

ACARA VII WEB (PHP)

DASAR TEORI

Bahasa pemrograman web bernama PHP merupakan singkatan rekursif dari *PHP: Hypertext Preprocessor*. Berbeda dengan HTML, dimana pemrosesan halaman dilakukan di sisi klien (browser), pada PHP pemrosesan dilakukan di sisi server sebagaimana bahasa sejenis, seperti ASP dan JSP.

Kelebihan PHP dibanding bahasa web sejenisnya diantaranya, merupakan bahasa perograman yang bebas digunakan dan didownload, mudah dipelajari, dan basis komunitasnya yang cukup besar.

File PHP dapat berisi teks, tag HTML dan skrip, nantinya isi file PHP ini akan dikirim dan diproses di server, dan hasilnya dikembalikan ke klien dalam bentuk halaman HTML.

Sebagai sebuah bahasa pemrograman, PHP dikenal akrab dengan berbagai jenis database, salah satunya adalah MySQL, yang seringkali dipaketkan menjadi satu menjadi WAMP(Windows Apache MySQL PHP), LAMP(Linux Apache MySQL PHP) atau XAMPP (X: banyak sistem operasi, Apache, MySQL PHP Perl). Oleh karenanya sebelum menggunakan PHP, terlebih dahulu harus memiliki program PHP, disarankan menggunakan paket yang sudah terintegrasi.

Membuat Program Latihan7a

Program Latihan7a berisi program PHP yang bersifat umum

Tujuan latihan ini adalah berlatih membuat web menggunakan sintak dasar pada PHP yang dipadukan dengan HTML.

LANGKAH KERJA

1. Jalankan server apache,
2. Uji apache dengan menjalankan Web Browser dan tuliskan localhost,
3. Buka program Notepad (disarankan Notepad++),
4. Tulislah sintaks-sintaks PHP,
5. Simpan file dengan ekstensi *.php (misal latihan7a.php)
6. Buka file yang telah dibuat dengan Web Browser dengan menuliskan localhost\latihan7a.php,
7. Cermati hasilnya.

SINTAKS – SINTAKS PHP

Perintah dasar

Tag pada PHP ditandai dengan <? atau <?php dan ditutup dengan >. Berikut contoh skrip sederhana

```
<html>
<body>
  <?php
    echo "Hallo semuanya!";
  ?>
</body>
</html>
```

Setiap perintah di PHP haruslah diakhiri dengan ;. Terdapat dua perintah keluaran yaitu print dan echo digunakan untuk menampilkan keluaran di layar. Sementara jika kita ingin membuat komentar maka kita gunakan // untuk komentar satu baris dan /* dan */ untuk komentar yang cukup panjang. Contoh:

```
<html>
<body>
  <?php
    //Ini adalah komentar yg akan diabaikan
    /*
      Dan ini adalah
      satu blok komentar
    */
  ?>
</body>
</html>
```

Variabel

Variabel digunakan untuk menyimpan nilai, dapat berupa karakter, string, atau angka, sehingga dapat dipakai berulang kali. Semua variabel di PHP dimulai dengan tanda \$. Berikut contoh penggunaannya:

```
<?php
$kata = "Ini Tes"
$angka = 88;
?>
```

Di PHP variabel tidak harus dideklarasikan sebelumnya, berbeda dengan C atau Pascal, dimana variabel harus dideklarasikan terlebih dahulu nama dan tipe datanya. Namun terdapat beberapa aturan perihal variabel:

- dimulai dengan tanda dollar (\$)
- sebuah variabel harus selalu diawali dengan huruf atau garis bawah (_)
- karakter yang diterima hanyalah karakter alfanumerik(a-Z, 0-9, dan _)
- sebuah variabel jika terdiri dari dua kata tidak boleh dipisah, alternatifnya bisa disambung garis bawah(\$nilai_a) atau dengan huruf kapital(\$nilaiA).
- tidak boleh menggunakan tanda baca.
- huruf besar dan huruf kecil berbeda (*case sensitif*)
- jangan menggunakan kata yang merupakan fungsi *built-in* PHP (misal: echo, include)

String

String digunakan untuk menyimpan karakter atau rangkaian karakter. Penulisan string selalu diawali dengan tanda petik ganda (") atau tanda petik tunggal ('). Sebagaimana contoh awal, kita juga bisa menyimpannya dalam sebuah variabel, contoh:

```
<?php
    $teks = "Hallo semuanya!";
    echo $teks;
?>
```

Keluaran dari kode diatas akan sama dengan keluaran dari contoh pertama.

Operator penggabungan string

String bisa digabungkan dengan menggunakan titik(.) saja, sebagai contohnya:

```
<?php
    $teks = "Hallo semuanya!";
    $teks2= "12345";
    echo $teks . " " . $teks2;
?>
```

Nantinya keluaran dari skrip diatas akan berupa:

Hallo semuanya! 12345

Mengetahui panjang string

Panjang suatu string sering digunakan dalam suatu perulangan, dimana panjang ini digunakan sebagai banyaknya perulangan yang terjadi. Di PHP kita bisa menggunakan fungsi strlen(). Contoh:

```
<?php
    echo strlen("Hallo semuanya!");
?>
```

Jika semuanya berjalan dengan baik seharusnya akan menghasilkan bilangan 15, sesuai jumlah karakter(termasuk blank space).

Mencari karakter di dalam string

Dengan fungsi strpos() kita bisa mencari satu string atau karakter di dalam string yang lain. Jika ditemukan, maka akan memberikan posisi dari kecocokan yang pertama. Namun jika tidak ada kecocokan akan menghasilkan FALSE. Masih dengan contoh diatas kita lihat penerapannya:

```
<?php
    echo strpos("Hallo semuanya!", "semua");
?>
```

Jika berjalan normal akan menghasilkan bilangan 6, bukan 7 yang mungkin sudah anda duga. Hal ini karena posisi pertama pada string adalah 0 bukan 1.

Array

Ketika bekerja dengan PHP, cepat atau lambat anda akan membutuhkan tipe data array. Tipe data array berisi beberapa elemen, yang masing-masing elemen dapat diisi data. Dan tiap elemen tersebut memiliki ID-nya sendiri sehingga dapat diakses dengan mudah. Terdapat tiga jenis array:

- Array numerik
- Array asosiatif
- Array multidimensi

Mari kita ulas array di atas satu persatu.

Array numerik

Array numerik merupakan array dengan ID numerik, seperti contoh berikut:

```
$nama = array("Joko", "Parto", "Jono");
```

secara otomatis, tiap anggota elemen diatas memiliki ID sesuai dengan urutannya.

Kita juga bisa memberi ID secara manual dengan cara:

```
$nama[0] = "Joko";
$nama[1] = "Parto";
$nama[2] = "Jono";
```

Selanjutnya isi dari array tersebut bisa kita panggil dalam sebuah skrip sebagaimana contoh berikut:

```
<?php
    $nama[0] = "Joko";
    $nama[1] = "Parto";
    $nama[2] = "Jono";
    echo $nama[1]." dan ".$nama[2]." adalah tetangga Pak ".$nama[0];
?>
```

Array asosiatif

Array asosiatif adalah array dengan ID yang diasosiasikan dengan suatu nilai. Ada kalanya untuk menyimpan data dengan nilai yang spesifik kurang pas jika menggunakan array numerik, di saat itulah kita gunakan array asosiatif. Untuk lebih jelasnya marilah kita lihat contoh berikut, dimana untuk tiap orang memiliki nilai umur yang berbeda:

```
$umur = array("Joko"=>33, "Parto"=>35, "Jono"=>29);
```

Untuk hasil yang sama dengan contoh diatas bisa juga dilakukan dengan cara:

```
$umur['Joko'] = "33";
$umur['Parto'] = "35";
$umur['Jono'] = "29";
```

Array multidimensi

Array multidimensi singkatnya merupakan array yang berisi array, untuk mengurangi kebingungan marilah kita lihat contoh berikut:

```
$keluarga = array
(
    "Joko"=>array
    (
        "Jojon",
        "Joni",
        "Joana"
    ),
    "Parto"=>array
    (
        "Parmi"
```

```
),  
"Warto"=>array  
(  
    "Warman",  
    "Warno",  
    "Warmin"  
)  
);
```

Marilah kita coba untuk menampilkan salah satu isi dari array diatas dengan menambahkan perintah berikut:

```
echo $keluarga["Joko"][2] ." adalah anggota keluarga Joko";
```

Pemilihan Kondisi

Dalam bahasa pemrograman prosedural salah satu fitur yang selalu ada adalah pemilihan kondisi, dimana kita anda aksi tersendiri untuk tiap keputusan yang dibuat atau kondisi yang ada. Jika hanya terdapat dua pilihan aksi untuk setiap kondisi atau keputusan maka kita bisa gunakan if..else. Dan jika pilihannya lebih dari dua maka kita bisa gunakan elseif. Untuk lebih jelasnya mari kita lihat penerapannya.

Menggunakan if..else

Misalkan kita ingin melakukan pengecekan hari dan kemudian memberikan respon yang tepat terhadap hari tersebut, maka kurang lebih kita bisa membuat kode sebagai berikut:

```
<html>  
<body>  
<?php  
    $d=date("D");  
    if ($d=="Sat")  
        echo "Selamat berakhir pekan!";  
    else  
        echo "Semoga hari anda menyenangkan!";  
?>  
</body>  
</html>
```

Pada kode diatas dilakukan pengecekan terhadap tanggal saat ini, kemudian akan diberikan respon yang sesuai. Jika responnya lebih dari satu perintah maka kita gunakan kurung kurawal, contoh:

```
<html>  
<body>  
<?php  
    $d=date("D");  
    if ($d=="Sat")  
    {  
        echo "Hallo!<br />";  
        echo "Selamat berakhir pekan! ";  
        echo "Sampai jumpa di hari Senin!";  
    }  
?>  
</body>  
</html>
```

Menggunakan elseif

Sebenarnya elseif merupakan perpanjangan dari if..else. Atau dengan kata lain merupakan perulangan dari pemilihan kondisi. Masih melanjutkan contoh diatas:

```
<html>
<body>
<?php
    $d=date("D");
    if ($d=="Sat")
        echo "Selamat berakhri pekan!";
    elseif ($d=="Sun")
        echo "Semoga hari Minggu anda menyenangkan!";
    else
        echo "Semoga hari anda menyenangkan!";
?>
</body>
</html>
```

Pada kode diatas respon khusus akan diberikan jika hari sama dengan Jumat atau Minggu.

Menggunakan switch

Penggunaan elseif akan menjadi tidak praktis bila terdapat banyak kondisi yang ingin kita tentukan responnya. Untuk itulah kita gunakan switch, berikut contoh implementasinya:

```
<html>
<body>
<?php
    $x = 2;
    switch ($x)
    {
        case 1: echo "Angka 1";
                    break;
        case 2: echo "Angka 2";
                    break;
        case 3: echo "Angka 3";
                    break;
        default:echo "Bukan angka antara 1 sampai 3";
    }
?>
</body>
</html>
```

Pada contoh diatas isi suatu variabel dibandingkan dengan berbagai pilihan yang tersedia, jika tidak ada yang sesuai maka default yang akan dipilih. Setelah pilihannya dijalankan maka break digunakan untuk menghentikan pemilihan kondisi dan kembali ke kode utama.

Perulangan

Seringkali dalam membuat program kita menginginkan suatu proses berjalan secara berulang-ulang, untuk itulah berbagai fungsi perulangan dibuat, agar sesuai dengan kondisi yang diinginkan.

Menggunakan while

Dengan while, perulangan akan terus terjadi selama kondisinya masih benar. Jika kondisi tidak sesuai lagi dengan syarat perulangan, maka perulangan akan dihentikan. Langsung saja kita bedah kode berikut:

```
<html>
  <body>
    <?php
      $i=1;
      while($i<=5)
      {
        echo "Angka " . $i . "<br />";
        $i++;
      }
    ?>
  </body>
</html>
```

Sebagaimana yang mungkin sudah anda duga, selama nilai \$i masih lebih kecil sama dengan 5 maka perulangan akan terus dilakukan.

Menggunakan do...while

Sekarang kita akan mencoba contoh yang sama tapi menggunakan bahasa do...while seperti berikut:

```
<html>
  <body>
    <?php
      $i=0;
      do
      {
        $i++;
        echo "Angka " . $i . "<br />";
      }
      while ($i<5);
    ?>
  </body>
</html>
```

Sedikit berbeda dengan while dimana do...while setidaknya akan menjalankan perulangan sekali dan terus mengulanginya selama kondisinya bernilai benar.

Menggunakan for

Sekarang kita akan belajar perulangan dalam bentuk yang lain, kita mulai dengan melihat kode berikut:

```
<html>
  <body>
    <?php
      for ($i=1; $i<=5; $i++)
      {
        echo "Hello World!<br />";
      }
    ?>
  </body>
</html>
```

Perulangan for memiliki tiga parameter. Parameter pertama mendeklarasikan nilai awal. Parameter kedua menentukan kondisi untuk perulangan terjadi. Dan parameter ketiga menentukan penambahan/pengurangan nilai dalam tiap perulangan.

Menggunakan foreach

Perulangan ini agak spesial karena dikhususkan untuk array, untuk lebih jelasnya kita lihat kode berikut:

```
<html>
  <body>
    <?php
      $arr=array("satu", "dua", "tiga");
      foreach ($arr as $nilai)
      {
        echo "Nilai: " . $nilai . "<br />";
      }
    ?>
  </body>
</html>
```

Function di PHP

Dalam pembuatan program seringkali kita membutuhkan sekumpulan perintah yang akan digunakan berulang kali. Tentunya sangat merepotkan apabila perintah-perintah tersebut harus diketik ulang setiap kali akan digunakan. Hal ini dapat dihindari dengan menggunakan subrutin. Subrutin adalah sekumpulan perintah yang diberi nama dan kemudian dapat kita panggil sewaktu-waktu. Dalam pemrograman dikenal dua macam subrutin, yaitu prosedur dan fungsi.

Menggunakan built-in function

Salah satu kelebihan PHP adalah banyak function yang siap untuk kita gunakan, setidaknya terdapat 700 function. strlen() dan strpos() yang sudah kita pelajari sebelumnya adalah contoh function. Pada kali ini kita akan mencoba menggunakan fungsi built-in yaitu include() yang sering digunakan untuk menyisipkan file yang disebutkan dalam argumen. Sebagai latihan buatlah dua file test.php dan vars.php yang berisi kode program sebagai berikut:

File test.php

```
<?php
  echo "Sebuah $buah $warna"; //Sebuah
  include 'vars.php';
  echo "Sebuah $buah $warna"; //Sebuah apel hijau
?>
```

File vars.php

```
<?php
  $buah = "apel";
  $warna = "hijau";
?>
```

Membuat dan menggunakan function

Sebelum berbicara lebih jauh alangkah baiknya anda mengetahui beberapa hal berikut:

- Semua function selalu dimulai dengan kata function()
- Buatlah nama yang menjelaskan kegunaan dari function tersebut

Berikut contoh penerapannya:

```
<html>
  <body>
    <?php
      function tulisNama()
      {
        echo "Merapi";
      }
      tulisNama();
    ?>
  </body>
</html>
```

Passing Variabel

Varibel dapat di-*passing* atau diteruskan ke halaman web berikutnya yang diakses user. Ada beberapa teknik untuk meneruskan variabel, diantaranya melalui:

- URL
- Form

URL

Variabel diteruskan melalui URL dengan format sebagai berikut:

```
[alamat web]?var1=nilai1&var2=nilai2
```

Sebagai contoh untuk memberikan variabel \$nama berisi “Merapi” dan \$alamat=”Sleman” ke file test.php ditulis sebagai berikut:

```
<?php
  echo "<a href=vars.php?nama=Merapi&alamat=Sleman>Gunung</a>";
?>
```

Kamudian tulislah kode untuk menerima variabel pada file vars.php sebagai berikut:

```
<?php
  $nama = $_GET['nama'];
  $alamat = $_GET['alamat'];
  echo "Nama gunung adalah Gunung $nama beralamat di $alamat";
?>
```

Form

Kelebihan lain dari PHP adalah kemudahan penanganan form dari halaman HTML untuk kemudian diproses menggunakan PHP. Berikut contoh sebuah form dalam bahasa HTML.

```
<html>
  <body>
    <form action="welcome.php" method="post">
      Nama: <input type="text" name="nama" />
      Umur: <input type="text" name="umur" />
      <input type="submit" value="Kirim"/>
    </form>
  </body>
</html>
```

Pada contoh diatas, terdapat dua buah masukan yaitu nama dan umur, satu tombol untuk mengirimkan masukan tersebut. Ketika pengguna memasukkan isian dan menekan tombol kirim maka masukan tadi akan dikirim ke halaman welcome.php yang berisi kode berikut:

```
<html>
  <body>
    Selamat datang <?php echo $_POST["nama"]; ?>.<br/>
    Umur anda <?php echo $_POST["umur"]; ?> tahun.
  </body>
</html>
```

Contoh diatas masih sangat sederhana, namun demikian dapat menjadi modal awal untuk membuat berbagai macam form sesuai keinginan anda.