

String	Constant	Datetime
<code>string (quotes, nowdoc, heredoc)</code>	<code>Syntax</code>	<code>date_default_timezone_set (string)</code>
<code>Semua fungsi string</code>	<code>Predefined Constant</code>	<code>time()</code>
<code>str_replace (search, replace, string)</code>	<code>__LINE__</code>	<code>date (format, timestamp = time())</code>
<code>strtoupper (string)</code>	<code>__FILE__</code>	<code>strtotime (string)</code>
<code>strtolower (string)</code>	<code>__DIR__</code>	<code>mktime (h, m, s, m, d, y)</code>
<code>ucfirst (string)</code>	<code>__FUNCTION__</code>	<code>getdate (timestamp = time())</code>
<code>ucwords (string)</code>	<code>__CLASS__</code>	<code>date_create (string = 'now' ,object DateTimeZone = null)</code>
<code>trim (string, character_mask)</code>	<code>Array</code>	<code>date_diff (datetime1, datetime2)</code>
<code>ltrim (string, character_mask)</code>	<code>Mendefinisikan Array</code>	<code>Format Tanggal</code>
<code>rtrim (string, character_mask)</code>	<code>array_push (array, value1, value2, ...)</code>	<code>-- DAY --</code>
<code>strops (string, needle)</code>	<code>array_pop (array)</code>	<code>d 01 s.d 31</code>
<code>substr (string, start, length)</code>	<code>array_unshift (array, value1, value2, ...)</code>	<code>j 1 s.d 31</code>
<code>strlen (string)</code>	<code>asort (array)</code>	<code>D Mon s.d Sun</code>
<code>strstr (string, needle, before_needle = false)</code>	<code>arsort (array)</code>	<code>I Sunday s.d Saturday</code>
<code>nl2br (string, is_xhtml = true)</code>	<code>ksort (array)</code>	<code>N 1 (Monday) s.d 7 (Saturday)</code>
<code>explode (delimiter, string)</code>	<code>krsort (array)</code>	<code>w 0 (Sunday) s.d 6 (Saturday)</code>
<code>implode (string = '', array)</code>	<code>array_keys (array)</code>	<code>-- MONTH --</code>
<code>. (dot) to concatenate string</code>	<code>array_values (array)</code>	<code>m 01 s.d 12</code>
<code>echo print_r var_dump</code>	<code>max (array)</code>	<code>n 1 s.d 12</code>
<code>is_string (variable)</code>	<code>min (array)</code>	<code>F January s.d December</code>
<code>Variable</code>	<code>count (array, mode = COUNT_NORMAL)</code>	<code>M Jan s.d Dec</code>
<code>Basic</code>	<code>join (array, separator)</code>	<code>t Jumlah hari dalam satu bulan (28 - 31)</code>
<code>Variable Scope</code>	<code>unset (array)</code>	<code>-- YEAR --</code>
<code>Variable Variables</code>	<code>array_search (needle, array)</code>	<code>Y 4 digit tahun (misal 2017)</code>
<code>Predefined Variables</code>	<code>array_map (callback, array)</code>	<code>y 2 digit tahun (misal 17)</code>
<code>\$_GET</code>	<code>list (array)</code>	<code>L Apakah tahun kabisat? (1 ya, 0 tidak)</code>
<code>\$_POST</code>	<code>in_array (needle, array)</code>	<code>-- TIME --</code>
<code>\$_FILES</code>	<code>key_exists (string, array)</code>	<code>a am atau pm</code>
<code>\$_REQUEST</code>	<code>array_merge (arr, arr)</code>	<code>A AM atau PM</code>
<code>\$_SESSION</code>	<code>is_array (array)</code>	<code>g Jam: 1 s.d 12</code>
<code>\$_COOKIE</code>	<code>array operators</code>	<code>h Jam: 01 s.d 12</code>
		<code>G Jam: 0 s.d 23</code>
		<code>H Jam: 00 s.d 23</code>
		<code>i Menit (00 s.d 59)</code>
		<code>s Detik (00 s.d 59)</code>

File and Folder

`fopen (file, mode)`
`fclose (handle)`
`fread (handle, length)`
`file_get_content (string)`
`file_put_contents (file, data)`
`filesize (file)`
`rename (old_name, new_name)`
`copy (source, destination)`
`unlink (file)`
`basename (path)`
`file_exists (file)`
`is_file (file)`
`include (file) | include 'file'`
`include_once (file) | include_once 'file'`
`require (file) | require 'file'`

Directory

`mkdir (path)`
`rmdir (path)`
`is_dir (path)`
`scandir (path)`
`file_exists (dir)`
`dirname (dir)`

Mode for fopen()

'r' File dibuka untuk dibaca, cursor berada di awal konten.
'r+' File dibuka untuk dibaca dan ditulis. 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 dibaca dan ditulis. 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 ditulis dan dibaca. Cursor diletakkan di akhir konten. Jika file tidak ditemukan, buat file baru.

IF ELSE dan Switch Case

`if ... elseif ... else`
`ternary operator`
`switch ... case`

PHP Loop

`for () {...}`
`while () {...}`
`do () while{...}`
`foreach () {...}`

PHP dan MySQL

`mysqli_connect(server, user, pass, db_name)`
`mysqli_select_db (db_name)`
`mysqli_query (sql)`
`mysqli_fetch_array (result)`
`mysqli_fetch_assoc (result)`
`mysqli_fetch_row (result)`
`mysqli_fetch_all (result, resulttype)`
`mysqli_affected_rows ($link)`
`mysqli_num_rows (result)`
`mysqli_free_result (result)`
`mysqli_close ($link)`
`mysqli_insert_id ($link)`
`mysqli_escape_string ($link, string)`
`mysqli_connect_error ($link)`
`mysqli_error ($link)`

Transaction

`mysqli_begin_transaction ($link)`
`mysqli_commit ($link)`
`mysqli_rollback ($link)`
`mysqli_savepoint ($conn, string)`
`mysqli_autocommit ($link)`

File Upload

`Handling file upload`
`$_POST | $_FILES | <form>`
`is_uploaded_file (file)`
`move_uploaded_file (tmp_file, new_file)`
`error code`
`php.ini | ini_set() | .htaccess`
`MAX_FILE_SIZE`

Array - Part I

array() | []

Mendefinisikan array

```
$arr = array(); // Semua versi PHP
$arr = []; // php >= 5.4
```

Mengakses array

```
$arr = array('foo', 'bar');
echo $arr[1]; // bar
$arr = array('a' => 'foo', 'b' => 'bar');
echo $arr['a']; // foo
```

array_push(arr, val1, ...) | array_pop(arr)

array_push: menambah elemen pada akhir array

```
$arr = array('foo', 'bar');
array_push($arr, 'alfa', 'beta');
```

array_pop: menghapus elemen terakhir array

```
$arr = array('foo', 'bar');
array_pop($arr); // $arr: Array ([0]=>foo)
```

array_unshift (arr, val) | array_shift (arr)

array_unshift: menambah elemen pertama array

```
$arr = array('foo', 'bar');
array_unshift($arr, 'alfa');
// $arr: Array ([0]=>alfa [1]=>foo [2]=>bar)
```

array_shift: menghapus elemen pertama array

```
$arr = array('foo', 'bar');
array_unshift($arr); // $arr: Array ([0]=>bar)
```

asort (arr) | arsort (arr)

asort: array sort, mengubah urutan value menjadi ascending

```
$arr = array('a' => 'foo', 'b' => 'bar');
asort($arr); // $arr: Array ([b]=>bar [a]=>foo)
```

arsort: array reverse sort, urutan value menjadi descending

```
$arr = array('bar', 'foo');
arsort($arr); // $arr: Array ([0]=>foo [1]=>bar)
```

ksort (arr) | krsort (arr)

ksort = key sort, mengubah urutan key secara ascending

```
$arr = array('b' => 'bar', 'a' => 'foo');
ksort($arr); // $arr: Array ([a]=>foo [b]=>bar)
```

krsort = key reverse sort, urutan key menjadi descending

```
$arr = array('a' => 'foo', 'b' => 'bar');
krsort($arr); // $arr: Array ([b]=>bar [a]=>foo)
```

array_keys (arr) | array_values (arr)

array_keys: mengambil key dari array, array_values: mengambil value dari array

```
$arr = array('a' => 'foo', 'b' => 'bar');
$key = array_keys ($arr); // Array ([0]=>a [1]=>b)
$key = array_values ($arr); // Array([0]=>foo [1]=>bar)
```

max (arr) | min (arr)

max mengambil value maksimal, min value minimal

```
$arr = array ('a'=>1, 'b'=>2, 'c'=>3);
max($arr); // 3
min($arr); // 1
```

count (arr)

Menghitung banyaknya elemen array

```
$arr = array('foo', 'bar');
echo count($arr); //2
```

join (arr, separator = '')

Menggabung value array menjadi string

```
$arr = array('a' => 'foo', 'b' => 'bar');
$string = join ($arr, ', '); // foo, bar
```

unset (arr[index])

Menghapus elemen array berdasarkan index

```
$arr = array('foo', 'bar');
unset($arr[1]); // $arr: Array ( [0]=>foo )
```

array_search (string, arr)

Mencari index berdasarkan value

```
$arr = array('a' => 'foo', 'b' => 'bar');
$key = array_search('bar', $arr); // $key: b
```

array_map (callback, arr)

Memanipulasi value dari array

```
$arr = array('foo', 'bar');
$upper = array_map('strtoupper', $arr);
// $upper: Array ([0]=>FOO [1]=>BAR)
```

Dengan fungsi sendiri:

```
$harga = array(1000, 2000);
$neto = array_map('neto', $harga);
function neto($arr_value) {
    return $arr_value * 0.5;
}
// $neto: Array ( [0] => 500 [1] => 1000 )
```

Array - Part II

list (arr)

Assign variabel pada value dari array

```
$arr = array ('merah', 'mobil');
list ($warna, $kendaraan) = $arr;
echo $warna; // merah
```

```
list (, $kendaraan) = $arr;
echo $kendaraan; // mobil
```

in_array (str, arr) | key_exists (str, arr)

in_array: cek elemen array

```
$arr = array('bar', 'foo');
if (in_array('foo', $arr)) { echo 'exists' }
```

key_exists: cek key array

```
$arr = array('a' => 'foo', 'b' => 'bar');
if (key_exists('b', $arr)) { echo 'exists' }
```

array_merge(arr, arr) | array(arr) + array(arr)

array_merge: menggabung array, hasil indexed array

```
$arr = array('foo', 'bar');
$arr2 = array('bar', 'alfa');
$merge = array_merge($arr, $arr2);
// Array ([0]=>foo [1]=>bar [2]=>bar [3]=>alfa)
```

Operator + untuk mengabung array berdasarkan index

```
$arr = array('bar', 'foo');
$arr2 = array('a' => 'alfa');
$arr3 = $arr + $arr2;
// $arr3: Array ([0]=>bar [1]=>foo [a]=>alfa)

$arr = array('bar', 'foo');
$arr2 = array('alfa', 'beta', 'charlie');
// $arr3: Array ([0]=>bar [1]=>foo [2]=>charlie)
```

is_array (arr)

Tes apakah argumen berbentuk array

```
$arg = 'string';
if (is_array($arg)) {
    echo 'array';
} else {
    echo 'bukan array';
} // bukan array
```

Jenis Array

Jenis Array: Indexed, associative, dan multidimensi. Array dapat digambarkan seperti tabel yang memiliki dua kolom:

Jenis Array

key	value
0	Mobil
1	Sepeda
2	Truk
3	Motor
4	Bus

Indexed Array

key	value
id	786
pass	123
role	admin
active	1
level	1

Associative Array

```
$kendaraan[1] // Sepeda
$user['pass'] // 123
```

Value dari array juga bisa berbentuk array (menjadi key dari array yang lain), misal:

Jenis Array

key	value
merk	Toyota
type	Vios
year	2016
0	Mobil [array]
1	Sepeda
2	Truk
3	Motor [array]
4	Bus

Array Mobil

key	value
0	Honda
1	Yamaha
2	Suzuki

Array Kendaraaan

key	value
0	Honda
1	Yamaha
2	Suzuki

Array Motor

Multidimensional Array

Pada contoh diatas, value Mobil dan Motor menjadi key dari array mobil dan array motor, bentuk ini disebut Array Multidimensi (Multidimensional Array)

```
$kendaraan = array( array('merk' => 'Toyota',
                           'type' => 'Vios',
                           'year' => 2016
                         ),
                     'Sepeda',
                     'Truk',
                     array('Honda', 'Yamaha',
                           'Suzuki'),
                     'Bus',
                   );
$kendaraan[0]['merk']; // Toyota
```

PHP dan FORM

\$_GET

HTML Form: gunakan method="get"

```
<form method="get" action="">
    <input type="text" name="keyword"/>
    <input type="submit" name="submit"
    value="Search"/>
</form>
```

PHP Code:

```
if (isset($_GET['keyword'])) {
    $keyword = $_GET['keyword'];
    $sql = 'SELECT *
        FROM article
        WHERE title
            LIKE "%' . $keyword . '%"';
}
```

Pada GET, data dikirim lewat url, gunakan GET untuk data yang tidak sensitif, seperti untuk pencarian dan pagination.

\$_POST

HTML Form: gunakan method="post"

```
<form method="post" action="">
    <input type="password" name="pass"/>
    <input type="submit" name="submit"
    value="Login"/>
</form>
```

PHP Code:

```
if (isset($_POST['submit'])) {
    $sql = 'SELECT user, password
        FROM user
        WHERE password = "'.$_POST['pass'].''";
    $result = mysqli_query($sql);
}
```

Pada POST, data dikirim lewat HTTP HEADER, sehingga tidak terlihat. Gunakan POST untuk data sensitif, seperti password, atau ketika terjadi perubahan data pada database seperti form edit atau delete data.

\$_FILES

HTML Form: gunakan enctype="multipart/form-data"

```
<form method="post" action=""
enctype="multipart/form-data">
    <input type="file" name="user_file"/>
    <input type="submit" name="submit"
    value="Upload"/>
</form>
```

PHP Code:

```
if (isset($_POST['submit'])) {
    echo '<pre>'; print_r($_FILES); echo'</pre>';
}
```

Nilai Input sama dengan data yang disubmit

Pada contoh dibawah, jika form disubmit, maka input keyword akan berisi string yang dicari (Method GET):

```
<form method="get" action="">
    <input type="text" name="keyword"
        value="<?=@$_GET['keyword'] ?: ""?>"/>
    <input type="submit" name="submit"
        value="Search"/>
</form>
```

Penambahan @ untuk menghilangkan warning ketika variabel \$_GET['keyword'] belum terdefinisi, misal ketika form pertama kali dibuka. Contoh untuk method POST

```
<form method="post" action="">
    <input type="text" name="nama"
        placeholder="Nama Anda"
        value="<?=@$_POST['nama'] ?: ""?>" />
    <input type="submit" name="submit"
        value="Simpan"/>
</form>
```

ELEMEN SELECT

Usahakan gunakan loop ketika membuat elemen select

```
<form method="post" action="">
    <?php
        $bulan = array('Januari',
                      'Februari',
                      'Maret'
                    );
        echo '<select name="bulan">';
        foreach ($bulan as $key => $nama_bulan) {
            $selected = @$_POST['bulan'] == $key
                ? ' SELECTED="SELECTED"' : '';
            echo '<option value="'.$key.'"'.
                $selected .'>' .
                $nama_bulan .
            '</option>';
        }
        echo '</select>';
    ?>
    <input type="submit" name="submit"
        value="Simpan"/>
</form>
```

Penggunaan loop akan memudahkan ketika ada penambahan maupun pengurangan elemen, disamping itu juga memudahkan pemberian atribut selected="selected". Opsi akan membuat opsi otomatis terpilih sesuai dengan yang dipilih user.

Example: Menampilkan data database

```
1. <?php
2. // SUBMIT
3. $hasil_query = false;
4. $error = $warning = array();
5. if (isset($_POST['submit']))
6. {
7.     $form_error = '';
8.     if (!trim($_POST['nama_barang'])) {
9.         $error = 'Nama barang harus diisi';
10.    }
11.
12.    if (!$error) {
13.        $conn = @mysqli_connect('localhost', 'root', '', 'db');
14.        if (!$conn) {
15.            $error[] = mysqli_connect_error($conn);
16.        } else {
17.            $where = trim($_POST['nama_barang'])
18.                  ? 'nama_barang LIKE "%'
19.                    .$_POST['nama_barang']. "%"
20.                  : '';
21.
22.            $sql = 'SELECT * FROM barang WHERE ' . $where;
23.            $result = mysqli_query ($conn, $sql);
24.
25.            // RESULT
26.            if (!$result) {
27.                $error[] = mysqli_error($conn)
28.                          . '<br><strong>SQL Query</strong>: ' . $sql;
29.            } else {
30.                $num_rows = mysqli_num_rows($result);
31.                if (!$num_rows) {
32.                    $warning[] = 'Data tidak ditemukan';
33.                } else {
34.                    $hasil_query = true;
35.                }
36.            }
37.        }
38.    }
39. }?>
<html>
<head> <title>PHP dan FORM</title> </head>
<body>
<?php
```

```
44. if ($error) {
45.     echo '<div>Error: ' . join($error, ', ') . '</div>';
46.
47.
48. if ($warning) {
49.     echo '<div>' . join($warning, ', ') . '</div>';
50. }
51. ?>
52. <form action="" method="post">
53.     <div>
54.         <label>Nama Barang</label>
55.         <input type="text" name="nama_barang"
56.               value=<?=@$_POST['nama_barang'] ?: ''?> />
57.     </div>
58.     <div>
59.         <input type="submit" name="submit" value="Submit"/>
60.     </div>
61. </form>
62. <?php
63. if (isset($_POST['submit'])) {
64.
65.     if ($hasil_query)
66.     {
67.         echo 'Ditemukan ' . $num_rows . ' data';
68.         $thead = ' <tr>
69.             <th>No</th>
70.             <th>Nama Barang</th>
71.         </tr> ';
72.
73.         echo ' <table> <thead>' . $thead . '</thead>
74.             <tbody>';
75.
76.         $no = 1;
77.         while($row = mysqli_fetch_array($result)) {
78.             echo '<tr> <td>' . $no . '</td>
79.                   <td>' . $row['nama_barang'] . '</td>
80.             </tr> ';
81.             $no++;
82.         }
83.         echo '</tbody></table>';
84.     }
85. }
86. </div>
87. </body>
88. </html>
```

Datetime ref >**date_default_timezone_set(string)**

Setting timezone

```
date_default_timezone_set('Asia/Jakarta');
```

Untuk perubahan permanen ubah file konfigurasi pada file php.ini

```
date.timezone = Asia/Jakarta
```

timestamp

timestamp adalah istilah untuk waktu dalam format UNIX berupa jumlah detik dari epoch time (01-01-1970 00:00:00), timestamp 50 berarti 01-01-1970 00:00:50. Untuk membuat timestamp gunakan fungsi time(), strtotime(), dan mktime()

date (format, timestamp = time())

Menampilkan tanggal dengan format tertentu

```
$tgl = date ('d-m-Y H:i:s'); // Sekarang: 22-03-2017 05:31:27
$tgl = date ('d-m-Y', strtotime('2017-05-01'));
// 01-05-2017
```

time()

fungsi time() tidak memiliki argumen, menghasilkan waktu sekarang dalam bentuk UNIX

```
time(); // 1490135985
date('d-m-Y H:i:s', time()); 22-03-2017 05:32:27
date('d-m-Y H:i:s', time() + 60); 22-03-2017 05:33:27
```

strtotime (str)

Menghasilkan tanggal format UNIX (GMT + 0) dari string

```
strtotime('now'); // 1490135639
date('d-m-Y H:i:s', strtotime('now')); // 22-03-2017 05:35:04
date('d-m-Y H:i:s', strtotime('+1 day')); // 23-03-2017 05:35:04
date('d-m-Y H:i:s', strtotime('-24 hours')); // 21-03-2017 05:35:04
```

mktime(hour, min, sec, month, day, year)

Menghasilkan tanggal format UNIX

```
date('d-m-Y', mktime(0,0,0,1,32,2017)); // 01-02-2017
date('d-m-Y', mktime( 0, 0, 0, date('n') - 1,
date('j'), date('Y')))
```

getdate (timestamp = time())

Menghasilkan informasi tanggal dan waktu dalam array

```
$today = getdate(); // hari ini
// Array ( [seconds] => 0 [minutes] => 50
[hours] => 23 [mday] => 21 [wday] => 2 [mon] =>
3 [year] => 2017 [yday] => 79 [weekday] =>
Tuesday [month] => March [0] => 1490136600 )
```

date_create (string,object DateTimeZone=null)

Menghasilkan objek DateTime

```
$now = date_create(); // waktu sekarang
$tgl = date_format($now, 'd-m-Y H:i:s'); // 22-03-2017 12:27:23
$now = date_create('now',
new DateTimeZone('Asia/Jakarta'));
// 22-03-2017 18:27:23
```

date_diff (datetime1, datetime2)

Menghitung selisih tanggal dan waktu, argumen harus berupa objek DateInterval (hasil fungsi date_create())

```
$awal = date_create('1988-08-10');
$akhir = date_create(); // waktu sekarang
$diff = date_diff( $awal, $akhir );
echo 'Selisih: ' . $diff->y. ' tahun, ' .
$diff->m. ' bulan, ' . $diff->d. ' hari, ' .
$diff->h. ' jam, ' . $diff->i. ' menit, ' .
$diff->s. ' detik, ';
// Selisih: 28 tahun, 5 bulan, 9 hari, 13 jam, 7 menit, 7 detik
```

```
echo 'Total selisih hari : ' . $diff->days;
// Total selisih hari: 10398
```

Format output:

```
$diff->format('Usia anda adalah %Y tahun %a');
// Usia anda adalah 29 tahun 9 hari.
```

Format: **Tahun:** %Y 4 digit tahun dan %y 2 digit tahun, **Bulan:** %M bulan dengan awalan 0 dan %m tanpa awalan 0, **Hari:** %D hari dengan awalan 0, %d hari tanpa awalan 0, dan %a untuk total hari

Tanggal Bahasa Indonesia

PHP tidak menyediakan fungsi untuk menampilkan tanggal dalam Bahasa Indonesia, contoh solusi:

```
$hari = array(1=>'Senin', 'Selasa', 'Rabu');
$bulan = array(1=>'Januari', 'Februari');
echo $hari[date('N')] . ', ' . date('d') . ' '
. $bulan[date('n')] . ' ' . date('Y');
// Rabu, 22 Februari 2017
```

Character untuk argumen format [ref »](#)

-- Day --

d	01 s.d 31
j	1 s.d 31
Hari dalam minggu	
D	Mon s.d Sun
l	Sunday s.d Saturday
N	1 (Monday) s.d 7 (Saturday)
w	0 (Sunday) s.d 6 (Saturday)

-- Month --

m	01 s.d 12
n	1 s.d 12
F	January s.d December
M	Jan s.d Dec
t	Jumlah hari dalam satu bulan (28 s.d 31)

-- Year --

Y	4 digit tahun (misal: 2017)
y	2 digit tahun (misal: 17)
L	Tes apakah tahun kabisat (1 ya, 0 tidak)

-- Time --

a	am atau pm
A	AM atau PM

Jam

g	1 s.d 12
h	01 s.d 12
G	0 s.d 23
H	00 s.d 23

Menit dan Detik

i	Menit (00 s.d 59)
s	Detik (00 s.d 59)

Relative Formats [ref »](#)

Used Symbol

Deskripsi	Format
ordinal	'first' 'second' 'third' 'fourth' 'fifth' 'sixth' 'seventh' 'eighth' 'ninth' 'tenth' 'eleventh' 'twelfth' 'next' 'last' 'previous' 'this'
daytext	'weekday' 'weekdays'
reltext	'next' 'last' 'previous' 'this'
unit	(('sec' 'second' 'min' 'minute' 'hour' 'day' 'fortnight' 'forthnight' 'month' 'year') 's'?') 'weeks' daytext

Day-based Notations

Format	Contoh
'yesterday'	"yesterday 14:00"
'midnight'	
'today'	
'now'	
'first day of'	"first day of April 2017"
'last day of'	"last day of next month"
ordinal [spasi] dayname [spasi] of'	"first sat of April 2017"
'last' [spasi] dayname [spasi] 'of'	"last sat of April 2017"

File ref >

fopen (file, mode)

Membuka file baik pada lokal maupun remote (URL), variabel penyimpannya 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 memasukkan 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);
```

File Upload [ref »](#)

HTML Form <form>

Gunakan atribut `enctype="multipart/form-data"` pada form HTML

```
<form method="post" action=""  
enctype="multipart/form-data">  
    <input type="file" name="user_file"/>  
    <input type="submit" name="submit"  
value="Upload"/>  
</form>
```

`$_FILES`

Data file yang diupload disimpan pada variabel `$_FILES`

```
Array  
(  
    [user_file] => Array  
        (  
            [name] => image.jpg  
            [type] => image/jpeg  
            [tmp_name] => E:\xampp\tmp\php995.tmp  
            [error] => 0  
            [size] => 392946  
        )  
)
```

`is_uploaded_file (file)`

Pertama kali file di upload, akan diletakkan di folder temporary php, jika berhasil `is_uploaded_file` bernilai true

```
if(is_uploaded_file(  
    $_FILES['user_file']['tmp_name']))  
) {...}
```

`move_uploaded_file (tmp_file, new_file)`

Pindahkan file dari direktori temp ke lokasi baru

```
$tmp_name = $_FILES['user_file']['tmp_name'];  
$destination = 'files/new_file.txt';  
move_uploaded_file($tmp_name, $destination);
```

WARNING: Jika file telah ada, maka akan ditimpas

Maksimal file upload

Secara default maksimal file yang dapat diupload sebesar 2M, untuk mengubahnya, ada tiga cara yang dapat dilakukan

`php.ini` [ref »](#)

Ubah parameter pada file `php.ini`

```
memory_limit = 128M  
upload_max_filesize = 128M  
post_max_size = 128M
```

`post_max_size` harus \geq `upload_max_filesize`. `memory_limit` harus \geq `post_max_size`

`ini_set()` PHP < 5.3

Cara kedua adalah menggunakan fungsi `ini_set` (hanya dapat digunakan pada PHP versi dibawah 5.3)

```
ini_set('upload_max_filesize', '64M');  
ini_set('post_max_size', '64M');
```

.htaccess

Tambahkan parameter pada file `.htaccess`

```
<IfModule mod_php5.c>  
    php_value upload_max_filesize 48M  
    php_value post_max_size 48M  
</IfModule>
```

Membatasi ukuran file

```
if ($_FILES['user_file']['size'] > 1048576) {  
    echo 'Error: Ukuran file maksimal 1 Mb';  
}
```

`MAX_FILE_SIZE` [ref »](#)

HTML input dengan nama `MAX_FILE_SIZE`, nilai dalam byte. Elemen harus diletakkan di paling awal.

```
<form method="post" action=""  
enctype="multipart/form-data">  
    <input type="hidden" name="MAX_FILE_SIZE"  
value="102400"/>  
    <input type="submit" name="submit"  
value="Upload"/>  
</form>
```

WARNING: jangan gunakan teknik ini, karena sangat tidak aman, gunakan file PHP

ERROR CODE

Jika terjadi error, maka key 'error' pada `$_FILES['user_file']['error']` akan bernilai > 0

- 1 Ukuran file melebihi `upload_max_filesize` pada `php.ini`
- 2 Ukuran file melebihi `MAX_FILE_SIZE` pada form html
- 3 File tidak selesai diupload
- 4 Tidak ada file yang diupload
- 6 Temporary folder tidak ditemukan
- 7 Tidak dapat menulis file ke hardisk

EXAMPLE

Contoh untuk single file upload

```
1. <form method="post" action="" enctype="multipart/form-data">
2.   <input type="file" name="user_file"/>
3.   <input type="submit" name="submit" value="Upload"/>
4. </form>
5.
6. <?php
7. if (isset($_POST['submit']))
8. {
9.   $error_code = $_FILES['user_file']['error'];
10.  if ($error_code) {
11.    $list_error = array(1 => 'Ukuran file melebihi batas yang ditentukan',
12.                      2 => 'Ukuran file melebihi batas yang ditentukan',
13.                      3 => 'File tidak selesai diupload',
14.                      4 => 'Tidak ada file yang diupload',
15.                      6 => 'Temporary folder tidak ditemukan',
16.                      7 => 'Tidak dapat menulis file ke hardisk'
17.                    );
18.
19.    echo 'ERROR: ' . $list_error[$_FILES['user_file']['error']];
20.  } else {
21.
22.    if (is_uploaded_file($_FILES['user_file']['tmp_name'])) {
23.
24.      // VALIDASI
25.      $size  = $_FILES['user_file']['size'];
26.      $type  = $_FILES['user_file']['type'];
27.      $mime  = array('image/jpeg', 'image/jpg');
28.      $error = false;
29.
30.      if ($size > (1024 * 1024)) { // 1 MB
31.        $error = 'ERROR: Maksimal ukuran file 1 MB';
32.      }
33.      elseif (!in_array($type, $mime)) {
34.        $error = 'ERROR: File harus JPG atau JPEG';
35.      }
36.
37.      // UPLOAD
38.      else {
39.        $tmp_name    = $_FILES['user_file']['tmp_name'];
40.        $new_file = 'files/'. $_FILES['user_file']['name'];
41.        if( move_uploaded_file($tmp_name, $new_file) ){
42.          echo 'File berhasil diupload';
43.        } else {
44.          $error = 'File gagal diupload';
45.        }
46.      }
47.
48.      // JIKA ERROR
49.      if ($error) {
50.        echo $error;
51.      }
52.    } else {
53.      echo 'File gagal diupload';
54.    }
55.  }
56.}
```

IF ... ELSE ref »

If ... elseif ... else

Satu statemen, tanpa kurung kurawal

```
if ($a > $b)
    echo $a . ' lebih besar dari ' . $b;
else
    echo $a . ' kurang dari ' . $b;
```

atau:

```
if ($a > $b) echo $a ' lebih besar';
$b = 10;
```

Menggunakan kurung kurawal

```
if ($a > $b) {
    echo $a . ' lebih besar dari ' . $b;
} elseif ($a == $b) {
    echo $a . ' sama dengan ' . $b;
} else {
    echo $a . ' lebih kecil dari ' . $b;
}
```

Menggunakan colon, gunakan elseif bukan else if

```
if ($a > $b):
    echo $a . ' lebih besar dari ' . $b;
elseif ($a == $b):
    echo $a . ' sama dengan ' . $b;
else:
    echo $a . ' lebih kecil dari ' . $b;
endif;
```

if...else (ternary operator)

```
$a = 9;
$val = $a > 7 ? 'lulus' : 'gagal'; // lulus
```

Mulai PHP 5.3, dapat menggunakan ?:

```
$val = isset($b) ? $b : 'lainnya'; // lainnya
$val = @$b ?: 'lainnya'; // lainnya
```

Keterangan: karena variabel \$b tidak dideklarasikan maka variabel \$val bernilai 'lainnya'

Mix: PHP dan HTML

```
<?php if ($a > $b) {?>
    <p> $a lebih besar dari $b </p>
<?php } else {?>
    <p> $a lebih kecil dari $b </p>
<?php } ?>
```

Dengan colon:

```
<?php if ($a > $b) :?>
    <p> $a lebih besar dari $b </p>
<?php else :?>
    <p> $a lebih kecil dari $b </p>
<?php endif ?>
```

Tes apakah variabel ada nilainya (kondisi: variabel sudah dideklarasikan)

```
$a = 5; $b = '';
if ($b) {
    echo $a * $b;
} else {
    echo $a * 3;
} // 15
```

Dengan operator !

```
$a = 5; $b = '';
if (! $b) {
    $b = 3;

echo $a * $b; // 15
```

Tes menggunakan operator perbandingan

```
if ($a != $b) { ... } // $a tidak sama $b
if ($a <> $b) { ... } // $a tidak sama $b
if ($a == $b) { ... } // $a sama dengan $b
if ($a === $b) { ... } // $a sama dengan $b
baik nilai maupun tipe datanya
if ($a >= $b) { ... } // $a lebih besar atau sama dengan $b
if ($a <= $b) { ... } // $a lebih kecil atau sama dengan $b
```

SWITCH ... CASE ref »

switch...case

Gunakan switch ... case jika yang diuji hanya satu nilai, contoh menguji nilai variabel \$a. Switch case lebih mudah dipahami dan lebih cepat dieksekusi daripada IF ELSE

Menguji variabel \$a dengan IF ELSE

```
if ($a == 'A') {
    echo 'Baik Sekali';
} elseif ($a == 'B') {
    echo 'Baik';
} elseif ($a == 'C') {
    echo 'Cukup';
} else {
    echo 'Kurang';
}
```

Dengan switch ... case

```
switch ($a) {
    case 'A':
        echo 'Baik Sekali'; break;
    case 'B':
        echo 'Baik'; break;
    case 'C':
        echo 'Cukup'; break;
    default:
        echo 'Kurang';
}
```

IF ELSE dengan OR

```
if ($a == 'A' || $a == 'B') {
    echo 'Lulus';
} elseif ($a == 'C') {
    echo 'Mengulang';
} else {
    echo 'Tidak lulus';
}
```

Dengan switch case

```
switch ($a) {
    case 'A':
    case 'B':
        echo 'Lulus';
        break;
    case 'C':
        echo 'Mengulang';
        break;
    default:
        echo 'Tidak lulus';
}
```

if ... else di dalam switch ... case

```
switch ($a) {
    case 'A':
    case 'B':
        if ($b >= 8) {
            echo 'Lulus';
        } else {
            echo 'Lulus bersyarat';
        }
        break;
    case 'C':
        echo 'Mengulang'; break;
    default:
        echo 'Tidak lulus';
}
```

FOR ref » | WHILE ref » | FOREACH ref »

for () {...}

```
for ($i = 1; $i <= 10; $i++) {
    echo $i;
} // 12345678910
```

Break digunakan untuk menghentikan loop

```
for ($i = 1; $i <= 10; $i++) {
    echo $i;
    if ($i == 5)
        break;
} // 12345678910
```

Continue digunakan untuk melanjutkan loop

```
for ($i = 1; $i <= 10; $i++) {
    echo $i;
    if ($i == 5) continue;
} // 1234678910 -> tanpa 5
```

while () {...}

```
$i = 1;
while ($i <= 10) {
    echo $i;
    $i++;
} // 12345678910
```

Break digunakan untuk menghentikan loop

```
$i = 1;
while ($i <= 10):
    echo $i;
    if ($i == 5)
        break;
    $i++;
endwhile; // 12345
```

do () while{...}

```
do {
    statement; // dieksekusi terlebih dahulu
} while (ekspresi);
```

Pada do while, setidaknya statement akan di eksekusi sekali, selanjutnya di dilakukan pengecekan pada ekspresi

```
$i = 1;
do {
    echo $i;
} while ($i < 0); // 1
```

Keterangan: pertama \$i dicetak, selanjutnya karena ekspresi tidak terpenuhi, loop dihentikan.

foreach () {...}

foreach digunakan pada array

```
$arr = array ('foo', 'bar');
foreach ($arr as $val) {
    echo $val . ', ';
} // foo, bar,
```

Associative array

```
$arr = array ('kendaraan' => 'mobil',
              'warna' => 'merah',
              'tipe' => 'mpv'
            );
foreach ($arr as $item => $deskripsi) {
    echo $item . ': ' . $deskripsi;
}
// kendaraan: mobil warna: merah tipe: mpv
```

Gunakan continue untuk melewati loop:

```
foreach ($arr as $item => $deskripsi) {
    if ($item == 'warna')
        continue;
    echo $item . ': ' . $deskripsi;
}
// kendaraan: mobil tipe: mpv
```

PHP dan MySQL [ref »](#)

`mysqli_connect(server, user, pass, db_name)`

Menghubungkan php dengan mysql:

```
$conn = @mysqli_connect('localhost', 'root', '',
                        'db_name');
if (!$conn) {
    die ('MySQL Error:' . mysqli_connect_error());
}
```

`mysqli_query (sql)`

Fungsi untuk menjalankan query MySQL. Fungsi ini harus dijalankan setiap kali menjalankan query, baik SELECT, INSERT, UPDATE maupun DELETE

```
$result = mysqli_query('SELECT * FROM tabel');
```

PENTING: jika data tidak ditemukan, variabel \$result akan bernilai false

`mysqli_fetch_array (result)`

Argumen result merupakan hasil query, salah satunya hasil dari fungsi mysqli_query

```
while ($row = mysqli_fetch_array($result)) {
    echo $row[col_name];
}
```

`mysqli_fetch_all (result, resulttype)`

Fungsi akan mengubah hasil query menjadi bentuk array. nilai argumen resulttype: MYSQLI_ASSOC, MYSQLI_NUM, dan MYSQLI_BOTH

```
mysqli_fetch_all ($result, MYSQLI_ASSOC);
```

WARNING: pemanggilan mysqli_fetch_xxx lebih dari sekali akan menghasilkan data kosong

`mysqli_affected_rows ($conn)`

Digunakan untuk mendapatkan jumlah baris yang terpengaruh pada query sebelumnya. Dapat digunakan untuk statemen SELECT, INSERT, UPDATE, dan DELETE

```
$result = mysqli_query($conn, 'SELECT ...');
$affected_rows = mysqli_affected_rows($conn);
```

WARNING: tes dengan `if($affected_rows > 0)` bukan `if($affected_rows)` karena ketika gagal insert atau update data, hasilnya -1 bukan 0

`mysqli_num_rows (result)`

Mendapatkan banyaknya baris hasil query

```
$result = mysqli_query($conn, 'SELECT ...');
$row_count = mysqli_num_rows($result);
```

`mysqli_free_result (result)`

Menghapus hasil query dari memory (RAM), tidak wajib digunakan, gunakan hanya jika hasil query sangat besar. PHP akan otomatis menghapus hasil query pada akhir eksekusi script

```
$result = mysqli_query($conn, 'SELECT ...');
mysqli_free_result($result);
```

`mysqli_close ($conn)`

Menutup koneksi, tidak wajib digunakan. PHP otomatis menutup semua koneksi pada akhir eksekusi script

```
mysqli_close ($conn);
```

`mysqli_insert_id ($conn)`

Mendapatkan nilai terakhir dari auto_increment

```
mysqli_query($conn, 'INSERT INTO ... ');
$last_id = mysqli_insert_id($conn);
```

`mysqli_escape_string ($conn, string)`

Sama dengan `mysqli_real_escape_string (string)` digunakan untuk men-escape tanda kutip

```
$str = "Hari Jum'at";
$str = mysqli_escape_string ($conn, $str);
mysqli_query($conn, 'INSERT INTO tabel VALUES
(".$str.")');
```

`mysqli_connect_error ($conn)`

Untuk mendapatkan pesan error jika koneksi gagal

`mysqli_error ($conn)`

Untuk mendapatkan pesan error pada query terakhir

Transaction [ref »](#)

`mysqli_begin_transaction ($conn)`

Memulai transaksi:

```
mysqli_begin_transaction($conn);
```

`mysqli_commit ($conn) | mysqli_rollback ($conn)`

Commit untuk menerapkan perubahan secara permanen, Rollback untuk membatalkan perubahan

`mysqli_savepoint ($conn, string)`

Membuat savepoint pada transaksi

```
mysqli_savepoint($conn, 'perubahan_1');
```

EXAMPLES

Example 1: SELECT

Contoh script untuk mengambil data

```
$conn = @mysqli_connect('localhost', 'root', '',  
                        'toko_buku');  
  
if (!$conn) {  
    die ('MySQL ERROR: ' .  
         mysqli_connect_error());  
}  
  
$result = mysqli_query($conn, 'SELECT * FROM  
buku');  
  
if ($result) {  
  
    // Jumlah Data  
    $row_count = mysqli_num_rows($result);  
    echo 'Jumlah data ' . $row_count;  
  
    // Fetch rows  
    while ($row = mysqli_fetch_array($result)) {  
        echo $row['judul'];  
    }  
  
    // Fetch all  
    $all_data = mysqli_fetch_all($result,  
                                 MYSQLI_ASSOC);  
    echo '<pre>';  
    print_r($all_data);  
    echo '</pre>';  
}  
mysqli_free_result($result);  
mysqli_close($conn);
```

EXAMPLE 2: UPDATE

Pada contoh berikut, meskipun query berhasil, banyaknya data yang diupdate (`mysqli_affected_rows`) adalah 0, karena buku dengan id 1 sudah memiliki judul PHP

```
$conn = @mysqli_connect('localhost', 'root', '',  
                        'toko_buku');  
if (!$conn) {  
    die ('MySQL ERROR: ' . mysqli_connect_error());  
}  
  
$result = mysqli_query($conn, 'UPDATE buku SET  
judul="PHP" WHERE id_buku = 1');  
  
if ($query) {  
    echo 'Data berhasil diupdate sebanyak ' .  
         mysqli_affected_rows($conn) . ' baris';  
} else {  
    echo 'Data gagal diupdate';  
}
```

EXAMPLE 3: TRANSACTION

Contoh transaksi dengan PHP

```
$conn = @mysqli_connect('localhost', 'root', '',  
                        'toko_buku');  
  
if (!$conn) {  
    die ('MySQL ERROR: ' . mysqli_connect_error());  
}  
  
// Mulai Transaksi  
mysqli_begin_transaction($conn);  
  
// Query 1  
$error = '';  
$result = mysqli_query($conn, 'INSERT INTO buku  
(judul) VALUES ("PHP")');  
if (!$result) {  
    $error .= 'Query 1: ' . mysqli_error($conn) . ', ';  
}  
  
// Query 2  
$result = mysqli_query($conn, 'INSERT INTO buku  
(judul) VALUES ("MySQL")');  
if (!$result) {  
    $error .= 'Query 2: ' . mysqli_error($conn) . ', ';  
}  
  
// Commit - Rollback  
if ($error) {  
    mysqli_rollback($conn);  
    echo 'MySQL ERROR: ' . rtrim($error, ', ');  
} else {  
    mysqli_commit($conn);  
    echo 'Data berhasil dimasukkan';  
}
```

EXAMPLE 2: INSERT

Contoh script untuk memasukkan data

```
$conn = @mysqli_connect('localhost', 'root', '',  
                        'toko_buku');  
  
if (!$conn) {  
    die ('MySQL ERROR: ' . mysqli_connect_error());  
}  
  
$val = 'slash"s';  
$val = mysqli_escape_string($conn, $val);  
$result = mysqli_query($conn, 'INSERT INTO buku  
(judul) VALUES ("' . $val . "')');  
  
if ($result) {  
    echo 'Data berhasil dimasukkan sebanyak' .  
         mysqli_affected_rows($conn) . ' baris,'.  
         ' id_buku: ' . mysqli_insert_id($conn);  
} else {  
    echo 'Data gagal dimasukkan';  
}
```

String ref >

```
str_replace (search, replace, str)
$str = str_replace('merah', 'biru', 'mobil
merah'); // mobil biru

Multiple string
$arr = array('merah', 'biru', 'kuning');
$str = str_replace($arr, 'hitam', 'motor merah
kuning'); // motor hitam hitam
```

strtoupper (str) | strtolower (str)

```
$str = strtoupper('mobil'); // MOBIL
$str = strtolower('MOBIL'); // mobil
```

ucfirst(str) | ucwords(str)

ucfirst: huruf pertama menjadi kapital, **ucwords:** huruf pertama pada setiap kata menjadi kapital

```
$str = ucfirst('mobil merah'); // Mobil merah
$str = ucwords ('mobil merah'); // Mobil Merah
```

trim(str) | ltrim(str) | rtrim(str)

trim menghilangkan spasi sebelum dan sesudah string, **ltrim** (left trim) sebelum string, dan **rtrim** (right trim) sesudah string. **Opsional:** argumen kedua dapat diisi karakter yang ingin dihilangkan

```
$str = trim(' mobil '); // 'mobil'
$str = ltrim(' mobil '); // 'mobil '
$str = rtrim(' mobil '); // ' mobil'
$str = rtrim('mobil+', '+'); // 'mobil'
```

strpos (str, needle) | stripos (str, needle)

strpos dan stripos digunakan untuk mendapatkan posisi karakter. strpos (case sensitive), stripos (case insensitive)

```
$pos = strpos('mobil', 'o'); // 1
$str = 'mobil';
if (strpos($str, 'mob') !== false) {
    echo 'exists'
} // 'exists'
```

substr (str, start, length)

Mengambil bagian string

```
$str = substr('mobil', 0, 3); // mob
$str = substr('mobil', -3); // il
$str = substr('mobil', 3); // il
```

strlen (str)

Menghitung banyaknya karakter

```
$str = strlen('mobil'); // 5
```

strstr (str, needle, before_string = false)

String pertama yang ditemukan

```
$str = strstr('nama@mail.com','@'); // @mail.com
$str = strstr('nama@mail.com','@', true); // name
```

nl2br (str)

New line to break (
)

```
$str = nl2br ("foo\r\nbar"); // foo<br />bar
```

explode (sep, str) | implode (sep, str)

explode: memecah string menjadi array

```
$str = explode (',', 'foo, bar');
// Array ([0]=>foo [1]=>bar)
```

implode: menggabung array menjadi string

```
$arr = array('foo', 'bar')
$str = implode (' ', $arr); // foo, bar
```

. (dot)

Digunakan untuk menggabungkan string

```
$str = 'Mobil';
$str = $str . ' merah'; // Mobil merah
$str .= 'baru' // Mobil merah baru
```

'' | ''"

Dua tanda kutip dapat digunakan untuk menyimpan variabel

```
$var = 'foo';
$str = $var . ' bar'; // foo bar
$str = "$var bar"; // foo bar
```

echo | print_r | var_dump

echo hanya untuk mencetak string. print_r dan var_dump untuk mencetak array dan object. var_dump menampilkan tipe data dan panjang data

```
echo 'foo bar';
$arr = array('foo', 'bar');
echo '<pre>'; print_r($arr); // Array ( [0] =>
foo [1] => bar
echo '<pre>'; var_dump($arr); // array(2) {
[0]=> string(3) "foo" [1]=> string(3) "bar" }
```

is_string (var)

Menguji apakah variabel berjenis string

```
if (is_string($var)) {
    echo 'string';
}
```

Variable ref >

Penulisan

Contoh penulisan valid

```
$variable = 1;
$_variable = 'string';
$variable2 = array();
```

Keterangan:

- Diawali \$ dan harus diikuti sebuah huruf atau underscore, sisanya: huruf, underscore, atau angka
- Case sensitive: \$var tidak sama dengan \$Var
- Tidak boleh menggunakan nama \$this

Built in variables

`$_GET, $_POST, $_FILES, $_SESSION, $_COOKIE, dan
$_SERVER`

`$_GET, $_POST, dan $_FILES` untuk memperoleh data form yang disubmit, `$_SESSION` untuk memperoleh data sesi yang dimulai dengan `start_session()`, `$_COOKIE` untuk memperoleh data cookie, `$_SERVER` untuk memperoleh data terkait server

Memberi Nilai

```
$value = 10;
$value = $value + 5;
echo $value; // 15;
```

Pertama tanda ekspresi sebelah kanan tanda = akan dievaluasi, hasilnya disimpan ke variabel di sebelah kiri.

Scope: GLOBAL & LOCAL

```
$nilai = 10; // Global
function angka() {
    echo $nilai; // Undefined
    $nilai = 5; // Lokal
    echo $nilai;
}
angka(); // 5
echo $nilai; // 10
```

Global: di luar fungsi, Lokal: di dalam fungsi. Secara default variabel global tidak dapat digunakan di dalam fungsi

Variable Variables

```
$nilai = 10;
$variabel = 'nilai';
echo $$variabel; // 10
```

Mengakses \$\$variabel sama dengan mengakses \$nilai.

Gunakan tanda kurung kurawal untuk membuat variabel dari gabungan string dan variabel

```
$nama = 'alfa';
${'nilai_'. $nama} = 10;
echo $nilai_alfa; //10;
```

Contoh lain:

```
$list = array("nama" => "Alfa", "nilai" => 10);
foreach ($list as $key => $val) {
    ${'input_'} . $key} = $val;
}
echo $input_nama; // Alfa
echo $input_nilai; // 10
```

Constant ref >

Membuat Konstanta

Dua cara mendefinisikan konstanta, fungsi atau statemen

```
define('BASE_URL', 'jagowebdev.com'); // Fungsi
const DB_NAME = 'toko_buku'; // Statemen
echo BASE_URL; // jagowebdev.com
```

Sifat konstanta:

- Nilai constant tidak dapat diubah, jika constant didefinisikan dua kali, akan muncul pesan error
- Nama bersifat case sensitive
- Fungsi define tidak dapat digunakan di dalam Class
- Sebelum PHP versi 5.3 statemen const hanya dapat digunakan pada Class

Penulisan:

- Harus diawali huruf atau underscore (_), selanjutnya hanya boleh: huruf, angka, dan underscore
- Konstanta hanya bisa diisi nilai scalar (integer, float, dll) dan string, mulai PHP 7 dapat diisi array
- Tips: untuk memberi tanda konstanta, gunakan huruf kapital semua

Membuat Konstanta dengan array (PHP 7)

Konstanta dengan array:

```
define ('DB_CONFIG', [
    'host' => 'localhost',
    'port' => '3306',
    'db_name' => 'jagowebdev',
    'user' => 'root',
    'pass' => ''
])
echo DB_CONFIG['host'];
```

Scope: GLOBAL

Konstanta bersifat global, dapat digunakan di luar maupun di dalam fungsi

```
const DB_NAME = 'toko_buku'; // PHP >= 5.3
function connect() {
    echo DB_NAME;
}
echo connect(); // toko_buku
```

Mendefinisikan Dengan Aman

Untuk menghindari error karena konstanta didefinisikan lebih dari sekali,, tes nama konstanta dengan IF

```
if (!defined('DB_NAME')) {
    const DB_NAME = 'toko_buku';
}
```

PRE DEFINED CONSTANT

PHP menyediakan beberapa konstanta yang dapat langsung digunakan, ciri dari konstanta bawaan PHP adalah diawali dan diakhiri dengan dua underscore (_)

<u>_LINE_</u>	Mendapatkan baris kode <code>echo __LINE__; // 1</code>
<u>_FILE_</u>	Full path dari file <code>E:\xampp\htdocs\tutorial\file.php</code>
<u>_DIR_</u>	Full path dari directory <code>E:\xampp\htdocs\tutorial\</code>
<u>_FUNCTION_</u>	Mendapatkan nama fungsi <code>function tes_fungsi() { echo __FUNCTION__; } echo tes_fungsi(); // tes_fungsi</code>
<u>_CLASS_</u>	Mendapatkan nama CLASS <code>class myClass { function getClassName() { echo __CLASS__; } } \$obj = new myClass; echo \$obj->getClassName(); // myClass</code>

EXAMPLE: PHP 7

File config.php

```
define ('SITE_NAME', 'jagowebdev');
define ('DB_CONFIG', ['host' => 'localhost',
    'port' => '3306',
    'db_name' => 'jagowebdev',
    'user' => 'root',
    'pass' => ''
]);

```

File koneksi.php

```
require_once 'config.php';
$conn = mysqli_connect(CONFIG_DB['host'],
    CONFIG_DB['user'],
    CONFIG_DB['pass'],
    CONFIG_DB['db_name']
);
if (!$conn) {
    die('ERROR: ' . mysqli_connect_error());
}
```

EXAMPLE: PHP ALL VERSION

File config.php

```
define ('DB_HOST', 'localhost');
define ('DB_USER', 'root');
define ('DB_PASS', '');
define ('DB_NAME', 'jagowebdev');
```

File koneksi.php

```
require_once 'config.php';
$conn = mysqli_connect(DB_HOST, DB_ROOT,
DB_PASS, DB_NAME);
if (!$conn) {
    die('ERROR: ' . mysqli_connect_error());
}
```