

Pengkondisian

IF – ELSEIF – ELSE

Seperti namanya, kode PHP ini berfungsi sebagai penyeleksi kondisi “jika kondisi terpenuhi” (IF), “jika kondisi lain yang terpenuhi (ELSEIF, dari kata ELSE IF yang digabung), dan “jika semua kondisi tidak terpenuhi” (ELSE). Sebagai contoh, kode di bawah ini berisi dua variabel yang berisi nilai ber-type data integer (angka) dan nantinya akan di bandingkan menggunakan operator pembandingan.

Contoh:

```
<?php
```

```
if ($a > $b) {
```

```
    echo "$a lebih besar dari $b";
```

```
} elseif ($a == $b) {
```

```
    echo "$a sama dengan $b";
```

```
} else {
```

```
    echo "$a lebih kecil dari $b";
```

```
}
```

```
?>
```

Pengulangan (Loop)

FOR

Pengulangan FOR memberikan pernyataan bahwa eksekusi program yang akan dilakukan berulang sesuai dengan kode program yang dimasukkan pada FOR tersebut. Penulisan pengulangan FOR tentu berbeda dengan WHILE. Oleh karena itu, perhatikan format penulisan sintaks FOR berikut :

```
For (Initialization statement ; conditional statement ; repetition statement)
```

```
{
```

```
Code block;
```

```
}
```

Ketiga atribut tersebut yang pertama adalah *Initialization statement* menuliskan pernyataan di awal (kondisi awal) dari suatu iterasi. *Conditional statement* dituliskan seperti kondisi awal if, maksudnya ialah kondisi dimana iterasi tersebut adalah harus dilakukan kapan iterasi itu berakhir. Dan ketiga adalah *repetition statement*, atribut yang ketiga ini disebut sebagai pernyataan pengulangan karena bisa diisi dengan *Increment* (penambahan) atau pun *Decrement* (pengurangan) .

Contoh :

```
<?php
```

```
$varawal = 1;
```

```
$varakhir = 10;
```

```
$jumlah = 0;
```

```
for ($i = $varawal; $i <= $varakhir; $i++)
```

```
{
```

```
$jumlah += $i;
```

```
echo $jumlah.", " ";
```

```
}
```

```
echo "<br>";
```

```
echo $jumlah;
```

```
?>
```

WHILE

Perulangan dengan WHILE adalah perulangan yang paling sederhana di PHP. Bentuk dasar dari perulangan menggunakan WHILE adalah sebagai berikut :

```
while (ekspresi) {
```

```
    pernyataan
```

```
}
```

Perulangan WHILE hanya membutuhkan satu ekspresi yang akan di periksa setiap awal perulangan di lakukan, sehingga apabila sejak awal ekspresi tidak memenuhi kondisi yang di persyaratkan (bernilai FALSE), maka pernyataan tidak akan di proses sama sekali. Lihat contoh kode PHP untuk perulangan sederhana di bawah ini, variabel \$i bernilai 1 dan ekspresi (kondisi yang di persyaratkan) adalah variabel \$i bernilai kurang dari sama dengan 10, artinya variabel \$i harus bernilai 1 hingga 10, jika tidak maka ekspresi bernilai FALSE dan perulangan akan di hentikan / tidak akan di jalankan.

Contoh :

```
<?php
```

```
$i = 1;
```

```
while ($i <= 10) {

    echo $i++;

}

?>
```

TIPE DATA

1.INTEGER

Tipe data integer digunakan untuk menyimpan bilangan bulat yang berisikan bilangan bulat negatif dan positif atau tanpa bilangan pecahan. Contoh 1,2,3,4,-1,-2,-3,-4
Pembuatan kode : \$namakode = 45;

2.FLOAT

Tipe Double Floating biasa disebut dengan double,float,real merupakan tipe data yang digunakan untuk menyimpan desimal,contoh : 0.1,1.2
Penulisan kode : \$namakode = 2.5;

3.STRING

Tipe data String merupakan sebuah tipe data yang terdiri dari kata/huruf.baik tunggal maupun kalimat,penulisan string diapit oleh petik ganda ("") dan petik tunggal (' ').
Contoh pembuatan kode :
-\$namakode = 'tulisan string ini petik tunggal';
-\$namakode = "tulisan string ini petik ganda";

Note: jika menggunakan petik tunggal,varibel yang kita buat tidak tampil,pada pembuatan petik ganda akan tampil di browser.

4.ARRAY

Array adalah sebuah objek tabel dengan index berurutan. Analoginya mirip gerbong Kereta mempunyai nama kereta(nama array,nomor gerbong(index),gerbong(elemen array),dan jenis kereta(tipe data elemen). Index array diawali dengan angka 0,
Jenis array ada 3 yaitu : Numeric Array, Associative Array ,dan Multidimensional Array.
Contoh :

Numerik Array :

```
<?php
$namakode[0] = "hello";
$namakode[1] = "dunia";
Echo $namakode[0]." " . $namakode[1];
```

Associative Array

```
<?php
$age= array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
echo "Peteris      " .      $age['Peter']      . "      years      old.";
?>
```

Multidimensional Array

```
<?php
$kelompok = array
(
    array("faldo","junaidi","bahrudin"),
    array("sinta","suci","yunita"),
    array("micel","andini","mumun"),
    array("arif","iik","haerudin")
);

echo $kelompok[0][0]."-".$kelompok[0][1]."".$kelompok[0][2].".<br>";
echo $kelompok[1][0]."-".$kelompok[1][1]."".$kelompok[1][2].".<br>";
echo $kelompok[2][0]."-".$kelompok[2][1]."".$kelompok[2][2].".<br>";
echo $kelompok[3][0]."-".$kelompok[3][1]."".$kelompok[3][2].".<br>";
?>
```

METHOD POST DAN GET

Ada 2 metode ketika anda ingin memparsing nilai inputan dari suatu file ke file yang lain, yaitu dengan memanfaatkan METHOD POST atau GET.

METHOD GET sendiri memiliki pengertian bahwa metode pengiriman data menggunakan query string ,

Jadi semua nilai pada form anda akan dikirim ke sisi server yang kemudian diterima oleh file tertentu ,

Tetapi nilai form itu akan tampil pada baris URL/addres bar

Contoh method get yang akan ditampikan pada addres bar browser anda akan seperti ini;

localhost/adminValidate.php?username=admin&password=demo

AdminValidate.php adalah file yang menerima nilai dari query string

Dan disana terdapat dua buah variabel username dan password , dimana "=" digunakan sebagai perintah untuk mengisi nilai variabel tersebut

Contoh penulisan program untuk mengambil nilai dari query string tersebut adalah seperti dibawah ini

```
<?php
```

```
echo $_GET(username); //menampilkan nilai variabel get username
```

```
<?php
```

```
Echo $_GET(password); //menampilkan nilai variabel get password
```

Sedangkan method post digunakan untuk memarsing hasil inputan form tetapi tidak menampilkan query string pada adres bar layaknya method get.

Method post akan sangat berguna penggunaanya untuk halaman – halaman yang membutuhkan keamanan lebih . karena data inputan kita tidak akan tampil pada adres browser .

Contoh (method post) pada adres bar tidak akan menampilkan informasi data input

```
<?php
```

```
echo $_POST(username); //menampilkan nilai variabel get username
```

```
echo $_POST(password); //menampilkan nilai variabel get password
```

CATATAN

Ketika anda menggunakan method get , maka query string yang ditampilkan pada adres bar hanya mampu menampung 255 karakter , oleh karena itu perhatikan banyaknya inputan ketika akan menggunakan method get.

FUNGSI VALIDASI

EMPTY

Kode empty pada php merupakan salah satu fungsi built-in yang berfungsi untuk memvalidasi nilai suatu variabel apakah kosong atau telah memiliki nilai, jadi fungsi EMPTY memeriksa nilai dari variabel yang sudah ada , bahkan memeriksa apakah suatu variabel ada atau tidak ada.

```
<?php
```

```
$usia = 0;
```

```
If (empty($usia))
```

```
{
```

```
echo 'variabel usia bernilai kosong atau 0' ;
```

```
}
```

ISSET

Kode isset pada php merupakan salah satu fungsi built-in yang berfungsi memvalidasi eksistensi suatu variabel apakah ada atau tidak ada,

Berbeda dengan fungsi empty yang memeriksa nilai dari variabel yang sudah ada

Fungsi ISSET memeriksa apakah variabel eksis (ada) atau tidak ada tanpa menghiraukan nilai yang ada dalam variabel tersebut (KECUALI NULL).

CONTOH

```
<?php
$nama = '';
If (isset ($nama))
{
echo "variabel nama ada , evaluasi bernilai TRUE";
}
If (isset($alamat))
{
Echo "variabel alamat tidak ada, evaluasi bernilai FALSE";
}
?>
```

UNSET

Kode UNSET pada PHP merupakan salah satu fungsi built-in dengan fungsi utama untuk menghilangkan suatu variabel termasuk nilai yang ada di dalamnya. Perhatikan potongan kode di bawah ini :

```
<?php

$nama="";

if (isset($nama)) {

echo "Variabel NAMA ada.<br>";

} else {

echo "Variabel NAMA tidak ada.";

}

unset($nama);

if (isset($nama)) {

    echo "Variabel NAMA ada.";

}else {

    echo "Variabel NAMA tidak ada.";

}

?>
```