



MODUL PRAKTIKUM PEMROGRAMAN WEB Semester III

Oleh:
Abdi Pandu Kusuma, S.Kom., M.T

Nama :

NIM :

Kelas :



**FAKULTAS TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS ISLAM BALITAR BLITAR**

MODUL 5

WEB DATABASE PROGRAMMING

(Pertemuan ke-5)

Petunjuk Praktikum

- Awali setiap kegiatan praktikum dengan doa, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.
- Tanyakan kepada dosen apabila ada hal-hal yang kurang jelas.

5.1 Koneksi Database

Akses dan manipulasi data pada database hanya dapat dilakukan jika terdapat koneksi antara file php dengan MySQL. Terdapat beberapa parameter yang digunakan untuk melakukan koneksi database, yaitu:

- Server name: nama server atau IP server dimana MySQL diinstal
- User name: nama user yang diberikan wewenang untuk mengakses database dalam MySQL
- Password: password yang dimiliki user dalam proses autentifikasi

Berikut adalah syntax php untuk melakukan koneksi terhadap database:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

$conn = mysqli_connect($servername, $username, $password);
?>
```

`mysqli_connect` merupakan fungsi php untuk menciptakan koneksi terhadap database.

a. Membaca Database

Salah satu bentuk akses dan manipulasi data yang bisa dilakukan melalui syntax php adalah membaca database. Parameter database name harus ditambahkan pada syntax agar database dapat terbaca oleh file php. Berikut adalah syntax php untuk membaca database:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$db = "web";

$conn = mysqli_connect($servername, $username, $password, $db);
```

```
?>
```

Selain menggunakan fungsi `mysqli_connect`, database dapat dibaca dengan fungsi `mysqli_select_db`. Berikut adalah syntax php untuk membaca database dengan fungsi `mysqli_select_db`:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

$conn = mysqli_connect($servername, $username, $password);

$db = mysqli_select_db($conn, "web");
?>
```

b. Membuat Database

Membuat database adalah bagian dari akses dan manipulasi data. Pernyataan query untuk membuat database adalah `CREATE DATABASE`, sehingga syntax php untuk membuat database adalah:

```
<?php
$sql = "CREATE DATABASE contoh";
mysqli_query($conn, $sql);
?>
```

Syntax tersebut menunjukkan bahwa fungsi `mysqli_query` melakukan eksekusi terhadap perintah `CREATE DATABASE` yang tersimpan pada variabel `$sql` yang berjalan di atas koneksi `$conn`.

c. Membuat Table

Bentuk akses dan manipulasi data selanjutnya adalah membuat table. Pernyataan query untuk membuat table adalah `CREATE TABLE`. Berikut adalah syntax php untuk membuat table:

```
<?php
$sql = "CREATE TABLE biodata(
    id INT NOT NULL,
    name VARCHAR(30),
    age INT,
    PRIMARY KEY(id))";
mysqli_query($conn, $sql);
?>
```

Sama halnya dengan membuat database, syntax tersebut menunjukkan bahwa fungsi `mysqli_query` mengeksekusi perintah `CREATE TABLE` yang tersimpan pada variabel `$sql` yang berjalan di atas koneksi `$conn`.

Note: sesuaikan server name, user name, dan password dengan setting pada server kalian

Praktik 1:**a. Koneksi Database**

Buat file koneksi.php dengan syntax sebagai berikut:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";

$conn = mysqli_connect($servername, $username, $password);
/* variabel $conn digunakan untuk menyimpan kegiatan koneksi
terhadap MySQL */

if (!$conn){
    die("Koneksi gagal: ".mysqli_connect_error());
}
//kondisi if dibuat untuk melakukan pengecekan terhadap koneksi

echo "Koneksi berhasil<br>";?>
```

Koneksi database dibuat secara terpisah agar dapat digunakan oleh file lain.

Buat file test.php untuk menjalankan file koneksi.php dengan syntax sebagai berikut:

```
<html>
<body>
<?php
require "koneksi.php";
?>
</body>
</html>
```

Pernyataan require digunakan untuk memasukkan file koneksi.php pada file test.php.

Jalankan file test.php dan amati hasilnya!

b. Membaca Database

Lakukan penambahan syntax, sehingga isi file koneksi.php menjadi sebagai berikut:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";

$conn = mysqli_connect($servername, $username, $password);
/* variabel $conn digunakan untuk menyimpan kegiatan koneksi
terhadap MySQL */

if (!$conn){
    die("Koneksi gagal");
}
//kondisi if dibuat untuk melakukan pengecekan terhadap koneksi

echo "Koneksi berhasil<br>";

$db = mysqli_select_db($conn, "web");

if (!$db){
    die("Gagal membaca database");
}

echo "Berhasil membaca database<br>";
?>
```

Jalankan kembali file test.php dan amati hasilnya!

c. Membuat Database

Buat file creatdb.php dengan syntax sebagai berikut:

```
<?php
require "koneksi.php";
echo "</br>";

$sql = "CREATE DATABASE contoh";
if (mysqli_query($connect, $sql)) {
    echo "Database berhasil dibuat";
} else {
    echo "Database gagal dibuat";
}
?>
```

Jalankan file creatdb.php dan amati hasilnya! Periksa juga perubahan yang terjadi pada database server Anda!

d. Membuat Table

Buat file creattb.php dengan syntax sebagai berikut:

```
<?php
require "koneksi.php";
echo "</br>";

$sql = "CREATE TABLE biodata(
        id INT NOT NULL,
        name VARCHAR(30),
        age INT,
        PRIMARY KEY(id))";

if (mysqli_query($connect, $sql)) {
    echo "Tabel berhasil dibuat";
} else {
    echo "Tabel gagal dibuat";
}
?>
```

Jalankan file creattb.php dan amati hasilnya! Periksa juga perubahan yang terjadi pada database server Anda!

5.2 Manipulasi Database.

Akses dan manipulasi data adalah berbagai aktifitas pengelolaan data pada database. Akses dan manipulasi data meliputi koneksi database, membaca database, membuat database dan table, membaca data pada database, menambah data pada database, mengubah data pada database, serta menghapus data pada database. Modul ini akan membahas lebih jelas tentang **manipulasi database** melalui syntax php. Fungsi `mysqli_query` digunakan untuk melakukan eksekusi terhadap pernyataan query yang kita kehendaki. Parameter fungsi `mysqli_query` terdiri dari koneksi dan pernyataan query itu sendiri.

a. Menambah Data

Menambah data adalah kemampuan akses dan manipulasi data yang paling mendasar. Pernyataan query untuk menambahkan data adalah INSERT. Berikut adalah syntax php untuk menambahkan data:

```
<?php
$sql = "INSERT INTO biodata(id, name, age)
      VALUES(1001, 'Budi', 20)";
mysqli_query($connect, $sql);
?>
```

Untuk mempermudah dan mempercepat proses penambahan data, umumnya aplikasi web disertai dengan form khusus.

b. Membaca Data

Secara umum membaca data adalah bentuk akses data yang paling sederhana. Pernyataan query untuk membuat table adalah SELECT. Pada syntax php, data yang dibaca dapat ditampilkan dalam beberapa cara. Berikut adalah syntax php untuk membaca dan menampilkan data:

```
<?php
$sql = "SELECT * from biodata";
$data = mysqli_query($connect, $sql);

while($show = mysqli_fetch_array($data)) {
    echo $show["id"]. "</br>". $show["name"]. "</br>". $show["age"];
    echo "</br></br>";
}
?>
```

Variable \$data menyimpan data yang berhasil dibaca oleh pernyataan query SELECT. Fungsi mysqli_fetch_array digunakan untuk menampilkan isi dari variable \$data tersebut.

Berikut fungsi php untuk menampilkan data dari database:

mysqli_fetch_array	Menyimpan hasil seleksi dalam bentuk array asosiatif \$show["id"] maupun array numerik \$show[2]
mysqli_fetch_assoc	Menyimpan hasil seleksi dalam bentuk array asosiatif \$show["age"]
mysqli_fetch_row	Menyimpan hasil seleksi dalam bentuk array numerik \$show[0]
mysqli_fetch_object	Menyimpan hasil seleksi dalam bentuk object \$show->name

c. Mengubah Data

Bentuk akses dan manipulasi data selanjutnya adalah mengubah data. Pernyataan query untuk mengubah data adalah UPDATE. Berikut adalah syntax php untuk mengubah data:

```
<?php
$sql = "UPDATE biodata SET age=22 WHERE id=1003";
mysqli_query($connect, $sql);
?>
```

Syntax tersebut menunjukkan bahwa fungsi `mysqli_query` mengeksekusi perintah UPDATE yang tersimpan pada variabel `$sql` yang berjalan di atas koneksi `$connect`.

d. Menghapus Data

Menghapus data adalah kegiatan akses dan manipulasi data yang tidak kalah penting. Pernyataan query untuk menghapus data adalah DELETE. Berikut adalah syntax php untuk menghapus data:

```
<?php
$sql = "DELETE FROM biodata WHERE age<21";
mysqli_query($connect, $sql);
?>
```

Syntax ini menunjukkan bahwa fungsi `mysqli_query` mengeksekusi perintah DELETE yang tersimpan pada variabel `$sql` yang berjalan di atas koneksi `$connect`.

Praktik 2:

a. Menambah Data

Buat file `insertdt.php` dengan syntax sebagai berikut:

```
<?php
require "koneksi.php";
echo "<br>";

$sql = "INSERT INTO biodata(id, name, age)
VALUES(1001, 'Budi', 20)";

if (mysqli_query($connect, $sql)) {
    echo "Data berhasil ditambahkan";
} else {
    echo "Data gagal ditambahkan";
}
?>
```

Jalankan file `insertdt.php` dan amati hasilnya! Periksa juga perubahan yang terjadi pada database server Anda!

b. Menambah Data melalui Form

Form digunakan untuk mempermudah proses penambahan data. Pada poin ini, dibutuhkan syntax html dan syntax php. Syntax html digunakan untuk membuat form sekaligus mengirim input data dari form. Syntax php digunakan untuk melakukan operasi penambahan data ke database.

Buat file form.html dengan syntax sebagai berikut:

```
<!DOCTYPE html>
<html>
<body>
<form action="insert.php" method="post">
  <p>
    Id:
    <input type="text" name="id" id="id">
  </p>
  <p>
    Name:
    <input type="text" name="name" id="name">
  </p>
  <p>
    Age:
    <input type="text" name="age" id="age">
  </p>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Buat file insert.php dengan syntax sebagai berikut:

```
<?php
require "koneksi.php";

$id = $_POST['id'];
$name = $_POST['name'];
$age = $_POST['age'];

$sql = "INSERT INTO biodata (id, name, age)
      VALUES ('$id', '$name', '$age')";

if(mysqli_query($connect, $sql)){
  echo "Data berhasil ditambahkan.";
} else{
  echo "Data gagal ditambahkan";
}
?>
```

Jalankan file form.html dan amati hasilnya! Periksa juga perubahan yang terjadi pada database server Anda!

c. Membaca Data

Buat file read.php dengan syntax sebagai berikut:

```
<?php
require "koneksi.php";
echo "<br>";

$sql = "select * from biodata";
```



```
$data = mysqli_query($connect,$sql);

while($show = mysqli_fetch_array($data)){
    echo $show["id"]."</br>".$show["name"]."</br>".$show["age"];
    echo "</br></br>";
}
?>
```

Jalankan file read.php dan amati hasilnya! Ubah fungsi `mysqli_fetch_array` menjadi `mysqli_fetch_assoc`, `mysqli_fetch_row`, dan `mysqli_fetch_object`.

d. Mengubah Data

Buat file update.php dengan syntax sebagai berikut:

```
<?php
require "koneksi.php";
echo "</br>";

$sql = "UPDATE biodata SET age=19 WHERE id=1003";
if (mysqli_query($connect, $sql)) {
    echo "Data berhasil diubah";
} else {
    echo "Data gagal diubah";
}
?>
```

Jalankan file update.php dan amati hasilnya! Periksa juga perubahan yang terjadi pada database server Anda!

e. Menghapus Data

Buat file delete.php dengan syntax sebagai berikut:

```
<?php
require "koneksi.php";
echo "</br>";

$sql = "DELETE FROM biodata WHERE age<21";
if (mysqli_query($connect, $sql)) {
    echo "Data berhasil dihapus";
} else {
    echo "Data gagal dihapus";
}
?>
```

Jalankan file delete.php dan amati hasilnya! Periksa juga perubahan yang terjadi pada database server Anda!

TUGAS PRAKTIKUM V:

- Buatlah tampilan web sederhana melanjutkan dari tugas praktikum ke-3 dengan penambahan CRUD!
- Tambahkan minimal 3 halaman pada tugas praktikum ini!