

MODUL PRAKTIKUM

MATAKULIAH BASIS DATA



**DISUSUN OLEH:
TIM LABORATORIUM**

**SEKOLAH TINGGI ILMU MANAJEMEN INFORMATIKA
DAN KOMPUTER
STIMIK 10 NOPEMBER
JAYAPURA
2018**

Pengantar

Membangun basis data merupakan sebuah ketrampilan. Tidak ada cara yang lebih cepat dalam mempelajarinya, selain melalui praktikum. Ini adalah salah satu cara untuk mengasah keterampilan dalam membangun basis data. Basis data yang digunakan/diperkenalkan adalah basis data Client-Server MySQL.

Penyusun berusaha agar pembahasan pada modul-modul praktikum ini dapat langsung diterapkan pada komputer, tanpa penjelasan yang bertele-tele. Dengan demikian mahasiswa dapat langsung menerapkannya, dan menjadi terampil.

Diharapkan dengan modul ini mahasiswa dapat lebih memahami dan dapat mengimplementasikan basis data dalam pemograman, baik itu desktop maupun website.

Jayapura, Februari 2018

LAB. STIMIK 10 NOPEMBER

MODUL 1

Pengenalan MySQL

1. Tujuan

- Mengenal Lingkungan kerja MySQL
- Mahasiswa dapat Mengakses MySQL Melalui Command Prompt
- Mahasiswa memahami beberapa sintaks Bahasa DDL (*Data Defenition Languange*) dan dapat menggunakannya

2. Landasan Teori



MySQL adalah suatu perangkat lunak database relasi (Relational Database Management System atau RDBMS), seperti halnya ORACLE, Postgresql, MS SQL, dan sebagainya. MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang software dan konsultan database bernama MYSQL AB yang berada di Swedia. Waktu itu perusahaan tersebut masih bernama TcX DataKonsult AB, dan tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasis web pada client. MySQL menyebut produknya sebagai database open source terpopuler di dunia. Berdasarkan riset dinyatakan bahwa bahwa di platform Web, dan baik untuk kategori open source maupun umum, MySQL adalah database yang paling banyak dipakai. Menurut perusahaan pengembangnya, MySQL telah terpasang di sekitar 3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan MySQL bekerja siang malam memompa data bagi para pengunjungnya.

Aturan dan ketentuan dalam perintah MySQL

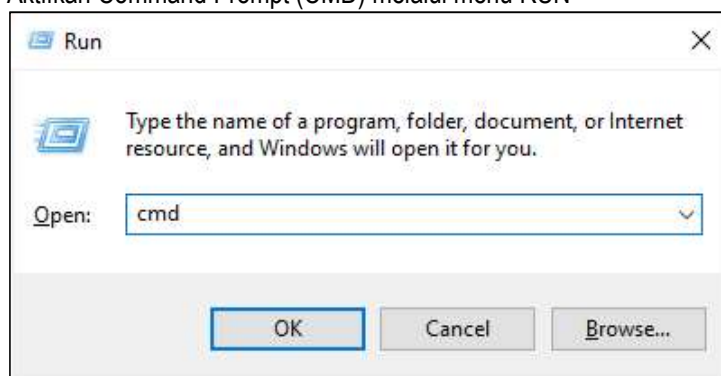
- Setiap perintah harus diakhiri dengan tanda titik koma, kecuali untuk perintah tertentu, misal : quit
- Setiap perintah akan disimpan dalam buffer (memori sementara) untuk menyimpan histori perintah-perintah yang pernah diberikan.
- Perintah dapat berupa perintah SQL atau perintah khusus MySQL.
- Perintah-perintah dalam lingkungan MySQL tidak menerapkan aturan case sensitive, tetapi case insensitive yaitu perintah bisa dituliskan dalam huruf besar atau pun huruf kecil.
- Aturan case sensitive diterapkan pada penamaan objek-objek dalam database seperti nama database atau nama table, namun aturan ini hanya ada dalam lingkungan Unix dan Linux.

Ada beberapa tanda yang sering muncul di 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 ("")

Langkah-langkah mengakses database melalui command prompr

- a. Aktifkan Command Prompt (CMD) melalui menu RUN



Ketik "cmd" pada kotak inputan kemudian klik "OK"

- b. Akses directory "xampp\mysql\bin" dengan perintah "cd\xampp\mysql\bin" seperti gambar di bawah ini

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.214]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Crow>cd\xampp\mysql\bin
C:\xampp\mysql\bin>_
```

- c. Aktifkan MySQL dengan perintah "mysql -uroot"

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.214]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Crow>cd\xampp\mysql\bin
C:\xampp\mysql\bin>mysql -uroot_
```

Jika di "enter" maka hasilnya akan seperti gambar di bawah ini:

```
C:\WINDOWS\system32\CMD.exe - mysql -uroot
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\LAB Soft 201>CD\XAMPP\MYSQL\BIN

C:\xampp\mysql\bin>mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.8 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

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

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

mysql> _
```

atau

```
C:\WINDOWS\system32\cmd.exe - mysql -uroot
Microsoft Windows [Version 10.0.16299.214]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Crow>cd\xampp\mysql\bin
C:\xampp\mysql\bin>mysql -uroot
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> _
```

Tampilan di atas menunjukkan bahwa kita telah terhubung dengan MySQL

3. Beberapa Peintah Dasar Pada Mysql

a. Membuat Database

Sintaks: **"create database <nama_database>;"**

Jika dieksekusi maka hasilnya seperti gambar di bawah ini:

```
MariaDB [(none)]> create database makanan;
Query OK, 1 row affected (0.32 sec)

MariaDB [(none)]> _
```

b. Menampilkan daftar Database dengan sintaks: **"show databases;"**

Seperti gambar di bawah ini:

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| dbpertanahan |
| information_schema |
| makanan |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
7 rows in set (0.26 sec)

MariaDB [(none)]> _
```

c. Menggunakan/mengaktifkan database yang telah dibuat

Sintaks: **"use <nama_database>;"**

Kita tidak dapat membuat table, proses CRUD jika tidak ada database yang diaktifkan.

```
MariaDB [(none)]> use makanan;
Database changed
MariaDB [makanan]> _
```

d. Menghapus Basisdata

sintaks: **"drop database <nama_database>;"**

```
MariaDB [makanan]> drop database makanan;
Query OK, 0 rows affected (0.61 sec)

MariaDB [(none)]> _
```

Untuk memastikan bahwa database sudah terhapus, lihat di daftar database dengan sintaks: **"show databases;"**

4. Kegiatan Praktikum

- Akses Database melalui Command Prompt (lakukan secara berulang-ulang sampai paham)
- Coba perintah DLL **"create"** sampai **"drop"** pada perintah di atas coba dengan nama database **"Mahasiswa"** (lakukan secara berulang-ulang sampai anda benar-memahaminya).
- Simpulkan Hasil praktikum anda pada modul ini pada sebuah kertas.

MODUL 2

Struktured Query Language (SQL)

1. Tujuan

- a. Mahasiswa dapat mengetahui dan menghafal bahasa SQL pada MySQL sehingga mahasiswa dapat menggunakannya perintah-perintah tersebut untuk mengakses Server MySQL guna membangun basis data.
- b. Mengerti dan memahami sintaks penulisan dari SQL.

2. Tugas Pendahuluan

- a. Apa yang ada ketahui tentang
 - Database
 - Tabel
 - Field
 - Record
- b. Uraikan beberapa engine yang ada pada Mysql

3. Landasan Teori

Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap *Database Management Systems* (DBMS), namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI. Artikel ini akan menggunakan bentuk paling umum yang dapat digunakan pada kebanyakan SDBD

DDL (Data Definition Language), DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut basis data, tabel, atribut(kolom), batasan-batasan terhadap suatu atribut, serta hubungan antar tabel. Yang termasuk dalam kelompok DDL ini adalah CREATE, ALTER, dan DROP.

Beberapa perintah dasar pada Mysql:

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel, seperti perintah Select, insert, delete, update, alter dan lain sebagainya

Di bawah ini beberapa perintah DDL dan DML

- a. Membuat Database
 - Sintaks: **"create database <nama_database>;"**
- b. Menampilkan daftar Database dengan sintaks: **"show databases;"**
- c. Menggunakan/mengaktifkan database yang telah dibuat
 - Sintaks: **"use <nama_database>;"**
- d. Membuat Tabel

Bentuk umum:

```
Create table <nama_tabel> (  
nama_field1 tipe_data [constraints],  
nama_field2 tipe_data [constraints],  
...  
)
```

Dengan:

nama_field adalah nama kolom (field) yang akan dibuat. Beberapa sistem manajemen basis data mengizinkan penggunaan spasi dan karakter nonhuruf pada nama kolom.

tipe_data tergantung implementasi sistem manajemen basis data. Misalnya, pada MySQL, tipe data dapat berupa VARCHAR, TEXT, BLOB, ENUM, dan sebagainya. constraints adalah batasan-batasan yang diberikan untuk tiap kolom. Ini juga

tergantung implementasi sistem manajemen basis data, misalnya NOT NULL, UNIQUE, dan sebagainya. Ini dapat digunakan untuk mendefinisikan kunci primer (primary key) dan kunci asing (foreign key). Satu tabel boleh tidak memiliki kunci primer sama sekali, namun sangat disarankan mendefinisikan paling tidak satu kolom sebagai kunci primer.

Contoh:

```
MariaDB [krs]> create table mahasiswa(
-> npm varchar(9) NOT NULL,
-> nama varchar(30) NOT NULL,
-> tempatlahir varchar(50) NOT NULL,
-> tgllahir date,
-> alamat text NOT NULL,
-> kontak varchar(30) NOT NULL,
-> primary key(npm))
-> engine=innodb;
Query OK, 0 rows affected (0.81 sec)

MariaDB [krs]>
```

Untuk menampilkan daftar tabel dalam database digunakan perintah:

"show tables;" seperti di bawah ini

```
MariaDB [krs]> show tables;
+-----+
| Tables_in_krs |
+-----+
| mahasiswa     |
+-----+
1 row in set (0.10 sec)

MariaDB [krs]>
```

Untuk melihat struktur tabel gunakan perintah "desc <nama_tabel>" seperti contoh di bawah ini:

```
MariaDB [krs]> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| npm   | varchar(9) | NO | PRI | NULL |  |
| nama  | varchar(30) | NO |  | NULL |  |
| tempatlahir | varchar(50) | NO |  | NULL |  |
| tgllahir | date | YES |  | NULL |  |
| alamat | text | NO |  | NULL |  |
| kontak | varchar(30) | NO |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.22 sec)

MariaDB [krs]>
```

4. Kegiatan Praktikum

- Praktekkan sintaks sql di atas secara berulang-ulang sampai anda hafal dan paham maksud penggunaannya.
- Simpulkan hasil praktikum anda

MODUL 3 TYPE DATA FIELD

1. TUJUAN

- a. Mahasiswa mengetahui dan memahami type-type data pada MySQL.
- b. Mahasiswa dapat menggunakannya type-type data pada MySQL.
- c. Mahasiswa dapat lebih memahami sintaks sql

2. Tugas Pendahuluan

- a. Pelajari type data yang ada pada mysql.
- b. Cari fungsi dan kegunaan dari constraints "AUTO INCREMENT"

3. Lansan Teori

a. Tipe Data Numerik

Tipe data numerik yaitu tipe data yang digunakan untuk menyimpan data numerik (angka).

Nama	Fungsi	Jangkauan	Ukuran
TINYINT	Menyimpan data bilangan bulat positif <u>dan</u> negatif.	-128 s/d 127	1 byte (8 bit).
SMALLINT	menyimpan data bilangan bulat positif <u>dan</u> negatif.	: -32.768 s/d 32.767	: 2 byte (16 bit).
MEDIUMINT	menyimpan data bilangan bulat positif <u>dan</u> negatif.	-8.388.608 s/d 8.388.607	Ukuran : 3 byte (24 bit).
INT	menyimpan data bilangan bulat positif <u>dan</u> negative	-2.147.483.648 s/d 2.147.483.647	4 byte (32 bit).
BIGINT	menyimpan data bilangan bulat positif dan negatif.	$\pm 9,22 \times 10^{18}$	8 byte (64 bit).
FLOAT	menyimpan data bilangan pecahan positif dan negatif presisi tunggal	-3.402823466E+38 s/d -1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+38.	4 byte (32 bit)
DOUBLE	menyimpan data bilangan pecahan positif dan negatif presisi ganda.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit)
REAL	menyimpan data bilangan pecahan positif dan negatif presisi ganda.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).
DECIMAL	menyimpan data bilangan pecahan positif dan negatif.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).
NUMERIC	menyimpan data bilangan pecahan positif dan negatif.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).

b. Tipe Data Date dan Time

Tipe data date dan time yaitu tipe data yang digunakan untuk menyimpan data tanggal dan waktu.

Nama	Fungsi	Jangkauan	Ukuran
DATE	menyimpan data tanggal	1000-01-01 s/d 9999-12-31 (YYYY-MM-DD)	3 byte.

TIME	menyimpan data waktu	-838:59:59 s/d +838:59:59 (HH:MM:SS)	3 byte
DATETIME	menyimpan data tanggal dan waktu.	'1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'	8 byte
YEAR	menyimpan data tahun dari tanggal	1900 s/d 2155	1 byte

c. Tipe Data String (Text)

Tipe data string yaitu tipe data yang digunakan untuk menyimpan data string (text).

Nama	Fungsi	Jangkauan
CHAR	menyimpan data string ukuran tetap.	0 s/d 255 karakter
VARCHAR	menyimpan data string ukuran dinamis.	0 s/d 255 karakter (versi 4.1), 0 s/d 65.535
TINYTEXT	menyimpan data text.	0 s/d 255 karakter (versi 4.1), 0 s/d 65.535
TEXT	menyimpan data text.	0 s/d 65.535
MEDIUMTEXT	menyimpan data text	0 s/d 224 - 1 karakter
LONGTEXT	menyimpan data text.	0 s/d 232 - 1 karakter

d. Tipe Data BLOB (Biner)

Tipe data blob digunakan untuk menyimpan data biner.

Nama	Fungsi	Jangkauan
BIT	Menyimpan data biner.	64 digit biner
TINYBLOB	menyimpan data biner/ Gambar ukuran kecil	255 byte
BLOB	Menyimpan data biner/ Gambar	4
MEDIUMBLOB	Menyimpan data biner/ Gambar kuran sedang	224-1 byte
LOB	Menyimpan data biner/ Gambar ukuran besar	232- 1 byte

e. Tipe Data yang Lain

Selain tipe data di atas, MySQL juga menyediakan tipe data yang lain, diantaranya adalah :

Nama	Fungsi	Jangkauan
ENUM	enumerasi (kumpulan data).	sampai dengan 65535 string.
SET	combination (himpunan data).	sampai dengan 255 string anggota

4. Kegiatan Praktikum

- Buat basisdata dengan nama "krs" kemudian aktifkan basisdata tersebut
- Buat tabel-tabel di bawah ini:

Tabel "mahasiswa"

No.	Field	Type Data	Width	Key
1.	Npm	Varchar	9	*
2.	Nm_Mhs	Varchar	30	
3.	Alamat	Varchar	50	
4.	No_kontak	Varchar	13	
5.	Tmp_lahir	Varchar	20	

6.	Tgl_lahir	Date		
7.	Jurusan	Varchar	50	

Tabel Matakuliah

No.	Field	Type Data	Width	Key
1.	Kdmk	Varchar	6	*
2.	Nm_Matakuliah	Varchar	100	
3.	Sks	int	1	
4.	Semester	Int	1	

Tabel krs

No.	Field	Type Data	Width	Key
1.	Npm	Varchar	9	*
2.	Kdmk	Varchar	6	*
3.	Thn_ajaran	Varchar	9	*
4.	Semester	Varchar	6	
5.	Dosen_wali	Varchar	50	

Tabel Dosen

No.	Field	Type Data	Width	Key
1.	Npd	Varchar	10	*
2.	Nm_Dosen	Varchar	50	
3.	Alamat	Varchar	50	
4.	No_kontak	Varchar	13	
5.	Tmp_lahir	Varchar	20	
6.	Tgl_lahir	Date		

Tabel Mengajar

No.	Field	Type Data	Width	Key
1.	Npd	Varchar	9	*
2.	Kdmk	Varchar	6	*
3.	Thn_Ajaran	Varchar	9	*

- Simpulkan hasil praktikum anda

MODUL 4

Mengubah Struktur Tabel

1. Tujuan

- Mahasiswa dapat mengubah struktur tabel sesuai kebutuhan.
- Mahasiswa dalam lebih memahami perintah pada MySQL

2. Tugas Pendahuluan

- Hafalkan macam-macam “[perubahan]” pada perintah “Alter” beserta fungsinya
- Buat laporan mengenai cara penggunaan alter

3. Landasan Teori

Perintah Perubahan Struktur Tabel pada MySQL

```
alter_specification:
  table_options
  | ADD [COLUMN] col_name column_definition
    [FIRST | AFTER col_name ]
  | ADD [COLUMN] (col_name column_definition,...)
  | ADD {INDEX|KEY} [index_name]
    [index_type] (index_col_name,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]] PRIMARY KEY
    [index_type] (index_col_name,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]]
    UNIQUE [INDEX|KEY] [index_name]
    [index_type] (index_col_name,...) [index_option] ...
  | ADD FULLTEXT [INDEX|KEY] [index_name]
    (index_col_name,...) [index_option] ...
  | ADD SPATIAL [INDEX|KEY] [index_name]
    (index_col_name,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]]
    FOREIGN KEY [index_name] (index_col_name,...)
    reference_definition
  | ALGORITHM [=] {DEFAULT|INPLACE|COPY}
  | ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
  | CHANGE [COLUMN] old_col_name new_col_name column_definition
    [FIRST|AFTER col_name]
  | LOCK [=] {DEFAULT|NONE|SHARED|EXCLUSIVE}
  | MODIFY [COLUMN] col_name column_definition
    [FIRST | AFTER col_name]
  | DROP [COLUMN] col_name
  | DROP PRIMARY KEY
  | DROP {INDEX|KEY} index_name
  | DROP FOREIGN KEY fk_symbol
  | DISABLE KEYS
  | ENABLE KEYS
  | RENAME [TO|AS] new_tbl_name
  | ORDER BY col_name [, col_name] ...
  | CONVERT TO CHARACTER SET charset_name [COLLATE collation_name]
  | [DEFAULT] CHARACTER SET [=] charset_name [COLLATE [=] collation_name]
  | DISCARD TABLESPACE
  | IMPORT TABLESPACE
  | FORCE
  | ADD PARTITION (partition_definition)
  | DROP PARTITION partition_names
  | TRUNCATE PARTITION {partition_names | ALL}
  | COALESCE PARTITION number
  | REORGANIZE PARTITION partition_names INTO (partition_definitions)
  | EXCHANGE PARTITION partition_name WITH TABLE tbl_name
  | ANALYZE PARTITION {partition_names | ALL}
  | CHECK PARTITION {partition_names | ALL}
  | OPTIMIZE PARTITION {partition_names | ALL}
  | REBUILD PARTITION {partition_names | ALL}
```

4. Kegiatan Praktikum

- Ulangi Kegiatan praktikum pada modul 3 point 1 dan 2
- Diberikan saat kegiatan praktikum berlangsung

MODUL 5 MENYISIPKAN DATA

1. Tujuan

- Mahasiswa dapat menyisipkan satu atau lebih data ke dalam tabel MySQL
- Mahasiswa dapat lebih memahami sintaks SQL

2. Tugas Pendahuluan

Pelajari dan pahami modul 5

3. Landasan Teori

Bentuk umum:

```
"insert into <nama_tabel> (field1, field2, ...,n) value
(value1,value2, ..., n);"
```

Contoh menyisipkan 1 data:

```
MariaDB [krs]> insert into mahasiswa(
-> npm, nama, tempatlahir, tgllahir, alamat, kontak)
-> value(
-> "201211056",
-> "Ajenkris Y. Kungkung",
-> "Makassar",
-> "1994-03-26",
-> "Blackland",
-> "082238281801");
Query OK, 1 row affected, 1 warning (0.06 sec)

MariaDB [krs]>
```

Contoh menyisipkan lebih dari 1 data:

```
MariaDB [makanan]> insert into mahasiswa
-> (npm, nama, tempatlahir, tgllahir, alamat, kontak)
-> values
-> ("201211056", "Ajenkris Y. K", "Makassar", "1994-03-26", "Blackland", "082238281801"),
-> ("201211009", "Amrin", "Jayapura", "1993-09-03", "Sentani", "08108381711");
Query OK, 2 rows affected (0.09 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [makanan]>
```

Untuk melihat data yang telah disisipkan ke dalam table gunakan perintah:

```
MariaDB [makanan]> select * from mahasiswa;
+----+-----+-----+-----+-----+-----+
| npm | nama      | tempatlahir | tgllahir | alamat   | kontak      |
+----+-----+-----+-----+-----+-----+
| 201211009 | Amrin      | jayapura    | 1993-09-03 | Sentani  | 08108381711 |
| 201211056 | Ajenkris Y. K | Makassar    | 1994-03-26 | Blackland | 082238281801 |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [makanan]>
```

Contoh di atas adalah perintah untuk menyisipkan data pada tabel mahasiswa yang telah di buat pada modul 2

4. Kegiatan Praktikum

- Ulangi Kegiatan praktikum pada modul 3 point 1 dan 2
- Isi tabel mahasiswa yang telah di buat dengan data di bawah ini:

Npm	Nm_mhs	Alamat	No_kontak	Tmp_lahi	Tgl_lahir	Jurusan
200711012	Dedy Salmon K.	Dok IX	081344550001	Jayapura	12-01-1989	Tek. Info S1
200711025	Wali Wonda	Dok V	081345456789	Malang	25-10-1990	Tek. Info S1
200711039	Yohanes Woppy	Klofkamp	085244551234	Timika	28-09-1990	Tek. Info S1
200711052	Desena Kogoya	Dok II	0811456345	Oksibil	24-03-189	Tek. Info S1

200711031	Yedida Amelia	Hamadi	-	Bandung	25-07-1988	Tek. Info S1
200711023	Emy Lenora T.	Polimak	081344550987	Ambon	29-05-1990	Tek. Info S1

- c. Ulangi point “b” secara berulang-ulang sampai anda paham
d. Kerjakan dirumah untuk tabel yang lain:

Tabel Dosen

Npd	Nm_dosen	Alamat	No_kontak	Tmp_lahir	Tgl_lahir
D-001	Siti Nur Alam, ST.	Dok IX	081344550012	Jayapura	12-01-1975
D-002	Rahmat H. Kiswanto, ST	Dok V	081345456798	Malang	25-10-1977
D-003	Syaiful Rizal Sjaib, ST	Klofkamp	085244551243	Timika	28-09-19
D-004	Tengadi Boney Bun, ST.	Dok II	0811456355	Oksibil	24-03-189

Tabel krs

Npm	Kdmk	Thn_ajaran	Semester	Dosen_Wali
200711012	BAD340	2008/2009	Genap	Siti Nur Alam, ST.
200711012	CSD140	2008/2009	Genap	Siti Nur Alam, ST.
200711039	BAD340	2008/2009	Genap	Siti Nur Alam, ST.
200711052	BAD340	2008/2009	Genap	Rahmat H. Kiswanto, ST
200711052	BHC240	2008/2009	Genap	Rahmat H. Kiswanto, ST
200711052	CBP140	2008/2009	Genap	Rahmat H. Kiswanto, ST
200711031	PAK340	2008/2009	Genap	Syaiful Rizal Sjaib, ST
200711023	STD340	2008/2009	Genap	Tengadi Boney Bun, ST.
200711023	STD340	2008/2009	Genap	Tengadi Boney Bun, ST.

Tabel Mengajar

Npd	Kdmk	Thn_Ajaran
D-001	CBP140	2008/2009
D-002	BHC240	2008/2009
D-003	PAK340	2008/2009
D-004	STD340	2008/2009
D-005	BAD340	2008/2009

- e. Diberikan saat kegiatan praktikum

MODUL 6

Select, Update dan Delete

1. Tujuan

- Mahasiswa dapat memahami perintah untuk Select, Update dan Delete record pada tabel.
- Mahasiswa dapat menggunakan perintah Select, Update dan Delete

2. Tugas Pendahuluan

- Pelajari dan pahami modul 6
- Pada perintah select terdapat beberapa atribut "join" uraikan fungsi dan kegunaan dari "join" (tulis tangan pada folio bergaris)

3. Landasan Teori

a. Perintah "Select"

Perintah "select" digunakan untuk menampilkan isi dari tabel, baik itu satu tabel atau lebih dari satu tabel yang saling berelasi.

Bentuk umum:

```
"SELECT [ * (all) | <nama_kolom> | <fungsi_aritmatika>]  
FROM [nama_tabel] [where <kondisi> (optional)]"
```

Contoh 1 menampilkan seluruh data dan seluruh kolom:

```
MariaDB [makanan]> select * from mahasiswa;  
+-----+-----+-----+-----+-----+-----+  
| npm      | nama      | tempatlahir | tgllahir  | alamat    | kontak    |  
+-----+-----+-----+-----+-----+-----+  
| 201211009 | Amrin     | jayapura    | 1993-09-03 | Sentani   | 08108381711 |  
| 201211056 | Ajenkris Y. K | Makassar    | 1994-03-26 | Blackland | 082238281801 |  
+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```

Contoh 2 menampilkan data berdasarkan kondisi dan beberapa kolom

```
MariaDB [makanan]> select npm, nama, kontak from mahasiswa where alamat = 'Sentani';  
+-----+-----+-----+  
| npm      | nama      | kontak    |  
+-----+-----+-----+  
| 201211009 | Amrin     | 08108381711 |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

Perintah di atas mengambil data mahasiswa yang tinggal di "Sentani" dan menampilkan hanya kolom npm, nama, kontak

Contoh 3 menampilkan data dari tabel yang berelasi berdasarkan kondisi

```
MariaDB [laboratorium]> select matakuliah.nmmk, matakuliah.smt, absen.shift  
-> from matakuliah, mhs, absen  
-> where  
-> mhs.npm=absen.npm  
-> and  
-> matakuliah.kdmk=absen.kdmk  
-> and mhs.npm='201211056';  
+-----+-----+-----+  
| nmmk      | smt | shift |  
+-----+-----+-----+  
| Prakt. Struktur Data | 4 | S |  
| Prakt. Basis Data    | 4 | S |  
| Prakt. Komputasi Bergerak | 4 | S |  
| Prakt. Bahasa Rakitan dan Mikroprosesor | 5 | S |  
| Prakt. Jaringan Komputer | 6 | S |  
| Prakt. Interaksi Manusia dan Komputer | 6 | S |  
| Prakt. Sistem Pakar   | 7 | S |  
+-----+-----+-----+  
7 rows in set (0.00 sec)
```

Perintah di atas dapat dilakukan dengan perintah join seperti di barah ini:

```
MariaDB [laboratorium]> SELECT
  -> matakuliah.nmmk,
  -> matakuliah.smt,
  -> absen.shift
  -> FROM
  -> absen
  -> INNER JOIN mhs ON mhs.npm = absen.npm
  -> INNER JOIN matakuliah ON matakuliah.kdmk = absen.kdmk
  -> WHERE
  -> mhs.npm = '201211056';
```

nmmk	smt	shift
Prakt. Struktur Data	4	S
Prakt. Basis Data	4	S
Prakt. Komputasi Bergerak	4	S
Prakt. Bahasa Rakitan dan Mikroprosesor	5	S
Prakt. Jaringan Komputer	6	S
Prakt. Interaksi Manusia dan Komputer	6	S
Prakt. Sistem Pakar	7	S

```
7 rows in set (0.00 sec)

MariaDB [laboratorium]>
```

Keterangan contoh 3

Perintah di atas digunakan jika ingin menampilkan matakuliah praktikum yang pernah dikontrak oleh mahasiswa dengan npm "201211056";

b. Perintah "Update"

Perintah update digunakan untuk memodifikasi atau merubah data pada suatu kolom yang ada pada sebuah tabel

Bentuk umum:

```
"UPDATE [nama_tabel] SET [kolom1 = nilai1, kolom2 =
nilai2,..., n] [where<kondisi>];"
```

Contoh:

Perhatikan data pada contoh 1 pont a. data kedua adalah mahasiswa dengan nama "Ajenkris Y. Kungkung" kita ingin rubah menjadi "Wahyu Sidiq A. Prakoso".

```
MariaDB [laboratorium]> use makanan
Database changed
MariaDB [makanan]> update mahasiswa set
  -> nama = "Wahyu Sidiq A. Prakoso",
  -> alamat="Entrop",
  -> kontak="0821232234"
  -> where npm="201211056";
Query OK, 1 row affected (0.09 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [makanan]>
```

Tampilkan data mahasiswa dengan perintah "Select * from mahasiswa" untuk melihat perubahan

```
MariaDB [makanan]> select * from mahasiswa;
```

npm	nama	tempatlahir	tgllahir	alamat	kontak
201211009	Amrin	Jayapura	1993-09-03	Sentani	08108381711
201211056	Wahyu Sidiq A. Prakoso	Makassar	1994-03-26	Entrop	0821232234

```
2 rows in set (0.00 sec)

MariaDB [makanan]>
```


c. Perintah “Delete”

Perintah delete digunakan untuk menghapus data dari tabel pada MySQL

Bentuk umum

`“DELETE FROM [nama_tabel] WHERE [kondisi];”`

Contoh:

Kita ingin menghapus data mahasiswa yang memiliki npm “201211056” dari tabel mahasiswa.

```
MariaDB [makanan]> delete from mahasiswa where npm='201211056';  
Query OK, 1 row affected (0.09 sec)
```

```
MariaDB [makanan]>
```

Tampilkan data pada tabel mahasiswa untuk memastikan

```
MariaDB [makanan]> select * from mahasiswa;
```

npm	nama	tempatlahir	tgllahir	alamat	kontak
201211009	Amrin	jayapura	1993-09-03	Sentani	08108381711

1 row in set (0.00 sec)

```
MariaDB [makanan]>
```

4. Kegiatan Praktikum

- Tampilkan data dari tabel-tabel yang telah diisi pada modul 5
- Analisa keseluruhan tabel kemudian tampilkan informasi yang dapat ditampilkan dari tabel-tabel yang berelasi
- Coba rubah beberapa data dari tabel yang ada dengan perintah “Update”
- Coba hapus beberapa data dari tabel yang ada dengan perintah “Delete”
- Tarik kesimpulan dari hasil praktikum

MODUL 7

GUI phpMyAdmin

1. Tujuan

- Mahasiswa dapat mengakses phpMyAdmin melalui Browser
- Mahasiswa dapat membuat database dan tabel melalui phpMyAdmin

2. Tugas Pendahuluan

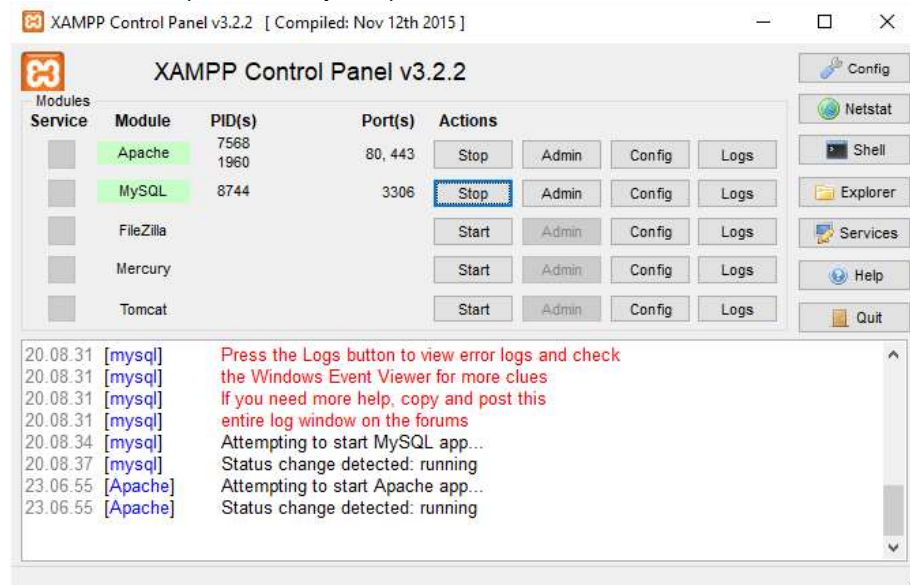
Siapkan database yang telah dinormalisasi pada pertemuan teori

3. Landasan Teori

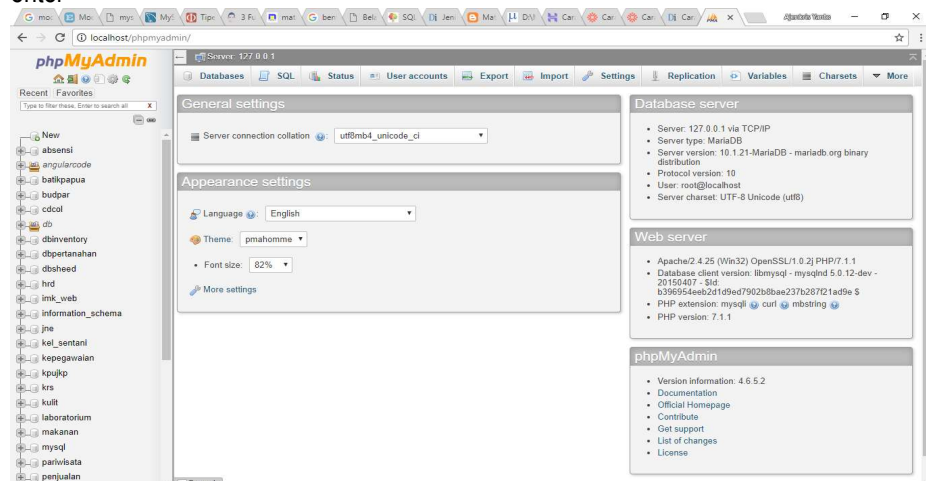
Phpmyadmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyadmin, anda dapat membuat database, membuat tabel, menginsert, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual.

Langkah-langkah untuk membuka phpMyAdmin adalah sebagai berikut:

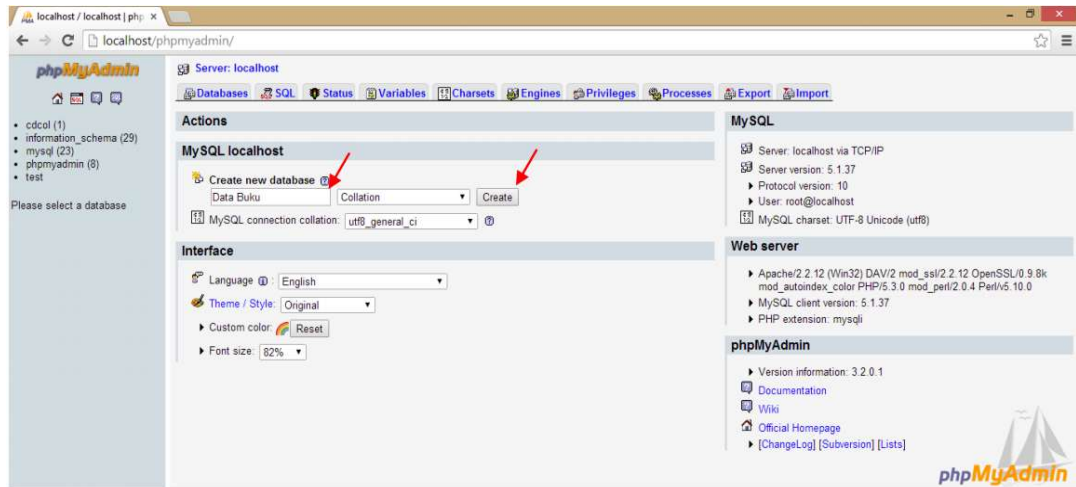
- Aktifkan server apache dan MySQL pada XAMPP



- Buka browser dan ketikkan "localhost/phpmyadmin" pada address bar kemudian enter



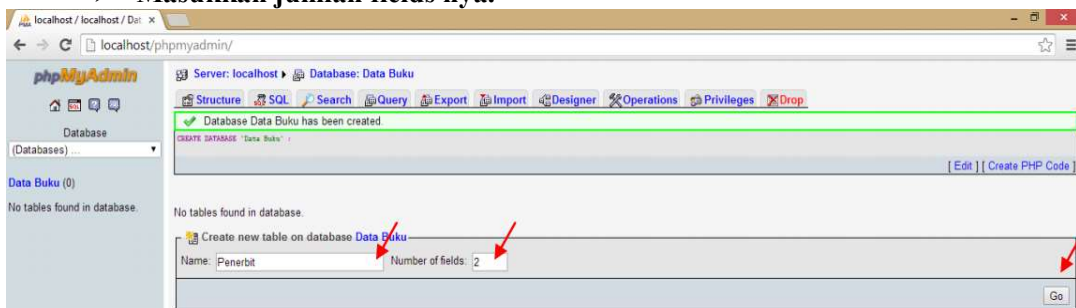
c. Membuat Database



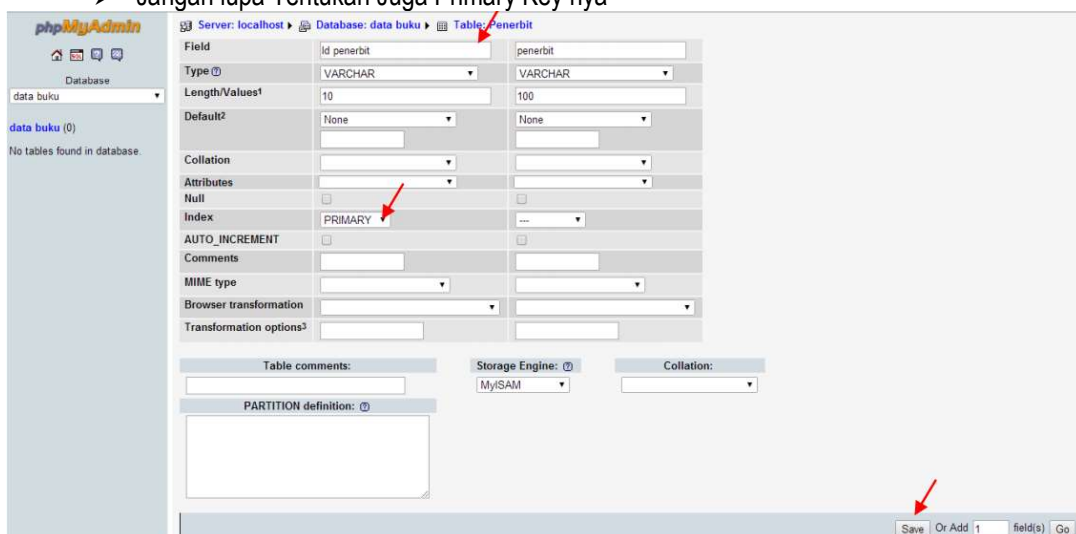
Pada kolom “create new database” ketikkan nama database yang akan dibuat, kemudian klik create untuk membuat database

d. Membuat tabel

- Pilih Database “Data Buku” dan masukkan nama tabel pada kolom Create new table on database data buku yang tersedi.
- **Masukkan jumlah fields nya.**



- **Isilah nama field, pilih jenis type dan isi jumlah length nya.**
- **Jangan lupa Tentukan Juga Primary Key nya**



- Kemudian klik save

4. Kegiatan Praktikum

- Buat Database yang sudah dinormalisasi dengan GUI phpMyAdmin
- Simpulkan

MODUL 8

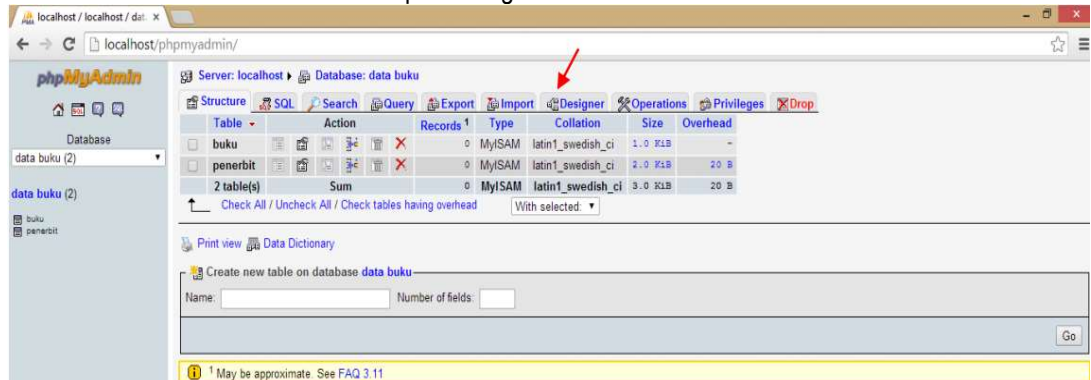
Relasi Tabel dan Insert Data

1. Tujuan

- Mahasiswa dapat membuat relasi tabel melalui phpMyAdmin
- Mahasiswa dapat menambahkan, Mengubah, Menghapus data melalui phpMyAdmin

2. Landasan Teori

- Langkah-langkah pembuatan relasi
 - Pilih Database kemudian pilih Designer

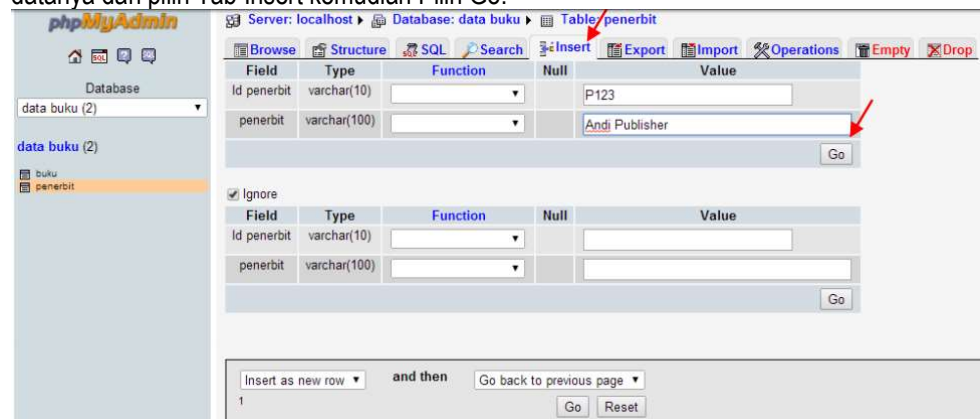


- Pilih simbol relasi kemudian pilih Primary Key pada tabel pertama dan hubungkan dengan memilih Foreign Key pada tabel kedua.



b. Menyisipkan Data

Setelah kita membuat Relasi, selanjutnya kita akan mencoba memasukan data. Memasukan data, dapat dilakukan dengan cara : Pilih tabel yang akan dimasukan datanya dan pilih Tab Insert kemudian Pilih Go.



3. Kegiatan Praktikum

- Sisipkan beberapa record data ke dalam tabel yang telah di buat
- Tarik kesimpulan

MODUL 9

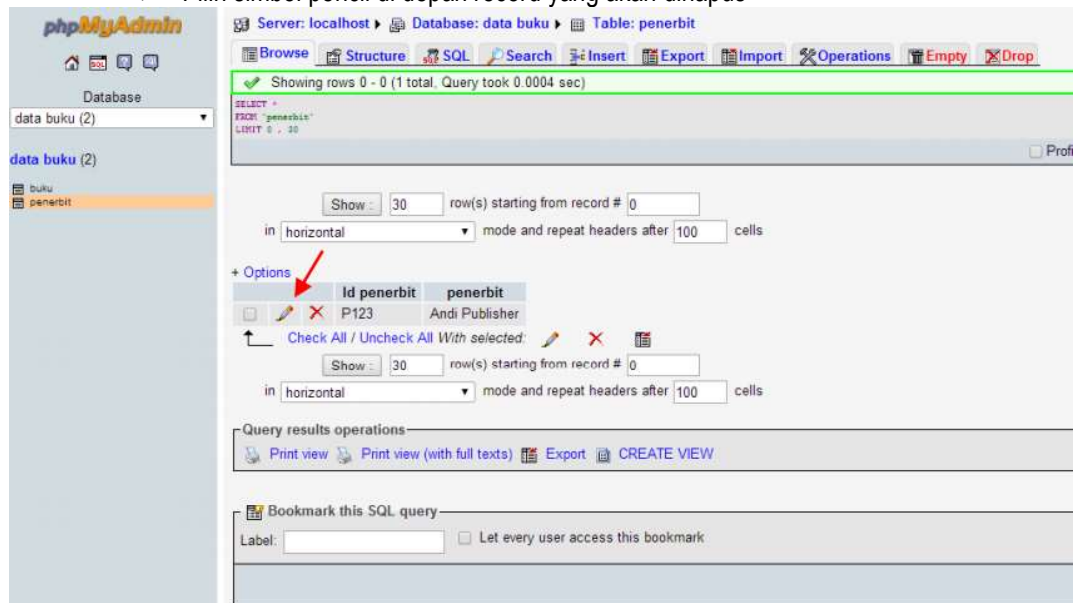
Mengubah dan menghapus Data

1. Tujuan

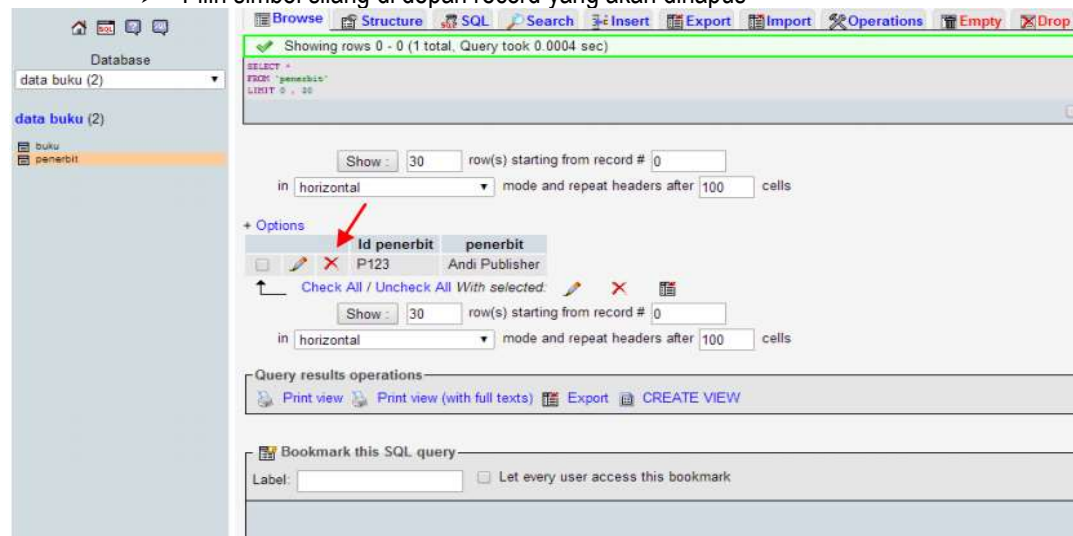
- Mahasiswa dapat memahami cara mengubah dan menghapus data melalui phpMyAdmin
- Mahasiswa dalam mempraktekkan cara mengubah dan menghapus data melalui phpMyAdmin

2. Landasan Teori

- Cara mengubah Data
 - Pilih tabel yang akan diubah datanya dan pilih Tab Browse.
 - Pilih simbol pencil di depan record yang akan dihapus



- Ubah data dan pilih Go
- Menghapus Data
 - Pilih tabel yang akan dihapus datanya dan pilih Tab Browse.
 - Pilih simbol silang di depan record yang akan dihapus



- Kemudian Pilih Go

DAFTAR ACUAN

- [1] <http://id.wikipedia.org/wiki/SQL> (February 20, 2014)
- [2] Hengky W. Pramana, [Database Desktop versi 7.0](#), Jakarta : PT. Elex Media Komputindo, 1999.
- [3] Jim Lahallo, S.T., M.Kom, Modul Praktikum Basis Data, Jayapura, STIMIK 10 Nopember Jayapura