2.1. Tipe data dan Variabel pada PHP

PHP mendukung tipe data berikut ini:

- integer
- double
- string
- array
- object

Tipe variabel ini tidak perlu ditentukan oleh programmer, melainkan ditentukan pada saat runtime oleh PHP dengan kebergantungan pada konteks pemakaian variabel tersebut.

2.2. Menginisialiasi Variabel

Untuk menginisialisasi suatu variabel dalam PHP, secara sederhana berikan saja suatu nilai padanya. Tetapi untuk tipe seperti array dan objek membutuhkan mekanisme yang berbeda.

```
$nama = "Ilham";
$umur = 6;
$berat = 20.5;
```

2.3. Menginisialisasi Array

Array dapat diinisialisasi dengan dua cara: dengan pemberian nilai langsung, dan menggunakan konstruksi array() (yang akan dijelaskan pada bagian Fungsi).

Untuk memberikan nilai secara langsung pada suatu array, secara sederhana adalah memberikan nilai ke variabel array dengan subscript kosong. Nilai tersebut akan ditambahkan sebagai elemen terakhir dari array.

```
$names[] = "Ilham";      // $names[0] = "Ilham"
$names[] = "Melati";     // $names[1] = "Melati"
```

Sesuatu yang perlu diingat adakah elemen array dimulai dari index 0 (bukan 1)

2.4. Menginisialiasi Objek

Untuk menginisialisasi suatu object, anda dapat mengunakan perintah new. Perintah ini digunakan untuk menginisialisasi suatu object kepada suatu variabel.

```
class kosong {
   function masa_bodo () {
      echo "Tidak melakukan apa-apa.";
   }
}
$bar = new kosong;
$bar -> masa_bodo ();
```

2.5. Jangkauan dari variabel

Jangkauan dari suatu variabel adalah tergantung pada dimana variabel tersebut didefinisikan. Pada umumnya semua variabel PHP hanya memiliki suatu jangkauan. Setiap variabel yang digunakan dalam suatu fungsi, maka secara default jangkauannya adalah lokal. Sebagai contoh:

```
$awal = 1;                              /* jangkauan global */

function Coba () {
   echo $awal;                  /* mengacu pada variabel jangkauan lokal */
}
Coba ();
```

Script diatas tidak mengeluarkan apapun, karena $awal dalam fungsi Coba mengacu pada $awal lokal yang tidak memiliki nilai apapun. Hal ini berbeda dengan bahasa C dimana setiap variabel global akan berlaku bagi semua function, kecuali didefinisikan sebagai local. Dalam PHP, variabel global harus dideklarasikan sebagai global didalam suatu fungsi jika mereka akan digunakan dalam fungsi tersebut, contoh :

```
$panjang = 10;
$lebar = 5;
function Hitungluas () {
   global $panjang, $lebar;$luas
   $luas = $panjang *  $lebar;
}
Hitungluas ();
echo $luas;
```

Script diatas akan menghasilkan keluaran "50". Dengan mendeklarasikan $panjang dan $lebar sebagai global didalam fungsi.
Cara kedua untuk mengakses variabel sebagai acuan global adalah dengan mengunakan definisi $GLOBAL array pada PHP. Contoh sebelumnya dapat juga ditulis menjadi:

```
$panjang= 1;
$lebar= 2;
function Hitungluas () {
   $GLOBALS["luas"] = $GLOBALS["panjang"] * $GLOBALS["lebar"];
}
Hitungluas ();
echo $luas;
```

$GLOBALS array adalah suatu asosiasi array dengan key adalah nama dari variabel global.
Hal lain yang penting dari jangkuan adalah variabel static. Suatu variabel hanya ada pada jangkauan lokal fungsi, dan nilainya akan tetap dipertahankan:

```
function Coba () {
   $a = 0;
   echo $a;
   $a++;
}
```

Pada fungsi diatas nilai $panjang kembali menjadi 0 untuk setiap pemanggilan. Artinya nilai $panjang akan hilang begitu program keluar dari fungsi. Contoh berikut menggunakan variabel static.

```
Function Coba () {
   static $a = 0;
   echo $a;
   $a++;
}
```

Sekarang, setiap kali fungsi Coba(), dipanggil maka nilai $a adalah nilai $a pada pemanggilan sebelumnya.

2.6. Variabel variabel

Kadang-kandang adalah lebih nyaman mengunakan variable-variabel; Yang mana adalah nama variabel yang dapat digunakan secara dinamis. Secara normalnya variabel dibuat dengan :

```
$a = "hello";
```

Suatu variabel variabel akan mengunakan nilai dari suatu variabel menjadi nama variabel, Contoh:

```
$a = "hello";
$$a = "world";
```

Pada contoh diatas akan terbentuk dua variabel yaitu $a dan $hello. Dimana $a berisi "hello" dan $hello berisi "world". Selanjutnya perintah berikut ini:

```
echo "$a ${$a}";
```

atau

```
echo "$a $hello"
```

Akan menghasilkan keluaran : hello world

2.7. Penentuan tipe variabel

PHP tidak membutuhkan deklarasi variabel secara eksplisit; tipe variabel ditentukan berdasarkan konteks pemakaiannya pada saat runtime. Dengan kata lain; jika anda memberi nilai string kesuatu variabel var, var menjadi suatu variabel tipe string. Jika anda memberi nilai integer ke var, maka otomatis berubah menjadi tipe integer.

Suatu contoh dari otomatisasi konversi tipe pada PHP adalah operator penjumlahan '+'. Jika salah satu operandnya adalah tipe double, maka semua operand lainya dievaluasi sebagai double dan hasilnya adalah double.

```
$coba = "0";                    // $coba adalah string (ASCII 48)
$coba++;                        // $coba adalah string "1" (ASCII 49)
$coba += 1;                     // $coba sekarang adalah integer (2)
$coba = $coba + 1.3;            // $coba sekarang adalaj double (3.3)
$coba = 5 + "10 kotak"; // $coba adalah integer (15)
$coba = 5 + "10 kotak kecil";   // $coba adalah integer (15)
```

2.8. Mengetahui tipe suatu variabel

Karena PHP menentukan tipe variabel sesuai dengan konteks pemakaiannya, maka anda dapat menggunakan fungsi berikut untuk memeriksa tipe pada suatu variabel gettype(), is_long(), is_double(), is_string(), is_array(), dan is_object().

2.9. Tipe casting

Tipe casting dalam PHP bekerja seperti dalam C: nama dari tipe yang diinginkan ditulis didalam kurang sebelum variabel yang akan di cast.

```
$coba = 10;                     // $coba adalah suatu integer
$bar = (double) $coba;          // $bar adalah suatu double
```

2.10. Cast yang diperbolehkan:

(int), (integer) - cast ke integer
(real), (double), (float) - cast ke double
(string) - cast ke string
(array) - cast ke array
(object) - cast ke object

2.11. Konversi String

Ketika suatu string dievaluasikan sebagai suatu nilai numerik, nilai hasil dan tipe ditentukan sebagai berikut.
String akan dievaluasikan sebagai suatu double jika mengandung salah satu karakter '.', 'e' tatau 'E'. Jika tidak akan dievaluasikan sebagai suatu string.
Jika string dimulai dengan data numerik yang sah, maka nilai tersebut akan digunakan, jika tidak akan memiliki nilai 0 (nol).

```
$coba = 1 + "10.5";         // $coba adalah double (11.5)
$coba = 1 + "-1.3e3";       // $coba adalah double (-1299)
$coba = 1 + "bob-1.3e3";            // $coba adalah integer (1)
$coba = 1 + "bob3";         // $coba adalah integer (1)
$coba = 1 + "10 Small Pigs";      // $coba adalah integer (11)
$coba = 1 + "10 Little Piggies"; // $coba adalah integer (11);
$coba = "10.0 pigs " + 1;         // $coba adalah int (11)
$coba = "10.0 pigs " + 1.0;       // $coba adalah double (11)
```

Tipe dari variabel tergantung pada ekspresi kedua, jika ekspresi pertamanya adalah string

2.12. Menangani variabel Form, Cookies dan Environment

Pada pemrograman CGI, program kita akan berinteraksi dengan variabel-variabel dari luar yang dikirim melalui form baik dengan metode GET maupun metode POST.
Ketika suatu form dikirim ke suatu PHP script, semua variabel dari form secara otomatis dapat diproses oleh script PHP sebagaimana variabel biasanya. Sebagai contoh, perhatikan form berikut ini:

```
<form action="kosong.php" method="post">
   Nama: <input type="text" name="nama"><br>
   <input type="submit" value="Kirim">
</form>
```

Top of Form

Nama: [ ]

Submit

Bottom of Form

Ketika form disubmit, maka PHP akan membuat variabel $nama, yang mana mengandung apa yang diketikkan pada field Nama: di form tersebut.
PHP juga mendukung variabel array dalam kontek form, tetapi dibatasi hanya 1 dimensi, Contoh:

```
<form action="array.html" method="post">
   Nama: <input type="text" name="personal[nama]"><br>
   Email: <input type="text" name="personal[email]"><br>
   Beer: <br>
   <select multiple name="beer[]">
      <option value="warthog">Warthog
      <option value="guinness">Guinness
      </select>
   <input type="submit">
</form>
```

Top of Form

Nama :

Email :

Beer:

Submit

Bottom of Form

2.13. Variabel pada IMAGE SUBMIT

Ketika mengirim suatu form, juga dimungkinkan untuk menggunakan suatu gambar sebagai pengganti tombol submit dengan tag HTML berikut ini:

<input type=image src="image.gif" name="sub">

Ketika pemakai melakukan klik pada gambar tersebut, makan form akan dikirim ke CGI dengan dua variabel tambahan, yaitu sub_x dan sub_y. Kedua variabel ini adalah koordinat dimana klik dilakukan pada gambar.

2.14. HTTP Cookies

PHP secara transparan mendukung HTTP cookies. Cookies adalah suatu mekanisme penyimpanan data secara remote pada browser klien. Hal ini dapat digunakan untuk mengidentifikasikan pemakai pada kunjungan berikutnya.

Anda dapat menggunakan fungsi SetCookie(). Cookie adalah bagian dari HTTP header, jadi fungsi SetCookie harus dipanggil sebelum output lainnya dikirim ke browser. Hal ini adalah sama batasannya untuk fungsi Header(). Semua cookies yang dikirim pada anda dari klien akan secara otomatis di ubah kedalam suatu variabel PHP seperti method GET dan POST data.

Jika anda ingin memberikan banyak nilai pada suatu cookie tunggal, tambahkan saja [] pada nama cookie.

Sebagai contoh:

SetCookie ("MyCookie[]", "Testing", time()+3600);

Catatan bahwa suatu cookie akan menimpa cookie sebelumnya yang memiliki nama yang sama dalam browser anda, kecuali path atau domainnya berbeda. Jadi untuk suatu aplikasi shopping cart anda perlu menyimpan suatu counter dan mengirimnya bersamaan.

Contoh.

```
$Count++;
SetCookie ("Count", $Count, time()+3600);
SetCookie ("Cart[$Count]", $item, time()+3600);
```

2.15. Variabel Environment

PHP secara otomatis membuat variabel lingkungan normalnya seperti variabel PHP.

```
echo $HOME;  /* Shows the HOME environment variable, if set. */
```

Sejak informasi datang dengan GET, POST dan mekanisme Cookies secara secara otomatis menjadi variabel PHP, adalah lebih baik membaca variabel langsung dari lingkungan untuk mendapatkan versi yang sebenarnya. Fungsi getenv() dapat digunakan untuk melakukan hal ini. Anda dapat juga dapat membentuk suatu variabel lingkungan dengan fungsi putenv().

2.16. Konstruksi Bahasa PHP

Suatu PHP script terdiri dari sejumlah perintah yang berurutan. Suatu perintah dapat berupa pemberi nilai, pemanggilan terhadap fungsi, perulangan, perintah kondisi dan baris kosong. Setiap perintah diakhiri dengan sebuah titik koma (;). Dan beberapa perintah dapat dikelompokan menjadi satu dalam kurung kurawal {dan }.
Catatan : Setiap perintah diakhir dengan sebuah titik koma (;).

Konstanta

PHP memiliki sejumlah konstanta yang telah didefinisikan, dan anda diberi kesempatan untuk membuat konstanta sesuai dengan kebutuhan anda.
Konstanta yang telah didefinisikan adalah __FILE__ (nama file yang sedang diproses) __LINE__ (nomor baris dari file yang sedang diproses)

Contoh :

```
<?php
function report_error($file, $line, $message) {
   echo "Suatu kesalahan terjadi pada file $file baris $line: $message.";
}
report_error(__FILE__,__LINE__, "Telah terjadi suatu kesalahan!");
?>
```

Anda dapat mendefinsikan konstanta ciptaan dengan menggunakan fungsi define().

Contoh :

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // akan mencetak "Hello world."
?>
```

2.17. Ekspresi

Hampir semua yang anda tulis dalam PHP script adalah ekspresi. Definisi yang paling mudah dari ekspresi adalah "segala sesuatu yang memiliki nilai".
Contohnya yang paling sederhana dari ekpresi adalah konstanta dan variabel. Ketika anda mengetik "$a = 5", anda telah memberikan nilai '5' ke $a.
Contoh yang lebih komplek untuk suatu ekspresi adalah fungsi. Untuk jelasnya perhatikan contoh berikut:

```
function hello () {
   return 5;
}
```

maka kalau anda menulis $c = hello() adalah sama dengan memberikan nilai 5 ke variabel $c, karena fungsi hello mengembalikan 5. Diatas adalah contoh fungsi yang sederhana.
PHP mendukung tiga tipe nilai skalar yaitu : nilai integer, floating point dan string. (nilai skalar adalah nilai yang tidak dapat dibagi menjadi bagian yang lebih kecil, seperti array). PHP juga mendukung dua tipe nilai bukan skalar yaitu array dan objek.