

MENANGANI DATA PADA FORM

Dalam hal menangani data yang dikirim ke server, PHP memiliki beberapa variabel **Superglobals** yang secara otomatis akan menangkap data berdasarkan metode (method) yang digunakan oleh pengirim. Variabel-variabel tersebut adalah:

- a. **\$_GET**, akan menangkap atau menampung data-data yang dikirimkan menggunakan metode (method) *get*.
- b. **\$_POST**, akan menangkap atau menampung data-data yang dikirimkan menggunakan metode (method) *post*.
- c. **\$_FILES**, akan menangkap atau menampung informasi file yang dikirim ke server.
- d. **\$_REQUEST**, akan menangkap atau menampung informasi file yang dikirim ke server menggunakan metode *get* maupun *post*.

Variabel-variabel tersebut bertipe data Array.

Data dikirim ke server biasanya melalui sebuah form. Metode yang digunakan ditentukan melalui atribut **method** pada elemen **<form>**. Sebagai contoh, form berikut:

```
<form name="form1" method="post" action="proses.php">
```

Form di atas akan mengirimkan data menggunakan metode *post*. Atribut **action** digunakan untuk menentukan ke mana data tersebut akan dikirim. Pada contoh di atas, form tersebut memiliki atribut **action="proses.php"**. Artinya, data akan dikirim ke sebuah halaman dengan nama file **proses.php**. Selanjutnya, file **proses.php** yang akan menangani data-data tersebut.

Dengan menggunakan metode *post*, maka PHP akan menangkap dan menampung data ke dalam variabel **\$_POST**.

Apabila pada form di atas digunakan **method="get"**, maka data yang dikirim akan ditangkap dan ditampung pada variabel **\$_GET**.

Khusus untuk metode *get*, data dapat dikirimkan melalui sebuah link atau URL. Sebagai contoh, jika terdapat *link* sebagai berikut:

<http://www.unikom.ac.id/index.php?fakultas=teknik&nim=10100246>

Maka, variabel **\$_GET** akan menangkap dan menampung dua buah data, yaitu data **"fakultas=teknik"** dan data **"nim=10100246"**. Sehingga, isi dari variabel **\$_GET** tersebut akan menjadi seperti berikut ini:

```
$_GET = Array( "fakultas" => "teknik", "nim" => "10100246" );
```

Sehingga, untuk menampilkan atau mengambil nilai dari variabel tersebut dapat dilakukan seperti contoh berikut:

```
echo $_GET["fakultas"]; //teknik  
echo $_GET["nim"]; //10100246
```

atau

```
$fak = $_GET["fakultas"];  
$nim = $_GET["nim"];
```

```
echo $fak; //teknik  
echo $nim; //10100246
```

LATIHAN 1
Menangani Data Form dan Menyimpan Data ke Dalam File

1. MEMBUAT HALAMAN FORMULIR

- Buat file baru dengan nama **formulir.php**
- Tulis kode di bawah ini ke dalam file tersebut:

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Form</title>
5   </head>
6   <body>
7     <h2>Formulir Pendaftaran</h2>
8     <p>Isi Formulir di bawah ini dengan data yang benar</p>
9     <form name="frmDaftar" id="frmDaftar" method="post" action="proses.php">
10    <table>
11      <tr>
12        <td width="40">Nama</td>
13        <td><input type="text" name="nama" size="40"></td>
14      </tr>
15      <tr>
16        <td>Alamat</td>
17        <td><textarea name="alamat" cols="35" rows="3"></textarea></td>
18      </tr>
19      <tr>
20        <td>Kota</td>
21        <td>
22          <select name="kota">
23            <option value="Kota Bandung">Kota Bandung</option>
24            <option value="Kota Bogor">Kota Bogor</option>
25            <option value="DKI Jakarta">DKI Jakarta</option>
26          </select>
27        </td>
28      </tr>
29      <tr>
30        <td></td>
31        <td><input type="submit" value="Daftar"></td>
32      </tr>
33    </table>
34  </form>
35  <a href="lihatdata.php">Lihat Data</a>
36 </body>
37 </html>
```

- Simpan file tersebut.

2. MEMBUAT FILE UNTUK MENANGANI (MEMPROSES) DATA YANG DIKIRIM KE SERVER

- a. Buat sebuah file baru dengan nama **proses.php**. File ini digunakan untuk menangani data yang dikirim ke server melalui file **formulir.php** di atas.
- b. Ketikkan kode-kode di bawah ini:

```
<?php
if ( !$_POST ){
    die( "Anda tidak berhak mengakses halaman ini" );
}

$namafile = "data.txt";
$nama = $_POST['nama'];
$salamat = $_POST['alamat'];
$kota = $_POST['kota'];
$data = $nama.'~'.$salamat.'~'.$kota;

$fp = fopen($namafile, 'a' );
fwrite( $fp, $data."\r\n" );
fclose( $fp );
echo '<p>Data Telah Disimpan</p>';
echo '<p>';
echo '<a href="lihatdata.php">Lihat Data</a>';
echo '<a href="formulir.php">Kembali ke halaman Formulir</a>';
echo '</p>';
?>
```

- c. Simpan file jika sudah selesai

Keterangan:

File ini digunakan untuk menangani data yang dikirim dengan menggunakan **method="post"** pada form. Data yang dikirim akan ditangkap oleh variabel **\$_POST** pada PHP. Jika data dikirim dengan menggunakan **method="get"**, maka data yang dikirim akan ditangkap oleh variabel **\$_GET**.

\$namafile memuat nama file yang akan digunakan untuk menyimpan data yang dikirim dari client. Variabel **\$nama**, **\$salamat**, **\$kota** diambil dari variabel **\$_POST** yang memuat data yang dikirim oleh client.

Data akan disimpan dengan format **nama~alamat~kota** pada file yang dideklarasikan pada variabel **\$namafile**.

3. MEMBUAT FILE UNTUK MELIHAT DATA YANG DISIMPAN PADA FILE

- Buat file baru dengan nama **lihatdata.php**. File ini akan mengakses file yang digunakan untuk menyimpan data.
- Ketikkan kode di bawah ini:

```
<html>
<head>
  <title>Lihat Data</title>
</head>
<body>
  <a href="formulir.php">Kembali ke Halaman Formulir</a>
  <?php
    $namafile = 'dimana/data.txt';
    if ( !file_exists( $namafile ) ){
      echo "<p>Tidak Ada Data</p>";
    }
    else {
      $fp = file($namafile);

      foreach( $fp as $linenum => $linecontent ){
        $data = explode('~', $linecontent );
        echo '<h3>Data ke-' . $linenum . '</h3>';
        echo '<p>';
        echo 'Nama: ' . $data[0] . '<br>';
        echo 'Alamat: ' . $data[1] . '<br>';
        echo 'Kota: ' . $data[2] . '<br>';
        echo '</p>';
      }
    }
  ?>
</body>
</html>
```

- Simpan file tersebut.

Function yang digunakan pada latihan di atas:

1. **fopen**(namafile, mode), function untuk membuka sebuah file.

pilihan mode terdiri dari:

- 'r': file hanya dibaca, posisi pointer ada di awal file
 - 'r+': file dapat dibaca dan ditulis, posisi pointer ada di awal file
 - 'w': file hanya dapat ditulis, posisi pointer ada di awal file dan akan menghapus isi file. Jika file tidak ada, akan dibuat terlebih dahulu.
 - 'w+': file dapat dibaca dan ditulis, posisi pointer ada di awal file dan akan menghapus isi file. Jika file tidak ada, akan dibuat terlebih dahulu.
 - 'a': file hanya dapat ditulis, posisi pointer di akhir file. Jika file tidak ada, akan dibuat terlebih dahulu.
 - 'a+': file dapat ditulis dan dibaca, posisi pointer di akhir file. Jika file tidak ada, akan dibuat terlebih dahulu.
 - 'x': membuat dan membuka file untuk ditulis, posisi pointer ada di awal file. Jika file sudah ada, **fopen()** akan gagal dan menghasilkan **FALSE** dan error. Jika file tidak ada, akan dibuat terlebih dahulu.
 - 'c': membuka file untuk ditulis saja. Jika file tidak ada, akan dibuatkan terlebih dahulu. Posisi pointer ada di awal file.
 - 'c+': membuka file untuk ditulis dan dibaca, selebihnya memiliki perilaku seperti 'c'.
2. **fwrite**(resource, string), function untuk menulis ke dalam file.
 3. **fclose**(resource), function untuk menutup file.

Contoh:

```
<?php
$namafile = "data.txt";
$fp = fopen( $namafile, 'a' );
fwrite( $fp, "Konten ditambahkan ke file" );
fclose( $fp );
?>
```

4. **file_exists**(namafile), digunakan untuk mengecek apakah file tersebut exists(ada) atau tidak ada. Menghasilkan **true** jika file sudah ada, dan **false** jika file tidak (belum) ada.
5. **file**(namafile), digunakan membaca file dan setiap baris file tersebut akan disimpan ke dalam sebuah **array**.
6. **die**(string) atau sama dengan **exit**(string), mengeluarkan pesan dan menghentikan proses pada script PHP. *Script* yang berada setelahnya tidak akan pernah dieksekusi.

LATIHAN 2

Mengupload File ke Server

1. MEMBUAT FORM

- Buat sebuah file baru dengan nama **upload.php**. File ini digunakan untuk membaca sistem file dan mengambil file yang akan diupload.
- Ketikkan kode berikut ini:

```
<!doctype html>
<html>
  <head>
    <title>Upload File</title>
  </head>
  <body>
    <h2>Upload File</h2>
    <form method="post" action="prosesfile.php" enctype="multipart/form-data">
      File <input type="file" name="file"> <input type="submit" value="Upload">
    </form>
  </body>
</html>
```

- Simpan file tersebut

Keterangan:

Pada form, untuk mengirim sebuah file, atribut **enctype** harus menggunakan "**multipart/form-data**". Secara *default*, apabila atribut **enctype** tidak didefinisikan, maka form akan menggunakan **enctype="application/x-www-form-urlencoded"**.

Sebagai contoh:

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

```
<form method="post" action="proses.php" enctype="multipart/form-data">
```

Form pertama, akan menggunakan **enctype="application/x-www-form-urlencoded"**, sedangkan form kedua akan menggunakan "**multipart/form-data**", yang dapat digunakan untuk mengirim file ke server.

Informasi mengenai penggunaan enctype dapat anda lihat pada link berikut ini: <http://www.w3.org/TR/html4/interact/forms.html#h-17.13.4>.

Elemen HTML yang digunakan untuk menentukan file yang akan di-*upload* adalah elemen **<input type="file">**.

MEMBUAT FILE UNTUK MENGUPLOAD FILE

- a. Buat file baru dengan nama **prosesfile.php**. File ini akan menangani dan memproses file yang dikirim ke server. Melalui halaman **upload.php**.
- b. Ketikkan *script* di bawah ini:

```
<?php
if ( !$_FILES ){
    die( "Tidak Ada File yang Diupload" );
}

$folderupload = "upload";
/* cek apakah folder sudah ada
   jika belum ada, buat folder tersebut
   menggunakan function mkdir()
*/
if ( !is_dir( $folderupload ) ){
    mkdir( $folderupload );
}

$error = $_FILES['file']['error'];
if ( $error > 0 ){
    die( "Upload Error!" );
}

$namafile = $_FILES['file']['name'];
$typefile = $_FILES['file']['type'];
$sizefile = $_FILES['file']['size'];
$tmpfile = $_FILES['file']['tmp_name'];

if ( move_uploaded_file( $tmpfile, $folderupload."/".$namafile ) ){
    echo "<h2>Upload Berhasil</h2>";
    echo "<p>Nama File: $namafile</p>";
    echo "<p>Ukuran: {$sizefile} bytes</p>";
    echo "<p>Type File: ".$typefile."</p>";
}
?>
```

- c. Simpan file tersebut

Keterangan:

File yang dikirimkan melalui halaman **upload.php** akan ditangkap oleh variabel **\$_FILES** pada PHP yang bertipe **array**. Format dari array tersebut adalah sebagai berikut:

- **\$_FILES[namafield]['name']**, menampung informasi nama file yang dikirim.
- **\$_FILES[namafield]['type']**, menampung informasi type file yang dikirim.
- **\$_FILES[namafield]['error']**, menampung kode error, jika 0 berarti tidak ada error.
- **\$_FILES[namafield]['tmp_name']**, menampung informasi nama file *temporary* (sementara) dari file yang di-upload. Sebelum disimpan secara permanen, file

yang di-*upload* tersebut akan disimpan terlebih dahulu pada folder *temporary* (tmp). Selanjutnya, file sementara inilah yang dipindahkan dari lokasi sementara ke lokasi tetap yang ditentukan oleh programmer.

- **\$_FILES**[*namafile*]['*size*'], menampung informasi ukuran dari file yang dikirim atau di-*upload* ke server. Ukuran yang digunakan adalah bytes.
- *namafile* akan tergantung pada atribut **name** yang digunakan pada elemen input file. Sebagai contoh, jika elemen yang digunakan adalah:

`<input type="file" name="file">`, maka variabel **\$_FILES** akan menjadi:

```
$_FILES['file']['name']  
$_FILES['file']['type']  
$_FILES['file']['tmp_name']  
$_FILES['file']['error']  
$_FILES['file']['size']
```

Function yang digunakan;

- **is_dir**(*namadirektori*), digunakan untuk mengecek apakah direktori sudah ada atau belum.
- **mkdir**(*namadirektori*), digunakan untuk membuat direktori (folder) baru, relatif terhadap folder dari file yang sedang diakses.
- **move_uploaded_file**(*namafile*, *tujuan*), digunakan untuk memindahkan file yang sudah di-*upload*, biasanya dari folder sementara (*temporary*) ke folder dan file tujuan. Bernilai **true** jika proses pemindahan sukses, dan **false** jika gagal. Misal:

```
move_uploaded_file( 'xad.tmp', 'file/test.doc');
```

perintah di atas akan memindahkan file **xad.tmp** ke folder **file** dengan nama **test.doc**.

REFERENSI:

1. Dealing With Forms, <http://id.php.net/manual/en/tutorial.forms.php>
2. Predefined Variables, <http://id.php.net/manual/en/reserved.variables.php>
3. Filesystem Functions, <http://www.php.net/manual/en/ref.filesystem.php>
4. Arrays, <http://id.php.net/manual/en/language.types.array.php>