

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB & APLIKASI I**



NAMA : Muhammad Ilyas Abdullah
NIM : 185110500111013
KELAS : C (Permata Merdeka)
MODUL : 1

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKARAYA
2021**

BAB 1

TUJUAN DAN LANDASAN TEORI

1. Tujuan Praktikum

- 1.1. Memahami fungsi dasar dari bahasa pemrograman PHP serta fungsi-fungsi umum yang ada di dalamnya
- 1.2 Mampu membuat program sederhana menggunakan bahasa pemrograman PHP

2. Landasan Teori

Script PHP bisa diletakkan di manapun dalam dokumen, dimulai dengan tanda “<?php” dan diakhiri 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 Word”.

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
    <title>hello word</title>
</head>
<body>

<?php
echo "hello word";
?>

</body>
</html>
```

Gambar 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 alpha-numeric 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 = "php"

echo(i LOVE $txt)

?>

echo "I love " . $txt . "!"; // menghasilkan keluaran
yang sama dengan perintah cetak diatas

?>
```

Gambar 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 di luar fungsi memiliki lingkup global, sedangkan variabel yang dideklarasikan di dalam fungsi memiliki lingkup lokal yang hanya bisa diakses di dalam 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
    function myTest() {
        echo "<p> variabel x di dalam fungsi adalah :
$x</p>";
    }

    myTest();

    echo "<p> variabel x di dalam fungsi adalah : $x</p>";
?>
```

Gambar 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 4. PHP

```
<?php  
$x = 5;  
$y = 10;  
  
function myTest() {  
    global $x, $y;  
    $y = $x + $y;  
}  
  
myTest();  
echo $y; // output 15  
?>
```

Gambar 5. PHP

```
myTest();  
echo $y; // output 15  
?>  
  
<?php  
$x = 5;
```

```

$y = 10;

function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}

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

```

Gambar 6. PHP

```

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

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

```

Gambar 7. PHP

Echo and 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 marginal 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 8. PHP

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

String adalah susunan dari karakter-karakter seperti “Hello world!”. String dituliskan di dalam 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 9. PHP

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 10. PHP

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)
?>
```

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 dideklarasikan secara eksplisit. Pertama kita harus mendeklarasikan class dari object. Class adalah struktur yang bisa memiliki properties dan method. Berikut contohnya:

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:

Resource

Tipe khusus resource bukanlah merupakan tipe data yang sebenarnya. Ia menyimpan referensi fungsi dan sumber daya 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:

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:

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 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 juga memiliki fungsi untuk mengganti teks didalam string, yaitu dengan menggunakan fungsi `str_replace()`. Contoh berikut adalah kode program yang digunakan untuk mengganti teks “world” pada string “Hello world!” dengan teks “Dolly”.

Konstanta / Constant

onstanta 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.

Operator PHP

Operator digunakan untuk melakukan operasi pada variabel-variabel 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-operator assignment (penugasan)

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)

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.

Operator-operator perbandingan

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

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

'!=='	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

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

Operator-operator logika

Operator logika digunakan untuk mengkombinasikan pernyataan kondisional.

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

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



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

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

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. Syntaks 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-ulang 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 ke dalam fungsi melalui argument. Argument adalah seperti variabel. Argument dispesifikasikan setelah nama fungsi didalam kurung. Argument bisa dibuat banyak, antara deklarasi argument harus dipisah dengan tanda koma. Jika argument memiliki nilai default, maka argument tersebut bisa langsung ditentukan nilainya dengan tanda sama dengan. Berikut adalah contoh penentuan argument dengan nilai default.

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:

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 Asosiatif

Array asosiatif adalah array dengan nama kunci yang ditentukan. Ada dua cara membuat array asosiatif, 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 asosiatif, 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.

Elemen dalam sebuah array bisa diurutkan dalam susunan alfabet atau numeris, descending atau ascending. Berikut adalah fungsi-fungsi yang bisa digunakan untuk mengurutkan elemen-elemen dalam array: `sort()` - Mengurutkan array dalam susunan 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 assosiatif 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 elemen, array tiga dimensi memerlukan tiga index untuk memilih sebuah element.

Name Stock Sold

Volvo 22 18

BMW 15 13

Saab 5 2

Land Rover 17 15

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 di dalam script PHP, termasuk dari dalam fungsi atau method. PHP menyimpan semua variabel-variabel global dalam sebuah array \$GLOBALS[index]. Index adalah nama dari variabel. Contoh berikut menunjukkan cara menggunakan variabel \$GLOBALS:

\$_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:

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']`

`$_REQUEST`

`$_REQUEST` PHP digunakan untuk mengumpulkan data setelah submit sebuah form HTML. Contoh berikut menunjukkan sebuah form dengan field input dan tombol submit. Ketika user melakukan submit data dengan menekan “Submit”, data form dikirimkan untuk file yang dispesifikasikan dalam atribut aksi dari tag. Dalam contoh ini, kita memperhatikan file ini memproses data form. Jika pengguna ingin menggunakan file PHP lain untuk memproses data form, menggantinya dengan nama file yang dipilih, maka gunakan variabel super global `$_REQUEST` untuk mengumpulkan field nilai input.

`$_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. 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.

\$_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 dalam URL. Asumsikan halaman HTML yang mengandung hyperlink dengan parameter berikut:

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.

BAB 2 PEMBAHASAN

Pertanyaan dalam modul 1:

1. Nama-nama disimpan di dalam 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.

Dari 4 soal yang diberikan di atas, berikut ini adalah program yang dibuat untuk memenuhi penugasan yang diberikan:

```
<?php
/* Buatlah program web untuk mengolah nama-nama anggota
keluarga anda dengan kriteria sebagai berikut:
1. Nama-nama disimpan di dalam 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*/

$keluarga= ['hatta', 'meida', 'furqon', 'vetha',
'keisha'];

function vokal($kata) {

    $a = substr_count($kata, 'a');
    $i = substr_count($kata, 'i');
    $u = substr_count($kata, 'u');
    $e = substr_count($kata, 'e');
    $o = substr_count($kata, 'o');

    $count = ($a+$i+$u+$e+$o);

    return $count;
}

function konsonan($kata) {
    $jumlah = strlen($kata);

    $a = substr_count($kata, 'a');
```



```

        $i = substr_count($kata, 'i');
        $u = substr_count($kata, 'u');
        $e = substr_count($kata, 'e');
        $o = substr_count($kata, 'o');

        $count = $jumlah - ($a+$i+$u+$e+$o);
        return $count;
    }

    ?>

```

```

<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Tugas 1</title>
</head>
<body>
    <p>Praktikum Pemrograman web dan app I</p>
    <p>Muhammad Ilyas Abdullah</p>
    <p>185110500111013</p>

    <?php foreach ($keluarga as $keluarga) : ?>
    <?php
    echo "<br>";
    echo "nama :". $keluarga. "<br>";
    echo "jumlah huruf:". strlen($keluarga). "<br>";

```

```

        echo "jumlah kata:". str_word_count($keluargaa).
        "<br>";
        echo "kebalikan huruf:". strrev($keluargaa). "<br>";
        echo "jumlah vokal:". vokal($keluargaa). "<br>";
        echo "jumlah konsonan:". konsonan($keluargaa). "<br>";
    ?>
<?php endforeach; ?>
</body>
</html>

```

Dalam tugas pertama disebutkan perintah untuk membuat nama-nama yang akan digunakan dalam penugasan lalu disimpan di dalam variabel array. Hal tersebut sudah dilaksanakan seperti yang terlihat di *line* ke-10 dan *line* ke-11 meliputi:

```
$keluarga= ['hatta', 'meida', 'furqon', 'vetha', 'keisha'];
```

Variabel berikut di atas adalah nama keluarga yang akan dimasukkan di dalam array yang ada.

Dalam tugas kedua disebutkan bahwa penulis disuruh untuk membuat program yang mana program tersebut menampilkan jumlah kata dan jumlah huruf pada nama keluarga yang dituliskan di dalamnya. Dalam melaksanakan hal tersebut kita dapat menggunakan perintah `strlen` dan `str_word_count` yang mana kedua perintah tersebut diarahkan ke dalam variabel `$keluargaa` adapun untuk penulisan fungsi dua perintah tersebut dapat kita lihat pada *line* 56 dan *line* 57. Berikut ini adalah *line* 56 dan *line* 57:

```

echo "Jumlah huruf :". strlen ($keluargaa). "<br>";
echo "jumlah kata:". str_word_count($keluargaa). "<br>";

```

Seperti yang sudah dijelaskan di dalam modul, perintah dari dua fungsi di atas adalah untuk menghitung variabel `$keluarga`. Yang mana variabel `$keluarga` disini dibuat menggunakan fungsi `foreach` sehingga diubah menjadi `$keluargaa`. Adapun isi

string dari variabel \$keluarga yakni ['hatta', 'meida', 'furqon', 'vetha', 'keisha'];

Selanjutnya penulisan penugasan yang ketiga yaitu untuk membuat program yang dapat menampilkan kebalikan nama. Yang artinya disini, kita akan membalikan nama pada variabel \$keluargaa. Dalam membuat program tersebut, kita dapat menggunakan fungsi `strrev`. Fungsi `strrev` sendiri adalah fungsi yang umumnya digunakan untuk membalik susunan string yang ada di dalam variabel. Adapun cara untuk menggunakan fungsi `strrev` adalah dengan menggunakan `strrev($keluargaa)`.

Pada penugasan yang keempat, penulis diminta untuk membuat program yang dapat menampilkan jumlah konsonan dan jumlah vokal pada nama yang telah diinputkan. Program di atas dapat dibuat menggunakan fungsi `substr_count` dan `strlen`. Untuk baris kode dari program di atas dapat dilihat melalui program di bawah ini:

```
function vokal($kata) {

    $a = substr_count($kata, 'a');
    $i = substr_count($kata, 'i');
    $u = substr_count($kata, 'u');
    $e = substr_count($kata, 'e');
    $o = substr_count($kata, 'o');

    $count = ($a+$i+$u+$e+$o);

    return $count;
}

function konsonan($kata) {
    $jumlah = strlen($kata);

    $a = substr_count($kata, 'a');
    $i = substr_count($kata, 'i');
    $u = substr_count($kata, 'u');
    $e = substr_count($kata, 'e');
    $o = substr_count($kata, 'o');
```

```
    $count = $jumlah - ($a+$i+$u+$e+$o);  
    return $count;  
}
```

Fungsi dari substr_count sendiri adalah agar dapat menghitung jumlah huruf pada string. Dalam pembuatan program di atas, penulis membuat fungsi \$vokal dan juga fungsi \$konsonan. Adapun logika yang digunakan dalam kedua fungsi di atas adalah dengan menggunakan dua rumus yakni `$count = ($a+$i+$u+$e+$o)` dan `$count = $jumlah - ($a+$i+$u+$e+$o)`. Dalam menghitung jumlah huruf vokal penggunaan (\$a+\$i+\$u+\$e+\$o) sebagai rumus adalah hal yang tepat karena disini hanya akan menghitung huruf aiueo yang ada pada string variabel \$keluarga. Kemudian penggunaan rumus:

`$jumlah - ($a+$i+$u+$e+$o);`

disini sebelumnya penulis membuat statement pada variabel \$jumlah:

`$jumlah = strlen($kata)`

Pengertian dari baris fungsi di atas adalah dengan menghitung jumlah string yang ada secara keseluruhan terlebih dahulu dengan menggunakan strlen kemudian selanjutnya hasil dari strlen akan dikurangi dengan hasil dari penghitungan huruf vokal (\$a+\$i+\$u+\$e+\$o) sehingga menghasilkan jumlah kata konsonan pada variabel yang akan digunakan nantinya (\$keluarga).

Selanjutnya, dalam menampilkan output dari program di atas, penulis menggunakan HTML seperti pada gambar di bawah ini:

```
<!DOCTYPE HTML>  
<html>  
<head>
```

```

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Tugas 1</title>
</head>
<body>
    <p>Praktikum Pemrograman web dan app I</p>
    <p>Muhammad Ilyas Abdullah</p>
    <p>185110500111013</p>

    <?php foreach ($keluarga as $keluargaa) : ?>
    <?php
    echo "<br>";
    echo "nama :". $keluargaa. "<br>";
    echo "jumlah huruf:". strlen($keluargaa). "<br>";
    echo "jumlah kata:". str_word_count($keluargaa).
"<br>";
    echo "kebalikan huruf:". strrev($keluargaa). "<br>";
    echo "jumlah vokal:". vokal($keluargaa). "<br>";
    echo "jumlah konsonan:". konsonan($keluargaa). "<br>";
    ?>
<?php endforeach; ?>
</body>
</html>

```

Disini penulis menggunakan meta charset utf-8 yang merupakan standar *default* saat ini untuk *web browser*. Selanjutnya penulis menggunakan:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Untuk memberikan kenyamanan responsibilitas jika kemudian tampilan web dibuka melalui web mobile. Program yang sudah dibuat kemudian diletakan di dalam body dengan memberikan sedikit informasi terkait penulis

<p>Praktikum Pemrograman web dan app I</p>

<p>Muhammad Ilyas Abdullah</p>

<p>185110500111013</p>

Kemudian untuk memanggil seluruh program yang dibuat, penulis menuliskan di dalam tag <?php dengan rincian menggunakan fungsi *foreach* dan *echo* masing-masing seperti tampilan baris kode di bawah ini:

```
<?php
    echo "<br>";
    echo "nama :". $keluarga. "<br>";
    echo "jumlah huruf:". strlen($keluarga). "<br>";
    echo "jumlah kata:". str_word_count($keluarga).
"<br>";
    echo "kebalikan huruf:". strrev($keluarga).
"<br>";
    echo "jumlah vokal:". vokal($keluarga). "<br>";
    echo "jumlah konsonan:". konsonan($keluarga).
"<br>";
?>
<?php endforeach; ?>
```

Untuk tampilan output dari program yang sudah dibuat di atas dapat dilihat melalui gambar di bawah ini:

Praktikum Pemrograman web dan app I

Muhammad Ilyas Abdullah

185110500111013

nama :hatta
jumlah huruf:5
jumlah kata:1
kebalikan huruf:attah
jumlah vokal:2
jumlah konsonan:3

nama :meida
jumlah huruf:5
jumlah kata:1
kebalikan huruf:adiem
jumlah vokal:3
jumlah konsonan:2

nama :furqon
jumlah huruf:6
jumlah kata:1
kebalikan huruf:qonruf
jumlah vokal:2
jumlah konsonan:4

nama :vetha
jumlah huruf:5
jumlah kata:1
kebalikan huruf:htev
jumlah vokal:2
jumlah konsonan:3

nama :keisha
jumlah huruf:6
jumlah kata:1
kebalikan huruf:ahsieK
jumlah vokal:3
jumlah konsonan:3

KESIMPULAN

Bahasa pemrograman PHP adalah bahasa pemrograman yang banyak digunakan dalam server side dari sebuah website. Dengan lebih dari 1000 fungsi built-in di dalam PHP, kita juga dapat membuat fungsi tersendiri untuk memenuhi

kebutuhan dalam pembuatan sebuah program. Salah satunya adalah pembuatan program yang digunakan dalam penugasan ini yang mana pemahaman terkait dengan manipulasi fungsi dasar merupakan hal yang sangat penting. Oleh karenanya, pembuatan program yang ada, bisa dijadikan awal untuk kemudian memahami lebih dalam terkait dengan bahasa pemrograman PHP.