

**LAPORAN HASIL PRAKTIKUM**  
**PEMROGRAMAN WEBSITE DAN MOBILE I**



**NAMA : Muhammad Fahriyan Mahmudi**  
**NIM : 193020503025**  
**KELAS : A**  
**MODUL : II**

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

# BAB I

## TUJUAN DAN 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. Landasan 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>
  </body>
</html>
```

**Gambar 1.1 HTML**

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>
</html>

```

**Gambar 1.2** HTML dan PHP

Jika field nama diinputkan dengan Tono dan email diinputkan dengan tonon@mail.com maka output yang akan tampil adalah sebagai berikut: Welcome Budi Your email address is tonon@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>
  </body>
</html>

```

**Gambar 1.3**

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

```

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

```

**Gambar 1.4** HTML dan PHP

## GET vs. POST

GET dan POST membuat sebuah array (contoh array(kunci => nilai, kunci2 => nilai2, kunci3 => nilai3, ...)). Array ini menyimpan pasangan kunci/nilai, dimana kuncikunci 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 \*

**Gambar 1.7 VALIDASI**

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, tombol pilihan (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>
```

**Gambar 1.8 Text Field**

## 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
```

**Gambar 1.9** Text Radio Button

## Form Element

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

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

**Gambar 1.10** From Element

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

`$_SERVER["PHP_SELF"]` adalah variabel super global yang mengembalikannama 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, fungsi tersebut akan mengkonversikan karakter `<` dan `>` menjadi `<` dan `>`. Fungsi ini mencegah injeksi yang bisa dilakukan dengan HTML atau javascript (Cross-site ScriptingAttack) 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"];?>">
```

#### **Gambar 1.11 PHP**

Kemudian user memasukkan URL pada address bar dengan alamat sebagaiberikut:

http://localhost/test\_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/s  
cript %3E

yang jika ditranslasikan akan menjadi:

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

#### **Gambar 1.12 PHP**

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

#### **Berhati-hatilah dengan kemungkinan penambahan kode javascript pada tag**

Hacker bisa mengarahkan user ke file pada server yang lain, dan file itubisa 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(). Fungsitersebut akan mengkonversikan karakter khusus ke entitas HTML. Ketika usermemasukkan 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;">
```

#### **Gambar 1.13 Script**

## 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);
    return $data;
}

?>
```

**Gambar 1.14 PHP**

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 = "";
    }
}
```

**Gambar 1.15 PHP**

```

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

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

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = test_input($_POST["gender"]);
    }
}
?>

```

**Gambar 1.16 PHP**

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;?></span> <br><br>
    E-mail:
    <input type="text" name="email">
    <span class="error">* <?php echo $emailErr;?></span>
    <br><br>
    Website:
    <input type="text" name="website">
    <span class="error"><?php echo $websiteErr;?></span>
    <br><br>
    Comment: <textarea name="comment" rows="5" cols="40"></textarea>
    <br><br>
    Gender:
    <input type="radio" name="gender" value="female">Female

```

**Gambar 1.17 PHP**

```

☐* <?php echo $genderErr;?></span>
<br><br>


```

**Gambar 1.18 PHP**

### 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";
}

```

**Gambar 1.19 Validasi Nama**

Fungsi preg\_match() mencari string berdasarkan pola, mengembalikannilai 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";
}

```

**Gambar 1.20 Validasi Email**

### 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, makapesan error akan disimpan kedalam variabel \$websiteErr:

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

**Gambar 1.21** Validasi 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 perluditambahkan 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>.**

**Gambar 1.22** Validasi URL

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

**Gambar 1.23** Validasi URL.

## **BAB II**

### **PEMBAHASAN**

#### **Tugas Praktikum**

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.

**Full code :**

```

tabel > index.php
1  <?php
2      if ($_SERVER["REQUEST_METHOD"] == "POST"){
3          $username = $_REQUEST["username"];
4          $password = $_REQUEST["password"];
5          $user = strlen($username);
6          $pass = strlen($password);
7          $x = false;
8
9          if($user>7){
10             echo "<script>alert('Username lebih dari 7');</script>";
11             $x = true;
12         }
13         if (!preg_match("/[A-Z]/", $password) ) {
14             echo "<script>alert('Password kapital');</script>";
15             $x = true;
16         }
17         if (!preg_match("/[a-z]/", $password)) {
18             echo "<script>alert('Password kecil');</script>";
19             $x = true;
20         }
21
22         if (!preg_match("/^[a-zA-Z\d]/", $password)) {
23             echo "<script>alert('Password special character');</script>";
24             $x = true;
25         }
26
27         if (!preg_match("/[0-9]/", $password)) {
28             echo "<script>alert('Password angka');</script>";
29             $x = true;
30         }
31         if($pass<10){

```

**Gambar 2.1** program bagian 1

```

tabel > index.php
32     echo "<script>alert('Password kurang dari 10');</script>";
33     $x = true;
34     }
35     if( $x == false ){
36         echo "<script>alert('Berhasil');</script>";
37     }
38 }
39
40 ?>
41
42 <!DOCTYPE html>
43 <html lang="en">
44 <head>
45     <meta charset="UTF-8">
46     <meta name="viewport" content="width=device-width, initial-scale=1.0">
47     <title>Document</title>
48     <style>
49         body {
50             background-image: url('yi.jpg');
51             height: 100%;
52             background-repeat: no-repeat;
53             background-size: cover;
54         }
55     }
56
57     table {
58         font-size: 16px;
59         font-family: Georgia, 'Times New Roman', Times, serif;
60     }

```

**Gambar 2.2** program bagian 2

```

tabel > index.php
61     </style>
62 </head>
63 <body>
64 <form action="" method="post" name="Login_Form">
65     <br><br> <table border="0" align="center">
66         <tr>
67             <td colspan="2" align="center"></td>
68         </tr>
69         <tr>
70             <td colspan="2" align="center"><h3>LOGIN</h3></td>
71         </tr>
72         <tr>
73             <td align="right">Username : </td>
74             <td><input name="username" type="text" placeholder="Username"></td>
75         </tr>
76         <tr>
77             <td align="right">Password : </td>
78             <td><input name="password" type="password" placeholder="Password" ></td>
79         </tr>
80         <tr>
81             <td> </td>
82             <td><input name="Submit" type="submit" value="Login"></td>
83         </tr>
84     </br></br>
85 </table>
86 </form>
87 </body>
88 </html>

```

**Gambar 2.3** program bagian 3

#### PEMBAHASAN NYA :

```

tabel > index.php
1  <?php
2      if ($_SERVER["REQUEST_METHOD"] == "POST"){
3          $username = $_REQUEST["username"];
4          $password = $_REQUEST["password"];
5          $user = strlen($username);
6          $pass = strlen($password);
7          $x = false;
8

```

**Gambar 2.4** sintaks awal

Sintaks pada **Gambar 2.4** menjelaskan jika request method nya adalah post, maka data pada username yang bisa melakukan login dengan mempertimbangkan inputan user via web sebelum itu. Inputan tersebut bisa melakukan login dengan mempertimbangkan terpenuhinya aturan yang diminta pada tugas yang diaplikasikan



ke dalam bentuk if pada bagian handling. Adanya strlen berfungsi untuk menghitung panjangnya inputan string pada username dan password yang telah diinput oleh user.

Berikut percabangan antara lain :

```
if($user>7){  
    echo "<script>alert('Username lebih dari 7');</script>";  
    $x = true;  
}
```

**Gambar 2.5** perbangan 1

Percabangan yang saya buat ini digunakan untuk mengecek karakter dari \$username apakah lebih besar dari 7 atau tidak, jika lebih besar maka program akan menampilkan tulisan dalam echo bisa dilihat pada **Gambar 2.5** dan \$x akan bernilai true.

```
if (!preg_match("/[A-Z]/", $password) ) {  
    echo "<script>alert('Password kapital');</script>";  
    $x = true;  
}
```

**Gambar 2.6** perbangan 2

Percabangan yang saya buat ini untuk mengecek huruf kapital \$password apakah mengandung huruf kapital atau tidak dapat dilihat, fungsi preg match dimana fungsi ini guna nya untuk mengecek apakah ada kecocokan dalam string atau tidak, dimana kecocokan itu harus sesuai dengan [A-Z], jika \$password tidak mengandung huruf kapital maka akan menampilkan tulisan dalam echo bisa dilihat pada **Gambar 2.6** dan \$x akan bernilai true.

```
if (!preg_match("/[a-z]/", $password)) {  
    echo "<script>alert('Password kecil');</script>";  
    $x = true;  
}
```

**Gambar 2.7** percabangan 3

Percabangan yang saya buat ini untuk mengecek huruf kecil pada \$password apakah mengandung huruf kecil atau tidak dapat dilihat, fungsi preg match dimana

fungsi ini guna nya untuk mengecek apakah ada kecocokan dalam string atau tidak, dimana kecocokan itu harus sesuai dengan [a-z], jika \$password tidak mengandung huruf kecil maka akan menampilkan tulisan dalam echo bisa dilihat pada **Gambar 2.7** dan \$x akan bernilai true.

```
if (!preg_match("/^[a-zA-Z\d]/", $password)) {  
    echo "<script>alert('Password special character');</script>";  
    $x = true;  
}
```

**Gambar 2.8** percabangan 4

Percabangan yang saya buat ini untuk mengecek karakter spesial pada \$password, apakah mengandung karakter spesial atau tidak dapat dilihat, fungsi preg match dimana fungsi ini guna nya untuk mengecek apakah ada kecocokan dalam string atau tidak, dimana kecocokan itu harus sesuai dengan [^a-zA-Z)/d], jika \$password tidak mengandung karakter spesial maka akan menampilkan tulisan dalam echo bisa dilihat pada **Gambar 2.8** dan \$x akan bernilai true.

```
if (!preg_match("/[0-9]/", $password)) {  
    echo "<script>alert('Password angka');</script>";  
    $x = true;  
}
```

**Gambar 2.9** percabangan 5

Percabangan yang saya buat ini untuk mengecek angka pada \$password, apakah mengandung angka atau tidak dapat dilihat, fungsi preg match dimana fungsi ini guna nya untuk mengecek apakah ada kecocokan dalam string atau tidak, dimana kecocokan itu harus sesuai dengan [0-9], jika \$password tidak mengandung angka maka akan menampilkan tulisan dalam echo bisa dilihat pada **Gambar 2.9** dan \$x akan bernilai true.

```

}
if( $x == false ){
    echo "<script>alert('Berhasil');</script>";
}

```

**Gambar 2.10** percabangan 6

Percabangan 6 digunakan jika login sudah berhasil, kondisi ini jika sudah melewati percabangan 1-5 maka akan berhasil login.

Lalu setelah menjelaskan bagian php selanjut nya saya akan menjelaskan bagian html.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        body {
            background-image: url('yi.jpg');
            height: 100%;
            background-repeat: no-repeat;
            background-size: cover;
        }

        table {
            font-size: 16px;
            font-family: Georgia, 'Times New Roman', Times, serif;
        }
    </style>

```

**Gambar 2.11** bagian html

Pada bagian awal html saya mendeklarasikan head, didalam head ada body yang nanti nya saya edit, saya memasukkan background dari image dan ukuran pada background saya deklarasikan dibawah nya begitu juga pada table , disitu saya buat ukuran font didalam table dengan ukuran 16 pixel dan saya jua mengubah font nya menjadi times neew roman .

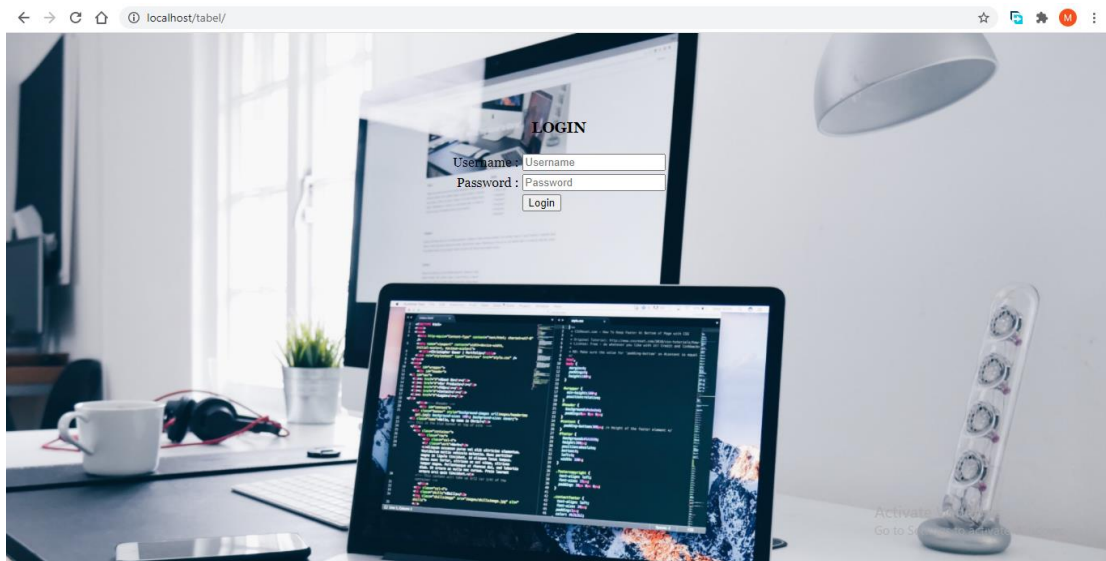
```

<body>
<form action="" method="post" name="Login_Form">
  <br><br> <table border="0" align="center">
    <tr>
      <td colspan="2" align="center"></td>
    </tr>
    <tr>
      <td colspan="2" align="center"><h3>LOGIN</h3></td>
    </tr>
    <tr>
      <td align="right">Username : </td>
      <td><input name="username" type="text" placeholder="Username"></td>
    </tr>
    <tr>
      <td align="right">Password : </td>
      <td><input name="password" type="password" placeholder="Password" ></td>
    </tr>
    <tr>
      <td></td>
      <td><input name="Submit" type="submit" value="Login"></td>
    </tr>
  </br></br>

```

**Gambar 2.12** bagian html

Lalu pada bagian body bisa dilihat pada sintaks diatas digunakan untuk sebagai nama, password yang dapat dipanggil di php. Lalu type adalah tipe dari inputan tersebut dalam hal ini tipe inputannya adalah teks dan terakhir placeholder digunakan untuk tulisan dalam wadah untuk inputan. di bagian input ketiga ini, tidak ada name, karena inputan ketiga tidak di panggil di php. Maka hasil nya adalah sebagai berikut :



**Gambar 2.13** Tampilan awal

localhost menyatakan  
Username lebih dari 7

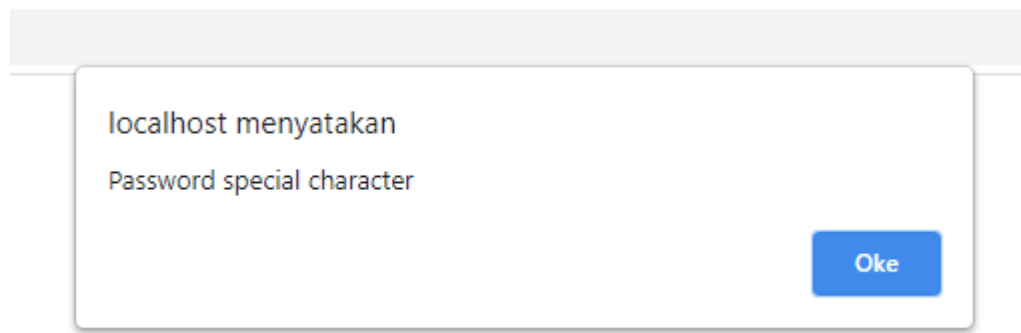
Oke

**Gambar 2.14** username lebih dari 7

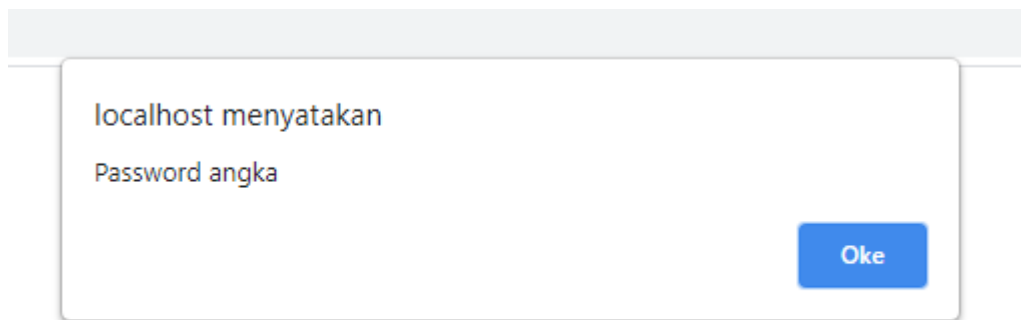
localhost menyatakan  
Password kapital

Oke

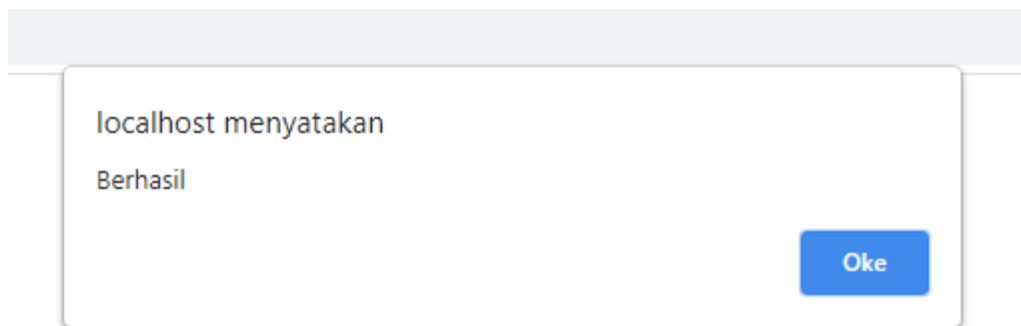
**Gambar 2.15** password kapital



**Gambar 2.16** password special karakter



**Gambar 2.17** password angka



**Gambar 2.18** ketika login berhasil

## **BAB III**

### **KESIMPULAN**

Handling digunakan untuk menangani berbagai inputan. Input akan dicek dan disesuaikan berdasarkan semua kondisi yang telah disiapkan sebelumnya, jika kondisi pertama tidak terpenuhi maka pengecakan berlanjut di kondisi kedua dan seterusnya hingga kondisi dari inputan tercapai.

Adanya method GET dan POST yang digunakan untuk mengambil dan atau mengirim nilai variabel ke halaman lain yang dikombinasikan dengan form sehingga user akan diarahkan ke halaman selanjutnya.

## DAFTAR PUSTAKA

Praktikum, K. (n.d.). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.*

*Belajar PHP : form Handling – WebHozzCode.(n.d).* Retrieved April 5, 2021,

From <https://www.webhozz.com/code/php-form-handling/>

Yukawa, K.(2019). *Tutorial mendesain form login menggunakan HTML&CSS\_by kumiyuki Yukawa\_Medium* HTML dan CSS



## LAMPIRAN

```
tabel > index.php
1  <?php
2      if ($_SERVER["REQUEST_METHOD"] == "POST"){
3          $username = $_REQUEST["username"];
4          $password = $_REQUEST["password"];
5          $user = strlen($username);
6          $pass = strlen($password);
7          $x = false;
8
9          if($user>7){
10             echo "<script>alert('Username lebih dari 7');</script>";
11             $x = true;
12         }
13         if (!preg_match("/[A-Z]/", $password) ) {
14             echo "<script>alert('Password kapital');</script>";
15             $x = true;
16         }
17         if (!preg_match("/[a-z]/", $password)) {
18             echo "<script>alert('Password kecil');</script>";
19             $x = true;
20         }
21
22         if (!preg_match("/^[a-zA-Z\d]/", $password)) {
23             echo "<script>alert('Password special character');</script>";
24             $x = true;
25         }
26
27         if (!preg_match("/[0-9]/", $password)) {
28             echo "<script>alert('Password angka');</script>";
29             $x = true;
30         }
31         if($pass<10){
```

Gambar 2.1 program bagian 1

```

tabel > index.php
32     echo "<script>alert('Password kurang dari 10');</script>";
33     $x = true;
34     }
35     if( $x == false ){
36         echo "<script>alert('Berhasil');</script>";
37     }
38 }
39
40 ?>
41
42 <!DOCTYPE html>
43 <html lang="en">
44 <head>
45     <meta charset="UTF-8">
46     <meta name="viewport" content="width=device-width, initial-scale=1.0">
47     <title>Document</title>
48     <style>
49         body {
50             background-image: url('yi.jpg');
51             height: 100%;
52             background-repeat: no-repeat;
53             background-size: cover;
54         }
55     }
56
57     table {
58         font-size: 16px;
59         font-family: Georgia, 'Times New Roman', Times, serif;
60     }

```

**Gambar 2.2** program bagian 2

```

tabel > index.php
61     </style>
62 </head>
63 <body>
64 <form action="" method="post" name="Login_Form">
65     <br><br> <table border="0" align="center">
66         <tr>
67             <td colspan="2" align="center"></td>
68         </tr>
69         <tr>
70             <td colspan="2" align="center"><h3>LOGIN</h3></td>
71         </tr>
72         <tr>
73             <td align="right">Username : </td>
74             <td><input name="username" type="text" placeholder="Username"></td>
75         </tr>
76         <tr>
77             <td align="right">Password : </td>
78             <td><input name="password" type="password" placeholder="Password" ></td>
79         </tr>
80         <tr>
81             <td> </td>
82             <td><input name="Submit" type="submit" value="Login"></td>
83         </tr>
84     </br></br>
85 </table>
86 </form>
87 </body>
88 </html>

```

**Gambar 2.3** program bagian 3

```

tabel > index.php
1  <?php
2      if ($_SERVER["REQUEST_METHOD"] == "POST"){
3          $username = $_REQUEST["username"];
4          $password = $_REQUEST["password"];
5          $user = strlen($username);
6          $pass = strlen($password);
7          $x = false;
8

```

**Gambar 2.4** sintaks awal

```

    if($user>7){
        echo "<script>alert('Username lebih dari 7');</script>";
        $x = true;
    }

```

**Gambar 2.5** perbangan 1

```

if (!preg_match("/[A-Z]/", $password) ) {
    echo "<script>alert('Password kapital');</script>";
    $x = true;
}

```

**Gambar 2.6** perbangan 2

```

if (!preg_match("/[a-z]/", $password)) {
    echo "<script>alert('Password kecil');</script>";
    $x = true;
}

```

**Gambar 2.7** percabangan 3

```

if (!preg_match("/[^a-zA-Z\d]/", $password)) {
    echo "<script>alert('Password special character');</script>";
    $x = true;
}

```

**Gambar 2.8** percabangan 4

```

if (!preg_match("/[0-9]/", $password)) {
    echo "<script>alert('Password angka');</script>";
    $x = true;
}

```

**Gambar 2.9** percabangan 5

```

}
if( $x == false ){
    echo "<script>alert('Berhasil');</script>";
}

```

**Gambar 2.10** percabangan 6

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    body {
      background-image: url('yi.jpg');
      height: 100%;
      background-repeat: no-repeat;
      background-size: cover;

    }

    table {
      font-size: 16px;
      font-family: Georgia, 'Times New Roman', Times, serif;
    }
  </style>

```

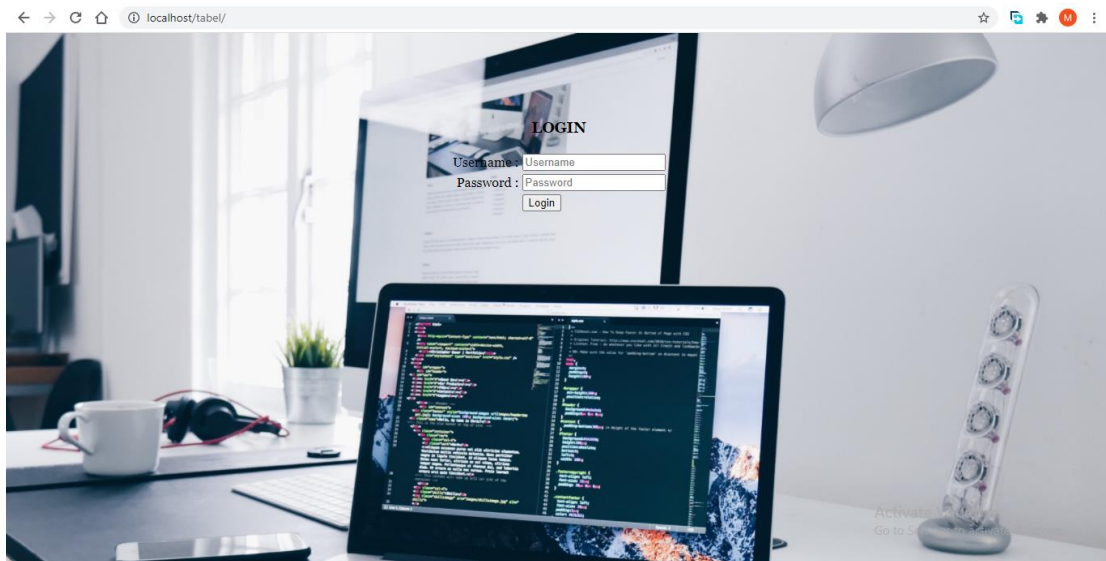
**Gambar 2.11** bagian html

```

<body>
<form action="" method="post" name="Login_Form">
  <br><br> <table border="0" align="center">
    <tr>
      <td colspan="2" align="center"></td>
    </tr>
    <tr>
      <td colspan="2" align="center"><h3>LOGIN</h3></td>
    </tr>
    <tr>
      <td align="right">Username : </td>
      <td><input name="username" type="text" placeholder="Username"></td>
    </tr>
    <tr>
      <td align="right">Password : </td>
      <td><input name="password" type="password" placeholder="Password" ></td>
    </tr>
    <tr>
      <td></td>
      <td><input name="Submit" type="submit" value="Login"></td>
    </tr>
  </br></br>

```

**Gambar 2.12** bagian html



**Gambar 2.13** Tampilan awal

localhost menyatakan  
Username lebih dari 7

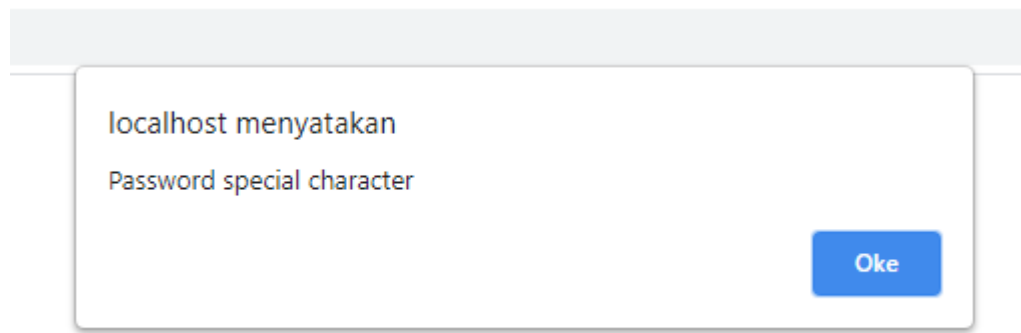
Oke

**Gambar 2.14** username lebih dari 7

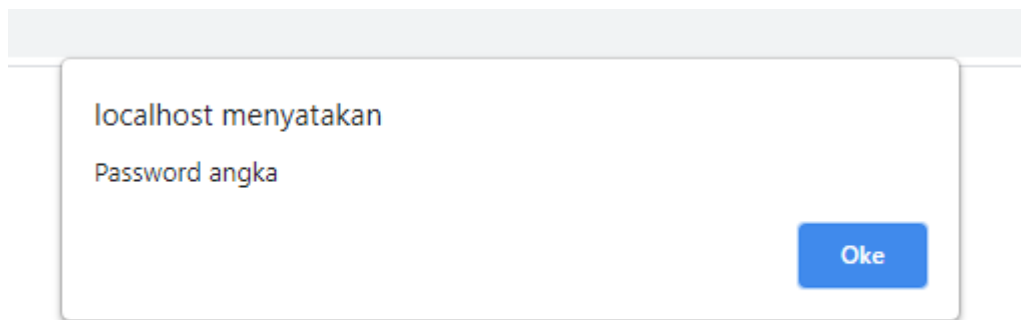
localhost menyatakan  
Password kapital

Oke

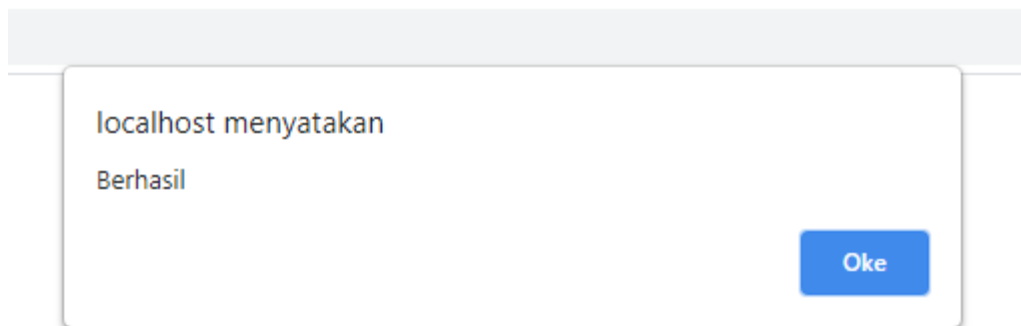
**Gambar 2.15** password kapital



**Gambar 2.16** password special karakter



**Gambar 2.17** password angka



**Gambar 2.18** ketika login berhasil