

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN WEB DAN MOBILE I**



**NAMA : MUHAMMAD FAUZAN**

**NIM : 193020503024**

**KELAS : A**

**MODUL : II (FORM HANDLING)**

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

# **BAB I**

## **TUJUAN DAN LANDASAN TEORI**

### **1.1. Tujuan**

- 1.1.1. Mahasiswa mampu membuat handling yang mampu mengolah data dari form HTML
- 1.1.2. Mahasiswa mampu membuat batasan – batasan untuk menangani inputan dari form HTML

### **1.2. Landasan Teori**

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. (Achmad Solichin. S.Kom, 2016)

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>
```

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>
```

### 1.2.1. GET & 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. (Praktikum, 2021.)

### 1.2.2. Validasi Form PHP

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: (Praktikum, n.d.)

Field	Rute 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

## BAB II

### PEMBAHASAN

Pada tugas praktikum modul ini para praktikan diperintahkan untuk membuat 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

Dengan kriteria tugas seperti diatas praktikan diharapkan mampu membuat sebuah website untuk menginputkan username dan juga password menggunakan form dan penanganan. Berikut adalah penjelesanan mengenai program html dari program yang dibuat.

```
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>Modul 2</title>
48 </head>
49 <body>
50 <form action="" method="post">
51   <table border="0" align="center">
52     <tr>
53       <td colspan="3" align="center"><h3>Login</h3></td>
54     </tr>
55     <tr>
56       <td align="right">Username : </td>
57       <td><input name="username" type="text" placeholder="Username"></td>
58     </tr>
59     <tr>
60       <td align="right">Password : </td>
61       <td><input name="password" type="password" placeholder="Password" ></td>
62     </tr>
63     <tr>
64       <td> </td>
65       <td><input name="Submit" type="submit" value="Login"></td>
66     </tr>
67   </table>
68 </form>
69 </body>
70 </html>
```

Gambar 2.1. HTML

Didalam struktur terdapat dua buah bagian yaitu header HTML dan body HTML. Bagian head pada html terdapat mulai pembukaan tag `<!DOCTYPE html>` hingga penutupan `</head>`. Dan bagian body terdapat mulai pembukaan tag `<body>` hingga penutupan `</body>`. Didalam struktur body html terdapat *form* untuk menjadi wadah inputan bagi user untuk menginputkan data didalamnya. Didalam *form* terdapat suatu elemen yaitu *action*, elemen ini bisa memproses suatu file namun didalam file yang berbeda. Namun jika dikosongkan seperti gambar diatas maka action tersebut tidak akan memproses di file yang lain dan hanya akan memproses file tersebut dan dengan menggunakan method POST.

Agar tampilan html terkesan lebih rapi maka disini saya membuatnya dengan menggunakan tabel namun dengan ketebalan border = 0 agar garis tidak terlihat. Pada tag pembukaan input terdapat beberapa elemen yang masih baru dan belum dijelaskan di modul sebelumnya seperti name dan type. Name didalam elemen ini berfungsi untuk memberi nama pada sebuah html agar nantinya dapat dipanggil kedalam penyelesaian php. Type adalah elemen yang berfungsi untuk mendefinisikan form input yang ada. Berikut adalah penjelasan mengenai PHP.

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

**Gambar 2.2.** Validasi PHP

Dapat dilihat pada syntax program php diatas terdapat tag pembukaan php atau `<?php` dan juga fungsi pemilihan atau percabangan if. Dalam fungsi ini terdapat variabel super global SERVER yang memiliki REQUEST\_METHOD berupa method POST. Fungsi ini berguna untuk menjadi validasi terhadap username dan password yang akan diinput.

Variabel username dan password dideklarasikan untuk menampung data inputan melalui name yang dipanggil yang sudah dideklarasikan didalam input form username dan juga password. Variabel user dan pass disini berfungsi untuk menghitung Panjang karakter dari variabel username dan juga variabel password. Terdapat juga variabel x yang bernilai = true sebagai Boolean check jika kondisi benar atau salah.

```
if($user>7){
    echo "<script>alert('Username Lebih dari 7');</script>";
    $x = false;
}
if (!preg_match("/[A-Z]/", $password) ) {
    echo "<script>alert('password kapital');</script>";
    $x = false;
}
if (!preg_match("/[a-z]/", $password)) {
    echo "<script>alert('password kecil');</script>";
    $x = false;
}

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

if (!preg_match("/[0-9]/", $password)) {
    echo "<script>alert('password digit');</script>";
    $x = false;
}
if($pass<10){
    echo "<script>alert('password lebih dari 10');</script>";
    $x = false;
}

if( $x == true ){
    echo "<script>alert('berhasil');</script>";
}
}
```

**Gambar 2.3.** Error Handling

Fungsi if diatas berfungsi sebagai error handling. Dengan kondisinya masing masing sesuai kriteria yang diminta pada tugas praktikum. Yaitu, username

lebih dari 7 karakter, password setidaknya menggunakan huruf kapital, password setidaknya menggunakan huruf kecil, password setidaknya menggunakan special karakter, password setidaknya menggunakan angka dan password tidak boleh kurang dari 10 karakter. Login hanya berhasil jika kondisi diatas tidak terpenuhi dan variabel x akan bernilai true yang artinya login berhasil.



### **BAB III**

#### **KESIMPULAN**

Setelah menyelesaikan praktikum ini saya dapat menyimpulkan bahwa penggunaan form handling dalam pembuatan sebuah form sangatlah penting, tidak hanya menghindari masalah pada program hal ini juga dapat untuk memastikan program berjalan baik dengan baik dan benar.

## DAFTAR PUSTAKA

- Achmad Solichin. S.Kom. (2016). Pemrograman Web dengan PHP dan MySQL - Achmad Solichin - Google Buku. *Universitas Budi Luhur, June*, 85.  
<https://books.google.co.id/books?id=kcD4BQAAQBAJ&printsec=frontcover&dq=aplikasi+berbasis+web+dengan&hl=id&sa=X&ved=0ahUKEwib-fT80ITYAhVBrI8KHT9GD6QQ6AEIJzAA#v=onepage&q=aplikasi+berbasis+web+dengan&f=false>
- Praktikum, K. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

## LAMPIRAN

```
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>Modul 2</title>
48 </head>
49 <body>
50 <form action="" method="post">
51   <table border="0" align="center">
52     <tr>
53       <td colspan="3" align="center"><h3>Login</h3></td>
54     </tr>
55     <tr>
56       <td align="right">Username : </td>
57       <td><input name="username" type="text" placeholder="Username"></td>
58     </tr>
59     <tr>
60       <td align="right">Password : </td>
61       <td><input name="password" type="password" placeholder="Password" ></td>
62     </tr>
63     <tr>
64       <td></td>
65       <td><input name="Submit" type="submit" value="Login"></td>
66     </tr>
67   </table>
68 </form>
69 </body>
70 </html>
```

Gambar 2.1. HTML

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

Gambar 2.2. Validasi PHP

```

if($user>7){
    echo "<script>alert('Username Lebih dari 7');</script>";
    $x = false;
}
if (!preg_match("/[A-Z]/", $password) ) {
    echo "<script>alert('password kapital');</script>";
    $x = false;
}
if (!preg_match("/[a-z]/", $password)) {
    echo "<script>alert('password kecil');</script>";
    $x = false;
}

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

if (!preg_match("/[0-9]/", $password)) {
    echo "<script>alert('password digit');</script>";
    $x = false;
}
if($pass<10){
    echo "<script>alert('password lebih dari 10');</script>";
    $x = false;
}

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

```

?>

**Gambar 2.3.** Error Handling