

LAPORAN HASIL PRAKTIKUM

PEMROGRAMAN WEBSITE DAN MOBILE I



NAMA : MUHAMMAD ARIFIN

NIM : 193020503025

KELAS : A

MODUL : II

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PALANGKA RAYA

2021

BAB I

TUJUAN DAN LANDASAN TEORI

1. Tujuan Praktikum

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

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>
</html>
```

Gambar 1.1 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>
```

Gambar 1.2 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 1.3 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 1.6 HTML dan PHP

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 *

Gambar 1.7 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 1.8 Text Field

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
```

Gambar 1.9 Text Radio Button

Form Element

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

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

Gambar 1.10 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.

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"];?>">
```

Gambar 1.11 PHP

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

http://localhost/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>
```

Gambar 1.12 PHP

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

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/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;";>
```

Gambar 1.13 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;
}

?>
```

Gambar 1.14 PHP

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 = "";
    }
}
```

Gambar 1.15 PHP

```

    } 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"]);
    }
}
?>

```

Gambar 1.16 PHP

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

```

Gambar 1.17 PHP

```
<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>
```

Gambar 1.18 PHP

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";  
}
```

Gambar 1.19 Validasi Nama

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";  
}
```

Gambar 1.20 Validasi Email

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):\\W|www\\.)([-a-z0-9+&@#V%?=_~!|:,;]*[-a-z0-9+&@#V%?=_~!|:,;])/i",$website)) {  
    $websiteErr = "Invalid URL";  
}
```

Gambar 1.21 Validasi 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>`.

Gambar 1.22 Validasi URL

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
```

Gambar 1.23 Validasi URL.

BAB II

PEMBAHASAN

Tugas Praktikum

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.

Full code :

```
<?php

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

    $username = $_REQUEST["username"];

    $password = $_REQUEST["password"];

    $user = strlen($username);

    $pass = strlen($password);

    $x = false;

    if($user>7){

        echo "<script>alert('Username lebih dari 7');</script>";

        $x = true;
```

```
}

if (!preg_match("/[A-Z]/", $password) ) {

    echo "<script>alert('Password kapital');</script>";

    $x = true;

}

if (!preg_match("/[a-z]/", $password)) {

    echo "<script>alert('Password kecil');</script>";

    $x = true;

}


if (!preg_match("/^[a-zA-Z\d]/", $password)) {

    echo "<script>alert('Password special character');</script>";

    $x = true;

}


if (!preg_match("/[0-9]/", $password)) {

    echo "<script>alert('Password angka');</script>";

    $x = true;

}

if($pass<10){
```

```
        echo "<script>alert('Password kurang dari 10');</script>";

        $x = true;

    }

    if( $x == false ){

        echo "<script>alert('Berhasil');</script>";

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        body {

            background: -webkit-linear-gradient(bottom, #2dbd6e, #a6f77b);

            background-repeat: no-repeat;
```

```
        height : 600px;

    }

</style>

</head>

<body>

<form action="" method="post" name="Login_Form">

    <table border="0" align="center">

        <tr>

            <td colspan="2" align="center"></td>

        </tr>

        <tr>

            <td colspan="2" align="center"><h3>LOGIN</h3></td>

        </tr>

        <tr>

            <td align="right">Username : </td>

            <td><input name="username" type="text" placeholder="Username"></td>

        </tr>

        <tr>

            <td align="right">Password : </td>
```



```
<td><input name="password" type="password" placeholder="Password"
></td>

</tr>

<tr>

<td> </td>

<td><input name="Submit" type="submit" value="Login"></td>

</tr>

</table>

</form>

</body>

</html>
```

Pembahasan

Request username dan password.

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

    $username = $_REQUEST["username"];

    $password = $_REQUEST["password"];

    $user = strlen($username);

    $pass = strlen($password);

    $x = false;
```

Sintaks di atas menerangkan jika server meminta atau request method nya adalah post, maka data username dan password yang bisa melakukan login diambil dari hasil inputan user via web sebelumnya. Inputan tersebut bisa melakukan login dengan mempertimbangkan terpenuhinya aturan yang diminta pada tugas yang diaplikasikan ke dalam bentuk if pada bagian handling. Adanya strlen berfungsi untuk menghitung panjangnya inputan string pada username dan password yang telah diinput oleh user.

Handling 1

Pada handling ini, kondisi yang harus dipenuhi ialah panjang string atau inputan dari username harus lebih dari 7. Jika kondisi tersebut terpenuhi maka akan muncul notifikasi bahwa “Username lebih dari 7”.

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

Handling 2

Pada handling ini, kondisi yang harus dipenuhi ialah panjang string atau inputan dari password benar berupa huruf kapital, preg_match digunakan untuk mengecek apakah nilai balik memiliki kecocokan dengan string. Jika kondisi tersebut terpenuhi maka akan muncul notifikasi bahwa “Password kapital”.

```
if (!preg_match("/[A-Z]/", $password) ) {  
  
    echo "<script>alert('Password kapital');</script>";  
  
    $x = true;
```

```
}
```

Handling 3

Pada handling ini, kondisi yang harus dipenuhi ialah panjang string atau inputan dari password benar berupa huruf kecil, `preg_match` digunakan untuk mengecek apakah nilai balik memiliki kecocokan dengan string. Jika kondisi tersebut terpenuhi maka akan muncul notifikasi bahwa “Password kecil”.

```
if (!preg_match("/[a-z]/", $password)) {  
  
    echo "<script>alert('Password kecil');</script>";  
  
    $x = true;  
  
}
```

Handling 4

Pada handling ini, kondisi yang harus dipenuhi ialah panjang string atau inputan dari password benar berupa karakter, `preg_match` digunakan untuk mengecek apakah nilai balik memiliki kecocokan dengan string. Jika kondisi tersebut terpenuhi maka akan muncul notifikasi bahwa “Password special character”.

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

Handling 5

Pada handling ini, kondisi yang harus dipenuhi ialah panjang string atau inputan dari password benar berupa angka, `preg_match` digunakan untuk mengecek apakah nilai balik memiliki kecocokan dengan string. Jika kondisi tersebut terpenuhi maka akan muncul notifikasi bahwa “Password angka”.

```
if (!preg_match("/[0-9]/", $password)) {  
  
    echo "<script>alert('Password angka');</script>";  
  
    $x = true;  
  
}
```

Handling 6

Pada handling ini, kondisi yang harus dipenuhi ialah panjang string atau inputan dari password kurang dari 10, Jika kondisi tersebut terpenuhi maka akan muncul notifikasi bahwa ”Password kurang dari 10”.

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

Handling 7

Jika inputan username dan password tidak memenuhi semua handling di atas maka dapat dinyatakan bahwa inputan tersebut berhasil melakukan login dengan ditandai notifikasi “Berhasil”.

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

Form Action

Action harus merujuk ke halaman di server web yang akan memproses form input dan mengembalikan HTML baru kepada user. Form yang menjadi rujukan action bisa berada pada lokasi yang sama dengan skrip PHP yang tengah dijalankan, atau digunakan untuk membuat file PHP terpisah untuk handling action.

```
<form action="" method="post" name="Login_Form">
```

Tabel

Tabel dibuat sebagai field untuk memasukkan username dan password yang nantinya akan diinput oleh user. Input username bertipe text sehingga ketika diinputkan maka inputan tersebut akan terlihat sedangkan password bertipe password agar inputan tidak terlihat dan hanya menampilkan simbol, inputan password bisa dilihat jika user menekan tombol view disamping.

Untuk button login bertipe submit yang berfungsi untuk mengirim hasil inputan untuk mentrigger handling dan mengecek apakah inputan sudah benar atau belum.

```

<table border="0" align="center">

    <tr>

        <td colspan="2" align="center"></td>

    </tr>

    <tr>

        <td colspan="2" align="center"><h3>LOGIN</h3></td>

    </tr>

    <tr>

        <td align="right">Username : </td>

        <td><input name="username" type="text" placeholder="Username"></td>

    </tr>

    <tr>

        <td align="right">Password : </td>

        <td><input name="password" type="password" placeholder="Password"
></td>

    </tr>

    <tr>

        <td> </td>

        <td><input name="Submit" type="submit" value="Login"></td>

    </tr>

```

```
</table>
```

Styling CSS

Styling css disini berfungsi untuk mengatur background dari tampilan web, background body dimasukkan pada bagian dalam tag style sehingga tidak diperlukan untuk membuat file css baru yang terpisah dari file utama.

```
<style>

    body {

        background: -webkit-linear-gradient(bottom, #2dbd6e, #a6f77b);

        background-repeat: no-repeat;

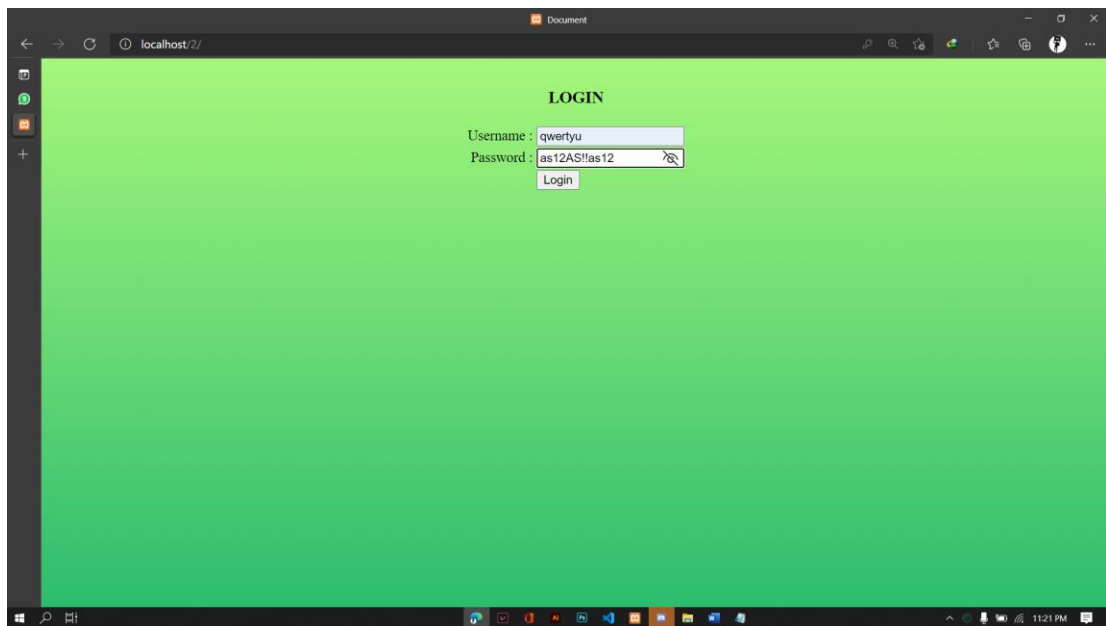
        height : 600px;

    }

</style>
```

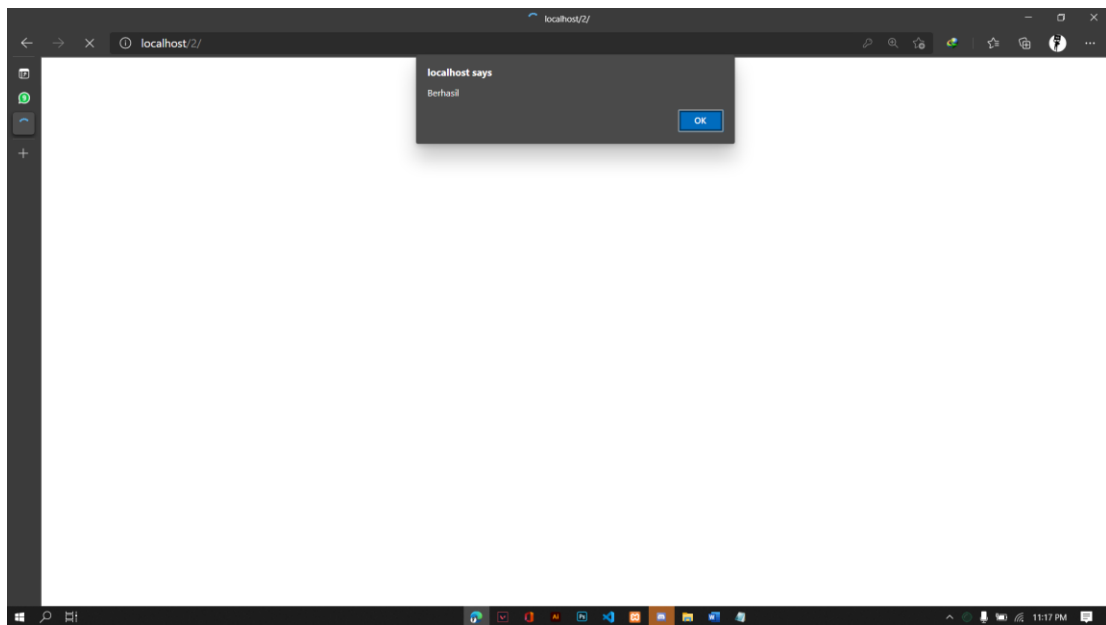
Berikut adalah hasil inputan username dan password dalam percobaan terkait beberapa handling sebelumnya.

Input Username : qwertyu | Password : as12AS!!as12



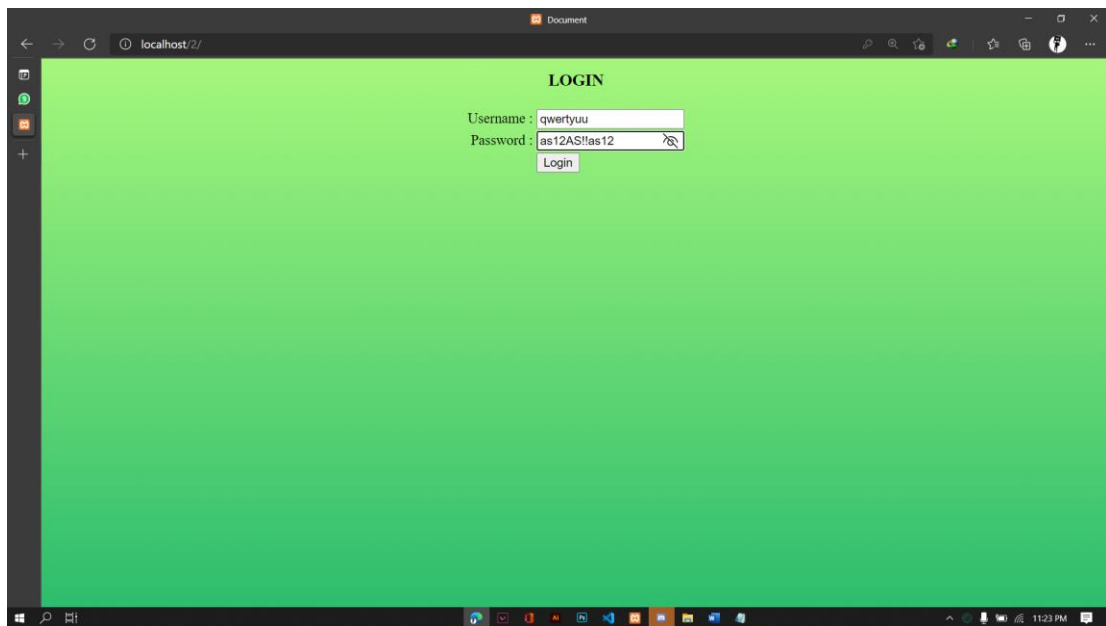
Gambar 2.1 Input pertama.

Status : Berhasil.



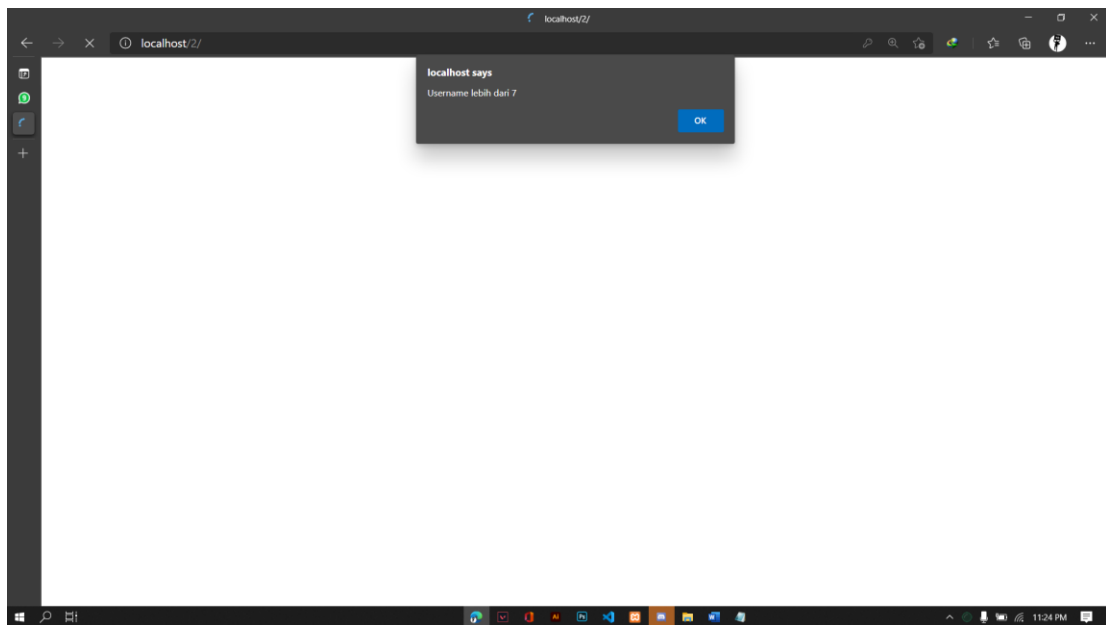
Gambar 2.2 Berhasil.

Input Username : qwertyuu | Password : as12AS!!as12



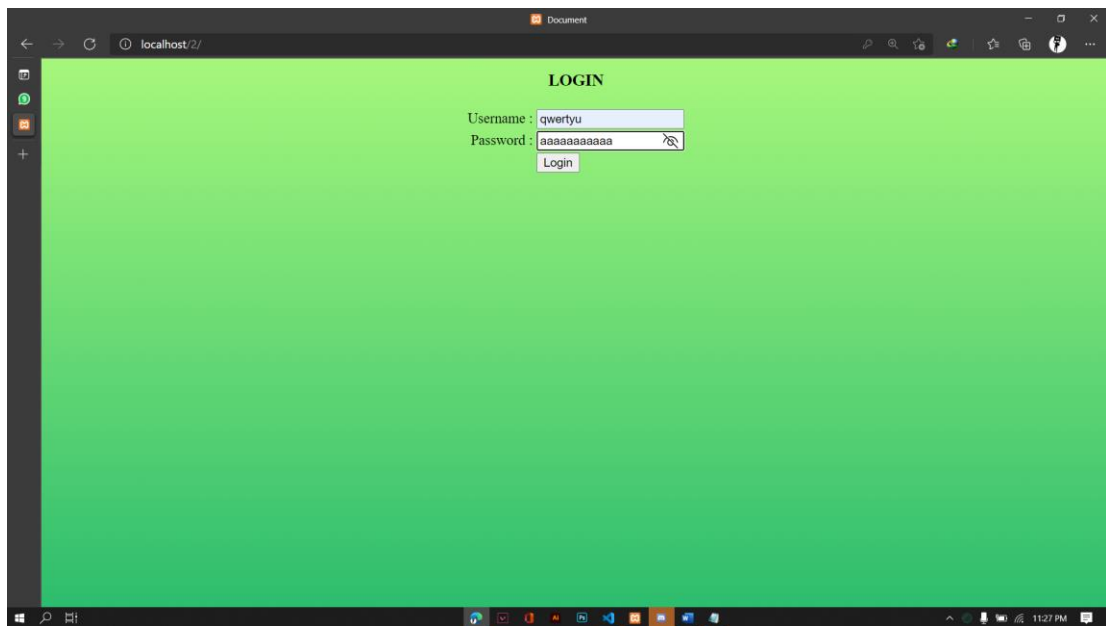
Gambar 2.3 Input Kedua.

Status : Username lebih dari 7.



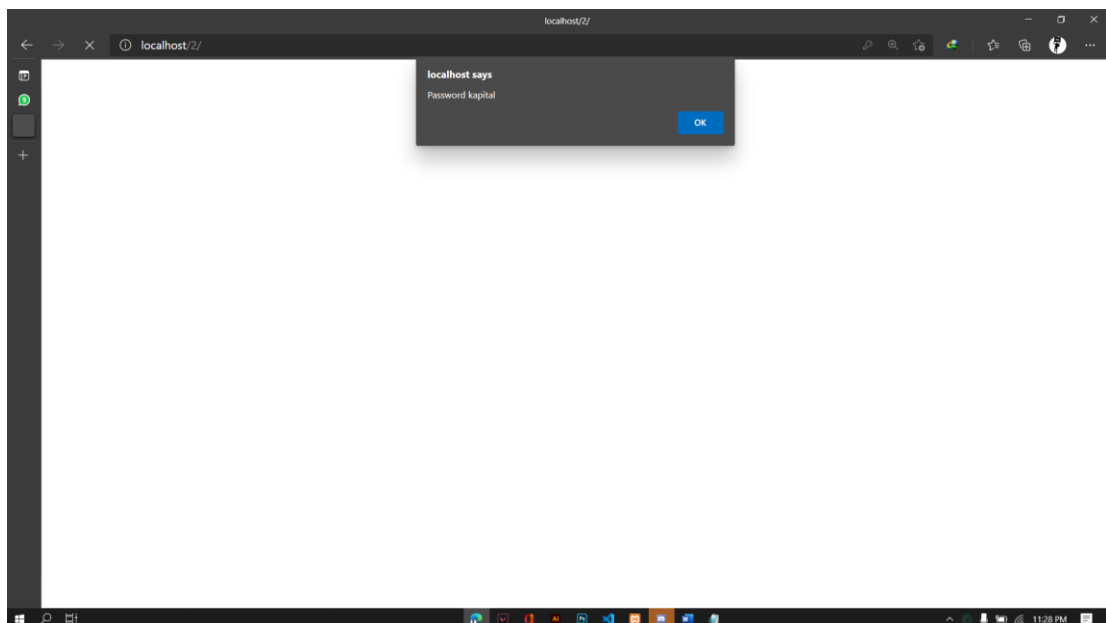
Gambar 2.4 Username lebih dari 7.

Input Username : qwertyu | Password : aaaaaaaaaa

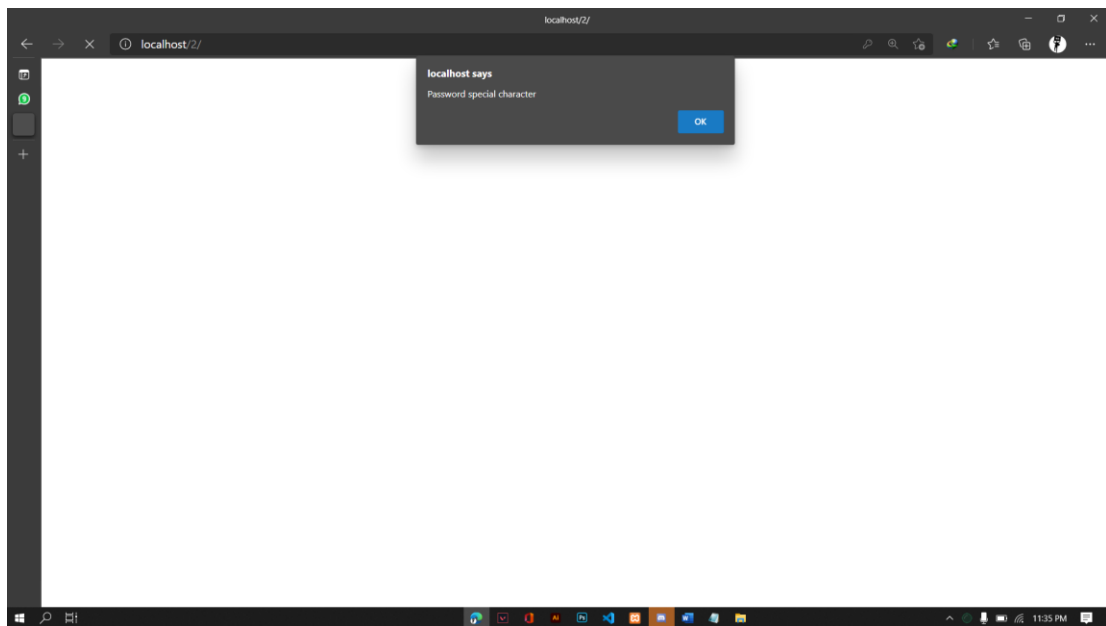


Gambar 2.5 Input ketiga.

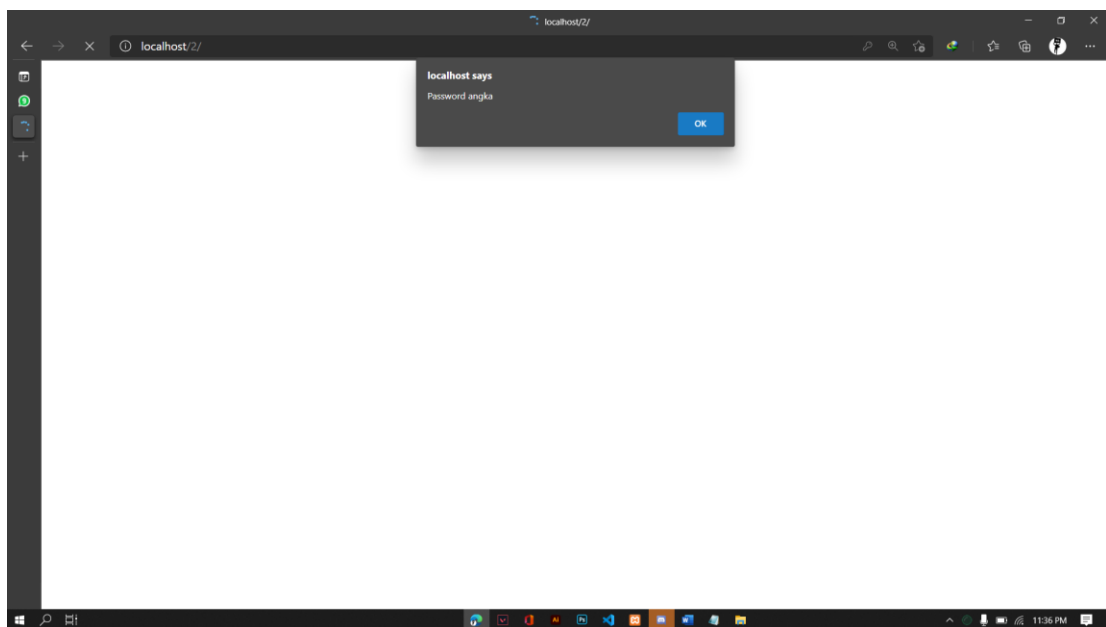
Status : Input password huruf kecil, maka akan muncul notifikasi untuk memasukkan jenis inputan yang kurang seperti huruf besar, karakter spesial, dan angka.



Gambar 2.6 Notifikasi untuk input huruf kapital.



Gambar 2.7 Notifikasi untuk input karakter spesial.



Gambar 2.8 Notifikasi untuk input angka.

BAB III

KESIMPULAN

Handling digunakan untuk menangani berbagai inputan. Input akan dicek dan disesuaikan berdasarkan semua kondisi yang telah disiapkan sebelumnya, jika kondisi pertama tidak terpenuhi maka pengecakan berlanjut di kondisi kedua dan seterusnya hingga kondisi dari inputan tercapai.

Adanya method GET dan POST yang digunakan untuk mengambil dan atau mengirim nilai variabel ke halaman lain yang dikombinasikan dengan form sehingga user akan diarahkan ke halaman selanjutnya.

DAFTAR PUSTAKA

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