

MODUL PRAKTIKUM
PEMROGRAMAN WEB



Materi 8 dan 9:
Database dan SQL

Dosen:

Febrianta Surya Nugraha, M.Kom

S1 INFORMATIKA
STMIK AMIKOM SURAKARTA
2020

Database dan SQL

1. PENDAHULUAN

Basis data (database) dapat dikatakan sebagai sekumpulan data yang disimpan, saling berhubungan, dan diorganisasi secara bersama. Perangkat lunak yang dapat digunakan untuk mengolah basis data sudah cukup banyak, tetapi pada pembahasan kali ini akan menggunakan MySQL. MySQL dipilih karena cukup mudah dipelajari dan bersifat opensource, serta mendukung banyak platform sistem operasi. Selain itu, MySQL juga memiliki dukungan komunitas yang banyak.

SQL (Structure Query Language)

SQL adalah bahasa pemrograman yang digunakan untuk mengakses data dalam basis data relasional. SQL biasanya berupa perintah sederhana yang berisi instruksi manipulasi data. Perintah ini sering disebut dengan *query*.

SQL dikenalkan pertama kali dalam IBM pada tahun 1970 dan sebuah standar ISO dan ANSI ditetapkan untuk SQL. Standar ini tidak tergantung software yang digunakan. Beberapa perintah SQL yang digunakan adalah:

a. Data Definition Language (DDL)

Data Definition Language (DDL) adalah perintah SQL yang digunakan untuk mendefinisikan sebuah obyek di dalam database dan tabel. Beberapa perintah dasar dalam DDL antara lain **CREATE**, **ALTER**, **RENAME**, **DROP**.

b. Data Modification Language (DML)

Data Modification Language (DML) adalah perintah SQL yang digunakan untuk melakukan manipulasi terhadap obyek yang ada di dalam database. Perintah SQL yang termasuk di dalam DML antara lain **SELECT**, **INSERT**, **UPDATE**, **DELETE**.

c. Data Control Language (DCL)

Data Control Language (DCL) adalah perintah SQL yang berhubungan dengan manipulasi user dan hak akses(priviledges). Perintah SQL dalam DCL antara lain **GRANT**, **REVOKE**.

2. MYSQL DENGAN COMMAND PROMPT

1. Masuk MySQL dengan Command Prompt

Untuk mengakses MySQL, kita dapat menggunakan command prompt. Akan tetapi, sebelum membuka MySQL pastikan terlebih dahulu servis MySQL pada XAMPP sudah diaktifkan. Selanjutnya buka command prompt dan masuk ke dalam direktori tempat MySQL telah diinstall. Berikut langkah-langkahnya.

1. Ketikkan: **cd C:\xampp\mysql\bin** di tempat XAMPP telah telah diinstal.
2. Jika sudah masuk ke direktori MySQL, ketikkan perintah:

mysql -u "namauser" -p "password"

untuk dapat masuk ke MySQL. *Setting default* MySQL biasanya tanpa password. Maka perintah masuk ke MySQL cukup sampai nama user saja: **mysql -u root**.

```
Command Prompt - mysql -u root
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Ubing>cd C:\xampp\mysql\bin

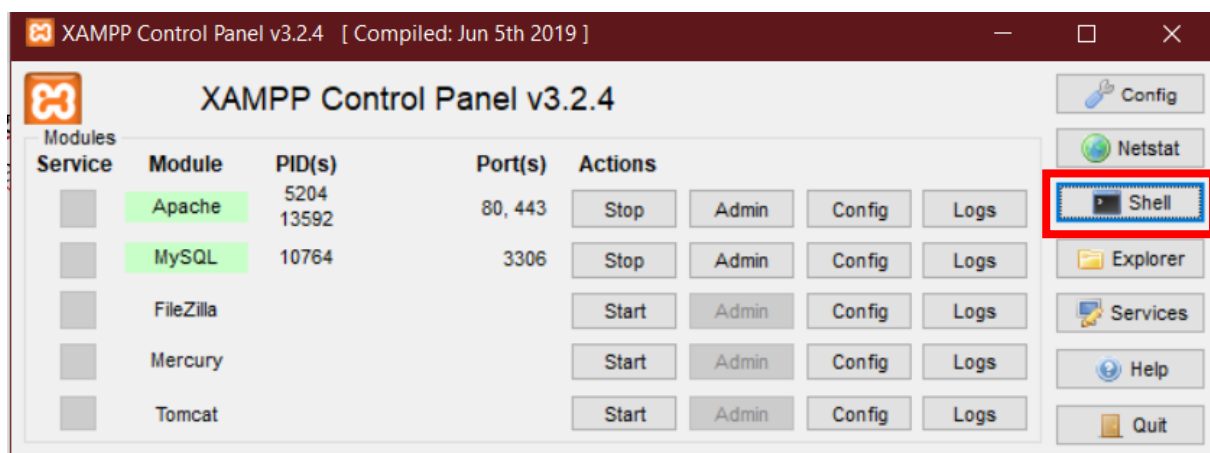
C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 43
Server version: 10.4.6-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Cara yang lebih ringkas anda dapat mengklik tombol shell pada control panel xampp kemudian login ke dalam mysql menggunakan ke 2 tahapan diatas.



```
XAMPP for Windows - mysql -u root
Ubing@CYBORG c:\xampp
# cd C:\xampp\mysql\bin

Ubing@CYBORG C:\xampp\mysql\bin
# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.6-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

2. DDL Menggunakan *Command Prompt*

2.1. *Membuat Database*

Sintaks umum SQL dalam membuat sebuah database adalah sebagai berikut.

CREATE DATABASE [nama_database];

Aturan penamaan database sama dengan aturan penamaan pada variabel yang terdiri dari huruf, angka, dan underscore. Penamaan database tidak boleh sama dengan database yang sudah ada. Jika penamaan sama, maka akan dimunculkan pesan error.

Latihan buatlah sebuah database contoh:

```
MariaDB [(none)]> create database data01;  
Query OK, 1 row affected (0.002 sec)
```

2.2. Menampilkan Database

Sintaks umum SQL untuk menampilkan database yang telah dibuat adalah seperti berikut.

SHOW DATABASES;

```
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| data01   |  
| information_schema |  
| mysql    |  
| performance_schema |  
| phpmyadmin |  
| test     |  
+-----+  
6 rows in set (0.001 sec)
```

2.3. Menghapus Database

Seorang administrator dapat menggunakan hak aksesnya untuk menghapus sebuah database yang memang sudah terpakai. Perintah umum untuk menghapus database adalah sebagai berikut.

DROP DATABASE [nama_database];

Perintah hapus database memiliki risiko yang tinggi. Karena **sebuah database yang dihapus akan hilang beserta data yang ada di dalamnya**. Sehingga seorang administrator harus sangat berhati-hati dalam menggunakan perintah drop. Perintah hapus database akan tidak berjalan ketika database yang dimaksud tidak ada.

```
MariaDB [(none)]> create database data02;  
Query OK, 1 row affected (0.002 sec)  
  
MariaDB [(none)]> drop database data02;  
Query OK, 0 rows affected (0.001 sec)
```

2.4. Mengaktifkan Database

Database yang telah dibuat harus diaktifkan terlebih dahulu sebelum digunakan. Sintaks umum untuk mengaktifkan database adalah sebagai berikut.

Setelah mengaktifkan database yang akan digunakan, langkah selanjutnya adalah membuat tabel yang akan disimpan di dalam database.

USE [nama_database];

```
MariaDB [(none)]> use data01;  
Database changed  
MariaDB [data01]>
```

Note: Tidak ada perintah untuk melakukan unselect, atau tidak memilih database, jika anda ingin berganti database lain tinggal gunakan use nama database yang ingin digunakan. Apabila ingin kembali kedalam keadaan tidak memilih satupun database, buat (create) database baru kemudian gunakan (use) database baru kemudian hapus (drop) database baru.

2.5. Membuat Tabel

Bentuk umum SQL untuk membuat sebuah tabel adalah sebagai berikut.

```
CREATE TABLE [nama_tabel] (  
  Nama_field_1 tippedata_1 (lebardata_1),  
  Nama_field_2 tippedata_2 (lebardata_2),  
  ....  
  Nama_field_n tippedata_n (lebardata_n),  
  PRIMARY KEY (field_key)  
);
```

Aturan penamaan tabel dan kolom (field) sama dengan aturan penamaan database.

Contoh tabel **mahasiswa**:

No	Nama kolom (Field)	Tipe Data	Lebar Data
1	nim*	int	5
2	mhs_nama	varchar	200

```
CREATE TABLE mahasiswa  
( nim int(5)NOT NULL,  
  mhs_nama VARCHAR(200) ,  
  CONSTRAINT mahasiswa_pk PRIMARY KEY (nim)  
);
```

```
MariaDB [data01]> CREATE TABLE mahasiswa  
-> ( nim int(5)NOT NULL,  
-> mhs_nama VARCHAR(200) ,  
-> CONSTRAINT mahasiswa_pk PRIMARY KEY (nim)  
-> );  
Query OK, 0 rows affected (0.107 sec)
```

Untuk melihat tabel sudah ada atau belum ketikkan perintah: **SHOW TABLES;**

```
MariaDB [data01]> show tables;  
+-----+  
| Tables_in_data01 |  
+-----+  
| mahasiswa        |  
+-----+  
1 row in set (0.001 sec)
```

untuk melihat struktur tabel gunakan perintah: **DESC [nama_tabel];**

jadi apabila sesuai dengan contoh diatas menjadi: **DESC mahasiswa;**

```
MariaDB [data01]> desc mahasiswa;
```

Field	Type	Null	Key	Default	Extra
nim	int(5)	NO	PRI	NULL	
mhs_nama	varchar(200)	YES		NULL	

```
2 rows in set (0.016 sec)
```

2.6. Mengubah Struktur Tabel dengan Alter

ALTER TABLE digunakan untuk melakukan perubahan pada tabel, antara lain mengubah nama tabel, struktur tabel, menambah field, dll. Berikut beberapa perintah yang dapat digunakan untuk mengubah tabel dengan menggunakan perintah ALTER.

a. Tambah field_baru

```
ALTER TABLE [nama_tabel] ADD [nama_kolom] [tipe_data] ([lebar_data]);
```

b. Menambah primary key (kolom kunci)

```
ALTER TABLE [nama_tabel] ADD PRIMARY KEY([nama_field]);
```

c. Mengubah field

```
ALTER TABLE [nama_tabel] CHANGE [nama_field_lama] [nama_field_baru]
[tipe_data] ([lebar_data]);
```

d. Menghapus nama field

```
ALTER TABLE [nama_tabel] DROP [nama_field];
```

e. RENAME nama tabel

```
ALTER TABLE [nama_tabel] RENAME TO [nama_tabel_baru];
```

2.7. Menghapus Tabel

Untuk menghapus sebuah tabel dapat menggunakan perintah SQL sebagai berikut.

```
DROP TABLE [nama_tabel];
```

NOTE: Perintah drop harus digunakan dengan sangat hati-hati karena terkait dengan ketersediaan data yang mungkin bersifat sangat penting.

Contoh:

a. Tambahkan *auto increment* untuk *field* nim agar penomoran nim menjadi otomatis terurut.

```
MariaDB [data01]> alter table mahasiswa change nim nim int(5) auto_increment;
Query OK, 0 rows affected (0.278 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [data01]> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| nim        | int(5)        | NO   | PRI | NULL    | auto_increment |
| mhs_nama   | varchar(200)  | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.022 sec)
```

b. Tambahkan field tabel mahasiswa: email type varchar (30)

```
MariaDB [data01]> alter table mahasiswa add email varchar(30);
Query OK, 0 rows affected (0.026 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [data01]> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| nim        | int(5)        | NO   | PRI | NULL    | auto_increment |
| mhs_nama   | varchar(200)  | YES  |     | NULL    |                |
| email      | varchar(30)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.050 sec)
```

3. DML Menggunakan *Command Prompt*

3.1. Menambah Record dengan *INSERT*

Bentuk umum perintah SQL untuk input data ke dalam sebuah tabel adalah sebagai berikut.

INSERT INTO [nama_tabel] **VALUES** ('nilai1', 'nilai2', ..., 'nilai_n');

Atau dapat menyebutkan nama *field*-nya seperti pada perintah berikut.

INSERT INTO [nama_tabel]('field1','field2') **VALUES** ('nilai1','nilai2');

3.2. Mengubah Record dengan *UPDATE*

Perintah update cukup penting dalam pengolahan database, karena biasanya banyak data yang perlu diperbaiki. Proses update adalah mengubah data lama dengan *record* yang baru. Perubahan dengan perintah update bersifat permanen. Bentuk umum perintah update adalah sebagai berikut.

UPDATE [nama_tabel] **SET** field1='recordbaru' **WHERE** kondisi;

3.3. Menghapus Record dengan DELETE

Proses *delete* adalah proses penghapusan *record* di dalam sebuah tabel. Perintah *delete* yang dilakukan bersifat permanen, sehingga perintah ini perlu dijalankan dengan sangat hati-hati. Perintah SQL untuk delete data adalah sebagai berikut.

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

3.4. Menampilkan Record dengan SELECT

Perintah *select* berfungsi untuk menampilkan data pada sebuah tabel. Dengan menggunakan perintah *select* kita dapat mengatur tampilan atau keluaran data sesuai dengan yang diinginkan. Bentuk dasar perintah SELECT adalah sebagai berikut.

```
SELECT [field | *] FROM [nama_tabel] WHERE kondisi;
```

Contoh:

3a.. Inputkan data mahasiswa dengan perintah insert seperti berikut berikut

```
MariaDB [data01]> insert into mahasiswa (mhs_nama,email) values ("paijo","paijo@email.com");
Query OK, 1 row affected (0.007 sec)
```

```
MariaDB [data01]> insert into mahasiswa (mhs_nama) values ("Dodi");
Query OK, 1 row affected (0.018 sec)
```

3b. Tampilkan data dengan perintah select seperti berikut

Menampilkan semua data dengan mengguna tanda asterik (*)

```
MariaDB [data01]> select * from mahasiswa;
+-----+-----+-----+
| nim | mhs_nama | email          |
+-----+-----+-----+
| 1 | paijo | paijo@email.com |
| 2 | Dodi | NULL           |
+-----+-----+-----+
2 rows in set (0.000 sec)
```

Menampilkan sebagian field dengan menyebutkan field yang ingin ditampilkan

```
MariaDB [data01]> select nim, mhs_nama from mahasiswa;
+-----+-----+
| nim | mhs_nama |
+-----+-----+
| 1 | paijo |
| 2 | Dodi |
+-----+-----+
2 rows in set (0.000 sec)
```


3c. Menampilkan dengan kondisi

Menampilkan nama yang emailnya kosong

```
MariaDB [data01]> select mhs_nama from mahasiswa where email is null;
+-----+
| mhs_nama |
+-----+
| Dodi      |
+-----+
1 row in set (0.005 sec)
```

Menampilkan data mahasiswa yang bernama "paijo"

```
MariaDB [data01]> select * from mahasiswa where mhs_nama like "paijo";
+-----+-----+-----+
| nim | mhs_nama | email          |
+-----+-----+-----+
| 1   | paijo    | paijo@email.com |
+-----+-----+-----+
1 row in set (0.003 sec)
```

Menampilkan data mahasiswa yang namanya mengandung huruf tertentu.

```
MariaDB [data01]> select* from mahasiswa where mhs_nama like"p%";
+-----+-----+-----+
| nim | mhs_nama | email          |
+-----+-----+-----+
| 1   | paijo    | paijo@email.com |
+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [data01]> select* from mahasiswa where mhs_nama like"%i";
+-----+-----+-----+
| nim | mhs_nama | email |
+-----+-----+-----+
| 2   | Dodi      | NULL  |
+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [data01]> select* from mahasiswa where mhs_nama like"%a%";
+-----+-----+-----+
| nim | mhs_nama | email          |
+-----+-----+-----+
| 1   | paijo    | paijo@email.com |
+-----+-----+-----+
1 row in set (0.001 sec)
```

3d. Ubah data di dalam tabel mahasiswa dengan menggunakan perintah update

Update mahasiswa set email="dodi@amikomsolo.ac.id" **where** mhs_nama="dodi";

```
MariaDB [data01]> Update mahasiswa set email="dodi@amikomsolo.ac.id" where mhs_nama="dodi";
Query OK, 1 row affected (0.023 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [data01]> select * from mahasiswa;
+-----+-----+-----+
| nim | mhs_nama | email |
+-----+-----+-----+
| 1 | paijo | paijo@email.com |
| 2 | Dodi | dodi@amikomsolo.ac.id |
+-----+-----+-----+
2 rows in set (0.000 sec)
```

Note: Jika tidak menggunakan kondisi (where) maka semua data di dalam table akan berubah.

3e. Hapus data dengan perintah delete

```
MariaDB [data01]> delete from mahasiswa where mhs_nama = "dodi";
Query OK, 1 row affected (0.026 sec)

MariaDB [data01]> select * from mahasiswa;
+-----+-----+-----+
| nim | mhs_nama | email |
+-----+-----+-----+
| 1 | paijo | paijo@email.com |
+-----+-----+-----+
1 row in set (0.000 sec)
```

```
MariaDB [data01]> delete from mahasiswa where mhs_nama like "P%";
Query OK, 1 row affected (0.005 sec)

MariaDB [data01]> select * from mahasiswa;
Empty set (0.000 sec)
```

4. Backup dan Restore

Database yang telah dibuat dapat di *backup* atau dapat juga melakukan *restore* dari file *database* yang lain.

4.1. Backup

Proses *backup database* dapat dilakukan dengan menuliskan perintah sebagai berikut.

```
C:\xampp\mysql\bin>mysqldump -u root -p pw10001 > D:\backup\pw1.sql
```

Dengan melakukan proses backup di atas, database pw10001 akan di-backup dan disimpan di direktori D:/ pada folder backup.

Jika ingin mem-backup struktur tabelnya saja, dapat menggunakan perintah sebagai berikut.

```
C:\xampp\mysql\bin>mysqldump -u root -p --no-data pw10001 > D:\backup\pw1.sql
```

Atau jika ingin mem-backup datanya saja tanpa mengikutsertakan strukturnya, dapat menggunakan perintah sebagai berikut.

```
C:\xampp\mysql\bin>mysqldump -u root -p --no-create-info pw10001 > D:\backup\pw1.sql
```

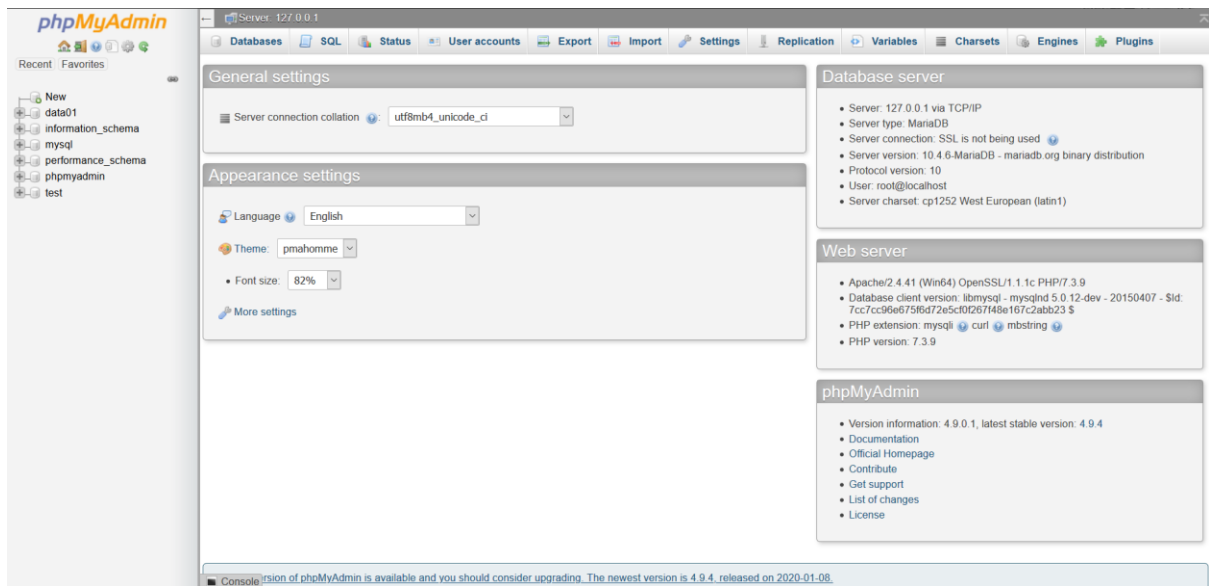
4.2. Restore

Untuk me-restore sebuah *database*, terlebih dahulu siapkan sebuah database, misal **pwl_baru**. Kemudian tuliskan perintah seperti berikut ini.

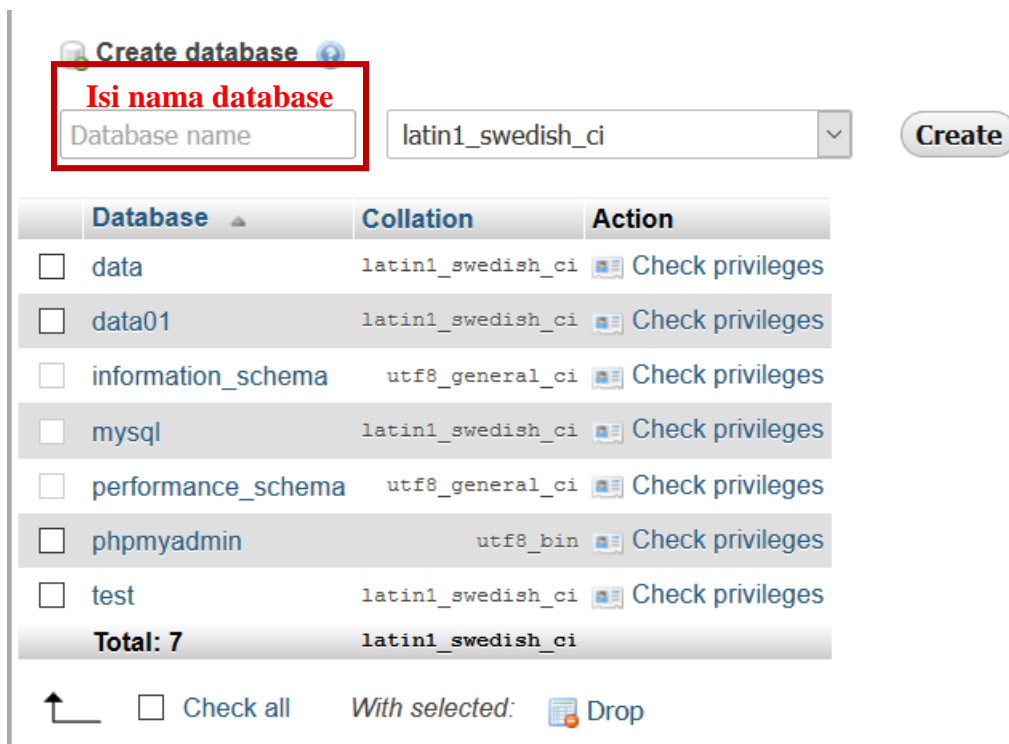
```
C:\xampp\mysql\bin>mysqldump -u root -p pwl_baru < D:\backup\pwl.sql
```

3. SQL DENGAN “PHPMYADMIN”

Masuk phpmyadmin dengan cara ketikan di browser apabila port apache 8080 menjadi <http://localhost:8080/phpmyadmin/> (pastikan apache dan mysql berjalan di xampp).



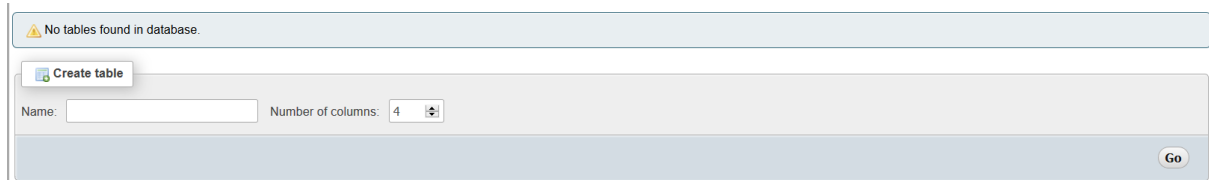
Klik new di kiri atas untuk membuat database baru.



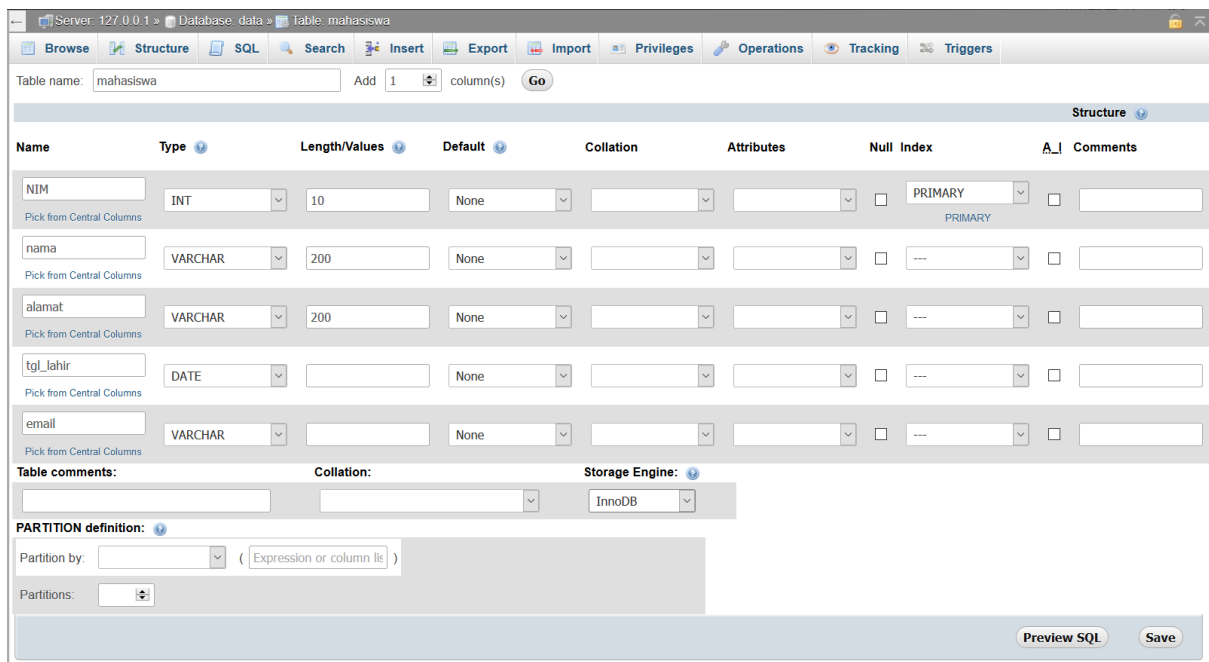
Collation adalah: Sekumpulan rule atau aturan yang digunakan oleh database untuk untuk membandingkan karakter yang ada pada sebuah character set. Sedangkan character set sendiri dapat diartikan sekumpulan character dengan jenis tertentu.

Setelah database terbuat pilih list database di bagian kiri sesuai nama yang telah dibuat tadi.

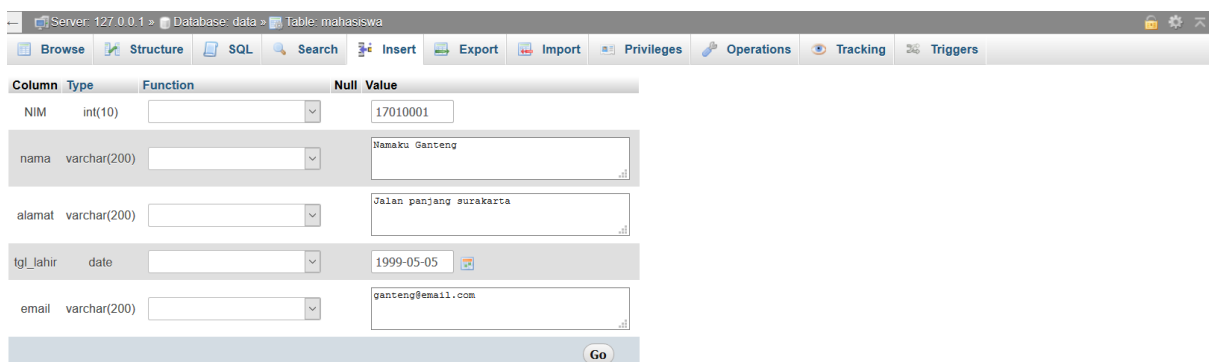
Karena tabel kosong maka otomatis anda akan masuk window untuk membuat tabel.



Setelah anda membuat tabel maka anda akan mengisi nama atribut-atributnya.



Untuk menginputkan data masuk dalam menu insert.



Silahkan eksplorasi lebih lanjut tentang phpmyadmin.

Latihan: Buatlah 2 tabel kemudian relasikan lah tabel tersebut menggunakan phpmyadmin. (Anda dapat melihat visualisasi relasi tabel dengan masuk ke database, kemdian menu designer)