

**LAPORAN PRAKTIKUM  
PENGEMBANGAN WEB DAN MOBILE**



**NAMA : DZIKRI AHMADILLAH  
NIM : 193020503039  
KELAS : A  
MODUL : I (PHP (Hypertext Preprocessor))**

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

# **BAB I**

## **TUJUAN DAN LANDASAN TEORI**

### **1.1 Tujuan**

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

### **1.2 Landasan Teori**

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).(Andre, 2019)

#### **1.2.1 Sintaks Dasar**

Ketika PHP menerima suatu file, maka PHP akan mencari tags pembuka (“<?php”) dan tags penutup (“?”), ini juga berarti sebagai pemberitahuan agar PHP mulai menterjemahkan baris-baris kode PHP tersebut, serta mengakhiri penterjemahannya sampai bertemu dengan tags penutup. Baris-baris kode yang berada diluar pasangan penanda “<?php ... ?>” akan diabaikan (tidak diterjemahkan) oleh PHP.(Suprianto, 2008)

Seperti bahasa C atau Perl, PHP membutuhkan penghentian baris pernyataan, serta memisahkan antara baris satu dengan baris lain dengan cara memberi tanda titik koma (“;”) diakhir setiap baris pernyataan.

#### **1.2.2 Komentar dan Remarks**

PHP mendukung pemberian komentar seperti yang ada di 'C', 'C++' dan Unix shell-style (Perl style). Terkadang dibutuhkan beberapa baris kalimat untuk memberi keterangan pada suatu baris program, hal ini sering disebut

remarks. Dimana remarks tidak akan ikut dieksekusi oleh server. Jika remarks hanya satu baris maka gunakanlah (“/”) atau (“#”), jika remarks terdiri dari beberapa baris secara berurutan lebih efektif jika menggunakan (“/\*...\*/”).(Suprianto, 2008)

### **1.2.3 Variabel**

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.(‘MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya’, no date)

### **1.2.4 Tipe Data**

PHP mendukung delapan jenis tipe data, antara lain:

1. Tipe scalar, tipe scalar tidak bisa dipecah lagi menjadi bagian yang lebih kecil, boleh dikatakan merupakan tipe dasar.
  - a. Boolean, mempresentasikan dua kemungkinan keadaan: TRUE atau FALSE.
  - b. Integer, atau bilangan bulat
  - c. Float (floating point, „double“), bilangan dengan tanda desimal atau bilangan dalam bentuk eksponensial.
  - d. String, Mempresentasikan susunan dari karakter-karakter.
2. Tipe compound / Tipe campuran
  - a. Array, variabel yang bisa menyimpan banyak nilai pada satu waktu.
  - b. Object, sebuah tipe data yang menyimpan data dan informasi bagaimana memproses data tersebut.
3. Tipe special / Tipe khusus
  - a. Resource, Tipe khusus resource bukanlah merupakan tipe data yang sebenarnya. Ia menyimpan referensi fungsi dan sumberdaya eksternal untuk PHP.

- b. Null, tipe data khusus yang bisa memiliki hanya satu nilai: NULL. Sebuah variabel dengan tipe data NULL adalah variabel yang tidak memiliki nilai. (*PHP Tutorial*, no date)

### 1.2.5 Operator

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

#### 1. Operator Aritmatika

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

**Tabel 1.1** Operator Aritmatika

Operator	Nama	Contoh	Keterangan
+	Penjumlahan	$\$x + \$y$	Nilai x ditambah dengan nilai y
-	Pengurangan	$\$x - \$y$	Nilai x dikurang 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

#### 2. Operator Assignment

Operator assignment pada PHP digunakan dengan nilai numeris untuk menuliskan nilai pada variabel.

**Tabel 1.2** Operator Assignment

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

### 3. Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan nilai dua variabel.

**Tabel 1.3** Operator Perbandingan

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		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 dari atau sama dengan	$x >= y$	Mengembalikan nilai true jika nilai \$x lebih besar atau sama dengan nilai \$y
'<='	Lebih kecil dari atau sama dengan	$x <= y$	Mengembalikan nilai true jika nilai \$x lebih kecil atau sama dengan nilai \$y

### 4. Operator Iterasi

Iterasi memiliki dua jenis, iterasi penambahan dan iterasi pengurangan.

**Tabel 1.4** Operator Iterasi

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

## 5. Operator Logika

Operator logika digunakan untuk mengkombinasikan pernyataan kondisional

**Tabel 1.5** Operator Logika

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

## 6. Operator String

Dua operator didesain khusus untuk string.

**Tabel 1.6** Operator 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

### 1.2.6 Percabangan

Pernyataan If merupakan percabangan bersyarat yang berfungsi untuk melewati suatu proses, jika syarat terpenuhi maka akan dilakukan proses yang berikutnya. Struktur if...else memiliki dua alur percabangan, sehingga mempunyai alternatif. If...else bisa menuju ke proses berikutnya walaupun syarat terpenuhi atau tidak terpenuhi.(Suprianto, 2008)

```
If (syarat) {  
    Instruksi;  
} else {  
    instruksi;  
}
```

**Gambar 1.1** Struktur penggunaan If...else

Selain Menggunakan If..else, terdapat pula sintaks 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.  
}
```

**Gambar 1.2** Struktur penggunaan switch case

### 1.2.7 Perulangan

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

#### 1. WHILE

Perulangan while akan melakukan pengecekan syarat di awal blok perulangan. Selama syarat bernilai benar maka perulangan terus berlanjut dan sebaliknya jika syarat bernilai salah maka perulangan akan dihentikan.

```
while (syarat){  
    instuksi;  
    instuksi;  
}
```

**Gambar 1.3** Struktur penggunaan WHILE

## 2. DO-WHILE

Perulangan do-while merupakan kebalikan dari perulangan for dan while, karena perulangan do-while melakukan pengecekan terhadap syarat diakhir blok perulangan. Jadi do-while paling sedikit akan melakukan satu kali proses perulangan.

```
do{  
    instruksi;  
    instruksi;  
} while (syarat)
```

**Gambar 1.4** Struktur penggunaan DO-WHILE

## 3. FOR

Struktur kontrol For lebih cocok untuk perulangan dengan jumlah pencacah yang pasti atau sudah diketahui.

```
do{  
    instruksi;  
    instruksi;  
} while (syarat)
```

**Gambar 1.5** Struktur penggunaan DO-WHILE

### 1.2.8 Menampilkan Output

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.

### **1.2.9 Fungsi**

Fungsi adalah sekumpulan baris-baris program yang merupakan serangkaian perintah-perintah program yang disusun sedemikian rupa sehingga bisa menjadi satu module saja. Ketika programmer membuat program yang cukup rumit/kompleks maka perlu digunakan fungsi, karena fungsi akan sangat membantu menyederhanakan penulisan program. (Suprianto, 2008)

Adapun manfaat adanya fungsi adalah sebagai berikut :

- a) Mengurangi penulisan program yang sama.

Jika kita sering menggunakan program-program yang sama, maka kita dapat membuatnya menjadi suatu module fungsi, sehingga dengan fungsi tersebut bisa dipanggil di tempat lain tanpa membuat ulang baris program tersebut, dengan demikian program yang dibuat akan menjadi lebih pendek.

- b) Kemudahan untuk melacak dan memperbaiki program.

Dengan fungsi akan sangat mudah dalam melacak dan memperbaiki baris program, karena jika ada kesalahan tidak perlu untuk memeriksa isi program secara keseluruhan tetapi cukup dilihat per bagian fungsi saja.

- c) Bisa dipanggil dimana saja termasuk di fungsi yang lain.

Karena fungsi bisa dipanggil di fungsi yang lain maka program yang kompleks bisa dibuat menjadi lebih sederhana dan mudah untuk dipahami.

Dalam segi pembuatan, fungsi terbagi dua :

#### **1. Fungsi Built-in yang disediakan PHP.**

Pemrogram tinggal memanggil nama fungsinya saja dan siap digunakan. Seluruh isi module fungsi sudah dibuatkan oleh PHP secara Built-in. berikut beberapa contoh dari fungsi yang telah disediakan PHP :

```
<?php
    echo strlen($string);
?>
```

**Gambar 1.6** Fungsi *strlen* yang digunakan untuk menghitung panjang sebuah string

```
<?php
    echo str_word_count($string);
?>
```

**Gambar 1.7** Fungsi *str\_word\_count* yang digunakan untuk menghitung banyak kata pada sebuah string

```
<?php
    echo strrev($string);
?>
```

**Gambar 1.8** Fungsi *strrev* yang digunakan untuk membalik urutan karakter pada sebuah string

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

**Gambar 1.9** Fungsi *str\_replace* yang digunakan untuk mencari karakter tertentu pada sebuah string dan menggantinya

## 2. Fungsi buatan sendiri.

Pemrogram juga diberi kebebasan untuk membuat fungsi sendiri, hal ini dilakukan jika dirasa fungsi yang dibutuhkan tidak disediakan oleh PHP secara built-in.

Tata cara penamaan sebuah fungsi mengikuti aturan yang sama dengan tata cara penamaan variable, selama masih sesuai dengan aturan PHP maka programmer diberi kebebasan memberi

nama fungsi. Nama fungsi yang benar diawali dengan huruf atau garis bawah (underscores) kemudian diikuti dengan angka huruf, angka, huruf atau garis bawah.

```
Function nama_fungsi(argumen1[,argumen2,...])  
{  
    Baris program1;  
    Baris program2;  
    Baris programN;  
    [return]  
}
```

**Gambar 1.10** Struktur deklarasi fungsi buatan sendiri

## BAB II

### PEMBAHASAN

#### 2.1 SOAL

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.

#### 2.2 JAWABAN

Dalam menyelesaikan permasalahan ini, saya membagi file menjadi lima bagian sesuai dengan intruksi. 4 file akan berisi masing-masing penyelesaian dari soal. File ke-5 akan menjadi index yang menggabungkan semua file.

##### 2.2.1 Soal 1, Menyimpan data menggunakan Array

Soal pertama adalah menyimpan nama-nama anggota keluarga saya didalam sebuah variabel array. Saya memberi nama array *\$nama*. Array langsung diberi index dan diberi nilai. File disimpan dengan nama *soal1.php*.

```
<?php

    $nama[0] = "Mudloaf";
    $nama[1] = "Sulastri";
    $nama[2] = "Wildanul Khadiq";
    $nama[3] = "Dzikri Ahmadillah";

?>
```

**Gambar 2.1** Menyimpan data dalam array

### 2.2.2 Soal 2, Menghitung huruf dan kata pada sebuah String

Soal kedua adalah menghitung banyak kata dan huruf pada nama yang telah disimpan tadi. Untuk menyelesaikannya, saya membuat fungsi bernama *cek\_nama* dengan sebuah parameter *\$nama*.

Untuk menghitung kata pada string *\$nama*, cukup memanggil fungsi *str\_word\_count* yang sudah tersedia pada PHP.

Untuk menghitung banyak huruf pada string *\$nama*, digunakan fungsi *strlen* yang sudah tersedia pada PHP. Sebelum menghitung banyak huruf yang ada, sebelumnya digunakan fungsi *str\_replace* untuk menghapus semua spasi pada string. Hal ini karena fungsi *strlen* menghitung spasi sebagai karakter. Untuk mencegah ini digunakan *str\_replace* untuk menghapus spasi.

Kedua perhitungan ditampilkan ke layar menggunakan sintaks *echo*. File akan disimpan dengan nama *soal2.php*.

```
<?php
    function cek_nama($nama){
        echo "Banyak kata = ".str_word_count($nama)."<br>";

        $nama = str_replace(' ', '', $nama);
        echo "Banyak huruf = ".strlen($nama)."<br>";
    }
?>
```

**Gambar 2.2** Fungsi untuk menghitung kata dan huruf pada string *\$nama*

### 2.2.3 Soal 3, Membalik String

Soal ketiga adalah membalik urutan dari string nama tadi. Untuk menyelesaikannya, saya membuat fungsi bernama *balik\_nama* dengan sebuah parameter *\$nama*.

Untuk membalik urutan karakter pada string, cukup menggunakan fungsi *strrev* yang sudah disediakan pada PHP. Hasil ditampilkan ke layar menggunakan *echo*. File akan disimpan dengan nama *soal3.php*.

```
<?php
    function balik_nama($nama){
        echo "kebalikan = ".strrev($nama)."<br>";
    }
?>
```

**Gambar 2.3** Menggunakan Fungsi *strrev* untuk membalik urutan karakter

#### 2.2.4 Soal 4, Menghitung huruf Vokal dan konsonan

Soal keempat adalah menghitung banyak huruf konsonan dan vokal dari string nama tadi. Untuk menyelesaikannya, saya membuat fungsi bernama *cek\_huruf* dengan sebuah parameter *\$nama*.

Tidak seperti sebelumnya, tidak ada fungsi built-in di PHP yang mampu langsung menampilkan banyak huruf vokal dan konsonan pada sebuah string. Untuk menyelesaikannya, pertama saya menghapus spasi dari string menggunakan *str\_replace* mirip seperti soal nomor 2.

Selanjutnya saya membuat dua variabel bantu. Pertama *\$sukuran* yang memiliki nilai ukuran string dikurang 1. Nilai variabel *\$sukuran* dikurang 1 karena nanti akan digunakan sebagai batas perulangan yang mana indeks perulangan dimulai dari 0. Selanjutnya membuat variabel array bernama *\$vokal* yang berisi kumpulan huruf vokal.

```
function cek_huruf ($nama){
    $nama = str_replace(' ', '', $nama);
    $sukuran = strlen($nama)-1;
    $huruf_vokal = array("a","i","u","e","o");
```

**Gambar 2.4.1** Menghilangkan spasi pada string dan membuat variabel bantu

Selanjutnya adalah membuat pengecekan huruf. Sebelumnya mendeklarasikan variabel *\$konsonan* dan *\$vokal* dengan nilai 0 yang digunakan untuk menyatakan banyak huruf konsonan dan vokal pada string nantinya.

Berikutnya saya membuat sebuah perulangan for. perulangan digunakan untuk mengecek per huruf pada string. Perulangan dilakukan sebanyak ukuran indeks. Didalamnya terdapat percabangan if yang menentukan jenis huruf.

Digunakan fungsi *in\_array* yang merupakan fungsi built-in pada PHP. Fungsi ini akan membandingkan kesamaan antara dua string. jika terdapat kesamaan pada kedua string, maka fungsi memberi nilai balik TRUE. (PHP: *in\_array* - Manual)

Fungsi *in\_array* akan digunakan untuk membandingkan tiap huruf pada string *\$nama* dan *\$vokal*. Jika nilai balik TRUE, maka huruf yang dibandingkan pastilah huruf vokal. Jika nilai balik FALSE, maka pastinya huruf tersebut merupakan huruf konsonan.

Terakhir variabel *\$konsonan* dan *\$vokal* akan ditampilkan ke layar menggunakan *echo*. File disimpan dengan nama *soal4.php*.

```
for($x = 0; $x <= $ukuran; $x++){  
    if(in_array($nama[$x], $huruf_vokal)){  
        $vokal++;  
    } else {  
        $konsonan++;  
    }  
}  
echo "Banyak huruf konsonan = ".$konsonan."<br>";  
echo "Banyak huruf vokal = ".$vokal."<br>";
```

**Gambar 2.4.2** membandingkan tiap huruf pada string menggunakan *in\_array*

### 2.2.5 Index

File terakhir yang merupakan kumpulan dari file-file sebelumnya. Oleh karena itu, hal pertama yang dilakukan adalah meng-include semua file yang telah dibuat.

```
<?php
    include_once('soal1.php');
    include_once('soal2.php');
    include_once('soal3.php');
    include_once('soal4.php');
?>
```

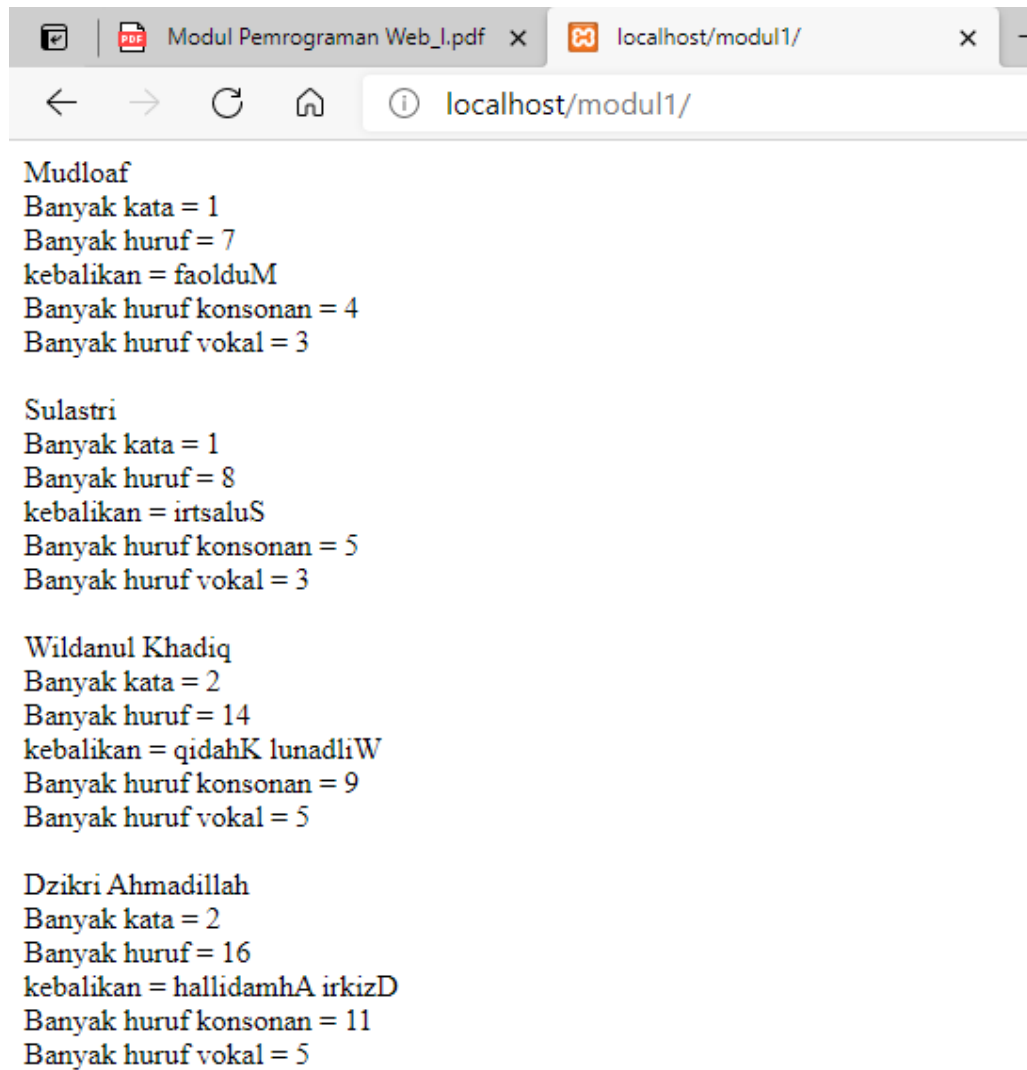
**Gambar 2.5.1** meng-include file yang telah dibuat sebelumnya agar bisa digunakan.

Selanjutnya adalah menggunakan fungsi yang telah dibuat pada masing-masing file ke array *\$nama*. Digunakan perulangan for agar perintah dilakukan ke tiap isi pada array *\$nama*.

```
for($x = 0; $x < count($nama); $x++){
    echo $nama[$x]."<br>";
    cek_nama($nama[$x]);
    balik_nama($nama[$x]);
    cek_huruf($nama[$x]);
    echo "<br>";
}
```

**Gambar 2.5.2** menjalankan tiap fungsi yan telah dibuat sebelumnya





**Gambar 2.6** Hasil run

### **BAB III**

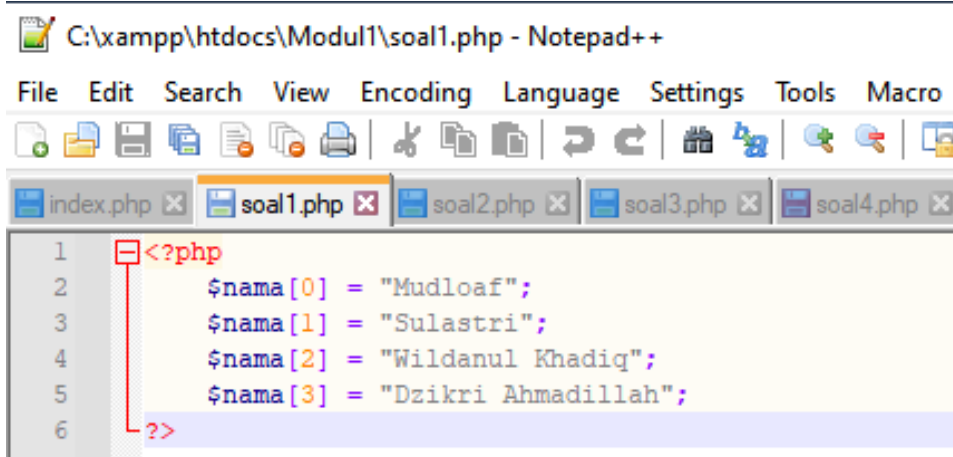
### **KESIMPULAN**

Pembuatan web menggunakan pemrograman PHP mempunyai berbagai macam kelebihan dan keunggulan bila dibandingkan dengan menggunakan program lain yang sejenis. Berbagai macam kemudahan yang ada pada program PHP sangat fleksibel dan akan memberikan kemudahan dalam aplikasinya. Karena PHP mencari tags pembuka `<?php` dan penutup `?>` untuk menerjemahkan code, kita dapat menggunakan HTML dan PHP secara bergantian tanpa takut terjadi error.

## DAFTAR PUSTAKA

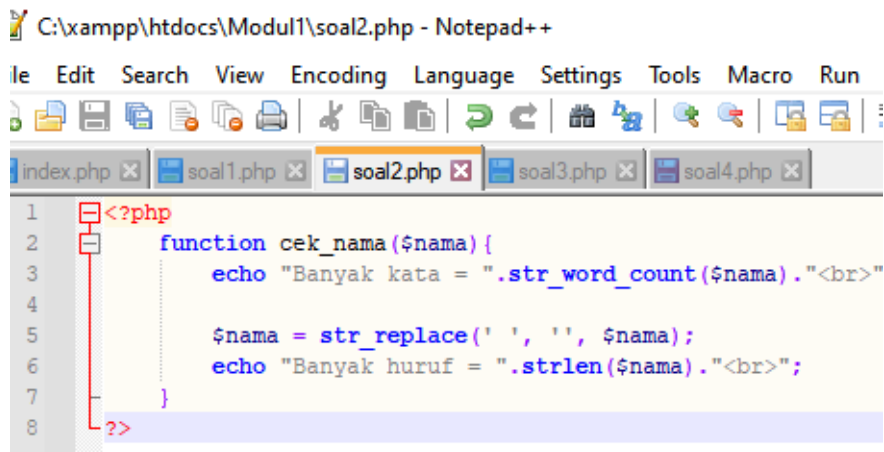
- Andre (2019) *Pengertian dan Fungsi PHP dalam Pemrograman Web - HCR TECH*. Available at: <https://www.duniailkom.com/pengertian-dan-fungsi-php-dalam-pemrograman-web/> (Accessed: 28 March 2021).
- ‘MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya’ (no date).
- PHP: in\_array - Manual* (no date). Available at: <https://www.php.net/manual/en/function.in-array.php> (Accessed: 28 March 2021).
- PHP Tutorial* (no date). Available at: <https://www.w3schools.com/php/> (Accessed: 28 March 2021).
- Suprianto, D. (2008) ‘Buku Pintar pemrograman php’, *Bandung: OASE Media*.

## LAMPIRAN



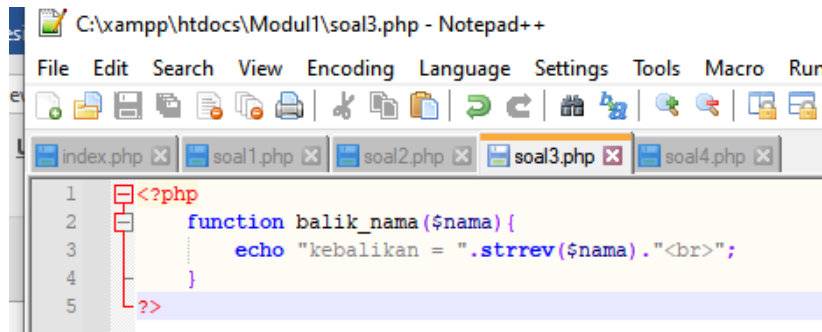
```
C:\xampp\htdocs\Modul1\soal1.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro
index.php x soal1.php x soal2.php x soal3.php x soal4.php x
1 <?php
2     $nama[0] = "Mudloaf";
3     $nama[1] = "Sulastri";
4     $nama[2] = "Wildanul Khadiq";
5     $nama[3] = "Dzikri Ahmadillah";
6 ?>
```

Source code soal1.php



```
C:\xampp\htdocs\Modul1\soal2.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run
index.php x soal1.php x soal2.php x soal3.php x soal4.php x
1 <?php
2 function cek_nama($nama){
3     echo "Banyak kata = ".str_word_count($nama). "<br>";
4
5     $nama = str_replace(' ', '', $nama);
6     echo "Banyak huruf = ".strlen($nama). "<br>";
7 }
8 ?>
```

Source code soal2.php



```
C:\xampp\htdocs\Modul1\soal3.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run
index.php x soal1.php x soal2.php x soal3.php x soal4.php x
1 <?php
2 function balik_nama($nama){
3     echo "kebalikan = ".strrev($nama). "<br>";
4 }
5 ?>
```

Source code soal3.php

```
index.php x soal1.php x soal2.php x soal3.php x soal4.php x
1 <?php
2 function cek_huruf ($nama){
3
4     $nama = str_replace(' ', '', $nama);
5     $ukuran = strlen($nama)-1;
6     $huruf_vokal = array("a","i","u","e","o");
7
8     $vokal = 0;
9     $konsonan = 0;
10
11     for($x = 0; $x<=$ukuran; $x++){
12         if(in_array($nama[$x], $huruf_vokal)){
13             $vokal++;
14         } else {
15             $konsonan++;
16         }
17     }
18     echo "Banyak huruf konsonan = ".$konsonan."<br>";
19     echo "Banyak huruf vokal = ".$vokal."<br>";
20 }
21 ?>
```

Source code soal4.php

```
index.php x soal1.php x soal2.php x soal3.php x soal4.php x
1 <!--
2 Dzikri Ahmadillah
3 193020503039
4 -->
5
6 <!DOCTYPE html>
7 <head>
8     <?php
9         include_once('soal1.php');
10        include_once('soal2.php');
11        include_once('soal3.php');
12        include_once('soal4.php');
13    ?>
14 </head>
15
16 <body>
17     <?php
18         for($x = 0; $x < count($nama); $x++){
19             echo $nama[$x]."<br>";
20             cek_nama($nama[$x]);
21             balik_nama($nama[$x]);
22             cek_huruf($nama[$x]);
23             echo "<br>";
24         }
25     ?>
26 </body>
27 </html>
```

Source code index.php