

PENGENALAN PEMROGRAMAN PHP

3/10/2011

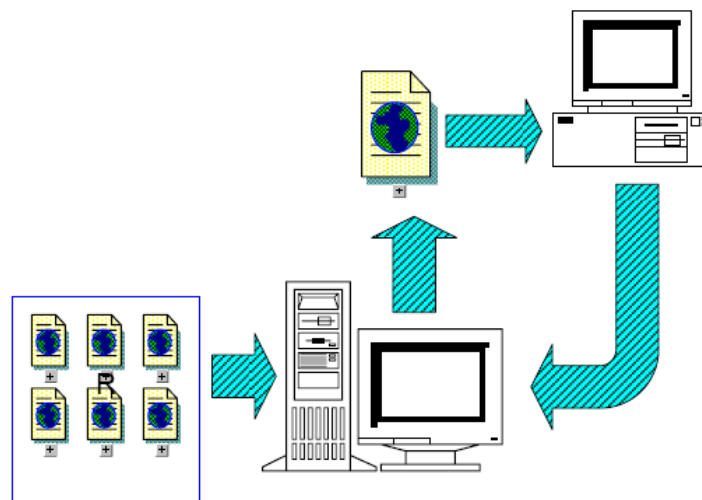
[Type the company name]

Imanuel Harkespan, S. Kom

1. Pengenalan PHP

PHP, 'PHP: Hypertext Preprocessor', merupakan bahasa pemrograman web dinamis layaknya ASP, JSP, Perl dan sebagainya. Lalu mengapa harus menggunakan bahasa pemrograman, apa HTML saja tidak cukup ? Ya, tentu saja HTML tidak cukup, salah satunya adalah karena sifat HTML yang statis. HTML hanya berguna untuk desain, saat *browsing*, sebenarnya yang terlihat adalah interpretasi dari HTML. Sebagai contoh saat menggunakan YahooMail untuk mengirim email, maka tampilan yang terlihat merupakan interpretasi dari HTML, sedangkan untuk dapat mengirim email, digunakan bahasa pemrograman web seperti PHP. Konsep dinamis yang dimaksud di sini bukan berarti web yang dikunjungi dapat bergerak-gerak, tetapi web tersebut dapat berubah-ubah tanpa mengubah skrip yang ada dalam *file*-nya.

PHP tergolong sebagai bahasa pemrograman yang berbasis *server* (*server-side scripting*). Ini berarti bahwa semua script PHP diletakkan di *server* dan diterjemahkan oleh *webserver* terlebih dahulu, kemudian hasil terjemahan itu ditampilkan di *browser client*.



Gambar 1. Standar Web Architecture

2. Koding Dasar PHP

a. HTML Escaping

Ada beberapa jenis pasangan tag pembuka dan penutup yang digunakan diantaranya adalah `<?php ... ?>` (tag penuh) dan `<? ... ?>` (tag sebagian). Keduanya akan sama-sama bekerja pada saat file php dijalankan, namun untuk hasil yang lebih maksimal disarankan untuk menggunakan tag penuh.

Sebuah file PHP dapat berisi sintaks-sintaks HTML dan beberapa kode skrip PHP. Berikut adalah contoh penggunaan skrip PHP sederhana yang dimasukkan ke dalam tag HTML.

```
<html>
<body>
<?php
echo "Hello World";
?>
</body>
</html>
```

Skrip 1 Hello World

Skrip di atas akan mengirimkan tulisan “Hello World” ke dalam browser client. Setiap akhir baris PHP selalu diakhiri oleh titik koma (;). Titik koma merupakan tanda pemisah sebuah set instruksi dengan instruksi lainnya.

b. Komentar

Terkadang pemrogram membutuhkan beberapa baris kalimat untuk memberikan keterangan pada suatu baris program yang disebut remarks. Remarks tidak akan ikut dieksekusi oleh server.

Ada tiga macam *style* penulisan komentar program yang dapat digunakan, yaitu :

- C *style*, komentar diawali dengan tag `/*` dan diakhiri dengan `*/`, style ini digunakan untuk komentar yang lebih dari satu baris.
- C++ *style*, komentar diawali dengan tag `//` dan hanya berlaku untuk satu baris komentar, untuk baris berikutnya harus diawali tag `//` lagi.
- Bourne Shell *style*, diawali dengan tag `#` untuk satu baris komentar.

```
<?php
echo "Halo semua!<br>"; // Ini komentar C style;
/*
Ini komentar
yang lebih dari satu baris
*/
echo "Halo lagi!<br>"; # Ini komentar shell-style
?>
```

Skrip 2 Komentar Program

c. Tipe Data

PHP mengenal beberapa jenis tipe data, misalnya integer, float (real), array, string, dan object. Di PHP tidak diperlukan pendeklarasian tipe data seperti halnya di pemrograman C/C++, Pascal, dan Java.

Integer dapat dituliskan dengan notasi desimal, oktal, atau heksadesimal. Angka *floating-point* dapat dituliskan dalam notasi standar atau *scientific*.

```
<?php
$int1 = 523; // desimal
$int2 = -523; // desimal negatif
$int3 = 01013; // representasi oktal dari 523
$int4 = 0x20B; // representasi heksa dari 523
$float1 = 523.197; // notasi float standar
$float2 = 5.23197e2; // notasi float scientific
// Cetak semua angka
echo "$int1<br>$int2<br>$int3<br>$int4<br>$float1<br>$float2";
?>
```

Skrip 3 Representasi Angka

String dalam PHP dibatasi oleh kutip ganda (") atau kutip tunggal ('). Perbedaanannya adalah bagaimana *string* tersebut diinterpretasikan. Jika string dibatasi oleh kutip ganda, variabel di dalam *string* akan ditampilkan.

```
<?php
$makanan = "eskrim";
echo "Saya suka $makanan<br>";
echo 'Saya suka $makanan';
?>
```

Skrip 4 Contoh Penggunaan String

Array dalam PHP dapat berupa *indexed array* (vektor) dan *hash tables* (*associative arrays*). PHP juga mendukung array multidimensi.

```
<?php
// Membuat array sederhana secara eksplisit
$a[0] = "Buku";
$a[1] = "Pinsil";
$a[] = "Bolpen"; // secara implisit ditempatkan pada indeks
// (key) 3
$a[] = "Penghapus"; // secara implisit ditempatkan pada indeks
// (key) 4
echo "$a[0], $a[1], $a[2], $a[3]<br>";
// Membuat array asosiatif
$tanaman["pangan"] = "padi";
$tanaman["hias"] = "anggrek";
$tanaman["sayuran"] = "wortel";
echo "Contoh tanaman hias adalah {$tanaman["hias"]}<br>";
```

```
// Membuat array asosiatif spt di atas, namun caranya lebih
// sederhana
$tanaman = array("pangan" => "padi", "hias" => "anggrek",
"sayuran" => "wortel");
echo "Contoh tanaman sayuran adalah {$tanaman["sayuran"]}<br>";
// Membuat array multidimensi
$b[0][0] = "Nol Nol";
$b[0][1] = "Nol Satu";
echo "Nilai \${b[0][1]} adalah {\${b[0][1]}}<br>";
// Membuat array asosiatif multidimensi
$kota["Jabar"][0] = "Bogor";
$kota["Jabar"][1] = "Bandung";
$kota["Jateng"][0] = "Semarang";
$kota["Jateng"][1] = "Solo";
echo "\${kota['Jabar']}[1] = {\${kota['Jabar']}[1]}<br>";
?>
```

Skrip 5 Inisialisasi dan penggunaan array

Object adalah sebuah tipe data yang dapat berupa sebuah bilangan, variabel, atau bahkan sebuah fungsi.

```
<?php
class tanaman
{
    function tanaman()
    {
        $this->komoditas = "apel";
    }
    function set_var($komoditas)
    {
        $this-> komoditas = $komoditas;
    }
}
$buah = new tanaman();
echo $buah -> komoditas;
$buah -> set_var("mangga");
echo "<br>".$buah -> komoditas;
?>
```

Skrip 6 Penggunaan Tipe Data Object

d. Variabel dan Konstanta

Variabel PHP diawali dengan tanda dolar (\$) kemudian diikuti dengan nama variabel. Semua nama variabel bersifat *case-sensitive*. Nama variabel yang valid harus diawali oleh huruf atau garis bawah yang diikuti oleh sejumlah huruf, angka, atau garis bawah.

```

<?
$variabel1 = "Printer";
$variabel2 = "Epson";
echo "$variabel1, $variabel2";
$1variabel = 123; // nama variabel yang salah
$_tes = "tes"; // benar, diawali oleh garis bawah
$test_Ç = "tes2"; // benar, Ç adalah ASCII 128
?>

```

Skrip 7 Penamaan Variabel

e. Struktur Kontrol

Skrip PHP dibangun oleh serangkaian pernyataan. Pernyataan tersebut bisa berupa pemberi nilai (*assignment*), pemanggilan fungsi, perulangan (*loop*), pernyataan kondisi, maupun pernyataan kosong. Pernyataan-pernyataan tersebut dapat diatur alur kerjanya dengan struktur kontrol. Berikut ini akan dijelaskan jenis struktur kontrol yang ada dalam PHP.

- IF

Pernyataan if merupakan percabangan bersyarat yang berfungsi untuk melewati suatu proses. Jika syarat terpenuhi akan dilakukan proses berikutnya. Proses bisa berupa satu instruksi atau beberapa instruksi.

Struktur if :

```

if (syarat)
{
    instruksi;
    instruksi;
    instruksi;
}

```

- If...else

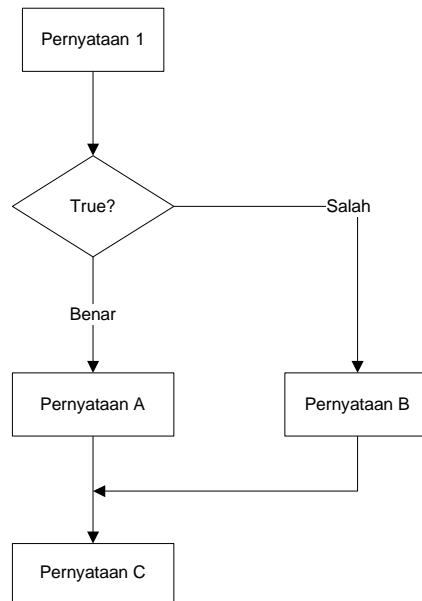
Struktur if..else memiliki dua alur percabangan sehingga memiliki alternatif. Jika syarat tidak terpenuhi, maka akan dijalankan instruksi lainnya.

Struktur if...else

```

if (syarat)
{
    instruksi;
    instruksi;
}
else
    instruksi;

```



Gambar 2 Pernyataan if...else

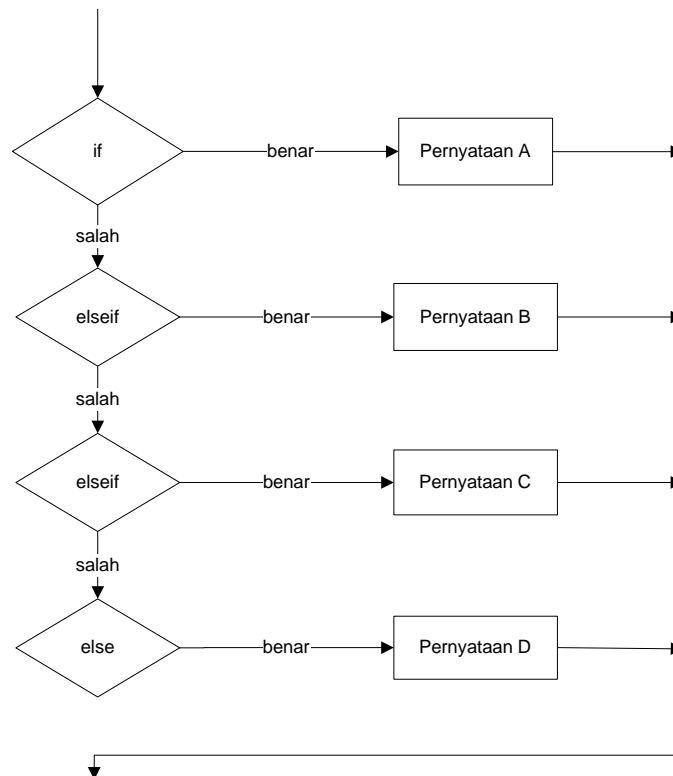
- If...elseif...else

Pernyataan if...elseif...else merupakan pengembangan dari pernyataan if...else, namun memiliki percabangan lebih dari dua. Pernyataan ini sesuai untuk pemecahan masalah yang membutuhkan banyak percabangan karena banyak alternatif yang diperoleh.

Struktur if...elseif...else

```

if (syarat)
{
    instruksi;
    instruksi;
}
elseif (syarat)
{
    instruksi;
    instruksi;
}
elseif (syarat)
{
    instruksi;
}
else
{
    instruksi;
}
  
```



Gambar 3 Pernyataan if...elseif...else

```

<?php
if (1 < 2)
echo "Ini akan dicetak.<br>";
else
echo "Ini tidak akan dicetak<br>";
$nilai = 2;
if ($nilai == 1)
{
// Gunakan kurawal untuk membatasi blok statement
// Hanya diperlukan untuk blok statement ganda (> 1)
echo "\$nilai == 1<br>";
}
elseif ($nilai == 2)
{
echo "\$nilai == 2<br>";
}
elseif($nilai == 3)
{
echo "\$nilai == 3<br>";
}
else
{
echo "\$nilai bukan 1, 2 atau 3<br>";
}
?>

```

Skrip 8 Contoh penggunaan if, if...else, dan if...elseif...else

f. Perulangan

- While

Perulangan while akan melakukan pengecekan syarat di awal blok perulangan. Selama syarat bernilai benar, berulangan akan terus berlanjut. Sebaliknya, jika syarat bernilai salah, perulangan akan dihentikan.

Struktur while :

```
while (syarat)
{
    instruksi;
    instruksi;
}
```

- Do..while

Meskipun serupa dengan pengulangan while, pengulangan do..while memeriksa kondisi pengulangan setelah eksekusi pertama dari *statement* pengulangan. Hal ini memastikan bahwa *statement* pengulangan akan dieksekusi paling sedikit satu kali.

Struktur do...while :

```
do
{
    Instruksi;
    Instruksi;
}
while (syarat)
```

```
<?php
echo "Menghitung dari kecil ke besar menggunakan
<b>while</b>.<br>";
$indeks = 0;
// cetak angka dari 0 sampai 9
while ($indeks < 10)
{
    echo "$indeks<br>";
    $indeks++;
}

echo "Menghitung dari besar ke kecil menggunakan
<b>do..while</b>. <br>";
// cetak angka dari 10 sampai 1
do
{
    print "$indeks<br>";
    $indeks--;
} while ($indeks > 0);
?>
```

Skrip 9 Contoh Penggunaan while dan do...while

- **For**

Pengulangan *for* adalah *statement* pengulangan yang paling kompleks di PHP.

Sintaksnya adalah sebagai berikut :

```
for (expr1; expr2; expr3)
{
    Statement;
    Statement;
}
```

Ekspresi pertama (*expr1*) dievaluasi sekali pada awal pengulangan. Ekspresi kedua (*expr2*) dievaluasi pada tiap awal iterasi. Jika nilainya *true*, pengulangan berlanjut, dan *statement(s)* dieksekusi. Pada akhir tiap iterasi, ekspresi ketiga (*expr3*) dievaluasi. Salah satu dari ketiga ekspresi tersebut dapat dikosongkan.

```
<?php
// cetak angka dari 0 sampai 9
for ($indeks = 0; $indeks < 10; $indeks++)
{
    echo "$indeks<br>";
}
/*
$indeks sekarang bernilai 10. Contoh di bawah ini
menunjukkan bahwa salah satu dari ketiga ekspresi
dapat dikosongkan.
*/
for (; $indeks > 0; $indeks--)
{
    echo "$indeks<br>";
}
?>
```

Skrip 10 Contoh penggunaan *for*

- ***Foreach***

Statement foreach adalah mekanisme yang baik untuk mengiterasi *array*. PHP mendukung dua sintaks :

```
foreach (ekspresi_array as nama_variabel_nilai) statement
foreach (ekspresi_array as nama_variabel_kunci => nama_variabel_nilai)
statement
```

```

<?php
$warna = array( "Merah", "Hijau", "Biru" );
foreach ($warna as $nilai)
{
    echo "Nilai warna saat ini adalah $nilai<br>";
}
$warna = array ( "Merah" => "#FF0000",
    "Hijau" => "#00FF00",
    "Biru" => "#0000FF" );
foreach ($warna as $kunci => $nilai)
{
    echo "Nilai heksa dari warna $kunci adalah $nilai<br>";
}
?>

```

Skrip 11 Penggunaan foreach

- Switch

Statement switch menyederahanakan evaluasi banyak kondisi. *Statement* ini sering digunakan untuk menggantikan *statement* if..elseif yang kompleks. Dalam PHP, *string* dapat digunakan sebagai ekspresi switch.

```

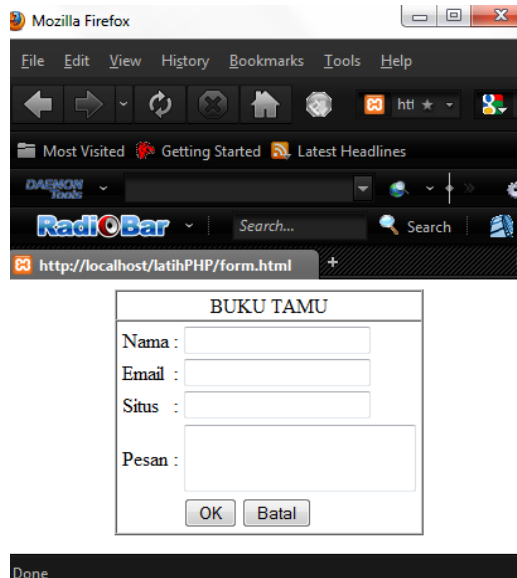
<?php
$indeks = 17;
switch ($indeks)
{
    case 0: echo "Nol<br>";
            break;
    case 1: echo "Satu<br>";
            break;
    case 2: echo "Dua<br>";
            break;
    default: echo "Bukan nol, satu, atau dua<br>";
            break;
}
?>

```

Skrip 12 Penggunaan switch

3. Aplikasi PHP dan HTML dalam Form

Seperti yang telah dipaparkan sebelumnya bahwa PHP dan HTML dapat berintegrasi membentuk sebuah aplikasi web yang dinamis. Berikut adalah contoh aplikasi yang sering dijumpai dalam website.



Gambar 4 Form Buku Tamu

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=windows-1250">
    <meta name="generator" content="PSPad editor, www.pspad.com">
    <title></title>
  </head>
  <body>

    <form method="post" action="proses.php">

      <table width="200" border="1" cellspacing="0" align="center">
        <tr>
          <td align="center">BUKU TAMU</td>
        </tr>
        <tr>
          <td><table border=0>
            <tr>
              <td>Nama</td>
              <td>:</td>
              <td><input type="text" name="nama"></td>
            </tr>
            <tr>
              <td>Email</td>
              <td>:</td>
              <td><input type="text" name="email"></td>
            </tr>
            <tr>
              <td>Situs</td>
              <td>:</td>
              <td><input type="text" name="situs"></td>
            </tr>
            <tr>
              <td>Pesan</td>
              <td>:</td>
              <td><textarea width=50 height=50 name="pesan"></textarea></td>
            </tr>
            <tr>
              <td>&nbsp;</td>
              <td>&nbsp;</td>
              <td><input type="submit" name="submit1" value="OK">
                <input type="reset" name="submit2" value="Batal"></td>
            </tr>
          </table></td>
        </tr>
      </table>
    </form>

  </body>
</html>

```

Form di atas sifatnya statis, karena belum adanya pemrosesan terhadap form tersebut. Untuk itu, dibuat sebuah file PHP yang akan digunakan untuk memproses form buku tamu. Disimpan dengan nama proses.php sesuai dengan action yang didefinisikan pada skrip form.

```
<?php
$nama=$_POST[nama];
$email=$_POST[email];
$situs=$_POST[situs];
$pesan=$_POST[pesan];

echo "Nama : ".$nama."<br>";
echo "Email : ".$email."<br>";
echo "Situs : ".$situs."<br>";
echo "Pesan : ".$pesan."<br>";
?>
```

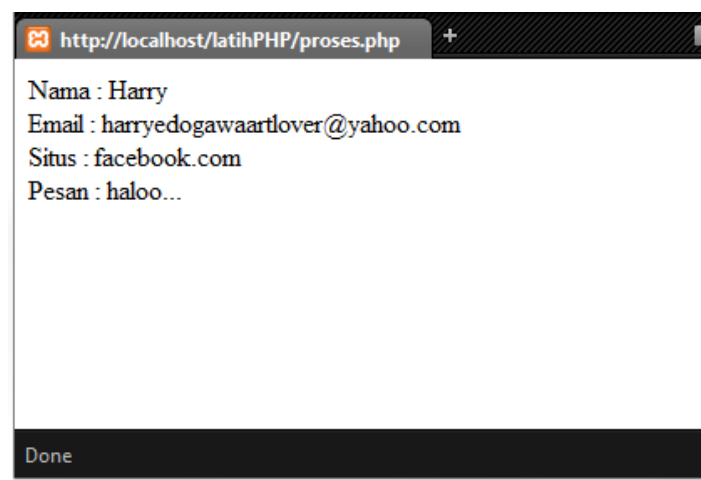
Skrip 14 File PHP pemroses

Hasilnya :



The screenshot shows a web browser window with the address bar displaying 'http://localhost/latihPHP/form.html'. The main content area contains a form titled 'BUKU TAMU'. The form has four input fields: 'Nama' with the value 'Harry', 'Email' with the value 'harryedogawaartlover@', 'Situs' with the value 'facebook.com', and 'Pesan' with the value 'haloo...'. Below the input fields are two buttons: 'OK' and 'Batal'. The browser's status bar at the bottom shows 'Done'.

Gambar 5 Form dengan isian pada form.html



The screenshot shows a web browser window with the address bar displaying 'http://localhost/latihPHP/proses.php'. The main content area displays the output of the PHP script, which is the form data: 'Nama : Harry', 'Email : harryedogawaartlover@yahoo.com', 'Situs : facebook.com', and 'Pesan : haloo...'. The browser's status bar at the bottom shows 'Done'.

Gambar 6 Hasil keluaran pada proses.php

Referensi :

1. Subagyo. 2008. *Pemrograman PHP*. Jakarta : Universitas Indraprasta PGRI.
2. Suprianto, Dodit. 2008. *Buku Pintar Pemrograman PHP*. Bandung : Oase Media.
3. Suwarno, Willy Bayuardi. 2008. *Pemrograman PHP dan MySQL*.
<http://willy.situshijau.co.id>
4. <http://www.w3schools.com>