

Modul Panduan Praktikum Pemrograman Web

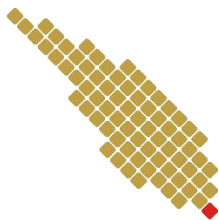
PERTEMUAN

6

PHP I

Asisten Praktikum

- Aminudin Fadila
- Defangga Aby Vonega
- Markus Togi Fedrian Rivaldi Sinaga



TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI PRODUKSI INDUSTRI DAN INFORMASI
INSTITUT TEKNOLOGI SUMATERA
2022

Dasar Teori

1. PHP

PHP adalah singkatan dari “*PHP: Hypertext Preprocessor*”, yaitu bahasa pemrograman sisi server yang digunakan secara luas untuk penanganan, pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. Sejarah PHP bermula pada tahun 1994 ketika programmer kelahiran Denmark yang sekarang berdomisili di Kanada, Rasmus Lerdorf membuat sebuah script (kode program) dengan bahasa Perl untuk web pribadinya.

Dengan alasan untuk meningkatkan performa, Rasmus Lerdorf kemudian membuat ulang kode program tersebut dalam bahasa C. Ia juga mengembangkannya lebih lanjut sehingga memiliki script tersebut memiliki kemampuan untuk memproses form HTML dan berkomunikasi dengan database. Lerdorf kemudian merilis kode tersebut ke publik dengan sebutan Personal Home Page Tools (PHP Tools) version 1.0.

Seiring dengan pengembangan dan penambahan fitur web pada saat itu, pada April 1996, Rasmus Lerdorf mengumumkan PHP/FI versi 2.0. PHP versi 2 ini dirancang Lerdorf pada saat mengerjakan sebuah proyek di University of Toronto yang membutuhkan pengolahan data dan tampilan web yang rumit. Hingga modul ini dibuat, PHP sudah berkembang hingga PHP versi 8.1.0.

2. Variabel

Variabel adalah sebuah wadah untuk menyimpan sebuah informasi (nilai). Variabel pada PHP diawali dengan tanda dollar \$ diikuti dengan nama variabelnya. Contoh penggunaan variabel dapat dilihat pada **Gambar 1**. Berbeda dengan bahasa pemrograman lain, pada PHP tidak ada perintah yang mengharuskan pendeklarasian sebuah variabel. Variabel tersebut tercipta ketika langsung diisi dengan nilai (inisialisasi). Beberapa aturan pada pemberian nama variabel adalah :

- Nama variabel harus diawali dengan huruf atau underscore (_)
- Nama variabel tidak boleh diawali dengan angka
- Nama variabel hanya boleh mengandung karakter *alphanumeric*
- Berlaku case-sensitive pada variabel PHP.



```
<?php
// Penulisan yang benar
$a = "Hello world";
$number = 100;
$nama_saya;

// Penulisan yang salah
$nama saya;
?>
```

Gambar 1. Contoh penggunaan variabel pada PHP

3. Operator

Operator merupakan suatu tool atau alat yang disediakan untuk keperluan manipulasi data. Operator dapat dibedakan menjadi :

- **Operator Aritmatika**

| Nama Operasi | Contoh | Keterangan |
|------------------|--------------|-------------------------------|
| Penjumlahan | $\$a + \b | Penjumlahan $\$a$ dan $\$b$ |
| Pengurangan | $\$a - \b | Selisih $\$a$ dan $\$b$ |
| Perkalian | $\$a * \b | Perkalian $\$a$ dan $\$b$ |
| Pembagian | $\$a / \b | Pembagian $\$a$ dan $\$b$ |
| Modulo / Modulus | $\$a \% \b | Sisa hasil $\$a$ dibagi $\$b$ |

- **Operator Logika**

| Nama Operasi | Contoh | Keterangan |
|--------------|------------------------|---|
| AND | $\$a \text{ and } \b | Bernilai benar jika $\$a$ dan $\$b$ benar |
| OR | $\$a \text{ or } \b | Bernilai benar jika $\$a$ atau $\$b$ benar |
| XOR | $\$a \text{ xor } \b | Bernilai benar jika salah satu $\$a$ atau $\$b$ benar, tidak keduanya |
| Not True | $!\$a$ | Inverse dari $\$a$ |
| && | $\$a \ \&\& \ \b | Bernilai benar jika $\$a$ dan $\$b$ benar |
| | $\$a \ \ \b | Bernilai benar jika $\$a$ atau $\$b$ benar |

- **Operator Perbandingan**

| Nama Operasi | Contoh | Keterangan |
|-------------------|---------------|--|
| Sama dengan | $\$a == \b | Bernilai benar jika $\$a$ sama dengan $\$b$ |
| Identik | $\$a === \b | Bernilai benar jika $\$a$ sama dengan $\$b$, dan memiliki tipe data yang sama (PHP4 ke atas) |
| Tidak sama dengan | $\$a != \b | Bernilai benar jika $\$a$ tidak sama dengan $\$b$ |
| Tidak identik | $\$a !== \b | Bernilai benar jika $\$a$ tidak sama dengan $\$b$ atau keduanya tidak memiliki tipe data yang sama |

| | | |
|-----------------------|----------------|--|
| Lebih kecil | $\$a < \b | Bernilai benar jika \$a lebih kecil dari \$b |
| Lebih kecil atau sama | $\$a \leq \b | Bernilai benar jika \$a lebih kecil atau sama dengan \$b |
| Lebih besar | $\$a > \b | Bernilai benar jika \$a lebih besar dari \$b |
| Lebih besar atau sama | $\$a \geq \b | Bernilai benar jika \$a lebih besar atau sama dengan \$b |

4. Array

Array adalah sebuah variabel yang bisa menampung lebih dari satu nilai. Di dalam sebuah array, kita bisa mengakses sebuah nilai dengan memanggil index-nya. Index sendiri merepresentasikan urutan posisi elemen, dimana urutan pertama dimulai dari 0.

The screenshot shows a PHP script on the left and its output on the right. The script defines an array with four elements: "nol", "satu", "dua", and "tiga". It then accesses the element at index 2 and uses `var_dump` to display the entire array.

```
<?php
$ini_array = [
    "nol",
    "satu",
    "dua",
    "tiga"
];
echo "<b>Mengakses 1 elemen array </b><br/>";
echo "\$ini_array[2] = ".$ini_array[2]."<br/>";
echo "<br/><b>Menampilkan array untuk debugging</b>";
echo "<pre>";
var_dump($ini_array);
echo "</pre>";
?>
```

The output on the right shows the value of the element at index 2 and the debug dump of the array:

```
Mengakses 1 elemen array
$ini_array[2] = dua

Menampilkan array untuk debugging

array(4) {
    [0]=>
        string(3) "nol"
    [1]=>
        string(4) "satu"
    [2]=>
        string(3) "dua"
    [3]=>
        string(4) "tiga"
}
```

- **Associative Array**

Pada array dalam PHP bisa dilihat key-nya selalu berupa angka integer dengan mulai dari angka 0. Dengan array asosiatif memungkinkan kita untuk membuat array dengan menggunakan key berupa string.

The screenshot shows a PHP script on the left and its output on the right. The script defines an associative array with string keys and values. It then accesses the value for the key "dua" and uses `var_dump` to display the entire array.

```
<?php
$ini_array = [
    "nol" => "apel",
    "satu" => "apel juga",
    "dua" => "yg ini juga apel",
    "tiga" => "semua-mua apel"
];
echo "<b>Mengakses 1 elemen array </b><br/>";
echo "\$ini_array[\"dua\"] = ".$ini_array["dua"]."<br/>";
echo "<br/><b>Menampilkan array untuk debugging</b>";
echo "<pre>";
var_dump($ini_array);
echo "</pre>";
?>
```

The output on the right shows the value for the key "dua" and the debug dump of the associative array:

```
Mengakses 1 elemen array
$ini_array["dua"] = yg ini juga apel

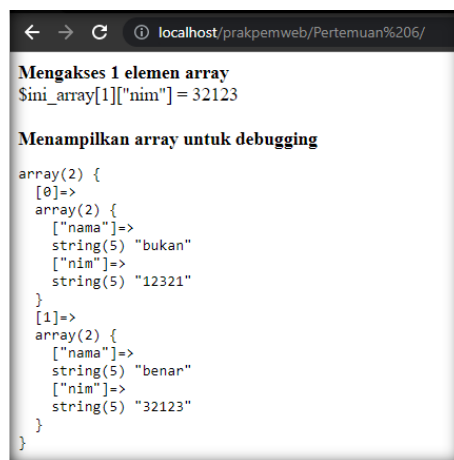
Menampilkan array untuk debugging

array(4) {
    ["nol"]=>
        string(4) "apel"
    ["satu"]=>
        string(9) "apel juga"
    ["dua"]=>
        string(16) "yg ini juga apel"
    ["tiga"]=>
        string(14) "semua-mua apel"
}
```

- **Multidimensional Array**

Multidimensional array bisa kita sebut sebagai array di dalam array. Contoh penggunaan array dalam array seperti kita mengenal sebuah matriks

```
<?php
$ini_array = [
    [
        "nama" => "bukan",
        "nim" => "12321",
    ],
    [
        "nama" => "benar",
        "nim" => "32123",
    ]
];
echo "<b>Mengakses 1 elemen array </b><br/>";
echo "\$ini_array[1][\"nim\"] = ".$ini_array[1]["nim"]."<br/>";
echo "<br/><b>Menampilkan array untuk debugging</b>";
echo "<pre>";
var_dump($ini_array);
echo "</pre>";
?>
```



```
Mengakses 1 elemen array
Sini_array[1][\"nim\"] = 32123

Menampilkan array untuk debugging
array(2) {
  [0]=>
  array(2) {
    [\"nama\"]=>
    string(5) \"bukan\"
    [\"nim\"]=>
    string(5) \"12321\"
  }
  [1]=>
  array(2) {
    [\"nama\"]=>
    string(5) \"benar\"
    [\"nim\"]=>
    string(5) \"32123\"
  }
}
```

5. Percabangan

- **If**

Percabangan if adalah percabangan paling dasar. Kegunaannya adalah memeriksa sebuah ekspresi logika. Jika suatu variabel logika bernilai true, maka proses yang ada di dalam blok kode if akan dijalankan. Jika tidak, maka kode yang ada di dalam blok if tidak akan dijalankan.

```
<?php
if( kondisi ) {
    // aksi yg akan dijalankan
    // jika kondisi terpenuhi
}
?>
```

- **If Else**

Percabangan if else digunakan disaat program memiliki 2 kondisi yang berbeda. Jika suatu variabel logika bernilai true, maka proses yang ada di dalam blok kode if akan dijalankan. Jika tidak, maka proses yang ada di dalam blok kode else yang akan dijalankan.

```
<?php
if( kondisi ) {
    // aksi yg akan dijalankan
    // jika kondisi terpenuhi
} else {
    // aksi yg akan dijalankan jika
    // kondisi pada 'if' tidak terpenuhi
}
?>
```

- **If Else If Else**

Percabangan if else if else digunakan disaat program memiliki lebih dari 2 kondisi yang berbeda. Jika kondisi pertama yang benar, maka proses yang ada di dalam blok kode if akan dijalankan. Jika kondisi pertama salah dan kondisi kedua benar, maka proses yang ada di dalam blok kode elseif akan dijalankan. Jika kondisi pertama dan kedua salah, maka proses yang ada di dalam blok kode else yang akan dijalankan.

```
<?php
if( kondisi_1 ) {
    // aksi yg akan dijalankan
    // jika kondisi_1 terpenuhi
} elseif ( kondisi_2 ) {
    // aksi yg akan dijalankan
    // jika kondisi_2 terpenuhi
} else {
    // aksi yg akan dijalankan jika
    // kondisi_1 dan kondisi_2
    // tidak terpenuhi
}
?>
```

- **Switch Case**

Percabangan switch case digunakan untuk kasus-kasus yang lebih sederhana dari if else. Percabangan switch case digunakan pada saat suatu kondisi yang dicek hanya 1 variabel saja dan tidak memakai variabel logika seperti :

```
<?php
switch ( $variabel ) {
    case nilai_1 :
        // aksi yg dijalankan jika
        // $variabel sama dengan nilai_1
        break;
    case nilai_2 :
        // aksi yg dijalankan jika
        // $variabel sama dengan nilai_2
        break;
    default :
        // aksi yg dijalankan jika tidak ada
        // nilai case yg sama dengan $variabel
        break;
}
?>
```

6. Perulangan

- **For**

Dengan perulangan for, kita bisa menentukan jumlah perulangan yang kita inginkan :

```
<?php
for ( nilai_awal; nilai_akhir; increment/decrement ) {
    // aksi yg akan dijalankan
    // untuk setiap pengulangan
}
?>
```

- **While**

Jika perulangan for kita sudah mengetahui ingin berapa kali kita melakukan perulangan, pada perulangan while ini kita belum tahu pasti berapa kali perulangan akan kita lakukan. Perulangan while ini akan berhenti jika kondisi perulangan sudah terpenuhi :

```
<?php
kondisi_awal;
while ( kondisi_loop_berhenti ) {
    // aksi yg akan dijalankan
    // untuk setiap pengulangan
    increment/decrement;
}
?>
```

- **Do-While**

Perulangan do-while hampir sama dengan perulangan while. Jika pada perulangan while perulangan akan berhenti jika kondisi while terpenuhi, pada perulangan do-while akan melakukan perulangan terlebih dahulu lalu mengecek kondisi perulangan.

```
<?php
kondisi_awal;
do {
    // aksi yg akan dijalankan
    // untuk setiap pengulangan
    increment/decrement;
} while ( kondisi_loop_berhenti );
?>
```

- **Foreach**

Perulangan foreach ini sedikit berbeda dengan ketiga perulangan diatas. Perulangan foreach ini dikhususkan untuk tipe data array. Dengan foreach kita dapat melakukan perulangan pada setiap elemen array tanpa harus tahu berapa panjang array tersebut.

```
<?php
$nameArray = ["Markus", "Togi", "Fedrian", "Rivaldi"];
foreach ( $nameArray as $name ) {
    echo $name."<br/>";
}
?>
```

7. Fungsi

Fungsi adalah suatu kumpulan blok kode, yang menerima suatu inputan, melakukan satu tugas tertentu, dan secara opsional ia bisa mengembalikan suatu nilai.

Ekspresi Fungsi

```
<?php
function namaFungsi( parameter ) {
    // aksi yang dilakukan ketika fungsi dipanggil
}
?>
```

Struktur fungsi diatas dapat dijelaskan sebagai berikut :

- ❖ **namaFungsi** merupakan deklarasi nama fungsi yang akan kita buat. Dalam mendeklarasikan nama fungsi memiliki beberapa aturan, antara lain:
 - Tidak boleh sama dengan fungsi yang sudah ada di dalam PHP, karena PHP tidak mendukung fasilitas overloading yaitu kondisi dimana fungsi dapat menggunakan nama yang sama namun keluaran yang berbeda dengan syarat parameter pada fungsi tersebut berbeda.
 - Hanya terdiri dari huruf, angka dan garis bawah (*underscore*).
 - Tidak boleh diawali dengan angka.
- ❖ **parameter** adalah data/nilai masukan yang akan diolah oleh fungsi

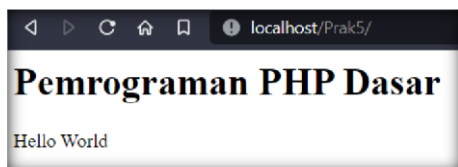
Implementasi Fungsi

- ❖ **Fungsi tanpa parameter**

```
1 <?php
2
3 function belajarPHP(){
4     echo "<h1>Pemrograman PHP Dasar</h1>";
5     echo "Hello World";
6 }
7
8 belajarPHP();
9
10 ?>
```

Fungsi **belajarPHP()** tanpa menggunakan parameter berisi 2 output yaitu "<h1>Pemrograman PHP Dasar</h1>" dan "Hello World".

Hasil yang didapatkan :

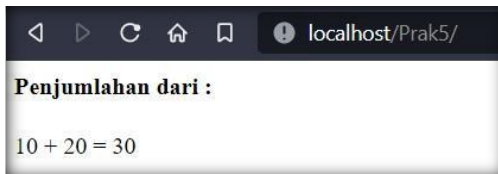


❖ Fungsi dengan parameter

```
1 <?php
2
3 function penjumlahan($x, $y){
4     $total = $x + $y;
5
6     echo "<h4>Penjumlahan dari : </h4>";
7     echo "$x + $y = $total";
8 }
9
10 penjumlahan(10, 20);
11
12 ?>
13
```

Fungsi penjumlahan memiliki parameter \$x dan \$y, di dalam fungsi penjumlahan terdapat operasi aritmatika yaitu $x + y$ yang disimpan dalam variabel \$total.

Hasil yang didapatkan :



Penjumlahan dari :
10 + 20 = 30

❖ Fungsi dengan return

```
1 <?php
2
3 function penjumlahan($x, $y){
4     $total = $x + $y;
5
6     return $total;
7 }
8
9 $total = penjumlahan(10, 20);
10 echo "Penjumlahan 10 + 20 adalah $total";
11 echo "<br>";
12 echo "penjumlahan 5 + 8 adalah".penjumlahan(5,8);
13
14 ?>
```

Pada kode program diatas, kita membuat fungsi penjumlahan dengan parameter \$x dan \$y. Return untuk mengembalikan nilai \$total.

Hasil yang didapatkan :



Penjumlahan 10 + 20 adalah 30
penjumlahan 5 + 8 adalah13

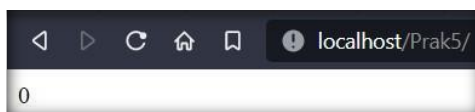
❖ **Passing parameter by value**

Passing by value adalah teknik memasukkan parameter ke dalam sebuah fungsi dengan cara membuat salinan dari variabel asli, sehingga variabel asli tidak terpengaruh. Untuk lebih jelasnya, perhatikan contoh berikut:

```
1 <?php
2
3 function increment($value){
4     $value++;
5 }
6
7 $x = 0;
8 increment($x);
9 echo $x;
10
11 ?>
```

Awalnya variabel \$x bernilai 0, ketika menjalankan fungsi increment, maka nilai variabel \$x akan disalin ke variabel \$value. Maka operasi yang dilakukan oleh fungsi increment adalah operasi yang dilakukan terhadap variabel \$value yang merupakan salinan variabel \$x. Sehingga nilai variabel \$x tidak berubah.

Hasil yang didapatkan :



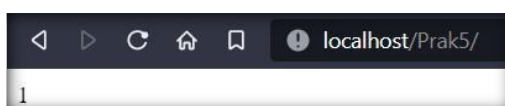
localhost/Prak5/
0

❖ **Passing parameter by reference**

Passing parameter by reference memungkinkan kita untuk melakukan manipulasi terhadap variabel yang menjadi parameter melalui sebuah fungsi. Untuk lebih jelasnya, perhatikan contoh latihan berikut yang merupakan modifikasi dari skrip program sebelumnya:

```
1 <?php
2
3 function increment(&$value){
4     $value++;
5 }
6
7 $x = 0;
8 increment($x);
9 echo $x;
10
11 ?>
```

Hasil yang didapatkan :



localhost/Prak5/
1

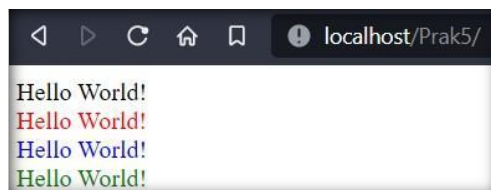
❖ **Default Parameter**

Default parameter adalah nilai otomatis yang digunakan saat pemanggilan fungsi parameter yang tidak berisi nilai.

```
1 <?php
2
3 function bar($color = 'black'){
4     echo '<font color="'. $color. '">Hello World!</font><br>';
5 }
6
7 bar();
8 bar('red');
9 bar('blue');
10 bar('green');
11
12 ?>
```

Secara default, teks tersebut akan berwarna hitam jika fungsi bar dipanggil tanpa ada parameter. Namun jika pemanggilan fungsi disertai dengan \$color, maka parameter default akan diabaikan dan menggunakan parameter yang dimasukkan.

Hasil yang didapatkan :



localhost/Prak5/

Hello World!
Hello World!
Hello World!
Hello World!

Latihan Percobaan

Buatlah program menggunakan bahasa PHP, dimana program tersebut memiliki sebuah fungsi yang dapat mengubah bilangan integer ke bilangan Romawi.

| Romawi | Nilai |
|----------|-------------|
| I | 1 |
| V | 5 |
| X | 10 |
| L | 50 |
| C | 100 |
| D | 500 |
| M | 1000 |

Ingat bahwa pada bilangan Romawi terdapat penempatan yang bertujuan untuk mengurangi nilai suatu bilangan, antara lain :

- **I** (1) sebelum **V** (5), akan mengurangi **V** (5) sebanyak **I** (1) menjadi **IV** (4)
- **I** (1) sebelum **X** (10), akan mengurangi **X** (10) sebanyak **I** (1) menjadi **IX** (9)
- **X** (10) sebelum **L** (50), akan mengurangi **L** (50) sebanyak **X** (10) menjadi **XL** (40)
- **X** (10) sebelum **C** (100), akan mengurangi **C** (100) sebanyak **X** (10) menjadi **XC** (90)
- **C** (100) sebelum **D** (500), akan mengurangi **D** (500) sebanyak **C** (100) menjadi **CD** (400)
- **C** (100) sebelum **M** (1000), akan mengurangi **M** (1000) sebanyak **C** (100) menjadi **CM** (900)

Contoh :

```
echo intToRoman(58);
```

← → ↻ localhost/prakpemweb/Pertemuan%206/
LVIII

```
echo intToRoman(1994);
```

← → ↻ localhost/prakpemweb/Pertemuan%206/
MCMXCIV

Daftar Pilihan Referensi

- [W3Schools](#)
- [PHP Dasar - Web Programming Unpas \(YouTube\)](#)