

**MODUL**

# **PEMOGRAMAN WEB II**

Oleh:

CHALIFA CHAZAR

# MODUL 10

## Penanganan Form HTML

### Tujuan:

Mahasiswa memahami cara penanganan permintaan dalam kode PHP yang dikirimkan melalui form HTML untuk menyelesaikan permasalahan tertentu.

### Pustaka:

Raharjo, B. (2015): *Mudah Belajar PHP Teknik Penggunaan Fitur-Fitur Baru Dalam PHP 5*. Informatika. Bandung

Prasetio, A. (2015): *Buku Pintar Webmaster*.

Form berperan sebagai suatu antarmuka (*user interface*) agar user dapat berkomunikasi dengan server web. Setiap permintaan yang diminta oleh user (disebut *request*) akan di respon oleh aplikasi, kemudian hasilnya akan dikembalikan ke user melalui web browser.

### 1. Membuat Form HTML

Untuk membuat form di dalam halaman web, kita dapat menggunakan tag `<form>` dan `</form>`. Dalam setiap form, kita juga perlu menyertakan element `<input>` dengan tipe submit (berupa tombol) yang berfungsi untuk mengirimkan data dari form ke server web.

Dalam penggunaan tag `<form>`, terdapat dua atribut yang penting yang perlu diketahui, yaitu:

- Atribut `action`
- Atribut `method`

#### Atribut `action`

Atribut `action` digunakan untuk menunjuk file PHP yang akan digunakan untuk memproses data yang dikirim melalui form. File tersebut akan dieksekusi ketika pengguna web memilih tombol submit.

Chalifa Chazar – edu.script.id – chalifa.chazar@gmail.com

Perhatikan contoh berikut ini.

```
<form action="proses.php">  
  ...  
</form>
```

Pada contoh diatas, file `proses.php` akan dipanggil akan dieksekusi pada saat user memilih tombol `submit`.

### Atribut `method`

Atribut `method` berfungsi untuk menentukan cara atau metode informasi dari suatu form dikirimkan ke web server. Kita dapat menggunakan salah satu dari dua metode: `get` dan `post`.

Perhatikan contoh berikut ini.

```
<form action="proses.php" method="get">  
  ...  
</form>
```

atau

```
<form action="proses.php" method="post">  
  ...  
</form>
```

Pada contoh diatas, file `proses.php` akan dipanggil akan dieksekusi pada saat user memilih tombol `submit`.

## 2. Menggunakan Metode `get`

Dengan menggunakan metode `get`, data yang dikirimkan ke server web akan ditampilkan pada URL yang terdapat dalam web browser. Format data yang akan ditampilkan di dalam URL adalah pasangan nama dan nilai, yang ditulis dalam format `nama=nilai`. `nama` yang dimaksud disini adalah nama dari elemen kontrol yang menyimpan data (misalnya elemen `<input>` dengan tipe `text`), sedangkan `nilai` adalah data yang dimasukan oleh user ke dalam elemen tersebut.

Perhatikan contoh berikut ini.

```
?nama=Bimo
```

Dimana, "nama" adalah elemen kontrol dan "bimo" adalah nilainya. Jika data yang dikirimkan melalui form lebih dari satu, maka akan ditambahkan tanda dan &.

Berikut ini contoh kode yang akan menunjukkan cara kerja metode **get**.

Simpan kode di bawah ini ke dalam format HTML (file .html) dan tempatkan di dalam direktori **htdocs** dalam server **Xampp**.

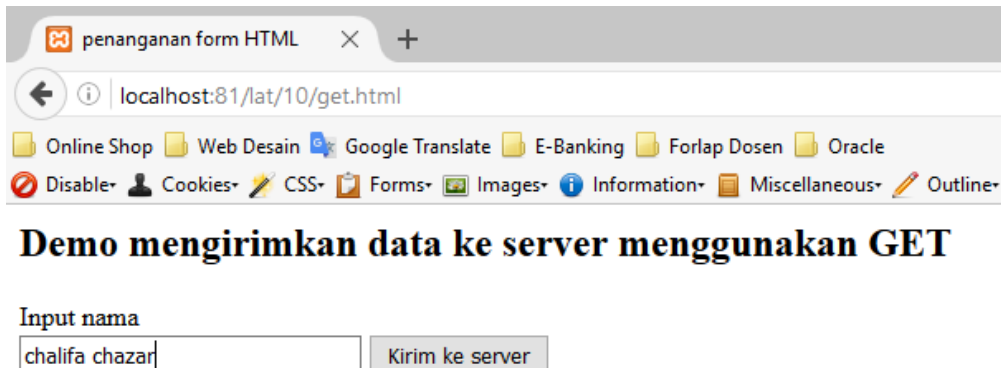
```
<!--nama file: get.html-->
<html>
<head>
    <title>penanganan form HTML</title>
</head>
<body>
<h2>Demo mengirimkan data ke server menggunakan GET</h2>
<form action="get.php" method="get">
Input nama <br>
<input type="text" name="txtname" size="30" />
<input type="submit" name="btsubmit" value="Kirim ke server" />
</form>
</body>
</html>
```

Simpan kode di bawah ini ke dalam format PHP (file .php) dan tempatkan di dalam direktori **htdocs** dalam server **Xampp**.

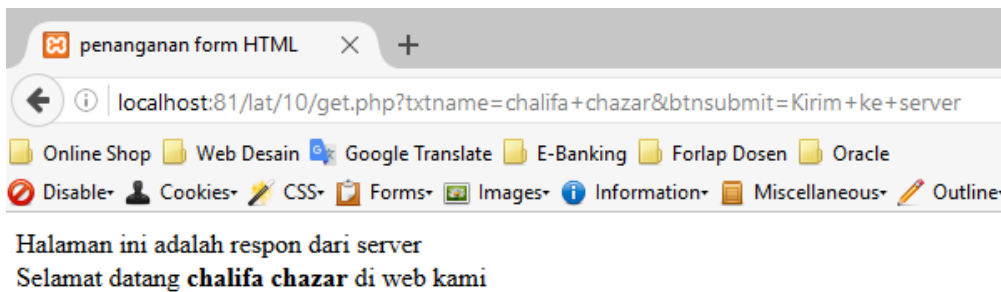
```
<!--nama file: get.php-->
<html>
<head>
    <title>penanganan form HTML</title>
</head>
<body>
<?php
$submit = $_GET['btsubmit'];
if (isset($submit)) {
    $nama = $_GET['txtname'];
    echo "Halaman ini adalah respon dari server";
    echo '<br>';
    echo "Selamat datang " . '<strong>' . $nama . '</strong>' . " di web kami";
}
?>
</body>
</html>
```

Eksekusi program ke dalam web browser dengan memanggil alamat **get.html**.

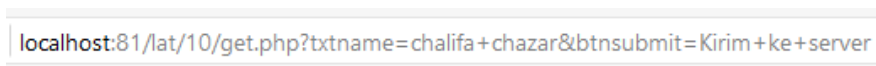
Berikut ini adalah tampilan hasil yang diperoleh ketika program dieksekusi.



Pada saat user mengirim data menggunakan tombol submit, maka server web akan meresponnya dengan cara memanggil dan mengeksekusi file `get.php` sehingga tampilan halaman web akan berubah menjadi sebagai berikut.



Karena menggunakan metode `get` dalam proses pengiriman informasinya, maka data yang dikirim oleh user akan ditampilkan pada bagian URL di adres bar yang terdapat pada web browser.



Dalam file `get.php`, coba perhatikan script berikut ini.

```
$submit = $_GET['btsubmit'];
$name = $_GET['txtname'];
```

Baris diatas berfungsi untuk menangkap data yang dikirimkan melalui elemen-elemen yang terdapat di dalam form. Karena data dikirimkan menggunakan `get`, maka cara penangkapan atau pengambilannya adalah dengan menggunakan `$_GET[""]`.

Pada contoh, `btnsubmit` dan `txtname` adalah nama-nama dari elemen kontrol yang ada dalam form, sedangkan `$submit` dan `$nama` adalah variabel normal yang kita buat atau definisikan sendiri untuk keperluan program.

Pada contoh, dalam file `get.php` juga terdapat bagian penting lainnya, yaitu fungsi `isset()`. Fungsi ini digunakan untuk mengetahui bahwa isi dari suatu variabel yang disertakan sebagai parameternya, apakah kosong atau tidak. Jika kosong (telah di-set dengan nilai tertentu) maka fungsi `isset()` akan mengembalikan nilai `true`.

### 3. Menggunakan Metode `post`

Nilai `post` pada atribut `method` dari tag `<form>` pada dasarnya memiliki fungsi yang sama seperti `get`. Perbedaannya, pada metode pengiriman ini, data yang dimasukkan oleh user tidak akan ditambahkan/ditampilkan ke dalam URL. Dengan demikian, pengiriman data yang dilakukan menggunakan metode `post` akan lebih terjaga kerahasiaannya.

Berikut ini contoh kode yang akan menunjukkan cara kerja metode **`post`**.

Simpan kode di bawah ini ke dalam format HTML (file `.html`) dan tempatkan di dalam direktori **`htdocs`** dalam server **`Xampp`**.

```
<!--nama file: post.html-->
<html>
<head>
    <title>penanganan form HTML</title>
</head>
<body>
<h2>Demo mengirimkan data ke server menggunakan POST</h2>
<form action="post.php" method="post">
Input nama : <br>
<input type="text" name="txtname" size="30" />
<input type="submit" name="btnsubmit" value="Kirim ke server" />
</form>
</body>
</html>
```

Simpan kode di bawah ini ke dalam format PHP (file .php) dan tempatkan di dalam direktori **htdocs** dalam server **Xampp**.

```
<!--nama file: post.php-->
<html>
<head>
    <title>penanganan form HTML</title>
</head>
<body>
<?php
$submit = $_POST['btnsubmit'];
if (isset($submit)) {
    $nama = $_POST['txtname'];
    echo "Halaman ini adalah respon dari server";
    echo '<br>';
    echo "Selamat datang " . '<strong>' . $nama . '</strong>' . " di web kami";
}
?>
</body>
</html>
```

Eksekusi program ke dalam web browser dengan memanggil alamat **post.html**.

Hasil program ketika dieksekusi terlihat sama seperti contoh penggunaan metode get, akan tetapi coba perhatikan bagian URL. Karena menggunakan metode post maka data tidak ditambahkan/ditampilkan dalam *address bar*.



localhost:81/lat/10/post.php

Hal lain yang perlu diperhatikan adalah cara pengambilan data dari form di dalam file post.php. Karena kali ini data dikirim menggunakan metode post maka cara pengambilannya menjadi sebagai berikut.

```
$submit = $_POST['btnsubmit'];
$nama = $_POST['txtname'];
```

### Tugas 1

Buat program untuk menampilkan inputan data mahasiswa seperti gambar berikut ini.

<u>Tampilan awal</u>	<u>Tampilan hasil</u>
<b>Input Data Mahasiswa</b>	Halaman ini adalah respon dari server
NIM : <input type="text"/>	Nim : 36
Nama : <input type="text"/>	Nama : chalifa
Jurusan : <input type="text"/>	Jurusan : informatika
Email : <input type="text"/>	Email : chal@script.id
Password : <input type="text"/>	Password : 12345
<input type="button" value="Simpan Data"/> <input type="button" value="Reset"/>	

### Tugas 2

Buat program untuk menghitung luas persegi panjang seperti gambar berikut ini.

### **Menghitung Luas Persegi Panjang dengan Form**

Nilai Panjang :	<input type="text"/>
Nilai Lebar :	<input type="text"/>
Luas :	<input type="text"/>
<input type="button" value="Proses"/> <input type="button" value="Reset"/>	