

BAB 8

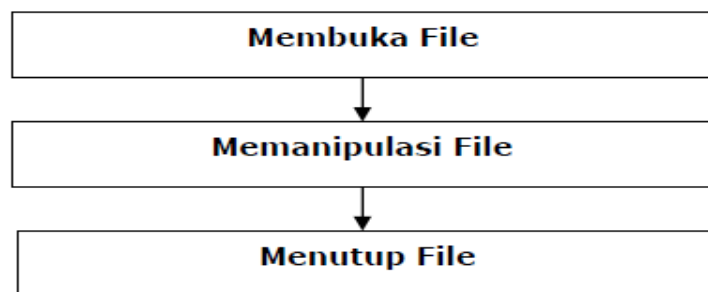
FILE

Pada BAB kali ini, kita akan membahas mengenai operasi file dengan PHP. Secara umum, operasi file pada php meliputi beberapa hal, yaitu membuka file, menutup file, membaca file, mengisi atau menulis file, menghapus file, menyimpan data dalam sebuah file, dan menampilkan isi file.

Prinsip Pemrosesan File

Pemrosesan file pada dasarnya meliputi 3 tahapan yaitu,

1. Proses membuka file
2. Proses memanipulasi file (menyimpan data ke file atau membaca data yang ada didalam file).
3. Proses menutup file.



A. MEMBUKA FILE

Perintah yang digunakan untuk operasi ini adalah *fopen* , sintaknya adalah sebagai berikut,

```
Fopen ("nama_file", 'mode_akses_file')
```

Secara umum, mode akses file terdapat 3 cara, yaitu

- Write ('w') adalah mode akses file yang hanya dapat digunakan untuk menulis atau memberikan isi baru ke dalam file dan menghapus isi file yang lama.

Contoh perintah;

```
<?
$nama_file = "tulis.txt";
$coba = fopen($nama_file, 'w') or die("File tidak bisa
dibuka");
fclose($coba);
?>
```

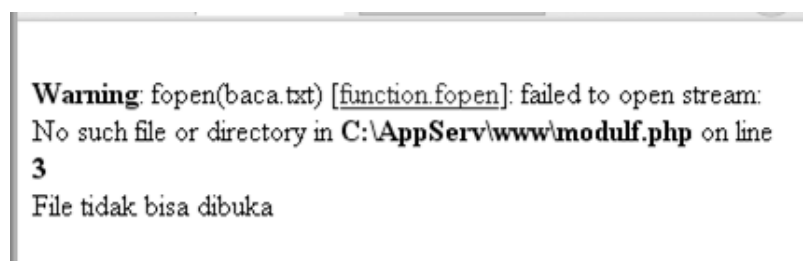
Cara menjalankan perintah tersebut (**membuka file**) adalah pada saat kita menjalankan atau memanggil perintah tersebut pada web browser. Pada mode ' w ', meskipun file "tulisan.txt" belum ada, secara otomatis setelah menjalankan perintah tersebut maka file "tulisan.txt" akan terbentuk.

- Read ('r') adalah mode akses file yang hanya dapat digunakan untuk membaca isi file. Dalam penggunaan mode ini, diharuskan file yang akan dibaca sudah ada atau sudah dibuat, jika file yang dibaca belum ada, maka akan muncul pesan error saat pembukaan file.

Contoh perintah;

```
<?
$nama_file = "baca.txt";
$coba = fopen($nama_file, 'r') or die("File tidak bisa
dibuka");
fclose($coba);
?>
```

Seperti pada mode write dan mode-mode lainnya, perintah dengan mode read dijalankan dengan menjalankan perintah tersebut pada web browser. *Jika file "baca.txt" belum ada, maka akan muncul pesan error (yang berarti tidak ada file atau directory baca.txt didalam C:\AppServ\www\modulf.php) dan akan menampilkan pesan bahwa " File tidak bisa dibuka " (or die ("File tidak bisa dibuka")).*



- Append ('a') adalah mode akses file yang dapat digunakan untuk menambah isi file atau dengan kata lain, memberikan isi baru kedalam file tanpa menghapus isi file lama.

Contoh perintah;

```
<?
$nama_file = "tambah.txt";
$coba = fopen($nama_file, 'a') or die("File tidak bisa
dibuka");
fclose($coba);
?>
```

modulf.php

```
1 <?
2 $nama_file = "tambah.txt";
3 $coba = fopen($nama_file, 'a') or die("File tidak bisa dibuka");
4 fclose($coba);
5 ?>
6
```

Cara kerja untuk menjalankan perintah tersebut sama dengan mode write.

Selain menggunakan ketiga mode diatas dapat juga menggunakan mode – mode dibawah ini,

| Mode akses file | keterangan |
|-----------------|---|
| r+ | Mode akses file ini digunakan membuka file untuk dibaca dan ditulis, tapi penulisannya diletakkan pada awal file |
| w+ | Mode akses file ini digunakan untuk membaca dan menulis file, seperti kegunaan pada mode ' w ', pada mode ini, isi file yang lama akan dihapus dan diganti dengan yang baru, dan bila belum ada isi pada file maka akan meng-create isi file. |
| a+ | Mode ini digunakan untuk membaca dan menambahkan isi file, pada saat penambahan, file akan ditambahkan sesuai letak pointer yaitu berada diakhir isi file. Seperti perintah w, w+, dan a, jika belum terdapat isi file, maka akan meng-create isi file. |

Untuk perintah or die (“File tidak bias dibuka”) dijalankan pada saat nama file yang dibuka tidak tepat atau salah. Dan biasanya, File_name (file yang akan dibuka) dan perintah fopen disimpan dalam sebuah variable, karena hasil dari perintah tersebut akan digunakan untuk parameter perintah-perintah akses file berikutnya.

B. MEMANIPULASI FILE

Untuk memanipulasi file, kita kelompokkan menjadi dua cara,

- Menambah isi file (menyimpan isi file)
- Membaca isi file

1. Menambah atau Menyimpan data pada File

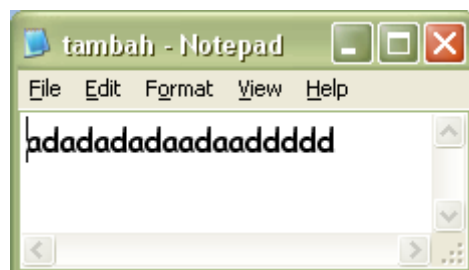
Untuk menyimpan data ke file, kita dapat menggunakan perintah `fputs()`, sintaknya

```
Fputs(variable_penyimpanan_buka_file, "data yang akan disimpan");
```

Contoh perintah,

```
<?
$nama_file = "tambah.txt";
$coba = fopen($nama_file, 'a') or die("File tidak bisa
dibuka");
fputs($coba, "adadadadaadaaddddd");
fclose($coba);
?>
```

Setelah perintah dijalankan pada browser, maka file `tambah.txt` akan berisi =
" adadadadaadaaddddd"



Selain menggunakan perintah `fputs()`, bisa juga menggunakan perintah `fwrite()`, sintaksnya

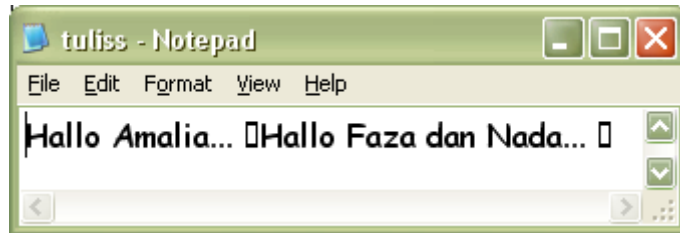
```
Fwrite(variable_penyimpanan_buka_file, "data yang akan disimpan");
```

Contoh perintah;

```
<?
$coba = "tuliss.txt";
$coba1 = fopen($coba, 'w') or die("File gagal dibuka");
$Data = "Hallo Amalia... \n";
fwrite($coba1, $Data);
$Data = "Hallo Faza dan Nada... \n";
fwrite($coba1, $Data);
fclose($coba1);
?>
```

Pada perintah diatas, menggunakan mode akses file “w” (write) jadi pada saat menulis file, isi dari file lama akan dihapus dan diganti isi file yang baru (overwriting). Jika kita menggunakan append (“a”), maka file lama akan ditambah file yang baru (data tambahan pada akhir pointer), begitu juga yang lain seperti penjelasan mode akses pada pembahasan buka file.

Hasilnya adalah,

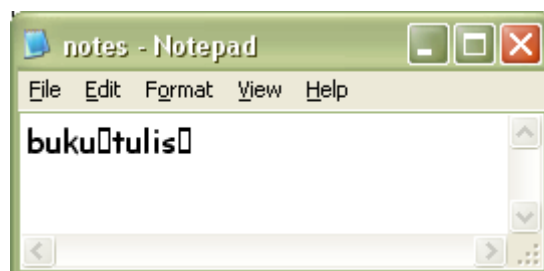


Bagaimana dengan menambah isi file ?

Untuk menambah isi file, kita gunakan append (“a”), lihat contoh perintah dibawah ini,

```
<?
$file = "notes.txt";
$opi = fopen($file, 'a') or die("File tidak bisa
dibuka");
$po1 = "buku\n";
fwrite($opi, $po1);
$po2 = "tuliss\n";
fwrite($opi, $po2);
fclose($opi);
?>
```

Maka hasilnya



Mode akses file lainnya yang dapat digunakan yaitu a+.

2. Membaca Isi File

Untuk dapat membaca isi suatu file, kita dapat menggunakan fread() dan fgetc(), perbedaan dari penggunaan perintah tersebut adalah,

- Fread() dapat melakukan pembacaan dari data awal suatu file hingga akhir.
- Fgetc() dapat melakukan pembacaan karakter demi karakter dalam suatu file.

Untuk membaca isi file, mode akses yang digunakan adalah read ('r').

a. Fread()

Fread() digunakan untuk membaca data dari awal hingga akhir isi file, sintaknya,

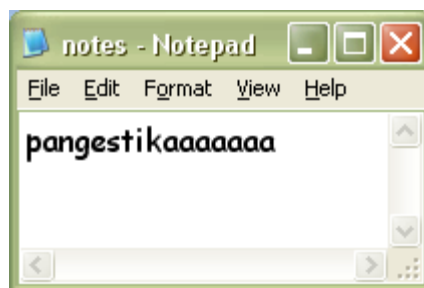
Fread(variable_penyimpanan_buka_file, panjang karakter yang akan dibaca);

Contoh perintah;

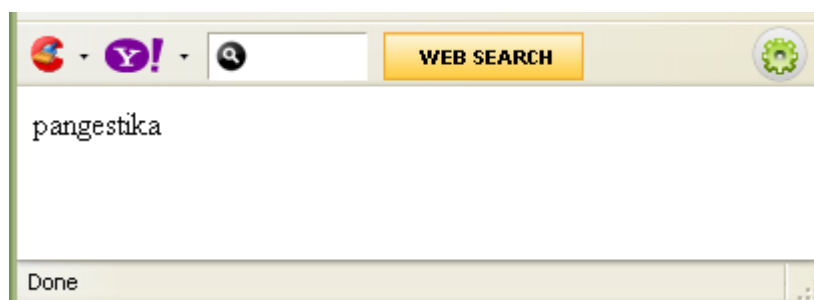
```
<?
$File = "notes.txt";
$po = fopen($File, 'r');
$Data = fread($po, 10);
fclose($po);
echo $Data;
?>
```

Perintah diatas digunakan untuk membaca file notes.txt, kemudian menampilkannya pada browser dengan panjang karakter yang akan dibaca 5.

Isi notes.txt



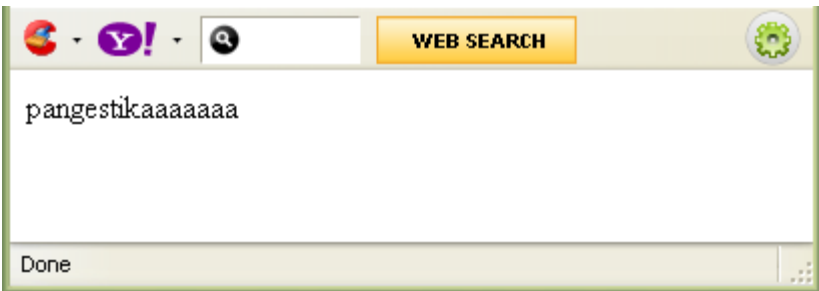
Tampilan perintah diatas pada browser,



Jika ingin menampilkan semua tulisan yang ada pada file notes.txt, menggunakan perintah fread(), yaitu dengan cara

```
<?
$File = "notes.txt";
$po = fopen($File, 'r');
$Data = fread($po,filesize($File));
fclose($po);
echo $Data;
?>
```

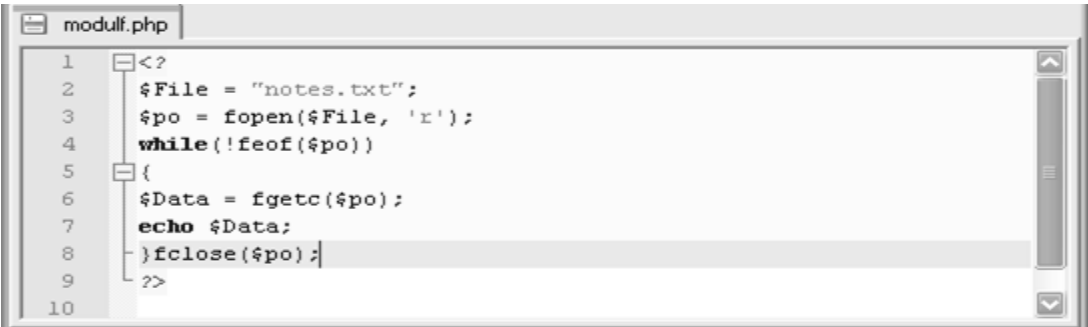
Maka hasil pada browser



b. Fgetc()

Fgetc() digunakan untuk membaca file karakter demi karakter, perhatikan dibawah ini untuk contoh penggunaanya.

```
<?
$File = "notes.txt";
$po = fopen($File, 'r');
while(!feof($po))
{
$Data = fgetc($po);
echo $Data;
}fclose($po);
?>
```



fgetc() digunakan pada perulangan, feof() digunakan untuk menyelidiki apakah pointer sudah berada di akhir dari file (end of file) atau belum. Selama belum end of file, maka proses pembacaan karakter akan berjalan terus. Setiap kali pembacaan, karakter yang dibaca akan ditampilkan di browser.

Tampilan pada browser



C. MENUTUP FILE

Jika ada file yang dibuka, maka file tersebut harus ditutup setelah pemrosesan atau kegiatan manipulasi file selesai. Jika file tidak ditutup, ada kemungkinan terjadi kerusakan pada struktur file. Setelah file ditutup, file tidak dapat dimanipulasi lagi isinya, jika ingin memanipulasi, file harus dibuka kembali. Perintah yang digunakan untuk menutup file yaitu, fclose();

Sintaknya yaitu,

```
Fclose(variable_penyimpanan_buka_file);
```

Contoh perintah

```
<?
$Nama= "notess.txt";
$File = fopen($Nama, 'w') or die("File tidak bisa dibuka");
fclose($File);
?>
```

D. MENGHAPUS FILE

Dalam PHP, function untuk menghapus file adalah unlink(). Sintaknya

```
Unlink(variable_penyimpanan_buka_file);
```

Argument ini berfungsi agar data tidak terhapus, nilai baliknya berupa

- TRUE jika berkas berhasil dihapus
- FALSE jika terjadi kegagalan dan menghapus.

Contoh perintahnya;

```
<?
$myFile = "notes.txt";
unlink($myFile);
?>
```

Jika tidak ditemukan file notes.txt maka argument menjadi FALSE dan pada browser akan muncul



E. UPLOAD FILE

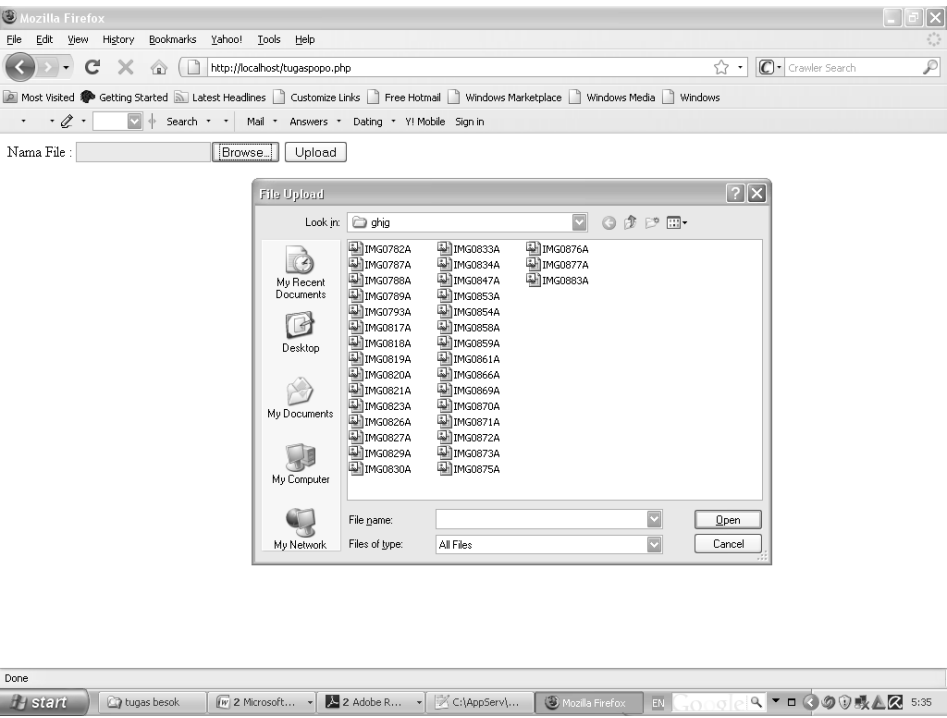
Untuk proses upload file ke server kita gunakan form untuk upload file, contoh sriptnya adalah,

```
<form enctype="multipart/form-data" action="upload.php"
method="post">
<input type="hidden" name="MAX_FILE_SIZE" value="40000" />
Nama File : <input name="userfile" type="file" />
<input type="submit" value="Upload" />
</form>
```

Untuk script diatas digunakan untuk input file dengan tujuan upload.php dengan method post, dimana file yang diambil dibatasi hingga 40kb.

Tampilan pada browser





Selanjutnya kita membuat file upload.php untuk proses uploadnya pada upload.php

```
<?php
$uploadaddir = 'uploads/';
$uploadfile = $uploadaddir . $_FILES['userfile']['name'];
if (move_uploaded_file($_FILES['userfile']['tmp_name'],
$uploadfile)) {
echo "File telah berhasil diupload";
} else {
echo "File gagal diupload";
}
?>
```

Maka bila mengambil file kurang dari 40 Kb maka file sukses diupload



A. FEOF();

Tests for end-of-file on a file pointer. Pengujian End-of-File pada file yang ditunjuk. Sintak : `int feof(FILE *stream);`. Untuk mengetest apakah posisi pointer sudah di end-of-file. Mengembalikan 0 jika belum end-of-file. Pointer file harus masih berlaku, dan harus menunjuk ke sebuah file berhasil dibuka oleh [fopen \(\)](#) (dan belum ditutup oleh [fclose \(\)](#)). Jika pointer file berlalu tidak valid Anda mungkin mendapatkan infinite loop, karena **feof ()** gagal untuk mengembalikan **TRUE**.

Latihan

1. Buatlah file untuk counter halaman yang diakses
2. Upload gambar pada foem dan tampilkan pada halalan yang lain