

## **BAB II**

### **PEMROSESAN FORM**

#### **A. TUJUAN**

- Memahami konsep dasar transfer data dari form HTML.
- Mampu menangani masukan data dari form HTML.
- Mampu melakukan inisiasi nilai pada elemen-elemen form HTML.

#### **B. ALOKASI WAKTU**

4 JS (4 x 50 menit)

#### **C. PETUNJUK**

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

#### **D. DASAR TEORI**

##### **1. Pemrosesan Form**

Pemrosesan form (*form processing*) merupakan operasi mendasar pada aplikasi web. Dalam konteks pengembangan aplikasi web dinamis, langkah ini pasti akan dilakukan. Sebagai contoh, untuk menerima masukan dari user, tentu diperlukan sekali form isian dan cara pemrosesannya. Begitu pula halnya ketika administrator ingin masuk ke sistem, tentu memerlukan suatu antarmuka penghubung. Intinya, keberadaan form dan pemrosesannya mutlak diperlukan dalam membangun aplikasi web yang interaktif.

Pada aplikasi web, teknik pengiriman form dapat dilakukan melalui tiga metode: POST, GET, dan kombinasi keduanya. Dalam konteks PHP, parameter-parameter GET dapat dibaca melalui superglobal `$_GET`, sedangkan POST melalui `$_POST`. Selain itu, keduanya juga dapat dibaca menggunakan `$_REQUEST`.

## E. LATIHAN

### 1. Menangani Masukan Data

Sebagaimana diketahui, kita dapat menggunakan metode GET atau POST untuk mengirim informasi ke kode PHP.

Contoh pengaksesan data dengan menggunakan metode GET diperlihatkan sebagai berikut:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Metode GET</title>
</head>

<body>

<form action="<?php $_SERVER['PHP_SELF'];?>" method="get">
Nama
<input type="text" name="nama" /> <br />

<input type="submit" value="OK" />
</form>

<?php
if (isset($_GET['nama'])) {
    echo 'Hallo, ' . $_GET['nama'];
}
?>

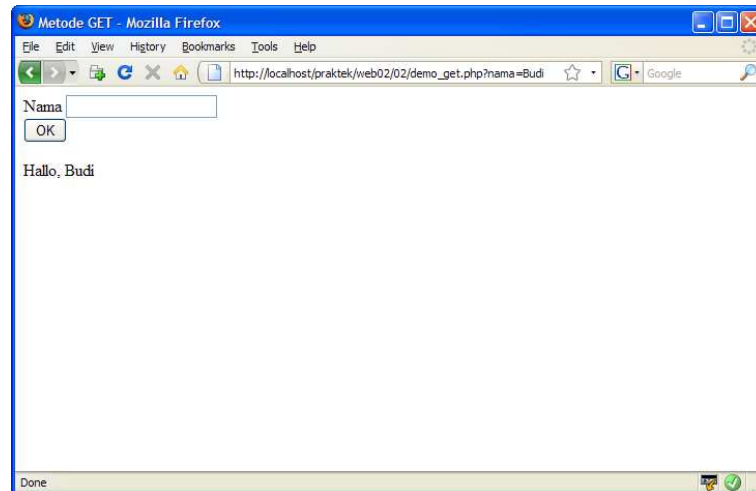
</body>
</html>
```

Simpan kode program di atas dengan nama misalnya demo\_get.php. Pada contoh di atas, ada beberapa hal penting yang perlu diperhatikan:

- Nilai `$_SERVER['PHP_SELF']` (variabel superglobal) menyatakan bahwa form akan ditangani oleh *current script*.
- Nilai `get` pada atribut `method` menyatakan bahwa transfer data menggunakan metode GET.
- Untuk mendapatkan nilai variabel, kita memanfaatkan superglobal `$_GET`.

Untuk mengetahui bagaimana kode program bekerja menangani transfer data, jalankan dari browser.

Begitu kita mengisi nilai dan melakukan submisi form, maka nama variabel beserta nilai yang kita isikan tadi akan ditampilkan pada alamat URL. Sebagai contoh, hasilnya diperlihatkan pada Gambar 1.



Gambar 1. Menangani data melalui metode GET

Untuk tujuan yang sama, kita sekarang akan mencoba menggunakan pendekatan metode POST.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Metode POST</title>
</head>

<body>

<form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
Nama
<input type="text" name="nama" /> <br />

<input type="submit" value="OK" />
</form>

<?php
if (isset($_POST['nama'])) {
  echo 'Hallo, ' . $_POST['nama'];
}
?>

</body>
</html>
```

Dalam konteks PHP, parameter-parameter GET dapat dibaca melalui superglobal `$_GET`, sedangkan POST melalui `$_POST`. Selain itu, keduanya juga dapat dibaca menggunakan `$_REQUEST`.

### **Latihan Kecil**

Bagaimana jika metode GET ditangani dengan superglobal `$_POST`; dan sebaliknya, metode POST ditangani dengan `$_GET`? Jelaskan!

Coba manfaatkan `$_REQUEST` untuk menggantikan fungsi `$_GET` atau `$_POST`.

Sebagai tambahan, kita juga dapat mengidentifikasi jenis metode request dengan memanfaatkan elemen REQUEST\_METHOD milik superglobal \$\_SERVER.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Identifikasi Metode</title>
</head>

<body>

<form action="<?php $_SERVER['PHP_SELF'];?>" method="get">
Nama
<input type="text" name="nama" /> <br />

<input type="submit" value="OK" />
</form>

<?php
if (isset($_REQUEST['nama'])) {
    echo 'Metode, ' . $_SERVER['REQUEST_METHOD'];
}
?>

</body>
</html>
```

## 2. Prefilling Text Field

Begitu form disubmisi, normalnya nilai elemen-elemen form akan di-reset (atau dikosongkan). Adapun jika diperlukan, kita sebenarnya juga dapat menahan agar nilainya tetap ada di dalam elemen, misalnya text field.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Prefilling Text Field</title>
</head>

<body>

<form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
Nama
<input type="text" name="nama"
      value="<?php
        echo isset($_POST['nama']) ? $_POST['nama'] : '';
      ?>"
/> <br />

<input type="submit" value="OK" />
</form>

<?php
if (isset($_POST['nama'])) {
    echo $_POST['nama'];
}
?>

</body>
</html>
```

### 3. Menangani Nilai Radio Button

- Mendapatkan Nilai Radio Button

Pada prinsipnya, cara mendapatkan nilai radio button tak ubahnya elemen text field.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Data Radio Button</title>
</head>

<body>

  <form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
    Jenis Kelamin
    <input type="radio" name="sex" value="Pria" />Pria
    <input type="radio" name="sex" value="Wanita" />Wanita <br />

    <input type="submit" value="ok" />
  </form>

  <?php
  if (isset($_POST['sex'])) {
    echo $_POST['sex'];
  }
  ?>

</body>

</html>
```

#### **Latihan Kecil**

Bagaimana memberikan nilai default pada koleksi radio button?

- Prefilling Radio Button

Untuk menginisiasi nilai item radio button, kita perlu melakukan per item yang ada. Artinya, identifikasi akan dilakukan untuk setiap item guna mengetahui apakah item terkait dipilih atau tidak.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Prefill Data Radio Button</title>
</head>

<body>

  <form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
    Jenis Kelamin
    <input type="radio" name="sex" value="Pria" checked
```

```
<?php
if ($_POST['sex'] == 'Pria') {
    echo 'checked="checked"';
}
?>
/>Pria
<input type="radio" name="sex" value="Wanita"
<?php
if ($_POST['sex'] == 'Wanita') {
    echo 'checked="checked"';
}
?>
/>Wanita <br />

<input type="submit" value="ok" />
</form>

<?php
if (isset($_POST['sex'])) {
    echo $_POST['sex'];
}
?>

</body>
</html>
```

#### 4. Menangani Nilai Seleksi

Cara menangani data-data dari elemen seleksi sebenarnya tak ubahnya data dari radio button. Ini tentu dengan asumsi bahwa item yang bisa dipilih hanya satu saja.

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
<title>Data Seleksi</title>
</head>

<body>

<form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
    Pekerjaan
    <select name="job">
        <option value="Mahasiswa">Mahasiswa
        <option value="ABRI">ABRI
        <option value="PNS">PNS
        <option value="Swasta">Swasta
    </select> <br />

    <input type="submit" value="ok" />
</form>

<?php
if (isset($_POST['job'])) {
    echo $_POST['job'];
}
?>

</body>
```

```
</html>
```

## 5. Menangani Nilai Check Box

Khusus dalam penanganan check box, kita akan melibatkan tipe data array (lebih jelasnya akan dibahas di pertemuan selanjutnya). Bagaimanapun, item check box normalnya boleh dipilih lebih dari satu.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Data Checkbox</title>
</head>

<body>

  <form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
    Jenis Kelamin
    <input type="checkbox" name="hobby[]" value="Membaca"
    />Membaca

    <input type="checkbox" name="hobby[]" value="Olahraga"
    />Olahraga

    <input type="checkbox" name="hobby[]" value="Menyanyi"
    />Menyanyi <br />

    <input type="submit" value="ok" />
  </form>

  <?php
  // Ekstraksi nilai
  if (isset($_POST['hobby'])) {
    foreach ($_POST['hobby'] as $key => $val) {
      echo $key . ' -> ' . $val . '<br />';
    }
  }
  ?>

</body>

</html>
```

## **F. STUDI KASUS**

1. Buat kode program untuk penanganan seleksi dengan menambahkan preselecting.
2. Buat kode program untuk penanganan check box dengan menambahkan preselecting.

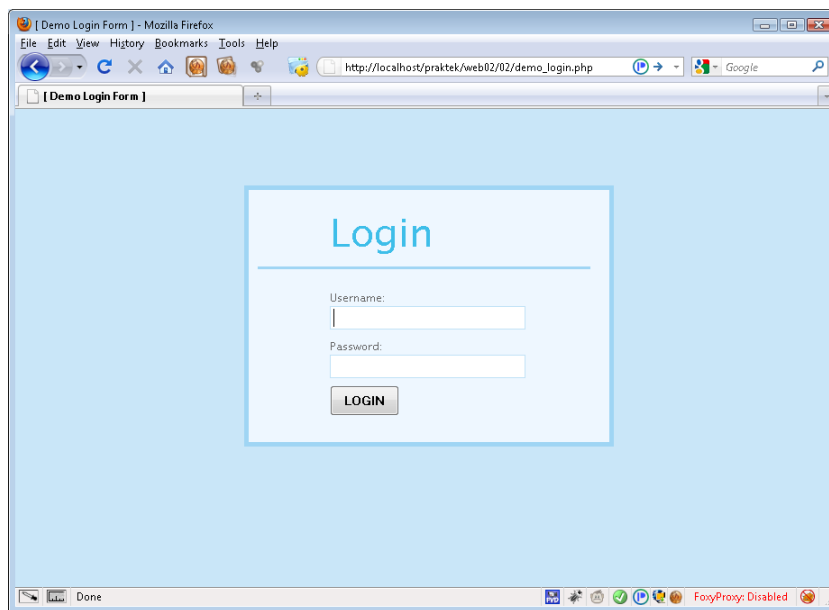


## G. TUGAS PRAKTIKUM

1. Buat aplikasi login form sederhana (tanpa database) yang terdiri dari field ID/Username dan Password. Gunakan variabel di PHP untuk mensimulasikan verifikasi data yang dimasukkan.

Spesifikasi:

- Terdapat scripting JavaScript untuk validasi awal (field tidak boleh kosong, masukan id dan password harus huruf).
- Jika field kosong dan langsung di-submit, kembalikan fokus kursor ke field pertama (id)
- Ada validasi sisi server (dari PHP) untuk memastikan bahwa nilai field adalah string.
- Jika nilai id dan password sesuai dengan pre-defined value di variabel, munculkan pesan selamat datang dan cetak nilai id. Sebaliknya, jika tidak sesuai, tampilkan pesan kegagalan.



Gambar 2. Demo login form