

## Bab 7 File dan Direktori

- ❖ Membuat, Menulis dan Membaca Isi File
  - + Membuka dan Membuat File
  - + Menulis ke File
  - + Membaca Isi File
  - + Menutup File
- ❖ Membuat dan Menghapus Direktori
- ❖ Manipulasi File
  - + Mendapatkan Informasi File
  - + Copy, Rename dan Delete File
- ❖ Upload File

### Penanganan File

Dalam management file dan direktori, PHP menyediakan lebih dari 70 fungsi. Beberapa fungsi utama yang berhubungan dengan management file (create, write, append, dan delete), antara lain :

#### Membuka dan Membuat File

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

*Keterangan :*

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

Mode	Keterangan
r	Hanya untuk baca file, pointer berada di awal file
r+	Untuk baca dan tulis file, pointer berada di awal file
w	Hanya untuk tulis file, isi file lama dihapus, jika file belum ada maka akan di-create
w+	Untuk baca dan tulis file, isi file lama dihapus, jika file belum ada maka akan di-create
a	Hanya untuk menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create
a+	Untuk membaca dan menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create

#### Program 7.1

Nama File : file01.php

Deskripsi : Program mengakses (membuka) file dengan mode r.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>File berhasil dibuka</b>";
8 }
9 fclose($handle);
10 ?>
```



Gambar 7.1 Tampilan Program 7.1

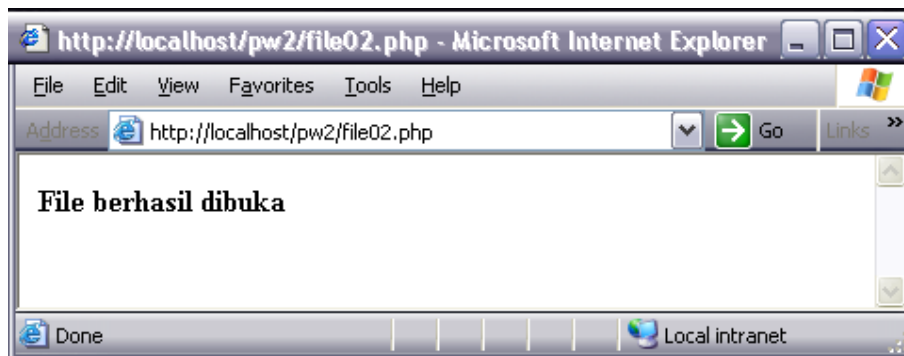
Program 7.2

Nama File : file02.php

Deskripsi : Program mengakses (membuka) file dengan mode w.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "w");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>File berhasil dibuka</b>";
8 }
9 fclose($handle);
10 ?>
    
```



Gambar 7.2 Tampilan Program 7.2

Menulis ke File

```

fwrite ($handle, $isi);
fputs ($handle, $isi);
    
```

Program 7.3

Nama File : file03.php

Deskripsi : Program menulis ke file.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "w");
    
```

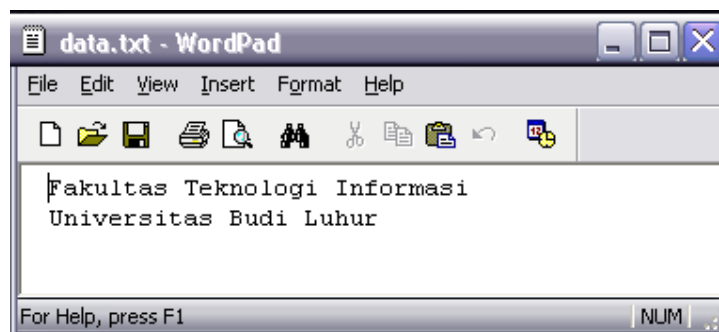
```

4  if (!$handle) {
5      echo "<b>File tidak dapat dibuka atau belum ada</b>";
6  } else {
7      fwrite ($handle, "Fakultas Teknologi Informasi\n");
8      fputs ($handle, "Universitas Budi Luhur\n");
9      //file_put_contents ($namafile, "Jakarta");
10     echo "<b>File berhasil ditulis</b>";
11 }
12 fclose($handle);
13 ?>

```

### Penjelasan Program 7.3

Untuk menulis ke file, dapat menggunakan fungsi **fwrite()** dan **fputs()**. Variabel **\$handle** merupakan mode pengaksesan file, dimana mode yang digunakan harus bisa menulis ke file. Contohnya mode **w** dan **a**. Selain itu, kita juga bisa menggunakan fungsi **file\_put\_contents()** dengan parameter berupa nama file dan isi yang akan ditambahkan. Jika program di atas dijalankan, maka isi file **data.txt** akan berisi seperti pada gambar 7.3 (buka dengan WordPad).



Gambar 7.3. File **data.txt** dibuka dengan WordPad

### Membaca Isi File

```

fread ($handle [, $panjang]);
fgets ($handle [, $panjang]);

```

### Program 7.4

Nama File : **file04.php**

Deskripsi : Program membaca dan menampilkan isi file.

```

1  <?php
2  $namafile = "data.txt";
3  $handle = fopen ($namafile, "r");
4  if (!$handle) {
5      echo "<b>File tidak dapat dibuka atau belum ada</b>";
6  } else {
7      $isi = fgets ($handle, 2048);
8      //$isi2 = fread ($handle, 20);
9      echo "Isi 1 : $isi<br>";
10     //echo "Isi 2 : $isi2<br>";
11 }
12 fclose($handle);
13 ?>

```

### Penjelasan Program 7.4

Untuk membaca isi file, bisa menggunakan fungsi **fgets()** dan **fread()**. Fungsi **fgets()** akan membaca isi file tiap baris. Sedangkan fungsi **fread()** akan membaca isi file setiap jumlah byte tertentu. Pada program di atas, baris ke-8 akan membaca file setiap 20 byte atau 20 karakter.

Program 7.5

Nama File : file05.php

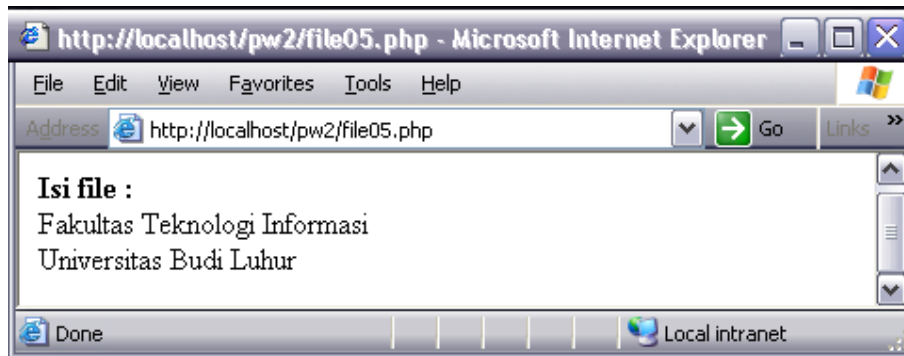
Deskripsi : Program membaca isi file baris tiap baris.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>Isi file : </b><br>";
8     while ($isi = fgets ($handle, 2048)) {
9         echo "$isi<br>";
10    }
11 }
12 fclose($handle);
13 ?>
    
```

Penjelasan Program 7.5

Program 7.5 di atas akan menampilkan keseluruhan isi file, dimana isi file akan dibaca tiap baris. Pembacaan dilakukan dari baris pertama sampai baris terakhir.



Gambar 7.4. Tampilan Program 7.5

Program 7.6

Nama File : file06.php

Deskripsi : Program membaca isi file baris tiap baris.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = @fopen($namafile, "r");
4 if ($handle) {
5     while (!feof($handle)) {
6         $buffer = fgets($handle, 4096);
7         echo $buffer."<br>";
8     }
9     fclose($handle);
10 }
11 ?>
    
```

Penjelasan Program 7.6

Program 7.6 di atas akan menampilkan keseluruhan isi file, dimana isi file akan dibaca tiap baris. Pembacaan dilakukan selama pointer file belum sampai di end-of-file. Fungsi feof() pada baris ke-5 berfungsi untuk memeriksa pointer file sudah berada di akhir file atau belum.

## Menutup File

```
fclose ($handle);
```

### Program 7.7

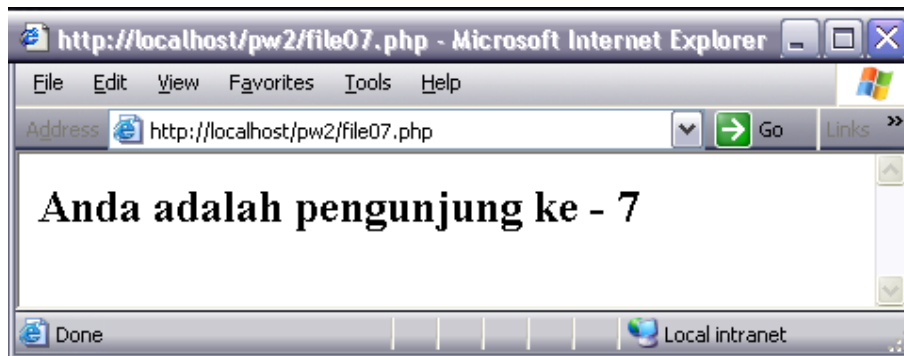
Nama File : file07.php

Deskripsi : Program aplikasi user counter dengan file.

```

1 <?php
2 $counter_file="counter.txt";
3 if (!file_exists ($counter_file)) {
4     fopen ($counter_file, "w");
5 }
6 $file = fopen($counter_file,"r");
7
8 $counter = fread($file,10);
9 fclose($file);
10
11 $counter++;
12
13 echo "<h2>Anda adalah pengunjung ke - $counter</h2>";
14 $file = fopen($counter_file, "w");
15 fwrite($file,$counter);
16 fclose($file);
17 ?>

```



Gambar 7.5. Tampilan Program 7.7

## Membuat dan Menghapus Direktori

### Membuat Direktori Baru

#### Program 7.8

Nama File : file08.php

Deskripsi : Program membuat direktori baru.

```

1 <?php
2 $dir = "images"; //nama direktori
3 $cek = mkdir ($dir);
4 if ($cek) {
5     echo "Direktori <b>$dir</b> berhasil dibuat";
6 } else {
7     echo "Direktori <b>$dir</b> gagal dibuat";
8 }
9 ?>

```

### Penjelasan Program 7.8

Program 7.8 di atas akan membuat direktori bernama "images" di direktori tempat program 7.8 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.

### Menghapus Direktori

#### Program 7.9

Nama File : file09.php

Deskripsi : Program menghapus direktori.

```

1 <?php
2 //buat direktori
3 $dir = "include"; //nama direktori
4 $cek = mkdir ($dir);
5 if ($cek) {
6     echo "Direktori <b>$dir</b> berhasil dibuat";
7 } else {
8     echo "Direktori <b>$dir</b> gagal dibuat";
9 }
10 //hapus direktori
11 $del = rmdir ($dir);
12 if ($del) {
13     echo "<br>Direktori <b>$dir</b> berhasil dihapus";
14 } else {
15     echo "<br>Direktori <b>$dir</b> gagal dihapus";
16 }
17 ?>

```

### Penjelasan Program 7.9

Program 7.9 di atas akan membuat direktori baru bernama "include" (lihat baris ke-4). Setelah itu, direktori akan dihapus dengan fungsi **rmdir()** pada baris ke-10. Fungsi **rmdir()** hanya bisa menghapus direktori kosong. Jika direktori ada isinya maka, gunakan fungsi rekursif untuk menghapus isi direktori. Lihat contoh di PHP Manual.

### Menampilkan Isi Direktori

#### Program 7.10

Nama File : file10.php

Deskripsi : Program menampilkan file dan direktori dalam suatu direktori.

```

1 <?php
2 $dir = "images";
3 if ($handle = opendir($dir)) {
4     while (false !== ($file = readdir($handle))) {
5         if ($file != "." && $file != "..") {
6             echo "$file<br>";
7         }
8     }
9     closedir($handle);
10 }
11 ?>

```

### Penjelasan Program 7.10

Program 7.10 di atas akan menampilkan di layar, semua file dan direktori yang berada dalam direktori "images". 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, terdapat dalam tabel sebagai berikut :

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 (dalam byte).
filetype(\$file)	Menghasilkan jenis file.

#### Program 7.11

Nama File : file11.php

Deskripsi : Program memeriksa keberadaan suatu file.

```

1 <?php
2 $filename = "data.txt";
3
4 if (file_exists($filename)) {
5     echo "File $filename ADA";
6 } else {
7     echo "File $filename TIDAK ADA";
8 }
9 ?>
```

#### Program 7.12

Nama File : file12.php

Deskripsi : Program mendapatkan beberapa informasi file.

```

1 <?php
2 $file = "data.txt";
3
4 if (is_file($file)) {
5     echo "File <b>$file</b> adalah FILE<br>";
6     //
7     if (is_executable ($file)) {
8         echo "File <b>$file</b> bisa dijalankan secara
9 langsung (executable)<br>";
10    } else {
11        echo "File <b>$file</b> TIDAK executable<br>";
12    }
13    //
14    if (is_writable ($file)) {
15        echo "File <b>$file</b> bisa ditulis / diedit<br>";
16    } else {
17        echo "File <b>$file</b> TIDAK bisa ditulis /
18 diedit<br>";
19    }
20 }
```

```

18 //
19 if (is_readable ($file)) {
20     echo "File <b>$file</b> bisa dibaca<br>";
21 } else {
22     echo "File <b>$file</b> TIDAK bisa dibaca<br>";
23 }
24 //
25 echo "Akses terakhir file <b>$file</b> = ". date("d-m-Y
H:i:s.", fileatime($file)). "<br>";
26 echo "File <b>$file</b> dibuat = ". date("d-m-Y H:i:s.",
filectime($file)). "<br>";
27 echo "Modifikasi terakhir file <b>$file</b> = ". date("d-
m-Y H:i:s.", filemtime($file)). "<br>";
28 echo "Ukuran file <b>$file</b> = ". filesize($file). "
bytes<br>";
29 echo "Jenis file <b>$file</b> = ". filetype($file). "<br>";
30 } else if (is_dir($file)){
31     echo "File <b>$file</b> adalah DIREKTORI<br>";
32 } else {
33     echo "File <b>$file</b> tidak dikenal<br>";
34 }
35 ?>

```

### Copy, Rename dan Delete File

#### Program 7.13

Nama File : file13.php

Deskripsi : Program meng-copy, rename dan delete file.

```

1 <?php
2 $file = "data.txt";
3 $filebaru = "databaru.txt";
4 $filebaru2 = "databaru2.txt";
5 if (copy ($file, $filebaru)) {
6     echo "File <b>$file</b> berhasil dicopy menjadi
<b>$filebaru</b>. <br>";
7 }
8 if (rename ($filebaru, $filebaru2)) {
9     echo "File <b>$filebaru</b> berhasil direname menjadi
<b>$filebaru2</b>. <br>";
10 }
11 if (unlink ($filebaru2)) {
12     echo "File <b>$filebaru2</b> berhasil dihapus. <br>";
13 }
14 ?>

```

#### Penjelasan Program 7.13

Program 7.13 di atas terdapat beberapa fungsi untuk meng-copy, rename dan delete file. Fungsi **copy()** pada baris ke-5 akan menduplikasi file \$file menjadi file \$filebaru. Selanjutnya fungsi **rename()** akan mengganti nama file \$filebaru menjadi \$filebaru2. Fungsi **unlink()** pada baris ke-11 akan menghapus file \$filebaru2.

### Upload File

**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"
```

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
4. *Destination folder* file upload harus bisa writable (bisa diakses), biasanya dengan permission 777 atau 775.

Program 7.14

Nama File : form\_upload.php

Deskripsi : Program menampilkan form upload file.

```

1 <html>
2 <head><title>Upload File</title></head>
3 <body>
4 <FORM ACTION="upload.php" METHOD="POST"
  ENCTYPE="multipart/form-data">
5     Upload File : <input type="file" name="file"><br>
6     <input type="submit" name="Upload" value="Upload">
7 </FORM>
8 </body>
9 </html>

```

Program 7.15

Nama File : upload.php

Deskripsi : Program proses upload file.

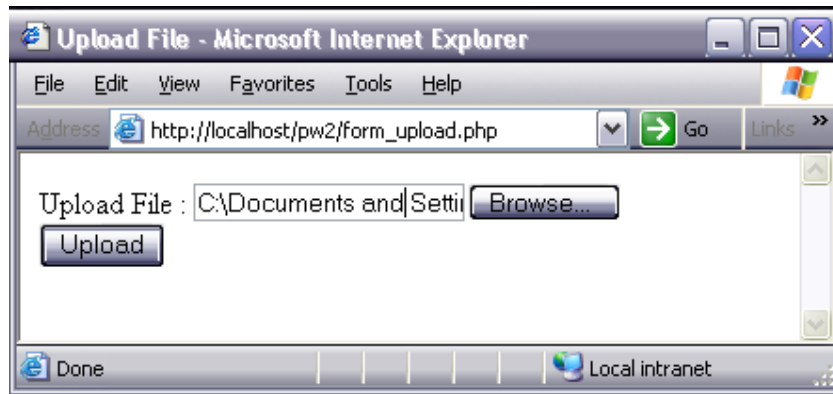
```

1 <?php
2 if (isset($_POST['Upload'])) {
3     $dir_upload = "images/";
4     $nama_file = $_FILES['file']['name'];
5     //
6     if (is_uploaded_file($_FILES['file']['tmp_name'])) {
7         $cek = move_uploaded_file ($_FILES['file']['tmp_name'],
  $dir_upload.$nama_file);
8         if ($cek) {
9             die ("File berhasil diupload");
10        } else {
11            die ("File gagal diupload");
12        }
13    }
14 }
15 ?>

```

Penjelasan Program 7.15

Program 7.15 di atas merupakan program sederhana penanganan upload file. Fungsi **is\_uploaded\_file()** pada baris ke-6 akan mengupload file yang telah dipilih melalui form pada program 7.14 ke folder temporary (sementara). Selanjutnya pada baris ke-7, file yang sudah diupload ke folder temporary akan dipindahkan (move) ke direktori yang diinginkan menggunakan fungsi **move\_uploaded\_file()**. Lihat baris ke-7!



Gambar 7.6. Tampilan Program 7.14