



Modul Pemrograman Web

Ilmu Komputer

Pertemuan 11:

Penganganan Form dan Login PHP

Disusun Oleh:

Asisten Dosen Pemrograman Web

Jurusan Ilmu Komputer Universitas Lampung

Tim Penyusun

Pengarah

Rizky Prabowo, M.Kom.

M. Iqbal Parabi, S.Si., M.T.

Penyusun

- M Arya Dzaky A (2217051007)
- Gilbert Hasiholan Sibuea (2217051118)
- Dafa Farhan Haqiqi (2267061001)
- Luthfi Aditya (2217051062)
- Berli Anta Atrizki (2217051076)
- Cakrawangsa Praba Alkautsar (2217051003)
- M Nurfaiz Satriani Azra (2217051001)
- Jhon Valentino Nababan (2217051087)

Gambaran Umum

Materi ini membahas tentang penanganan form dan proses login menggunakan PHP native. Dalam pengembangan aplikasi web, form merupakan elemen penting untuk menerima input dari pengguna, seperti data login. Oleh karena itu, memahami cara kerja form dan bagaimana data ditangani oleh PHP menjadi dasar yang sangat penting. Selain itu, sistem login adalah salah satu fitur keamanan utama dalam aplikasi web. Materi ini akan membimbing peserta untuk memahami alur kerja form login, mulai dari pembuatan tampilan form, pengambilan data input, proses validasi terhadap database, hingga pengelolaan sesi pengguna menggunakan session.

Capaian Pembelajaran

Setelah menyelesaikan materi ini, mahasiswa diharapkan mampu:

1. Menjelaskan alur kerja form login berbasis web.
2. Mengimplementasikan sistem login sederhana menggunakan PHP dan MySQL.
3. Menggunakan session untuk menjaga otentikasi pengguna.
4. Mengamankan data login menggunakan teknik hashing password.
5. Menangani form input dan validasi dasar.

Materi Praktikum

Jenis-jenis form

Form inputan dibuat dengan tag-tag HTML. Halaman yang mengandung form murni (tidak ada script php) tidak harus disimpan dalam bentuk php, bisa dalam bentuk html. Untuk merancang sebuah form inputan, setidaknya ada 3 (tiga) hal penting, yaitu:

1. METHOD Method dari sebuah form menentukan bagaimana data inputan form dikirim. Method ini ada dua macam, yaitu GET dan POST. Method ini menentukan bagaimana data inputan dikirim dan diproses oleh PHP.
2. ACTION Action dari sebuah form menentukan dimana data inputan dari form diproses. Jika action ini dikosongkan, maka dianggap proses form terjadi di halaman yang sama. Jadi halaman form dan halaman proses bisa saja dipisah atau dijadikan satu.
3. SUBMIT BUTTON Submit button merupakan sebuah tombol (pada umumnya) yang berfungsi sebagai trigger pengiriman data dari form inputan. Jika tombol ini ditekan, maka data form akan dikirimkan (diproses) di halaman yang sudah ditentukan pada atribut action.

Jenis – jenis method

1. GET

Metode GET digunakan untuk mengirimkan data melalui URL. Data yang dikirimkan dengan metode GET akan muncul secara terbuka di URL, sehingga dapat terlihat oleh user atau tersimpan dalam log server. Beberapa karakter khusus seperti spasi akan diubah menjadi tanda "+" atau "%20"

Contoh dari method GET :

index.php

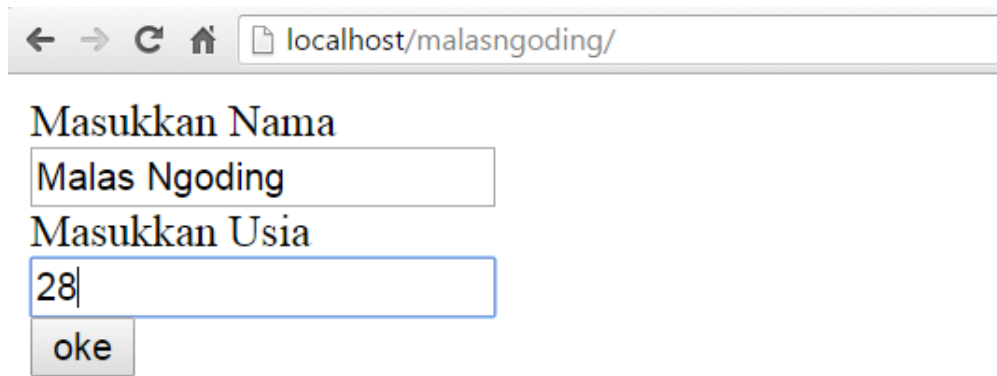
```
<!-- penanganan form dengan method GET -->
<form method="get" action="tampil.php">
    <label>Masukkan Nama</label><br/>
    <input type="text" name="nama"><br/>
    <label>Masukkan Usia</label><br/>
    <input type="text" name="usia"><br/>
    <input type="submit" value="oke">
</form>
```

tampil.php

```
<?php
// menangkap data nama dengan method nama
$nama = $_GET['nama'];
// menangkap data usia dengan method nama
$usia = $_GET['usia'];

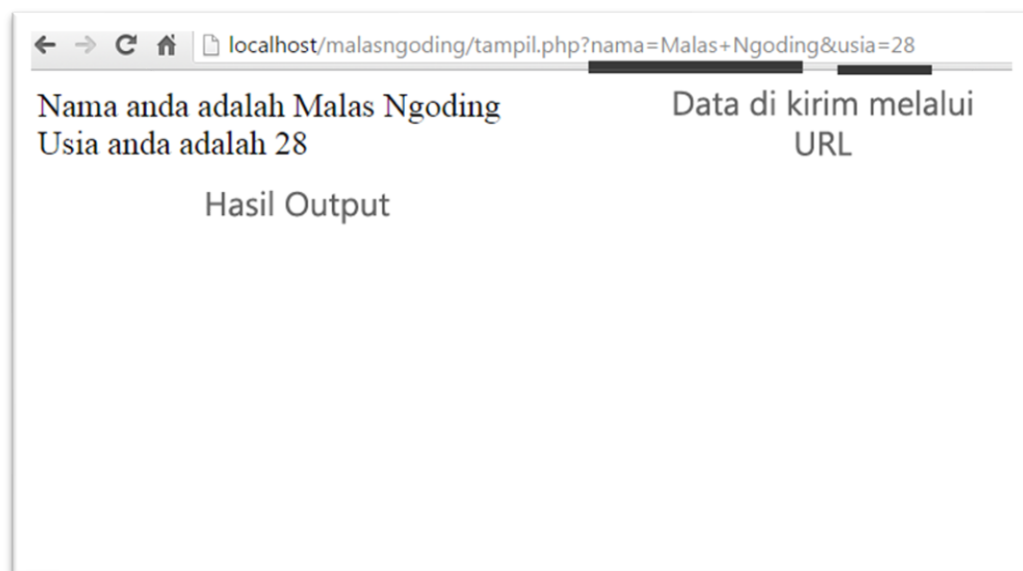
// menampilkan data nama
echo "Nama anda adalah " . $nama;
echo "<br/>";
// menampilkan data usia
echo "Usia anda adalah " . $usia;
?>
```

kemudian jalankan index.php



A screenshot of a web browser window. The address bar shows 'localhost/malasngoding/'. The page content includes the text 'Masukkan Nama' followed by a text input field containing 'Malas Ngoding'. Below this is the text 'Masukkan Usia' followed by a text input field containing '28'. At the bottom of the form is a button labeled 'oke'.

dan output yang di hasilkan



A screenshot of a web browser window showing the output of a POST request. The address bar shows 'localhost/malasngoding/tampil.php?nama=Malas+Ngoding&usia=28'. The page content displays the text 'Nama anda adalah Malas Ngoding' and 'Usia anda adalah 28' on the left, and 'Data di kirim melalui URL' on the right. Below this, the text 'Hasil Output' is centered.

2. POST

Metode POST digunakan untuk mengirimkan data secara tersembunyi melalui permintaan HTTP. Data yang dikirimkan dengan metode POST tidak akan muncul di URL, sehingga lebih aman dan lebih cocok untuk mengirim data sensitif seperti kata sandi atau informasi pribadi. Data dikirimkan sebagai bagian dari badan permintaan HTTP dan tidak terlihat oleh pengguna atau tersimpan dalam log server.

Contohnya :

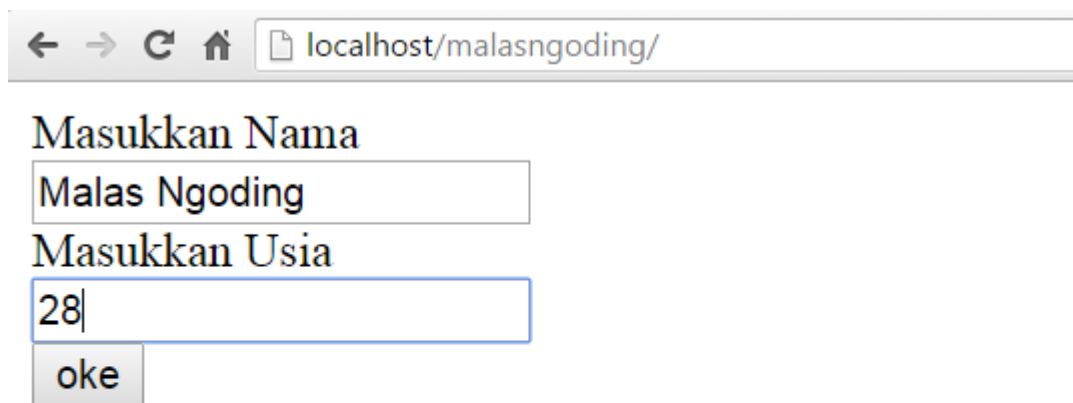
index.php

```
<!-- penanganan form dengan method POST -->
<form method="post" action="tampil.php">
    <label>Masukkan Nama</label><br/>
    <input type="text" name="nama"><br/>
    <label>Masukkan Usia</label><br/>
    <input type="text" name="usia"><br/>
    <input type="submit" value="oke">
</form>
```

tampil.php

```
<?php
// menangkap data nama dengan method post
$nama = $_POST['nama'];
// menangkap data usia dengan method post
$usia = $_POST['usia'];

// menampilkan data nama
echo "Nama anda adalah " . $nama;
echo "<br/>";
// menampilkan data usia
echo "Usia anda adalah " . $usia;
?>
```



A screenshot of a web browser window. The address bar shows 'localhost/malasngoding/'. The page content includes the text 'Masukkan Nama' followed by a text input field containing 'Malas Ngoding'. Below that is the text 'Masukkan Usia' followed by a text input field containing '28'. At the bottom of the form is a button labeled 'oke'.

Hasilnya

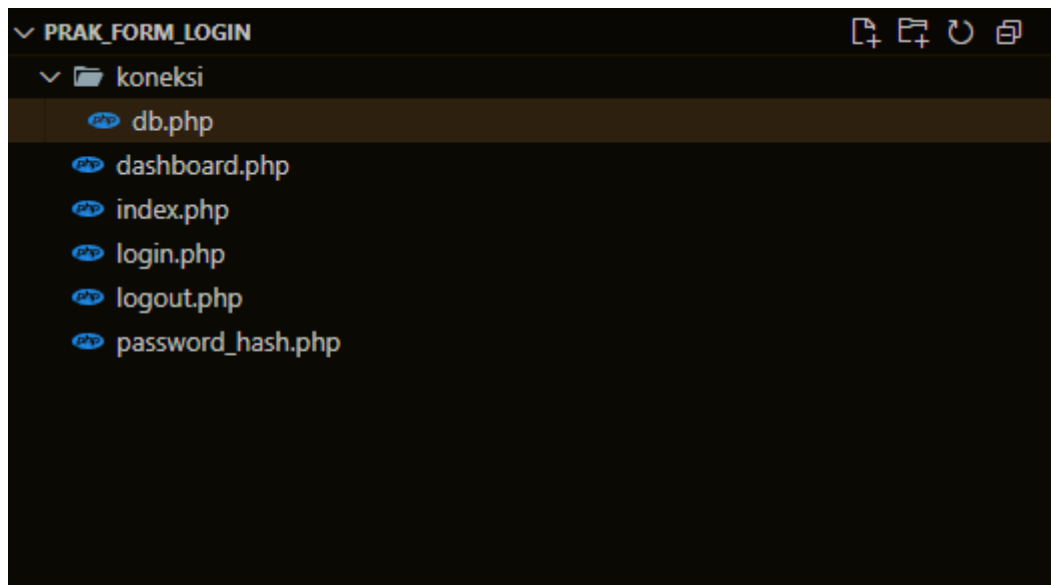


A screenshot of a web browser window. The address bar shows 'localhost/malasngoding/tampil.php'. The page content displays the result of the form submission: 'Nama anda adalah Malas Ngoding' and 'Usia anda adalah 28'.

Latihan Praktikum

Kita akan membuat bagaimana cara login bekerja pada php native.
Berikut Langkah-langkahnya :

1. Struktur Folder



2. Buat file koneksi ke database pada koneksi/db.php

```
<?php

$host = "localhost";
$user = "root";
$pass = "";
$db   = "latihan_login_db";

$conn = mysqli_connect($host, $user, $pass, $db);

if (!$conn) {
    die("Koneksi gagal: " . mysqli_connect_error());
}
?>
```

Setelah itu buatkan table user dengan SQL sebagai berikut :

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL
);
```

- Setelah itu, generate password_hash untuk dimasukkan ke dalam database

password_hash.php

```
<?php
// misalkan password disini menggunakan 12345
$password_hash = password_hash("12345", PASSWORD_DEFAULT);
echo $password_hash;
```

Lalu jalankan di terminal kalian php password_hash.php

```
PS D:\laragon\www\Prak_Form_Login> php "d:\laragon\www\Prak_Form_Login\password_hash.php"
$2y$10$4.ht5LuZG1W9srXqRAMad.1CqCDo5XANS1kdWIdtJQ5xR5G2ENL06
```

Setelah itu, masukan username dan password_hash yang kita dapatkan ke dalam database.

☐ Profiling
 [\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PHP code \]](#)
[\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

	id	username	password
<input type="checkbox"/> Edit Copy Delete	1	aryadzar	\$2y\$10\$4.ht5LuZG1W9srxqRAMad.ICqCD05XANSIkdWIdtJQ5...

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

4. Buat file index.php dengan form login

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
  <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap
.min.css" rel="stylesheet">
</head>
<body class="bg-light">

<div class="container mt-5">
  <div class="row justify-content-center">
    <div class="col-md-4">
      <div class="card shadow-sm rounded-4">
        <div class="card-body">
          <h3 class="text-center mb-4">Login</h3>
          <form method="POST" action="login.php">
            <div class="mb-3">
              <label for="username" class="form-
label">Username</label>
              <input type="text" name="username"
class="form-control" id="username" required>
            </div>
            <div class="mb-3">
              <label for="password" class="form-
label">Password</label>
              <input type="password" name="password"
class="form-control" id="password" required>
            </div>
            <button type="submit" class="btn btn-primary
w-100">Login</button>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>

</body>
</html>
```

5. Buat file login.php

```
<?php
session_start();
include 'koneksi/db.php';

$username = $_POST['username'];
$password = $_POST['password'];

$query = "SELECT * FROM users WHERE username='$username'";
$result = mysqli_query($conn, $query);
$user = mysqli_fetch_assoc($result);

if ($user && password_verify($password, $user['password'])) {
    $_SESSION['user'] = $user['username'];
    header("Location: dashboard.php");
} else {
    echo "Login gagal. Username atau password salah.";
}
?>
```

- **session_start() :**
Digunakan untuk memulai session. Session ini penting agar kita bisa menyimpan data login (seperti username) ke dalam memori server dan digunakan antar halaman. Misalnya, kalau login berhasil, kita akan menyimpan username ke dalam session.
- **include 'koneksi/db.php';:**
Baris ini menyisipkan file koneksi/db.php yang berisi koneksi ke database MySQL.
Jadi, kita tidak perlu menulis ulang konfigurasi koneksi di file ini.
- **Ambil data dari form login (method POST):**
 - `$_POST['username']` : Mengambil data dari input form `name="username"`.
 - `$_POST['password']` : Mengambil data dari input form `name="password"`.

Catatan: Di tahap lanjut, kita sebaiknya juga melakukan validasi input atau filter input agar lebih aman (contohnya `htmlspecialchars()` atau validasi string).

- **Pengecekan:**

- `if ($user)`: Cek apakah username ditemukan di database.
- `password_verify()`: Fungsi bawaan PHP untuk membandingkan password yang diinput user (`$password`) dengan password yang sudah di-hash di database (`$user['password']`).

Ini adalah cara aman memverifikasi password karena kita tidak membandingkan password secara langsung (plaintext), tetapi lewat hash yang aman.

- **Jika login berhasil:**

- Simpan username ke dalam session (`$_SESSION['user']`) agar bisa digunakan di halaman lain (misalnya `dashboard.php`).
- Redirect ke halaman dashboard menggunakan `header()`.

6. Buat file dashboard.php

```
<?php
session_start();
if (!isset($_SESSION['user'])) {
    header("Location: index.php");
    exit;
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Dashboard</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap
.min.css" rel="stylesheet">
</head>
<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
        <a class="navbar-brand" href="#">MyApp</a>
        <div class="d-flex">
            <a href="logout.php" class="btn btn-outline-light">Logout</a>
        </div>
    </div>
```

```

    </div>
</nav>

<div class="container mt-5">
    <div class="alert alert-success rounded-4 shadow-sm"
    role="alert">
        <h4 class="alert-heading">Selamat Datang, <?php echo
        htmlspecialchars($_SESSION['user']); ?>!</h4>
        <p>Kamu berhasil login ke dalam sistem. Ini adalah halaman
        dashboard sederhana.</p>
    </div>
</div>

</body>
</html>

```

7. Buat file logout.php

```

<?php
session_start();
session_destroy();
header("Location: index.php");

```

- **session_destroy()**

Baris ini berfungsi untuk **menghapus seluruh data session yang tersimpan**, termasuk data login seperti `$_SESSION['user']`.

Ini digunakan saat user ingin **logout** dari sistem. Setelah dijalankan, semua informasi yang tersimpan di session akan hilang, artinya user dianggap sudah keluar dari aplikasi

- **header("Location: index.php");**

Redirect ke halaman login

Setelah session dihapus, user langsung diarahkan ke `index.php` (halaman login).

Ini membuat pengalaman pengguna menjadi jelas: setelah logout, dia langsung kembali ke halaman untuk login lagi.

Silahkan Dicoba loginnya dengan menggunakan database

Tugas

1. Dari program di atas, Ketika user salah memasukkan password tidak Kembali ke halaman login, buat agar Kembali ke halaman login !
2. Dari program di atas, Ketika user sudah login ke dashboard, masih bisa ke halaman login, silahkan perbaiki agar redirect ke dashboard lagi !

Form pengumpulan akan diumumkan nanti, selamat mengerjakan

Refrensi

<https://pilarsolusi.co.id/pemrograman/perbedaan-antara-metode-get-dan-post-dalam-pemrograman-php/>

<https://www.malasngoding.com/belajar-php-penanganan-form-pada-php/>