

JOBSHEET 6
PEMROGRAMAN WEB



NAFISAH ALIYAH KHUMAINI

244107060066
SIB 2C

PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2025



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	Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/array_1.php</code>
3	Amati hasil yang ditampilkan
4	<p>Untuk menampilkan array, selain menggunakan indeks kita juga bisa menggunakan perulangan. Coba tampilan hasil dari kode program diatas dengan menggunakan perulangan. Tampilkan kode program dan hasilnya. (soal no.1)</p> <pre>foreach (\$Listdosen as \$dosen) { echo \$dosen . "
"; }</pre> <p>Array Terindeks</p> <p>Elok Nur Hamdana Unggul Pamenang Bagas Nugraha</p>

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 (`=>`) untuk mendefinisikan nilai kepada kunci. Berikut adalah cara penulisan *associative* array:

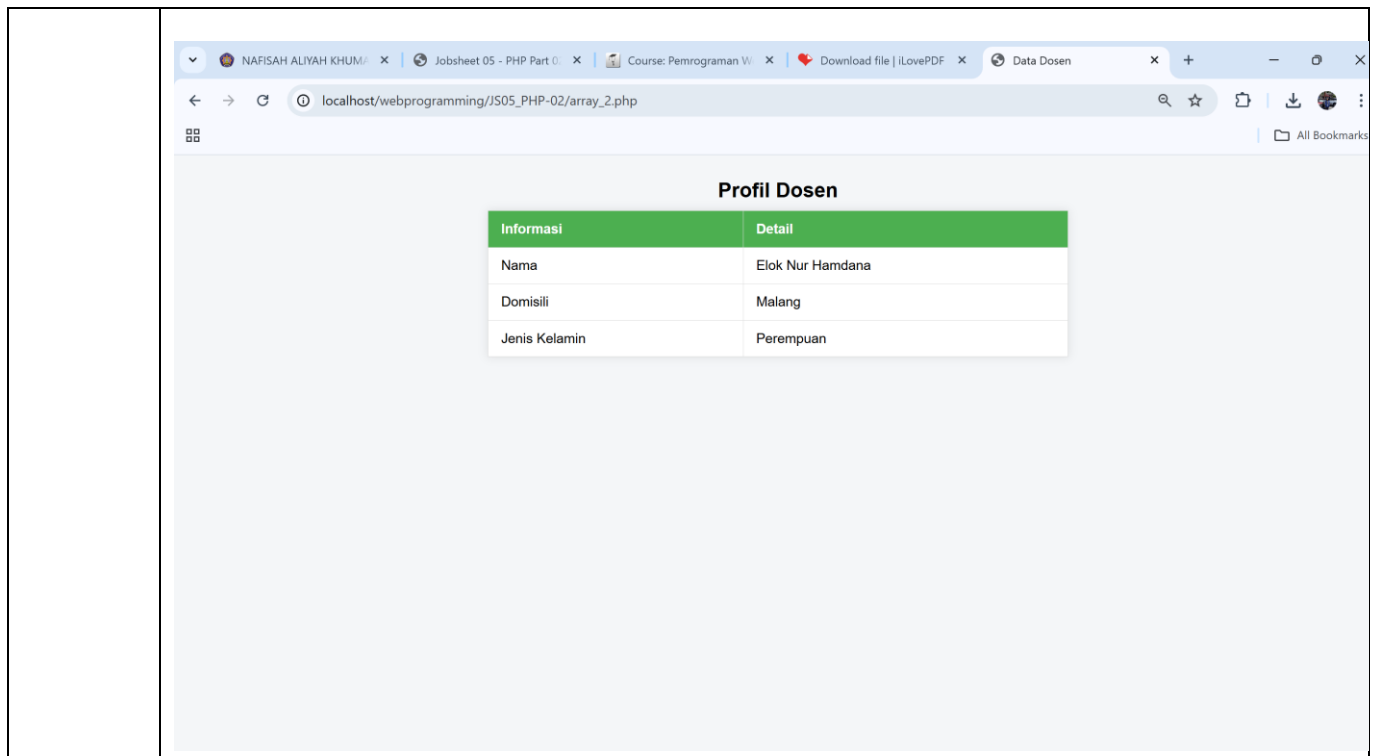
```
<?php
    $variable = array(
        "Key0" => "value0",
        "key1" => "value1",
        "key2" => "value2",
        :
        "key-n" => "value-n"
    );
?>
```

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)</p> <p>*Untuk penggunaan style bebas boleh internal atau eksternal file.</p> <pre><style> body { font-family: Arial, sans-serif; background-color: #f4f6f8; padding: 20px; } table { width: 50%; margin: auto; border-collapse: collapse; background-color: #ffffff; box-shadow: 0 0 10px rgba(0,0,0,0.1); } th, td { padding: 12px 15px; border: 1px solid #ddd; text-align: left; } th { background-color: #4CAF50; color: white; } caption { caption-side: top; font-size: 1.5em; margin-bottom: 10px; font-weight: bold; } </style></pre>



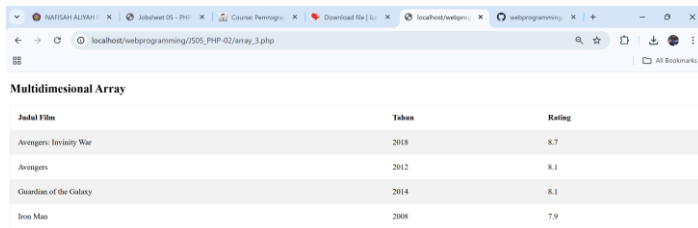
Praktikum Bagian 3. *Multidimensional* Array

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
Avengers	2012	8.1
Guardian of the Galaxy	2014	8.1
Iron Man	2008	7.9

Jadi untuk membuat judul kolomnya di html. Isinya di php menggunakan array. Untuk menampilkannya menggunakan indeks array. Array bisa ditambahkan style.

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><?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 Elok Senang berkenalan dengan Anda Assalamualaikum, Perkenalkan, nama saya Elok Senang berkenalan dengan Anda</p> <p>Penggunaan fungsi pada php menggunakan sintaks function</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.</p> <p>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> <?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> <hr/> <p>Hello, Perkenalkan, nama saya Hamdana Senang berkenalan dengan Anda</p> <hr/> <p>Selamat Pagi, Perkenalkan, nama saya Elok Senang berkenalan dengan Anda</p> <p>Menggunakan function berparameter. Yang pertama itu langsung mengisi parameter di function. Yang kedua bisa buat variabel dulu kemudian ketika memanggil fungsi mengisi parameter dengan variabel yang sudah dibuat</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.</p> <p>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> <?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>Hello, Perkenalkan, nama saya Hamdana Senang berkenalan dengan Anda</p> <hr/> <p>Assalamualaikum, Perkenalkan, nama saya Elok Senang berkenalan dengan Anda</p> <p>Fungsi perkenalan() memiliki parameter default salam = 'assalamualaikum'. Jadi ketika memanggil fungsi tersebut jika tidak mengisi parameter salam maka sistem otomatis mengisinya dengan assalamualaikum</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><?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 35 tahun</p> <p>Fungsi hitungUmur memakai parameter thn_lahir dan thn_sekarang. Kemudian ada variabel umur yang menggunakan operasi pengurangan thn_skrng dengan thn_lahir. Setelah itu mengembalikan nilai umur</p>
Memanggil Fungsi di dalam fungsi	
12	<p>Fungsi yang sudah kita buat, dapat juga dipanggil di dalam fungsi lain.</p>
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>

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>Ini adalah rekursi yang terkontrol. Base case nya adalah \$indeks < \$jumlah. Kemudian ada step rekursif</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 data-bbox="277 147 735 1010"><?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 data-bbox="277 1111 1002 1417">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 data-bbox="341 1503 469 1637" style="list-style-type: none"> • Beranda • Berita • Tentang • Kontak <p>Menu bertingkat belum menangani submenu secara rekursif</p>

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

```
function tampilkanMenuBertingkat(array $menu){
    echo "<ul>";
    foreach ($menu as $item) {
        echo "<li>{$item['nama']}";

        // Cek apakah ada submenu dan pastikan itu array
        if (isset($item['subMenu']) && is_array($item
            ['subMenu'])) {
            tampilkanMenuBertingkat($item
                ['subMenu']); // Rekursi
        }

        echo "</li>";
    }
    echo "</ul>";
}
```

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
<code>ltrim()</code>	Menghilangkan karakter spasi di awal string

<code>rtrim()</code>	Menghilangkan karakter spasi di akhir string
<code>strtoupper()</code>	Mengubah huruf menjadi huruf capital
<code>strtolower()</code>	Mengubah huruf menjadi huruf kecil (<i>lowercase</i>)
<code>str_replace()</code>	Menggantikan nilai dari beberapa bagian dari string dengan string yang lain
<code>ucwords()</code>	Mengubah huruf awal dari sebuah kata dengan huruf besar
<code>explode()</code>	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
<code>\n</code>	Baris baru
<code>\r</code>	Karakter carriage-return
<code>\t</code>	Karakter tab
<code>\\$</code>	Karakter \$ itu sendiri
<code>\"</code>	Untuk menampilkan tanda petik dua
<code>\\</code>	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>Di php dapat menghitung jumlah karakter termasuk spasi dan tanda baca menggunakan strlen. Str_word_count dapat menghitung jumlah kata dalam string. strtoupper mengubah semua huruf jadi kapital sedangkan strtolower mengubah semua huruf menjadi huruf kecil</p>
Escape Character	
3	Buat file <code>string2.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut:
	<pre><?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> <pre>Baris baru Baris\nbaru Halo Dunia Halo\rDunia Halo Dunia! Halo\rDunia! Katakanlah "Tidak pada narkoba" Katakanlah "Tidak pada narkoba"</pre> <p>Kalau memakai kutip ganda (“) \n untuk baris baru, \r untuk carriage turun, \” escape diperlukan. Kalau memakai kutip tunggal (‘) \n teks literal, \r teks literal, \’tidak perlu escape</p>
Membalik String menggunakan perintah <i>strrev()</i> .	
5	Buat file <code>string3.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut: <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>gnalam kera ayaS</p> <p>Mencetak variabel pesan tapi terbalik</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>gnalam kera ayaS</p> <p>Explode memecah string “saya arek malang” menjadi array [“Saya”,”arek”,”malang”]. Array map ini untuk membalik setiap kata dalam array. [“ayaS”, “kera”, “gnalam”]. Lalu digabungkan dengan menggunakan implode dan terakhir dicetak menggunakan echo \$pesan</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 tag <i>echo</i> .
5	<p>Dari dua cara tersebut mana yang lebih mudah menurut kalian, kemukakan jawaban disertai dengan alasan (soal no 17)</p> <p>Metode pertama lebih mudah dan lebih bersih untuk tampilan sederhana. Metode kedua lebih fleksibel untuk logika kompleks dan manipulasi string.</p>

--	--

Entities HTML

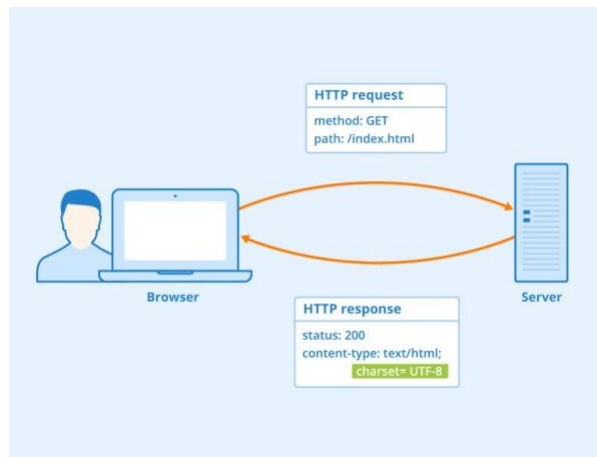
Ketika membuat konten website, tidak akan terlepas dari menggunakan karakter khusus seperti simbol copyright (©), ampersand (&), lebih kecil (<), lebih besar (>), titik koma (;) dan sebagainya. Untuk menampilkan simbol-simbol tersebut di HTML perlu menggunakan entitas karakter (*character entities*). 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	&

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 R\ *respond*.



HTTP Header

HTTP *respond* 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 pengunjung web.

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

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>Date</p> <p>Today is 2025/09/21 Today is 2025.09.21 Today is 2025-09-21 Today is Sunday</p> <p>Y/m/d → Tahun/Bulan/Tanggal dengan garis miring → 2025/09/21 Y.m.d → Tahun.Bulan.Tanggal dengan titik → 2025.09.21 Y-m-d → Tahun-Bulan-Tanggal dengan tanda minus → 2025-09-21 l → Nama hari dalam bahasa Inggris → Sunday</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>Time</p> <p>10:43:52pm</p> <p>Menampilkan waktu terkini dengan format h:i:sa (hours, minute, second)</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.

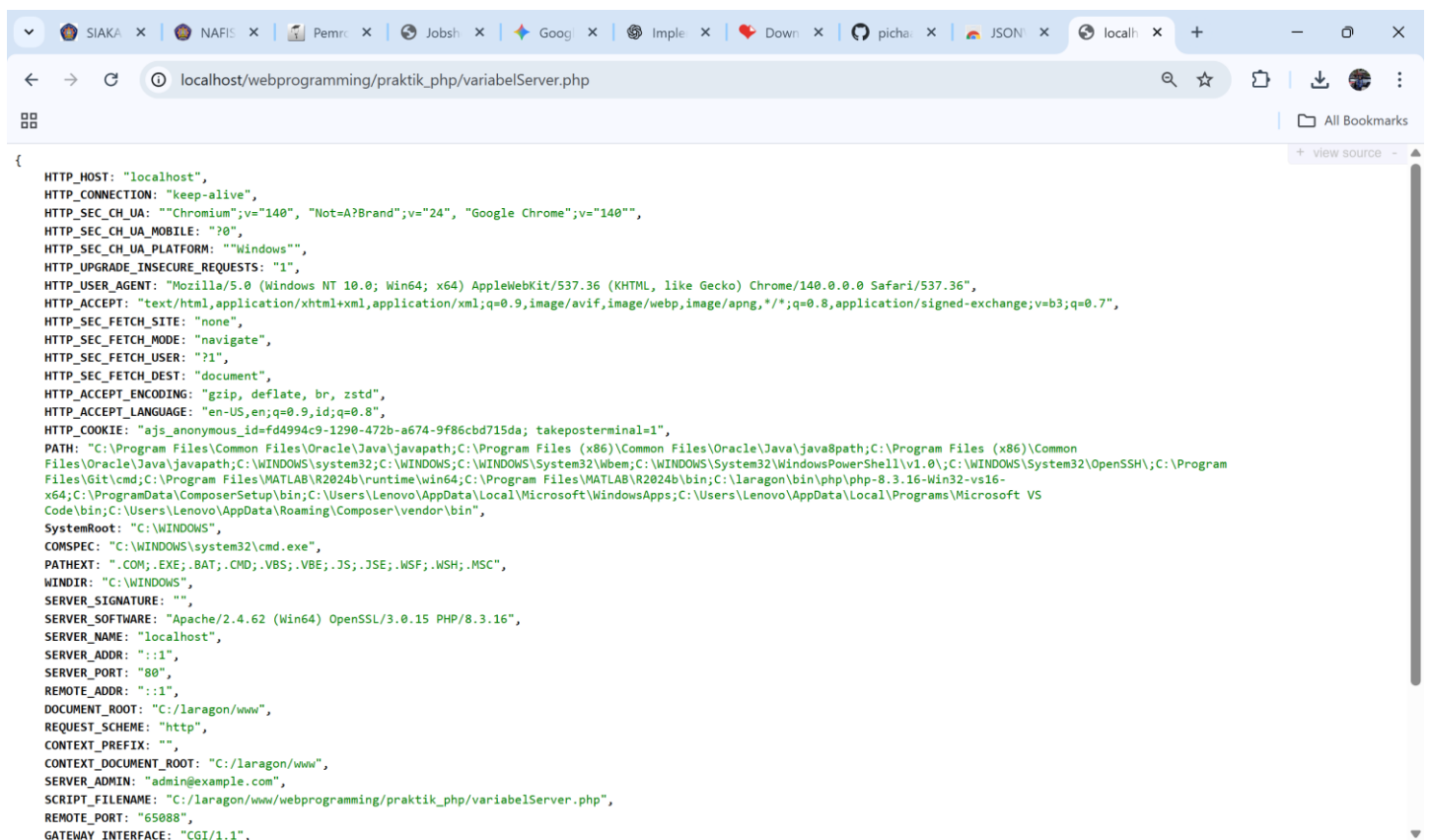
Nilai yang tersimpan pada variabel \$_SERVER disediakan oleh web server, oleh karena itu tidak ada jaminan khusus bahwa setiap web server yang kita gunakan akan memberikan semua data-data standar yang ada.

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

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

Bagaimana output yang didapatkan (**Soal no 21**) sebelum di jalan silakan install ekstensi JSON View pada link berikut ini

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



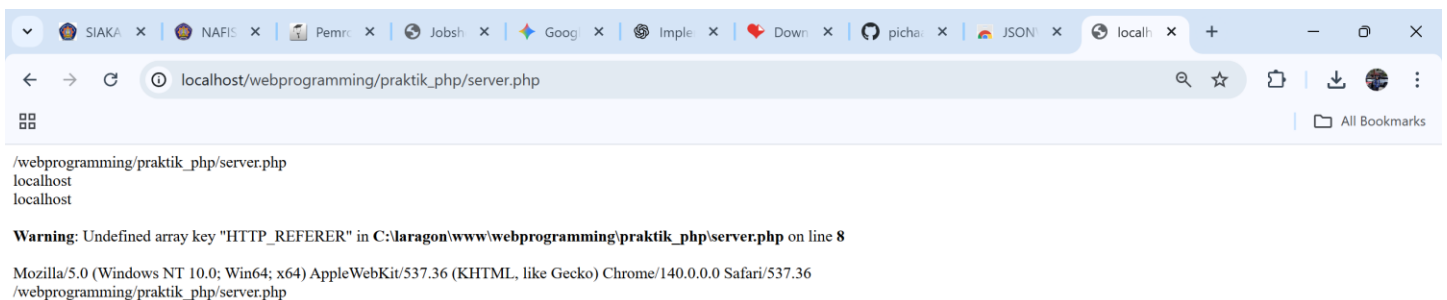
```
{
  HTTP_HOST: "localhost",
  HTTP_CONNECTION: "keep-alive",
  HTTP_SEC_CH-UA: "\"Chromium\";v=\"140\"\", \"Not=A?Brand\";v=\"24\"\", \"Google Chrome\";v=\"140\"\",",
  HTTP_SEC_CH-UA-MOBILE: "?0",
  HTTP_SEC_CH-UA-PLATFORM: "\"Windows\"",
  HTTP_UPGRADE_INSECURE_REQUESTS: "1",
  HTTP_USER_AGENT: "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 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: "en-US,en;q=0.9,id;q=0.8",
  HTTP_COOKIE: "ajs_anonymous_id=fd4994c9-1290-472b-a674-9f86cbd715da; takepostterminal=1",
  PATH: "C:\\Program Files\\Common Files\\Oracle\\Java\\javapath;C:\\Program Files (x86)\\Common Files\\Oracle\\Java\\javapath;C:\\Program Files (x86)\\Common Files\\Oracle\\Java\\javapath;C:\\WINDOWS\\system32\\cmd.exe;C:\\WINDOWS\\system32\\WindowsPowerShell\\v1.0\\cmd.exe;C:\\WINDOWS\\system32\\OpenSSH\\C:\\Program Files\\Git\\cmd;C:\\Program Files\\MATLAB\\R2024b\\runtime\\win64;C:\\Program Files\\MATLAB\\R2024b\\bin;C:\\laragon\\bin\\php\\php-8.3.16-Win32-vs16-x64;C:\\ProgramData\\ComposerSetup\\bin;C:\\Users\\Lenovo\\AppData\\Local\\Microsoft\\WindowsApps;C:\\Users\\Lenovo\\AppData\\Local\\Programs\\Microsoft VS Code\\bin;C:\\Users\\Lenovo\\AppData\\Roaming\\Composer\\vendor\\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/2.4.62 (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/webprogramming/praktik_php/variabelServer.php",
  REMOTE_PORT: "65088",
  GATEWAY_INTERFACE: "CGI/1.1",
```

```

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)



PHP SELF itu untuk menampilkan file yang lagi dieksekusi = /webprogramming/praktik_php/server.php

SELF NAME itu untuk menampilkan nama server = localhost

SERVER HOST itu untuk menampilkan host dari permintaan HTTP = localhost

HTTP REFERE itu URL halaman sebelumnya jika ada = unidentified

HTTP USER AGENT itu info tentang browser dan sistem operasi pengguna = mozilla chrome

SCRIPT NAME itu nama script yang dijalankan = sama dengan php self

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

No	Variabel dan Deskripsi
1	<code>\$_SERVER['PHP_SELF']</code> Berisi informasi nama file yang sedang dieksekusi, nama file diambil berdasarkan dari dokumen <i>root</i>
2	<code>\$_SERVER['SERVER_ADDR']</code> Alamat IP server dari file yang sedang dieksekusi.
3	<code>\$_SERVER['SERVER_NAME']</code>

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	<code>\$_SERVER['SERVER_PROTOCOL']</code> Ini adalah protokol komunikasi yang sedang berjalan. Apakah HTTP atau HTTPS. Misal: 'HTTP/0.1'
5	<code>\$_SERVER['REQUEST_METHOD']</code> Berisi jenis metode <i>request</i> dari file PHP yang sedang dieksekusi. Misalkan: <code>GET POST PUT DELETE OPTIONS</code>
6	<code>\$_SERVER['QUERY_STRING']</code> Akan mengembalikan query string dari file atau skrip PHP yang sedang dieksekusi. Misalkan user mengakses halaman <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> Mengembalikan konten dari host, seperti misalnya: header (jika memang ada).
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> Variabel ini berisi informasi dari user yang sedang melakukan <i>request</i> konten: mulai dari browser yang digunakan, bahasa, hingga OS. Contoh nilai dari variabel ini adalah: Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586
11	<code>\$_SERVER['REMOTE_ADDR']</code> Berisi alamat IP dari user yang sedang mengakses halaman web PHP
12	<code>\$_SERVER['SCRIPT_FILENAME']</code> Nama path absolut dari file yang sedang dieksekusi.
13	<code>\$_SERVER['REQUEST_URI']</code> Alamat uri dari file yang sedang dieksekusi. Misalkan: <code>"/php/halo-dunia"</code> .

2. Variabel \$_GET

Variabel `$_GET` adalah array asosiatif yang berisi nilai dari *query string*. Misalkan kita memiliki file `halo-dunia.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>

| Halo Elok! Apakah benar Anda berusia 37 tahun?

`$_GET['nama']` dan `$_GET['usia']` mengambil nilai dari parameter URL.

Tanda `@` digunakan untuk **menyembunyikan error** jika parameter tidak tersedia (meskipun tidak disarankan untuk penggunaan jangka panjang).

`echo` mencetak kalimat dengan interpolasi variabel.

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)

Karena form menggunakan `method="post"`, maka \$_POST adalah cara yang **paling tepat dan aman** untuk mengambil data. Ini lebih spesifik dibandingkan \$_REQUEST, yang bisa mengambil data dari \$_GET, \$_POST, dan \$_COOKIE.

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)

Name:

picha

\$_REQ = gabungan dari get, post, dan cookie. Dipakai saat tidak yakin metode pengiriman.

\$_POS hanya data dari metode POST. Saat form menggunakan metode POST.

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 skrip 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

\$x dan \$y adalah variabel global yang nilainya masing-masing 75 dan 25. Fungsi addition() menjumlahkan \$x dan \$y menggunakan array superglobal \$GLOBALS. Hasil penjumlahan disimpan dalam \$GLOBALS['z'], sehingga \$z menjadi variabel global juga. Setelah fungsi dipanggil, echo \$z; akan mencetak hasilnya:

100

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