

**LAPORAN PRAKTIKUM  
PENGEMBANGAN WEB DAN MOBILE**



**NAMA : DZIKRI AHMADILLAH  
NIM : 193020503039  
KELAS : A  
MODUL : II (Form Handling)**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PALANGKARAYA  
2021**

# BAB I

## TUJUAN DAN LANDASAN TEORI

### 1.1 Tujuan

1. Mahasiswa mampu membuat handling yang mampu mengolah data dari form HTML.
2. Mahasiswa mampu membuat batasan-batasan untuk menangani inputan dari form HTML.

### 1.2 Landasan Teori

#### 1.2.1 HTML

HTML adalah singkatan dari Hypertext Markup Language. HTML memungkinkan seorang user untuk membuat dan menyusun bagian paragraf, heading, link atau tautan, dan blockquote untuk halaman web dan aplikasi.

HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan user untuk mengorganisir dan memformat dokumen, sama seperti Microsoft Word

#### 1.2.2 HTML Form

HTML Form adalah cara yang digunakan pada web untuk mengambil inputan dari user. Untuk membuat form pada HTML, digunakan tag `<form>`. Didalamnya akan diisi dengan tag `<input>`. Tag `<input>` dapat ditampilkan dengan beragam cara sesuai dengan kebutuhan input yang diinginkan (*HTML Forms*)

**Tabel 1.1** tipe-tipe dari `<input>`

Tipe	Deskripsi
<code>&lt;input type="text"&gt;</code>	Menampilkan inputan satu baris
<code>&lt;input type="radio"&gt;</code>	Menampilkan tombol radio(digunakan untuk memilih satu opsi diantara banyak pilihan)

<code>&lt;input type="checkbox"&gt;</code>	Menampilkan sebuah checkbox (digunakan untuk memilih 0 atau lebih opsi dari banyak pilihan)
<code>&lt;input type="submit"&gt;</code>	Menampilkan sebuah submit button (untuk mengirim form)
<code>&lt;input type="button"&gt;</code>	menampilkan button yang bisa di klik

Agar komponen-komponen FORM HTML bisa dikenali sebagai variable yang bernilai data, maka semua komponen harus diletakkan diantara tanda `<form>...</form>`. Khusus untuk komponen file upload, TAG HTML FORM harus diubah menjadi `<form enctype="multipart/form-data" >...</form>`.

Struktur penulisan HTML Form adalah sebagai berikut (Suprianto 2008):

```

<form name="nama_form" action = "tujuan.php"> method =
"metode yang digunakan">
    Komponen form
</form>

```

**Gambar 1.1** struktur penulisan <FORM>

### 1.2.3 Method GET dan POST

Method adalah jenis pengiriman variable data, yang terbagi menjadi dua jenis yaitu METHOD=GET dan METHOD=POST.

Dengan method GET, maka nama dan nilai variable akan tampak di address URL browser. Method GET lebih cocok untuk pengiriman variable disertai argumen yang panjang, serta tidak membutuhkan keamanan lebih.

Dengan method POST, maka nama dan nilai variable tidak akan tampak di address URL browser. Method POST lebih cocok untuk pengiriman variable yang membutuhkan pengamanan. Misalnya : untuk halaman web yang memerlukan login user dan password. ('MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya')

### 1.2.4 Htmlelentities

Fungsi htmlelentities adalah fungsi yang digunakan untuk mengkonversikan semua character menjadi entitas HTML. Fungsi ini identik dengan fungsi

`htmlspecialchars()`, dimana semua character yang memiliki kesamaan atau dimiliki oleh character entitas HTML akan diterjemahkan ke entitas tersebut.(Suprianto 2008)

## BAB II PEMBAHASAN

### 2.1 SOAL

Buatlah program web untuk menginputkan username dan password menggunakan form dan penanganan input data dengan kriteria sebagai berikut:

1. username yang diinputkan tidak boleh lebih dari tujuh karakter.
2. password yang diinputkan harus terdiri dari huruf kapital, huruf kecil, angka dan karakter khusus.
3. Jumlah karakter password tidak boleh kurang dari sepuluh karakter.

### 2.2 JAWABAN

Untuk menyelesaikan permasalahan ini saya hanya menggunakan sebuah file. Isi secara garis besar terbagi menjadi dua bagian yaitu bagian input form dan bagian pengolahan data.

#### 2.2.1 Input Form

Pertama adalah membuat form yang diperlukan. Method yang akan digunakan adalah POST. Saya menggunakan post karena form ini akan meminta data username dan password. Pada method POST, nama dan nilai variable tidak akan tampak di address URL browser sehingga pengiriman menjadi lebih aman.

Pada bagian *action*, dikarenakan hanya menggunakan satu file maka, *action* dari form akan mereferensikan ke file itu sendiri. Dilakukan dengan menggunakan variabel global `$_SERVER ["PHP_SELF"]`).

Reference file juga disertai dengan fungsi *htmlspecialchars* digunakan untuk mengkonversikan karakter-karakter spesial ke entitas HTML. Hal ini dilakukan untuk mencegah kemungkinan injeksi dan pencurian data user.

```
<?php>  
<form method="post" action="<?php echo htmlspecialchars($_SERVER ["PHP_SELF"]);?>">  
<div>
```

**Gambar 2.1** Format yang digunakan untuk *form*

Selanjutnya adalah membuat tampilan agar user dapat menginputkan data ke web. Terdapat dua data yang diperlukan yaitu input untuk username dan untuk password kedua input akan menggunakan tipe input *“text”* karena hanya memerlukan satu baris inputan.

Agar isi inputan sudah terisi dengan nilai field yang telah diinput sebelumnya, tiap field inputan diberi nilai value. Untuk melakukannya pertama digunakan fungsi *isset* untuk mengetahui apakah variabel masih bernilai NULL(belum ada inputan) atau tidak. Jika tidak NULL, maka akan diberikan nilai dari variabel yang telah dikonversi balik menggunakan *htmlspecialchars* karena saat melakukan *post* data, char diubah karakter special entitas html sehingga perlu diubah balik ke karakter char normal.

Diakhir ditambahkan inputan untuk mengirim data hasil inputan.

```
<div>
  <div>
    <label for="username">username</label>
  </div>
  <div>
    <input type="text" id="username" name="username" placeholder="Username" value=
      "<?php echo isset($_POST['username']) ? htmlspecialchars($_POST['username'], ENT_QUOTES) :
    </div>
  </div>
```

**Gambar 2.2** Input untuk username

```
<div>
  <div>
    <label for="fname">password</label>
  </div>
  <div>
    <input type="text" id="password" name="password" placeholder="password" value=
      "<?php echo isset($_POST['password']) ? htmlspecialchars($_POST['password'], ENT_QUOTES) :
    </div>
  </div>
```

**Gambar 2.3** Input untuk password

### 2.2.2 Pengolahan data

Hal yang dilakukan selanjutnya adalah mengolah data yang didapat. Kita harus memastikan data yang diminta sesuai dengan yang diinginkan. Namun sebelumnya, saya akan membuat sebuah fungsi untuk menampilkan sebuah notifikasi ke layar agar pengguna tau dimana letak kesalahan yang dilakukannya. Fungsi diberi nama *giveAlert*.

```
function giveAlert($error){
    >>
    <div class="alert">
        <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
        <?php echo $error;>
    </div>
    <?php
}
}
```

**Gambar 2.4** Fungsi yang digunakan untuk menampilkan peringatan

Sebelum dilakukan pengecekan dibuat 4 variabel bantu. 2 variabel yaitu *\$username* dan *\$password* mewakili username dan password. Selanjutnya variabel yang *\$usernameERR* dan *\$passwordERR* yang memiliki nilai awal 0. Variabel tersebut digunakan untuk menyatakan apakah terjadi sebuah kesalahan.

Pengecekan pertama dilakukan untuk menentukan apakah user telah melakukan input ke kolom username dan password. Jika tidak, maka akan diberikan peringatan serta mengubah nilai *\$usernameERR* dan *\$passwordERR* menjadi 1. Pengecekan nilai menggunakan fungsi *empty*.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["username"])) {
        giveAlert("Username masih kosong!!!");
        $usernameERR = 1;
    }
    if (empty($_POST["password"])) {
        giveAlert("Password masih kosong!!!");
        $passwordERR = 1;
    }
}
```

**Gambar 2.5** mengecek apakah user sudah memasukan data

Setelahnya adalah mengecek apakah panjang username tidak lebih dari 7 karakter. Hal ini dilakukan untuk menyesuaikan dengan kondisi satu dari soal. Untuk mengecek panjang inputan username digunakan fungsi *strlen*. Jika lebih dari 7, akan diberi peringatan dan nilai *\$usernameERR* menjadi bernilai 1.

```
}
if (strlen($_POST["username"]) > 7){
    giveAlert("Username tidak boleh lebih dari tujuh karakter!!!");
    $usernameERR = 1;
}
```

**Gambar 2.6** Mengecek panjang username tidak lebih dari 7 karakter

Pengecekan selanjutnya adalah pengecekan apakah password yang diinputkan sudah memiliki huruf kapital, huruf kecil dan angka. sesuai dengan kondisi 2 pada soal. Pengecekan dilakukan menggunakan fungsi *preg\_match* untuk memastikan bahwa string memiliki capital, huruf kecil dan angka. Jika tidak, , akan diberi peringatan dan nilai *\$passwordERR* menjadi bernilai 1.

```
if (!preg_match('^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])^', $_POST["password"]))
{
    giveAlert("password yang diinputkan harus terdiri dari huruf kapital, huruf kecil dan angka!!");
    $passwordERR = 1;
}
```

**Gambar 2.7** Mengecek apakah password terdiri atas huruf kapital, huruf kecil dan angka

Selanjutnya mengecek apakah password minimal terdiri dari 10 karakter. Jika tidak, maka akan diberi peringatan dan nilai *\$passwordERR* menjadi bernilai 1. Sama seperti sebelumnya, pengecekan menggunakan

```
if (strlen($_POST["password"]) < 10){
    giveAlert("password harus lebih dari 10 karakter!!!");
    $passwordERR = 1;
}
```

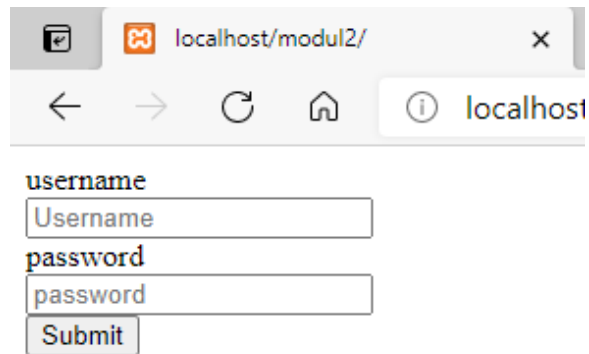
**Gambar 2.8** Mengecek panjang username minimal 10 karakter

Terakhir adalah notifikasi yang ditampilkan ketika submit berhasil. Jika tidak kesalahan yaitu ketika nilai variabel *\$usernameERR* dan *\$passwordERR* bernilai 0, maka akan ditampilkan notifikasi bahwa submit berhasil

```
//jika semua benar
if($usernameERR == 0 && $passwordERR == 0) {
    ?>
    <div class="alert2">
        <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
        submit data berhasil
    </div>
<?php
}
```

**Gambar 2.9** Menampilkan notifikasi submit data berhasil





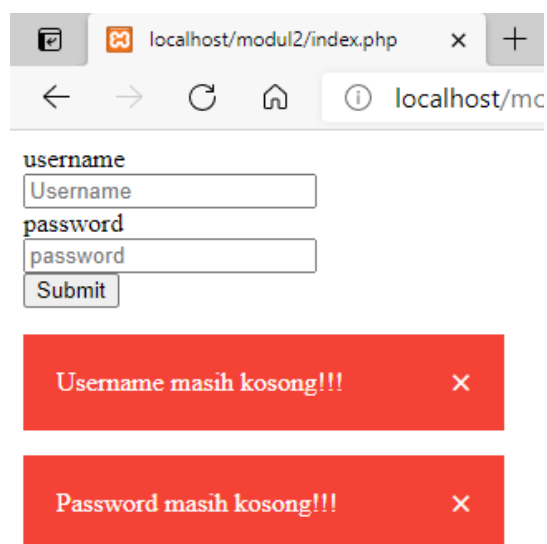
localhost/modul2/

username  
Username

password  
password

Submit

**Gambar 2.10** kondisi awal



localhost/modul2/index.php

username  
Username

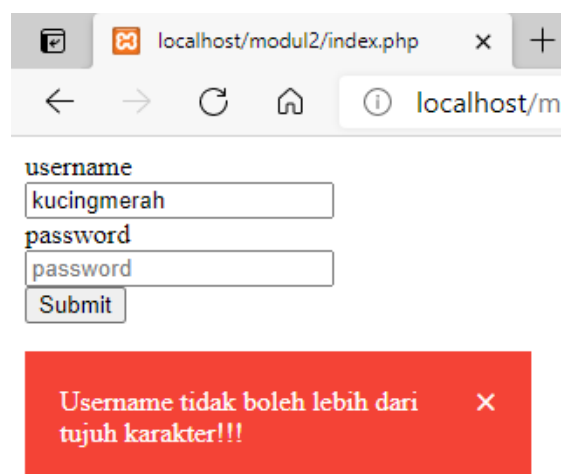
password  
password

Submit

Username masih kosong!!!

Password masih kosong!!!

**Gambar 2.11** Ketika submit tanpa mengisi data



localhost/modul2/index.php

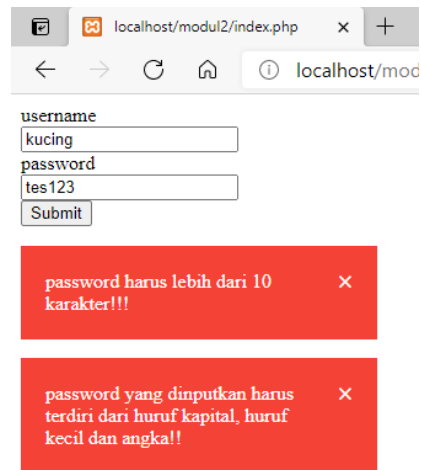
username  
kucingmerah

password  
password

Submit

Username tidak boleh lebih dari tujuh karakter!!!

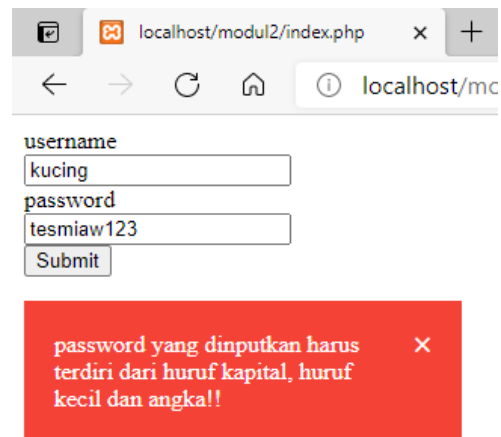
**Gambar 2.12** Username melebihi batas



A screenshot of a web browser window with the address bar showing 'localhost/modul2/index.php'. The page contains a login form with two input fields: 'username' with the value 'kucing' and 'password' with the value 'tes123'. Below the fields is a 'Submit' button. Two red error messages are displayed below the form:

- password harus lebih dari 10 karakter!!!
- password yang diinputkan harus terdiri dari huruf kapital, huruf kecil dan angka!!

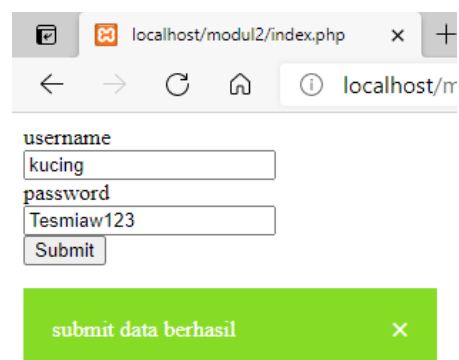
**Gambar 2.13** Password belum sesuai kriteria



A screenshot of a web browser window with the address bar showing 'localhost/modul2/index.php'. The page contains a login form with two input fields: 'username' with the value 'kucing' and 'password' with the value 'tesmiaw123'. Below the fields is a 'Submit' button. One red error message is displayed below the form:

- password yang diinputkan harus terdiri dari huruf kapital, huruf kecil dan angka!!

**Gambar 2.14** password tidak memiliki kapital



A screenshot of a web browser window with the address bar showing 'localhost/modul2/index.php'. The page contains a login form with two input fields: 'username' with the value 'kucing' and 'password' with the value 'Tesmiaw123'. Below the fields is a 'Submit' button. A green success message is displayed below the form:

- submit data berhasil

**Gambar 2.15** username dan password sudah sesuai kriteria

### **BAB III**

### **KESIMPULAN**

Form adalah cara yang dilakukan pada web untuk mendapatkan data dari inputan user. Kita harus berhati-hati dalam memilih method apa yang digunakan ketika menggunakan form. Method POST lebih aman daripada get karena nilai variabel tidak Nampak di url. Hal ini berguna untuk input data password dan semacamnya. Namun hal ini juga menyebabkan alamat url tidak bisa di bookmark. Sedangkan GET menampilkan variabel di url sehingga alamat dapat di bookmark. GET berguna untuk input pencarian.


## DAFTAR PUSTAKA

“HTML Forms.” [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp) (April 4, 2021).

“MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika  
Fakultas Teknik Universitas Palangka Raya.”

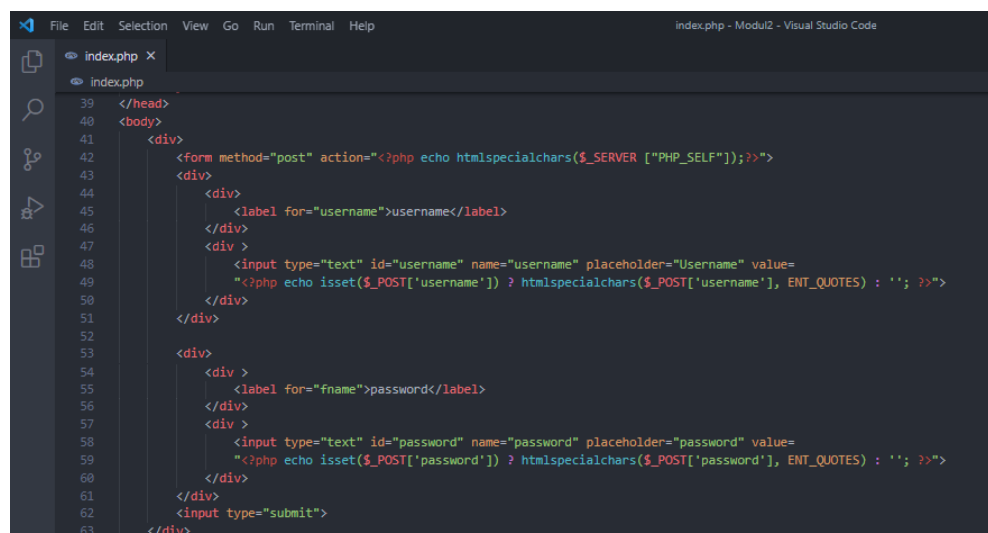
Suprianto, Dodit. 2008. “Buku Pintar Pemrograman Php.” *Bandung: OASE Media*.

## LAMPIRAN



```
1 <html>
2 <head>
3
4 <style>
5     .alert {
6         padding: 20px;
7         background-color: #f44336; /* merah */
8         color: white;
9         margin-bottom: 15px;
10        width:250px;
11        max-width:250px;
12    }
13    .alert2{
14        padding: 20px;
15        background-color: #860c24; /* hijau */
16        color: white;
17        margin-bottom: 15px;
18        width:250px;
19        max-width:250px;
20    }
21
22    /* The close button */
23    .closebtn {
24        margin-left: 15px;
25        color: white;
26        font-weight: bold;
27        float: right;
28        font-size: 22px;
29        line-height: 20px;
30        cursor: pointer;
31        transition: 0.3s;
32    }
33
34    /* When moving the mouse over the close button */
35    .closebtn:hover {
36        color: black;
37    }
38 </style>
```

Source code pada head berisi style untuk alert



```
39 </head>
40 <body>
41 <div>
42     <form method="post" action="<?php echo htmlspecialchars($_SERVER ["PHP_SELF"]);>">
43     <div>
44         <div>
45             <label for="username">username</label>
46         </div>
47         <div>
48             <input type="text" id="username" name="username" placeholder="Username" value=
49                 "<?php echo isset($_POST['username']) ? htmlspecialchars($_POST['username'], ENT_QUOTES) : '' ; >">
50         </div>
51     </div>
52
53     <div>
54         <div>
55             <label for="fname">password</label>
56         </div>
57         <div>
58             <input type="text" id="password" name="password" placeholder="password" value=
59                 "<?php echo isset($_POST['password']) ? htmlspecialchars($_POST['password'], ENT_QUOTES) : '' ; >">
60         </div>
61     </div>
62     <input type="submit">
63 </div>
```

Source code untuk form

```
index.php X
index.php
64
65 <?php
66     $username = $password = "";
67     $usernameERR = $passwordERR = 0;
68
69     if ($_SERVER["REQUEST_METHOD"] == "POST") {
70         if (empty($_POST["username"])) {
71             giveAlert("Username masih kosong!!!");
72             $usernameERR = 1;
73         }
74         if (empty($_POST["password"])) {
75             giveAlert("Password masih kosong!!!");
76             $passwordERR = 1;
77         }
78         if (strlen($_POST["username"]) > 7){
79             giveAlert("Username tidak boleh lebih dari tujuh karakter!!!");
80             $usernameERR = 1;
81         }
82
83         if (strlen($_POST["password"]) < 10){
84             giveAlert("password harus lebih dari 10 karakter!!!");
85             $passwordERR = 1;
86         }
87
88         if (!preg_match('^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])$', $_POST["password"]))
89         {
90             giveAlert("password yang diinputkan harus terdiri dari huruf kapital, huruf kecil dan angka!!!");
91             $passwordERR = 1;
92         }
93
94         //jika semua benar
95         if($usernameERR == 0 && $passwordERR == 0) {
96             >>
97             <div class="alert2">
98                 <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
99                 submit data berhasil
100             </div>
101         <?php
102     }
```

Percabangan sesuai dengan inputan data

```
index.php - Modul2 - Visual Studio Code
index.php
102     }
103
104     $username = $_POST["username"];
105     $password = $_POST["password"];
106 }
107
108 function giveAlert($error){
109     >>
110     <div class="alert">
111         <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
112         <?php echo $error;>
113     </div>
114 <?php
115 }
116
117 >>
118
119
120
121
122 </body>
123 </html>
```

Fungsi untuk menampilkan alert