

MODUL PRAKTIKUM
PEMROGRAMAN WEB



Materi 7:
File Handling

Dosen:

Febrianta Surya Nugraha, M.Kom

S1 INFORMATIKA
STMIK AMIKOM SURAKARTA
2020

File Handling

PHP INCLUDE FILES

Pernyataan **include** (atau **require**) mengambil semua teks / kode / markup yang ada dalam file yang ditentukan dan menyalinnya ke file yang menggunakan pernyataan include.

Pernyataan **include** dan **require** identik, kecuali jika gagal:

- **require** akan menghasilkan kesalahan fatal (E_COMPILE_ERROR) dan menghentikan skrip.
- **include** hanya akan menghasilkan peringatan (E_WARNING) dan skrip akan berlanjut.

Syntax:

```
include 'filename';  
  
require 'filename';
```

*Gunakan **require** ketika file diperlukan oleh aplikasi.*

*Gunakan **include** ketika file tidak diperlukan dan aplikasi harus dilanjutkan ketika file tidak ditemukan.*

Latihan

Contoh kita akan membuat menu sederhana dengan nama menu.php

```
<?php
echo '<a href="index.php">Home</a> -
<a href="index.php">HTML Tutorial</a> -
<a href="index.php">CSS Tutorial</a> -
<a href="index.php">JavaScript Tutorial</a> -
<a href="index.php">PHP Tutorial</a>';
?>
```

Dan kita mendefinisikan variabel di halaman lain yaitu var.php

```
<?php
$nama='Paijo';
$nim='007';
?>
```

Kemudian kita buat halaman yang akan diakses.

```
<html>
<body>
<div class="menu">
<?php include 'menu.php';?>
</div>

<h1>Welcome to my home page!</h1>
<?php include 'var.php';
echo "Nama: $nama <br>";
echo "Nim: $nim <br>";
?>

</body>
</html>
```

PHP READFILE() FUNCTION

Fungsi readfile adalah fungsi untuk membaca file dan menuliskannya ke buffer output.

Contoh buatlah file .txt yang ditempatkan ke dalam server (1 patch dengan file php yang akan diakses).

Contoh: identitas.txt

```
Nama: Namaku
Nim: 08007
Jurusan: D3 Manajemen Informatika
Mata Kuliah : Pemrograman web lanjut
```

Kemudian kita akan buka file tersebut dengan menggunakan PHP:

```
fopen ($namafile, $mode);
```

\$namafile merupakan nama file yang akan dibuat / dibaca, sedangkan \$mode merupakan mode akses file. Mode akses file yang bisa digunakan yaitu :

Mode	Keterangan
r	Open a file for read only. Penunjuk file dimulai pada awal file
r+	Open a file for read/write. Penunjuk file dimulai pada awal file
w	Open a file for write only. Hapus konten file atau buat file baru jika tidak ada. Penunjuk file dimulai pada awal file
w+	Open a file for read/write. Hapus konten file atau buat file baru jika tidak ada. Penunjuk file dimulai pada awal file
a	Open a file for write only. Data yang ada dalam file dipertahankan. Penunjuk file dimulai di akhir file. Membuat file baru jika file tersebut tidak ada
a+	Open a file for read/write. Data yang ada dalam file dipertahankan. Penunjuk file dimulai di akhir file. Membuat file baru jika file tersebut tidak ada
x	Creates a new file for write only. Mengembalikan FALSE dan kesalahan jika file sudah ada
x+	Creates a new file for read/write. Mengembalikan FALSE dan kesalahan jika file sudah ada

fread()

Fungsi fread () membaca dari file yang terbuka.

Parameter pertama dari fread () berisi nama file yang akan dibaca dan parameter kedua menentukan jumlah byte maksimum untuk dibaca.

fclose()

Fungsi fclose () digunakan untuk menutup file yang terbuka.

Ini adalah praktik pemrograman yang baik untuk menutup semua file setelah Anda selesai menggunakannya. Anda tidak ingin file terbuka berjalan di server Anda mengambil sumber daya!

```
<?php
$myfile = fopen("identitas.txt", "r") or die("Unable to open file!");
echo fread($myfile, filesize("identitas.txt"));
fclose($myfile);
?>
```

or die merupakan menghasilkan pesan jika fungsi fopen () tidak dapat membuka file yang ditentukan.

Fungsi fungsi lain:

1. **fgets()** : fungsi digunakan untuk membaca satu baris dari suatu file. **Catatan:** Setelah panggilan ke fungsi fgets (), penunjuk file telah pindah ke baris berikutnya.
2. **feof()** : fungsi memeriksa apakah "end-of-file" (EOF) telah tercapai. Fungsi feof () berguna untuk perulangan melalui data yang panjangnya tidak diketahui.

```

<?php
$myfile = fopen("identitas.txt", "r") or die("Unable to open file!");
echo "Fungsi fgets <br>";
echo fgets($myfile);
fclose($myfile);
?>

<?php
echo "<br>" . "<br>" . "Fungsi feof <br>";
$myfile = fopen("identitas.txt", "r") or die("Unable to open file!");
// Output one line until end-of-file
while(!feof($myfile)) {
    echo fgets($myfile) . "<br>";
}
fclose($myfile);
?>

```

PHP FILE CREATE/WRITE

Jika Anda mengalami kesalahan saat mencoba menjalankan kode ini, periksa apakah Anda telah memberikan akses file PHP Anda untuk menulis informasi ke hard drive.

fwrite()

Fungsi yang digunakan untuk menulis kedalam file.

Parameter pertama dari fwrite () berisi nama file yang akan ditulis dan parameter kedua adalah string yang akan ditulis (r w a dst).

Contoh:

```

<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "Nama\n";
fwrite($myfile, $txt);
$txt = "Nim\n";
fwrite($myfile, $txt);
fclose($myfile);
?>

```

Buka folder yang berisi file php tersebut, maka akan ada file newfile.txt beserta isinya yang di inputkan apabila script berhasil dijalankan.

PHP Overwriting

Kita akan menghapus semua isi data dari file newfile.txt dan mengisinya dengan data baru.

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "Belajar \n";
fwrite($myfile, $txt);
$txt = "PWL\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

Apa yang menyebabkan script diatas meng-overwrite isi file pada latihan sebelumnya?

MEMBUAT DAN MENGHAPUS DIREKTORI

Membuat direktori baru

```
<?php
$dir = "folderbaru"; //nama direktori
$cek = mkdir ($dir);
if ($cek) {
echo "Direktori <b>$dir</b> berhasil dibuat";
} else {
echo "Direktori <b>$dir</b> gagal dibuat";
}
?>
```

Program di atas akan membuat direktori bernama “folderbaru” di direktori tempat program disimpan. Fungsi untuk membuat direktori baru dalam PHP adalah **mkdir()**. Jika direktori yang akan dibuat berada di luar tempat program disimpan, maka sertakan path direktori secara lengkap.

Contoh jika kita akan membuat subfolder di folderbaru yang telah dibuat sebelumnya bernama new, maka pada \$dir = "folderbaru/new"

Menghapus Direktori

```
<?php
//buat direktori
$dir = "folderbaru/new"; //nama direktori
$cek = mkdir ($dir);
if ($cek) {
echo "Direktori <b>$dir</b> berhasil dibuat";
} else {
echo "Direktori <b>$dir</b> gagal dibuat";
}
//hapus direktori
$del = rmdir ($dir);
if ($del) {
echo "<br>Direktori <b>$dir</b> berhasil dihapus";
} else {
echo "<br>Direktori <b>$dir</b> gagal dihapus";
}
?>
```

Fungsi rmdir() hanya bisa menghapus direktori kosong. Jika direktori ada isinya maka, gunakan fungsi rekursif untuk menghapus isi direktori. Lihat contoh di PHP Manual: <https://www.php.net/manual/en/function.rmdir.php> .

Menampilkan Isi Direktori

Misal kita akan menampilkan isi dari folder folderbaru yang kita buat sebelumnya. Karena isi folderbaru kosong, maka silahkan isi folder tersebut dengan file.

```
<?php
$dir = "folderbaru";
if ($handle = opendir($dir)) {
while (false !== ($file = readdir($handle))) {
if ($file != "." && $file != "..") {
echo "$file<br>";
}
}
closedir($handle);
}
?>
```

Fungsi **opendir()** pada baris ke-3 adalah untuk membuka direktori. Sedangkan fungsi **readdir()** akan membaca file dalam direktori satu per satu secara urut. Fungsi **closedir()** pada baris ke-9 adalah untuk menutup pembukaan direktori.

MANIPULASI FILE

Mendapatkan Informasi File

Beberapa Fungsi yang berhubungan dengan informasi dan manipulasi file, diantaranya adalah:

Fungsi	Keterangan
file_exists(\$file)	Memeriksa apakah \$file ada atau tidak
is_file(\$file)	Memeriksa apakah \$file adalah file atau bukan.
is_dir(\$file)	Memeriksa apakah \$file adalah direktori atau bukan.
is_executable(\$file)	Memeriksa apakah \$file termasuk file yang bisa dijalankan secara langsung atau bukan.
is_writable(\$file)	Memeriksa apakah \$file termasuk file yang bisa ditulis atau diedit.
is_readable(\$file)	Memeriksa apakah \$file termasuk file yang bisa dibaca atau tidak.
fileatime(\$file)	Menghasilkan waktu akses terakhir file (unix timestamp).
filectime(\$file)	Menghasilkan waktu pembuatan file (unix timestamp).
filemtime(\$file)	Menghasilkan waktu modifikasi terakhir file (dalam unix timestamp).
filesize(\$file)	Menghasilkan ukuran besar file

PHP Filesystem reference dapat dilihat lengkap di:

https://www.w3schools.com/php/php_ref_filesystem.asp

Mendapatkan Beberapa Informasi File

```
<?php
$file = "identitas.txt";
if (is_file($file)) {
echo "File <b>$file</b> adalah FILE<br>";
//
if (is_executable ($file)) {
echo "File <b>$file</b> bisa dijalankan secara langsung
(executable)<br>";
} else {
echo "File <b>$file</b> TIDAK executable<br>";
}
//
if (is_writable ($file)) {
echo "File <b>$file</b> bisa ditulis / diedit<br>";
} else {
echo "File <b>$file</b> TIDAK bisa ditulis /
diedit<br>";
}
//
if (is_readable ($file)) {
echo "File <b>$file</b> bisa dibaca<br>";
} else {
echo "File <b>$file</b> TIDAK bisa dibaca<br>";
}
echo "Akses terakhir file <b>$file</b> = ". date("d-m-Y
H:i:s.", filemtime($file)). "<br>";
echo "File <b>$file</b> dibuat = ". date("d-m-Y H:i:s.",
filectime($file)). "<br>";
echo "Modifikasi terakhir file <b>$file</b> = ". date("dm-
Y H:i:s.", filemtime($file)). "<br>";
echo "Ukuran file <b>$file</b> = ". filesize($file). "
bytes<br>";
echo "Jenis file <b>$file</b> = ". filetype($file). "<br>";
} else if (is_dir($file)){
echo "File <b>$file</b> adalah DIREKTORI<br>";
} else {
echo "File <b>$file</b> tidak dikenal<br>";
}
?>
```

Copy, Rename dan Delete File

Sebelumnya jika belum mempunyai file data.txt , dibuat terlebih dahulu.

Fungsi **copy()** akan menduplikasi file \$file menjadi file \$filebaru.

Fungsi **rename()** akan mengganti nama file \$filebaru menjadi \$filebaru2.

Fungsi **unlink()** akan menghapus file \$filebaru2.


```

<?php
$file = "data.txt";
$filebaru = "databaru.txt";
$filebaru2 = "databaru2.txt";
if (copy ($file, $filebaru)) {
echo "File <b>$file</b> berhasil dicopy menjadi
<b>$filebaru</b>. <br>";
}
if (rename ($filebaru, $filebaru2)) {
echo "File <b>$filebaru</b> berhasil direname menjadi
<b>$filebaru2</b>. <br>";
}
if (unlink ($filebaru2)) {
echo "File <b>$filebaru2</b> berhasil dihapus. <br>";
}
?>

```

PHP FILE UPLOAD

PHP, sebagai server-side-scripting, sangat memungkinkan untuk menangani upload file ke server. Ada beberapa hal yang perlu diperhatikan dalam upload file ini, yaitu :

1. Pada Form HTML harus ditambahkan atribut :

ENCTYPE="multipart/form-data"

Secara sederhana atribut enctype dimana merupakan atribut yang berperan menentukan bagaimana data form dikirim ke server. Atribut enctype itu sendiri hanya digunakan saat form menggunakan **method='post'**.

2. Form inputan upload file dapat menggunakan tag <input> dengan value atribut TYPE="FILE".
3. Untuk menangani inputan, PHP menyediakan sebuah variabel array global yaitu \$_FILES. Index dari variabel ini antara lain :
 - \$_FILES['file']['name'] : Nama asli dari file yang diupload
 - \$_FILES['file']['tmp_name'] : Nama temporary file yang diupload
 - \$_FILES['file']['size'] : Ukuran file asli (dalam byte)
 - \$_FILES['file']['type'] : MIME type file yang diupload

Multipurpose Internet Mail Extensions. Ini adalah cara mengidentifikasi file di Internet sesuai dengan sifat dan formatnya. Misalnya, menggunakan nilai header "Tipe konten" yang ditentukan dalam respons HTTP, browser dapat membuka file dengan ekstensi / plugin yang tepat.

4. Destination folder file upload harus bisa writable (bisa diakses), biasanya dengan permission 777 atau 775 untuk linux dan unix.

Nama file: index.php

Deskripsi: Program menampilkan form upload file.

```
<html>
<head><title>Upload File</title></head>
<body>
<FORM ACTION="upload.php" METHOD="POST" ENCTYPE="multipart/form-data">
Upload File : <input type="file" name="file"><br>
<input type="submit" name="Upload" value="Upload">
</FORM>
</body>
</html>
```

Nama file: upload.php

Deskripsi: Program proses upload file.

```
<?php
if (isset($_POST['Upload'])) {
$dir_upload = "images/";
$name_file = $_FILES['file']['name'];
//
if (is_uploaded_file($_FILES['file']['tmp_name'])) {
$cek = move_uploaded_file($_FILES['file']['tmp_name'],
$dir_upload.$name_file);
if ($cek) {
die ("File berhasil diupload");
} else {
die ("File gagal diupload");
}
}
}
?>
```

Fungsi `is_uploaded_file()` akan mengupload file yang telah dipilih melalui form pada program `index.php` ke folder temporary (sementara).

File yang sudah diupload ke folder temporary akan dipindahkan (move) ke direktori yang diinginkan menggunakan fungsi `move_uploaded_file()`.

Pastikan kedua file tersebut berada didalam 1 folder, dan buat folder `images` didalam folder yang sama dengan kedua file tersebut. Untuk membuka program akses menggunakan halaman `index.php`