**PHP**

***(Hypertext Prepocessor)***

**PHP** adalah suatu bahasa pemrograman berbasiskan kode-kode(script) yang berfungsi untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML.

PHP mempunyai ciri-ciri, antara lain :

1. PHP hanya dapat berjalan diatas *web server*, contoh : Apache.
2. PHP dapat digunakan untuk mengakses database, seperti: MysQL, PostgreSQl, Oracle, Sqlite, dll
3. PHP bersifat multiplatform, dapat dijalankan menggunakan sistem operasi apapun. contoh: Linux, Windows, Unix, Mac.
4. bersifat *open source* gratis untuk di-*download* dan digunakan.

PHP banyak digunakan khususnya di Indonesia untuk membangun sebuah aplikasi berbasis Web, pada Bab sebelumnya telah dijelaskan bagaiman meng-*install* Web Server. dan kini akan kita coba untuk mempelajari berbagai macam syntax PHP yang akan dijelaskan dalam modul ini.

**A. Dasar Pemrograman PHP**

setelah persiapan web server, kini saatnya kita mempelajari kode-kode PHP. sytnax PHP diawali dengan **<?php** dan diakhiri dengan **?>** Blok kode PHP selalu diampit oleh tanda tersebut. seperti pada pemrograman yang lain untuk setiap baris perintah akan diakhiri dengan tanda ; (titik koma) . kita akan membuat program pertama dengan PHP , simpan dengan nama **hello.php**

Contoh :

<?php

echo "Hallo Dunia";

?>

fungsi **echo** pada kode diatas berfungsi untuk menulisakan keluaran PHP di web browser. untuk memahami suatu bahasa pemrograman anda harus mengetahui komponen,syntax penyusun program. berikut adalah hal-hal yang sering digunakan dalam pemrograman PHP

**1. Variable**

variabel dalam PHP selalu dinyatakan dengan tanda dollar ($). nama sebuah variabel tidak bisa dipisahkan dengan spasi. jika ingin variable dengan spasi bisa menggunakan tanda hubung bawah/underscore("\_"). dan huruf pertama variable tidak boleh berupa angka, dan PHP bersifat *case-sensitive*

contoh :

$Nama = "Najib Abdillah";

$Nilai\_Makul\_PHP =100;

contoh penulisan :

<?php

$Nama = "Najib Abdillah";

$Nilai\_Makul\_PHP =100;

echo "Nama : $Nama<br>";

echo "Nilai Praktikum :

$Nilai\_Makul\_PHP";

?>

simpan dan jalankan script diatas dengan *web browser*, tentunya jangan lupa disimpan pada folder *htdocs* pada *Xampp* . dan *output*nya adalah :

Nama : Najib Abdillah  
Nilai Praktikum : 100

Untuk memanipulasi beberapa string PHP sudah mempunyai beberapa fungsi untuk meng*handle* string antata lain adalah.

1. **strlen()**

fungsi strlen() digunakan untuk menghitung panjang teks. contoh :

<?php

$Nama = "Najib Abdillah";

$panjangteks = "Panjang teks :".strlen($Nama);

echo "$panjangteks";

?>

Outputnya :

Panjang teks :14

1. **trim()**

fungsi trim() berfungsi untuk menghilangkan spasi baik di kanan maupun di kiri

contoh :

<?php

$Nama = "Najib Abdillah";

$Nama = trim($Nama);

echo $Nama.$Nama;

?>

Outputnya :

Najib AbdillahNajib Abdillah

1. **htmlentities()**

Fungsi ini digunakan untuk menghentikan proses penerjemaahan tag HTML oleh

browser, sehingga tag HTML akan dibiarkan muncul seperti apa adanya, tanpa

diterjemahkan oleh browser. Sintaksnya adalah sebagai berikut:

*Htmlentities(string)*

<?PHP

$str = "<A HREF='abc.htm'> Klik Di Sini </A>";

$result = htmlentities($str);

echo "$str";

echo "$result";

?>

1. **strstr(), stristr(), dan strchr()**

Ketiga fungsi ini digunakan untuk mencari keberadaan suatu string di dalam string

lain. Sintaksnya adalah sebagai berikut:

**Strstr(tujuan,pencari)**

**Stristr(tujuan,pencari)**

**Strchr(tujuan,pencari)**

Dari ketiga fungsi tersebut, hanya stristr() yang tidak bersifat case sensitive. Parameter

tujuan adalah string yang dicari oleh pencari. Hasil yang didapat adalah mulai dari

karakter pencari pertama yang ditemukan sampai akhir string tujuan. Jika string

pencari tidak terdapat pada string tujuan, maka fungsi akan menghasilkan nilai False.

contoh :

$email = "user@domain.com";

$domain = strstr($email,'@');

// echo $domain akan menghasilkan "@domain.com"

1. **strtolower() dan strtoupper()**

Kedua fungsi ini digunakan untuk mengkonversi string menjadi huruf kapital semua

atau huruf kecil semua. Sintaksnya adalah sebagai berikut:

Strtolower(string)

Strtoupper(string)

Contoh :

$str = "StrINg inI terDiRi DARi hurUF beSAr dan KECIl";

echo strtolower($str)."<br>";

echo strtoupper($str);

Outputnya :

string ini terdiri dari huruf besar dan kecil  
STRING INI TERDIRI DARI HURUF BESAR DAN KECIL

1. **strpos()**

Fungsi ini digunakan untuk mengetahui posisi sebuah string di dalam string yang lain.

Jika pada string tujuan terdapat lebih dari satu karakter yang dicari oleh karakter

pencari, maka karakter pertama yang ditemukan yang akan dipakai. Sintaksnya adalah

sebagai berikut:

Strpos(tujuan,pencari)

contoh :

$str = "bambang";

$cari = strpos($str,"b");

// echo $cari akan menghasilkan 0

1. **strrev()**

Fungsi ini digunakan untuk membalik urutan karakter-karakter penyusun string dari

depan ke belakang menjadi dari belakang ke depan. Dengan kata lain fungsi ini

digunakan untuk membaca string secara terbalik.

Strrev(string)

contoh

$str = "Belajar PHP itu mudah";

echo strrev($str);

//hasilnya adalah "hadum uti PHP rajaleB"

1. **str\_replace()**

Fungsi ini digunakan untuk mengganti suatu string dengan string yang lain.

Sintaksnya adalah sebagai berikut:

str\_replace(yang\_diganti,pengganti,tujuan)

Parameter yang\_diganti menunjukkan string yang akan diganti. Parameter pengganti

menunjukkan string yang akan menggantikannya. Parameter tujuan adalah string

secara keseluruhan yang didalamnya mengandung string yang\_diganti. Bisa juga

merujuk pada variabel tertentu yang berisi sebuah string.

Contoh:

$str = "Belajar PHP itu sulit";

echo str\_replace("sulit", "mudah", $str)

//hasilnya adalah "Belajar PHP itu mudah"

1. **substr()**

Fungsi ini digunakan untuk mengambil atau memotong suatu bagian sebuah string dan

menampilkannya sebagai sebuah string tersendiri. Sintaksnya adalah sebagai berikut:

substr(string,mulai[,panjang])

String akan diambil atau dipotong mulai dari karakter yang terletak pada nomor yang

ditunjukkan oleh parameter mulai dengan jumlah sebanyak angka yang ditunjukkan

oleh parameter panjang. Contoh:

$rest = substr ("abcdef", 1); // menghasilkan "bcdef"

$rest = substr ("abcdef", 1, 3); // menghasilkan "bcd"

Parameter mulai juga dapat diisi dengan negatif. Jika negatif, maka perhitungan

dimulai dari karakter yang paling belakang.

Contoh:

$rest = substr ("abcdef", -1); // menghasilkan "f"

$rest = substr ("abcdef", -2); // menghasilkan "ef"

$rest = substr ("abcdef", -3, 1); // menghasilkan "d"

1. **is\_numeric()**

Fungsi ini digunakan untuk mengecek apakah masukkan berupa nomor atau bukan.

Sintaksnya adalah sebagai berikut:

is\_numeric(masukkan)

Contoh:

$inputan = "027454321";

If (is\_numeric($inputan)) {

echo "Inputan berupa bilangan numeric";

} else {

echo "Inputan bukan berupa bilangan numeric";

}

**2. Operator Aritmatika, pertidaksamaan dan logika**

Dalam pemrograman tidak lepas untuk pengolahan dara dan berbagai keperluan. kita tidak bisa lepas dari operasi-operasi yang bersifat aritmatik, maupun operasi yang bersifat logika.

1. **Operasi aritmatika**

operasi aritmatika hanya untuk tipe data numerik.

|  |  |  |
| --- | --- | --- |
| **Operasi** | **Operator** | **Contoh Sintaks** |
| Penambahan | + | $x + $y |
| Pengurangan | - | $x – $y |
| Perkalian | \* | $x \* $y |
| Pembagian | / | $x / $y |
| Sisa Pembagian | % | $x % $y |
| Increment | ++ | $x++ |
| Decrement | -- | $x–- |

1. **Operator persamaan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Deskripsi** | **Contoh** | **Hasil Boolean** |
| == | Apakah sama | 2==3 | **False** |
| **!=** | Tidak sama | 2!=3 | **True** |
| **>** | Lebih besar dari | 2>3 | **False** |
| **<** | kurang dari | 2<3 | **True** |
| **>=** | Lebih besar atau sama dengan | 2>=3 | **False** |
| **<=** | Kurang dari atau sama dengan | 2<=3 | **True** |

1. **Operator Logika**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Deskripsi** | **Contoh** | **Hasil Boolean** |
| **&&** | Dan | x=3  y=2  (x <5 && y >1) | **True** |
| **||** | Atau | x=3  y=2  (x <5 || y >1) | **True** |
| **!** | Not Atau Negasi | x=3  y=2  !(x==y) | **True** |

**3. Kondisi**

kondisi atau suatu pernyataan digunakan untuk memilih aksi atau proses tertentu.

**a. if**

ada tiga bentuk kondisi if yaitu :

* **if(kondisi) {code program} .** kode program dijalankan apabila kondisi terpenuhi atau dalam keadaan true.
* **if(kondisi) {code program} else {code program lain}** . apabila kondisi terpenuhi, maka kode program satu dijalankan, namun apabila tidak , maka kode program lain yang dijalankan.
* **if(kondisi satu) {kode program} else if (kondisi dua) {kode program kedua} else{kode program ketiga}.** jika kondisi satu terpenuhi, maka kode program satu dijalankan, jika tidak, jika kondisi dua terpenuhi, maka kode program dua dijalankan, jika tidak terpenuhi keduanya, maka kode program tiga dijalankan

Contoh :

<?php

$x=5;

$y=3;

if($x==3){

echo "salah jika x==3";

}

else if (!($x==$y)) {

echo "Benar x tidak sama dengan y";

}

else

echo "yang lainnya"

?>

Outputnya:

Benar x tidak sama dengan y

**b. switch case**

digunakan untuk melakukan suatu proses berdasarkan suatu kondisi yang berbeda. sintaks umumnya seperti contoh berikut.

<?php

$nilai =3;

switch($nilai)

{

case 1:

echo "Angka 1";

break;

case 2:

echo "Angka 2";

break;

case 3:

echo "Angka 3";

break;

default:

echo "tidak ada angka yang cocok";

}

?>

Outputnya

Angka 3

**4. Array**

*Array* adalah sebuah data yang mengandung satu atau lebih data dan dapat diindek

berdasarkan numerik maupun *string*. Pada PHP tipe masing-masing data variabel *array*

tidak harus sama dan variabel *array* juga bisa menyimpan suatu variabel *array*

didalamnya. *Array* dapat dibuat dengan menggunakan fungsi *array()* dan *list()* atau dapat

dideklarasikan secara eksplist.

Jika digunakan angka sebagai indeks, maka angka yang digunakan dimulai dari nol.

Untuk mendeklarasikan array dengan indeks banyak dapat dipersingkat denganmengosongkan kurung siku. PHP akan otomatis memberikan angka yang berurutan.

contoh :

<?php

$a[0]="Saya";

$a[]="belajar";

$a[]="berhitung";

echo "$a[0]";

echo "$a[1]";

echo "$a[2]";

echo "<br>";

$a["aku"]["kamu"][3]="array lho";

echo "Saya belajar {$a["aku"]["kamu"][3]}";

?>

Outputnya :

Sayabelajarberhitung  
Saya belajar array lho

**5. Looping atau perulangan**

Di dalam PHP, looping atau perulangan mempunyai peranan penting, yang berfungsi untuk menampilkan informasi tertentu yang bersifat daftar atau untuk memproses suatu data yang membutuhkan perulangan tertentu. pada program PHP dikenal 4 jenis perulangan yaitu

**a. for**

syntax for :

for (inisiasi variable; kondisi; increment)

{

kode program yang dijalankan;

}

contoh :

<?php

for ($i=1; $i<=10; $i++)

{

echo "perulangan ke ".$i."<br>";

}

?>

Outputnya

perulangan ke1  
perulangan ke2  
perulangan ke3  
perulangan ke4  
perulangan ke5  
perulangan ke6  
perulangan ke7  
perulangan ke8  
perulangan ke9  
perulangan ke10

**b.foreach**

foreach digunakan dalam perulangan yang berbentuk daftar data yang dimasukkan ke dalam array.syntax for :

foreach ($array as $valueArray)

{

kode program yang dijalankan;

}

contoh :

<?php

$daftarnama=array("najib","sigit","sabana");

$i=1;

foreach($daftarnama as $nama ){

echo "Nama ke ".$i." : ".$nama."<br>";

$i++;

}

?>

Outputnya

Nama ke 1 : najib  
Nama ke 2 : sigit  
Nama ke 3 : sabana

**c. While**

Dengan while, perulangan akan dilakukan terus-menerus selama kondisi bernilai *true*, atau selama kondisinya masih terpenuhi . syntax :

while (kondisi)

{

kode program yang akan dieksekusi;

}

contoh :

<?php

$nilai=4;

$i=1;

while($nilai<=32)

{

echo "Nilai ke-".$i." Adalah ".$nilai."<br>";

$nilai = $nilai+5;

$i++;

}

Outputnya

Nilai ke1 Adalah 4  
Nilai ke2 Adalah 9  
Nilai ke3 Adalah 14  
Nilai ke4 Adalah 19  
Nilai ke5 Adalah 24  
Nilai ke6 Adalah 29

**d. do while**

hampir sama dengan perulangan **while, do while** memiliki bentuk umum seperti berikut:

do

{

kode program yang akan dieksekusi;

}

while (kondisi);

contoh :

<?php

$nilai=4;

$i=1;

do

{

echo "Nilai ke-".$i." Adalah ".$nilai."<br>";

$nilai = $nilai+5;

$i++;

}

while($nilai<=32)

?>

Outputnya

Nilai ke1 Adalah 4  
Nilai ke2 Adalah 9  
Nilai ke3 Adalah 14  
Nilai ke4 Adalah 19  
Nilai ke5 Adalah 24  
Nilai ke6 Adalah 29

**6. Function**

Function atau fungsi merupakan sebuah subprogram. pada dasarnya PHP sudah menyediakan banyak fungsi yang digunakan secara instan oleh programmer. sebagai contoh fungsi strlen() yang digunakan untuk mengetahui panjang *string.* fungsi juga dapat Anda buat dan definisikan sendiri . dengan mendifinisikan sebuah fungsi, maka kode program akan lebih mudah ditelusuri apabila terjadi kesalahan. bentuk syntax:

function namafunction($parameter)

{

kode program;

}

contoh :

<?php

function cetakTebal($angka)

{

if($angka==16)

{

echo "<b>"."Nilai ".$angka."</b>"."</br>";

}

else {

echo "Nilai ".$angka."</br>";

}

}

$nilai=4;

while($nilai<=32)

{

cetakTebal($nilai);

$nilai = $nilai+4;

}

?>

Outputnya

Nilai 4  
Nilai 8  
Nilai 12  
**Nilai 16**  
Nilai 20  
Nilai 24  
Nilai 28  
Nilai 32

contoh lain membuat fungsi bilangan berpangkat. :

<?php

function pangkat($bilangan,$pangkat)

{

$hasil=$bilangan;

$i=1;

while($i<$pangkat)

{

$hasil = $hasil \* $bilangan;

$i++;

}

return $hasil;

}

echo "2 pangkat 4 adalah <b><i>".

pangkat (2,4)."</b></i>";

?>

Outputnya

2 pangkat 4 adalah ***16***

**7. Session**

Di dalam sebuah aplikasi web, terutama yang memiliki jumlah halaman yang

banyak, seringkali dibutuhkan suatu kontinuitas dari informasi-informasi tertentu pada

saat pengunjung berpindah-pindah dari satu halaman ke halaman lain. Oleh karena itu

diperlukan suatu alternatif lain. Session adalah jawabannya. Session memiliki

kemampuan untuk menyimpan informasi secara kontinyu asalkan session itu masih

“hidup”.

Ada dua cara yang dapat digunakan dalam mengaktifkan session tanpa mengedit

php.ini, yaitu secara eksplisit dan secara implisit. Pengaktifan session secara eksplisit

dilakukan dengan fungsi **session\_start() ,** sedangkan secara implisit menggunakan fungsi

**session\_register()**.Ada beberapa fungsi dalam session,diantaranya yaitu:

**a. session\_start(),** Dengan menjalankan fungsi tersebut, maka sebuah session

akan dimulai. Fungsi ini tidak memerlukan parameterapapun, sehingga sintaks

penggunaanya adalah dengan langsung saja menuliskanfungsi session\_start()

tersebut.

**b. session\_register()** pada prinsipnya adalah “mendaftarkan” variabel sehingga

variabel-variabel tersebut akan dikenal sepanjang sebuahsessionsekaligus memulai

sebuah session apabila belum dimulai. Sintaks penggunaan fungsi session\_register()

adalah sebagai berikut:

**c. session\_unregister()** adalah kebalikan dari fungsi session\_register(),yaitu

untuk "mencabut" variabel dari sebuah session.

**d. session\_is\_registered()** digunakan untuk memeriksa apakah sebuah variabel

pernah didaftarkan pada sebuah sessionatau tidak.

**e. unset()** digunakan untuk menghapus variable

**f. isset()** digunakan untuk memeriksa apakah variabel tersebutsudah pernah

dideklarasikan atau belum.

**g. session\_destroy()** digunakan untuk menghapus seluruh informasi session ini.

contoh sederhana session :

<?php

session\_start();

$\_SESSION['nama'] = "Muhammad Hasio";

$\_SESSION['umur'] = "21";

$\_SESSION['email'] = "pemrogramanweb@sukses-selalu.com";

$\_SESSION['waktu'] = date("F j, Y, g:i a");

echo "Halo ".$\_SESSION['nama'].'<br>';

echo "Umur Anda ".$\_SESSION['umur'].'<br>';

echo "email anda ".$\_SESSION['email'].'<br>';

echo "Anda login pada ".$\_SESSION['waktu'].'<br>';

?>

Outputnya :

Halo Muhammad Hasio  
Umur Anda 21  
email anda pemrogramanweb@sukses-selalu.com  
Anda login pada May 21, 2013, 3:26 pm