

## Belajar Mysql dengan Phpmyadmin

**Akhmad Sofwan**

[sofwan@sofwan.net](mailto:sofwan@sofwan.net)

<http://www.sofwan.net>

<http://blog.sofwan.net>

### **Lisensi Dokumen:**

Copyright © 2003-2006 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Tulisan ini adalah modul kuliah Graphical User Interface I (GUI I) di Perguruan Tinggi Raharja ([www.raharja.ac.id](http://www.raharja.ac.id)) dengan editing seperlu nya . Materi GUI I terdiri dari 2 bagian, yaitu Mengenai dasar-dasar Linux yang terdiri dari 6 pertemuan dan Materi Database Mysql dengan Phpmyadmin. Dan di dalam tulisan ini hanya membahas Mysql dan penggunaan phpmyadmin.

### **Pendahuluan**

Mysql adalah sebuah Database Open Source populer di dunia. Penggunaannya sebagai database bahasa pemrograman populer seperti PHP dan Java adalah hal umum. Untuk memudahkan penggunaan Mysql, terdapat Software open source berbasis GUI, yakni phpmyadmin, yang dapat di download secara gratis di situs [www.phpmyadmin.net](http://www.phpmyadmin.net). Phpmyadmin ini juga terdapat secara default pada Xampp ([www.apachefriends.org](http://www.apachefriends.org)), yaitu software yang membundle apache,php dan mysql serta Perl, di tambah modul-modul tambahan.

## Database Mysql dan Phpmyadmin

### Pengertian

Database : Database adalah sebuah system yang di buat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. Database terdiri dari kumplan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital.

Database digital di manage menggunakan Database Management System (DBMS), yang menyimpan isi database, mengizinkan pembuatan dan maintenance data dan pencarian dan akses yang lain.

Beberapa Database yang ada saat ini adalah : Mysql, Sql Server, Ms.Access, Oracle, dan PostgreSql.

### Pengenalan MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses database nya. Lisensi Mysql adalah FOSS License Exception dan ada juga yang versi komersial nya. Tag Mysql adalah "The World's most popular open source database".

MySQL tersedia untuk beberapa platform, di antara nya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap Mysql, anda dapat menggunakan software tertentu, di antara nya adalah phpmyadmin dan mysql yog. Pada kesempatan kali ini, kita akan menggunakan phpmyadmin, yang terdapat dalam bundle xampp, yang dapat di peroleh di [www.apachefriends.org](http://www.apachefriends.org).

### Phpmyadmin.

Phpmyadmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyadmin, anda dapat membuat database, membuat tabel, menginsert, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual.

PhpMyadmin dapat di download secara free di <http://www.phpmyadmin.net>. Saat tulisan ini di buat, versi phpmyadmin terbaru adalah phpmyadmin 3.3.10. Karena berbasis web, maka phpmyadmin dapat di jalankan di banyak OS, selama dapat menjalankan webserver dan Mysql.

### Xampp

Xampp adalah sebuah paket kumpulan software yang terdiri dari apache, mysql, phpmyadmin, php, Perl, Freetype2,dll. Xampp berfungsi untuk memudahkan instalasi lingkungan php, di mana biasa nya lingkungan pengembangan web memerlukan php,apache,mysql dan phpmyadmin serta software-software yang terkait dengan pengembangan web. Dengan menggunakan xampp, kita tidak perlu menginstall aplikasi-aplikasi tsb satu persatu.

Paket aplikasi perlu di extract dan di install terlebih dahulu, dengan memilih jenis xampp sesuai dengan jenis OS nya.

Setelah sukses menginstall xampp, kita dapat langsung mengaktifkan Mysql dengan cara mengaktifkan xampp. Pada Linux, cara mengaktifkan Mysql dan phpMyadmin sbb :

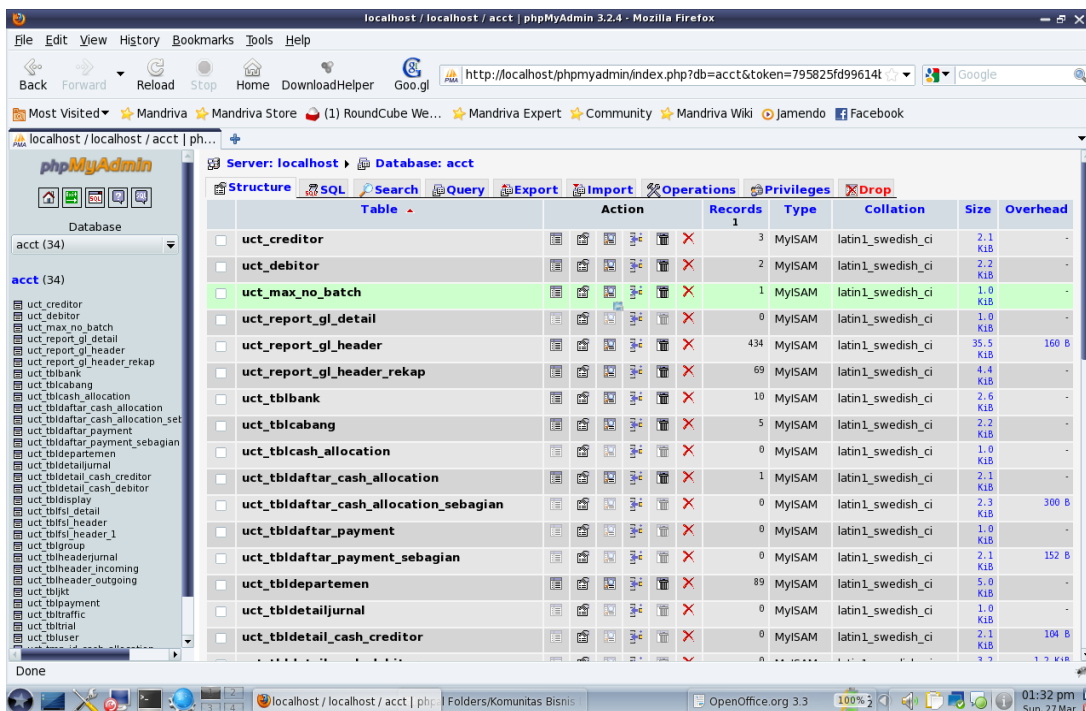
1. Login ke user root
2. Masuk ke direktory /opt/lampp

3. Ketik : #./lammpp start
4. Maka akan memulai xampp , termasuk mengaktifkan phpmyadmin.
5. Selanjutnya, anda bisa masuk ke phpmyadmin, dengan cara :  
<http://localhost/phpmyadmin>

Pada Windows, untuk mengaktifkan phpmyadmin dengan cara sbb :

1. Aktifkan phpmyadmin pada Panel
2. Ketik : <http://localhost/phpmyadmin>

Maka akan terlihat tampilan layar seperti ini :



Gambar VII.1 : Tampilan tables sebuah database

### **Membuat Database :**

Membuat Database, dapat di lakukan dengan cara :

1. Klik link localhost pada bagian atas dan isilah nama database baru nya.

### **Membuat dan Mengelola Tabel**

Setelah kita membuat Database, maka selanjut nya adalah kita akan mencoba membuat Tabel, sbb :

1. Pilih Database dan masukkan nama tabel pada kolom yang tersedia.
2. Masukkan jumlah fields nya.
3. Isilah nama field, pilih jenis type dan isi jumlah length nya.
4. Tentukan juga primary field nya

Untuk mengisi data di dalam sebuah tabel, dapat dengan mengklik link insert.

Tabel adalah sebuah set elemen data (nilai) yang di organisasikan menggunakan sebuah model vertikal (kolom) atau di sebut field dan horizontal (baris). Sebuah tabel memiliki jumlah kolom tertentu . Tabel dapat juga di sebut dengan Relasi.

Sebuah database dapat memiliki 1 atau lebih Tabel. Sebuah tabel dapat memiliki feature sbb :

- Primary key : Sebuah key yang unique yang menyatakan id dari setiap baris
- auto\_increment : Sebuah field yang bertambah nilai 1 secara otomatis, ketika ada data yang di masukkan.

Beberapa Feature yang ada di dalam phpmyadmin adalah sbb :

- Feature menjalankan SQL secara manual
- Status Information
- Pengaturan Hak akses user
- Export dan Import database dan table

## Database Mysql dan Phpmyadmin

### Tipe Data

Pada Mysql, terdapat beberapa tipe data yang dapat di gunakan di dalam pembuatan tabel. Contoh pada tabel Mahasiswa :

Pada tipe data integer, dapat di peroleh informasi sbb :

No	Nama Tipe Data	Storage Bytes	Range Value (Signed/Unsigned)
1	Tinyint	1	-128 – 127 / 0 - 255
2	smallint	2	-32768 – 32767 / 0 - 65535
3	mediumint	3	-8388608 – 8388607 / 0 - 16777215
4	int	4	-2147483648 – 2147483647 / 0 - 4294967295
5	bigint	8	-9223372036854775808 – 9223372036854775807 / 0 - 18446744073709551615

Tipe data lain yang ada pada Mysql adalah sbb :

Data float :

- Float
- Real
- Double
- Decimal

Data String :

- Varcha r
- Text
- Date

### Server Variables dan Setting:

Server variables dan settings adalah variabel yang di miliki oleh Phpmyadmin untuk mengatur sistem dari Phpmyadmin itu sendiri. Beberapa variabel yang di miliki oleh phpmyadmin adalah sbb :

basedir : Adalah direktori di mana phpmyadmin di install.

Contoh value : /opt/lampp

dateformat : Adalah format tanggal default.

Contoh value : %Y-%m-%d

datetime format : Adalah format tanggal dan waktu default.

Contoh value : %Y-%m-%d %H:%i:%s

general log file : Adalah file log general, berikut letak direktori nya.

Contoh value : /opt/lampp/var/mysql/sofwan.log

version compile os : OS di mana phpmyadmin di install  
Contoh value : pc-linux-gnu

### **Status dan Informasi**

Untuk menjalankan Phpmyadmin, kita membutuhkan apache dan Mysql dan untuk mengetahui informasi singkat mengenai apache dan mysql yang terinstall yang di gunakan oleh phpmyadmin, kita dapat mengetahui nya di phpmyadmin itu sendiri, dengan cara melihat dari Home dan tab status.

Pada page Home, anda dapat melihat informasi mengenai MySQL, Web Server dan phpMyAdmin. Pada tab Status, anda dapat melihat informasi berikut :  
Server Traffic dan Query Statistics.

### **Server Traffic**

Beberapa data pada server traffic adalah sbb :

Aliran data : Received dan Sent

Connections : Max.concurrent connections, Failed attempts dan Aborted.

### **Query Statistics**

Adalah statistic mengenai query yang pernah di jalankan di dalam phpmyadmin.

Beberapa query type, yaitu :

- |                  |                   |
|------------------|-------------------|
| - admin commands | - show collations |
| - change db      | - show databases  |
| - select         | - show grants     |
| - set option     | - show plugins    |
| - show binlogs   | - show status     |
| - show charsets  | - show tables     |

**MySQL**

- Server: MySQL host info: Localhost via UNIX socket
- Server version: 5.1.41
  - Protocol version: 10
  - User: root@localhost
- MySQL charset: UTF-8 Unicode (utf8)

**Web server**

- Apache/2.2.14 (Unix) DAV/2 mod\_ssl/2.2.14 OpenSSL/0.9.8l PHP/5.3.1 mod\_apreq2-20090110/2.7.1 mod\_perl/2.0.4 Perl/v5.10.1
- MySQL client version: mysqlnd 5.0.5-dev - 081106 - \$Revision: 289630 \$
- PHP extension: mysql

**phpMyAdmin**

- Version information: 3.2.4
- [Documentation](#)
- [Wiki](#)
- [Official Homepage](#)
- [\[ChangeLog\]](#) [\[Subversion\]](#) [\[Lists\]](#)



Gambar VIII.1 : Tampilan informasi Home

Server: localhost

Databases SQL Status VariablesCharsetsEnginesPrivilegesProcessesExportImport

**Runtime Information**

[Refresh] [Reset] [ @ ]

This MySQL server has been running for 0 days, 0 hours, 1 minutes and 51 seconds. It started up on Mar 27, 2011 at 10:52 AM.

[SQL query] [SSL] [Handler] [Query cache] [Threads] [Binary log] [Temporary data] [Delayed inserts] [Key cache] [Joins] [Replication] [Sorting] [Tables] [Transaction coordinator]

**Server traffic: These tables show the network traffic statistics of this MySQL server since its startup.**

Traffic <sup>1</sup>		ø per hour	Connections		ø per hour	%
Received	1,573 B	50 KiB	max. concurrent connections	1	---	---
Sent	19 KiB	623 KiB	Failed attempts	0	0.00	0.00%
Total	21 KiB	673 KiB	Aborted	0	0.00	0.00%
			Total	5	162.16	100.00%

**Query statistics: Since its startup, 51 queries have been sent to the server.**

Total	ø per hour	ø per minute	ø per second
51	1.65 k	27.57	0.46

Query type	ø per hour	%	Query type	ø per hour	%
admin commands	0	0.00 0.00%	restore table	0	0.00 0.00%
assign to keycache	0	0.00 0.00%	revoke	0	0.00 0.00%
alter db	0	0.00 0.00%	revoke all	0	0.00 0.00%
alter db upgrade	0	0.00 0.00%	rollback	0	0.00 0.00%
alter event	0	0.00 0.00%	rollback to savepoint	0	0.00 0.00%

Gambar VIII.2 : Tampilan Informasi tab Status (Runtime Information)

## Pengoperasian Database I

Untuk mengoperasikan database Mysql dengan phpmyadmin, kita perlu mengerti bahasa SQL, walaupun kita tetap bisa manage database tanpa SQL, namun untuk kepentingan pemrograman dan pengolahan database lebih lanjut, SQL adalah hal yang penting untuk di ketahui.

### Pengertian SQL

SQL (Structured Query Language) adalah sebuah bahasa scripting yang di gunakan untuk mengolah database. Database besar seperti Mysql, PostgreSQL dan SQL Server sudah menggunakan SQL untuk mengolah database nya. SQL yang di gunakan software database tsb adalah sama kecuali sedikit perbedaan di beberapa tempat.

Pada kesempatan kali ini, kita akan menggunakan SQL pada Mysql. Kita akan menggunakan database sbb :

Nama Database : test  
nama tabel : mahasiswa  
struktur tabel :

No	Field	Tipe Data	Length/Values	Keterangan
1	Nrp	Varchar	15	PK
2	Nama	Varchar	40	
3	tempat_lahir	Varchar	35	
4	tgl_lahir	Date		
5	Alamat	Varchar	50	
6	no_telp	Varchar	30	

Berikut adalah beberapa syntax di dalam SQL, yaitu :

#### create database

Fungsi : Untuk membuat database  
Struktur : create database **nama\_database**  
Keterangan : Untuk membuat sebuah database

Contoh :

```
create database test
```

#### create table

Fungsi : Untuk membuat sebuah tabel  
Struktur :  
create table nama\_tabel  
(field1 tipe data(jumlah karakter), field2 tipe data (jumlah karalter));

contoh :

```
create tabel siswa (nama varchar(45), alamat varchar(60));
```



### Select

Fungsi : Untuk menampilkan isi dari sebuah tabel

Struktur : select **nama\_kolom** from **nama\_tabel**

Keterangan : Untuk menampilkan data di nama kolom di tabel nama\_tabel

Contoh :

```
select nama from mahasiswa
```

Keterangan : Menampilkan isi **nama** di tabel **mahasiswa**.

```
Select nrp,nama from mahasiswa
```

Keterangan : Menampilkan isi **nrp dan nama** di tabel **mahasiswa**

Untuk menampilkan semua field, gunakan \*.

### where

Fungsi : Untuk filter dengan kriteria mana saja, sebuah tabel di tampilkan

Contoh : select \* from mahasiswa where nama="Akhmad Sofwan"

Perintah where dapat di gunakan untuk perintah select, update dan delete.

### and

Fungsi : Untuk penghubung operator and atau dan

Perintah and dapat di gunakan untuk perintah select, update dan delete

### or

Fungsi : Untuk penghubung operator or

Perintah or dapat di gunakan untuk perintah select, update dan delete

Tabel hasil nilai dari **and** dan **or**

Nilai A	Nilai B	<b>Or</b>	<b>And</b>
0	0	<b>0</b>	<b>0</b>
1	0	<b>1</b>	<b>0</b>
0	1	<b>1</b>	<b>0</b>
1	1	<b>1</b>	<b>1</b>

Contoh : select nama,alamat from mahasiswa where nama="Budiman"  
and tempat\_lahir="Jakarta"

Keterangan : Menampilkan nama dan alamat dari tabel mahasiswa untuk nama Budiman dan tempat lahir di Jakarta

### order

Fungsi : Untuk mengurutkan data ascending atau descending berdasarkan field tertentu.

Struktur : select \* from nama\_tabel order by field1

Perintah order umum nya di gunakan pada perintah select.

**Contoh**

```
select * from mahasiswa order by nama
```

Keterangan : Menampilkan data tabel mahasiswa dan mengurutkan data nya berdasarkan nama

**Insert**

Fungsi : Untuk memasukkan data ke dalam sebuah tabel

Struktur : insert into nama\_tabel (field1,field2,field3) values ("nilai1","nilai2","nilai3")

**Contoh :**

```
insert into mahasiswa (nrp,nama,tempat_lahir) values ("101101","andi ali","jakarta")
```

Keterangan : Memasukkan nilai 101101 ke field nrp, andi ali ke field nama dan jakarta ke field

tempat\_lahir.

**Update**

Fungsi : Untuk mengupdate isi dari sebuah tabel.

Struktur : update nama\_tabel set field1="isi1"

**contoh :**

```
update mahasiswa set tempat_lahir="Surabaya"
```

Keterangan : merubah field tempat lahir menjadi surabaya

```
update mahasiswa set tempat_lahir="Surabaya" where nama="Budiman"
```

Keterangan : merubah field tempat lahir menjadi surabaya yang nama nya adalah Budiman

**Delete**

Fungsi : Untuk menghapus sebuah atau beberapa record dalam sebuah tabel

Struktur : delete from nama\_tabel

**Contoh:**

```
delete from mahasiswa where nama="Budiman"
```

Keterangan : Menghapus isi dari tabel mahasiswa yang nama nya adalah Budiman

## Pengoperasian Database II

Bab ini adalah kelanjutan dari Bab IX, yang membahas mengenai pengoperasian database.

Kita akan menggunakan tabel mahasiswa dengan nilai sbb :

<b>nrp</b>	<b>nama</b>	<b>tempat_lahir</b>	<b>tgl_lahir</b>	<b>alamat</b>	<b>no_telp</b>
1011500001	Budiman	Jakarta	27 Juni 1990	Jl.Beo No.11	021-888888
1011500006	Siti Aminah	Semarang	1 Juni 1991	Jl.Kerinci No.15	'021.78787878
1011500018	Ujang	Bandung	2 Juni 1992	Jl.Kerbau No.8	'021-7355375
1011500019	Dino	Surabaya	4 Juni 1990	Jl.Kecapi No.18	'021-7459087
1011500025	Dini	Jogjakarta	6 Juni 1990	Jl.Rambutan No.99	'021-7767981
1011500044	Amin	Jakarta	1 Juli 1991	Jl.Sungai No.100	'021-7756789

### **Between**

Fungsi : Untuk memilih data pada range tertentu

Struktur : `select field1,field2 from nama_tabel where field 2 between nilai1 and nilai2`

Contoh :

```
select nrp,nama from mahasiswa where right(nrp,3) between '001' and '006'
```

Keterangan : Menampilkan field **nrp** dan **nama** dari tabel **mahasiswa** yang memiliki nilai nrp dari

kanan di antara '001' dan '006'

Hasil nya akan tampil sbb :

<b>nrp</b>	<b>nama</b>
1011500001	Budiman
1011500006	Siti Aminah

### **Max**

Fungsi : Untuk menampilkan nilai terbesar dari suatu field dengan tipe apa saja, tidak hanya

untuk tipe numerik saja.

Struktur : `select max(nama_field) from nama_tabel`

Contoh :

```
select max(nrp) from mahasiswa
```

Keterangan : Menampilkan nrp maximal yang di ambil dari nilai nrp.

Hasil nya akan tampil sbb :

```
max(nrp)
1011500044
```

**Min**

Fungsi : Untuk menampilkan nilai terkecil dari suatu field  
Struktur : select min(**nama\_field**) from **nama\_tabel**

Contoh :

```
select min(nrp) from mahasiswa
```

Keterangan : Menampilkan nrp minimal yang di ambil dari nilai nrp.

Hasil nya akan tampil sbb :

```
min(nrp)
1011500001
```

**Sum**

Fungsi : Untuk menampilkan jumlah dari sebuah field  
Struktur : select sum(nama\_field) from mahasiswa

Contoh :

```
select sum(nilai) from nilai
```

Keterangan : Menjumlahkan data field nilai dari tabel nilai

Maka akan menghasilkan output sbb :

```
sum(nilai)
-----
    505
```

**Group by**

Fungsi : Untuk menampilkan jumlah sebuah nilai dan di group berdasarkan suatu field tertentu.

Struktur : select nama\_field1,sum(nama\_field2) as nama\_nilai from nama\_tabel  
group by nama\_field1

Contoh : select nrp,sum(nilai) as jumlah\_nilai from nilai group by nrp

Keterangan : Menampilkan field nrp dan jumlah dari field nilai di group berdasarkan nrp.

Hasil :

nrp	jumlah_nilai
-----	-----
1011500001	185
1011500006	170
1011500025	70
1011500044	80

### **Having**

Fungsi : Untuk memfilter tampilan berdasarkan nilai group by

Struktur : `select nama_field1,sum(nama_field2) as nama_nilai from nama_tabel  
group by nama_field1 having sum(nama_field2)>number`

Contoh : `select nrp,sum(nilai) as jumlah_nilai from nilai group by nrp having  
sum(nilai)>100`

Keterangan : Menampilkan field nrp dan jumlah dari field nilai di group berdasarkan nrp dengan

kriteria penjumlahan nilai masing-masing nrp lebih dari 100.

### **As**

Fungsi : Untuk memberi nama lain kepada sebuah tabel atau field, dengan maksud untuk

menyingkat penamaan atau memberi nama lain.

Struktur : `select nama_field1 as nama_field2 from nama_tabel1`

Contoh : `select nrp,nama as nama_mahasiswa from mahasiswa`

Keterangan : Merename field nama menjadi nama\_mahasiswa

Hasil :

nama\_mahasiswa

-----  
Budiman  
Siti Aminah  
Ujang  
Dino  
Dini  
Amin

### **Use**

Fungsi : Untuk menggunakan sebuah database, terutama jika database yang akan di pergunakan

tersebut belum aktif.

Struktur : `use nama_database`

Contoh : `use mahasiswa`

### **Length**

Fungsi : Untuk menghitung banyak karakter data yang di tampilkan, termasuk spasi.

Struktur : `select length(nama_field) from nama_tabel`

Contoh : `select nama,length(nama) as panjang_nama from mahasiswa`

Hasil :

nama            panjang\_nama

-----  
Budiman            7  
Siti Aminah        11  
Ujang                5  
Dino                 4  
Dini                 4  
Amin                 4

### **Trim**

Fungsi : Untuk menghapus spasi pada sebuah karakter

Struktur : `select trim(karakter) from nama_tabel`

contoh : `select trim(" Sofwan ") as hasilnya;`

Hasil :

hasilnya

-----

Sofwan

Terdapat varian nya, yaitu :

Ltrim → Untuk menghilangkan spasi yang terletak di sebelah kiri dari sebuah karakter.

Rtrim → Untuk menghilangkan spasi yang terletak di sebelah kanan dari sebuah karakter.

**PENGOPERASIAN DATABASE III**

**A.Mengoperasikan database dengan tabel lebih dari 1**

Seringkali dalam menampilkan sebuah data, kita perlu menggunakan lebih dari 1 tabel, bisa 2, 3 atau lebih, karena data yang akan di tampilkan terletak pada beberapa tabel dan setiap tabel memiliki keterkaitan melalui primary key dan foreign key nya.

Contoh :

Terdapat 2 tabel, yaitu :

Tabel : mahasiswa

<b>nrp</b>	<b>nama</b>	<b>tempat_lahir</b>	<b>tgl_lahir</b>	<b>alamat</b>	<b>no_telp</b>
1011500001	Budiman	Jakarta	27 Juni 1990	Jl.Beo No.11	021-888888
1011500006	Siti Aminah	Semarang	1 Juni 1991	Jl.Kerinci No.15	'021.787878 78
1011500018	Ujang	Bandung	2 Juni 1992	Jl.Kerbau No.8	'021- 7355375
1011500019	Dino	Surabaya	4 Juni 1990	Jl.Kecapi No.18	'021- 7459087
1011500025	Dini	Jogjakarta	6 Juni 1990	Jl.Rambutan No.99	'021- 7767981
1011500044	Amin	Jakarta	1 Juli 1991	Jl.Sungai No.100	'021- 7756789

Tabel : nilai\_algoritma

<b>id</b>	<b>Nrp</b>	<b>nilai</b>
1	1011500001	80
2	1011500006	70
3	1011500018	60
4	1011500044	90

Kita akan menampilkan data sbb :

Nrp,nama dan nilai yang hanya berisi record yang ada nilai nya di tabel nilai\_algoritma.

**1. Inner Join**

Fungsi : Untuk menampilkan 1 atau lebih field di 2 atau lebih tabel yang berbeda, dengan mengacu kepada sebuah field yang memiliki data yang sama . Data yang di tampilkan adalah data yang ada di kedua tabel tsb.

Contoh : Dalam kedua tabel tsb, kita akan menampilkan data sbb :  
 nrp,nama dan nilai.

Field **nrp** dan **nama** di ambil dari tabel **mahasiswa**  
 Field **nilai** di ambil dari tabel **nilai\_algoritma**

Command :

```
select a.nrp,a.nama,b.nilai from mahasiswa as a inner join nilai_ algoritma as b on a.nrp=b.nrp
```

Keterangan : Menampilkan field nrp, nama dan nilai berdasarkan kesamaan nrp pada kedua tabel yang masing-masing memiliki data pada field nya. Jika ada data yang tidak ada nilai nya, maka tidak di tampilkan.

Hasil :

nrp	nama	nilai
1011500001	Budiman	100
1011500001	Budiman	85
1011500006	Siti Aminah	90
1011500006	Siti Aminah	90
1011500025	Dini	70
1011500044	Amin	80

## 2. left outer join

Fungsi : Untuk menampilkan 1 atau lebih field di 2 atau lebih tabel yang berbeda, dengan mengacu kepada sebuah field yang memiliki data yang sama . Data yang di tampilkan adalah data yang ada di tabel yang di sebelah kiri atau yang pertama kali di sebut pada perintah SQL.

```
select a.nrp,a.nama,b.nilai from mahasiswa as a left outer join nilai_ algoritma as b on a.nrp=b.nrp
```

nrp	nama	nilai
1011500001	Budiman	100
1011500001	Budiman	85
1011500006	Siti Aminah	90
1011500006	Siti Aminah	80
1011500018	Ujang	NULL
1011500019	Dino	NULL
1011500025	Dini	70
1011500044	Amin	80

Pada hasil di atas, terdapat nilai NULL, itu karena syntax left outer join menjadikan tabel yang sebelah kiri atau yang pertama kali di sebut (tabel : mahasiswa) sebagai referensi. Dan jika tidak ada nilai nya berdasarkan tabel nilai\_ algoritma, maka data tetap akan di tampilkan, namun nilai nya adalah NULL.

## 3. right outer join

Fungsi : Untuk menampilkan 1 atau lebih field di 2 atau lebih tabel yang berbeda, dengan mengacu kepada sebuah field yang memiliki data yang sama . Data yang di tampilkan adalah data yang ada di tabel yang di sebelah kanan atau yang urutan kedua di sebut pada perintah SQL.

```
select a.nrp,a.nama,b.nilai from mahasiswa as a right outer join nilai_ algoritma as b on a.nrp=b.nrp
```



nrp	nama	nilai
1011500001	Budiman	100
1011500006	Siti Aminah	90
1011500001	Budiman	85
1011500006	Siti Aminah	80
1011500018	Dini	70
1011500019	Amin	80

Pada hasil di atas, terdapat nilai NULL, itu karena syntax left outer join menjadikan tabel yang sebelah kanan atau yang kedua kali di sebut (tabel : nilai\_algoritma) sebagai referensi. Dan jika tidak ada nilai nya berdasarkan tabel nilai\_algoritma, maka data tetap akan di tampilkan, namun nilai nya adalah NULL. Pada contoh di atas, kebetulan data nrp di tabel nilai, juga ada nrp nya di tabel mahasiswa.

#### 4. **Join**

Untuk mengambil data dari field pada 1 atau lebih tabel yang berbeda, dapat menggunakan command dengan where. Contoh :

Menampilkan nrp,nama dan nilai .

Data-data nya di ambil dari nrp yang sama yang ada di kedua tabel tsb.

Contoh :

```
select a.nrp,a.nama,b.nilai from mahasiswa as a right outer join nilai as b on a.nrp=b.nrp
```

Hasil yang tampil adalah :

nrp	nama	nilai
1011500001	Budiman	100
1011500006	Siti Aminah	90
1011500001	Budiman	85
1011500006	Siti Aminah	80
1011500025	Dini	70
1011500044	Amin	80

Keterangan :

Hasil yang tampil adalah field nrp, nama dan nilai, yang mana, data-data nya di ambil dari kedua tabel yang nrp nya sama. Cara ini menghasilkan hasil yang sama dengan inner join.

Pada perintah penggabungan tabel di atas (inner join, left outer join, right outer join dan where), dapat juga di terapkan pada 3, 4 atau lebih tabel. Yang penting, antara 1 tabel dengan tabel yang lainnya, memiliki field yang sama, sebagai acuan atau kunci untuk mengambil data.

#### B. **Mengisi sebuah tabel dengan mengambil data di tabel lain**

Misalkan kita akan membuat sebuah tabel yang bernama mahasiswa\_a.

Tabel ini memiliki struktur sbb : nrp (varchar (10)) dan nama (varchar(40))

Tabel mahasiswa\_a ini berisi data yang di ambil dari tabel mahasiswa yang memiliki huruf awalan "A". Untuk itu, kita dapat menggunakan perintah berikut :

```
insert into mahasiswa_a (nrp,nama)  
select nrp,nama from mahasiswa where left(nama,2)="A"
```

## PENGOPERASIAN DATABASE IV

### 1. Syntax SQL

Pada bab ini, kita kembali mempelajari beberapa syntax SQL, yaitu :  
Count, Concatenate, left, right, substring.

Untuk tabel yang di gunakan dalam contoh script di bawah, kita menggunakan tabel mahasiswa dan nilai\_algoritma, di bab XI di atas.

#### a. **Count**

Struktur : count(nama\_field)  
Keterangan : Menghitung jumlah record pada field tertentu  
Contoh : select count(nrp) from mahasiswa

Hasil :  
count(nrp)  
-----  
7

#### b. **Concatenate**

Struktur : concat (nama\_field,"keterangan")  
Keterangan : Menambahkan string tertentu pada data sebuah field  
Contoh : select concat(nrp ," mhs") as hasil from mahasiswa

Hasil :  
1011500001 mhs  
1011500006 mhs  
1011500018 mhs  
1011500019 mhs  
1011500025 mhs  
1011500044 mhs  
1011500008 mhs

#### c. **left**

Struktur : left(karakter, jumlah\_karakter)  
Keterangan : Mengambil karakter dari sebelah kiri  
Contoh : select left(nrp,3) as hasil from mahasiswa

Hasil :  
hasil  
-----  
101  
101  
101  
101

#### d. **right**

Struktur : right (karakter,jumlah\_karakter)  
Keterangan : Mengambil karakter dari sebelah kanan  
Contoh : select right(nrp,3) as hasil from mahasiswa

hasil

-----

001  
006  
018  
019  
025  
044  
008

e. **substring**

Struktur : substring(karakter,angka 1,angka 2)  
Keterangan : Untuk mengambil karakter yang di mulai pada angka1,  
dengan panjang angka 2.  
Contoh : substring("Jakarta",2,4);

2. **Index**

Index di pergunakan di dalam database untuk mempercepat pencarian sebuah data. Index dalam database sama dengan index di dalam sebuah buku. Jika pada sebuah buku tentang Binatang dan kita mencari semut, maka dari pada kita membuka halaman buku tersebut satu persatu hingga kita mencari bab tentang semut, maka lebih baik dan lebih cepat, kita melihat index, di halaman berapa bab Semut ada dan setelah itu, kita membuka halaman yang tertera di index pada buku tsb. Maka proses pencarian menjadi lebih cepat.

Sebuah index dapat berdasarkan pada 1 atau lebih kolom.

**Membuat index**

Misal, pada tabel mahasiswa, kita akan membuat index pada nrp.  
Create index idx\_mahasiswa\_nrp on mahasiswa (nrp)

**Strategi membuat Index**

- a. Membuat index pada kolom yang bertipe integer  
Integer mengambil space lebih sedikit, sehingga query berjalan lebih cepat.
- b. Menjaga index agar sesempit mungkin  
Index yang lebih sempit membutuhkan sedikit space, membutuhkan waktu yang lebih sedikit untuk proses.
- c. Urutan kolom adalah penting  
Untuk index yang memiliki beberapa kolom, urutan kolom di dalam index adalah penting.  
Penggunaan terbaik, adalah menggunakan kolom dengan kardinalitas yang paling rendah sebagai index pertama dan kardinalitas tertinggi sebagai yang terakhir. Untuk mengetahui kardinalitas sebuah kolom, dapat menggunakan perintah distinct.
- d. Kolom yang di setting index tidak bernilai NOT NULL

### **3. View**

View adalah tabel bayangan atau virtual tabel. Sebuah view terdiri dari baris dan kolom, seperti tabel. Perbedaan antara view dan tabel adalah, view di definisikan terletak di atas dari tabel lain dan tidak memegang data secara real atau dengan kata lain, data sesungguhnya berupa numerik dan karakter tidak berada di view. Jika data berubah di tabel sebenarnya, maka view juga ikut berubah.

Keuntungan menggunakan view adalah :

**a. Mudah di gunakan**

View menyembunyikan ke kompleksitasan sebuah tabel dari end user, sehingga end user tidak

merasa berat melihat tabel yang kompleks.

**b. Menghemat space**

Views mengambil sedikit sekali space untuk di simpan, karena mereka tidak menyimpan data sebenarnya.

**c. Menambah keamanan data**

Views hanya dapat kolom tertentu yang berada di dalam tabel, jadi hanya kolom yang tidak sensitif yang di insert dan di ekspos ke dalam end user.

**Perintah Membuat view**

```
create view view_name as  
select nama_kolom(s) from table_name where condition
```

Contoh :

```
create view view_mhs as  
select nrp,nama from mahasiswa where tempat_lahir="Jakarta"
```

**Perintah update view**

Sama dengan create view. Jika sebuah view di buat kembali, itu arti nya view tersebut di update. Update di gunakan jika terdapat perubahan pada view, misal : penambahan field.

**Menghapus view**

```
drop view nama_view
```

## Stored Procedure

### Pengertian

Stored Procedure adalah sebuah kelompok kode SQL yang di simpan di katalog database dan dapat di panggil kemudian oleh program, trigger atau bahkan stored procedure.

Sebuah Stored Procedure yang memanggil diri nya sendiri di sebut rekursif stored procedure. Mysql sendiri mendukung penggunaan Stored Procedure sejak MySQL versi 5.x ke atas.

### Keuntungan penggunaan Stored Procedure

- Meningkatkan performance aplikasi.  
Sebuah Stored Procedure di simpan dan di compile di katalog database yang mana dapat di eksekusi lebih cepat di bandingkan SQL yang tidak di compile dari kode aplikasi.
- Mengurangi traffic antara aplikasi dan database server.  
Aplikasi hanya mengirim nama stored procedure untuk mengeksekusi SQL.
- Dapat di gunakan kembali dan transparent ke aplikasi yang ingin menggunakannya.
- Aman.  
Penggunaan Stored Procedure dapat di akses hak nya oleh aplikasi oleh Database Administrator

### Kerugian penggunaan Stored Procedure

- Dapat mengakibatkan Database server membutuhkan memory dan prosessor lebih tinggi.
- Stored procedure hanya berisi SQL deklaratif, sehingga sangat sulit untuk menulis sebuah procedure dengan kompleksitas logika, seperti bahasa pemrograman yang di gunakan untuk memprogram aplikasi.
- Stored procedure tidak dapat di debug di hampir RDBMS, termasuk MySQL.
- Membutuhkan keahlian khusus untuk menulis dan me maintain stored procedure yang tidak setiap developer memiliki, sehingga dapat membuat ribet.

### Membuat Stored Procedure

```
delimiter //  
create procedure nama_mhs()  
begin  
select * from mahasiswa;  
end //  
delimiter ;
```

Kita memulainya dengan **delimiter //** dan di akhiri dengan **// delimiter ;**  
Body sql di mulai dengan **begin** dan di akhiri dengan **end**.

### **Memanggil Stored Procedure**

call nama\_stored\_procedure( )

Contoh :

call nama\_mhs( );

### **Variabel di Stored Procedure**

Variabel di gunakan untuk menyimpan prosedur ke penyimpanan hasil dengan segera. Anda dapat mendeklarasikan sebuah variabel dengan syntax berikut :  
DECLARE nama variabel tipe data (ukuran) DEFAULT nilai default;

Contoh :

DECLARE total\_sales INT DEFAULT 0

### **Memberi Nilai ke variabel**

Terdapat beberapa cara memberi nilai ke variabel, yaitu dengan cara :

- SET

Contoh :

DECLARE total\_count INT DEFAULT 0

SET total\_count =0

- SELECT ... INTO

DECLARE total\_product INT DEFAULT 0

SELECT COUNT(\*) INTO total\_products FROM products

### **Ruang Lingkup Variabel**

Sebuah variabel hanya berlaku di dalam ruang lingkup nya masing-masing, yaitu di antara BEGIN dan END. Sebuah variabel yang diawali dengan tanda @, di sebut dengan variabel session, yang tetap ada hingga session berakhir.

### **Parameter dalam Stored Procedure**

Parameter, terdiri dari 3 bentuk, yaitu :

- IN

- OUT

- INOUT

IN : Mode Default. Dapat di gunakan di dalam sebuah stored procedure, namun stored procedure tidak dapat merubah nilai nya.

OUT : Parameter ini dapat di rubah oleh sebuah stored procedure yang di lewatinya.

INOUT : Dapat melewati stored procedure dan mendapatkan kembali nilai nya yang berbeda dari program yang memanggil.

Syntax untuk mendefinisikan sebuah parameter:

MODE nama\_parameter tipe\_parameter (ukuran parameter)

**Contoh :**

```
DELIMITER //
CREATE PROCEDURE tahun_mahasiswa(IN tahun_nrp VARCHAR (255))
BEGIN
    SELECT nrp,nama FROM mahasiswa WHERE left(nrp,2)=tahun_nrp
END //
DELIMITER ;
```

Dalam contoh di atas, kita mendapatkan, nrp dan nama yang filter nya adalah left(nrp,2)=tahun\_nrp

Untuk memanggil nya, kita menggunakan :  
CALL tahun\_mahasiswa('10')

**Conditional Control**

Di Bahasa Pemrograman, Conditional Control adalah hal yang umum keberadaannya. Stored Procedure juga memiliki Conditional Control tsb. Penggunaan Conditional Control, menjadikan kita dapat mengeksekusi sebuah perintah berdasarkan kriteria tertentu.

**IF Statement**

```
IF expression
    THEN commands
END IF
```

```
IF expression
    THEN commands
    ELSE commands
END IF
```

**Contoh :**

```
CREATE PROCEDURE procedure1 (IN parameter1 INTEGER)
BEGIN
    DECLARE variabel CHAR(10);
    IF parameter1=17 THEN
        SET variable1="birds";
    ELSE
        SET variable1="beasts";
    END IF
    INSERT INTO table1 VALUES (variable1);
END
```

**CASE Statement**

```
CASE
    WHEN expression THEN commands
    ...
    WHEN expression THEN commands
END CASE
```



Contoh :

```
SELECT
  nrp,nama
CASE
  WHEN left(nrp,2)="10" THEN SELECT "Tahun 2010"
  WHEN left(nrp,2)= "11" THEN SELECT "Tahun 2011"
END CASE
FROM mahasiswa
```

## Menghubungkan Database MySQL dengan PHP

Selain dapat di gunakan sendiri untuk pengolahan data, Database juga dapat di hubungkan dengan bahasa pemrograman untuk membuat sebuah aplikasi. Peran database di dalam pembuatan aplikasi adalah sama, tetap sebagai penyimpan data. Di dalam pembuatan aplikasi, terdapat berbagai macam operasi data dengan menggunakan bahasa SQL, seperti memasukkan , menampilkan dan menghapus data.

Sejak lama, database sudah menjadi penyimpanan data bagi aplikasi, selain file. MySQL sendiri dapat berhubungan dengan beberapa bahasa pemrograman, seperti Visual Basic, Gambas, Visual Foxpro, Java dan PHP.

Pada kesempatan kali ini, akan di tampilkan teknik menghubungkan bahasa pemrograman PHP dengan database Mysql. Penulis mengasumsikan, pembaca sudah dapat mengeksekusi file php dengan sebuah web server di dalam sebuah web browser.

### Script menghubungkan database dan menampilkan isi data

```
1.<?php
2.  $hubung_db=mysql_connect("localhost","root","opan");
3.  if (!$hubung_db)
4.      die ("Database MySQL tidak dapat di buka");
5.
6.  $pilih_db=mysql_select_db("test");
7.  if (! $pilih_db)
8.      die ("Database test tidak dapat di pilih");
9.
10.  $hasil=mysql_query("select * from mahasiswa");
11.
12.  while ($db=mysql_fetch_array($hasil))
13.  {
14.      echo $db["nrp"]."<br/>";
15.  }
16. ?>
```

Script di atas menerangkan tentang mengakses database dan menampilkan isi dari sebuah tabel.

Penjelasannya ada di bawah ini :

#### 1.<?php

Adalah tag pembuka script php. PHP di buka dengan <?php atau <? dan di tutup dengan ?>

2. \$hubung\_db=mysql\_connect("localhost","root","opan");

3. if (!\$hubung\_db)

4. die ("Database MySQL tidak dapat di buka");

Baris 2 → Script untuk mengakses database MySQL.

Baris 3 – 4 → Jika tidak terhubung, maka menampilkan pesan kegagalan.

Baris 6 – 8 → Memilih database dan jika database error terpilih, di tampilkan pesan kesalahan.

Baris 10 → Mengeksekusi sebuah query

Baris 12 – 15 → Mencetak hasil script

Script php dapat di gabungkan dengan script lain, seperti javascript, html dan css serta SQL. Dan untuk membuat sebuah aplikasi web yang tangguh, kita memang harus menggunakan script-script tsb. Pada pembahasan selanjut nya, akan di bahas mengenai Insert, Update dan Delete, dengan bantuan script html, selain php dan SQL.

### **Memasukkan Data ke dalam tabel**

- Membuat Form input data

```
1 <html>
2 <head>
3 <title>Entry Data</title>
4 </head>
5 <body>
6 <form method="post" action="proses.php">
7 <table>
8 <tr>
9 <td>Nrp</td><td>:</td><td><input type="text" name="vnrp" size="15"/></td>
10 </tr>
11 <tr>
12 <td>Nama</td><td>:</td><td><input type="text" name="vnama"
size="25"/></td>
13 </tr>
14 <tr>
15 <td colspan="3"><input type="submit" value="Simpan">
16 <input type="reset" value="Reset" >
17 </td>
18 </tr>
19 </table>
20 </form>
21 </body>
22 </html>
```

### **Keterangan :**

Script di atas adalah script html, yang membuat form untuk input data. Tampilan dari script di atas adalah :

Nrp :

Nama :

Ketika nrp dan nama telah di isi, lalu user menekan tombol simpan, maka akan menjalankan script php, dengan nama file proses.php.

```
1 <?php
2 $hubung_db=mysql_connect("localhost","root","opan");
3 if (!$hubung_db)
4 die ("Database MySQL tidak dapat di buka");
```

```
5
6 $pilih_db=mysql_select_db("test");
7 if (! $pilih_db)
8     die ("Database test tidak dapat di pilih");
9
10 if (isset($_POST['vnrp']))
11     { $xnrp=$_POST['vnrp']; }
12
13 if (isset($_POST['vnama']))
14     { $xnama=$_POST['vnama']; }
15
16 // Memasukkan data ke dalam tabel //
17 $query_masuk="insert into mahasiswa_a (nrp,nama) values ('$xnrp','$xnama')";
18 $jalan_query=mysql_query($query_masuk);
19 if ($jalan_query)
20     {
21     echo 'Data Berhasil di Simpan';
22     }
23 else
24     {
25     echo 'Data Gagal di simpan';
26     }
27 ?>
```

**Keterangan :**

Baris 1 – 3 : Menghubungkan Database MySQL

Baris 6 -8 : Memilih database

Baris 10 – 14 :

```
10 if (isset($_POST['vnrp']))
```

Memeriksa, apakah variabel vnrp sudah ada nilai nya atau belum.

```
11. $xnrp=$_POST['vnrp'];
```

Mengeset variabel \$xnrp dengan nilai dari variabel vnrp, dari form entry data.

Baris 13 – 14 : Sama dengan baris 10 – 14, namun yang di ambil adalah variabel vnama.

Baris 17 :

```
17 $query_masuk="insert into mahasiswa_a (nrp,nama) values ('$xnrp','$xnama')";
```

Query untuk memasukkan data dari form entry ke dalam tabel mahasiswa\_a

Baris 18 :

```
$jalan_query=mysql_query($query_masuk);
```

Menjalankan query

Baris 19 – 27 : Jika query berhasil di jalankan, tampilkan pesan sukses, jika gagal, tampilkan pesan gagal.

**Penutup :**

Demikianlah materi GUI I yang saya persembahkan. Semoga para mahasiswa dan pembaca modul ini dapat menarik manfaat nya. Modul ini tentu saja tidak lepas dari kekurangan di sana-sini, oleh karna itu, saran, masukan dan kritik sangat kami harapkan dan dapat di sampaikan melalui e-mail : [sofwan@sofwan.net](mailto:sofwan@sofwan.net)

Terima kasih atas perhatiannya dan Mohon maaf atas kekurangannya.

### **Referensi :**

1. "Wikipedia", <http://en.wikipedia.org/wiki/Database>, (19 Mar 11)
2. "Wikipedia", [http://en.wikipedia.org/wiki/Table\\_%28database%29](http://en.wikipedia.org/wiki/Table_%28database%29), (27 Mar 2011)
3. Chapple, Mike, "Primary Key Definition",  
<http://databases.about.com/cs/administration/g/primarykey.htm>, (27 Mar 2011)
4. SQL Tutorial, "SQL Database Table", <http://www.sql-tutorial.net/sql-database-table.asp>  
(27 Mar 2011)
5. "MySQL 5.0 Reference Manual," "Numeric Types",  
<http://dev.mysql.com/doc/refman/5.0/en/numeric-types.html>
6. <http://www.1keydata.com/sql/sqlselect.html>
7. Stored Procedure, "MySQL Stored Procedure", <http://www.mysqltutorial.org/getting-started-with-mysql-stored-procedures.aspx>
8. [http://www.experts-exchange.com/Database/MySQL/Q\\_22691771.html](http://www.experts-exchange.com/Database/MySQL/Q_22691771.html)
9. Kadir, Abdul, Dasar Pemrograman Web Dinamis menggunakan PHP, Yogyakarta: Andi Yogyakarta, 2002

### **Biografi Penulis**

**Akhmad Sofwan.** Menyelesaikan S1 di Universitas Budi Luhur (d/h STMIK Budi Luhur), Fakultas Teknologi Informasi, jurusan Teknik Informatika pada tahun 2001. Penulis saat ini adalah Fulltime Freelancer dalam pengembangan Web dan Aplikasi berbasis web dan konsultan Linux. Selain itu menulis buku Komputer dan mengajar di Perguruan Tinggi Raharja ([www.raharja.ac.id](http://www.raharja.ac.id)), Tangerang sebagai dosen tidak tetap.