

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



NAMA : Kevin Jeremia S.P
NIM : 193020503034
KELAS : A
MODUL : II (FORM HANDLING)

**JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021**

BAB I

TUJUAN DAN LANDASAN TEORI

1.1 Tujuan Praktikum

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

1.2 Landasan Teori

1.2.1 Get dan Post

Form pada PHP mempunyai dua method pengiriman data, yaitu GET dan POST. Dengan menggunakan method GET, maka nilai variabel yang dikirimkan ke server melalui url pada address bar browser. Sedangkan method POST akan mengirimkan nilai variabel ke server secara terpisah, sehingga nilai variabelnya tidak terlihat.

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 1.1 HTML dan PHP

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 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 1.2 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 1.3 HTML dan PHP

Berikut beberapa pertimbangan kapan sebaiknya Anda menggunakan method GET dan POST, diantaranya :

- Apabila data yang dikirim ke server bersifat rahasia, Anda dapat menggunakan method POST, misalnya untuk mengirimkan data username dan password.
- Apabila data yang dikirim ke server berupa data umum dan biasanya untuk memperjelas data yang dimasukkan di form, Anda dapat menggunakan method GET, misalnya untuk form pencarian data, polling dan lainnya.

- Jika data yang dikirimkan cukup banyak, gunakanlah method POST
- Jika data yang dikirimkan berupa file, gunakan method POST

1.2.2 Validasi Form PHP

Pertimbangkan keamanan ketika memproses form PHP!

PHP Form Validation Example

* required field.

Name: *

E-mail: *

Website:

Comment:

Gender: ☐ Female ☐ Male *

Gambar 1.4 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 1.5 Text Field

- **Form Element**

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

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

Gambar 1.6 From Element

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, fungsi tersebut akan mengkonversikan karakter `<` dan `>` menjadi `<` dan `>`. Fungsi ini mencegah injeksi yang bisa dilakukan dengan HTML atau javascript (Cross-site Scripting Attack) pada form tersebut.

BAB II

PEMBAHASAN

Pada praktikum kali ini, diminta untuk membuat sebuah 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.

Web yang akan di buat menggunakan dua buah Bahasa. Bahasa pertama untuk Bahasa yang akan mengatur algoritma dari web ini adalah PHP dan untuk tampilan nya menggunakan Bahasa HTML. Dalam pembuatan web ini kedua Bahasa ini akan disimpan kedalam sebuah file saja, file php. Berikut ini adalah penjelasan baris kode web ini yang akan menyelesaikan soal nomor 1 sampai 3.

Pertama yang akan di jelaskan adalah baris kode HTML. Baris kode HTML yang di gunakan di halaman web ini berfungsi untuk menginputkan username dan password di dalama web yang di buat. Berikut ini adalah baris kode dan penjelasan dari HTML nya.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Modul 2</title>
</head>
```

```

<body>
<form action="" method="post" name="Login_Form">
    <h3>Login</h3>
    <p> Username : <input name="username" type="text" placeholder
="Username"></p>
    <p>Password : <input name="password" type="text" placeholder="
Password" ></p>
    <p><input name="Submit" type="submit" value="Login"></p>
</form>
</body>
</html>

```

Baris kode diatas memiliki, head dan body. Di dalam head, terdapat title Modul2 lalu setelah itu penutup head. Lalu body, di dalam body terdapat form, element yang digunakan untuk menampung macam-macam element yang berkaitan dengan sebuah form. Form HTML dibentuk menggunakan tag, dan tag form memiliki attribute method yang digunakan untuk menentukan bagaimana cara form mengirim data ke server (data dikirim ke halaman yang diset di dalam attribute action).

Kemudian di dalam form tersebut terdapat sebuah tempat untuk menginputkan username dan password dan tombol login. Untuk menginputkan username baris kodenya dimulai dengan input yang merupakan elemen dari form. Input berfungsi untuk membuat tampilan Ketika login nya. name="username" di gunakan untuk menyimpan username yang di tuliskan user dan di masukan kedalam PHP. Untuk type="text" memiliki fungsi untuk membuat Ketika melakukan typing yang akan muncul huruf atau karakter yang sesuai dengan yang di masukan. placeholder="Username", merupakan sebuah tulisan yang terdapat di dalam box pengisian username.

Setelah itu untuk menginputkan password, baris kode yang di pakai sama saja seperti baris kode membuat penginputan username. Namun yang berbeda hanyalah nama yang akan di panggil di PHP di ubah menjadi Password:

<input name="password". Setelah itu untuk tombol login type yang di gunakan submit dengan valuenya adalah login. Setelah itu di baris kode untuk menutup form, body, dan html.

Dilanjutkan untuk pembahasan Bahasa PHP. PHP digunakan sebaga penanganan input data, dimana setiap username dan password yang akan di inputkan haruslah sesuai dengan kondisi kondisi yang sesuai soal diatas sebelum nya. Berikut ini adalah penjelasan dari baris kode nya.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $username = $_POST["username"];
    $password = $_POST["password"];
    $user = strlen($username);
    $pass = strlen($password);
    $a = false;
```

Baris program diatas di buka dengan tag php, yang artinya Bahasa php akan di gunakan. Setelah itu terdapat kondisi percabangan if dengan parameter \$_SERVER ["REQUEST_METHOD"] == "POST" sebagai kondisi dari percabangan ini . Fungsi dari baris program ini adalah jika request_method yang digunakan adalah post maka sintaks tersebut akan dijalankan. Bari program yang akan di jalankan di bawah nya ada sebua variabel \$username dan \$ password. Variable ini memiliki nilai yang berasal dari html. Jadi variabel tersebut akan mengambil nilai dari usernamedan passorwd yang telah disubmit pada bagian html.

Sesudah itu terdapat \$user dan \$pass yang memiliki nilai strlen dengan parameter \$username dan \$password. Dimana strlen berfungsi untuk menghitung panjag string oleh sebab itu mengapa parameter dari strlen ini variabel \$username dan \$password. \$user dan \$pass ini berfungsi menghitung Panjang dari string, dimana variable ini dapat di gunakan sebagai sebuah kondisi di baris program selanjutnya. Lalu terdapat variable \$a yang bertipe data bool dan bernilai false yang akan di

gunakan nanti juga pada saat membuat percabangan. Berikut ini setiap percabangan dan masing masing pembahasan nya

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

Pembahasan Percabangan pertama adalah sebuah percabangan yang memiliki parameter \$user>7 yang artinya variable yang digunakan dapat menghitung Panjang string. Artinya Ketika Panjang string kurang dari 7 maka akan menampilkan peringatan Username lebih dari 7 dan variable \$a =true; menyatakan bahwa kondisi yang salah ini bernilai True. Kondisi memiliki fungsi untuk menghitung Panjang username yang di input oleh user.

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

Selanjutnya baris program diatas berfungsi untuk mengecek apakah password yang di inputkan oleh user harus lah terdapat huruf besar/kapital. Fungsi preg_match digunakan untuk mengecek apakah suatu kecocokan terdapat dalam string atau tidak. Preg_match memiliki paramterer"/[A-Z]/". Yang berfungsi untuk mencocokkan apakah ada sebuah huruf kapital yang terdapat dari password yang di inputkan oleh user. Jika kondisi terpenuhi yang artinya tidak terdapat huruf kapital maka kondisi ini akan melakukan echo dan menampilkan peringatan kepada user. \$a =true; menyatakan bahwa kondisi yang salah ini bernilai True

```

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

```

Berikutnya adalah baris program diatas berfungsi untuk mengecek apakah password yang di inputkan oleh user harus lah terdapat huruf kecil. Fungsi preg_match digunakan untuk mengecek apakah suatu kecocokan terdapat dalam string atau tidak. Preg_match memiliki paramterer"/[a-z]/". Yang berfungsi untuk mencocokkan apakah ada sebuah huruf kecil yang terdapat dari password yang di inputkan oleh user. Jika kondisi terpenuhi yang artinya tidak terdapat huruf kecil maka kondisi ini akan melakukan echo dan menampilkan peringatan kepada user. \$a =true; menyatakan bahwa kondisi yang salah ini bernilai True

```

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

```

Dan ini baris program berfungsi untuk mengecek apakah password yang di inputkan oleh user harus lah terdapat angka 0-9. Fungsi preg_match digunakan untuk mengecek apakah suatu kecocokan terdapat dalam string atau tidak. Preg_match memiliki paramterer"/[0-9]/". Yang berfungsi untuk mencocokkan apakah ada sebuah angka yang terdapat dari password yang di inputkan oleh user. Jika kondisi terpenuhi yang artinya tidak terdapat angka maka kondisi ini akan melakukan echo dan menampilkan peringatan kepada user. \$a =true; menyatakan bahwa kondisi yang salah ini bernilai True

```

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

```

```
    $a = true;
}
```

Baris program diatas berfungsi untuk mengecek apakah password yang di inputkan oleh user harus lah terdapat Special karakter seperti !, @, #, \$. Fungsi preg_match digunakan untuk mengecek apakah suatu kecocokan terdapat dalam string atau tidak. Preg_match memiliki parameter "/[^a-zA-Z\d]/". Yang berfungsi untuk mencocokkan apakah ada sebuah special yang terdapat dari password yang di inputkan oleh user. Jika kondisi terpenuhi yang artinya tidak terdapat special character maka kondisi ini akan melakukan echo dan menampilkan peringatan kepada user. \$a = true; menyatakan bahwa kondisi yang salah ini bernilai True

```
if($pass<10){
    echo "<script>alert('password kurang dari 10');</script>";
    $a = true;
}
```

Baris kode diatas adalah sebuah percabangan yang memiliki parameter \$pass<10 yang artinya variable yang digunakan dapat menghitung Panjang string. Artinya Ketika Panjang string kurang dari 10 maka akan menampilkan peringatan Password kurang dari 10 dan variable \$a = true; menyatakan bahwa kondisi yang salah ini bernilai True. Kondisi memiliki fungsi untuk menghitung Panjang username yang di input oleh user.

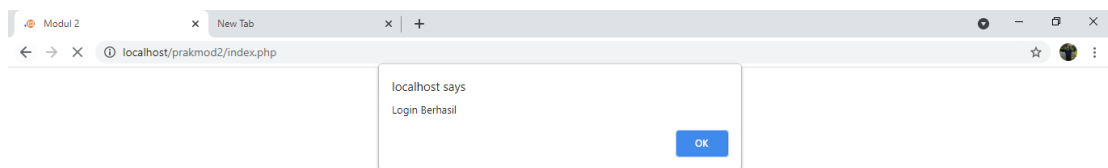
```
if( $a == false ){
    echo "<script>alert('Login Berhasil');</script>";
}
```

Dan Perbangan terakhir ini memiliki parameter \$a == false yang artinya Ketika tidak ada kesalahan yang terjadi / bernilai true maka login dinyatakan berhasil

dengan menampilkan alert Login Berhasil yang di sudah di echo. Berikut ini adalah tampilan web dari login dan Ketika login berhasil.



Gambar2.1 Halaman web login



Gambar2.2 Login Berhasil

BAB III

Kesimpulan

Kesimpulan dari praktikum ini adalah menggunakan form handling dalam membuat sebuah program login dapat membantu program untuk terhindar dari kesalahan dan membuat program dapat berjalan dengan baik serta membuat login menjadi aman. Selain itu GET dan POST jika di gunakan pada kondisi yang tepat maka akan memiliki kegunaan nya masing masing.

DAFTAR PUSTAKA

Praktikum, K. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

Request, H., Nama, G. E. T., Praktis, P., Php, P., & Pemula, U. (2000). *Bab 5 memahami method get dan post pada php*. 18–22.

LAMPIRAN

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

Gambar 1.1 HTML dan PHP

```
<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 1.2 HTML dan PHP

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

Gambar 1.3 HTML dan PHP

PHP Form Validation Example

* required field.

Name: *

E-mail: *

Website:

Comment:

Gender: ☐ Female ☐ Male *

Gambar 1.4 VALIDASI

Name: `<input type="text" name="name">`

E-mail: `<input type="text" name="email">`

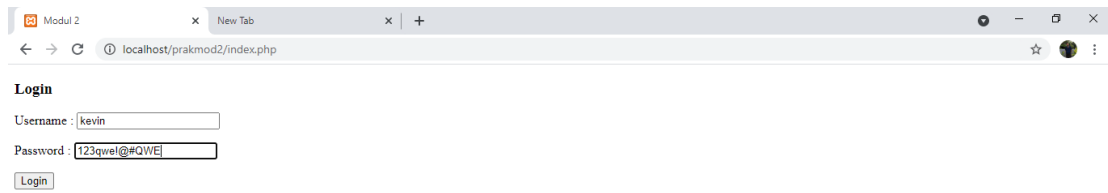
Website: `<input type="text" name="website">`

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

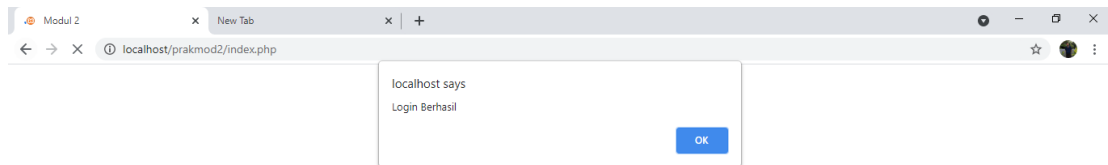
Gambar 1.5 Text Field

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

Gambar 1.6 From Element



Gambar2.1 Halaman web login



Gambar2.2 Login Berhasil