

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE I



NAMA : Rakhel Cakra Kusumadinata Sera
NIM :193030503062
KELAS : A
MODUL : I (PHP(Hypertext Preprocessor))

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021

BAB I

LANDASAN TEORI

1.1. Tujuan

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

1.2. Landasan Teori

PHP (Hypertext Preprocessor) merupakan bahasa script yang dijalankan pada sisi server (SSS: ServerSide Scripting). Database yang didukung PHP antara lain: MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC. PHP adalah software Open Source, bebas untuk diunduh dan digunakan.

File PHP dapat berisi teks, tag HTML, dan Script. File PHP dikembalikan ke browser dalam bentuk plain HTML. File PHP dapat berekstensi .php, .php3, atau .phtml.

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

Gambar 1.2 PHP

Tipe Data PHP

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

a. 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 1.3 Tipe data

b. 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 1.4 Tipe Data

c. 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 1.5 Tipe Data

d. Boolean

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

e. Object

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 1.6 Object/objek

f. 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 1.7 Nilai Null

g. 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 1.8 Manipulasi String

Panjang string juga bisa diukur perkata menggunakan fungsi `str_word_count()` seperti contoh berikut:

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

Gambar 1.9 Manipulasi String

PHP juga memiliki fungsi yang bisa digunakan untuk membalikkan susunan string, yaitu menggunakan fungsi `strrev()`

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

Gambar 1.10 Manipulasi String

Operator PHP

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

a. 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 1.11 Operator

b. 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 1.12 Operator

c. 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 1.13 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 1.14 Operator

d. 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 1.15 Operator

e. 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 1.16 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 1.17 Operator

f. 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 1.18 Operator

g. 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 1.19 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;
```

```
}
```

a. 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.

```
}
```

b. 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.

```
}
```

c. 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:

a. 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.  
}
```

b. Perulangan dengan do ... while

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

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

c. 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.

d. 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 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 penentuan 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 1.20 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 1.21 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:

a. 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[]`.

b. Array Assosiatif

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.

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.

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

c. 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.

Name	Stock Sold	
Volvo	22	18
BMW	15	13
Saab	5	2
Land Rover	17	15

Gambar 1.22 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)  
);
```


BAB II

PEMBAHASAN

2.1. Tugas

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 huruf pada nama yang anda tuliskan.
3. Program menampilkan kebalikan dari nama yang diinputkan.
4. Program menampilkan jumlah konsonan dari jumlah vokal pada nama yang telah diinputkan.

2.2. Pembahasan

```
$daftarNama= ["Namasia Mey", "Katie M", "Abi aja", "leon thunder", "Agus"];
```

Pertama tabel dibuat di variabel *\$daftarNama*. Untuk menyatakan sebuah variabel harus menggunakan tanda \$ untuk menjadikan ciri khas dari variabel. Untuk menyimpan array hanya perlu menuliskan [], lalu masukkan nilai-nilai pada array.

```
function jumlahTanpaSpasi($kalimat){  
    return strlen($kalimat) - substr_count($kalimat, ' ');  
}
```

```
Jumlah Huruf: <?php echo jumlahTanpaSpasi($name); ?>  
Jumlah Kata: <?php echo str_word_count($name); ?>
```

Untuk menampilkan jumlah kata, perlu untuk memanggil fungsi *jumlahTanpaSpasi*. Untuk parameter yang digunakan yaitu *\$kalimat*. Nilai balik dari fungsi *jumlahTanpaSpasi* yaitu panjang dari kalimat dikurangi jumlah karakter spasi pada kalimat. Untuk menyatakan panjang dari kalimat menggunakan fungsi yang telah disediakan oleh php yaitu *strlen()*. Karena *strlen()* menghitung seluruh karakter yang terbentuk sehingga spasi antar kalimat juga dihitung, maka diperlukan untuk menghitung panjang spasi antar kalimat menggunakan *substr_count()*.

Untuk mencetak menggunakan php, kita perlu menggunakan *echo* diikuti perintah yang diinginkan. Menghitung banyak kata, hanya perlu memanggil fungsi string dari php yaitu *str_word_count()*.

```
Kebalikan Nama: <?php echo strrev($name); ?>
```

Untuk kebalikan nama, php telah menyediakan fungsi string untuk membalikkan string yaitu *strrev(\$nama_string)*.

```
function Konsonan($kalimat) {
return strlen($kalimat) - (
    substr_count($kalimat, 'a')
    + substr_count($kalimat, 'i')
    + substr_count($kalimat, 'u')
    + substr_count($kalimat, 'e')
    + substr_count($kalimat, 'o')
);
}

function Vokal($kalimat) {
return
    substr_count($kalimat, 'a')
    + substr_count($kalimat, 'i')
    + substr_count($kalimat, 'u')
    + substr_count($kalimat, 'e')
    + substr_count($kalimat, 'o');
}
```

Untuk menghitung panjang konsonan, maka dibuatlah sebuah fungsi bernama *Konsonan()*. Nilai parameter-nya tetap menggunakan daftar nilai-nilai array yang digunakan. Nilai balik yang diberikan adalah panjang kalimat dikurangi jumlah huruf vokal. Agar dapat menemukan jumlah vokal, menggunakan *substr_count(\$nama_string, karakter)* merupakan cara yg sangat tepat. Hanya perlu memasukkan karakter yang ingin dicari seperti huruf vokal yaitu a, i, u, e, dan o.

Fungsi *Vokal* kurang menggunakan algoritma yang sama dengan fungsi *Konsonan*. Hanya menggunakan *substr_count(\$nama_string, karakter)* dan mencari masing-masing karakter seperti a, i, u, e dan o. Lalu jumlahkan total huruf vokal sehingga menjadi nilai balik.

```
| Vokal: <?php echo Vokal( $name); ?>  
| Konsonan: <?php echo Konsonan( $name); ?>
```

Untuk menampilkan jumlah vokal dan kosonan, perlu untuk memanggil *echo* diikuti fungsinya seperti *Vokal()* dan *Konsonan()*.

Output program:

```
Pemrograman Web dan Mobille I  
Nama: Rakhel Cakra Kusumadinata Sera  
NIM: 193030503062  
Modul I  
Nama: Namasia Mey | Jumlah Huruf: 10 | Jumlah Kata: 2 | Kebalikan Nama: yeM aisamaN | Vokal: 5 | Konsonan: 6  
Nama: Katie M | Jumlah Huruf: 6 | Jumlah Kata: 2 | Kebalikan Nama: M eitaK | Vokal: 3 | Konsonan: 4  
Nama: Abi aja | Jumlah Huruf: 6 | Jumlah Kata: 2 | Kebalikan Nama: aja ibA | Vokal: 3 | Konsonan: 4  
Nama: leon thunder | Jumlah Huruf: 11 | Jumlah Kata: 2 | Kebalikan Nama: rednuht noel | Vokal: 4 | Konsonan: 8  
Nama: Agus | Jumlah Huruf: 4 | Jumlah Kata: 1 | Kebalikan Nama: sugA | Vokal: 1 | Konsonan: 3
```

Gambar 2.1 Output program

BAB III

KESIMPULAN

Penggunaan PHP sangat memudahkan dengan fungsi-fungsinya yang telah disediakan. Pembuatan array di PHP sangat memudahkan, karena tidak diperlukan pembuatan tipe data array. Dan fungsi seperti *substr_count()*, *str_word_count()*, dan *strrev(\$nama_string)* sangat membantu dalam pembuatan sebuah program.

Daftar Pustaka

Dosen Teknik Informatika. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I* Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya. *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

Muhardian, & Ahmad. (2015). *Belajar PHP: 7 Hal yang Harus Kamu Ketahui Tentang Array di PHP*. <https://Www.Petanikode.Com/Php-Array/>.
<https://www.petanikode.com/php-array/>

w3schools. (n.d.). *PHP substr_count() Function*. W3schools. Retrieved March 29, 2021, from https://www.w3schools.com/php/func_string_substr_count.asp

LAMPIRAN

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

Gambar 1.1 PHP

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

Gambar 1.2 PHP

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

Gambar 1.3 Tipe data

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

Gambar 1.4 Tipe Data

```
<?php
  $x = 10.365;
  var_dump($x);
?>
```

Gambar 1.5 Tipe Data

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

Gambar 1.6 Object/objek

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

Gambar 1.7 Nilai Null

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

Gambar 1.8 Manipulasi String

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

Gambar 1.9 Manipulasi String

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

Gambar 1.10 Manipulasi String

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 1.11 Operator

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 1.12 Operator

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 1.13 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 1.14 Operator

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 1.15 Operator

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

Gambar 1.16 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 1.17 Operator

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 1.18 Operator

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 1.19 Operator

```
<?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 1.20 Fungsi

```
<?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 1.21 Fungsi

Name	Stock Sold	
Volvo	22	18
BMW	15	13
Saab	5	2
Land Rover	17	15

Gambar 1.22 Multidimensi

Source Code:

```
<?php
    $daftarNama= ["kevin rui", "jeremia", "evan", "abigael", "pratama"];
    function Konsonan($kalimat) {
    return strlen($kalimat) - (
        substr_count($kalimat, 'a')
        + substr_count($kalimat, 'i')
        + substr_count($kalimat, 'u')
        + substr_count($kalimat, 'e')
        + substr_count($kalimat, 'o')
    );
    }

    function Vokal($kalimat) {
    return
        substr_count($kalimat, 'a')
        + substr_count($kalimat, 'i')
        + substr_count($kalimat, 'u')
        + substr_count($kalimat, 'e')
        + substr_count($kalimat, 'o');
    }

    function jumlahTanpaSpasi($kalimat){
    return strlen($kalimat) - substr_count($kalimat, ' ');
    }
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <p> Pemrograman Web dan Mobbille I </p>
    <p> Nama: Rakhel Cakra Kusumadinata Sera </p>
    <p> NIM: 193030503062 </p>
    <p> Modul I </p>
```

```
<?php foreach ($daftarNama as $name) : ?>
    Nama: <?php echo $name; ?>
    | Jumlah Huruf: <?php echo jumlahTanpaSpasi($name); ?>
    | Jumlah Kata: <?php echo str_word_count($name); ?>
    | Kebalikan Nama: <?php echo strrev($name); ?>
    | Vokal: <?php echo Vokal( $name); ?>
    | Konsonan: <?php echo Konsonan( $name); ?>
    </BR>
<?php endforeach; ?>

</table>

</body>
</html>
```