

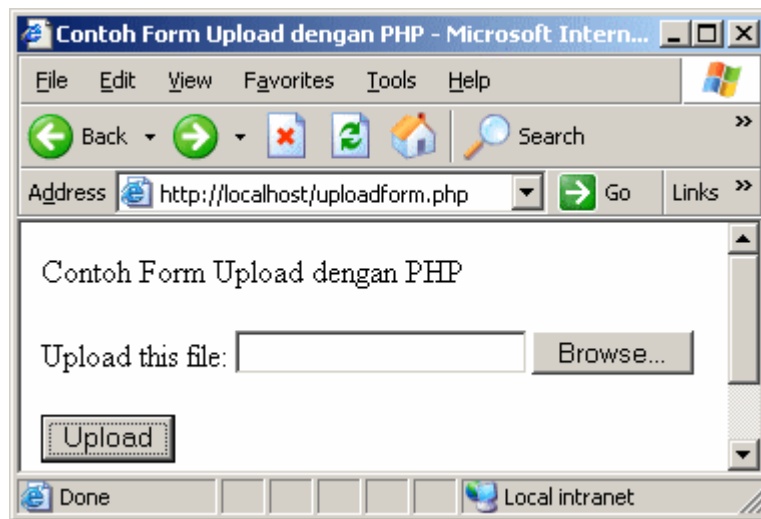
Cara Melakukan Upload File pada PHP Apache

Persiapan

Untuk melakukan ini kita harus sudah siap dengan beberapa infrastruktur seperti Apache web server dan juga PHP. Pada kasus ini penulis menggunakan sistem operasi Windows XP yang dilengkapi dengan Apache web server versi 1.3.33 (Win32) dan PHP dengan versi 4.3.11. Untuk mendapatkan Apache Anda bisa download dari www.apache.org dan PHP bisa Anda dapatkan di www.php.net. Di sini penulis tidak akan lagi menjelaskan proses instalasi keduanya. Penulis asumsikan semua sudah *diinstall* pada konfigurasi default dan folder web root terletak pada folder `c:/program files/apache group/apache/htdocs`.

Membuat form upload

Langkah pertama yang perlu dibuat adalah merancang form uploadnya. Screen shot dari form upload nantinya adalah sebagai berikut.



Adapun code HTML untuk membuat code di atas adalah sebagai berikut.

```
<html>
<head>
<title>Contoh Form Upload dengan PHP</title>
</head>
<body>
<p>Contoh Form Upload dengan PHP</p>
<form enctype="multipart/form-data" action="p_upload.php" method="post">
<input type="hidden" name="MAX_FILE_SIZE" value="2000000" />
Upload this file: <input name="userfile" type="file" /><p>
<input type="submit" value="Upload" /> </p>
</form>
<p>&nbsp;</p>
</body>
</html>
```

Simpan code tersebut dengan nama `uploadform.php` dan letakkan pada folder web root dari Apache. Panggil file ini dengan mengetikkan `http://localhost/uploadform.php`. Pastikan Anda sudah menyalakan service Apache dengan mengetikkan perintah `net start apache` pada command prompt Windows. Apabila semuanya sudah benar maka form upload akan muncul pada browser Anda.

Kita lihat pada script di atas encoding type-nya menggunakan `multipart/form-data` dan method-nya POST. Untuk action script-nya diarahkan pada file `p_upload.php`. Script `p_upload.php` inilah yang akan kita buat sebagai script yang berguna untuk memproses file yang diupload oleh user.

Membuat script yang memproses upload

Setelah form upload di atas jadi, maka saat ini kita harus membuat script yang fungsinya untuk menerima dan

memproses file yang diupload oleh user. Sesuai dengan script sebelumnya maka script pemroses upload ini bernama p_upload.php. Berikut ini adalah codenya.

```
<?php
$uploaddir = 'c:/files/';
$uploadfile = $uploaddir . $_FILES['userfile']['name'];

if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile))
{
echo "File sudah sukses diupload ke server pada folder dan nama file
$uploadfile";
}
else
{
echo "Proses upload gagal, kode error = " . $_FILES['userfile']['error'];
}
?>
```

Pada kasus ini penulis akan meletakkan file yang diupload pada folder c:\files\. Untuk proses upload ini PHP menyediakan autoglobal \$FILES yang ada sejak PHP 4.1.0. Melalui autoglobal \$_FILES ini Anda bisa mengetahui nama file yang diupload, type file, size, temporary file name dan juga kode error yang terjadi.

Intinya semua file yang diupload oleh user sebenarnya akan ditampung pada suatu folder temporary sistem dengan nama file yang juga temporary. Untuk folder temporary bisa diatur pada konfigurasi di file php.ini. Pada kondisi default Windows, temporary folder akan menggunakan folder yang sama dengan yang digunakan oleh Windows. Fungsi move_uploaded_file merupakan fungsi yang bertujuan untuk memindahkan file yang diupload dari folder temporary tadi menuju folder yang sudah kita tentukan. Pada kasus di atas semua file yang diupload akan dipindahkan ke dalam folder c:\files\. Fungsi ini memiliki dua parameter yaitu yang pertama adalah nama file yang ada di temporary folder dan parameter kedua adalah folder tujuan beserta juga nama filenya. Jika proses pemindahan file ini berjalan sukses maka fungsi ini akan mengembalikan nilai TRUE, demikian pula sebaliknya jika gagal maka akan mengembalikan nilai FALSE.

Berikut ini keterangan dari autoglobal \$_FILES.

`$_FILES['xxx']['name']`

Berguna untuk mengetahui nama file yang diupload.

`$_FILES['xxx']['type']`

Berguna untuk mengetahui tipe dari file, contohnya seperti text/xml.

`$_FILES['xxx']['size']`

Berguna untuk mengetahui ukuran file yang telah diupload.

`$_FILES['xxx']['tmp_name']`

Berguna untuk mengetahui nama sementara dari file yang diupload.

`$_FILES['xxx']['error']`

Berguna untuk mengetahui error code yang terjadi pada saat proses upload.

Jika semua proses upload berjalan lancar maka error code yang dihasilkan adalah 0. Tanda xxx di atas bisa disesuaikan dengan nama input untuk upload file seperti pada form upload di atas. Pada kasus ini kita menggunakan nama input `userfile`.