

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



NAMA : Kevin Jeremia S.P
NIM : 193020503034
KELAS : A PHP (Hypertext Preprocessor)
MODUL : I

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

BAB I

TUJUAN DAN LANDASAN TEORI

1.1 Tujuan Praktikum

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

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

1.2.1 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

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 kita bisa memiliki variabel lokal dengan nama yang sama pada fungsi

yang berbeda, karena variabel lokal hanya dikenali oleh fungsi yang mendeklarasikannya saja. Keyword global digunakan untuk mengakses variabel global dari dalam fungsi. Php juga menyimpan semua variabel global dalam sebuah array yang disebut \$GLOBALS[index]. Index adalah nama dari variabel, array ini juga bisa diakses dari dalam fungsi dan bisa digunakan untuk mengupdate variabel global secara langsung. Normalnya, ketika fungsi dieksekusi, semua variabel-variabelnya dihapus. Namun terkadang kita perlu untuk mempertahankan nilai variabel lokal supaya tidak dihapus. Untuk itu, gunakan keyword static dalam pendeklarasian variabel yang nilainya ingin dipertahankan.

```
<?php
    $x = 5; // 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 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 PHP

```

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

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

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

```

Gambar 1.5 PHP

```

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

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

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

```

Gambar 1.6 PHP

```

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

```

Gambar 1.7 PHP

```

        echo $x;
        $x++;
    }

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

```

Gambar 1.8 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 and Print

1.2.2 Tipe Data PHP

Variabel bisa menyimpan data dengan tipe yang berbeda-beda, berikut adalah tipe-tipe data yang didukung oleh : PHP String, Integer, Float, Boolean, Array, Object, NULL, Resource

1.2.3 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 1.10 Tipe data

1.2.4 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 1.11 Tipe Data

1.2.5 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 1.12 Tipe Data

1.2.6 Boolean

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

1.2.7 Object / objek

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 1.13 Object/objek

1.2.8 Nilai NULL

NULL adalah tipe data khusus yang bisa memiliki hanya satu nilai: NULL. Sebuah variabel dengan tipe data NULL adalah variabel yang tidak memiliki nilai. Jika sebuah variabel dibuat tanpa nilai, maka otomatis akan ditentukan dengan nilai NULL. Variabel bisa juga dikosongkan dengan mengatur nilainya menjadi NULL:


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

Gambar 1.14 Nilai Null

1.2.9 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.

1.2.10 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 1.15 Manipulasi String

Hasil dari kode tersebut adalah panjang dari string “Hello world!” yaitu 12 karakter. Panjang string juga bisa diukur perkata menggunakan fungsi `str_word_count()` seperti contoh berikut:

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

Gambar 1.16 Manipulasi String

Keluaran dari kode tersebut berupa jumlah kata pada string “Hello world!” yaitu 2. PHP juga memiliki fungsi yang bisa digunakan untuk membalikkan susunan string, yaitu menggunakan fungsi `strrev()` seperti contoh kode program berikut yang menghasilkan string “!dlrow olleH” dari inputan string “Hello world!”.

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

Gambar 1.17 Manipulasi String

PHP juga memiliki fungsi untuk menemukan teks yang spesifik didalam sebuah string, yaitu dengan menggunakan fungsi strpos(). Jika teks yang dicari ditemukan didalam string, maka fungsi akan mengembalikan index posisi dari karakter awal teks pertama yang cocok. Jika tidak ditemukan, maka fungsi strpos() akan mengembalikan nilai FALSE. Kode program berikut adalah contoh dari penggunaan fungsi strpos() pada string “Hello world!” untuk menemukan teks “world”, sehingga keluaran yang dihasilkan adalah 6.

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

Gambar 1.18 PHP

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

Gambar 1.19 PHP

1.2. 11 Konstanta / Constant

Konstanta seperti variabel, tetapi setelah dideklarasikan nilai konstanta tidak bisa dirubah. Konstanta adalah nama atau indintifikasi untuk nilai sederhana. Nilainya tidak bisa diubah selama program berjalan. Nama konstanta yang valid dimulai dengan huruf atau garis bawah, tidak ada tanda \$ sebelum nama konstanta. Konstanta memiliki lingkup global. Gunakan fungsi define(nama, nilai, case-sensitif) untuk mendeklarasikan konstanta. Parameter nama adalah nama dari konstanta yang akan digunakan, parameter nilai adalah

nilai dari konstanta yang akan digunakan pada program, dan parameter case-sensitive digunakan untuk menspesifikasikan apakah nama konstanta harus case-sensitive. Secara default, nilai case-sensitive adalah false.

```
<?php
// nama constant adalah case-sensitive
define("GREETING", "Welcome to W3Schools.com!");
echo GREETING;
?>
```

Gambar 1.20 Konstanta / Constant

1.2. 12 Operator PHP

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

1.2. 13 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 1.21 Operator

1.2. 14 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 1.22 Operator

1.2. 15 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 1.23 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 1.24 Operator

1.2. 16 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 1,25 Operator

1.2. 17 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 1.26 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 1.27 Operator

1.2. 18 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 1.28 Operator

1.2. 19 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 1.29 Operator

1.2. 20 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. Syntakdari kondisi ini adalah sebagai berikut:

```
if (kondisi) {  
    kode yang akan dieksekusi jika kondisi true; }  

```

1.2. 21 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. }  

```

1.2. 22 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. }
```

1.2. 23 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. }
```

1.2. 24 Perulangan

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

1.2. 25 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. }
```

1.2. 26 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 );
```

1.2. 27 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.

1.2. 28 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.

1.2. 29 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 1.30 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 1.31 Fungsi

1.2. 30 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:

1.2. 31 HTML

Hypertext Markup Language (HTML) adalah bahasa markup yang umum digunakan untuk membuat halaman web. Sebenarnya HTML bukanlah sebuah bahasa pemrograman. Apabila di tinjau dari namanya, HTML merupakan bahasa markup atau penandaan terhadap sebuah dokumen teks. Tanda tersebut di gunakan untuk menentukan format atau style dari teks yang di tandai.

HTML dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN dan dipopulerkan pertama kali oleh browser Mosaic. Selama awal tahun 1990 HTML mengalami perkembangan yang sangat pesat. Setiap pengembangan HTML pasti akan menambahkan kemampuan dan fasilitas yang lebih baik dari versi sebelumnya.

Ciri-ciri dari dokumen HTML :

1. Terdapat Tag HTML

- a. Tag ditandai dengan “”.
- b. Dalam penulisan kode HTML harus diawali dengan < dan diakhiri dengan >.
- c. Tag HTML tidak ‘case sensitive’, sama dengan .
- d. Jika dalam suatu ag ada tag lagi maka penulisan tag akhir harus berurutan, misal ... *Huruf Tebal Miring* .

2. Terdapat ELEMENT HTML, yang terdiri dari 2 bagian :

- a. HEAD merupakan kepala dokumen dari HTML, yang berfungsi untuk memberikan informasi tentang dokumen.
- b. BODY merupakan tubuh dari HTML, yang berfungsi untuk menentukan bagaimana isi suatu dokumen yang akan ditampilkan pada web browser.

Struktur dari dokumen HTML adalah sebagai berikut :

struktur	Keterangan
<HTML>	Tag awal HTML, sebagai tanda awal dokumen HTML
<HEAD> </HEAD>	Head HTML: untuk tag informasi halaman judul, base, isindex, link, script, style & meta.
<BODY> </BODY>	Body HTML: untuk bagian isi dari dokumen dan di dalamnya terdapat atribut seperti background, Warna teks, warna link, warna visited link, warna active link, dll.
</HTML>	Tag penutup HTML, sebagai tanda penutup dokumen HTML

Gambar 1.32 Pendklarasian HTML

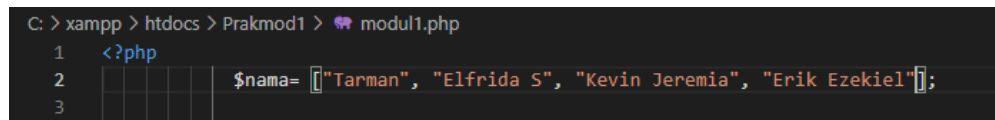
BAB II

PEMBAHASAN

Pada praktikum ini diminta untuk membuat sebuah program web untuk mengolah nama nama anggota keluarga. Dalam membuat program web ini menggunakan Bahasa PHP dan untuk tampilan pada web menggunakan Bahasa HTML. Berikut ini adalah kriteria – kriteria untuk mengolah nama nama anggota keluarga dan penjelasan nya.

2.1 Nama – nama disimpan didalam Variable array

Di nomor yang pertama ini diminta untuk menyimpan nama anggota keluarga yang ingin di olah kedalam Variable array. Di gambar 2.1 di bawah ini dapat dilihat nama nama anggota keluarga yang sudah di simpan kedalam array. Nama nama tersebut di simpan dalam Variable array dengan nama Variable \$nama.

A screenshot of a code editor window. The title bar shows the path 'C: > xampp > htdocs > Prakmod1 > modul1.php'. The code is as follows:

```
1 <?php
2 $nama= ["Tarman", "Elfrida S", "Kevin Jeremia", "Erik Ezekiel"];
3
```

Gambar 2.1 Variable Array

Array adalah variabel khusus, yang dapat menampung lebih dari satu nilai pada satu waktu. Oleh sebab itu variable ini sangat cocok untuk menyimpan banyak data seperti nama anggota keluarga. Nama nama yang telah di simpan ini nanti akan di lakukan pengolahan.

2.2 Program menampilkan jumlah kata dan jumlah huruf pada nama yang anda tuliskan

Selanjutnya di minta untuk menampilkan jumlah kata dan jumlah huruf yang telah di simpan sebelumnya di dalam array. Untuk menampilkan jumlah kata dapat menggunakan fungsi `str_word_count` untuk menghitung kalimat nya. Untuk mengaktifkan nya fungsi tinggal di panggil seperti gambar 2.2 . Sementara itu

untuk menghitung jumlah huruf menggunakan sebuah fungsi dengan nama `jumlahHuruf` seperti gambar 2.3.

```
<td><?php echo str_word_count($name)."<br>"; ?></td>  
<td><?php echo jumlahHuruf($name)."<br>"; ?></td>
```

Gambar 2.2 Pemanggilan Fungsi

Fungsi di gambar 2.3 Memiliki cara kerja memmbalikan nilai dari fungsi `strlen` dikurangi fungsi `substr_count`. `Strln` berfungsi menghitung Panjang dari string dan `substr_count` berfungsi untuk menghitung berapa kali substring spasi. Dari kombinasi ini fungsi akan mengembalikan hanyalah panjag dari stirng tersebut tanpa ikut menghitung karakter spasi.

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

Gambar 2.3 Fungsi jumlahHuruf

2.3 Program menampilkan kebalikan dari nama yang diinputkan.

Selanjut nya di nomor yang ketiga ini diminta untuk menampilkan nama yang telah di simpan di dalam array. Namun nama yang akan di tampilkan akan dalam posisi yang terbalik. Di dalam PHP terdapat sebuah fungsi manipulasi array yang dapat di gunakan yaitu `strrev`. `Strrev` berfungsi untuk membalikan susunan karakter string dapat dilihat pada gambar 2.4. Di gambar 2.4 pemanggilan fungsi `strev` dengan parameter nya adalah array lalu di tampilkan dengan `Echo`.

```
<td><?php echo strrev($name)."<br>"; ?></td>
```

Gambar 2.4 Menampilkan Nama terbalik

2.4 Program menampilkan jumlah konsonan dan jumlah vokal pada nama yang telah diinputkan.

Dan ini nomor yang terakhir ini diminta untuk menampilkan jumlah huruf konsonan dan jumlah huruf vokal. Huruf yang akan di tampilkan berasal dari array yang berisi nama nama anggota keluarga. Huruf Vokal terdiri dari A, I, U, E, O dan sedangkan konsonan merupakan huruf selain huruf vokal. Di dalam program yang telah di buat ini untuk menampilkan jumlah konsonan dan jumlah vocal di bagi menjadi 2 buah fungsi. Fungsi Vokal seperti di gambar 2.5 dan fungsi konsonan di gambar 2.6.

```
3
4  ✓
5  function vokal($kalimat) {
6      $a = substr_count($kalimat, 'a');
7      $i = substr_count($kalimat, 'i');
8      $u = substr_count($kalimat, 'u');
9      $e = substr_count($kalimat, 'e');
10     $o = substr_count($kalimat, 'o');
11     $A = substr_count($kalimat, 'A');
12     $I = substr_count($kalimat, 'I');
13     $U = substr_count($kalimat, 'U');
14     $E = substr_count($kalimat, 'E');
15     $O = substr_count($kalimat, 'O');
16
17     $count = ($a+$i+$u+$e+$o+$A+$I+$U+$E+$O);
18
19     return $count;
20 }
```

Gambar 2.5 Fungsi Vokal

Gambar 2.5 merupakan fungsi untuk menghitung jumlah huruf vocal. Fungsi ini memiliki parameter \$kalimat yang berisi objek. Di dalam fungsi ini berisi variabel dari huruf vocal yang kapital mau tidak. Di dalam setiap variable nya memiliki substr_count yang berfungsi untuk mencari setiap huruf. Dan pada variable \$count di hitung berapa banyak huruf vocal yang di temukan didalam nama tersebut kemudian fungsi ini memberikan sebuah nilai balikan yang akan di tampilkan pada halaman web, untuk pemanggilan fungsi dapat di lihat pada gamabr 2.7.

```

21 function konsonan($kalimat) {
22     $jumlah = strlen($kalimat)-substr_count($kalimat, ' ');
23     $a = substr_count($kalimat, 'a');
24     $i = substr_count($kalimat, 'i');
25     $u = substr_count($kalimat, 'u');
26     $e = substr_count($kalimat, 'e');
27     $o = substr_count($kalimat, 'o');
28     $A = substr_count($kalimat, 'A');
29     $I = substr_count($kalimat, 'I');
30     $U = substr_count($kalimat, 'U');
31     $E = substr_count($kalimat, 'E');
32     $O = substr_count($kalimat, 'O');
33
34     $count = $jumlah - ($a+$i+$u+$e+$o+$A+$I+$U+$E+$O);
35
36     return $count;
37 }

```

Gambar 2.6 Fungsi Konsonan

Selanjutnya 2.6 merupakan fungsi untuk menghitung jumlah huruf konsonan. Sama seperti fungsi vocal fungsi ini memiliki parameter \$kalimat yang berisi objek. Cara kerja fungsi ini sama seperti seperti fungsi vocal, namun terdapat yang berbeda yaitu variabel \$jumlah. Variable ini punya fungsi strlen yang menghitung Panjang karakter dan dikurangi dengan substr_count “ ” untuk tidak menghitung karakter spasi. Setiap huruf vocal kapital maupun tidak di hitung untuk digunakan pada variable \$count. \$count merupakan jumlah huruf konsonan karena variable \$jumlah- seluruh jumlah variable vocal. Setelah itu di dapatkanlah hasil huruf konsonan nya.

```

<td><?= konsonan( $name); ?></td>
<td><?= vokal( $name); ?></td>

```

Gambar 2.7 Memanggil Fungsi Konsonan Dna Fungsi Vokal

Dan Di gambar 2.7 adalah pemanggilan fungsi fungsi. Nya. Setiap fungsi di tampilkan memiliki parameter \$name yang merupakan alias Dari \$nama yang berisi nama nama anggota keluarga di dalam string.

2.5 Baris Program HTML dan Tampilan di Web

Pada 2.5 ini akan di jelaskan baris program html dan tampilan dari web nya. HTML berfungsi untuk menampilkan data yang telah diolah di Bahasa PHP. Tampilan dari web ini sangat sederhana karena hanya menampilkan table yang berisi nama dan kondisi nama yang telah di olah. Dapat dilihat tampilan web pada gambar 2.8 dan baris program dapat di lihat pada gambar 2.9

Nama : Kevin Jeremia S. P
NIM : 193020503034

Pengolahan Nama Anggota Keluarga

No	Nama	Jumlah Kata	Jumlah Huruf	Kebalikan Nama	Konsonan	Vokal
1	Tarman	1	6	namraT	4	2
2	Elfrida S	2	8	S adirflE	5	3
3	Kevin Jeremia	2	12	aimereJ niveK	6	6
4	Erik Ezekiel	2	11	leikezE kirE	5	6

Gambar 2.8 Tampilan Program di web

Baris program html untuk membuat table di mulai dari body nya. Untuk membuat table menggunakan perintah `<table>` lalu dengan `border="2"` `cellpadding="10"` `cellspacing="1"` dan di table di letakan di tengah dengan `align=center`. Setelah itu untuk table head nya di berikan no, Nama, Jumlah Kata, Jumlah Huruf, Kebalikan Nama, Konsonan, Vokal. Untuk pengisian data setiap kolom menggunakan bahas PHP dan melakukan perulangan Foreach. Lalu setiap fungsi akan di echo secara berulang dengan parameter nya \$nama yang berisi stirng. Yang artinya akan di lakukan perulangan sebanayak data yang terdapat di dalam array.

```
<!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>
Nama : Kevin Jeremia S. P
<Br>NIM : 193020503034<Br>
<h3 align=center>Pengolahan Nama Anggota Keluarga</h3>
<table border="2" cellpadding="10" cellspacing="1" align=center>
<thead>
<tr>
<th>No</th>
<th>Nama</th>
<th>Jumlah Kata</th>
<th>Jumlah Huruf</th>
<th>Kebalikan Nama</th>
<th>Konsonan</th>
<th>Vokal</th>
</tr>
</thead>
<?php $i = 1; ?>
<?php foreach ($nama as $name) : ?>

<tr>
<td><?= $i; ?></td>
<td><?php echo $name."<br>"; ?></td>
<td><?php echo str_word_count($name)."<br>"; ?></td>
<td><?php echo jumlahHuruf($name)."<br>"; ?></td>
<td><?php echo strrev($name)."<br>"; ?></td>
<td><?= konsonan( $name); ?></td>
<td><?= vokal( $name); ?></td>
</tr>
<?php $i++; ?>
<?php endforeach; ?>
</table>

</body>
</html>

```

Gambar 2.9 Baris Program HTML

BAB III

Kesimpulan

Dari praktikum ini dapat di simpulkan bahwa PHP (Hypertext Preprocessor) merupakan sebuah bahasa pemrograman yang memiliki berbagai banyak fitur. Bahasa Pemrograman PHP di gunakan untuk mendevelop Pembuatan Back End sebuah website. Bahasa PHP memiliki banyak fitur seperti Penggunaan bermacam variable, dapat menggunakan perulangan, Meyabungkan website dengan database, Memanipulasi Array dan hal lain nya yang sangat berguna dalam proses pembuatan website. Selain itu Bahasa PHP bisa di kombinasikan dengan Bahasa HTML. HTML digunakan untuk mengatur tampilan antar muka di sebuah website agar website dapat menampilkan sebuah data yang sangat baik.

BAB IV

DAFTAR PUSTAKA

- Nuryake Fajaryati, M. P. (1999). Hypertext Markup Language (HTML). *World Wide Web*, 191–291. https://doi.org/10.1007/978-3-642-59944-6_6
- Praktikum, K. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

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 = 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 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 PHP

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

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

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

Gambar 1.5 PHP

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

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

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

Gambar 1.6 PHP

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

Gambar 1.7 PHP

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

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

Gambar 1.8 PHP

```
<?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 and Print

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

Gambar 1.10 Tipe data

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

Gambar 1.11 Tipe Data

```
<?php
    $x = 10.365;

    var_dump($x);
?>
```

Gambar 1.12 Tipe Data

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

Gambar 1.13 Object/objek

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

Gambar 1.14 Nilai Null

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

Gambar 1.15 Manipulasi String

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

Gambar 1.16 Manipulasi String

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

Gambar 1.17 Manipulasi String

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

Gambar 1.18 PHP


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

Gambar 1.19 PHP

```
<?php
// nama constant adalah case-sensitive
define("GREETING", "Welcome to W3Schools.com!");
echo GREETING;
?>
```

Gambar 1.20 Konstanta / Constant

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 1.21 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 1.22 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 1.23 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 1.24 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 1.25 Operator

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 1.26 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 1.27 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 1.28 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 1.29 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 1.30 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 1.31 Fungsi

Struktur dari dokumen HTML adalah sebagai berikut :

struktur	Keterangan
<HTML>	Tag awal HTML, sebagai tanda awal dokumen HTML
<HEAD> </HEAD>	Head HTML: untuk tag informasi halaman judul, base, isindex, link, script, style & meta.
<BODY> </BODY>	Body HTML: untuk bagian isi dari dokumen dan di dalamnya terdapat atribut seperti background, Warna teks, warna link, warna visited link, warna active link, dll.
</HTML>	Tag penutup HTML, sebagai tanda penutup dokumen HTML

Gambar 1.32 Pendklarasian HTML

```
C: > xampp > htdocs > Prakmod1 > modul1.php
1  <?php
2      $nama= ["Tarman", "Elfrida S", "Kevin Jeremia", "Erik Ezekiel"];
3
```

Gambar 2.1 Variable Array

```
<td><?php echo str_word_count($name)."<br>"; ?></td>
<td><?php echo jumlahHuruf($name)."<br>"; ?></td>
```

Gambar 2.2 Pemanggilan Fungsi

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

Gambar 2.3 Fungsi jumlahHuruf

```
<td><?php echo strrev($name)."<br>"; ?></td>
```

Gambar 2.4 Menampilkan Nama terbalik

```

3
4  ✓
5     function vokal($kalimat) {
6         $a = substr_count($kalimat, 'a');
7         $i = substr_count($kalimat, 'i');
8         $u = substr_count($kalimat, 'u');
9         $e = substr_count($kalimat, 'e');
10        $o = substr_count($kalimat, 'o');
11        $A = substr_count($kalimat, 'A');
12        $I = substr_count($kalimat, 'I');
13        $U = substr_count($kalimat, 'U');
14        $E = substr_count($kalimat, 'E');
15        $O = substr_count($kalimat, 'O');
16
17        $count = ($a+$i+$u+$e+$o+$A+$I+$U+$E+$O);
18
19        return $count;
20    }

```

Gambar 2.5 Fungsi Vokal

```

21     function konsonan($kalimat) {
22         $jumlah = strlen($kalimat)-substr_count($kalimat, ' ');
23         $a = substr_count($kalimat, 'a');
24         $i = substr_count($kalimat, 'i');
25         $u = substr_count($kalimat, 'u');
26         $e = substr_count($kalimat, 'e');
27         $o = substr_count($kalimat, 'o');
28         $A = substr_count($kalimat, 'A');
29         $I = substr_count($kalimat, 'I');
30         $U = substr_count($kalimat, 'U');
31         $E = substr_count($kalimat, 'E');
32         $O = substr_count($kalimat, 'O');
33
34         $count = $jumlah - ($a+$i+$u+$e+$o+$A+$I+$U+$E+$O);
35
36         return $count;
37     }

```

Gambar 2.6 Fungsi Konsonan

```

<td><?= konsonan( $name); ?></td>
<td><?= vokal( $name); ?></td>

```

Gambar 2.7 Memanggil Fungsi Konsonan Dna Fungsi Vokal

Nama : Kevin Jeremia S. P
NIM : 193020503034

Pengolahan Nama Anggota Keluarga

No	Nama	Jumlah Kata	Jumlah Huruf	Kebalikan Nama	Konsonan	Vokal
1	Tarman	1	6	namraT	4	2
2	Elfrida S	2	8	S adirflE	5	3
3	Kevin Jeremia	2	12	aimereJ niveK	6	6
4	Erik Ezekiel	2	11	leikezE kirE	5	6

Gambar 2.8 Tampilan Program di web

```
<!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>
Nama : Kevin Jeremia S. P
<Br>NIM : 193020503034<Br>
<h3 align=center>Pengolahan Nama Anggota Keluarga</h3>
  <table border="2" cellpadding="10" cellspacing="1" align=center>
    <thead>
      <tr>
        <th>No</th>
        <th>Nama</th>
        <th>Jumlah Kata</th>
        <th>Jumlah Huruf</th>
        <th>Kebalikan Nama</th>
        <th>Konsonan</th>
        <th>Vokal</th>
      </tr>
    </thead>
    <?php $i = 1; ?>
    <?php foreach ($nama as $name) : ?>

      <tr>
```

```

        <td><?= $i; ?></td>
        <td><?php echo $name."<br>"; ?></td>
        <td><?php echo str_word_count($name)."<br>"; ?></td>
        <td><?php echo jumlahHuruf($name)."<br>"; ?></td>
        <td><?php echo strrev($name)."<br>"; ?></td>
        <td><?= konsonan( $name); ?></td>
        <td><?= vokal( $name); ?></td>
    </tr>
    <?php $i++; ?>
<?php endforeach; ?>
</table>

</body>
</html>

```

Gambar 2.9 Baris Program HTML