

DASAR-DASAR PHP

1. Sekilas tentang PHP

PHP adalah skrip yang dijalankan di server. Keuntungan penggunaan PHP, kode yang menyusun program tidak perlu diedarkan ke user sehingga kerahasiaan kode dapat dilindungi.

Hal menarik yang didukung oleh PHP adalah kenyataan bahwa PHP bisa digunakan untuk mengakses berbagai macam database seperti Access, Oracle, MySQL, dan lain-lain.

Pengkodean PHP

latihan1.php

```
<HTML>
<HEAD>
<TITLE> Latihan pertama </TITLE>
</HEAD>
<BODY>
<H1>Contoh PHP pertama</H1>
<?
//latihan pertama
    print"Selamat belajar PHP <BR>";
    print"Semoga sukses !";
/* menampilkan pesan
    Selamat Belajar PHP
    Semoga sukses
    Di halaman browser */
?>
</BODY>
</HTML>
```

pada kode program di atas terdapat kode HTML dan kode PHP. Kode PHP diawali dengan <? dan diakhiri dengan ?>. Di dalam pasangan tersebut terdapat pernyataan-pernyataan PHP. Kode :

```
print "Selamat belajar PHP <BR>";
```

disebut sebagai sebuah pernyataan atau statement. Antarpernyataan dipisahkan dengan titik-koma (;).

Teks yang berada setelah tanda // dan di antara tanda /* dan */ disebut komentar program dan tidak akan dieksekusi. Tanda // hanya berlaku untuk satu baris, tanda /* ... */ berlaku untuk satu baris atau lebih.

2. Variabel dan Tipe Data pada PHP

Variable berfungsi untuk menyimpan suatu nilai dan nilai yang ada di dalamnya dapat diubah sewaktu-waktu. Dalam membuat suatu nama variable nama yang dipilih harus memenuhi aturan pengenalan (identifier). Identifier banyak digunakan dalam program untuk memberi nama variable, fungsi, atau kelas. Aturan yang berlaku untuk identifier adalah :

- Karakter yang digunakan adalah huruf, angka atau garis bawah.
- Karakter pertama harus berupa huruf atau garis bawah.
- Panjang pengenalan bisa berapa saja.
- Case-sensitive

Tipe data

Tipe data pada PHP ada tiga macam, yaitu : integer, double, dan string. Integer menyatakan tipe data bilangan bulat. Double menyatakan tipe data bilangan real yaitu bilangan yang mempunyai bagian pecahan. String menyatakan tipe data teks.

latihan2.php

```
<HTML>
<HEAD>
<TITLE> Latihan kedua </TITLE>
</HEAD>
<BODY>
<H1>Contoh PHP kedua</H1>
<?
    $nama="Anwar";
    $nomor=2;
    $tanggal=date("d F Y");
    $jam=date(H:i:s);
    print"Ini Halaman ke-$nomor diakses pada tanggal $tanggal<BR> ";
    print"Ini Halaman ke-$nomor diakses pada jam $jam<BR> ";
    print"Selamat belajar PHP $nama <BR> ";
    print"Semoga sukses !";
?>
</BODY>
</HTML>
```

Parsing Variabel Melalui Form

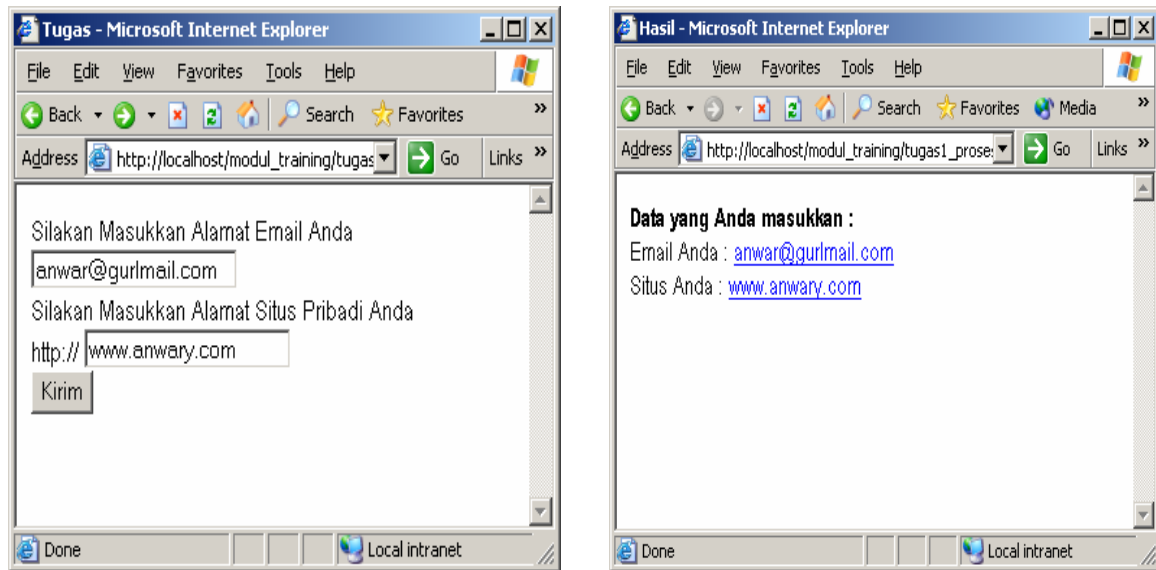
latihan3.php

```
<HTML>
<HEAD>
<TITLE> Latihan ketiga </TITLE>
</HEAD>
<BODY>
<FORM METHOD=POST ACTION="latihan3_hasil.php">
Silakan Masukkan Nama Anda <BR>
<INPUT TYPE="text" NAME="nama"><BR>
Silakan Masukkan Pesan Anda <BR>
<TEXTAREA NAME="pesan" ROWS="5" COLS="30"></TEXTAREA><BR>
<INPUT TYPE="submit" value="Kirim">
</FORM>
</BODY>
</HTML>
```

latihan3_hasil.php

```
<HTML>
<HEAD>
<TITLE> Hasil Latihan ketiga </TITLE>
</HEAD>
<BODY>
<?
print"<B>Data yang Anda masukkan : </B> <BR> ";
print"Nama Anda : $nama <BR> ";
print"Pesan Anda : $pesan";
?>
</BODY>
</HTML>
```

Latihan : buat program form pengisian alamat email dan situs pribadi sehingga ketika tombol Submit diklik menghasilkan tampilan email dan situs pribadi yang mempunyai link.



Parsing Variabel Melalui Link *latihan4.php*

```
<HTML>
<HEAD>
<TITLE> Latihan keempat </TITLE>
</HEAD>
<BODY>
Silakan Klik di
<A HREF="latihan4_proses.php?nama=Anwar&pesan=Terima+kasih">sini</A>
</BODY>
</HTML>
```

latihan4_proses.php

```
<HTML>
<HEAD>
<TITLE> Hasil Latihan keempat </TITLE>
</HEAD>
<BODY>
<?
print"Variabel dan nilai yang terkirim adalah : <BR>";
print"\$nama = $nama <BR>";
print"\$pesan = $pesan";
?>
</BODY>
</HTML>
```

3. Pernyataan Kontrol

Pernyataan IF

Pernyataan if pada PHP juga berguna untuk melakukan pengambilan keputusan terhadap lebih dari satu alternative. Bentuknya :

- If
- If else
- If elseif

Bentuk pernyataan If

Bentuk pernyataan if yang paling sederhana adalah sebagai berikut :

```
If (ekspresi)
    pernyataan
```

Pada bagian pernyataan akan dijalankan kalau bagian ekspresi bernilai benar. Sekiranya jumlah pernyataan yang akan dijalankan oleh if lebih dari satu, maka bentuknya adalah :

```
If (ekspresi){
    Pernyataan_1;
    ...
    Pernyataan_n;
}
```

Bentuk If – else

Bentuk ini melibatkan bagian else, formatnya adalah sebagai berikut :

```
If (ekspresi)
    Pernyataan_1
Else
    Pernyataan_2
```

Bagian pernyataan_1 akan dijalankan jika ekspresi bernilai benar, jika salah maka yang dijalankan adalah pernyataan_2.

Sekiranya jumlah pernyataan yang akan dijalankan oleh if atau else lebih dari satu, maka bentuknya adalah :

```
If (ekspresi){
    Pernyataan_1;
    ...
    Pernyataan_n;
}
else{
    Pernyataan_1;
    ...
    Pernyataan_n;
}
```

Bentuk ini digunakan untuk mengambil keputusan yang melibatkan banyak alternatif.

```

If (ekspresi1){
    Pernyataan_1;
    ...
    Pernyataan_n;
}
else if (ekspresi2){
    Pernyataan_1;
    ...
    Pernyataan_n;
}
...
else{
    Pernyataan_1;
    ...
    Pernyataan_n;
}

```

Latihan5.php

```

<HTML>
<HEAD>
<TITLE> Latihan kelima </TITLE>
</HEAD>
<BODY>
<?
$nama_hari=date("l");
if ($nama_hari=="Sunday"){
    print"Hari ini hari Ahad";
}
else{
    print"Hari ini bukan hari Ahad";
}
?>
</BODY>
</HTML>

```

Latihan : Buat program konversi hari English ke hari Indonesia

Pernyataan Switch

Pernyataan ini berfungsi untuk menyeleksi sejumlah alternative. Bentuk umumnya sebagai berikut :

```

switch (ekspresi){
    case ekspresi_case_1 :
        pernyataan_1;
        break;
    case ekspresi_case_2 :
        pernyataan_2;
        break;
    ...
    default :
        pernyataan_n;
}

```

Kegunaan break untuk mengarahkan eksekusi ke akhir switch. Setiap case bisa mengandung lebih dari satu pernyataan, antarpernyataan dipisahkan dengan titik-koma. Bagian default akan dieksekusi kalau tak ada yang cocok pada bagian case.

Latihan6.php

```
<HTML>
<HEAD>
<TITLE> Latihan keenam </TITLE>
</HEAD>
<BODY>
<?
$nama_hari=date("l");
switch ($nama_hari){
case "Sunday" :
    print"Hari ini hari Ahad";
break;
default :
    print"Hari ini bukan hari Ahad";
}
?>
</BODY>
</HTML>
```

Latihan : Buat program konversi hari English ke hari Indonesia menggunakan switch

4. Pernyataan Perulangan

Pernyataan For

Sebagaimana while, pernyataan for berguna juga untuk perulangan. Bentuknya :

```
for (ekspresi1; ekspresi2; ekspresi3){
    pernyataan_pernyataan;
}
```

ekspresi1 untuk memberi nilai awal terhadap variable yang akan digunakan untuk melakukan pencacahan pengulangan.

ekspresi2 untuk menentukan pengulangan terhadap pernyataan yang ada dalam tanda 2 kurung kurawal ({ ... }) akan dilakukan atau tidak.

ekspresi3 untuk mengatur nilai variable yang digunakan dalam ekspresi1.

Latihan7.php

```
<HTML>
<HEAD>
<TITLE> Latihan ketujuh </TITLE>
</HEAD>
<BODY>
<?
for ($i=1; $i<=7; $i++){
    print"<FONT SIZE=\"\$i\" COLOR=\"silver\" FACE=\"Verdana\">$i</FONT>
<BR>";
}
?>
</BODY>
</HTML>
```

Latihan : buat program dengan tampilan sebagai berikut :

```
*
**
***
****
```

```
          *
        ***
      *****
    *********
```

Pernyataan While

Pernyataan while berguna untuk melakukan suatu perulangan. Bentuknya :

```
while (ekspresi){  
    pernyataan_pernyataan;  
}
```

Latihan8.php

```
<HTML>  
<HEAD>  
<TITLE> Latihan kedelapan </TITLE>  
</HEAD>  
<BODY>  
<?  
$i=1;  
while ($i<=7){  
    print"<FONT    SIZE=\"$i\"    COLOR=\"silver\"    FACE=\"Verdana\">$i</FONT>  
<BR>";  
    $i++;  
}  
?>  
</BODY>  
</HTML>
```

Latihan : buat program dengan tampilan seperti latihan sebelumnya namun menggunakan while

4. Operator

Operator adalah symbol yang digunakan dalam program untuk melakukan suatu operasi, misalnya penjumlahan atau perkalian, perbandingan kesamaan dua buah nilai, atau bahkan memberikan nilai ke variable.

Operator Aritmatika

- +=** menambahkan variable di sisi kiri dengan nilai di sisi kanan
- =** mengurangi isi variable di sisi kiri dengan nilai di sisi kanan
- /=** membagi isi variable di sisi kiri dengan nilai di sisi kanan
- %=** memperoleh sisa pembagian antara variable di sisi kiri dengan nilai di sisi kanan

Operator Perbandingan

- ==** sama dengan
- <** kurang dari
- >** lebih dari
- <=** kurang dari atau sama dengan
- >=** lebih dari atau sama dengan
- !=** tidak sama dengan (bisa juga menggunakan <>)

latihan9.php

```
<?
switch ($operator){
case "+": $hasil=$angka1+$angka2;
        break;
case "-": $hasil=$angka1-$angka2;
        break;
case "*": $hasil=$angka1*$angka2;
        break;
case "/": if ($angka2==0) $hasil="E"; else $hasil=$angka1/$angka2;
        break;
}
?>
<HTML>
<HEAD>
<TITLE> Latihan kesembilan </TITLE>
</HEAD>
<BODY>
<FORM METHOD=POST ACTION="latihan9.php">
<TABLE style="font-family: verdana;font-size: 10pt" align=center border=5
cellspacing=0>
<TR bgcolor=f1f1f1>
<TD colspan=4>Kalkulator Sederhana</TD>
</TR>
<TR>
<TD colspan=4><INPUT TYPE="text" NAME="angka1" value="<? print $angka1
?>"></TD>
</TR>
<TR>
<TD colspan=4><INPUT TYPE="text" NAME="angka2" value="<? print $angka2
?>"></TD>
</TR>
<TR align=center>
<TD><INPUT TYPE="submit" name="operator" value="+"></TD>
<TD><INPUT TYPE="submit" name="operator" value="-"></TD>
<TD><INPUT TYPE="submit" name="operator" value="*"></TD>
<TD><INPUT TYPE="submit" name="operator" value="/"></TD>
</TR>
<TR>
<TD colspan=4><INPUT TYPE="text" NAME="hasil" value="<? print
$hasil?>"></TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

5. Array

Pengertian Array

Array adalah wadah yang dapat menyimpan sejumlah nilai scalar. Masing-masing nilai diakses melalui indeks yang biasa disebut subskrip. Dalam hal ini indeks berupa bilangan bulat yang dimulai dari nol.

Menciptakan Array

Array dalam PHP diciptakan dengan bentuk sebagai berikut :

```
$namaArray=array(elemen_1, ... , elemen_N);
```


Mengubah Variabel Biasa Menjadi Array

Sebuah variable biasa bisa diubah keadaannya menjadi sebuah array, contoh : variable \$kalimat="Saya belajar PHP"; bisa diubah menjadi array \$kata dengan masing-masing elemen \$kata[0]="Saya"; \$kata[1]="belajar"; \$kata[2]="PHP"; Fungsi untuk melakukan hal tersebut adalah menggunakan **explode**. Sintaknya adalah :

explode("pemisah", "string yang akan dijadikan array");

oleh sebab itu pernyataan di atas dapat dituliskan dalam program berikut ini :

```
$kalimat="Saya belajar PHP";  
$kata=explode(" ", $kalimat);
```

Kebalikan explode adalah **implode**. Sintaknya adalah :

implode("perekat", "array yang akan digabung")

jika variable array di atas akan digabung kembali dengan penggabungannya tanda strip tengah (-) misalnya maka hal tersebut dapat dituliskan dalam program berikut ini :

```
$gabung=implode("-", $kata);
```

Mengetahui Jumlah Elemen Array

Untuk mengetahui jumlah elemen array digunakan fungsi **count()** dengan argument berupa nama array.

latihan10.php

```
<?  
$kalimat="Saya belajar PHP";  
$kata=explode(" ", $kalimat);  
$jumlahArray=count($kata);  
for ($i=0; $i<$jumlahArray; $i++){  
    print "\$kata[$i]=$kata[$i]<BR>";  
}  
$gabung=implode("-", $kata);  
print"<BR>";  
print "\$gabung=$gabung";  
?>
```

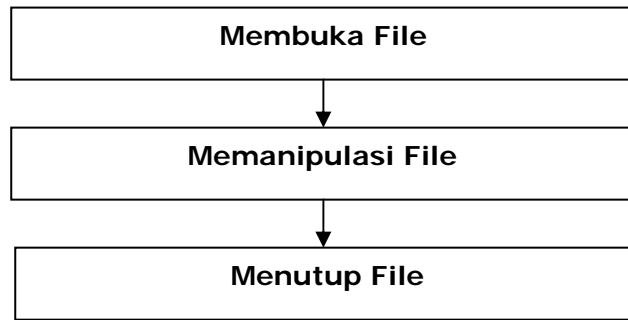
6. File

Bagian ini membahas cara menyimpan data ke dalam file dan sekaligus cara membaca data yang tersimpan dalam file.

Prinsip Pemrosesan pada File

Pemrosesan pada file pada dasarnya hanya ada tiga proses, yaitu : proses membuka file, proses memanipulasi file (menyimpan data ke file atau membaca data yang ada dalam file), dan proses menutup file.

Hal tersebut dapat digambarkan pada skema di bawah ini :



Membuka File

Perintah untuk membuka file menggunakan **fopen**, sintaknya adalah :

```
fopen("nama_file", "mode buka")
```

Mode buka file pada ada tiga macam, yaitu r (untuk membaca/read file), w (untuk menyimpan/write data ke file), dan a (untuk menambah/append data ke file).

Biasanya perintah fopen disimpan dalam sebuah variable karena hasil perintah tersebut akan digunakan untuk parameter perintah-perintah akses file berikutnya. Untuk mode buka r file sudah harus ada, sedangkan untuk mode yang lain file boleh ada atau boleh juga tidak.

Memanipulasi File

Memanipulasi file ada dua menyimpan data ke file dan membaca data dari file. Untuk menyimpan data ke file digunakan perintah :

```
fputs(variable_penyimpan_buka_file, "data yang akan disimpan ke file");
```

Untuk membaca data ke file digunakan perintah :

```
fread(variable_penyimpan_buka_file, panjang karakter yang akan dibaca);
```

```
fgets(variable_penyimpan_buka_file, panjang karakter yang akan dibaca);
```

perbedaan **fread** dan **fgets** adalah fgets hanya mampu membaca perbaris sedangkan fread pembacaan yang dilakukan adalah dari data awal pada file sampai akhirnya. Seharusnya fungsi **fread** maupun **fgets** disimpan dalam sebuah variable, dikarenakan nantinya apa yang sudah dibaca selanjutnya dicetak ke browser.

Manipulasi file harus selaras dengan mode bukannya. Jika manipulasi filenya adalah membaca data maka digunakan mode buka r. jika manipulasi filenya adalah menyimpan data maka digunakan mode buka w atau a. Perbedaan antara mode buka w dan a adalah w menyebabkan overwriting sedangkan a menyebabkan data tambahan akan terletak pada akhir pointer.

Menutup File

Jika akses file sudah selesai maka seharusnya file ditutup. Perintah untuk menutup file adalah **fclose**, sintaknya adalah :

```
fclose(variable_penyimpan_buka_file);
```

latihan11.php

```
<?
    if ($submit=="Simpan ke File"){
        $data=fopen("data.txt","a");
        fputs($data,"Nama : $nama \n ");
        fputs($data,"Email : $email \n ");
        fputs($data,"-----\n");
        fclose($data);
    }
?>
<HTML>
<HEAD>
<TITLE> Latihan kesebelas </TITLE>
</HEAD>
<BODY>
<CENTER><H1>Menyimpan Data ke File</H1></CENTER>
<FORM METHOD=POST ACTION="latihan11.php">
<TABLE align=center style="font-family: verdana;font-size: 10pt">
<TR>
    <TD>Nama</TD>
    <TD>: <INPUT TYPE="text" NAME="nama" size=40></TD>
</TR>
<TR>
    <TD>Email</TD>
    <TD>: <INPUT TYPE="text" NAME="email" size=50></TD>
</TR>
<TR>
    <TD colspan=2><INPUT TYPE="submit" name="submit" value="Simpan
ke File"></TD>
</TR>
</TABLE>
</FORM><BR><BR>
<CENTER><H1>Hasil Simpanan</H1></CENTER>
<?
    $data=fopen("data.txt","r");
    $baca=fread($data,1024);
    $baca=str_replace("\n","<BR>",$baca);
    print $baca;
?>
</BODY>
</HTML>
```

Tugas :

1. buat tampilan pada latihan di atas menggunakan **fgets**, petunjuk gunakan while dan feof(variable_penyimpan_buka_file). Fungsi feof akan mengembalikan nilai TRUE jika pointer file terletak di akhir file.
2. buat aplikasi teks updater untuk copyright sebuah web.
3. buat aplikasi buku tamu menggunakan file. Petunjuk : gunakan akses file, explode dan for.
4. buat aplikasi counter. Petunjuk : data yang ada pada file dibaca kemudian ditambahkan 1 baru ditulis/simpan ke file sembari ditampilkan. Buat file teksnya yang berisi angka 0 (nol) simpan dengan nama counter.txt.
5. setelah nomor 3 selesai cobalah agar tampilan counter berupa image. Petunjuk : setiap angka yang dihasilkan diganti dengan gambar yang sudah ada,

7. Fungsi

Istilah fungsi menyatakan blok kode yang diberi nama. Selain menyediakan sejumlah fungsi bawaan, PHP juga memperkenalkan pemrogram mendefinisikan sendiri suatu fungsi. Bentuk pendefinisian sebagai berikut :

```
function nama (daftar_parameter){  
    pernyataan_1;  
    ...  
    pernyataan_n ;  
}
```

pernyataan return digunakan untuk memberikan nilai balik fungsi. Perlu diketahui bahwa pernyataan return akan membuat fungsi berakhir dan menghasilkan nilai yang sesuai dengan isi variable.

latihan12.php

```
<?  
function tampilAlert($pesan){  
    print"<script>alert('Pesan : $pesan');window.history.go(-1);</script>";  
}  
function kuadrat($bilangan){  
    $kuadrat=$bilangan*$bilangan;  
    $hasil="<script> alert('hasil $bilangan kuadrat = $kuadrat'); window.history.go(-1); </script>";  
    return $hasil;  
}  
switch($submit){  
    case "Tampilkan" : tampilAlert($pesan);  
                      break;  
    case "Kuadrat"   : print (kuadrat($bilangan));  
                      break;  
}  
?>  
<HTML>  
<HEAD>  
<TITLE> Latihan kedua belas </TITLE>  
</HEAD>  
<BODY style="font-family: Verdana; font-size: 10pt; font-weight: bold">  
<FORM METHOD=POST ACTION="latihan12.php">  
Tuliskan Pesan Anda <INPUT TYPE="text" NAME="pesan"> <INPUT TYPE="submit"  
name="submit" value="Tampilkan"><BR><HR>  
Tuliskan Bilangan <INPUT TYPE="text" NAME="bilangan" size=3> <INPUT  
TYPE="submit" name="submit" value="Kuadrat">  
</FORM>  
</BODY>  
</HTML>
```