

**LAPORAN HASIL PRAKTIKUM  
PEMROGRAMAN WEB DAN MOBILE**



**Nama** : Ahmad Daffa Fahrezi  
**NIM** : 193010503008  
**Kelas** : C  
**Modul** : II (Form Handling)

**JURUSAN 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**

##### **1.2.1. Pengertian Form**

(DOSEN, 2016)Form merupakan salah satu bentuk halaman web yang digunakan untuk menerima masukan dari pengguna, untuk selanjutnya masukan dari pengguna tersebut diolah menggunakan bahasa pemrograman web, baik secara server side scripting(misalkan PHP, JSP) ataupun client-side scripting (javascript).

Form dapat digunakan untuk berbagai keperluan seperti keperluan login, transaksi penjualan, mengumpulkan informasi atau meminta umpan balik dari pengguna, menawarkan barang/jasa secara on-line dan sebagainya. Form sangat sering kita jumpai dalam satu aplikasi sistem informasi berbasis web.

##### **1.2.2. Anatomi dari form**

Standard atau format umum penulisan dari form adalah sebagai berikut :

```
<Form      metode="post/get"
action="...">
.....
.....
.....
```

```
</form>
```

Berikut ini adalah beberapa variasi yang dapat diterapkan pada atribut **<form>** :

```
<FORM ACTION=action base> form tags </FORM>  
<FORM METHOD=method> form tags </FORM>  
<FORM ENCTYPE=media type> form tags </FORM>  
<FORM ACTION=action base TARGET="target  
window name"> form tags  
</FORM> <FORM SCRIPT=URL> form tags </FORM>
```

**\_blank**

**\_self**

**\_parent**

**\_top**

Di antara elemen **<form>** dan **</form>** digunakan elemen – elemen HTML **<input>** yang berfungsi untuk mendefinisikan input yang akan dimasukkan oleh pengguna. Format umum dari penulisan elemen HTML **<input>** adalah sebagai berikut :

```
<input  
  name="name"  
  size="number"  
  type="text"|"checkbox"|"radio"|"submit"|"reset"
```

```
value="value"  
checked >
```

Pada jenis masukan (input) dari halaman web minimal harus mempunyai atribut jenis (type), nama (name) dan nilai (value) default. Disamping atribut type, masing-masing elemen input memiliki atribut-atribut yang lain dan beberapa atribut tersebut berlaku untuk beberapa jenis elemen input.

### 1.2.3. Get dan Post Pada PHP dan HTTP.

(Anakit, 2018) Berikut adalah kegunaan dari Get dan post

#### **METODE GET.**

- Menampilkan variabel pada URL pencarian
- Kurang aman
- batasan panjang string hanya sampai 2047 karakter
- Menggunakan \$\_GET dalam pengambilan variabel
- Lebih sering digunakan untuk input data melalui link
- Penggunaan pada data-data yang kurang penting

#### **METODE POST.**

- Menyembunyikan variabel pada URL pencarian
- Lebih aman
- Tidak ada batasan panjang string dalam metode post
- Menggunakan \$\_POST dalam pengambilan variabel
- Lebih sering digunakan pada input data melalui form
- Penggunaan pada data-data penting seperti password

Dari kedua penjelasan perbedaan POST dan GET diatas tentunya kita harus menggunakan tempat atau program yang mana akan menggunakan POST atau GET. Karena dari penjelasan perbedaanya metode Get diyakini sangat rentan terhadap hacker karena informasi penting atau informasi yang dikirim tertera didalam URL pencarian. Sedangkan metode POST diyakini aman karena hanya menampilkan alamat tujuan atau alaman laman tertentu pada URL.

#### 1.2.4. Memvalidasi data Form dengan PHP

(Praktikum, n.d.)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()`).

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().

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.

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:

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:

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:

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 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.

## BAB II

### PEMBAHASAN

```
?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $user = $_request["user"];
    $pass = $_request["pass"];
    $_user = strlen($user);
    $_pass = strlen($pass);
    $x = false;

    if ($_user > 7) {
        echo "Username harus kurang dari 7 karakter<br>";
        $x = true;
    }

    if (!preg_match("/[A-Z]/", $pass)) {
        echo "Password harus terdapat huruf kapital<br>";
        $x = true;
    }

    if (!preg_match("/[a-z]/", $pass)) {
        echo "Password harus terdapat huruf kecil<br>";
        $x = true;
    }

    if (!preg_match("/[0-9]/", $pass)) {
        echo "Password harus terdapat karakter khusus (angka)<br>";
        $x = true;
    }

    if ($_pass < 10) {
        echo "Password harus lebih dari 10 karakter<br>";
        $x = true;
    }

    if ($x == false) {
        echo "Username dan Password memenuhi syarat";
    }
}
?>
```



Pada source code diatas adalah penggunaan request method post. `$_SERVER["REQUEST_METHOD"]` digunakan untuk memeriksa apakah form sudah disubmit. 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.

Variabel superglobal PHP `$_GET` dan `$_POST` digunakan untuk mengumpulkan data-form. 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`.

Pernyataan `if` digunakan untuk Mengeksekusi beberapa kode jika satu kondisi benar. `Strlen` adalah tipe data berfungsi menghitung panjang dari string dari inputan yang nantinya akan diinput oleh user yang berupa username dan password.

`Echo` berguna untuk menampilkan output di PHP. `Echo` bisa menggunakan banyak parameter sedangkan `print` bisa menggunakan satu argument.

```
<<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>mod2</title>
</head>
<body>
  <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST"
  >
    <ul>
      <li>
        <label for="user">Username<br></label>
        <input type="text" name="user" id="user">
      </li>
      <li>
        <label for="pass">Password<br></label>
        <input type="text" name="pass" id="pass">
      </li>
      <li>
```

```

        <button type="submit"> submit </button>
    </li>
</ul>
</form>

</body>
</html>

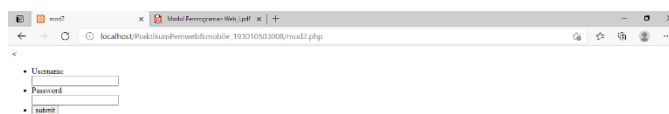
```

Pada bagian diatas adalah Script PHP yang bisa diletakkan di manapun dalam dokumen, dimulai dengan tanda “<”. Dokumen PHP disimpan dengan ekstensi “.php”, dimana file ini bisa menyimpan tag HTML dan beberapa script PHP. Setiap statement dalam PHP harus diakhiri dengan semicolon “;”.

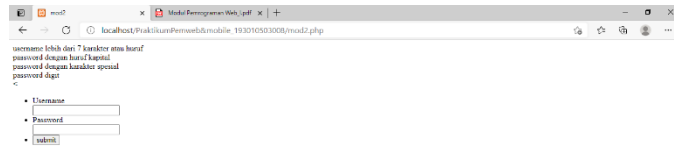
Meta tag sendiri adalah rangkaian kode HTML yang biasa digunakan untuk mempermudah Google dan browser untuk mengidentifikasi seluruh konten yang ada dalam website tersebut. Pada saat membuat satu halaman website dengan memberikan tag **<meta charset="UTF-8">** berarti halaman tersebut telah memberi informasi terhadap browser dan search engine untuk melakukan pengkodean karakter sesuai ketentuan UTF-8.

`$_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.

Berikut adalah tampilan dari programnya :



Gambar 2.1 output program



Gambar 2.2. output program false

## **BAB III**

### **KESIMPULAN**

Form HTML dibentuk menggunakan tag, dan tag form memiliki attribute method yang digunakan untuk menentukan bagaimana cara form mengirim data ke server (data dikirim ke halaman yang diset di dalam attribute action).

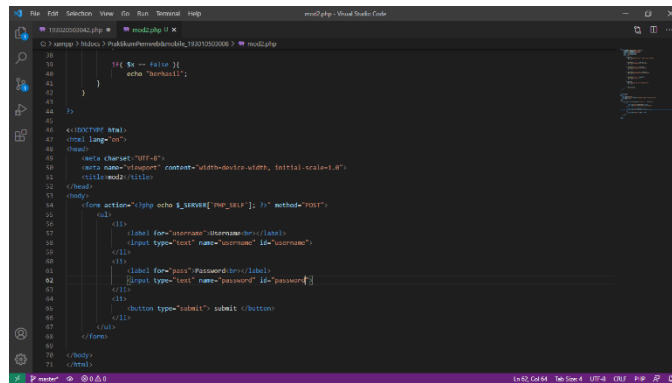
`$_POST` termasuk dalam kategori Predefined Variable yang disediakan oleh library PHP. Kegunaan dari `$_POST` hampir sama dengan `$_GET` yaitu digunakan untuk mengambil suatu nilai yang dikirimkan oleh form bedanya `$_POST` mengambil nilai yang dikirim oleh form dengan `method="post"`

Semua nilai / informasi yang dikirimkan oleh form yang menggunakan method POST tidak akan terlihat oleh user yang mengakses, dikarenakan informasi yang dikirim akan tidak ditampilkan di Address Bar Web Browser. Selain **`$_POST`** juga tidak memiliki batasan pada jumlah informasi yang dikirim.

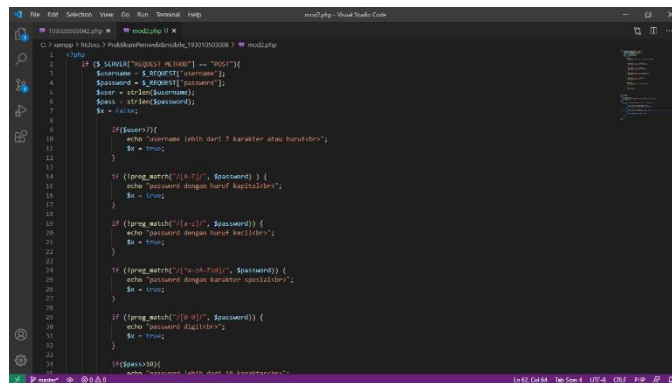
## DAFTAR PUSTAKA

- Anakit. (2018). *Metode Get dan Post (Penjelasan Dengan Contoh Program)*.  
2018/11. <https://www.anakit.id/2018/11/metode-get-dan-post-penjelasan-dengan.html>
- DOSEN, L. (2016). *Pemrograman Web I Pengertian dan fungsi form html*.  
2016/11. <https://www.lowongandosen.com/2016/11/pengertian-dan-fungsi-form-html.html>
- Praktikum, K. (n.d.). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

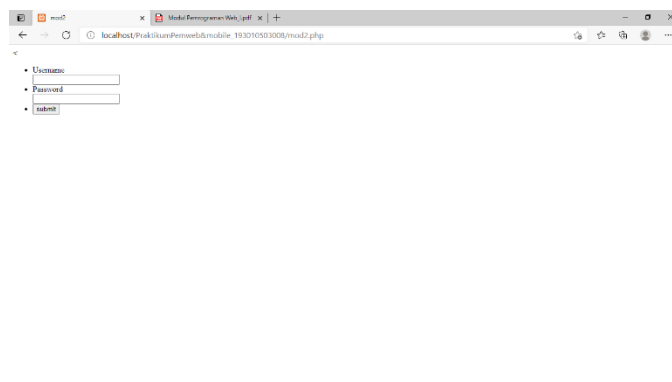
## LAMPIRAN



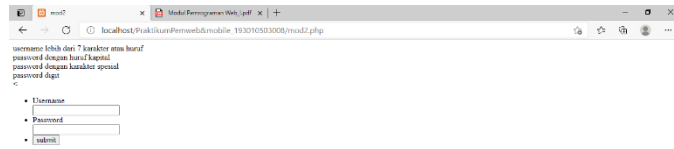
### Gambar 1. Sourcode program



**Gambar 2. Sourcecode program 2**



**Gambar 3. output program**



**Gambar 4 output program false**