

File ref >>

fopen (file, mode)

Membuka file baik pada lokal maupun remote (URL), variabel penyimpanannya disebut handle

```
$fh = fopen('files/teks.txt', 'r'); // $fh = handle
```

Untuk mode, gunakan r untuk membaca file, w untuk menulis file, ketika menggunakan w, isi file akan terhapus.

PENTING: DIRECTORY_SEPARATOR

Ketika menulis path untuk lokasi file, selalu gunakan slash (/) atau constant DIRECTORY_SEPARATOR, keduanya akan kompatibel pada sistem operasi baik windows, mac os, maupun linux, jangan gunakan back slash (\), karena tidak bisa digunakan lintas sistem operasi.

```
$file = 'files/teks.txt'; // cara pertama
$file = 'files'. DIRECTORY_SEPARATOR
        . 'teks.txt';
```

fclose (handle)

Menutup file yang dibuka oleh fopen. Perintah ini wajib dijalankan agar file tidak terkunci

```
$fh = fopen($file, 'r');
fclose($fh);
```

fread (handle, length) | file_get_contents

fread untuk membaca isi file

```
$file = 'files/teks.txt';
$fh = fopen($file, 'r');
$content = fread($fh, filesize($file));
fclose($fh);
```

file_get_contents akan sekaligus: membuka, membaca, dan menutup file, sehingga lebih praktis

```
$content = file_get_contents($file);
```

fwrite (handle, string)

fwrite digunakan untuk menulis string ke file, mirip file_put_contents

```
$file = 'files/teks.txt';
$fh = fopen($file, 'w');
$content = fwrite($fh, 'new phrase');
fclose($fh);
```

file_put_contents (file, data)

file_put_contents akan menjalankan fopen, fwrite, dan fclose sekaligus. Terlebih dahulu akan menghapus isi file.

```
$file = 'files/teks.txt';
```

```
file_put_contents($file, 'new phrase');
```

untuk mengunci file selama proses penulisan, sehingga tidak dapat ditulis oleh user lain, gunakan opsi LOCK_EX

```
file_put_contents($file, 'new phrase', LOCK_EX);
```

untuk menggabung isi file lama dengan teks yang baru, gunakan opsi FILE_APPEND

```
$file = 'files/teks.txt';
$new = 'New Contents';
file_put_contents($file, $new, FILE_APPEND | LOCK_EX);
```

cara lain:

```
$file = 'files/teks.txt';
$current = file_get_contents($file);
$current .= 'New Contents';
file_put_contents($file, $current);
```

filesize (file)

Menghasilkan ukuran file dalam byte.

```
filesize('files/teks.txt');
```

rename (old_name, new_name)

Mengubah nama file atau direktori

```
$old = 'files/teks.txt';
$new = 'files/new_teks.txt';
rename($old, $new);
```

rename juga dapat digunakan untuk memindah file (file lama akan terhapus)

```
$old = 'files/teks.txt';
$new = 'files/map/new_teks.txt';
rename($old, $new);
```

WARNING: Jika file baru telah ada, maka akan ditimpa

copy (source, destination)

Mengcopy file

```
$old = 'files/teks.txt';
$new = 'files/new_teks.txt';
copy($old, $new);
```

unlink (file)

Menghapus file

```
unlink('files/teks.txt');
```

basename (path)

Menghasilkan nama file

```
echo basename(__FILE__);
// file_folder.php
```

__FILE__

Menghasilkan full path file php yang sedang dijalankan

```
echo __FILE__;  
// E:\xampp\htdocs\cheat_sheet\file_folder.php
```

file_exists (file)

Tes apakah file atau direktori ada

```
if (file_exists('files/teks.txt')) { ... }
```

is_file (file)

Tes apakah nama file merupakan sebuah file

```
if (is_file('files/teks.txt')) { ... }
```

include | include_once

Include digunakan untuk memasukkan file, jika tidak ditemukan, hanya muncul warning, script tetap berjalan. include_once memasukan file hanya sekali.

```
include ('header.php'); // Bentuk fungsi  
include_once 'header.html'; // Bentuk statemen
```

Include dan include_once digunakan untuk menyertakan file umum seperti template, teks, dll

require | require_once

Require berarti perlu, karena perlu, maka jika file tidak ditemukan, script akan error dan berhenti

```
require ('inc/functions.php');  
require_once 'inc/functions.php';
```

require dan require_once digunakan untuk menyertakan file penting seperti file konfigurasi, daftar fungsi, dll

xxx_once

Pada include_once dan require_once, akan dilakukan pengecekan apakah file sebelumnya telah disertakan, jika belum maka file akan disertakan

Mode untuk fopen()

'r'	File dibuka untuk dibaca, cursor berada di awal konten.
'r+'	File dibuka untuk <u>dibaca dan ditulis</u> . Cursor berada di awal konten.
'w'	File dibuka untuk ditulis. Isi file akan dihapus. Jika file tidak ditemukan, buat file baru.
'w+'	File dibuka untuk <u>dibaca dan ditulis</u> . Isi file akan dihapus. Jika file tidak ditemukan, buat file baru.
'a'	Buka file untuk ditulis. Cursor diletakkan di akhir konten. Jika file tidak ditemukan, buat file baru.
'a+'	Buka file untuk <u>ditulis dan dibaca</u> . Cursor diletakkan di akhir konten. Jika file tidak ditemukan, buat file baru.

Folder (direktori) [ref >>](#)

mkdir (path)

Membuat direktori

```
mkdir ('files/backup');
```

pada contoh diatas, fungsi mkdir akan membuat folder backup di dalam folder files, untuk itu, folder files harus ada, jika tidak, maka akan error

rmdir (path)

Menghapus direktori

```
rmdir ('files/backup');
```

WARNING: direktori tidak dapat dihapus jika ada isinya, lihat script pada bagian EXAMPLE

is_dir (path)

Tes apakah path berupa direktori

```
if (is_dir('files/backup')) { ... }
```

rename (old_name, new_name)

Mengubah nama folder, lihat fungsi rename() pada file

scandir (path)

Scan isi direktori, hasil berbentuk array

```
$dir_content = scandir('folder');  
print_r($dir_content);  
// Array ( [0] => . [1] => .. [2] => file1.php  
           [3] => subfolder )
```

Keterangan: meskipun direktori kosong, variabel array tetap memiliki nilai . dan ..

file_exists (dir)

Tes apakah direktori ada, fungsi ini sama seperti yang digunakan untuk menguji file

dirname (dir)

Menghasilkan parent directory (direktori file)

```
echo dirname(__FILE__);  
// E:\xampp\htdocs\cheat_sheet\
```

__DIR__

Menghasilkan full path direktori dari file

```
echo __DIR__;  
// E:\xampp\htdocs\cheat_sheet\
```

EXAMPLE

Script yang kami buat ini akan menghapus folder beserta semua file dan subdirectory yang ada di dalamnya:

```
<?php
$del = array();
function deleteAll($dir)
{
    global $del;
    if (is_dir($dir) && file_exists($dir))
    {
        $dir_content = @scandir($dir);
        if (count($dir_content) > 2)
        {
            foreach ($dir_content as $isi) {
                if ($isi != "." && $isi != "..") {
                    if(@filetype($dir.'/'.$isi)=="dir"){
                        deleteAll($dir.'/'.$isi);
                    } else {
                        if (@unlink ($dir.'/'.$isi)) {
                            $del['files'][]=$dir.'/'.$isi;
                        }
                    }
                }
            }
            deleteAll($dir);
        } else {
            @rmdir($dir);
            $del['dir'][] = $dir;
        }
    }
}

deleteAll('folder');
echo '<pre>'; print_r($del);
```