



Topik

1. konsep *function* pada pemrograman PHP
2. konsep penggabungan pada pemrograman PHP dan HTML

Tujuan

Mahasiswa diharapkan mampu:

1. Mahasiswa paham dengan konsep *function* pada pemrograman PHP
2. Mahasiswa paham dengan konsep penggabungan pada pemrograman PHP dan HTML

Pendahuluan

Pengenalan Array dan Fungsi

Array atau larik adalah salah satu tipe data. Array bukan merupakan tipe data dasar seperti integer atau boolean, tetapi array adalah sebuah tipe data yang terdiri dari kumpulan tipe data lainnya. Array memudahkan dalam membuat kelompok data, menghemat penulisan, dan penggunaan variabel. Di dalam PHP terdapat 3 jenis array, yaitu *indexed array*, *associative array*, dan *multidimensional array*.

Selain tipe data, pemahaman tentang fungsi adalah salah satu yang dibutuhkan ketika membuat program. Dalam bahasa pemrograman, fungsi didefinisikan sebagai kode program yang dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Ketika merancang kode program kadang kita sering membuat kode yang melakukan tugas yang sama secara berulang-ulang, seperti membaca tabel dari database, menampilkan penjumlahan, dan lain-lain. Tugas yang sama ini akan lebih efektif jika dipisahkan dari program utama, dan dirancang menjadi sebuah fungsi.

Praktikum Bagian 1. *Indexed Array*

Indexed array adalah array dengan indeks numerik. Penulisan *indexed array* bisa dilakukan secara otomatis atau diberikan indeks secara manual. Indeks array selalu dimulai dari 0. Berikut adalah cara penulisan *indexed array*:

```
<?php
    $variable = array(
        "value0",
        "value1",
        "value2",
        :
        "value-n"
    ); ?>
```

atau bisa juga dituliskan seperti berikut:

```
<?php
    $variable[0] = "value0";
```

```

$variable[1] = "value1";
$variable[2] = "value2";      :
$variable[n] = "value-n";
?>

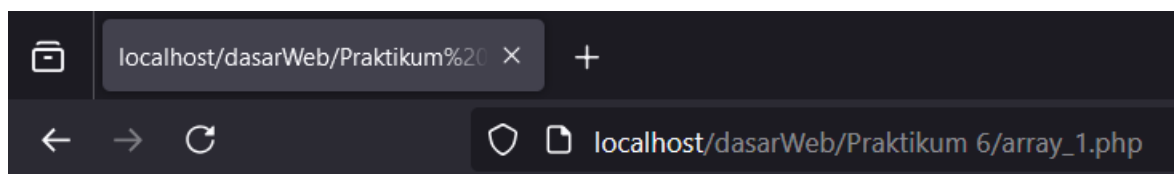
```

Ikuti langkah-langkah berikut untuk memahami *indexed* array di dalam PHP:

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>array_1.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> <!DOCTYPE html> <html> <head> </head> <body> <h2>Array Terindeks</h2> <?php \$Listdosen=["Elok Nur Hamdana","Unggul Pamenang", "Bagas Nugraha"]; echo \$Listdosen[2] . "
"; echo \$Listdosen[0] . "
"; echo \$Listdosen[1] . "
"; ?> </body> </html> </pre>
2	<p>Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/array_1.php</code></p>
3	<p>Amati hasil yang ditampilkan</p>
4	<p>Untuk menampilkan array, selain menggunakan indeks kita juga bisa menggunakan perulangan. Coba tampilkan hasil dari kode program diatas dengan menggunakan perulangan. Tampilkan kode program dan hasilnya. (soal no.1)</p> <p>Kode Program:</p>

```
Praktikum 6 > array_1.php > html > body
You, 2 minutes ago | 1 author (You)
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      // echo $Listdosen[2] . "<br>";
11      // echo $Listdosen[0] . "<br>";
12      // echo $Listdosen[1] . "<br>";
13      foreach ($Listdosen as $dosen) {
14        echo $dosen . "<br>";
15      }
16    ?>
17  </body>
18 </html>
```

Hasil:



Array Terindeks

Elok Nur Hamdana
Unggul Pamenang
Bagas Nugraha

Praktikum Bagian 2. *Associative Array*

Komponen *associative array* terdiri dari pasangan kunci (*key*) dan nilai (*value*). Kunci menunjukkan posisi dimana nilai disimpan. PHP menggunakan tanda panah (\Rightarrow) untuk mendefinisikan nilai kepada kunci. Berikut adalah cara penulisan *associative array*:

```
<?php
$variable = array(
    "Key0" => "value0",
    "key1" => "value1",
```

```

        "key2" => "value2",
        :
        "key-n" => "value-n"
    );
?>

```

atau bisa juga ditulis seperti berikut:

```

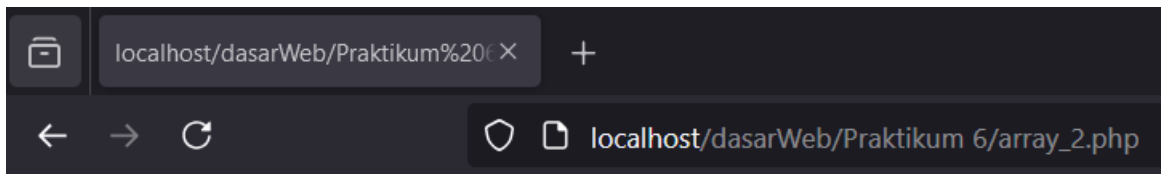
<?php
$variable['key0'] = "value0";
$variable['key1'] = "value1";
$variable['key2'] = "value2";      :
$variable['key-n'] = "value-n";    ?>

```

Ikuti langkah-langkah berikut untuk memahami *associative* array di dalam PHP:

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>array_2.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> <!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title></title> </head> <body> <?php \$Dosen = ['nama' => 'Elok Nur Hamdana', 'domisili' => 'Malang', 'jenis_kelamin' => 'Perempuan']; echo "Nama : {\$Dosen ['nama']}
"; echo "Domisili : {\$Dosen ['domisili']}
"; echo "Jenis Kelamin : {\$Dosen ['jenis_kelamin']}
"; ?> </body> </html> </pre>
2	<p>Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/array_2.php</code></p>
3	<p>Amati hasil yang ditampilkan. Kemudian tambahkan style tabel pada output tampilan tersebut supaya lebih menarik. (soal no.2) *Untuk penggunaan style bebas boleh internal atau eksternal file.</p>

Sebelum Style:



Nama : Elok Nur Hamdana
Domisili : Malang
Jenis Kelamin : Perempuan

Sesudah Style:

Informasi	Detail
Nama	Elok Nur Hamdana
Domisili	Malang
Jenis Kelamin	Perempuan

Praktikum Bagian 3. *Multidimensional* Array

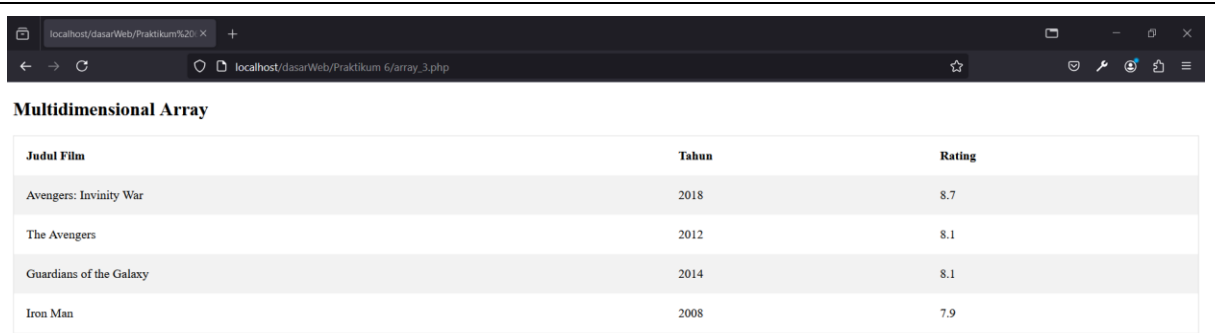
Multidimensional array adalah array yang terdiri dari satu atau lebih array. Berikut adalah contoh penulisan array dengan dimensi 2:

```
<?php
$variable = array(
    array ("value00", "value01", ..., "value-nm"),
    array ("value10", "value11", ..., "value-nm"),
    array ("value20", "value21", ..., "value-nm"),
);
?>
```

Ikuti langkah-langkah berikut untuk memahami multidimensional array di dalam PHP:

Langkah	Keterangan
---------	------------

1	<p>Buat file baru dengan nama <code>style.css</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan 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 dengan nama <code>array_3.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 <link rel="stylesheet" type="text/css" href="style.css"/> 5 </head> 6 <body> 7 <h2> Multidimensional Array </h2> 8 <table> 9 <tr> 10 <th>Judul Film</th> 11 <th>Tahun</th> 12 <th>Rating</th> 13 </tr> 14 <?php 15 \$movie = array(16 array("Avengers: Infinity 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 "<tr>"; 22 echo "<td>". \$movie[0][0] . "</td>"; 23 echo "<td>". \$movie[0][1] . "</td>"; 24 echo "<td>". \$movie[0][2] . "</td>"; 25 echo "</tr>"; 26 echo "<tr>"; 27 echo "<td>". \$movie[1][0] . "</td>"; 28 echo "<td>". \$movie[1][1] . "</td>"; 29 echo "<td>". \$movie[1][2] . "</td>"; 30 echo "</tr>"; 31 echo "<tr>"; 32 echo "<td>". \$movie[2][0] . "</td>"; 33 echo "<td>". \$movie[2][1] . "</td>"; 34 echo "<td>". \$movie[2][2] . "</td>"; 35 echo "</tr>"; 36 echo "<tr>"; 37 echo "<td>". \$movie[3][0] . "</td>"; 38 echo "<td>". \$movie[3][1] . "</td>"; 39 echo "<td>". \$movie[3][2] . "</td>"; 40 echo "</tr>"; 41 <?> 42 </table> 43 </body> 44 </html></pre>
3	<p>Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/array_3.php</code></p>
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 3)</p>



Multidimensional Array

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

Membuat tabel yang menampilkan judul film, tahun rilis, dan rating untuk empat film yang berbeda dengan menggunakan array multidimensional untuk menampilkan data dalam format yang terstruktur.

\$movie: Sebuah array multidimensional yang menyimpan data film. Setiap elemen dari array adalah sebuah array yang berisi tiga nilai:

- Judul film
- Tahun rilis
- Rating film

Setiap film diambil dari array \$movie menggunakan indeks. Misalnya, \$movie[0] mengambil array pertama (film pertama).

Fungsi

Banyak fungsi *build-in* dari php yang sering kita gunakan, seperti `print()`, `print_r()`, `unset()`, dll. Selain fungsi-fungsi tersebut, kita juga dapat membuat fungsi sendiri sesuai kebutuhan.

Fungsi adalah sekumpulan intruksi yang dibungkus dalam sebuah blok. Fungsi dapat digunakan ulang tanpa harus menulis ulang instruksi di dalamnya.

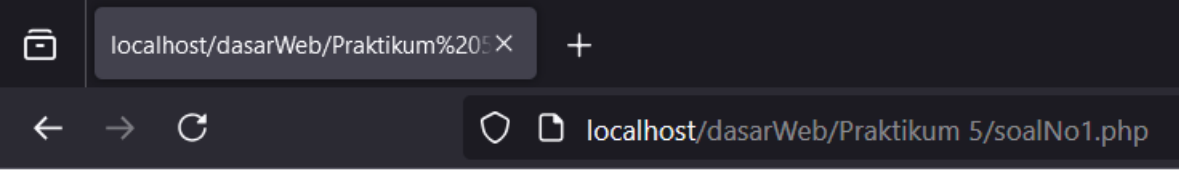
Fungsi pada PHP dapat dibuat dngan kata kunci *function*, lalu diikuti dengan nama fungsinya. Contoh:

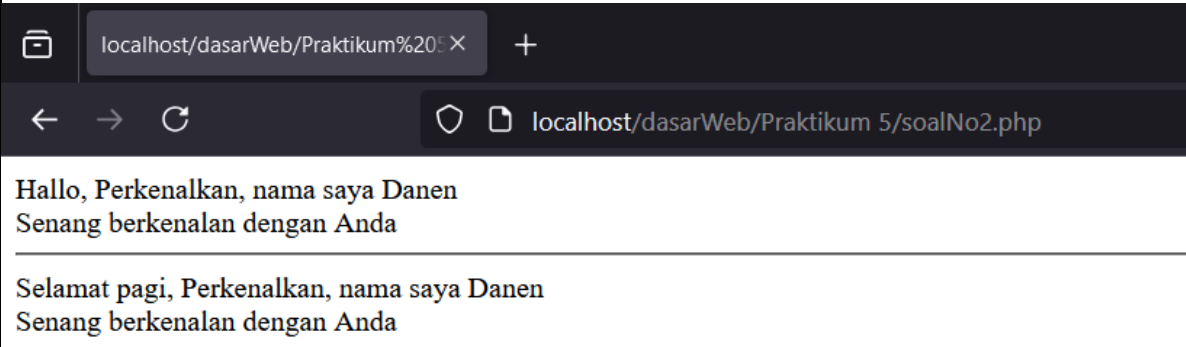
```
function namaFungsi(){
    //...
}
```

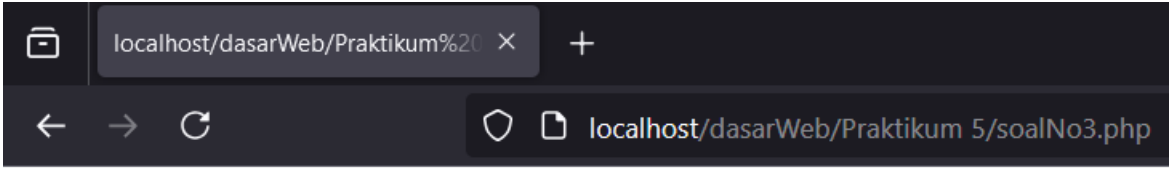
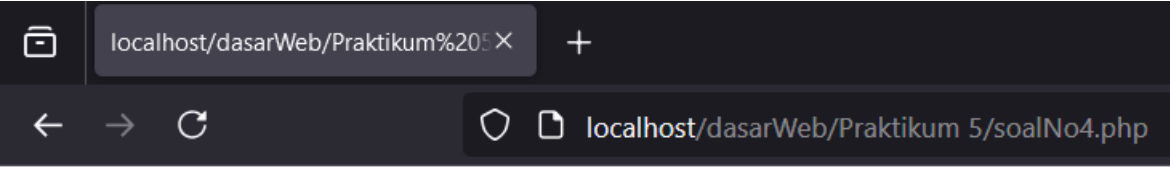
Kode instruksi dapat di tulis di dalam kurung kurawal ({...})

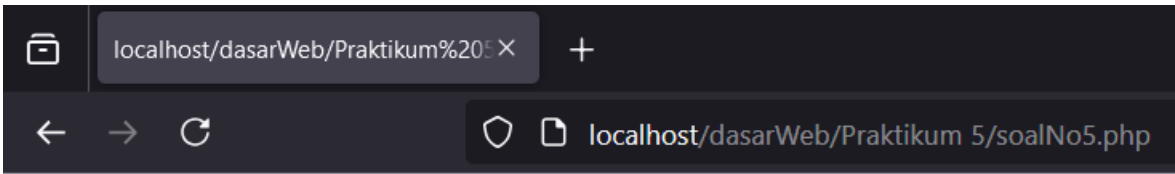
Penamaan fungsi di dalam PHP diawali dengan huruf atau tanda garis bawah dan tidak boleh diawali dengan angka. Penulisan nama fungsi ini tidak **case-sensitive**. Ikuti langkah-langkah berikut untuk memahami penggunaan fungsi di dalam PHP:

Langkah	Keterangan
---------	------------

1	<p>Buatlah satu file baru di dalam direktori praktik_php , beri nama fungsi.php</p> <pre data-bbox="284 163 1118 465"><?php function perkenalan(){ echo "Assalamualaikum, "; echo "Perkenalkan, nama saya Elok
"; //Tulis sesuai nama kalian echo "Senang berkenalan dengan Anda
"; } //memanggil fungsi yang sudah dibuat perkenalan(); ?></pre>
2	<p>Simpan file dan jalankan kode program dengan cetak sebanyak 2 kali, Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 4)</p>  <p>Assalamualaikum, Perkenalkan, nama saya Danen Senang berkenalan dengan Anda Assalamualaikum, Perkenalkan, nama saya Danen Senang berkenalan dengan Anda</p> <p>Membuat Fungsi Perkenalan, kemudian memanggil fungsi perkenalan sebanyak 2 kali.</p>
Fungsi dengan Parameter	
3	<p>Supaya instruksi yang di dalam fungsi lebih dinamis, kita dapat menggunakan parameter untuk memasukkan sebuah nilai ke dalam fungsi. Nilai tersebut akan diolah di dalam fungsi. Misalkan, pada contoh fungsi yang tadi, tidak mungkin nama yang dicetak adalah <i>elok</i> saja dan salam yang dipakai tidak selalu <i>assalamualaikum</i>.</p>
4	<p>Tambahkan parameter seperti pada kode program berikut ini:</p> <pre data-bbox="284 1417 863 1888"><?php //membuat fungsi function perkenalan(\$nama, \$salam){ echo \$salam.", "; echo "Perkenalkan, nama saya ".\$nama."
"; echo "Senang berkenalan dengan Anda
"; } //memanggil fungsi yang sudah dibuat perkenalan("Hamdana","Hallo"); echo "<hr>"; \$saya = "Elok"; \$ucapanSalam = "Selamat pagi"; //memanggil lagi perkenalan(\$saya,\$ucapanSalam); ?></pre>

5	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 5)</p>  <p>Hallo, Perkenalkan, nama saya Danen Senang berkenalan dengan Anda</p> <hr/> <p>Selamat pagi, Perkenalkan, nama saya Danen Senang berkenalan dengan Anda</p> <p>Membuat fungsi dengan menggunakan parameter untuk memasukkan sebuah nilai ke dalam fungsi. Nilai tersebut akan diolah di dalam fungsi. kemudian fungsi akan dipanggil untuk memunculkan nilai dalam parameter.</p>
Parameter dengan Nilai Default	
6	<p>Nilai <i>default</i> dapat kita berikan di parameter. Nilai <i>default</i> berfungsi untuk mengisi nilai sebuah parameter, kalau parameter tersebut tidak diisi nilainya. Misalnya: lupa mengisi parameter salam, maka program akan <i>error</i>. Oleh karena itu, kita perlu memberikan nilai <i>default</i> supaya tidak error.</p>
7	<p>Ketikkan kode program berikut</p> <pre data-bbox="292 1167 938 1624"><?php //membuat fungsi function perkenalan(\$nama, \$salam="Assalamualaikum"){ echo \$salam.", "; echo "Perkenalkan, nama saya ".\$nama."
"; echo "Senang berkenalan dengan Anda
"; } //memanggil fungsi yang sudah dibuat perkenalan("Hamdana","Hallo"); echo "<hr>"; \$saya = "Elok"; \$ucapanSalam = "Selamat pagi"; //memanggil lagi tanpa mengisi parameter salam perkenalan(\$saya); ?></pre>
8	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 6)</p>

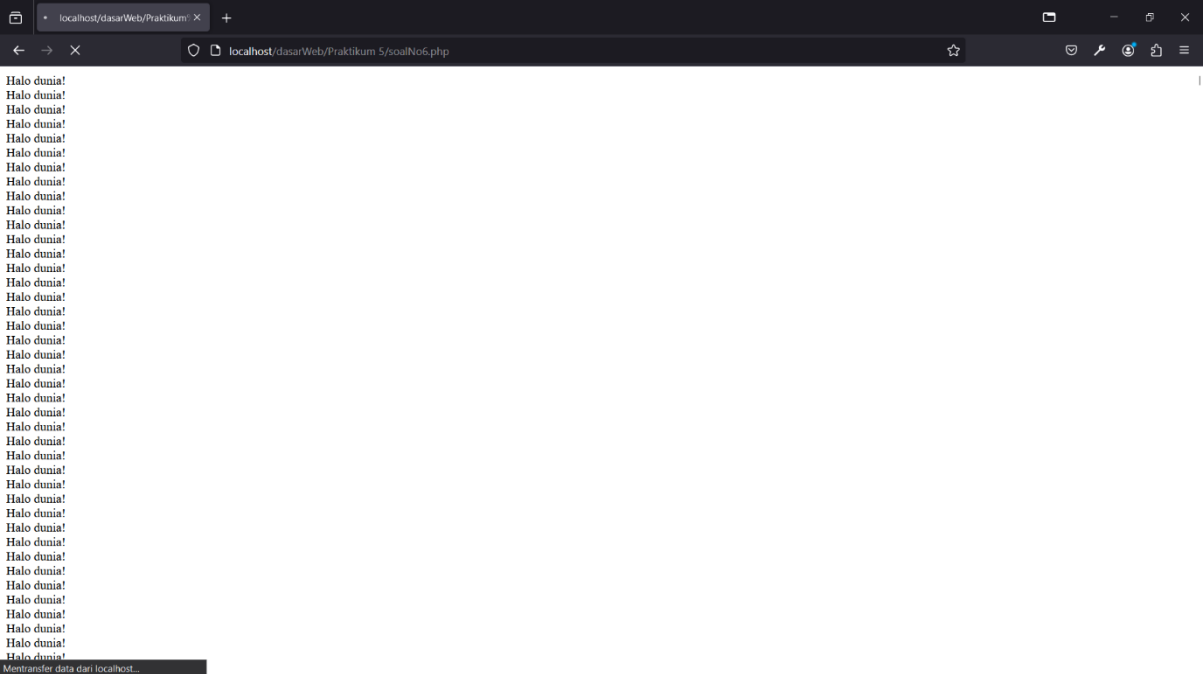
	 <p>Hallo, Perkenalkan, nama saya Danen Senang berkenalan dengan Anda</p> <hr/> <p>Assalamualaikum, Perkenalkan, nama saya Danendra AP Senang berkenalan dengan Anda</p> <p>Memanggil Fungsi menggunakan parameter tetapi dengan menggunakan Nilai Default. Nilai default berfungsi untuk mengisi nilai sebuah parameter, kalau parameter tersebut tidak diisi nilainya. Contoh diatas adalah memanggil fungsi tanpa mengisi parameter salam.</p>
Fungsi yang Mengembalikan Nilai	
9	<p>Hasil pengolahan nilai dari fungsi mungkin saja kita butuhkan untuk pemrosesan berikutnya. Oleh karena itu, kita harus membuat fungsi yang dapat mengembalikan nilai.</p> <p>Pengembalian nilai dalam fungsi dapat menggunakan kata kunci return.</p>
10	<p>Ketikkan kode program berikut</p> <pre data-bbox="284 1176 1029 1456"><?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 ?></pre>
11	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 7)</p>  <p>Umur saya adalah 20tahun</p> <p>Membuat perhitungan umur dengan fungsi Return. Fungsi return adalah fungsi yang dapat mengembalikan nilai.</p>

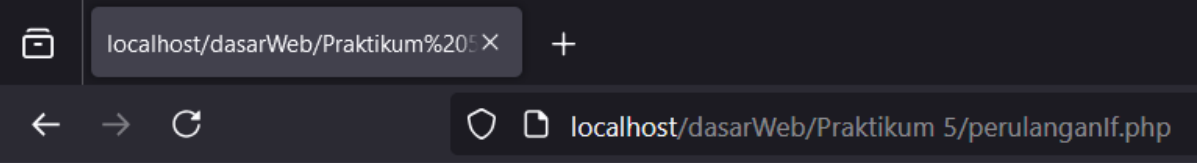
Memanggil Fungsi di dalam fungsi	
12	Fungsi yang sudah kita buat, dapat juga dipanggil di dalam fungsi lain.
13	<p>Ketikkan kode program berikut</p> <pre> <?php //membuat fungsi function hitungUmur(\$thn_lahir, \$thn_sekarang){ \$umur = \$thn_sekarang - \$thn_lahir; return \$umur; } function perkenalan (\$nama, \$salam="Assalamualaikum") { echo \$salam.", "; echo "Perkenalkan, nama saya ".\$nama."
"; //memanggil fungsi lain echo "Saya berusia ". hitungUmur(1988, 2023) ." tahun
"; echo "Senang berkenalan dengan anda
"; } //memanggil fungsi perkenalan perkenalan ("Elok"); ?> </pre>
14	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 8)</p>  <p>Saya berusia 20tahun Senang berkenalan dengan Anda Assalamualaikum, Perkenalkan, nama saya Danen</p> <p>Memanggil Fungsi di dalam fungsi, Fungsi yang sudah dibuat dapat dipanggil di fungsi lain, contoh fungsi hitungUmur dan perkenalan.</p>

Fungsi Rekursif

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Fungsi ini biasanya digunakan untuk menyelesaikan masalah seperti faktorial, bilangan fibonacci, pemrograman dinamis

Langkah	Keterangan
---------	------------

1	<p>Buat file baru dengan nama <code>rekursif.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre data-bbox="284 219 841 465"><?php function tampilkanHaloDunia(){ echo "Halo dunia!
"; tampilkanHaloDunia(); } tampilkanHaloDunia(); ?></pre>
2	<p>Jika kode program di atas dijalankan, apa yang akan terjadi dan bagaimana dampaknya jika itu di lakukan kemukakan pendapat kalian! (soal no 9)</p>  <p>Membuat fungsi <code>tampilkanHaloDunia</code> tetapi dengan Rekursif, maka fungsi ini akan memanggil Dirinya sendiri.</p>
3	<p>Untuk menampilkan angka 1 sampai 25, kita bisa dengan mudah menggunakan perulangan <code>for</code> seperti berikut:</p> <pre data-bbox="284 1592 829 1780"><?php for (\$i=1; \$i <=25; \$i++){ echo "Perulangan ke-{\$i}
"; } ?></pre>

4	<p>Akan tetapi jika kita ingin menggunakan konsep fungsi rekursif untuk menjalankan tugas yang sama, ketikkan kode program berikut.</p> <pre><?php function tampilkanAngka (int \$jumlah, int \$indeks = 1) { echo "Perulangan ke-{\$indeks}
"; //panggil diri sendiri selama \$indeks <= \$jumlah if (\$indeks < \$jumlah) { tampilkanAngka(\$jumlah, \$indeks + 1); } } tampilkanAngka(20); ?></pre>
5	<p>Jalankan kode program di atas dan bagaimana outputnya kemudian jelaskan kenapa bisa seperti itu. (soal no 10)</p>  <p>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</p> <p>Membuat Perulangan secara rekursif dengan fungsi dan if sebanyak 20 kali.</p>

Contoh Kasus Menu Bertingkat

Langkah	Keterangan
---------	------------

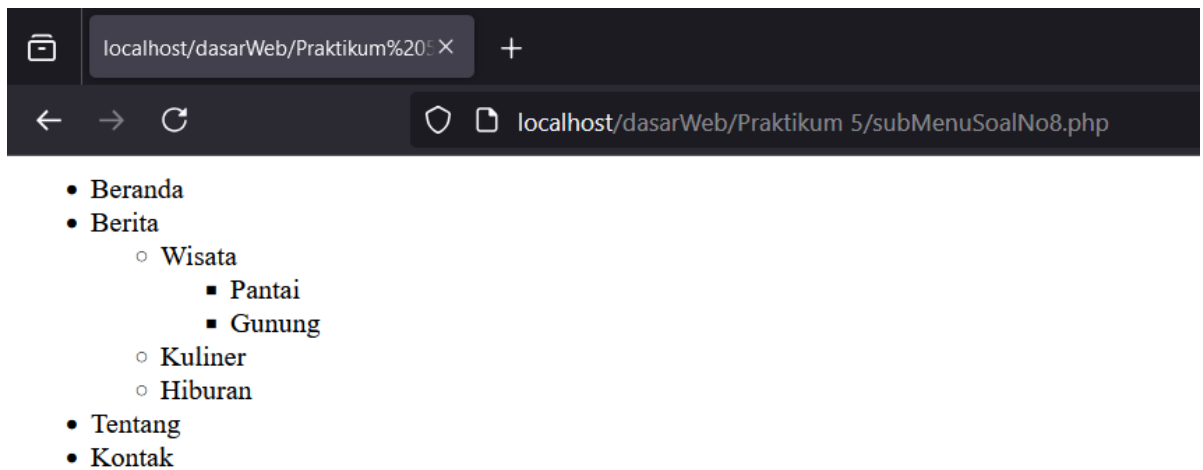
1	<p>Buat variabel \$menu. Variable ini adalah gabungan antara <i>array</i> terindeks dan <i>array</i> asosiatif multidimensi. Dikatakan multidimensi karena ia adalah suatu <i>array</i> yang memiliki array lain di dalamnya.</p> <p>Selanjutnya kita akan coba menampilkan semua item dari <i>array</i> \$menu menggunakan fungsi rekursif.</p>
2	<p>Buatlah kode program untuk variabel \$menu berikut</p> <pre><?php \$menu = [["nama" => "Beranda"], ["nama" => "Berita", "subMenu" => [["nama" => "Wisata", "subMenu" => [["nama" => "Pantai"], ["nama" => "Gunung"]]]], "nama" => "Kuliner"], ["nama" => "Hiburan"]], ["nama" => "Tentang"], ["nama" => "Kontak"],];</pre>
3	<p>Kemudian buatlah fungsi untuk menampilkan <i>array</i> utama</p>
	<pre>function tampilkanMenuBertingkat (array \$menu) { echo ""; foreach (\$menu as \$key => \$item) { echo "{\$item['nama']}"; } echo ""; } tampilkanMenuBertingkat(\$menu); ?></pre>
4	<p>Jalankan program diatas dan bagaimana hasil outputnya (soal no 11)</p>  <ul style="list-style-type: none"> • Beranda • Berita • Tentang • Kontak

5

Selanjutnya buatlah fungsi di atas menjadi rekursif dengan memanggil dirinya sendiri ketika suatu item dari menu memiliki atribut subMenu. Sehingga tampilanya menjadi seperti berikut. (soal no 12)

- Beranda
- Berita
 - Wisata
 - Pantai
 - Gunung
 - Kuliner
 - Hiburan
- Tentang
- Kontak

Hasil:



String

String tipe data karakter yang biasanya diekspresikan dengan diapit oleh tanda petik ganda (“ ”) atau petik tunggal (‘ ’). Keduanya hampir sama akan tetapi cara kerjanya sedikit berbeda. Tanda petik dua akan mengisi variabel dengan nilai aslinya, ada pun dengan tanda petik satu, kita tidak bisa melakukan hal tersebut.

Contoh strings adalah “Hello world!”. Beberapa operasi dapat dilakukan pada data bertipe string. PHP menyediakan fungsi-fungsi *built-in* yang siap untuk digunakan dalam operasi string, yaitu:

Fungsi	Keterangan
<code>strlen()</code>	Untuk mengetahui panjang string
<code>str_word_count()</code>	Untuk mengetahui jumlah kata di dalam string
<code>strpos()</code>	Untuk mengetahui posisi suatu string dalam string
<code>strrev()</code>	Untuk membalik urutan string
<code>strstr()</code>	Untuk mencari substring suatu string
<code>substr()</code>	Untuk mengambil substring dari posisi awal dan akhir dalam sebuah string
<code>trim()</code>	Menghilangkan karakter spasi di awal dan akhir string

ltrim()	Menghilangkan karakter spasi di awal string
rtrim()	Menghilangkan karakter spasi di akhir string
strtoupper()	Mengubah huruf menjadi huruf capital
strtolower()	Mengubah huruf menjadi huruf kecil (<i>lowercase</i>)
str_replace()	Menggantikan nilai dari beberapa bagian dari string dengan string yang lain
ucwords()	Mengubah huruf awal dari sebuah kata dengan huruf besar
explode()	Memecah sebuah kalimat berdasarkan sebuah karakter dan mengubah menjadi array

Escape Character

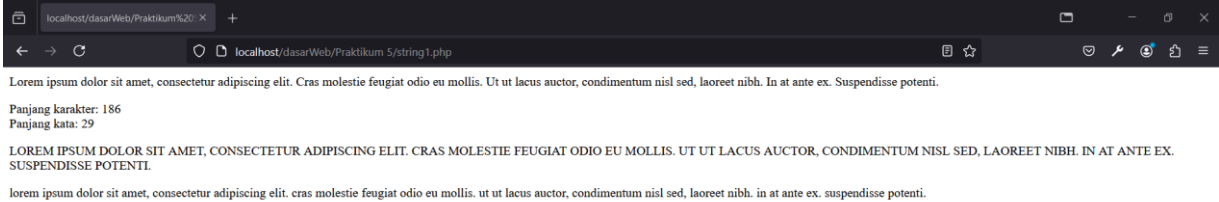
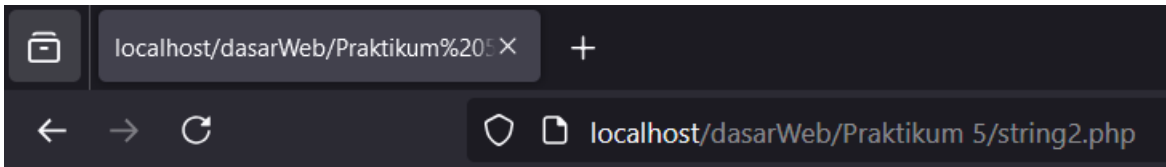
Karakter-karakter khusus yang tidak bisa ditampilkan secara langsung, melainkan harus diikuti dengan tanda \. String yang dirangkai dengan tanda petik dua akan mengganti *escape character* dengan karakter yang merepresentasikannya. Hal ini berbeda dengan string yang dirangkai dengan tanda petik satu. Yang mana ia hanya akan menampilkan apa adanya tanpa mereplace apa pun dengan apa pun (kecuali sedikit kasus).

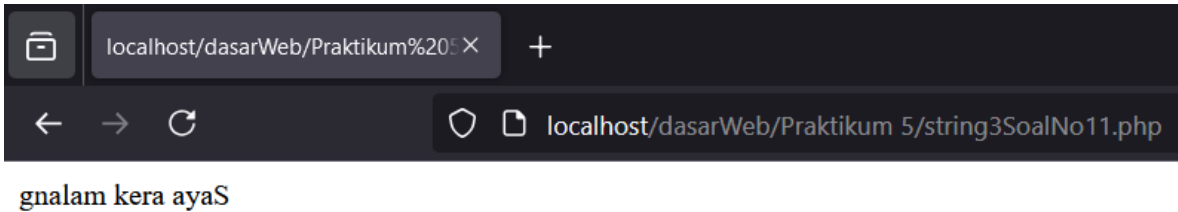
Ada pun *escape character* pada PHP adalah:

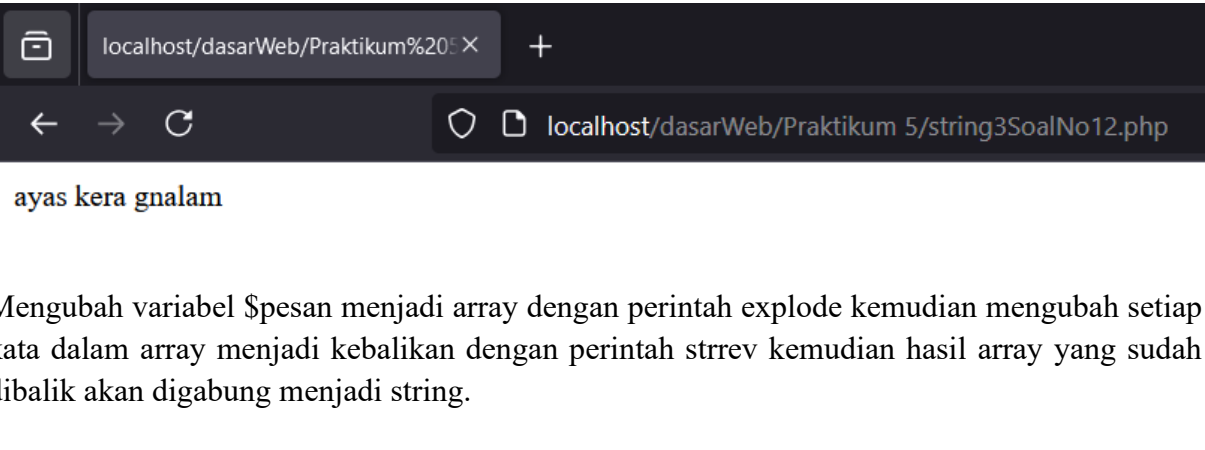
Fungsi	Keterangan
\n	Baris baru
\r	Karakter carriage-return
\t	Karakter tab
\\$	Karakter \$ itu sendiri
\"	Untuk menampilkan tanda petik dua
\\	Untuk menampilkan tanda slash \ itu sendiri

Ikuti langkah-langkah praktikum berikut ini:

Langkah	Keterangan
1	<p>Buat file <code>string1.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre><?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 "<p>{\$loremIpsum}</p>"; echo "Panjang karakter: " . strlen(\$loremIpsum) . "
"; echo "Panjang kata: " . str_word_count(\$loremIpsum) . "
"; echo "<p>" . strtoupper(\$loremIpsum) . "</p>"; echo "<p>" . strtolower(\$loremIpsum) . "</p>"; ?></pre>
2	Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 13)

	 <p>1. Menampilkan lorem dengan menggunakan fungsi</p> <p>2. Menghitung panjang karakter/string dengan strlen</p> <p>3. Menghitung panjang kata di dalam string dengan str_word_count</p> <p>4. Mengubah huruf menjadi huruf kapital dengan strtoupper</p> <p>5. Mengubah huruf menjadi huruf kecil dengan strtolower</p>
Escape Character	
3	<p>Buat file <code>string2.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre data-bbox="288 815 944 1115"><?php echo "Baris\nbaru
"; //soal 10.a echo 'Baris\nbaru
'; //soal 10.b echo "Halo\rDunia
"; //soal 10.c echo 'Halo\rDunia
'; //soal 10.d echo "<pre>Halo\tDunia!</pre>"; //soal 10.e echo '<pre>Halo\tDunia!</pre>'; //soal 10.f echo "Katakanlah \"Tidak pada narkoba!\"
"; //soal 10.g echo 'Katakanlah \'Tidak pada narkoba!\'
'; //soal 10.h ?></pre>
4	<p>Dari kode program di atas, kalian bisa mengetahui perbedaan antara tanda petik dua dan tanda petik satu dari segi cara kerjanya menangani <i>escape string</i>. Amati hasil yang ditampilkan dan jelaskan hasil dari masing-masing outputnya dan apa yang dapat kalian simpulkan dari soal percobaan tersebut (soal no 14)</p>  <p>Baris baru Baris\nbaru Halo Dunia Halo\rDunia Halo Dunia! Halo\tDunia! Katakanlah "Tidak pada narkoba!" Katakanlah 'Tidak pada narkoba!'</p> <p>perbedaan antara tanda petik dua (") dan tanda petik satu (') terlihat jelas dalam penanganan escape string.</p>

	<ol style="list-style-type: none"> 1. <code>echo "Baris\nbaru
";</code> - Menampilkan "Baris" diikuti oleh newline dan "baru", dengan <code>
</code> untuk line break. 2. <code>echo 'Baris\nbaru
';</code> - Menampilkan "Baris\nbaru
" secara literal, tanpa interpretasi escape sequence. 3. <code>echo "Halo\rDunia
";</code> - Menampilkan "Dunia" di atas "Halo" karena <code>\r</code> membawa kursor kembali ke awal baris. 4. <code>echo 'Halo\rDunia
';</code> - Menampilkan "Halo\rDunia
" secara literal. 5. <code><pre></code> tags memformat teks, sehingga <code>\t</code> (tab) ditampilkan sebagai tab pada output. 6. Tanda kutip dalam string diizinkan menggunakan escape karakter (<code>\</code> dan <code>\'</code>), memungkinkan kita untuk menyisipkan tanda kutip tanpa konflik. <p>Kesimpulannya, tanda petik dua menginterpretasi escape sequences, sementara tanda petik satu menampilkan string secara literal.</p>
Membalik String menggunakan perintah <code>strrev()</code> .	
5	<p>Buat file <code>string3.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre><?php \$pesan = "Saya arek malang"; echo strrev(\$pesan) . "
"; ?></pre>
6	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 15)</p>  <p>Menggunakan string <code>strrev</code> untuk membalik urutan string atau membalik kata.</p>
8	<p>untuk membalik string per kata, ketikkan kode program berikut:</p> <pre><?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) => strrev(\$pesan), \$pesanPerKata); # gabungkan kembali array menjadi string \$pesan = implode(" ", \$pesanPerKata); echo \$pesan . "
"; ?></pre>
8	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 16)</p>

	 <p>ayas kera gnam</p> <p>Mengubah variabel \$pesan menjadi array dengan perintah explode kemudian mengubah setiap kata dalam array menjadi kebalikan dengan perintah strrev kemudian hasil array yang sudah dibalik akan digabung menjadi string.</p>
--	--

Menggabungkan HTML dan PHP

Ada dua cara menggabungkan HTML dan PHP yaitu PHP yang berada didalam HTML, dan HTML yang ada di dalam PHP.

Langkah	Keterangan
1	Cara pertama adalah php di dalam HTML. Seperti kode berikut.
	<pre><html> <head> <title>Cara 01</title> </head> <body> <p>Tanggal Hari ini : <?php echo date("d M Y")?></p> </body> </html></pre>
2	Pada kode diatas merupakan kode html yang berisi kode php untuk menampilkan tanggal server yang ditandai dengan tag <?php dan ?> .
3	<p>Cara kedua adalah HTML di dalam PHP. Di dalam PHP Tag HTML diperlakukan sebagai <i>string</i> yang di apit dengan tanda petik dan bisa dilakukan berbagai fungsi untuk memanipulasi <i>string</i> seperti menyambung dll. Contoh kode seperti pada potongan kode berikut.</p> <pre><?php echo '<html>'; echo '<head><title>Cara02</title></head>'; echo '<body>'; echo '<p>Tanggal Hari ini : '.date('d M Y').'</p>'; echo '</body>'; echo '</html>'; ?></pre>
4	Kode diatas mengeluarkan output sama dengan potongan kode sebelumnya. Namun yang berbeda adalah penulisan kode dimana HTML berada di dalam PHP sebagai string dan untuk menampilkannya menggunakan <i>tag echo</i> .
5	<p>Dari dua cara tersebut mana yang lebih mudah menurut kalian, kemukakan jawaban disertai dengan alasan (soal no 17)</p> <p>Menurut saya, PHP di dalam HTML lebih mudah. Karena, dengan menggabungkan PHP di dalam HTML, Kode Program lebih rapi dan terstruktur, sehingga, tidak perlu echo untuk setiap tag HTML.</p>

Entities HTML

Ketika membuat konten website, tidak akan terlepas dari menggunakan karakter khusus seperti simbol copyright (©), ampersand (&), lebih kecil (<), lebih besar (>), titik koma (;) dan sebagainya. Untuk menampilkan simbol-simbol tersebut di HTML perlu menggunakan entitas karakter (*character entities*). Entitas Karakter mempunyai 3 bagian yaitu: sebuah *ampersand* (&), nama dan nomor entitas (#), dan titik koma/semicolon (;).

Sebagai contoh, untuk menampilkan simbol copyright (©) ➤ © atau ©

Berikut ini adalah tabel beberapa entitas yang umum digunakan:

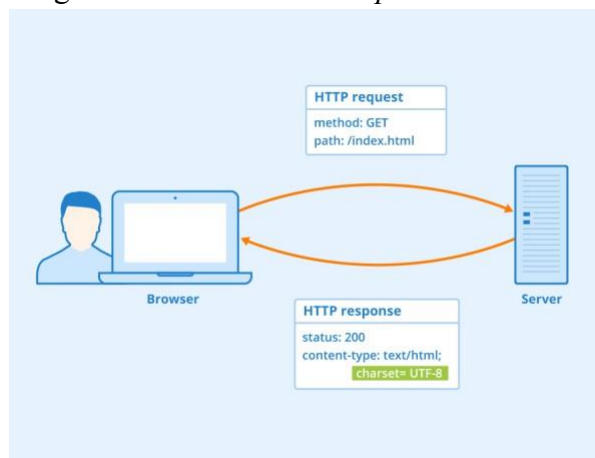
Nama Entitas	Nomor Entitas	Keterangan	Hasil
©	©	Copyright	©
®	®	Registered	®
-	™	Trademark	™
 	¡	Non break space	
&	&	Ampersand	&
«	«	Angle question left	«
»	»	Angle question right	»
"	"	Tanda kutip dua	"
'	-	Tanda kutip satu	'
<	<	Lebih kecil	<
>	>	Lebih besar	>
×	×	Tanda kali	×
÷	÷	Tanda bagi	÷

Langkah	Keterangan
---------	------------

HTTP Header

HTTP *header* adalah sebuah data yang dikirim antara web browser dengan web server sebagai sarana komunikasi antar keduanya. Di dalam HTTP header terdapat informasi tentang bagaimana cara menangani file yang dikirim/diminta.

Siklus *request respond* untuk halaman web: Ketika kita mengakses suatu halaman web, web browser secara otomatis mengirim sebuah HTTP *request* kepada web server. HTTP *request* berisi banyak informasi, salah satunya adalah HTTP *header*. Di dalam HTTP *header* (yang dikirim pada saat proses *request*) terdapat informasi tentang file apa yang diminta (apakah file HTML, file PHP, file PDF, atau yang lain), serta berbagai info tambahan seperti jenis web browser yang dipakai, sistem operasi, dan alamat IP. Setelah sampai di web server, informasi yang ada di HTTP *header* akan dibaca dan web server mempersiapkan file-file yang diminta. Setelah itu, web server mengirim kembali file-file tersebut kepada web browser. Proses pengembalian ini dikenal juga dengan sebutan HTTP *Response*.



HTTP Header

HTTP *response* ini terdiri dari 2 bagian: HTTP header dan file web. HTTP *header* berisi informasi mengenai file web yang dikirim, seperti tipe data, tanggal dikirim, nama web server, dan sistem operasi yang digunakan oleh web server. Sedangkan file web sendiri terdiri dari file HTML yang menyusun halaman web, termasuk juga file gambar (jika ada).

Sebagai analogi, jika presiden ingin berkunjung ke rumah kalian, tentu akan ada tim paspampres yang datang lebih awal. Mereka menginformasikan bahwa presiden akan datang pada jam sekian, dengan membawa sekian orang, dan informasi-informasi lain. Tim paspampres ini bisa disamakan dengan HTTP header yang datang lebih dahulu sebelum file asli dikirim. Dalam prakteknya, kita tidak menyadari adanya HTTP header, bahkan mungkin tidak pernah mendengar istilah ini. Dan itu tidak salah, karena isi dari HTTP header ditujukan kepada pemrosesan di web browser, bukan untuk pengunjungan web.

Bagaimana cara melihat http header pada web browser kalian jelaskan disertai Langkah-langkahnya (Soal no 19)

Menggunakan fitur Developer Tools yang tersedia di browser seperti Google Chrome, Firefox, atau Microsoft Edge. Untuk langkah – langkah sebagai berikut:

☐ Buka Developer Tools:

- Klik kanan di halaman web mana saja, lalu pilih "**Inspect**" atau tekan tombol keyboard **F12**.

□ Navigasi ke tab "Network":

- Setelah Developer Tools terbuka, pilih tab "**Network**" di bagian atas panel Developer Tools.

□ Muat Ulang Halaman:

- Jika halaman sudah terbuka, muat ulang halaman dengan menekan tombol **F5** atau klik ikon **refresh** di browser.
- Ini akan memungkinkan browser menangkap semua request jaringan termasuk HTTP Headers.

□ Pilih Request:

- Di bagian **Network**, kamu akan melihat daftar request yang dibuat oleh browser untuk memuat halaman web.
- Klik salah satu request (biasanya yang pertama adalah request utama untuk halaman HTML).

□ Lihat HTTP Headers:

- Di panel sebelah kanan, klik tab "**Headers**".
- Di sana kamu bisa melihat **Request Headers** (yang dikirim oleh browser) dan **Response Headers** (yang diterima dari server).

Date and Time

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

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

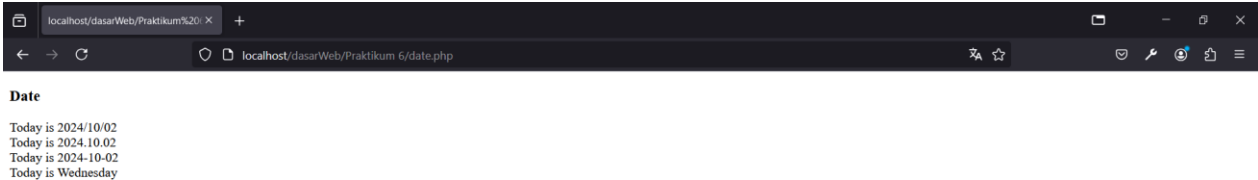
Parameter format bersifat wajib (*required*). Parameter format digunakan untuk menentukan bagaimana format tanggal dan/atau waktu yang akan digunakan. Berikut beberapa contoh karakter yang umum digunakan untuk format tanggal:


1. `d`, merepresentasikan hari (01 sampai 31)
2. `m`, merepresentasikan bulan (01 sampai 12)
3. `Y`, merepresentasikan tahun (dalam 4 digit)
4. `l`, merepresentasikan hari dalam 1 minggu

Selain menampilkan tanggal, fungsi tersebut juga menampilkan waktu. Berikut adalah beberapa karakter yang umum digunakan untuk format waktu:

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

Ikuti langkah-langkah berikut untuk memahami bagaimana penggunaan fungsi `date()`

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>date.php</code> di dalam direktori <code>dasarWeb</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <h3> Date </h3> 7 <?php 8 echo "Today is " . date("Y/m/d") . "
"; 9 echo "Today is " . date("Y.m.d") . "
"; 10 echo "Today is " . date("Y-m-d") . "
"; 11 echo "Today is " . date("l"); 12 ?> 13 </body> 14 </html> </pre>
2	Simpan file dan jalankan kode program
3	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 19)</p>  <p>Menggunakan kode PHP untuk menampilkan tanggal saat ini dalam berbagai format pada halaman HTML.</p> <p>Parameter format bersifat wajib (required). Parameter format digunakan untuk menentukan bagaimana format tanggal dan/atau waktu yang akan digunakan.</p>

4	<p>Buat file baru dengan nama <code>time.php</code> di dalam direktori <code>dasarWeb</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 </head> 5 <body> 6 <h3> Time </h3> 7 <?php 8 date_default_timezone_set("asia/jakarta"); 9 echo date("h:i:sa"); 10 ?> 11 </body> 12 </html> </pre>
5	Simpan file dan jalankan kode program
6	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 20)</p>  <p>Menggunakan kombinasi antara HTML dan PHP untuk menampilkan waktu saat ini dalam format 12-jam, dengan zona waktu yang diatur ke Asia/Jakarta.</p>

Variabel Superglobal

Variabel super global di PHP adalah variabel bawaan yang bersifat global. Variabel bawaan yang dimaksud adalah: variabel yang sudah otomatis ada tanpa perlu kita definisikan sendiri. Dan ia bersifat global dalam artian bisa kita akses dari mana pun dan kapan pun.

Variabel super global menyimpan banyak sekali data penting dan juga bermanfaat yang bisa kita gunakan dalam menyelesaikan proyek yang sedang kita kerjakan. Terdapat **9 variabel** super global di PHP.

1. Variabel \$_SERVER

Variabel yang pertama dan utama adalah variabel \$_SERVER. Ia adalah sebuah array asosiatif yang menyediakan berbagai macam informasi tentang request yang ditangkap oleh server. Data yang dimuat berupa *headers*, *paths*, lokasi skrip, dan sebagainya.

Nilai yang tersimpan pada variabel \$_SERVER disediakan oleh web server, oleh karena itu tidak ada jaminan khusus bahwa setiap web server yang kita gunakan akan memberikan semua data-data standar yang ada.

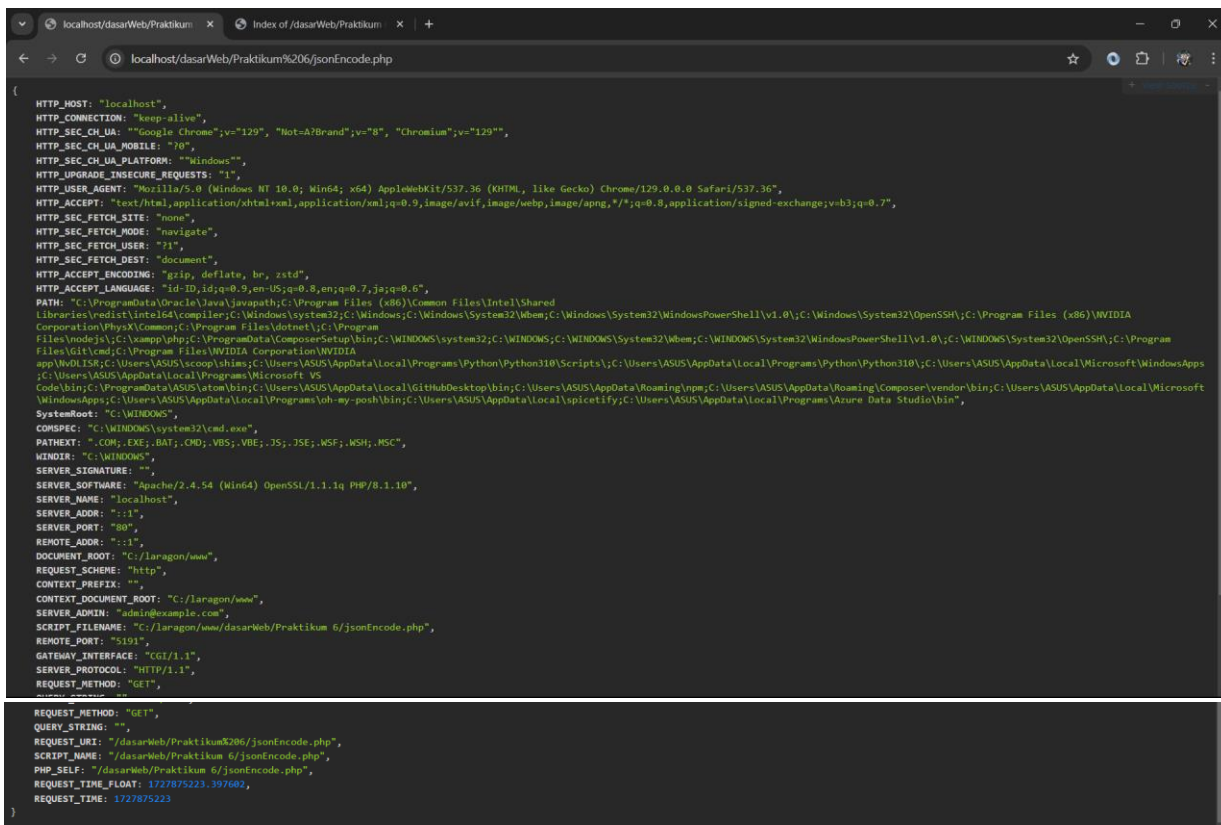
Untuk bisa mengetahui nilai apa saja yang tersedia pada variabel \$_SERVER, kita bisa mengeksekusi perintah berikut:

```
<?php
echo
    json_encode($_SERVER);
```

Bagaimana output yang dapatkan (Soal no 21) sebelum di jalan silakan menginstall ekstensi JSON View pada link berikut ini

<https://chrome.google.com/webstore/detail/jsonvue/chklaanhfefbnpoihckbnefhakgolnmc?hl=id>

Output:



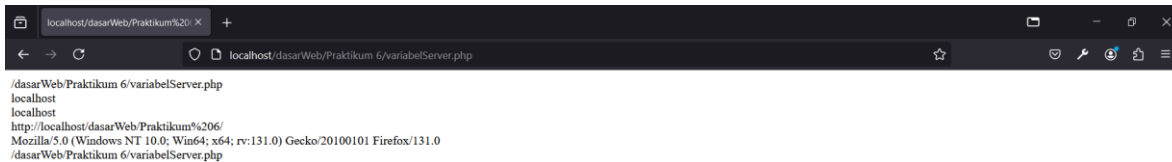
```
{
  "HTTP_HOST": "localhost",
  "HTTP_CONNECTION": "keep-alive",
  "HTTP_SEC_CH_UA": "\"Google Chrome\";v=\"129\", \"Not-A.Brand\";v=\"8\", \"Chromium\";v=\"129\"",
  "HTTP_SEC_CH_UA_MOBILE": "\"0\"",
  "HTTP_SEC_CH_UA_PLATFORM": "\"Windows\"",
  "HTTP_UPGRADE_INSECURE_REQUESTS": "\"1\"",
  "HTTP_USER_AGENT": "\"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 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": "\"1\"",
  "HTTP_SEC_FETCH_DEST": "\"document\"",
  "HTTP_ACCEPT_ENCODING": "\"gzip, deflate, br, zstd\"",
  "HTTP_ACCEPT_LANGUAGE": "\"id-ID;id;q=0.9,en-US;q=0.8,en;q=0.7,jav;q=0.6\"",
  "PATH": "\"C:\\ProgramData\\Oracle\\Java\\javapath;C:\\Program Files (x86)\\Common-Files\\Intel\\SharedLibraries\\vcredist\\intel64\\compiler;C:\\Windows\\system32;C:\\Windows;C:\\Windows\\System32\\Wbem;C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\;C:\\Windows\\System32\\OpenSSH;C:\\Program Files (x86)\\NVIDIA Corporation\\PhysX\\Common;C:\\Program Files\\dotnet\\;C:\\Program Files\\Node.js\\;C:\\xampp\\php;C:\\ProgramData\\ComposerSetup\\bin;C:\\WINDOWS\\system32;C:\\WINDOWS;C:\\WINDOWS\\System32\\Wbem;C:\\WINDOWS\\System32\\WindowsPowerShell\\v1.0\\;C:\\WINDOWS\\System32\\OpenSSH;C:\\Program Files\\Git\\cmd;C:\\Program Files\\NVIDIA Corporation\\NVidia app\\NvDLSS;C:\\Users\\ASUS\\AppData\\Local\\Programs\\Python\\Python310\\Scripts\\;C:\\Users\\ASUS\\AppData\\Local\\Programs\\Python\\Python310\\;C:\\Users\\ASUS\\AppData\\Local\\Microsoft\\WindowsApps\\;C:\\Users\\ASUS\\AppData\\Local\\Programs\\Microsoft VS Code\\bin;C:\\ProgramData\\ASUS\\atom\\bin;C:\\Users\\ASUS\\AppData\\Local\\GitHubDesktop\\bin;C:\\Users\\ASUS\\AppData\\Roaming\\npm;C:\\Users\\ASUS\\AppData\\Roaming\\Composer\\vendor\\bin;C:\\Users\\ASUS\\AppData\\Local\\Microsoft\\WindowsApps;C:\\Users\\ASUS\\AppData\\Local\\Programs\\oh-my-posh\\bin;C:\\Users\\ASUS\\AppData\\Local\\Programs\\Azure Data Studio\\bin\"",
  "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/2.4.54 (Win64) OpenSSL/1.1.1q PHP/8.1.10\"",
  "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\\dasarWeb\\Praktikum 6\\jsonEncode.php\"",
  "REMOTE_PORT": "\"5191\"",
  "GATEWAY_INTERFACE": "\"CGI/1.1\"",
  "SERVER_PROTOCOL": "\"HTTP/1.1\"",
  "REQUEST_METHOD": "\"GET\"",
  "REQUEST_METHOD": "\"GET\"",
  "QUERY_STRING": "\"\"",
  "REQUEST_URI": "\"/dasarWeb/Praktikum6/jsonEncode.php\"",
  "SCRIPT_NAME": "\"/dasarWeb/Praktikum 6/jsonEncode.php\"",
  "PHP_SELF": "\"/dasarWeb/Praktikum 6/jsonEncode.php\"",
  "REQUEST_TIME_FLOAT": "1727875223.397602",
  "REQUEST_TIME": "1727875223"
}
```

```

1  <?php
2  echo $_SERVER['PHP_SELF'];
3  echo "<br>";
4  echo $_SERVER['SERVER_NAME'];
5  echo "<br>";
6  echo $_SERVER['HTTP_HOST'];
7  echo "<br>";
8  echo $_SERVER['HTTP_REFERER'];
9  echo "<br>";
10 echo $_SERVER['HTTP_USER_AGENT'];
11 echo "<br>";
12 echo $_SERVER['SCRIPT_NAME'];
13 ?>

```

Jalankan kode program diatas kemudian jelaskan output dari masing-masing perintah echonya (Soal no.22)



Penjelasan:

\$_SERVER['PHP_SELF']:

- **\$_SERVER['PHP_SELF']** mengembalikan path dari file PHP yang sedang dijalankan relatif terhadap root. berguna untuk mendapatkan URL dari halaman yang sedang dieksekusi.

2. \$_SERVER['SERVER_NAME']:

- **\$_SERVER['SERVER_NAME']** mengembalikan nama domain dari server di mana skrip ini dijalankan. Jika skrip diakses melalui **localhost**, maka nilai yang dikembalikan adalah **localhost**. Jika diakses melalui domain seperti **example.com**, outputnya adalah **example.com**.

3. \$_SERVER['HTTP_HOST']:

- **\$_SERVER['HTTP_HOST']** mengembalikan nilai host header dari permintaan HTTP, yang biasanya berisi nama domain atau alamat IP dari server. Ini hampir sama dengan **SERVER_NAME**, namun diambil dari header HTTP. Outputnya bisa berupa **localhost** atau **example.com**.

4. \$_SERVER['HTTP_REFERER']:

- \$_SERVER['HTTP_REFERER'] mengembalikan URL dari halaman yang mengarahkan pengguna ke halaman ini, atau lebih dikenal sebagai **referer**.
- Jika pengguna mengklik tautan dari halaman lain ke halaman ini, URL halaman asal akan ditampilkan.
- Namun, **HTTP referer** mungkin tidak selalu tersedia karena beberapa browser atau server bisa memblokirnya.

5. \$_SERVER['HTTP_USER_AGENT']:

- \$_SERVER['HTTP_USER_AGENT'] mengembalikan informasi terkait **browser** atau **client** yang digunakan oleh pengunjung untuk mengakses halaman.
- Ini berisi string yang menyatakan jenis browser, sistem operasi, dan informasi lainnya yang berguna untuk mengidentifikasi perangkat pengguna.
- Misalnya, output bisa berupa sesuatu seperti **Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36**.

6. \$_SERVER['SCRIPT_NAME']:

- \$_SERVER['SCRIPT_NAME'] mengembalikan path dari file PHP yang sedang dijalankan, mirip dengan \$_SERVER['PHP_SELF'].
- Output ini menunjukkan lokasi dari file saat ini relatif terhadap root server. Contohnya, **/public_html/index.php**.

Beberapa contoh data pada variabel \$_SERVER yang akan sering dibutuhkan:

No	Variabel dan Deskripsi
1	\$_SERVER['PHP_SELF'] file yang sedang dieksekusi, nama file diambil berdasarkan dari dokumen <i>root</i>
2	\$_SERVER['SERVER_ADDR'] ng sedang Alamat IP server dari file ya lieksekusi.
3	\$_SERVER['SERVER_NAME']
No	Variabel dan Deskripsi
	Hostname dari server dari file php yang sedang dieksekusi. Hostname biasanya adalah nama PC yang terlihat di dalam jaringan. Jika skrip PHP dijalankan di atas Virtual Host, maka nama virtual host tersebut akan dijadikan sebagai <i>server name</i>
4	\$_SERVER['SERVER_PROTOCOL'] ang sedang berjalan. Apakah HTTP atau HTTPS. Misal: Ini adalah protokol komunikasi y HTTP/0.1'
5	\$_SERVER['REQUEST_METHOD'] le PHP yang sedang Berisi jenis metode <i>request</i> dari fi dieksekusi. Misalkan: GET , POST , PUT , DELETE , OPTIONS
6	\$_SERVER['QUERY_STRING'] Akan mengembalikan query s tring dari file atau skrip PHP yang sedang dieksekusi. Misalkan user

	mengakses <code>http://localhost/halo-dunia?nama=Budi&umur=20&asal=Surabaya</code> , maka variabel tersebut akan mengembalikan nilai <code>nama=Budi&umur=20&asal=Surabaya</code> .
7	<code>\$_SERVER['DOCUMENT_ROOT']</code> Nama direktori root dari dokumen/file PHP yang sedang dieksekusi. Hasil yang dikembalikan berdasarkan dari pengaturan server.
8	<code>\$_SERVER['HTTP_HOST']</code> host, seperti misalnya: header (jika memang ada). Mengembalikan konten dari
9	<code>\$_SERVER['HTTP_REFERER']</code> Halaman url yang menjadi referensi ke halaman yang sedang dieksekusi. Jika tidak ada, nilainya kosong.
10	<code>\$_SERVER['HTTP_USER_AGENT']</code> user yang sedang melakukan <i>request</i> konten: mulai dari browser Variabel ini berisi informasi dari browser yang ontho nilai dari variabel ini adalah: Mozilla/4.5 [en] (X11; U; digunakan, bahasa, hingga OS. C Linux 2.2.9 i586
11	<code>\$_SERVER['REMOTE_ADDR']</code> yang sedang mengakses halaman web Berisi alamat IP dari user yang PHP
12	<code>\$_SERVER['SCRIPT_FILENAME']</code> yang sedang Nama path absolut dari file yang dieksekusi.
13	<code>\$_SERVER['REQUEST_URI']</code> dieksekusi. Misalkan: <code>/php/halo-dunia</code> . Alamat uri dari file yang sedang

2. Variabel \$_GET

Variabel `$_GET` adalah array asosiatif yang berisi nilai dari *query string*. Misalkan kita memiliki file `halodunia.php` sebagaimana berikut:

```
<?php

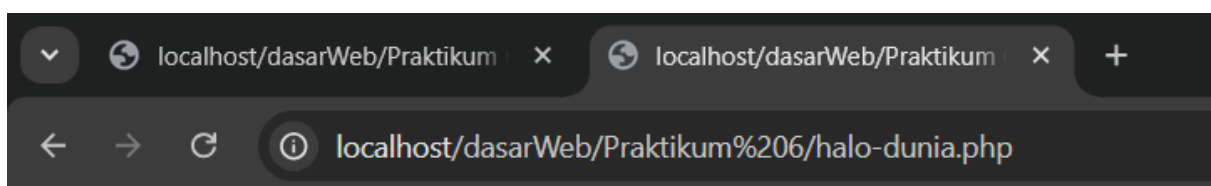
$nama = @$_GET['nama']; //tanda @ agar tidak ada peringatan error
ketika key-nya kosong
$usia = @$_GET['usia']; //tanda @ agar tidak ada peringatan error
ketika key-nya kosong

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

Jika kita mengakses file tersebut dengan *query string* dibawah ini, output apa yang dihasilkan, amati dan bagaimana hasilnya (soal no 23)

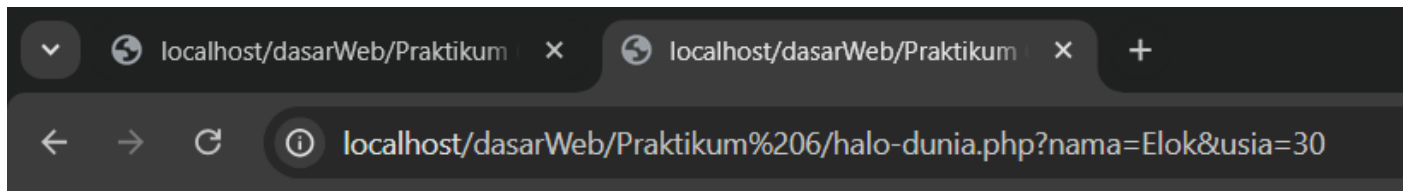
<http://localhost/halo-dunia.php?nama=Elok&usia=30>

Ketika belum di inputkan query string:



Halo ! Apakah benar anda berusia tahun?

Ketika sudah di input query string:



Halo Elok! Apakah benar anda berusia 30 tahun?

3. Variabel \$_POST

Variabel \$_POST mirip dengan variabel \$_GET. Hanya saja data yang di-passing tidaklah melalui query string pada URL, akan tetapi pada *body request*. Dan *request method* yang dilakukan haruslah dengan metode **POST**.

```
<html>
<body>

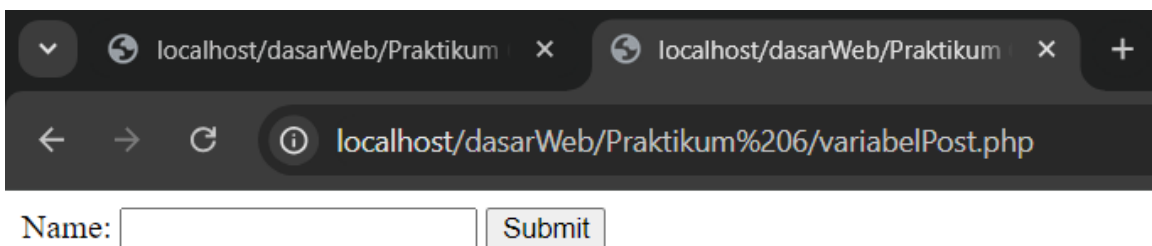
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

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

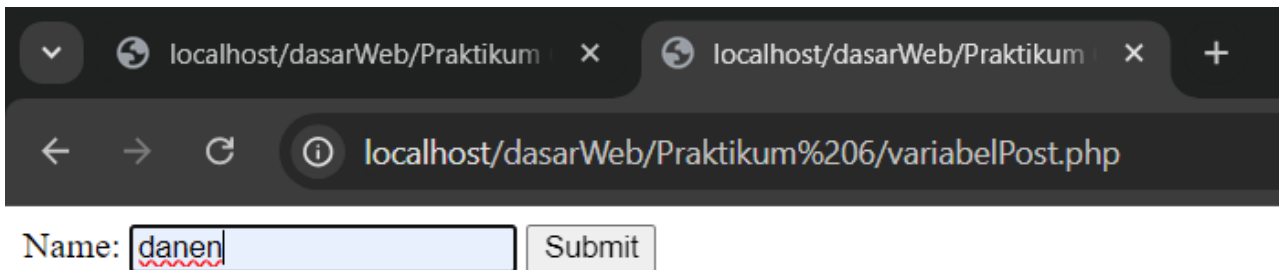
</body>
</html>
```

Jalankan kode program berikut ini. Apa yang bisa kalian simpulkan dari output yang dihasilkan (Soal no 24)

Tampilan Awal:

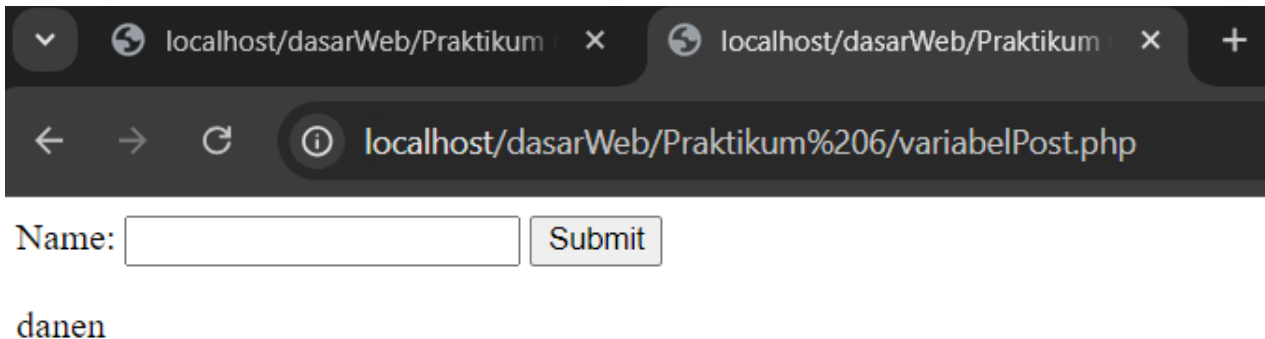


Setelah di isi dan di submit:



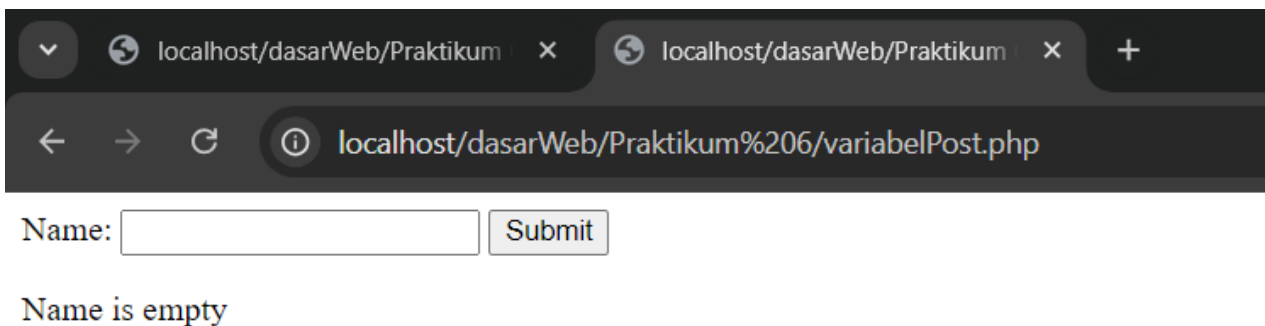
A screenshot of a web browser with two tabs. The active tab shows the URL `localhost/dasarWeb/Praktikum%206/variabelPost.php`. Below the address bar, there is a form with a label "Name:" followed by a text input field containing the text "danen". To the right of the input field is a "Submit" button.

Hasil:



A screenshot of a web browser showing the result of a successful form submission. The URL is the same as the previous image. Below the address bar, the "Name:" label is followed by an empty text input field. Below the input field, the text "danen" is displayed, indicating the value that was submitted.

Ketika Nama kosong kemudian submit:



A screenshot of a web browser showing the result of an empty form submission. The URL is the same. Below the address bar, the "Name:" label is followed by an empty text input field. Below the input field, the text "Name is empty" is displayed, indicating that the submitted name was empty.

Menggunakan kombinasi HTML dan PHP yang berfungsi untuk membuat form sederhana di mana pengguna dapat memasukkan nama, kemudian mengirimkannya ke server melalui metode POST.

❑ `if ($_SERVER["REQUEST_METHOD"] == "POST"):`

- Bagian ini memeriksa apakah metode pengiriman request adalah **POST**. Hal ini memastikan bahwa kode di dalam blok ini hanya dijalankan ketika form dikirim (submit) dengan metode POST.

❑ `$name = $_POST['fname'];:`

- Mengambil nilai yang dikirimkan dari input form dengan nama **fname** melalui array superglobal **\$_POST**. Nilai ini kemudian disimpan dalam variabel **\$name**.

□ **if (empty(\$name)):**

- Memeriksa apakah input nama kosong (tidak diisi oleh pengguna). Jika kosong, maka pesan "**Name is empty**" akan ditampilkan.

□ **echo \$name;**

- Jika input nama tidak kosong, nilai yang diisi oleh pengguna akan ditampilkan di halaman sebagai output.

Alur Eksekusi:

- Ketika pengguna mengakses halaman, mereka akan melihat form untuk memasukkan nama.
- Setelah pengguna memasukkan nama dan menekan tombol "Submit", data dikirim menggunakan metode POST ke halaman yang sama.
- PHP memproses data yang dikirim:
- Jika input kosong, menampilkan "Name is empty".
- Jika input berisi nama, menampilkan nama tersebut di halaman.

4. Variabel **\$_SESSION**

Variabel **\$_SESSION** adalah array asosiatif yang menyimpan data sesi pengguna. Variabel ini bisa kita gunakan untuk menyimpan user yang login pada satu sesi tertentu. Atau juga bisa digunakan untuk menyimpan data *cart* pada toko online. Secara *default*, umur sesi pada PHP adalah **1440 detik** atau **24 menit**.

5. Variabel **\$_COOKIE**

Mirip dengan **\$_SESSION**, variabel **\$_COOKIE** bisa kita gunakan untuk menyimpan suatu data yang berkaitan dengan user: misal informasi login, informasi *cart* pada toko online, dan sebagainya.

Bedanya, *cookie* adalah file berukuran kecil yang disimpan pada browser pengguna. File tersebut akan senantiasa dikirim setiap kali browser mengirimkan *request* ke server. Umur *cookie* umumnya lebih panjang dari pada umur sesi.

6. Variabel \$_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.

```
<html>
<body>

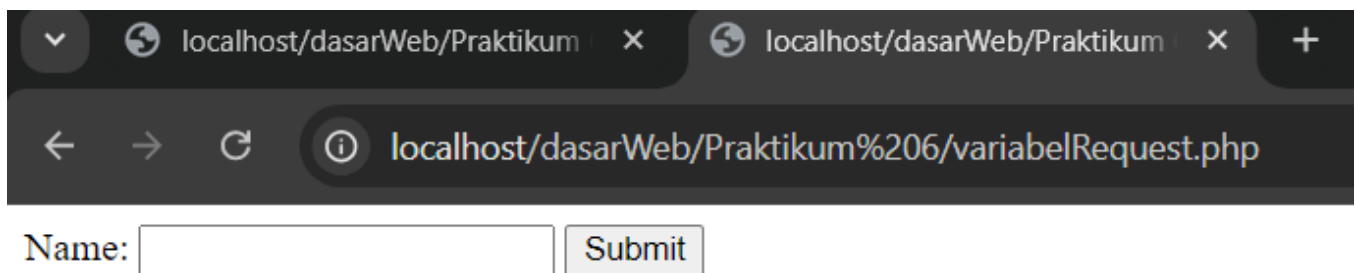
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?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>
```

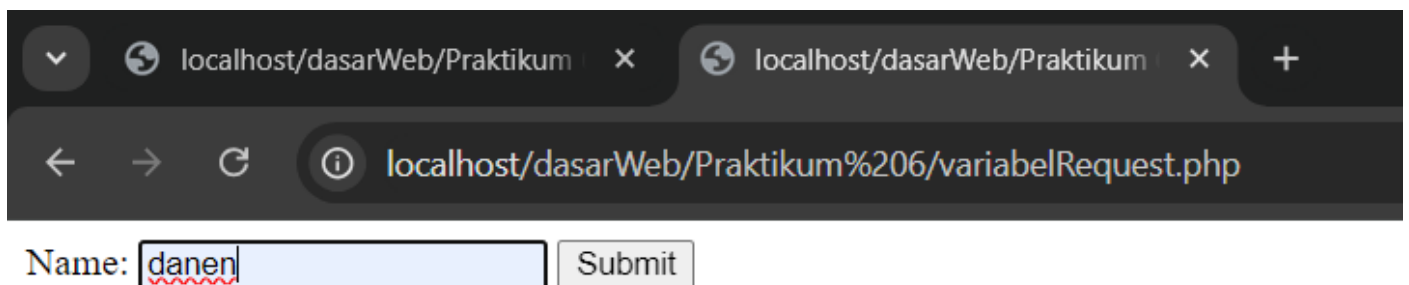
Jalankan kode program berikut ini. Apa yang bisa kalian simpulkan dari output yang dihasilkan. Dan apa bedanya dengan variable global \$POST (Soal no 25)

Tampilan awal:



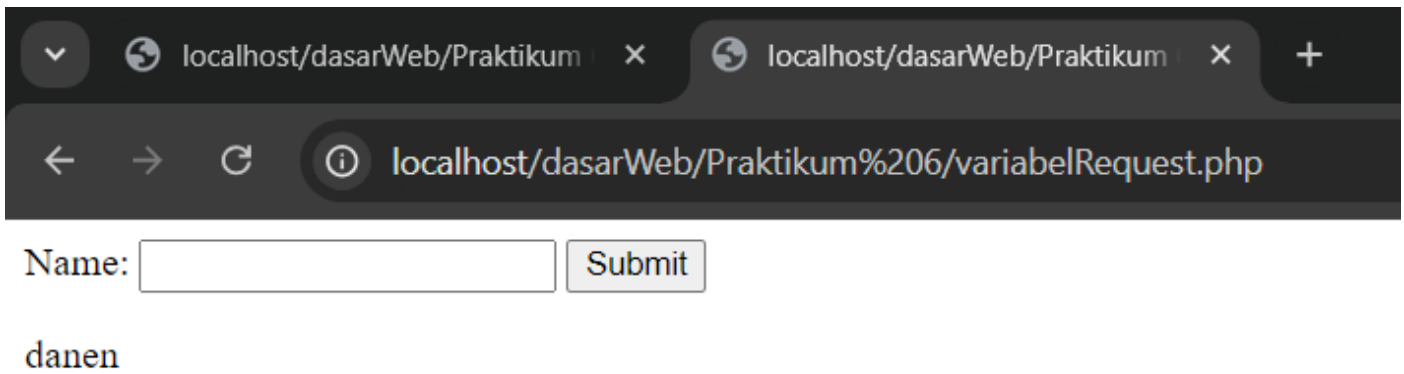
The screenshot shows a web browser with two tabs. The active tab is titled 'localhost/dasarWeb/Praktikum' and the address bar shows 'localhost/dasarWeb/Praktikum%206/variabelRequest.php'. Below the browser, there is a form with the label 'Name:' followed by an empty text input field and a 'Submit' button.

Setelah di isi dan submit:



The screenshot shows the same web browser as before, but the text input field now contains the name 'danen'. The 'Submit' button is still present next to it.

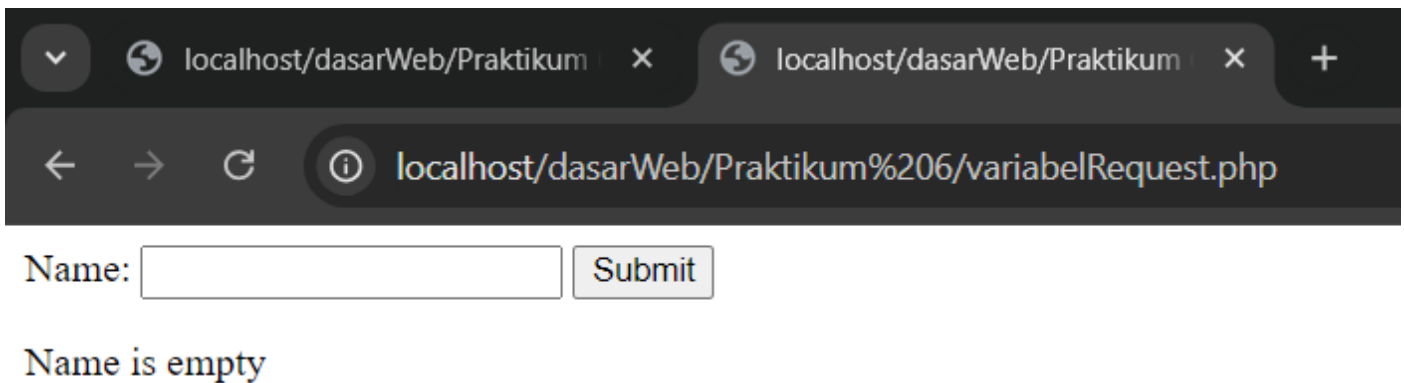
Hasil:



Name:

danen

Saat input nama kosong kemudian submit:



Name:

Name is empty

menggunakan metode POST untuk mengirimkan data ke server, kemudian diproses dengan menggunakan variabel superglobal `$_REQUEST`.

Alur Eksekusi Program:

1. Form HTML:

- Form meminta pengguna untuk memasukkan nama dalam input teks dengan atribut `name="fname"`.
- Saat pengguna menekan tombol **Submit**, data dikirim ke server dengan metode POST.

2. PHP Processing:

- PHP mengecek apakah request dikirim menggunakan metode POST
- Kemudian, menggunakan `$_REQUEST['fname']`, PHP mengumpulkan nilai dari input dengan nama **fname**.
- Jika nilai input kosong, PHP akan menampilkan **"Name is empty"**.
- Jika nilai input tidak kosong, PHP akan menampilkan nama yang diisi oleh pengguna.

Kesimpulan dari Output:

- **Output yang dihasilkan** tidak berbeda ketika menggunakan **\$_REQUEST** atau **\$_POST** dalam hal ini, karena form menggunakan metode POST, sehingga data yang dikirim akan diambil oleh kedua variabel superglobal tersebut.

Perbedaan antara \$_REQUEST dan \$_POST:

1. \$_POST:

- **\$_POST** hanya mengumpulkan data yang dikirimkan melalui **metode POST**. Biasanya digunakan ketika ada inputan form yang dikirimkan dengan metode POST (misalnya, form pengisian data).
- Lebih spesifik untuk menangani data yang dikirim via POST dan lebih sering digunakan jika kita yakin bahwa metode yang digunakan adalah POST.

2. \$_REQUEST:

- **\$_REQUEST** mengumpulkan data dari tiga sumber:
 - **\$_POST**: Data dari formulir yang dikirim dengan metode POST.
 - **\$_GET**: Data yang dikirim melalui URL (query string) dengan metode GET.
 - **\$_COOKIE**: Data yang tersimpan di cookie.
- Karena **\$_REQUEST** bisa mengumpulkan data dari berbagai sumber, dalam beberapa kasus bisa membingungkan atau menimbulkan masalah keamanan (karena bisa menerima data yang tidak dimaksudkan).

Kesimpulan Akhir:

- **\$_REQUEST** adalah superglobal yang lebih umum, karena mencakup POST, GET, dan COOKIE, sedangkan **\$_POST** hanya mencakup data yang dikirim dengan metode POST.
- **\$_POST** lebih aman dan direkomendasikan jika kita hanya bekerja dengan data yang dikirimkan melalui metode POST, sedangkan **\$_REQUEST** mungkin digunakan ketika kita ingin menangani input dari beberapa metode sekaligus (POST, GET, atau COOKIE).

7. Variabel \$_FILES

Variabel **\$_FILES** adalah array asosiatif yang menyimpan data file yang diunggah pengguna dalam satu *request* dengan metode **POST** atau **PUT**.

8. Variabel \$_ENV

Variabel **\$_ENV** adalah array asosiatif yang berisi data tentang *environment* yang skrip PHP berjalan di atasnya. Variabel **\$_ENV** disediakan oleh *shell* yang menjalankan skrip PHP, sehingga nilainya bisa bervariasi tergantung dengan sistem operasi yang digunakan.

Di dalam *framework* PHP modern seperti laravel, variabel `$_ENV` juga digunakan untuk menyimpan hal-hal yang berkaitan dengan *environment* seperti nama database, password database, dan nilai lainnya untuk melakukan konfigurasi *framework*.

9. Variabel \$GLOBALS

Variabel `$GLOBALS` adalah array asosiatif yang menyimpan semua variabel global yang didefinisikan saat program dijalankan. Variabel `$GLOBALS` merupakan variabel super global PHP yang digunakan untuk mengakses variabel global dari mana saja dalam scrip PHP (juga dari dalam fungsi atau metode).

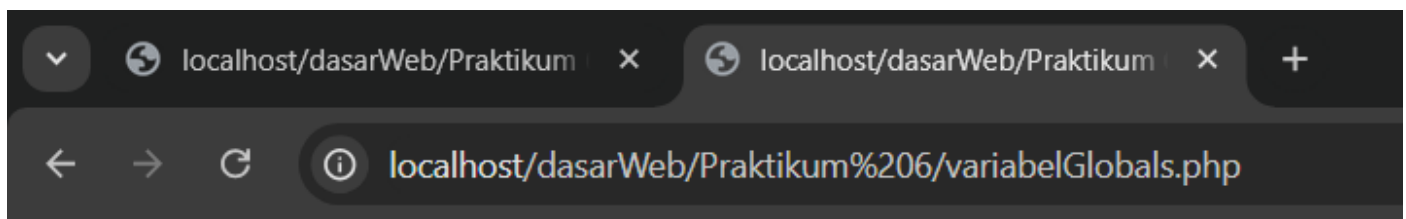
Cara menggunakan variabel super global `$GLOBALS`:

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

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

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

Bagaimana output dari kode soal di atas kemudian jelaskan! (Soal no 26)



100

menunjukkan bagaimana variabel global dapat diakses di dalam fungsi dengan menggunakan array `$GLOBALS` dalam PHP.

1. Deklarasi Variabel Global:

- Variabel `$x` dan `$y` dideklarasikan di luar fungsi dengan nilai masing-masing **75** dan **25**.
- Karena variabel ini dideklarasikan di luar fungsi, mereka adalah **variabel global**.

2. Fungsi `addition()`:

- Fungsi `addition()` didefinisikan untuk melakukan penjumlahan variabel global `$x` dan `$y`.
- `$GLOBALS` adalah array superglobal yang menyimpan referensi semua variabel global yang ada di skrip. Oleh karena itu:
 - `$GLOBALS['x']` mengakses nilai dari variabel `$x`.
 - `$GLOBALS['y']` mengakses nilai dari variabel `$y`.
- Hasil penjumlahan dari `$x` dan `$y` disimpan ke dalam `$GLOBALS['z']`, yang membuat variabel `$z` tersedia secara global.

3. Memanggil Fungsi `addition()` dan Menampilkan Variabel Global `z`:

- Fungsi `addition()` dipanggil untuk melakukan penjumlahan.
- Setelah fungsi dipanggil, variabel `$z` (yang dibuat dalam array `$GLOBALS`) akan berisi hasil penjumlahan **75 + 25**, yaitu **100**.

- **echo \$z;** menampilkan nilai dari **\$z** yang sekarang bernilai **100**.

Kesimpulan:

1. Program ini menunjukkan penggunaan array superglobal **\$GLOBALS** untuk mengakses dan memodifikasi variabel global di dalam fungsi.
2. Variabel **\$x** dan **\$y** diakses menggunakan **\$GLOBALS** di dalam fungsi **addition()**, kemudian hasilnya disimpan dalam variabel global **\$z**.
3. Output dari program ini adalah **100**, yang merupakan hasil penjumlahan dari **75 + 25**.

Referensi:

- 1) Nixon, Robin. (2018). Learning PHP, MySQL, JavaScript, CSS & HTML: A Step-by-step Guide to Creating Dynamic Websites, 5th Edition. O'Reilly Media, Inc.
- 2) Forbes, Alan. (2012). The Joy of PHP: A Beginners's Guide to Programming Interactive Web Applications with PHP and MySQL, 5th Edition. Plum Island Publishing