



SERVER SIDE PROGRAMMING





PHP?

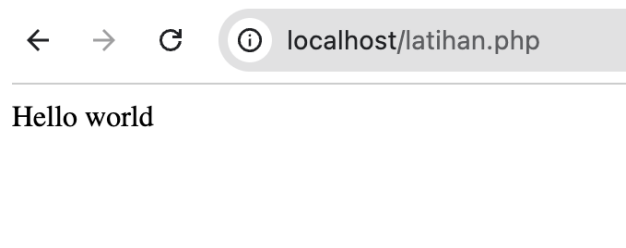
PHP adalah bahasa pemrograman skrip yang dijalankan di server, yang berarti kode PHP dieksekusi di server dan hasilnya dikirimkan ke browser pengguna sebagai HTML. PHP awalnya singkatan dari "Personal Home Page," tetapi sekarang merupakan singkatan rekursif untuk "PHP: Hypertext Preprocessor."

karena php adalah bahasa pemrograman server side, untuk menjalankan aplikasi php dibutuhkan beberapa aplikasi seperti laragon

First Code

buatlah sebuah file dalam folder document root (www) dengan diakhiri dengan ekstensi .php kemudian ketikkan kode:

```
<?php
echo "Hello world";
?>
```



variabel dan tipe data

Variabel

Variabel di PHP digunakan untuk menyimpan nilai atau data yang dapat digunakan dan dimanipulasi selama eksekusi script. Berikut adalah beberapa poin penting tentang variabel di PHP:

- Deklarasi Variabel: Variabel di PHP diawali dengan simbol \$ diikuti dengan nama variabel.
- Aturan Penamaan: Nama variabel harus dimulai dengan huruf atau underscore (_), diikuti oleh huruf, angka, atau underscore. Tidak boleh dimulai dengan angka.
- Case-Sensitive: Nama variabel bersifat case-sensitive, artinya \$nama dan \$Nama dianggap sebagai dua variabel yang berbeda.

```
<?php
$nama = "John";
$umur = 25;
?>
```

Integer: Tipe data yang digunakan untuk menyimpan bilangan bulat (positif atau negatif).

```
<?php
$angka1 = 123;
$angka2 = -456;
?>
```

Float (Double): Tipe data yang digunakan untuk menyimpan bilangan desimal.

```
<?php
$desimal1 = 1.23;
$desimal2 = -4.56;
?>
```

Boolean: Tipe data yang hanya memiliki dua nilai: true atau false.

```
<?php
$benar = true;
$salah = false;
?>
```

Tipe Data

String: Tipe data yang digunakan untuk menyimpan teks. String dapat dideklarasikan menggunakan tanda kutip tunggal (') atau ganda (").

```
<?php
$teks1 = "hello world";
$teks2 = "halo, PHP";
?>
```

Array: Tipe data yang digunakan untuk menyimpan beberapa nilai dalam satu variabel. Array dapat berupa array berindeks atau array asosiatif.

```
//array berindex
<?php
$buah = array("Apel", "Pisang",
"Jeruk");
echo $buah[0]; // Apel
```

```
//Array Asosiatif
<?php
$usia = array("John" => 25,
"Jane" => 30, "Doe" => 35);
echo $usia["John"]; // 25
```

Kondisi

Dalam pemrograman, struktur kontrol kondisional digunakan untuk membuat keputusan berdasarkan kondisi tertentu. PHP mendukung beberapa pernyataan kondisional, seperti if, else, elseif, switch, dan operator ternary.

1. pernyataan if

Pernyataan if digunakan untuk mengeksekusi blok kode jika kondisi tertentu bernilai benar (true).

```
<?php
$umur = 20;

if ($umur >= 18) {
    echo "Anda sudah dewasa.";
}
```



2. Pernyataan if-else

Pernyataan if-else digunakan untuk mengeksekusi satu blok kode jika kondisi bernilai benar, dan blok kode lain jika kondisi bernilai salah (false).

```
<?php
$umur = 16;

if ($umur >= 18) {
    echo "Anda sudah dewasa.";
} else {
    echo "Anda masih anak-anak.";
}
```

3. Pernyataan if-elseif-else

Pernyataan if-elseif-else digunakan untuk mengecek beberapa kondisi dan mengeksekusi blok kode yang sesuai dengan kondisi pertama yang bernilai benar.

```
<?php
$nilai = 85;
if ($nilai >= 90) {
    echo "Nilai Anda A.";
} elseif ($nilai >= 80) {
    echo "Nilai Anda B.";
} elseif ($nilai >= 70) {
    echo "Nilai Anda C.";
} else {
    echo "Nilai Anda D.";
}
?>
```

4. Pernyataan switch

Pernyataan switch digunakan untuk membandingkan satu nilai dengan beberapa nilai yang mungkin dan mengeksekusi blok kode yang sesuai.

```
<?php
$hari = "Senin";

switch ($hari) {
    case "Senin":
        echo "Hari ini adalah Senin.";
        break;
    case "Selasa":
        echo "Hari ini adalah Selasa.";
        break;
    case "Rabu":
        echo "Hari ini adalah Rabu.";
        break;
    default:
        echo "Hari tidak diketahui.";
        break;
}
?>
```

5. Operator Ternary

Operator ternary adalah cara singkat untuk menulis pernyataan if-else. Operator ini menggunakan tanda ? dan :.

```
<?php
$umur = 20;
$status = ($umur >= 18) ? "dewasa" : "anak-anak";
echo "Anda adalah " . $status . ".";
?>
```

Struktur kontrol kondisional adalah bagian penting dari pemrograman, memungkinkan Anda untuk membuat keputusan berdasarkan kondisi tertentu. Menguasai penggunaan if, else, elseif, switch, dan operator ternary akan membantu Anda menulis kode yang lebih fleksibel dan dinamis.

Perulangan

Perulangan digunakan untuk mengeksekusi blok kode yang sama berulang kali dengan jumlah iterasi yang telah ditentukan atau selama kondisi tertentu terpenuhi. PHP mendukung beberapa jenis perulangan, seperti for, while, do-while, dan foreach. Berikut adalah penjelasan dan contoh untuk masing-masing:

1. Perulangan for

Perulangan for biasanya digunakan ketika Anda tahu berapa kali blok kode harus diulang. Sintaks dari perulangan for melibatkan tiga bagian utama: inisialisasi, kondisi, dan pengubahan nilai.

```
<?php
for ($i = 0; $i < 5; $i++) {
    echo "Nomor: $i <br>";
}
```

Penjelasan:

\$i = 0: Inisialisasi variabel penghitung.

\$i < 5: Kondisi yang harus dipenuhi untuk melanjutkan perulangan.

\$i++: Pengubahan nilai variabel penghitung setelah setiap iterasi.

2. Perulangan while

Perulangan while digunakan ketika Anda ingin mengulangi blok kode selama kondisi tertentu bernilai benar (true).

```
<?php
$i = 0;
while ($i < 5) {
    echo "Nomor: $i <br>";
    $i++;
}
?>
```

Penjelasan:

\$i < 5: Kondisi yang harus terpenuhi untuk melanjutkan perulangan.

\$i++: Pengubahan nilai variabel penghitung di dalam blok perulangan.

3. Perulangan do-while

Perulangan do-while mirip dengan while, tetapi kondisi diperiksa setelah blok kode dieksekusi sekali. Ini berarti blok kode dijalankan setidaknya sekali.

```
<?php
$i = 0;
do {
    echo "Nomor: $i <br>";
    $i++;
} while ($i < 5);
?>
```

4. Perulangan foreach

Perulangan foreach digunakan khusus untuk mengunjungi elemen-elemen dalam array. Ini sangat berguna untuk memproses setiap elemen dalam array tanpa mengetahui jumlah elemen sebelumnya.

```
<?php
$buah = array("Apel", "Pisang", "Jeruk");

foreach ($buah as $item) {
    echo "Buah: $item <br>";
}
```

Fungsi

Fungsi dalam PHP adalah blok kode yang dapat digunakan kembali yang melakukan tugas tertentu.

Fungsi membantu dalam mengorganisasi kode dan mengurangi redundansi, sehingga membuat kode lebih modular dan mudah dikelola. PHP memiliki fungsi bawaan (built-in) dan juga memungkinkan Anda untuk mendefinisikan fungsi sendiri (user-defined).

Untuk mendefinisikan fungsi di PHP, gunakan kata kunci function diikuti dengan nama fungsi dan sepasang tanda kurung. Blok kode yang merupakan tubuh fungsi diletakkan dalam tanda

kurung kurawal.

```
function namaFungsi() {
    // kode yang akan dieksekusi
}

<?php
function sapa() {
    echo "Hello, World!";
}

// Memanggil fungsi
sapa();
```

1. Fungsi dengan Parameter

Fungsi dapat menerima parameter yang memungkinkan Anda untuk mengirim informasi ke fungsi. Parameter didefinisikan di dalam tanda kurung setelah nama fungsi.

```
<?php
function sapaNama($nama) {
    echo "Hello, $nama!";
}

// Memanggil fungsi dengan parameter
sapaNama("John");
?>
```

2. Fungsi dengan Nilai Kembali (Return Value)

Fungsi dapat mengembalikan nilai menggunakan kata kunci return. Ini memungkinkan fungsi untuk mengirim hasil kembali ke pemanggilnya.

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

// Menyimpan hasil fungsi dalam variabel
$hasil = tambah(5, 3);
echo "Hasil penjumlahan: $hasil";
?>
```

3. Parameter Default

Anda dapat menetapkan nilai default untuk parameter fungsi. Ini berguna ketika parameter opsional dan tidak selalu harus diberikan nilai.

```
<?php
function sapaDenganDefault($nama = "Tamu") {
    echo "Hello, $nama!";
}

// Memanggil fungsi tanpa parameter
sapaDenganDefault(); // Output: Hello, Tamu!

// Memanggil fungsi dengan parameter
sapaDenganDefault("John"); // Output: Hello, John!
?>
```

Excercise

Fungsi dapat menerima parameter yang memungkinkan Anda untuk mengirim informasi ke fungsi. Parameter didefinisikan di dalam tanda kurung setelah nama fungsi.

1. Manakah dari pernyataan berikut yang benar untuk mendeklarasikan variabel di PHP?

- a. var \$nama = "John";
- b. \$nama = "John";
- c. let \$nama = "John";
- d. string \$nama = "John";
- e. declare \$nama = "John";

2. Bagaimana cara mendeklarasikan tipe data string di PHP?

- a. \$teks = Hello;
- b. \$teks = 123;
- c. \$teks = true;
- d. \$teks = 'Hello';
- e. \$teks = Hello World;

3. Apa hasil dari kode berikut ini?

```
<?php
$namaDepan = "andi";
$namaBelakang = "baso";
$namaLengkap = $namaDepan . " " . $namaBelakang;
echo $namaLengkap;
?>
```

- a. andi
- b. andibaso
- c. andi. baso
- d. baso
- e. andi baso

4. Bagaimana cara mendeklarasikan tipe data boolean di PHP?

- a. \$benar = 1;
- b. \$benar = true;
- c. \$benar = Yes;
- d. \$benar = "true";
- e. \$benar = 'true';

5. Apa output dari kode berikut ini?

```
<?php
$nilai = 70;
if ($nilai >= 90) {
    echo "A";
} elseif ($nilai >= 80) {
    echo "B";
} elseif ($nilai >= 70) {
    echo "C";
} else {
    echo "D";
}
?>
```

- a. A
- b. B
- c. C
- d. D

6. Apa output dari kode berikut ini?

```
<?php
    $warna = "merah";
    switch ($warna) {
        case "biru":
            echo "Warna biru";
            break;
        case "merah":
            echo "Warna merah";
            break;
        case "hijau":
            echo "Warna hijau";
            break;
        default:
            echo "Warna tidak diketahui";
    }
?>
```

- a. Warna biru
- b. Warna hijau
- c. Tidak ada output
- d. Warna merah
- e. Warna tidak diketahui

7. Apa output dari kode berikut ini?

```
<?php
    $umur = 18;
    $status = ($umur >= 18) ? "dewasa" : "anak-anak";
    echo $status;
?>
```

- a. dewasa
- b. anak-anak
- c. 18
- d. dewasa anak-anak
- e. Tidak ada output

8. Apa output dari kode berikut ini?

```
<?php
    $buah = array("apel", "pisang", "jeruk");
    foreach ($buah as $item) {
        echo $item . " ";
    }
?>
```

- a. apel
- b. pisang
- c. jeruk
- d. apel pisang jeruk
- e. Tidak ada output

9. Manakah dari pernyataan berikut yang benar untuk mendefinisikan fungsi di PHP?

- a. define function sapa() { echo "Hello"; }
- b. function sapa { echo "Hello"; }
- c. function sapa() { echo "Hello"; }
- d. def sapa() { echo "Hello"; }
- e. create function sapa() { echo "Hello"; }

10. Apa output dari kode berikut ini?

```
<?php
    function sapa($nama) {
        echo "Hello, $nama!";
    }
    sapa("John");
?>
```

- | | | |
|-----------|----------|-----------------|
| a. Hello | c. John | e. Hello, John! |
| b. Hello, | d. Error | |