
MODUL 6

WEB PROGRAMMING : PHP 1

Tujuan :

1. Memahami tentang PHP
2. Memahami instalasi Apache dan PHP
3. Memahami tag-tag dalam PHP
4. Memahami tentang struktur kontrol

PENGENALAN PHP

PHP merupakan akronim dari "PHP : Hypertext Presprocessor". PHP merupakan bahasa script yang biasa digunakan untuk web development yang dapat diselipkan dalam HTML.

Berbeda dengan script-script lainnya seperti Java Script atau VB script, PHP dieksekusi di lingkungan server, client hanya menerima hasil dari script yang telah dieksekusi, tanpa bisa mengetahui kode yang digunakan.

Apa saja yang dapat dilakukan PHP?

PHP difokuskan pada scripting server-side, jadi Anda dapat melakukan apa yang bisa dilakukan CGI dengan menggunakan PHP seperti mengambil data inputan form, meng-generate konten halaman dinamis, mengirim dan menerima cookies dan masih banyak lagi. Kemampuan dan supportnya untuk database juga sangat dapat diandalkan.

Sekarang ini, PHP bahkan dikembangkan untuk menjadi bahasa pemrograman. Untuk lebih jelasnya, Anda dapat mencari informasi lebih lanjut di distro-distro Linux keluaran baru.

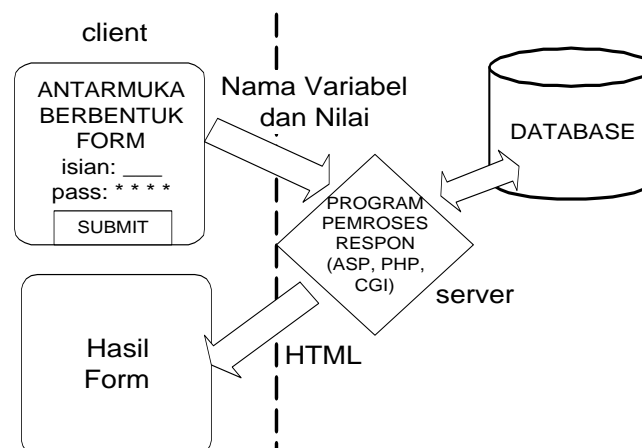
Di lingkungan mana PHP bekerja?

Awalnya, PHP bekerja di lingkungan Linux, karena merupakan 'turunan' bahasa PEARL yang merupakan bahasa pemrograman native di lingkungan Linux dan Unix. Namun seiring perkembangan dan kebutuhan maka PHP dibuat secara cross platform. PHP dapat bekerja di lingkungan Windows maupun Linux.

PHP mengeksekusi server script lebih cepat dari server script manapun. PHP menempati ranking pertama dalam hal kecepatan eksekusi, diikuti oleh ASP dan JSP.

Dengan kemampuannya, apakah PHP gratis?

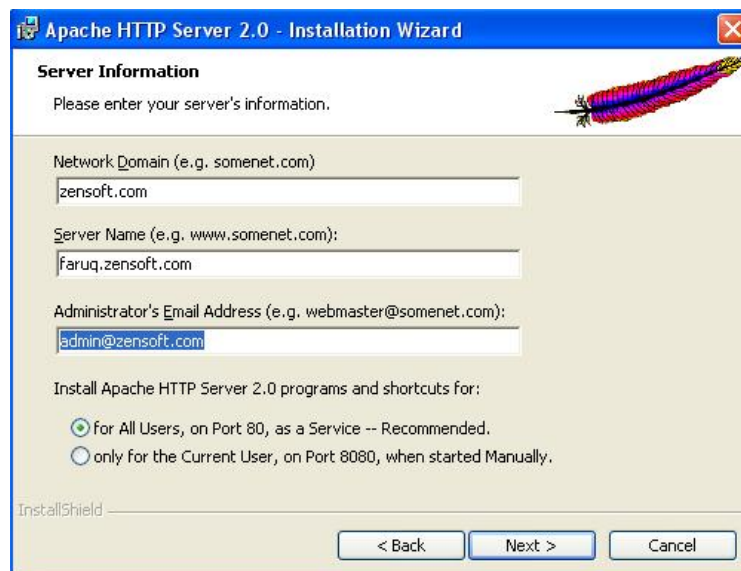
PHP dapat anda Download secara free atau cuma-cuma. Kunjungi saja situs www.php.net, dan download versi terbarunya.



INSTALASI

Instalasi apache :

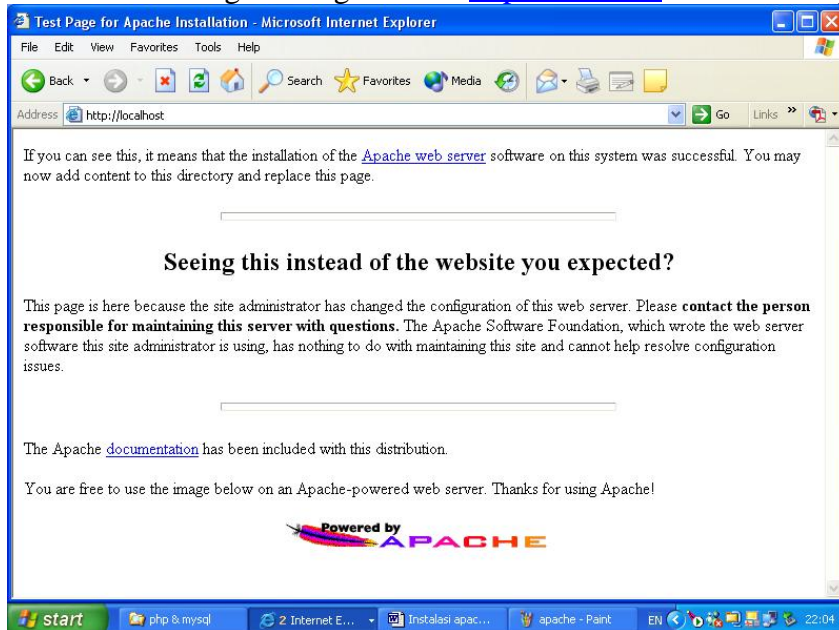
1. Jalankan apache_2.0.48-win32-x86-no_ssl.msi
2. Klik next, sampai ada isian seperti berikut dan isikan :
domain anda seperti : zensoft.com
server name : faruq.zensoft.com
administrator email address : admin@zensoft.com



3. Pilih Typical dan klik next.
4. Klik finish, dan di kanan bawah akan muncul tanda ikon apache.



5. Test koneksi dengan mengetikkan : <http://localhost> di internet explorer

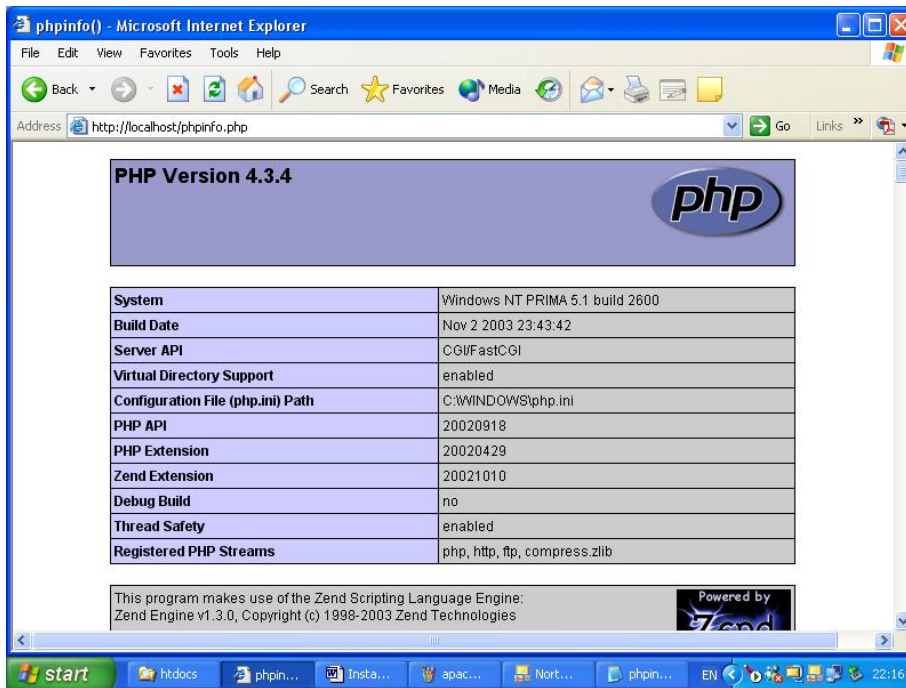


Instalasi php :

1. Ekstrak php-4.3.4-Win32.zip ke drive c:/
2. Ganti nama c:/php-4.3.4-Win32.zip dengan c:/php
3. Copykan file php.ini-dist ke C:/windows
4. Ubah nama file diatas menjadi php.ini
5. Edit httpd.conf di C:/Program Files/Apache Group/Apache2/conf
 - a. Tambahkan script alias berikut ini :
`ScriptAlias /cgi-bin/ "C:/Program Files/Apache Group/Apache2/cgi-bin/"`
ScriptAlias /php/ "C:/php/"
 - b. Tambahkan addtype :
`AddType application/x-compress .Z`
`AddType application/x-gzip .gz .tgz`
AddType application/x-httpd-php .php
Action application/x-httpd-php "/php/php.exe"
 - c. Tambahkan index.php untuk pathnya :
`DirectoryIndex index.html index.html.var index.php`
6. Restart Apache
7. Buat file php, simpan dengan nama phpinfo.php dan masukkan di C:/Program Files/Apache Group/Apache2/htdocs

```
<? phpinfo(); ?>
```

8. Test apakah php sudah terinstal dengan baik :
<http://localhost/phpinfo.php>



TAG DALAM PHP

Banyak cara untuk menyisipkan PHP dalam script HTML, ada berbagai macam bentuk tag yang dapat digunakan, antara lain :

- Cara I

<code><?php</code>	—————>	menandai awal tag
...		
<code>?></code>	—————>	menandai akhir tag
- Cara II

<code><?</code>	—————>	menandai awal tag
...		
<code>?></code>	—————>	menandai akhir tag
- Cara III

<code><%</code>	—————>	menandai awal tag
...		
<code>%></code>	—————>	menandai akhir tag
- Cara IV

<code><script language="php"></code>	——>	menandai awal script
...		
<code></script></code>	——>	menandai akhir script

Namun untuk lebih mudahnya, kita akan menggunakan cara II mulai sekarang dan seterusnya.

Ada kalanya kita sebagai pemrogram, karena banyaknya kode program atau variabel dalam program, perlu menandai atau memberi komentar pada program. Komentar pada program merupakan tulisan pada program yang tidak dieksekusi. Pada PHP, ada 3 macam cara penulisannya :

1. `/* komentar */`
Tulisan apapun yang berada di antara `'/*'` dan `'*/'` akan dianggap sebagai komentar. Cara seperti ini sangat berguna dan efisien untuk pemberian komentar yang memakan banyak baris.
2. `// komentar`
Tulisan di baris yang sama setelah `'//'` akan dianggap sebagai komentar. Cara ini berguna untuk pemberian komentar singkat yang tak lebih dari 1 baris saja.
3. `# komentar`
Sama seperti `'//'`, tulisan di baris yang sama setelah `'#'` akan dianggap sebagai komentar. Cara ini berguna untuk pemberian komentar singkat yang tak lebih dari 1 baris saja.

VARIABEL DAN TIPE DATA

Dalam PHP, penulisan variabel diawali dengan karakter '\$', kemudian diikuti dengan huruf sebagai karakter pertama. Setelah itu, dapat dilanjutkan dengan kombinasi huruf dan angka. Variabel tidak boleh mengandung spasi maupun tanda baca di dalamnya, kecuali underscore ('_').

Variabel pada PHP bersifat "case sensitif", yang berarti Anda harus memperhatikan penulisan huruf besar dan huruf kecil. Variabel `$jumlah` berbeda dengan `$Jumlah`.

Contoh penamaan variabel yang benar :

```
$nama_pemakai  
$kota_3  
$user1
```

Contoh penamaan variabel yang salah :

```
$nama pemakai  
$3kota  
$us\er1
```

Tipe Data

Tipe data yang dapat diolah menggunakan PHP, antara lain :

1. Boolean

Nilai boolean menyatakan nilai kebenaran. Suatu variabel boolean dikatakan salah (**FALSE**) jika :

1. Variabel boolean bernilai **FALSE**
2. Variabel integer 0 (nol)
3. Variabel float 0.0 (nol)
4. Variabel string kosong atau string "0"
5. Variabel array dengan nol elemen
6. Variabel objek dengan nol anggota
7. Dan tipe spesial yaitu **NULL**

Selain hal-hal tersebut di atas, maka nilai boolean dianggap **TRUE**.

2. Integer

```
5  
-9
```

- 1500
- 0123 octal
- 0xfb hexadecimal

3. Floating Point

- 0.8998
- 23e2

4. String

- 'Teks menggunakan apostrof' ...menggunakan tanda petik tunggal
- "Teks menggunakan quote" ...menggunakan tanda kutip

Pada tipe data string, terdapat escape character, yaitu karakter khusus yang digunakan mewakili karakter ASCII dengan fungsi khusus. Untuk mendefinisikan karakter yang akan ditampilkan jika karakter tersebut merupakan escape character yaitu dengan mengawali karakter tersebut dengan tanda \ (backslash).

Daftar escape character :

\n	linefeed (LF or 0x0A (10) in ASCII)
\r	carriage return (CR or 0x0D (13) in ASCII)
\t	horizontal tab (HT or 0x09 (9) in ASCII)
\\	backslash
\\$	dollar sign
\"	double-quote
\[0-7]{1,3}	the sequence of characters matching the regular expression is a character in octal notation
\x[0-9A-Fa-f]{1,2}	the sequence of characters matching the regular expression is a character in hexadecimal notation

5. Array

Tipe data array akan dibahas pada suatu bab tersendiri.

6. Object

Ini merupakan pengembangan PHP berorientasi objek. Tipe data objek merupakan tipe data yang didalamnya mempunyai data dan method. Data tersebut didefinisikan sebagai class terlebih dahulu.

Operator

Operator merupakan suatu tool/alat untuk keperluan manipulasi data. Operator dapat dibedakan menjadi :

1. Operator Aritmetika

Contoh	Nama	Hasil
\$a + \$b	Penjumlahan	Jumlah \$a dan \$b.
\$a - \$b	Pengurangan	Selisih \$a dan \$b.
\$a * \$b	Perkalian	Hasil kali \$a dan \$b.
\$a / \$b	Pembagian	Quotient dari \$a dan \$b.
\$a % \$b	Modulus	Remainder dari \$a dibagi \$b.

2. Operator String

Menggunakan '.' (tanda titik)

```
$a = "Hello ";
$b = $a . "World!"; // sekarang $b berisi "Hello World!"
```

Menggunakan ‘.=’ (tanda titik dan sama dengan)

```
$a = "Hello ";
$a .= "World!"; // sekarang $a berisi "Hello World!"
```

3. Operator Increment/Decrement

Contoh	Nama	Keterangan
++\$a	Pre-increment	Increment \$a dengan 1, mengembalikan \$a.
\$a++	Post-increment	Mengembalikan \$a, lalu increments \$a dengan 1
--\$a	Pre-decrement	Decrements \$ dengan 1, lalu mengembalikan \$a.
\$a--	Post-decrement	Mengembalikan \$a, lalu decrements \$a dengan 1

4. Operator Logika

Contoh	Name	Hasil
\$a and \$b	And	TRUE jika \$a dan \$b TRUE.
\$a or \$b	Or	TRUE jika \$a atau \$b TRUE.
\$a xor \$b	Xor	TRUE jika salah satu dari \$a atau \$b TRUE
! \$a	Not	TRUE jika \$a bukan TRUE.
\$a && \$b	And	TRUE jika \$a dan \$b TRUE.
\$a \$b	Or	TRUE jika \$a atau \$b TRUE.

5. Operator Ternary

Menggunakan tanda '??', dengan syntax :

(ekspresi1) ? (ekspresi2) : (ekspresi3)

Operator ternary melakukan pengujian pada ekspresi1, jika hasilnya TRUE, maka ekspresi 2 yang dijalankan, jika hasil ekspresi1 FALSE, maka ekspresi3 yang dijalankan.

Contoh :

```
<?
$a = 2;
$hasil = ($a<4) ? ($a*2) : ($a*3)
echo "$hasil";
?>
```

Dari contoh di atas, maka hasil yang terlihat adalah 4, namun jika nilai \$a diganti 5, maka hasilnya adalah 15.

6. Operator Perbandingan

Contoh	Nama	Hasil
\$a == \$b	Equal	TRUE jika \$a sama dengan \$b.
\$a === \$b	Identical	TRUE jika \$a sama dengan \$b, dan

Contoh	Nama	Hasil
		mempunyai tipe yang sama (hanya di PHP 4)
<code>\$a != \$b</code>	Not equal	TRUE jika \$a tidak sama dengan \$b.
<code>\$a <> \$b</code>	Not equal	TRUE jika \$a tidak sama dengan \$b.
<code>\$a !== \$b</code>	Not identical	TRUE jika \$a tidak sama dengan \$b, atau mempunyai tipe yang berbeda (hanya di PHP 4)
<code>\$a < \$b</code>	Less than	TRUE jika \$a kurang dari \$b.
<code>\$a > \$b</code>	Greater than	TRUE jika \$a lebih dari \$b.
<code>\$a <= \$b</code>	Less than or equal to	TRUE jika \$a lebih kecil sama dengan \$b.
<code>\$a >= \$b</code>	Greater than or equal to	TRUE jika \$a lebih besar sama dengan \$b.

7. Operator Assignment

Contoh :

```
$a = 4;           // $a diisi dengan nilai 4
$a += 5;        // sama dengan $a = $a + 5;
```

8. Operator Kontrol Error

Dengan memberikan tanda '@' pada ekspresi PHP, maka pesan error yang dihasilkan oleh suatu ekspresi akan diabaikan. Jika fitur **track_error** enable, maka setiap pesan error yang diterima akan disimpan dalam variabel global `$php_errormsg`.

Contoh :

```
<?php
$my_file = @file ('non_existent_file') or
die ("Failed opening file: error was '$php_errormsg'");
?>
```

9. Operator Eksekusi

Operator ini hanya bekerja di lingkungan sistem operasi Linux dan Unix, dengan webserver yang digunakan adalah Apache.

Operator eksekusi menggunakan karakter ``...perintah...`` (perintah di antara backticks), dimana hasilnya ditampung pada sebuah variabel dan kemudian ditampilkan di halaman web.

Lingkup Variabel

1. Variabel Global

Variabel global adalah variabel dan data di dalamnya dikenali oleh seluruh bagian script. Suatu variabel yang dibuat pada bagian utama script bukan pada bagian suatu fungsi, variabel-variabel ini akan bersifat global.

Untuk fungsi yang menggunakan variabel global, maka nama variabel pada fungsi harus dideklarasikan global.

Contoh :

```
function test() {
    global $var;
    echo $var;
```

```

    }
    $var = 2;
    test();

```

2. Variabel Lokal

Variabel lokal merupakan variabel yang dideklarasikan pada suatu fungsi, dan hanya dapat digunakan oleh fungsi tersebut.

3. Variabel Static

Variabel static merupakan variabel lokal dalam fungsi yang memungkinkan nilai terakhir di dalamnya dipertahankan.

Sintaks untuk deklarasi variabel statik :

```

function namafungsi() {
    static $namaVariabel = InitValue;
    perintah perubahan $namaVariabel;
    ...
}

```

4. Variabel Web

Variabel web merupakan variabel yang dihasilkan oleh PHP secara otomatis, ketika ada request (permintaan) ke server yang mendukung PHP. Contoh variabel web yaitu form isian dari user, baik dengan metode POST maupun GET, selain itu ada pula SESSION, COOKIES dan lain sebagainya. Topik ini akan dibahas nanti pada pengolahan data form.

Untuk menampilkan data dapat digunakan print maupun echo. Sintaks yang digunakan antara lain :

```

print(data);
atau
print data;
atau
echo(data);
atau
echo data;

```

Kedua perintah baik print maupun echo adalah sama saja, tergantung kebiasaan atau kesukaan Anda dalam menggunakannya. Yang perlu diperhatikan di sini adalah bahwa untuk data string gunakanlah apostrophe (petik tunggal ') atau quote (petik ganda "). Namun untuk pencetakan variabel, agar ditampilkan isinya, gunakan quote. Seperti pada contoh di bawah ini.

Contoh :

```

<?
    $a = 10;
    $b = "test";
    echo "Dengan quote -> ";
    echo "nilai $b ku $a ";
    echo "Dengan apostrophe -> ";
    echo 'nilai $b ku $a'
?>

```

Hasil dari script di atas adalah :

```

    Dengan quote -> nilai test ku 10
    Dengan apostrophe -> nilai $b
    ku $a

```

STRUKTUR KONTROL

Struktur kontrol merupakan inti dari logika pemrograman. Secara mendasar ada 3 struktur kontrol yaitu :

- Sequence atau urutan
- Selection atau pilihan
- Iteration atau pengulangan

if / if else

Sintaks untuk struktur kontrol menggunakan if else/ifelse :

```
if (ekspresi) {
    perintah1;
    perintah2;
    ...
}
```

Struktur kontrol if memeriksa isi ekspresi, dimana jika terpenuhi atau bernilai benar, maka ia akan menjalankan perintah-perintah yang ada dalam blok.

Untuk banyak kondisi, bisa dilakukan dengan sintaks :

```
if (ekspresi1) {
    perintah1;
    perintah2;
    ...
}
elseif(ekspresi2) {
    perintah3;
    perintah4;
    ...
}
else {
    perintah5;
    perintah6;
    ...
}
```

Dimana jika ekspresi1 tidak tercapai, maka struktur kontrol akan memeriksa apakah ekspresi berikutnya tercapai, hingga ia mencapai struktur else sebagai pilihan default atau terakhir.

switch

Dapat dikatakan kalau switch adalah merupakan penyederhanaan dari bentuk if else ataupun ifelse. Sintaks dari struktur kontrol menggunakan switch antara lain :

```
switch($variabel) {
    case nilai1 :
        perintah_jika_nilai1;
        break;
    case nilai2 :
        perintah_jika_nilai2;
        break;
    case nilai3 :
        perintah_jika_nilai3;
        break;
    ...
    [default :
        perintah_untuk_nilai_default;
    ]
}
```

Untuk nilai default, tidaklah harus ada.

while

Sintaks :

```
while(ekspresi) {
    perintah1;
    perintah2;
    ...
}
```

do .. while

Sintaks :

```
do {
    perintah1;
    perintah2;
    ...
} while (ekspresi);
```

for

Sintaks :

```
for(nilai_awal ;syarat_terjadinya_loop ; inc_atau_dec)
{
    perintah1;
    perintah2;
    ...
}
```

foreach

Struktur kontrol foreach digunakan untuk pengulangan terhadap variabel array. Struktur kontrol foreach akan melakukan loop mulai dari awal indeks array hingga indeks terakhir array tersebut, meskipun array itu bersifat assosiatif.

Sintaks :

```
foreach (var_array as value) {
    perintah1;
    perintah2;
    ...
}
```

Untuk array asosiatif :

```
foreach (var_array as key => value) {
    perintah1;
    perintah2;
    ...
}
```

break

Perintah break digunakan untuk keluar dari suatu pengulangan/loop.

Contoh :

```
for($a = 0; $a < 100 ; $a++)
{
    if($a == 30) { break; }
    else echo "$a ,";
}
```

Pada contoh di atas, akan memunculkan tulisan berupa nilai variabel a, mulai dari 0,1,2,3...hingga ketika mencapai nilai 30 maka loop akan berhenti karena perintah break.

continue

Perintah continue digunakan untuk menuju step atau iterasi berikutnya dari suatu loop.

Contoh :

```
for($a = 0; $a<10; $a++)
{
    if($a%2==0) continue;
    echo "$a ";
}
```

Pada contoh di atas, hasil yang akan didapat adalah "1 3 5 7 9"

LATIHAN – LATIHAN :

1. Pemakaian Variabel

```
<HTML>
<HEAD>
<TITLE> Contoh Skrip PHP </TITLE>
<BODY>
<?
$nama = "Muhammad Faruq";
echo "$nama berkata \"Katakan namamu!\"";
?>
</BODY>
</HTML>
```

2. Operator Penugasan

```
<?
$a = 3;
$b = 7;
$a += 5;
$b = ($c = 11) + 3;
echo "Nilai variabel a adalah = $a";
echo "<BR>";
echo "Nilai variabel b adalah = $b";
echo "<BR>";
echo "Nilai variabel c adalah = $c";
?>
```

3. Operasi Aritmatika

```
<?
$a=10;
$b=3;
echo "\$a = $a <BR>";
echo "\$b = $b <BR>";
echo '$a + $b = ';
print $a+$b;
echo "<BR>";
echo '$a - $b = ';
print $a-$b;
echo "<BR>";
```

```
echo '$a * $b = ';
print $a*$b;
echo "<BR>";
echo '$a / $b = ';
print $a/$b;
echo "<BR>";
echo '$a % $b = ';
print $a%$b;
?>
```

4. Operator Logika

```
<?
$b = 4!=4;
$c = 3+7 == 10;
$a = ($b and $c);
echo "\$a=$a <BR>";
$a = ($b or $c);
echo "\$a=$a <BR>";
$a = ($b xor $c);
echo "\$a=$a <BR>";
$a = (!$b or $c);
echo "\$a=$a <BR>";
$a = $b && $c;
echo "\$a=$a <BR>";
$a = $b || $c;
echo "\$a=$a <BR>";
?>
```

5. Operator Perbandingan

```
<?
$x = 4;
$a = ($x == 4);
echo "\$a = $a <BR>";
$b = ($x === "4");
echo "\$b = $b <BR>";
$c = ($x != 4);
echo "\$c = $c <BR>";
$d = ($x !== "4");
echo "\$d = $d <BR>";
$e = ($x < 5);
echo "\$e = $e <BR>";
$f = ($x > 5);
echo "\$f = $f <BR>";
$g = ($x <= 4);
echo "\$g = $g <BR>";
$h = ($x >= 5);
echo "\$h = $h <BR>";
?>
```

6. Struktur Kontrol : If

```
<?
$a = 5;
$b = 7;
if ($a < $b)
{
    echo "\$a lebih kecil daripada \$b";
}
?>
```

7. Struktur Kontrol : If - else

```
<?
$a = 5;
$b = 3;
echo "\$a = $a <BR>";
echo "\$b = $b <BR>";
if ($a < $b)
    {
    echo '$a lebih kecil daripada $b';
    }
else
    {
    echo '$a lebih besar daripada $b';
    }
?>
```

8. Struktur Kontrol : Syntax if – else yang lain

```
<?
$a = 5;
$b = 7;
echo "\$a = $a <BR>";
echo "\$b = $b <BR>";
if ($a == $b):
    echo '$s sama dengan $b';
elseif ($a > $b):
    echo '$a lebih besar daripada $b';
else:
    echo '$a lebih kecil daripada $b';
endif;
?>
```

9. Struktur Kontrol : Switch

```
<?
$a = 5;
switch ($a) {
    case 0:
        echo '$a sama dengan 0';
        break;
    case 1:
        echo '$a sama dengan 1';
        break;
    case 2:
        echo '$a sama dengan 2';
        break;
    default:
        echo '$a tidak sama dengan 0, 1, atau 2';
}
?>
```

10. Struktur Kontrol : while

```
<?
echo "<H3> Cara 1 </H3>";
$i = 1;
while ($i <= 10)
{
    print $i++;
}
echo "<BR>";
echo "<H3> Cara 2 </H3>";
$i = 1;
while ($i <= 10):
    print $i;
    $i++;
endwhile;
?>
```

11. Struktur Kontrol : do...while

```
<?
$i = 2;
do
{
    echo "\$i = $i <BR>";
    $i++;
}
while ($i < 5);
?>
```

12. Struktur Kontrol : for

```
<?
for ($i=1;$i<=10;$i++) {
    print $i;
}
?>
```

13. Struktur Kontrol : break

```
<?
for ($i = 1; $i <= 10; $i++) {
    if ($i == 6) {
        break;
    }
    echo "\$i = $i <BR>";
}
?>
```

14. Struktur Kontrol : continue

```
<?
for ($i = 1;$i <= 10;$i++) {
    if (!(($i%2)) {
        continue;
    }
    echo "\$i = $i <BR>";
}
?>
```

TUGAS :

1. Buatlah program php untuk menyelesaikan tes psyko berikut ini :

- a. 4 6 9 13 18 ? ?
- b. 2 2 3 3 4 ? ?
- c. 1 9 2 10 3 ? ?