



### Topik

1. Konsep *array* dalam pemrograman PHP
2. Konsep *fungsi* dalam pemrograman PHP

### Obyektif

Siswa diharapkan untuk:

1. Memahami konsep *array* dalam pemrograman PHP
2. Memahami konsep fungsi dalam pemrograman PHP

## INTRODUCTION

### Pengantar *Array* dan Fungsi

*Array* dan fungsi adalah konsep dasar dalam pemrograman yang membantu mengatur dan mengoptimalkan kode. **● *Array***

*Array*, atau list, adalah salah satu tipe data. *Array* bukanlah tipe data dasar seperti bilangan bulat atau boolean, melainkan tipe data yang terdiri dari kumpulan tipe data lainnya. *Array* memudahkan untuk mengelompokkan data, menghemat penulisan, dan membuat penggunaan variabel lebih efisien. *Array* adalah struktur data yang menyimpan kumpulan elemen, *biasanya* dari tipe data yang sama, di bawah satu variabel. *Array* menyederhanakan penanganan kumpulan data yang besar, memungkinkan beberapa nilai disimpan dan diakses dengan mudah menggunakan indeks atau kunci. Dalam PHP, *array* diklasifikasikan menjadi tiga jenis:

- *Indexed Arrays*: *Array* dengan indeks numerik.
- *Associative Arrays*: *Array* dengan kunci (*key*) yang menetapkan nilai (*value*) pada tiap elemennya.
- *Multidimensional Arrays*: *Array* yang berisi *array* lain sebagai elemennya.

### **● Fungsi**

Fungsi adalah blok kode yang dirancang untuk melakukan tugas tertentu, yang dapat digunakan kembali beberapa kali di seluruh program. Fungsi mengurangi redundansi, membuat kode lebih mudah dibaca, dan meningkatkan efisiensi. Dengan memecah tugas-tugas berulang seperti kueri database atau perhitungan matematis menjadi fungsi, program menjadi modular, dan pemeliharaan menjadi lebih mudah.

### Praktikum 1. *Indexed Arrays*

*Indexed Arrays* PHP adalah *array* di mana elemen disimpan dengan indeks numerik, mulai dari 0 secara default. Setiap elemen dalam *array* dikaitkan dengan nomor indeks, yang digunakan untuk mengakses atau mereferensikan elemen tersebut.

```
<?php
// Creating an indexed array
$variable = array("Value0", "Value1", "Value2", "Value3");

// Accessing the elements of the indexed array
echo $variable[0]; // Outputs: Value0
echo $variable[1]; // Outputs: Value1

// Adding a new element
$variable[] = "Value4"; // Adds Value4 to the array
?>
```

Bisa juga ditulis sebagai berikut:

```
<?php
// Creating an indexed array
$variable[0] = "Value0";
$variable[1] = "Value1";
$variable[2] = "Value2";
$variable[3] = "Value3";

// Accessing the elements of the indexed array
echo $variable[0]; // Outputs: Value0
echo $variable[1]; // Outputs: Value1

// Adding a new element
$variable[] = "Value4"; // Adds Value4 to the array
?>
```

Dalam bentuk ini, *array* dibuat secara langsung menetapkan nilai ke indeks tertentu tanpa menggunakan **fungsi `array()`**. PHP secara otomatis menetapkan indeks berikutnya yang tersedia jika elemen baru ditambahkan tanpa menentukan indeks, seperti yang ditunjukkan saat menambahkan "Value4".

Ikuti langkah-langkah ini untuk memahami *array* yang diindeks di PHP:

Langkah	Deskripsi
1	<p>Buat file baru bernama <b>array_1.php</b> di dalam direktori <b>JS05_PHP-2</b>, lalu ketik kode berikut:</p> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;/head&gt; &lt;body&gt; &lt;h2&gt;Array Terindeks&lt;/h2&gt; &lt;?php     \$Listdosen=["Elok Nur Hamdana","Unggul Pamenang", "Bagas Nugraha"];      echo \$Listdosen[2] . "&lt;br&gt;";     echo \$Listdosen[0] . "&lt;br&gt;";     echo \$Listdosen[1] . "&lt;br&gt;"; ?&gt; &lt;/body&gt; &lt;/html&gt;</pre>
2	<p>Simpan file dan jalankan program Kita di browser. Ketik tautan ini ke browser Kita <b><a href="localhost/dasarWeb/JS05_PHP-2/array_1.php">localhost/dasarWeb/JS05_PHP-2/array_1.php</a></b></p>
3	<p>Amati <i>output</i> yang ditampilkan</p>

Untuk menampilkan *array*, selain menggunakan indeks, kita juga bisa menggunakan *loop*. Coba tampilkan *output* program di atas menggunakan *loop*.

(Pertanyaan No.1)

**Jawaban:**

localhost/dasarWeb/Praktikum05%20Pemrograman%20Web/Array1.php

#### Array Terindeks

Elok Nur Hamdana  
Unggul Pamenang  
Bagas Nugraha

4

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 <h2>Array Terindeks</h2>
7 <?php
8     $Listdosen=["Elok Nur Hamdana", "Unggul Pamenang", "Bagas Nugraha"];
9
10    for ($i = 0; $i < count($Listdosen); $i++) {
11        echo $Listdosen[$i] . "<br>";
12    }
13 ?>
14 </body>
15 </html>
```

## Praktikum 2. Associative Array

*Array* asosiatif PHP adalah *array* di mana kuncinya bukan numerik melainkan **string**, memungkinkan kita untuk mengaitkan nilai tertentu dengan kunci yang bermakna (sesuai keinginan kita). Hal ini dapat membuat lebih mudah untuk mengakses dan memanipulasi data berdasarkan nama kunci daripada indeks numerik.

Komponen *array* asosiatif terdiri dari pasangan kunci-nilai (**key-value**). Kunci menunjukkan posisi di mana nilai disimpan. PHP menggunakan simbol panah (**=>**) untuk menetapkan nilai ke kunci. Berikut adalah sintaks untuk menulis *array* asosiatif:

```
<?php
// Creating an indexed array
$variable = array(
    'Key0' => 'value0',
    'Key1' => 'value1',
    'Key2' => 'value2',
    'Key3' => 'value3'
);

// Accessing the elements of the indexed array
echo $variable['Key0']; // Outputs: Value0
echo $variable['Key1']; // Outputs: Value1

// Adding a new element
$variable['Key4'] = "Value4"; // Adds Value4 to the array
?>
```

Bisa juga ditulis sebagai berikut:

```

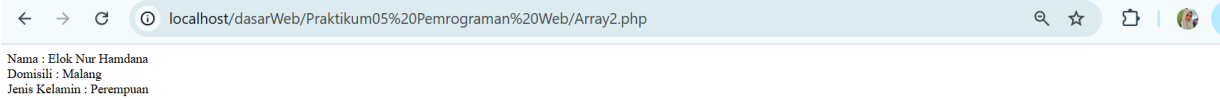
<?php
// Creating an indexed array
$variable['Key0'] = 'value0';
$variable['Key1'] = 'value1';
$variable['Key2'] = 'value2';
$variable['Key3'] = 'value3';

// Accessing the elements of the indexed array
echo $variable['Key0']; // Outputs: Value0
echo $variable['Key1']; // Outputs: Value1

// Adding a new element
$variable['Key4'] = "Value4"; // Adds Value4 to the array
?>

```

Ikuti langkah-langkah ini untuk memahami *array* asosiatif di PHP:

Langkah	Deskripsi
1	<p>Buat file baru bernama <b>array_2.php</b> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt;   &lt;meta charset="utf-8"&gt;   &lt;meta name="viewport" content="width=device-width, initial-scale=1"&gt;   &lt;title&gt;&lt;/title&gt; &lt;/head&gt; &lt;body&gt;   &lt;?php     \$Dosen = [       'nama' =&gt; 'Elok Nur Hamdana',       'domisili' =&gt; 'Malang',       'jenis_kelamin' =&gt; 'Perempuan' ];      echo "Nama : {\$Dosen ['nama']} &lt;br&gt;";     echo "Domisili : {\$Dosen ['domisili']} &lt;br&gt;";     echo "Jenis Kelamin : {\$Dosen ['jenis_kelamin']} &lt;br&gt;";    ?&gt; &lt;/body&gt; &lt;/html&gt; </pre>
2	<p>Simpan file dan jalankan program Kita di browser. Ketik tautan ini ke browser Kita <b><a href="localhost/dasarWeb/JS05_PHP-2/array_2.php">localhost/dasarWeb/JS05_PHP-2/array_2.php</a></b></p>
3	<p>Amati hasil yang ditampilkan. Kemudian tambahkan style tabel pada output tampilan tersebut supaya lebih menarik.</p> <p><b>(Pertanyaan No.2)</b></p> <p>*Untuk penggunaan style bebas boleh internal atau eksternal file.</p> <p><b>Jawaban:</b></p> <p><b>Tanpa style</b></p> 

## Dengan style menggunakan eksternal file

localhost/dasarWeb/Praktikum05%20Pemrograman%20Web/Array2.php

### Informasi Dosen

Nama	Elok Nur Hamdana
Domisili	Malang
Jenis Kelamin	Perempuan

### Praktikum 3. Multidimensional Array

Array multidimensi di PHP adalah *array* yang berisi satu atau lebih *array* sebagai elemennya. Ini berarti bahwa setiap elemen dalam *array* dapat berupa *array* itu sendiri, memungkinkan Kita menyimpan data dalam struktur seperti kisi atau matriks. *Array* multidimensi sering digunakan untuk mewakili struktur data yang kompleks seperti tabel, matriks, atau database.

Contoh *Array* Multidimensi di PHP:

```
<?php
// Creating an multidimensional array
$variable = array(
    array('value00', 'value01', 'value02'),
    array('value10', 'value11', 'value12'),
    array('value20', 'value21', 'value22')
);

// Accessing the elements
echo $variable[0][0]; // Outputs: value00
echo $variable[1][1]; // Outputs: value11
echo $variable[2][2]; // Outputs: value22
?>
```

Atau bisa juga ditulis sebagai berikut:

```
<?php
// Creating a multidimensional array
$students = array(
    array("name" => "John", "age" => 20, "grade" => "A"),
    array("name" => "Sarah", "age" => 19, "grade" => "B"),
    array("name" => "Mike", "age" => 21, "grade" => "A")
);

// Accessing elements of the multidimensional array
echo $students[0]["name"]; // Outputs: John
echo $students[1]["name"]; // Outputs: Sarah
echo $students[2]["name"]; // Outputs: Mike
echo $students[0]["age"]; // Outputs: 20
echo $students[1]["age"]; // Outputs: 19
echo $students[2]["age"]; // Outputs: 21
?>
```

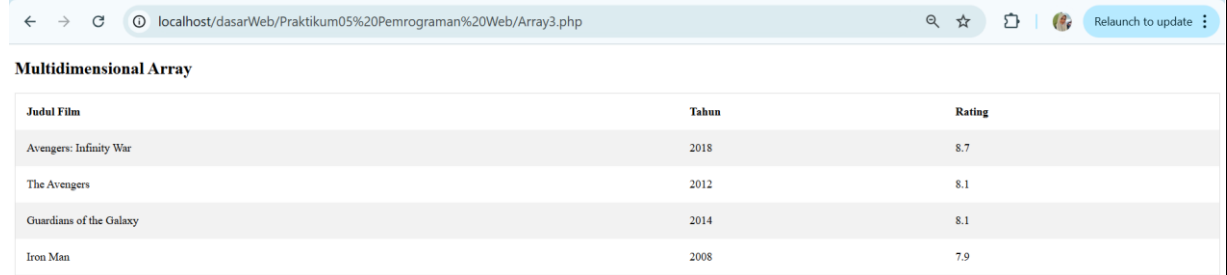


Ikuti langkah-langkah ini untuk memahami *array* multidimensi di PHP:

Langka h	Deskripsi
1	<p>Buat file baru bernama <b>style.css</b> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p> <pre> 1  table { 2      border-collapse: collapse; 3      border-spacing: 0; 4      width: 100%; 5      border: 1px solid #ddd; 6  } 7 8  th, td { 9      text-align: left; 10     padding: 16px; 11 } 12 13 tr:nth-child(even) { 14     background-color: #f2f2f2 15 }</pre>
2	<p>Buat file baru bernama <b>array_3.php</b> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p>
	<pre> 1  &lt;!DOCTYPE HTML&gt; 2  &lt;html&gt; 3      &lt;head&gt; 4          &lt;link rel="stylesheet" type="text/css" href="style.css"/&gt; 5      &lt;/head&gt; 6      &lt;body&gt; 7          &lt;h2&gt; Multidimensional Array &lt;/h2&gt; 8          &lt;table&gt; 9              &lt;tr&gt; 10                 &lt;th&gt;Judul Film&lt;/th&gt; 11                 &lt;th&gt;Tahun&lt;/th&gt; 12                 &lt;th&gt;Rating&lt;/th&gt; 13             &lt;/tr&gt; 14             &lt;?php 15                 \$movie = array( 16                     array("Avengers: Invinity War", 2018, 8.7), 17                     array("The Avengers", 2012, 8.1), 18                     array("Guardians of the Galaxy", 2014, 8.1), 19                     array("Iron Man", 2008, 7.9) 20                 ); 21                 echo "&lt;tr&gt;"; 22                 echo "&lt;td&gt;". \$movie[0][0] . "&lt;/td&gt;"; 23                 echo "&lt;td&gt;". \$movie[0][1] . "&lt;/td&gt;"; 24                 echo "&lt;td&gt;". \$movie[0][2] . "&lt;/td&gt;"; 25                 echo "&lt;/tr&gt;"; 26                 echo "&lt;tr&gt;"; 27                 echo "&lt;td&gt;". \$movie[1][0] . "&lt;/td&gt;"; 28                 echo "&lt;td&gt;". \$movie[1][1] . "&lt;/td&gt;"; 29                 echo "&lt;td&gt;". \$movie[1][2] . "&lt;/td&gt;"; 30                 echo "&lt;/tr&gt;"; 31                 echo "&lt;tr&gt;"; 32                 echo "&lt;td&gt;". \$movie[2][0] . "&lt;/td&gt;"; 33                 echo "&lt;td&gt;". \$movie[2][1] . "&lt;/td&gt;"; 34                 echo "&lt;td&gt;". \$movie[2][2] . "&lt;/td&gt;"; 35                 echo "&lt;/tr&gt;"; 36                 echo "&lt;tr&gt;"; 37                 echo "&lt;td&gt;". \$movie[3][0] . "&lt;/td&gt;"; 38                 echo "&lt;td&gt;". \$movie[3][1] . "&lt;/td&gt;"; 39                 echo "&lt;td&gt;". \$movie[3][2] . "&lt;/td&gt;"; 40                 echo "&lt;/tr&gt;"; 41             &lt;?&gt; 42             &lt;/table&gt; 43         &lt;/body&gt; 44     &lt;/html&gt;</pre>
3	<p>Simpan file dan jalankan program Kita di browser. Ketik tautan ini ke browser Kita <b><a href="localhost/dasarWeb/JS05_PHP-2/array_3.php">localhost/dasarWeb/JS05_PHP-2/array_3.php</a></b></p>

Amati *output* yang ditampilkan dan jelaskan hasil pengamatanmu (Pertanyaan No 3)

**jawaban :**



The screenshot shows a web browser window with the address bar displaying 'localhost/dasarWeb/Praktikum05%20Pemrograman%20Web/Array3.php'. The page title is 'Multidimensional Array'. Below the title is a table with three columns: 'Judul Film', 'Tahun', and 'Rating'. The table contains four rows of data:

Judul Film	Tahun	Rating
Avengers: Infinity War	2018	8.7
The Avengers	2012	8.1
Guardians of the Galaxy	2014	8.1
Iron Man	2008	7.9

#### Jawaban

Berdasarkan kode program dan *output* yang ditampilkan, dapat diamati bahwa kode tersebut berhasil membuat dan menampilkan array multidimensi di PHP dalam format tabel HTML. Array bernama \$movie menyimpan data film, di mana setiap elemen array utamanya adalah array lain yang berisi Judul Film (indeks [0]), Tahun (indeks [1]), dan Rating (indeks [2]). *Output* menunjukkan keempat film—"Avengers: Infinity War," "The Avengers," "Guardians of the Galaxy," dan "Iron Man"—telah berhasil diekstrak nilainya satu per satu menggunakan akses indeks \$movie[i][j]

## Fungsi

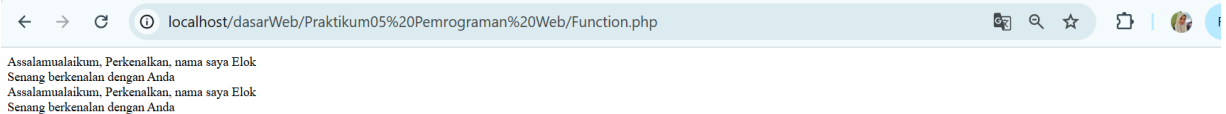
Ada banyak fungsi PHP bawaan yang sering kita gunakan, seperti `print()`, `print_r()`, `unset()`, dll. Selain fungsi-fungsi tersebut, kita juga dapat membuat fungsi kustom kita sendiri sesuai dengan kebutuhan kita. Fungsi adalah sekumpulan instruksi yang dibungkus dalam blok. Fungsi dapat digunakan kembali tanpa harus menulis ulang instruksi di dalamnya. Di PHP, fungsi dapat dibuat menggunakan kata kunci `function`, diikuti dengan nama fungsi. Contoh:

```
function namaFungsi(){  
    //...  
}
```

Kode instruksi dapat ditulis di dalam kurung kurawal (`{...}`). Nama fungsi di PHP harus dimulai dengan huruf atau garis bawah dan tidak boleh dimulai dengan angka. Penamaan fungsi di PHP tidak peka huruf besar/kecil (*case-insensitive*). Ikuti langkah-langkah ini untuk memahami penggunaan fungsi di PHP:

```
<?php  
// Defining a function in PHP  
function greet() {  
    echo "Hello, welcome to PHP functions!";  
}  
  
// Calling the function  
greet();  
?>
```

## Praktikum 4. Fungsi

Langka h	Deskripsi
1	<p>Buat file baru di dalam direktori JS05_PHP-2 dan beri nama <code>function.php</code></p> <pre>&lt;?php  function perkenalan(){     echo "Assalamualaikum, ";     echo "Perkenalkan, nama saya Elok&lt;br/&gt;"; //Tulis sesuai nama kalian     echo "Senang berkenalan dengan Anda&lt;br/&gt;"; }  //memanggil fungsi yang sudah dibuat perkenalan();  ?&gt;</pre>
2	<p>Simpan file dan jalankan kodenya. Ubah program sehingga dapat menampilkan <i>output</i> dua kali. Jelaskan pengamatan Kita!</p> <p><b>(Pertanyaan No 4)</b></p> <p><b>Jawaban:</b></p>  <p>Kode PHP tersebut mendemonstrasikan prinsip fungsi reusable dalam pemrograman, di mana sebuah blok kode bernama <code>perkenalan()</code> didefinisikan untuk mencetak tiga baris teks perkenalan ke <i>output</i>. Karena fungsi tersebut dipanggil dua kali secara berturut-turut di luar definisi fungsi, seluruh isinya dieksekusi sebanyak dua kali, menghasilkan pengulangan pesan perkenalan sebanyak dua kali secara berurutan pada <i>output browser</i>. Hal ini menunjukkan efisiensi fungsi dalam menjalankan tugas yang sama berulang kali tanpa perlu menulis ulang baris kode yang identik.</p>
<b>Fungsi dengan Parameter</b>	
3	<p>Untuk membuat instruksi di dalam fungsi lebih dinamis, kita dapat menggunakan parameter untuk meneruskan nilai ke dalam fungsi. Nilai-nilai ini akan diproses dalam fungsi. Misalnya, pada fungsi sebelumnya, tidak ideal jika nama yang dicetak selalu "Elok" dan sapaannya selalu "Assalamualaikum".</p> <p>Kita dapat mengubah nama orang dan sapaan menjadi kata lain.</p>
4	<p>Tambahkan parameter seperti pada kode program berikut:</p>



```
<?php
//membuat fungsi
function perkenalan($nama, $salam){
    echo $salam.", ";
    echo "Perkenalkan, nama saya ".$nama."<br/>";
    echo "Senang berkenalan dengan Anda<br/>";
}

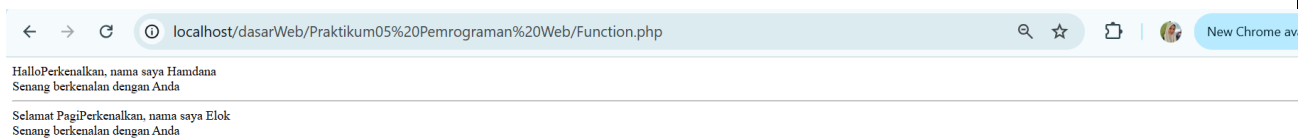
//memanggil fungsi yang sudah dibuat
perkenalan("Hamdana","Hallo");

echo "<hr>";

$saya = "Elok";
$ucapanSalam = "Selamat pagi";

//memanggil lagi
perkenalan($saya,$ucapanSalam);
?>
```

Amati *output* yang ditampilkan dan jelaskan hasil pengamatanmu! (Pertanyaan No 5)



- 5 Kode PHP ini mendemonstrasikan fungsi yang menerima parameter dan fleksibilitas pemanggilan fungsi tersebut. Fungsi `perkenalan()` didefinisikan untuk mengambil dua argumen, `$nama` dan `$salam`, yang kemudian digunakan untuk menyusun pesan perkenalan yang dinamis. Fungsi ini dipanggil dua kali: pertama, dengan meneruskan nilai *string* literal "Hamdana" dan "Hallo", dan kedua, dengan meneruskan variabel `$saya` dan `$ucapanSalam` yang telah ditentukan sebelumnya. Hasilnya, *output* menampilkan dua pesan perkenalan yang berbeda, dipisahkan oleh garis horizontal (`<hr>`), menegaskan bahwa fungsi berhasil menggunakan nilai yang berbeda untuk setiap pemanggilan.

### Fungsi dengan Parameter dan menggunakan nilai default

- 6 Kita dapat menetapkan *nilai default* ke parameter. *Nilai default* berfungsi untuk memberikan nilai untuk parameter jika tidak disediakan.  
Misalnya: jika kita lupa memberikan parameter salam, program biasanya akan melemparkan kesalahan. Oleh karena itu, kita perlu menetapkan nilai default untuk menghindari kesalahan ini.

Tulis kode ini ke program Kita

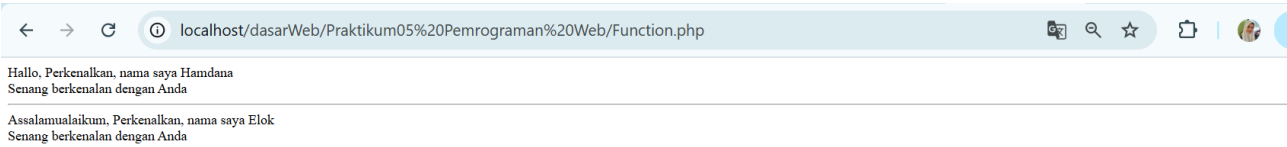

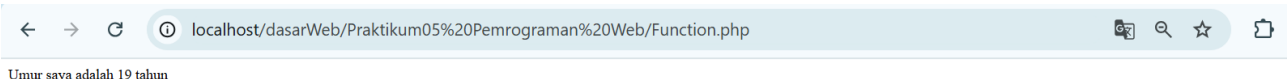
```
<?php
//membuat fungsi
function perkenalan($nama, $salam="Assalamualaikum"){
    echo $salam.", ";
    echo "Perkenalkan, nama saya ".$nama."<br/>";
    echo "Senang berkenalan dengan Anda<br/>";
}

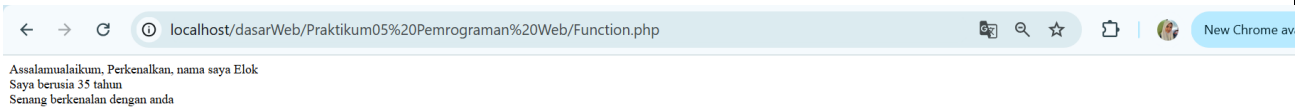
//memanggil fungsi yang sudah dibuat
perkenalan("Hamdana","Hallo");

echo "<hr>";

$saya = "Elok";
$ucapanSalam = "Selamat pagi";

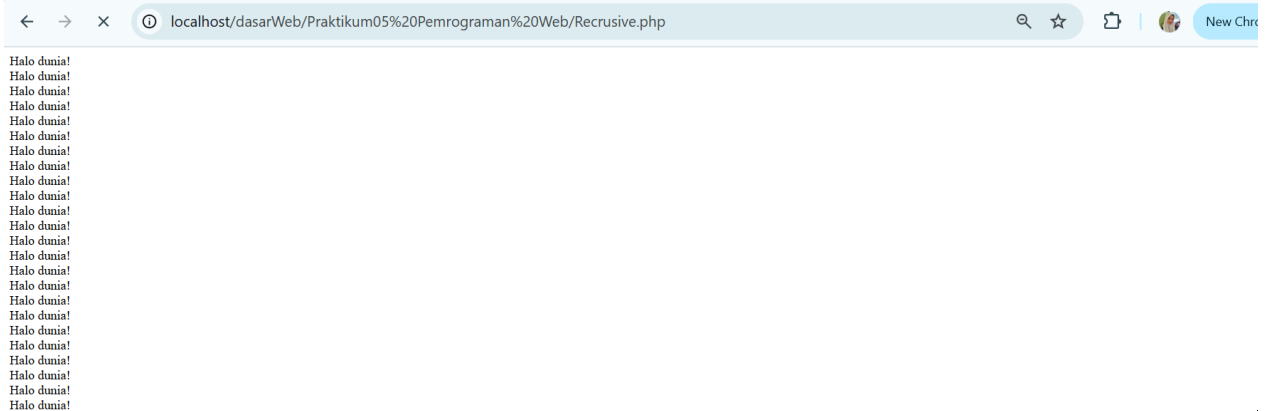
//memanggil lagi tanpa mengisi parameter salam
perkenalan($saya);
?>
```

8	<p>Amati <i>output</i> yang ditampilkan dan jelaskan hasil pengamatanmu! (Pertanyaan No 6)</p>  <p>Kode PHP ini mendemonstrasikan fitur nilai parameter <i>default</i> (default parameter value) pada fungsi. Fungsi perkenalan() menerima dua parameter, \$nama yang wajib diisi dan \$salam yang memiliki nilai <i>default</i> "Assalamualaikum". Pada pemanggilan pertama, nilai "Halo" secara eksplisit diberikan, sehingga <i>output</i> pertama menggunakan salam tersebut. Namun, pada pemanggilan kedua, parameter \$salam tidak diisi, yang kemudian secara otomatis menggunakan nilai <i>default</i> "Assalamualaikum" untuk salam, membuktikan bahwa parameter <i>default</i> berfungsi sebagai <i>fallback</i> ketika argumen tidak disediakan saat fungsi dipanggil.</p>
<p><b>Fungsi dengan nilai pengembalian</b></p>	
9	<p>Fungsi <b>dengan mengembalikan nilai</b> adalah fungsi yang dirancang untuk memproses data dan mengirim hasilnya kembali ke titik di mana ia dipanggil. Di PHP, Kita dapat</p>
<p>menggunakan pernyataan return untuk mengembalikan nilai dari suatu fungsi. Ini berguna ketika Kita memerlukan fungsi untuk melakukan perhitungan atau operasi dan memberikan hasilnya kembali ke program utama untuk digunakan lebih lanjut.</p>	
10	<p>Buat file baru di dalam direktori JS05_PHP-2 dan beri nama <code>function_return.php</code></p>  <pre>&lt;?php //membuat fungsi function hitungUmur(\$thn_lahir, \$thn_sekarang){     \$umur = \$thn_sekarang - \$thn_lahir;     return \$umur; }  echo "Umur saya adalah ". hitungUmur(1988, 2023) ."tahun" // isi sesuai dengan tahun lahir kalian  ?&gt;</pre>
11	<p>Amati <i>output</i> yang ditampilkan dan jelaskan hasil pengamatanmu! (Pertanyaan No 7)</p>  <p>Kode PHP ini mengilustrasikan fungsi yang melakukan perhitungan dan mengembalikan nilai (return value). Fungsi hitungUmur() didefinisikan untuk menerima dua parameter, \$thn_lahir dan \$thn_sekarang, kemudian menghitung dan mengembalikan selisihnya, yang merupakan usia. Ketika dipanggil dengan argumen 1988 (tahun lahir) dan 2023 (tahun sekarang), fungsi akan mengembalikan nilai 35. Namun, berdasarkan <i>output</i> yang ditampilkan ("Umur saya adalah 19 tahun"), dapat disimpulkan bahwa nilai tahun lahir yang digunakan dalam pemanggilan hitungUmur() yang</p>

	<p>menghasilkan <i>output</i> tersebut bukanlah 1988, melainkan 2004 (2023–2004=19). <i>Output</i> menunjukkan bahwa fungsi berhasil menjalankan perhitungannya dan nilai <i>return</i> tersebut dimasukkan langsung ke dalam perintah <code>echo</code> untuk ditampilkan</p>
	<p><b>Memanggil fungsi di dalam fungsi lain</b></p>
1 2	<p><b>Memanggil Fungsi Di Dalam Fungsi Lain</b> adalah praktik pemrograman umum dalam PHP dan bahasa lain. Ini memungkinkan Kita untuk memecah tugas kompleks menjadi bagianbagian yang lebih kecil dan dapat digunakan kembali dengan mengaturnya menjadi fungsi terpisah dan kemudian memanggil satu fungsi dari fungsi lainnya.</p>
1 3	<p>Ubah <code>function_return.php</code> seperti kode ini</p> <pre> &lt;?php //membuat fungsi function hitungUmur(\$thn_lahir, \$thn_sekarang){     \$umur = \$thn_sekarang - \$thn_lahir;     return \$umur; } function perkenalan (\$nama, \$salam="Assalamualaikum") {     echo \$salam.",";     echo "Perkenalkan, nama saya ".\$nama."&lt;br/&gt;";      //memanggil fungsi lain     echo "Saya berusia ". hitungUmur(1988, 2023) ." tahun&lt;br/&gt;";     echo "Senang berkenalan dengan anda&lt;br/&gt;"; }  //memanggil fungsi perkenalan perkenalan ("Elok");  ?&gt; </pre>
1 4	<p>Amati <i>output</i> yang ditampilkan dan jelaskan hasil pengamatanmu! (Pertanyaan No 8)</p>  <p><b>Jawaban:</b></p> <p>Kode PHP ini mendemonstrasikan pemanggilan fungsi di dalam fungsi lain, yang merupakan konsep modularitas. Fungsi <code>perkenalan()</code> dipanggil dengan argumen "Elok" dan menggunakan nilai <i>default</i> "Assalamualaikum". Di dalam fungsi <code>perkenalan()</code>, fungsi <code>hitungUmur(1988, 2023)</code> dipanggil. Fungsi <code>hitungUmur</code> mengembalikan hasil 35, yang kemudian digabungkan dengan <i>string</i> untuk menampilkan usia. Hasil akhirnya di <i>browser</i> adalah urutan pesan perkenalan yang menggunakan data dari parameter dan hasil perhitungan dari fungsi lain yang dipanggil di dalamnya.</p>

## Praktikum 5. Fungsi Rekursif

**Fungsi Rekursif** di PHP adalah fungsi yang memanggil dirinya sendiri selama eksekusi. Teknik ini berguna untuk memecahkan masalah yang dapat dibagi menjadi submasalah yang lebih kecil dan serupa, sering disebut sebagai membagi dan menaklukkan. Fungsi rekursif biasanya digunakan untuk memecahkan masalah seperti menghitung faktorial, bilangan Fibonacci, dan pemrograman dinamis.

Langkah	Deskripsi
1	<p>Buat file baru bernama <code>recursive.php</code> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p> <pre>&lt;?php function tampilkanHaloDunia(){     echo "Halo dunia! &lt;br&gt;";      tampilkanHaloDunia(); }  tampilkanHaloDunia(); ?&gt;</pre>
2	<p>Jika kode program di atas dieksekusi, apa yang akan terjadi dan apa dampaknya dari melakukannya? Silakan bagikan pendapat Kita!</p> <p>(Pertanyaan No 9)</p>  <p><b>Jawaban:</b></p> <p>Kode PHP ini mendemonstrasikan fungsi rekursif tanpa kondisi berhenti (<i>base case</i>), yang menyebabkan <i>output</i> "Halo dunia!" ditampilkan berulang kali tanpa batas (atau hingga <i>stack</i> memori <i>server</i> habis). Fungsi <code>tampilkanHaloDunia()</code> memanggil dirinya sendiri di akhir definisinya. Karena fungsi ini dipanggil sekali di <i>scope</i> global, ia akan terus mencetak <i>string</i> tersebut dan memanggil dirinya lagi, menghasilkan <i>loop</i> tak terbatas yang secara visual terlihat di <i>browser</i> seperti pada gambar.</p>

Untuk menampilkan angka 1 hingga 25, kita dapat dengan mudah menggunakan *loop* for sebagai berikut:

```
<?php
for ($i=1; $i <=25; $i++){
    echo "Perulangan ke-{$i} <br>";
}
?>
```

localhost/dasarWeb/Praktikum05%20Pemrograman%20Web/Recursive.php

Perulangan ke-1  
Perulangan ke-2  
Perulangan ke-3  
Perulangan ke-4  
Perulangan ke-5  
Perulangan ke-6  
Perulangan ke-7  
Perulangan ke-8  
Perulangan ke-9  
Perulangan ke-10  
Perulangan ke-11  
Perulangan ke-12  
Perulangan ke-13  
Perulangan ke-14  
Perulangan ke-15  
Perulangan ke-16  
Perulangan ke-17  
Perulangan ke-18  
Perulangan ke-19  
Perulangan ke-20  
Perulangan ke-21  
Perulangan ke-22  
Perulangan ke-23  
Perulangan ke-24  
Perulangan ke-25

Kita dapat membuat tampilan angka 1 hingga 25 menggunakan fungsi rekursif (tanpa *for loop*).

```
<?php
function tampilkanAngka (int $jumlah, int $indeks = 1) {
    echo "Perulangan ke-{$indeks} <br>";

    //panggil diri sendiri selama $indeks <= $jumlah
    if ($indeks < $jumlah) {
        tampilkanAngka($jumlah, $indeks + 1);
    }
}
tampilkanAngka(20);
?>
```

Jalankan kode program di atas dan jelaskan *output* nya, lalu jelaskan mengapa ia berperilaku seperti itu.

(Pertanyaan No 10)

localhost/dasarWeb/Praktikum05%20Pemrograman%20Web/Recursive.php

Perulangan ke-1  
Perulangan ke-2  
Perulangan ke-3  
Perulangan ke-4  
Perulangan ke-5  
Perulangan ke-6  
Perulangan ke-7  
Perulangan ke-8  
Perulangan ke-9  
Perulangan ke-10  
Perulangan ke-11  
Perulangan ke-12  
Perulangan ke-13  
Perulangan ke-14  
Perulangan ke-15  
Perulangan ke-16  
Perulangan ke-17  
Perulangan ke-18  
Perulangan ke-19  
Perulangan ke-20

**Jawaban :**

Kode PHP ini mendemonstrasikan fungsi rekursif yang terstruktur dengan baik untuk melakukan perulangan. Fungsi `tampilkanAngka()` menerima dua parameter, `$jumlah` (target total perulangan) dan `$indeks` (nilai *default* 1 untuk *counter*), dan mencetak nilai indeks saat ini. Fungsi ini

menggunakan kondisi if (\$indeks < \$jumlah) sebagai syarat berhenti (*base case*). Saat dipanggil dengan tampilanAngka(20), fungsi akan memanggil dirinya sendiri, dengan nilai \$indeks yang terus bertambah satu, sampai \$indeks mencapai 20. Hasilnya adalah *output* yang menampilkan perulangan secara berurutan dari "Perulangan ke-1" hingga "Perulangan ke-20".

### Menu Multi-Level menggunakan Array

Menu Multi-Level menggunakan *Array* di PHP mengacu pada pembuatan struktur menu hierarkis atau bersarang di mana setiap item menu dapat memiliki sub-item. Hal ini berguna untuk membuat menu navigasi di situs web yang memiliki struktur yang lebih kompleks, seperti bagian dengan sub-bagian atau kategori dengan subkategori.

#### Contoh Menu Multi-Level menggunakan Array di PHP:

Untuk membuat struktur menu multi-level atau bersarang, *array* dapat digunakan untuk mewakili item menu dan sub-itemnya. Berikut adalah contoh yang menunjukkan bagaimana Kita dapat menggunakan *array* multidimensi untuk membuat menu semacam itu.

```
<?php
// Menentukan menu multi-level menggunakan array asosiatif
$menu = array(
    "Rumah" => "#home",
    "Tentang Kami" => array(
        "Tim kami" => "#team",
        "Kisah Kami" => "#story",
        "Misi & Visi" => "#mission"
    ),
    "Layanan" => array(
        "Pengembangan Web" => "#web",
        "Pengembangan Seluler" => "#mobile",
        "Optimasi SEO" => "#seo"
    ),
    "Kontak" => "#contact"
);

// Fungsi untuk menampilkan menu
function displayMenu($menu) {
    echo "<ul>";
    foreach ($menu as $key => $value) {
        // Periksa apakah item menu adalah array (artinya memiliki sub-item)
        if (is_array($value)) {
            echo "<li>$key";
            displayMenu($value); // Tampilkan sub-menu secara rekursif
        } else {
            echo "</li>";
        }
    }
    echo "</ul>";
}

// Panggil fungsi untuk menampilkan menu displayMenu($menu);
?>
```

Kita dapat mencoba kode di atas di file `array_menu.php`, dan menjalankannya di browser.

## Praktikum 6. Menu Multi-Level

Langkah	Deskripsi
---------	-----------



1	Buat <code>\$menu</code> variabel. Variabel ini adalah kombinasi dari <i>array</i> terindeks dan <i>array</i> asosiatif multidimensi. Ini disebut multidimensi karena merupakan <i>array</i> yang berisi <i>array</i> lain
---	--

	<b>di dalamnya.</b> Selanjutnya, kita akan mencoba menampilkan semua item dari <code>array \$menu</code> menggunakan fungsi rekursif
--	--

2	<p>Tulis kode ini ke dalam file <code>multi_menu.php</code></p> <pre>&lt;?php \$menu = [     [         "nama" =&gt; "Beranda"     ],     [         "nama" =&gt; "Berita",         "subMenu" =&gt; [             [                 "nama" =&gt; "Wisata",                 "subMenu" =&gt; [                     [                         "nama" =&gt; "Pantai"                     ],                     [                         "nama" =&gt; "Gunung"                     ]                 ]             ],             [                 "nama" =&gt; "Kuliner"             ],             [                 "nama" =&gt; "Hiburan"             ]         ]     ],     [         "nama" =&gt; "Tentang"     ],     [         "nama" =&gt; "Kontak"     ] ];</pre>
---	---

3	<p>Setelah menulis kode di langkah 2, tulis kode ini setelahnya, untuk menampilkan menu di browser</p> <pre>function tampilkanMenuBertingkat (array \$menu) {     echo "&lt;ul&gt;";     foreach (\$menu as \$key =&gt; \$item) {         echo "&lt;li&gt;{\$item['nama']}&lt;/li&gt;";     }     echo "&lt;/ul&gt;"; }  tampilkanMenuBertingkat(\$menu); ?&gt;</pre>
---	---

4	<p>Jalankan program di atas dan jelaskan <i>output</i> nya. (Pertanyaan No 11)</p>  <p><b>Jawaban:</b></p>
---	--

	<p>Program PHP ini bertujuan untuk menampilkan menu bertingkat (multilevel menu) menggunakan array multidimensi tetapi hanya berhasil menampilkan tingkat menu utamanya. Array \$menu menyimpan struktur menu yang kompleks, termasuk <i>submenu</i> (seperti 'Berita' yang memiliki 'Wisata', 'Kuliner', dan 'Hiburan'), namun fungsi <code>tampilkanMenuBertingkat()</code> yang digunakan tidak dirancang secara rekursif. Fungsi tersebut hanya menggunakan perulangan <code>foreach</code> tunggal untuk mencetak elemen 'nama' dari array utama. Akibatnya, <i>output</i> yang ditampilkan di <i>browser</i> hanyalah daftar menu tingkat atas: Beranda, Berita, Tentang, dan Kontak, dan semua <i>submenu</i> diabaikan karena fungsi tidak memiliki logika untuk memproses kedalaman array lebih lanjut.</p>
5	<p>Selanjutnya, buat fungsi di atas rekursif dengan memanggil dirinya sendiri ketika item dari menu memiliki atribut <code>subMenu</code>. Ini akan menghasilkan tampilan seperti berikut.</p>
	<div data-bbox="170 591 592 992" data-label="Image"> <pre> graph TD     Beranda --&gt; Beranda     Berita --&gt; Berita     Berita --&gt; Wisata     Wisata --&gt; Pantai     Wisata --&gt; Gunung     Berita --&gt; Kuliner     Berita --&gt; Hiburan     Tentang --&gt; Tentang     Kontak --&gt; Kontak   </pre> </div> <p>(Pertanyaan No 12)</p> <div data-bbox="170 1135 1468 1388" data-label="Image"> </div>

```

1  <?php
2
3  $menu = [
4      [
5          "nama" => "Beranda"
6      ],
7      [
8          "nama" => "Berita",
9          "subMenu" => [
10             [
11                 "nama" => "Wisata",
12                 "subMenu" => [
13                     [
14                         "nama" => "Pantai"
15                     ],
16                     [
17                         "nama" => "Gunung"
18                     ]
19                 ]
20             ],
21             [
22                 "nama" => "Kuliner"
23             ],
24             [
25                 "nama" => "Hiburan"
26             ]
27         ]
28     ],
29     [
30         "nama" => "Tentang"
31     ],
32     [
33         "nama" => "Kontak"
34     ]
35 ];
36
37 function tampilkanMenuBertingkat (array $menu) {
38     echo "<ul>";
39
40     foreach ($menu as $item) {
41         echo "<li>{$item['nama']}";
42
43         if (isset($item['subMenu'])) {
44             tampilkanMenuBertingkat($item['subMenu']);
45         }
46
47         echo "</li>";
48     }
49
50     echo "</ul>";
51 }
52
53 ?>
54 <!DOCTYPE html>
55 <html lang="id">
56 <head>
57     <title>Menu Bertingkat Rekursif</title>
58     <link rel="stylesheet" href="style1.css">
59 </head>
60 <body>
61
62     <?php
63     // Panggil fungsi untuk menampilkan menu
64     tampilkanMenuBertingkat($menu);
65     ?>
66
67 </body>
68 </html>

```

```

1
2  ul {
3      list-style: none;
4      padding-left: 0;
5  }
6
7  ul li {
8      margin: 5px 0;
9      padding-left: 20px;
10 }
11
12 body > ul > li {
13     list-style-type: disc;
14     padding-left: 0;
15     margin-left: 20px;
16 }
17
18 body > ul > li > ul > li {
19     list-style-type: circle;
20     margin-left: 40px;
21 }
22
23 body > ul > li > ul > li > ul > li {
24     list-style-type: square;
25     margin-left: 60px;
26 }

```

**Jawaban:**

## Tali

Dalam PHP, string dapat didefinisikan menggunakan **tanda kutip ganda** (" ") atau **tanda kutip tunggal** (' '). Meskipun keduanya memungkinkan Kita untuk membuat variabel string, ada perbedaan penting dalam cara mereka menangani variabel dan karakter khusus.

### 1. Tanda Kutipan Ganda (" ") :

- Tanda kutip ganda memungkinkan **interpolasi variabel**, yang berarti bahwa variabel dalam string akan dievaluasi dan diganti dengan nilainya.
- Karakter khusus (urutan escape) seperti `\n` (baris baru) atau `\t` (tab) dikenali dan diproses.
- **Contoh:**

```

<?php
$name = "John Wick"; echo "Halo, $name!"; //
Keluaran: Halo, John!
echo "Ini adalah baris baru.\nbaris berikutnya."; // Output: Ini adalah baris baru

```

### 2. Tanda Kutip Tunggal (' ') :

- Tanda kutip tunggal memperlakukan string secara **harfiah**, artinya variabel di dalam string tidak dievaluasi.
- Urutan escape tidak diproses, kecuali untuk `\\` (garis miring terbalik) dan `\'` (tanda kutip tunggal).
- **Contoh:**

```
<?php
$name = 'Yohanes'; echo 'Halo, $name!';    //
Keluaran: Halo, $name!
echo 'Ini adalah baris baru.\nbaris berikutnya.';    // Output: Ini adalah baris baru.\nBaris
berikutnya.
```

### Perbedaan Utama:

- **Penghuraian Variabel:** Tanda kutip ganda akan menggantikan variabel dengan nilainya, sedangkan tanda kutip tunggal akan menampilkan nama variabel sebagai teks biasa.
- **Karakter Escape:** Tanda kutip ganda mengenali urutan escape khusus seperti `\n`, `\t`, sedangkan tanda kutip tunggal hanya mengenali `\'` dan `\\`.

Secara umum, **gunakan tanda kutip ganda** saat Kita membutuhkan interpolasi variabel atau karakter khusus. Jika Kita tidak memerlukan fitur ini, **tanda kutip tunggal** lebih cepat dan lebih efisien untuk definisi string sederhana.

Beberapa operasi dapat dilakukan pada data tipe string. PHP menyediakan fungsi bawaan yang siap digunakan untuk operasi string

Fungsi	Deskripsi
<code>strlen()</code>	Untuk mengetahui panjang string
<code>str_word_count()</code>	Untuk menghitung jumlah kata dalam string
<code>strpos()</code>	Untuk menemukan posisi substring dalam string
<code>strrev()</code>	Untuk membalikkan urutan string
<code>strstr()</code>	Untuk mencari substring dalam string
<code>substr()</code>	Untuk mengekstrak substring dari posisi awal hingga akhir dalam string
<code>trim()</code>	Untuk menghapus spasi kosong dari awal dan akhir string
<code>ltrim()</code>	Untuk menghapus spasi kosong dari awal string
<code>rtrim()</code>	Untuk menghapus spasi kosong dari akhir string
<code>strtoupper()</code>	Untuk mengonversi semua karakter dalam string menjadi huruf besar
<code>strtolower()</code>	Untuk mengonversi semua karakter dalam string menjadi huruf kecil
<code>str_replace()</code>	Untuk mengganti bagian string dengan string lain
<code>ucwords()</code>	Untuk menggunakan huruf besar huruf pertama dari setiap kata dalam string
<code>meledak()</code>	Untuk membagi string menjadi <i>array</i> berdasarkan karakter tertentu

### Karakter Melarikan Diri

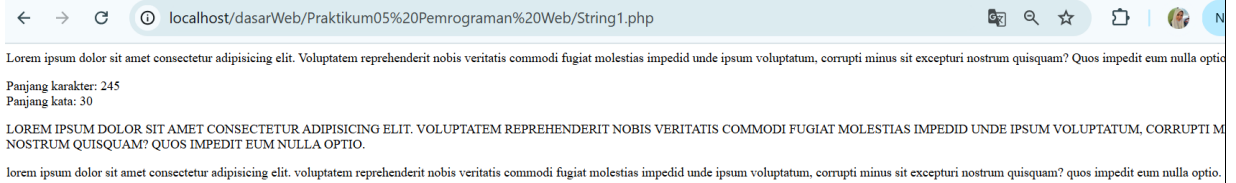
Karakter khusus yang tidak dapat ditampilkan secara langsung harus didahului dengan garis miring terbalik (`\`). String yang diapit dalam tanda kutip ganda akan mengganti karakter escape dengan karakter yang diwakilinya. Ini berbeda dari string yang diapit dalam tanda kutip tunggal, di mana ia hanya akan menampilkan konten apa adanya tanpa mengganti apa pun (dengan beberapa pengecualian).

Karakter escape di PHP adalah:

Fungsi	Deskripsi
--------	-----------

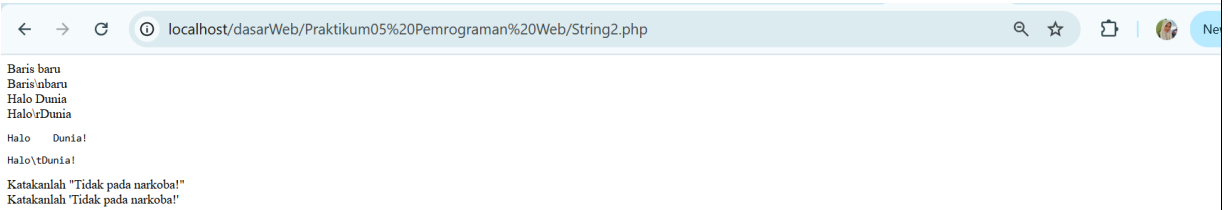
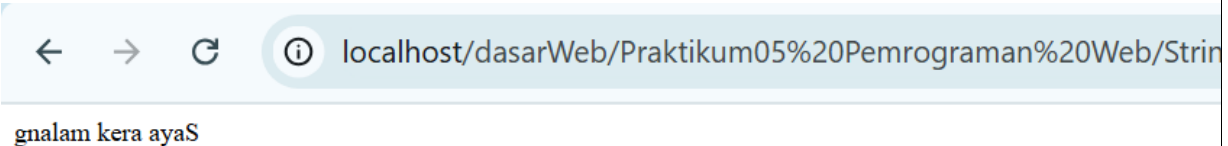
<code>\n</code>	Baris baru
<code>\r</code>	Karakter kereta kembali
<code>\t</code>	Karakter tab
<code>\\$</code>	Karakter <code>\$</code> itu sendiri
<code>\"</code>	Untuk menampilkan tanda kutip ganda
<code>\\</code>	Untuk menampilkan garis miring terbalik ( <code>\</code> ) itu sendiri

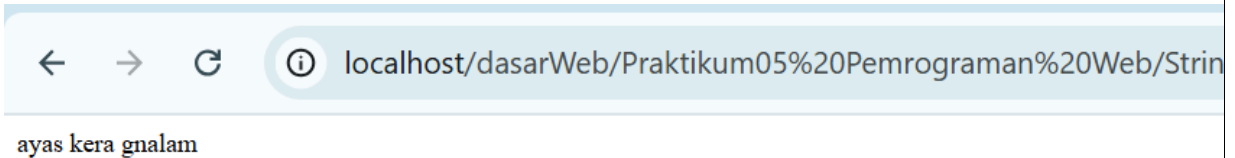
## Praktikum 7. String

Langkah	Deskripsi
1	<p>Buat file bernama <code>string1.php</code> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p> <pre>&lt;?php  \$loremIpsum = "Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatem reprehenderit nobis veritatis commodi fugiat molestias impedit unde ipsum voluptatum, corrupti minus sit excepturi nostrum quisquam? Quos impedit eum nulla optio.";  echo "&lt;p&gt;{\$loremIpsum}&lt;/p&gt;"; echo "Panjang karakter: " . strlen(\$loremIpsum) . "&lt;br&gt;"; echo "Panjang kata: " . str_word_count(\$loremIpsum) . "&lt;br&gt;"; echo "&lt;p&gt;" . strtoupper(\$loremIpsum) . "&lt;/p&gt;"; echo "&lt;p&gt;" . strtolower(\$loremIpsum) . "&lt;/p&gt;";  ?&gt;</pre>
2	<p>Amati <i>output</i> yang ditampilkan dan jelaskan hasil pengamatanmu. (Pertanyaan No 13)</p>  <p><b>Jawaban:</b></p> <p><i>Output</i> yang ditampilkan pada <i>browser</i> adalah hasil dari serangkaian operasi manipulasi <i>string</i> PHP pada variabel <code>\$loremIpsum</code>. Awalnya, teks paragraf asli ditampilkan. Kemudian, kode mencetak dua statistik penting dari <i>string</i> tersebut: panjang karakter (245) yang dihitung oleh fungsi <code>strlen()</code>, dan panjang kata (30) yang dihitung oleh <code>str_word_count()</code>. Terakhir, <i>output</i> menunjukkan dua versi teks yang sama persis, di mana yang pertama diubah sepenuhnya menjadi huruf kapital (<i>uppercase</i>) oleh fungsi <code>strtoupper()</code>, dan yang kedua diubah sepenuhnya menjadi huruf kecil (<i>lowercase</i>) oleh fungsi <code>strtolower()</code>.</p>

**Karakter Melarikan Diri**



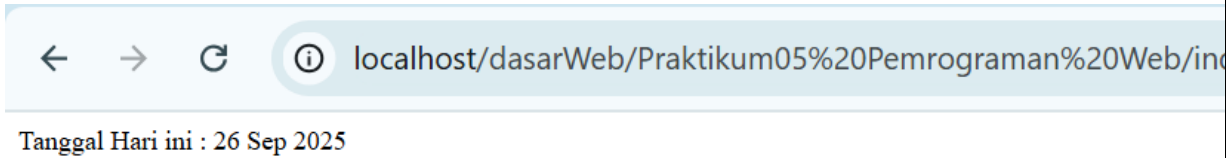
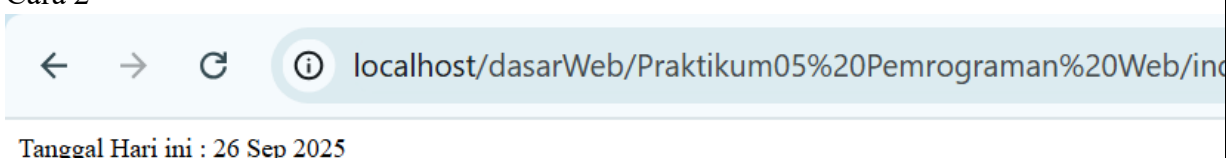
3	<p>Buat file bernama <code>string2.php</code> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p> <pre>&lt;?php echo "Baris\nbaru &lt;br&gt;"; //soal 10.a echo 'Baris\nbaru &lt;br&gt;'; //soal 10.b echo "Halo\rDunia &lt;br&gt;"; //soal 10.c echo 'Halo\rDunia &lt;br&gt;'; ///soal 10.d  echo "&lt;pre&gt;Halo\tDunia!&lt;/pre&gt;"; //soal 10.e echo '&lt;pre&gt;Halo\tDunia!&lt;/pre&gt;'; //soal 10.f  echo "Katakanlah \"Tidak pada narkoba!\" &lt;br&gt;"; //soal 10.g echo 'Katakanlah \"Tidak pada narkoba!\" &lt;br&gt;'; //soal 10.h ?&gt;</pre>
4	<p>Dari kode program di atas, Kita dapat mengamati perbedaan antara tanda kutip ganda dan tanda kutip tunggal dalam hal bagaimana mereka menangani string escape. Amati <i>output</i> dan jelaskan hasil dari setiap <i>output</i> . Kesimpulan apa yang dapat Kita tarik dari percobaan ini? (Pertanyaan No 14)</p>  <p><b>Jawaban:</b></p> <p>Dari percobaan tersebut terlihat bahwa tanda kutip ganda ( " ") di PHP akan memproses <i>escape sequence</i> seperti \n (baris baru), \r (carriage return), \t (tab), dan \" (kutip ganda), sehingga hasil output dapat berubah sesuai instruksi khusus tersebut. Sebaliknya, tanda kutip tunggal ( ' ') menampilkan hampir semua <i>escape sequence</i> sebagai teks biasa, sehingga karakter seperti \n, \r, atau \t ditampilkan apa adanya kecuali \\ dan \'. Dengan demikian, kutip ganda lebih tepat dipakai untuk string yang memerlukan karakter khusus atau interpolasi variabel, sedangkan kutip tunggal lebih sederhana dan efisien untuk teks biasa.</p>
<p><b>Membalikkan String menggunakan fungsi <code>strrev()</code></b></p>	
5	<p>Buat file bernama <code>string3.php</code> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p> <pre>&lt;?php \$pesan = "Saya arek malang"; echo strrev(\$pesan) . "&lt;br&gt;"; ?&gt;</pre>
6	<p>Amati <i>output</i> yang ditampilkan dan jelaskan hasil pengamatanmu (Pertanyaan No 15)</p>  <p><b>Jawaban:</b></p> <p>Kode PHP ini menunjukkan penggunaan fungsi manipulasi <i>string</i> <code>strrev()</code> untuk membalik urutan karakter dari sebuah <i>string</i>. Variabel <code>\$pesan</code> diinisialisasi dengan nilai "Saya arek malang". Ketika fungsi <code>strrev(\$pesan)</code> dieksekusi, ia membalikkan urutan seluruh karakter dalam <i>string</i> tersebut, termasuk spasi. Hasilnya adalah <i>output</i> yang terbalik, yaitu "gnalam kera ayaS".</p>

	ayaS", yang dicetak ke <i>browser</i> . Fungsi ini secara efektif menghasilkan <i>string</i> yang dibaca dari belakang ke depan.
7	<p>Untuk membalikkan string kata demi kata, ketik kode program berikut:</p> <pre>&lt;?php \$pesan = "saya arek malang"; # ubah variabel \$pesan menjadi array dengan perintah explode \$pesanPerKata = explode(" ", \$pesan); # ubah setiap kata dalam array menjadi kebalikannya \$pesanPerKata = array_map(fn(\$pesan) =&gt; strrev(\$pesan), \$pesanPerKata); # gabungkan kembali array menjadi string \$pesan = implode(" ", \$pesanPerKata);  echo \$pesan . "&lt;br&gt;"; ?&gt;</pre>
8	<p>Amati <i>output</i> yang ditampilkan dan jelaskan hasil pengamatanmu (Pertanyaan No 16)</p>  <p><b>Jawaban:</b></p> <p>Kode PHP ini bertujuan untuk membalik setiap kata dalam <i>string</i> dengan mempertahankan urutan kata aslinya. Proses dimulai dengan memecah <i>string</i> \$pesan menjadi array kata menggunakan <code>explode(" ", \$pesan)</code>. Selanjutnya, fungsi <code>array_map()</code> diterapkan untuk membalik setiap elemen (kata) dalam array menggunakan <code>strrev()</code>. Terakhir, array kata yang sudah dibalik tersebut digabungkan kembali menjadi satu <i>string</i> menggunakan <code>implode(" ", \$pesanPerKata)</code>, dipisahkan oleh spasi. Oleh karena itu, <i>output</i> yang akan dihasilkan adalah "ayas kera gnamal".</p>

## Menggabungkan HTML dan PHP

**Menggabungkan HTML dan PHP** adalah praktik umum dalam pengembangan web di mana PHP disematkan dalam HTML untuk menghasilkan konten secara dinamis di halaman web. Kode PHP dapat dimasukkan ke dalam HTML untuk memproses data, menangani input pengguna, atau menampilkan konten dinamis.

## Praktikum 8. HTML dan PHP

Langka h	Deskripsi
1	<p>Metode pertama adalah PHP di dalam HTML. Berikut adalah contoh kodenya:</p> <pre>&lt;html&gt; &lt;head&gt;   &lt;title&gt;Cara 01&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;p&gt;Tanggal Hari ini : &lt;?php echo date("d M Y")?&gt;&lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>
2	<p>Kode di atas adalah kode HTML yang berisi kode PHP untuk menampilkan tanggal server, ditandai dengan tag <code>&lt;?php</code> dan <code>?&gt;</code></p>
3	<p>Metode kedua adalah HTML di dalam PHP. Dalam PHP, tag HTML diperlakukan sebagai string yang diapit dalam tanda kutip, dan berbagai fungsi dapat diterapkan untuk memanipulasi string, seperti penggabungan, dll. Berikut adalah contoh cuplikan kode:</p> <pre>&lt;?php echo '&lt;html&gt;'; echo '&lt;head&gt;&lt;title&gt;Cara02&lt;/title&gt;&lt;/head&gt;'; echo '&lt;body&gt;'; echo '&lt;p&gt;Tanggal Hari ini : '.date('d M Y').'&lt;/p&gt;'; echo '&lt;/body&gt;'; echo '&lt;/html&gt;'; ?&gt;</pre>
4	<p>Kode di atas menghasilkan <i>output</i> yang sama dengan cuplikan kode sebelumnya. Namun, perbedaannya terletak pada cara kode ditulis, di mana HTML berada di dalam PHP sebagai string, dan untuk menampilkannya menggunakan tag <code>echo</code>.</p>
5	<p>Manakah dari dua metode yang menurut Kalian lebih mudah? Berikan jawabanmu bersama dengan alasan. (Pertanyaan No 17)</p> <p><b>Cara 1</b></p>  <p><b>Cara 2</b></p>  <p><b>Jawaban:</b> Menggunakan cara01 karena lebih mudah kode HTML-nya tetap bersih dan mudah dibaca (HTML dan PHP-nya terpisah), mempermudah <i>debugging</i> dan kolaborasi, sementara Cara 02 membuat sintaks HTML sulit dibedakan dari <i>string</i> PHP dan mempersulit <i>debugging</i>.</p>

## Entitas HTML

**Entitas HTML** digunakan untuk menampilkan karakter yang dicadangkan dalam HTML atau karakter yang memiliki arti khusus, seperti `<`, `>`, dan `&`. Karakter ini harus ditulis sebagai entitas untuk mencegahnya ditafsirkan sebagai kode HTML.

### Entitas HTML Umum:

Nama Entitas	Nomor Entitas	Deskripsi	Hasil
&menyalin;	&#169;	Hak cipta	©
&reg;	&#174;	Terdaftar	®
–	&#8482;	Merek dagang	™
&nbspsp;	&#161;	Ruang tanpa putus	¡
&	&	Tanda dan	&
&laquo;	&#171;	Kutipan sudut kiri	«
&raquo;	&#187;	Kutipan sudut kanan	»
"	"	Tanda kutip ganda	"
'	–	Tanda kutip tunggal	'
<	<	Kurang	<
>	&#61;	Lebih besar dari	>
&kali;	&#215;	Tanda perkalian	×
&membagi;	&#247;	Tanda pembagian	÷

## Praktikum 9. Entitas HTML

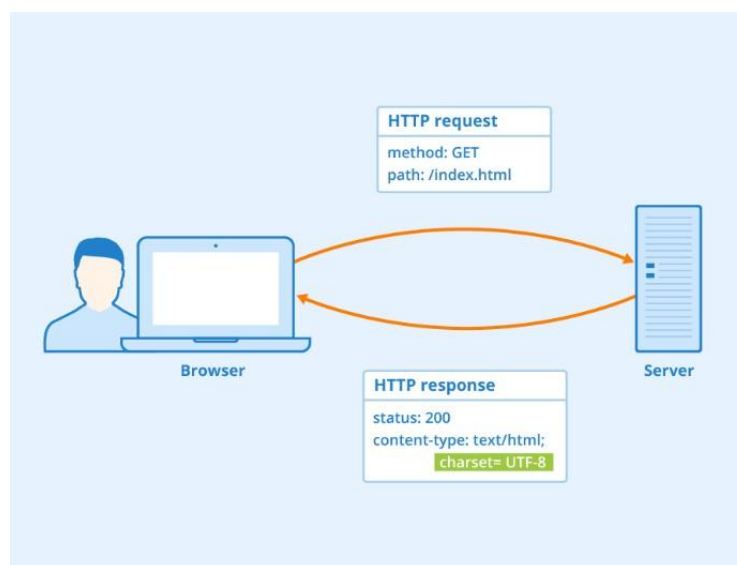
[illegible]

&#39;. Selanjutnya, paragraf kedua dimulai dengan empat *named entity* &nbsp; yang menciptakan indentasi spasi yang jelas di awal baris. Terakhir, simbol hak cipta (©) di kaki halaman dihasilkan menggunakan *numerical entity* &#169;. Hasil pengamatan pada *output* menunjukkan bahwa *browser* berhasil menerjemahkan semua *entity* ini dengan benar, menampilkan apostrof, empat spasi, teks Indonesia, dan simbol hak cipta diikuti oleh "2023 jti.com"

## Header HTTP

Header HTTP adalah data yang dikirim antara browser web dan server web sebagai sarana komunikasi antara keduanya. Header HTTP berisi informasi tentang cara menangani file yang dikirim atau diminta.

Siklus *permintaan-respons* untuk halaman web: Saat kita mengakses halaman web, browser web secara otomatis mengirim permintaan HTTP ke server web. Permintaan HTTP berisi banyak informasi, salah satunya adalah header HTTP. Di header HTTP (dikirim selama proses permintaan), ada informasi tentang file yang diminta (apakah itu file HTML, file PHP, file PDF, atau yang lainnya), serta info tambahan seperti jenis browser web yang digunakan, sistem operasi, dan alamat IP. Setelah mencapai server web, informasi di header HTTP dibaca, dan server web menyiapkan file yang diminta. Setelah itu, server web mengirim file-file tersebut kembali ke browser web. Proses pengembalian ini juga dikenal sebagai respons HTTP.



Header HTTP

Respons HTTP ini terdiri dari *dua bagian*: **header HTTP** dan **file web**. **Header HTTP** berisi informasi tentang file web yang dikirim, seperti tipe data, tanggal pengiriman, nama server web, dan sistem operasi yang digunakan oleh server web. **File web** itu sendiri terdiri dari file HTML yang membentuk halaman web, termasuk file gambar apa pun (jika ada).

Sebagai analogi, jika presiden mengunjungi rumah Kita, akan ada tim keamanan (paspampres) yang datang sebelumnya. Mereka akan memberi tahu Kita bahwa presiden akan tiba pada waktu tertentu, dengan jumlah orang tertentu, dan memberikan informasi relevan lainnya. Tim keamanan ini dapat dibandingkan dengan header HTTP, yang tiba sebelum file yang sebenarnya dikirim. Dalam praktiknya, kita sering tidak mengetahui header HTTP, dan banyak yang bahkan mungkin belum pernah mendengar istilah tersebut. Ini normal, karena konten header HTTP dimaksudkan untuk pemrosesan browser web, bukan untuk pengunjung situs web.

Bagaimana Kita bisa melihat header HTTP di browser web Kita? Jelaskan dan sertakan langkah-langkahnya. (Pertanyaan No 19)

**Jawaban:**

Untuk melihat header HTTP di browser, kita dapat memanfaatkan fitur Developer Tools yang tersedia di hampir semua browser. Caranya, buka halaman web yang ingin diperiksa, lalu tekan F12 atau klik kanan dan pilih Inspect/Periksa. Setelah itu, masuk ke tab Network, kemudian lakukan refresh halaman agar permintaan HTTP muncul. Klik salah satu request (biasanya yang paling atas adalah request utama halaman), lalu pilih sub-tab Headers untuk melihat detail request dan response, termasuk informasi penting seperti metode HTTP, status code, content-type, cookie, dan header lainnya.

## Tanggal dan Waktu

Fungsi `date()` di PHP digunakan untuk menampilkan tanggal dan waktu. Sintaks fungsi `date()` adalah sebagai berikut:

```
<?php  
date(format, timestamp);
```

**format:** Parameter wajib yang menentukan bagaimana tanggal/waktu harus diformat. Ini dapat mencakup karakter untuk hari, bulan, tahun, jam, menit, dan detik.

**timestamp:** Parameter opsional yang menentukan stempel waktu. Jika dihilangkan, tanggal dan waktu saat ini akan digunakan.

Parameter **format** diperlukan. Parameter **format** digunakan untuk menentukan bagaimana tanggal dan/atau waktu akan diformat. Di bawah ini adalah beberapa karakter umum yang digunakan untuk pemformatan tanggal:

1. **d**: Mewakili hari (01 hingga 31)
2. **m**: Mewakili bulan (01 hingga 12)
3. **y**: Mewakili tahun (dalam 4 digit)
4. **l**: Mewakili hari dalam seminggu

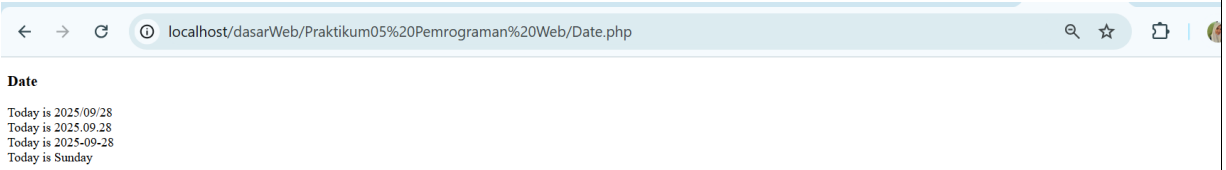
Selain menampilkan tanggal, fungsi ini juga dapat menampilkan waktu. Berikut adalah beberapa karakter umum yang digunakan untuk pemformatan waktu:

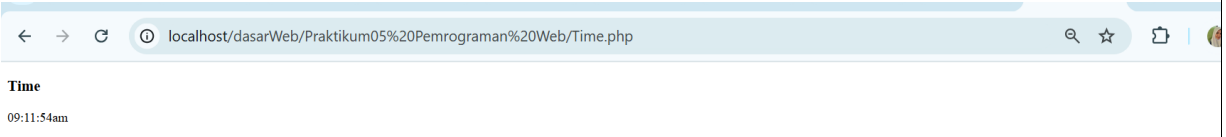
1. **H**: Mewakili jam dalam format 24 jam
2. **h**: Mewakili jam dalam format 12 jam
3. **i**: Mewakili menit (00 hingga 59)
4. **s**: Mewakili detik (00 hingga 59)
5. **a**: Mewakili ante meridiem (am) atau post meridiem (pm).



## Praktikum 10. Tanggal

Ikuti langkah-langkah ini untuk memahami cara menggunakan fungsi `date()`:

Langka h	Deskripsi
1	<p>Buat file baru bernama <code>date.php</code> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p> <pre>1 &lt;!DOCTYPE HTML&gt; 2 &lt;html&gt; 3   &lt;head&gt; 4   &lt;/head&gt; 5   &lt;body&gt; 6     &lt;h3&gt; Date &lt;/h3&gt; 7     &lt;?php 8       echo "Today is " . date("Y/m/d") . "&lt;br&gt;"; 9       echo "Today is " . date("Y.m.d") . "&lt;br&gt;"; 10      echo "Today is " . date("Y-m-d") . "&lt;br&gt;"; 11      echo "Today is " . date("l"); 12    ?&gt; 13  &lt;/body&gt; 14 &lt;/html&gt;</pre>
2	Simpan file dan jalankan program
3	<p>Amati <i>output</i> yang ditampilkan dan jelaskan hasil pengamatanmu! (Pertanyaan No 19)</p> <p><b>Jawaban :</b></p>  <p><i>Output</i> yang ditampilkan adalah hasil dari kode PHP yang menggunakan fungsi <code>date()</code> untuk mencetak tanggal dan hari dalam berbagai format. Baris pertama hingga ketiga menunjukkan tanggal yang sama 2025/09/28 dalam format berbeda-beda (dipisahkan oleh garis miring /, titik ., dan <i>hyphen</i> -), yang ditentukan oleh <i>format string</i> "Y/m/d", "Y.m.d", dan "Y-m-d". Baris terakhir menampilkan nama lengkap hari ini, "Sunday", yang dicetak menggunakan <i>format character</i> "l" (huruf kecil 'L'). Secara keseluruhan, <i>output</i> ini berhasil mendemonstrasikan fleksibilitas fungsi <code>date()</code> PHP dalam menampilkan informasi waktu yang berbeda berdasarkan <i>format character</i> yang diberikan.</p>
4	<p>Buat file baru bernama <code>time.php</code> di dalam direktori JS05_PHP-2, lalu ketik kode berikut:</p> <pre>1 &lt;!DOCTYPE HTML&gt; 2 &lt;html&gt; 3   &lt;head&gt; 4   &lt;/head&gt; 5   &lt;body&gt; 6     &lt;h3&gt; Time &lt;/h3&gt; 7     &lt;?php 8       date_default_timezone_set("asia/jakarta"); 9       echo date("h:i:sa"); 10    ?&gt; 11  &lt;/body&gt; 12 &lt;/html&gt;</pre>

5	Simpan file dan jalankan program
6	<p>Amati <i>output</i> yang ditampilkan dan jelaskan hasil pengamatanmu! (Pertanyaan No 20)</p>  <p><b>Jawaban :</b></p> <p>Kode PHP ini bertujuan untuk menampilkan waktu lokal saat ini setelah zona waktu telah ditetapkan. Baris kode <code>date_default_timezone_set("asia/jakarta");</code> berfungsi untuk mengatur zona waktu default PHP ke Asia/Jakarta. Setelah zona waktu ditetapkan, fungsi <code>date("h:i:sa")</code> dipanggil. <i>Format string</i> "h:i:sa" menginstruksikan PHP untuk menampilkan jam dalam format 12-jam (h), menit (i), detik (s), dan penanda am/pm (a). Oleh karena itu, <i>output</i> 09:11:56am yang ditampilkan di <i>browser</i> merupakan waktu saat ini di zona waktu Jakarta dengan format 12-jam.</p>

## Variabel Superglobal PHP

Variabel Superglobal PHP adalah variabel yang telah ditentukan sebelumnya dalam PHP yang dapat diakses dari mana saja dalam skrip. Mereka tersedia secara global, artinya mereka dapat digunakan dalam fungsi, kelas, atau file apa pun tanpa perlu mendeklarasikannya sebagai global. Superglobal digunakan untuk menangani berbagai jenis data, seperti input formulir, detail server, informasi sesi, dan banyak lagi.

Variabel superglobal menyimpan banyak data penting dan berguna yang dapat kita gunakan untuk menyelesaikan proyek yang sedang kita kerjakan. Ada 9 variabel superglobal di PHP.

## Praktikum 11. Variabel Superglobal

### 1. `$_SERVER`

Variabel pertama dan terpenting adalah variabel `$_SERVER`. Ini adalah *array* asosiatif yang menyediakan berbagai jenis informasi tentang permintaan yang ditangkap oleh server. Data tersebut mencakup header, jalur, lokasi skrip, dan lainnya.

Nilai yang disimpan dalam variabel `$_SERVER` disediakan oleh server web, yang berarti tidak ada jaminan khusus bahwa setiap server web yang kami gunakan akan menyediakan semua data standar yang tersedia.

Untuk mengetahui nilai apa yang tersedia dalam variabel `$_SERVER`, kita dapat menjalankan perintah berikut dan menyimpannya di `global_server.php`

```
<?php
// Menampilkan semua informasi yang tersedia dalam array $_SERVER echo '<pra>';
print_r($_SERVER); echo '</pra>';
?>
```

Jalankan kode program di atas, lalu jelaskan *output* dari setiap perintah **echo**. (Soal no.22)

**Jawaban:**

```
Array ( [HTTP_HOST] => localhost [HTTP_CONNECTION] => keep-alive [HTTP_SEC_CH_UA] => "Google Chrome" [HTTP_SEC_CH_UA_MOBILE] => ? [HTTP_SEC_CH_UA_PLATFORM] => "Windows" [HTTP_UPGRADE_INSECURE_REQUESTS] => 1 [HTTP_USER_AGENT] => Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 [HTTP_SEC_FETCH_SITE] => none [HTTP_SEC_FETCH_MODE] => navigate [HTTP_SEC_FETCH_USER] => ? [HTTP_SEC_FETCH_DEST] => document [HTTP_ACCEPT_ENCODING] => gzip, deflate, br, zstd [HTTP_ACCEPT_LANGUAGE] => en-US,en;q=0.9,id;q=0.8 [PATH] => C:\Program Files\Common Files\Oracle\Java\javapath;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0;C:\WINDOWS\System32\OpenSSH;C:\Program Files\Java\jdk-24\bin;C:\xampp\mysql\bin;C:\xampp\php\bin;C:\laragon\bin\php\php-8.3.16-Win32-vs16-x64;C:\Program Files\Git\cmd;C:\Program Files\nodejs\;C:\Users\Umi Maharani\AppData\Local\Microsoft\WindowsApps\;C:\Users\Umi Maharani\AppData\Local\Programs\Microsoft VS Code\bin;C:\Users\Umi Maharani\AppData\Roaming\Composer\vendor\bin;C:\Users\Umi Maharani\AppData\Roaming\upm; [SystemRoot] => C:\WINDOWS [COMSPEC] => C:\WINDOWS\system32\cmd.exe [PATHEXT] => .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC [WINDIR] => C:\WINDOWS [SERVER_SIGNATURE] => (SERVER_SOFTWARE) => Apache/2.4.62 (Win64) OpenSSL/3.0.15 PHP/8.3.16 [SERVER_NAME] => localhost [SERVER_ADDR] => ::1 [SERVER_PORT] => 80 [REMOTE_ADDR] => ::1 [DOCUMENT_ROOT] => C:\laragon\www [REQUEST_SCHEME] => http [CONTEXT_PREFIX] => [CONTEXT_DOCUMENT_ROOT] => C:\laragon\www [SERVER_ADMIN] => admin@example.com [SCRIPT_FILENAME] => C:\laragon\www\dasarWebPraktikum05 Pemrograman Web GlobalServer.php [REMOTE_PORT] => 50634 [GATEWAY_INTERFACE] => CGI/1.1 [SERVER_PROTOCOL] => HTTP/1.1 [REQUEST_METHOD] => GET [QUERY_STRING] => [REQUEST_URI] => /dasarWebPraktikum05 Pemrograman Web GlobalServer.php [SCRIPT_NAME] => /dasarWebPraktikum05 Pemrograman Web GlobalServer.php [PHP_SELF] => /dasarWebPraktikum05 Pemrograman Web GlobalServer.php [REQUEST_TIME_FLOAT] => 1759025605.2589 [REQUEST_TIME] => 1759025605 )
```

**Jawaban:**

Berdasarkan kode PHP yang Anda berikan, perintah `echo '<pre>'; print_r($_SERVER); echo '</pre>';` bertujuan untuk menampilkan semua data yang ada di dalam array superglobal `$_SERVER`. Array ini berisi informasi tentang *server web*, lingkungan eksekusi, dan permintaan HTTP saat ini.

Perintah `echo` pertama mencetak tag `<pre>`, yang berfungsi untuk mempertahankan pemformatan teks *preformatted* di HTML, memastikan *output* array yang kompleks dapat dibaca. Selanjutnya, perintah `print_r($_SERVER)` mencetak seluruh isi array superglobal `$_SERVER`, yang memuat detail lingkungan *server* dan permintaan klien, seperti alamat IP (*Remote Address*), nama *server*, *browser* klien (*User Agent*), dan *request method*. Terakhir, perintah `echo` ketiga mencetak tag penutup `</pre>`, yang mengakhiri pemformatan teks yang mudah dibaca tersebut.

Berikut adalah beberapa contoh data dari **variabel `$_SERVER`** yang sering dibutuhkan:

Tidak	Variabel	Deskripsi
1	<code>\$_SERVER['PHP_SELF']</code>	Berisi nama file yang sedang dijalankan, diambil dari akar dokumen.
2	<code>\$_SERVER['SERVER_ADDR']</code>	Alamat IP server tempat file sedang dieksekusi.
3	<code>\$_SERVER['SERVER_NAME']</code>	Nama host server tempat file PHP sedang dijalankan. Nama host biasanya adalah nama PC dalam jaringan. Jika skrip PHP dijalankan pada Host Virtual, nama host virtual akan digunakan sebagai nama server.
4	<code>\$_SERVER['SERVER_PROTOCOL']</code>	Protokol komunikasi yang saat ini digunakan, seperti HTTP atau HTTPS. Contoh: 'HTTP/0.1'.
5	<code>\$_SERVER['REQUEST_METHOD']</code>	Berisi metode permintaan file PHP yang sedang dieksekusi, seperti GET, POST, PUT, DELETE, OPTIONS.
6	<code>\$_SERVER['QUERY_STRING']</code>	Mengembalikan string kueri dari file PHP yang sedang dieksekusi. Misalnya, jika pengguna mengakses
Tidak	Variabel	Deskripsi

		<code>http://localhost/halo-dunia?nama=Budi&amp;umur=20&amp;asal=Surabaya</code> , variabel ini akan mengembalikan nilai <code>nama=Budi umur=20 asal=Surabaya</code>
7	<code>\$_SERVER['DOCUMENT_ROOT']</code>	Direktori akar dokumen dari file PHP yang sedang dieksekusi, dikembalikan berdasarkan pengaturan server.
8	<code>\$_SERVER['HTTP_HOST']</code>	Mengembalikan konten host, seperti header (jika tersedia).
9	<code>\$_SERVER['HTTP_REFERER']</code>	URL halaman yang merujuk ke halaman saat ini yang sedang dijalankan. Jika tidak ada, nilainya kosong.
10	<code>\$_SERVER['HTTP_USER_AGENT']</code>	Berisi informasi tentang pengguna yang membuat permintaan, termasuk browser, bahasa, dan sistem operasi. Contoh: Mozilla/4.5 [id] (X11; U; Linux 2.2.9 i586).
11	<code>\$_SERVER['REMOTE_ADDR']</code>	Alamat IP pengguna yang mengakses halaman PHP.
12	<code>\$_SERVER['SCRIPT_FILENAME']</code>	Nama jalur absolut dari file yang sedang dieksekusi.
13	<code>\$_SERVER['REQUEST_URI']</code>	URI file yang sedang dieksekusi. Contoh: <code>"/php/halodunia"</code> .

## 2. `$_GET`

Variabel `$_GET` adalah *array* asosiatif yang berisi nilai dari string kueri. Misalnya, Buat file `global_get.php`, dan tulis kode sebagai berikut:

```
<?php

$nama = @$_GET['nama']; //tanda @ agar tidak ada peringatan error
ketika key-nya kosong
$usia = @$_GET['usia']; //tanda @ agar tidak ada peringatan error
ketika key-nya kosong

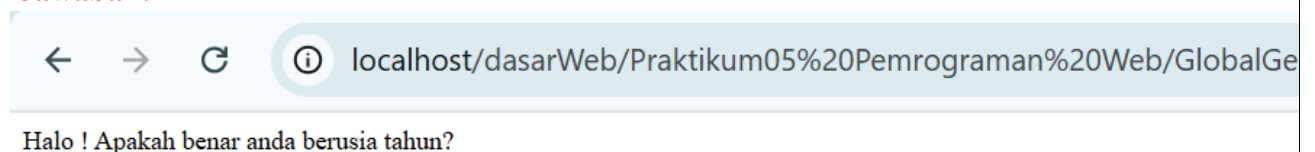
echo "Halo {$nama}! Apakah benar anda berusia {$usia} tahun?";
?>
```

Jalankan url ini di browser Kita

`localhost/dasarWeb/JS05_PHP-2/global_get.php?nama=Elok&usia=37`

Output apa yang dihasilkan, amati, dan jelaskan hasilnya. (Pertanyaan No 23)

**Jawaban:**



**Jawaban:**

Kode PHP ini mendemonstrasikan cara mengambil nilai dari URL menggunakan array superglobal `$_GET` dan menampilkannya, meskipun tidak ada parameter yang diberikan. Variabel `$nama` dan

\$usia mencoba mengambil nilai dari *key* 'nama' dan 'usia' dari *query string* URL. Penggunaan simbol @ di depan setiap pengambilan nilai (@\$\_GET[...]) berfungsi untuk menekan pesan kesalahan *notice* yang muncul ketika *key* tersebut tidak ada dalam URL. Karena URL yang dieksekusi (GlobalGet.php) tidak memiliki parameter *query* apa pun, \$nama dan \$usia diinisialisasi sebagai *empty* atau *null*, yang kemudian ditampilkan dalam *output* sebagai: "Halo ! Apakah benar anda berusia tahun?"

### 3. \$\_POST

Variabel \$\_POST mirip dengan variabel \$\_GET. Namun, data **tidak diteruskan melalui** string kueri di URL, melainkan di **isi permintaan**. Selain itu, metode permintaan yang digunakan harus POST.

Buat file `global_post.php`, dan tulis kode sebagai berikut:

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_POST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
</html>
```

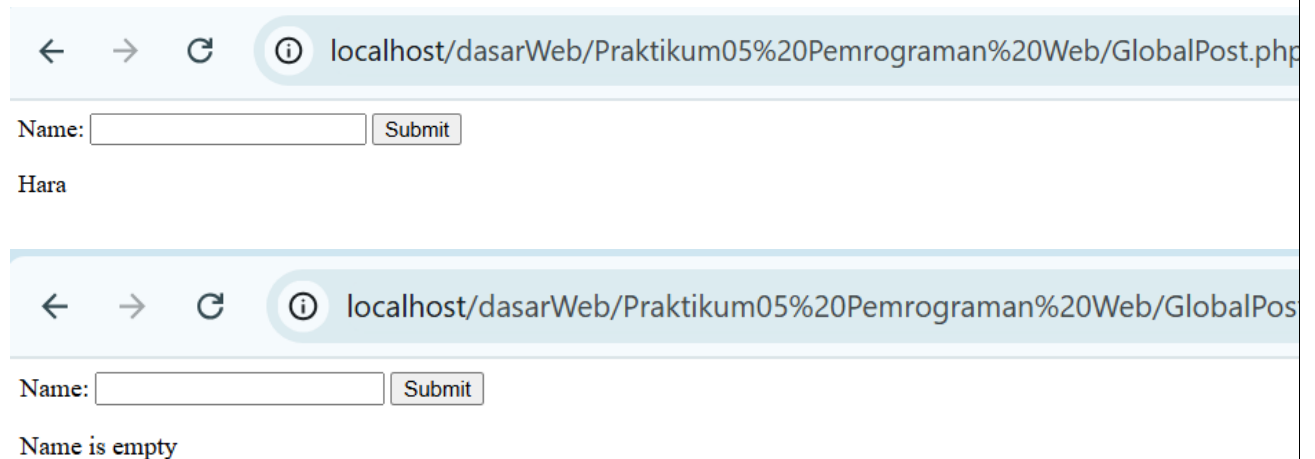
Jalankan url ini di browser Kita

`localhost/dasarWeb/JS05_PHP-2/global_post.php`

Kirim formulir dan *output* apa yang dihasilkan? Amati dan jelaskan hasilnya.

(Pertanyaan No 24)

**Jawaban:**



The first screenshot shows a web browser with the URL `localhost/dasarWeb/Praktikum05%20Pemrograman%20Web/GlobalPost.php`. The form has a text input field labeled "Name:" and a "Submit" button. Below the form, the text "Hara" is displayed.

The second screenshot shows the same browser with the URL `localhost/dasarWeb/Praktikum05%20Pemrograman%20Web/GlobalPos`. The form has a text input field labeled "Name:" and a "Submit" button. Below the form, the text "Name is empty" is displayed.

**Jawaban:**

Kode program tersebut menggunakan form HTML dengan method POST untuk mengirim data ke halaman yang sama, lalu PHP memproses data dari input fname. Jika pengguna mengetikkan sebuah nilai, misalnya *Hara*, maka output yang ditampilkan adalah teks Hara karena variabel \$name berisi data dari \$\_POST['fname']. Sebaliknya, jika pengguna mengirimkan form tanpa mengisi apapun di kolom teks, kondisi empty(\$name) bernilai benar sehingga muncul pesan Name is empty. Dari percobaan ini dapat dilihat bahwa metode POST berhasil mengirimkan data input ke server, dan script PHP bisa membedakan apakah field kosong atau berisi nilai.

**4. \$\_SESSION**

Variabel **\$\_SESSION** adalah *array* asosiatif yang menyimpan data sesi pengguna. Variabel ini dapat digunakan untuk menyimpan pengguna yang masuk untuk sesi tertentu. Ini juga dapat digunakan untuk menyimpan data keranjang di toko online. Secara default, masa pakai sesi di PHP adalah 1440 detik atau 24 menit.

**5. \$\_COOKIE**

Mirip dengan **\$\_SESSION**, variabel **\$\_COOKIE** dapat digunakan untuk menyimpan data yang terkait dengan pengguna, seperti informasi login, detail keranjang di toko online, dan sebagainya. Perbedaannya adalah bahwa cookie adalah file kecil yang disimpan di browser pengguna. File ini dikirim setiap kali browser membuat permintaan ke server. Masa pakai cookie umumnya lebih lama daripada sesi.

**6. \$\_REQUEST**

Variabel **\$\_REQUEST** adalah *array* asosiatif yang menyimpan gabungan nilai dari variabel **\$\_GET**, **\$\_POST**, dan **\$\_COOKIE** yang kesemuanya berhubungan dengan data yang dikirim bersamaan dengan *request* user.

Buat file **global\_request.php**, dan tulis kode sebagai berikut:

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_REQUEST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
</html>
```



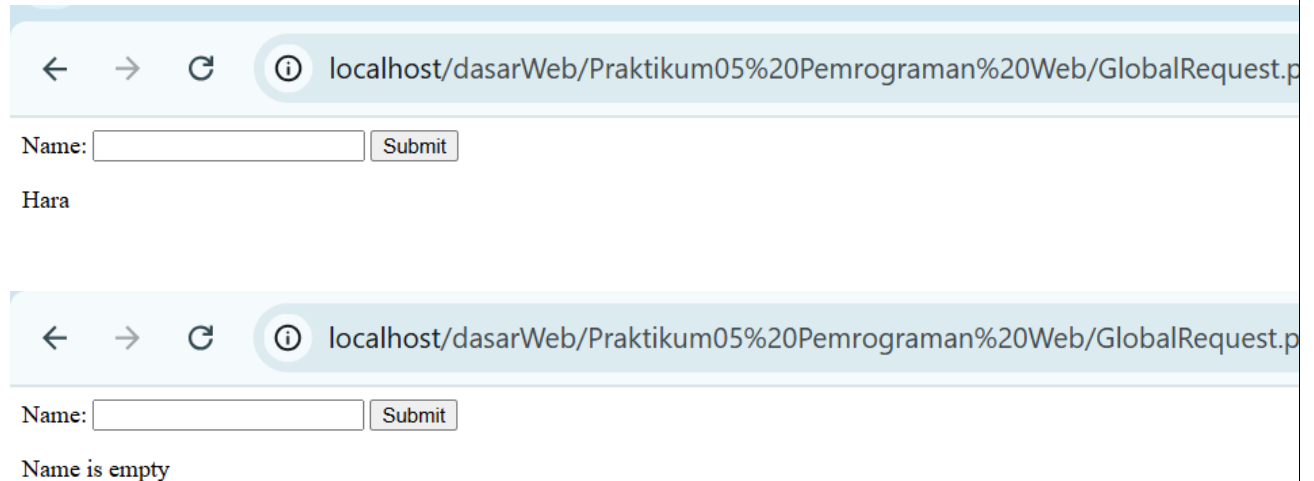
Jalankan url ini di browser Kita

`localhost/dasarWeb/JS05_PHP-2/global_request.php`

Kirim formulir dan *output* apa yang dihasilkan? Amati dan jelaskan hasilnya. Dan apa bedanya dengan variabel global `$_POST`?

(Pertanyaan No 25)

**Jawaban:**



The first screenshot shows a web browser at the URL `localhost/dasarWeb/Praktikum05%20Pemrograman%20Web/GlobalRequest.p`. It displays a form with a label 'Name:' followed by an input field and a 'Submit' button. Below the form, the output 'Hara' is displayed.

The second screenshot shows the same browser after the form is submitted. The output now reads 'Name is empty'.

Perbedaan utama kedua kode terletak pada penggunaan array superglobal PHP: `$_POST` pada Kode 1 dan `$_REQUEST` pada Kode 2. `$_POST` hanya mengambil data yang dikirim melalui metode HTTP POST (seperti yang digunakan formulir Anda), menjadikannya lebih spesifik dan aman untuk data formulir. Sebaliknya, `$_REQUEST` secara *default* dapat mengambil data dari `$_POST`, `$_GET` (dari URL), dan `$_COOKIE`, sehingga lebih umum dan kurang spesifik. Karena kedua formulir Anda menggunakan `method="post"` dan PHP-nya memvalidasi metode POST, output dari kedua kode akan sama persis—baik menampilkan nama yang dimasukkan atau pesan "Name is empty".

## 7. `$_FILES`

Variabel `$_FILES` adalah *array* asosiatif yang menyimpan data tentang file yang diunggah oleh pengguna dalam satu permintaan menggunakan **metode POST** atau **PUT**.

## 8. `$_ENV`

Variabel `$_ENV` adalah *array* asosiatif yang berisi data tentang lingkungan tempat skrip PHP berjalan.

Variabel `$_ENV` disediakan oleh shell yang menjalankan skrip PHP, sehingga nilainya dapat bervariasi tergantung pada sistem operasi yang digunakan.

Dalam kerangka kerja PHP modern seperti Laravel, **variabel `$_ENV`** juga digunakan untuk menyimpan informasi terkait lingkungan, seperti nama database, kata sandi database, dan nilai lain yang diperlukan untuk mengonfigurasi kerangka kerja.

## 9. \$GLOBALS

Variabel \$GLOBALS adalah *array* asosiatif yang menyimpan semua variabel global yang ditentukan saat program berjalan. Variabel \$GLOBALS adalah superglobal PHP yang digunakan untuk mengakses variabel global dari mana saja dalam skrip PHP (termasuk fungsi atau metode di dalam).

Buat file `global_globals.php`, dan tulis kode sebagai berikut:

```
<?php
$x = 75;
$y = 25;

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

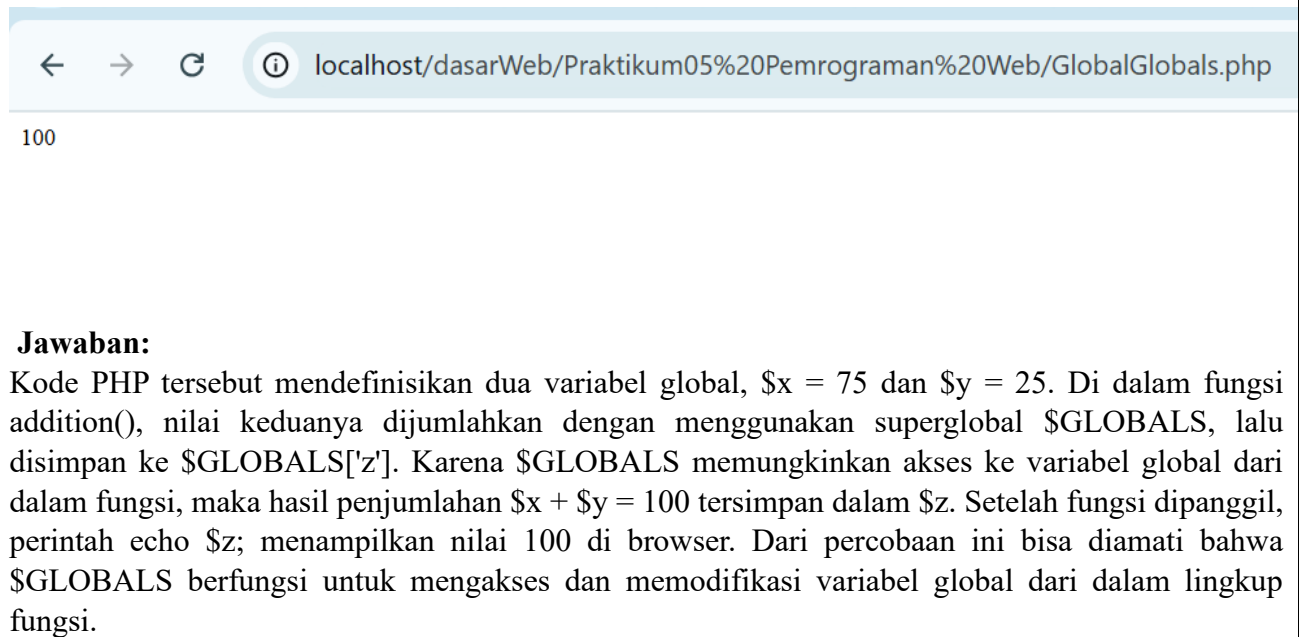
addition();
echo $z;
?>
```

Jalankan url ini di browser Kita

[localhost/dasarWeb/JS05\\_PHP-2/global\\_globals.php](localhost/dasarWeb/JS05_PHP-2/global_globals.php)

Kirim formulir dan *output* apa yang dihasilkan? Amati dan jelaskan hasilnya. (Pertanyaan No 26)

**Jawaban:**



**Jawaban:**

Kode PHP tersebut mendefinisikan dua variabel global,  $x = 75$  dan  $y = 25$ . Di dalam fungsi `addition()`, nilai keduanya dijumlahkan dengan menggunakan superglobal `$GLOBALS`, lalu disimpan ke `$GLOBALS['z']`. Karena `$GLOBALS` memungkinkan akses ke variabel global dari dalam fungsi, maka hasil penjumlahan  $x + y = 100$  tersimpan dalam `$z`. Setelah fungsi dipanggil, perintah `echo $z;` menampilkan nilai 100 di browser. Dari percobaan ini bisa diamati bahwa `$GLOBALS` berfungsi untuk mengakses dan memodifikasi variabel global dari dalam lingkup fungsi.

## Referensi:

- 1) Nixon, Robin. (2018). Belajar PHP, MySQL, JavaScript, CSS & HTML: Panduan Langkah demi Langkah untuk Membuat Situs Web Dinamis, Edisi ke-5. O'Reilly Media, Inc.
- 2) Forbes, Alan. (2012). Kegembiraan PHP: Panduan Pemula untuk Pemrograman Aplikasi Web Interaktif dengan PHP dan MySQL, Edisi ke-5. Penerbitan Pulau Plum