

**MODUL PRAKTIKUM**  
**PEMROGRAMAN WEB**



**Materi 10 & 11:**  
**Koneksi Database**  
**&**  
**Cookie dan Session**

**Dosen:**  
**Febrianta Surya Nugraha, M.Kom**

**S1 INFORMATIKA**  
**STMIK AMIKOM SURAKARTA**  
**2020**

## 1. KONEKSI DATABASE

MySQLi atau MySQL Improved merupakan komponen yang digunakan untuk mengakses / berkomunikasi dengan database MySQL melalui PHP. Dengan menggunakan MySQLi, sebagai programmer PHP Anda dapat mengakses seluruh fasilitas terbaru yang ada pada MySQL versi 4.1 atau yang lebih baru.

### a) Koneksi PHP dan MySQLi

Bentuk koneksi dengan menggunakan mysqli adalah sebagai berikut.

```
mysqli_connect("hostname","user","password","database_name");
```

### b) Menampilkan Data

Setelah melakukan koneksi terhadap basis data yang akan dioperasikan, selanjutnya kita bisa memulai untuk melakukan manipulasi terhadap data yang terkandung dalam basis data tersebut.

Agar proses “migrasi” dari mysql ke mysqli tidak terlalu menyusahkan, PHP memberikan 2 alternatif cara penulisan mysqli.

#### *1. Procedural Style*

cara ini mirip dengan extension mysql. Kita dapat menggunakan fungsi-fungsi untuk mengakses database MySQL.

##### **a. mysqli\_query()**

Digunakan untuk melakukan eksekusi perintah SQL untuk memanipulasi database yang berhasil dilakukan koneksinya.

##### **b. mysqli\_fetch\_array()**

Digunakan untuk melakukan pemrosesan hasil query yang dilakukan dengan perintah mysqli\_query(), dan memasukkannya ke dalam array asosiatif, array numeris atau keduanya.

##### **c. mysqli\_fetch\_assoc()**

Fungsi ini hampir sama dengan fungsi mysqli\_fetch\_array(), hanya saja array yang dihasilkan hanya array asosiatif.

##### **d. mysqli\_fetch\_row()**

Fungsi ini hampir sama dengan fungsi mysqli\_fetch\_array(), hanya saja array yang dihasilkan hanya array numeris.

##### **e. mysqli\_num\_rows()**

Fungsi ini digunakan untuk menghitung jumlah record yang ada pada database.

## 2. Object Oriented Style,

cara ini lebih dekat dengan aturan penulisan pemrograman berorientasi object. Kedua jenis style ini menggunakan nama fungsi dan method yang kurang lebih sama. Sebagai contoh, pada procedural style mysqli terdapat fungsi `mysqli_query()`, sedangkan dalam Object oriented style dapat menggunakan method `$mysqli->query()`

**Pada praktikum ini, kita akan menggunakan Procedural style, untuk OOP style dapat dipelajari secara mandiri**

## 2. COOKIE DAN SESSION

### a) Cookie

Cookie adalah sebuah nilai yang dikirimkan dan ditanamkan server pada komputer client. Biasanya informasi-informasi yang disimpan dalam cookie ini adalah informasi yang berkaitan dengan user. Cookie diletakkan di sisi client, sehingga pengguna dapat melihat bahkan memodifikasi dan menghapus cookie tersebut. Hal tersebut seringkali membuat penyimpanan data menggunakan cookie menjadi tidak efektif, apalagi user dapat menonaktifkan penggunaan cookie melalui setting browser yang mengakibatkan penggunaan cookie menjadi sia-sia.

Setelah sebuah variabel cookie dideklarasikan, ia akan disimpan di sisi client dan selalu tersedia saat browser mengakses website (selama masa kadaluarsa cookie belum habis). Berdasarkan sifat yang telah di uraikan diatas, cookie dapat digunakan antara lain.

1. Menyimpan nama pengunjung.
2. Merekam daftar barang yang ingin dibeli pengunjung.
3. Menyimpan pilihan-pilihan yang diatur oleh pengunjung.
4. Menciptakan suatu sesi yang memungkinkan seseorang dapat masuk ke halaman-halaman lain tanpa perlu melakukan login kembali.

Untuk membuat sebuah cookie PHP telah menyediakan fungsi sebagai berikut.

**`setcookie(name, value, expire);`**

atau secara lebih lengkap:

**`setcookie(name, value, expire, path, domain, secure, httponly);`**

Hanya Paramater nama yang disyaratkan, opsi lain bersifat optional.

Keterangan:

1. **Name**, untuk nama cookie. Digunakan sebagai pengenalan cookie.
2. **Value**, berisi nilai yang akan disimpan dalam cookie.
3. **Expire**, merupakan batas waktu ketika cookie akan terhapus otomatis.

### b) Session

Session adalah salah satu fasilitas yang ada pada PHP yang digunakan untuk menyimpan data sementara ke dalam variabel (variabel session) sehingga data tadi dapat di akses oleh client selama variabel session tadi tidak dikosongkan atau dihilangkan.

Nilai variabel di dalam session di simpan di sisi server (web server). Berbeda dengan cookies yang nilai variabelnya disimpan di sisi client (browser). Jadi session relatif lebih aman digunakan untuk menyimpan variabel nilai yang bersifat rahasia seperti username dan password pada saat login.

### 1. Mengawali Session

Untuk menunjukkan bahwa suatu halaman menggunakan session maka pada awal halaman harus ada awal session yaitu dengan **session\_start ( )**

### 2. Penggunaan Variabel Session

Setelah session dimulai maka variabel session sudah dapat mulai digunakan. Penggunaannya menggunakan format **\$\_SESSION['nama\_variabel']**

### 3. Menghapus Session

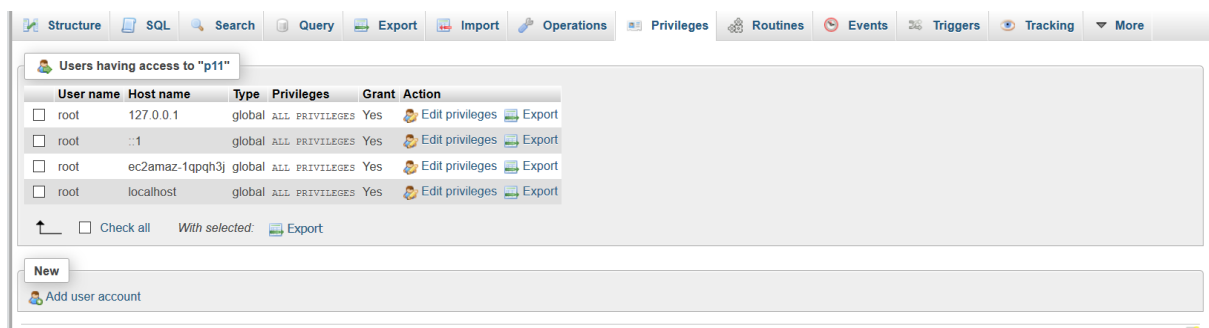
Setelah variabel session digunakan, variabel tersebut dapat dihapus. Banyak cara untuk menghapus session di antaranya sebagai berikut:

- \$\_SESSION['nama\_variabel'] = ""** untuk memberikan atau mengganti nilai dari variabel session menjadi null atau kosong.
- unset(\$\_SESSION['nama\_variabel'])** untuk menghapus sebuah variabel session.
- session\_destroy( )** untuk menghapus semua variabel session yang mungkin ada banyak variabel session yang dibuat.

## PRAKTIKUM

### 1. Membuat database

**Buat database**, kemudian klik database dan masuk ke menu privilege.



buat akun baru untuk mengakses database.

## Add user account

**Login Information**

User name:

Use text field:

Host name:

Any host

Password:

Use text field:

Strength:  Very weak

Re-type:

Authentication Plugin:

Native MySQL authentication

Generate password:

Generate

**Database for user account**

☐ Create database with same name and grant all privileges.

☐ Grant all privileges on wildcard name (username\_%).

☒ Grant all privileges on database p11.

**Global privileges** ☒ Check all

Note: MySQL privilege names are expressed in English.

**Data**

☒ SELECT  
☒ INSERT  
☒ UPDATE  
☒ DELETE  
☒ FILE

**Structure**

☒ CREATE  
☒ ALTER  
☒ INDEX  
☒ DROP  
☒ CREATE TEMPORARY TABLES  
☒ SHOW VIEW  
☒ CREATE ROUTINE  
☒ ALTER ROUTINE  
☒ EXECUTE  
☒ CREATE VIEW  
☒ EVENT  
☒ TRIGGER

**Administration**

☒ GRANT  
☒ SUPER  
☒ PROCESS  
☒ RELOAD  
☒ SHUTDOWN  
☒ SHOW DATABASES  
☒ LOCK TABLES  
☒ REFERENCES  
☒ REPLICATION CLIENT  
☒ REPLICATION SLAVE  
☒ CREATE USER

**Resource limits**

Note: Setting these options to 0 (zero) removes the limit.

MAX QUERIES PER HOUR

MAX UPDATES PER HOUR

MAX CONNECTIONS PER HOUR

MAX USER\_CONNECTIONS

**SSL**

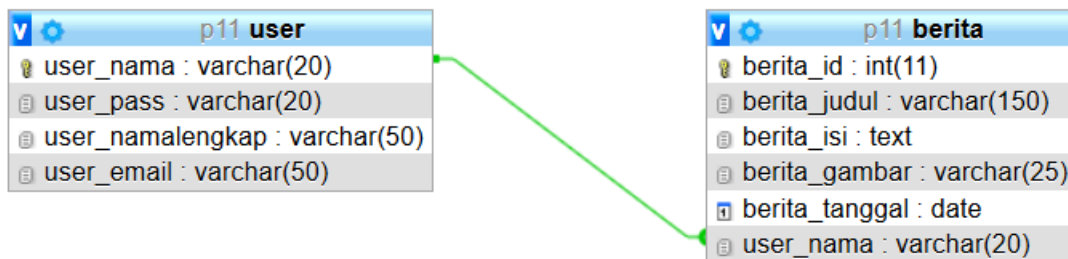
☒ REQUIRE NONE  
☐ REQUIRE SSL  
☐ REQUIRE X509  
☐ SPECIFIED

REQUIRE CIPHER

REQUIRE ISSUER

Console

Buatlah sebuah database menggunakan phpMyAdmin sesuai struktur di bawah ini



Langkah-langkahnya:

1. Buat tabel user

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	Comments
user_nama	VARCHAR	20	None			<input type="checkbox"/>	PRIMARY	
user_pass	VARCHAR	20	None			<input type="checkbox"/>		
user_namalengkap	VARCHAR	50	None			<input type="checkbox"/>		
user_email	VARCHAR	50	None			<input type="checkbox"/>		

Table comments: Collation: Storage Engine: InnoDB

PARTITION definition:

Partition by: ( Expression or column list )

Partitions:

Preview SQL Save

atau dapat dengan menggunakan query dibawah ini:

```
CREATE TABLE `p11`.`user` ( `user_nama` VARCHAR(20) NOT NULL , `user_pass` VARCHAR(20) NOT NULL , `user_namalengkap` VARCHAR(50) NOT NULL , `user_email` VARCHAR(50) NOT NULL , PRIMARY KEY (`user_nama`)) ENGINE = InnoDB;
```

## 2. Buat tabel berita

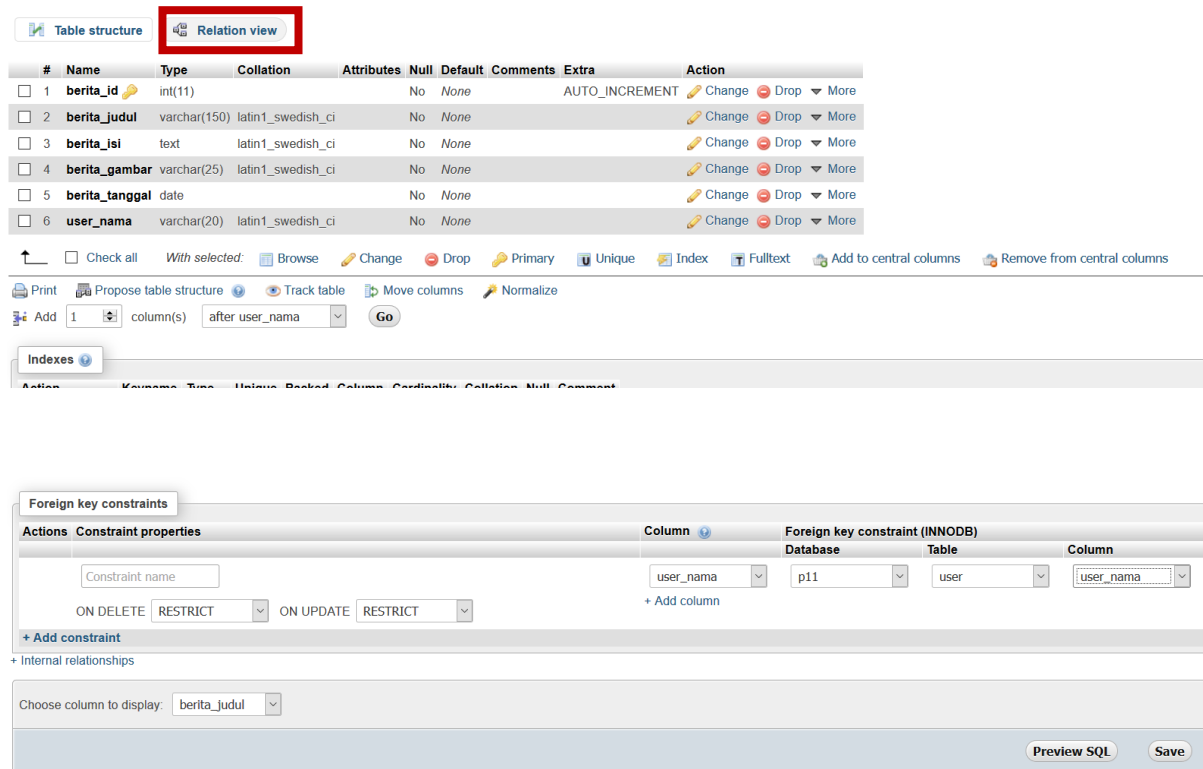
Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	Comments	Virtuality
berita_id	INT	11	None			<input type="checkbox"/>	<input checked="" type="checkbox"/>		
berita_judul	VARCHAR	150	None	latin1_swedish_ci		<input type="checkbox"/>	<input type="checkbox"/>		
berita_isi	TEXT		None	latin1_swedish_ci		<input type="checkbox"/>	<input type="checkbox"/>		
berita_gambar	VARCHAR	25	None	latin1_swedish_ci		<input type="checkbox"/>	<input type="checkbox"/>		
berita_tanggal	DATE		None			<input type="checkbox"/>	<input type="checkbox"/>		
user_nama	VARCHAR	20	None	latin1_swedish_ci		<input type="checkbox"/>	<input type="checkbox"/>		

Preview SQL Save

atau dapat menggunakan query dibawah:

```
CREATE TABLE `p11`.`berita` (
  `berita_id` INT(11) NOT NULL AUTO_INCREMENT,
  `berita_judul` VARCHAR(150) NOT NULL,
  `berita_isi` TEXT NOT NULL,
  `berita_gambar` VARCHAR(25) NOT NULL,
  `berita_tanggal` DATE NOT NULL,
  `user_nama` VARCHAR(20) NOT NULL,
  PRIMARY KEY(`berita_id`)
) ENGINE = INNODB;
```

**3. Buatlah koneksi antar tabel untuk membuat hubungan antara tabel user dan tabel berita.** Gunakan langkah berikut.



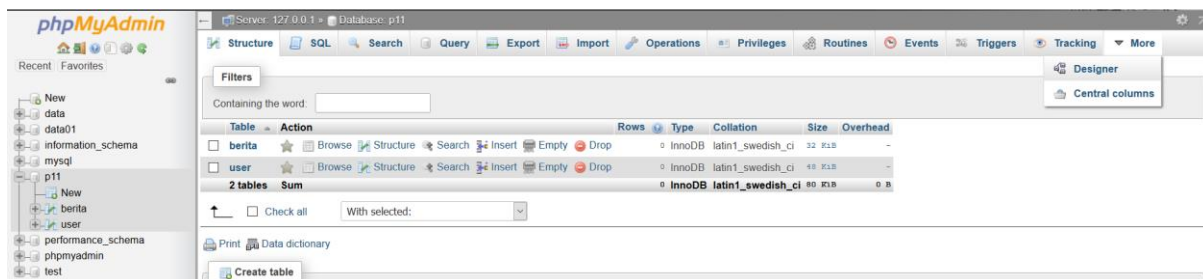
Keterangan gambar diatas adalah:

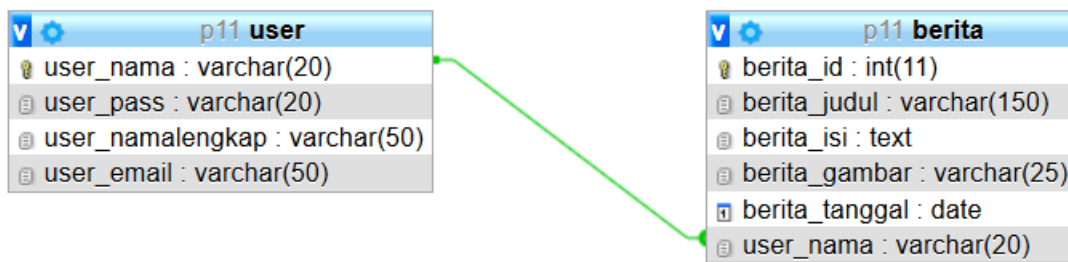
- Column** berisi field *foreign key* (**user\_nama**) dari tabel berita
- Database** berisi nama database yang saat ini digunakan
- Table** adalah pilihan untuk tabel induk (tabel user)
- Column** adalah pilihan field *primary key* (**user\_nama**) dari tabel user

Jika **user\_nama** pada kolom FK tidak muncul, periksa kembali apakah **user\_nama** pada kolom user sudah menjadi **Primary Key** atau belum.

Apabila terdapat kesalahan periksa kembali apakah **user\_nama** pada tabel user maupun berita sudah sama **penulisannya, tipe data nya dan ukuran datanya**.

**4. Anda dapat melihat visualisasi relasi tabel dengan masuk ke database, kemudian menu designer.**





## 5. Isikan masing-masing data untuk tabel user dan berita melalui phpMyadmin.

Column	Type	Function	Null	Value
user_nama	varchar(20)			admin
user_pass	varchar(20)	MD5		admin123
user_namalengkap	varchar(50)			administrator
user_email	varchar(50)			admin@pwl.id

**Go**

Anda dapat memberikan enkripsi pada atribut user\_pass dengan memilih function md5 dengan mengubah ukuran panjang character pada user\_pass menjadi 255.

Column	Type	Function	Null	Value
berita_id	int(11)			1
berita_judul	varchar(150)			Belajar Web
berita_isi	text			Belajar web lanjut menggunakan PHP
berita_gambar	varchar(25)			
berita_tanggal	date			2020-01-21
user_nama	varchar(20)			paijo - paijo

**Go**

Atribut berita gambar kita kosongkan terlebih dahulu.



## 2. Membuat file Koneksi di PHP dan file PHP untuk menampilkan data

### a. Membuat Koneksi Database config.php

```
1 <?php
2 $config = mysqli_connect("localhost", "pwl", "██████", "p11");
3 if (!$config) {
4     die('Gagal terhubung ke MySQL: '.mysqli_connect_error());
5 }
6 ?>
```

### b. Menampilkan data tunggal username.php

```
1 <?php
2 include "config.php"; //memanggil file koneksi database
3
4 $sql = "Select * from user";
5 $query= mysqli_query($config,$sql);
6 $row= mysqli_fetch_assoc($query);
7
8 //data ditampilkan dalam bentuk array bentuk pertama
9 echo "<pre>";
10 print_r($row);
11 echo "<pre>";
12
13 //bentuk kedua
14 echo "username : " . $row['user_nama'] . "<br />";
15 echo "Nama : " . $row['user_namalengkap'] . "<br />";
16 echo "email : " . $row['user_email'] . "<br />";
17
18 ?>
```

#### Keterangan:

- Fungsi include bertugas untuk memasukkan file config.php ke dalam file username.php. Tugasnya membuat koneksi ke server database.
- Query “SELECT \* from user” disimpan di dalam variabel \$sql. Kemudian dijalankan dengan perintah mysqli\_query(\$config,\$sql) yang disimpan di dalam variabel \$query.
- Hasil query yang tersimpan di variabel \$row adalah data mentah yang akan diolah dengan fungsi mysqli\_fetch\_assoc(\$query) yang berguna untuk mengolah data mentah tersebut ke dalam bentuk array. Selanjutnya disimpan di dalam variabel \$row.
- Selanjutnya adalah perintah untuk menampilkan data. Variabel \$query disimpan dalam bentuk array sehingga dapat memanfaatkan fungsi print\_r untuk menampilkan struktur array tersebut. Atau dalam bentuk kedua array \$query yang ditampilkan merupakan kolom yang sebelumnya didefinisikan pada query (username, nama lengkap, dan email), sehingga untuk menampilkan kita perlu menyebut array dengan index asosiatifnya.

### c. Menampilkan data jamak

Tambahkan 1 data lagi kedalam tabel user.

Untuk menampilkan data yang banyak perlu dideskripsikan perintah `mysqli_fetch_assoc($query)` yang dilakukan secara berulang hingga data yang diinginkan telah mencukupi atau sampai kondisi dimana data yang dihasilkan dari query telah habis. Oleh sebab itu, perlu sebuah perulangan yang akan menampilkan data yang jumlahnya banyak.

```
config.php x username.php x username2.php x
1  <?php
2  include "config.php"; //memanggil file koneksi database
3
4  $sql = "Select * from user";
5  $query= mysqli_query($config,$sql);
6
7  echo "<pre>";
8  while ($row =mysqli_fetch_assoc($query)){
9      //menampilkan data jamak
10     print_r($row);
11 }
12 echo "<pre>";
13
14 ?>
15
```

## 3. Latihan Session dan Cookie

### a. Latihan Cookie

#### latcookie.php

Membuat cookie dengan script sebagai berikut:

```
latcookie.php x aksescookie.php x hapuscookie.php x newcookie.php x modc
1  <?php
2      $value="12";
3      setcookie("barang",@value,time()+100);
4      echo "Cookie telah dibuat";
5  ?>
6
```

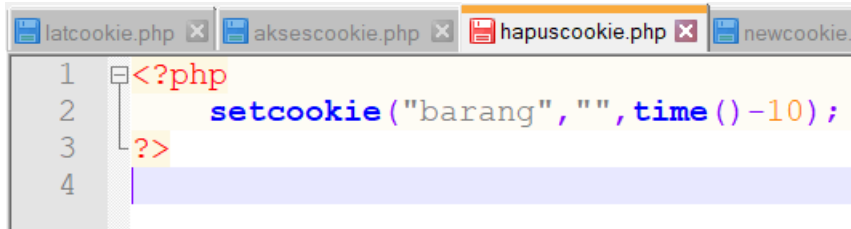
Script membaca / mengakses cookie

#### aksescookie.php

```
latcookie.php x aksescookie.php x hapuscookie.php x newcookie.php
1  <?php
2      echo "Jumlah Barang :";
3      if (isset ($_COOKIE['barang'])) {
4          echo $_COOKIE['barang'];
5      }
6  ?>
7
```

Membuat script untuk menghapus cookie. Untuk menghapus nilai cookie gunakan `setcookie()` dengan menyebutkan nama cookie pada argumen pertama dan string kosong pada argumen kedua dan mengisikan nilai dari parameter `expire`-nya. Cara lain yaitu argumen kedua tidak disebutkan.

### hapuscookie.php



Mengetahui bilangan yang digunakan pada time cookie. dibawah ini adalah contoh script cookie sederhana yang diatur dengan durasi 1 hari. Berpakah nilai 1 pada angka yang digunakan secara default pada script cookie?

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
// 86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

**Note:** The `setcookie()` function must appear BEFORE the `<html>` tag.

**Note:** The value of the cookie is automatically URLencoded when sending the cookie, and automatically decoded when received.

The example creates a cookie named "user" with the value "John Doe". The cookie will expire after 30 days (86400 \* 30). The "/" means that the cookie is available in entire website (otherwise, select the directory you prefer).

## b. Latihan Session untuk Autentikasi Login

Buatlah tabel berikut

The screenshot shows the 'Structure' tab for a table named 'user' in a database named 'data'. The table has three columns: 'id' (INT(11), PRIMARY, NOT NULL), 'username' (VARCHAR(255), NOT NULL), and 'password' (VARCHAR(255), NOT NULL). The storage engine is InnoDB. The 'PARTITION definition' section is empty.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	Comments
id	INT	11	None			<input type="checkbox"/>	PRIMARY	
username	VARCHAR	255	None			<input type="checkbox"/>		
password	VARCHAR	255	None			<input type="checkbox"/>		

```
CREATE TABLE `data`.`user` ( `id` INT(11) NOT NULL AUTO_INCREMENT ,  
`username` VARCHAR(255) NOT NULL , `password` VARCHAR(255) NOT NULL , PRIMARY  
KEY (`id`)) ENGINE = InnoDB;
```

Isikan data berikut

The screenshot shows the 'Insert' tab for the 'user' table. The 'id' column is set to 'int(11)' and is empty. The 'username' column is set to 'varchar(255)' and contains the value 'admin'. The 'password' column is set to 'varchar(255)' and contains the value 'admin'. The 'Go' button is at the bottom right.

Column	Type	Function	Null	Value
id	int(11)			
username	varchar(255)			admin
password	varchar(255)			admin

Buatlah sebuah folder, kemudian buatlah file koneksi database:

**koneksi.php**

```

<?php
$koneksi = mysqli_connect("localhost","pwl","330187","data");

// Check connection
if (mysqli_connect_errno()){
    echo "Koneksi database gagal : " . mysqli_connect_error();
}

?>

```

## Membuat halaman utama untuk login: index.php

```

<!DOCTYPE html>
<html>
<head>
    <title>Membuat Login Dengan PHP dan MySQLi</title>
</head>
<body>
    <h2>Login </h2>
    <br/>
    <!-- cek pesan notifikasi -->
    <?php
    if(isset($_GET['pesan'])){
        if($_GET['pesan'] == "gagal"){
            echo "Login gagal! username dan password salah!";
        }else if($_GET['pesan'] == "logout"){
            echo "Anda telah berhasil logout";
        }else if($_GET['pesan'] == "belum_login"){
            echo "Anda harus login untuk mengakses halaman admin";
        }
    }
    ?>
    <br/>
    <br/>
    <form method="post" action="cek_login.php">
        <table>
            <tr>
                <td>Username</td>
                <td>:</td>
                <td><input type="text" name="username"
placeholder="Masukkan username"></td>
            </tr>
            <tr>
                <td>Password</td>
                <td>:</td>
                <td><input type="password" name="password"
placeholder="Masukkan password"></td>
            </tr>
            <tr>
                <td></td>
                <td></td>
                <td><input type="submit" value="LOGIN"></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

Kemudian buat file cek\_login.php sebagai tindak lanjut atas aksi button form pada index.php

### cek\_login.php

```
<?php
// mengaktifkan session php
session_start();

// menghubungkan dengan koneksi
include 'koneksi.php';

// menangkap data yang dikirim dari form
$username = $_POST['username'];
$password = $_POST['password'];

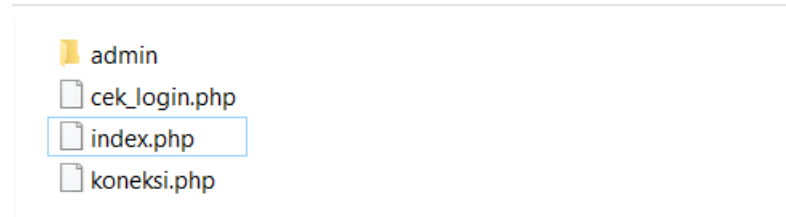
// menyeleksi data admin dengan username dan password yang sesuai
$data = mysqli_query($koneksi,"select * from user where
username='$username' and password='$password'");

// menghitung jumlah data yang ditemukan
$cek = mysqli_num_rows($data);

if($cek > 0){
    $_SESSION['username'] = $username;
    $_SESSION['status'] = "login";
    header("location:admin/index.php");
}else{
    header("location:index.php?pesan=gagal");
}
?>
```

Buatlah folder bernama admin pada path sesuai dengan file-file yang ada diatas. contohnya:

is PC > OS (C:) > xampp > htdocs > webp11 >



Kemudian kita buat file index.php tetapi yang berada didalam folder admin.

### admin/index.php

```

<!DOCTYPE html>
<html>
<head>
    <title>Membuat Login Dengan PHP dan MySQLi</title>
</head>
<body>
    <h2>Halaman Admin</h2>

    <br/>

    <!-- cek apakah sudah login -->
    <?php
        session_start();
        if($_SESSION['status']!="login"){
            header("location:../index.php?pesan=belum_login");
        }
    ?>

    <h4>Selamat datang, <?php echo $_SESSION['username']; ?>! anda telah login.</h4>

    <br/>
    <br/>

    <a href="logout.php">LOGOUT</a>

</body>
</html>

```

Kemudian buatlah file untuk melakukan logout sekaligus untuk menghapus session.

```

<?php
// mengaktifkan session
session_start();

// menghapus semua session
session_destroy();

// mengalihkan halaman sambil mengirim pesan logout
header("location:../index.php?pesan=logout");
?>

```

## Menggunakan enkripsi md5 pada login

tambahlah data admin2 dengan menggunakan fungsi enkripsi md5.

← Server: 127.0.0.1 » Database: data » Table: user

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#)

Column	Type	Function	Null	Value
id	int(11)			
username	varchar(255)			admin2
password	varchar(255)	MD5		admin2

[Go](#)

← Server: 127.0.0.1 » Database: data » Table: user

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Help](#)

✓ Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

`SELECT * FROM `user``

☐ Show all | Number of rows: 25 | Filter rows:  | Sort by key:

+ Options

	id	username	password
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	admin	admin
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	2	admin2	c84258e9c39059a89ab77d846ddab909

kemudian cobalah untuk login dengan menggunakan akun admin2.

Meskipun kita telah menuliskan password dengan benar, sistem tidak dapat menerima password pada admin2 karena dalam database terenkripsi. jadi kita perlu melakukan perubahan script pada cek\_login.php.



```
cek_login.php x koneksi.php x index.php x logout.php x index.php x
1 <?php
2 // mengaktifkan session php
3 session_start();
4
5 // menghubungkan dengan koneksi
6 include 'koneksi.php';
7
8 // menangkap data yang dikirim dari form
9 $username = $_POST['username'];
10 $password = md5($_POST['password']);
11
12 // menyeleksi data admin dengan username dan password yang sesuai
13 $data = mysqli_query($koneksi,"select * from user where username='$username'
14 ' and password='$password'");
15
16 // menghitung jumlah data yang ditemukan
17 $cek = mysqli_num_rows($data);
18
19 if($cek > 0){
20     $_SESSION['username'] = $username;
21     $_SESSION['status'] = "login";
22     header("location:admin/index.php");
23 }else{
24     header("location:index.php?pesan=gagal");
25 }
26 ?>
```

Cobalah login kembali dengan menggunakan admin2.

admin2 dapat login, akan tetapi kita tidak dapat login dengan menggunakan admin karena perubahan diatas.

# Ujian Akhir Semester

Tugas Ujian Akhir Semester anda bersifat pribadi yaitu membuat website dinamis php dengan tema yang anda pilih dengan memiliki minimal halaman login, dan halaman yang dapat mengoperasikan CRUD (Create, Read, Update Delete) (materi yang akan datang).

Selain Tugas tersebut, UAS juga mengumpulkan laporan praktikum.

Untuk Ketentuan ujian akan diperjelas saat materi CRUD.