

PRAKTIKUM 5

Query Data

TUJUAN BELAJAR

Mahasiswa dapat menggunakan PHP untuk menyimpan data ke database MySQL

MATERI

PHP, Manajemen database dengan SQLYog, perintah-perintah SQL, perintah-perintah query dalam PHP, mendayagunakan PHP dalam inputan form

TUGAS

Menambahkan file pengolah data untuk kelima form input data

TEORI

PHP

Bahasa script jenis server-side (Javascript adalah bahasa script jenis client-side) artinya file-file PHP akan dieksekusi pada server. PHP terlihat bersama HTML, CSS, dan Javascript serta MySQL untuk membuat aplikasi atau situs web. Untuk menggunakan PHP harus menginstall dulu server web seperti XAMPP.

Sintaks PHP

Skrip PHP dibungkus dengan simbol `<?php` dan `?>`.

```
<?php  
echo "Hello world!"  
?>
```

Komentar

Catatan atau komentar dalam program diperlukan untuk penjelas atau pengingat kode program. Komentar tidak akan dieksekusi:

```
<?php
    /*Ini komentar jika
    lebih dari satu baris*/
    //Ini komentar satu baris
    echo "Hello world!"
?>
```

Case Sensitive

Perintah PHP bersifat non case sensitive, artinya PHP tidak membedakan huruf kecil dan kapital. Perintah echo dan ECHO sama saja. Tetapi untuk penamaan variabel, tidak demikian. Variabel \$COLOR dengan \$color adalah dua variabel yang berbeda.

Variabel

Untuk membuat variabel menggunakan simbol dollar (\$) diikuti nama variabel:

```
<?php
$teks1="Suhu udara saat ini ";
$cel=30;
echo $teks.$cel." derajat celcius"
?>
```

Akan menampilkan 'Suhu udara saat ini 30 derajat celcius'. Baris perintah terakhir menggunakan operator titik (.) untuk menggabungkan semua isi variabel dan teks.

Aturan penulisan variabel:

- Diawali dengan simbol dollar diikuti nama variabel
- Nama variabel harus dimulai dengan huruf atau garis bawah
- Nama variabel tidak boleh diawali dengan huruf
- Nama variabel hanya bisa menggunakan karakter alpa numerik (A-z, 0-9, _)
- Nama variabel bersifat case sensitive

Perintah echo dan print

Kedua perintah berfungsi sama untuk mencetak output ke layar. Bedanya perintah echo tidak mengembalikan nilai sementara print mengembalikan nilai 1. Output yang akan dicetak dibungkus dengan tanda petik. Untuk mencetak isi variabel, ada dua cara:

Meletakkan variabel didalam tanda petik:

```
$teks="UDINUS";  
echo "Belajar PHP di $teks";
```

atau menggunakan operator penggabungan string:

```
$teks="UDINUS";  
echo "Belajar PHP di ".$teks;
```

Tipe data PHP

Dalam PHP ada beberapa tipe data:

Tipe data	Contoh
string	"Warga Udinus disebut Dinusian"
integer	bilangan bulat mulai dari -2.147.483.648 s/d 2.147.483.648
float	bilangan pecahan, misalnya: 35,70
boolean	hanya mempunyai dua nilai true dan false. Biasa digunakan untuk pengujian suatu kondisi
array	<p>tipe data yang memiliki banyak nilai didalamnya yang disebut elemen array. Untuk membuat array menggunakan fungsi array(), contoh:</p> <pre>\$warna=array("merah","hijau","biru");</pre> <p>Untuk mengakses nilai variabel array harus menyebutkan indeks elemennya:</p> <pre>echo \$warna[0]; //akan mencetak 'merah'</pre>
Object	<p>tipe data yang menyimpan data dan informasi bagaimana mengakses data tersebut. Contoh:</p> <pre><?php class mobil { function mobil() { \$this->model = "MPV"; } }</pre>

	<pre>} // membuat sebuah object \$avanza = new mobil(); // menampilkan properti object echo \$avanza->model; ?></pre> <p>Tipe data ini harus dideklarasikan, tidak seperti tipe data lainnya</p>
null	<p>tipe data khusus yang hanya punya satu nilai: null. Sebuah variabel akan bertipe data null jika belum ada nilainya. Contoh:</p> <pre>\$x=null; var_dump(\$x);</pre>

Perintah Kondisional

Perintah kondisional adalah perintah percabangan. Dalam PHP sintaks perintah percabangan:

```
if (kondisi) {  
    kumpulan perintah jika kondisi bernilai true;  
} else {  
    kumpulan perintah jika kondisi bernilai false;  
}
```

Contoh:

```
<?php  
$t = date("H");  
if ($t < "9") {  
    echo "Selamat pagi!";  
} elseif ($t < "13") {  
    echo "Selamat siang!";  
} else {  
    echo "Selamat malam!";  
}  
?>
```

Perintah Perulangan

Perintah perulangan dalam PHP bersintaks:

```
while(kondisi true) {  
    perintah-perintah yang akan dieksekusi berulang;  
}
```

Contoh:

```
$x=1;  
while($x<=5) {  
    echo "Bilangan : $x <br>";  
    $x++;  
}
```

Array PHP

Merupakan tipe data khusus yang menyimpan serangkaian nilai dalam satu variabel.

Deklarasi array menggunakan fungsi array():

```
$web=array("HTML","CSS","Javascript","PHP");
```

untuk memperoleh panjang array menggunakan fungsi count():

```
$panj=count($web); //hasilnya 4
```

untuk mencetak nilai array menggunakan perulangan:

```
for ($x=0; $x<$panj; $x++) {  
    echo $web[$x]."<br>";  
}
```

Sorting Array

Elemen array dapat diurutkan:

```
sort($web);  
for ($x=0; $x<$panj; $x++) {  
    echo $web[$x]."-";  
}
```

/* Hasilnya:

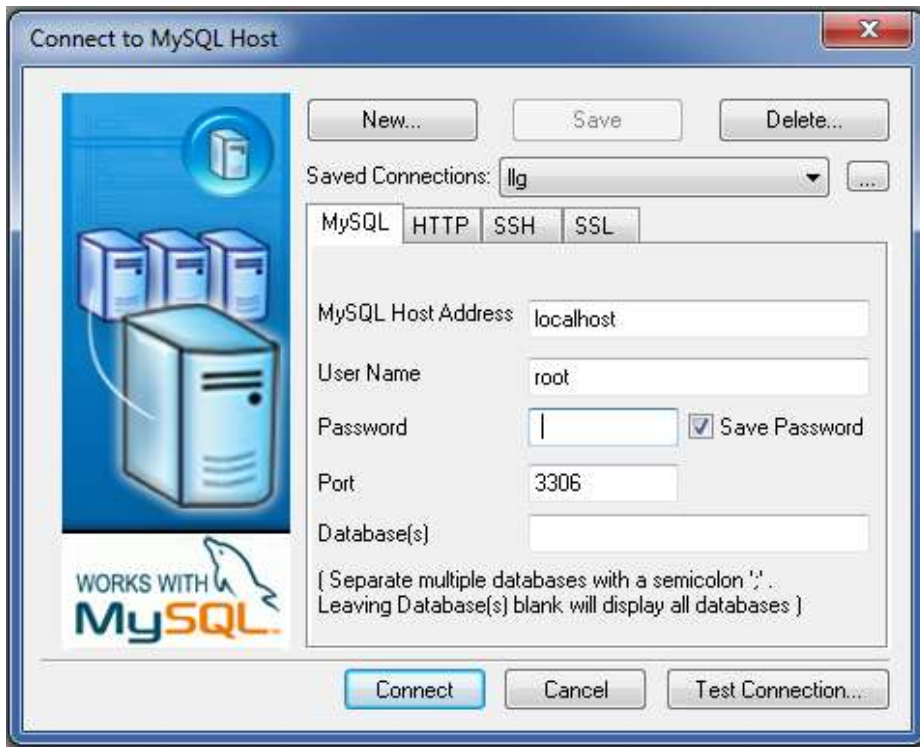
CSS-HTML-Javascript-PHP */

Manajemen Database MySQL

Untuk memanajemen data yang disimpan ke database MySQL dapat menggunakan berbagai perangkat lunak bantu, salah satunya adalah SQLYog.

Membuat koneksi

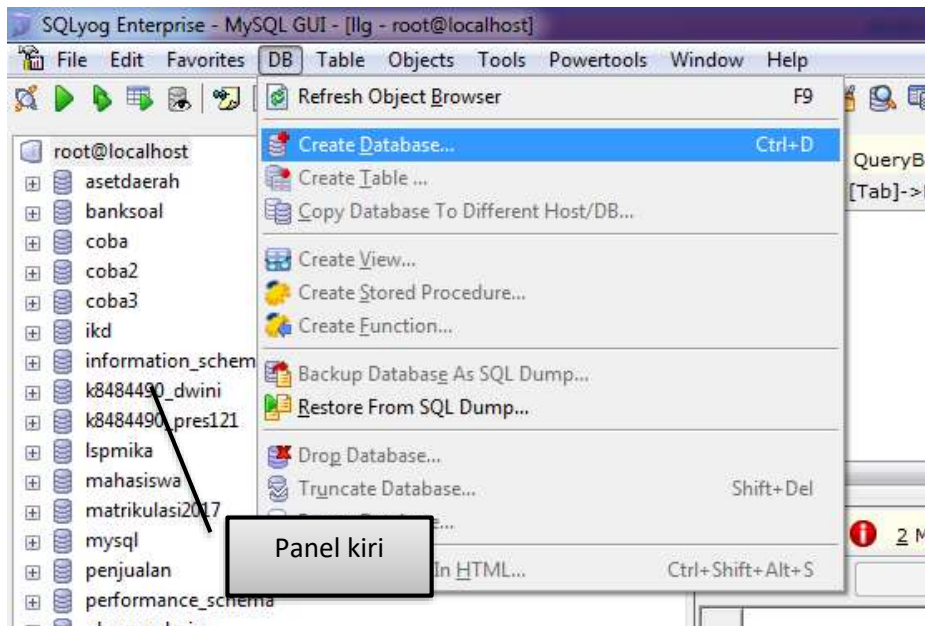
Untuk menggunakan SQLYog, pertama buat koneksi dulu ke server:



Isi MySQL Host Address dengan nama atau nomor IP server (192.168.10.250) atau localhost jika menggunakan pc atau laptop sendiri. User Name dengan nama user (a12xxxxxxx) atau root jika di pc sendiri, dan Password biasanya dikosongkan saja. Klik tombol Connect.

Membuat Database

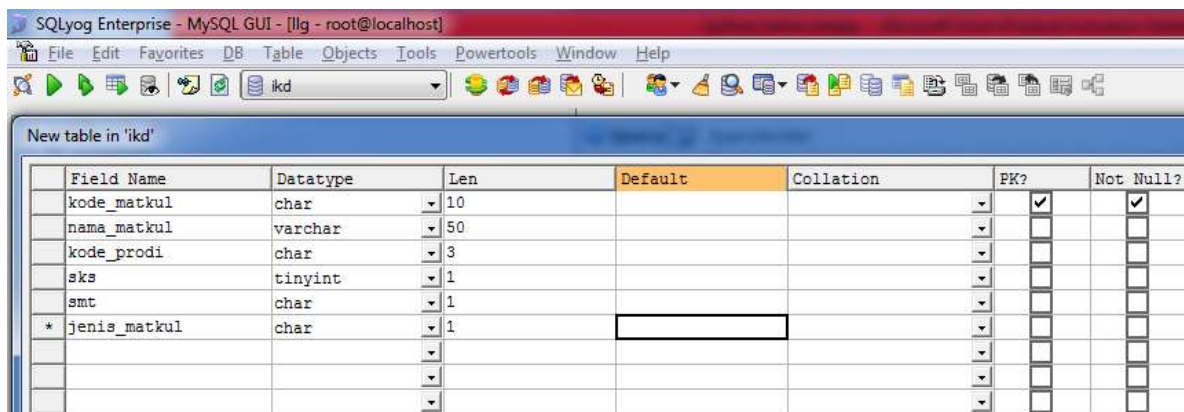
Buat database dengan menu DB – Create Database:



Tentukan nama database pada saat diminta.

Membuat Tabel

Pilih nama database yang baru saja dibuat di panel kiri lalu buat tabel dengan menu DB – Create Table:

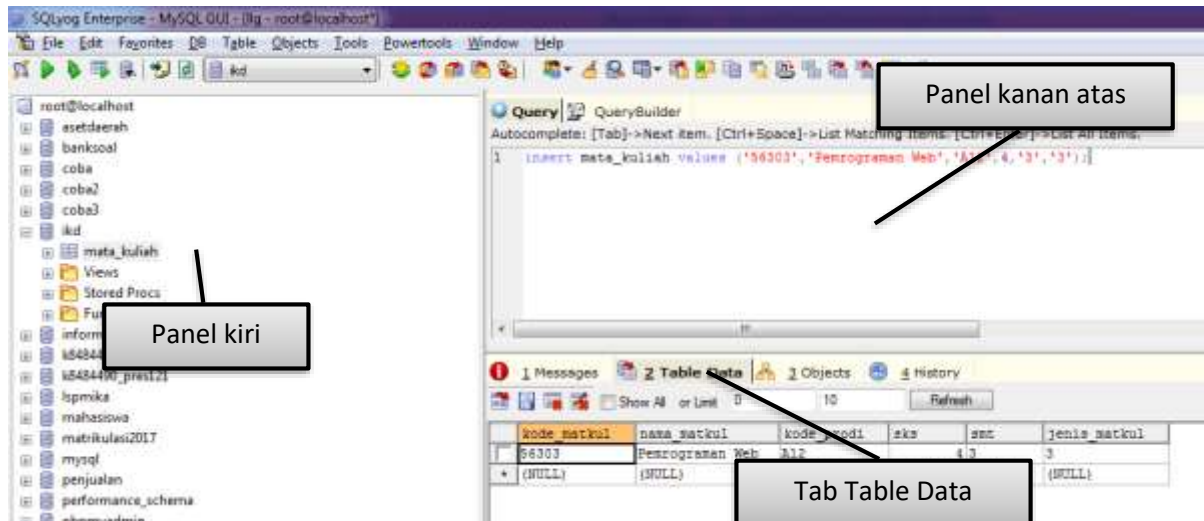


- kolom Field Name isi dengan nama field
- kolom Datatype isi dengan tipe data
- kolom Len isi dengan panjang dari field
- kolom PK? dicentang untuk memilih field kunci tabel.
- kolom Not Null? otomatis akan tercentang

Klik tombol Create Table.... lalu beri nama tabel ketika diminta.

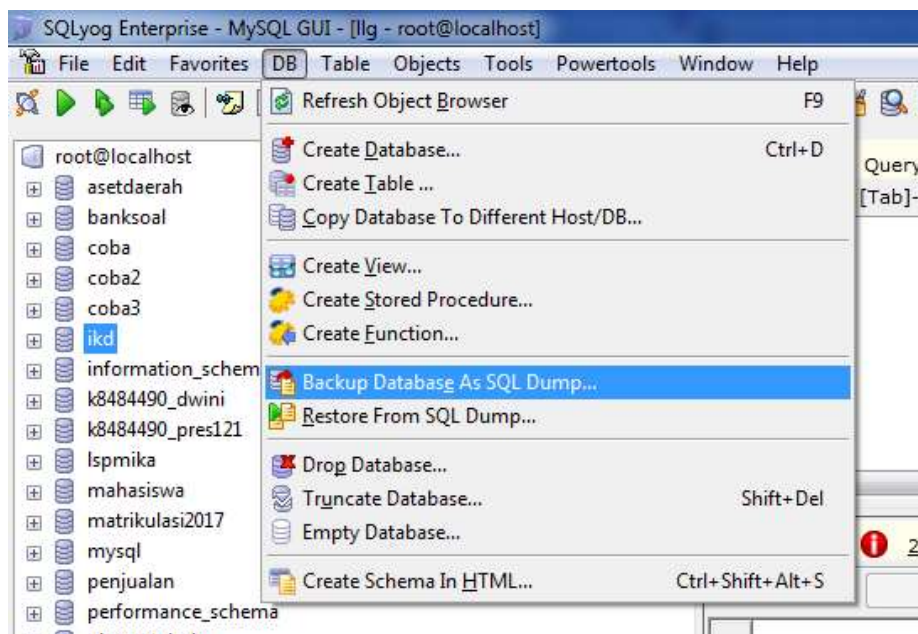
Memajemen Data Tabel

Untuk melihat isi tabel yang baru dibuat klik nama tabel di panel kiri lalu klik tab Table Data di panel kanan bawah. Untuk memasukkan rekod tulis perintah sql di panel kanan atas:

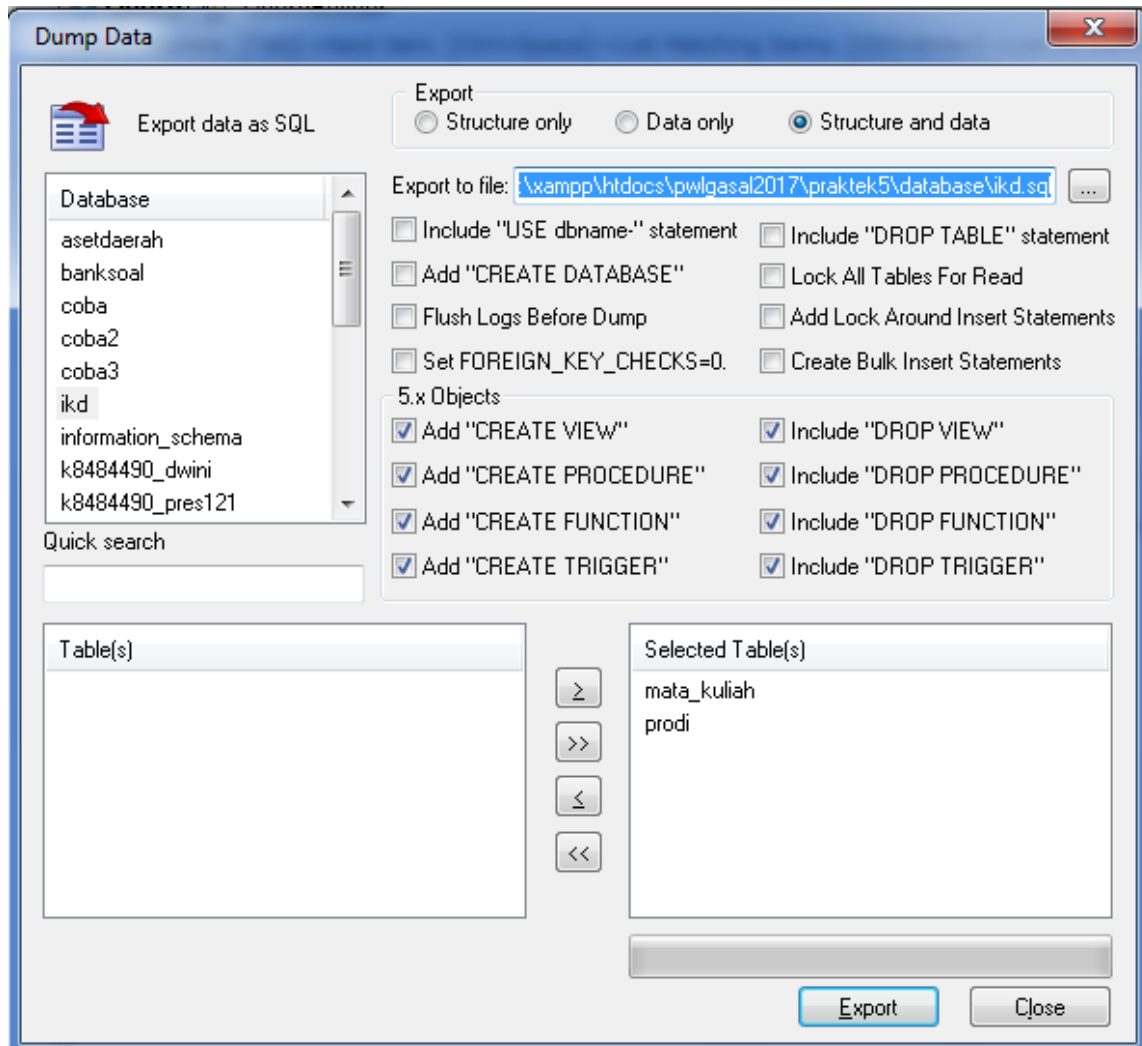


Membuat Backup Database

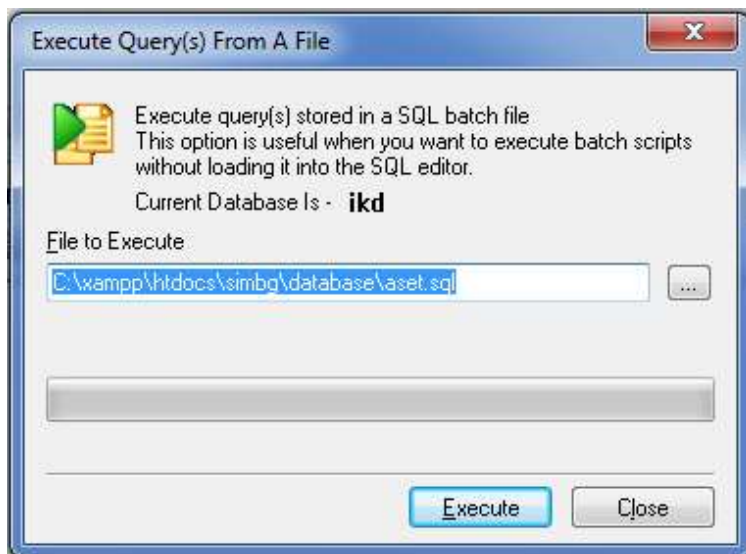
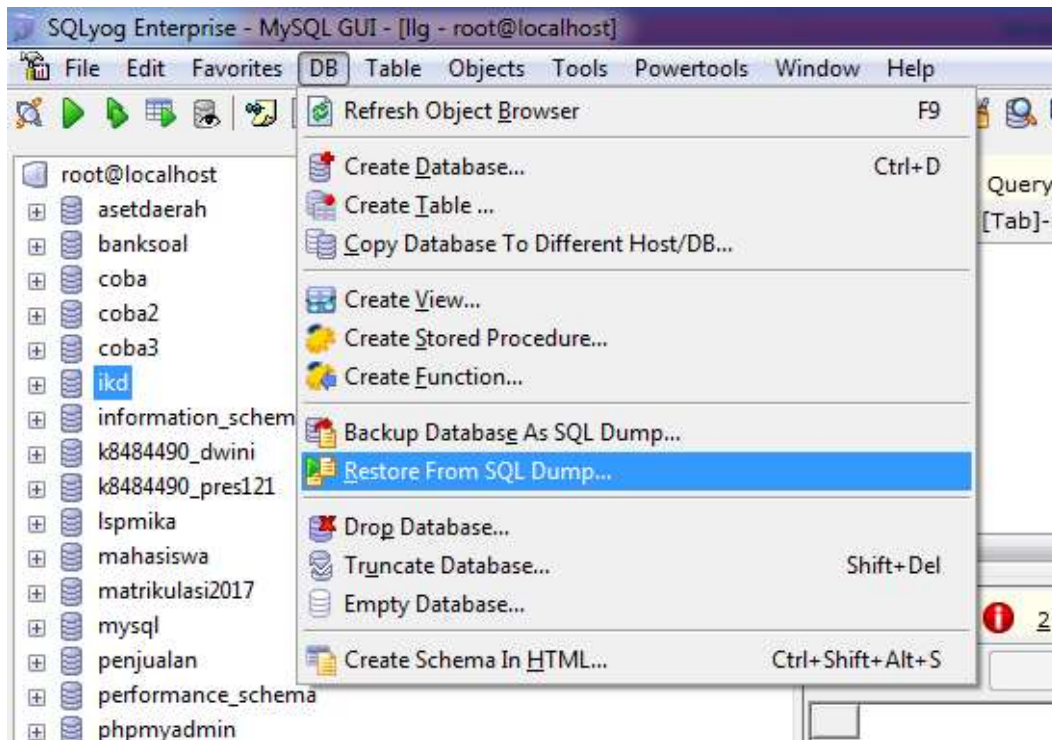
Untuk memindahkan database dari satu komputer ke komputer yang lain diperlukan backup database. Pilih Menu DB – Backup Database As SQL Dump:



Pilih database yang akan dibackup (Export data as sql), tentukan tabel apa saja yang akan dibackup (Selected Table(s)), dan lokasi file hasil backup (export to file). Klik tombol Export:



Hasilnya berupa file sql. Sedangkan untuk merestore pilih menu DB – Restore From SQL Dump lalu tentukan lokasi dan nama file backup:



SQL

Untuk manajemen data melalui program perlu bahasa SQL singkatan dari Structure Query Language, standar bahasa untuk menyimpan dan mengolah data database. Beberapa perintah penting:

- **Select**

Mengambil rekod dari tabel. Sintaks:

```
select kolom1, kolom2, ... from nama_tabel where kondisi;
```

Contoh:

- `select kode_matkul, nama_matkul, sks from mata_kuliah where sks=2;`
- `select * from mata_kuliah;`

- **Insert**

Menyimpan rekod baru ke tabel. Sintaks:

```
insert into nama_tabel(kolom1, kolom2,...) values (nilai1,nilai2,...);
```

Contoh:

```
insert into mata_kuliah(kode_matkul, nama_matkul, kode_prodi, sks, smt, jenis_matkul) values ('56303', 'Pemrograman Web', 'A12', 4, '3', '3');
```

- **Update**

Memodifikasi rekod dalam tabel. Sintaks:

```
update nama_tabel set kolom1=nilai1, kolom2=nilai2,... where kondisi;
```

Contoh:

```
update mata_kuliah set nama_matkul="Pemrograman Web Yuniior" where kode_matkul='56303';
```

- **Delete**

Menghapus rekod dalam tabel. Sintaks:

```
delete from nama_tabel where kondisi;
```

Contoh:

```
delete from mata_kuliah where kode_matkul='56303';
```

Perintah Query PHP

Query adalah pengolahan data di database. PHP memiliki kelompok perintah untuk memanipulasi data yang disimpan dalam database MySQL, yaitu:

- **mysqli_connect**

Untuk membuat koneksi ke server. Contoh:

```
$namaserver="localhost";
$namauser="a12xxxxxxx";
$password="123";
$db="ikd";
$kon=mysqli_connect($namaserver, $namauser, $password, $db);
if (!$kon) {
    die("Koneksi gagal: ".mysqli_connect_error());
}
```

- **mysqli_query**

Untuk mengeksekusi query. Contoh:

```
$namaserver="localhost";
$namauser="a12xxxxxxx";
$password="123";
$db="ikd";
$kon=mysqli_connect($namaserver, $namauser, $password, $db);
if (!$kon) {
    die("Koneksi gagal: ".mysqli_connect_error());
}
$sql="insert into mata_kuliah values ('56303', 'Pemrograman Web', 'A12', 4, '3', '3')";
mysqli_query($kon, $sql) or die("Query gagal: ".mysqli_error());
```

- **mysqli_close**

Untuk menutup koneksi. Contoh:

```
mysqli_close($kon);
```

- **mysql_fetch_row**

Untuk mengambil data dari variabel yang digunakan untuk query (mysqli_query). Contoh:

```
<select class="form-control" name="fkode_prodi">
    <?php
        $kon=mysqli_connect("localhost","root","","ikd") or
        die(mysqli_connect_error());
        $sql="select kode_prodi,nama_prodi from prodi";
        $qry=mysqli_query($kon,$sql) or die(mysqli_error());
        while($hasil=mysqli_fetch_row($qry)) {
            echo "<option value=$hasil[0]>$hasil[1]</option>"
        }
    ?>
</select>
```

Menggunakan PHP Dalam Input Form

Dalam suatu form terdapat inputan jenis select. Pilihan select dapat dibuat dengan menggunakan perulangan, array atau database. Contoh:

```
<label>Tanggal lahir</label>
<div>
    <select name="ftgl_lahir">
        <?php
            for ($i=1; $i<=31; $i++){
                echo "<option value=$i>$i</option>";
            }
        ?>
    </select>
</div>
```

Pilihan select dibuat dengan menggunakan perulangan for.

Session

Variabel-variabel web bersifat stateless. Artinya hanya bisa digunakan di halaman dimana variabel tersebut dibuat. Misalnya variabel \$username dan \$password yang dibuat di halaman login tidak bisa digunakan di halaman lain misalnya di halaman administrator. Padahal di halaman administrator variabel tersebut masih dibutuhkan untuk proses otentifikasi pengguna yang berhak masuk kedalam halaman administrator.

Session memungkinkan variabel-variabel tersebut dapat digunakan di berbagai halaman. Karena sekali variabel session dibuat maka tetap ada sampai browser ditutup atau dihapus dengan sengaja.

Menggunakan Session

Session harus diaktifkan terlebih dulu untuk mulai menggunakannya. Perintah mengaktifkan session adalah `session_start()`. Setelah diaktifkan kita dapat mulai menyimpan data kedalam variabel session yaitu `$_SESSION`.

Kemudian untuk menggunakan variabel session tersebut di halaman lain dimulai dengan mengaktifkan session terlebih dulu. Setelah itu variabel-variabel session dapat digunakan, misalnya dicetak atau untuk keperluan otentifikasi seorang pengguna.

Selanjutnya silahkan buka w3schools.com untuk penjelasan lebih detail tentang session.

PRAKTIKUM

Program login adalah program yang dibuat untuk masuk kedalam sebuah sistem atau aplikasi. Sering ditemui pada situs-situs internet seperti facebook, tokopedia, dan lain-lain.

Dalam praktek kali ini kalian diminta untuk membuat program login yang nanti akan dipakai untuk masuk kedalam aplikasi pengolahan data yang akan dibuat. Program login ini akan memerlukan mekanisme session.

Proses login:

- Pengguna memasukkan data akun berupa nama user dan password
- Data akun tersebut dicocokkan dengan data di database
- Jika ada maka data akun tersebut disimpan kedalam variabel session kemudian pengguna akan diarahkan ke halaman administrator. Jika tidak ada, akan ditampilkan pesan kegagalan login

Berikut adalah contoh kode program login:

```
<?php session_start()?>
<!DOCTYPE html>
<html>
<head>
    <title>Login Sistem</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="bootstrap4/css/bootstrap.css"
>
    <script src="bootstrap4/js/bootstrap.js"></script>
    <script src="bootstrap4/jquery/3.3.1/jquery-3.3.1.js"></script>
</head>
<body>
    <div class="container">
        <div class="w-25 mx-auto text-center mt-5">
            <div class="card bg-dark text-light">
                <div class="card-body">
                    <h2 class="card-title">LOGIN</h2>
                    <form method="post" action="">
                        <div class="form-group">
                            <label for="username">Nama user</label>
                            <input class="form-
control" type="text" name="username" id="username" autofocus>
                        </div>
                        <div class="form-group">
                            <label for="passw">Password</label>
                            <input class="form-
control" type="password" name="passw" id="passw">
                        </div>
                        <div>
                            <button class="btn btn-
info" type="submit">Login</button>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
<?php
if (isset($_POST['username'])){
    require "fungsi.php";
    $username=$_POST['username'];
    $passw=md5($_POST['passw']);
    $sql="select * from user where username='$username' and password='$pas
sw'";
    $hasil=mysqli_query($koneksi,$sql) or die(mysql_error($koneksi));
    $row=mysqli_fetch_assoc($hasil);
    if (mysqli_affected_rows($koneksi)>0){
```

```
$_SESSION['username']=$username;
header("location:homeadmin.php");
}else{
    echo "<div class='alert alert-danger w-25 mx-auto text-center mt-1 alert-dismissible'>
        <button type='button' class='close' data-dismiss='alert'>&times;</button>
        Maaf, login gagal. Ulangi login.
    </div>";
}
}
?>
</body>
</html>
```

Perintah `nilai` kosong pada atribut `action` dalam perintah `<form method="post" action="">` artinya data-data form akan dikirim ke file itu sendiri.

Perintah `isset($_POST['username'])` akan memeriksa data yang dikirim form. Jika ada kiriman data dari form maka serangkaian perintah dilakukan.

Fungsi `mysqli_affected_rows($koneksi)` untuk memeriksa apakah perintah `mysqli_fetch_assoc($hasil)` menghasilkan rekod yang berarti data login ada di database. Jika ya maka dibuat sebuah variabel session dengan perintah `$_SESSION['username']=$username` lalu pengguna diarahkan ke halaman administrator. Jika tidak maka akan ditampilkan pesan gagal login.

Untuk program login ini memerlukan tabel user yang terdiri dari field id, username, dan password. Field id menjadi kunci primer dan auto_increment.