

LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB & MOBILEI



NAMA : FIRDAUS ZAILANI

NIM : 193020503040

KELAS : A

MODUL : II (Form Handling)

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PALANGKA RAYA

2021

BAB I

TUJUAN DAN LANDASAN TEORI

A. Tujuan

- Mahasiswa mampu membuat handling yang mampu mengolah data dari form HTML
- 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>
```

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 ton@mail.com maka output yang akan tampil adalah sebagai berikut:

Welcome Budi

Your email address is ton@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>
```

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>
```

- **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 *

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, 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.

- **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"];?>">
```

Kemudian user memasukkan URL pada address bar dengan alamat sebagai berikut:

```
http://localhost/<nama_folder>/test_form.php/%22%3E%3Cscript%3Eale  
t('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”.

Berhati-hatilah dengan kemungkinan penambahan kode javascript pada tag <script>!

Hacker bisa mengarahkan user ke file pada server yang lain, dan file itu bisa 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(). 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/"><script>alert('hacked')</script>";>
```

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, 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;
    }
?>
```

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 = "";
    } 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 field input 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
```

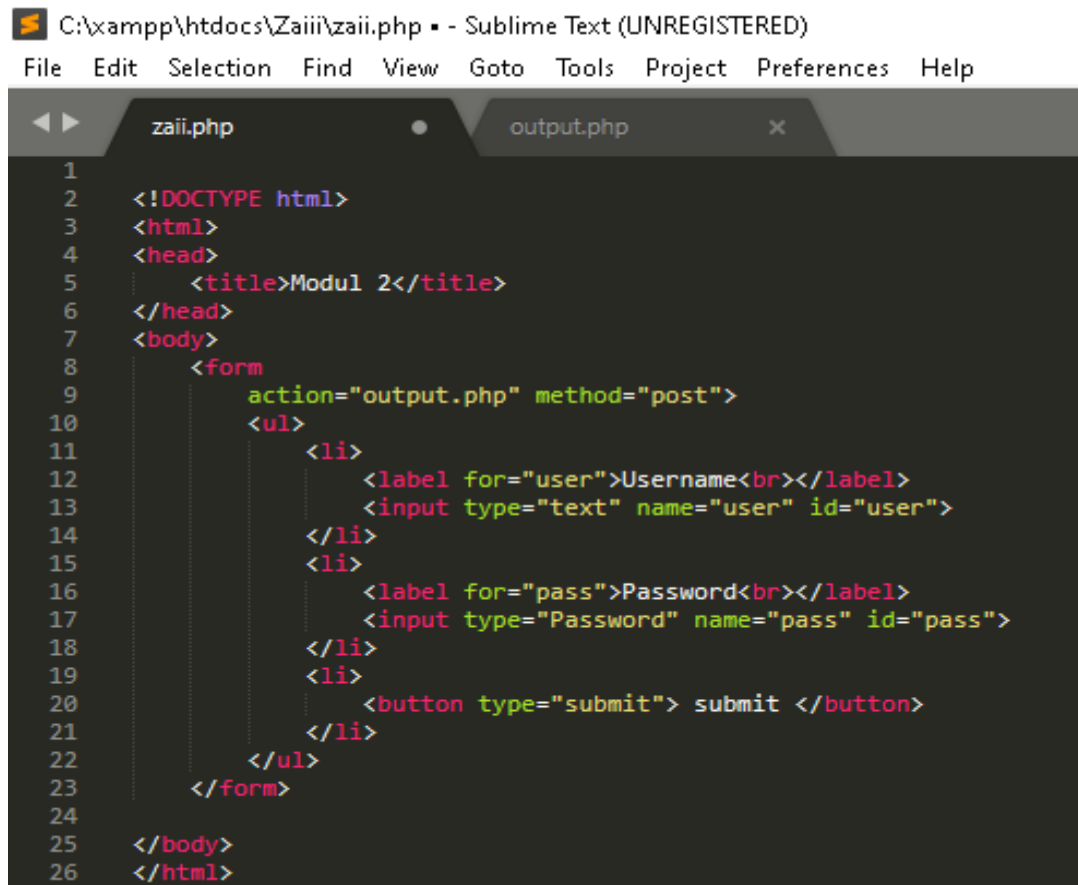
TUGAS

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.

BAB II

PEMBAHASAN



```
1
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>Modul 2</title>
6  </head>
7  <body>
8      <form
9          action="output.php" method="post">
10         <ul>
11             <li>
12                 <label for="user">Username<br></label>
13                 <input type="text" name="user" id="user">
14             </li>
15             <li>
16                 <label for="pass">Password<br></label>
17                 <input type="Password" name="pass" id="pass">
18             </li>
19             <li>
20                 <button type="submit"> submit </button>
21             </li>
22         </ul>
23     </form>
24
25 </body>
26 </html>
```

Gambar 1.1. *Sourcecode zaii.php.*

Pada gambar *zaii.php* baris 2 `<!DOCTYPE html>` berfungsi sebagai mendeklarsi untuk mendefinisikan dokumen menjadi HTML. Pada Source Code di atas merupakan untuk memunculkan judul pada sebuah browser `<!DOCTYPE html>` adalah sebuah deklarasi atau untuk mengidentifikasi jenis dokumen HTML yang digunakan agar browser dapat menentukan bagaimana memperlakukan kode tersebut. title bar adalah menampilkan judul sebuah program/aplikasi/web.

Sedangkang pada baris 3 `<html> </html>` adalah tag pembuka untuk membuat dokumen HTML. Pada baris ke 4 dan `<head> </head>` informasi meta tentang dokumen. Pada baris ke 7 `<body></body>` berfungsi sebagai tempat dibuatnya semua konten webside menggunakan hmtl. Selanjutnya `<form` untuk membuat formulir untuk mengumpulkan input pengguna dan penanganan action

yang kedua adalah diproses pada halaman lain. Carra ini dilakukan dengan cara mengisi parameter action sesuai dengan nama file yang terletak di script PHP contoh nya seperti pada baris 9: “ action="output.php" method="post"> “..Pada Source Code di atas berfungsi untuk mengirim data ke halam output.php (action). <head> berfungsi sebagai Informasi meta tentang dokumen. <body> berfungsi sebagai Tempat dibuatnya semua konten website menggunakan HTML. <form> adalah Membuat formulir untuk mengumpulkan input pengguna.Action berfungsi sebagai untuk mendefinisikan tindakan apa yang akan dilakukan ketika formulir dikirimkan.Action ini mau dieksekusi ke mana data itu ketika tombol Login diklik.

Selanjutnya pada baris ke 10 untuk membuat tampilan daftar menggunakan symbol seperti bullet, cyrcle dan lainnya. Sedangkan menentukan berbagai item yang ingin ditampilkan. Pada baris 12,13,16,17 untuk meampilkan sebuah text atau kolom yang nantinya akan ditampilkan di situs (localhost, web). Username yang artinya memasukan username atau id dan password adalah password dari pengguna, password dapat ditampilkan atau pun di sembunyikan dengan cara mengganti pada baris 17 text “password” jika ingin menampilkan maka di gati “text”. Pada Source Code diatas berfungsi untuk memasukkan/input Username ke dalam sebuah website. berfungsi Untuk membuat tampilan daftar menggunakan simbol seperti bullet, cyrcle, dan lainnya. berfungsi untuk Menentukan berbagai item yang ingin ditampilkan. <label> untuk Memberikan label pada elemen input.For merupakan sebuah perulangan.<label for="user">Username
</label> fungsi untuk menampilkan text Username pada website tersebut. <input> adalah Membuat tipe inputan pada form yang akan dibuat.Type berfungsi untuk menentukan jenis text.Name adalah Agar data yang dikirimkan bisa diterima dengan valid, maka masing-masing bidang input harus memiliki name atribut. berfungsi Menentukan berbagai item yang ingin ditampilkan.id berfungsi Memberikan id pada tabel atau kolom Pada Source Code diatas berfungsi untuk memasukkan/input Password ke dalam sebuah website. berfungsi Untuk membuat tampilan daftar menggunakan simbol seperti bullet, cyrcle, dan lainnya. berfungsi untuk Menentukan berbagai item yang ingin ditampilkan. <label> untuk Memberikan label pada elemen input.For merupakan sebuah perulangan.<label for="pass">Password
</label>fungsi

untuk menampilkan text Password pada website tersebut. <input> adalah Membuat tipe inputan pada form yang akan dibuat. Type berfungsi untuk menentukan jenis text. Name adalah Agar data yang dikirimkan bisa diterima dengan valid, maka masing-masing bidang input harus memiliki name atribut. berfungsi Menentukan berbagai item yang ingin ditampilkan. id berfungsi Memberikan id pada tabel atau kolom.

Dan pada baris 20 adalah sama halnya seperti baris sebelumnya hanya saya nanti setelah selesai memasukan username dan password maka dapat mengklik pada bagian ini dan nantinya ada di konfirmasi apakah username dan password dimasukan benar apa salah. Pada Source Code diatas adalah untuk menampilkan button Login pada sebuah website. <button> berfungsi Membuat Button/tombol. Type adalah menentukan jenis penomoran untuk button yaitu digunakan type submit. Login adalah title untuk Submit pada text tersebut. berfungsi menentukan berbagai item yang ingin ditampilkan. berfungsi Untuk membuat tampilan daftar menggunakan simbol seperti bullet, cyrcle, dan lainnya. <form> adalah Membuat formulir untuk mengumpulkan input pengguna. <body> berfungsi Tempat dibuatnya semua konten website menggunakan HTML. <html> berfungsi Tag pembuka untuk membuat dokumen HTML.



```
C:\xampp\htdocs\Zaiii\output.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

zail.php x output.php x
1 <?php
2
3 if ($_SERVER["REQUEST_METHOD"] == "POST") {
4     $user = $_REQUEST["user"];
5     $pass = $_POST["pass"];
6     $user = strlen($user);
7     $pass = strlen($pass);
8     $x = false;
9
10    if ($_user > 7) {
11        echo "Username harus kurang dari 7 karakter<br>";
12        $x = true;
13    }
14
15    if (!preg_match("/[A-Z]/", $pass)) {
16        echo "Password harus terdapat huruf kapital<br>";
17        $x = true;
18    }
19
20    if (!preg_match("/[a-z]/", $pass)) {
21        echo "Password harus terdapat huruf kecil<br>";
22        $x = true;
23    }
24
25    if (!preg_match("/[0-9]/", $pass)) {
26        echo "Password harus terdapat karakter khusus (angka)<br>";
27        $x = true;
28    }
29
30    if ($_pass < 10) {
31        echo "Password harus lebih dari 10 karakter<br>";
32        $x = true;
33    }
34
35    if ($x == false) {
36        echo "Username dan Password memenuhi syarat";
37    }
38
39 }
40
41
```

Gambar 1.2. Sourcecode output.php.

Pada gambar diatas baris ke 3-8 Pada Source Code diatas if (`$_SERVER["REQUEST_METHOD"] == "POST"`) adalah itu kode nya menggunakan operator comparison yaitu '=='. jadi itu bacanya apakah `$_SERVER["REQUEST_METHOD"]` sama dengan POST.maka hasilnya antara true atau false begitu juga dengan isset. `$username = $_POST["user"]`; memberikan nilai pada variable agar tidak kepanjangan. `$pass = $_POST["pass"]`;sama seperti user memberikan nilai pada variable agar tidak kepanjangan. `strlen()` fungsi menghitung panjang string.\$ merupakan variable.jadi `$_user = strlen($user)`; berfungsi untuk menghitung Panjang string pada username. `$_pass = strlen($pass)`; berfungsi untuk menghitung Panjang string pada password. `$x = false`; berfungsi untuk mengecek apakah username dan password benar dimasukkan/salah di masukkan.

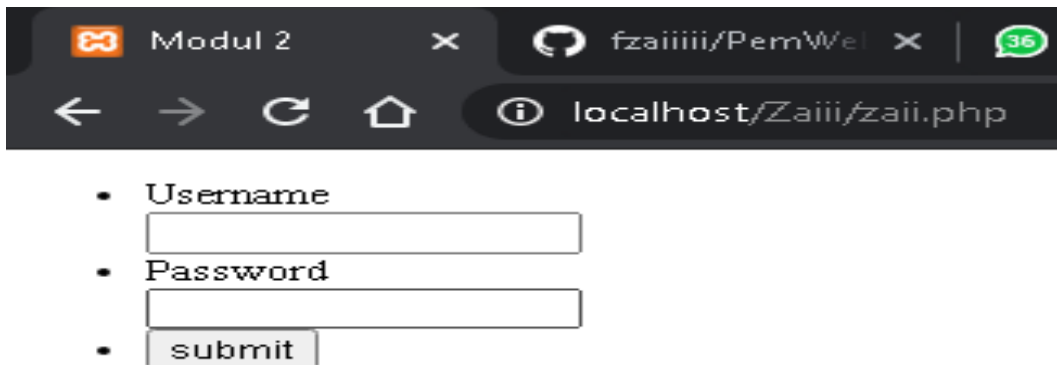
Pada baris 10-38 adalah syarat untuk memasukan password Pada Source Code diatas adalah berfungsi jika username kurang dari 7 kata maka tidak bisa login ke website tersebut.Jika memasukkan username lebih 7 kata maka berhasil login dan bernilai false . If digunakan untuk satu kondisi saja. Jika pernyataan benar (terpenuhi) maka akan dijalankan, jika salah (tidak terpenuhi) maka abaikan. `echo()` adalah fungsi untuk menampilkan teks ke layar. Fungsi ini dapat digunakan dengan tanda kurung maupun tanpa tanda kurung. “Password harus terdapat huruf kapital”.Jika berhasil ada huruf kapital maka berhasil login. `preg_match()` digunakan untuk dapat mengembalikan apakah kecocokan ditemukan dalam string.`echo()` adalah fungsi untuk menampilkan teks ke layar. Fungsi ini dapat digunakan dengan tanda kurung maupun tanpa tanda kurung.`$x = true` berfungsi mengecek nilai true/false. “Password harus terdapat huruf kecil”.Jika berhasil ada huruf kecil maka berhasil login. `echo()` adalah fungsi untuk menampilkan teks ke layar. `.$x = true` berfungsi mengecek nilai true/false. Terdapat apakah ada karakter khusus/tidak.Jika ada maka akan berhasil login ke dalam website tersebut.Jika password tidak ada karakter khusus maka tidak bias login ke dalam website dan ada tulisan “Password harus terdapat karakter khusu”. `echo()` adalah fungsi untuk menampilkan teks ke layar.`$x = true` berfungsi mengecek nilai true/false. untuk mengecek password lebih dari 10 karakter/kurang.Jika kurang dari 10 karakter maka tidak bisa login dan akan muncul tulisan “Password harus lebih dari 10

karakter”.Jika memasukkan karakter lebih dari 10 karakter maka berhasil login. echo() adalah fungsi untuk menampilkan teks ke layar.\$x = true berfungsi mengecek nilai true/false. untuk jika berhasil login maka akan muncul tulisan seperti”Username dan Password memenuhi syarat” dan jika gagal login maka tidak ada tulisan seperti ”Username dan Password memenuhi syarat” echo() adalah fungsi untuk menampilkan teks ke layar.\$x = false berfungsi mengecek nilai true/false.Jika nilai true maka login berhasil,Jika false maka gagal login ke website tersebut.

```
42 <!DOCTYPE html>
43 <html>
44 <head>
45     <title></title>
46 </head>
47 <body>
48     <br>
49     <a href="zaii.php">Kembali Ke Menu
50     <br>
51     <a href="zaii.php">Log out</a>
52 </body>
53 </html>
```

Gambar 1.3. Sourcecode output.php.

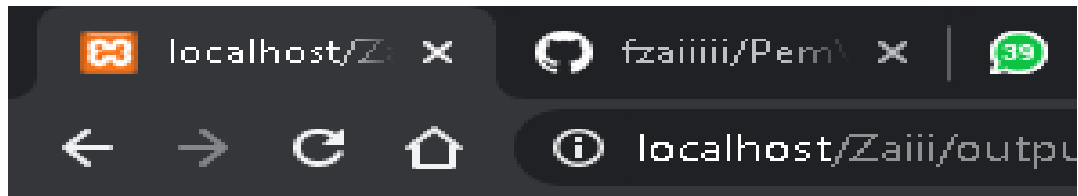
Pada baris 49 dan 51 adalah perintah yang nantinya ditampilkan di website jika selesai memasukan username dan password salah ataupun benar. Tampilannya seperti pada “ Gambar 1.5. dan Gambar 1.6.”.



- Username
- Password
-

Gambar 1.4. Tampilan Awal Login.

Pada gambar diatas menampilkan awal login yang dimana kita disuruh agar memasukan username dan password, jika memasukan username dan password benar dan juga memenuhi syarat maka akan seperti gambar berikut.



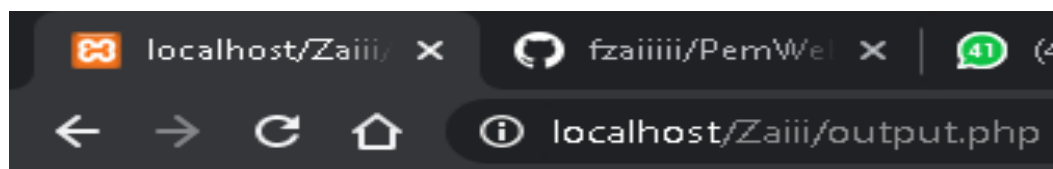
Username dan Password memenuhi syarat

[Kembali Ke Menu](#)

[Log_out](#)

Gambar 1.5. *Login Berhasil.*

Dan jika salah memasukkan username dan password atau pun tidak memenuhi syaratnya maka akan tampil sebagai berikut.



Password harus terdapat huruf kapital

Password harus terdapat karakter khusus (angka)

Password harus lebih dari 10 karakter

[Kembali Ke Menu](#)

[Log_out](#)

Gambar 1.6. *Login Gagal.*

Diatas menampilkan bahwa username dan password yang dimasukkan salah ataupun tidak memenuhi syarat, setelah itu kita dapat mengklik “Kembali Ke Menu/Log out” untuk mengulang(masuk kembali).

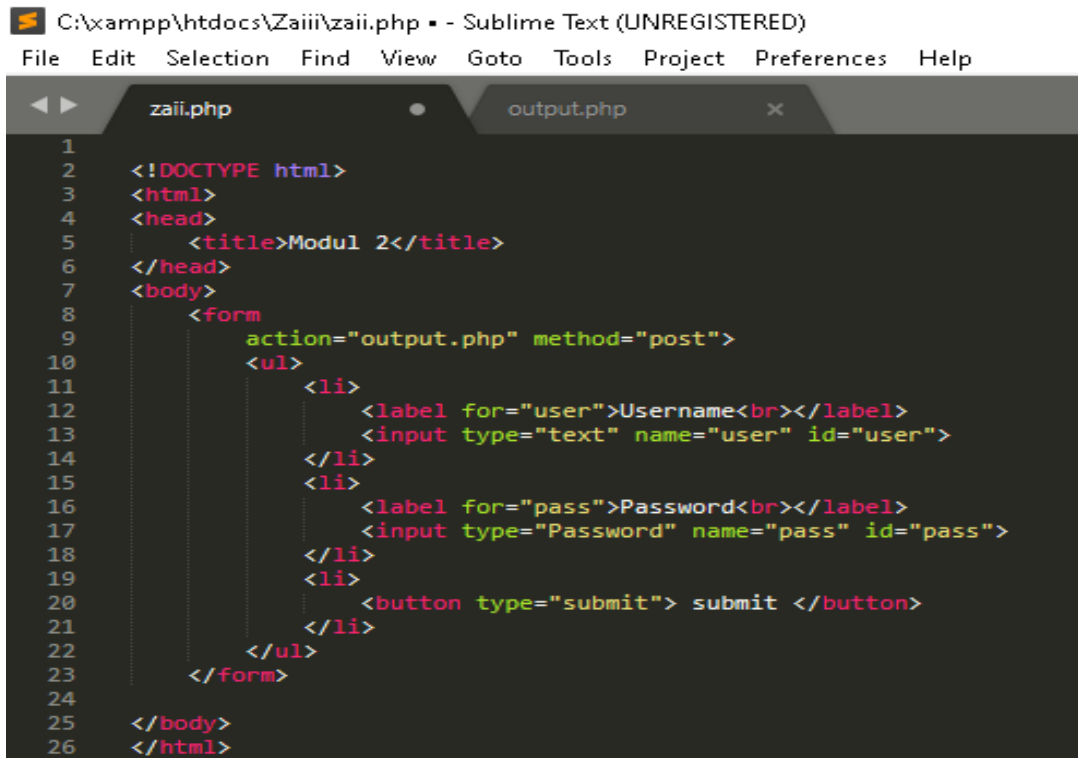
KESIMPULAN

Dari praktikum yang sudah saya kerjakan Form Handling adalah suatu aplikasi banyak menerima masukan/input data dari pengguna, misalkan ketika registrasi kesuatu website atau ketika login pada halaman email dan sebagainya. Untuk memungkinkan hal tersebut diperlukan suatu antarmuka pada klien dalam bentuk form HTML dan suatu mekanisme untuk penanganan data yang dikirim oleh pengguna pada sisi server. Form handling ialah suatu mekanisme untuk menangani suatu masukan dari form yang dikirim oleh pengguna. Form handling berhubungan dengan metode yang dikirim dan bagaimana menangani pengiriman data berdasarkan metode yang digunakan. Pada bab ini akan membahas mengenai form HTML dan elemennya beserta implementasinya, juga form handling baik data teks maupun file.

DAFTAR PUSTAKA

- Guntoro. 17 Januari 2021. *65 Tag HTML dilengkapi Fungsi dan Contoh Penggunaannya*. <https://badoystudio.com/tag-html/>. (Diakses Pada Tanggal 04 April 2021 Pukul 23:40 WIB).
- Noname. 25 Februari 2017. *Fungsi Action Dalam Form PHP*. <http://info-chana.blogspot.com/2017/02/fungsi-action-dalam-form-php.html> . (Diakses Pada Tanggal 04 April 2021 Pukul 23:34 WIB).
- Reza Indra. 04 September 2014. *Perbedaan Metode POST Dan GET*. <https://www.dumetschool.com/blog/Perbedaan-Metode-POST-Dan-GET#:~:text=Method%20POST%20akan%20mengirimkan%20data,POST%20untuk%20menampung%20data%2Fnilai>. (Diakses Pada Tanggal 04 April 2021 Pukul 23:40 WIB).


LAMPIRAN



```
C:\xampp\htdocs\Zaiii\zaii.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

zaii.php
1
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title>Modul 2</title>
6 </head>
7 <body>
8   <form
9     action="output.php" method="post">
10    <ul>
11      <li>
12        <label for="user">Username<br></label>
13        <input type="text" name="user" id="user">
14      </li>
15      <li>
16        <label for="pass">Password<br></label>
17        <input type="Password" name="pass" id="pass">
18      </li>
19      <li>
20        <button type="submit"> submit </button>
21      </li>
22    </ul>
23  </form>
24
25 </body>
26 </html>
```

Gambar 1.1. *Sourcecode zaii.php.*



```
C:\xampp\htdocs\Zaiii\output.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

zaii.php x output.php x
1 <?php
2
3 if ($_SERVER["REQUEST_METHOD"] == "POST") {
4   $user = $_REQUEST["user"];
5   $pass = $_POST["pass"];
6   $user = strlen($user);
7   $pass = strlen($pass);
8   $x = false;
9
10  if ($user > 7) {
11    echo "Username harus kurang dari 7 karakter<br>";
12    $x = true;
13  }
14
15  if (!preg_match("/[A-Z]/", $pass)) {
16    echo "Password harus terdapat huruf kapital<br>";
17    $x = true;
18  }
19
20  if (!preg_match("/[a-z]/", $pass)) {
21    echo "Password harus terdapat huruf kecil<br>";
22    $x = true;
23  }
24
25  if (!preg_match("/[0-9]/", $pass)) {
26    echo "Password harus terdapat karakter khusus (angka)<br>";
27    $x = true;
28  }
29
30  if ($pass < 10) {
31    echo "Password harus lebih dari 10 karakter<br>";
32    $x = true;
33  }
34
35  if ($x == false) {
36    echo "Username dan Password memenuhi syarat";
37  }
38 }
39
40
41 >>
```

Gambar 1.2. *Sourcecode output.php.*

```

42 <!DOCTYPE html>
43 <html>
44 <head>
45     <title></title>
46 </head>
47 <body>
48     <br>
49     <a href="zaii.php">Kembali Ke Menu
50     <br>
51     <a href="zaii.php">Log out</a>
52 </body>
53 </html>

```

Gambar 1.3. Sourcecode output.php.

Modul 2 x fzaiiiii/PemWel x 36

localhost/Zaiii/zaii.php

- Username
- Password
-

Gambar 1.4. Tampilan Awal Login.

localhost/Z x fzaiiiii/PemWel x 39

localhost/Zaiii/output.php

Username dan Password memenuhi syarat

Kembali Ke Menu

Log_out

Gambar 1.5. Login Berhasil.

localhost/Zaiii/ x fzaiiiii/PemWel x 41

localhost/Zaiii/output.php

Password harus terdapat huruf kapital

Password harus terdapat karakter khusus (angka)

Password harus lebih dari 10 karakter

Kembali Ke Menu

Log_out

Gambar 1.6. Login Gagal.