

# BAB X

## OPERASI FILE

Pada bab ini akan dipaparkan bagaimana menggunakan perintah PHP untuk melakukan operasi file mulai dari proses membuka dan menutup file. Setelah itu dilanjutkan dengan proses membaca, menulis, menambah isi, menghapus dan meng-upload file.

### Membuka File

Secara umum terdapat 3 cara membuka file, yaitu membuka file hanya untuk dibaca (read: 'r'), hanya untuk ditulisi baru (write: 'w'), dan hanya untuk ditambahi isinya (append: 'a'). Selain 3 cara membuka file tersebut, terdapat pula cara lain membuka file yaitu dapat dibaca dan ditulisi (read/write: 'r+'), serta dapat dibaca dan ditambahi isinya (append: 'a+').

Berikut ini adalah contoh kode PHP untuk membuka file

```
<?
$nama_file = "test.txt";
$fh = fopen($nama_file, 'X') or die("File tidak bisa dibuka");
fclose($fh);
?>
```

dengan 'X' dapat diganti dengan 'w', 'r', 'a', 'r+', 'a+'.

Apabila nama file yang akan dibuka ternyata salah, atau letak filenya yang tidak tepat, maka function `die()` yang akan dijalankan. Function `die()` akan menampilkan teks sebagai peringatan apabila proses membuka file gagal.

### Menutup File

Setelah file dibuka, hendaknya file tersebut juga ditutup ketika pemrosesan selesai. File yang tidak ditutup kemungkinan dapat terjadi kerusakan pada strukturnya. Berikut ini adalah contoh kode PHP untuk menutup file yang telah dibuka

```
$NamaFile = "testFile.txt";
$FileHandle = fopen($NamaFile, 'w') or die("File tidak bisa dibuka");
fclose($FileHandle);
```

File yang telah ditutup tidak bisa untuk dibaca, ditulisi, dan ditambah. Untuk bisa melakukan hal itu kembali, file terlebih dahulu harus dibuka lagi seperti sebelumnya.

### Menulis ke File

File yang telah dibuka dapat ditulisi dengan data di dalamnya. Berikut ini adalah contoh kode PHP untuk menulis suatu string ke dalam file.

```
<?
$FileKu = "testFile.txt";
```

```
$FileHandle = fopen($Fileku, 'w') or die("File gagal dibuka");
$DataString = "Hallo Amalia... \n";
fwrite($FileHandle, $DataString);
$DataString = "Hallo Faza dan Nada... \n";
fwrite($FileHandle, $DataString);
fclose($FileHandle);
?>
```

Apabila Anda menggunakan mode 'w' pada `fopen()`, dan selanjutnya Anda menuliskan data pada file, maka isi file yang lama (jika sebelumnya terdapat isi pada file tersebut) akan terhapus dan diganti dengan isi yang baru. Sedangkan apabila Anda menginginkan data yang lama pada suatu file tidak dihapus, maka Anda gunakan mode append 'a+' atau 'a'.

## Membaca Isi File

Misalkan kita memiliki file dengan nama `test.txt` yang isinya adalah sbb:

```
Selamat berjumpa lagi kawan!!
Senang bertemu Anda.
```

Kita dapat menggunakan kode PHP untuk membaca file tersebut dan selanjutnya hasil pembacaan dapat ditampilkan di browser atau diproses lebih lanjut. Berikut ini adalah contoh kode pembacaannya.

```
<?
$FileKu = "test.txt";
$FileHandle = fopen($FileKu, 'r');
$Data = fread($FileHandle, 5);
fclose($FileHandle);
echo $Data;
?>
```

Apabila kode di atas dijalankan, maka pada browser hanya akan menampilkan teks

```
Selam
```

Hal ini dikarenakan adanya nilai 5 pada `fread()`. Nilai tersebut menyatakan jumlah karakter awal yang dibaca dari file. Supaya seluruh isi file dapat dibaca, gunakan function `filesize()`.

```
<?
$FileKu = "test.txt";
$FileHandle = fopen($FileKu, 'r');
$Data = fread($FileHandle, filesize($FileKu));
fclose($FileHandle);
echo $Data;
?>
```

Setelah Anda lihat di browser, ternyata ganti baris pada isi file `test.txt` diabaikan. Meskipun teks dalam file tersebut ditulis dalam 2 baris, oleh perintah `fread()` akan dibaca dalam satu baris. Hal ini dikarenakan dalam file `test.txt` tidak terdapat tag html `<br>` untuk pindah baris.

Anda juga dapat menggunakan function `fgetc()` untuk membaca file. Function ini membaca isi file karakter demi karakter. Berikut ini adalah contoh penggunaannya.

```
<?
$FileKu = "test.txt";
```

```
$FileHandle = fopen($FileKu, 'r');
while(!feof($FileHandle))
{
    $Data = fgetc($FileHandle);
    echo $Data;
}
fclose($FileHandle);
?>
```

Keterangan:

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

## Menghapus File

Dalam PHP, function untuk menghapus file adalah `unlink()`. Berikut ini contohnya.

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

## Menambah Isi File

Maksud dari menambah isi file di sini adalah, menambah data baru pada file (diasumsikan data sudah ada sebelumnya). Untuk menambah isi file dalam PHP, mode pembukaan file nya menggunakan 'a' atau 'a+' dan `fwrite()` untuk menulis data ke dalam file. Berikut ini adalah contohnya.

Misalkan kita mempunyai file `buah.txt` yang isinya adalah sbb:

```
Jeruk
Apel
Mangga
```

Selanjutnya kita punya kode PHP sbb:

```
<?
$myFile = "buah.txt";
$fh = fopen($myFile, 'a') or die("File tidak bisa dibuka");
$buah1 = "Anggur\n";
fwrite($fh, $buah1);
$buah2 = "Nanas\n";
fwrite($fh, $buah2);
fclose($fh);
?>
```

Kode di atas akan menambahkan 2 buah data baru ke file `buah.txt`. Perintah `\n` (new line) digunakan untuk ganti baris pada file.

## Upload File

Di beberapa aplikasi web, sering kita menjumpai proses upload file ke server. Berikut ini akan dibahas cara melakukan hal itu.

Langkah pertama untuk membuat aplikasi web guna upload file adalah membuat formnya terlebih dahulu. Berikut ini adalah salah satu contoh form dalam bentuk HTML

```
<form enctype="multipart/form-data" action="upload.php" method="post">
  <input type="hidden" name="MAX_FILE_SIZE" value="30000" />
  Nama File : <input name="userfile" type="file" />
  <input type="submit" value="Upload" />
</form>
```

Pada form di atas, kita membatasi ukuran file yang dapat diupload adalah 30 Kb.

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

```
<?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";
}

?>
```