

## Bab 4

# Penanganan Form

- ❖ Berbagai Cara Penanganan Form
- ❖ Form Input Type TEXT dan PASSWORD
- ❖ Form Input Type RADIO
- ❖ Form Input Type CHECKBOX
- ❖ Form Input Type COMBO BOX
- ❖ Form Input Type TEXTAREA

Form inputan dibuat dengan tag-tag HTML. Halaman yang mengandung form murni (tidak ada *script* php) tidak harus disimpan dalam bentuk php, bisa dalam bentuk html.

Untuk merancang sebuah form inputan, setidaknya ada 3 (tiga) hal penting, yi :

1. **METHOD**

Method dari sebuah form menentukan bagaimana data inputan form dikirim. Method ini ada dua macam, yaitu GET dan POST. Method ini menentukan bagaimana data inputan dikirim dan diproses oleh PHP.

2. **ACTION**

Action dari sebuah form menentukan dimana data inputan dari form diproses. Jika action ini dikosongkan, maka dianggap proses form terjadi di halaman yang sama. Jadi halaman form dan halaman proses bisa saja dipisah atau dijadikan satu.

3. **SUBMIT BUTTON**

Submit button merupakan sebuah tombol (pada umumnya) yang berfungsi sebagai *trigger* pengiriman data dari form inputan. Jika tombol ini ditekan, maka data form akan dikirimkan (diproses) di halaman yang sudah ditentukan pada atribut action.

### Berbagai Cara Penanganan Form

Cara 1 : Menyatukan antara Form dan Proses

Proses pengolahan form dilakukan di halaman yang sama dengan form inputannya. Jika proses penanganan form berada di satu halaman, maka *value* atribut **action** pada tag **form** tidak perlu diisi (dikosongkan).

Program 4.1

Nama File : *input01.php*

Deskripsi : *Program Contoh pengolahan form dimana antara form inputan dan proses pengolahan inputan berada dalam satu halaman.*

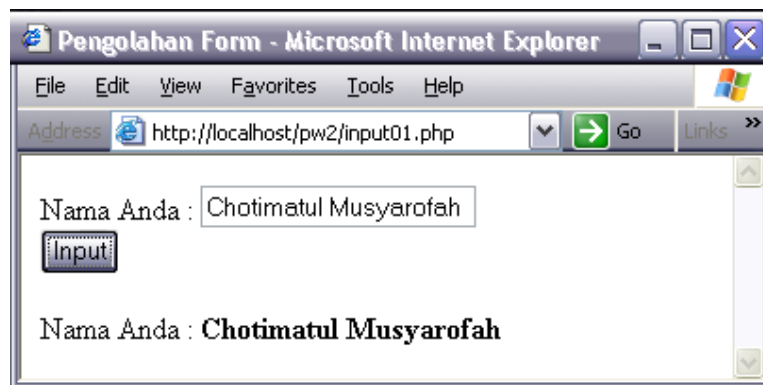
```

1  <html>
2  <head><title>Pengolahan Form</title></head>
3  <body>
4      <FORM ACTION="" METHOD="POST" NAME="input">
5          Nama Anda : <input type="text" name="nama"><br>
6          <input type="submit" name="Input" value="Input">
7      </FORM>
8  </body>
9  </html>
10
11 <?php
12 if (isset($_POST['Input'])) {
13     $nama = $_POST['nama'];
14     echo "Nama Anda : <b>$nama</b>";
15 }
16 ?>
    
```

### Penjelasan Program 4.1

Beberapa hal yang perlu diperhatikan dari program 4.1 di atas, di antaranya mengenai nama setiap komponen form karena nama ini akan menjadi index array dalam PHP. Pada program 4.1 di atas, *value* atribut **action** pada tag **form** tidak diisi (baris 4), ini berarti bahwa proses pengolahan form berada di halaman yang sama. Selanjutnya (baris 4) method yang digunakan dalam penanganan form adalah POST. Cara ini lebih disarankan dalam penanganan sebuah form inputan.

Mulai baris ke-11 hingga 16, terdapat script / program PHP yang akan menangani (mengolah) nilai yang diinputkan melalui form. Letak proses inputan ini boleh sebelum atau sesudah form, tergantung kebutuhan. Pada baris ke-12, terdapat pemeriksaan kondisi apakah tombol dengan nama "**Input**" (perhatikan index array **\$\_POST** dan bandingkan dengan name tombol **submit** pada form) benar-benar ditekan atau tidak oleh user. Selanjutnya pada baris ke-13, nilai inputan dari form akan diambil dengan cara mengakses array **\$\_POST** sesuai dengan komponen form yang akan diambil. Pada baris 13 ini, isi komponen inputan dengan nama "**nama**" akan diambil dan dimasukkan ke variabel **\$nama**. Index array **\$\_POST** pada baris 13 harus sama dengan *value* atribut **name** pada baris ke-5.



Gambar 4.1. Tampilan Program 4.1

### Cara 2 : Memisahkan antara Form dan Proses

Proses pengolahan form dilakukan di halaman yang terpisah dengan form inputannya. Jika proses penanganan form berada di halaman yang berbeda, maka *value* atribut **action** pada tag **form** harus diisi dengan alamat halaman tempat proses pengolahan form.

### Program 4.2

Nama File : *input02.php*

Deskripsi : Program menampilkan form inputan dengan method POST

```

1 <html>
2   <head><title>Pengolahan Form</title></head>
3   <body>
4     <FORM ACTION="proses02.php" METHOD="POST" NAME="input">
5       Nama Anda : <input type="text" name="nama"><br>
6       <input type="submit" name="Input" value="Input">
7     </FORM>
8   </body>
9 </html>

```

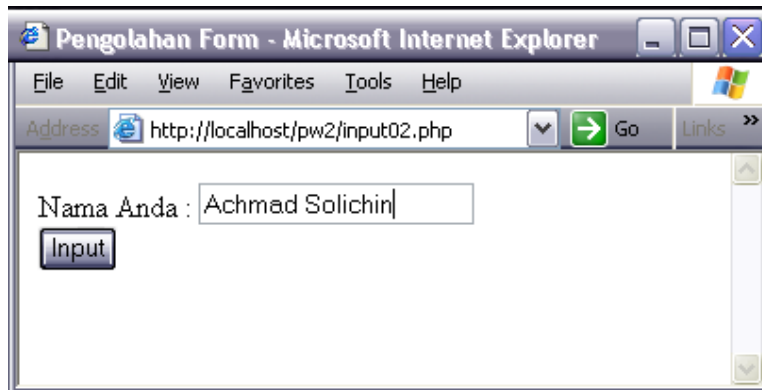
**Program 4.3**

Nama File : proses02.php

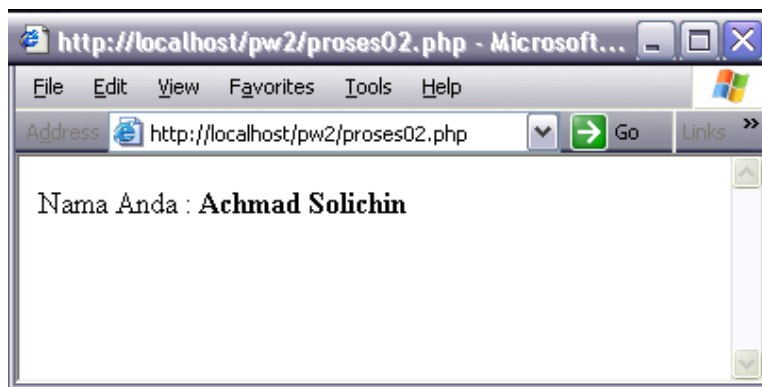
Deskripsi : Program penanganan form inputan untuk Program 4.2.

```

1 <?php
2 if (isset($_POST['Input'])) {
3     $nama = $_POST['nama'];
4     echo "Nama Anda : <b>$nama</b>";
5 }
6 ?>
```



Gambar 4.2. Tampilan Program 4.2



Gambar 4.3. Tampilan Program 4.3

**Program 4.4**

Nama File : input03.php

Deskripsi : Program menampilkan form inputan dengan method GET

```

1 <html>
2 <head><title>Pengolahan Form</title></head>
3 <body>
4 <FORM ACTION="proses03.php" METHOD="GET" NAME="input">
5     Nama Anda : <input type="text" name="nama"><br>
6     <input type="submit" name="Input" value="Input">
7 </FORM>
8 </body>
9 </html>
```

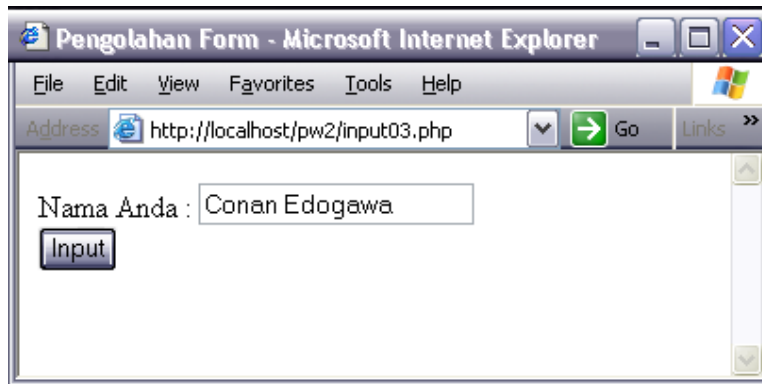
**Program 4.5**

Nama File : proses03.php

Deskripsi : Program penanganan form inputan untuk Program 4.4.

```

1 <?php
2 if (isset($_GET['Input'])) {
3     $nama = $_GET['nama'];
4     echo "Nama Anda : <b>$nama</b>";
5 }
6 ?>
    
```



Gambar 4.4. Tampilan Program 4.4



Gambar 4.5. Tampilan Program 4.5

**Penjelasan Program 4.2, 4.3, 4.4 dan 4.5**

Program 4.2 dan program 4.3 pada prinsipnya sama dengan program 4.1. Hasilnya pun sama. Akan tetapi, proses penanganan form (PHP) terpisah dengan tampilan form inputannya (HTML). Pada program 4.2 dan 4.3, penanganan form menggunakan method POST, sedang pada program 4.4 dan 4.5, penanganan form dilakukan dengan method GET. Perhatikan, method ini menentukan bagaimana sebuah form diproses dan bagaimana variabel inputan diambil. Bandingkan baris ke-2 dan ke-3 pada program 4.3 dan program 4.5 !

**Form Input Type TEXT dan PASSWORD**

**Program 4.6**

Nama File : input04.php

Deskripsi : Program menampilkan form inputan text dalam jumlah banyak.

```

1 <html>
2 <head><title>Pengolahan Form ~ Text</title></head>
    
```

```

3     <body>
4     <FORM ACTION="proses04.php" METHOD="POST" NAME="input">
5     Sahabat-sahabat Dekatku<br>
6     <input type="text" name="nama1"><br>
7     <input type="text" name="nama2"><br>
8     <input type="text" name="nama3"><br>
9     <input type="text" name="nama4"><br>
10    <input type="submit" name="Input" value="Input">
11    </FORM>
12  </body>
13 </html>

```

Program 4.7

Nama File : proses04.php

Deskripsi : Program penanganan form inputan untuk Program 4.6.

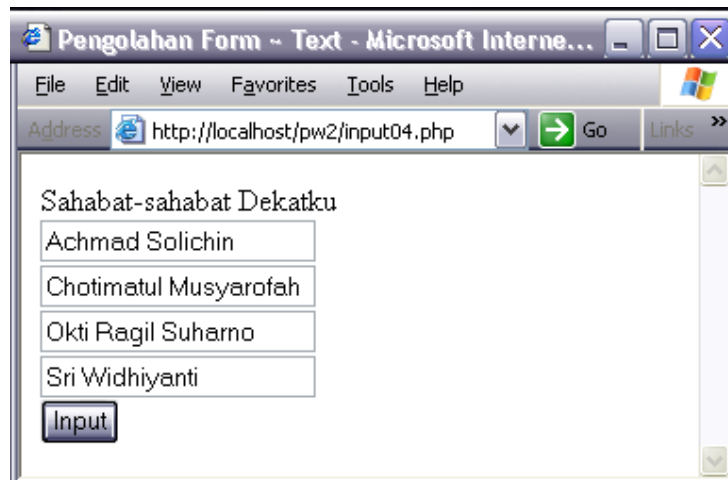
```

1 <?php
2 if (isset($_POST['Input'])) {
3     $nama1 = $_POST['nama1'];
4     $nama2 = $_POST['nama2'];
5     $nama3 = $_POST['nama3'];
6     $nama4 = $_POST['nama4'];
7     echo "<b>Nama Sahabat-sahabat Dekatku :</b> <br>";
8     echo $nama1. "<br>";
9     echo $nama2. "<br>";
10    echo $nama3. "<br>";
11    echo $nama4. "<br>";
12 }
13 ?>

```

Penjelasan Program 4.6 dan 4.7

Baris 3-6 program 4.7 merupakan proses pengambilan nilai dari masing-masing form inputan text yang terdapat pada program 4.6. Selanjutnya variabel ini ditampilkan di layar (baris 8-11, program 4.7).



Gambar 4.6. Tampilan Program 4.6



Gambar 4.7. Tampilan Program 4.7

**Program 4.8**

Nama File : *input05.php*

Deskripsi : Program menampilkan form login (inputan text dan password).

```

1 <html>
2   <head><title>Login Here</title></head>
3   <body>
4     <FORM ACTION="proses05.php" METHOD="POST" NAME="input">
5       <h2>Login Here...</h2>
6       Username : <input type="text" name="username"><br>
7       Password : <input type="password" name="password"><br>
8       <input type="submit" name="Login" value="Login">
9       <input type="reset" name="reset" value="Reset">
10    </FORM>
11  </body>
12 </html>

```

**Program 4.9**

Nama File : *proses05.php*

Deskripsi : Program sederhana untuk memeriksa inputan username dan password pada program 4.8

```

1 <?php
2 if (isset($_POST['Login'])) {
3   $user = $_POST['username'];
4   $pass = $_POST['password'];
5   if ($user == "achmatim" && $pass == "123") {
6     echo "<h2>Login Berhasil</h2>";
7   } else {
8     echo "<h2>Login Gagal</h2>";
9   }
10 }
11 ?>

```

**Penjelasan Program 4.8 dan 4.9**

Program 4.8 akan menampilkan form login sederhana yang terdiri dari inputan **username** dan **password**. Selanjutnya nilai yang diinput akan diproses di program 4.9. Jika username dan password diinput dengan benar maka akan ditampilkan pesan berhasil (lihat gambar) dan jika login salah maka akan ditampilkan pesan kesalahan (lihat gambar).