

SEPINTAS LALU

Asas PHP

*php*8

PHP

PHP: Hypertext Processor

The logo for PHP 8, featuring the word "php" in a bold, lowercase, italicized sans-serif font, followed by a large, stylized blue "8" that has a circular, swirling design.

Pengenalan PHP

1. Apa itu PHP
2. Persiapan
3. Memahami Server-Side & Client-Side

PHP (Pengaturcaraan)

Asas Pengaturcaraan

- Sintaks
- PHP & HTML
- Pembolehubah / Variable
- Array
- Logic (if-else, switch)
- Operator
- Loop (for, foreach, while)
- Fungsi (function)
- Include & Require
- Superglobals

Apa itu PHP?

1. PHP adalah bahasa pengaturcaraan untuk web
2. Pemprosesan PHP berlaku di web server (server side)
3. PHP boleh membuat sambungan ke database dan menyimpan data untuk jangka panjang
4. PHP menjana HTML untuk dihantar ke pelawat, dan HTML akan dipaparkan oleh browser
5. Dengan PHP, HTML yang berbeza boleh dijana bergantung kepada situasi, input, dll... menjadikan website dinamik.

Apa itu PHP?

6. Rujukan Rasmi PHP

- a. <https://www.php.net>
- b. <https://www.php.net/manual/en>

Persiapan

- PHP perlu diletakkan di server
- Untuk pembangunan, kita perlu pasang “aplikasi server” dalam komputer
- Termudah, gunakan :
 - Laragon untuk Windows, atau <https://laragon.org>
 - MAMP untuk Mac OS <https://mamp.info>

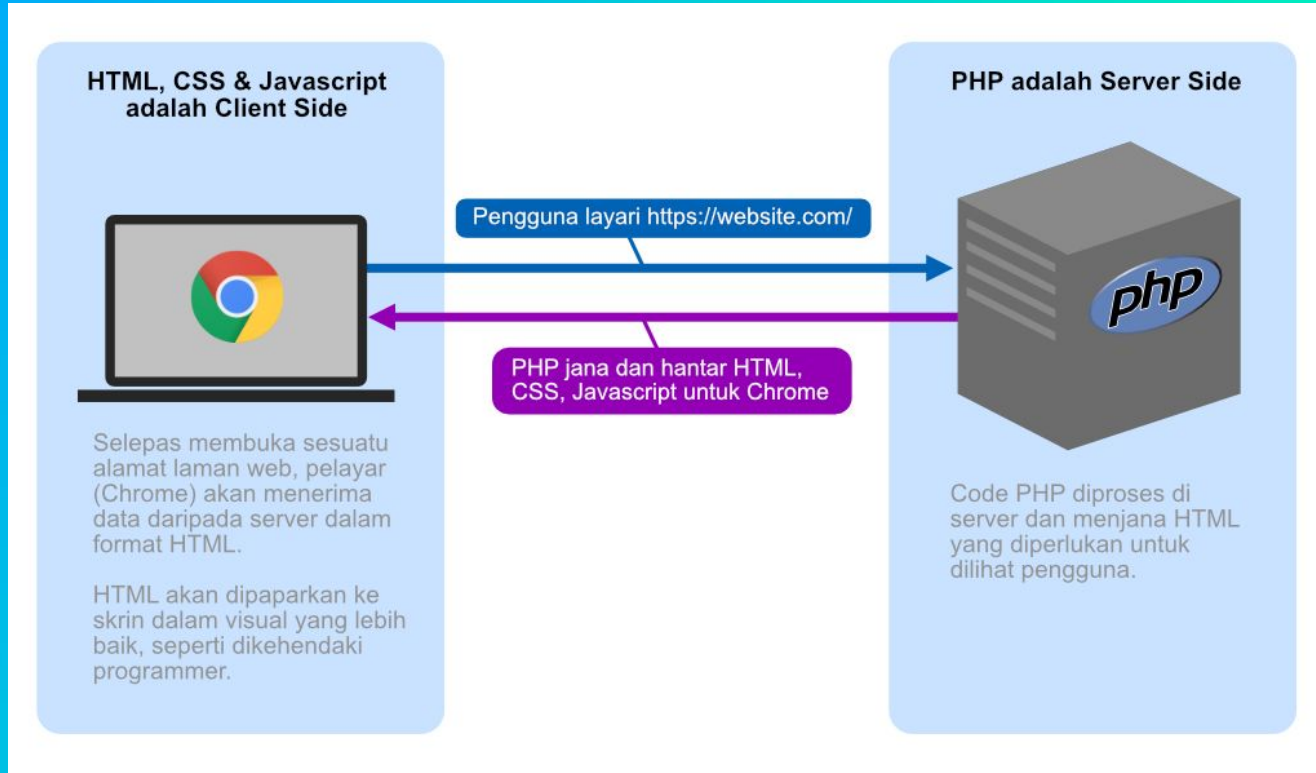


Laragon



MAMP

Memahami Server-Side & Client-Side



Sintaks

```
<?php
    echo "Selamat pagi";
?>
```

1. Fail kod PHP mempunyai ekstensyen .php
2. Kod PHP bermula dengan `<?php` dan diakhiri dengan `?>`
3. Setiap pernyataan PHP berakhir dengan semicolon `;`

Sintaks

```
<?php
    $Nama = 'Ali';
    $nama = 'Muthu';
    $NAMA = 'Ah Hock';
?>
```

1. Kod PHP adalah case-sensitive
2. \$Nama, \$nama, \$NAMA adalah variable berbeza

Sintaks

```
<?php
    // Ini komen satu baru -- tidak akan dilaksanakan

    /* -- ini juga komen banyak baris
        echo "Selamat pagi";
    */
?>
```

1. Komen tidak akan dilaksanakan oleh PHP
2. Kita boleh menulis komen dengan `//`
3. Atau dengan `/* ... */`

PHP & HTML

```
<h1>Selamat pagi  
<?php echo $nama; ?>  
</h1>
```

1. Kod PHP dan HTML boleh diselang-selikan
2. Pastikan kod PHP berada di antara `<?php` dan `?>`

Pembolehubah (Variable)

```
<?php
    $nama = "Ali";
    $umur = 11;
?>
```

1. Pembolehubah wujud dalam semua pengaturcaraan
2. Pembolehubah membantu memegang nilai
3. Pembolehubah boleh berubah nilai

Pembolehubah (Variable)

```
<?php
    $nama = "Ali"; // string, atau teks
    $umur = 11;    // integer, atau nombor bulat
    $berat = 30.5; // float, atau nombor perpuluhan
    $murid = true; // boolean, atau benar / salah
?>
```

1. Pembolehubah mempunyai jenis data (data type)

Pembolehubah (Variable)

```
<?php
    $nama = "Ali";
    echo "Nama $nama"; // menghasilkan -- Nama Ali
    echo 'Nama $nama'; // menghasilkan -- Nama $nama
?>
```

1. Apabila memaparkan data, berhati-hati dengan single quote (') dan double quote (")
2. Double quote memaparkan nilai pembolehubah
3. Single quote memaparkan teks

Membina Array

```
<?php
    $murid = []; // membina array kosong
    $murid = ['Siti','Aliyah']; // membina array dengan nilai awal
    $murid[] = 'Ali'; // menambah item dalam array di hujung
    $murid[2] = 'Abu'; // mengubah nilai item dalam array
?>
```

1. Array adalah sejenis data type yang boleh menyimpan variable

Membina Array

```
<?php
    $murid = ['Siti','Aliyah'];
    var_dump($murid);
    print_r($murid);
?>
```

1. Kita boleh gunakan fungsi `var_dump` atau `print_r` untuk melihat kandungan array

Associative Array

```
<?php
    $murid = [
        'nama' => 'Siti',
        'umur' => 11
    ];
    echo $murid['nama'];
?>
```

1. Array dengan kunci teks dipanggil associative array

Nested Array

```
<?php
    $murid1 = ['nama' => 'Siti', 'umur' => 11];
    $murid2 = ['nama' => 'Atan', 'umur' => 11];
    $kelas[] = $murid1;
    $kelas[] = $murid2;
    echo $kelas[0]['nama'];
?>
```

1. Dalam array boleh ada array lain

Logic

1. Pengaturcaraan banyak menggunakan LOGIK
2. Beberapa pernyataan LOGIK dalam PHP
 - a. if
 - b. if ... else
 - c. if ... elseif ... else
 - d. switch ... case

Logic (if-elseif-else)

```
<?php
    if ($nama == 'Ali') {
        echo "Selamat Datang Ali";
    } elseif ($nama == 'Raju') {
        echo "Apa Khabar Raju";
    } else {
        echo "Saya PHP. Awak siapa?";
    }
?>
```

Logic (switch-case)

```
<?php
    switch($nama) {
        case 'Ali':
            echo "Selamat Datang Ali"; break;
        case 'Raju':
            echo "Apa Kabar Raju"; break;
        default:
            echo "Saya PHP. Awak siapa?";
    }
?>
```

Operator

1. Operator adalah simbol seperti `+ - / * = . > <`
2. Operator membantu dalam operasi matematik
3. Operator membantu mengubah nilai pembolehubah
4. Operator membantu memproses logik

Operator (Matematik)

```
<?php
```

```
$x = 15 - 4; // tolak, hasil 11
```

```
$x = 15 + 4; // tambah, hasil 19
```

```
$x = 15 / 4; // bahagi, hasil 3.75
```

```
$x = 15 * 4; // darab, hasil 60
```

```
$x = 15 % 4; // modulus (baki), hasil 3
```

```
$x = 15 ** 4; // kuasa, hasil 50,625
```

```
?>
```


Operator (Assignment, Meletakkan Nilai)

```
<?php
    $x = 15;    // memberi nilai 15
    $x += 15;   // ditambah 15
    $x -= 15;   // ditolak 15
    $x *= 15;   // didarab
    $x /= 15;   // dibagi 15
    $x %= 15;   // operasi modulus
?>
```

Operator (Assignment, Meletakkan Nilai)

```
<?php
    $x++;      // memberi nilai, kemudian ditambah 1
    ++$x;      // ditambah 1, kemudian memberikan nilai
    $x--;      // memberi nilai, kemudian ditolak 1
    --$x;      // ditolak 1, kemudian memberikan nilai
?>
```

Operator (Pembandingan)

```
<?php
    if ($x == $y) { ... } // sama dengan
    if ($x === $y) { ... } // sama dengan, dan membandingkan data type sekali
    if ($x != $y) { ... } // tidak sama dengan
    if ($x !== $y) { ... } // tidak sama dengan, dan membandingkan data type sekali
    if ($x > $y) { ... } // lebih besar
    if ($x >= $y) { ... } // lebih besar atau sama dengan
    if ($x < $y) { ... } // lebih kecil
    if ($x <= $y) { ... } // lebih kecil atau sama dengan
?>
```

Loop (Gelung)

1. Loop membantu mengulangi kod PHP sehingga syarat tertentu
2. Jenis loop
 - a. for
 - b. foreach
 - c. while
 - d. do ... while

Loop (for)

```
<?php
    for($x = 0; $x < 10; $x++) {
        echo "$x <br>";
    }
?>
```

1. Menghasilkan 10 baris nombor, dari 0 hingga 9
2. `for(syarat awal; syarat tamat; pengubah nilai) { ... }`

Loop (for)

```
<?php
    for($x = 0; $x < 1000; $x += 10) {
        echo "$x <br>";
    }
?>
```

Menghasilkan nombor gandaan 10, bermula dari 0, 10, 20 ...

Loop (for)

```
<?php
    for($x = 1000; $x > 0; $x -= 100) {
        echo "$x <br>";
    }
?>
```

Menghasilkan nombor menurun dari 1000 hingga 100

Loop (foreach)

```
<?php
    $kelas = [];
    $kelas[] = ['nama' => 'Siti', 'umur' => 11];
    $kelas[] = ['nama' => 'Atan', 'umur' => 11];
    foreach($kelas as $murid) {
        echo $murid['nama'];
    }
?>
```

Biasa digunakan dengan Array

Loop (foreach)

```
<?php
    $murid = ['nama' => 'Siti', 'umur' => 11];
    foreach($murid as $key => $value) {
        echo "$key : $value <br>"; // nama : Siti, kemudian umur : 11
    }
?>
```

Format lain, boleh memberikan kunci dan nilai untuk item

Loop (while)

```
<?php
    $a = 0;
    while($a < 10) {
        echo "$a <br>";
        $a++;
    }
?>
```

Melaksanakan kod PHP sehingga syarat dipenuhi

Loop (do...while)

```
<?php
    $a = 0;
    do {
        echo "$a <br>";
        $a++;
    } while($a < 10);
?>
```

Melaksanakan kod PHP sehingga syarat dipenuhi

Fungsi (function)

1. Fungsi mengumpulkan kod PHP dalam satu set untuk tujuan yang khusus
2. PHP mempunyai banyak fungsi-fungsi untuk pelbagai kegunaan seperti `print_r()`
3. Kita boleh menulis fungsi kita sendiri
4. Fungsi boleh menerima input (parameter)
5. Fungsi boleh memulangkan hasil

Fungsi (function)

```
<?php
    function tambah ( $a, $b ) {
        return $a + $b;
    }
    echo tambah(4, 3);
?>
```

Include & Require

1. Kita boleh memanggil fail PHP lain untuk digunakan dalam fail PHP semasa.
2. Ini memudahkan perkongsian kod antara fail-fail PHP.
3. Kod yang sama, tidak perlu diulang-taip semula.
4. `include()` membenarkan fail dipanggil, walaupun terdapat ralat dalam kod, tapi akan ada WARNING.
5. `require()` tidak membenarkan fail dengan ralat untuk dipanggil. Akan ada ERROR.

Include & Require

```
<?php
    include('functions.php');

    // fungsi daripada functions.php
    echo tambah($a, $b);

?>
```

Superglobals

1. PHP mempunyai beberapa variable khusus.
 2. Superglobals boleh digunakan di mana-mana dalam kod PHP.
- `$GLOBALS`
 - `$_SERVER`
 - `$_REQUEST`
 - `$_POST`
 - `$_GET`
 - `$_FILES`
 - `$_ENV`
 - `$_COOKIE`
 - `$_SESSION`

Superglobals (\$_GET & \$_POST)

1. Mengandung data daripada borang yang dihantar
2. Data untuk \$_GET boleh diberi bersama alamat URL dipisahkan dengan tanda soalan (?)

website.com?nama=Ali&umur=11

3. Boleh diakses melalui `$_GET['nama']` dan `$_GET['umur']`

Terima Kasih