pre>	
8.	<pre>MySQL&gt;SELECT -&gt; USER () -&gt; \c</pre>	
9.	<pre>MySQL&gt;SELECT USER () Apa yang terjadi ?  Kemudian ketikkan : -&gt; ; Apa yang terjadi ?</pre>	

10.	MySQL>SHOW DATABASES;	
11.	MySQL>USE TEST	
12.	MySQL>USE COBA	
13.	MySQL>CREATE DATABASE coba; MySQL> use coba; MySQL> show tables;	
14.	MySQL>CREATE TABLE buku (kodebuku CHAR (5) PRIMARY KEY NOT NULL, -> judul VARCHAR (25) NOT NULL, pengarang VARCHAR (20), -> penerbit VARCHAR (15) , kategori VARCHAR (15));	
15.	MySQL>desc buku;	

## 5. EVALUASI DAN PERTANYAAN

- 1) Tulis semua perintah – perintah SQL percobaan di atas beserta outputnya!
- 2) Perintah apa yang digunakan untuk mengaktifkan salah satu basis data maupun tabel
- 3) Apakah perintah – perintah di MySQL bersifat case sensitive?
- 4) Apa perbedaan pengetikan lebih dari satu perintah, yang dipisahkan dengan tanda koma (,) dan tanda baca titik koma (;)? Bandingkan hasil query percobaan 3 dan 6
- 5) Dapatkah kita melakukan operasi aritmatik dengan menggunakan perintah SQL?berikan contoh yang lain?
- 6) Haruskah kita mengetik perintah dalam satu baris?
- 7) Bagaimana untuk mengakhiri atau membatalkan suatu perintah?
- 8) Sebelum membuat tabel, langkah – langkah apa saja yang sebaiknya kita lakukan?(Berdasarkan percobaan di atas)

## 6. STUDI KASUS

- a. Presentasikan program yang telah anda buat di depan dosen / instruktur

## 7. KESIMPULAN

## 8. Hasil Review

Date	Result (1 – 5)	Signature



## KOMPETENSI

Setelah menyelesaikan modul ini, mahasiswa dapat membuat database dan tabel dengan data definition language (DDL) dan mampu memodifikasi tabel

**1. TUJUAN**

Setelah menyelesaikan modul ini, mahasiswa diharapkan dapat :

- a. Membuat database dan tabel dengan menerapkan data definition language (DDL)
- b. Mampu memodifikasi tabel

**2. TUGAS PENDAHULUAN**

1. Apa yang anda ketahui tentang database dan tabel? (Jelaskan perbedaannya)
2. Apa yang anda ketahui tentang DDL?
3. Sebutkan beberapa contoh perintah dalam DDL!

**3. DASAR TEORI**

DDL (Data Definition Language) merupakan sekumpulan set perintah yang bertujuan untuk mendefinisikan atribut – atribut database, tabel, atribut kolom (field), maupun batasan – batasan terhadap suatu atribut dan relasi/hubungan antar tabel. Yang termasuk dalam kelompok perintah DDL adalah : `CREATE`, `ALTER`, dan `DROP`.

`CREATE` merupakan perintah DDL yang digunakan untuk membuat database maupun tabel. Nama database maupun tabel tidak boleh mengandung spasi (space). Nama database tidak boleh sama antar database.

`ALTER` merupakan perintah DDL yang digunakan untuk mengubah nama/struktur tabel.

`DROP` merupakan perintah DDL yang digunakan untuk menghapus database ataupun tabel.

## a. DATABASE

Syntax untuk membuat database sebagai berikut :

- `CREATE DATABASE namadatabase;`

Berikut merupakan syntax untuk membuat database universitas :

```
CREATE DATABASE db_universitas;
```

Sedangkan syntax tambahan untuk menampilkan daftar nama database yang terdapat dalam database server pada MySQL menggunakan perintah :

- `SHOW DATABASES;`

Sebelum kita membuat suatu tabel yang digunakan untuk menyimpan data, terlebih dahulu harus memilih/mengaktifkan salah satu database sebagai database aktif yang akan digunakan untuk menyimpan beberapa tabel yang akan kita buat. Untuk memilih/mengaktifkan salah satu database menggunakan syntax :

- `USE namadatabase;`

Contohnya kita akan mengaktifkan database db\_universitas :

```
USE db_universitas;
```

Selain perintah - perintah di atas, terdapat perintah yang berfungsi untuk menghapus database maupun tabel. Perintah tersebut adalah sebagai berikut :

- `DROP DATABASE namadatabase;`

Contohnya kita akan menghapus database coba:

```
DROP DATABASE coba;
```

## b. TABEL

- Membuat Tabel

Nama tabel tidak boleh mengandung spasi (space). Ketika membuat tabel ada beberapa yang harus dideklarasikan dalam pembuatannya, yaitu antara lain meliputi : nama tabel, nama Field (Kolom), type data dari field dan panjang data. Adapun syntax yang digunakan untuk membuat tabel secara umum adalah sebagai berikut:

```
CREATE TABLE namatabel (Field1, TypeData1,  
Field2, TypeData2);
```

Contoh berikut syntax untuk membuat Tabel *mahasiswa*:

```
CREATE TABLE mahasiswa (nim CHAR (20),  
nama_mhs CHAR (50),  
login CHAR(20),  
pass CHAR(20),  
umur INT,  
ipk real,  
PRIMARY KEY(nim));
```

- Menampilkan Tabel

Untuk menampilkan daftar nama tabel yang terdapat dalam database yang sedang aktif/digunakan menggunakan perintah :

```
SHOW TABLES;
```

- Menampilkan deskripsi atribut tabel

Untuk menampilkan deskripsi atribut – atribut yang terdapat pada suatu tabel dengan menggunakan perintah:

```
DESC namatabel; misalkan DESC mahasiswa;
```

- Menghapus Tabel

Untuk menghapus Tabel perintahnya sama dengan untuk menghapus database yaitu dengan menggunakan perintah DROP. Syntax yang digunakan adalah:

```
DROP TABLE namaTabel;
```

Tabel yang akan dihapus harus sesuai dengan nama tabel. Misal kita akan menghapus tabel mahasiswa, maka syntax nya adalah:

```
DROP TABLE mahasiswa;
```

- Mendefinisikan Null/Not Null

Null ataupun Not Null merupakan pernyataan yang digunakan untuk membuat kolom yang kita buat boleh kosong (Null) atau tidak boleh kosong (Not Null). Ketika pada kolom tabel tidak diset, maka secara default akan bernilai Null (boleh kosong). Untuk mendefinisikannya maka perintah yang digunakan adalah:

```
CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL,  
nama_mhs CHAR (50) NOT NULL,
```

```
login CHAR(20) NOT NULL,  
pass CHAR(20) NOT NULL,  
umur INT,  
ipk real,  
PRIMARY KEY(nim));
```

- Mendefinisikan Nilai Default

Nilai default merupakan nilai yang diberikan secara otomatis oleh sistem untuk suatu kolom ketika terjadi penambahan baris baru, sementara nilai pada kolom tersebut tidak diisi oleh pengguna.

Contohnya adalah :

```
CREATE TABLE mahasiswa (nim CHAR (20),  
nama_mhs CHAR (50),  
login CHAR(20),  
pass CHAR(20),  
umur INT DEFAULT 0,  
ipk real,  
PRIMARY KEY(nim));
```

- Mendefinisikan PRIMARY KEY pada Tabel

Suatu keharusan dalam suatu tabel adalah harus memiliki satu kolom yang dijadikan sebagai perwakilan dari tabel tersebut. Pembuatan perwakilan tabel ini berfungsi untuk melakukan hubungan/relasional dengan tabel lain. Bentuk perwakilan ini dalam database disebut sebagai PRIMARY KEY yang aturan pembuatannya adalah sebagai berikut:

- Satu tabel hanya diperbolehkan memiliki satu kolom kunci.
- Nama kolom kunci tidak digunakan pada kolom lain dalam satu tabel
- Nama kolom kunci tidak boleh sama dengan kolom kunci yang ada pada tabel lain
- Bentuk kolom kunci harus diset NOT NULL.

Terdapat tiga cara untuk mendefinisikan primary key. Berikut ini syntax yang digunakan:

```
CREATE TABLE mahasiswa (nim CHAR (20),  
nama_mhs CHAR (50),  
login CHAR(20),  
pass CHAR(20),  
umur INT,  
ipk real,  
PRIMARY KEY(nim));
```

Atau

```
CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL  
PRIMARY KEY,  
nama_mhs CHAR (50),  
login CHAR(20),  
pass CHAR(20),  
umur INT,  
ipk real);
```

atau

```
ALTER TABLE mahasiswa ADD CONSTRAINT  
namaconstraint PRIMARY KEY(namakolom);
```

Coba buat perintah untuk membuat/menambahkan kolom pada tabel mahasiswa dengan kolom alamat type datanya char(100), sex type datanya enum('P','W') dengan mendefinisikan nilai Not Null dan Primary key untuk kolom nim serta nilai default untuk kolom alamat.

- Menghapus PRIMARY KEY pada Tabel

Cara 1: Jika primary key dibuat menggunakan alter table:

```
ALTER TABLE namatabel DROP CONSTRAINT  
namaconstraint;
```

Cara 2: jika primary key dibuat melalui create table:

```
ALTER TABLE namatabel DROP PRIMARY KEY;
```

- Menambah kolom baru pada Tabel

Pada saat kita membuat tabel terkadang kita ingin menambahkan kolom lagi pada tabel yang sudah kita buat. Dalam database, hal ini dapat dilakukan dengan menggunakan perintah sebagai berikut:

```
ALTER TABLE namatabel ADD fieldbaru  
typedata(lebar);
```

Namatabel merupakan nama tabel yang akan ditambahkan kolomnya. Filedbaru merupakan nama kolom yang akan ditambahkan, typedata(lebar) merupakan type data dan lebar data yang akan ditambahkan. Misal kita akan menambahkan kolom telepon pada tabel mahasiswa setelah kolom umur:

```
ALTER TABLE mahasiswa ADD COLUMN telepon  
CHAR(15) AFTER umur;
```

- Mengubah Tipe Data atau Lebar Kolom pada Tabel

Perintah yang digunakan adalah:

```
ALTER TABLE namatabel MODIFY COLUMN field  
type(lebar);
```

Contoh :

```
ALTER TABLE mahasiswa MODIFY COLUMN  
telepon(12);
```

- Mengubah Nama Kolom (Field)

Perintah yang digunakan adalah:

```
ALTER TABLE namatabel CHANGE COLUMN  
namakolomlama namakolombaru  
typedatabaru(lebarbaru);
```

Contoh:

```
ALTER TABLE mahasiswa CHANGE COLUMN telepon  
phone CHAR(25);
```

- Menghapus Kolom pada Tabel

Perintah yang digunakan:

```
ALTER TABLE namatabel DROP COLUMN namakolom;
```

Contoh:

```
ALTER TABLE mahasiswa DROP COLUMN phone;
```

- Membuat dan Menghapus Index

Index berfungsi untuk mempercepat proses pencarian data dalam suatu tabel. Dengan adanya index pada suatu field tabel menyebabkan proses pencarian otomatis akan dilakukan terlebih dahulu ke dalam index, apabila ditemukan baru akan diambilkan data sesungguhnya dari tabel, apabila tidak ditemukan dalam index, sudah dapat dipastikan bahwa data tersebut tidak ada dalam tabel. Terdapat perintah untuk membuat dan menghapus index, tapi tidak ada perintah untuk merubah index. Perhatikan contoh berikut:

```
CREATE INDEX IDXNIM ON mahasiswa (nim);
```

Atau

```
ALTER TABLE mahasiswa ADD INDEX IDXNIM(nim);
```

Sedangkan untuk menghapus:

```
DROP INDEX IDXNIM ON mahasiswa;
```

Atau

```
ALTER TABLE mahasiswa DROP INDEX IDXNIM;
```

#### 4. PRAKTIKUM

- a) Buatlah sebuah database dengan nama *db\_universitas* atau *db\_nama\_mhs*
- b) Buatlah beberapa tabel dalam database tersebut sesuai dengan kriteria berikut :

- a. Tabel *mahasiswa*

Field	Type Data
nim	Int (8) Primary Key
nama_mhs	Char (50)
sex	Enum ('L', 'P') DEFAULT 'L'
alamat	Varchar (50)
kota	Varchar (20) DEFAULT 'Purwokerto'
asal_sma	Char (30)
nohp	Varchar (12)
Login	Char (20)
Pass	Char (20)
Umur	Integer
Kode_prodi	Char (6) foreign key fk0

	(kode_prodi) references prodi (kode_prodi)
--	--

b. Tabel *prodi*

Field	Type Data
kode_prodi	Char (6) Primary Key
nama_prodi	Char (30)

c. Tabel *mata\_kuliah*

Field	Type Data
mk_id	Char (10) Primary Key
nama_mk	Char (50)
jumlah_jam	Float (4,2)
Sks	Integer

\*1 SKS = 13,33 jam; 2 SKS = 26,66 jam; 3 SKS = 40 jam; 4 SKS = 53,33 jam

d. Tabel *ruang*

Field	Type Data
ruang_id	Char (3) Primary Key
nama_ruang	Char (20)
Kapasitas	Integer

e. Tabel *dosen*

Field	Type Data
Nik	Int (11) Auto Increment Primary Key
Inisial	Char (3) UNIQUE KEY
nama_dosen	Char (50)
Status	Enum ('T','LB') Default 'T'
Sex	enum ('L','P') Default 'L'
Agama	Char (20)
Login	Char (20)
Pass	Char (20)
Alamat	Varchar (50)
Kota	Varchar (20) Default 'Purwokerto'
Email	Varchar (50)
Nohp	Varchar (12)
Salary	Int

f. Tabel *mengajar*

Field	Type Data
Id_mengajar	Int Auto Increment Primary Key



jam_ke	Integer
Hari	Varchar (10)
mk_id	Char (10) foreign key fk1 (mk_id) references mata_kuliah (mk_id)
Inisial	Char (3) foreign key fk2 (inisial) references dosen (inisial)
kode_prodi	Char (6) foreign key fk3 (kode_prodi) references prodi (kode_prodi)
ruang_id	Char (3) foreign key fk4 (ruang_id) references ruang (ruang) id

g. Tabel *nilai*

Field	Type Data
Nim	Int foreign key fk5 (nim) references mahasiswa (nim)
mk_id	Char (10) foreign key fk6 (mk_id) references mata_kuliah (mk_id)
kode_prodi	Char (6) foreign key fk7 (kode_prodi) references prodi (kode_prodi)
Inisial	Char (3) foreign key fk8 (inisial) references dosen (inisial)
Nilai_UTS	Integer
Nilai_UAS	Integer
Nilai_akhir	Integer

- c) Tambahkan sebuah kolom Agama (varchar (10)) pada tabel mahasiswa sebagai kolom terakhir.
- d) Tambahkan kolom rid(Char 10) di awal kolom (sebagai kolom pertama) pada tabel ruang.
- e) Sisipkan sebuah kolom dengan nama grade (char) pada tabel nilai setelah kolom inisial.
- f) Ubah nama tabel mahasiswa menjadi student.
- g) Jadikan nim sebagai primary key pada table mahasiswa.

## 5. EVALUASI DAN PERTANYAAN

- 1) Tuliskan perintah – perintah SQL percobaan di atas beserta outputnya!
- 2) Apa maksud dari Char(10)?

- 3) Pada saat kita melihat struktur tabel dengan perintah desc, terdapat kolom Null yang berisi YES dan No. Apa maksudnya?

## 6. STUDI KASUS

- a. Tentukan kunci utama pada masing – masing tabel
- b. Tentukan kunci tamu pada tabel mengajar dan tabel nilai

## 7. KESIMPULAN

## 8. Hasil Review

Date	Result (1 – 5)	Signature

## Modul

# 3

## MANIPULASI DAN RETRIEVE DATA (1)

### KOMPETENSI

Setelah menyelesaikan modul ini, mahasiswa dapat mengenal Data Manipulation Language (DML) dan menggunakannya, mampu mengelola record dan retrieve data

#### 1. TUJUAN

Setelah melaksanakan kegiatan praktikum ini mahasiswa dapat

- a) Menerapkan perintah – perintah SQL seperti INSERT, DELETE, UPDATE maupun SELECT untuk memanipulasi data dalam database

#### 2. TUGAS PENDAHULUAN

#### 3. DASAR TEORI

DML (Data Manipulation Language) merupakan kelompok perintah yang berfungsi untuk memanipulasi data dalam database. Contohnya untuk memasukkan, pengambilan, pengubahan maupun penghapusan data. Perintah yang termasuk dalam DML adalah:

INSERT, DELETE, UPDATE, dan SELECT

##### a. INSERT

Perintah INSERT bertujuan untuk menambahkan record data pada suatu tabel. Terdapat beberapa cara untuk menambahkan record, yaitu:

Cara 1: Menambahkan record dengan mengisi data pada setiap kolom:

```
INSERT INTO namatabel VALUES (nilai1, nilai2, nilai-n);
```

Cara 2: menambahkan baris dengan hanya mengisi pada kolom tertentu:

```
INSERT INTO namatabel (field1, field2, field-n) VALUES (nilai1, nilai2, nilai-n);
```

Ket : Jika data bertipe string, date, atau time (contoh : didi, basis data, 1984-03-18) maka pemberian nilainya diapit menggunakan tanda petik tunggal ('Didi') atau petik ganda ("Basis Data"). Jika data bertipe numerik (29, 4) maka pemberian nilainya tidak diapit tanda petik tunggal maupun ganda.

##### b. DELETE

Perintah DELETE digunakan untuk menghapus satu baris, baris dengan kondisi tertentu maupun seluruh baris. Syntax yang digunakan:

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

Perintah dalam tanda [] bersifat pilihan/opsional untuk menghapus suatu baris dengan kondisi tertentu yang dipersyaratkan. Contoh perintah untuk menghapus suatu baris dalam tabel dengan kondisi persyaratan tertentu :

```
DELETE FROM mahasiswa WHERE nim 13120070;
```

c. UPDATE

Perintah UPDATE digunakan untuk mengubah isi data pada satu atau beberapa kolom pada suatu tabel. Syntax yang digunakan secara umum adalah sebagai berikut:

```
UPDATE namatabel SET field1=nilai1, field2=nilai2  
[WHERE kondisi];
```

Perintah dalam tanda [] bersifat pilihan/opsional untuk mengubah suatu baris dengan kondisi tertentu yang dipersyaratkan.

d. SELECT

Perintah SELECT digunakan untuk menampilkan isi dari suatu tabel yang dapat dihubungkan dengan beberapa tabel lainnya.

- Menampilkan data semua kolom dengan menggunakan asterisk (\*):

```
SELECT * FROM namatabel;
```

- Menampilkan data untuk field/kolom tertentu:

```
SELECT filed1, filed2, filed-n FROM namatabel;
```

- Menampilkan data dengan kondisi tertentu menggunakan klausa WHERE:

```
SELECT * FROM namatabel WHERE kondisi;
```

Beberapa operator perbandingan yang dapat digunakan pada klausa WHERE adalah “=” (sama dengan), > (kurang dari), < > (tidak sama dengan), >= (lebih dari sama dengan), <= (kurang dari sama dengan). Adapun operator lain yaitu : AND, OR, NOT, BETWEEN-AND, IN dan LIKE.

#### 4. PRAKTIKUM

1. Aktifkan database *db\_universitas* atau *db\_nama\_mahasiswa*.

2. Isikan data – data (record) ke dalam tabel mahasiswa :

```
INSERT into mahasiswa VALUES
('13120001','Anton', 'L', 'Jl. Jenderal Sudirman No.12','Brebes', 'SMA 1
Brebes', '081574568921','anton','anton','18','S1 IF'),
('13120002','Bany', 'L', 'Jl. Letnan Jenderal Sutoyo No.15','Tegal', 'SMA 1
Tegal', '081574568346','bany','bany','18','S1 IF'),
('13120003','Cika', 'P', 'Jl. Jenderal Sudirman No.38','Purwokerto', 'SMA 4
Purwokerto', '081874568000','cika','cika','17','S1 IF'),
('13120004','Doni', 'L', 'Jl. Pahlawan No. 4','Brebes', 'SMK N 1 Adiwerna',
'081674568900','doni','doni','19','S1 TT'),
('13120005','Eka', 'P', 'Jl. Buntu No.01','Pekalongan', 'SMA 3 Pekalongan',
'088874568945','eka','eka','17','S1 TT'),
('13120006','Fendi', 'L', 'Jl. Kemerdekaan No.15','Pemalang', 'SMA 1
Pemalang', '081257456892','fendi','fendi','18','D3 TT'),
('13120007','Galuh', 'P', 'Jl. Mawar Putih','Brebes', 'SMA 2 Brebes',
'081273458921','galuh','galuh','18','D1 PR')
```

3. Isikan data berikut ke dalam tabel dosen :

```
INSERT into dosen VALUES
('1','DS','Didi Supriyadi','T', 'L','Islam','didi','didi', 'Jl. Sunan
Bonang RT 01/03 Banjarnayar','Brebes', 'didisupriyadi@st3telkom.ac.id',
'085743622236','5000000'),
('2','STS','Sisilia Thya Safitri','T', 'P','Kristen','sisil','sisil', 'Jl.
Antah berantah RT 001/005 Banjarbaru','Jambi', 'sisil@st3telkom.ac.id',
'085875997546','5000000'),
('3','TW','Tenia Wahyuningrum','T', 'P','Islam','tenia','tenia', 'Jl. Ledug
RT 01/03 Banyumas','Purwokerto', 'tenia@st3telkom.ac.id',
'085746795432','5500000'),
('4','DJ','Dwi Januarita AK','T', 'P','Islam','ita','ita', 'Jl. Sunan Kudus
RT 01/03 Jekulo','Kudus', 'ita@st3telkom.ac.id', '085852829809','4000000'),
('5','IS','Irwanto Susanto','T', 'L','Islam','irwan','irwan', 'Jl. Tanjung RT
01/03 Banyumas','Purwokerto', 'irwan@st3telkom.ac.id',
'081327593529','5500000'),
('6','SL','Sarlan','LB', 'L', 'Islam', 'sarlan', 'sarlan', 'Jl. Jend.
Sudirman RT 01/03 Banyumas','Purwokerto', 'sarlan@st3telkom.ac.id',
'081634462738','3500000'),
('7','DN','Daniel','LB', 'L', 'Kristen', 'daniel', 'daniel', 'Jl. Tanjung
RT 04/07 Banyumas','Purwokerto', 'daniel@st3telkom.ac.id',
'081365047309','4500000')
```

4. Lakukan insert data ke tabel – tabel lain sesuai dengan field, type data dan panjang datanya masing – masing tabel

5. Tampilkan semua isi record tabel mahasiswa.

6. Ubah data salah satu nama mahasiswa (misal :Anton menjadi Dino)

7. Tampilkan satu baris data/record data yang telah diubah tadi yaitu record dengan nama Dino saja.

8. Hapus data mahasiswa yang bernama Dino.

9. Tampilkan record/data mahasiswa yang usianya lebih dari atau sama dengan 17 tahun.

10. Tampilkan semua data dosen yang berjenis kelamin Pria dan agamanya Islam.
11. Dengan menggunakan satu perintah SELECT, tampilkan semua nama dosen yang berstatus Tetap dengan agamanya adalah Islam atau dosen Luar Biasa (LB) dengan agama kristen.

#### **5. EVALUASI DAN PERTANYAAN**

1. Tulis semua perintah SQL percobaan di atas dan beserta outputnya.
2. Presentasikan di depan dosen pengampu.
3. Sampaikan kesimpulan anda pada kolom kesimpulan.

#### **6. STUDI KASUS**

-

#### **7. KESIMPULAN**

#### **8. Hasil Review**

Date	Result (1 – 5)	Signature

## Modul

# 4

## MANIPULASI DAN RETRIEVE DATA (2)

### KOMPETENSI

Setelah menyelesaikan modul ini, mahasiswa dapat mencari dan menampilkan data dengan perintah SELECT, mampu mengkombinasikan perintah SELECT dengan perintah lainnya

#### 1. TUJUAN

Setelah menyelesaikan modul ini, diharapkan mahasiswa dapat:

- a. Mencari dan menampilkan data dengan perintah SELECT
- b. Mengkombinasikan perintah SELECT dengan perintah lainnya menggunakan klausa WHERE untuk memanipulasi data dengan syarat atau kondisi tertentu, SORT BY, DISTINCT, BETWEEN, AND & OR

#### 2. TUGAS PENDAHULUAN

a.

#### 3. DASAR TEORI

SELECT merupakan instruksi yang populer digunakan dalam SQL. Instruksi ini berfungsi untuk memilih spesifik kolom dari satu atau beberapa tabel. Secara umum instruksi SELECT adalah sebagai berikut :

```
SELECT kolom1, kolom2, kolom-n,... FROM nama_tabel WHERE predikat/kondisi
```

Jika klausa WHERE tidak digunakan atau diberikan, maka record atau data yang diseleksi adalah seluruh Data dalam tabel. Predikat atau kondisi yang diberikan setelah klausa WHERE menyatakan kualifikasi dari record yang harus ditemukan, apabila memenuhi syarat, maka record tersebut akan dipilih dan ditampilkan. Artinya, bahwa dengan menggunakan klausa WHERE, maka seleksi yang dilakukan bukan pada seluruh record, melainkan hanya pada record yang memenuhi syarat. Bentuk umum klausa WHERE adalah sebagai berikut :

```
WHERE kolom <operator> <nilai>
```

Beberapa operator yang berlaku meliputi :

No	Operator	Arti
1	=	Sama dengan
2	< >	Tidak sama, atau dapat juga !=

3	<	Kurang dari
4	<=	Kurang dari sama dengan
5	>	Lebih dari
6	>=	Lebih dari sama dengan

Logika AND, OR dan NOT

Selain menggunakan operator di atas, klausa WHERE juga dapat digabungkan dengan menggunakan logika AND, OR, dan NOT. Hal ini bertujuan untuk menggabungkan lebih dari satu kondisi maupun negasi.

SELECT \* merupakan karakter khusus yang menyatakan bahwa kolom yang akan dipilih adalah seluruh kolom yang terdapat pada tabel tersebut.

- a. Memberi nama lain pada kolom

```
SELECT nama_kolom_lama AS nama_kolom_baru FROM namatabel
```

Contoh :

```
SELECT salary AS Gaji from dosen;
```

- b. Menggunakan alias untuk nama tabel

```
SELECT nama_alias .namakolom1, nama_alias .namakolom2 FROM namatabel nama_alias;
```

Contoh:

```
SELECT m.nama_mhs, m.alamat FROM mahasiswa m;
```

- c. Menampilkan data lebih dari dua tabel

```
SELECT * FROM nama_tabel1, nama_tabel2, nama_tabel-n;
```

Contoh :

```
SELECT * FROM ruang,prodi;
```

- d. Nested queries/subquery (IN, NOT IN, EXISTS, NOT EXISTS)

Subquery dapat diartikan sebagai query dalam query. Dengan subquery, hasil dari query akan menjadi bagian dari query di atasnya. Subquery terletak dalam klausa WHERE atau HAVING. Pada klausa WHERE, subquery digunakan untuk memilih baris – baris tertentu yang kemudian digunakan oleh query. Sedangkan pada klausa HAVING, subquery digunakan untuk memilih kelompok baris yang kemudian digunakan oleh query.

Contoh IN:

```
SELECT nama_mhs, alamat, kota FROM mahasiswa WHERE kota IN ("Purwokerto", "Brebes");
```



	nama_mhs	alamat	kota
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Anton	Jl. Jenderal Sudirman No.12	Brebes
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Cika	Jl. Jenderal Sudirman No.38	Purwokerto
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Doni	Jl. Pahlawan No. 4	Brebes
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Galuh	Jl. Mawar Putih	Brebes

Contoh NOT IN:

```
SELECT nama_mhs, alamat, kota FROM mahasiswa WHERE
kota NOT IN ("Purwokerto", "Brebes");
```

	nama_mhs	alamat	kota
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Bany	Jl. Letnan Jenderal Sutoyo No.15	Tegal
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Eka	Jl. Buntu No.01	Pekalongan
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Fendi	Jl. Kemerdekaan No.15	Pemalang

Contoh : EXISTS

```
SELECT * FROM mahasiswa AS m WHERE EXISTS (SELECT
* FROM nilai AS n WHERE m.nim = n.nim)
```

	nim	nama_mhs	sex	alamat	kota	asal_sma	nohp	login	pass	umur	kode_prodi
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13120001	Anton	L	Jl. Jenderal Sudirman No.12	Brebes	SMA 1 Brebes	081574568921	anton	anton	18	S1 IF

Contoh : NOT EXISTS

```
SELECT * FROM mahasiswa AS m WHERE NOT EXISTS
(SELECT * FROM nilai AS n WHERE m.nim = n.nim)
```

	nim	nama_mhs	sex	alamat	kota	asal_sma	nohp	login	pass	umur	kode_prodi
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13120002	Bany	L	Jl. Letnan Jenderal Sutoyo No.15	Tegal	SMA 1 Tegal	081574568346	bany	bany	18	S1 IF
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13120003	Cika	P	Jl. Jenderal Sudirman No.38	Purwokerto	SMA 4 Purwokerto	081874568000	cika	cika	17	S1 IF
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13120004	Doni	L	Jl. Pahlawan No. 4	Brebes	SMK N 1 Adiwerna	081674568900	doni	doni	19	S1 TT
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13120005	Eka	P	Jl. Buntu No.01	Pekalongan	SMA 3 Pekalongan	088874568945	eka	eka	17	S1 TT
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13120006	Fendi	L	Jl. Kemerdekaan No.15	Pemalang	SMA 1 Pemalang	081257456892	fendi	fendi	18	D3 TT
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13120007	Galuh	P	Jl. Mawar Putih	Brebes	SMA 2 Brebes	081273458921	galuh	galuh	18	D1 PR

#### e. Operator Comparison ANY dan ALL

Operator ANY digunakan untuk menampilkan record yang terkait dengan instruksi subquery. Operator ANY akan menghasilkan nilai TRUE (benar) jika paling tidak salah satu perbandingan dengan hasil subquery menghasilkan nilai TRUE.

Contoh : ANY

```
SELECT d.nama_dosen, d.inisial FROM dosen d,
mengajar m WHERE d.inisial = m.inisial > ANY
(SELECT m.inisial FROM mengajar m);
```

	nama_dosen	inisial
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Didi Supriyadi	DS

Contoh :ALL

```
SELECT d.nama_dosen, d.inisial FROM dosen d,
mengajar m WHERE d.inisial = m.inisial > ALL
(SELECT m.inisial FROM mengajar m);
```

```
SELECT d nama_dosen, d inisial
FROM dosen d, mengajar m
WHERE d.inisial = m.inisial >=
ALL (
SELECT m.inisial
FROM mengajar m
)
```

Show : 30 row(s) starting from row # 0 in horizontal

+ Options

	nama_dosen	inisial
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Didi Supriyadi	DS
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Sisilia Thya Safitri	STS
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Tenia Wahyuningrum	TW
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Dwi Januarita AK	DJ
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Inwan Susanto	IS
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Sarlan	SL
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Daniel	DN

f. Penyusunan (Sort) menggunakan ORDER BY

Klausa ORDER BY digunakan untuk mengurutkan data berdasarkan kolom tertentu sesuai dengan tipe yang dimiliki.

Contoh ORDER BY:

```
SELECT nama_dosen, inisial FROM dosen ORDER BY
nama_dosen;
```

```
SELECT nama_dosen, inisial
FROM dosen
ORDER BY nama_dosen
LIMIT 0, 30
```

Show : 30 row(s) starting from row # 0 in horizontal

Sort by key: None

+ Options

	nama_dosen	inisial
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Daniel	DN
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Didi Supriyadi	DS
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Dwi Januarita AK	DJ
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Irwan Susanto	IS
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Sarlan	SL
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Sisilia Thya Safitri	STS
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Tenia Wahyuningrum	TW

Atau dengan menggunakan sintak ASC atau DESC seperti berikut:

```
SELECT nama_dosen, inisial FROM dosen ORDER BY
nama_dosen ASC;
```

```
SELECT nama_dosen, inisial FROM dosen ORDER BY
nama_dosen DESC;
```

g. DISTINCT

Digunakan untuk menghilangkan nilai ganda, sehingga data yang ditampilkan hanya data tunggal.

Contoh :

```
SELECT DISTINCT kode_prodi FROM mahasiswa;
```

```
SELECT DISTINCT kode_prodi
FROM mahasiswa
LIMIT 0, 30
```

Show : 30 row(s) starting from row # 0

Sort by key: None

+ Options

	kode_prodi
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	D1 PR
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	D3 TT
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	S1 IF
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	S1 TT

h. BETWEEN dan NOT BETWEEN

Digunakan untuk menyederhanakan pencarian “antara” (range).

Contoh :

```
SELECT nama_mhs, kode_prodi, umur FROM mahasiswa  
WHERE umur BETWEEN 18 and 19;
```

	nama_mhs	kode_prodi	umur
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Anton	S1 IF	18
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Bany	S1 IF	18
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Doni	S1 TT	19
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Fendi	D3 TT	18
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Galuh	D1 PR	18

i. LIKE dan NOT LIKE

Digunakan untuk mencari teks berdasarkan prefix (kata depan), suffix (kata akhir) atau kata tengah.

Contoh :

```
SELECT nama_mhs, kota FROM mahasiswa WHERE kota  
LIKE 'Purwokerto';
```

	nama_mhs	kota
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Cika	Purwokerto

```
SELECT nama_mhs, kota FROM mahasiswa WHERE kota  
LIKE 'B%';
```

	nama_mhs	kota
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Anton	Brebes
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Doni	Brebes
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Galuh	Brebes

```
SELECT nama_mhs, kota FROM mahasiswa WHERE kota  
LIKE '%MAL%';
```

	nama_mhs	kota
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Fendi	Pemalang

```
SELECT nama_mk, mk_id FROM mata_kuliah WHERE mk_id  
LIKE 'MKB%';
```

	nama_mk	mk_id
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Arsitektur dan Organisasi Komputer	MKB3012
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Pengantar Teknologi Informasi	MKB3032
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Praktikum Dasar Komputer	MKB3971

```
SELECT nama_dosen, inisial FROM dosen WHERE  
inisial LIKE 'D_';
```

	nama_dosen	inisial
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Didi Supriyadi	DS
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Dwi Januarita AK	DJ
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Daniel	DN

j. UNION, INTERSECT, dan EXCEPT

#### 4. PRAKTIKUM

- Tampilkan data mata kuliah yang berlangsung selama 40 jam
- Tampilkan mata kuliah yang mempunyai jumlah jam lebih dari 13,33 jam
- Tampilkan mahasiswa laki – laki yang berasal dari Purwokerto.
- Tampilkan mahasiswa wanita dari Purwokerto dan Jakarta.
- Tampilkan mahasiswa yang bukan berasal dari Purwokerto
- Tampilkan seluruh mahasiswa menurut abjad dari Z – A.
- Berasal dari kota mana saja mahasiswa yang ada?(hilangkan data kota yang duplikasi).
- Tampilkan daftar nama mata kuliah, jumlah jam dan jumlah SKSnya yang jumlah jamnya antara 26 dan 40.
- Tampilkan daftar nama mahasiswa, kode prodi dan umur yang usianya tidak diantara 18 dan 19.
- Tampilkan nama dosen yang jumlah gajinya antara 4.500.000 sampai 5.500.000
- Cari nama mahasiswa yang tempat tinggalnya dimulai dengan 'P'.

- l. Carilah nama mahasiswa yang tempat tinggalnya ditengahnya terdapat kata 'kalong'
- m. Carilah nama dosen, inisial yang inisial namanya diawali dengan huruf 'S'.
- n. Tampilkan kolom nik dosen, nama dosen, salary, tunjangan dan sebuah kolom baru yaitu salary+tunjangan yang berisi jumlah salary dan tunjangan.
- o. Tampilkan kolom nik dosen, nama dosen, salary, tunjangan dan sebuah kolom baru (gunakan alias) yaitu total gaji yang berisi jumlah salary dan tunjangan.

## 5. EVALUASI DAN PERTANYAAN

- a. Jelaskan fungsi subquery EXISTS dan NOT EXISTS
- b. Jelaskan fungsi subquery IN dan NOT IN
- c. Jelaskan fungsi subquery ANY dan ALL

## 6. STUDI KASUS

-

## 7. KESIMPULAN

## 8. Hasil Review

Date	Result (1 – 5)	Signature

## Modul

# 5

## MANIPULASI DAN RETRIEVE DATA (2) KOMPETENSI

Setelah menyelesaikan modul ini, mahasiswa dapat mencari dan menampilkan data dengan perintah SELECT, mampu mengkombinasikan perintah SELECT dengan perintah UNION, INTERSECT, EXCEPT dan fungsi AGREGAT

### 1. TUJUAN

Setelah mengikuti kegiatan praktikum ini, mahasiswa dapat :

- Menerapkan operasi himpunan menggunakan perintah UNION, INTERSECT maupun EXCEPT pada database untuk memanipulasi data.
- Menerapkan operasi fungsi agregat pada database untuk memanipulasi data.
- Menerapkan operasi group function pada database untuk memanipulasi data.

### 2. TUGAS PENDAHULUAN

- Jelaskan macam – macam operasi himpunan aljabar dan kalkulus pada database
- Jelaskan macam – macam operasi fungsi agregat
- Jelaskan tentang operasi group function

### 3. DASAR TEORI

#### A. UNION

UNION merupakan operator atau perintah yang digunakan untuk *menggabungkan* hasil query atau isi data dari 2 (dua) tabel atau lebih dengan ketentuan jumlah, nama dan tipe data atau kolom dari masing – masing tabel yang akan ditampilkan datanya harus sama. Perintah ini terdiri dari dua jenis yaitu UNION dan UNION ALL. Untuk menghasilkan suatu data set perintah UNION harus disisipkan diantara perintah SELECT. Untuk lebih jelasknya kita ambil contoh sederhana dari sekumpulan data sebagai berikut :

Tabel mengajar:

	id_mengajar	jam_ke	hari	mk_id	inisial	kode_prodi	ruang_id
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	Senin	MKB3012	DS	S1 IF	TLA1
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	3	Senin	MKB3971	STS	S1 IF	TLA2

Tabel dosen :

	nik	inisial	nama_dosen	status	sex	agama	login	pass	alamat	kota	email	nohp	salary	tunjangan
Inline Edit Copy Delete	1	DS	Didi Supriyadi	T	L	Islam	didi	didi	Jl. Sunan Bonang RT 01/03 Banjaranyar	Erebes	didisupriyadi@st3telkom.ac.id	085743622236	5000000	1000000
Inline Edit Copy Delete	2	STS	Sisilia Thya Saffri	T	P	Kristen	sisil	sisil	Jl. Antah berantah RT 001/005 Banjarbaru	Jambi	sisil@st3telkom.ac.id	085875997546	5000000	700000
Inline Edit Copy Delete	3	TW	Tenia Wahyuningrum	T	P	Islam	tenia	tenia	Jl. Ledug RT 01/03 Banyumas	Purwokerto	tenia@st3telkom.ac.id	085746795432	5500000	1000000
Inline Edit Copy Delete	4	DJ	Dwi Januarita Ak	T	P	Islam	ita	ita	Jl. Sunan Kudus RT 01/03 Jekulo	Kudus	ita@st3telkom.ac.id	085852829809	4000000	500000
Inline Edit Copy Delete	5	IS	Irwan Susanto	T	L	Islam	irwan	irwan	Jl. Tanjung RT 01/03 Banyumas	Purwokerto	irwan@st3telkom.ac.id	081327593529	5500000	1000000
Inline Edit Copy Delete	6	SL	Sarlan	LB	L	Islam	sarlan	sarlan	Jl. Jend. Sudirman RT 01/03 Banyumas	Purwokerto	sarlan@st3telkom.ac.id	081634462738	3500000	200000
Inline Edit Copy Delete	7	DN	Daniel	LB	L	Kristen	daniel	daniel	Jl. Tanjung RT 04/07 Banyumas	Purwokerto	daniel@st3telkom.ac.id	081365047309	4500000	300000

Kalau kita ingin menggabungkan data inisial dari dua tabel tersebut dengan menggunakan UNION maka akan menghasilkan himpunan data berikut :

inisial

DJ

DN

DS

IS

SL

STS

TW

```
SELECT inisial FROM mengajar
UNION SELECT inisial FROM dosen
```

Secara umum bentuk perintah SQL nya sebagai berikut :

```
(SELECT* FROM tabel_A) UNION (SELECT* FROM tabel_B)
```

Berbeda dengan perintah UNION, perintah UNION ALL menggabungkan keseluruhan data walaupun data tersebut memiliki isi yang sama. Untuk lebih jelasnya kita kembali contoh di atas dengan menggunakan perintah UNION ALL

inisial

DS

STS

DJ

DN

DS

IS

SL

STS

TW

```
SELECT inisial FROM mengajar
UNION ALL SELECT inisial FROM
dosen
```



Perintahnya secara umum adalah sebagai berikut :

```
SELECT* FROM tabel_A UNION ALL SELECT* FROM tabel_B
```

#### B. INTERSECT

INTERSECT merupakan operator atau perintah yang digunakan untuk memperoleh *irisan* data hasil query atau isi data dari 2 (dua) tabel atau lebih dengan ketentuan jumlah, nama dan tipe data atau kolom dari masing – masing tabel yang akan ditampilkan datanya harus sama.

Pada MySQL tidak terdapat operator INTERSECT akan tetapi sebagai gantinya dapat menggunakan operator IN seperti yang telah di jabarkan di sub praktikum sebelumnya.

#### C. EXCEPT

EXCEPT merupakan operator atau perintah yang digunakan untuk memperoleh data hasil query dari luar irisan data dari 2 (dua) tabel atau lebih dengan ketentuan jumlah, nama dan tipe data atau kolom dari masing – masing tabel yang akan ditampilkan datanya harus sama (kebalikan dari INTERSECT)

Pada MySQL tidak terdapat operator EXCEPT akan tetapi sebagai gantinya dapat menggunakan operator NOT IN seperti yang telah di jabarkan di sub praktikum sebelumnya.

#### D. ARITMATIKA dan FUNGSI AGREGAT

##### ARITMATIKA

Perhitungan aritmatika pada suatu database dapat dilakukan dengan menggunakan \* / + - seperti pada umumnya. Sebagai contoh untuk menghitung salary dosen yang dinaikkan 20% adalah sebagai berikut :

```
SELECT nama_dosen, salary * 1.2 AS "Salary Up"  
FROM dosen
```

nama_dosen	Salary Up
Didi Supriyadi	6000000.0
Sisilia Thya Safitri	6000000.0
Tenia Wahyuningrum	6600000.0
Dwi Januarita AK	4800000.0
Irwan Susanto	6600000.0
Sarlan	4200000.0
Daniel	5400000.0

## FUNGSI AGREGAT

### a. COUNT

Merupakan perintah atau fungsi yang digunakan untuk menghitung jumlah baris suatu kolom (record) pada tabel. Contoh:

```
SELECT COUNT( nama_dosen ) AS 'Jumlah Dosen'
FROM dosen
```

Jumlah Dosen
7

Sedangkan untuk menampilkan jumlah dosen yang salarynya di atas rata – rata adalah sebagai berikut:

```
SELECT @avgsalary:=AVG(salary) FROM dosen;
SELECT COUNT(nama_dosen) FROM dosen WHERE salary >
@avgsalary;
```

COUNT(nama_dosen)
4

### b. SUM

Merupakan fungsi yang digunakan untuk menghitung jumlah nilai suatu kolom pada tabel. Perintah umumnya adalah sebagai berikut:

```
SELECT SUM(namakolom) FROM namatabel;
```

Contoh : Tampilkan total salary dosen

```
SELECT SUM(salary) FROM dosen;
```

SUM(salary)
33000000

Sedangkan untuk menampilkan jumlah salary dosen yang salarynya di atas rata – rata adalah sebagai berikut:

```
SELECT @avgsalary:=AVG(salary) FROM dosen;
```

```
SELECT SUM(salary) FROM dosen WHERE salary > @avgsalary;
```

```
SUM(salary)
21000000
```

### c. AVG

Merupakan fungsi yang digunakan untuk menghitung nilai rata – rata suatu kolom pada tabel. Perintah umum untuk menghitung rata – rata adalah sebagai berikut :

```
SELECT AVG(namakolom) FROM namatabel;
```

Contoh:

```
SELECT AVG(salary) FROM dosen;
```

```
AVG(salary)
4714285.7143
```

Sedangkan untuk menampilkan dosen yang salarynya di atas rata – rata adalah sebagai berikut:

```
SELECT @rataratasalary:=AVG(salary) FROM dosen;
```

```
@rataratasalary:=AVG(salary)
4714285.714285714
```

```
SELECT @rataratasalary:=AVG(salary) FROM dosen;
```

```
SELECT nama_dosen, salary FROM dosen WHERE salary >
@rataratasalary
```

+ Options

	nama_dosen	salary
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Didi Supriyadi	5000000
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Sisilia Thya Safitri	5000000
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Tenia Wahyuningrum	5500000
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Irwan Susanto	5500000

MySQL membutuhkan variabel temporary yang didefinisikan dengan `@namavariabel:=ekspresi` yang kemudian digunakan untuk komparasi bagi query dibawahnya.

d. MIN

Merupakan fungsi untuk menampilkan nilai terkecil dari suatu kolom pada tabel. Syntax umumnya adalah sebagai berikut:

```
SELECT MIN(namakolom) FROM namatabel;
```

Contoh : tampilkan nilai salary terkecil dosen

```
SELECT MIN(salary) FROM dosen;
```

MIN(salary)
3500000

Sedangkan untuk menampilkan dosen yang salarynya paling kecil adalah sebagai berikut:

```
SELECT @minsalary:=MIN(salary) FROM dosen;
```

```
SELECT nama_dosen, salary FROM dosen WHERE salary = @minsalary
```

←T→	nama_dosen	salary
<input type="checkbox"/> Edit  Inline Edit  Copy  Delete Sarlan		3500000

e. MAX

Merupakan fungsi untuk menampilkan nilai terbesar dari suatu kolom pada tabel. Syntax umumnya adalah sebagai berikut:

```
SELECT MAX(namakolom) FROM namatabel;
```

Contoh : tampilkan nilai salary terbesar dosen

```
SELECT MAX(salary) FROM dosen;
```

MAX(salary)
5500000

Sedangkan untuk menampilkan dosen yang salarynya paling besar adalah sebagai berikut:

```
SELECT @maxsalary:=MAX(salary) FROM dosen;
```

```
SELECT nama_dosen, salary FROM dosen WHERE salary = @maxsalary
```

←T→	nama_dosen	salary
<input type="checkbox"/> Edit  Inline Edit  Copy  Delete Tenia Wahyuningrum		5500000
<input type="checkbox"/> Edit  Inline Edit  Copy  Delete Irwan Susanto		5500000

#### **4. PRAKTIKUM**

1. Tampilkan dosen yang penghasilan bersihnya (salary + tunjangan) bukan 5.500.000 dan 6.500.000
2. Tampilkan jumlah baris/record tabel dosen
3. Tampilkan/hitung jumlah total penghasilan bersih (salary + tunjangan) di tabel dosen
4. Tampilkan/hitung rata – rata penghasilan bersih (salary + tunjangan) dosen
5. Tampilkan penghasilan bersih terbesar
6. Tampilkan penghasilan bersih terkecil
7. Tampilkan daftar penghasilan bersih dosen (salary + tunjangan)
8. Tampilkan daftar salary dosen dan dikurangi dengan pajak penghasilan 15%
9. Tampilkan jumlah dosen yang penghasilan bersihnya (salary + tunjangan) dibawah rata – rata.
10. Hitung jumlah penghasilan bersih (Salary + tunjangan) yang penghasilan bersihnya diatas rata – rata penghasilan bersih.
11. Tampilkan/hitung banyaknya dosen yang penghasilan bersihnya (Salary + Tunjangan) di atas rata – rata
12. Gunakan tabel mahasiswa. Tampilkan jumlah mahasiswa berdasarkan jenis kelamin.
13. Gunakan tabel mahasiswa. Tampilkan jumlah mahasiswa berdasarkan program studi.
14. Gunakan tabel mahasiswa. Tampilkan jumlah mahasiswa berdasarkan jenis kelamin dan program studi.

#### **5. EVALUASI DAN PERTANYAAN**

1. Dapatkah fungsi agregat dan aritmetika dikombinasikan dengan alias?
2. Dapatkah fungsi agregat dan aritmetika digunakan untuk menampilkan data dari dua tabel atau lebih?

#### **6. STUDI KASUS**

-

## 7. KESIMPULAN

--

## 8. Hasil Review

Date	Result (1 – 5)	Signature

## Modul

# 6

## MANIPULASI DAN RETRIEVE DATA (2) KOMPETENSI

Setelah menyelesaikan modul ini, mahasiswa dapat mencari dan menampilkan data dengan perintah SELECT, mampu mengkombinasikan perintah SELECT dengan perintah GROUP BY, HAVING, dan JOIN

### 1. TUJUAN

Setelah mengikuti kegiatan praktikum ini, mahasiswa dapat :

- Mampu melakukan pencarian kembali data dan mengelompokkannya
- Mampu menampilkan data dengan pencocokan pola tau karakter

### 2. TUGAS PENDAHULUAN

### 3. DASAR TEORI

#### a. GROUP BY

Klausa GROUP BY digunakan untuk menyeleksi himpunan yang dihitung berdasarkan fungsi spesifik atau digunakan untuk melakukan pengelompokan data. Fungsi spesifik yang dapat digunakan antara lain AVG, COUNT dan lainnya.

**Contoh :** Gunakan tabel mahasiswa. Tampilkan jumlah mahasiswa berdasarkan jenis kelamin.

```
SELECT sex, COUNT(*) AS Jumlah FROM mahasiswa GROUP BY sex
```

sex	Jumlah
L	4
P	3

#### b. GROUP BY...HAVING

HAVING berlaku untuk kelompok query group dan berfungsi seperti WHERE. Klausa HAVING digunakan untuk menentukan kondisi bagi klausa GROUP BY. Hanya kelompok atau group yang memenuhi kriteria HAVING saja yang akan diproses atau dihasilkan.

Contoh : perintah untuk menampilkan data hanya kolom kode\_prodi yang dikelompokkan berdasarkan kolom kode\_prodi, dimana jumlah prodi

berdasarkan kelompoknya harus lebih besar dari satu pada tabel mahasiswa.

```
SELECT kode_prodi, COUNT(*) AS Jumlah FROM mahasiswa GROUP BY kode_prodi HAVING COUNT(kode_prodi) >1
```

kode_prodi	Jumlah
S1 IF	3
S1 TT	2

Penggunaan klausa Where, Group By, dan Having ada baiknya memperhatikan hal berikut :

- Where digunakan untuk menampilkan data yang difilter /baris – baris dari operasi yang dinyatakan oleh perintah From.
- Group By digunakan untuk mengelompokkan hasil dari klausa Where
- Having digunakan untuk memfilter baris – baris dari hasil pengelompokkan.

### c. Pattern Matching (Pencocokan pola/Karakter)

Fungsi string dapat digunakan untuk menampilkan data dengan didasarkan pada pencarian dengan karakter. Syntak yang digunakan dalam pencarian data adalah LIKE. Syntak LIKE hampir sama dengan syntak =, perbedaannya adalah kalau syntak = maka pencarian datanya harus sama dengan kata yang diminta, akan tetapi ketika menggunakan LIKE, karakter yang akan ditampilkan tidak harus lengkap melainkan hanya dengan menuliskan salah satu huruf atau kata saja, maka semua data yang dicari akan ditampilkan.

Simbol yang digunakan untuk pencocokan pola yaitu:

- ✓ % : digunakan untuk mencocokkan karakter sebelum atau sesudah tanda %
- ✓ \_ : digunakan untuk mencari karakter sebanyak jumlah tanda \_.

Bentuk umum penggunaan syntak LIKE :

- SELECT \* FROM nama\_tabel WHERE nama\_kolom LIKE 'char%';
- SELECT \* FROM nama\_tabel WHERE nama\_kolom LIKE '%char';



- `SELECT * FROM nama_tabel WHERE nama_kolom LIKE '%char%';`
- `SELECT * FROM nama_tabel WHERE nama_kolom NOT LIKE '%char%';`
- `SELECT * FROM nama_tabel WHERE nama_kolom LIKE '_';`

#### d. JOIN

Join merupakan operasi yang digunakan untuk menggabungkan dua tabel atau lebih dengan hasil merupakan gabungan dari kolom – kolom yang berasal dari tabel – tabel tersebut.

- Inner Join

Digunakan untuk menampilkan gabungan data dari dua arah/dua tabel yang berisi data sesuai dengan syarat dibelakang on (tidak bernilai null), artinya semua data tabel di sebelah kiri mendapatkan pasangan data dari tabel sebelah kanan.

Contoh : kita akan menggabungkan tabel mahasiswa dan tabel prodi dimana kita akan menampilkan daftar mahasiswa dan prodi yang dipilih

Cara 1 : Penggabungan dengan klausa WHERE

```
SELECT mahasiswa.nim, mahasiswa.nama_mhs,
prodi.kode_prodi, prodi.nama_prodi FROM mahasiswa,
prodi WHERE mahasiswa.kode_prodi=prodi.kode_prodi
```

nim	nama_mhs	kode_prodi	nama_prodi
13120007	Galuh	D1 PR	D1 Pemrograman
13120006	Fendi	D3 TT	D3 Teknik Telekomunikasi
13120001	Anton	S1 IF	S1 Informatika
13120002	Bany	S1 IF	S1 Informatika
13120003	Cika	S1 IF	S1 Informatika
13120004	Doni	S1 TT	S1 Teknik Telekomunikasi
13120005	Eka	S1 TT	S1 Teknik Telekomunikasi

Cara 2 : Penggabungan dengan Inner Join

```
SELECT mahasiswa.nim, mahasiswa.nama_mhs,
prodi.kode_prodi, prodi.nama_prodi FROM mahasiswa
INNER JOIN prodi ON
mahasiswa.kode_prodi=prodi.kode_prodi
```

nim	nama_mhs	kode_prodi	nama_prodi
13120007	Galuh	D1 PR	D1 Pemrograman
13120006	Fendi	D3 TT	D3 Teknik Telekomunikasi
13120001	Anton	S1 IF	S1 Informatika
13120002	Bany	S1 IF	S1 Informatika
13120003	Cika	S1 IF	S1 Informatika
13120004	Doni	S1 TT	S1 Teknik Telekomunikasi
13120005	Eka	S1 TT	S1 Teknik Telekomunikasi

- Outer Join

Digunakan untuk menggabungkan satu arah, sehingga memungkinkan ada data yang Null (kosong) di satu sisi. Contoh kita akan menggabungkan tabel dosen dan mengajar dimana kita akan menampilkan daftar dosen yang pernah melakukan pengajaran mata kuliah. Outer join terbagi menjadi dua yaitu Left Join dan Right Join.

- Left Join

```
SELECT dosen.nik, dosen.inisial,
dosen.nama_dosen, mengajar.mk_id FROM dosen LEFT
JOIN mengajar ON dosen.inisial=mengajar.inisial
```

nik	inisial	nama_dosen	mk_id
1	DS	Didi Supriyadi	MKB3012
2	STS	Sisilia Thya Safitri	MKB3971
3	TW	Tenia Wahyuningrum	NULL
4	DJ	Dwi Januarita AK	NULL
5	IS	Irwan Susanto	NULL
6	SL	Sarlan	NULL
7	DN	Daniel	NULL

penggunaan left join akan menampilkan juga data dosen dengan inisial TW, DJ, IS, SL, dan DN, walaupun dosen tersebut belum pernah mengajar. Dan pada kolom mk\_id untuk dosen – dosen tersebut isinya NULL, artinya di tabel kanan (mengajar) dosen tersebut tidak ada.

- Right Join

```
SELECT dosen.nik, dosen.inisial,
dosen.nama_dosen, mengajar.mk_id FROM dosen
```

```
RIGHT JOIN mengajar ON
dosen.inisial=mengajar.inisial
```

nik	inisial	nama_dosen	mk_id
1	DS	Didi Supriyadi	MKB3012
2	STS	Sisilia Thya Safitri	MKB3971

Dengan right join, tabel yang menjadi acuan adalah tabel sebelah kanan (tabel mengajar), jadi semua isi tabel pesan akan ditampilkan. Jika data dosen tidak ada di tabel mengajar, maka isi tabel mengajar tetap ditampilkan.

- Join 3 Tabel

Untuk menggabungkan tiga atau lebih tabel, secara prinsip sama dengan penggabungan dengan dua buah tabel. Sebagai contoh misalnya kita akan menampilkan data dosen beserta mata kuliah yang diajar dan detail mata kuliahnya. Berikut perintah SQL-nya :

```
SELECT dosen.nik, dosen.inisial, dosen.nama_dosen,
mengajar.mk_id, mata_kuliah.nama_mk, mata_kuliah.sks FROM
dosen, mengajar, mata_kuliah WHERE
dosen.inisial=mengajar.inisial AND
mengajar.mk_id=mata_kuliah.mk_id
```

nik	inisial	nama_dosen	mk_id	nama_mk	sks
1	DS	Didi Supriyadi	MKB3012	Arsitektur dan Organisasi Komputer	2
2	STS	Sisilia Thya Safitri	MKB3971	Praktikum Dasar Komputer	1

#### 4. PRAKTIKUM

1. Tuliskan syntax SQL sehingga diperoleh hasilnya sebagai berikut :

kode_prodi	Jumlah
D1 PR	1
D3 TT	1
S1 IF	3
S1 TT	2

2. Tuliskan syntax SQL sehingga diperoleh hasilnya sebagai berikut :

sex	Jumlah
L	4
P	3

3. Tuliskan syntax SQL sehingga diperoleh hasilnya sebagai berikut :

nim	nama_mhs	nilai_uts	nilai_uas
13120001	Anton	90	80
13120002	Bany	NULL	NULL
13120003	Cika	NULL	NULL
13120004	Doni	NULL	NULL
13120005	Eka	NULL	NULL
13120006	Fendi	NULL	NULL
13120007	Galuh	NULL	NULL

4. Tuliskan syntax SQL sehingga diperoleh hasilnya sebagai berikut :

nim	nama_mhs	nilai_uts	nilai_uas
13120001	Anton	90	80

5. Tuliskan syntax SQL sehingga diperoleh hasilnya sebagai berikut :

mk_id	nama_mk	jumlah_jam	sks
MKB3971	Praktikum Dasar Komputer	13.33	1

6. Tuliskan syntax SQL sehingga diperoleh hasilnya sebagai berikut :

nik	inisial	nama_dosen	mk_id	nama_mk	sks
1	DS	Didi Supriyadi	MKB3012	Arsitektur dan Organisasi Komputer	2

## 5. EVALUASI DAN PERTANYAAN

1. Jelaskan perbedaan antara Outer Left Join dengan Natural Left Join, sertakan bentuk umum dari syntax keduanya
2. Jelaskan perbedaan antara Outer Right Join dengan Natural Right Join, sertakan bentuk umum dari syntax keduanya

**6. STUDI KASUS**

**7. KESIMPULAN**

**8. Hasil Review**

<b>Date</b>	<b>Result (1 – 5)</b>	<b>Signature</b>

**KOMPETENSI**

Setelah menyelesaikan modul ini, mahasiswa dapat membuat relasi antar table dan membuktikan integritas data dengan tabel yang saling berhubungan (relasi)

**1. TUJUAN**

- a. Mahasiswa mampu membuat relasi antar tabel
- b. Mahasiswa mampu membuktikan integritas data dengan tabel yang saling berhubungan (relasi)

**2. TUGAS PENDAHULUAN****3. DASAR TEORI**

Relational Database merupakan salah satu konsep penyajian dan penyimpanan data. Sebelum ada relasional database, sudah ada dua model database yaitu network database dan hirarki database. Pada database relasional, data disimpan dalam bentuk relasi atau tabel dua dimensi, serta antar tabel yang satu dengan tabel lainnya terdapat hubungan atau relationship. Kemudian kumpulan dari data yang diorganisasikan sebagai tabel disimpan dalam bentuk data elektronik di dalam media penyimpanan komputer. Untuk membuat struktur tabel, mengisi, mengubah maupun menghapus data dari tabel pada suatu database elektronik diperlukan software.

Software yang digunakan untuk membuat strktur tabel, mengisi, mengubah maupun menghapus data disebut Relasional Database Management System (RDBMS). Perintah yang digunakan menggunakan perintah Structure Query Language (SQL). Selain fungsi di atas, RDBMS juga digunakan untuk manajemen data dalam skala besar serta mendukung proses bisnis yang berkelanjutan dan real time. RDBMS harus memiliki kemampuan manajemen user dan keamanan data, backup, recovery serta kemampuan lain yang berkaitan dengan kecepatan pemrosesan data (performance).

Primary key adalah suatu field atau kombinasi field yang secara unik mengidentifikasi setiap record dalam tabel.

Foreign key adalah primary key pada suatu tabel yang dimasukkan pada tabel lain dan dijadikan salah satu key pada tabel tersebut.

References menunjukkan bahwa foreign key pada suatu tabel merupakan key pada tabel lain. Pada perubahan suatu record baik update maupun delete, dapat diikuti dengan kata : cascade, restrict, atau set null sesuai dengan kebutuhan.

Contoh :

-on update CASCADE on delete RESTRICT

-on update RESTRICT on delete SET NULL

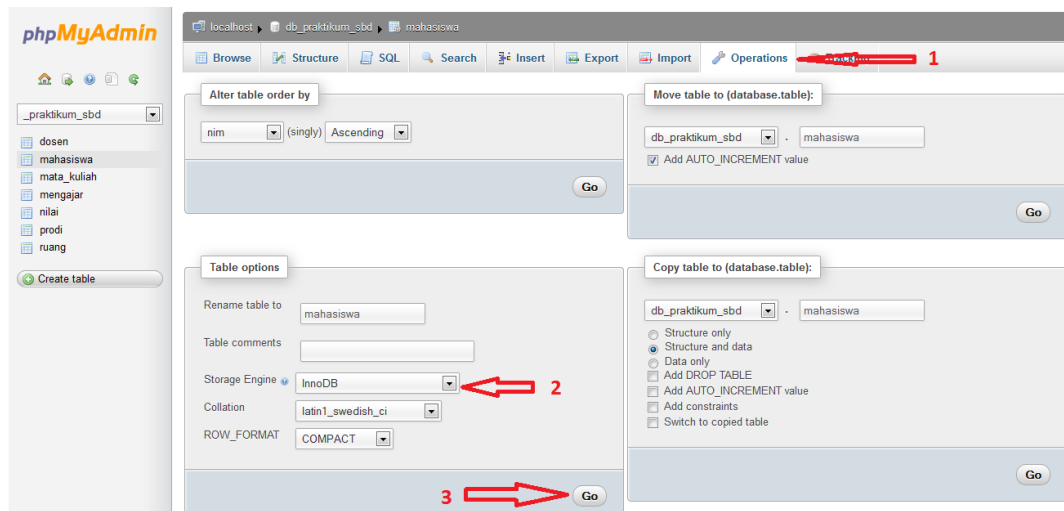
#### 4. PRAKTIKUM

1. Buat sebuah database dengan nama kursus

Create database kursus;

2. Buat/ubah tipe tabel peserta menjadi tipe InnoDB.

```
create table peserta (  
nrp varchar(10) not null,  
nama varchar(40) ,  
primary key(nrp) );
```



3. Buat/ubah tipe tabel tutor menjadi tipe InnoDB dan relasikan ke tabel mahasiswa dengan mengambil key NRP dari tabel peserta\_kursus sebagai foreign key di tabel tutor

```
create table tutor (
```

```

id int(5) not null auto_increment,
kode_tutor varchar(5),
nama varchar(40),
nrp varchar(10),
primary key(id),
index(nrp),
foreign key(nrp) references peserta(nrp) on update
cascade on delete cascade)
engine innodb

```

4. Isi data pada tabel seperti data berikut:

nrp	nama	id	kode_tutor	nama	nrp
001	Anton Wahyu Nugroho	1	A001	Beni S	001
002	Budi Surya	2	A002	Wahyu	002
003	Cici Paramida	3	A001	Beni S	003
004	Denny Dargo	4	A002	Wahyu	004

dan

5. Update satu record pada tabel peserta:

```

UPDATE peserta SET nrp='005' WHERE nama='Denny
Dargo'

```

6. Lihat perubahan isi pada tabel tutor. Tunjukkan perubahannya.
7. Hapus sebuah record pada tabel peserta :
- ```

delete from peserta where nrp='005'

```
8. Lihat perubahan isi pada tabel tutor. Tunjukkan perubahannya.

## 5. EVALUASI DAN PERTANYAAN

1. Lakukan percobaan pada langkah 3 (tiga) (baris sintak terakhir) diubah menjadi on update cascade on delete restrict dan perhatikan apa yang terjadi.
2. Lakukan percobaan pada langkah 3 (tiga) (baris sintak terakhir) diubah menjadi on update cascade on delete set null dan perhatikan apa yang terjadi.
3. Lakukan percobaan pada langkah 3 (tiga) (baris sintak terakhir) diubah menjadi on update restrict on delete set null dan perhatikan apa yang terjadi.



## 6. STUDI KASUS

Dalam sebuah kelompok kecil, buat dan tunjukkan sebuah database yang di dalamnya terdapat minimal 3 buah tabel. Setelah itu buat relasi untuk tabel - tabel tersebut dan buktikan relasi tersebut dengan melakukan update atau delete terhadap suatu tabel berpengaruh terhadap tabel lainnya.

## 7. KESIMPULAN

### 8. Hasil Review

| Date | Result (1 – 5) | Signature |
|------|----------------|-----------|
|      |                |           |