

LAPORAN PRATIKUM
PEMROGRAMAN WEB DAN MOBILE I



NAMA : Rio Geradi Muke
NIM : 193030503051
MODUL : II
KELAS : A

JURUSAN/PROGRAM STUDI 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 ton@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/scr ipt %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):\\V|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.

Pembahasan:

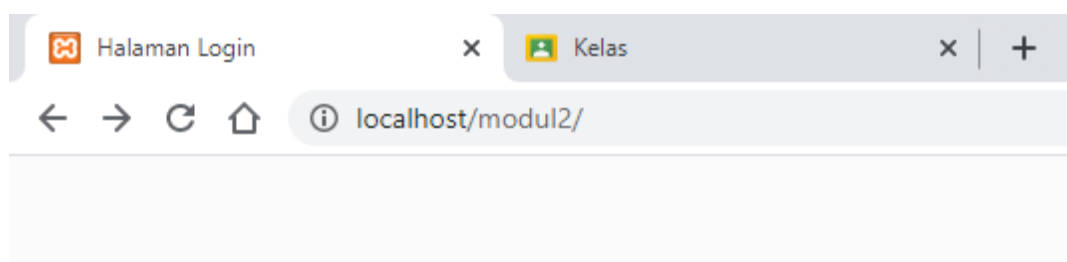


LOGIN

Username :

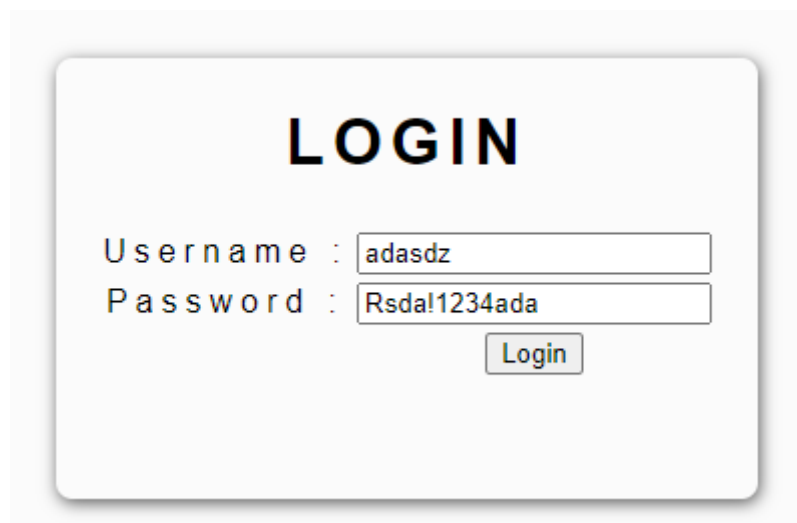
Password :

Gambar 2.1 Tampilan Login Dashboard



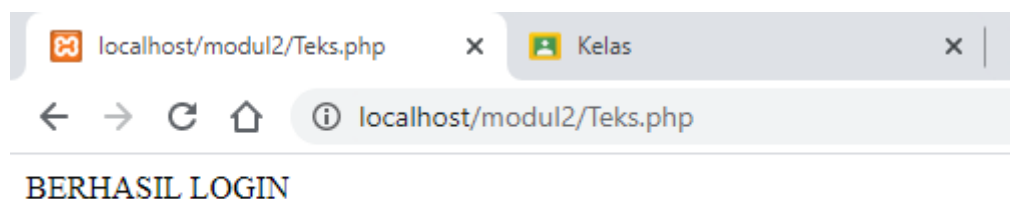
Gambar 2.2 Tampilan Halaman Login website

Tampilan Website dari LOGIN ini saya menggunakan Bahasa pemrograman dari HTML, PHP dan CSS. Karena saya membuat Fitur border kotak tersebut dari CSS jadi saya menambahkan Bahasa pemrograman CSS yang saya pelajari dari Website atau situ belajar Bahasa pemrograman setelah itu apa bila kita memasukan username dan password yang sesuai dengan soal tersebut maka akan bertuliskan Login Berhasil. Sesuai yang di soal tersebut harus memasukan username kurang dari 7 dan untuk password harus melebihi 10 karakter dan berisi sebuah huruf kapital, spesial karakter, dan huruf kecil. Setelah itu apa bila sudah memasukan username dan password sama maka akan Login Berhasil.

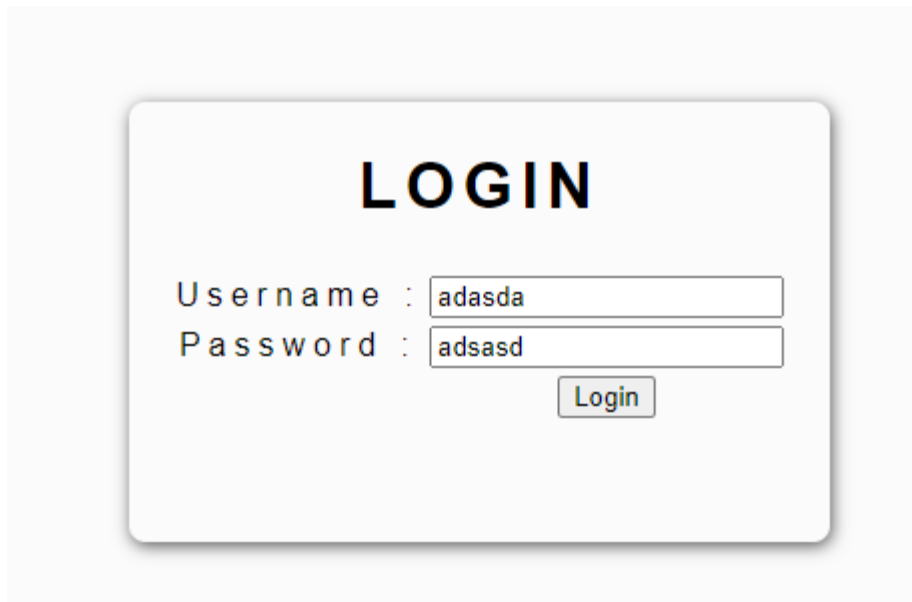


The image shows a login form with a light gray background and a white rounded rectangle in the center. At the top of the rectangle, the word "LOGIN" is written in large, bold, black capital letters. Below it, there are two input fields. The first is labeled "Username :" and contains the text "adasdz". The second is labeled "Password :" and contains the text "Rsdal1234ada". Below the password field is a button labeled "Login".

Gambar 2.3 Memasukan Username dan Password



Gambar 2.4 Login Berhasil

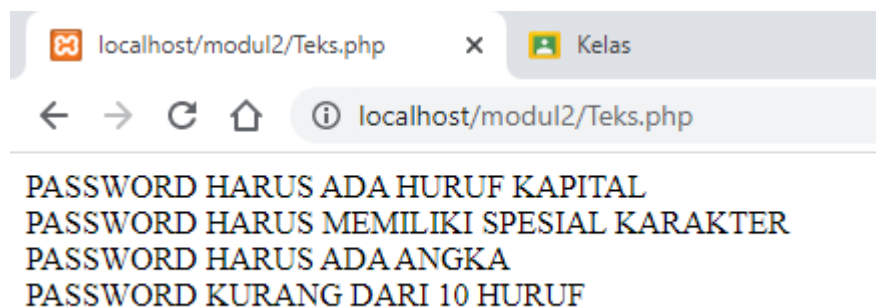


LOGIN

Username :

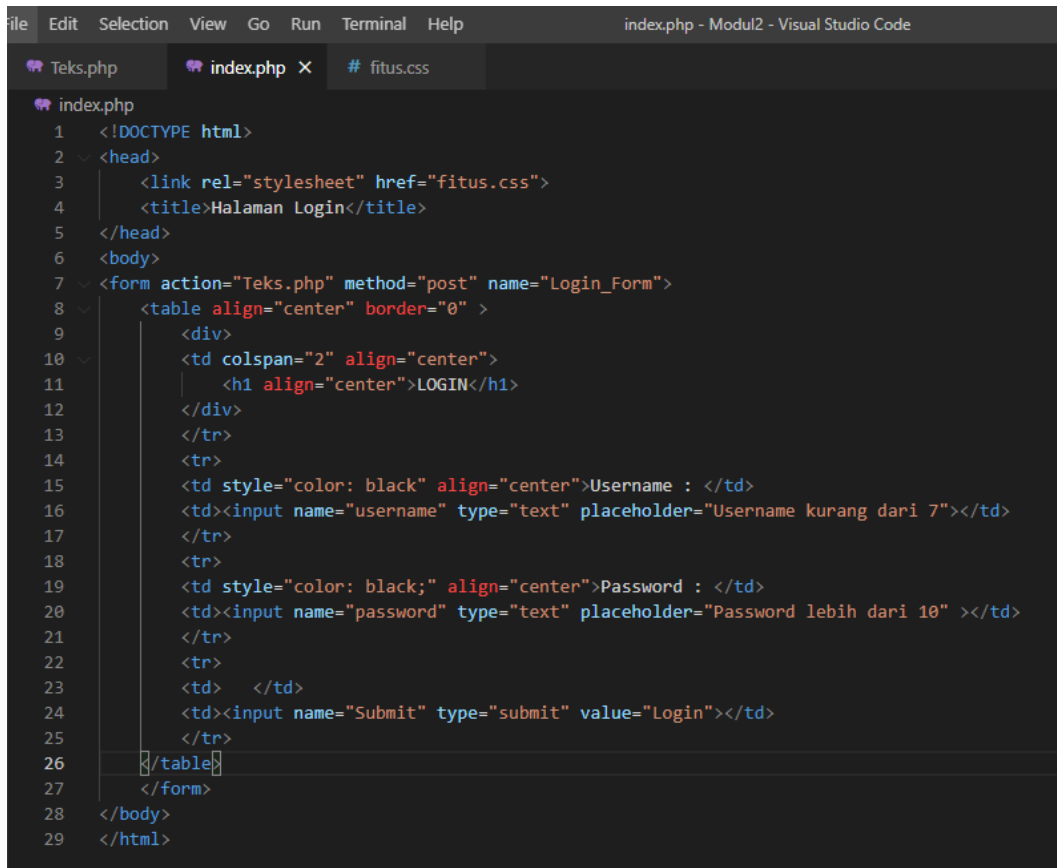
Password :

Gambar 2.5 Login username benar dan password salah



Gambar 2.6 Tampilan username benar dan password salah.

Untuk selanjut apa bila tidak menyesuaikan kondisi username dan password dengan benar maka hasilnya salah /atau akan bertuliskan seperti gambar di atas setelah itu harus menyesuaikan kondisi seperti soal di modul 2 tersebut. Agar hasilnya benar dan login berhasil.



```
1 <!DOCTYPE html>
2 <head>
3   <link rel="stylesheet" href="fitus.css">
4   <title>Halaman Login</title>
5 </head>
6 <body>
7 <form action="Teks.php" method="post" name="Login_Form">
8   <table align="center" border="0" >
9     <div>
10      <td colspan="2" align="center">
11        <h1 align="center">LOGIN</h1>
12      </div>
13    </tr>
14    <tr>
15      <td style="color: black" align="center">Username : </td>
16      <td><input name="username" type="text" placeholder="Username kurang dari 7"></td>
17    </tr>
18    <tr>
19      <td style="color: black;" align="center">Password : </td>
20      <td><input name="password" type="text" placeholder="Password lebih dari 10" ></td>
21    </tr>
22    <tr>
23      <td> </td>
24      <td><input name="Submit" type="submit" value="Login"></td>
25    </tr>
26  </table>
27 </form>
28 </body>
29 </html>
```

Gambar 2.7 Program Index.php berisi HTML

Pertama saya akan menjelaskan cara membuat file index.php yaitu terdapat aplikasi yang bernama XAMPP setelah itu membuka file manager tersebut dan di dalam file manager XAMPP tersebut terdapat file HTDOCS lalu masuk kedepan file tersebut setelah itu membuka VISUAL STUDIO CODE lalu membuat new file lalu save setelah itu masuk kedalam file HTDOCS save di tempat tersebut dan filenya bernama sesuai Bahasa programnya contohnya index.php dan index.html dan fitur.css. Setelah itu saya di gambar 2.7 menggunakan Bahasa PHP dan HTML lalu kegunaanya <link rel="stylesheet" href="fitus.css"> adalah untuk memanggil atau mengabungkan file CSS ke dalam website. Bisa di bilang CSS di program saya adalah fitur untuk border box dari Web Login tersebut. Setelah itu <form action="Teks.php" method="post" name="Login_Form"> kegunaan dari codingan ini adalah apa bila kita memasukan username dan password benar atau salah maka akan di tampilkan kondisi di dalam sebuah teks.php tersebut. Karena di dalam teks.php tersebut berisi sebuah program untuk menjalankan username dan password apa bila ada salah dan beanr.


```
<td style="color: black" align="center">Username : </td>
      <td><input name="username" type="text" placeholder="Username
kurang dari 7"></td>
```

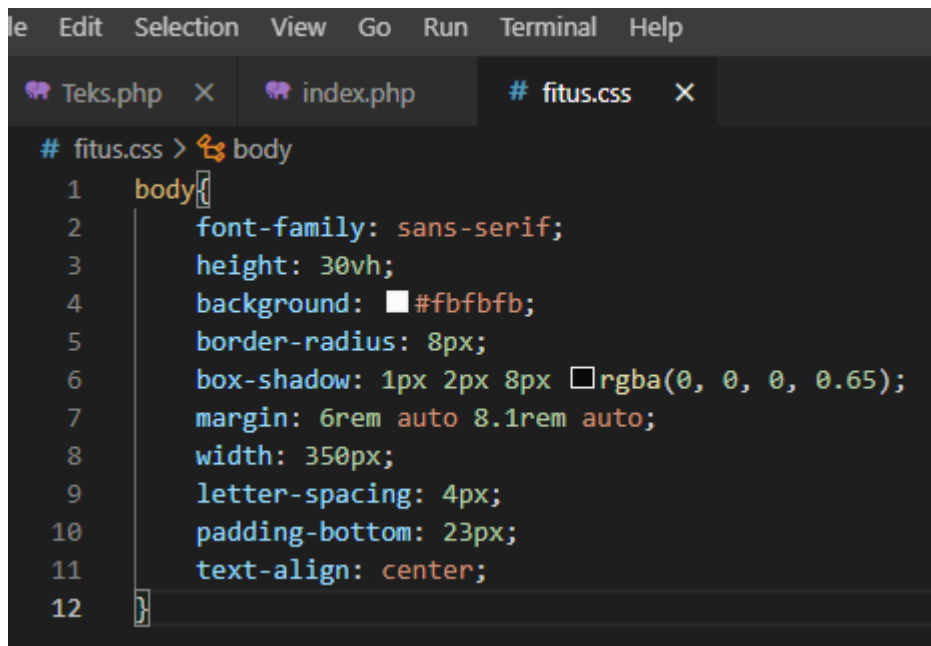
Untuk Maksud Coding tersebut untuk memasukan username yang dan berisi username kurang dari 7 .

```
<td style="color: black;" align="center">Password : </td>
      <td><input name="password" type="text" placeholder="Password
lebih dari 10" ></td>
```

Untuk maksud dari coding tersebut juga sama akan memasukan password yang berisi password lebih dari 10 dan untuk maksud placeholder nama dari inputan.

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

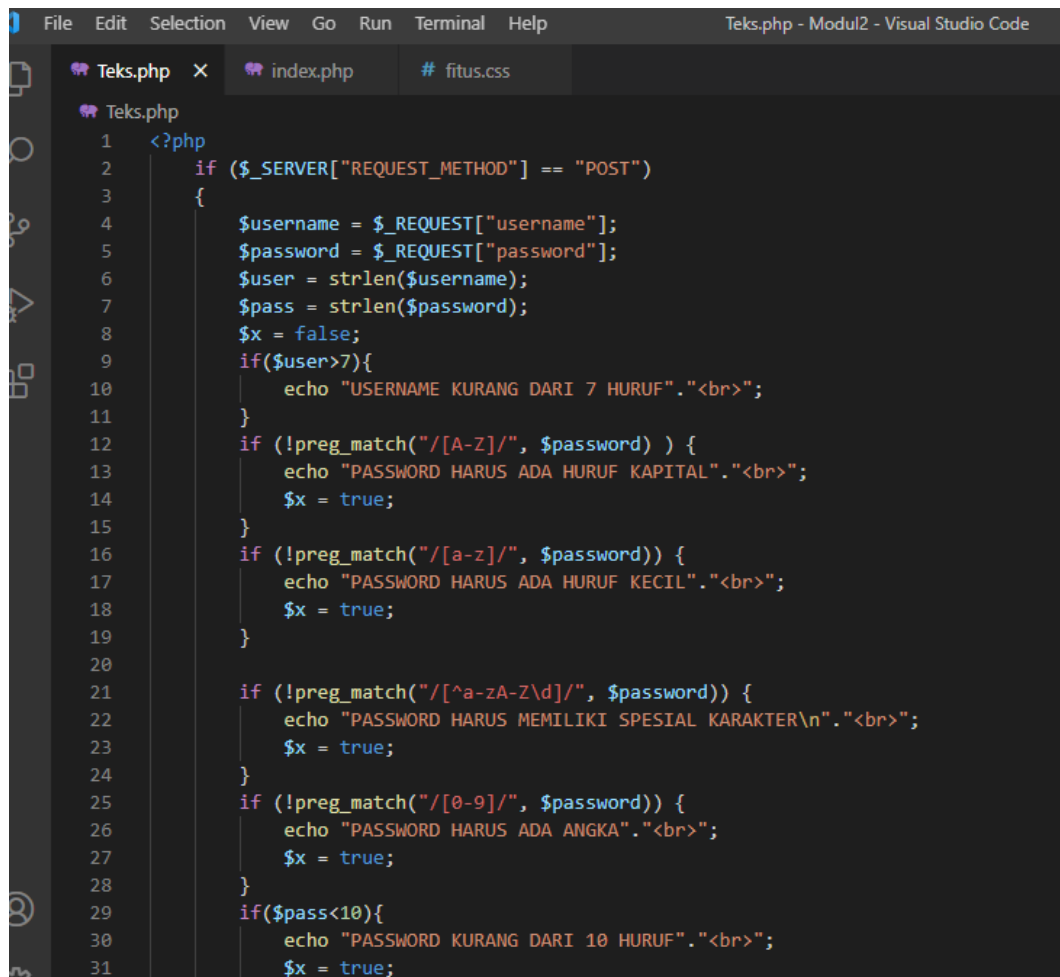
Lalu untuk sumbit ini adalah untuk menyerahkan data tersebut dan bertulisan login maka akan masuk ke perintah masuk ke website.



Gambar 2.8 Program CSS

Pertama harus memasukan kedalam body untuk memasukan codingnya artu body tubuh dan di dalam tubuh ada organ begitu juga Bahasa pemrograman CSS ini. Kegunaan dari font-family adalah jenis font dari teks website login saya setelah itu height : 30vh; kegunanya adalah untuk mengatur border kotak lalu saya menggunakan background warna hitam dan box-shadow maksudnya adalah sisi border hitam tersebut menggunakan box-shadow. Dan juga kegunaanya yang lebih

penting yaitu margin untuk ukuran website tersebut agar rata dengan desktop web dan text juga harus ke center atau rata tengah.



```
1 <?php
2 if ($_SERVER["REQUEST_METHOD"] == "POST")
3 {
4     $username = $_REQUEST["username"];
5     $password = $_REQUEST["password"];
6     $user = strlen($username);
7     $pass = strlen($password);
8     $x = false;
9     if($user>7){
10         echo "USERNAME KURANG DARI 7 HURUF."<br>";
11     }
12     if (!preg_match("/[A-Z]/", $password) ) {
13         echo "PASSWORD HARUS ADA HURUF KAPITAL."<br>";
14         $x = true;
15     }
16     if (!preg_match("/[a-z]/", $password)) {
17         echo "PASSWORD HARUS ADA HURUF KECIL."<br>";
18         $x = true;
19     }
20
21     if (!preg_match("/^[a-zA-Z\d]/", $password)) {
22         echo "PASSWORD HARUS MEMILIKI SPESIAL KARAKTER\<br>";
23         $x = true;
24     }
25     if (!preg_match("/[0-9]/", $password)) {
26         echo "PASSWORD HARUS ADA ANGKA."<br>";
27         $x = true;
28     }
29     if($pass<10){
30         echo "PASSWORD KURANG DARI 10 HURUF."<br>";
31         $x = true;
```

Gambar 2.9 Program Text.php



```
32     }
33     if( $x == false ) {
34         echo "BERHASIL LOGIN."<br>";
35     }
36
37 ?>
38
```

Gambar 2.10 Program Text.php

Pertama saya membuat file php lalu menuliskan <?php \$SERVER adalah untuk menampilkan data di dalam website. if (\$_SERVER["REQUEST_METHOD"] == "POST") Sintaks ini merupakan sintaks untuk memeriksa request method yang telah digunakan atau pernah diterapkan pada web tersebut. if (\$_SERVER["REQUEST_METHOD"] == "POST") saya menggunakan

\$REQUEST variabel superglobal \$_REQUEST salah satu yang tepat. Fungsi dari STRLEN adalah untuk memperpanjang String dari sebuah variabel username dan password. Setelah itu ada yang namanya variabel X di yaitu nilai x=false. Dan untuk `if($user>7){ echo "USERNAME KURANG DARI 7 HURUF". "
";` adalah untuk jika username di bawah atau kurang dari 7 maka akan berhasil jadi menggunakan user >7 apa bila melebihi 7 karakter maka akan tidak berhasil login. Jadi `x = true.` `if (!preg_match("/[A-Z]/", $password)) { echo "PASSWORD HARUS ADA HURUF KAPITAL". "
";` . `preg match` adalah kegunaannya mencari akan tipe string. Dan Password tersebut harus ada huruf Kapital jadi maka di tuliskan `[A-Z]` . dan `x=true` dan sampai akhirnya seterusnya . `if($x == false) { echo "BERHASIL LOGIN". "
";` lalu untuk kegunaan dari codingan ini adalah fungsi ketika nilai variabel x tetap false karena diawal diinisiasikan false. Jika salah input maka x akan menjadi true lalu muncul jika tetap, False maka login akan berhasil seperti tampilan website berhasil LOGIN.

BAB III

KESIMPULAN

Kesimpulan dari pratikum modul 2 yang saya dapatkan ini adalah kegunaan dari GET dan 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.

DAFTAR PUSTAKA

- Muhardin, A. (2018). *Belajar HTML #03: Membuat Paragraf pada HTML*. HTML. <https://www.petanikode.com/html-paragraf/>
- Praktikum, K. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.
- Yukawa, K. (2019). *Tutorial Mendesain Form Login Menggunakan HTML&CSS _ by Kimiyuki Yukawa _ Medium*. HTML dan CSS.

LAMPIRAN

```
<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

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

Gambar 1.2 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 1.3 HTML dan PHP

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

Gambar 1.6 HTML dan PHP

PHP Form Validation Example

* required field.

Name: *

E-mail: *

Website:

Comment:

Gender: ☐ Female ☐ Male *

Gambar 1.7 VALIDASI

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

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

Gender:
`<input type="radio" name="gender" value="female">Female`
`<input type="radio" name="gender" value="male">Male`

Gambar 1.9 Text Radio Button

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

Gambar 1.10 From Element

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

Gambar 1.11 PHP

```
<form method="post" action="test_form.php/"><script>alert('hacked')</script>
```

Gambar 1.12 PHP

```
<form method="post"  
action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

Gambar 1.13 Script

```
<?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


```

<?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

```

<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

```

$name = test_input($_POST["name"]);
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
    $nameErr = "Only letters and white space allowed";
}

```

Gambar 1.19 Validasi Nama

```

$email = test_input($_POST["email"]);
if (!filter_var($email, FILTER_VALIDATE_EMAIL))
    { $emailErr = "Invalid email format";
    }

```

Gambar 1.20 Validasi Email

```

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

```

Gambar 1.21 Validasi URL

`<textarea>` dan tag `</textarea>`.

Gambar 1.22 Validasi URL

```
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



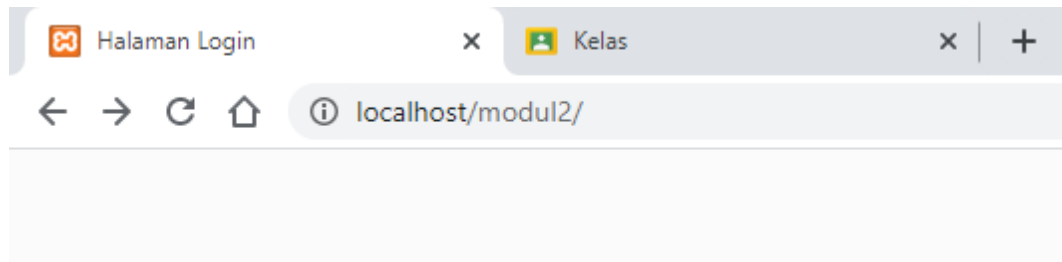
LOGIN

Username : Username kurang dari 7

Password : Password lebih dari 10

Login

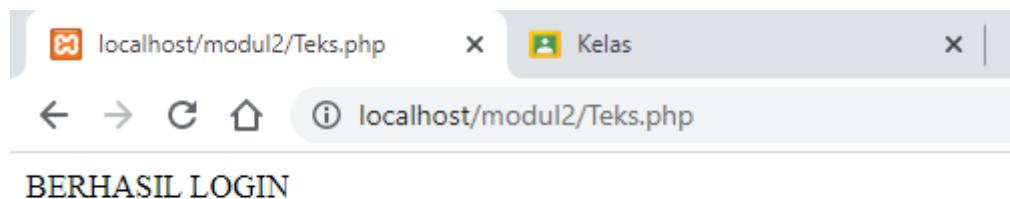
Gambar 2.1 Tampilan Login Dashboard



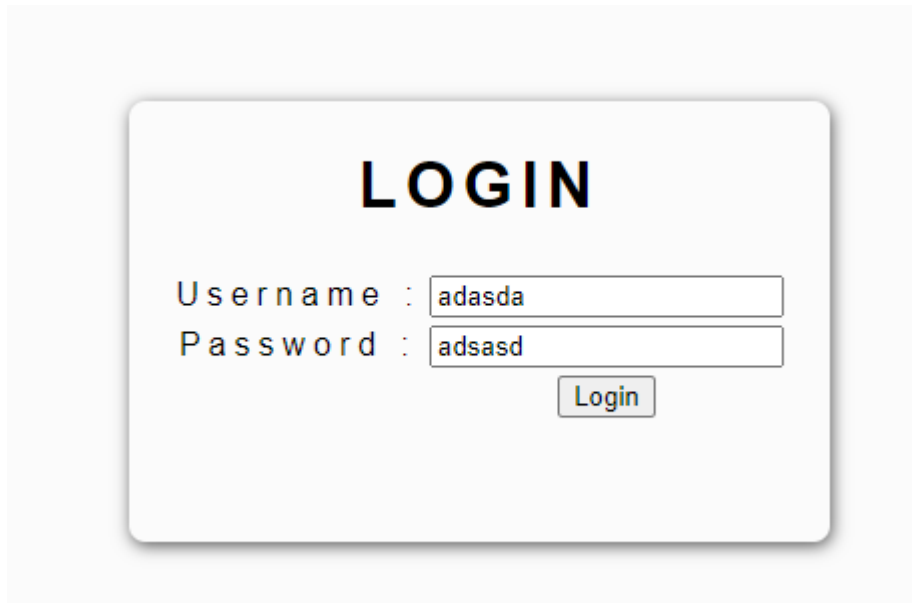
Gambar 2.2 Tampilan Halaman Login website

A login form titled 'LOGIN' in large, bold, black letters. Below the title, there are two input fields. The first is labeled 'Username :' and contains the text 'adasdz'. The second is labeled 'Password :' and contains the text 'Rsdal1234ada'. Below the password field is a button labeled 'Login'.

Gambar 2.3 Memasukan Username dan Password



Gambar 2.4 Login Berhasil

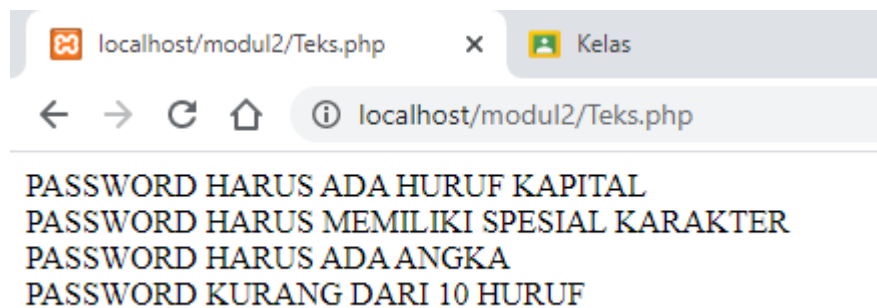


LOGIN

Username :

Password :

Gambar 2.5 Login username benar dan password salah



localhost/modul2/Teks.php x Kelas

← → ↻ 🏠 ⓘ localhost/modul2/Teks.php

PASSWORD HARUS ADA HURUF KAPITAL
PASSWORD HARUS MEMILIKI SPESIAL KARAKTER
PASSWORD HARUS ADA ANGKA
PASSWORD KURANG DARI 10 HURUF

Gambar 2.6 Tampilan username benar dan password salah.

```
File Edit Selection View Go Run Terminal Help index.php - Modul2 - Visual Studio Code

Teks.php index.php X # fitus.css

index.php
1 <!DOCTYPE html>
2 <head>
3   <link rel="stylesheet" href="fitus.css">
4   <title>Halaman Login</title>
5 </head>
6 <body>
7   <form action="Teks.php" method="post" name="Login_Form">
8     <table align="center" border="0">
9       <div>
10        <td colspan="2" align="center">
11          <h1 align="center">LOGIN</h1>
12        </td>
13      </div>
14      <tr>
15        <td style="color: black" align="center">Username : </td>
16        <td><input name="username" type="text" placeholder="Username kurang dari 7"></td>
17      </tr>
18      <tr>
19        <td style="color: black;" align="center">Password : </td>
20        <td><input name="password" type="text" placeholder="Password lebih dari 10" ></td>
21      </tr>
22      <tr>
23        <td> </td>
24        <td><input name="Submit" type="submit" value="Login"></td>
25      </tr>
26    </table>
27  </form>
28 </body>
29 </html>
```

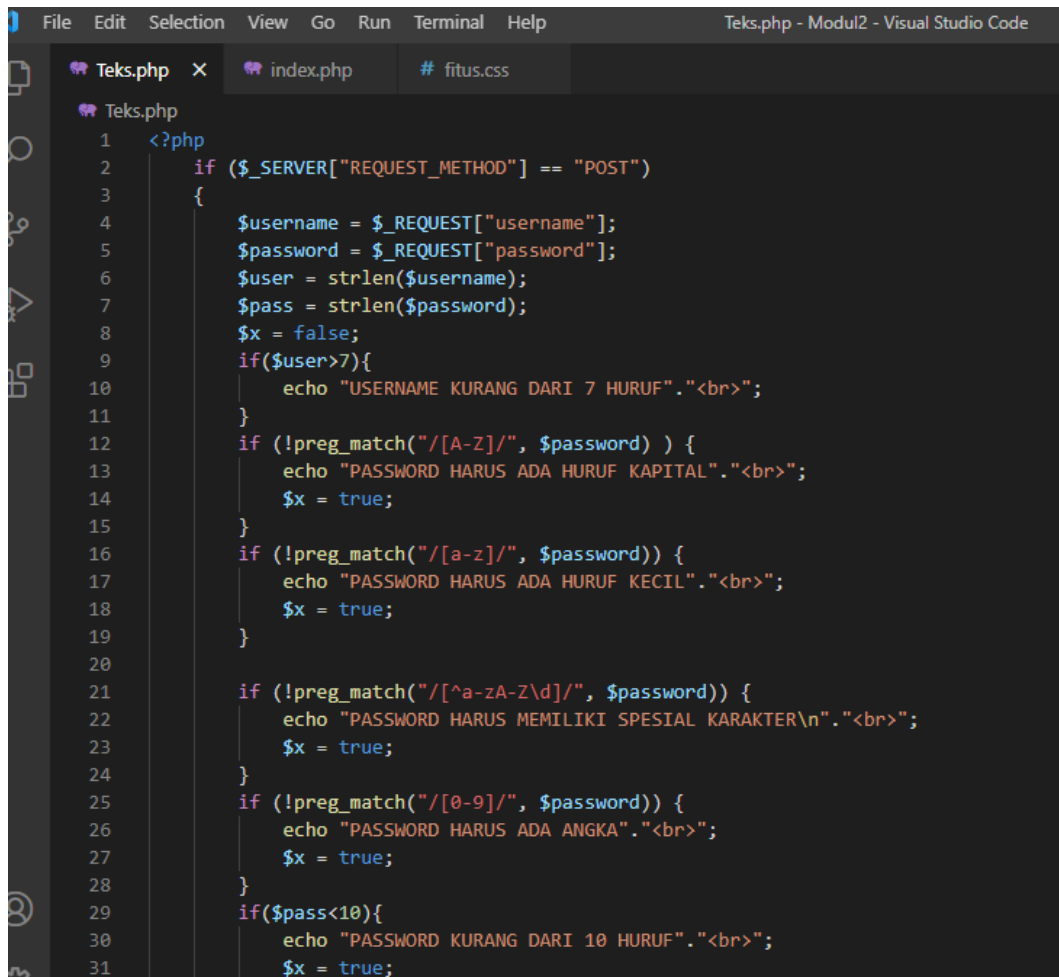
Gambar 2.7 Program Index.php berisi HTML

```
File Edit Selection View Go Run Terminal Help

Teks.php X index.php # fitus.css X

# fitus.css > body
1 body
2   font-family: sans-serif;
3   height: 30vh;
4   background: #fbfbfb;
5   border-radius: 8px;
6   box-shadow: 1px 2px 8px rgba(0, 0, 0, 0.65);
7   margin: 6rem auto 8.1rem auto;
8   width: 350px;
9   letter-spacing: 4px;
10  padding-bottom: 23px;
11  text-align: center;
12 }
```

Gambar 2.8 Program CSS



```
1 <?php
2 if ($_SERVER["REQUEST_METHOD"] == "POST")
3 {
4     $username = $_REQUEST["username"];
5     $password = $_REQUEST["password"];
6     $user = strlen($username);
7     $pass = strlen($password);
8     $x = false;
9     if($user>7){
10         echo "USERNAME KURANG DARI 7 HURUF."<br>";
11     }
12     if (!preg_match("/[A-Z]/", $password) ) {
13         echo "PASSWORD HARUS ADA HURUF KAPITAL."<br>";
14         $x = true;
15     }
16     if (!preg_match("/[a-z]/", $password)) {
17         echo "PASSWORD HARUS ADA HURUF KECIL."<br>";
18         $x = true;
19     }
20
21     if (!preg_match("/[^\a-zA-Z\d]/", $password)) {
22         echo "PASSWORD HARUS MEMILIKI SPESIAL KARAKTER\n."<br>";
23         $x = true;
24     }
25     if (!preg_match("/[0-9]/", $password)) {
26         echo "PASSWORD HARUS ADA ANGKA."<br>";
27         $x = true;
28     }
29     if($pass<10){
30         echo "PASSWORD KURANG DARI 10 HURUF."<br>";
31         $x = true;
```

Gambar 2.9 Program Text.php



```
32     }
33     if( $x == false ) {
34         echo "BERHASIL LOGIN."<br>";
35     }
36 }
37 ?>
38
```

Gambar 2.10 Program Text.php