

**LAPORAN HASIL PRAKTIKUM**  
**PEMROGRAMAN WEB & MOBILE**



**Nama** : Tirza Yolanda Hevin  
**NIM** : 193030503049  
**Kelas** : A  
**Modul** : II ( Form Handling )

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS PALANGKA RAYA**  
**2021**

# BAB I

## LANDASAN TEORI

### A. Tujuan Praktikum

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

### B. Dasar Teori

Variabel superglobal PHP `$_GET` dan `$_POST` digunakan untuk mengumpulkan data-form. Contoh berikut menunjukkan form HTML sederhana dengan dua field input dan tombol submit:

```
<html>
  <body>
    <form action="welcome.php"
      method="post"> Name:
      <input type="text"
        name="name"><br>E-
      mail: <input type="text"
        name="email"><br>
      <input type="submit">
    </form>
```

Ketika user mengisi form, dan menekan tombol click, data form dikirim untuk memproses file PHP dengan nama “welcome.php”. Data form dikirimkan dengan method HTTP POST. Untuk menampilkan data yang sudah disubmit bisa dilakukan dengan mencetak data tersebut menggunakan perintah echo. File “welcome.php” adalah sebagai berikut:

```
<html>
  <body>
    Welcome <?php echo $_POST["name"];
    ?><br> Your email address is: <?php
    echo $_POST["email"];
    ?> </body>
```

Jika field nama diinputkan dengan Tono dan email diinputkan dengan [tono@mail.com](mailto:tono@mail.com) maka output yang akan tampil adalah sebagai berikut:

Welcome Budi

Your email address is [tono@mail.com](mailto:tono@mail.com)

Hasil yang sama juga akan tampil dengan menggunakan method get sebagai berikut:

```
<html>
  <body>

    <form action="welcome_get.php"
      method="get"> Name: <input
      type="text" name="name"><br>
      E-mail: <input type="text"
      name="email"><br>
      <input type="submit">
    </form>
```

dengan file “welcome\_get.php” sebagai berikut:

```
<html>
  <body>
    Welcome <?php echo $_GET["name"];
    ?><br> Your email address is: <?php
    echo $_GET["email"];
    ?> </body>
```

## GET vs. POST

GET dan POST membuat sebuah array (contoh array(kunci => nilai, kunci2 => nilai2, kunci3 => nilai3, ...)). Array ini menyimpan pasangan kunci/nilai, dimana kunci- kunci adalah nama-nama dari form control dan nilai-nilai adalah data input dari user. Method GET diakses menggunakan `$_GET` dan method POST diakses menggunakan

`$_POST`. Kedua variabel ini adalah variabel superglobal, yang selalu bisa diakses, tanpa memperhatikan lingkup dan bisa diakses dari fungsi, class atau file yang berbeda tanpa harus melakukan teknik khusus. `$_GET` adalah sebuah array dari variabel yang dikirimkan ke skrip melalui parameter URL. `$_POST` adalah sebuah array dari variabel yang dikirimkan ke skrip melalui method HTTP POST.

### Kapan sebaiknya menggunakan GET?

Informasi dikirim dari sebuah form dengan method GET bisa dilihat oleh semua orang (semua nama dan nilai variabel ditampilkan di URL). GET juga memiliki batas pada jumlah informasi yang dikirim. Batasannya adalah sekitar 2000 karakter. Namun, karena variabel ditunjukkan di URL, ia memungkinkan untuk dilakukan bookmark halaman. Dalam beberapa kasus, hal ini sangat bermanfaat. GET bisa digunakan untuk mengirimkan data yang tidak sensitif.

**Ingat!** GET tidak boleh digunakan untuk mengirimkan password atau informasi sensitif lainnya!

### Kapan menggunakan POST?

Informasi yang dikirim dari sebuah form dengan method POST tidak bisa dilihat oleh siapapun (semua nama-nama atau nilai-nilai tertanam didalam body request HTTP) dan tidak memiliki batasan jumlah informasi yang akan dikirim. POST juga mendukung fungsionalitas lanjutan seperti dukungan untuk input biner multi-part ketika sedang melakukan upload file ke server. Namun, karena variabel tidak ditampilkan di URL, tidak mungkin untuk dilakukan bookmark halaman

(data tidak ter-bookmark). Developer lebih baik menggunakan POST untuk mengirimkan data form.

## Validasi Form PHP

Pertimbangkan keamanan ketika memproses form PHP!

### PHP Form Validation Example

\* required field.

Name:  \*

E-mail:  \*

Website:

Comment:

Gender: ☐ Female ☐ Male \*

Form HTML yang akan kita gunakan pada modul ini, mengandung bermacam- macam field input, misalnya text field yang harus diisi dan text field yang opsional, tombolpilihan (radio button), dan tombol submit. Rule atau aturan validasi untuk form diatas adalah sebagai berikut:

Field	Rule Validasi
Name	Dibutuhkan. + Harus hanya mengandung huruf dan spasi
E-mail	Dibutuhkan. + Harus mengandung sebuah alamat email yang valid dengan @ dan .
Website	Opsional. Jika ada, harus mengandung URL yang valid.
Comment	Opsional. Field input multi-line (text area).

Gender	Dibutuhkan. Harus memilih salah satu
--------	--------------------------------------

Kode HTML untuk membentuk Form tersebut adalah sebagai berikut

### *Text Field*

Field nama, email dan website adalah elemen-elemen text input, dan field komentar adalah textarea yaitu sebagai berikut:

Name: <input

type="text"

name="name">E-

mail: <input

type="text"

name="email">

Website: <input type="text" name="website">

Comment: <textarea name="comment" rows="5" cols="40"></textarea>

### *Radio Button*

Field jenis kelamin adalah radio button yaitu sebagai berikut:

Gender:

<input type="radio" name="gender" value="female">Female

<input type="radio" name="gender" value="male">Male

### *Form Element*

Kode HTML untuk membentuk form pada gambar diatas adalah sebagai berikut:

<form method="post"

action="<?php echo

```
htmlspecialchars($_SERVER  
["PHP_SELF"]);?>">
```

Ketika form disubmit, data pada form dikirim dengan method “post”.

`$_SERVER["PHP_SELF"]` adalah variabel super global yang mengembalikan nama file dari skrip yang sedang dieksekusi. Sehingga kode form diatas mengirim data pada form ke halaman itu sendiri. Sedangkan fungsi **htmlspecialchars()** adalah fungsi yang mengkonversikan karakter-karakter spesial ke entitas HTML. Sebagai contoh, fungsitersebut akan mengkonversikan karakter < dan > menjadi &lt; dan &gt;. Fungsi ini mencegah injeksi yang bisa dilakukan dengan HTML atau javascript (Cross-site Scripting Attack) pada form tersebut.

### **Catatan Penting pada Keamanan Form PHP**

Variabel `$_SERVER["PHP_SELF"]` bisa digunakan oleh hacker! Jika `PHP_SELF` digunakan pada halaman web, user bisa memasukkan skrip dengan terlebih dahulu memasukkan garis miring (/) kemudian beberapa perintah Cross Site Scripting (XSS) untuk dieksekusi. XSS adalah tipe kelemahan keamanan komputer yang secara tipikal ditemukan dalam aplikasi web.

Asumsikan kita memiliki halaman web dengan nama “test\_form.php”, dan form hanya kita deklarasikan sebagai berikut:

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

Kemudian user memasukkan URL pada address bar dengan alamat sebagai berikut:

[http://localhost/<nama\\_folder>/test\\_form.php/%22%3E%3Cscript%3Ealert\('hacked'\)%3C/script%3E](http://localhost/<nama_folder>/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E)

yang jika ditranslasikan akan menjadi:

```
<form method="post" action="test_form.php/"><script>alert('hacked')</script>
```

Kode ini menambah tag script dan perintah alert atau peringatan, ketika halaman dibuka, kode javascript tersebut akan dieksekusi, maka user akan melihat kotak peringatan dengan tulisan “hacked”.

**Berhati-hatilah dengan kemungkinan penambahan kode javascript pada tag <script>!**

Hacker bisa mengarahkan user ke file pada server yang lain, dan file itu bisa mengandung kode yang bisa merubah variabel global atau melakukan submit form pada alamat web yang berbeda untuk mencuri data user.

Bagaimana menghindari penyalahgunaan \$\_SERVER["PHP\_SELF"]?

Caranya adalah dengan menggunakan fungsi htmlspecialchars(). Fungsi tersebut akan mengkonversikan karakter khusus ke entitas HTML. Ketika user memasukkan URL dengan tag script seperti contoh sebelumnya, maka akan ditranslasikan sebagai berikut:

```
<form method="post" action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

dengan cara ini, percobaan penyalahgunaan akan gagal.

### *Memvalidasi data Form dengan PHP*

Hal pertama yang akan kita lakukan adalah memasukkan semua variabel melalui fungsi htmlspecialchars(). Kemudian ada juga dua hal ketika user melakukan submit form:

1. Membuang karakter-karakter yang tidak dibutuhkan (seperti spasi extra,



tab extra, dan baris baru yang ekstra) dari data input user (dengan fungsi trim()).

2. Membuang backslash (\) satu garis miring dari data input user (dengan fungsi stripslashes()).

Langkah berikutnya adalah membuat fungsi yang akan melakukan pemeriksaan kebenaran data yang diinputkan oleh user. Contohnya adalah sebagai berikut:

```
<?php

// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

function
test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
}
```

```
        ata);return  
        $data;  
  
    }  
?>
```

Ingat bahwa pada permulaan skrip, adalah pemeriksaan apakah form sudah disubmit menggunakan `$_SERVER["REQUEST_METHOD"]`. Jika `REQUEST_METHOD` adalah `POST`, maka form telah disubmit dan seharusnya tervalidasi. Jika belum tersubmit, lewati langkah validasi dan tampilkan form kosong. Namun pada contoh diatas semua field input adalah opsional. Skrip bekerja baik bahkan jika user tidak melakukan entri data.

### *Field yang Dibutuhkan*

Kode program berikut terdapat tambahan variabel baru yaitu: `$nameErr`, `$emailErr`, `$genderErr`. Variabel-variabel error ini akan menangani pesan error untuk field yang dibutuhkan. Percabangan dengan `if else` juga akan ditambahkan untuk setiap variabel `$_POST`. Fungsinya untuk memeriksa apakah variabel `$_POST` kosong, hal ini dilakukan dengan menggunakan fungsi `empty()`. Jika kosong, maka pesan error disimpan dalam variabel error yang berbeda, dan jika tidak kosong, ia akan mengirim data input user melalui fungsi `test_input()`:

```
<?php  
  
    // define variables and set to empty values  
  
    $nameErr = $emailErr = $genderErr = $websiteErr = "";  
  
    $name = $email = $gender = $comment = $website = "";  
  
    if ($_SERVER["REQUEST_METHOD"] == "POST")
```

```

{ if (empty($_POST["name"])) {
    $nameErr = "Name is required";
} else {
    $name = test_input($_POST["name"]);
}

if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = test_input($_POST["email"]);
}

if (empty($_POST["website"])) {
    $website = "";

```

```

} else {
    $website =
    test_input($_POST["website"]);
}

if (empty($_POST["comment"])) {
    $comment = "";
} else {
    $comment =
    test_input($_POST["comment"]);
}

if (empty($_POST["sender"])) {

```

?>

Setelah kode diatas ditambahkan, beberapa skrip ditambahkan pada setiap field yang dibutuhkan pada form, fungsinya untuk menampilkan

pesan error jika field yang dibutuhkan tidak diisi. Form HTMLnya adalah sebagai berikut:

```
<form method="post" action="<?php echo  
htmlspecialchars($_SERVER["PHP_SELF"]);?>">  
  
    Name: <input type="text" name="name">  
    <span class="error">* <?php echo  
    $nameErr;?></sp  
    an> <br><br>E-  
    mail:  
    <input type="text" name="email">  
    <span class="error">* <?php echo $emailErr;?></span>  
    <  
    b  
    r  
    >  
    <  
    b  
    r  
    >
```

```
    <input type="radio" name="gender" value="male">Male  
    <span class="error">* <?php echo $genderErr;?></span>  
    <br><br>  
    <input type="submit" name="submit" value="Submit">  
  
</form>
```

### Validasi Nama

Kode berikut menunjukkan cara sederhana untuk memeriksa apakah field nama hanya mengandung huruf dan spasi. Jika nilai dari nama tidak valid, maka pesan error akan disimpan didalam variabel \$nameErr:

```
$name = test_input($_POST["name"]);  
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {  
    $nameErr = "Only letters and white space allowed";  
}
```

Fungsi `preg_match()` mencari string berdasarkan pola, mengembalikan nilai `true` jika polanya ada, `false` jika polanya tidak ada.

### *Validasi Email*

Cara paling mudah dan paling aman untuk memeriksa apakah sebuah alamat email memiliki pola yang sesuai adalah dengan menggunakan fungsi `filter_var()`. Kode dibawah memeriksa apakah alamat email yang dimasukkan menggunakan pola yang sesuai atau tidak, jika tidak, maka pesan error akan disimpan kedalam variabel

`$emailErr`:

```
$email = test_input($_POST["email"]);  
if (!filter_var($email, FILTER_VALIDATE_EMAIL))  
    { $emailErr = "Invalid email format";  
}
```

### *Validasi URL*

Kode program berikut menunjukkan cara untuk memeriksa apakah sintaks alamat URL valid atau tidak. Ekspresi reguler ini mengizinkan keberadaan tanda pisah pada URL. Jika sintaks alamat URL tidak valid, maka pesan error akan disimpan kedalam variabel `$websiteErr`:

```

$website = test_input($_POST["website"]);
if (!preg_match("/^b(?:(:https?|ftp):\\\/|www\\.)*[-a-z0-9+&@#\/%?~_!:,.;]*[-a-z0-9+&@#\/%?~_]/i",$website)) {
    $websiteErr = "Invalid URL";
}

```

Biasanya, jika user salah menginputkan nilai, maka halaman yang tampil adalah halaman yang sama dengan field yang sudah terisi dengan nilai field yang sudah diinput sebelumnya. Untuk menunjukkan nilai dalam field input setelah user menekan tombol submit, ada beberapa skrip PHP yang perlu ditambahkan didalam atribut value pada field input name, email, dan website. Khusus untuk field textarea, akan skrip tersebut akan ditambahkan antara tag <textarea> dan tag </textarea>. Skrip yang singkat akan mengeluarkan nilai dari variabel \$name, \$email, \$website dan \$comment. Untuk radio button atau tombol radio, akan ditambahkan kode yang membuat salah satu pilihan terpilih.

```

Name: <input type="text" name="name" value="<?php echo
$name;?>">E-mail: <input type="text" name="email"
value="<?php echo $email;?>">

Website: <input type="text" name="website" value="<?php echo $website;?>">

Comment: <textarea name="comment" rows="5" cols="40"><?php echo
$comment;? ></textarea>

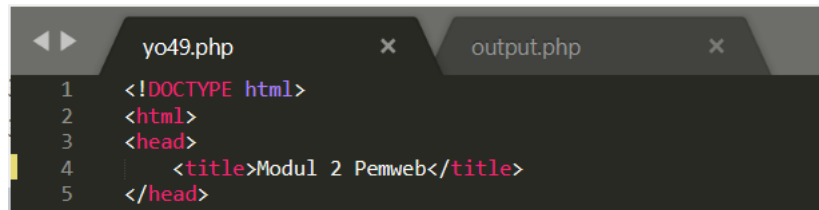
Gender:
<input type="radio" name="gender"
<?php if (isset($gender) &&
$gender=="female") echo"checked";?>
value="female">Female

```

## BAB II

### PEMBAHASAN

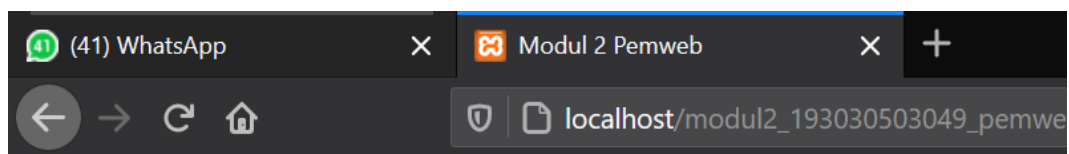
#### A. Tugas Praktikum



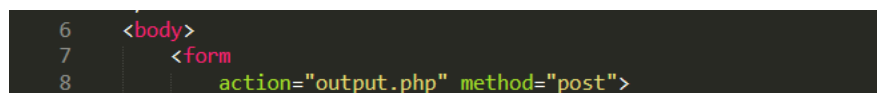
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Modul 2 Pemweb</title>
5 </head>
```

Gambar 2.1 Header

Pada gambar di atas merupakan bagian header untuk membuat judul website yang akan dibuat. Hasil dari header di atas adalah seperti pada Gambar 2.2 di bawah ini.



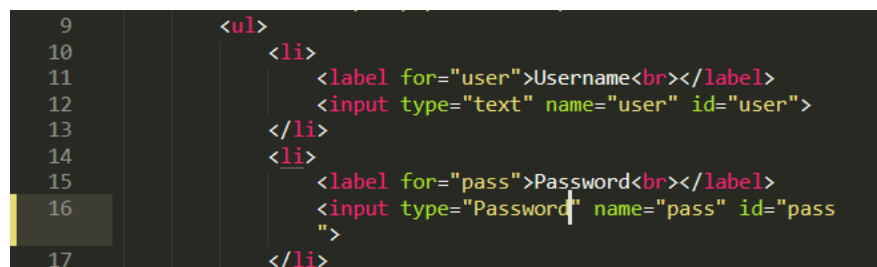
Gambar 2.2 Title Website



```
6 <body>
7   <form
8     action="output.php" method="post">
```

Gambar 2.3 Body Program

Pada gambar di atas terdapat action yang berfungsi untuk menentukan tindakan yang akan dilakukan selanjutnya setelah formulir dikirimkan, action akan mengarahkan formulir ke laman selanjutnya sesuai yang diinginkan dimana di sini akan dilanjutkan ke program output.php.



```
9   <ul>
10     <li>
11       <label for="user">Username<br></label>
12       <input type="text" name="user" id="user">
13     </li>
14     <li>
15       <label for="pass">Password<br></label>
16       <input type="Password" name="pass" id="pass"
17     </li>
```

Gambar 2.4

Pada gambar 2.4 di atas terdapat perintah <ul> untuk membuat list

menggunakan simbol seperti bullet, cyrcle, dan lainnya. Selain itu terdapat pula `<li>` untuk menentukan berbagai item yang ingin ditampilkan. Label untuk user di sini menggunakan text Username dengan inputan yang bertype text, nama user, dan idnya adalah pass. Kemudian juga terdapat bagian pass atau bagian password dimana labelnya menggunakan text Password untuk inputnya bertype Password ( saat memasukkan password hanya tampil bullet saja tidak terlihat langsung password yang diketik ), dan nama serta id menggunakan pass.

```
18         </li>
19         <button type="Submit"> Submit </button>
20     </li>
21 </ul>
22 </form>
23
24 </body>
25 </html>
```

Gambar 2.5 Submit

Pada gambar di atas adalah merupakan untuk membuat tombol Submit dimana nantinya setelah menekan tombol submit ini form akan diarahkan ke output.php.

```
output.php  x  yo49.html  x
1 <?php
2
3 if ($_SERVER["REQUEST_METHOD"] == "POST") {
4     $user = $_REQUEST["user"];
5     $pass = $_POST["pass"];
6     $_user = strlen($user);
7     $_pass = strlen($pass);
8     $x = false;
9 }
```

Gambar 2.6 Output.php

Selanjutnya terdapat perintah untuk membuat sebuah perulangan yaitu dengan menggunakan if, dimana jika server yang digunakan menggunakan method == post seperti method yang terdapat pada program html sebelumnya maka program akan dijalankan. Pada gambar di atas juga didefinisikan bahwa \$user = \$request dimana disini menggunakan input dari label user, dan \$pass = \$post disini akan menggunakan input dari label pass. Kemudian juga terdapat \$\_user = strlen(\$user) yang berfungsi untuk menghitung karakter inputan dari label user, dan \$\_pass = strlen(\$pass) yang berfungsi untuk menghitung jumlah karakter inputan dari label pass. \$x untuk memeriksa apakah username dan



password yang dimasukkan sudah benar.

```
10     if ($_user > 7) {  
11         echo "Username harus kurang dari 7 karakter<br>  
12         ";  
13         $x = true;  
14     }
```

Gambar 2.7 Output.php

Kemudian pada gambar di atas juga dilakukan proses pengulangan menggunakan if, dimana perulangan ini merupakan bagian dari persyaratan perulangan pertama. Pada gambar diatas program menunjukkan jika \$\_user yang diinputkan lebih dari 7 karakter maka nilai dari perulangan ini adalah true dan akan menampilkan text “Username harus kurang dari 7 karakter”.

```
14  
15     if (!preg_match("/[A-Z]/", $pass)) {  
16         echo "Password harus terdapat huruf kapital<br>  
17         ";  
18         $x = true;  
19     }
```

Gambar 2.8 Output.php

Pada gambar di atas perintah perulangan yang dilakukan untuk mencocokkan apakah password yang dimasukkan terdapat huruf kapital atau tidak sebagai salah satu persyaratan untuk password yang benar. Jika pada password yang dimasukkan tidak terdapat huruf kapital maka \$x akan bernilai true dan akan menampilkan text “Password harus terdapat huruf kapital”.

```
19  
20     if (!preg_match("/[a-z]/", $pass)) {  
21         echo "Password harus terdapat huruf kecil<br>";  
22         $x = true;  
23     }  
24
```

Gambar 2.9 Output.php

Pada bagian program diatas fungsi nya sama seperti program pada gambar 2.8, hanya saja yang membedakannya adalah pada bagian ini digunakan untuk mencari huruf kecil pada password yang dimasukkan.

```

24
25
26     if (!preg_match("/[0-9]/", $pass)) {
27         echo "Password harus terdapat karakter khusus (
28             angka)<br>";
29         $x = true;
30     }

```

Gambar 2.10 Output.php

Selanjutnya selain untuk mencocokkan huruf terdapat juga perulangan untuk mencari angka pada password yang dimasukkan, dimana password yang diinputkan harus mengandung angka sebagai syaratnya. Jika tidak terdapat angka pada password yang dimasukkan maka nilai dari \$x adalah true dann text akan ditampilkan.

```

29
30
31     if ($_pass < 10) {
32         echo "Password harus lebih dari 10 karakter<br>
33             ";
34         $x = true;
35     }

```

Gambar 2.11 Output.php

Selain itu juga sebagai salah satu syaratnya adalah password yang diinputkan berjumlah lebih dari 10 karakter, jika password yang diinputkan kurang dari 10 karakter maka \$x akan bernilai true dan text akan ditampilkan.

```

34
35
36     if ($x == false) {
37         echo "Username dan Password memenuhi syarat<br>
38             ";
39         echo "Selamat Datang<br>";
40     }

```

Gambar 2.12 Output.php

Program diatas digunakan untuk memeriksa apakah username dan password yang dimasukkan sudah memenuhi semua persyaratannya, jika pada semua persyaratan diatas \$x bernilai false atau memenuhi syarat maka text yang akan ditampilkan adalah seperti gambar di atas.

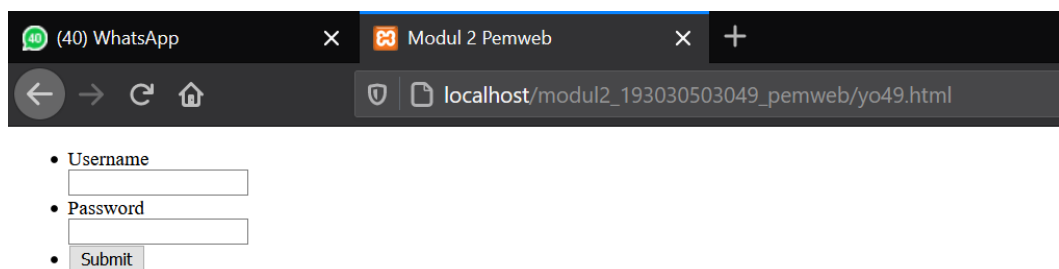
```

42
43 <!DOCTYPE html>
44 <html>
45 <head>
46 <title></title>
47 </head>
48 <body>
49 <br>
50 <a href="yo49.html">Keluar
51 </body>
52 </html>

```

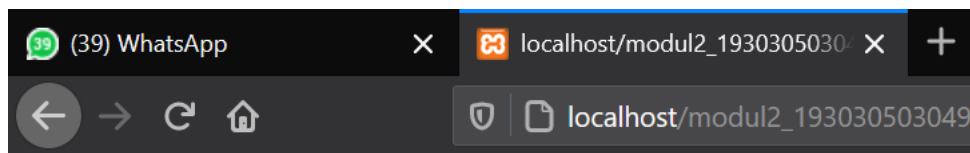
Gambar 2.13 Output.php

Html diatas digunakan untuk memanggil kembali yo49.html dengan mengklik pada tulisan “Keluar”.



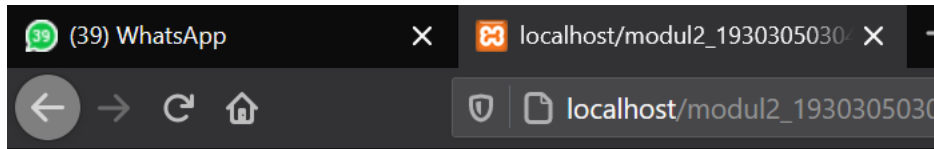
Gambar 2.14 Tampilan Web

Pada gambar di atas merupakan tampilan web pada saat program pertama kali di jalankan.



Gambar 2.15 Tampilan Web

Pada gambar di atas adalah contoh tampilan web apabila username yang dimasukkan lebih dari 7 karakter.



Password harus terdapat huruf kapital

[Keluar](#)

Gambar 2.16 Tampilan Web

Tampilan web apabila password yang dimasukkan tidak terdapat huruf kapitalnya.



Password harus terdapat huruf kecil

[Keluar](#)

Gambar 2.17 Tampilan Web

Tampilan web jika password yang dimasukkan tidak mengandung huruf kecil.



Password harus terdapat karakter khusus (angka)

[Keluar](#)

Gambar 2.18 Tampilan Web

Tampilan web yang akan muncul jika password yang dimasukkan tidak terdapat amgkanya.



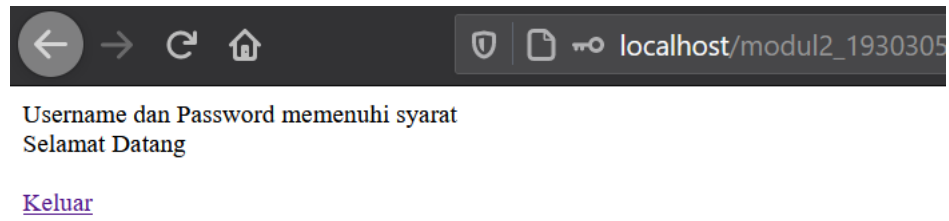
Password harus lebih dari 10 karakter

[Keluar](#)

Gambar 2.19 Tampilan Web

Tampilan web apabila password yang diinputkan tidak lebih dari 10

karakter.



Gambar 2.19 Tampilan Web

Tampilan web jika username dan password yang dimasukkan memenuhi syarat yang ada.

### **BAB III**

#### **KESIMPULAN**

Pada praktikum kali ini, mempelajari cara menggunakan form pada kasus penginputan username dan password dengan beberapa syarat yang harus disesuaikan. Form handling terdapat 2 method yaitu GET dan POST yang memiliki ciri khasnya masing - masing yaitu GET akan mengirimkan data ke URL sementara POST mengambil hasil inputan lalu ditampilkan pada file bagian web browser bukan URL. Pada praktikum kali ini menggunakan method POST. Kelebihan dari method POST adalah method ini digunakan untuk membuat login karena membuat data yang di inputkan tidak terlihat selain itu method ini sangat cocok untuk menyimpan data.

## **DAFTAR PUSTAKA**

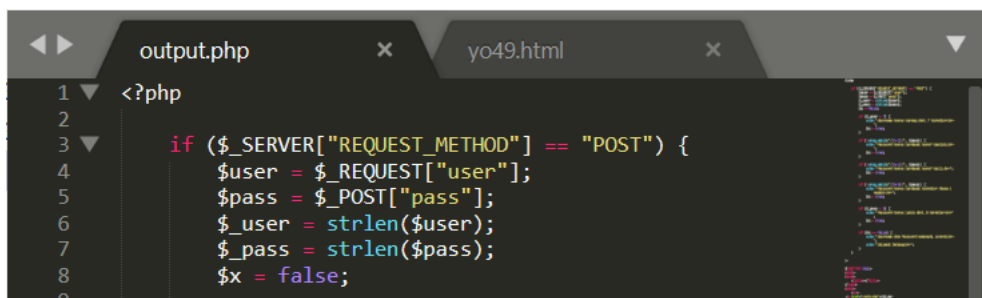
Tim Dosen Pemrograman Web & Mobile. Modul Pemrograman Web & Mobile.

2021. Palangka Raya. Jurusan Teknik Informatika Fakultas Teknik  
Universitas Palangka Raya (UPR).

## LAMPIRAN

```
18         <li>
19             <button type="Submit"> Submit </button>
20         </li>
21     </ul>
22 </form>
23
24 </body>
25 </html>
```

Gambar 2.5 Submit



```
1 <?php
2
3 if ($_SERVER["REQUEST_METHOD"] == "POST") {
4     $user = $_REQUEST["user"];
5     $pass = $_POST["pass"];
6     $user = strlen($user);
7     $pass = strlen($pass);
8     $x = false;
```

Gambar 2.6 Output.php

```
10     if ($_user > 7) {
11         echo "Username harus kurang dari 7 karakter<br>";
12         $x = true;
13     }
```

Gambar 2.7 Output.php

```
14
15     if (!preg_match("/[A-Z]/", $pass)) {
16         echo "Password harus terdapat huruf kapital<br>";
17         $x = true;
18     }
19
```

Gambar 2.8 Output.php

```
19
20     if (!preg_match("/[a-z]/", $pass)) {
21         echo "Password harus terdapat huruf kecil<br>";
22         $x = true;
23     }
24
```

Gambar 2.9 Output.php

```
24
25     if (!preg_match("/[0-9]/", $pass)) {
26         echo "Password harus terdapat karakter khusus (
27             angka)<br>";
28         $x = true;
29     }
```

Gambar 2.10 Output.php



```

29
30     if ($_pass < 10) {
31         echo "Password harus lebih dari 10 karakter<br>";
32         $x = true;
33     }
34

```

Gambar 2.11 Output.php

```

34
35     if ($x == false) {
36         echo "Username dan Password memenuhi syarat<br>";
37         echo "Selamat Datang<br>";
38     }
39 }
40

```

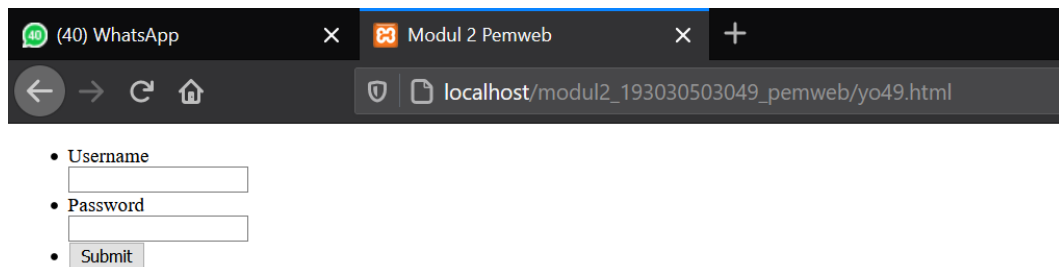
Gambar 2.12 Output.php

```

42
43 <!DOCTYPE html>
44 <html>
45 <head>
46     <title></title>
47 </head>
48 <body>
49     <br>
50     <a href="yo49.html">Keluar
51 </body>
52 </html>

```

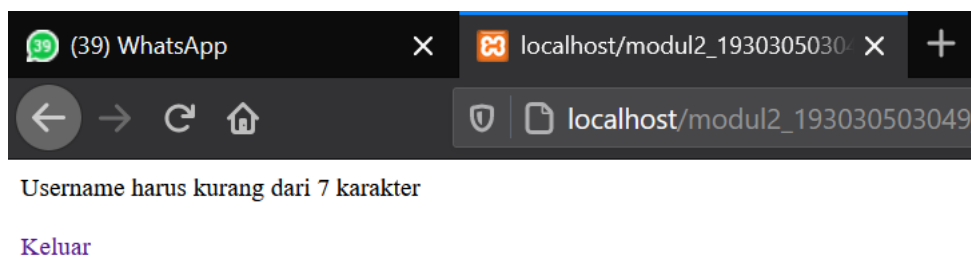
Gambar 2.13 Output.php



The screenshot shows a web browser window with two tabs: "(40) WhatsApp" and "Modul 2 Pemweb". The address bar displays "localhost/modul2\_193030503049\_pemweb/yo49.html". Below the browser, a login form is visible with the following elements:

- Username:
- Password:
- Submit:

Gambar 2.14 Tampilan Web

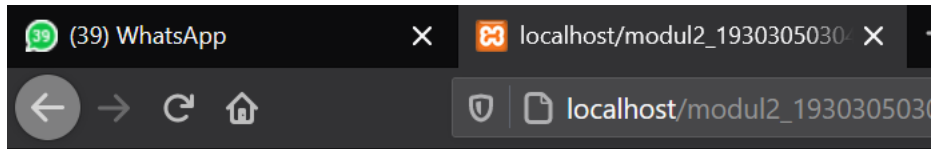


The screenshot shows a web browser window with two tabs: "(39) WhatsApp" and "localhost/modul2\_193030503049\_pemweb/yo49.html". The address bar displays "localhost/modul2\_193030503049\_pemweb/yo49.html". Below the browser, an error message is displayed:

Username harus kurang dari 7 karakter

Below the error message, there is a link labeled "Keluar" with a purple underline.

Gambar 2.15 Tampilan Web



Gambar 2.16 Tampilan Web



Password harus terdapat huruf kecil

[Keluar](#)

Gambar 2.17 Tampilan Web



Password harus terdapat karakter khusus (angka)

[Keluar](#)

Gambar 2.18 Tampilan Web



Password harus lebih dari 10 karakter

[Keluar](#)

Gambar 2.19 Tampilan Web



Username dan Password memenuhi syarat  
Selamat Datang

[Keluar](#)

Gambar 2.19 Tampilan Web