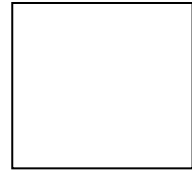


LATIHAN PRAKTIKUM PEMROGRAMAN WEB

BAB : FORM
NAMA : WAHYU SEPTIADI
NIM : 175150201111015
ASISTEN : 1. DESE NARFA FIRMANSYAH
2. IQBAL SANTOSO PUTRA
TGL PRAKTIKUM : 05 MARET 2019



LATIHAN 1

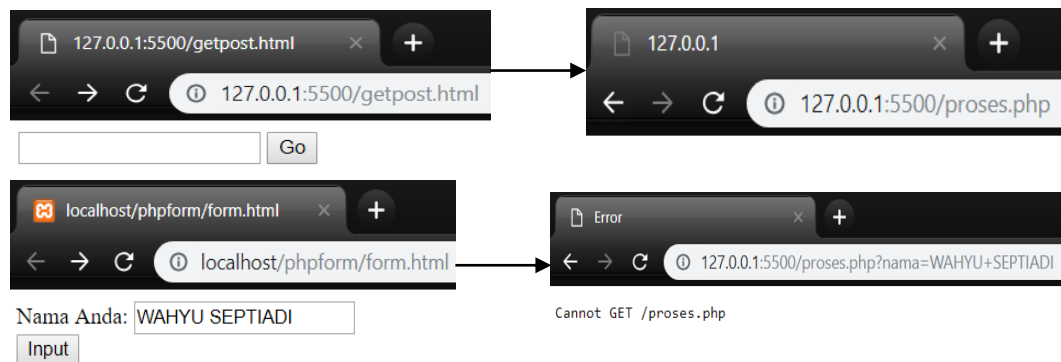
A. Soal

Tulis kode di bawah ini.

```
<body>
  <form action="proses.php" method="post">
    <input type="text" name="nama">
    <input type="submit" value="Go">
  </form>
</body>
```

Jalankan kode tersebut di browser lalu ganti metodenya dengan GET. Lihat perbedaannya pada URL di browser. Beri kesimpulan.

B. Screenshot



C. Syntax

```
1 <body>
2   <form action="proses.php" method="get">
3     <input type="text" name="nama">
4     <input type="submit" value="Go">
5   </form>
6 </body>
```

D. Penjelasan

Pada syntax diatas kita membuat form inputan, dimana inputan pertama yaitu bertipe text yang diperintahkan untuk mengisikan nama, kemudian pada inputan kedua yaitu bertipe submit bernilai/berlabel Go yaitu ketika kita klik tombol Go tersebut akan di alihkan ke action selanjutnya yaitu file proses.php dengan method post (output akan ditampilkan di bagian body), tetapi output disini masih error karena proses.php tidak ada. Perbedaan dengan method get yaitu ketika kita klik

tombol Go inputan yang kita berikan di inputan pertama nilainya akan dicetak pada URL sebagai query string seperti gambar kedua.

LATIHAN 2

A. Soal

Untuk beberapa kasus berikut, tentukan metode apakah yang cocok untuk digunakan dan beri alasan.

1. Form untuk login.
2. Form untuk mengirim gambar desain poster ke website percetakan.
3. Form untuk mencari suatu artikel di situs berita.
4. Form untuk search engine.

B. Screenshot

C. Syntax

1	
---	--

D. Penjelasan

1. Form untuk login, metode yang cocok adalah metode POST (mengirimkan username dan password) karena data dari form yang dikirimkan ke server melalui HTTP request header tidak mengubah URL/ tidak ditampilkan di URL sehingga bisa digunakan sebagai keamanan data yaitu tidak akan terbaca oleh pengguna yang tidak bersangkutan karena data yang dikirim di login ini mengandung informasi yang sensitif / privasi yaitu username dan password.
2. Form untuk mengirim gambar desain poster ke website percetakan, metode yang cocok adalah metode POST (mengirimkan gambar) karena data dari form yang dikirimkan ke server melalui HTTP request header tidak mengubah URL, selain itu untuk mengirim gambar harus menyediakan ID pada server, supaya mempermudah jika kita ambil nantinya. Jadi termasuk merubah data di server.
3. Form untuk mencari suatu artikel di situs berita, metode yang cocok adalah metode GET karena kita tidak merubah isi dari server untuk request data, maka semua pengguna dapat membaca data tersebut pada URL melalui Address bar di browser, biasanya dipisah dengan tanda tanya (?). Selain itu hasil URL metode ini bisa dibagi (share) data ke orang lain, karena tidak ada konten yang sensitif.
4. Form untuk search engine, metode yang cocok adalah metode GET karena data yang dikirimkan melalui URL untuk request data yang ingin dicari dan kita tidak merubah server hanya membaca/ menampilkan response dari server, Selain itu hasil URL metode ini bisa dibagi (share) data ke orang lain dan juga dapat di-bookmark di browser pengguna, karena tidak ada konten yang sensitif, akan tetapi kapasitas pada URL metode ini sangat terbatas.

Pada intinya, untuk penggunaan metode POST dan GET, kita harus mengajukan pertanyaan: Apakah akan terjadi perubahan pada server? Jika ya, maka kita gunakan POST, jika tidak maka kita gunakan GET. Singkatnya POST untuk Create, Update, Delete data, sedangkan GET untuk Read data. Pertanyaan selanjutnya yaitu Apakah ada data sensitif yang dikirim ? jika ya maka kita gunakan POST supaya data tidak terekspose di URL, begitu sebaliknya.

LATIHAN 3

A. Soal

Tulis kode di bawah ini.

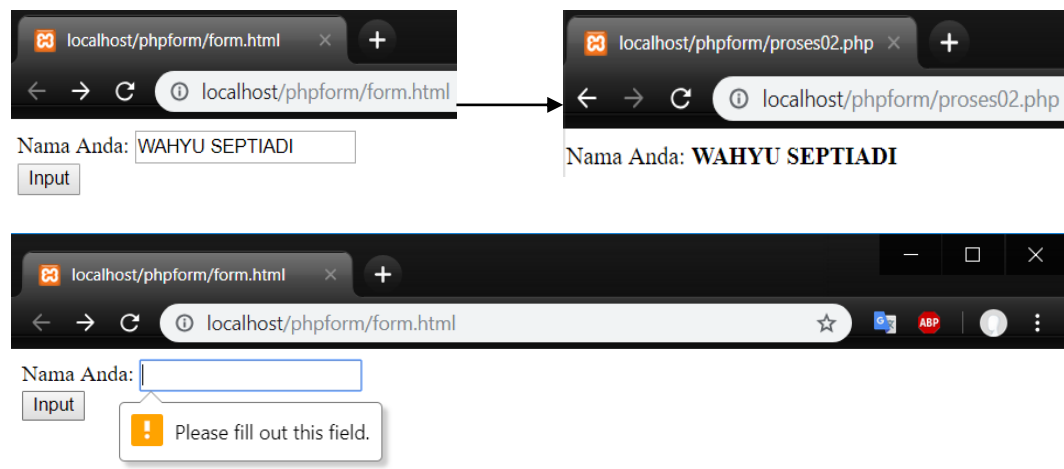
```
<form action="proses02.php" method="post" name="input">
    Nama Anda: <input type="text" name="nama" required>
    <br>
    <input type="submit" name="input" value="Input">
</form>
```

Kemudian tulis kode di bawah ini.

```
<?php
if (isset($_POST['input'])) {
    $nama = $_POST['nama'];
    echo "Nama Anda: <b>$nama</b>";
}
```

- Jalankan kode di atas kemudian isikan data input dan tekan tombol input. Apa yang terjadi?
- Ulangi dengan mengosongkan data input dan amati apa yang terjadi.
- Jelaskan alur pengiriman data dari form yang ada di kode tersebut!

B. Screenshot



C. Syntax

```
1 <form action="proses02.php" method="post" name="input">
2     Nama Anda: <input type="text" name="nama" required>
3     <br>
4     <input type="submit" name="input" value="Input">
5 </form>
```

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

D. Penjelasan

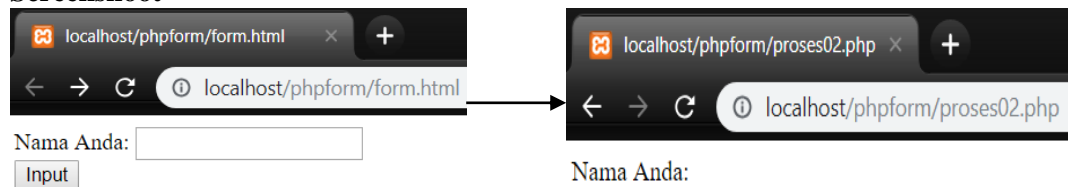
- Ketika menginputkan nilai pada inputan form diatas bernilai WAHYU SEPTIADI, dan menekan tombol input maka akan dialihkan pada action selanjutnya yaitu di file proses02.php yang akan mencetak di bagian body HTML seperti gambar pertama diatas.
- Ketika kita tidak menginputkan data pada form diatas dan langsung menekan tombol input akan memunculkan pop up seperti gambar kedua diatas.
- Kita dapat menyimpulkan bahwa syntax diatas menggunakan method POST, dimana data yang di kirimkan ke server melalui HTTP request header ditangkap oleh action selanjutnya yaitu proses02.php yang meminta untuk ngambil data yang ada di form di file form.html tadi dengan \$_POST yang dimasukkan di variabel nama dan mencetaknya seperti pada nomor 3 dan 4 file proses02.php. Untuk pop up tersebut dapat muncul karena adanya syntax required seperti nomor 2 pada file form.html.

LATIHAN 4

A. Soal

Ubah kembali kode pada Latihan 3 dengan menghapus tulisan required pada baris 2. Jalankan kode dan kosongi data nama kemudian tekan tombol Input. Apa yang terjadi? Jelaskan apa fungsi dari kode required.

B. Screenshoot



C. Syntax

```
1 <form action="proses02.php" method="post" name="input">
2     Nama Anda: <input type="text" name="nama">
3     <br>
4     <input type="submit" name="input" value="Input">
5 </form>
```

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

D. Penjelasan

Berbeda dengan latihan sebelumnya, ketika kita mengkosongkan data form input dan langsung menekan tombol input kita tidak menemui pop up saran untuk harus mengisi data, tetapi kita akan dialihkan ke action selanjutnya yaitu pada file proses02.php seperti gambar diatas. Fungsi kode required yaitu memberi perintah kepada user untuk harus mengisi data form tersebut, jika tidak akan muncul pop up seperti gambar pada latihan 3.

LATIHAN 5

A. Soal

Tulis kode berikut.

```
<form enctype="multipart/form-data" action="upload.php"
method="post">
    Choose a file to upload:
    <input name="uploadedfile" type="file" /> <br>
    <input type="submit" value="Upload File" />
</form>
```

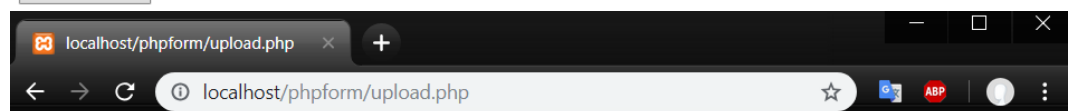
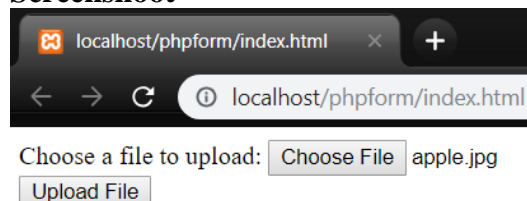
Tulis kode berikut.

```
<?php
$target_path = "uploads/";
$target_path = $target_path.
basename($_FILES['uploadedfile']['name']);
if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'],
$target_path)) {
    echo "The file ".basename(
$_FILES['uploadedfile']['name'])." has been uploaded";
} else {
    echo "There was an error uploading the file, please try
again!";
}
```

Di manakah posisi file setelah proses upload berhasil?

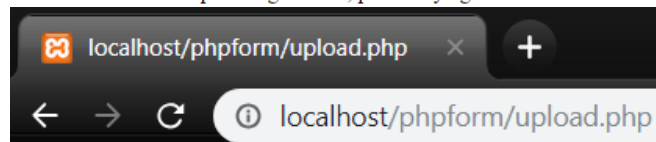
Apakah hasil output dari kode tersebut setelah dijalankan?

B. Screenshoot



Warning: move_uploaded_file(uploads/apple.jpg): failed to open stream: No such file or directory in C:\xampp\htdocs\phpform\upload.php on line 4

Warning: move_uploaded_file(): Unable to move 'C:\xampp\tmp\phpBBE2.tmp' to 'uploads/apple.jpg' in C:\xampp\htdocs\phpform\upload.php on line 4
There was an error uploading the file, please try again!



The file apple.jpg has been uploaded

Hasil dapat di keluarkan tanpa error setelah pembuatan folder uploads di didalam direktori file upload.php berada.

C. Syntax

```
1 <form enctype="multipart/form-data" action="upload.php"
2 method="post">
3     Choose a file to upload:
4     <input name="uploadedfile" type="file" /> <br>
5     <input type="submit" value="Upload File" />
6 </form>
```

```
1 <?php
2 $target_path = "uploads/";
3 $target_path = $target_path.
4 basename($_FILES['uploadedfile']['name']);
5 if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'],
6 $target_path)) {
7     echo "The file ".basename(
8 $_FILES['uploadedfile']['name'])." has been uploaded";
9 } else {
10     echo "There was an error uploading the file, please
11 try again!";
12 }
```

D. Penjelasan

Posisi file setelah proses upload berhasil seharusnya akan disimpan pada folder uploads seperti yang disebutkan pada nomer 2 file upload.php diatas, dimana variabel target_path diinisialisasi dengan uploads/ yang maksudnya adalah nama folder untuk menyimpan file yang terupload. Akan tetapi program saat dijalankan pertama akan terjadi ERROR, karena folder uploads belum dibuat sebelumnya. Hasil yang ditampilkan dari kode program diatas yaitu terjadi ERROR, karena folder uploads belum dibuat sebelumnya seperti gambar kedua. Setelah dibuatkan folder nya akan menampilkan keluaran seperti gambar ketiga diatas.