

**LAPORAN HASIL PRAKTIKUM
PEMROGAMAN WEB & MOBILE**



NAMA : STEVEN SULING
NIM : 193030503058
KELAS : A
MODUL : I PHP (Hypertext Preprocessor)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALNGKA RAYA
2021

BAB I

TUJUAN DAN LANDASAN TEORI

Tujuan

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

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

Script PHP bisa diletakkan di manapun dalam dokumen, dimulai dengan tanda “<?php” dan diakhiri 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!”

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 =
    "W3Schoo
    ls.com";
    echo "I
    love
    $txt!";

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

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.

```
<?ph
p
    $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();

?
>
```

```
<?php
    function myTest() {
        $x = 5; // Lingkup lokal

        echo "<p>Variabel x di dalam fungsi adalah :
        $x</p>";
    }
?>
```

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

    function myTest()
    {
        global $x,
        $y;

        $y = $x +
        $y;
    }

    myTest();

    echo $y; // output
    15

?>
```

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

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

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

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

```
?>
```

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.

```
<?ph
    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.";

?>
```

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

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
?
```

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
?>
```


Float

Float (bilangan floating-point) adalah bilangan dengan tanda desimal atau bilangan dalam bentuk eksponensial. Berikut contoh penggunaan variabel float.

```
<?php
    $x = 10.365;

?>
```

Boolean

Boolean merepresentasikan dua kemungkinan keadaan: TRUE atau FALSE. Boolean sering digunakan pada kondisi percobaan. Berikut contoh penggunaannya:

```
$x = true;
$y = false;
```

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

?>
```

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!";

    $x = null;
    var_dump($
?>
```

Resource

Tipe khusus resource bukanlah merupakan tipe data yang sebenarnya. Ia menyimpan referensi fungsi dan sumberdaya eksternal untuk PHP. Contoh umum penggunaan tipe data resource adalah pada pemanggilan database.

Fungsi-fungsi Manipulasi String

String merupakan susunan dari karakter yang memiliki fungsi khusus untuk mengolahnya. Seperti fungsi `strlen()`, berfungsi menghitung panjang dari string. Contohnya sebagai berikut:

```
<?php
    echo strlen("Hello
?>
```

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
?>
```

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
?>
```

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!",
?>
```

PHP juga memiliki fungsi untuk mengganti teks didalam string, yaitu dengan menggunakan fungsi `str_replace()`. Contoh berikut adalah kode program yang digunakan untuk mengganti teks “world” pada string “Hello world!” dengan teks “Dolly”.

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

?>

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-sensitive) 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!")\n echo GREETING.
?>
```

Operator PHP

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

Operator-operator aritmatika

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

Oper ator	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)

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

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
'!=='	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

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

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
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

Operator-operator string

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

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
--	--	--	---

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. Syntaks dari kondisi ini adalah sebagai berikut:

if (kondisi) {

kode yang akan dieksekusi jika kondisi true;
}

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.
}

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.

```
}
```

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. Sintax dari pernyataan ini adalah sebagai berikut:

```
switch (n) {
```

```
case label1:
```

kode yang akan dieksekusi jika $n = \text{label1}$; break;

```
case label2:
```

kode yang akan dieksekusi jika $n = \text{label2}$; break;

```
case label3:
```

kode yang akan dieksekusi jika $n = \text{label3}$;

```
break;
```

```
...
```

```
default:
```

kode yang akan dieksekusi jika n berbeda dari semua label yang ada.

```
}
```

Perulangan

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

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.  
}
```

Perulangan dengan do ... while

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

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

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.

Perulangan dengan foreach

Perulangan foreach hanya bekerja pada array, dan digunakan untuk perulangan melalui setiap nilai pada array. Syntax dari perulangan foreach adalah sebagai berikut:

```
foreach ($array as $variabel) { kode yang akan dieksekusi;
}
```

Setiap nilai dalam array akan dimasukkan kedalam variabel dalam setiap perulangan.

Fungsi

Kekuatan dari PHP berasal dari fungsi-fungsinya. PHP memiliki lebih dari 1000 fungsi built-in. Namun dalam PHP bisa juga dibuat fungsi sendiri. Fungsi adalah blok kode yang bisa digunakan secara berulang-ulang kali dalam program. Fungsi tidak akan dieksekusi secara langsung jika halaman web sedang dimuat. Sebuah fungsi akan dieksekusi oleh fungsi yang memanggilnya. Syntax dari fungsi yang dibuat sendiri adalah sebagai berikut:

```
function nama_fungsi() {
kode yang akan dieksekusi.
}
```

Nama fungsi bisa dimulai dengan huruf atau garis bawah (bukan bilangan). Tentukan nama fungsi berdasarkan apa kegunaannya. Informasi dari luar fungsi bisa dimasukkan kedalam fungsi melalui argument. Argumen adalah seperti variabel. Argumen dispesifikasikan setelah nama fungsi didalam kurung. Argument bisa dibuat banyak, antara deklarasi argument harus dipisah dengan tanda koma. Jika argumen memiliki nilai default, maka argument tersebut bisa langsung ditentukan nilainya dengan tanda sama dengan. Berikut adalah contoh penetapan argument dengan nilai default.

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:

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:

Array Terindex

Array terindex adalah array dengan index numerik. Ada dua cara untuk membuat array terindex, pertama, array bisa ditentukan secara otomatis dimana indexnya selalu mulai dari 0, contoh : `$mobil = array("volvo","BMW","Toyota");` dan yang kedua, array bisa ditentukan secara manual, contohnya sebagai berikut:

```
$mobil[0] = "Volvo";  
$mobil[1] = "BMW";  
$mobil[2] = "Toyota";
```

Panjang array atau jumlah element bisa juga dihitung menggunakan fungsi count(), untuk menghitung jumlah element pada variabel array \$mobil[] dapat ditulis count(\$mobil), fungsi ini akan mengembalikan nilai integer sebagai jumlah element dalam variabel \$mobil[].

Array Assosiatif

Array asosiatif adalah array dengan nama kunci yang ditentukan. Ada dua cara membuat array asosiatif, pertama dengan mendeklarasikan ketika memanggil fungsi array(), contoh : `$umur = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");` yang kedua dengan mendeklarasikan nilainya satu-persatu, contoh:

```
$umur['Peter'] = "35";  
$umur['Ben'] = "37";  
$umur['Joe'] = "43";
```

Untuk melakukan perulangan pada array asosiatif, dapat menggunakan `foreach($variabel as $kunci => $nilai)` dimana `$variabel` adalah variabel array, `$kunci` adalah variabel yang akan berisi kunci array dan `$nilai` adalah variabel yang berisi nilai dari array.

Element dalam sebuah array bisa diurutkan dalam susunan alpabet atau numeris, descending atau ascending. Berikut adalah fungsi-fungsi yang bisa digunakan untuk mengurutkan element-element dalam array:

`sort()` - Mengurutkan array dalam susunan ascending. Misalkan fungsi ini digunakan untuk mengurutkan array `$mobil`, maka isi array `$mobil` akan terurut berdasarkan alphabet, yaitu sebagai berikut:

Variabel Global dan Superglobal

Superglobal diperkenalkan pada PHP 4.1.0 dan merupakan variabel built-in yang selalu ada pada semua scope atau lingkup. Beberapa variabel yang didefinisi sebelumnya dalam PHP adalah “superglobal”, yang berarti bahwa mereka selalu bisa diakses tanpa terpengaruh oleh lingkup atau scope sehingga bisa diakses oleh semua fungsi atau class tanpa harus melakukan suatu teknik khusus.

Variabel PHP superglobal adalah sebagai berikut:

PHP \$GLOBALS

`$GLOBALS` adalah sebuah variabel PHP yang super global yang digunakan untuk mengakses variabel dari mana saja didalam script PHP, termasuk dari dalam fungsi atau method. PHP menyimpan semua variabel-variabel global dalam sebuah array `$GLOBALS[index]`. Index adalah nama dari variabel. Contoh berikut menunjukkan cara menggunakan variabel `$GLOBALS`:

\$_SERVER

`$_SERVER` adalah variabel super global yang menyimpan informasi tentang header-header, path-path (jalur), dan skrip lokasi. Berikut adalah contoh penggunaan variabel global `$_SERVER`:

TUGAS

Buatlah program web untuk mengolah nama-nama anggota keluarga anda dengan kriteria sebagai berikut:

1. Nama-nama disimpan didalam variabel array.
2. Program menampilkan jumlah kata dan 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.

BAB II

PEMBAHASAN

```
<!DOCTYPE html>
<html>
<head>
    <title>Anggota Keluarga</title>
</head>
<body>
```

Gambar 2.1

Pada Gambar di atas adalah fungsi untuk menampilkan judul pada sebuah website.

```
<?php foreach ($anggota_keluarga = ["Santo Suling", "Santara", "Stella Suling", "Steven Suling"] as $ak){ ?>
<br>Nama: <?=$ak; ?>
<br>Jumlah kata: <?=$ str_word_count($ak) ?>
<br>Jumlah huruf: <?=$ strlen($ak) ?>
<br>Kebalikan nama: <?=$ strrev($ak) ?> <br>
```

Gambar 2.2

Pada Gambar diatas adalah Foreach di gunakan untuk perulangan yang datanya dalam bentuk array. terkadang kita memiliki data dalam bentuk array yang harus kita tampilkan dengan menggunakan perulangan for. Pada Source Code diatas untuk mengisi daftar nama yang ingin di tampilkan. Pada
 berfungsi untuk merapikan kata supaya tidak menyambung ke samping. Nama: <?=\$ak; ?> fungsinya untuk menampilkan nama. Jumlah kata: <?=\$ str_word_count(\$ak) ?> untuk menghitung kata pada anggota keluarga tersebut. Jumlah huruf: <?=\$ strlen(\$ak) ?> untuk menampilkan jumlah huruf yang terdiri dari anggota keluarga tersebut/. Kebalikan nama: <?=\$ strrev(\$ak) ?>
 membalikkan nama dari kanan ke kiri.


```
<?php
$teks_awal = trim($ak);
$teks_modif = strtolower($teks_awal);
$teks=str_replace(' ','',$teks_modif);
$jumlah = strlen($teks);
```

Gambar 2.3

Pada Gambar di atas adalah `$teks_awal = trim($_POST['kalimat']);` berfungsi untuk membuang spasi di awal dan akhir kalimat. `$teks_modif = strtolower($teks_awal);` adalah mengganti kalimat menjadi huruf kecil semua. `$teks=str_replace(" ","",$teks_modif);` adalah membuang spasi yang ada di tengah kata. `$jumlah = strlen($teks);` adalah menghitung jumlah karakter pada variable `$teks`.

```
$a = substr_count($teks, "a"); //hitung jumlah huruf "a"
$i = substr_count($teks, "i"); //hitung jumlah huruf "i"
$u = substr_count($teks, "u"); //hitung jumlah huruf "u"
$e = substr_count($teks, "e"); //hitung jumlah huruf "e"
$o = substr_count($teks, "o"); //hitung jumlah huruf "o"
```

Gambar 2.4

Pada gambar di atas adalah source code untuk menghitung jumlah kata/huruf. `substr_count()` fungsi menghitung jumlah kali substring terjadi dalam sebuah string. Jadi source code diatas berguna untuk menghitung jumlah kata/huruf pada anggota keluarga tersebut.

```
$vocal = $a+$i+$u+$e+$o; //hitung total jumlah huruf vocal
$konsonan = $jumlah - $vocal;
```

Gambar 2.5

Pada Gambar diatas merupakan source code untuk menghitung total jumlah huruf vocal dan juga menghitung huruf konsonan. Menggunakan rumus Jumlah kata – vocal maka akan menghasilkan huruf konsonan

```
echo "Kalimat \"$. $teks_awal.\" memiliki :\" .\"<br>";  
echo "Jumlah karakter = $jumlah \" .\"<br>";  
echo "jumlah huruf vocal = $vocal \" .\"<br>";  
echo "Jumlah huruf kosonan = $konsonan\" .\"<br>";
```

Gambar 2.6

Pada Gambar diatas merupakan source code untuk ouput untuk menampilkan sebuah huruf pada saat menginput didalam source code tersebut. echo adalah fungsi untuk menampilkan teks ke layar. Dan juga menampilkan output setiap variable apa aja yang ingin di keluarkan.

Nama: Santo Suling
Jumlah kata: 2
Jumlah huruf: 12
Kebalikan nama: gniluS otnaS
Kalimat "Santo Suling" memiliki :
Jumlah karakter = 11
jumlah huruf vocal = 4
Jumlah huruf kosonan = 7

Nama: Santara
Jumlah kata: 1
Jumlah huruf: 7
Kebalikan nama: aratnaS
Kalimat "Santara" memiliki :
Jumlah karakter = 7
jumlah huruf vocal = 3
Jumlah huruf kosonan = 4

Nama: Stella Suling
Jumlah kata: 2
Jumlah huruf: 13
Kebalikan nama: gniluS alletS
Kalimat "Stella Suling" memiliki :
Jumlah karakter = 12
jumlah huruf vocal = 4
Jumlah huruf kosonan = 8

Nama: Steven Suling
Jumlah kata: 2
Jumlah huruf: 13
Kebalikan nama: gniluS nevetS
Kalimat "Steven Suling" memiliki :
Jumlah karakter = 12
jumlah huruf vocal = 4
Jumlah huruf kosonan = 8

Gambar 2.7

Pada Gambar diatas adalah hasil dari source code tersebut didalam web tersebut memiliki nama,jumlah kata,jumlah huruf,kebalikan nama,kalimat.jumlah karakter,jumlah huruf vocal dan juga huruf kosonan

BAB III

KESIMPULAN

Index array di PHP dihitung mulai dari 0. Sebuah array bisa menyimpan lebih dari satu nilai. Sebuah array asosiatif adalah sebuah array yang kuncinya bukan merupakan angka. Sebuah array multidimensi adalah array di dalam array. Untuk me-loop array biasa anda bisa gunakan for atau foreach, untuk array asosiatif gunakan foreach.

DAFTAR PUSTAKA

Dosen Teknik Informatika.2021.Modul PRAKTIKUM PEMROGRAMAN WEB
I.Palangkaraya:

Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya
(Diakses pada tanggal 28 Maret 2021).

Ahmad Muhardian. 19 Nov 2015. Belajar PHP: 7 Hal yang Harus Kamu Ketahui
Tentang Array di PHP. <https://www.petanikode.com/php-array/>(Diakses
pada tanggal 28 Maret 2021).

No name. Cara Menghitung Jumlah Kata Tertentu Dalam Kalimat Menggunakan
Fungsi substr_count() Pada Bahasa Pemrograman PHP. <https://www.kodefungsi.com/fungsi/php/8ce39fbeb9116a5604105d7ff1a94a9/Cara-Menghitung-Jumlah-Kata-Tertentu-Dalam-Kalimat-Menggunakan-Fungsi-substr-count---Pada-Bahasa-Pemrograman-PHP>(Diakses pada tanggal 28
Maret 2021).

Nurul Huda. 2 February 2020. PHP Dasar: Macam Macam Operator.
<https://jagongoding.com/web/php/dasar/macam-macam-operator/>(Diakses
pada tanggal 28 Maret 2021)

LAMPIRAN

```
<!DOCTYPE html>
<html>
<head>
    <title>Anggota Keluarga</title>
</head>
<body>
```

Gambar 2.1

```
<?php foreach ($anggota_keluarga = ["Santo Suling", "Santara", "Stella Suling", "Steven Suling"] as $ak){ ?>
<br>Nama: <?= $ak; ?>
<br>Jumlah kata: <?= str_word_count($ak) ?>
<br>Jumlah huruf: <?= strlen($ak) ?>
<br>Kebalikan nama: <?= strrev($ak) ?> <br>
```

Gambar 2.2

```
<?php
$teks_awal = trim($ak);
$teks_modif = strtolower($teks_awal);
$teks=str_replace(' ', '', $teks_modif);
$jumlah = strlen($teks);
```

Gambar 2.3

```
$a = substr_count($teks, "a"); //hitung jumlah huruf "a"
$i = substr_count($teks, "i"); //hitung jumlah huruf "i"
$u = substr_count($teks, "u"); //hitung jumlah huruf "u"
$e = substr_count($teks, "e"); //hitung jumlah huruf "e"
$o = substr_count($teks, "o"); //hitung jumlah huruf "o"
```

Gambar 2.4

```
$vocal = $a+$i+$u+$e+$o; //hitung total jumlah huruf vocal
$konsonan = $jumlah - $vocal;
```

Gambar 2.5

```

echo "Kalimat \"$.teks_awal.\"\" memiliki :\" .\"<br>";

echo "Jumlah karakter = $jumlah \" .\"<br>";

echo "jumlah huruf vocal = $vocal \" .\"<br>";

echo "Jumlah huruf kosonan = $konsonan\" .\"<br>";

```

Gambar 2.6

Nama: Santo Suling
 Jumlah kata: 2
 Jumlah huruf: 12
 Kebalikan nama: gniluS otnaS
 Kalimat "Santo Suling" memiliki :
 Jumlah karakter = 11
 jumlah huruf vocal = 4
 Jumlah huruf kosonan = 7

Nama: Santara
 Jumlah kata: 1
 Jumlah huruf: 7
 Kebalikan nama: aratnaS
 Kalimat "Santara" memiliki :
 Jumlah karakter = 7
 jumlah huruf vocal = 3
 Jumlah huruf kosonan = 4

Nama: Stella Suling
 Jumlah kata: 2
 Jumlah huruf: 13
 Kebalikan nama: gniluS alletS
 Kalimat "Stella Suling" memiliki :
 Jumlah karakter = 12
 jumlah huruf vocal = 4
 Jumlah huruf kosonan = 8

Nama: Steven Suling
 Jumlah kata: 2
 Jumlah huruf: 13
 Kebalikan nama: gniluS nevetS
 Kalimat "Steven Suling" memiliki :
 Jumlah karakter = 12
 jumlah huruf vocal = 4
 Jumlah huruf kosonan = 8

Gambar 2.7