

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



NAMA : Okki Andaresta
NIM : DBC 118 068
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. Mahasiswa mampu membuat handling yang mampu mengolah data dari form HTML.
2. Mahasiswa mampu membuat batasan-batasan untuk menangani input dari form HTML.

1.2. 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>
```

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>
```

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

Welcome Budi

Your email address is tono@mail.com

Hasil yang sama juga akan tampil dengan menggunakan method get :

```
<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>
```

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>
```

Perbedaan GET dan 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 padajumlah 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.

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, 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>`

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 `<` dan `>`. Fungsi inimencegah injeksi yang bisa dilakukan dengan HTML atau javascript (Cross-site Scripting Attack) pada form tersebut.

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”.

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, tabextra, dan baris baru yang ekstra) dari data input user (dengan fungsi `trim()`).
2. Membuang backslash (\) satu garis miring dari data input user (dengan

funksistripslashes()).

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;

}

?>
```

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

}

?>
```

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
```

```
    <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 fieldinput 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

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

BAB II

PEMBAHASAN

Pada Praktikum modul 2 ini membuat program web untuk menginputkan username dan password menggunakan form dan penanganan input data dengan kriteria sebagai berikut, username yang diinputkan tidak boleh lebih dari tujuh karakter, password yang diinputkan harus terdiri dari huruf kapital, huruf kecil, angkadan karakter khusus dan jumlah karakter password tidak boleh kurang dari sepuluh karakter. Berikut ini merupakan kode program yang telah dibuat :

```
modu2validationform.php
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <style>
5 .error {color: #FF0000;}
6 </style>
7 </head>
8 <body>
9
10 <?php
11 $username_pesan= $password_pesan="";
12 $username = $password = "";
13
14 //Mengecek apakah ada kiriman form dari method post
15 if ($_SERVER["REQUEST_METHOD"] == "POST") {
16
17     //Mengecek apakah field name kosong ?
18     if (empty($_POST["username"])) {
19         $username_pesan = "Nama harus diisi";
20     } else {
21         $username = htmlspecialchars($_POST["username"]);
22     }
23
24     if (empty($_POST["password"])) {
25         $password_pesan = "Password harus diisi";
26     } else {
27
28         //Cek apakah jumlah karakter password kurang dari 5 karakter ?
29         if (strlen($password)<10){
30             $password_pesan = "Password kurang dari 10 karakter";
31         }else {
32             $password = htmlspecialchars($_POST["password"]);
33         }
34     }
35 }
36
37 ?>
38
39 <h2>Form Penanganan input data</h2>
40
41 <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
42     Username: <input type="text" name="username">
43     <span class="error">* <?php echo $username_pesan;?></span>
44     <br><br>
45     Password: <input type="text" name="password">
46     <span class="error">* <?php echo $password_pesan;?></span>
47     <br><br>
48     <input type="submit" name="submit" value="Submit">
49 </form>
50
51 <?php
52 echo "<h2>Hasil inputan:</h2>";
53 echo "Nama : " . $username;
54 echo "<br>";
55 echo "Password : " . $password;
56 ?>
57
58 </body>
59 </html>
```

Gambar 2.1 Code Program

Program Web form ini bertujuan untuk menginput username dan password dibuat agar pengguna dapat memasukkan username dan menekan tombol submit

maka inputan tersebut akan langsung tampil di halaman itu sendiri tanpa berpindah halaman.

Pada elemen formnya menggunakan variabel super global yaitu `$_SERVER["PHP_SELF"]`; Dengan menggunakan variabel super global ini dapat mengirim data lewat form pada halaman itu sendiri tanpa harus berpindah halaman. Hal ini berfungsi untuk menampilkan pesan error saat menggunakan form validation di halaman yang sama.

Pada Program ini terdapat fungsi pada PHP yaitu `htmlspecialchars()`. Fungsi `htmlspecialchars()` akan mengubah/mengkonversi karakter khusus HTML menjadi named entity atau karakter string biasa. Sehingga tidak akan dieksekusi oleh web browser.

```
if ($_SERVER["REQUEST_METHOD"] == "POST")
```

Syntax tersebut berfungsi untuk memeriksa apakah ada nilai kiriman form dengan menggunakan method POST, jika ada maka akan dieksekusi pada bagian sintaks berikut.

Berikut ini merupakan program web yang telah dijalankan melalui localhost pada browser Untuk membuat validasi ini menggunakan function `strlen()`. Fungsi ini akan menghitung jumlah karakter pada sebuah string dengan ketentuan jumlah karakter password tidak boleh kurang dari 10 karakter.

Form Penanganan input data

Username: *

Password: * Password kurang dari 10 karakter

Hasil inputan:

Nama : okki
Password :

Gambar 2.2 Program Web Penanganan Input Data

BAB III

KESIMPULAN

Pada praktikum ini dapat disimpulkan bahwa :

Penggunaan Form HTML dibuat dengan cara memakai tag, sehingga tag form memiliki attribute method yang dapat digunakan untuk menyalurkan data dikirim ke halaman yang diset di dalam attribute action. Pada PHP terdapat Superglobals variabel yaitu variabel \$ _GET dan variabel \$ _POST. Perbedaan dari keduanya adalah untuk mengambil data dari get request, harus menggunakan \$ _GET, untuk post request harus menggunakan \$ _POST.

DAFTAR PUSTAKA

Alfa Adhitya. 2012. “*PHP Form Handling*”.

<https://www.lab-informatika.com/php-form-handling/>. Diakses pada Jum’at,
9 April 2021 pukul 19.45 WIB.

Dosen Teknik Informatika. 2021. *Pemrograman Web Dan Mobile I*. Universitas
Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.

LAMPIRAN

```
modul2validationform.php
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <style>
5   error {color: #FF0000;}
6 </style>
7 </head>
8 <body>
9
10 <?php
11 $username_pesan= $password_pesan="";
12 $username = $password = "";
13
14 //Mengecek apakah ada kiriman form dari method post
15 if ($_SERVER["REQUEST_METHOD"] == "POST") {
16
17   //Mengecek apakah field name kosong ?
18   if (empty($_POST["username"])) {
19     $username_pesan = "Nama harus diisi";
20   } else {
21     $username = htmlspecialchars($_POST["username"]);
22   }
23
24   if (empty($_POST["password"])) {
25     $password_pesan = "Password harus diisi";
26   } else {
27
28     //Cek apakah jumlah karakter password kurang dari 5 karakter ?
29     if (strlen($password)<10){
30       $password_pesan = "Password kurang dari 10 karakter";
31     }else {
32       $password = htmlspecialchars($_POST["password"]);
33     }
34   }
35 }
36
37 ?>
38
39 <h2>Form Penanganan input data</h2>
40
41 <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
42   Username: <input type="text" name="username">
43   <span class="error">*<?php echo $username_pesan;?></span>
44   <br><br>
45   Password: <input type="text" name="password">
46   <span class="error">*<?php echo $password_pesan;?></span>
47   <br><br>
48   <input type="submit" name="submit" value="Submit">
49 </form>
50
51 <?php
52 echo "<h2>Hasil inputan:</h2>";
53 echo "Nama : ".$username;
54 echo "<br>";
55 echo "Password : ".$password;
56 ?>
57
58 </body>
59 </html>
```

Gambar 2.1 Code Program

Form Penanganan input data

Username: *

Password: * Password kurang dari 10 karakter

Hasil inputan:

Nama : olki
Password :

Gambar 2.2 Program Web Penanganan Input Data

