
PHP dan MySQL

Hands-on Labs 6

I Made Arsa Suyadnya
Program Studi Teknik Elektro, FT - UNUD

A. Pengenalan Basis Data

Basis data terdiri atas 2 kata, yaitu Basis dan Data. **Basis** kurang lebih dapat diartikan sebagai markas atau gudang, sedangkan **Data** adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Sebagai satu kesatuan istilah, basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

- Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan **cepat** dan **mudah**.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan **tanpa pengulangan (redundansi)** yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- Kumpulan tabel yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

B. Sistem Basis Data

Basis data hanyalah sebuah objek yang pasif. Ia ada karena ada pembuatnya. Ia tidak akan pernah berguna jika tidak ada pengelola dan penggerakannya. Yang menjadi pengelola atau penggerakannya secara langsung adalah program/aplikasi (*software*). Gabungan keduanya (basis data dan pengelolanya) menghasilkan sebuah sistem. Karena itu, secara umum sebuah sistem basis data merupakan sistem yang terdiri atas kumpulan tabel data yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program (yang biasa disebut *DBMS/Database Management System*) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi tabel-tabel data tersebut.

Adapun secara lengkap, sebuah sistem basis data terdiri atas komponen-komponen utama sebagai berikut.

1. Perangkat Keras (*Hardware*)
2. Sistem Operasi (*Operating System*)
3. Basis Data (*Database*)
4. Sistem (Aplikasi/Perangkat Lunak) Pengelola Basis Data (DBMS)
5. Pemakai (*User*)
6. Aplikasi (Perangkat Lunak) lain (yang bersifat opsional, artinya tidak harus ada)

C. DBMS (Database Management System)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus. Perangkat lunak inilah (disebut **DBMS/Database Management System**) yang akan menentukan bagaimana data diorganisasikan, disimpan, diubah dan diambil kembali. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan/konsistensi data, dan sebagainya.

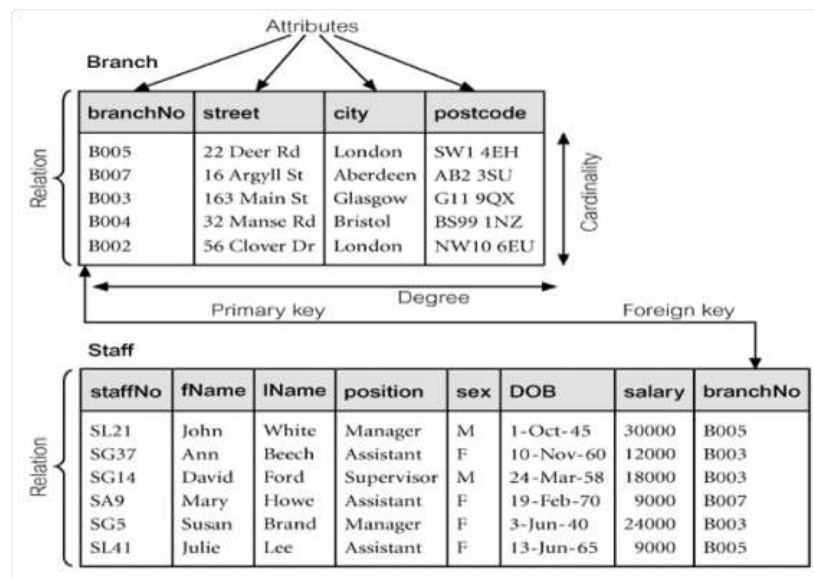
Perangkat lunak yang termasuk DBMS diantaranya Microsoft SQL Server, Oracle Database, IBM DB2, MySQL, dan PostgreSQL. Pembahasan dalam materi ini menggunakan DBMS MySQL. MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis di bawah lisensi GPL (General Public License). Dengan kehandalan, kecepatan dan kemudahan

penggunaannya, MySQL menjadi pilihan utama bagi banyak pengembang perangkat lunak dan aplikasi baik di *platform* web maupun desktop. MySQL pertama kali dibuat dan dikembangkan di Swedia, yaitu oleh David Axmark, Allan Larsson dan Michael "Monty" Widenius. Mereka mengembangkan MySQL sejak tahun 1980-an. Saat ini versi MySQL yang sudah stabil mencapai versi 5.x. Informasi lengkap mengenai MySQL dapat diperoleh melalui situs resminya: <http://www.mysql.com/>.

D. Basis Data Relasional

Salah satu jenis basis data yang populer saat ini adalah basis data relasional (*relational database*). Basis data relasional adalah jenis basis data yang menggunakan model relasional. Pada model relasional, data disusun dalam bentuk sejumlah relasi atau tabel. Setiap tabel tersusun atas sejumlah baris dan kolom. Baris mewakili data kesatuan data dan kolom menyatakan bagian-bagian yang menyusun sebuah baris.

Pada model relasional, satu tabel bisa berhubungan dengan tabel lain. Hubungan dibentuk melalui mekanisme kunci primer (*primary key*) dan kunci asing/tamu (*foreign key*). Kunci primer adalah suatu kolom (atau gabungan beberapa kolom) yang dapat digunakan untuk membedakan antara satu baris dengan baris yang lain. Berikut ini, pada Gambar 1 merupakan bentuk model relasional, yakni hubungan antara tabel Branch dengan tabel Staff.



Gambar 1. Hubungan antara tabel Branch dengan tabel Staff

E. SQL

DBMS merupakan perantara bagi pemakai dengan basis data dalam *disk*. Cara berinteraksi antara pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa itu dapat kita sebut sebagai bahasa basis data yang terdiri atas sejumlah perintah (*statement*) yang diformulasikan dan dapat diberikan user dan dikenali/diproses oleh DBMS untuk melakukan suatu aksi tertentu.

SQL atau *Structured Query Language* merupakan bahasa standar yang digunakan untuk mengakses basis data relasional dan bersifat non-prosedural. Pengaksesan data dapat dilakukan dengan bentuk perintah yang sederhana. Perintah SQL ini sering juga disingkat dengan sebutan '**query**'.

Adapun perintah atau instruksi SQL dapat dikelompokkan berdasarkan jenis dan fungsinya. Terdapat 3 jenis perintah dasar SQL, yakni DDL (*Data Definition Language*), DML (*Data Manipulation Language*) dan DCL (*Data Control Language*).

1. DDL (*Data Definition Language*)

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, view, user, dan sebagainya. Secara umum, DDL yang digunakan adalah CREATE untuk membuat objek baru, USE untuk menggunakan objek, ALTER untuk mengubah objek yang sudah ada, dan DROP untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah basis data.

▪ CREATE

CREATE digunakan untuk membuat basis data maupun objek-objek basis data. Perintah SQL yang umum digunakan adalah:

```
CREATE DATABASE nama_basis_data
```

Perintah CREATE DATABASE digunakan untuk membuat sebuah basis data baru.

CREATE TABLE membuat tabel baru pada basis data yang sedang aktif. Secara umum, perintah ini memiliki bentuk:

```
CREATE TABLE [nama_tabel]
(
    nama_field1 tipe_data [,
    nama_field2 tipe_data,
    ...]
    [CONSTRAINT nama_field constraints]
)
```

dengan:

- *nama_field* adalah nama kolom (*field*) yang akan dibuat. Beberapa sistem manajemen basis data mengizinkan penggunaan spasi dan karakter non huruf 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:

```
CREATE TABLE tbl_pelanggan (
    id_pelanggan VARCHAR(5) NOT NULL,
    nama_pelanggan VARCHAR(25) NOT NULL,
    alamat VARCHAR(50) NOT NULL,
```

```
no_hp VARCHAR(15) NOT NULL,  
PRIMARY KEY (id_pelanggan))
```

- **ALTER**

Perintah alter digunakan untuk merubah struktur tabel yang telah dibuat sebelumnya dengan perintah CREATE. Berikut perintah SQL yang digunakan, diantaranya:

Merubah tipe data kolom:

```
ALTER TABLE nama_tabel MODIFY nama_kolom tipe_data_baru;
```

Merubah nama kolom tabel:

```
ALTER TABLE nama_tabel CHANGE nama_kolom nama_kolom_baru tipe_data;
```

Merubah nama tabel:

```
ALTER TABLE nama_tabel_lama RENAME TO nama_tabel_baru;
```

Menambahkan kolom baru pada tabel:

```
ALTER TABLE nama_tabel ADD nama_kolom_baru tipe_data;
```

Menghapus kolom tabel:

```
ALTER TABLE nama_tabel DROP nama_kolom;
```

- **DROP**

Perintah drop digunakan untuk menghapus basis data maupun objek-objek basis data. Berikut perintah SQL yang digunakan, diantaranya:

Menghapus database:

```
DROP DATABASE nama_database;
```

Menghapus tabel:

```
DROP TABLE nama_tabel;
```

2. DML (*Data Manipulation Language*)

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah untuk menampilkan data (SELECT), menambahkan data baru (INSERT), mengubah data yang sudah ada (UPDATE), dan menghapus data (DELETE).

- **SELECT**

SELECT adalah perintah yang paling sering digunakan pada SQL, sehingga terkadang istilah query dirujuk pada perintah SELECT. SELECT digunakan untuk menampilkan data dari satu atau lebih tabel, biasanya dalam sebuah basis data yang sama. Secara umum, perintah SELECT memiliki bentuk lengkap seperti berikut ini.

```
SELECT [nama_tabel|alias.]nama_field1 [AS alias1] [, nama_field2, ...]  
FROM nama_tabel1 [AS alias1] [INNER|LEFT|RIGHT JOIN tabel2 ON kondisi_penghubung]  
[, nama_tabel3 [AS alias3], ...]  
[WHERE kondisi]  
[ORDER BY nama_field1 [ASC|DESC][, nama_field2 [ASC|DESC], ...]]  
[GROUP BY nama_field1[, nama_field2, ...]]  
[HAVING kondisi_agregat]
```

dengan:

- *kondisi* adalah syarat yang harus dipenuhi suatu data agar ditampilkan.
- *kondisi_agregat* adalah syarat khusus untuk fungsi agregat, contoh fungsi agregat: SUM untuk menghitung total nominal data, COUNT untuk menghitung jumlah kemunculan data, AVG untuk menghitung rata-rata sekelompok data, MAX dan MIN untuk mendapatkan nilai maksimum/minimum dari sekelompok data

Kondisi dapat dihubungkan dengan operator logika, misalnya AND, OR, dan sebagainya.

Contoh:

```
SELECT * FROM tbl_pelanggan
WHERE id_pelanggan = 'P0005'
```

Perintah SQL tersebut berarti menampilkan seluruh data (tanda asterik * untuk menampilkan seluruh kolom tabel) dari tabel pelanggan yang memiliki ID pelanggan: P0005.

■ INSERT

Untuk menyimpan data dalam tabel dipergunakan sintaks:

```
INSERT INTO [nama_table] ([daftar_field]) VALUES ([daftar_nilai])
```

Contoh:

```
INSERT INTO test (nama, alamat, password) VALUES ('test', 'alamat', 'pass');
```

■ UPDATE

Untuk mengubah data menggunakan sintaks:

```
UPDATE [nama_table] SET [nama_kolom]=[nilai] WHERE [kondisi]
```

Contoh:

```
UPDATE user SET password="123456" WHERE username="abc";
```

■ DELETE

Untuk menghapus data dipergunakan sintaks:

```
DELETE FROM [nama_table] WHERE [kondisi]
```

Contoh:

```
DELETE FROM TEST WHERE NAMA='test';
```

3. DCL (*Data Control Language*)

DCL merupakan jenis instruksi SQL yang berkaitan dengan manajemen hak akses dan pengguna (user) yang dapat mengakses database maupun tabel. Contoh perintah SQL yang termasuk dalam DCL, diantaranya: GRANT dan REVOKE

F. Implementasi Database Menggunakan DBMS MySQL

Implementasi database, seperti menciptakan database, membuat tabel, memodifikasi tabel, menghapus tabel dan lainnya dapat dilakukan dengan menggunakan perintah SQL yang termasuk dalam kelompok DDL (*Data Definition Language*). Selain menggunakan perintah SQL, implementasi database dapat dilakukan menggunakan suatu *tool* yang dirancang untuk memudahkan proses tersebut. Apabila implementasi database menggunakan DBMS MySQL, salah satu *tool* seperti phpmyadmin yang berbasis

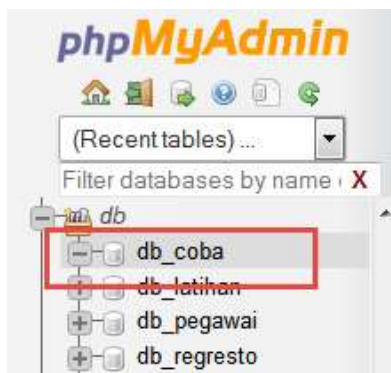
web juga cukup membantu dalam mengimplementasikan database tanpa menggunakan SQL. Namun, dalam *tool* phpmyadmin juga tetap menyediakan fasilitas kepada pengembang aplikasi untuk mengeksekusi query secara manual. Pengetahuan tentang SQL mutlak diperlukan bagi pengembang aplikasi database, mengingat aplikasi yang menggunakan database tetap berinteraksi dengan database melalui SQL.

Adapun untuk *tool* phpmyadmin, apabila telah meng-install menggunakan paket XAMPP pada komputer lokal, maka dapat diakses melalui URL: <http://localhost/phpmyadmin>. Untuk pembuatan database dan tabel menggunakan phpmyadmin diawali dengan proses *crate database* seperti terlihat pada Gambar 2 berikut ini (dengan asumsi pengguna telah berhasil login pada aplikasi phpmyadmin).



Gambar 2. Pembuatan database dengan phpmyadmin

Setelah database berhasil dibuat, maka database **db_coba** akan muncul pada daftar database yang ada di sebelah kiri, seperti terlihat pada Gambar 3.



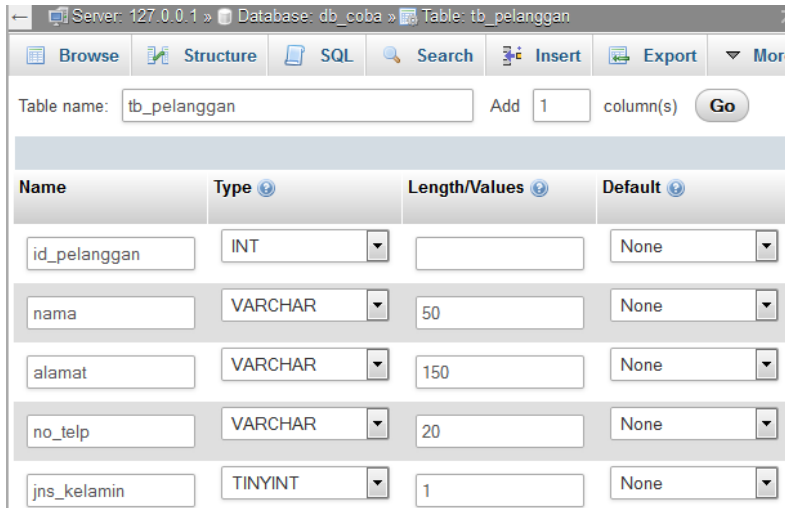
Gambar 3. Database db_coba berhasil dibuat

Selanjutnya, dengan mengklik database **db_coba** pada daftar tersebut, maka akan tampil fasilitas untuk menambahkan tabel, seperti tampak pada Gambar 4.



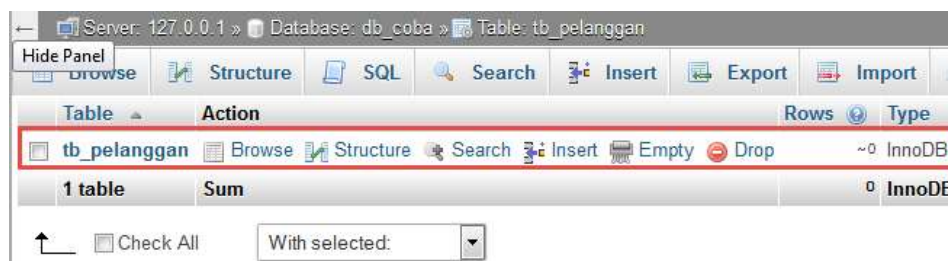
Gambar 4. Fasilitas untuk pembuatan tabel pada database

Misalkan nama tabel yang akan ditambahkan adalah **tb_pelanggan** dengan jumlah kolom sebanyak 5, setelah menekan tombol **Go** maka pengguna akan dibawa menuju ke fasilitas untuk mengisi struktur tabel tersebut, seperti terlihat pada Gambar 5 berikut ini.



Gambar Error! No text of specified style in document.. Pengisian struktur tabel tb_pelanggan

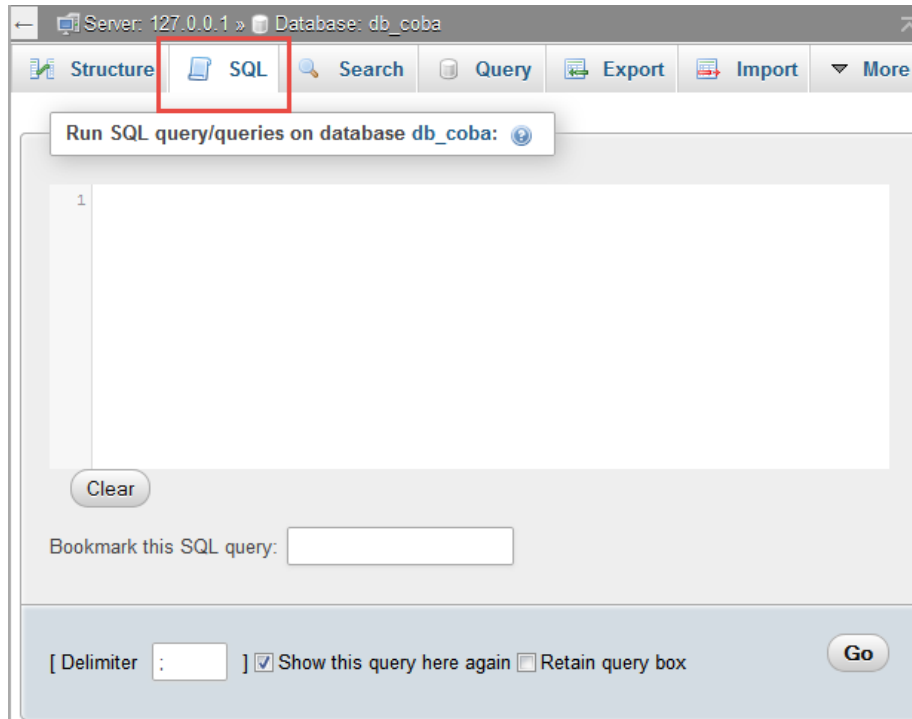
Setelah melengkapi struktur tabel, untuk menyimpan/menambahkan tabel tersebut ke database dilakukan dengan menekan tombol **Save** yang ada pada bagian bawah. Apabila tabel berhasil disimpan, maka tabel tersebut akan tampil pada list tabel, seperti terlihat pada Gambar 6.



Gambar 5. Tabel tb_pelanggan berhasil dibuat

Untuk melakukan pembuatan/menambahkan tabel baru pada database **db_coba** dapat dilakukan dengan langkah-langkah yang sama seperti sebelumnya.

Selain pembuatan tabel tanpa menggunakan SQL, phpmyadmin juga menyediakan fasilitas untuk melakukan eksekusi query atau perintah SQL secara manual. Hal ini dapat dilakukan dengan mengklik tab SQL pada halaman phpmyadmin, seperti terlihat pada Gambar 7.



Gambar 6. Fasilitas eksekusi query secara manual pada phpmyadmin

G. Insert Data dengan PHP – MySQL

Setelah mengenal beberapa bentuk perintah SQL untuk memanipulasi atau mengolah data di MySQL, pada bagian ini akan diperkenalkan bagaimana menerapkan konsep tersebut yang terintegrasi dengan script PHP.

Pada bagian ini, akan diawali dengan konsep inserting data ke MySQL melalui PHP. Kasus yang akan digunakan adalah pengelolaan data barang. Kasus ini juga akan digunakan pada konsep menampilkan data, update data dan hapus data.

▪ Pembuatan tabel pada database

Sebelum melakukan proses insert data, maka perlu dibuat terlebih dahulu tabel untuk penyimpanan data di database. Adapun struktur tabel yang dibutuhkan adalah sebagai berikut.

id_barang	int(11)	auto increment	primary key
nama_barang	varchar(50)		
stok	smallint(6)		
harga_satuan	int(11)		

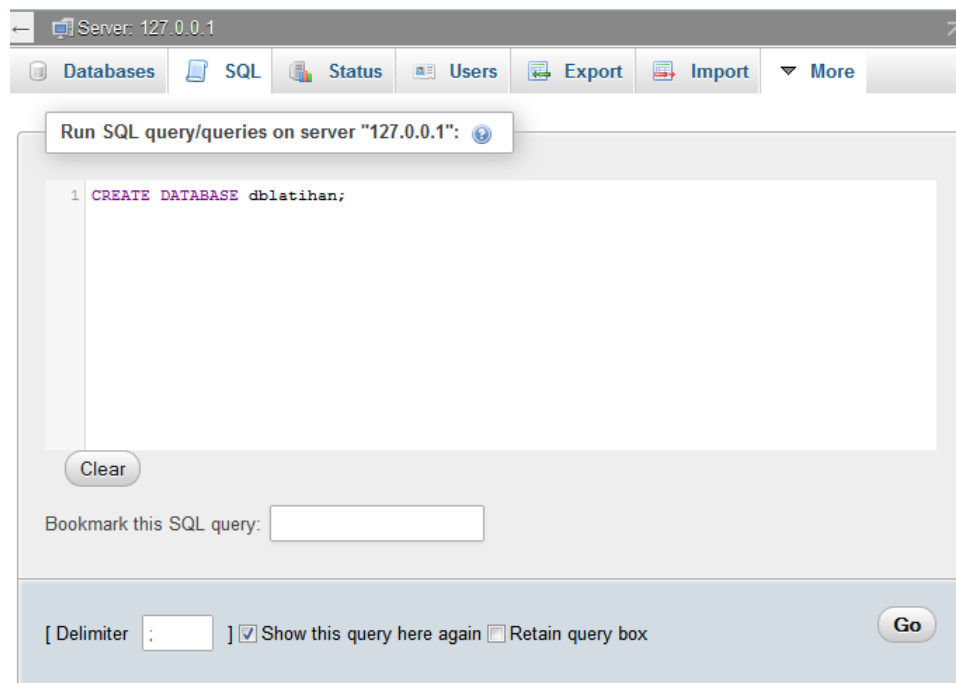
Selanjutnya struktur tabel tersebut kita konversi ke dalam bentuk perintah SQL seperti berikut ini.

```
CREATE TABLE `tb_barang` (
  `id_barang` INT(11) NOT NULL AUTO_INCREMENT,
  `nama_barang` VARCHAR(50) NOT NULL,
  `stok` SMALLINT(6) NOT NULL DEFAULT '0',
  `harga_satuan` INT(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_barang`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB;
```

Adapun untuk pembuatan database dan tabel, dapat menggunakan *tool* phpmyadmin. Database yang dibuat adalah dengan nama **dblatihan**. Dalam hal ini akan digunakan fasilitas eksekusi query secara manual yang disediakan oleh phpmyadmin. Perintah SQL untuk membuat database adalah sebagai berikut.

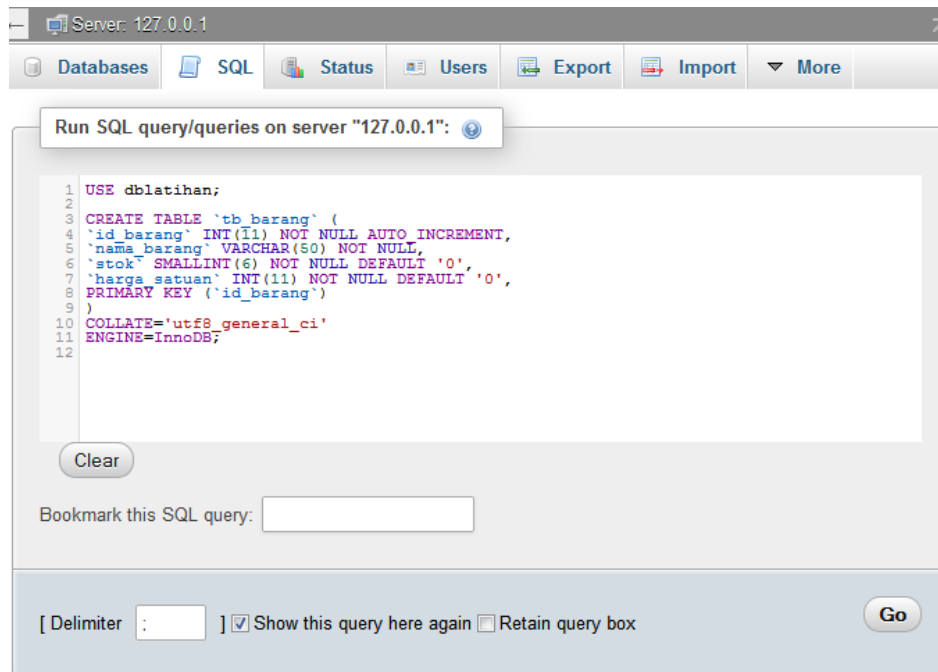
```
CREATE DATABASE dblatihan;
```

Input dan eksekusi perintah SQL ini, dapat dilihat seperti pada Gambar 8 (dengan asumsi telah berhasil login/masuk ke aplikasi phpmyadmin). Eksekusi dilakukan dengan menekan tombol Go.



Gambar 7. Perintah SQL untuk membuat database dblatihan

Setelah itu dengan cara yang sama, kita inputkan perintah SQL untuk membuat tabel barang seperti terlihat pada Gambar 9.



Gambar 8. Membuat tabel `tb_barang` pada database `dlatihan`

Pada perintah SQL yang di-inputkan terdapat perintah `USE`. Perintah `USE` ini digunakan untuk menset penggunaan objek database **`dlatihan`** yang telah dibuat sebelumnya.

▪ Koneksi PHP ke MySQL

Setelah database dan tabel berhasil dibuat, selanjutnya akan dibahas mengenai koneksi dari PHP ke MySQL. Untuk dapat melakukan koneksi dari PHP ke MySQL digunakan ekstensi `MySQLi`. Huruf `i` pada bagian akhir artinya *Improved* (yang disempurnakan dari versi ekstensi sebelumnya).

Perintah PHP untuk koneksi ke MySQL adalah:

```
mysqli_connect(host,username,password,dbname);
```

Keterangan:

- `host` : nama komputer dimana DBMS MySQL berada, apabila berada di komputer lokal maka nama hostnya: `'localhost'`, atau bisa diisi dengan alamat IP server database MySQL.
- `username` : nama user yang memiliki hak akses ke DBMS MySQL
- `password` : password untuk nama user yang memiliki hak akses ke DBMS MySQL
- `dbname` : nama database yang digunakan

Contoh dalam script PHP:

```

<?php
$mysql_hostname = "localhost";
$mysql_user = "root";
$mysql_password = "root";
$mysql_database = "dlatihan";

```

```
$conn = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or
die("Koneksi ke database gagal!");
?>
```

Sebaiknya script di atas disimpan dalam file script tersendiri, misalkan **configdb.php**. Selanjutnya script tersebut dapat di-includekan ke dalam script lain yang di dalamnya terdapat query-query SQL.

▪ Menjalankan query SQL di PHP

Setelah melakukan koneksi dari PHP ke MySQL, selanjutnya kita bisa menjalankan query melalui script PHP. Untuk menjalankan query SQL di PHP, dapat menggunakan perintah:

```
mysqli_query(connection,query);
```

Berikut ini contoh script PHP untuk menjalankan query SQL untuk insert data ke tabel MySQL.

```
<?php
include "configdb.php";
$query = "INSERT INTO namaTabel(field1, field2) VALUES(value1, value2) ";
mysqli_query($conn,$query);
?>
```

▪ Insert data barang

Setelah mengetahui teknik koneksi ke database, selanjutnya adalah mengimplementasikan konsep inserting data ke MySQL melalui script PHP yang diintegrasikan dengan form input. Berikut ini contoh script PHP beserta form input dalam HTML untuk insert data barang.

insert.php

```
<?php
include "configdb.php";
if (isset($_POST['act'])) {
    $namabarang = $_POST['nmbrg'];
    $stokbarang = $_POST['stkbrg'];
    $hargasatuan = $_POST['hrgrg'];

    $sql = "INSERT INTO tb_barang(nama_barang,stok,harga_satuan)
        VALUES('$namabarang',$stokbarang,$hargasatuan)";
    $ex = mysqli_query($conn,$sql) or die("Insert data gagal!");
    $result = mysqli_affected_rows($conn);
    //cek jika eksekusi query berhasil dilakukan
    if ($result>0) {
        echo "<script>alert('Insert/Tambah data berhasil..');
        location.href='tampil.php'; </script>";
    } else {
        echo "<script>alert('Insert/Tambah data gagal..');
        </script>";
    }
}
?>
<h2>Insert/Tambah Data Barang</h2>
<form method="post">
<table border="1">
<tr><td>Nama Barang</td><td><input type="text" name="nmbrg" /></td></tr>
<tr><td>Stok</td><td><input type="text" name="stkbrg" /></td></tr>
<tr><td>Harga Satuan</td><td><input type="text" name="hrgrg" /></td></tr>
```

```

<tr><td colspan="2"><input type="submit" name="act" value="Save" onClick="return
confirm('Apakah benar akan menambahkan data ini?')" />
<input type="button" value="Kembali" onclick="self.history.back()" /></td></tr>
</table>
</form>

```

Hasil form input data barang dapat dilihat pada Gambar 10 berikut ini.



Gambar 9. Form insert/tambah data barang

Dalam script tersebut, setelah form input data barang diisi dan kemudian menekan tombol Save maka nilai input data tersebut akan dikirimkan menggunakan method POST untuk diproses pada bagian script PHP. Nilai-nilai yang diinputkan akan ditambahkan ke `tb_barang` dengan query SQL INSERT. Apabila query berhasil dieksekusi, maka akan menampilkan alert: "Insert/Tambah data berhasil" dan kemudian diarahkan menuju ke halaman `tampil.php` untuk menampilkan data barang yang berhasil ditambahkan.

H. Menampilkan Data dengan PHP – MySQL

Konsep menampilkan data dengan script PHP ini adalah menggabungkan query SELECT di SQL yang dijalankan dalam script PHP.

Konsep dasar menampilkan data dengan script PHP ini adalah

1. Membuat query SQL untuk menampilkan data (menggunakan SELECT)
2. Jalankan query tersebut dengan menggunakan `mysqli_query()` di PHP
3. Baca data hasil query tersebut dengan proses *fetching*. Gunakan perintah `mysqli_fetch_array()` di PHP. Hasil dari `mysqli_fetch_array()` ini adalah suatu nilai dalam bentuk array asosiatif dengan nama field tabel sebagai indeksnya
4. Tampilkan data dari proses *fetching* sesuai nama field yang ada dalam array

Berikut ini contoh script PHP untuk menampilkan data barang.

tampil.php

```

<!DOCTYPE html>
<html>
<head>
<title>Tampil Data Barang</title>

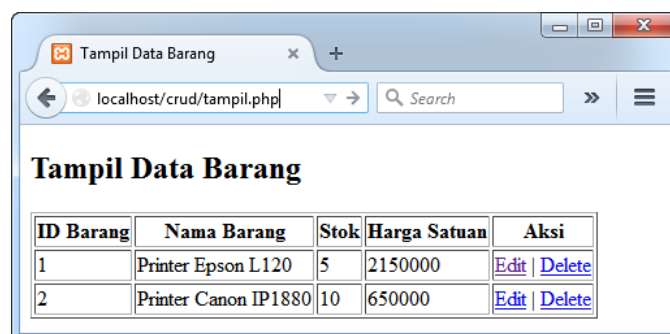
```

```

</head>
<body>
<h2>Tampil Data Barang</h2>
<table border="1">
<thead>
  <tr>
    <th>ID Barang</th>
    <th>Nama Barang</th>
    <th>Stok</th>
    <th>Harga Satuan</th>
    <th>Aksi</th>
  </tr>
</thead>
<tbody>
<?php
include "configdb.php";
$res = mysqli_query($conn,"SELECT * FROM tb_barang");
while ($row = mysqli_fetch_array($res)){
  ?>
  <tr>
    <td><?php echo $row['id_barang']; ?></td>
    <td><?php echo $row['nama_barang']; ?></td>
    <td><?php echo $row['stok']; ?></td>
    <td><?php echo $row['harga_satuan']; ?></td>
    <td>
      <a href="update.php?id=<?php echo $row['id_barang']; ?>">Edit</a> |
      <a href="action.php?id=<?php echo $row['id_barang']; ?>" onClick="return
confirm('Apakah benar akan menghapus data ini?')>Delete</a>
    </td>
  </tr>
</tbody>
<?php
}
?>
</tbody>
</table>
</body>
</html>

```

Hasil dari script tampil.php di atas dapat dilihat pada Gambar 11 berikut ini.



Gambar 10. Tampil data barang

Dalam script tersebut, proses fetching data menggunakan fungsi `mysqli_fetch_array()` dilakukan per baris data dari hasil query. Sehingga untuk mem-fetching semua data hasil query, digunakan looping `WHILE`. Selama masih ada baris data hasil query yang bisa di-fetch, maka looping ini akan terus berjalan dan baru akan berhenti bila sudah tidak ada baris lagi yang bisa di-fetch (akhir dari baris data). Setiap kali data di-fetch, selanjutnya langsung ditampilkan dengan `echo`. Selain itu, pada setiap baris akan ditambahkan link (kolom aksi) untuk melakukan edit dan delete data.

I. Update Data dengan PHP – MySQL

Proses update suatu data dengan menggunakan script PHP dan MySQL biasanya memiliki mekanisme sebagai berikut:

1. Tampilkan list data terlebih dahulu
2. Membuat sebuah link pada setiap baris data yang menuju ke proses update. Link tersebut memuat sebuah parameter yang di dalamnya terdapat value yang nantinya digunakan sebagai acuan dalam proses update data.
3. Selanjutnya apabila link tersebut diklik akan muncul form berisi data yang akan di-edit.

Dari contoh script PHP sebelumnya: **tampil.php**, yang digunakan untuk menampilkan data barang telah ditambahkan link pada setiap baris data menuju ke proses update.

```
<a href="update.php?id=<?php echo $row['id_barang']; ?>">Edit</a>
```

Link yang ditambahkan tersebut menuju ke script **update.php** dengan parameter acuan untuk update data adalah id barang.

Berikut ini merupakan contoh script PHP untuk menampilkan form edit data dan sekaligus untuk memproses update data tersebut.

update.php

```
<?php
include "configdb.php";
if(isset($_GET['id'])){
    $idbrg = mysqli_real_escape_string($conn,$_GET['id']);
    if (isset($_POST['act'])) {
        $idbrg = mysqli_real_escape_string($conn,$_POST['idbrg']);
        $namabarang = $_POST['nmbrg'];
        $stokbarang = $_POST['stkbrg'];
        $hargasatuan = $_POST['hrgrg'];

        $sql = "UPDATE tb_barang SET nama_barang = '$namabarang',
            stok = $stokbarang, harga_satuan = $hargasatuan
            WHERE id_barang = '$idbrg'";

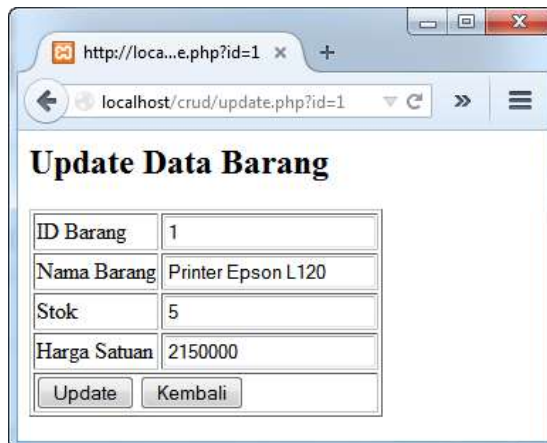
        $ex = mysqli_query($conn,$sql) or die("Update data gagal!");
        $result = mysqli_affected_rows($conn);
        //cek jika eksekusi query berhasil dilakukan
        if ($result>0) {
            echo "<script>alert('Update data berhasil..');
            location.href='tampil.php'; </script>";
```

```

    } else {
        echo "<script>alert('Update data gagal..'); </script>";
    }
}
$res = mysqli_query($conn,"SELECT * FROM tb_barang WHERE
    id_barang=".$idbrg);
$row = mysqli_fetch_array($res);
?>
<h2>Update Data Barang</h2>
<form method="post">
<table border="1">
<tr><td>ID Barang</td><td><input type="text" name="idbrg" readonly value="<?php echo
$row['id_barang']; ?>" /></td></tr>
<tr><td>Nama Barang</td><td><input type="text" name="nmbrg" value="<?php echo
$row['nama_barang']; ?>" /></td></tr>
<tr><td>Stok</td><td><input type="text" name="stkbrg" value="<?php echo $row['stok']; ?>"
/></td></tr>
<tr><td>Harga Satuan</td><td><input type="text" name="hrgrg" value="<?php echo
$row['harga_satuan']; ?>" /></td></tr>
<tr><td colspan="2"><input type="submit" name="act" value="Update" onClick="return
confirm('Apakah benar akan meng-update data ini?')"/>
<input type="button" value="Kembali" onclick="self.history.back()"/></td></tr>
</table>
</form>
<?php
}
?>

```

Dari script tersebut, dapat dilihat bahwa terdapat pengecekan terhadap parameter id barang: `isset($_GET['id'])`. Apabila `$_GET['id']` telah diset maka akan mengembalikan nilai true dan mengeksekusi query SELECT untuk mengambil data barang sesuai dengan parameter id barang. Data barang kemudian ditampilkan dalam form untuk selanjutnya dapat di-edit dan dikirimkan ke proses update data. Berikut ini, pada Gambar 12 merupakan form edit data barang yang ditampilkan setelah mengklik link edit pada kolom aksi (halaman tampil data barang).



The screenshot shows a web browser window with the address bar displaying `http://localhost/crud/update.php?id=1`. The page title is "Update Data Barang". The form contains the following data:

ID Barang	1
Nama Barang	Printer Epson L120
Stok	5
Harga Satuan	2150000

At the bottom of the form, there are two buttons: "Update" and "Kembali".

Gambar 11. Form update data barang

J. Hapus Data dengan PHP - MySQL

Konsep utama dari proses penghapusan data MySQL dengan PHP adalah menjalankan query SQL DELETE pada script PHP. Hampir sama dengan proses update data, untuk melakukan proses delete memerlukan link yang memuat sebuah parameter yang di dalamnya terdapat value yang nantinya digunakan sebagai acuan dalam proses delete data.

Dari contoh script PHP sebelumnya: **tampil.php**, yang digunakan untuk menampilkan data barang telah ditambahkan link pada setiap baris data menuju ke proses delete.

```
<a href="action.php?id=<?php echo $row['id_barang']; ?>" onClick="return confirm('Apakah benar akan menghapus data ini?')">Delete</a>
```

Link yang ditambahkan tersebut menuju ke script **delete.php** dengan parameter acuan untuk delete data adalah id barang.

Berikut ini merupakan contoh script PHP untuk melakukan proses delete data barang.

delete.php

```
<?php
include "configdb.php";
if(isset($_GET['id'])){
    $id = mysqli_real_escape_string($_GET['id']);
    $sql = "DELETE FROM tb_barang WHERE id_barang='$id'";
    $ex = mysqli_query($conn,$sql) or die("Delete data gagal!");
    $result = mysqli_affected_rows($conn);
    //cek jika eksekusi query berhasil dilakukan
    if ($result>0) {
        echo "<script>alert('Delete data berhasil..');
        location.href='tampil.php'; </script>";
    } else {
        echo "<script>alert('Delete data gagal..'); </script>"
    }
}
?>
```

Dari script tersebut, untuk melakukan proses delete data menggunakan id barang sebagai acuan. Hal ini dapat dilihat pada query SQL: "DELETE FROM tb_barang WHERE id_barang='\$id'", variabel \$id merupakan variabel yang berisikan id barang. Apabila proses delete data ini berhasil, maka akan menampilkan alert: "Delete data berhasil" dan kemudian diarahkan menuju ke halaman tampil.php untuk menampilkan keseluruhan data barang.

Latihan:

1. Cobalah kode PHP dan MySQL yang ada dalam modul ini, kemudian perhatikan dan pahami maksud dari kode PHP tersebut.

Tugas:

1. Buatlah sebuah aplikasi sederhana berbasis web untuk menangani keperluan pengisian buku tamu (guestbook). Data yang diisikan dalam form adalah sebagai berikut.

Nama
Email
Website
Komentar

Adapun tabel dalam database untuk menyimpan data isian buku tamu tersebut terdiri atas beberapa field sebagai berikut.

Id	integer	auto increment	primary key
Nama	varchar(30)		
Email	varchar(30)		
Website	varchar(30)		
Tanggal	datetime		
Komentar	text		

Keterangan:

Field "tanggal" digunakan untuk menyimpan informasi tanggal dan waktu kapan komentar tersebut dikirim (tanggal ini secara otomatis membaca tanggal dan waktu ketika komentar buku tamu di-submit). Tambahkan pula pagination untuk data buku tamu yang ditampilkan.

Petunjuk:

Uploadlah tugas tersebut pada folder “modul6” pada web hosting masing-masing. Buatlah laporan pembahasan singkat mengenai kode yang telah dibuat dalam format pdf dan laporan pembahasan tersebut diupload pula pada folder “modul6”.