

LAPORAN PROYEK 1

EKSPRESI NOTASI ALGORITMA 2

EKSPRESI,VARIABEL,CARA CETAK



OLEH:

Muhammad ilham

(0082761073)

REKAYASA PERANGKAT LUNAK

SMK NEGERI 1 KARANG BARU

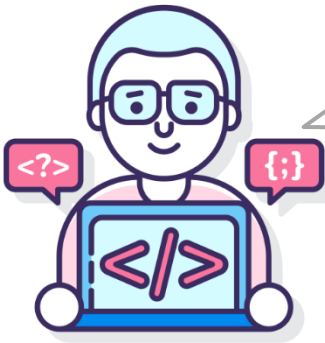
PEMERINTAH PROVINSI ACEH

2024



MATERI PERTEMUAN 5

Sintaks Dasar PHP yang Wajib Kamu Pahami

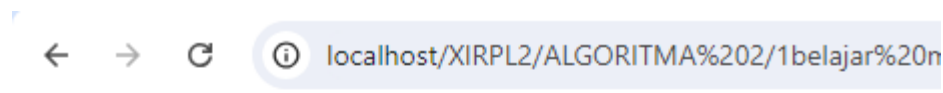


TAHUKAH KAMU...?

- “Bagaimana Struktur Program PHP yang Paling Dasar?”
- “Bagaimana Menulis Kode HTML dan PHP?”
- “Bagaimana Penulisan Statement dan Ekspresi?”
- “Bagaimana Aturan Penulisan Case PHP?”
- “Bagaimana Penulisan Komentar di PHP?”
- “Bagaimana Penulisan Blok Program?”

Kita sudah belajar membuat program **hello_world.php** dengan kode seperti ini:

```
<?php  
echo "Hello World!";  
?>
```



Hello World!
Hello NAMA SAYA MUHAMMAD ILHAM!

by. MUHAMMAD ILHAM

Ini adalah contoh program PHP yang paling sederhana.

Program tersebut hanya berfungsi untuk menampilkan teks **Hello World** saja.

Tapi...

Apa maksudnya **<?php** dan **?>** ?

Apa itu echo?

...dan kenapa harus ditulis seperti itu?

Apa itu sintak?

Sintak adalah aturan penulisan kode program.

Pada dasarnya setiap bahasa pemrograman itu sama, yang membedakan adalah sintak dan fitur.

Struktur Program PHP yang Paling Dasar

Ini adalah bentuk paling dasar program PHP:

```
<?php
```

```
echo "Hello World!";
```

Keterangan:

- **<?php** ini adalah pembuka program PHP. Pembuka ini wajib ada di setiap program PHP.
- **echo "Hello World!";** adalah sebuah statement atau perintah untuk menampilkan teks.

Tunggu dulu...

Kenapa tidak ditutup dengan **?>** ?

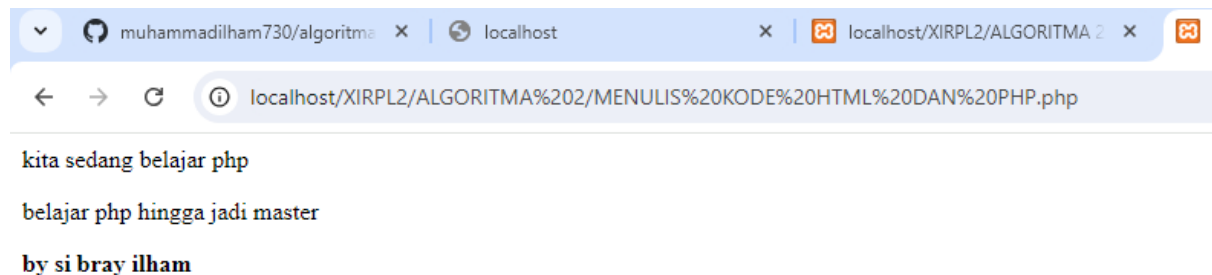
Tutup sebenarnya bersifat opsional. Tutup program dibutuhkan saat kita menggabungkan kode PHP dengan HTML.

Menulis Kode HTML dan PHP

Saat kita menulis kode PHP di dalam HTML, maka wajib hukumnya membuat tutup program.

Contoh:

```
<!DOCTYPE html>
<html>
  <head>
    <title><?php echo "Belajar PHP" ?></title>
  </head>
  <body>
    <?php
      echo "kita sedang belajar PHP<br>";
      echo "<p>Belajar PHP hingga jadi master</p>";
    ?>
  </body>
</html>
```



Perhatikan contoh di atas!

Apa yang akan terjadi bila kita menghapus tutup PHP (**?>**)?

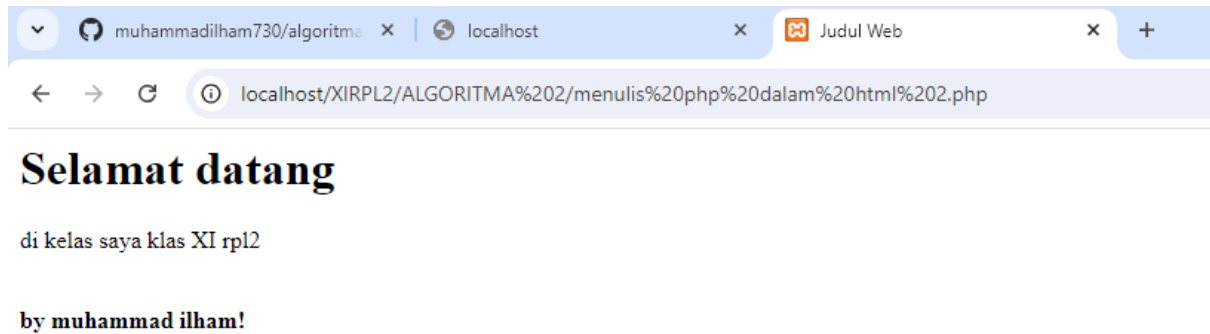
Tentunya program akan error.

Oya, PHP yang ditulis di dalam HTML, filenya harus disimpan dengan ekstensi **.php** bukan **.html** meskipun isinya HTML dan PHP.

Lalu ada juga yang menulis seperti ini:

```
<?php
```

```
echo "<html>";
echo "<head>";
echo "<title>Judul Web</title>";
echo "</head>";
echo "<body>";
echo "<h1>Selamat datang</h1>";
echo "</body>";
echo "</html>";
```



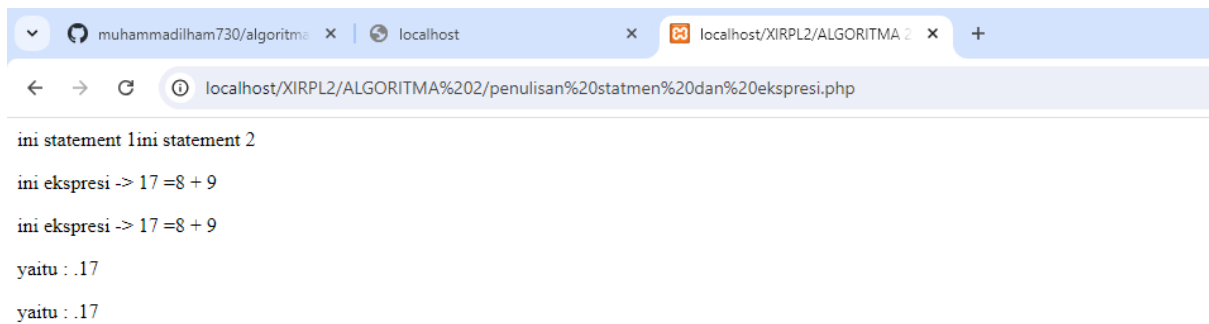
Nah kalau yang ini bisa tidak ditutup, karena kode HTML-nya ditulis di dalam sintak PHP. Masih bingung?

Penulisan Statement dan Ekspresi

Statement dan ekspresi adalah intruksi yang akan diberikan ke komputer. Setiap statement dan ekspresi di PHP harus diakhiri dengan titik koma (;).

Contoh:

```
<?php
echo "ini statement 1";
echo "ini statement 2";
$a = $b + $c;
```



The screenshot shows a web browser with three tabs: 'muhammadiyah730/algorithm', 'localhost', and 'localhost/XIRPL2/ALGORITMA'. The address bar shows the URL 'localhost/XIRPL2/ALGORITMA%202/penulisan%20statmen%20dan%20ekspresi.php'. The page content displays the output of a PHP script:

```
ini statement 1 ini statement 2
ini ekspresi -> 17 = 8 + 9
ini ekspresi -> 17 = 8 + 9
yaitu : .17
yaitu : .17
```

Gimana kalau kita lupa menuliskan titik koma? tentu programnya akan error.

Aturan Penulisan Case PHP

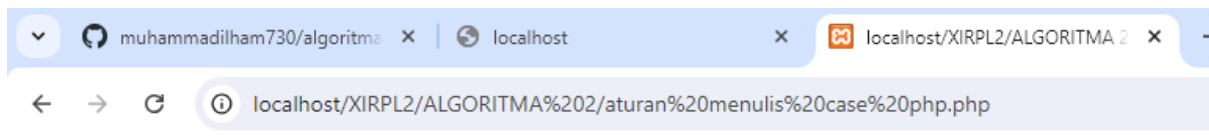
PHP adalah bahasa pemrograman yang bersifat *case sensitive*. Artinya, huruf besar (kapital) dan huruf kecil akan dibedakan.

Contoh:

```
<?php
```

```
$nama = "ahmadimuslim";
$NAMA = "dian";
```

```
?>
```



The screenshot shows a web browser with three tabs: 'muhammadiyah730/algorithm', 'localhost', and 'localhost/XIRPL2/ALGORITMA 2'. The address bar shows 'localhost/XIRPL2/ALGORITMA%202/aturan%20menulis%20case%20php.php'. The page content is as follows:

```
==== mencetak variabel ====  
  
nama saya adalah muhammad ilham  
nama kawan saya adalah ilham  
umur saya adalah 16  
coba pakai blokk program  
nama saya adalah muhammad ilham  
nama kawan saya adalah ilham  
umur saya adalah 16  
  
by muhammad ilham!
```

Variabel **\$nama** dan **\$NAMA** adalah dua variabel yang berbeda. Mereka tidak sama. Penulisan huruf besar dan kecil dalam program harus diperhatikan, karena bisa menyebabkan error bila salah.

Kita sering menemukannya, banyak yang salah tulis.

Contoh:

```
$umur = 19;
```

```
echo "Umur kita adalah $Umur";
```

Pada kode diatas, kita membuat variabel **\$umur** dengan huruf kecil. Lalu saat menggunakan variabel kita menggunakan **\$Umur**.

Jelas sekali ini akan menyebabkan error.

Untuk menghindari ini, kita harus konsisten dalam penamaan variabel dan konstanta.

Gunakan nama variabel dengan huruf kecil saja dan konstanta dengan huruf kapital.

Contoh:

```
const INI_KONSTANTA = 123;
```

```
$ini_variabel = 23;
```

```
$iniJugaVariabel = 49;
```

kok ada huruf kapital di **\$iniJugaVariabel**?

Ini namanya *camelCase*.

Jika variabel terdiri dari dua atau lebih suku kata, maka kita bisa memisahkannya dengan huruf kapital atau bisa juga dengan garis bawah (*underscore*).

Penulisan Komentar di PHP

Komentar adalah bagian yang tidak akan dieksekusi oleh komputer. Biasanya digunakan untuk keterangan, penjelasan, dan dokumentasi kode program.

Komentar di PHP dapat ditulis dengan dua cara:

1. Menggunakan tanda `//` untuk komentar satu baris;
2. Menggunakan tanda `/*` untuk komentar lebih dari satu baris.

Contoh:

```

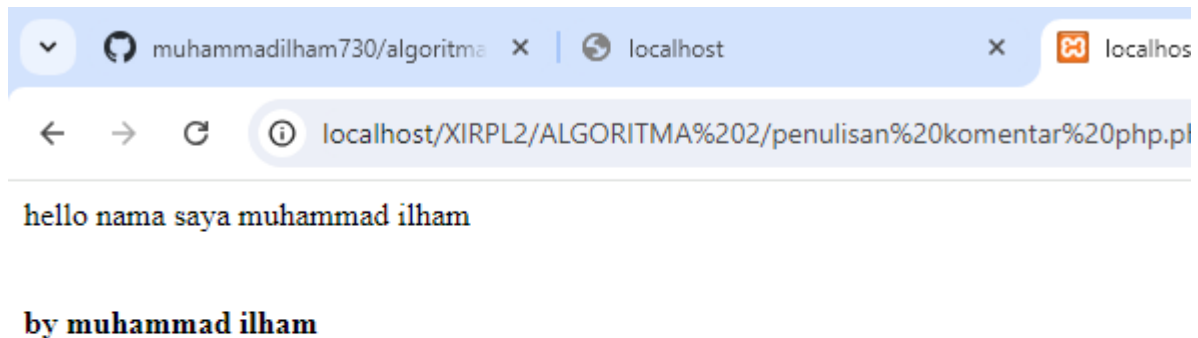
<?php

// ini adalah komentar
echo "Hello world";

/*
ini adalah komentar
yang lebih dari satu
baris
*/

?>

```



Penulisan Blok Program

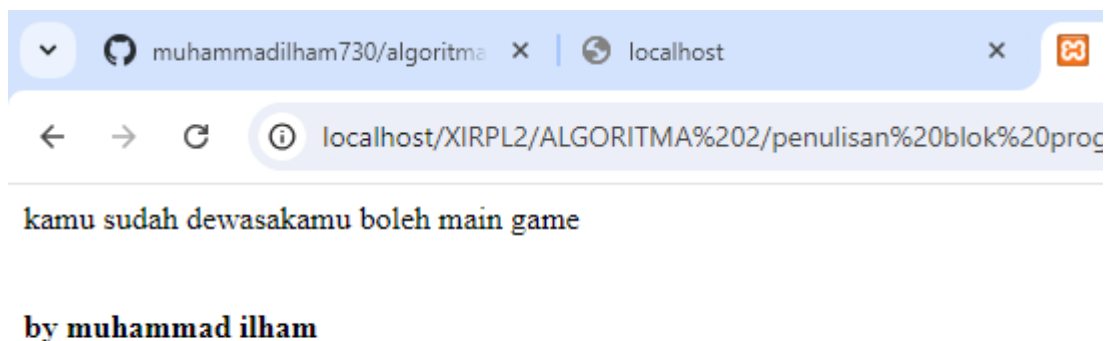
Blok program merupakan kumpulan dari statement dan ekspresi. Blok program di PHP dibungkus dengan kurung kurawal { ... }.

Contoh:

```

if ($umur > 18) {
    echo "Kamu sudah dewasa";
    echo "Selamat datang";
    echo "Kamu boleh minum kopi";
}

```



Program di atas adalah contoh blok kode *if* yang berisi tiga statement.

▪ LATIHAN 5

Kerjakan soal-soal berikut dengan baik dan benar!

- 1) Buatlah contoh bentuk paling sederhana dari program PHP..
- 2) Kapan membuat tutup program PHP, dan kapan tidak harus...
- 3) Apa yang dimaksud Statement dan ekspresi, dan bagaimana cara menuliskannya...
- 4) PHP adalah bahasa pemrograman yang bersifat case sensitive. Apa maksudnya...
- 5) Apa yang dimaksud dengan Komentar di PHP, dan bagaimana cara menuliskannya...
- 6) Apa yang dimaksud dengan Blok program, dan bagaimana cara menuliskannya...



MATERI PERTEMUAN 6

Cara Mencetak, Variabel Dan Tipe Data Pada PHP



TAHUKAH KAMU...?

PHP memiliki beberapa fungsi untuk mencetak teks ke layar:

- fungsi **echo()**;
- fungsi **print()**;
- fungsi **printf()**.

Apa saja perbedaan fungsinya...?

A. Variabel Pada PHP

Apa kamu pernah menemukan **x** dan **y** dalam pelajaran matematika?

Mereka berdua sebenarnya adalah variabel yang menyimpan sesuatu.

Kadang, kita sering diberi tugas untuk mencari tahu isi dari **x** dan **y**.

Contohnya:

jika $x + 3 = 5$, Berapakah **x**?

Variabel dalam pemrograman juga memiliki arti yang sama seperti dalam matematika.

Variabel adalah tempat kita menyimpan nilai sementara.

Variabel akan ada selama program dijalankan. Namun kita juga bisa menghapusnya dari memori.

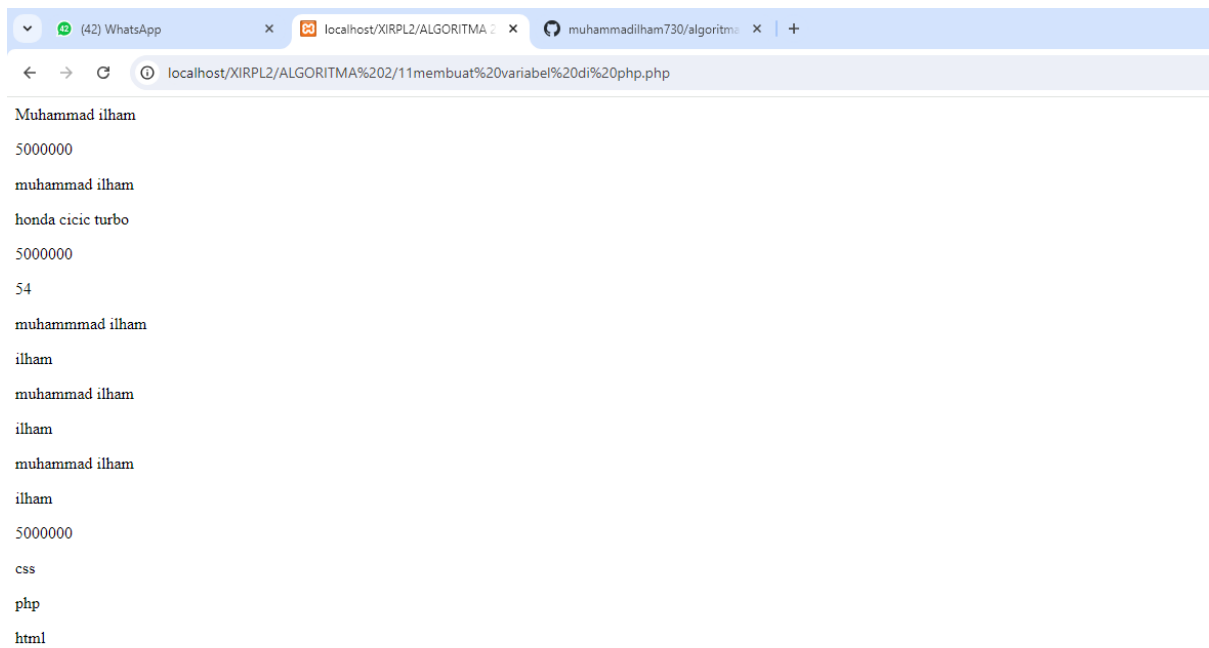
1. Membuat Variabel di PHP

Pada PHP, kita membuat variabel dengan tanda dolar (\$), lalu diikuti dengan nama variabelnya serta nilai yang ingin kita simpan.

Contoh:

```
<?php
```

```
$harga = 1000;
```



By ilham Activate Wi
Go to Settinas t

Kita baru saja membuat variabel bernama **\$harga** dengan isi **1000**.

Tanda sama dengan (=) adalah simbol atau operator yang digunakan untuk mengisi nilai ke variabel.

Mudah bukan?

Mari kita coba contoh yang lain:

```
<?php
```

```
$nama_barang = "Kopi C++";  
$harga = 4000;  
$stok = 40;
```

Oya, variabel juga dapat diisi ulang dengan nilai yang baru.

Contoh:

```
<?php
```

```
// membuat variabel baru  
$stok = 40;  
  
// mengisi ulang variabel dengan nilai baru  
$stok = 34;
```

Dalam membuat nama variabel ada 4 hal yang harus diperhatikan:

- 1) Awal dari nama variabel tidak boleh menggunakan angka dan simbol, kecuali underscore.

Contoh: ✗ Salah

```
$!nama = "";  
$46rosi = "";
```

Contoh: ✓ □ Benar

```
$nama = "";  
$rosi46 = "";
```

- 2) Nama variabel yang terdiri dari dua suku kata, bisa dipisah dengan underscore (_) atau menggunakan style camelCase.

Contoh:

```
$nama_barang = "Buku PHP";
```

```
$namaPembeli = "Ahmadi"; // ← menggunakan camelCase
```

- 3) Variabel harus diisi saat pembuatannya. Bila kita tidak ingin mengisi, cukup isi dengan nilai kosong.

Contoh:

```
$nama_barang = "";  
$namaPembeli = "";  
$harga = 0;
```

- 4) Nama variabel bersifat *Case Sensitive*, artinya huruf besar dan huruf kecil dibedakan.

Contoh: Tiga variabel ini akan dianggap berbeda.

```
$Belajar = "";  
$BELAJAR = "";  
$belajar = "";
```

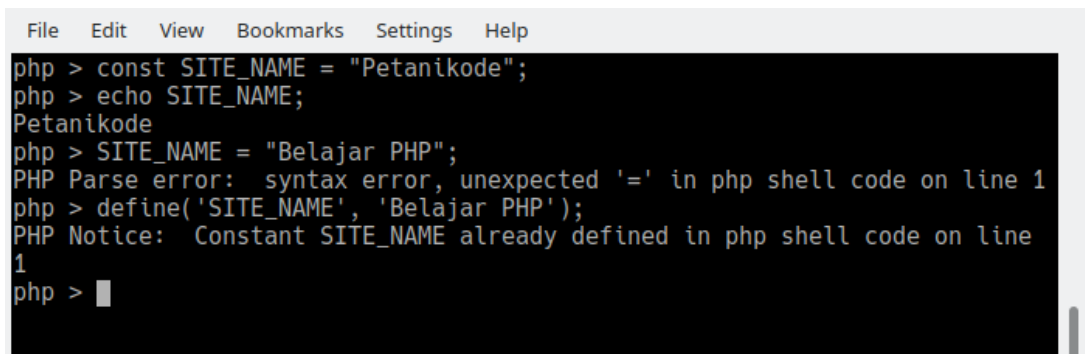
B. Konstanta Pada PHP

Konstanta seperti variabel. Dapat digunakan untuk menyimpan nilai, tapi tidak bisa diubah.

Contoh:

```
// kita punya konstanta dengan nilai 5  
const SEBUAH_NILAI = 5  
  
// lalu kita ubah menjadi 10  
SEBUAH_NILAI = 10 // <-- maka akan terjad error di sini
```

Mari kita buktikan:



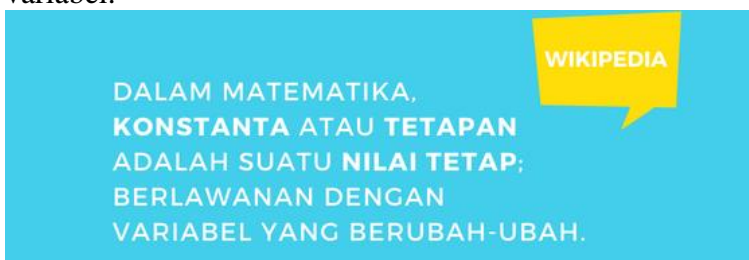
```
File Edit View Bookmarks Settings Help  
php > const SITE_NAME = "Petanikode";  
php > echo SITE_NAME;  
Petanikode  
php > SITE_NAME = "Belajar PHP";  
PHP Parse error: syntax error, unexpected '=' in php shell code on line 1  
php > define('SITE_NAME', 'Belajar PHP');  
PHP Notice: Constant SITE_NAME already defined in php shell code on line 1  
php > █
```

Pada percobaan tersebut, kita membuat konstanta dengan nama **SITE_NAME** dengan nilai **Ahmadimuslim**.

Lalu kita coba ubah nilainya menjadi **Belajar PHP**, tapi error.

Kenapa bisa error?

Karena sifat konstanta memang begitu. Nilai konstanta tidak bisa kita ubah-ubah seperti variabel.



Kapan Kita Harus Menggunakan Konstanta?

Konstanta biasanya digunakan untuk menyimpan nilai yang tidak pernah berubah.

Contoh:

```
const PHI = 3.14;  
const API_KEY = "182939812739812478u12ehj1h2u3123h12";
```

Pada dunia nyata, nilai **PHI** memang tidak akan pernah berubah. Lalu untuk **API_KEY** biasanya kita gunakan untuk mengakses sebuah web service dan nilai ini tidak akan pernah berubah di dalam program. Jadi, kita bisa memikirkan...kapan harus menggunakan konstanta dan kapan harus menggunakan variabel.

Intinya:

Jika kita ingin menyimpan nilai yang tak akan pernah berubah, maka gunakanlah konstanta. Sedangkan apabila nilai tersebut berubah-ubah di dalam program, maka gunakanlah variabel.

Cara Membuat Konstanta dan Contohnya

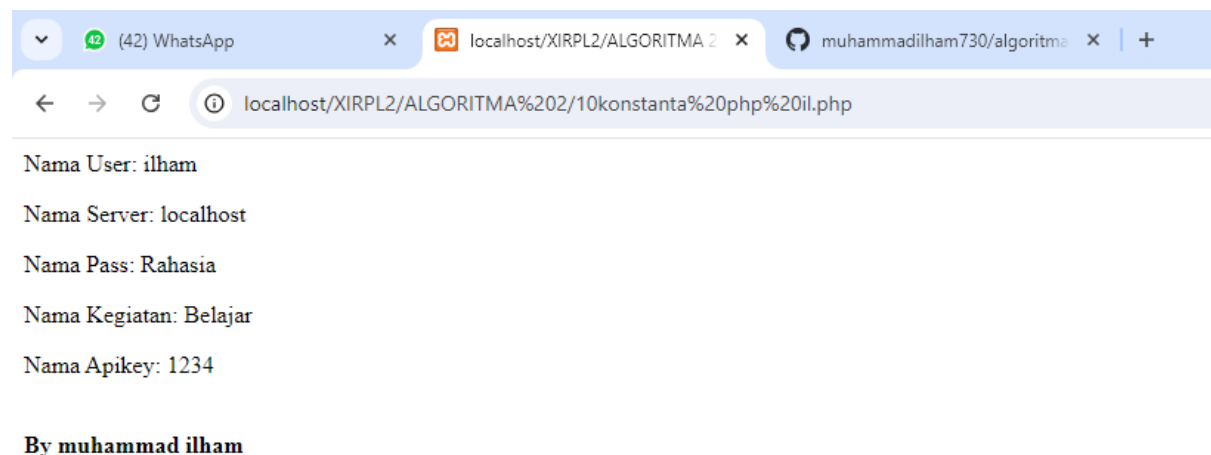
Pada PHP, kita dapat membuat konstanta dengan dua cara.

1. Menggunakan fungsi **define()**;
2. Menggunakan kata kunci **const**.

Contoh:

```
<?php
// membuat konstanta dengan fungsi define()
define('DB_SERVER', 'localhost');
define('DB_USER', 'ahmadimuslim');
define('DB_PASS', 'R4Hasia');
define('DB_NAME', 'belajar');

// membuat konstanta dengan kata kunci const
const API_KEY = "1234";
?>
```



Nama konstanta diharuskan menggunakan huruf kapital agar mudah dibedakan dengan variabel. Meskipun di PHP menggunakan simbol dolar (\$) untuk variabel, kita harus mengikuti aturan ini.



Lalu, bagaimana cara mengambil nilai dari konstanta?

Sama seperti variabel, kita tinggal tulis namanya.

Contoh:

```
const SITE_NAME = "Ahmadimuslim";
```

```
echo "Nama situs: " . SITE_NAME;
```

Perhatikan!

Kita menggunakan titik (.) untuk menggabungkan dua buah string. Karena konstanta tidak menggunakan dolar, kita tidak bisa langsung menulisnya seperti ini:

```
echo "Nama situs: SITE_NAME";
```

Biar makin mantap, coba contoh program berikut:

```
<?php // file: belajar-konstanta.php
```

```
// membuat konstanta
```

```
define('VERSION', '1.0.0');
```

```
const SITE_NAME = "Ahmadimuslim";
```

```
const BASE_URL = "https://www.ahmadimuslim.com";
```

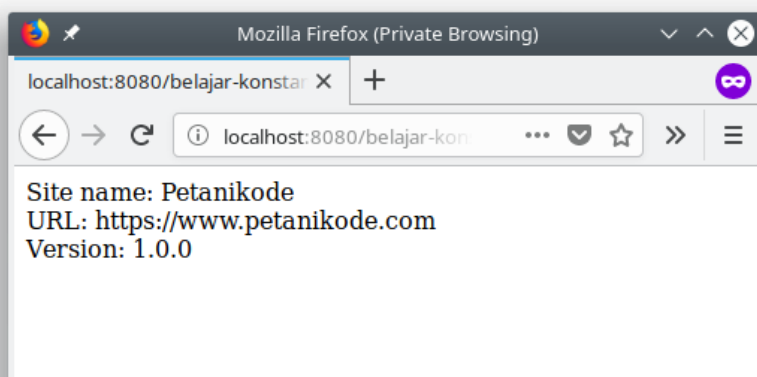
```
// cetak nilai konstanta
```

```
echo "Site name: " . SITE_NAME . "<br/>";
```

```
echo "URL: " . BASE_URL . "<br/>";
```

```
echo "Version: " . VERSION . "<br/>";
```

Hasilnya:



C. Cara Menampilkan/Mencetak Nilai dari Variabel Pada PHP

Setelah kita membuat variabel, biasanya akan kita gunakan pada proses berikutnya dengan mengambil nilainya.

Mengambil nilai dari variabel bisa kita lakukan dengan menuliskan namanya dalam perintah **echo** maupun ekspresi yang lain.

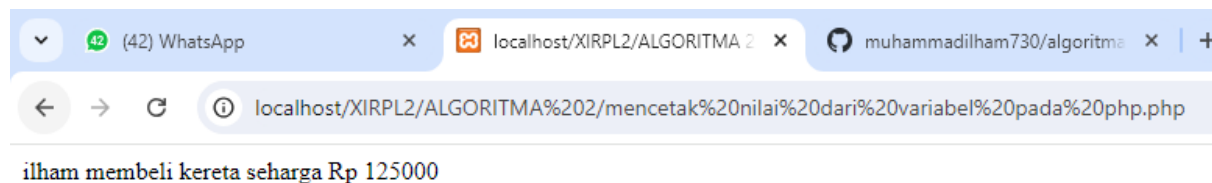
Contoh:

```
<?php
```

```
// membuat variabel baru
$nama_barang = "Minyak Goreng";
$harga = 15000;

// menampilkan isi variabel
echo "Ibu membeli $nama_barang seharga Rp $harga";
```

Hasilnya:



Jika kita menggunakan tanda petik ganda (") dalam **echo**, maka kita bisa menuliskan langsung nama variabelnya seperti ini:

```
$judul = "Belajar PHP dari nol sampai expert";
echo "Judul artikel: $judul";
```

Namun...

Apabila kita menggunakan tanda petik tunggal ('), maka kita harus menggunakan titik untuk menggabungkan teks dengan variabelnya.

Contoh:

```
$judul = "Tutorial PHP untuk Pemula";
echo 'Judul artikel: '.$judul;
```

Tanda titik pada perintah **echo** berfungsi untuk menggabungkan teks yang ada di dalam variabel dengan teks yang akan kita cetak.

Dalam PHP ada beberapa cara mencetak / menampilkan nilai dari ekspresi/deklarasi variabel sbb:

1. Fungsi echo()

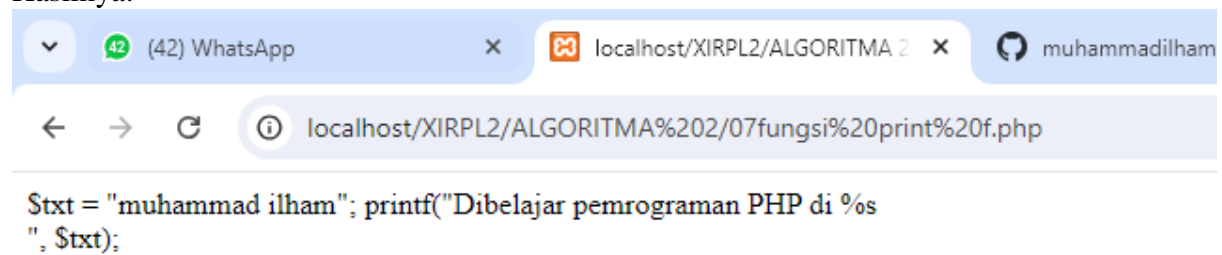
Fungsi **echo()** adalah fungsi untuk menampilkan teks ke layar. Fungsi ini dapat digunakan dengan tanda kurung maupun tanpa tanda kurung.

Contoh:

```
<?php
echo "<h2>Belajar PHP itu mudah!</h2>";
echo("Hello world!<br>");
echo "Disedang belajar PHP!<br>";
```

```
echo "Ini ", "teks ", "yang ", "dibuat ", "terpisah.";
?>
```

Hasilnya:



Fungsi **echo()** tidak akan mengembalikan apa-apa setelah dieksekusi. Dia hanya bertugas menampilkan teks saja.

2. Fungsi **print()**

Fungsi **print()** sama seperti fungsi **echo()**. Dia juga digunakan untuk menampilkan teks ke layar. Fungsi **print()** juga bisa digunakan tanpa tanda kurung.

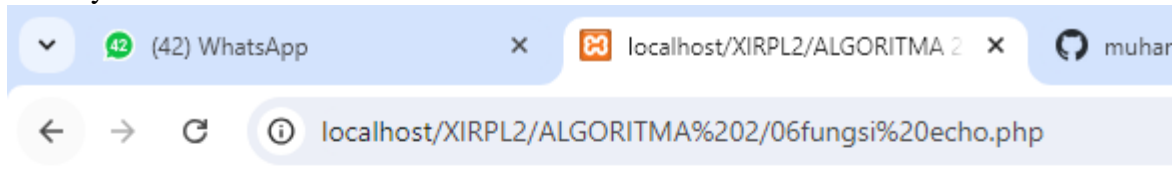
Perbedaannya dengan **echo()**:

- Fungsi **print()** akan selalu mengembalikan nilai **1** saat dieksekusi, sedangkan **echo()** tidak mengembalikan apa-apa.
- Fungsi **print()** hanya boleh diberikan satu parameter saja, sedangkan **echo()** boleh lebih dari satu.

Contoh:

```
<?php
print "<h2>Belajar PHP dari Nol!</h2>";
print "Hello world!<br>";
print "Belajar mencetak teks di PHP!";
?>
```


Hasilnya:



Belajar PHP itu mudah!

Hello muhammad ilham!
Disedang belajar PHP!
Ini teks yang dibuat terpisah.

By muhammad ilham

Apabila kita membuat variabel lalu mengisinya dengan fungsi **print()** seperti ini:

```
$cetak = print("Hello World!");
```

Maka variabel **\$cetak** akan bernilai **1**.

Apabila kita memberikan dua paramater ke dalam fungsi **print()**, maka akan terjadi error.

```
print("Hello", "World"); // <-- ini akan error
```

Percobaan pada console PHP:

```
petanikode@imajinasi ~  
petanikode@imajinasi ~ $ php -a  
Interactive mode enabled  
  
php > $cetak = print("Hello World!");  
Hello World!  
php > echo $cetak;  
1  
php > print("Hello", "World");  
PHP Parse error: syntax error, unexpected ',' in php shell code on line 1  
php > 
```

Menggabungkan String atau Teks di PHP

Pada fungsi **echo()**, kita menggabungkan teks atau string dengan memberikan sebagai argumen (dipisah dengan tanda koma).

```
echo "Ini ", "teks ", "yang ", "dibuat ", "terpisah.";
```

Selain cara ini, kita juga bisa melakukannya dengan tanda titik (.).

Titik adalah operator untuk menggabungkan dua teks di PHP.

Kenapa tidak menggunakan simbol plus (+)?

Pada PHP, simbol plus digunakan untuk penjumlahan, bukan untuk menggabungkan string.

Jadi:

```
echo "1" + "1"; // akan menghasilkan 2
```

```
echo "1" . "1"; // akan menghasilkan 11
```

Contoh:

```
<?php
```

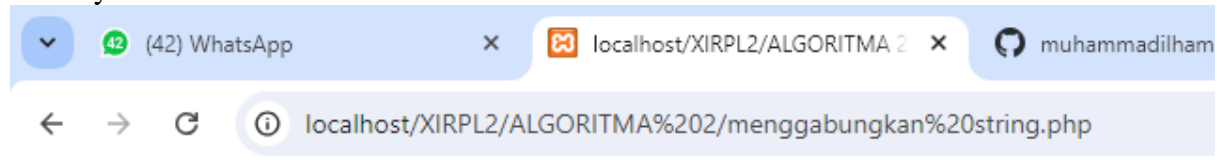
```

$txt1 = "Tutorial PHP";
$txt2 = "ahmadimuslim";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Dibelajar pemrograman PHP di " . $txt2 . "<br>";
echo $x + $y;
?>

```

Hasilnya:



Tutorial PHP

Dibelajar pemrograman PHP di m ilham
9

3. Fungsi printf()

Fungsi **printf()** adalah fungsi untuk memformat teks atau string. Fungsi ini akan mengembalikan panjang dari teks saat dieksekusi.

Biasanya saat kita menggunakan fungsi **echo**, kita akan menulis seperti ini:

```

$txt = "ahmadimuslim";
echo "Dibelajar pemrograman PHP di " . $txt . "<br>";

```

Apabila kita ingin menggunakan fungsi **printf()**, maka kita bisa lakukan seperti ini:

```

$txt = "ahmadimuslim";
printf("Dibelajar pemrograman PHP di %s<br>", $txt);

```

Simbol **%s** adalah sebuah *placeholder* untuk teks (string). Selain simbol **%s** ada juga simbol:

- **%d** untuk bilangan desimal (integer);
- **%f** untuk pecahan (float);
- **%b** untuk boolean.

Salah satu yang patut kita coba adalah **%f**, karena dengan ini kita bisa mengatur bagaimana bilangan pecahan ditampilkan.

Contoh:

```

// misalkan kita punya bilangan dengan notasi E seperti ini
$harga = 100000;

```

```

// jika kita cetak dengan echo:
echo "Harganya adalah Rp $harga";

```

```

// jika kita cetak dengan printf
printf("Harganya adalah Rp %.2f", $harga);

```

Hasilnya:

```

Harganya adalah Rp 100000
Harganya adalah Rp 100000.00

```

Simbol **%.2f** artinya kita akan menetak bilangan dengan dua angka di belakang koma.

```
localhost/XIRPL2/ALGORITMA 2 x muhammadilham730/algoritma
localhost/XIRPL2/ALGORITMA%202/07fungsi%20print%20f.php
$txt = "muhammad ilham"; printf("Dibelajar pemrograman PHP di %s", $txt);
```

Mencetak Tanda Kutip

Kadang kita ingin mencetak tanda kutip seperti ini:

```
echo 'hari ini hari jum'at';
```

Maka akan terjadi error!

Karena di sana dianggap tutup teksnya berakhir pada teks **jum**.

Lalu bagaimana caranya kita mencetak tanda kutip?

Kita bisa menggunakan tanda *back slash* (\).

Contoh:

```
echo 'hari ini hari jum\'at';
```

Maka akan menghasilkan:

```
hari ini hari jum'at
```

```
localhost/XIRPL2/ALGORITMA 2
localhost/XIRPL2/ALGORITMA%202/mencetak%20pa
echo 'hari ini hari jum\'at';
```

D. Tipe Data Pada PHP

Variabel yang sudah kita buat bisa kita simpan dengan berbagai jenis data.

Jenis-jenis data ini disebut tipe data.

Pada PHP, kita tidak harus mendeklarasikan tipe data secara eksplisit. Karena PHP sudah mampu mengenali tipe data dari nilai yang kita berikan.

Contoh:

```
<?php
```

```
// tipe data char (karakter)
$jenis_kelamin = 'L';
```

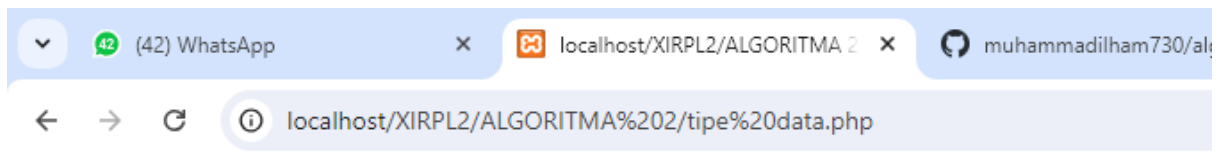
```
// tipe data string (teks)
$nama_lengkap = "Ahmadi";
```

```
// tipe data integer
$umur = 20;
```

```
// tipe data float
$berat = 48.3;
```

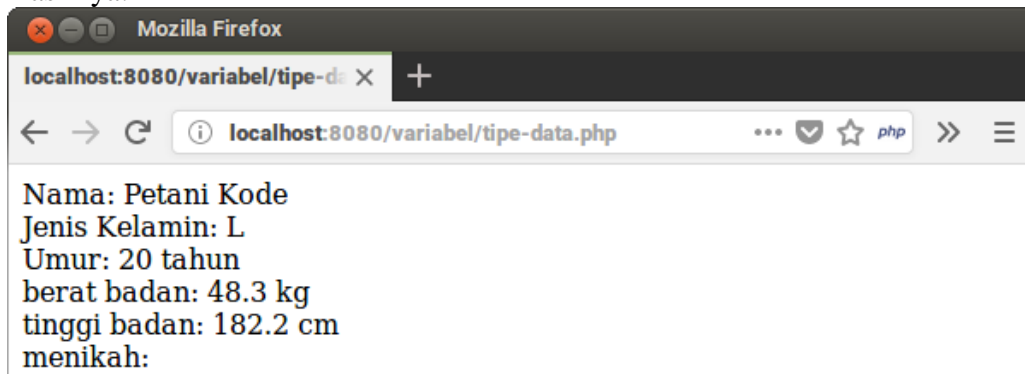
```
// tipe data float
$tinggi = 182.2;
```

```
// tipe data boolean
$menikah = false;
```



```
echo "Nama: $nama_lengkap<br>";
echo "Jenis Kelamin: $jenis_kelamin<br>";
echo "Umur: $umur tahun<br>";
echo "berat badan: $berat kg<br>";
echo "tinggi badan: $tinggi cm<br>";
echo "menikah: $menikah";
```

Hasilnya:



Variabel **\$menikah** akan ditampilkan kosong, karena nilai **false** akan dikonversi menjadi kosong dalam string.

Nanti kita akan bahas cara konversi tipe data di PHP.

Sekarang mari kita bahas masing-masing tipe data lebih detail...

Tipe Data Char dan String di PHP

Char adalah tipe data yang terdiri dari karakter. Penulisannya diapit dengan tanda petik satu.

Contoh:

```
$huruf = 'E';
```

Lalu, *String* adalah tipe data yang terdiri dari kumpulan karakter. Penulisannya diapit dengan tanda petik ganda.

Contoh:

```
$salamat = "Jl. Mawar, Jakarta";
```

```
localhost/XIRPL2/ALGORITMA%202/tipe%20data%20char%20dan%20string.php

Salamat = "Jl. Mawar, Jakarta";
```

Tipe Data Integer di PHP

Integer adalah tipe data angka. Penulisanya tidak menggunakan tanda petik.

Contoh:

```
$nilai = 98; // angka positif
$poins = -31; // angka negatif
```

```
localhost/XIRPL2/ALGORITMA%202/tipe%20data%20integer.php

$nilai = 98; // angka positif $poins = -31; // angka negatif
```

Tipe Data Float di PHP

Float adalah tipe data bilangan pecahan. Sama seperti integer, tipe data ini ditulis tanpa tanda petik.

Contoh:

```
$panjang = 233.12;
$kapasitas = 13232.12;
```

Kada juga tipe data float ditulis dalam notasi seperti ini:

```
$jarak = 1.2E-5;
```

E-5 artinya eksponen dari **10**.

Contoh di atas akan sama dengan 1.2×10^{-5} . Kalau kita jabarkan akan menjadi **0.000012**.

Agar format float tidak tercetak dalam notasi **E**, kita bisa gunakan fungsi **sprintf()**.

Contoh:

```
echo sprintf('%f', $a);
// batasi angka di belakang koma
echo sprintf('%.3f', $a);
```

```
localhost/XIRPL2/ALGORITMA%202/tipe%20data%20dan%20float.php

echo sprintf('%f', $a); // batasi angka di belakang koma echo sprintf('%.3f', $a);
```

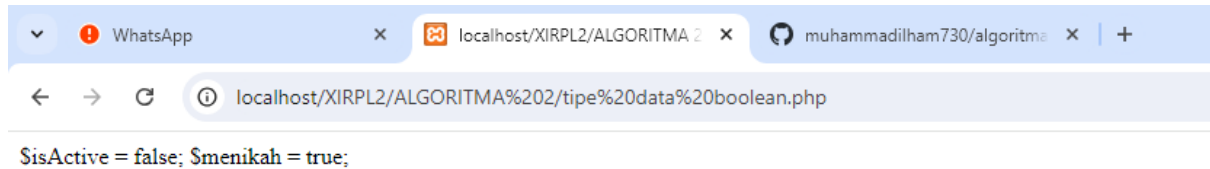
Tipe data Boolean di PHP

Tipe data *boolean* adalah tipe data yang hanya bernilai **true** dan **false**.

Penulisan **true** dan **false** tidak diapit dengan tanda petik.

Contoh:

```
$isActive = false;  
$menikah = true;
```



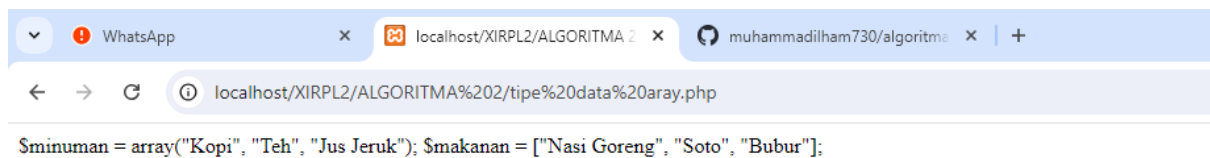
Tipe Data Array dalam PHP

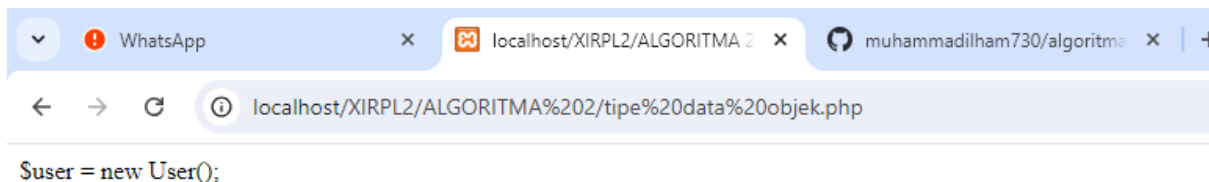
Array adalah tipe data yang berisi sekumpulan data.

Contoh:

```
$minuman = array("Kopi", "Teh", "Jus Jeruk");  
$makanan = ["Nasi Goreng", "Soto", "Bubur"];
```

Kita akan membahas lebih dalam tentang array di: 7 Hal yang Harus Kamu Ketahui Tentang Array.





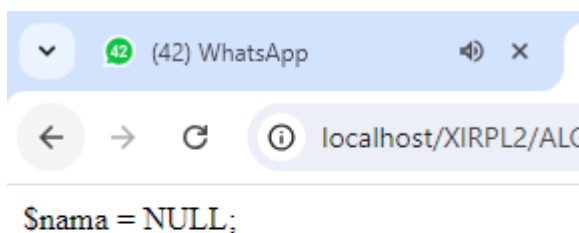
Tipe Data NULL di PHP

Tipe data **NULL** adalah tipe data yang menyatakan kosong.

Artinya: Jika kita mengisi variabel dengan nilai **NULL**, maka variabel tersebut akan dianggap kosong atau tidak punya nilai.

Contoh:

```
$nama = NULL;
```



4. Konversi Tipe data di PHP

Apa yang akan terjadi bila kita melakukan pembagian dengan tipe data teks dengan angka seperti ini:

```
$a = 3;  
$b = "angka 10";
```

```
$c = $b / $a;
```

Tentunya akan terjadi error saat kita ingin mencetak isi variabel **\$c**.

Karena itu, kita harus konversi dulu tipe datanya agar bisa dilakukan operasi yang lain.

Pada PHP konversi tipe data bisa dilakukan dengan operator kali (*).

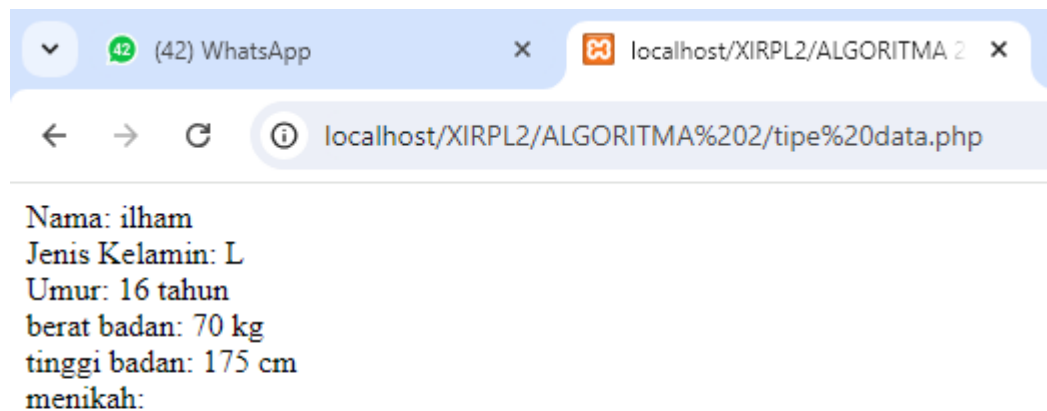
Contoh:

```
<?php  
$foo = "1"; // mula-mula $foo dalam bentuk string (ASCII 49)  
$foo *= 2; // $foo sekarang adalah integer (2)  
$foo = $foo * 1.3; // $foo sekarang adalah float (2.6)  
$foo = 5 * "10 Little birds"; // $foo sekarang adalah integer (50)  
$foo = 5 * "10 Small birds"; // $foo sekarang adalah integer (50)  
?>
```

Selain cara ini, kita juga bisa melakukannya seperti di bahasa C:

```
<?php
```

```
$a = "32";  
$a = (int) $a; // ubah nilai a menjadi integer  
$a = (float) $a; // ubah nilai a menjadi float  
$a = (string) $a; // ubah nilai a menjadi string  
  
?>
```



5. Menghapus Variabel dari Memori

Apabila ada variabel yang sudah tidak dibutuhkan lagi, maka kita bisa menghapusnya untuk meningkatkan performa program.

Cara menghapus variabel di PHP dapat menggunakan fungsi `unset()`.

Contoh:

```
// membuat variabel $tmp  
$tmp = 2901;  
  
// menghapus variabel $tmp  
unset($tmp);  
  
// mencoba mengakses variabel $tmp  
echo $tmp;
```

Jika kita eksekusi kode di atas, maka akan terjadi error:

PHP Notice: Undefined variable: tmp

Karena variabel **\$tmp** sudah kita hapus.

Kenapa harus menghapus variabel? nanti juga akan terhapus sendiri saat programnya selesai?

Biasanya saat kita ingin optimasi, variabel-variabel yang tidak terpakai harus dihapus agar tidak membebani memori pada server.

▪ LATIHAN 6

Kerjakan soal-soal berikut dengan baik dan benar!

- 1) Sebutkan 3 fungsi untuk mencetak teks ke layar, dan bagaimana cara menuliskannya....
- 2) Jelaskan perbedaan Fungsi print() dengan echo()....
- 3) Jelaskan Fungsi printf() dalam PHP, dan sebutkan 4 jenis penerapannya (dengan %)...
- 4) Bagaimana jika kita ingin menetak tanda kutip...
- 5) Bagaimana cara membuat variabel bernama harga dengan isi 1000?
- 6) Bagaimana cara mengisi ulang dengan nilai yang baru?
- 7) Jelaskan 4 hal yang harus diperhatikan membuat nama variabel!
- 8) Kapan saatnya Kita Harus Menggunakan Konstanta?
- 9) Jelaskan dua cara membuat konstanta pada PHP!
- 10) Bagaimana cara membuat konstanta bernama site_name dengan isi Ahmadimuslim;
- 11) Jelaskan Tipe Data berikut ini :
 - a) Char dan String
 - b) Integer
 - c) Float
 - d) Boolean
 - e) Array
 - f) Objek
 - g) NULL
- 12) Mengapa kita harus melakukan konversi tipe data?
- 13) Mengapa kita harus menghapus variabel yang tidak terpakai di PHP? Bagaimana Cara menghapusnya?