

Apa Itu Dinamis?

Web dinamis adalah jenis situs web yang menggunakan sisi server untuk menghasilkan konten yang berubah secara dinamis sesuai dengan permintaan pengguna. Ini berarti bahwa konten yang ditampilkan kepada pengguna dapat berubah berdasarkan interaksi mereka atau data yang diterima oleh server.

Apa Itu PHP?

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang umum digunakan untuk pengembangan web dinamis. PHP bekerja di sisi server, yang berarti kode PHP dijalankan di server web dan hasilnya dikirimkan ke browser pengguna dalam bentuk HTML. Dengan PHP, Anda dapat membuat halaman web yang menanggapi input pengguna, berinteraksi dengan basis data, menghasilkan konten dinamis, dan melakukan banyak tugas lainnya yang diperlukan untuk mengembangkan aplikasi web yang dinamis dan interaktif.

Program pertama PHP

Cara

1. Buka XAMPP, kemudian klik tombol start pada `apache`
2. Open file manager
3. Open di Drive D
4. Kemudian Open folder XAMPP
5. Lalu open folder htdocs
6. Buat folder baru di dalam htdocs
7. Lalu open folder yang sudah anda buat
8. Open VSCode
9. Open folder yang sudah anda buat di VSCode
10. Buat file baru di VSCode dengan ekstensi `.php` di folder yang sudah anda buka

Berikut adalah beberapa konsep dasar dalam PHP:

Echo

Kutip satu

kutip satu hanya bisa membaca teks atau STRING, variabel dan string dipisahkan dengan tanda titik

Kutip dua

kutip dua bisa membaca nilai dari variabel

Komentar

Single line

php

//ini komentar 1 baris menggunakan dua slice

Multiple line

php

/

ini

komentar

multibaris

menggunakan satu slice dan bintang

/

Variabel & Konstanta

Apa itu web dinamis dan PHP?

Web dinamis

Web dinamis adalah jenis situs web yang dapat menghasilkan konten yang berubah atau disesuaikan secara dinamis berdasarkan permintaan pengguna. Konten dinamis dapat dihasilkan melalui interaksi dengan pengguna, pengambilan data dari basis data, integrasi dengan layanan eksternal, dan pemrosesan logika bisnis. Web dinamis menggunakan bahasa pemrograman server-side untuk menghasilkan dan mengelola konten dinamis ini.

Php

PHP (PHP: Hypertext Preprocessor) adalah salah satu bahasa pemrograman yang sering digunakan untuk mengembangkan web dinamis. PHP adalah bahasa pemrograman sisi server yang dirancang khusus untuk pengembangan web. Itu dapat diintegrasikan dengan HTML untuk menghasilkan konten dinamis. PHP berjalan di sisi server, yang berarti kode PHP dieksekusi di server web sebelum hasilnya dikirim ke browser pengguna.

Program pertama PHP

Cara

1. Buka XAMPP, kemudian klik tombol start pada 'apache'
2. Open file manager
3. Open di Drive D
4. Kemudian Open folder XAMPP
5. Lalu open folder htdocs
6. Buat folder baru di dalam htdocs
7. Lalu open folder yang sudah Anda buat
8. Open VSCode
9. Open folder yang sudah Anda buat di VSCode
10. Buat file baru di VSCode dengan ekstensi .php di folder yang sudah Anda buka

Echo & commentar

Echo

1. Penggunaan Dasar:
 - echo adalah perintah dasar dalam PHP yang digunakan untuk menampilkan teks atau nilai pada halaman web.
 - Anda dapat menggunakan echo untuk menampilkan teks statis atau untuk menampilkan nilai dari variabel atau ekspresi.
 - echo dapat menerima satu atau lebih argumen yang dipisahkan oleh tanda titik (.) untuk menggabungkan teks dan variabel.
2. Menampilkan Teks:
 - Anda dapat menggunakan echo untuk langsung menampilkan teks statis dalam tanda kutip ganda ("") atau tanda kutip tunggal (").
 - Contoh: `echo "Halo, dunia!";`
3. Menampilkan Nilai Variabel:
 - Anda dapat menggunakan echo untuk menampilkan nilai dari variabel dalam teks yang ditampilkan.
 - Gunakan operator konkatenasi (tanda titik) untuk menggabungkan teks dan variabel.
 - Contoh:

```
php
$nama = "John Doe";
echo "Halo, nama saya " . $nama . " .";
```
4. Menampilkan HTML:
 - Selain teks, Anda juga dapat menggunakan echo untuk menampilkan kode HTML di dalam skrip PHP.

- Uji coba berikut menunjukkan penggunaan echo untuk menampilkan elemen HTML:

```
php
```

```
echo "
```

```
Selamat Datang!
```

```
";
```

```
echo "
```

```
Ini adalah halaman web PHP.
```

```
";
```

5. Penggunaan Kutipan:

- Anda dapat menggunakan tanda kutip ganda (") atau tanda kutip tunggal (') untuk mengelilingi teks dalam echo.
- Misalnya, echo "Halo, nama saya 'John'.";

6. Menggunakan Pemisah:

- Saat menggunakan echo dengan beberapa argumen, Anda dapat menggunakan koma (,) sebagai pemisah antara argumen.
- Contoh: echo "Halo,", " nama saya ", \$nama, ".";

Materi echo pada PHP memungkinkan Anda untuk menampilkan teks, nilai variabel, dan bahkan kode HTML di dalam skrip PHP. Hal ini berguna dalam menghasilkan tampilan dinamis pada halaman web yang sedang Anda bangun.

Struktur dasar echo

1. : Adalah penanda awal dan akhir dari blok kode PHP. Semua kode PHP harus berada di antara penanda ini agar dapat dieksekusi dengan benar.
2. echo: Adalah perintah dalam PHP yang digunakan untuk menampilkan teks atau nilai ke dalam halaman web.
3. teks atau nilai yang ingin ditampilkan: Adalah bagian di mana Anda memasukkan teks statis, nilai variabel, atau ekspresi matematika yang ingin ditampilkan di halaman web.

Jadi, ketika kode tersebut dieksekusi, teks atau nilai yang ditentukan akan ditampilkan di halaman web sebagai hasil dari perintah echo.

Kode program

```
php
```

Hasil:

Halo, dunia!

Commentar

Dalam PHP, komentar digunakan untuk memberikan penjelasan, catatan, atau dokumentasi di dalam kode program. Komentar tidak akan dieksekusi oleh interpreter PHP, sehingga

komentar tidak akan mempengaruhi hasil program.

Berikut adalah beberapa hal yang perlu diketahui tentang komentar dalam PHP:

1. Komentar Satu Baris:

- Komentar satu baris dimulai dengan tanda // atau #.
- Semua teks setelah tanda tersebut hingga akhir baris diabaikan oleh interpreter PHP.
- Contoh:
php
// Ini adalah komentar satu baris
\$nama = "John Doe"; // Ini adalah komentar di samping kode

2. Komentar Multibaris:

- Komentar multibaris dimulai dengan / *dan diakhiri dengan* /.
- Semua teks di antara tanda tersebut diabaikan oleh interpreter PHP.
- Komentar multibaris dapat mencakup beberapa baris kode.
- Contoh:
php
/
Ini adalah komentar multibaris.
Ini dapat mencakup beberapa baris kode.
/
\$umur = 25;

3. Komentar Dokumentasi:

- Komentar dokumen atau DocBlock digunakan untuk memberikan dokumentasi formal pada fungsi, kelas, atau metode.
- Komentar dokumen dimulai dengan /* *dan diakhiri dengan* /.
- Komentar ini sering digunakan dengan format khusus seperti PHPDoc untuk menghasilkan dokumentasi otomatis.
- Contoh:
php
/**
Fungsi ini menghitung jumlah dua angka.

@param int \$a Angka pertama
@param int \$b Angka kedua * @return int Hasil penjumlahan * /function tambah(\$a, \$b) {
 return \$a + \$b;
}

4. Tujuan Penggunaan Komentar:

- Memberikan penjelasan tentang tujuan dan fungsionalitas kode.
- Membantu dalam pemeliharaan dan pemahaman kode untuk pengembang dan anggota tim lainnya.

- Menyembunyikan atau menonaktifkan sebagian kode untuk uji coba atau sementara.
- Meninggalkan catatan atau pesan kepada diri sendiri atau pengembang lain tentang kode tertentu.

Penggunaan komentar yang baik sangat penting dalam pemrograman untuk menjaga kejelasan dan keberlanjutan kode. Dengan menggunakan komentar yang tepat, Anda dapat meningkatkan kemudahan pemeliharaan dan kolaborasi dalam pengembangan perangkat lunak.

Variable, const, operator

Variable

Variabel pada PHP merupakan suatu wadah atau tempat untuk menyimpan data. Variabel digunakan untuk menyimpan nilai-nilai yang akan digunakan dalam program PHP. Anda dapat memberikan nama pada variabel untuk mengidentifikasinya, dan kemudian menggunakan nama tersebut untuk mengakses nilai yang disimpan di dalamnya.

Dalam PHP, variabel bersifat dinamis, yang berarti Anda tidak perlu mendeklarasikan tipe data variabel sebelum menggunakannya. PHP akan secara otomatis menentukan tipe data variabel berdasarkan nilai yang Anda berikan.

Simbol

 pada PHP digunakan untuk menandakan sebuah variabel. Dalam PHP, setiap kali Anda ingin membuat variabel, Anda harus memulai dengan simbol \$ diikuti dengan nama variabelnya.

Contoh penggunaan variabel pada PHP:

php

```
nama = "JohnDoe"; // Variabel $namadengan nilai "JohnDoe"
umur = 25; // Variabel $umurdengan nilai 25
gaji = 2500.50; // Variabel $gajidengan nilai 2500.50
isActive = true; // Variabel $isActive dengan nilai true
```

Const

Dalam PHP, const adalah kata kunci yang digunakan untuk mendefinisikan konstanta. Konstanta adalah nilai yang tidak dapat diubah selama eksekusi program. Setelah konstanta didefinisikan, nilainya tidak dapat diubah atau dilakukan penugasan ulang.

- Konstanta didefinisikan menggunakan kata kunci const diikuti dengan nama konstanta dan nilainya.
- Nama konstanta biasanya ditulis dalam huruf besar (uppercase) untuk membedakan dengan variabel biasa.
- Nilai konstanta dapat berupa bilangan, string, boolean, atau ekspresi konstan lainnya.

Contoh:

php

```
const PI = 3.14;  
const NAMA_SITUS = "Website ABC";  
const STATUS_AKTIF = true;
```

Operator

Operator dalam PHP adalah simbol atau tanda yang digunakan untuk melakukan operasi atau manipulasi pada nilai atau variabel. Operator memungkinkan Anda untuk melakukan perhitungan matematika, perbandingan, penggabungan teks, dan operasi logika dalam program PHP.

Berikut ini beberapa jenis operator yang tersedia dalam PHP:

1. Operator Aritmatika:

- Operator aritmatika digunakan untuk melakukan operasi matematika seperti penjumlahan, pengurangan, perkalian, pembagian, modulus, dan sebagainya.
- Contoh:
php
`a = 10; b = 5;
c = $a + $b; // Penjumlahan
d = a - $b; // Pengurangan
e = a * $b; // Perkalian
f = a / $b; // Pembagian
g = $a % $b; // Modulus (sisanya)`

2. Operator Penugasan:

- Operator penugasan digunakan untuk memberikan nilai ke variabel.
- Contoh:
php
`a = 10; // Penugasan nilai 10 ke variabel $a
b += 5; // Penugasan nilai $b + 5 ke variabel $b (sama dengan $b = $b + 5)`

3. Operator Perbandingan:

- Operator perbandingan digunakan untuk membandingkan dua nilai dan menghasilkan nilai kebenaran (true atau false).
- Contoh:
php
`a = 10; b = 5;
c = $a == $b; // Sama dengan
d = a != $b; // Tidak sama dengan
e = a > $b; // Lebih besar dari
f = a < $b; // Lebih kecil dari
g = a >= $b; // Lebih besar dari atau sama dengan
h = $a <= $b; // Lebih kecil dari atau sama dengan`

4. Operator Logika:

- Operator logika digunakan untuk menggabungkan atau memanipulasi nilai kebenaran (true atau false).
- Contoh:
php
`a = true; b = false;
c = $a && $b; // Logika AND
d = a || $b; // Logika OR
e = !$a; // Logika NOT`

5. Operator String:

- Operator string digunakan untuk menggabungkan atau memanipulasi teks.
- Contoh:
php
`a="//Hello, //";b = "World!";`
`$c = $a . $b; // Penggabungan teks`

6. Operator Lainnya:

- Selain operator-operator di atas, PHP juga memiliki operator lain seperti operator increment (++) dan decrement (--), operator ternary (?:), operator array, dan lain-lain.

Operator dalam PHP memungkinkan Anda untuk melakukan berbagai operasi dan manipulasi pada data, dan memungkinkan logika dan fungsionalitas yang lebih kompleks dalam program Anda. Dalam pengembangan PHP, pemahaman tentang operator sangat penting untuk membuat kode yang efisien dan efektif.

`<?php`

```
//komentar 1 baris menggunakan dua slice
/*
ini
komentar
multibaris
menggunakan satu slice dan bintang
*/

//variabel
$meja = 30;
$tk_kelas = "XI";
$ketua_kelas = "Juli";
$wali_kelas = "Saleh";
$ketua_gank = "bombom";
$ketua_gank = "Rahmat"; //pengubahan nilai variabel

//konstanta
const KepSek = "Herwelis";
define('kelas', 'RPL 1');

//echo berfungsi untuk menampilkan
/*kutip satu hanya bisa membaca teks atau STRING, variabel dan string
dipisahkan dengan tanda titik
*/
echo 'jumlah meja di kelas : ' . $meja . ' buah';
echo "<br>";

//kutip dua bisa membaca nilai dari variabel
echo "Sholat gess, nabilang Pak $wali_kelas dan $ketua_kelas";
```



```
echo "<br>";

//kutip satu pada kata diracca', dibaca string disini
echo "Klo tidak, diracca' sama ketua gank $ketua_gank";
echo "<br>";
echo 'KepSeknya Pak ' . KepSek;
echo "<br>";
echo 'Kelasnya ' . $tk_kelas . ' ' . kelas;
?>
```

Hasil

"Cuplikan layar 2024-06-01 012834.png" could not be found.

Coditional Statement

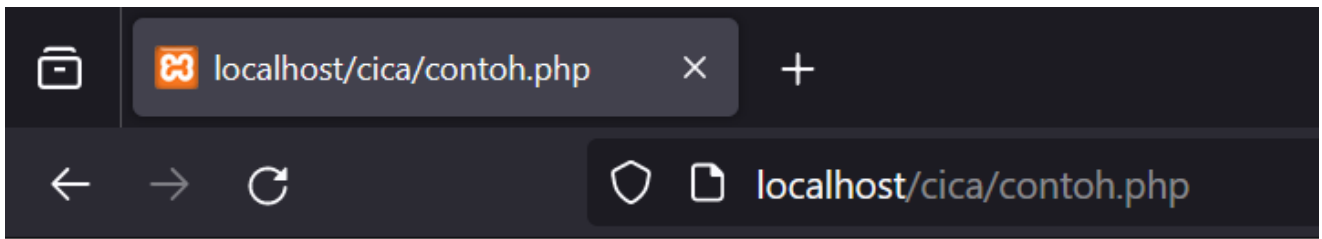
IF

Pernyataan if digunakan untuk mengevaluasi sebuah kondisi dan menjalankan blok kode jika kondisi tersebut benar. Jika kondisi yang dievaluasi adalah salah, maka blok kode tersebut tidak akan dijalankan.

Program

```
<?php
$umur = 17;
if ($umur >= 20) {
    echo "Anda sudah dewasa";
}
?>
```

Hasil



Anda masih di bawah umur

Penjelasan

Program ini mengevaluasi variabel `$umur` dan mencetak pesan "Anda sudah dewasa" jika umur lebih besar atau sama dengan 20.

Analisis

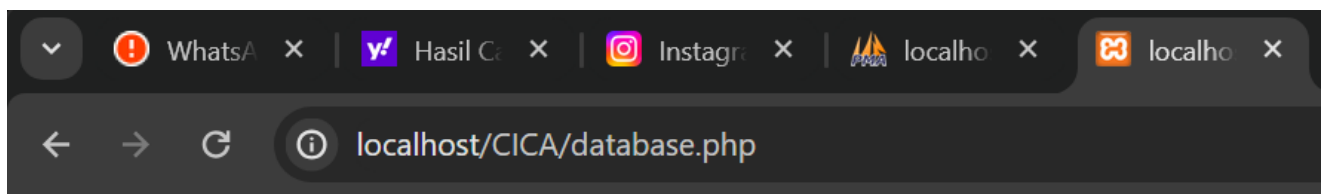
Karena nilai variabel `$umur` adalah 17, kondisi `if` tidak terpenuhi, sehingga pesan tidak dicetak.

IF-ELSE

Program

```
<?php
$umur = 17;
if ($umur >= 20) {
    echo "Anda sudah dewasa";
} else {
    echo "Anda masih di bawah umur";
}
?>
```

Hasil



koneksi aman

Berikut mobil-mobil beserta pemiliknya

1. DD4455JK : DILLA
2. DD1310JH : JIMIN
3. DD2812HF : HASIZA
4. DD5853PT : AL

Penjelasan

Program ini mengevaluasi variabel \$umur dan mencetak pesan "Anda sudah dewasa" jika umur lebih besar atau sama dengan 20. Jika umur kurang dari 20, pesan "Anda masih di bawah umur" akan dicetak.

Analisis

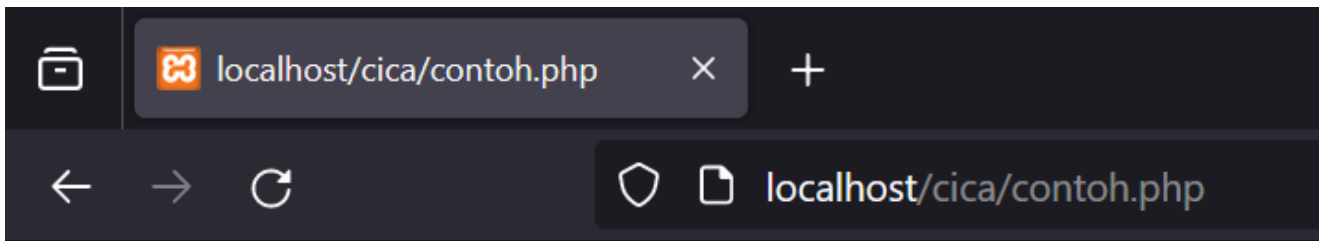
Karena nilai variabel \$umur adalah 17, maka blok kode di dalam else yang mencetak "Anda masih di bawah umur" akan dieksekusi.

IF-ELSE IF-ELSE

Program

```
<?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";
}
?>
```

Hasil



Nilai Anda B

Penjelasan

Program ini mengevaluasi variabel `$nilai` dan mencetak grade berdasarkan rentang nilai. Misalnya, jika nilai lebih dari atau sama dengan 90, maka grade A dicetak.

Analisis

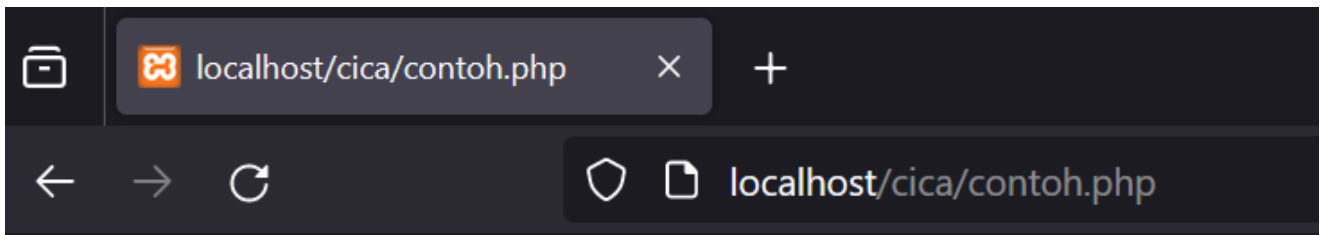
Karena nilai variabel `$nilai` adalah 85, maka blok kode yang dicetak adalah "Nilai Anda B", karena nilai tersebut memenuhi kondisi kedua.

SWITCH-CASE

Program

```
<?php
$warna = "merah";
switch ($warna) {
    case "merah":
        echo "Warna favorit Anda adalah merah";
        break;
    case "biru":
        echo "Warna favorit Anda adalah biru";
        break;
    default:
        echo "Warna favorit Anda bukan merah atau biru";
}
?>
```

Hasil



Warna favorit Anda adalah biru

Penjelasan

Program ini mengevaluasi variabel \$warna dan mencetak pesan berdasarkan nilai \$warna. Dalam kasus ini, karena nilai \$warna adalah "biru", maka pesan "Warna favorit Anda adalah biru" akan dicetak.

Analisis

Karena nilai variabel \$warna adalah "biru", maka blok kode yang dicetak adalah "Warna favorit Anda adalah biru".

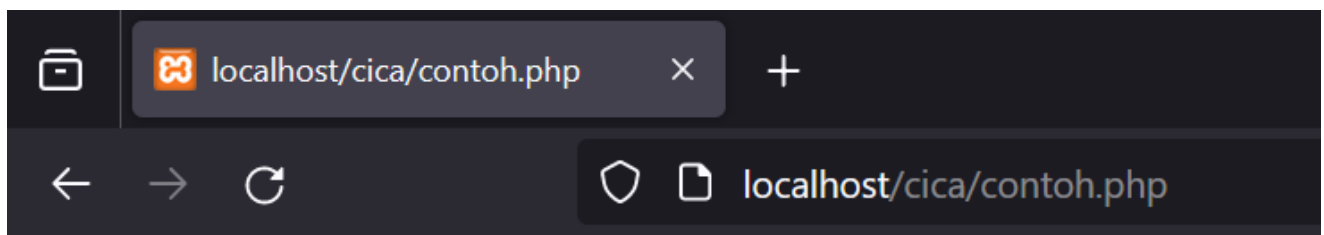
Array

Array 1 Dinamis

Program

```
<?php
$buah = array("Alpukat", "Durian", "Apel");
foreach ($buah as $item) {
    echo $item . "<br>";
}
?>
```

Hasil



Alpukat
Durian
Apel

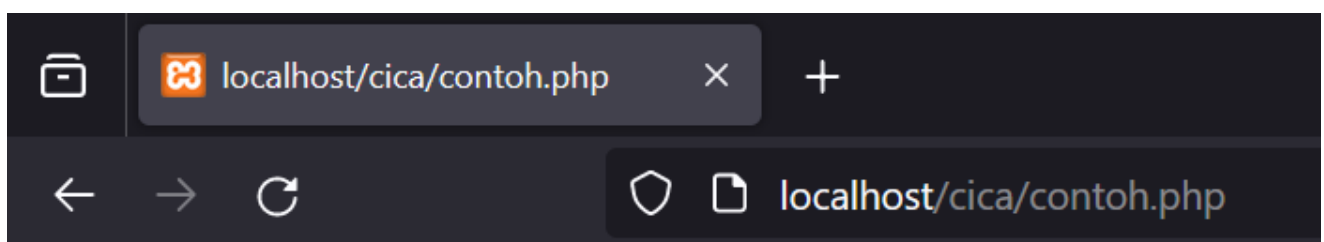
Penjelasan

Program ini membuat sebuah array dinamis dengan isi "apel", "pisang", dan "jeruk". Kemudian menggunakan loop foreach untuk menampilkan setiap nilai dalam array.

Array Asosiatif

```
<?php
$umur = array("Hasiza" => 17, "Dilla" => 16 , "Doe" => 35);
echo "Umur Hasiza: " . $umur['Hasiza'];
?>
```

Hasil



Umur Hasiza: 17

Penjelasan

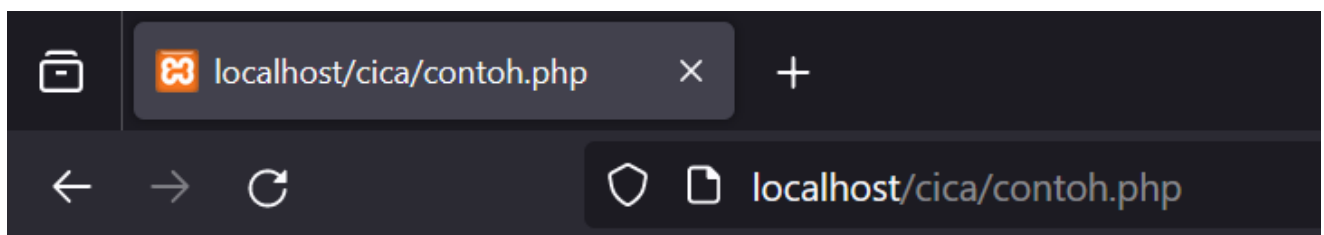
Program ini membuat sebuah array asosiatif dengan nama sebagai kunci dan umur sebagai nilainya. Kemudian mencetak umur John dengan menggunakan kunci 'John'.

Array Multidimensi

Program

```
<?php
$Siswa = array(
    array("Hasiza", 17, "RPL"),
    array("Dilla", 16, "Perkantoran"),
    array("Specky",15, "RPL")
);
echo "Nama: " . $Siswa[0][0] . ", Umur: " . $Siswa[0][1] . ", Jurusan: " .
$Siswa[0][2];
?>
```

Hasil



Nama: Hasiza, Umur: 17, Jurusan: RPL

Penjelasan

Program ini membuat sebuah array multidimensi yang menampung data mahasiswa. Setiap elemen dalam array adalah array sendiri yang berisi nama, umur, dan jurusan mahasiswa. Kemudian mencetak detail mahasiswa pertama.

Var-Dump

Penjelasan

```
<?php
$variabel = "Siti Nur Hasiza.A";
```

```
var_dump($variabel);  
?>
```

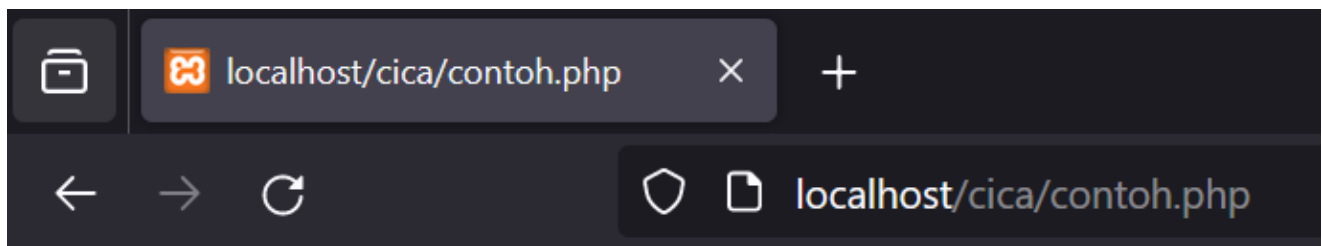
Struktur

```
<?php  
$variabel = "contoh";  
var_dump($variabel);  
?>
```

Program

```
<?php  
$umur = 17;  
$nama = "Hasiza";  
$nilai = array(80, 85, 90);  
var_dump($umur, $nama, $nilai);  
?>
```

Hasil



string(17) "Siti Nur Hasiza.A"

Analisis

Fungsi `var_dump` sangat berguna dalam debugging karena memberikan informasi rinci tentang variabel. Dengan melihat hasilnya, kita dapat memahami tipe data dan nilai variabel dengan lebih baik.

Kesimpulan Program

Dengan menggunakan `var_dump`, pengembang dapat mengeksplorasi struktur data dalam variabel dan mengidentifikasi masalah atau kesalahan dalam kode dengan lebih efektif.

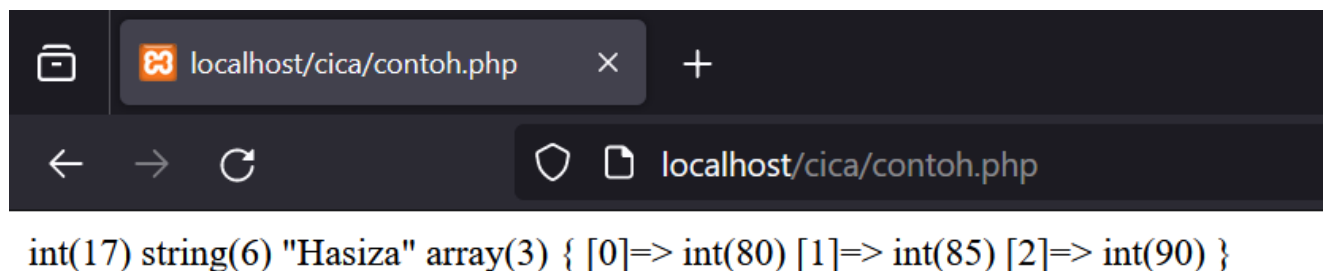
Looping(Perulangan)

FOR

Program

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

Hasil



Penjelasan

Program ini menggunakan loop for untuk mencetak angka dari 0 hingga 4.

Analisis

Loop for adalah cara yang efisien untuk melakukan iterasi berulang dengan jumlah iterasi yang sudah diketahui sebelumnya.

Kesimpulan

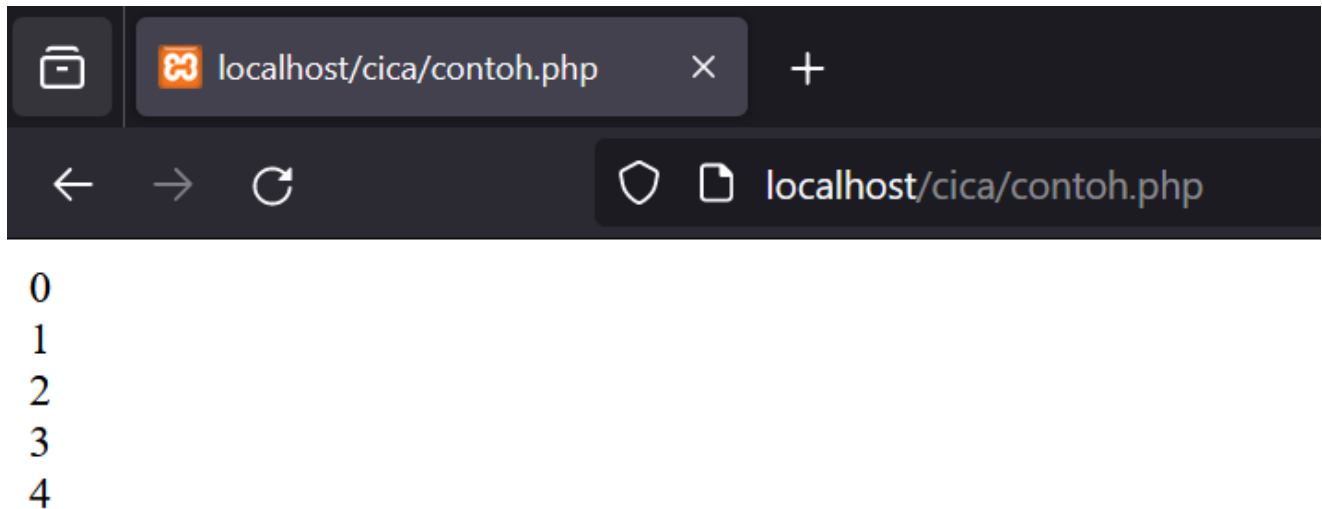
Loop for sangat berguna dalam situasi di mana kita perlu melakukan iterasi sejumlah tertentu.

While

Program

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

Hasil



Penjelasan

Program ini menggunakan loop while untuk mencetak angka dari 0 hingga 4.

Analisis

Loop while berguna ketika kita perlu melakukan iterasi berdasarkan kondisi tertentu.

Kesimpulan

Loop while dapat digunakan untuk melakukan iterasi selama kondisi tertentu terpenuhi.

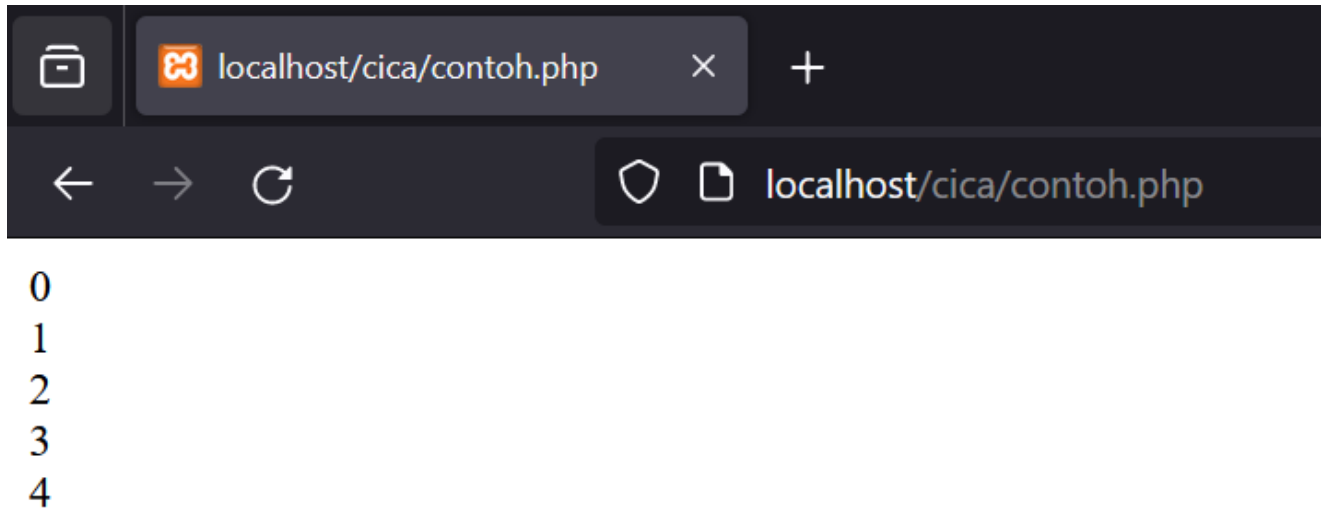
Do-While

Program

```
<?php
$i = 0;
do {
    echo $i . "<br>";
    $i++;
}
```

```
} while ($i < 5);  
?>
```

Hasil



Penjelasan

Program ini menggunakan loop do-while untuk mencetak angka dari 0 hingga 4.

Analisis

Loop do-while berguna ketika kita ingin menjalankan blok kode setidaknya sekali sebelum mengevaluasi kondisi.

Kesimpulan

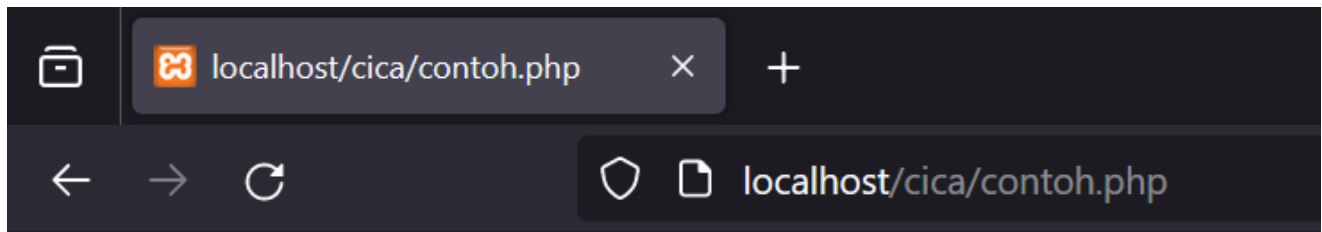
Loop do-while berguna dalam situasi di mana kita perlu menjalankan blok kode minimal satu kali.

Foreach

Program

```
<?php  
$buah = array("Alpukat", "Durian", "Apel");  
foreach ($buah as $item) {  
    echo $item . "<br>";  
}  
?>
```

Hasil



Alpukat
Durian
Apel

Penjelasan

Program ini menggunakan loop foreach untuk mencetak setiap nilai dalam array buah.

Analisis

Loop foreach sangat berguna ketika kita perlu melakukan iterasi melalui setiap elemen dalam array atau objek.

Kesimpulan

Loop foreach memudahkan kita dalam melakukan iterasi melalui array atau objek tanpa perlu mengkhawatirkan indeks atau kunci.

FUNCTION

Penjelasan

Fungsi di PHP adalah blok kode yang dapat digunakan kembali untuk melakukan tugas-tugas tertentu. Fungsi dapat menerima parameter (input) dan dapat mengembalikan nilai (output).

Struktur

```
function namaFungsi($parameter1, $parameter2, ...) {  
  
    // Blok kode yang akan dieksekusi  
  
    // Dapat berisi berbagai jenis pernyataan (statement)  
}
```

```
        return nilai_yang_dikembalikan;

    }
}
```

Program

```
<?php

function tambah($a, $b) {

    $hasil = $a + $b;

    return $hasil;

}

$angka1 = 5;

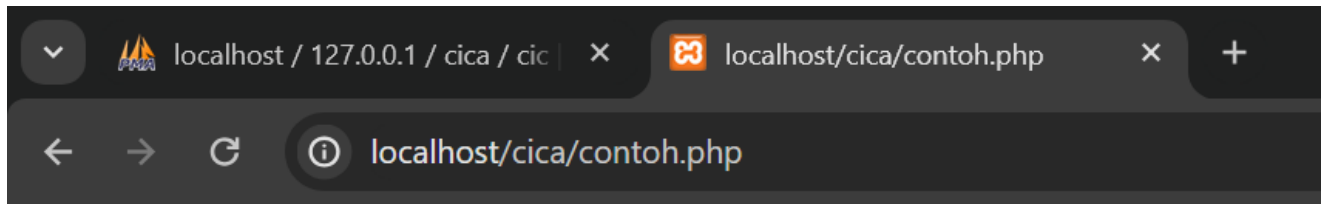
$angka2 = 3;

$hasil_penjumlahan = tambah($angka1, $angka2);

echo "Hasil penjumlahan: " . $hasil_penjumlahan;
```

?>

Hasil



Hasil penjumlahan: 8

Analisis

1. Program ini mendefinisikan sebuah fungsi bernama `tambah()` yang menerima dua parameter `$a` dan `$b`.
2. Dalam fungsi `tambah()`, variabel `$hasil` akan menyimpan hasil penjumlahan antara `$a` dan `$b`.
3. Fungsi `tambah()` mengembalikan nilai yang disimpan dalam `$hasil`.
4. Di luar fungsi `tambah()`, program mendeklarasikan dua variabel `$angka1` dan `$angka2` dengan nilai 5 dan 3.
5. Program memanggil fungsi `tambah()` dengan argumen `$angka1` dan `$angka2`, dan menyimpan hasilnya dalam variabel `$hasil_penjumlahan`.
6. Akhirnya, program mencetak pesan "Hasil penjumlahan: " diikuti dengan nilai yang tersimpan dalam `$hasil_penjumlahan`.

Kesimpulan

1. Program ini mengdemonstrasikan penggunaan fungsi dalam PHP untuk melakukan operasi penjumlahan.
2. Fungsi `tambah()` bertindak sebagai sebuah modul yang dapat digunakan kembali untuk melakukan penjumlahan antara dua angka.
3. Penggunaan fungsi memungkinkan program menjadi lebih modular, reusable, dan mudah dibaca dan dipelihara.

4. Pemanggilan fungsi `tambah()` dengan argumen `$angka1` dan `$angka2` menunjukkan fleksibilitas fungsi, di mana Anda dapat memanggil fungsi dengan nilai yang berbeda setiap kali.
5. Secara keseluruhan, program ini mendemonstrasikan pemahaman dasar tentang fungsi dalam pemrograman PHP, yang merupakan konsep penting bagi pelajar kelas 11 SMK jurusan RPL.

PHP Form

GET Method

Penjelasan

Metode GET adalah salah satu cara untuk mengirimkan data dari sisi klien (browser) ke sisi server (PHP) melalui URL. Data yang dikirimkan melalui metode GET akan terlihat dalam URL.

Program

FORM

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Document</title>
```

```
</head>
```

```
<body>
```

`<!-- Pada atribut action, kalian tuliskan nama file php yang bertugas untuk mengelola atau menangkap data dari form tersebut. -->`

```
<form action="proses_get.php" method="GET">
```

```
<input type="text" name="nama_lengkap" placeholder="Masukkan nama">
```

```
<input type="number" name="umur" placeholder="Masukkan umur"> <br>
```

```
<button type="submit">Kirim</button>
```

```
</form>
```

```
</body>
```

```
</html>
```

PROSES

```
<?php
```

```
// Key dari array-nya, sesuai dengan nama dari atribut name di setiap input-nya
```

```
$nama = $_GET["nama_lengkap"];
```



```
$umur = $_GET["umur"];
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>XI RPL 1 - GET</title>
```

```
</head>
```

```
<body>
```

```
<p>Nama anda
```

```
<!-- Ini adalah versi singkatnya dari php echo,
```

```
yang fungsinya untuk menampilkan data -->
```

```
<?= $nama ?>
```

```
</p>
```

```
<p>Umur anda <?= $umur ?> tahun</p>
```

```
</body>
```

```
</html>
```

Hasil

“Cuplikan layar 2024-06-01 113425.png” could not be found.

“Cuplikan layar 2024-06-01 020002.png” could not be found.

POST Method

Penjelasan

Metode POST adalah salah satu cara untuk mengirimkan data dari sisi klien (browser) ke sisi server (PHP) melalui body request HTTP. Data yang dikirimkan melalui metode POST tidak akan terlihat dalam URL.

Program

FORM

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Document</title>
```

```
</head>
```

```
<body>
```

```
<!-- Pada atribut action, kalian tuliskan nama file php yang bertugas untuk  
mengelola atau menangkap data dari form tersebut. -->
```

```
<form action="proses_post.php" method="POST">
```

```
<input type="text" name="nama_lengkap" placeholder="Masukkan nama">
```

```
<input type="number" name="umur" placeholder="Masukkan umur">
```

```
<input type="password" name="password" placeholder="Masukkan password">
```

```
<br>

<button type="submit">Kirim</button>

</form>

</body>

</html>
```

PROSES

```
<?php

// Key dari array-nya, sesuai dengan nama dari atribut name di setiap input-nya

// $nama = $_GET["nama"];

$umur = $_POST["umur"];

var_dump($_POST);

?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <title> XI RPL 1 - POST</title>
```

```
</head>
```

```
<body>
```

```
  <p>Nama anda <?= $_POST["nama_lengkap"] ?></p>
```

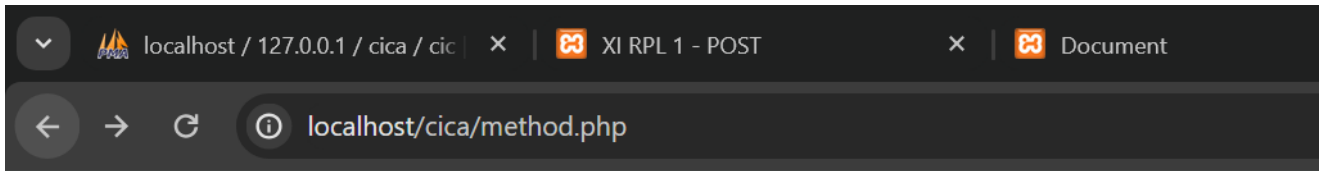
```
  <p>Umur anda <?= $umur ?> tahun</p>
```

```
  <p>Password anda aman!</p>
```

```
</body>
```

```
</html>
```

Hasil

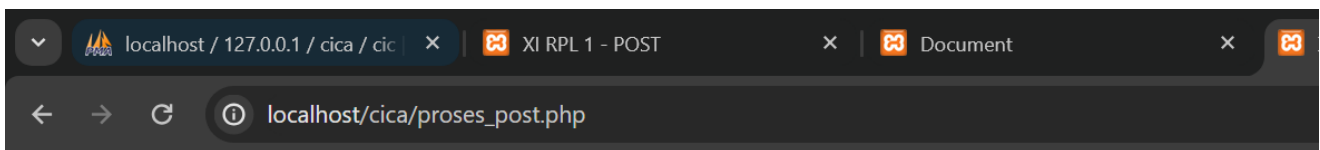


Masukkan nama

Masukkan umur

Masukkan password

Kirim



```
array(3) { ["nama_lengkap"]=> string(6) "Hasiza" ["umur"]=> string(2) "17" ["password"]=> string(5) "12345" }
```

Nama anda Hasiza

Umur anda 17 tahun

Password anda aman!