

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



**Nama : Mitchel Christian Timothyus Singaraca**

**Nim : 193020503041**

**Kelas : A**

**Modul : II (Form Handling)**

**JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKA RAYA**

**2021**

# **BAB I**

## **TUJUAN DAN LANDASAN TEORI**

### **1.1. Tujuan**

**1.1.1.** Mahasiswa mampu membuat handling yang mampu mengolah data dari form HTML.

**1.1.2.** Mahasiswa mampu membuat batasan-batasan untuk menangani inputan dari form HTML.

### **1.2. Landasan Teori**

Tag-tag dalam membuat form sebagai berikut:

- `<form>` Tag untuk membuat sebuah form HTML untuk input pengguna
- `<input>` Tag untuk membuat sebuah kontrol input
- `<textarea>` Tag untuk membuat sebuah kontrol input multibaris (text area)
- `<button>` Tag untuk membuat sebuah tombol yang dapat diklik
- `<select>` Tag untuk membuat sebuah daftar drop-down
- `<optgroup>` Tag untuk membuat sebuah kelompok pilihan yang terkait dalam daftar drop-down
- `<option>` Tag untuk membuat pilihan dalam daftar drop-down
- `<label>` Tag untuk membuat sebuah label untuk sebuah elemen `<input>`
- `<fieldset>` Grup unsur terkait dalam bentuk
- `<legend>` Tag untuk membuat sebuah caption untuk sebuah elemen `<fieldset>`, `<figure>`, atau `<details>`
- `<datalist>` Menentukan daftar pilihan yang telah ditetapkan untuk kontrol input (tag baru HTML5)
- `<keygen>` Tag untuk membuat key-pair generator kolom input (tag baru HTML5)
- `<output>` Tag untuk membuat hasil penghitungan (tag baru HTML5)

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>
```

Perhatikan bahwa ada dua atribut di dalam tag `<form>` pembuka:

Atribut `action` merujuk kepada file PHP “process-form.php” yang menerima data yang dimasukkan ke dalam formulir ketika pengguna mengirimkannya dengan menekan tombol kirim.

Atribut `method` memberi tahu browser untuk mengirim data formulir melalui metode POST.

### **Mengambil Data Formulir dengan PHP**

Untuk mengakses nilai bidang formulir tertentu, kalian bisa menggunakan variabel superglobal berikut. Variabel-variabel ini tersedia di semua cakupan pada seluruh skrip.

Superglobal	Description
<code>\$_GET</code>	Berisi daftar semua nama bidang dan nilai yang dikirim oleh formulir menggunakan metode get (mis. Melalui parameter URL).
<code>\$_POST</code>	Berisi daftar semua nama bidang dan nilai yang dikirim oleh formulir menggunakan metode posting (data tidak akan terlihat di URL).
<code>\$_REQUEST</code>	Berisi nilai-nilai dari variabel <b><code>\$_GET</code></b> dan <b><code>\$_POST</code></b> serta nilai dari variabel superglobal <b><code>\$_COOKIE</code></b> .

**Gambar 1.1** Tabel data form

## 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 sensitive. Jangan pernah gunakan GET 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 1.2** Manipulasi String

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

**Gambar 1.3** Tabel validasi

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.

## BAB II

### PEMBAHASAN

**2.1. Buatlah program web untuk menginputkan username dan password menggunakan form dan penanganan input data dengan kriteria sebagai berikut:**

- a. username yang diinputkan tidak boleh lebih dari tujuh karakter.**
- b. password yang diinputkan harus terdiri dari huruf kapital, huruf kecil, angka dan karakter khusus.**
- c. Jumlah karakter password tidak boleh kurang dari sepuluh karakter.**

Pada praktikum ini diminta untuk membuat sebuah web yang dapat digunakan untuk menginputkan username dan password dengan beberapa kondisi yang telah ditentukan.

Program ini terdiri dari 2 bagian yaitu bagian utama dan bagian kondisi. Gambar dibawah merupakan bagian sintaks utama yang dimana merupakan bagian head dan body.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <form action="" method="post" name="Login_Form">
    <table>
      <tr>
        <td><h3>Login</h3></td>
      </tr>
      <tr>
        <td>Username : </td>
        <td><input name="username" type="text" placeholder="Username"></td>
      </tr>
      <tr>
        <td>Password : </td>
        <td><input name="password" type="text" placeholder="Password" ></td>
      </tr>
      <tr>
        <td></td>
        <td><input name="Submit" type="submit" value="Login"></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

**Gambar 2.1** Gambar sintaks body

Pada bagian head terdapat sebuah judul yaitu Dokumen. Pada bagian body terdapat form, dimana form berisi tabel, tabel tersebut digunakan untuk tempat inputan password, username serta tombol untuk menyerahkan inputan password dan username tersebut. Pada baris <input username...> merupakan sintaks yang membuat sebuah tempat untuk melakukan input username sehingga nantinya username tersebut dapat dipanggil kedalam pembukaan php. <input password...> merupakan sintaks yang membuat sebuah tempat untuk melakukan input password sehingga nantinya password tersebut dapat dipanggil kedalam pembukaan php. <input submit...> merupakan sintaks untuk membuat tombol submit yang mana tombol ini digunakan untuk menjalankan program.

## Login



Username :

Password :

**Gambar 2.2** Tampilan didalam web

Pada pembukaan php atau kondisi ini terdapat sebuah konten untuk melakukan penanganan terhadap beberapa kondisi yang diminta pada soal. Berikut ini gambar isi dari pembukaan php.



```

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

    if($user>7){
        echo ("Error: Username harus lebih dari 7 huruf<br>");
        $x = true;
    }
    if (!preg_match("/[A-Z]/", $password) ) {
        echo "Error: Password harus mengandung 1 huruf kapital<br>";
        $x = true;
    }
    if (!preg_match("/[a-z]/", $password)) {
        echo "Error: Password harus mengandung 1 huruf kecil<br>";
        $x = true;
    }
    if (!preg_match("/^[^a-zA-Z\d]/", $password)) {
        echo "Error: Password harus mengandung 1 special character<br>";
        $x = true;
    }
    if (!preg_match("/[0-9]/", $password)) {
        echo "Error: Password harus mengandung 1 digit<br>";
        $x = true;
    }
    if($pass>10){
        echo "Error: Password harus lebih dari 10<br>";
        $x = true;
    }
    if( $x == false ){
        echo "Login berhasil";
    }
}
?>

```

**Gambar 2.3** Isi pembukaan php atau kondisi

Pada bagian awal pembukaan php berisi fungsi pemilihan if. Pada fungsi ini terdapat variabel SERVER yang memiliki REQUEST\_METHOD yang mana method yang digunakan adalah POST. Sintaks ini digunakan untuk memvalidasi form yang nantinya jika sudah disubmit akan tervalidasi. Variabel username dan password ditentukan untuk menampung nilai username dan password yang akan kita inputkan nantinya. Variabel user dan pass untuk merupakan variabel yang berfungsi untuk menghitung panjang string dari username dan password. Variabel x ditentukan sebagai nilai false.

Pada fungsi pemilihan menggunakan tipe data Boolean, tipe data ini hanya bisa diisi dengan salah satu dari 2 nilai: true atau false. Dalam fungsi if ini digunakan untuk untuk memutuskan apa yang mesti dijalankan ketika sebuah kondisi terjadi. Pada bagian dalam percabangan tersebut terdapat 7 fungsi pemilihan if yang mana

fungsi tersebut menggunakan tipe data boolean, dibuat agar memenuhi kriteria yang ditentukan. Berikut penjelasan tiap fungsi.

Fungsi if pertama merupakan fungsi untuk menangani input username. Variabel yang digunakan adalah user yang berisi panjang username. Sehingga bagian ini adalah untuk menangani batas input username yang tidak boleh lebih dari 7 karakter.

Fungsi if kedua merupakan fungsi untuk menangani input password yang mana fungsi tersebut akan memeriksa data yang diinputkan apakah sudah memiliki huruf kapital sebanyak 1 atau lebih.

Fungsi if ketiga merupakan fungsi untuk menangani input password yang mana fungsi tersebut akan memeriksa data yang diinputkan apakah sudah memiliki huruf kecil sebanyak 1 atau lebih.

Fungsi if keempat merupakan fungsi untuk menangani input password yang mana fungsi tersebut akan memeriksa data yang diinputkan apakah sudah memiliki karakter spesial sebanyak 1 atau lebih.

Fungsi if kelima merupakan fungsi untuk menangani input password yang mana fungsi tersebut akan memeriksa data yang diinputkan apakah sudah memiliki angka sebanyak 1 atau lebih.

Fungsi if keenam merupakan fungsi untuk menangani input password. Variabel yang digunakan adalah pass yang berisi panjang password. Sehingga bagian ini adalah untuk menangani batas input password yang harus lebih 10 karakter.

Fungsi if ketujuh ini merupakan fungsi yang dimana apabila tidak ada satupun fungsi yang terpenuhi maka login berhasil.

#### **Login**

Username :   
Password :   
  
Error: Username harus lebih dari 7 huruf  
Error: Password harus mengandung 1 huruf kapital  
Error: Password harus mengandung 1 huruf kecil  
Error: Password harus mengandung 1 special character  
Error: Password harus mengandung 1 digit

**Gambar 2.4** Tampilan saat kondisinya tidak terpenuhi

### Login

Username :

Password :

Login berhasil

**Gambar 2.5** Tampilan saat semua kondisinya terpenuhi

## **BAB III**

### **KESIMPULAN**

Dari praktikum ini dapat disimpulkan bahwa dalam pembuatan form peran form handling sangat membantu, yang mana form handling sangat membantu dalam memastikan program dapat berjalan dengan baik serta menghindari masalah pada saat program dijalankan. Form handling biasanya menggunakan method GET dan POST. GET merupakan sebuah array dari variabel yang dikirimkan ke skrip melalui parameter URL. Sedangkan POST adalah sebuah array dari variabel yang dikirimkan ke skrip melalui method HTTP POST. Yang mana masing-masing method memiliki keunggulannya pada kondisi tertentu

## DAFTAR PUSTAKA

Dosen Teknik Informatika. Pemrograman Web dan Mobile 1. 2021.

MODUL PRAKTIKUM PEMROGRAMAN WEB I.

Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika

*Belajar PHP : Form Handling - WebHozz Code.* (n.d.). Retrieved April 5, 2021,  
from <https://www.webhozz.com/code/php-form-handling/>

*PHP: Validasi Data Form [1/3] | Jago Ngoding.* (n.d.). Retrieved April 5, 2021,  
from <https://jagongoding.com/web/php/web-dinamis/validasi-data-form-part-1/>

*Tag-tag Pada HTML beserta Fungsinya - Gilacoding.* (n.d.). Retrieved April 5,  
2021, from <https://gilacoding.com/read/tag-tag-pada-html-beserta-fungsinya>

## LAMPIRAN

Superglobal	Description
<code>\$_GET</code>	Berisi daftar semua nama bidang dan nilai yang dikirim oleh formulir menggunakan metode get (mis. Melalui parameter URL).
<code>\$_POST</code>	Berisi daftar semua nama bidang dan nilai yang dikirim oleh formulir menggunakan metode posting (data tidak akan terlihat di URL).
<code>\$_REQUEST</code>	Berisi nilai-nilai dari variabel <b><code>\$_GET</code></b> dan <b><code>\$_POST</code></b> serta nilai dari variabel superglobal <b><code>\$_COOKIE</code></b> .

**Gambar 1.1** Tabel data form

### PHP Form Validation Example

\* required field.

Name:  \*

E-mail:  \*

Website:

Comment:

Gender: ☐ Female ☐ Male \*

**Gambar 1.2** Manipulasi String

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

**Gambar 1.3** Tabel validasi

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
<form action="" method="post" name="Login_Form">
  <table>
    <tr>
      <td><h3>Login</h3></td>
    </tr>
    <tr>
      <td>Username : </td>
      <td><input name="username" type="text" placeholder="Username"></td>
    </tr>
    <tr>
      <td>Password : </td>
      <td><input name="password" type="text" placeholder="Password" ></td>
    </tr>
    <tr>
      <td></td>
      <td><input name="Submit" type="submit" value="Login"></td>
    </tr>
  </table>
</form>
</body>
</html>

```

**Gambar 2.1** Gambar sintaks body

## Login

Username :

Password :

**Gambar 2.2** Tampilan didalam web

```

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

    if($user>7){
        echo ("Error: Username harus lebih dari 7 huruf<br>");
        $x = true;
    }
    if (!preg_match("/[A-Z]/", $password) ) {
        echo "Error: Password harus mengandung 1 huruf kapital<br>";
        $x = true;
    }
    if (!preg_match("/[a-z]/", $password)) {
        echo "Error: Password harus mengandung 1 huruf kecil<br>";
        $x = true;
    }
    if (!preg_match("/[^a-zA-Z\d]/", $password)) {
        echo "Error: Password harus mengandung 1 special character<br>";
        $x = true;
    }
    if (!preg_match("/[0-9]/", $password)) {
        echo "Error: Password harus mengandung 1 digit<br>";
        $x = true;
    }
    if($pass>10){
        echo "Error: Password harus lebih dari 10<br>";
        $x = true;
    }
    if( $x == false ){
        echo "Login berhasil";
    }
}
?>

```

**Gambar 2.3** Isi pembukaan php

### Login

Username :

Password :

Error: Username harus lebih dari 7 huruf  
 Error: Password harus mengandung 1 huruf kapital  
 Error: Password harus mengandung 1 huruf kecil  
 Error: Password harus mengandung 1 special character  
 Error: Password harus mengandung 1 digit

**Gambar 2.4** Tampilan saat kondisinya tidak terpenuhi

### Login

Username :

Password :

Login berhasil

**Gambar 2.5** Tampilan saat semua kondisinya terpenuhi