

**LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE I**



Nama : Oscar Oktorian Almando
NIM : 193020503037
Kelas : A
Modul : I (PHP (Hypertext Preprocessor))

JURUSAN/PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021

BAB I

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 Pengertian PHP

PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman server side scripting yang bersifat open source.

Sebagai sebuah scripting language, PHP menjalankan instruksi pemrograman saat proses runtime. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses.

PHP merupakan bahasa pemrograman server-side, maka script dari PHP nantinya akan diproses di server. Jenis server yang sering digunakan bersama dengan PHP antara lain Apache, Nginx, dan LiteSpeed.

Selain itu, PHP juga merupakan bahasa pemrograman yang bersifat open source. Pengguna bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan mereka (Awwaabiin 2020).

1.2.2 Fungsi PHP

Secara umum, fungsi PHP adalah digunakan untuk pengembangan website. Baik website statis seperti situs berita yang tidak membutuhkan banyak fitur. Atau pun website dinamis seperti toko online dengan segudang fitur pendukung.

Namun, penggunaan PHP tidak terbatas pada pengembangan website saja, lho. Karena fleksibilitasnya yang tinggi, PHP juga bisa digunakan untuk membuat aplikasi komputer sekalipun (Awwaabiin 2020).

1.2.3 Sintaks Dasar PHP

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 Sintaks dasar PHP "Hello World".

a. 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) (Jurusan Teknik Informatika 2021). 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 Sintaks PHP.

Lingkup dari sebuah variabel adalah bagian dari skrip dimana variabel bisa dirujuk atau digunakan. Ada tiga lingkup variabel: Lokal, Global dan Static. Variabel yang dideklarasikan diluar fungsi memiliki lingkup global, sedangkan variabel yang dideklarasikan didalam fungsi memiliki lingkup lokal yang hanya bisa diakses didalam fungsi saja. Kelebihan dari penggunaan variabel lokal adalah 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 perlu untuk mempertahankan nilai variabel lokal supaya tidak dihapus. Untuk itu, gunakan keyword static dalam pendeklarasian variabel yang nilainya ingin dipertahankan.

```

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

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

```

Gambar 1. 3 Sintaks PHP.

```

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

```

Gambar 1. 4 Sintaks PHP.

```

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

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

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

```

Gambar 1. 5 Sintaks PHP.

```

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

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

```

Gambar 1. 6 Sintaks PHP.

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

Gambar 1. 7 Sintaks PHP.

```
        echo $x;
        $x++;
    }

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

Gambar 1. 8 Sintaks PHP.

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

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

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

Gambar 1. 9 Echo dan Print.

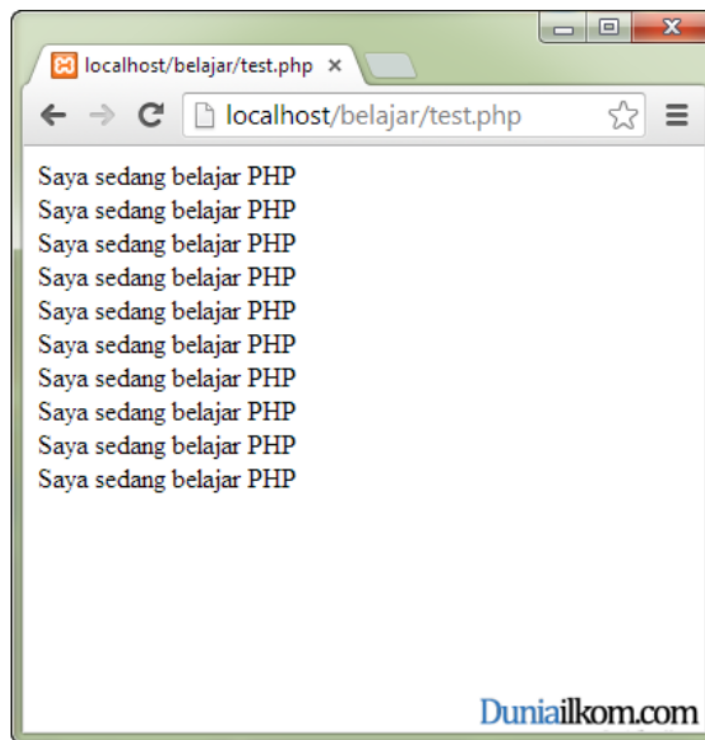
1.2.4 Struktur Perulangan For dalam PHP

Struktur perulangan (atau dalam bahasa inggris disebut dengan loop) adalah instruksi program yang bertujuan untuk mengulang beberapa baris perintah. Dalam merancang perulangan kode program, kita setidaknya harus mengetahui 3 komponen, yaitu kondisi awal dari perulangan, perintah program yang akan diulang, serta kondisi akhir dimana perulangan akan berhenti (Pratama 2014b).

Di dalam bahasa pemrograman, terdapat beberapa jenis instruksi perulangan, salah satunya: struktur perulangan FOR.

Sebagai contoh sederhana untuk perulangan for, akan dibuat program PHP untuk menampilkan 10 baris kalimat “Saya sedang belajar PHP”. Berikut adalah kode program yang digunakan:

```
1 <?php
2 for ($i= 1; $i <= 10; $i++)
3 {
4     echo "Saya sedang belajar PHP";
5     echo "<br />";
6 }
7 ?>
```



Gambar 1. 10 Penulisan perulangan For pada PHP.

Jika menjalankan perintah tersebut, maka di *web browser* akan tampil sebanyak 10 kalimat. Kemampuan bahasa pemrograman untuk melakukan perulangan ini sangat praktis jika yang diinginkan adalah mengulang beberapa perintah yang sama sebanyak beberapa kali.

1.2.5 Cara Penulisan Struktur Perulangan For dalam PHP

Untuk kondisi perulangan for, kita setidaknya membutuhkan 3 kondisi, yaitu di kondisi awal perulangan, kondisi pada saat perulangan, dan kondisi yang harus dipenuhi agar perulangan berhenti.

Penulisan dasar format perulangan for PHP adalah sebagai berikut:

```
for (start; condition; increment)  
{  
    statement;  
}
```

- a. Start adalah kondisi pada saat awal perulangan. Biasanya kondisi awal ini digunakan untuk membuat dan memberikan nilai kepada variabel counter yang digunakan untuk mengontrol perulangan. Misalkan, kita akan membuat variabel counter \$i, maka untuk kondisi start ini, kita juga harus memberikan nilai awal untuk variabel \$i, misalnya dengan 1, maka \$i=1.
- b. Condition adalah kondisi yang harus dipenuhi agar perulangan dijalankan. Selama kondisi ini terpenuhi, maka PHP akan terus melakukan perulangan. Biasanya variabel counter digunakan untuk mengatur akhir perulangan. Misalkan kita ingin menghentikan perulangan jika variabel \$i telah mencapai nilai 20, maka pada bagian condition ini kita membuat perintah \$i<=20, yang berarti selama nilai \$i kurang atau sama dengan 20, terus lakukan perulangan.
- c. Increment adalah bagian yang digunakan untuk memproses variabel counter agar bisa memenuhi kondisi akhir perulangan. Biasanya, pada bagian inilah kita akan membuat kondisi dari variabel counter.
- d. Statement adalah bagian kode program yang akan diproses secara terus-menerus selama proses perulangan berlangsung. Untuk statement ini, kita membuat blok program di antara tanda kurung kurawal ({ dan }) sebagai penanda bahwa bagian di dalam kurung kurawal inilah yang akan dikenai proses perulangan.

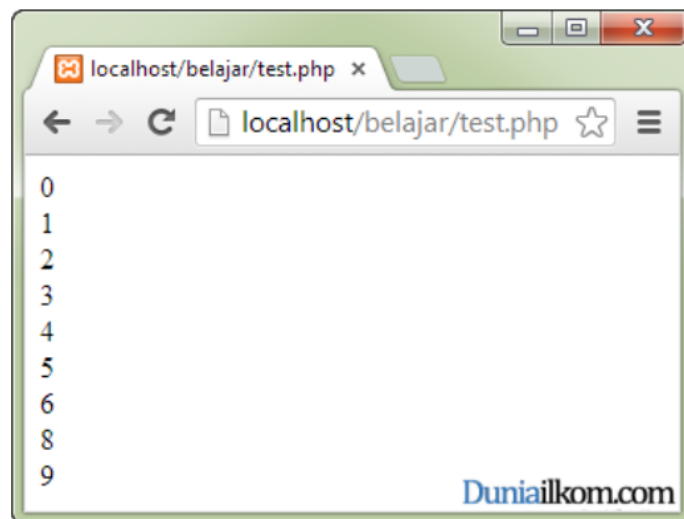
(Pratama 2014b)

1.2.6 Fungsi Perintah Continue dalam PHP

Perintah continue digunakan untuk men-interupsi perulangan dalam PHP, namun jika perintah break digunakan untuk menghentikan perulangan, maka perintah continue hanya akan menghentikan perulangan untuk 1 iterasi saja, lalu proses perulangan akan dilanjutkan.

Berikut contoh kode PHP penggunaan perintah continue:

```
1 <?php
2 for ($i=0; $i <10; $i++)
3 {
4     if ($i==7)
5     {
6         continue;
7     }
8     echo $i;
9     echo "<br />";
10 }
```



Gambar 1. 11 Penulisan perulangan For dengan perintah *continue* pada PHP.

Setelah perintah for, terdapat sebuah kondisi IF yang jika variabel counter \$i bernilai 7, maka jalankan continue.

Arti dari continue ini adalah sebuah instruksi kepada PHP untuk melewati sisa perintah dalam perulangan, dan langsung lompat ke nilai counter berikutnya, yakni 8. Dari hasil program, tidak akan terlihat angka 7 ditampilkan. (Pratama 2014a)

1.2.7 Pengertian Fungsi (Function) dalam PHP

Dalam merancang kode program, terkadang sering membuat kode yang melakukan tugas yang sama secara berulang-ulang, seperti membaca tabel dari database, menampilkan penjumlahan, dan lain-lain. Tugas yang sama ini akan lebih efektif jika dipisahkan dari program utama, dan dirancang menjadi sebuah fungsi.

Fungsi (atau Function) dalam bahasa pemrograman adalah kode program yang dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Seorang programmer dapat membuat fungsi sendiri, atau menggunakan fungsi yang dibuat oleh programmer lain.

Dalam dunia pemrograman terdapat istilah ‘lazy programming’ yang artinya bukanlah programmer yang malas. Tetapi, daripada membuat kode program umum dari dasar, programmer bisa menggunakan fungsi yang telah dibuat oleh programmer lain. PHP bahkan menyediakan ribuan fungsi bawaan yang tersedia untuk membantu dalam merancang program.

Mengetahui cara penggunaan fungsi ini akan menghemat waktu pembuatan program dan membuat kode program menjadi lebih efisien. Lazy programming is smart programming. (Pratama 2014c)

Menggunakan fungsi dalam teori pemrograman sering juga disebut dengan istilah ‘memanggil fungsi’ (calling a function). Fungsi dipanggil dengan menulis nama dari fungsi tersebut, dan diikuti dengan argumen (jika ada). Argumen ditulis di dalam tanda kurung, dan jika jumlah argumen lebih dari satu, maka diantaranya dipisahkan oleh karakter koma.

Setelah memproses nilai inputan, hampir semua fungsi akan memberikan nilai hasil pemrosesan tersebut (walaupun ada fungsi yang tidak memberikan nilai). Cara fungsi memberikan nilainya ini sering disebut dengan ‘mengembalikan nilai’ (return a value). Nilai yang dikembalikan oleh sebuah fungsi dapat ditampung ke dalam variabel, atau langsung ditampilkan ke web browser. (Pratama 2014c)

BAB II

PEMBAHASAN

Dalam praktikum modul 1 ini, aplikasi yang digunakan untuk menuliskan sintaks PHP dan HTML adalah aplikasi Notepad++. Sintaks yang telah dibuat akan disimpan ke dalam format *.php dan dieksekusi untuk ditampilkan hasilnya menggunakan aplikasi XAMPP (untuk memproses PHP) dan browser untuk menampilkannya.

2.1 Program PHP menampilkan nama yang disimpan dalam array

Program yang dibuat akan menampilkan nilai dari masing-masing indeks dalam sebuah array yang telah dideklarasikan. Berikut ini merupakan sintaks dari program yang dibuat.

Langkah pertama, membuat struktur awal dari HTML, hal ini dilakukan agar memudahkan dalam mengatur tata letak, struktur dan jenis teks yang akan ditampilkan dari hasil sintaks PHP. Berikut ini merupakan struktur awal HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title> Modul 1 - 1 | Oscar Oktorian Almando </title>
  </head>
  <body>
  </body>
</html>
```

Pada bagian <head> terdapat tag <title> yang berfungsi untuk memberikan judul pada tab di browser. Setelah membuat struktur awal, langkah selanjutnya yaitu mendeklarasikan tipe data array sekaligus memberikan nilai. Berikut ini merupakan sintaksnya.

```
<!DOCTYPE html>

<?php

    $nama=["OscarOkt", "Oktorian Amario", "Almando bin
Aliando", "Salman Simpati", "Suka Suka"];

?>

<html>

    <head>

        <title> Modul 1 - 1 | Oscar Oktorian Almando </title>

    </head>

    <body>

        </body>

</html>
```

Karena mendeklarasikan variabel untuk pemrograman PHP, maka sintaks harus dimasukkan ke dalam sebuah tag khusus PHP yang diawali dengan **<?php** dan diakhiri dengan **?>**. Tag ini dapat diletakkan di mana saja, namun perlu diperhatikan untuk pemakaiannya, khususnya fungsi dan variabel yang harus dideklarasikan di bagian atas sebelum digunakan atau dipanggil. Hal ini karena sintaks akan dibaca dari atas ke bawah. Tanpa menggunakan tag ini, sintaks PHP (pemanggilan fungsi, variabel, operasi-operasi) akan ditampilkan secara harfiah.

Selanjutnya mendeklarasikan variabel yang diawali dengan simbol **\$** dan dinamai dengan kata *nama*. Variabel *\$nama* akan dijadikan array agar dapat menampung lebih dari 1 indeks nilai, dengan cara menuliskan kurung siku '[' dan ']' kemudian mengisi nama-nama sesuai yang ada pada sintaks di atas. Karena yang akan diisi merupakan nama-nama, maka tipe data yang digunakan adalah String dan penggunaannya yaitu menuliskan nilainya diantara tanda petik (""). Setelah menuliskan deklarasi variabel dan mengisi

nilainya, tutup perintah tersebut dengan tanda *semicolon* (;) agar tidak terjadi error.

Kemudian setelah mendeklarasikan variabel array, langkah selanjutnya adalah untuk menampilkan nilai pada masing-masing indeks array yang telah dideklarasikan. Berikut ini merupakan sintaksnya.

```
<!DOCTYPE html>
<?php
    $nama=["OscarOkt", "Oktorian Amario", "Almando bin Aliando",
    "Salman Simpati", "Suka Suka"];
?>

<html>
    <head>
        <title> Modul 1 - 1 | Oscar Oktorian Almando </title>
    </head>
    <body>
        <h2>Isi pada masing-masing indeks Array : </h2>
        <p align="left">
            <?php
                for($i=0; $i<count($nama); $i++){
                    echo "Indeks ke-[$i] : ",$nama[$i],"<br><br>";
                }
            ?>
        </p>
    </body>
</html>
```

Kalimat “*Isi pada masing-masing indeks Array :*” ditulis di dalam tag `<h2>` yang merupakan tag untuk menampilkan sebuah teks atau objek sebagai *header-2*.

Kemudian, barulah menuliskan sintaks PHP untuk menampilkan nilai pada masing-masing indeks variabel array. Sintaks yang digunakan menggunakan perulangan *for* yang menampilkan sebanyak indeks yang ada.

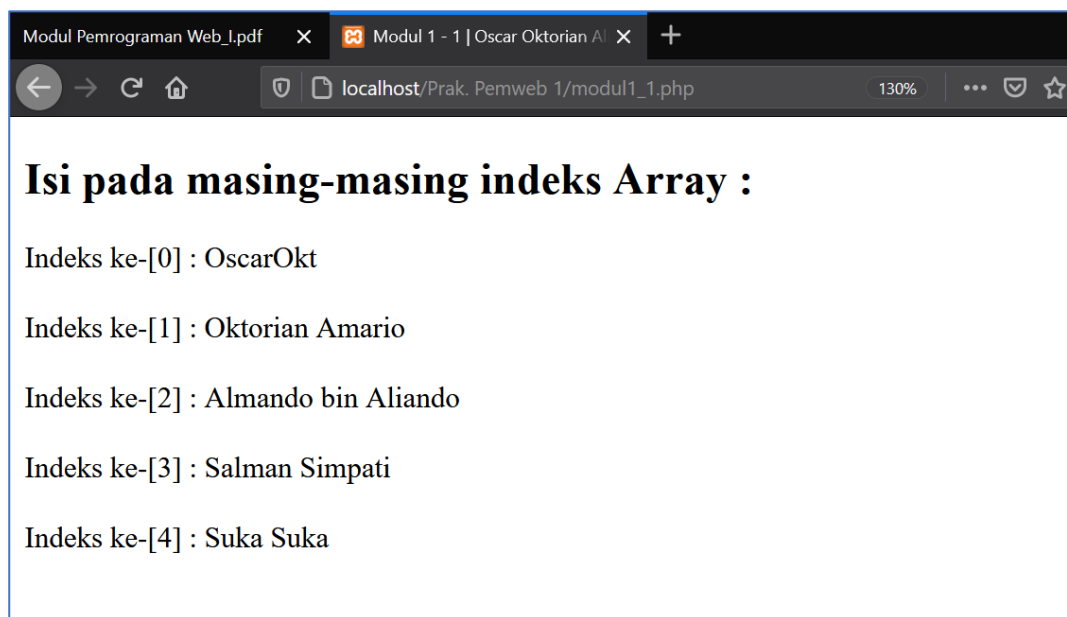
Agar mengetahui banyaknya indeks yang ada dalam suatu array, menggunakan fungsi `count($variabel)`. Semua sintaks yang terdapat dalam `<?php ?>` tersebut kemudian akan dimasukkan lagi ke dalam tag HTML `<p>` dengan penambahan aturan `align="left"` yang digunakan untuk pengaturan tulisan dan sebagai paragraf.

Setelah sintaks dibuat, maka akan dilakukan *save* pada *Notepad++* dan disimpan dengan ekstensi `*.php`. Berikut ini merupakan tampilan sintaks yang dibuat menggunakan *Notepad++*.

```
1 <!DOCTYPE html>
2 <?php
3     $nama=["OscarOkt", "Oktorian Amario", "Almando bin Aliando", "Salman Simpati", "Suka Suka"];
4     ?>
5
6 <html>
7     <head>
8         <title> Modul 1 - 1 | Oscar Oktorian Almando </title>
9     </head>
10    <body>
11        <h2>Isi pada masing-masing indeks Array : </h2>
12        <p align="left">
13            <?php
14                for($i=0; $i<count($nama); $i++){
15                    echo "Indeks ke-[$i] : ",$nama[$i], "<br><br>";
16                }
17            ?>
18        </p>
19    </body>
20 </html>
```

Gambar 2. 1 Sintaks PHP 1 - Menampilkan nilai pada masing-masing indeks variabel array.

Berikut ini merupakan output hasil dari sintaks PHP yang telah dibuat.



Gambar 2. 2 Output Sintaks PHP 1 - Menampilkan nilai pada masing-masing indeks variabel array.

2.2 Program PHP menampilkan jumlah kata dan jumlah huruf

Program yang dibuat akan menampilkan jumlah kata dan jumlah huruf dari masing-masing indeks dalam sebuah array yang telah dideklarasikan. Berikut ini merupakan sintaks dari program yang dibuat.

Sebagian sintaks akan dituliskan sama seperti sintaks pada tugas praktikum nomor sebelumnya, namun ditambahkan beberapa sintaks fungsi yang akan digunakan pada tugas praktikum di nomor ini. Berikut ini merupakan sintaks yang dibuat.

```
<!DOCTYPE html>
<?php
    $nama=["OscarOkt", "Oktorian Amario", "Almando bin Aliando",
    "Salman Simpati", "Suka Suka"];

    function hitungHuruf($sttr){
        $num= strlen($sttr)-(str_word_count($sttr)-1);
        return $num;
    }
?>

<html>
    <head>
        <title> Modul 1 - 2 / Oscar Oktorian Almando </title>
    </head>
    <body>
        <h2>Jumlah kata dan huruf pada nilai di masing-masing
indeks Array : </h2>
        <p align="left">
            <?php
                for($i=0; $i<count($nama); $i++){
```

```

                echo "Indeks ke-[$i] : ",$nama[$i],"<br>";
                echo          "Jumlah          Kata          :
",str_word_count($nama[$i])," kata <br>";
                echo          "Jumlah          Huruf          :
",hitungHuruf($nama[$i])," huruf <br><br>";
            }
        ?>
    </p>
</body>
</html>

```

Pada baris bagian atas, setelah mendeklarasikan variabel array, dituliskan sebuah fungsi bernama *hitungHuruf()*. Fungsi ini dibuat untuk menghitung huruf (atau String secara keseluruhan) dalam sebuah variabel tanpa menghitung spasi. Fungsi ini memiliki parameter yang akan menampung nilai string. Selanjutnya di dalam fungsi terdapat variabel *\$num* yang akan menampung jumlah dari huruf berdasarkan operasi perhitungan yang ditentukan. Operasi perhitungan yang dimaksud adalah *\$num* akan menampung hasil hitung panjang awal sebuah string termasuk spasi (dengan menggunakan fungsi *strlen()*) *dikurangi* jumlah spasi yang ada (dengan menggunakan fungsi *str_word_count()* dikurang 1). Selanjutnya *\$num* akan direturn agar fungsi tersebut dapat memberikan nilai.

Kemudian pada bagian <title> dan beberapa kalimat di struktur <body> terdapat beberapa perubahan judul untuk menyesuaikan dengan konteks soal praktikum. Kemudian, dalam perulangan yang sintaksnya sama seperti pada soal praktikum sebelumnya, terdapat penambahan baris sintaks. Penambahan baris itu meliputi sintaks untuk menampilkan data jumlah kata dan sintaks untuk menampilkan data jumlah huruf. Untuk menampilkan data jumlah kata, cukup menggunakan fungsi *str_word_count()*, yang melengkapi perintah *echo* agar data dapat diberikan keterangan.

Untuk menampilkan jumlah huruf, maka akan menggunakan fungsi yang telah dibuat di bagian atas sintaks yaitu *hitungHuruf()*. Fungsi tersebut

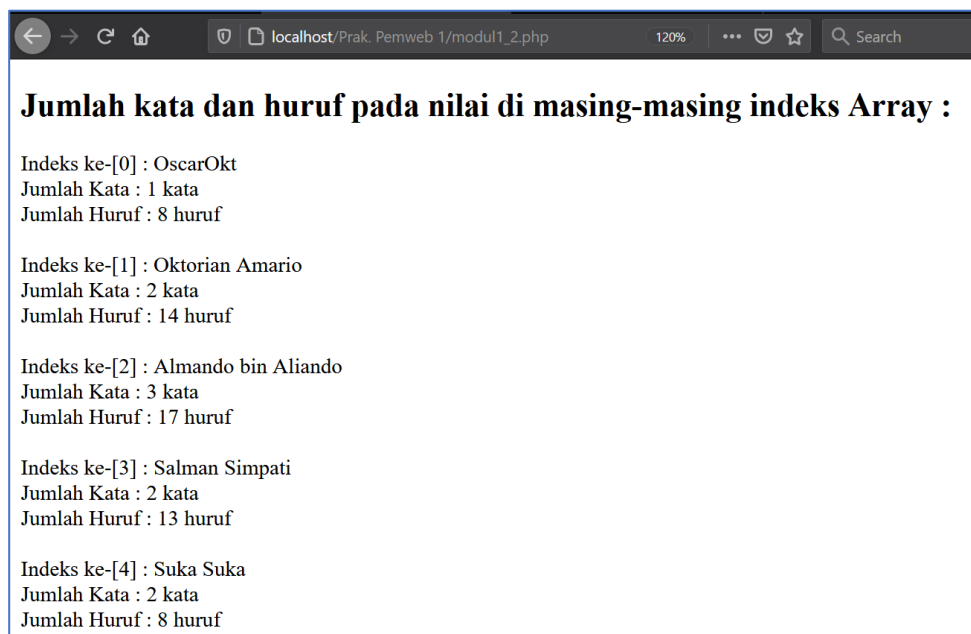
dimasukkan dalam sintaks *echo* agar fungsi tersebut dapat ditampilkan beserta keterangan yang ditambah.

Setelah sintaks dibuat, akan dilakan *save* seperti halnya pada tugas praktikum nomor sebelumnya. Berikut ini merupakan tampilan sintaks PHP pada aplikasi Notepad++.

```
1 <!DOCTYPE html>
2 <?php
3     $nama=["OscarOkt", "Oktorian Amario", "Almando bin Aliando", "Salman Simpati", "Suka Suka"];
4     function hitungHuruf($sttr){
5         $num= strlen($sttr)-(str_word_count($sttr)-1);
6         return $num;
7     }
8 ?>
9
10 <html>
11     <head>
12         <title> Modul 1 - 2 | Oscar Oktorian Almando </title>
13     </head>
14     <body>
15         <h2>Jumlah kata dan huruf pada nilai di masing-masing indeks Array : </h2>
16         <p align="left">
17             <?php
18             for($i=0; $i<count($nama); $i++){
19                 echo "Indeks ke-[$i] : ",$nama[$i],"<br>";
20                 echo "Jumlah Kata : ",str_word_count($nama[$i])," kata <br>";
21                 echo "Jumlah Huruf : ",hitungHuruf($nama[$i])," huruf <br><br>";
22             }
23             ?>
24         </p>
25     </body>
26 </html>
```

Gambar 2. 3 Sintaks PHP 2 - Menampilkan jumlah kata dan huruf pada masing-masing indeks variabel array.

Berikut ini merupakan output dari hasil sintaks yang dibuat.



Jumlah kata dan huruf pada nilai di masing-masing indeks Array :

Indeks	Nama	Jumlah Kata	Jumlah Huruf
ke-[0]	OscarOkt	1 kata	8 huruf
ke-[1]	Oktorian Amario	2 kata	14 huruf
ke-[2]	Almando bin Aliando	3 kata	17 huruf
ke-[3]	Salman Simpati	2 kata	13 huruf
ke-[4]	Suka Suka	2 kata	8 huruf

Gambar 2. 4 Output Sintaks PHP 2 - Menampilkan jumlah kata dan huruf pada masing-masing indeks variabel array.

2.3 Program PHP menampilkan kebalikan nama yang diinputkan

Program yang dibuat akan menampilkan kebalikan dari string yang telah diinputkan pada halaman web oleh user. Berikut ini merupakan sintaks yang telah dibuat.

```
<!DOCTYPE html>

<html>
    <head>
        <title> Modul 1 - 3 | Oscar Oktorian Almando </title>
    </head>
    <body>
        <h2>Kalimat yang dibalik pada nilai di masing-masing
indeks Array : </h2>
        <form      method="post"      action="<?php      echo
$_SERVER['PHP_SELF']; ?>">
            Nama: <input type="text" name="nama">
            <input type="submit">
        </form>
        <?php

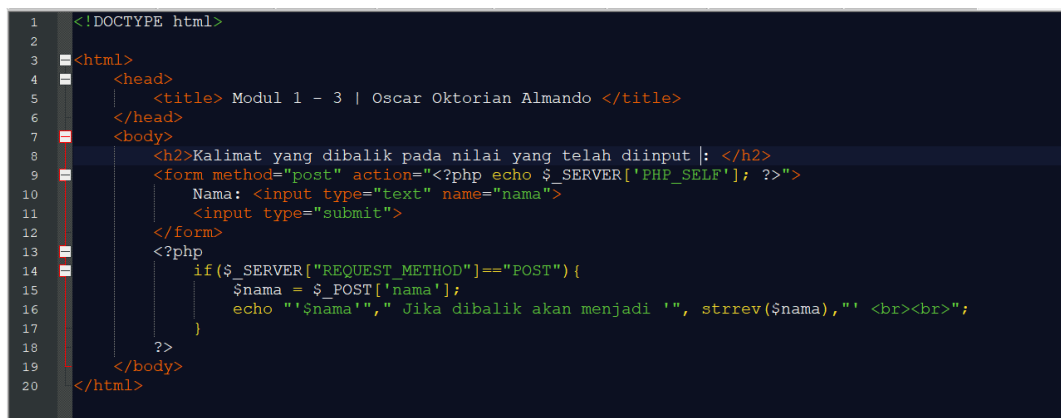
            if($_SERVER["REQUEST_METHOD"]=="POST"){
                $nama = $_POST['nama'];
                echo "'$nama'," Jika dibalik akan menjadi
'", strrev($nama),'<br><br>";
            }
        ?>
    </body>
</html>
```

Pertama, untuk menginput sebuah string, maka dibuat sintaks untuk menampilkan form sebagai tempat memasukkan nilai dan tombol untuk mengeksekusi hasil input. Di dalam tag <form>, terdapat atribut *method* yang

menandakan form tersebut akan diproses sebagai *post*, dan atribut *action* agar form tersebut dapat mengirimkan informasi mengenai data yang masuk ke dalam form tersebut.

Kemudian, setelah form dibuat, maka akan dituliskan sintaks PHP di antara tag ‘<?php’ dan ‘?>’. Sintaks yang dimasukkan pertama akan memeriksa apakah informasi yang terdapat pada `$_SERVER["REQUEST_METHOD"]` sama dengan informasi yang ada pada “POST”. Jika bernilai *true* maka sintaks akan mengeksekusi baris variabel `$nama` untuk diisi nilai dari `$_POST['nama']` yang mana menampung nilai yang diinputkan. Setelah itu variabel `$nama` akan digunakan pada fungsi `strrev()` untuk menghasilkan nilai dengan urutan yang dibalik sekaligus menampilkannya dengan perintah *echo*.

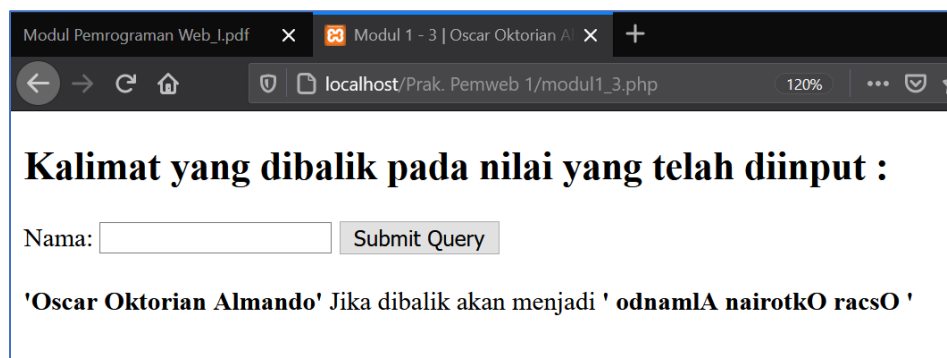
Berikut ini merupakan tampilan sintaks pada aplikasi *Notepad++*.



```
1 <!DOCTYPE html>
2
3 <html>
4 <head>
5 <title> Modul 1 - 3 | Oscar Oktorian Almando </title>
6 </head>
7 <body>
8 <h2>Kalimat yang dibalik pada nilai yang telah diinput |: </h2>
9 <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>"
10 Nama: <input type="text" name="nama">
11 <input type="submit">
12 </form>
13 <?php
14 if($_SERVER["REQUEST_METHOD"]=="POST"){
15     $nama = $_POST['nama'];
16     echo "$nama", " Jika dibalik akan menjadi '", strrev($nama), "' <br><br>";
17 }
18 ?>
19 </body>
20 </html>
```

Gambar 2. 5 Sintaks PHP 3 - Menampilkan kebalikan dari nilai yang diinput (versi pertama).

Berikut ini merupakan output dari sintaks yang dibuat, yang dibuka melalui browser.



Modul Pemrograman Web_I.pdf x Modul 1 - 3 | Oscar Oktorian Al x +

localhost/Prak. Pemweb 1/modul1_3.php 120%

Kalimat yang dibalik pada nilai yang telah diinput :

Nama: Submit Query

'Oscar Oktorian Almando' Jika dibalik akan menjadi ' odnamlA nairotkO racso '

Gambar 2. 6 Output Sintaks PHP 3 - Menampilkan kebalikan dari nilai yang diinput (versi pertama).

Terdapat versi kedua dari tugas praktikum nomor 3 ini, yakni dengan membalikkan nama dari variabel array yang telah dideklarasikan pada tugas praktikum di nomor sebelumnya.

Berikut ini merupakan sintaksnya.

```
<!DOCTYPE html>
<?php
    $name=["OscarOkt", "Oktorian Amario", "Almando bin Aliando",
"Salman Simpati", "Suka Suka"];
?>
<html>
    <head>
        <title> Modul 1 - 3 | Oscar Oktorian Almando </title>
    </head>
    <body>
        <h2>Kalimat yang dibalik pada nilai yang telah diinput :
    </h2>
        <form      method="post"      action="<?php      echo
$_SERVER['PHP_SELF']; ?>">
            Nama: <input type="text" name="nama">
            <input type="submit">
        </form>
        <?php

            if($_SERVER["REQUEST_METHOD"]=="POST"){
                $nama = $_POST['nama'];
                echo "<br><b>'$nama'</b>"," Jika dibalik
akan menjadi <b>' ", strrev($nama),' </b> <br><br>";
            }

            for($i=0; $i<count($name); $i++){
```

```

        echo "Indeks ke-[$i] : ",$name[$i],"<br>";
        echo  "Jika  dibalik  akan  menjadi  '",
strrev($name[$i]),"' <br><br>";

    }
    ?>

</p>

</body>

</html>

```

Versi kedua ini hanyalah penambahan kecil untuk menampilkan kebalikan nama dari nilai di masing-masing variabel array yang telah dibuat. Untuk menampilkan semua data yang di dalam array, maka digunakan perintah perulangan for. Untuk menampilkan kebalikan nama itu, digunakan perintah *strrev()* yang diisi parameter variabel array itu sendiri, yang nantinya diletakkan bersamaan dengan perintah *echo* agar dapat diberikan keterangan.

Berikut ini merupakan tampilan sintaks pada aplikasi *Notepad++*.



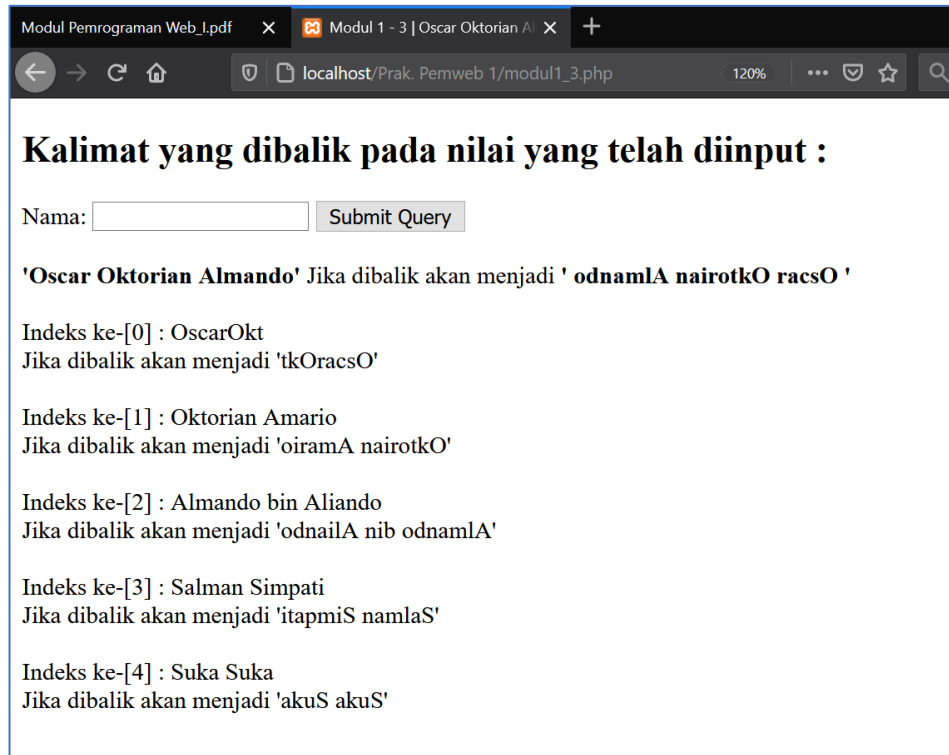
```

1  <!DOCTYPE html>
2  <?php
3      $name=["OscarOkt", "Oktorian Amario", "Almando bin Aliando", "Salman Simpati", "Suka Suka"];
4  ?>
5  <html>
6      <head>
7          <title> Modul 1 - 3 | Oscar Oktorian Almando </title>
8      </head>
9      <body>
10         <h2>Kalimat yang dibalik pada nilai yang telah diinput : </h2>
11         <form method="post" action="<?php echo $ _SERVER['PHP_SELF']; ?>">
12             Nama: <input type="text" name="nama">
13             <input type="submit">
14         </form>
15         <?php
16             if($_SERVER["REQUEST_METHOD"]=="POST"){
17                 $nama = $_POST['nama'];
18                 echo "<br><b>'$nama'</b>"," Jika dibalik akan menjadi <b>' ", strrev($nama)," '</b> <br><br>";
19             }
20
21             for($i=0; $i<count($name); $i++){
22                 echo "Indeks ke-[$i] : ",$name[$i],"<br>";
23                 echo "Jika dibalik akan menjadi '", strrev($name[$i]),"' <br><br>";
24             }
25         ?>
26     </p>
27 </body>
28 </html>

```

Gambar 2. 7 Sintaks PHP 3 - Menampilkan kebalikan dari nilai yang diinput dan nilai masing-masing indeks variabel array (versi kedua).

Berikut ini merupakan output dari sintaks yang dibuat, yang dibuka melalui browser.



Gambar 2. 8 Output Sintaks PHP 3 - Menampilkan kebalikan dari nilai yang diinput dan nilai masing-masing indeks variabel array (versi kedua).

2.4 Program PHP menampilkan jumlah huruf konsonan dan vokal

Program yang dibuat akan menampilkan jumlah huruf konsonan dan huruf vokal pada nilai yang ada di masing-masing indeks variabel array. Sintaks yang digunakan sama dengan sintaks pada tugas praktikum nomor sebelumnya, namun dengan tambahan fungsi yang digunakan sebagai solusi pada tugas praktikum nomor ini.

Berikut ini merupakan sintaks fungsi yang ditambahkan.

```
<?php
    $nama=["OscarOkt", "Oktorian Amario", "Almando bin Aliando",
    "Salman Simpati", "Suka Suka"];

    function hitungVokal($kata){
        $jumlah=0;
        for($i=0; $i<strlen($kata); $i++){
            if($kata[$i]=='a' || $kata[$i]=='i' || $kata[$i]=='u'
            || $kata[$i]=='e' || $kata[$i]=='o' ||
```

```

        $kata[$i]=='A' || $kata[$i]=='T' || $kata[$i]=='U'
// $kata[$i]=='E' || $kata[$i]=='O')
        {
            $jumlah++;
        } else {
            continue;
        }
    }
    echo "Jumlah huruf vokal dalam kata '$kata,' adalah
    ",$jumlah;
}
function hitungKonsonan($kata){
    $jumlah=0;
    $spasi=(str_word_count($kata)-1);
    for($i=0; $i<strlen($kata); $i++){
        if($kata[$i]=='a' || $kata[$i]=='i' || $kata[$i]=='u'
// $kata[$i]=='e' || $kata[$i]=='o' ||
        $kata[$i]=='A' || $kata[$i]=='T' || $kata[$i]=='U'
// $kata[$i]=='E' || $kata[$i]=='O')
        {
            continue;
        } else {
            $jumlah++;
        }
    }
    $jumlah-=$spasi;
    echo "Jumlah huruf konsonan dalam kata '$kata,' adalah
    ",$jumlah;
}
?>

```

Fungsi pertama yaitu fungsi *hitungVokal*. Fungsi ini mempunyai parameter yang akan menampung nama yang telah diinput. Di dalam fungsi

ini, terdapat variabel lokal *\$jumlah* yang nantinya akan menampung jumlah huruf vokal dan di-return ke fungsi. Kemudian terdapat perintah perulangan *for* untuk memeriksa setiap huruf dalam variabel *\$kata*. Di dalam perulangan *for*, terdapat sintaks percabangan *if*, yang mana akan memeriksa apakah huruf tersebut merupakan huruf vokal atau tidak. Kondisi yang ditentukan pada percabangan adalah jika huruf tersebut sama dengan huruf vokal (a,i,u,e,o,A,I,U,E atau O) maka variabel *\$jumlah* akan bernilai ditambah 1 (*\$jumlah++*). Jika tidak maka akan melewati baris perintah dengan kata kunci *continue*. Setelah semua huruf diperiksa, barulah perintah untuk menampilkan informasi mengenai jumlah huruf vokal dijalankan dengan menggunakan variabel *\$jumlah* yang telah diakumulasi dan *\$kata* sebagai acuan dalam menghitung huruf vokal.

Fungsi *hitungKonsonan()* sebenarnya mempunyai sintaks yang sama dengan fungsi *hitungVokal()*, namun perbedaannya terletak pada baris yang akan dieksekusi di dalam percabangan *if-nya*. Perintah untuk mengakumulasi *\$jumlah* akan dijalankan jika suatu huruf **tidak** termasuk dalam huruf vokal. Kemudian perbedaan selanjutnya adalah terdapat variabel *\$spasi* yang mana menampung jumlah spasi (nilai didapat dari hasil fungsi *str_word_count()* dikurangi 1). Variabel *\$spasi* akan dijadikan pengurang terhadap variabel *\$jumlah* sehingga mendapat jumlah konsonan yang presisi tanpa menghitung sebuah spasi.

Berikut ini merupakan sintaks lengkap dari tugas praktikum nomor 4 ini.

```
<!DOCTYPE html>

<?php

    $nama=["OscarOkt", "Oktorian Amario", "Almando bin Aliando",
"Salman Simpati", "Suka Suka"];

    function hitungVokal($kata){

        $jumlah=0;
```



```

        for($i=0; $i<strlen($kata); $i++){

            if($kata[$i]=='a' || $kata[$i]=='i' || $kata[$i]=='u'
// $kata[$i]=='e' || $kata[$i]=='o' ||

                $kata[$i]=='A' || $kata[$i]=='T' || $kata[$i]=='U'
// $kata[$i]=='E' || $kata[$i]=='O')

                {

                    $jumlah++;

                } else {

                    continue;

                }

            }

            echo "Jumlah huruf vokal dalam kata '$kata' adalah
", $jumlah;

        }

function hitungKonsonan($kata){

    $jumlah=0;

    $spasi=(str_word_count($kata)-1);

    for($i=0; $i<strlen($kata); $i++){

        if($kata[$i]=='a' || $kata[$i]=='i' || $kata[$i]=='u'
// $kata[$i]=='e' || $kata[$i]=='o' ||

                $kata[$i]=='A' || $kata[$i]=='T' || $kata[$i]=='U'
// $kata[$i]=='E' || $kata[$i]=='O')

                {

```

```

        continue;

    } else {

        $jumlah++;

    }

}

$jumlah-=$spasi;

echo "Jumlah huruf konsonan dalam kata '$kata' adalah
", $jumlah;

}

?>

<html>

    <head>

        <title> Modul 1 - 4 | Oscar Oktorian Almando </title>

    </head>

    <body>

        <h2>Jumlah huruf vokal dan konsonan pada nilai di
masing-masing indeks Array : </h2>

        <p align="left">

            <?php

                for($i=0; $i<count($nama); $i++){

                    echo "Indeks ke-[$i] : '$nama[$i],'<br>";

                    hitungVokal($nama[$i]);

                    echo "<br>";

```

```

        hitungKonsonan($nama[$i]);

        echo "<br><br>";

    }

?>

</p>

</body>

</html>

```

Pemanggilan nilai di masing-masing indeks variabel array sama seperti pemanggilan nilai di tugas praktikum nomor sebelumnya, dengan perbedaan yaitu pemanggilan fungsi *hitungVokal()* dan *hitungKonsonan()*.

Berikut ini merupakan tampilan sintaks yang ada pada aplikasi notepad++.

```

1  <!DOCTYPE html>
2  <?php
3      $nama=["OscarOkt", "Oktorian Amario", "Almando bin Aliando", "Salman Simpati", "Suka Suka"];
4
5
6      function hitungVokal($kata){
7          $jumlah=0;
8          for($i=0; $i<strlen($kata); $i++){
9              if($kata[$i]=='a' || $kata[$i]=='i' || $kata[$i]=='u' || $kata[$i]=='e' || $kata[$i]=='o' ||
10                 $kata[$i]=='A' || $kata[$i]=='I' || $kata[$i]=='U' || $kata[$i]=='E' || $kata[$i]=='O')
11              {
12                  $jumlah++;
13              } else {
14                  continue;
15              }
16          }
17          echo "Jumlah huruf vokal dalam kata '$kata,' adalah '$jumlah';
18      }
19
20      function hitungKonsonan($kata){
21          $jumlah=0;
22          $spasi=(str_word_count($kata)-1);
23          for($i=0; $i<strlen($kata); $i++){
24              if($kata[$i]=='a' || $kata[$i]=='i' || $kata[$i]=='u' || $kata[$i]=='e' || $kata[$i]=='o' ||
25                 $kata[$i]=='A' || $kata[$i]=='I' || $kata[$i]=='U' || $kata[$i]=='E' || $kata[$i]=='O')
26              {
27                  continue;
28              } else {
29                  $jumlah++;
30              }
31          }
32          $jumlah-=$spasi;
33          echo "Jumlah huruf konsonan dalam kata '$kata,' adalah '$jumlah';
34      }
35  }
36  ?>

```

Gambar 2. 9 Sintaks PHP 4- Menampilkan jumlah konsonan dan vokal pada nilai di masing-masing indeks variabel array (1).

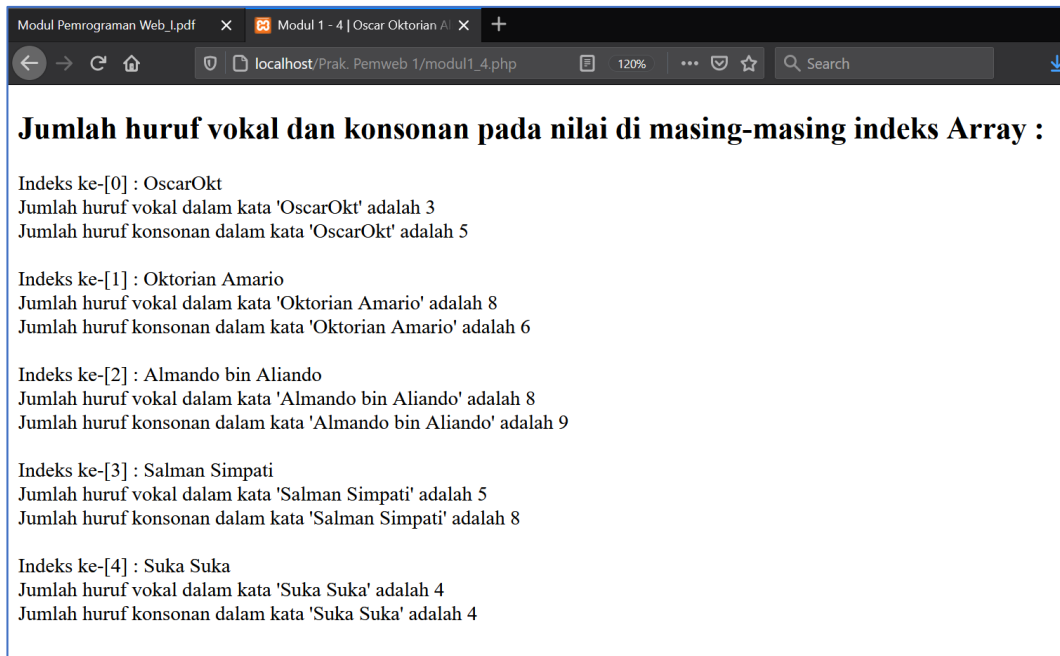
```

35
36
37 <html>
38   <head>
39     <title> Modul 1 - 4 | Oscar Oktorian Almando </title>
40   </head>
41   <body>
42     <h2>Jumlah huruf vokal dan konsonan pada nilai di masing-masing indeks Array : </h2>
43     <p align="left">
44       <?php
45         for($i=0; $i<count($nama); $i++){
46           echo "Indeks ke-[$i] : ",$nama[$i],"<br>";
47           hitungVokal($nama[$i]);
48           echo "<br>";
49           hitungKonsonan($nama[$i]);
50           echo "<br><br>";
51         }
52       ?>
53     </p>
54   </body>
55 </html>

```

Gambar 2. 10 Sintaks PHP 4- Menampilkan jumlah konsonan dan vokal pada nilai di masing-masing indeks variabel array (2).

Berikut ini merupakan output dari sintaks yang dibuat.



Jumlah huruf vokal dan konsonan pada nilai di masing-masing indeks Array :

Indeks ke-[0] : OscarOkt
 Jumlah huruf vokal dalam kata 'OscarOkt' adalah 3
 Jumlah huruf konsonan dalam kata 'OscarOkt' adalah 5

Indeks ke-[1] : Oktorian Amario
 Jumlah huruf vokal dalam kata 'Oktorian Amario' adalah 8
 Jumlah huruf konsonan dalam kata 'Oktorian Amario' adalah 6

Indeks ke-[2] : Almando bin Aliando
 Jumlah huruf vokal dalam kata 'Almando bin Aliando' adalah 8
 Jumlah huruf konsonan dalam kata 'Almando bin Aliando' adalah 9

Indeks ke-[3] : Salman Simpati
 Jumlah huruf vokal dalam kata 'Salman Simpati' adalah 5
 Jumlah huruf konsonan dalam kata 'Salman Simpati' adalah 8

Indeks ke-[4] : Suka Suka
 Jumlah huruf vokal dalam kata 'Suka Suka' adalah 4
 Jumlah huruf konsonan dalam kata 'Suka Suka' adalah 4

Gambar 2. 11 Output Sintaks PHP 4- Menampilkan jumlah konsonan dan vokal pada nilai di masing-masing indeks variabel array.

BAB III

KESIMPULAN

Berdasarkan hasil dari tugas praktikum yang dikerjakan, hal yang dapat disimpulkan dari praktikum modul 1 ini adalah bahwa PHP merupakan sebuah bahasa pemrograman yang dikhususkan untuk web-programming dan side-server programming. PHP dapat digunakan bersamaan dengan bahasa pemrograman web lainnya ataupun bahasa mark-up. Sintaks yang digunakan pada PHP sama halnya seperti bahasa pemrograman lainnya, namun berbeda cara penulisannya serta aturan penulisan yang berlaku.

DAFTAR PUSTAKA

- Awwaabiin, Salmaa. 2020. "Apa Itu PHP? Pengertian, Fungsi Dan Sintaks Dasarnya!" <https://www.niagahoster.co.id/blog/pengertian-php/> (March 27, 2021).
- Jurusan Teknik Informatika. 2021. "MODUL PRAKTIKUM PEMROGRAMAN WEB I."
- Pratama, Andre. 2014a. "Fungsi Dan Cara Penulisan Perintah Continue Dalam Perulangan PHP." <https://www.duniailkom.com/tutorial-belajar-php-fungsi-dan-cara-penulisan-continue-dalam-perulangan-php/> (March 27, 2021).
- . 2014b. "Pengertian Dan Cara Penulisan Perulangan For Dalam PHP | Duniailkom." <https://www.duniailkom.com/tutorial-belajar-php-pengertian-dan-cara-penulisan-perulangan-for-dalam-php/> (March 27, 2021).
- . 2014c. "Pengertian Fungsi (Function) PHP Dan Cara Penggunaan Fungsi." <https://www.duniailkom.com/tutorial-belajar-php-pengertian-fungsi-function-php-dan-cara-penggunaan-fungsi-php/> (March 27, 2021).