



INSTITUT TEKNOLOGI DEL

MATERI PRAKTIKUM

IF31103 – Pengembangan Situs Web II

SEMESTER II TAHUN AJAR 2018/2019

Session Date	:	21 Februari 2019
Semester	:	2
Courses	:	Pengembangan Situs Web II
Week/Session	:	4/2
Key Topics	:	Image Processing dan File Handling
Activity	:	Observasi Code
Duration	:	110 menit
Delivery	:	-
Deadline of delivery	:	-
Place of delivery	:	-
Goal	:	Mengerti file handling dan basic image processing

Sangat disarankan bagi anda untuk :

1. Melakukan observasi pada setiap kode yang ada.
2. Menggunakan IDE (NetBeans) atau JetBrains PhpStorm untuk membantu anda belajar.
3. Mencari sumber-sumber lain seperti buku, artikel, bahkan video untuk memperkaya wawasan dan meningkatkan pemahaman anda.
4. Jika anda merasa ada hal yang belum dipahami, silakan untuk berkonsultasi pada TA.
5. Setiap file php yang Anda buat harus ditambahkan comment sebagai keterangan seperti di bawah ini:

<!--

Nama : Nama Anda

NIM : XXX

Kelas : XXX

-->

Dengan demikian diharapkan anda mampu mengikuti materi kuliah dan praktikum sebaik mungkin.
Selamat belajar & good luck!

Introduction

Pada praktikum ini anda akan mempelajari bagaimana:

1. Berinteraksi dengan file mulai dari membuka, membaca, menulis hingga menutup file.
2. Mengembangkan fitur upload file
3. Membuat/ memanipulasi image hingga membuat chart (diagram).

File

File adalah objek yang ada pada suatu sistem komputer. *File* dibedakan atas dua, yakni dokumen (yang sering diasosiasikan dengan *file* itu sendiri) dan direktori. Direktori adalah salah satu jenis *file* yang menyimpan informasi tentang *file* lainnya secara hirarkis. Kegunaan *file* adalah sebagai media penyimpanan yang *persistent*.

Pada praktikum kali ini anda akan berinteraksi dengan *file*, mulai dari membuka dan menutup, membaca dan menulis informasi dari dan ke dalam *file*, hingga melakukan *upload*.

Pada dasarnya terdapat tiga tahap yang perlu dilakukan untuk berinteraksi dengan *file*: membuka 'koneksi', mengoperasikan dan menutup 'koneksi'.

Silakan download *file students.txt* dan meletakkannya tepat pada direktori praktikum.

Opening & closing file

Untuk membuka sebuah *file*, gunakan fungsi `fopen()` sementara untuk menutup gunakan fungsi `fclose()`. Fungsi `fopen()` memerlukan setidaknya dua parameter: *target file path*, dan *mode* yang digunakan serta mengembalikan *id* dari koneksi (*resource*) yang terbentuk. Sementara itu, fungsi `fclose()` hanya memerlukan satu parameter: *id* dari *resource* yang akan ditutup. Berikut ini adalah contoh penggunaan fungsi `fopen()` dan `fclose()`.

File: 01_opening_&_closing_file.php

```
<?php
    $file_name = 'students.txt';
    $mode      = 'r';
    $file      = fopen($file_name, $mode);
    echo(($file)? 'opened<br>' : 'unable to open<br>');
    $closed    = fclose($file);
    echo(($closed)? 'closed<br>' : 'unable to close<br>');
?>
```

Reading file

Salah satu operasi yang dapat dilakukan adalah membaca isi *file*, kali ini anda akan membaca *file students.txt* dan menampilkan isinya.

File: 02a_reading_using_fread.php

```
<?php
    $file_name = 'students.txt';
    $mode      = 'r';
    $file      = fopen($file_name, $mode);
    $file_size = filesize($file_name);
    $content   = fread($file, $file_size);
    echo($content);
    $closed    = fclose($file);
?>
```

Selain `fread()`, anda juga dapat menggunakan fungsi `fgets()`.

Reading file shortcuts

Fungsi `fread()` dan `fgets()` perlu didahului dengan pembukaan *file (resource opening)* dan penutupan (*resource closing*). Terdapat beberapa fungsi yang dapat digunakan sebagai *shortcuts* untuk membaca *file*, diantaranya: `file()` dan `file_get_contents()`. Fungsi `file()` mengembalikan seluruh isi *file* dalam bentuk *array* berdimensi satu dengan setiap elemen berisi satu baris isi *file*, sementara fungsi `file_get_contents()` mengembalikan seluruh isi *file* dalam sebuah *string*. Berikut ini contoh penggunaan fungsi `file()`.

File: 03b_reading_using_file.php

```
<?php
    $file_name = 'students.txt';
    // reads an entire file into a single string
    $content   = file($file_name);
    echo('<pre>');
    print_r($content);
?>
```

Writing file

Selain membaca *file*, operasi lain yang dapat dilakukan adalah menulis *file*. Anda dapat menggunakan `fwrite()` seperti pada contoh berikut.

File: 04a_writing_using_fwrite.php

```
<?php
    $file_name = 'new_students.txt';
    $mode      = 'w+';
    $file      = fopen($file_name, $mode);

    $content   = '11113212#Wiro Sableng#32TI1';
    fwrite($file, $content);
    $closed    = fclose($file);
?>
```

Seperti pada pembacaan *file*, juga terdapat *shortcut function* untuk penulisan *file*, yaitu `file_put_contents()`.

File: 04b_writing_using_file_put_contents.php

```
<?php
    $file_name = 'new_students.txt';
    $content    = '11113303#Jaka Sembung#32TK';
    file_put_contents($file_name, $content);
?>
```

Useful functions

Selain beberapa fungsi yang dipaparkan di atas, terdapat fungsi-fungsi lain yang juga dapat dipergunakan untuk memanipulasi *file*, berikut diantaranya:

No	Function	Usability
1	<code>copy()</code>	Copies a file
2	<code>delete()</code>	See <code>unlink()</code> or <code>unset()</code>
3	<code>dirname()</code>	Returns the directory name component of a path
4	<code>feof()</code>	Tests for end-of-file on an open file
5	<code>fflush()</code>	Flushes buffered output to an open file
6	<code>fgetc()</code>	Returns a character from an open file
7	<code>fgetcsv()</code>	Parses a line from an open file, checking for CSV fields
8	<code>fgets()</code>	Returns a line from an open file
9	<code>file_exists()</code>	Checks whether or not a file or directory exists
10	<code>fileatime()</code>	Returns the last access time of a file
11	<code>filectime()</code>	Returns the last change time of a file
12	<code>filemtime()</code>	Returns the last modification time of a file
13	<code>filesize()</code>	Returns the file size
14	<code>filetype()</code>	Returns the file type
15	<code>fpasssthru()</code>	Reads from an open file, until EOF , and writes the result to the output buffer
16	<code>fputcsv()</code>	Formats a line as CSV and writes it to an open file
17	<code>fscanf()</code>	Parses input from an open file according to a specified format
18	<code>fseek()</code>	Seeks in an open file
19	<code>fstat()</code>	Returns information about an open file
20	<code>ftell()</code>	Returns the current position in an open file
21	<code>is_dir()</code>	Checks whether a file is a directory
22	<code>is_executable()</code>	Checks whether a file is executable
23	<code>is_file()</code>	Checks whether a file is a regular file
24	<code>is_link()</code>	Checks whether a file is a link
25	<code>is_readable()</code>	Checks whether a file is readable
26	<code>is_uploaded_file()</code>	Checks whether a file was uploaded via HTTP POST
27	<code>is_writable()</code>	Checks whether a file is writeable
28	<code>mkdir()</code>	Creates a directory
29	<code>move_uploaded_file()</code>	Moves an uploaded file to a new location

30	rename()	Renames a file or directory
31	rewind()	Rewinds a file pointer
32	rmdir()	Removes an empty directory
33	set_file_buffer()	Sets the buffer size of an open file
34	stat()	Returns information about a file
35	unlink()	Deletes a file

diambil dari: w3schools.com, detail informasi setiap fungsi dapat dilihat di PHP documentation.

Working with directory

Direktori merupakan objek yang memiliki kemampuan menyimpan informasi tentang objek *file* lainnya secara *hirarkis*. Dengan ide yang mirip dengan operasi terhadap *file*, untuk mengoperasikan (membaca *entry* dari objek yang berada pada direktori) *file*, terlebih dahulu untuk membuat koneksi dengan fungsi `opendir()`.

File: 05_directory.php

```
<?php
    $dir_name    = 'C:\\xampp\\htdocs\\ibad\\w04s02p_file_handling';
    $dir         = opendir($dir_name);
    echo "$dir, contains:<br>";
    while (false !== ($entry = readdir($dir))) {
        echo "$entry<br>";
    }
?>
```

Uploading file

Dengan memanfaatkan PHP, anda dapat menyediakan layanan untuk mengunggah (*upload*) *file*. Berikut ini adalah contoh sederhana dari layanan *upload file*.

File: 06a_file_upload.php

```
upload the file:<br>
<form
    action="06b_file_upload_process.php"
    method="post"
    enctype="multipart/form-data">
    <input type="file" name="uploaded_file">
    <input type="submit" value="upload">
</form>
```

File: 06b_file_upload_process.php

```
<?php
    // file: 06b_file_upload_process.php
    // see what inside the $_FILES super global
    echo('<pre>');
    var_dump($_FILES);
```

```

// build the upload location
$uploaded_path = __DIR__ .
    DIRECTORY_SEPARATOR .
    $_FILES['uploaded_file']['name'];

// storing the file
$copied = copy(
    $_FILES['uploaded_file']['tmp_name'],
    $uploaded_path
);
echo(($copied)? "uploaded succesfully": "failed");
?>

```

Image

Pada akhir praktikum ini anda diharapkan dapat membuat atau memanipulasi *image* yang sudah ada menggunakan PHP hingga membuat *chart* (diagram). Anda perlu kreativitas dan imajinasi untuk mendesain *chart* yang lebih rumit seperti *bar chart*, dsb.

Basic image

Berikut ini adalah contoh manipulasi *image* dasar.

File: 01_basic_image.php

```

<?php

//create the canvas

$myImage = ImageCreate(150,150);

//set up some colors

$black = ImageColorAllocate($myImage, 0, 0, 0);
$white = ImageColorAllocate($myImage, 255, 255, 255);
$red = ImageColorAllocate($myImage, 255, 0, 0);
$green = ImageColorAllocate($myImage, 0, 255, 0);
$blue = ImageColorAllocate($myImage, 0, 0, 255);

//draw some rectangles

```

```
ImageRectangle($myImage, 15, 15, 55, 85, $red);

ImageRectangle($myImage, 55, 85, 125, 135, $white);

//output the image to the browser
header ("Content-type: image/gif");

ImageGif($myImage);

//clean up after yourself
ImageDestroy($myImage);

?>
```

Filled shapes

Berikut ini adalah contoh manipulasi *image* dengan *filled shapes*.

File: 02_color_fill.php

```
<?php

//create the canvas

$myImage = ImageCreate(150,150);

//set up some colors

$black = ImageColorAllocate($myImage, 0, 0, 0);

$white = ImageColorAllocate($myImage, 255, 255, 255);

$red = ImageColorAllocate($myImage, 255, 0, 0);

$green = ImageColorAllocate($myImage, 0, 255, 0);

$blue = ImageColorAllocate($myImage, 0, 0, 255);

//draw some rectangles

ImageFilledRectangle($myImage, 15, 15, 55, 85, $red);
```

```
ImageFilledRectangle($myImage, 55, 85, 125, 135, $white);

//output the image to the browser

header ("Content-type: image/jpeg");

ImageJpeg($myImage);

//clean up after yourself

ImageDestroy($myImage);

?>
```

Thumb image

Berikut ini merupakan contoh manipulasi yang umum kita jumpai, *thumb nail image*. Dengan memodifikasi praktikum sesi sebelumnya (`06b_file_upload_process.php`), anda dapat mengembangkan mekanisme *thumb nailing* sederhana yang secara spesifik hanya menerima *file* berformat **JPG**.

File: `06b_file_upload_process.php`

```
<?php

// we accept only JPG-typed file

if($_FILES['uploaded_file']['type'] != 'image/jpeg'){

    header("location: 06a_file_upload.php");

}

// set the content-type to JPG

header("content-type:image/jpg");

// set the thumb width size

$thumb_width = 100;
```



```

// create an image resource from an 'existing' image

$ori_image = imagecreatefromjpeg
($_FILES['uploaded_file']['tmp_name']);

// find out it's size to get the scale

// this is good to create a proper-sized thumb image

$ori_width = imagesx ($ori_image);

$ori_height = imagesy ($ori_image);

$scale = $ori_height/$ori_width;

// create a new true color image

$thumb_image = imagecreatetruecolor($thumb_width, $scale *
$thumb_width);

imagecopyresized ($thumb_image, $ori_image, 0, 0, 0, 0,
    $thumb_width, $scale * $thumb_width,$ori_width, $ori_height);

// output the image

imagejpeg($thumb_image);

// clean up

imagedestroy($thumb_image);

imagedestroy($ori_image);

?>

```

Pie chart

Manipulasi *image* lebih luas dapat digunakan sebagai mekanisme berkomunikasi dalam bentuk diagram, salah satunya *pie chart*.

File: 03_pie_chart.php

```
<?php

    //create the canvas

    $myImage = ImageCreate(300,300);

    //set up some colors hal 2 dari 6

    $white = ImageColorAllocate($myImage, 255, 255, 255);

    $red = ImageColorAllocate($myImage, 255, 0, 0);

    $green = ImageColorAllocate($myImage, 0, 255, 0);

    $blue = ImageColorAllocate($myImage, 0, 0, 255);

    //draw a pie

    ImageFilledArc($myImage, 100,100,200,150,0,90, $red, IMG_ARC_PIE);

    ImageFilledArc($myImage, 100,100,200,150,90,180, $green, IMG_ARC_PIE);

    ImageFilledArc($myImage, 100,100,200,150,180,360, $blue, IMG_ARC_PIE);

    //output the image to the browser

    header ("Content-type: image/png");

    ImagePng($myImage);

    //clean up after yourself

    ImageDestroy($myImage);

?>
```

3d pie chart

Berikut ini adalah contoh diagram *pie chart* dalam bentuk 3D.

File: 04_3d_pie_chart.php

```
<?php

//create the canvas

$myImage = ImageCreate(300,300);

//set up some colors hal 2 dari 6

$white = ImageColorAllocate($myImage, 255, 255, 255);

$red = ImageColorAllocate($myImage, 255, 0, 0);

$green = ImageColorAllocate($myImage, 0, 255, 0);

$blue = ImageColorAllocate($myImage, 0, 0, 255);

$lt_red = ImageColorAllocate($myImage, 255, 150, 150);

$lt_green = ImageColorAllocate($myImage, 150, 255, 150);

$lt_blue = ImageColorAllocate($myImage, 150, 150, 255);

//draw the shaded area

for ($i = 120;$i > 100;$i--) {

    ImageFilledArc ($myImage,100,$i,200,150,0,90, $lt_red,
IMG_ARC_PIE);

    ImageFilledArc ($myImage,100,$i,200,150,90,180, $lt_green,
IMG_ARC_PIE);

    ImageFilledArc ($myImage,100,$i,200,150,180,360, $lt_blue,
IMG_ARC_PIE);

}

//draw a pie

ImageFilledArc($myImage, 100,100,200,150,0,90, $red, IMG_ARC_PIE);

ImageFilledArc($myImage, 100,100,200,150,90,180, $green, IMG_ARC_PIE);
```

```
ImageFilledArc($myImage, 100,100,200,150,180,360, $blue, IMG_ARC_PIE);

//output the image to the browser

header ("Content-type: image/png");

ImagePng($myImage);

//clean up after yourself

ImageDestroy($myImage);

?>
```

Captcha Image

Pada bagian ini, anda akan mencoba menciptakan captcha sederhana dan contoh penggunaannya pada sebuah form sederhana untuk validasi.

File : captcha_creator.php

```
<?php

session_start();

header ("Content-type: image/png");

if(isset($_SESSION['my_captcha']))

{ unset($_SESSION['my_captcha']); }

$string1="abcdefghijklmnopqrstuvwxyz";

$string2="1234567890";

$string=$string1.$string2;

$string= str_shuffle($string);

$random_text= substr($string,0,8);

$_SESSION['my_captcha'] =$random_text;
```

```

$im = @ImageCreate (80, 20)
or die ("Cannot Initialize new GD image stream");
$background_color = ImageColorAllocate ($im, 204, 204, 204);
$text_color = ImageColorAllocate ($im, 51, 51, 255);
ImageString($im,5,8,2,$_SESSION['my_captcha'],$text_color);
ImagePng ($im);
imagedestroy($im);

?>

```

File : captcha_form.php

```

<script type="text/javascript">
function reload()
{
    img = document.getElementById("capt");
    img.src="captcha_creator.php?rand_number=" + Math.random();
}
</script>

Input Captcha shown<br>
<form type=post action=captcha_proses.php><input type=text name=t1>
<img src=captcha_creator.php id="capt">
<input type=button onClick=reload(); value='Reload Captcha'>
<input type=submit value='submit'></form>
</body>
</html>

```

File : captcha_proses.php

```
<?php
session_start();
echo "Text as entered by user  : $_GET[t1] <br>";
echo "Captcha shown : $_SESSION[my_captcha] <br>";
if($_GET['t1'] == $_SESSION['my_captcha']){
echo "Captcha validation passed ";
}else {
echo "Captcha validation failed ";
}
unset($_SESSION['my_captcha']);
?>
```