

**LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB I**



NAMA : ADITYA ANANDA RAMADHANI
NIM : 193030503072
KELAS : A
MODUL : II (Form Handling)

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

BAB I

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

1.2.1. ELEMEN FORM

Element form berfungsi untuk mendefinisikan form intraktif. Elemen ini mempunyai attribute yaitu action dan method. Attribute action berisikan aksi terhadap form yang dikirim (url) dan attribute methode berisikan metode form melakukan proses pengiriman data (GET/POST). (Jayanti & Raja, 2020)

Contoh :

```
<form action="..." method="...">
</form>
```

1.2.2. ELEMEN INPUT

Element INPUT berfungsi untuk mendefinisikan input yang akan dimasukkan pengguna. Element ini mempunyai attribute yaitu name, size, type, value, checked. Attribute name mendefinisikan nama dari input kontrol form, attribute size mendefinisikan ukuran teks pada input kontrol, type mendefinisikan bentuk-bentuk input kontrol, value mendefinisikan nilai awal/reset/submit, checked mendefinisikan pilihan terpilih pada type radio/checkbox. Element ini tidak mempunyai tag penutup dan harus berada di dalam element FORM. (Jayanti & Raja, 2020)

Contoh :

```
<input name = "name" size = "number" type = "text" | "checkbox" |  
"radio" | "submit" | "reset" value = "value" checked >
```

1.2.3. \$_REQUEST (REQUEST)

\$_REQUEST PHP digunakan untuk mengumpulkan data setelah submit sebuah form HTML. Contoh berikut menunjukkan sebuah form dengan field input dan tombol submit. Ketika user melakukan submit data dengan menekan “Submit”, data form dikirimkan untuk file yang dispesifikasikan dalam atribut aksi dari tag <form>. Dalam contoh ini, kita memperhatikan file ini memproses data form. Jika pengguna ingin menggunakan file PHP lain untuk memproses data form, menggantinya dengan nama file yang dipilih, maka gunakan variabel super global \$_REQUEST untuk mengumpulkan field nilai input.

1.2.4. \$_POST (POST)

\$_POST PHP sangat banyak digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan method=“post”. \$_POST juga banyak digunakan untuk melewati variabel. Contoh berikut menunjukkan sebuah form dengan sebuah input field dan sebuah tombol submit. Ketika user melakukan submit data dengan menekan tombol submit, data form dikirim ke file yang dispesifikasikan dalam atribut aksi dari tag <form>. Pada contoh ini, perhatikan file untuk pemrosesan data. Jika ingin menggunakan file PHP lain untuk memproses data form, ganti dengan nama file yang diinginkan. Kemudian, gunakan variabel super global \$_POST untuk mengumpulkan nilai field input.

1.2.5. \$_GET (GET)

\$_GET PHP bisa digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan method=“get”. Juga bisa digunakan untuk mengumpulkan data yang dikirimkan dalam URL.

1.2.6. 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.

1.2.7. REGULAR EXPRESSION

Regular Expression atau biasa disingkat regex, adalah suatu metode untuk mengenali atau mendeteksi suatu pola tertentu pada suatu string. Dengan menggunakan regex, kita bisa mendeteksi pola string seperti email, hashtag, link dan pola-pola kompleks lainnya dengan hanya satu ekspresi saja. regex juga merupakan metode standar dan independen, tidak mengenal bahasa pemrograman

Di dalam PHP, terdapat beberapa fungsi yang berkaitan dengan penggunaan regex. Di antaranya:

Fungsi	Deskripsi
<code>preg_match()</code>	Mencari kata/karakter yang sesuai dengan pola regex.
<code>preg_match_all()</code>	Mencari <i>semua</i> kata/karakter yang sesuai dengan pola regex.
<code>preg_replace()</code>	Mencari kata/karakter yang sesuai dengan pola regex, lalu menyimpannya dengan data baru.
<code>preg_grep()</code>	Mengembalikan kata/karakter yang sesuai dengan pola regex.
<code>preg_split()</code>	Membagi string menjadi sebuah array menggunakan pola regex.

Gambar 1.1 tabel fungsi-fungsi bawaan PHP untuk bekerja dengan regex

Terdapat banyak sekali meta karakter atau pola dalam regex. Akan tetapi, di bawah ini adalah yang mungkin paling sering anda gunakan.

Karakter	Deskripsi
<code>\s</code>	karakter putih (spasi, tab, baris baru)
<code>\d</code>	angka digit (0-9)
<code>\w</code>	karakter huruf (a-z, A-Z, 0-9, _)
<code>[aeiou]</code>	pola yang terdiri dari set karakter di dalam kurung siku
<code>[^aeiou]</code>	pola yang tidak terdiri dari set karakter di dalam kurung siku
<code>(foo bar baz)</code>	pilihan dari kata yang didefinisikan

Gambar 1.2 tabel beberapa meta karakter regex yang sering digunakan

BAB II

PEMBAHASAN

pembahasan berdasarkan materi soal yang ada pada modul Pemograman Web adalah sebagai berikut :

pada bagian awal dilakukan deklarasi variable adalah sebagai berikut :

`$username_h = $password_h = $usernameErr = $passwordErr = $berhasil = "";`

Variable pada bagian ini dideklarasikan dengan nilai string kosong. Berikut merupakan penjelasan kegunaan dari setiap variable :

- a) Variable `username_h` berguna untuk menyimpan username yang berhasil di validasi.
- b) Variable `password_h` berguna untuk menyimpan password yang berhasil di validasi.
- c) Variable `usernameErr` berguna untuk menyimpan pesan error untuk username.
- d) Variable `passwordErr` berguna untuk menyimpan pesan error untuk password.

Pada bagian berikut nya terdapat percabangan if POST yang digunakan mengecek apakah form telah diisi. Percabangan if POST adalah sebagai berikut :

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $username = $_REQUEST["username"];  
    $password = $_REQUEST["password"];  
    $user = strlen($username);  
    $pass = strlen($password);  
    $x = false;  
    if($user > 7){  
        $usernameErr = "Username lebih dari 7 karakter";  
        $x = true;  
    }  
    if (!preg_match("/[A-Z]/", $password)) {  
        $passwordErr = "password tidak memiliki huruf kapital";  
        $x = true;  
    }  
    if (!preg_match("/[a-z]/", $password)) {  
        $passwordErr = "password tidak memiliki huruf kecil";  
    }  
}
```

```

    $x = true;
}

if (!preg_match("/^[a-zA-Z\d]/", $password)) {
    $passwordErr = "password tidak memiliki special character";
    $x = true;
}

if (!preg_match("/[0-9]/", $password)) {
    $passwordErr = "password tidak memiliki digit/angka";
    $x = true;
}

if ($pass > 10) {
    $passwordErr = "password lebih dari 10 karakter";
    $x = true;
}

if ($x == false) {
    $berhasil = "<h2>BERHASIL</h2>";
    $username_h = $_POST["username"];
    $password_h = $_POST["password"];
}
}

```

Percabangan if (`$_SERVER["REQUEST_METHOD"] == "POST"`) digunakan untuk melakukan pengecekan terhadap form apakah telah terisi atau belum. Didalam percabangan if ini dijalankan beberapa percabangan if lainnya. Berikut merupakan penjelasan dari setiap percabangan if :

1. username yang diinputkan tidak boleh lebih dari tujuh karakter.

```

if($user>7){
    $usernameErr = "Username lebih dari 7 karakter";
    $x = true;
}

```

Pada bagian ini menggunakan dengan menggunakan percabangan if dengan kondisi yaitu variable user lebih dari 7 karakter maka akan dilakukan :

- a) variable usernameErr akan memiliki nilai string “Username Lebih dari 7 karakter”.
 - b) Variable x akan bernilai true.
2. password yang diinputkan harus terdiri dari huruf kapital, huruf kecil, angka dan karakter khusus.

- a. Percabangan IF untuk pemeriksaan password menggunakan huruf kapital

```
if (!preg_match("/[A-Z]/", $password)) {  
    $passwordErr = "password tidak memiliki huruf kapital";  
    $x = true;  
}
```

Pada bagian ini menggunakan dengan menggunakan percabangan if dengan kondisi yaitu fungsi regex(preg_match) dengan regex yaitu “/[A-Z]/” atau pola yang dicari adalah huruf kapital pada variable password maka akan dilakukan :

- a) variable usernameErr akan memiliki nilai string “password tidak memiliki huruf kapital”.
- b) Variable x akan bernilai true.

- b. Percabangan IF untuk pemeriksaan password menggunakan huruf kecil

```
if (!preg_match("/[a-z]/", $password)) {  
    $passwordErr = "password tidak memiliki huruf kecil";  
    $x = true;  
}
```

Pada bagian ini menggunakan dengan menggunakan percabangan if dengan kondisi yaitu fungsi regex(preg_match) dengan regex yaitu “/[a-z]/” atau pola yang dicari adalah huruf kecil pada variable password maka akan dilakukan :

- a) variable usernameErr akan memiliki nilai string “password tidak memiliki huruf kecil”.
- b) Variable x akan bernilai true.

- c. Percabangan IF untuk pemeriksaan password menggunakan special karakter

```
if (!preg_match("/^[a-zA-Z\d]/", $password)) {  
    $passwordErr = "password tidak memiliki special character";  
    $x = true;  
}
```

Pada bagian ini menggunakan dengan menggunakan percabangan if dengan kondisi yaitu fungsi regex(preg_match) dengan regex yaitu

“/[a-zA-Z\d]/” atau pola yang dicari adalah special karakter pada variable password maka akan dilakukan :

- a) variable usernameErr akan memiliki nilai string “password tidak memiliki special character”.
- b) Variable x akan bernilai true.

- d. Percabangan IF untuk pemeriksaan password menggunakan angka atau digit

```
if (!preg_match("/[0-9]/", $password)) {  
    $passwordErr = "password tidak memiliki digit/angka";  
    $x = true;  
}
```

Pada bagian ini menggunakan dengan menggunakan percabangan if dengan kondisi yaitu fungsi regex(preg_match) dengan regex yaitu “/[0-9]/” atau pola yang dicari adalah angka atau digit pada variable password maka akan dilakukan :

- a) variable usernameErr akan memiliki nilai string “password tidak memiliki digit/angka”.
- b) Variable x akan bernilai true.

- 3. Jumlah karakter password tidak boleh kurang dari sepuluh karakter.

```
if($pass>10){  
    $passwordErr = "password lebih dari 10 karakter";  
    $x = true;  
}
```

Pada bagian ini menggunakan dengan menggunakan percabangan if dengan kondisi yaitu variable user lebih dari 10 karakter maka akan dilakukan :

- a) variable usernameErr akan memiliki nilai string “Password Lebih dari 7 karakter”.
- b) Variable x akan bernilai true.

- 4. Menampilkan hasil dari username dan password yang berhasil melewati validasi

```

if( $x == false ){
    $berhasil = "<h2>BERHASIL</h2>";
    $username_h = $_POST["username"];
    $password_h = $_POST["password"];
}

```

Pada bagian ini menggunakan dengan menggunakan percabangan if dengan kondisi yaitu variable variabel x bernilai false maka akan dilakukan :

- a) variable berhasil akan memiliki nilai string “BERHASIL”.
- b) Variable username_h akan menyimpan data username yang dimasukan pada form.
- c) Variable password_h akan menyimpan data password yang dimasukan pada form.

Pada bagian selajutnya terdapat HTML yang digunakan untuk menampilkan form pada browser. HTML untuk form adalah sebagai berikut :

```

<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<h2>FORM VALIDASI</h2>
<p><span class="error">* WAJIB DI ISI.</span></p>
<form method="post" action="<?php echo ($_SERVER["PHP_SELF"]);?>">
    Username : <input type="text" name="username">
    <span class="error">* <?php echo $usernameErr;?></span>
    <br><br>
    Password : <input type="text" name="password">
    <span class="error">* <?php echo $passwordErr;?></span>
    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>
<?php
echo $berhasil;
if( $x == false ){
echo "Username : ".$username_h;
echo "<br>";
echo "Password : ".$password_h;
}

```

```
?>
</body>
</html>
```

Berikut merupakan penjelasan dari setiap bagian-bagian HTML form:

a. Head HTML

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
```

Pada bagian Head terdapat style yang digunakan untuk mengatur kelas error agar berwarna #FF0000(Merah).

b. Body HTML

```
<body>
<h2>FORM VALIDASI</h2>
<p><span class="error">* WAJIB DI ISI.</span></p>
<form method="post" action="<?php echo
($_SERVER["PHP_SELF"]);?>">
  Username : <input type="text" name="username">
  <span class="error">* <?php echo $usernameErr;?></span>
  <br><br>
  Password : <input type="text" name="password">
  <span class="error">* <?php echo $passwordErr;?></span>
  <br><br>
  <input type="submit" name="submit" value="Submit">
</form>
<?php
echo $berhasil;
echo "Username : ".$username_h;
echo "<br>";
echo "Password : ".$password_h;
?>
</body>
</html>
```

Pada bagian ini dimulai dengan menampilkan heading 2 yaitu “FORM VALIDASI”. Pada bagian berikutnya terdapat tag span dengan class error yang menampilkan “* WAJIB DI ISI.”. selanjutnya terdapat <form method="post" action="<?php echo (\$_SERVER["PHP_SELF"]);?>">

menyimpan form tersebut ke dalam PHP itu sendiri. Kemudian terdapat text field, yaitu :

- a) Username : `<input type="text" name="username">` yang digunakan untuk memasukan username.
- b) Password : `<input type="text" name="password">` yang digunakan untuk memasukan password.

Disetiap text field terdapat span yang pertama adalah `*<?php echo $usernameErr;?>` yang digunakan untuk menampilkan error dari username dan kedua adalah `*<?php echo $passwordErr;?>`.

c. PHP pada HTML

```
<?php
echo $berhasil;
if( $x == false ){
echo "Username : ".$username_h;
echo "<br>";
echo "Password : ".$password_h;
}
?>
```

Pada bagian ini yang pertama akan menampilkan variable berhasil. Selanjutnya adalah menampilkan username yang disimpan oleh variable username_h dan password yang disimpan oleh variable password_h.

Berikut merupakan output program web untuk menginputkan username dan password menggunakan form dan penanganan input data adalah sebagai berikut :

- a. Tampilan awal web.

FORM VALIDASI

*** WAJIB DI ISI.**

Username : *

Password : *

Username :

Password :

Gambar 2.1 tampilan awal web

- b. Tampilan jika memasukan lebih dari 7 karakter.

FORM VALIDASI

*** WAJIB DI ISI.**

Username : * Username lebih dari 7 karakter

Password : * password tidak memiliki digit/angka

Username :

Password :

Gambar 2.2. tampilan jika username yang dimasukan lebih dari 7 karakter

- c. Tampilan jika password yang dimasukan tidak memiliki special karakter

FORM VALIDASI

*** WAJIB DI ISI.**

Username : *

Password : * password tidak memiliki special character

Username :

Password :

Gambar 2.3 tampilan jika password yang dimasukan tidak memiliki special karakter

- d. Tampilan jika password yang dimasukan tidak memiliki angka atau digit.

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : * password tidak memiliki digit/angka

Username :

Password :

Gambar 2.4 tampilan jika password yang dimasukan tidak memiliki digit/angka

- e. Tampilan jika password tidak memiliki huruf kecil.

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : * password tidak memiliki huruf kecil

Username :

Password :

Gambar 2.5 tampilan jika password yang dimasukan tidak memiliki huruf kecil

- f. Tampilan jika password tidak memiliki huruf kapital.

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : * password tidak memiliki huruf kapital

Username :

Password :

Gambar 2.6 tampilan jika password yang dimasukan tidak memiliki huruf kapital

- g. Tampilan jika password lebih dari 10 karakter.

FORM VALIDASI

*** WAJIB DI ISI.**

Username : *

Password : * password lebih dari 10 karakter

Username :

Password :

Gambar 2.7 tampilan jika password lebih dari 10 karakter

- h. Tampilan jika username dan password berhasil di validasi.

FORM VALIDASI

*** WAJIB DI ISI.**

Username : *

Password : *

BERHASIL

Username : aditya

Password : Aditya_1

Gambar 2.8 tampilan jika username dan password berhasil divalidasi

BAB III

KESIMPULAN

PHP adalah bahasa pemrograman server side karena PHP diproses pada komputer server. Ketika dipanggil dari web browser, program yang ditulis dengan PHP akan di-parsing di dalam web server oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke web browser. Pada program ini banyak menggunakan `$_POST` PHP yang digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan `method="post"`. `$_POST` juga banyak digunakan untuk melewati variable dan `$_GET` PHP bisa digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan `method="get"`.

DAFTAR PUSTAKA

Jayanti, S., & Raja, J. (2020). *Jago Ngoding Pemrograman Web dengan PHP* (Issue January).

Tim Dosen Algoritma Pemograman. *Modul Praktikum Pemograman Web I*. 2021. Palangka Raya. Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya (UPR).

Andre. (2014). *Tutorial HTML Lanjutan: Perbedaan tag Span dan tag Div*.
<https://www.duniailkom.com/belajar-html-pengertian-tag-span-dan-div/>
(Diakses 04 April 2021).

Andre. (2014). *Tutorial Form PHP Part 5: Cara Membuat Validasi Form PHP (fungsi isset dan empty)*.
<https://www.duniailkom.com/tutorial-form-php-cara-membuat-validasi-form-php-fungsi-isset-dan-empty/#:~:text=Untuk%20memeriksa%20apakah%20sebuah%20objek,jika%20variabel%20tersebut%20belum%20dibuat>
(Diakses 04 April 2021)

Huda, Nurul. 2020. *PHP : Belajar Regular Expression*.
<https://jagongoding.com/web/php/menengah/regular-expression/> (Diakses 04 April 2021)

LAMPIRAN

Fungsi	Deskripsi
<code>preg_match()</code>	Mencari kata/karakter yang sesuai dengan pola regex.
<code>preg_match_all()</code>	Mencari <i>semua</i> kata/karakter yang sesuai dengan pola regex.
<code>preg_replace()</code>	Mencari kata/karakter yang sesuai dengan pola regex, lalu menyimpannya dengan data baru.
<code>preg_grep()</code>	Mengembalikan kata/karakter yang sesuai dengan pola regex.
<code>preg_split()</code>	Membagi string menjadi sebuah array menggunakan pola regex.

Gambar 1.1 tabel fungsi-fungsi bawaan PHP untuk bekerja dengan regex

Karakter	Deskripsi
<code>\s</code>	karakter putih (spasi, tab, baris baru)
<code>\d</code>	angka digit (0-9)
<code>\w</code>	karakter huruf (a-z, A-Z, 0-9, _)
<code>[aeiou]</code>	pola yang terdiri dari set karakter di dalam kurung siku
<code>[^aeiou]</code>	pola yang tidak terdiri dari set karakter di dalam kurung siku
<code>(foo bar baz)</code>	pilihan dari kata yang didefinisikan

Gambar 1.2 tabel beberapa meta karakter regex yang sering digunakan

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : *

Username :

Password :

Gambar 2.1 tampilan awal web

FORM VALIDASI

* WAJIB DI ISI.

Username : * Username lebih dari 7 karakter

Password : * password tidak memiliki digit/angka

Username :

Password :

Gambar 2.2. tampilan jika username yang dimasukan lebih dari 7 karakter

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : * password tidak memiliki special character

Username :

Password :

Gambar 2.3 tampilan jika password yang dimasukan tidak memiliki special karakter

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : * password tidak memiliki digit/angka

Username :

Password :

Gambar 2.4 tampilan jika password yang dimasukan tidak memiliki digit/angka

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : * password tidak memiliki huruf kecil

Username :

Password :

Gambar 2.5 tampilan jika password yang dimasukan tidak memiliki huruf kecil

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : * password tidak memiliki huruf kapital

Username :

Password :

Gambar 2.6 tampilan jika password yang dimasukan tidak memiliki huruf kapital

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : * password lebih dari 10 karakter

Username :

Password :

Gambar 2.7 tampilan jika password lebih dari 10 karakter

FORM VALIDASI

* WAJIB DI ISI.

Username : *

Password : *

BERHASIL

Username : aditya
Password : Aditya_1

Gambar 2.8 tampilan jika username dan password berhasil divalidasi