

**LAPORAN HASIL PRAKTIKUM**  
**PEMWEB DAN MOBILE**



**NAMA : AXEL BERKATI**  
**NIM : 193010503007**  
**KELAS : A**  
**MODUL : 1 (PHP (HYPERTEXT**  
**PREPROCESSOR))**

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

**2021**

# BAB 1

## TUJUAN DAN 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

#### 1.2.1. PHP

PHP Merupakan singkatan recursive dari PHP : Hypertext Preprocessor yang pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994 (Achmad Solichin. S.Kom, 2016). PHP Harus ditulis di antara tag :

- a. `<? dan ?>`
- b. `<?php dan ?>`
- c. `<script language="php">` dan `</script>`
- d. `<% dan %>`

Setiap satu statement (perintah) pada PHP biasanya diakhiri dengan titik-koma (;). CASE SENSITIVE untuk nama identifier yang dibuat oleh user (berupa variable, konstanta, fungsi dll), namun TIDAK CASE SENSITIVE untuk identifier built-in dari PHP. Jadi :

- \$nama ≠ \$Nama ≠ \$NAMA
- hitungLuas() ≠ HitungLuas()
- echo = ECHO
- while = WHILE

#### 1.2.2. Variabel

Berikut merupakan penjelasan variabel pada PHP:

- Digunakan untuk menyimpan sebuah value, data atau informasi
- Nama variabel diawali dengan tanda \$

- Panjang tidak terbatas
- Setelah tanda \$ diawali oleh huruf atau under-score (\_). Karakter berikutnya bisa terdiri dari huruf, angka, dan karakter tertentu yang diperbolehkan (karakter ASCII dari 127 – 255).
- Bersifat case-sensitive.
- Tidak perlu dideklarasikan.
- Tidak boleh mengandung spasi.

Contoh variabel yang benar dan salah :

Tabel 1.1 contoh variabel yang benar dan salah

Benar	Salah
<code>\$_name</code>	<code>\$3name</code>
<code>\$first_name</code>	<code>\$name?</code>
<code>\$name3</code>	<code>\$first+name</code>
<code>\$name_3</code>	<code>\$first.name</code>
-	<code>\$first name</code>

### 1.2.3. Tipe Data

Pada PHP, tipe data variabel tidak didefinisikan oleh programmer, akan tetapi secara otomatis ditentukan oleh interpreter PHP. Namun demikian, PHP mendukung 8 (delapan) buah tipe data primitif, yaitu :

1. Boolean
2. Integer
3. Float
4. String
5. Array
6. Object
7. Resource
8. NULL

#### 1.2.4. Konstanta

Konstanta merupakan variabel konstan yang nilainya tidak berubah-ubah. Untuk mendefinisikan konstanta dalam PHP, menggunakan fungsi `define()`. Konstanta seperti variabel, tetapi setelah dideklarasikan nilai konstanta tidak bisa dirubah. Konstanta adalah nama atau indentifikasi 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.

#### 1.2.5. Operator Dalam PHP

Jenis Operator	Operator	Contoh	Keterangan
<b>Aritmatika</b>	+	$\$a + \$b$	Pertambahan
	-	$\$a - \$b$	Pengurangan
	*	$\$a * \$b$	Perkalian
	/	$\$a / \$b$	Pembagian
	%	$\$a \% \$b$	Modulus, sisa pembagian
<b>Penugasan</b>	=	$\$a = 4;$	\$a diisi dengan 4
<b>Bitwise</b>	&	$\$a \& \$b$	Bitwise AND
		$\$a   \$b$	Bitwise OR
	^	$\$a \wedge \$b$	Bitwise XOR
	~	$\sim \$b$	Bitwise NOT
	<<	$\$a \ll \$b$	Shift Left
	>>	$\$a \gg \$b$	Shift Right
<b>Perbandingan</b>	==	$\$a == \$b$	Sama dengan

	===	\$a === \$b	Identik
	!=	\$a != \$b	Tidak sama dengan
	<>	\$a <> \$b	Tidak sama dengan
	!==	\$a !== \$b	Tidak identik
	<	\$a < \$b	Kurang dari
	>	\$a > \$b	Lebih dari
	<=	\$a <= \$b	Kurang dari sama dengan
	>=	\$a >= \$b	Lebih dari sama dengan
<b>Logika</b>	and	\$a and \$b	TRUE jika \$a dan \$b TRUE
	&&	\$a && \$b	TRUE jika \$a dan \$b TRUE
	or	\$a or \$b	TRUE jika \$a atau \$b TRUE
		\$a    \$b	TRUE jika \$a dan/atau \$b TRUE
	xor	\$a xor \$b	TRUE jika \$a atau \$b TRUE, tapi tidak keduanya
	!	!\$a	TRUE jika \$a FALSE
<b>String</b>	.	\$a . \$b	Penggabungan string \$a dan \$b

### 1.2.6. Komentar Program

Dalam PHP, komentar program bisa menggunakan :

- /\* dan \*/
- // dan
- #

### 1.2.7. Struktur Kondisi

#### a. Struktur Kondisi If :

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

Keterangan :

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

#### b. Struktur Kondisi If ... Else

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

Keterangan :

kondisi *merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya. Jika kondisinya TRUE maka statemen yang berada di blok if akan dieksekusi, sebaliknya jika kondisinya FALSE maka statemen yang berada di blok else yang akan dieksekusi.*

#### c. Struktur Kondisi Khusus ? :

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

Keterangan :

kondisi *merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya. Statement pada blok benar dan salah hanya boleh satu statemen saja*

#### d. Struktur Kondisi Switch ... Case

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

Keterangan :

- \$var *merupakan variabel yang akan diperiksa isi atau nilainya. Tipe data variabel ini tidak dibatasi.*

- *Value pada case juga bisa berupa string, integer, boolean, bahkan bisa berupa conditional-statement. Boleh memakai kutip tunggal maupun kutip ganda.*

### 1.2.8. Struktur Perulangan

#### a. Struktur Perulangan For

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

Keterangan :

- *init\_awal merupakan inisialisasi atau nilai awal variable.*
- *kondisi merupakan statemen kondisi yang akan membatasi perulangan.*
- *counter merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.*

#### b. Struktur Perulangan While

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

Keterangan :

- *init\_awal merupakan inisialisasi atau nilai awal variable.*
- *kondisi merupakan statemen kondisi yang akan membatasi perulangan.*
- *counter merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.*

#### c. Struktur Perulangan Do ... while

```
init_awal; do {
    statement-yang-diulang; counter;
```

```
} while (kondisi);
```

Keterangan :

- *init\_awal merupakan inisialisasi atau nilai awal variable.*
- *kondisi merupakan statemen kondisi yang akan membatasi perulangan.*
- *counter merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.*
- *Pada struktur do...while, pemeriksaan kondisi ada di bawah, sehingga statement yang berada dalam block do...while setidaknya akan dieksekusi sebanyak satu kali.*

#### d. Struktur Perulangan Foreach

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

Keterangan :

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

### 1.2.9. Array

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

### 1.2.10. Fungsi-Fungsi Array Dalam PHP

PHP menyediakan lebih dari 70 fungsi untuk manipulasi array.

#### a. Fungsi Pengurutan Array

- `arsort()` – Pengurutan array berdasarkan value secara descending
- `asort()` – Pengurutan array berdasarkan value secara ascending



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

#### **b. Fungsi Pengaturan Pointer Array**

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

#### **c. Fungsi Pencarian pada Array**

- array\_search() – Mencari posisi (key) dari suatu value dalam array
- array\_key\_exists() – Memeriksa suatu key ada dalam array atau tidak
- in\_array() – Memeriksa suatu elemen ada dalam array atau tidak

### **1.2.11. Fungsi Dalam PHP**

Bentuk umum pendefinisian fungsi dalam PHP :

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

## BAB II

### PEMBAHASAN

```
C:\> xampp > htdocs > Pemweb > index.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Praktikum Modul 1</title>
5  </head>
6  <body>
7
8  <?php
9  $family=["Yuwesli","Murniati Bajenta","Billy Asi Dirgantara","Verenty Eugenia","Axel Berkati"];
10
```

Gambar 2.1 Deklarasi array

program diawali dengan Deklarasi `<!DOCTYPE html>` yang mendefinisikan bahwa dokumen ini adalah dokumen HTML5. Kemudian dilanjutkan dengan Elemen `<html>` yang dimana adalah elemen root dari halaman HTML. Elemen `<body>` mendefinisikan badan dokumen, dan merupakan wadah untuk semua konten yang terlihat, seperti judul, paragraf, gambar, hyperlink, tabel, daftar, dll.

Masuk ke dalam program isi PHP. Program PHP, selalu dimulai dengan `<?php` dan diakhiri dengan `?>`. Pada awal program dideklarasikan variabel array dengan nama `$family`, yang berisi Yuwesli, Murniati Bajenta, Billy Asi Dirgantara Verenty Eugenia, dan Axel Berkati.

```
10
11  foreach($family as $data){
12      $teks=str_replace(" ","",$data); //buang spasi yang ada di tengah kata
13      echo "jumlah kata dari nama $data: ". str_word_count($data) . "<br>";
14      echo "jumlah huruf dari nama $data: ". strlen($teks) . "<br>";
15  }
16
```

Gambar 2.2 Potongan Code Menghitung Jumlah Kata dan Huruf

Pada potongan code diatas dibuat sebuah perulangan foreach, yang memiliki parameter dari array yang telah didefinisikan sebelumnya yaitu `$family` dan variabel `$data` digunakan sebagai variabel perantaranya yang berisi array pada perulangan tersebut. Isi dari perulangan foreach ini, spasi pada isi variabel `$data`

akan dibuang menggunakan perintah `str_replace(" ","",$data)`, yang kemudian pendefinisian ini akan di lempar ke variabel baru yaitu `$teks`. Fungsi `str_replace()` yaitu untuk menggantikan beberapa karakter dengan beberapa karakter lain dalam sebuah string. Pada kasus diatas, “ (spasi) digantikan dengan “”(tanpa spasi) Kemudian digunakan statement `echo` untuk menampilkan hasil penghitungan jumlah kata dengan perintah `str_word_count($data)`. Fungsi `str_word_count()` yaitu untuk menghitung jumlah kata dalam sebuah string. pada baris selanjutnya, dibuat statement `echo`, untuk menampilkan jumlah huruf pada variabel `$teks` dapat menggunakan pemanggilan perintah `strlen($teks)`. Fungsi `strlen()` yaitu untuk mengembalikan panjang string. Perulangan ini akan berakhir setelah semua isi dari array telah dilooping.

```

jumlah kata dari nama Yuwesli: 1
jumlah huruf dari nama Yuwesli: 7
jumlah kata dari nama Murniati Bajenta: 2
jumlah huruf dari nama Murniati Bajenta: 15
jumlah kata dari nama Billy Asi Dirgantara: 3
jumlah huruf dari nama Billy Asi Dirgantara: 18
jumlah kata dari nama Verenty Eugenia: 2
jumlah huruf dari nama Verenty Eugenia: 14
jumlah kata dari nama Axel Berkati: 2
jumlah huruf dari nama Axel Berkati: 11

```

Gambar 2.3 Output dari Potongan Code Menghitung Jumlah kata dan Huruf

```

17
18     foreach($family as $data){
19         echo "Kebalikan dari nama $data: ". strrev($data). "<br>";
20     }

```

Gambar 2.4 Potongan Code Untuk Membalikan Nama

Perulangan `foreach` diatas dibuat agar setiap isi dari variabel array `$family` dapat ditampilkan terbalik. Perulangan `foreach` ini memiliki parameter `$family` dan variabel perantaranya yaitu `$data`. Isi dari perulangan ini yaitu isi dari array `$data` akan di balik menggunakan `strrev`. Fungsi `strrev()` yaitu untuk membalikkan string. Perulangan ini akan berakhir setelah semua isi dari array telah dilooping.

Kebalikan dari nama Yuwesli: ilsewuY  
 Kebalikan dari nama Murniati Bajenta: atnejaB itainruM  
 Kebalikan dari nama Billy Asi Dirgantara: aratnagriD isA ylliB  
 Kebalikan dari nama Verenty Eugenia: aineguE ytnereV  
 Kebalikan dari nama Axel Berkati: itakreB lexA

Gambar 2.5 Output dari Potongan Code Membalikan Nama

```

22
23 function vowel_konsonant($word){
24     $word_len = strlen($word)-1;
25     $vowel_letters = array("a","e","i","o","u");
26
27     $vowel = 0;
28     $consonant = 0;
29
30     for($i = 0; $i <= $word_len; $i++){
31         if(in_array($word[$i], $vowel_letters)){
32             $vowel++;
33         }else{
34             $consonant++;
35         }
36     }
37
38     echo "<br>jumlah huruf vocal : ".$vowel;
39     echo "<br>jumlah huruf konsonan : ".$consonant;
40 }

```

Gambar 2.6 Potongan Code Fungsi vowel\_Konsonant

Pada potongan code diatas, dibuat sebuah function untuk menghitung jumlah huruf vokal dan konsonan, yang diberi nama vowel\_konsonant dengan parameter nya yaitu \$word. Isi dari fungsi ini yaitu, pertama-tama didefinisikan variabel baru yaitu \$word\_len, yang dimana variabel ini berisi strlen(\$word)-1, kemudian dideklarasikan variabel array yaitu \$vowel\_letters yang berisi karakter a,e,i,o,u. Kemudian didefinisikan pada variabel baru yaitu \$vowel dan \$consonant agar sama dengan 0. Pada fungsi ini juga dibuat perulangan for yang memiliki kondisi awal \$i=0, dengan kondisi jika \$i lebih kecil dari variabel \$word\_len, maka variabel \$i akan di increment atau perulangan akan dilanjutkan. Kemudian isi dari perulangan ini terdapat kondisi if yang dimana jika didalam array pada variabel array \$word terdapat isi dari variabel \$vowel\_letters yaitu a,e,i,o,u. Maka variabel \$vowel akan diincrement atau ditambah dengan 1, jika tidak maka \$consonant yang akan diincrement atau dijumlahkan 1. Setelah perulangan telah selesai dilakukan,

maka akan ditampilkan jumlah dari huruf vokal dan konsonant, dengan memanggil variabel \$vowel dan \$consonant.

```

41
42     foreach($family as $data){
43         echo "<br>Menghitung jumlah vokal dan konsonan dari nama $data";
44         $teks=str_replace(" ","",$data); //buang spasi yang ada di tengah kata
45         $teks_modif=strtolower($teks); //ganti kalimat menjadi huruf kecil semua
46
47         vowel_konsonant($teks_modif);
48     }
49     ?>
50 </body>
51 </html>

```

Gambar 2.7 Potongan Code Pemanggilan Fungsi vowel\_konsonant

Pada potongan code diatas dibuat perulangan foreach yang memiliki parameter \$family dan variabel perantaranya yaitu \$data. Isi dari perulangan foreach ini yaitu, pertama-tama isi dari array \$family akan ditampilkan, dengan memanggil variabel \$data. Kemudian, spasi pada isi variabel \$data akan dibuang menggunakan perintah `str_replace(" ","",$data)`, yang kemudian pendefinisian ini akan di lempar ke variabel baru yaitu \$teks.

Kemudian, kalimat pada \$teks akan diubah menjadi kecil dengan cara memanggil `strtolower($teks)`, yang kemudian dilempar ke variabel baru yaitu \$teks\_modif. `strtolower()` ini adalah perintah PHP untuk memanipulasi sebuah string menjadi huruf kecil. Kemudian, dipanggil fungsi yang telah dibuat sebelumnya yaitu `vowel_konsonant($teks_modif)` yang dimana \$teks\_modif ini digunakan sebagai parameter parameter pada fungsi `vowel_konsonant`.

```

Menghitung jumlah vokal dan konsonan dari nama Yuwesli
jumlah huruf vocal : 3
jumlah huruf konsonan : 4
Menghitung jumlah vokal dan konsonan dari nama Murniati Bajenta
jumlah huruf vocal : 7
jumlah huruf konsonan : 8
Menghitung jumlah vokal dan konsonan dari nama Billy Asi Dirgantara
jumlah huruf vocal : 7
jumlah huruf konsonan : 11
Menghitung jumlah vokal dan konsonan dari nama Verenty Eugenia
jumlah huruf vocal : 7
jumlah huruf konsonan : 7
Menghitung jumlah vokal dan konsonan dari nama Axel Berkati
jumlah huruf vocal : 5
jumlah huruf konsonan : 6

```

Gambar 2.8 Output Pemanggilan Fungsi

### **BAB III**

### **KESIMPULAN**

Dari praktikum yang telah dilaksanakan, dapat disimpulkan bahwa PHP adalah bahasa pemrograman yang bisa membuat halaman web bisa diatur dan diolah secara dinamis. PHP juga memiliki aturan menulis tertentu. Penting diketahui, aturan-aturan yang telah ditetapkan sangat kompleks. Meski terkesan rumit, pengembangan web sebenarnya bisa dilakukan siapa saja, baik yang baru menginjak tingkat dasar maupun yang sudah ekspert.

## DAFTAR PUSTAKA

Achmad Solichin. S.Kom. (2016). Pemrograman Web dengan PHP dan MySQL -

Achmad Solichin - Google Buku. *Universitas Budi Luhur, June*, 85.

<https://books.google.co.id/books?id=kcD4BQAAQBAJ&printsec=frontcover&dq=aplikasi+berbasis+web+dengan&hl=id&sa=X&ved=0ahUKEwib-fT80ITYAhVBrI8KHT9GD6QQ6AEIJzAA#v=onepage&q=aplikasi+berbasis+web+dengan&f=false>

Dosen Teknik Informatika. 20121. *Modul Pemrograman WEB dan Moblie.*

Univsersitas Palangkaraya : Jurusan Teknik Informatika Fakultas Teknik.

PHP Tutorial, <https://www.w3schools.com/php/>

## LAMPIRAN

```
index.php X
C: > xampp > htdocs > Pemweb > index.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Praktikum Modul 1</title>
5  </head>
6  <body>
7
8  <?php
9  $family=["Yuwesli","Murniati Bajenta","Billy Asi Dirgantara","Verenty Eugenia","Axel Berkati"];
10
11  foreach($family as $data){
12      $teks=str_replace(" ","",$data); //buang spasi yang ada di tengah kata
13      echo "jumlah kata dari nama $data: ". str_word_count($data) . "<br>";
14      echo "jumlah huruf dari nama $data: ". strlen($teks) . "<br>";
15  }
16
17
18  foreach($family as $data){
19      echo "Kebalikan dari nama $data: ". strrev($data). "<br>";
20  }
```

Gambar 1 program PHP(1)

```
22
23  function vowel_konsonant($word){
24      $word_len = strlen($word)-1;
25      $vowel_letters = array("a","e","i","o","u");
26
27      $vowel = 0;
28      $consonant = 0;
29
30      for($i = 0; $i <= $word_len; $i++){
31          if(in_array($word[$i], $vowel_letters)){
32              $vowel++;
33          }else{
34              $consonant++;
35          }
36      }
37
38      echo "<br>jumlah huruf vocal : ".$vowel;
39      echo "<br>jumlah huruf konsonan : ".$consonant;
40  }
41
42  foreach($family as $data){
43      echo "<br>Menghitung jumlah vokal dan konsonan dari nama $data";
44      $teks=str_replace(" ","",$data); //buang spasi yang ada di tengah kata
45      $teks_modif=strtolower($teks); //ganti kalimat menjadi huruf kecil semua
46
47      vowel_konsonant($teks_modif);
48  }
49  ?>
50  </body>
51  </html>
```

Gambar 2 Program PHP(2)



jumlah kata dari nama Yuwesli: 1  
 jumlah huruf dari nama Yuwesli: 7  
 jumlah kata dari nama Murniati Bajenta: 2  
 jumlah huruf dari nama Murniati Bajenta: 15  
 jumlah kata dari nama Billy Asi Dirgantara: 3  
 jumlah huruf dari nama Billy Asi Dirgantara: 18  
 jumlah kata dari nama Verenty Eugenia: 2  
 jumlah huruf dari nama Verenty Eugenia: 14  
 jumlah kata dari nama Axel Berkati: 2  
 jumlah huruf dari nama Axel Berkati: 11  
 Kebalikan dari nama Yuwesli: ilsewuY  
 Kebalikan dari nama Murniati Bajenta: atnejaB itainruM  
 Kebalikan dari nama Billy Asi Dirgantara: aratnagriD isA ylliB  
 Kebalikan dari nama Verenty Eugenia: aineguE ytnereV  
 Kebalikan dari nama Axel Berkati: itakreB lexA

Gambar 3 Output Program PHP(1)

Menghitung jumlah vokal dan konsonan dari nama Yuwesli  
 jumlah huruf vocal : 3  
 jumlah huruf konsonan : 4  
 Menghitung jumlah vokal dan konsonan dari nama Murniati Bajenta  
 jumlah huruf vocal : 7  
 jumlah huruf konsonan : 8  
 Menghitung jumlah vokal dan konsonan dari nama Billy Asi Dirgantara  
 jumlah huruf vocal : 7  
 jumlah huruf konsonan : 11  
 Menghitung jumlah vokal dan konsonan dari nama Verenty Eugenia  
 jumlah huruf vocal : 7  
 jumlah huruf konsonan : 7  
 Menghitung jumlah vokal dan konsonan dari nama Axel Berkati  
 jumlah huruf vocal : 5  
 jumlah huruf konsonan : 6

Gambar 4 Output Program PHP(2)