

**LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB & MOBILE I**



**NAMA : SATRIA SHANO  
NIM : 11191068  
KELAS : C  
MODUL : I**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PALANGKA RAYA  
2021**

# BAB I

## TUJUAN DAN LANDASAN TEORI

### 1. Tujuan Praktikum

- 1.1. Mahasiswa memahami dasar-dasar pemrograman PHP yang biasa digunakan pada umumnya..
- 1.2. Mahasiswa mampu membuat program sederhana menggunakan bahasa pemrograman PHP.

### 2. Landasan Teori

Script PHP bisa diletakkan di manapun dalam dokumen, dimulai dengan tanda “<”. Dokumen PHP disimpan dengan ekstensi “.php”, dimana file ini bisa menyimpan tag HTML dan beberapa script PHP. Setiap statement dalam PHP harus diakhiri dengan semicolon “;”. Berikut adalah contoh script PHP untuk menampilkan text “Hello World!”

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Halaman PHP Pertama ku</h1>
    <?php
      echo "Hello World!";
    ?>
  </body>
</html>
```

**Gambar 1.1 PHP**

### Mendeklarasikan variabel PHP

Pada pemrograman php, variabel dimulai dengan tanda “\$” dan diikuti dengan nama variabel. Tidak seperti bahasa pemrograman lainnya, php tidak memiliki perintah khusus untuk mendeklarasikan variabel. Tipe data ditentukan ketika nilai dimasukkan kedalam variabel tersebut. Sebuah variabel bisa memiliki nama yang singkat (seperti x dan y) atau nama deskriptif (contoh : umur, namamobil,

total\_volume). Aturan-aturan untuk variabel php adalah sebagai berikut:

Sebuah variabel dimulai dengan tanda “\$”, diikuti oleh nama variabel.

- Sebuah nama variabel harus mulai dengan huruf atau karakter garis bawah “\_”.
- Sebuah nama variabel tidak boleh dimulai dengan angka.
- Sebuah nama variabel hanya bisa mengandung karakter alphanumeric dan garis bawah (A-z, 0-9, dan \_)
- Nama-nama variabel adalah case-sensitive, yaitu huruf besar dan huruf kecil mempengaruhi perbedaan variabel (\$umur dan \$UMUR adalah dua variabel yang berbeda)

Perintah yang biasa digunakan untuk mencetak isi dari sebuah variabel adalah echo. Berikut adalah contoh program menentukan nilai variabel dan mencetaknya di browser:

```
<?php
$txt = "W3Schools.com";
echo "I love $txt!";
echo "I love " . $txt . "!"; // menghasilkan keluaran yang sama dengan
perintah cetak diatas
?>
```

**Gambar 1.2 PHP**

Lingkup dari sebuah variabel adalah bagian dari skrip dimana variabel bisa dirujuk atau digunakan. Ada tiga lingkup variabel: Lokal, Global dan Static. Variabel yang dideklarasikan diluar fungsi memiliki lingkup global, sedangkan variabel yang dideklarasikan didalam fungsi memiliki lingkup lokal yang hanya bisa diakses didalam fungsi saja. Kelebihan dari penggunaan variabel lokal adalah kita bisa memiliki variabel lokal dengan nama yang sama pada fungsi yang berbeda, karena variabel lokal hanya dikenali oleh fungsi yang mendeklarasikannya saja. Keyword global digunakan untuk mengakses variabel global dari dalam fungsi. Php juga menyimpan semua variabel global dalam sebuah array yang disebut

`$GLOBALS[index]`. Index adalah nama dari variabel, array ini juga bisa diakses dari dalam fungsi dan bisa digunakan untuk mengupdate variabel global secara langsung. Normalnya, ketika fungsi dieksekusi, semua variabel-variabelnya dihapus. Namun terkadang kita perlu untuk mempertahankan nilai variabel lokal supaya tidak dihapus.

Untuk itu, gunakan keyword `static` dalam pendeklarasian variabel yang nilainya ingin dipertahankan.

```
<?php
    $x = 5; // lingkup global
    function myTest() {
        //menggunakan x didalam fungsi ini akan menghasilkan error.
        echo "<p>Variabel x di dalam fungsi adalah : $x</p>";
    }
    myTest();

    echo "<p>Variabel x di luar fungsi adalah : $x</p>";
?>
```

**Gambar 1.3 PHP**

```
<?php
    function myTest() {
        $x = 5; // Lingkup lokal
        echo "<p>Variabel x di dalam fungsi adalah : $x</p>";
    }
    myTest();
    // Menggunakan x diluar fungsi akan menghasilkan error
    echo "<p>Variabel x di luar fungsi adalah: $x</p>";
?>
```

**Gambar 1.4 PHP**

```

<?php
    $x = 5;
    $y = 10;

    function myTest() {
        global $x, $y;
        $y = $x + $y;
    }

    myTest();
    echo $y; // output 15
?>

```

**Gambar 1.5 PHP**

```

<?php
    $x = 5;
    $y = 10;

    function myTest() {
        $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
    }
    myTest();
    echo $y; // output 15
?>

```

**Gambar 1.6 PHP**

```

<?php
    function myTest() {
        static $x = 0;
    }

```

**Gambar 1.7 PHP**

```

        echo $x;
        $x++;
    }

    myTest();
    myTest();
    myTest();
?>

```

**Gambar 1.8 PHP**

Echo dan Print Ada dua cara untuk menampilkan output di PHP yaitu dengan menggunakan perintah echo dan print. Terdapat sedikit perbedaan dalam penggunaan echo dan print, echo tidak mengembalikan nilai sedangkan print mengembalikan nilai 1 sehingga bisa digunakan dalam ekspresi. Echo bisa menggunakan banyak parameter sedangkan print bisa menggunakan satu argument. Secara marjinal echo lebih cepat dari print.

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";

print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

**Gambar 1.9** Echo and Print

### **Tipe Data PHP**

Variabel bisa menyimpan data dengan tipe yang berbeda-beda, berikut adalah tipe-tipe data yang didukung oleh PHP

- String
- Integer
- Float
- Boolean
- Array
- Object
- NULL
- Resource

### **String**

String adalah susunan dari karakter-karakter seperti “Hello world!”. String dituliskan didalam kutip satu maupun kutip dua. Berikut adalah contoh variabel yang menyimpan data string.

```
<?php
    $x = "Hello world!";
    $y = 'Hello world!';
    echo $x;
    echo "<br>";
    echo $y;
?>
```

**Gambar 1.10** Tipe data

### **Integer**

Tipe data integer atau bilangan bulat adalah bilangan bukan pecahan antara - 2.147.483.648 hingga 2.147.483.647. Aturan-aturan pada integer adalah sebagai berikut:

- integer harus memiliki sedikitnya satu digit.
- Integer tidak boleh memiliki koma.
- Integer bisa positif maupun negatif.
- Integer bisa dispesifikasikan dalam tiga format: desimal berbasis 10, hexadesimal berbasis 16 atau oktal berbasis 8. Berikut adalah contoh variabel integer. Fungsi var\_dump() mengembalikan tipe data dan nilainya:

```
<?php
    $x = 5985;
    var_dump($x);
?>
```

**Gambar 1.11** Tipe Data

### **Float**

Float (bilangan floating-point) adalah bilangan dengan tanda desimal atau bilangan dalam bentuk eksponensial. Berikut contoh penggunaan variabel float.

```
<?php
    $x = 10.365;

    var_dump($x);
?>
```

**Gambar 1.12** Tipe Data

### **Boolean**

Boolean merepresentasikan dua kemungkinan keadaan: TRUE atau FALSE. Boolean sering digunakan pada kondisi percobaan. Berikut contoh penggunaannya: \$x = true; \$y = false

### **Object / objek**

Object adalah sebuah tipe data yang menyimpan data dan informasi bagaimana memproses data tersebut. Object harus di deklarasikan secara eksplisit. Pertama kita harus mendeklarasikan class dari object. Class adalah struktur yang bisa memiliki properties dan method. Berikut contohnya:

```
<?php
class Car {
    function Car() {
        $this->model = "VW";
    }
}
// membuat sebuah objek
$herbie = new Car();
// Menunjukkan isi dari
objek echo $herbie->model;
?>
```

**Gambar 1.13** Object/objek

### **Nilai NULL**

NULL adalah tipe data khusus yang bisa memiliki hanya satu nilai: NULL. Sebuah variabel dengan tipe data NULL adalah variabel yang tidak memiliki nilai. Jika sebuah variabel dibuat tanpa nilai, maka otomatis akan ditentukan dengan nilai NULL. Variabel bisa juga dikosongkan dengan mengatur nilainya menjadi NULL:



```
<?php
    $x = "Hello world!";
    $x = null;
    var_dump($x);
?>
```

**Gambar 1.14** Nilai Null

### Resource

Tipe khusus resource bukanlah merupakan tipe data yang sebenarnya. Ia menyimpan referensi fungsi dan sumberdaya eksternal untuk PHP. Contoh umum penggunaan tipe data resource adalah pada pemanggilan database.

### Fungsi-fungsi Manipulasi String

String merupakan susunan dari karakter yang memiliki fungsi khusus untuk mengolahnya. Seperti fungsi strlen(), berfungsi menghitung panjang dari string. Contohnya sebagai berikut:

```
<?php
    echo strlen("Hello world!");
?>
```

**Gambar 1.15** Manipulasi String

Hasil dari kode tersebut adalah panjang dari string “Hello world!” yaitu 12 karakter. Panjang string juga bisa diukur perkata menggunakan fungsi str\_word\_count() seperti contoh berikut:

```
<?php
    echo str_word_count("Hello world!");
?>
```

**Gambar 1.16** Manipulasi String

Keluaran dari kode tersebut berupa jumlah kata pada string “Hello world!” yaitu 2. PHP juga memiliki fungsi yang bisa digunakan untuk membalikkan susunan string, yaitu menggunakan fungsi strrev() seperti contoh kode program berikut yang menghasilkan string “!dlrow olleH” dari inputan string “Hello world!”.

```
<?php
    echo strrev("Hello world!");
?>
```

**Gambar 1.17** Manipulasi String

PHP juga memiliki fungsi untuk menemukan teks yang spesifik didalam sebuah string, yaitu dengan menggunakan fungsi strpos(). Jika teks yang dicari ditemukan didalam string, maka fungsi akan mengembalikan index posisi dari karakter awal teks pertama yang cocok. Jika tidak ditemukan, maka fungsi strpos() akan mengembalikan nilai FALSE. Kode program berikut adalah contoh dari penggunaan fungsi strpos() pada string “Hello world!” untuk menemukan teks “world”, sehingga keluaran yang dihasilkan adalah 6.

```
<?php
    echo strpos("Hello world!", "world");
?>
```

**Gambar 1.18** PHP

```
<?php
    echo str_replace("world", "Dolly", "Hello world!");
?>
```

**Gambar 1.19** PHP

### **Konstanta / Constant**

Konstanta seperti variabel, tetapi setelah dideklarasikan nilai konstanta tidak bisa dirubah. Konstanta adalah nama atau indintifikasi untuk nilai sederhana. Nilainya tidak bisa diubah selama program berjalan. Nama konstanta yang valid dimulai dengan huruf atau garis bawah, tidak ada tanda \$ sebelum nama konstanta. Konstanta memiliki lingkup global. Gunakan fungsi define(nama, nilai, case-sensitif) untuk mendeklarasikan konstanta. Parameter nama adalah nama dari konstanta yang akan digunakan, parameter nilai adalah nilai dari konstanta yang akan digunakan

pada program, dan parameter case-sensitive digunakan untuk menspesifikasikan apakah nama konstanta harus case-sensitive. Secara default, nilai case-sensitive adalah false.

```
<?php
// nama constant adalah case-sensitive
define("GREETING", "Welcome to W3Schools.com!");
echo GREETING;
?>
```

**Gambar 1.20** Konstanta / Constant

### Operator PHP

Operator digunakan untuk melakukan operasi pada variabelvariabel dan nilai-nilai. Operator dibagi menjadi beberapa kelompok sebagai berikut:

#### Operator-operator aritmatika

Operator aritmatika pada PHP digunakan dengan nilai numeris untuk melakukan operasi aritmatika umum, seperti penjumlahan, pengurangan, perkalian, dan lain-lain.

Operator	Nama	Contoh	Keterangan
+	Penjumlahan	$\$x + \$y$	Nilai x ditambah dengan nilai y
-	Pengurangan	$\$x - \$y$	Nilai x dikurangi dengan nilai y
*	Perkalian	$\$x * \$y$	Kelipatan nilai x sebanyak y
/	Pembagian	$\$x / \$y$	Nilai x dibagi dengan nilai y
%	Modulus	$\$x \% \$y$	Menghasilkan sisa pembagian dari nilai x dengan nilai y
**	Perpangkatan	$\$x ** \$y$	Nilai x pangkat y (baru diperkenalkan pada PHP 5.6)

**Gambar 1.21** Operator

#### Operator-operator assignment (penugasan)

Operator assignment pada PHP digunakan dengan nilai numeris untuk menuliskan nilai pada variabel. Operator assignment dasar adalah “=” yang memiliki maksud menentukan nilai variabel sebelah kiri dengan hasil operasi atau nilai variabel sebelah kanan.

Assignment	Persamaan	Keterangan
$x = y$	$x = y$	Memasukkan nilai variabel y kedalam variabel x
$x += y$	$x = x + y$	Menambahkan nilai variabel x dengan nilai variabel y
$x -= y$	$x = x - y$	Mengurangi nilai variabel x dengan nilai variabel y
$x *= y$	$x = x * y$	Mengalikan nilai variabel x dengan nilai variabel y
$x /= y$	$x = x / y$	Membagi nilai variabel x dengan nilai variabel y
$x \% = y$	$x = x \% y$	Modulus

**Gambar 1.22 Operator**

### Operator-operator perbandingan

Operator perbandingan digunakan untuk membandingkan nilai dua variabel

Operator	Nama	Contoh	Keterangan
'=='	persamaan	$\$x == \$y$	Mengembalikan nilai true jika \$x sama dengan \$y
'==='	Identik	$\$x === \$y$	Mengembalikan nilai true jika \$x sama dengan \$y dan tipe mereka sama

**Gambar 1.23 Operator**

'!=='	Tidak sama	$\$x != \$y$	Mengembalikan nilai true jika \$x tidak sama dengan \$y
'<>'	Tidak sama	$\$x <> \$y$	Mengembalikan nilai true jika \$x tidak sama dengan \$y
'!=='	Tidak identik	$\$x !== \$y$	Mengembalikan nilai true jika \$x tidak sama dengan \$y dan tipe mereka tidak sama
'>'	Lebih besar dari	$\$x > \$y$	Mengembalikan nilai true jika nilai \$x lebih besar dari nilai \$y
'<'	Lebih kecil dari	$\$x < \$y$	Mengembalikan nilai true jika nilai \$x lebih kecil dari nilai \$y
'>='	Lebih besar atau sama dengan	$\$x >= \$y$	Mengembalikan nilai true jika nilai \$x lebih besar atau sama dengan nilai \$y
'<='	Lebih kecil atau sama dengan	$\$x <= \$y$	Mengembalikan nilai true jika nilai \$x lebih kecil atau sama dengan nilai \$y

**Gambar 1.24 Operator**

### Operator-operator iterasi

Iterasi memiliki dua jenis, iterasi penambahan dan iterasi pengurangan. operator iterasi penambahan digunakan untuk

menambah nilai variabel, sedangkan operator iterasi pengurangan digunakan untuk mengurangi nilai variabel.

Operator	Nama	Keterangan
++\$x	Pre-increment	Menambah nilai x dengan satu, kemudian kembalikan nilai x
\$x++	Post-increment	Kembalikan nilai x, kemudian tambah nilai x dengan satu
--\$x	Pre-decrement	Mengurangi nilai x dengan satu, kemudian kembalikan nilai x
\$x--	Post-decrement	Kembalikan nilai x, kemudian kurangi nilai x dengan satu

**Gambar 1.25** Operator

### Operator-operator logika

Operator logika digunakan untuk mengkombinasikan pernyataan kondisional.

Operator	Nama	Contoh	Hasil
and	dan	\$x and \$y	True jika x dan y adalah true
or	atau	\$x or \$y	True jika x atau y adalah true

**Gambar 1.26** Operator

xor	Atau eksklusif	\$x xor \$y	True jika x atau y adalah true, tetapi tidak dua-duanya true.
&&	Dan	\$x && \$y	True jika x dan y adalah true
	atau	\$x    \$y	True jika x atau y adalah true
!	tidak	!\$x	True jika x tidak true

**Gambar 1.27** Operator

### Operator-operator string

Dua operator didesain khusus untuk string

Operator	Nama	Contoh	Keterangan
.	Concatenation	\$text1 . \$text2	Menggabungkan isi dari string text1 dengan isi dari string text2
.=	Concatenation assignment	\$text1 .= \$text2	Membubuhi string pada text2 ke text1

**Gambar 1.28** Operator

## Operator-operator array

Operator array php digunakan untuk membandingkan dua array.

Operator	Nama	Contoh	Keterangan
'+'	Union	$\$x + \$y$	Penggabungan dari x dan y
'=='	Equality	$\$x == \$y$	Mengembalikan nilai true jika x dan y memiliki pasangan nilai yang sama
'==='	Identity	$\$x === \$y$	Mengembalikan nilai true jika x dan y memiliki pasangan nilai yang sama dengan susunan dan tipe yang sama
'!='	Inequality	$\$x != \$y$	Mengembalikan nilai true jika x tidak sama dengan y
'<>'	Inequality	$\$x <> \$y$	Mengembalikan nilai true jika x tidak sama dengan y
'!=='	Non-identity	$\$x !== \$y$	Mengembalikan nilai true jika x tidak identik dengan y

**Gambar 1.29** Operator

### Percabangan dengan if ... else ... elseif

Percabangan atau Pernyataan kondisional digunakan untuk melakukan aksi yang berbeda berdasarkan pada kondisi yang berbeda. dalam PHP terdapat beberapa pernyataan kondisional yaitu sebagai berikut:

Pernyataan if Mengeksekusi beberapa kode jika satu kondisi benar.

Syntax dari kondisi ini adalah sebagai berikut: if (kondisi) { kode yang akan dieksekusi jika kondisi true; }

### Pernyataan If ... else

Mengeksekusi beberapa kode jika kondisi benar dan mengeksekusi kode yang lain jika kondisinya tidak benar. Syntax dari kondisi ini adalah sebagai berikut: if (kondisi) { kode yang akan dieksekusi jika kondisi true.

} else { kode yang akan dieksekusi jika kondisi false. }

### **Pernyataan if ... elseif ... else**

Mengeksekusi kode yang berbeda untuk lebih dari dua kondisi. elseif bisa dibuat lebih dari satu. Syntax dari kondisi ini adalah sebagai berikut:

```
if (kondisi pertama) { kode yang akan
dieksekusi jika kondisi pertama true.
} elseif (kondisi kedua) { kode yang akan
dieksekusi jika kondisi kedua true.
} else { kode yang akan dieksekusi jika
semua kondisi false.
}
```

### **Pernyataan switch**

Pernyataan switch digunakan untuk melakukan aksi yang berbeda berdasarkan kondisi yang berbeda. Pernyataan ini digunakan untuk memilih satu dari banyak blok kode yang akan dieksekusi. Syntax dari pernyataan ini adalah sebagai berikut: switch (n) { case label1: kode yang akan dieksekusi jika n = label1; break; case label2: kode yang akan dieksekusi jika n = label2; break; case label3: kode yang akan dieksekusi jika n = label3; break; ... default: kode yang akan dieksekusi jika n berbeda dari semua label yang ada. }

### **Perulangan**

Perulangan digunakan untuk mengulang eksekusi kode program satu kali atau lebih dari satu kali. Terdapat beberapa macam perulangan yaitu sebagai berikut:

### **Perulangan dengan while**

While loop pada PHP akan mengeksekusi blok kode selama kondisi yang dispesifikasikan benar. Syntax dari while loop adalah sebagai berikut:

```
while (kondisi  
true) { kode yang  
akan dieksekusi. }
```

### **Perulangan dengan do ... while**

Perulangan do ... while akan langsung mengeksekusi blok kode kemudian memeriksa kondisi. Blok kode akan terus dieksekusi selama kondisi true. Syntax untuk perulangan do ... while adalah sebagai berikut:

```
do { kode yang  
akan dieksekusi. }  
while ( kondisi  
true );
```

### **Perulangan dengan for loop**

Perulangan for loop digunakan ketika sudah diketahui lebih lanjut berapa banyak kode harus diulang. Syntax perulangan loop adalah sebagai berikut:

```
for (Inisialisasi; kondisi; peningkatan)  
{ kode yang akan dieksekusi;  
}
```

Parameter:

Inisialisasi : menentukan nilai awal iterasi.

Kondisi : evaluasi terhadap kondisi nilai iterasi, jika kondisi nilai iterasi true, maka perulangan akan dilanjutkan, jika kondisi nilai iterasi false, maka perulangan berhenti.

Penjumlahan : peningkatan nilai iterasi.



## **Perulangan dengan foreach**

Perulangan foreach hanya bekerja pada array, dan digunakan untuk perulangan melalui setiap nilai pada array. Syntax dari perulangan foreach adalah sebagai berikut: `foreach ($array as $variabel) {` kode yang akan dieksekusi;

`}`

Setiap nilai dalam array akan dimasukkan kedalam variabel dalam setiap perulangan.

## **Fungsi**

Kekuatan dari PHP berasal dari fungsi-fungsinya. PHP memiliki lebih dari 1000 fungsi built-in. Namun dalam PHP bisa juga dibuat fungsi sendiri. Fungsi adalah blok kode yang bisa digunakan secara berulang kali dalam program. Fungsi tidak akan dieksekusi secara langsung jika halaman web sedang dimuat. Sebuah fungsi akan dieksekusi oleh fungsi yang memanggilnya. Syntax dari fungsi yang dibuat sendiri adalah sebagai berikut:

`function`

`nama_fungsi() { kode`

`yang akan dieksekusi.`

`}`

Nama fungsi bisa dimulai dengan huruf atau garis bawah (bukan bilangan). Tentukan nama fungsi berdasarkan apa kegunaannya. Informasi dari luar fungsi bisa dimasukkan kedalam fungsi melalui argument. Argumen adalah seperti variabel. Argumen dispesifikasikan setelah nama fungsi didalam kurung. Argument bisa dibuat banyak, antara deklarasi argument harus dipisah dengan tanda koma. Jika argumen memiliki nilai default, maka argument tersebut bisa langsung ditentukan nilainya dengan tanda sama

dengan. Berikut adalah contoh penetapan argument dengan nilai default.

```
<?php
function setHeight($minheight = 50) {
    echo "The height is : $minheight <br>";
}

setHeight(350);
setHeight(); // will use the default value of 50
setHeight(135);
setHeight(80);
?>
```

**Gambar 1.30** Fungsi

Fungsi juga bisa memberikan nilai kembalian. Untuk memberikan nilai kembalian, fungsi tersebut harus menggunakan perintah return. Berikut adalah contoh dari fungsi yang mengembalikan sebuah nilai:

```
<?php
function sum($x, $y) {
    $z = $x + $y;
    return $z;
}

echo "5 + 10 = " . sum(5, 10) . "<br>";
echo "7 + 13 = " . sum(7, 13) . "<br>";
echo "2 + 4 = " . sum(2, 4);
?>
```

**Gambar 1.31** Fungsi

## Array

Array adalah variabel khusus, yang bisa menyimpan banyak nilai pada satu waktu. Nilai dalam array bisa diakses dengan cara merujuk pada indexnya. Pada PHP fungsi array() digunakan untuk membuat sebuah array. Pada PHP, terdapat tiga macam tipe array:

### Array Terindex

Array terindex adalah array dengan index numerik. Ada dua cara untuk membuat array terindex, pertama, array bisa ditentukan secara otomatis dimana indexnya selalu mulai dari 0, contoh :

`$mobil = array("volvo","BMW","Toyota");` dan yang kedua, array bisa ditentukan secara manual, contohnya sebagai berikut:

`$mobil[0] = "Volvo";`

`$mobil[1] = "BMW";`

`$mobil[2] = "Toyota";`

Panjang array atau jumlah element bisa juga dihitung menggunakan fungsi `count()`, untuk menghitung jumlah element pada variabel array `$mobil[]` dapat ditulis `count($mobil)`, fungsi ini akan mengembalikan nilai integer sebagai jumlah element dalam variabel `$mobil[]`.

### **Array Assosiatif**

Array asosiatif adalah array dengan nama kunci yang ditentukan. Ada dua cara membuat array assosiatif, pertama dengan mendeklarasikan ketika memanggil fungsi `array()`, contoh : `$umur = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");` yang kedua dengan mendeklarasikan

nilainya satu-persatu, contoh: `$umur['Peter'] = "35";`

`$umur['Ben'] = "37";`

`$umur['Joe'] = "43";`

Untuk melakukan perulangan pada array assosiatif, dapat menggunakan `foreach($variabel as $kunci => $nilai)` dimana `$variabel` adalah variabel array, `$kunci` adalah variabel yang akan berisi kunci array dan `$nilai` adalah variabel yang berisi nilai dari array.

Untuk melakukan perulangan pada array assosiatif, dapat menggunakan `foreach($variabel as $kunci => $nilai)` dimana `$variabel` adalah variabel array, `$kunci` adalah variabel yang akan berisi kunci array dan `$nilai` adalah variabel yang berisi nilai dari array.

**sort()** - Mengurutkan array dalam susunan ascending. Misalkan fungsi ini digunakan untuk mengurutkan array `$mobil`, maka isi

array \$mobil akan terurut berdasarkan alphabet, yaitu sebagai berikut:

BMW

Toyota

Volvo

**rsort()** - Mengurutkan array dalam susunan descending. Misalkan fungsi ini digunakan untuk mengurutkan array \$mobil, maka urutan nilai pada array mobil akan menjadi sebagai berikut:

Volvo

Toyota

BMW

**asort()** - Mengurutkan array asosiatif dalam susunan ascending, berdasarkan pada nilainya. Misalkan fungsi ini digunakan untuk mengurutkan nilai pada variabel array \$umur, maka urutannya akan berdasarkan nilai pada variabel tersebut. Jika dicetak dengan kuncinya tampilan isi dari array \$umur adalah sebagai berikut:

Peter 35

Ben 37 Joe 43 **ksort()** - Mengurutkan array asosiatif dalam susunan ascending, berdasarkan kuncinya. Misalkan fungsi ini digunakan untuk mengurutkan kunci pada array \$umur, maka kunci pada variabel tersebut akan terurut ascending seperti berikut: Joe 43 Ben 37 Peter 35

**arsort()** - Mengurutkan array asosiatif dalam susunan descending, berdasarkan pada nilainya. Misalkan fungsi ini digunakan untuk mengurutkan data pada variabel \$umur, maka urutan data pada variabel tersebut akan mengikuti susunan nilai dari besar ke kecil.

Joe 43

Ben 37 Peter 35 **krsort()** - Mengurutkan array asosiatif dalam susunan descending, berdasarkan kuncinya. Misalkan fungsi ini digunakan untuk mengurutkan data pada variabel array \$umur,

maka susunannya akan mengurutkan kunci dari besar akhir pada alphabet ke awal pada alphabet.

Peter 35

Joe 43

Ben 37

### **Array Multidimensi**

Array multidimensi adalah array yang mengandung satu atau banyak array. Dimensi dari sebuah array mengindikasikan jumlah index yang diperlukan untuk memilih element. Array dua dimensi memerlukan dua index untuk memilih sebuah element, array tiga dimensi memerlukan tiga index untuk memilih sebuah element.

<b>Name</b>	<b>Stock Sold</b>	
Volvo	22	18
BMW	15	13
Saab	5	2
Land Rover	17	15

**Gambar 1.32** Multidimensi

Misalkan array dua dimensi digunakan untuk menyimpan data-data pada tabel diatas, maka contoh kode program untuk menyimpan data-data tersebut adalah sebagai berikut:

```
$cars = array(  
array("Volvo",22,18),  
array("BMW",15,13),  
array("Saab",5,2),  
array("Land  
Rover",17,15)  
);
```

### **Variabel Global dan Superglobal**

Superglobal diperkenalkan pada PHP 4.1.0 dan merupakan variabel built-in yang selalu ada pada semua scope atau lingkup. Beberapa variabel yang didefinisi sebelumnya dalam PHP adalah “superglobal”, yang berarti bahwa mereka selalu bisa diakses tanpa

terpengaruh oleh lingkup atau scope sehingga bisa diakses oleh semua fungsi atau class tanpa harus melakukan suatu teknik khusus.

Variabel PHP superglobal adalah sebagai berikut:

### **PHP \$GLOBALS**

\$GLOBALS adalah sebuah variabel PHP yang super global yang digunakan untuk mengakses variabel dari mana saja didalam script PHP, termasuk dari dalam fungsi atau method. PHP menyimpan semua variabelvariabel global dalam sebuah array \$GLOBALS[index]. Index adalah nama dari variabel. Contoh berikut menunjukkan cara menggunakan variabel \$GLOBALS:

```
<?php
$x = 75;
$y = 25;
function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z;
?>
```

**Gambar 1.33 PHP**

### **\$\_SERVER \$\_**

SERVER adalah variabel super global yang menyimpan informasi tentang header-header, path-path (jalur), dan skrip lokasi. Berikut adalah contoh penggunaan variabel global \$\_SERVER:

```

<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>

```

**Gambar 1.34 PHP**

Tabel berikut adalah daftar dari elemen-elemen penting yang bisa digunakan pada variabel \$\_SERVER:

**Elemen / Kode**

```

$_SERVER['PHP_SELF']
$_SERVER['GATEWAY_INTERFACE']
$_SERVER['SERVER_ADDR']
$_SERVER['SERVER_NAME']
$_SERVER['SERVER_SOFTWARE']
$_SERVER['SERVER_PROTOCOL']
$_SERVER['REQUEST_METHOD']
$_SERVER['REQUEST_TIME']
$_SERVER['QUERY_STRING']
$_SERVER['HTTP_ACCEPT']
$_SERVER['HTTP_ACCEPT_CHARSET']
$_SERVER['HTTP_HOST']
$_SERVER['HTTP_REFERER']
$_SERVER['HTTPS']
$_SERVER['REMOTE_ADDR']
$_SERVER['REMOTE_HOST']
$_SERVER['REMOTE_PORT']
$_SERVER['SCRIPT_FILENAME']

```

`$_SERVER['SERVER_ADMIN']`  
`$_SERVER['SERVER_PORT']`  
`$_SERVER['SERVER_SIGNATURE']`  
`$_SERVER['PATH_TRANSLATED']`  
`$_SERVER['SCRIPT_NAME']`  
`$_SERVER['SCRIPT_URI']`

### **Deskripsi**

Memberikan nama file dari skrip yang sedang dieksekusi.

Memberikan versi dari CGI (Common Gateway Interface) yang digunakan oleh server. Memberikan alamat IP dari server host.

Memberikan nama dari server host. Misal : `www.umrah.ac.id`

Memberikan string identifikasi server seperti `Apache/2.2.24`

Memberikan nama dan revisi dari protokol informasi (seperti `HTTP/1.1`) Memberikan method request yang digunakan untuk mengakses halaman (seperti `POST`)

Memberikan waktu atau timestamp dari permulaan request (seperti `1377687496`)

Memberikan string query jika halaman diakses via string query.

Memberikan header penerima dari request saat ini.

Memberikan header `Accept_Charset` dari request saat ini (seperti `utf-`

`8,ISO-8859-1`)

Memberikan header host dari permintaan atau request saat ini.

Memberikan URL lengkap dari halaman saat ini (kurang handal karena tidak semua useragents mendukung) Skrip yang di-query melalui protokol keamanan HTTP.

Memberikan alamat IP user yang sedang mengakses halaman saat ini.

Memberikan nama host dari user yang mengakses halaman saat ini.

Memberikan port yang sedang digunakan pada komputer user untuk berkomunikasi pada server web.

Memberikan nama path / jalur yang sedang mengeksekusi skrip.



Memberikan nilai yang diberikan untuk yang menunjuk pada file konfigurasi server web (jika skrip berjalan pada host virtual, ia akan menjadi nilai yang ditentukan untuk virtual host) (seperti seseorang@w3schools.com)

Memberikan port pada komputer server yang sedang digunakan oleh server web untuk komunikasi (seperti port 80)

Memberikan versi server dan nama virtual host yang ditambahkan pada halaman yang dibangkitkan oleh server.

Memberikan sistem file yang berdasarkan pada jalur/path ke skrip saat ini.

Memberikan jalur / path dari skrip saat ini.

Memberikan URI dari halaman saat ini.

## **\$\_REQUEST**

\$\_REQUEST PHP digunakan untuk mengumpulkan data setelah submit sebuah form HTML. Contoh berikut menunjukkan sebuah form dengan field input dan tombol submit. Ketika user melakukan submit data dengan menekan “Submit”, data form dikirimkan untuk file yang dispesifikasikan dalam atribut aksi dari tag

```
<html>
  <body>
    <form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
      Name: <input type="text" name="fname">
      <input type="submit">
    </form>
    <?php
      if ($_SERVER["REQUEST_METHOD"] == "POST") {
        // mengumpulkan nilai input field
        $name = $_REQUEST['fname'];
        if (empty($name)) {
          echo "Name is empty";
        }
      }
    </?php>
  </body>
</html>
```

**Gambar 1.35** Request

```
        } else {  
            echo $name;  
        }  
    }  
    ?>  
</body>  
</html>
```

**Gambar 1.36** Request

## **\$\_POST**

\$\_POST PHP sangat banyak digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan method="post". \$\_POST juga banyak digunakan untuk melewati variabel. Contoh berikut menunjukkan sebuah form dengan sebuah input field dan sebuah tombol submit. Ketika user melakukan submit data dengan menekan tombol submit, data form dikirim ke file yang dispesifikasikan dalam atribut aksi dari tag Top of Form Pada contoh ini, perhatikan file untuk pemrosesan data. Jika ingin menggunakan file PHP lain untuk memproses data form, ganti dengan nama file yang diinginkan. Kemudian, gunakan variabel super global \$\_POST untuk mengumpulkan nilai field input.

```
<html>  
  <body>  
    <form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">  
      Name: <input type="text" name="fname">  
      <input type="submit">  
    </form>  
    <?php  
      if ($_SERVER["REQUEST_METHOD"] == "POST") {  
        // collect value of input field  
        $name = $_POST['fname'];  
        if (empty($name)) {  
          echo "Name is  
empty"; } else {  
          echo $name;  
        }  
      }  
    ?>  
  </body>  
</html>
```

**Gambar 1.37** Post

## **\$\_GET**

\$\_GET PHP bisa digunakan untuk mengumpulkan data form setelah submit sebuah form HTML dengan method="get". Juga bisa digunakan untuk mengumpulkan data yang dikirimkan dalalam URL. Asumsikan halaman HTML yang mengandung hyperlink dengan parameter berikut:

```
<html>
  <body>
    <a href="test_get.php?subject=PHP&web=umrah.ac.id">Test $GET
  </a> </body>
</html>
```

**Gambar 1.37** Get

kemudian user meng-klik link "Test \$GET", parameter "subject" dan "web" dikirimkan ke skrip "test\_get.php", nilai dari kedua parameter tersebut bisa diakses dengan \$\_GET. Contoh berikut adalah skrip yang bisa mengakses nilai dari dua parameter tersebut.

```
<html>
  <body>
    <?php
      echo "Study " . $_GET['subject'] . " at " . $_GET['web'];
    ?>
  </body>
</html>
```

**Gambar 1.38** Get

## BAB II

### PEMBAHASAN

Buatlah program web untuk mengolah nama-nama anggota keluarga anda dengan kriteria sebagai berikut:

1. Nama-nama disimpan didalam variabel array.
2. Program menampilkan jumlah kata dan jumlah huruf pada nama yang anda tuliskan.
3. Program menampilkan kebalikan dari nama yang diinputkan.
4. Program menampilkan jumlah konsonan dan jumlah vokal pada nama yang telah diinputkan.

#### *Source Code*

1	<!DOCTYPE html>
2	<html>
3	<body>
4	<div class="judul" style="text-align: center;">
5	<h2> Pemrograman Web Modul 1 </h2>
6	<p><b>Satria Shano</b></p>
7	<p><b>11191068</b></p>
8	</div>
9	</body>
10	</html>
11	<?php
12	\$listnama= ["Sutrisno", "Mirsa Handayani", "Satria Shano", "Nabila Salsabila Shano"];
13	function hurufkonsonan(\$nama) {
14	\$all = strlen(\$nama);
15	\$a = substr_count(\$nama, "a");
16	\$i = substr_count(\$nama, "i");
17	\$u = substr_count(\$nama, "u");
18	\$e = substr_count(\$nama, "e");
19	\$o = substr_count(\$nama, "o");
20	
21	\$jumlah = \$all - (\$a + \$i + \$u + \$e + \$o);
22	
23	return \$jumlah;
24	}

```

25 function hurufvocal($nama) {
26     $a = substr_count($nama, "a");
27     $i = substr_count($nama, "i");
28     $u = substr_count($nama, "u");
29     $e = substr_count($nama, "e");
30     $o = substr_count($nama, "o");
31
32     $jumlah = ($a + $i + $u + $e + $o);
33
34     return $jumlah;
35 }
36 ?>
37 <?php foreach ($listnama as $listnama) : ?>
38     <?php
39     echo "<ul>";
40     echo "<li> Nama : $listnama </li>";
41     echo "<li> Jumlah Kata : " .str_word_count($listnama) . "</li>";
42     echo "<li> Jumlah Huruf : " .strlen($listnama);
43     echo "<li> Kebalikan : " .strrev($listnama);
44     echo "<li> Jumlah Huruf Konsonan : " .hurufkonsonan($listnama);
45     echo "<li> Jumlah Huruf Vocal : " .hurufvocal($listnama);
46     echo "</ul>";
47     ?>
48 <?php endforeach; ?>

```

Pada soal nomor 1 kita diminta untuk menyimpan nama-nama didalam sebuah variable array. Disini saya membuat variable array dengan nama "listnama" seperti pada line nomor 12 diatas.

```

$listnama= ["Sutrisno", "Mirsa Handayani", "Satria Shano", "Nabila S
alsabila Shano"];

```

Pada soal nomor 2 kita diminta untuk membuat program yang menampilkan jumlah kata dan jumlah huruf pada nama yang dituliskan didalam list. Untuk menampilkan jumlah kata-nya, kita gunakan fungsi .str\_word\_count dan untuk menampilkan jumlah huruf-nya kita gunakan fungsi .strlen() seperti pada line 41 dan 42 diatas. Untuk memasukkannya kedalam program kita, panggil variable array yang kita buat sebelumnya.

```

echo "<li> Jumlah Kata : " .str_word_count($listnama) . "</li>";
echo "<li> Jumlah Huruf : " .strlen($listnama);

```

Pada soal nomor 3 kita diminta untuk menampilkan kebalikan dari nama yang diinputkan. Untuk menampilkan kebalikan dari kata kita bisa menggunakan fungsi `.strrev()` seperti pada line 43 diatas. Untuk memasukkannya kedalam program kita, panggil variable array yang kita buat sebelumnya.

```
echo "<li> Kebalikan : " .strrev($listnama);
```

Pada soal nomor 4 kita diminta untuk menampilkan jumlah huruf konsonan dan huruf vocal pada nama yang kita inputkan. Untuk menampilkan jumlah huruf konsonan kita harus membuat fungsi baru dengan nama fungsi “hurufkonsonan” seperti pada line 13 diatas.

```
function hurufkonsonan($nama) {  
    $all = strlen($nama);  
    $a = substr_count($nama, "a");  
    $i = substr_count($nama, "i");  
    $u = substr_count($nama, "u");  
    $e = substr_count($nama, "e");  
    $o = substr_count($nama, "o");  
  
    $jumlah = $all - ($a + $i + $u + $e + $o);  
  
    return $jumlah;  
}
```

Fungsi ini mengambil semua huruf yang ada pada nama dengan menggunakan fungsi `strlen()`, lalu selanjutnya mengurangi atau membuang huruf a, i, u, e, o yang ada didalamnya dengan menggunakan fungsi `substr_count` dimana fungsi ini menghitung satu huruf secara spesifik. Setelah itu buat variable jumlah yang berisi rumus `all` (semua huruf) dikurang huruf (`a + i + u + e + o`) lalu setelah itu return jumlah untuk menampilkan hasil akhirnya.

```
echo "<li> Jumlah Huruf Konsonan : " .hurufkonsonan($listnama);
```

jangan lupa panggil fungsi yang telah kita buat tadi dan masukkan listnama seperti pada line nomor 44 diatas untuk menampilkan kedalam halaman web.

Sama hal-nya untuk menampilkan jumlah huruf vocal kita harus membuat fungsi baru seperti pada huruf konsonan diatas, untuk code nya hampir sama hanya saja pada rumus jumlah kita langsung menghitung total dari jumlah huruf konsonan yang ada pada kata/kalimat.

```
function hurufvocal($nama) {  
    $a = substr_count($nama, "a");  
    $i = substr_count($nama, "i");  
    $u = substr_count($nama, "u");  
    $e = substr_count($nama, "e");  
    $o = substr_count($nama, "o");  
  
    $jumlah = ($a + $i + $u + $e + $o);  
  
    return $jumlah;  
}
```

jangan lupa panggil fungsi yang telah kita buat tadi dan masukkan listnama seperti pada line nomor 45 diatas untuk menampilkan kedalam halaman web.

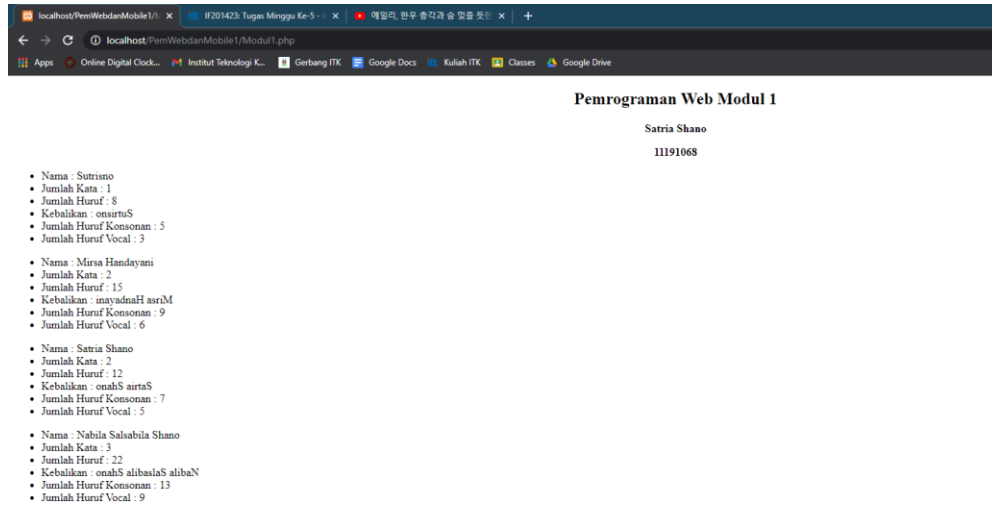
```
echo "<li> Jumlah Huruf Vocal : " .hurufvocal($listnama);
```

```
<!DOCTYPE html>  
<html>  
    <body>  
        <div class="judul" style="text-align: center;">  
            <h2> Pemrograman Web Modul 1 </h2>  
            <p><b>Satria Shano</b></p>  
            <p><b>11191068</b></p>  
        </div>  
    </body>  
</html>  
<?php
```

Untuk line nomor 1 hingga 11 diatas adalah format utama untuk membuat file format HTML didalam PHP. Kode diatas ditulis untuk membuat Judul, Nama, dan NIM.

```
<?php foreach ($listnama as $listnama) : ?>  
    <?php  
<?php endforeach; ?>
```

Untuk line nomor 37,38 dan 48 adalah fungsi perulangan dengan foreach(), fungsi ini hanya bisa digunakan untuk tipe data array. Fungsi ini berjalan sebanyak setiap masing-masing data yang ada didalam array. Untuk data diatas ada 4 data, maka akan dijalankan sebanyak 4 kali.



**Gambar 2.1** Output tampilan di localhost



### **BAB III**

### **KESIMPULAN**

PHP merupakan suatu bahasa pemrograman yang dapat disematkan kedalam HTML dalam pembuatan sebuah website. PHP merupakan bahasa pemrograman tingkat atas yang memiliki berbagai macam fitur untuk menunjang aktivitas pembangunan website yang sifatnya server-base. PHP juga memiliki berbagai macam operator serta fungsi yang bisa digunakan untuk mengolah data yang kita masukkan secara cepat dan akurat. Dengan menggunakan PHP kita dapat dengan mudah membuat sebuah website yang menarik juga dinamis.

## **DAFTAR PUSTAKA**

Praktikum, K. (n.d.). *MODUL PRAKTIKUM PEMROGRAMAN WEB I*  
*Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.*

# LAMPIRAN

