

MODUL 8
PHP BAGIAN III
(OPERASI BILANGAN, STRUKTUR KONDISI DAN PERULANGAN,
ARRAY DAN FUNGSI)

1. OPERASI BILANGAN

Operator dalam PHP

Jenis Operator	Operator	Contoh	Keterangan
Aritmatika	+	\$a + \$b	Pertambahan
	-	\$a - \$b	Pengurangan
	*	\$a * \$b	Perkalian
	/	\$a / \$b	Pembagian
	%	\$a % \$b	Modulus, sisa pembagian
Penugasan	=	\$a = 4;	\$a diisi dengan 4
Bitwise	&	\$a & \$b	Bitwise AND
		\$a \$b	Bitwise OR
	^	\$a ^ \$b	Bitwise XOR
	~	~\$b	Bitwise NOT
	<<	\$a << \$b	Shift Left
	>>	\$a >> \$b	Shift Right
Perbandingan	==	\$a == \$b	Sama dengan
	===	\$a === \$b	Identik
	!=	\$a != \$b	Tidak sama dengan
	<>	\$a <> \$b	Tidak sama dengan
	!==	\$a !== \$b	Tidak identik
	<	\$a < \$b	Kurang dari
	>	\$a > \$b	Lebih dari
	<=	\$a <= \$b	Kurang dari sama dengan
	>=	\$a >= \$b	Lebih dari sama dengan
Logika	and	\$a and \$b	TRUE jika \$a dan \$b TRUE
	&&	\$a && \$b	TRUE jika \$a dan \$b TRUE
	or	\$a or \$b	TRUE jika \$a atau \$b TRUE
		\$a \$b	TRUE jika \$a dan/atau \$b TRUE
	xor	\$a xor \$b	TRUE jika \$a atau \$b TRUE, tapi tidak keduanya
	!	!\$a	TRUE jika \$a FALSE
String	.	\$a . \$b	Penggabungan string \$a dan \$b

Program 8.1

Nama File : operator.php

Deskripsi : Program beberapa Operator aritmatika dalam PHP.

```
<?
$gaji = 1000000;
$pajak = 0.1;
$thp = $gaji - ($gaji * $pajak);
echo "Gaji sebelum pajak = Rp. $gaji <br>"; echo "Gaji yang dibawa pulang = Rp.
$thp";
?>
```

Program 8.2

Nama File : operator2.php

Deskripsi : Program operator logika dan perbandingan dalam PHP.

```
<?
$a = 5;
$b = 4;

echo "$a == $b : ". ($a == $b);
echo "<br>$a != $b : ". ($a != $b); echo "<br>$a > $b :
". ($a > $b); echo "<br>$a < $b : ". ($a < $b);
echo "<br>($a == $b) && ($a > $b) : ". (($a != $b) && ($a > $b));
echo "<br>($a == $b) || ($a > $b) : ". (($a != $b) || ($a > $b));
?>
```

2. STRUKTUR KONDISI DAN PERULANGAN

Struktur Kondisi If

```
if (kondisi) {
    statement-jika-kondisi-true;
}
```

Keterangan :

kondisi merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya.

Struktur Kondisi If ... Else

```
if (kondisi) {
    statement-jika-kondisi-true;
} else {
    statement-jika-kondisi-false;
}
```

Keterangan :

kondisi merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya.

Jika kondisinya TRUE maka statemen yang berada di blok if akan dieksekusi, sebaliknya jika kondisinya FALSE maka statemen yang berada di blok else yang akan dieksekusi.

Struktur Kondisi Khusus ? :

```
(kondisi) ? benar : salah;
```

Keterangan :

kondisi merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya.

Statement pada blok benar dan salah hanya boleh satu statemen saja

Struktur Kondisi Switch ... Case

```
switch ($var) {
    case '1' : statement-1; break; case '2' : statement-2; break;
    ....
}
```

Keterangan :

\$var merupakan variabel yang akan diperiksa isi atau nilainya. Tipe data variabel ini tidak dibatasi. Value pada **case** juga bisa berupa string, integer, boolean, bahkan bisa berupa conditional-statement. Boleh memakai kutip tunggal maupun kutip ganda.

Program 8.3

Nama File : *if.php*

Deskripsi : *Program Struktur Kondisi If.*

```
<?php
$nilai = 80;
if ($nilai >= 60) {
    echo "Nilai Anda $nilai, AndaLULUS";
}
?>
```

Penjelasan Program 8.3

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** lebih dari 60 (baris 3). Jika kondisi ini bernilai **TRUE** (variabel \$nilai berisi nilai yang lebih besar dari 60) maka *statement* pada baris ke-4 akan dijalankan. Sebaliknya jika kondisinya **FALSE**, maka *statement* pada baris ke-4 tidak akan dijalankan.

Program 8.4

Nama File : *if_else.php*

Deskripsi : *Program Struktur Kondisi If..Else.*

```
<?php
$nilai = 50;
if ($nilai >= 60) {
    echo "Nilai Anda $nilai, AndaLULUS";
} else {
    echo "Nilai Anda $nilai, AndaGAGAL";
}
?>
```

Penjelasan Program 8.4

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** lebih dari 60 (baris 3). Jika kondisi ini bernilai **TRUE** (variabel \$nilai berisi nilai yang lebih besar dari 60) maka *statement* pada baris ke-4 akan dijalankan. Sebaliknya jika kondisinya **FALSE**, maka *statement* pada baris ke-6 yang akan dijalankan. Pada contoh program di atas, kondisi pada baris ke-3 bernilai **FALSE** karena isi variabel \$nilai adalah 50.

Program 8.5

Nama File : *if_else2.php*

Deskripsi : Program Memeriksa username dan password dengan If..Else.

```
<?php
$user = "achmatim";
$pass = "123";
if ($user == "achmatim" && $pass == "123") { echo "Login Berhasil";
} else {
echo "Login Gagal";
}
?>
```

Penjelasan Program 8.5

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** sama dengan “achmatim” **dan** apakah isi variabel **\$pass** sama dengan “123” (baris 4). Jika kedua kondisi tersebut bernilai **TRUE** maka statement pada baris ke-5 akan dijalankan. Sebaliknya jika salah satunya bernilai **FALSE**, maka statement pada baris ke-7 yang akan dijalankan.

Program 8.6

Nama File : *if_var.php*

Deskripsi : Program Memeriksa suatu variabel ada atau tidak.

```
<?php
$user="";
if (!isset($user)) {
    echo "Variabel tidak ada/belum terbentuk";
} else {
    echo "Variabel ada";
}
?>
```

Penjelasan Program 8.6

Pada program di atas, terdapat fungsi **isset()** yang merupakan fungsi untuk memeriksa apakah suatu variabel sudah terbentuk (ada) atau belum. Fungsi akan bernilai **TRUE** jika variabelnya ada. Jadi jika baris ke-2 dari program di atas di- *remark* atau dihilangkan maka *statement* pada baris ke-4 yang akan dijalankan.

Program 8.7

Nama File : *switch.php*

Deskripsi : Program Struktur Switch..Case untuk menampilkan nama hari dalam bahasa Indonesia.

```
<?php
$day = date ("D");
switch ($day) {
case 'Sun' : $hari           = "Minggu"; break;
case 'Mon' : $hari           = "Senin"; break;
case 'Tue' : $hari           = "Selasa"; break;
case 'Wed' : $hari           = "Rabu"; break;
case 'Thu' : $hari           = "Kamis"; break;
case 'Fri' : $hari           = "Jum'at"; break;
case 'Sat' : $hari           = "Sabtu"; break;
}
echo "Hari ini hari <b>$hari</b>";
?>
```

Penjelasan Program 8.7

Pada program di atas, variabel **\$day** (baris ke-2) akan berisi 3 (tiga) digit pertama dari nama hari dalam bahasa Inggris. Dengan fungsi **date()**, kita akan memperoleh informasi tanggal, hari dan waktu sekarang. Selanjutnya isi variabel **\$day** akan diperiksa dengan switch (baris ke-3), jika isinya adalah ‘Sun’ maka **\$hari** adalah “Minggu” dan seterusnya.

Program 8.8

Nama File : *if_khusus.php*

Deskripsi : Program Struktur Kondisi Khusus ? untuk memeriksa tahun kabisat.

```
<?php
$tahun = date ("Y");
$skabisat = ($tahun%4 == 0) ? "KABISAT" : "BUKANKABISAT";
echo "Tahun <b>$tahun</b> $skabisat";
?>
```

Penjelasan Program 8.8

Pada program di atas, fungsi **date()** dengan parameter “Y” (baris ke-2) akan menghasilkan 4 digit tahun sekarang. Pada baris ke-3 akan diperiksa apakah isi variabel \$tahun jika dimodulus dengan 4 akan menghasilkan nilai 0. Jika **TRUE** maka \$skabisat akan berisi “KABISAT” dan jika **FALSE** \$skabisat akan berisi “BUKAN KABISAT”

Struktur Perulangan

Struktur Perulangan For

```
for (init_awal, kondisi, counter)
{ statement-yang-diulang;
}
```

Keterangan :

- a) **init_awal** merupakan inisialisasi atau nilai awal variable.
- b) **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- c) **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.

Struktur Perulangan While

```
init awal;  
while (kondisi)  
{ statement-yang-diulang;  
  counter;  
}
```

Keterangan :

- a) **init_awal** merupakan inisialisasi atau nilai awal variable.
- b) **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- c) **counter** merupakan pertambahan atau pengurangan nilai variabel
- d) sehingga perulangan tetap berjalan.

Struktur Perulangan Do ... while

```
init_awal; do  
{  
  statement-yang-diulang;  
  counter;  
} while (kondisi);
```

Keterangan :

- a) **init_awal** merupakan inisialisasi atau nilai awal variable.
- b) **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- c) **counter** merupakan pertambahan atau pengurangan nilai variabel
- d) sehingga perulangan tetap berjalan.
- e) Pada struktur do...while, pemeriksaan kondisi ada di bawah, sehingga
- f) statement yang berada dalam block do...while setidaknya akan dieksekusi sebanyak satu kali.

Struktur Perulangan Foreach

```
foreach (array_expression as $value) statement;  
foreach (array_expression as $key => $value)  
  statement;
```

Keterangan :

- a) Struktur foreach biasanya digunakan untuk melakukan perulangan berdasarkan isi suatu array. Perulangan akan berakhir jika isi array telah habis.

Program 8.9

Nama File : *for.php*

Deskripsi : *Program Struktur Perulangan For dan beberapa variasinya.*

```
<?php
// contoh 1
for ($i = 1; $i <= 10; $i++) { echo "$i ";
}
echo "<br><br>";
```

```
// contoh 2
for ($i = 1; ; $i++) { if ($i > 10) {
break;
}
echo "$i ";
}
echo "<br><br>";
```

```
// contoh 3
$i = 1;
for (; ; ) {
if ($i > 10) { break;
}
echo "$i ";
$i++;
} echo "<br><br>";
```

```
// contoh 4
for ($i = 1; $i <= 10; print "$i ", $i++);
?>
```

Penjelasan Program 8.9

Program di atas merupakan bentuk-bentuk perulangan dengan menggunakan for. Contoh yang pertama (baris 4-6) merupakan bentuk yang paling umum. Pada contoh 2, batas akhir perulangan tidak disebutkan dalam for, tapi diatur dengan if dan **break** (baris 11-13).

Program 8.10

Nama File : *while.php*

Deskripsi : *Program Struktur Perulangan While dan beberapa variasinya.*

```
<?php
// contoh 1 /
$i = 1;
```

```
while ($i <= 10) { echo $i++;  
}  
echo "<br><br>";  
  
// contoh 2 /  
$i = 1;  
while ($i <= 10): echo "$i";  
    $i++; endwhile;  
echo "<br><br>";  
  
// contoh 3 /  
$i = 1;  
while ($i <= 6) {  
    echo "<h$i>Heading $i</h$i>";  
    $i++;  
}  
?>
```

Penjelasan Program 8.10

Program di atas merupakan beberapa variasi perulangan dengan while. Contoh yang pertama (baris 3-6) merupakan bentuk yang paling umum dipakai. Blok perintah (baris 5) akan dijalankan selama pemeriksaan kondisi pada baris ke-4 bernilai **TRUE**. Contoh kedua merupakan bentuk lain dari while, bentuk ini jarang digunakan. Pada contoh ketiga, penggunaan perulangan untuk menampilkan tulisan dengan format <H1> sampai <H6>.

Program 8.11

Nama File : *dowhile.php*

Deskripsi : *Program Struktur Perulangan dengan Do...While.*

```
<?php  
$i = 1; do {  
    echo "$i ";  
    $i+=2;  
} while ($i <= 20);  
?>
```

Penjelasan Program 8.11

Program di atas akan menampilkan bilangan ganjil antara 1 sampai 20 menggunakan struktur perulangan **do..while**. Pada struktur perulangan do...while, pemeriksaan kondisi berada di bawah.

Struktur Break dan Continue

Program 8.12

Nama File : *break.php*

Deskripsi : *Program Struktur Break dan Continue.*


```
<?php
for ($i=1; $i<10; $i++) { if ($i == 5)
    continue; if ($i == 8)
    break; echo "$i ";
}
?>
```

Penjelasan Program 8.12

Dari program di atas, dapat disimpulkan bahwa perintah **continue** akan melanjutkan proses perulangan tanpa melewati (meng-eksekusi) baris perintah di bawahnya. Jadi perintah pada baris 6-8 akan dilewati (5 tidak akan tercetak) Sedangkan dengan perintah **break** akan menyebabkan program menghentikan perulangan (langsung keluar dari perulangan)

3. ARRAY DAN FUNGSI

Pengantar Array

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Bagian yang menyusun array disebut elemen array, yang masing-masing elemen dapat diakses tersendiri melalui index array. Index array dapat berupa bilangan integer atau string.

Mendeklarasikan dan Menampilkan Array

Program 8.13

Nama File : array01.php
Deskripsi : Program mendeklarasikan array dengan index numeric.

```
<?php
$sarrBuah = array ("Mangga", "Apel", "Pisang", "Jeruk");
echo $sarrBuah[0]; //Mangga
echo $sarrBuah[3]; //Jeruk

$sarrWarna = array();
$sarrWarna[] = "Merah";
$sarrWarna[] = "Biru";
$sarrWarna[] = "Hijau";
$sarrWarna[] = "Putih";
echo $sarrWarna[0]; //Merah
echo $sarrWarna[2]; //Hijau
?>
```

Penjelasan Program 8.13

Untuk mendeklarasikan atau mendefinisikan sebuah array di PHP bisa menggunakan *keyword* **array()**. Pada program di atas, terdapat pendeklarasian dan pendefinisian array pada baris ke-3. Jumlah elemen array tidak perlu disebutkan saat deklarasi. Sedangkan untuk menampilkan isi array pada elemen tertentu, cukup dengan menyebutkan nama array beserta index array-nya (lihat baris ke-4 dan ke-5). Pendeklarasian dan pengisian array juga bisa dilakukan seperti pada baris ke-7 sampai baris ke-11. Dengan pendefinisian array seperti pada dua cara di atas, index (key) array secara otomatis akan diisi dengan bilangan bulat integer yang dimulai dari 0.

Program 8.14

Nama File : *array02.php*

Deskripsi : Program mendeklarasikan array dengan index string (array assosiatif).

```
<?php
$arrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75, "Budi" => 85);
echo $arrNilai['Ani'];
echo $arrNilai['Otim'];

$arrNilai = array();
$arrNilai['Ami'] = 80;
$arrNilai['Asma'] = 95;
$arrNilai['Sri'] = 77;
echo $arrNilai['Asma'];
echo $arrNilai['Ami'];
?>
```

Penjelasan Program 8.14

Untuk mendeklarasikan atau mendefinisikan sebuah array *assosiatif* (array yang menggunakan index selain integer), dapat dengan cara menyebutkan index-nya terlebih dahulu diikuti operator **=>** dan diikuti value atau nilai elemennya. Lihat program baris ke-3! Sedangkan untuk menampilkan value atau nilai dari suatu array, bisa dengan menyebutkan nama array yang diikuti index arraynya (lihat baris ke-4 dan ke-5). Pendefinisian array *assosiatif* juga bisa dilakukan seperti pada baris ke-7 sampai baris ke-10

Program 8.15

Nama File : *array03.php*

Deskripsi : Program menampilkan seluruh isi array dengan index numerik.

```
<?php
$arrWarna = array("Blue", "Black", "Red", "Yellow", "Green");

echo "Menampilkan isi array dengan for :<br>"; for ($i=0;
```

```
$i<count($arrWarna); $i++){  
    echo "Do you like <font color=$arrWarna[$i]>". $arrWarna[$i]  
    . "</font> ?<br>";  
}  
  
echo "<br>Menampilkan isi array dengan foreach :<br>"; foreach ($arrWarna as  
$warna) {  
    echo "Do you like <font color=$warna>". $warna."</font>  
    ?<br>";  
}  
?>
```

Penjelasan Program 8.15

Untuk menampilkan keseluruhan isi array bisa dilakukan dengan perulangan for, while, do..while atau foreach. Baris 5 – 7 program di atas merupakan contoh menampilkan array dengan perulangan for. Fungsi **count()** pada baris ke-5 berfungsi untuk mencari jumlah elemen dari array \$arrWarna. Untuk menampilkan array juga bisa menggunakan struktur **foreach()**. Struktur foreach akan mengambil elemen array mulai dari posisi pointer sampai akhir pointer. Pada contoh program di atas (baris ke-10) bisa dinyatakan dengan pernyataan bahwa “untuk setiap elemen array \$arrWarna, masukkan ke variabel \$warna”.

Program 8.16

Nama File : array04.php

Deskripsi : Program menampilkan seluruh isi array asosiatif.

```
<?php  
$arrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75, "Budi" => 85);  
  
echo "Menampilkan isi array dengan foreach : <br>";  
foreach ($arrNilai as $nama => $nilai) {  
    echo "Nilai $nama = $nilai<br>";  
}  
reset ($arrNilai);  
echo "<br>Menampilkan isi array dengan while dan list : <br>";  
while (list ($nama, $nilai) = each($arrNilai)) {  
    echo "Nilai $nama = $nilai<br>";  
}  
?>
```

Penjelasan Program 8.16

Untuk menampilkan array *asosiatif* bisa menggunakan perulangan foreach dan while dengan fungsi list() dan each(). Baris ke-5 dari program di atas bisa dinyatakan dengan

MODUL PRAKTIKUM PEMROGRAMAN WEB

JURUSAN TEKNIK INFORMATIKA

UIN MAULANA MALIK IBRAHIM MALANG

pernyataan “untuk setiap elemen array \$arrNilai, masukkan key/index-nya ke variabel \$nama dan masukkan value-nya ke variabel \$nilai”. Pada baris ke-8 terdapat fungsi **reset()** yang berfungsi mengembalikan pointer array ke elemen pertama. Pada baris ke-10, array asosiatif ditampilkan dengan struktur while yang menggunakan fungsi **list()** dan **each()**.

Program 8.17

Nama File : array05.php
Deskripsi : Program mencetak struktur array.

```
<?
SarrWarna = array("Blue","Black","Red","Yellow","Green");
SarrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75, "Budi" => 85);
echo "<pre>"; print_r ($arrWarna);
echo "<br>"; print_r ($arrNilai); echo
"</pre>";
?>
```

Penjelasan Program 8.17

Pada program di atas terdapat fungsi **print_r()** (baris ke-5) yang berfungsi untuk menampilkan struktur dari array.

Fungsi Pengurutan Array

- arsort() – Pengurutan array berdasarkan value secara descending
- asort() – Pengurutan array berdasarkan value secara ascending
- krsort() - Pengurutan array berdasarkan index/key secara descending
- ksort() - Pengurutan array berdasarkan index/key secara ascending
- rsort() - Pengurutan array berdasarkan value secara descending dengan
- mengubah index/key
- sort() - Pengurutan array berdasarkan value secara ascending dengan
- mengubah index/key
- shuffle() – Random pengurutan array

Fungsi Pengaturan Pointer Array

- current() – Mendapatkan elemen array yang ditunjuk oleh pointer
- end() – Pointer menunjuk pada elemen array terakhir
- key() – Mendapatkan key yang ditunjuk oleh pointer
- next() – Pointer menunjuk pada elemen selanjutnya
- prev() – Pointer menunjuk pada elemen sebelumnya
- reset() – Memindahkan pointer array ke awal (elemen pertama)
- count() – Menghitung jumlah elemen array

Fungsi Pencarian pada Array

- array_search() – Mencari posisi (key) dari suatu value dalam array
- array_key_exists() – Memeriksa suatu key ada dalam array atau tidak
- in_array() – Memeriksa suatu elemen ada dalam array atau tidak

Program 8.18

Nama File : array06.php
Deskripsi : Program mengurutkan array dengan sort() dan rsort().

```
<?php
$arrNilai = array ("Ani" => 80, "Otim" => 90, "Sri" => 75, "Budi" => 85);
echo "<b>Array sebelumpengurutan</b>"; echo "<pre>";
print_r($arrNilai); echo "</pre>";

sort($arrNilai); reset($arrNilai);
echo "<b>Array setelah pengurutan dengansort()</b>"; echo "<pre>";
print_r($arrNilai); echo "</pre>";

rsort($arrNilai); reset($arrNilai);
echo "<b>Array setelah pengurutan denganrsort()</b>"; echo "<pre>";
print_r($arrNilai); echo "</pre>";
?>
```

Program 8.19

Nama File : array07.php
Deskripsi : Program mengurutkan array dengan asort() dan arsort().

```
<?php
$arrNilai = array ("Ani" => 80, "Otim" => 90, "Sri" => 75, "Budi" => 85);
echo "<b>Array sebelumpengurutan</b>"; echo "<pre>";
print_r($arrNilai); echo "</pre>";

asort($arrNilai); reset($arrNilai);
echo "<b>Array setelah pengurutan denganasort()</b>"; echo "<pre>";
print_r($arrNilai); echo "</pre>";

arsort($arrNilai); reset($arrNilai);
echo "<b>Array setelah pengurutan denganarsort()</b>"; echo "<pre>";
print_r($arrNilai); echo "</pre>";
?>
```

Program 8.20

Nama File : array08.php
Deskripsi : Program mengurutkan array dengan ksort() dan krsort().

```
<?php
$arrNilai = array ("Ani"=>80, "Otim"=>90,"Sri"=>75, "Budi"=>85);
echo "<b>Array sebelumpengurutan</b>"; echo "<pre>";
print_r($arrNilai); echo "</pre>";

ksort($arrNilai); reset($arrNilai);
echo "<b>Array setelah pengurutan denganksort()</b>"; echo "<pre>";
print_r($arrNilai); echo "</pre>";

krsort($arrNilai); reset($arrNilai);
```

```
echo "<b>Array setelah pengurutan dengankrsort()</b>"; echo "<pre>";  
print_r($arrNilai); echo "</pre>";  
?>
```

Program 8.21

Nama File : array09.php

Deskripsi : Program mengatur posisi pointer dalam array.

```
<?php  
$transport = array('foot', 'bike', 'car', 'plane'); echo "<pre>";  
print_r ($transport); echo "</pre>";  
$mode = current($transport);  
echo $mode."<br>"; // $mode = 'foot';  
$mode = next($transport);  
echo $mode."<br>"; // $mode = 'bike';  
$mode = current($transport);  
echo $mode."<br>"; // $mode = 'bike';  
$mode = prev($transport);  
echo $mode."<br>"; // $mode = 'foot';  
$mode = end($transport);  
echo $mode."<br>"; // $mode = 'plane';  
$mode = current($transport);  
echo $mode."<br>"; // $mode = 'plane';  
?>
```

Program 8.22

Nama File : array10.php

Deskripsi : Program mencari elemen array.

```
<?php  
$arrBuah = array ("Mangga", "Apel", "Pisang", "Kedondong", "Jeruk");  
if (in_array ("Kedondong", $arrBuah)) { echo "Ada buah Kedondong di sini";  
} else {  
    echo "Tidak ada buah Kedondong disini";  
}  
?>
```

Fungsi dalam PHP

Bentuk umum pendefinisian fungsi dalam PHP

```
function nama_fungsi(parameter1,...,n) { statement2;  
}
```

Program 8.23

Nama File : fungsi01.php

Deskripsi : Program sederhana mendefinisikan dan memanggil fungsi tanpa parameter dan tanpa return value.

```
<?php
function cetak_ganjil () {
    for ($i=0; $i<100; $i++) {
        if ($i%2 == 1) { echo "$i ";
        }
    }
}
//pemanggilan fungsi
cetak_ganjil();
?>
```

Program 8.24

Nama File : array02.php
Deskripsi : Program fungsi dengan parameter.

```
<?php
function cetak_ganjil ($awal, $akhir)
{ for ($i=$awal; $i<$akhir; $i++) {
    if ($i%2 == 1) {
        echo "$i ";
    }
}
}
//pemanggilan fungsi
$a = 10;
$b = 50;

echo "<b>Bilangan ganjil dari $a sampai $b : </b><br>"; cetak_ganjil($a, $b);
?>
```

Program 8.25

Nama File : fungsi03.php
Deskripsi : Program fungsi yang mengembalikan nilai.

```
<?php
function luas_lingkaran ($jari) {
    return 3.14*$jari*$jari;
}
//pemanggilan fungsi
$r = 10;
echo "Luas lingkaran dengan jari-jari $r = ";
echo luas_lingkaran($r);
?>
```

Program 8.26

Nama File : fungsi04.php
Deskripsi : Program passing by value dalam fungsi.

```
<?php
function tambah_string ($str) {
```

MODUL PRAKTIKUM PEMROGRAMAN WEB
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

```
        $str = $str . ", Jakarta"; return $str;
    }
```

```
$str = "Universitas Budi Luhur";
echo "\$str = $str<br>";
echo tambah_string ($str). "<br>";
echo "\$str = $str<br>";
?>
```

Program 8.27

Nama File : fungsi05.php
Deskripsi : Program passing by reference dalam fungsi.

```
<?php
function tambah_string (&$str) {
    $str = $str . ", Jakarta";
    return $str;
}
```

```
$str = "Universitas Budi Luhur";
echo "\$str = $str<br>";
echo tambah_string ($str). "<br>";
echo "\$str = $str<br>";
?>
```

Program 8.28

Nama File : fungsi06.php
Deskripsi : Program menampilkan defined-function yang didukung oleh PHP.

```
<?php
function luas_lingkaran ($jari) {
    return 3.14*$jari*$jari;
}
$arr = get_defined_functions();
echo "<pre>";
print_r($arr);
echo "</pre>";
?>
```

Program 8.29

Nama File : fungsi07.php
Deskripsi : Program memeriksa suatu fungsi tersedia di PHP atau tidak.

```
<?php
if (function_exists('exif_read_data')) {
    echo "Fungsi exif_read_data() ada di PHP.<br />\n";
} else {
    echo "Fungsi exif_read_data() tidak ada di PHP.<br />\n";
}
?>
```