

MATERI I - STRUKTUR DASAR PHP

Materi

- ❖ Sebelum Belajar PHP
- ❖ Mari Mengenal PHP
- ❖ Say Hello to PHP
- ❖ Variabel
- ❖ Tipe Data
- ❖ Konstanta
- ❖ Operator dalam PHP
- ❖ Komentar Program

Sebelum Belajar PHP

1. Saya asumsikan Anda telah mengenal dengan cukup baik komputer Anda, dan juga bisa mengoperasikannya. Jadi Anda seharusnya sudah mengerti perbedaan file dan direktori, bagaimana menyimpan file atau document, bagaimana meng-copy file, dan seterusnya.
2. Saya asumsikan Anda telah mengerti bagaimana membuka dan menggunakan salah satu atau salah semua dari editor seperti Notepad, EditPlus, Crimson Editor, atau yang lainnya. Saya asumsikan juga, Anda telah mengerti bagaimana membuat, mengedit, dan menyimpan suatu file dengan editor tersebut.
3. Saya asumsikan bahwa Apache, PHP, dan MySQL di komputer yang Anda pakai sudah berjalan sebagaimana mestinya.
4. Saya asumsikan Anda telah mengerti bagaimana cara menggunakan Web Browser (IE, Chrome, Mozilla, Opera, dsb)
5. Saya asumsikan Anda telah mengenal dan mengerti dengan cukup baik tentang tag-tag HTML dan juga bagaimana menggunakannya.
6. Saya asumsikan bahwa penglihatan Anda masih normal sehingga bisa membedakan antara :
 - Tulisan nama, Nama dan NAMA
 - Koma (,), titik-koma (;) dan titik-dua (:)
 - Kutip-tunggal (') dan kutip-ganda (")
 - Tulisan namamhs, nama_mhs dan nama mhs
7. Saya asumsikan Anda pernah menggunakan www.google.com.

Mari Mengenal PHP

Tahukah Anda, PHP itu.....

1. Merupakan singkatan recursive dari PHP : Hypertext Preprocessor
2. Pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994.
3. Harus ditulis di antara tag :
 - <? dan ?>
 - <?php dan ?>
 - <script language="php"> dan </script>
 - <% dan %>
4. Setiap satu statement (perintah) biasanya diakhiri dengan titik-koma (;)
5. CASE SENSITIVE untuk nama identifier yang dibuat oleh user (berupa variable, konstanta, fungsi dll), namun TIDAK CASE SENSITIVE untuk identifier built-in dari PHP. Jadi :

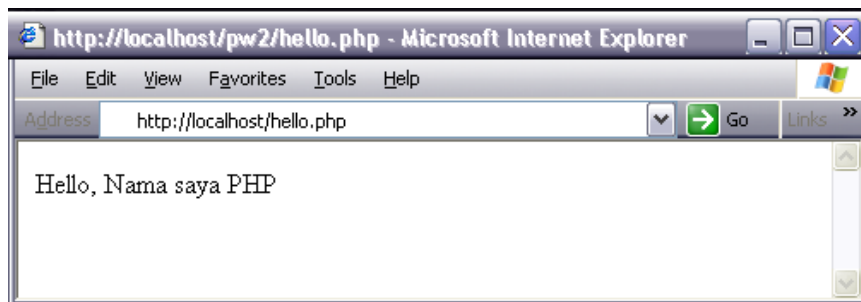
- \$nama ≠ \$Nama ≠ \$NAMA
 - hitungLuas() ≠ HitungLuas()
 - echo = ECHO
 - while = WHILE
6. Mudah dipelajari.

Say Hello to PHP

1. Ketikkan perintah di bawah ini di editor :

```
1 <?php
2 echo "Hello, nama saya PHP";
3 ?>
```

2. Buka browser dan ketikkan alamat <http://localhost/hello.php>
3. Lihat hasilnya di browser. Akan terlihat tampilan seperti pada gambar dibawah ini.



Variabel

- Digunakan untuk menyimpan sebuah value, data atau informasi
- Nama variabel diawali dengan tanda \$
- Panjang tidak terbatas
- Setelah tanda \$ diawali oleh huruf atau under-score (_). Karakter berikutnya bisa terdiri dari huruf, angka, dan karakter tertentu yang diperbolehkan (karakter ASCII dari 127 – 255).
- Bersifat case-sensitive.
- Tidak perlu dideklarasikan.
- Tidak boleh mengandung spasi.

Contoh :

Benar

- \$_name
- \$first_name
- \$name3
- \$name_3

Salah

- \$3name
- \$name?
- \$first+name
- \$first.name
- \$first name

Komentar Program

Dalam PHP, komentar program bisa menggunakan :

- /* dan */
- // dan
- #

Tipe Data

Pada PHP, tipe data variabel tidak didefinisikan oleh programmer, akan tetapi secara otomatis ditentukan oleh interpreter PHP. Namun demikian, PHP mendukung 8 (delapan) buah tipe data primitif, yaitu :

1. boolean
2. integer
3. float
4. string
5. array
6. object
7. resource
8. NULL

Konstanta

Konstanta merupakan variabel konstan yang nilainya tidak berubah-ubah. Untuk mendefinisikan konstanta dalam PHP, menggunakan fungsi define().

Operator dalam PHP

Jenis Operator	Operator	Contoh	Keterangan
Aritmatika	+	\$a + \$b	Pertambahan
	-	\$a - \$b	Pengurangan
Penugasan	=	\$a = \$b	Pada saat ini
	+=	\$a += \$b	Pertambahan dengan 4
Bitwise	%	\$a % \$b	Pembagian
	&	\$a & \$b	Bitwise AND
		\$a \$b	Bitwise OR
	^	\$a ^ \$b	Bitwise XOR
	~	~\$b	Bitwise NOT
	<<	\$a << \$b	Shift Left
	>>	\$a >> \$b	Shift Right
Perbandingan	==	\$a == \$b	Sama dengan
	===	\$a === \$b	Identik
	!=	\$a != \$b	Tidak sama dengan
	<>	\$a <> \$b	Tidak sama dengan
	!==	\$a !== \$b	Tidak identik
	<	\$a < \$b	Kurang dari
	>	\$a > \$b	Lebih dari
	<=	\$a <= \$b	Kurang dari sama dengan
	>=	\$a >= \$b	Lebih dari sama dengan
Logika	and	\$a and \$b	TRUE jika \$a dan \$b TRUE
	&&	\$a && \$b	TRUE jika \$a dan \$b TRUE
	or	\$a or \$b	TRUE jika \$a atau \$b TRUE
		\$a \$b	TRUE jika \$a dan/atau \$b TRUE
	xor	\$a xor \$b	TRUE jika \$a atau \$b TRUE, tapi tidak keduanya
	!	!\$a	TRUE jika \$a FALSE
String	.	\$a . \$b	Penggabungan string \$a dan \$b

MATERI II - PENANGANAN FORM

Materi

- ❖ Berbagai Cara Penanganan Form
- ❖ Form Input Type TEXT dan PASSWORD
- ❖ Form Input Type RADIO
- ❖ Form Input Type CHECKBOX
- ❖ Form Input Type COMBO BOX
- ❖ Form Input Type TEXTAREA

Form inputan dibuat dengan tag-tag HTML. Untuk merancang sebuah form inputan, setidaknya ada 3 (tiga) hal penting, yaitu :

1. Method

Method dari sebuah form menentukan bagaimana data inputan form dikirim. Method ini ada dua macam, yaitu GET dan POST. Method ini menentukan bagaimana data inputan dikirim dan diproses oleh PHP.

2. Action

Action dari sebuah form menentukan dimana data inputan dari form diproses. Jika action ini dikosongkan, maka dianggap proses form terjadi di halaman yang sama. Jadi halaman form dan halaman proses bisa saja dipisah atau dijadikan satu.

3. Submit Button

Submit button merupakan sebuah tombol (pada umumnya) yang berfungsi sebagai trigger pengiriman data dari form inputan. Jika tombol ini ditekan, maka data form akan dikirimkan (diproses) di halaman yang sudah ditentukan pada atribut action.

Cara Penanganan Form

Cara 1 : Menyatukan antara Form dan Proses

Proses pengolahan form dilakukan di halaman yang sama dengan form inputannya. Jika proses penanganan form berada di satu halaman, maka value atribut action pada tag form tidak perlu diisi (dikosongkan).

```

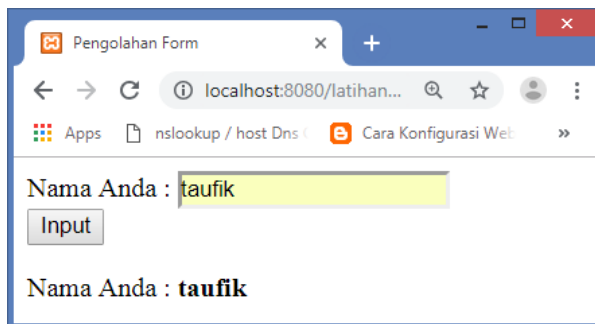
1 <html>
2 <head>
3   <title>Pengolahan Form</title>
4 </head>
5 <body>
6   <form action="" method="post" name="input">
7     Nama Anda : <input type="text" name="nama"/><br/>
8     <input type="submit" name="Input" value="Input"/>
9   </form>
10 </body>
11 </html>
12
13 <?php
14 if(isset($_POST['Input'])) {
15     $nama=$_POST['nama'];
16     echo "Nama Anda : <b>$nama</b>";
17 }
18 ?>

```

Penjelasan Program

Beberapa hal yang perlu diperhatikan dari program di atas, di antaranya mengenai nama setiap komponen form karena nama ini akan menjadi index array dalam PHP. Pada program di atas, value atribut action pada tag form tidak diisi (baris 6), ini berarti bahwa proses pengolahan form berada di halaman yang sama. Selanjutnya (baris 6) method yang digunakan dalam penanganan form adalah POST. Cara ini lebih disarankan dalam penanganan sebuah form inputan.

Mulai baris ke-13 hingga 18, terdapat script / program PHP yang akan menangani (mengolah) nilai yang diinputkan melalui form. Letak proses inputan ini boleh sebelum atau sesudah form, tergantung kebutuhan. Pada baris ke-14, terdapat pemeriksaan kondisi apakah tombol dengan nama "Input" (perhatikan index array \$_POST dan bandingkan dengan name tombol submit pada form) benar-benar ditekan atau tidak oleh user. Selanjutnya pada baris ke-15, nilai inputan dari form akan diambil dengan cara mengakses array \$_POST sesuai dengan komponen form yang akan diambil. Pada baris 15 ini, isi komponen inputan dengan nama "nama" akan diambil dan dimasukkan ke variabel \$nama. Index array \$_POST pada baris 15 harus sama dengan value atribut name pada baris ke-7.



Cara 2 : Memisahkan antara Form dan Proses

Proses pengolahan form dilakukan di halaman yang terpisah dengan form inputannya. Jika proses penanganan form berada dilakukan di halaman yang berbeda, maka value atribut action pada tag form harus diisi dengan alamat halaman tempat proses pengolahan form.

Nama File : input02.php

```

5 <body>
6     <form action="proses.php" method="post" name="input">
7         Nama Anda : <input type="text" name="nama"/><br/>
8         <input type="submit" name="Input" value="Input"/>
9     </form>
10 </body>
11 </html>

```

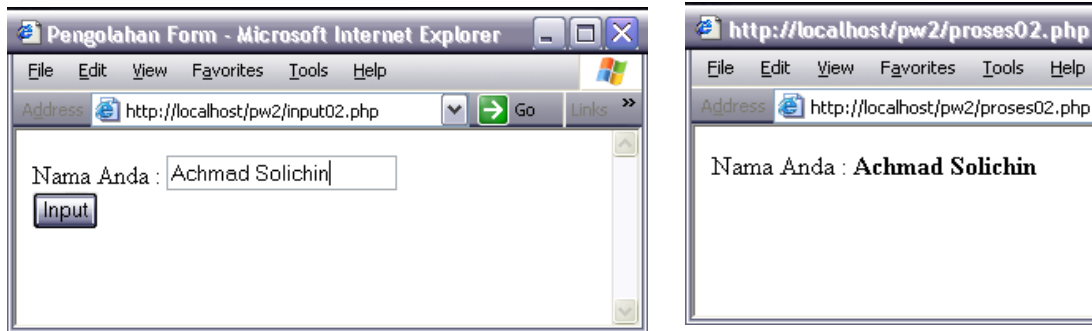
Nama File : proses02.php

```

1 <?php
2 if(isset($_POST['Input'])) {
3     $nama=$_POST['nama'];
4     echo "Nama Anda : <b>$nama</b>";
5 }
6 ?>

```

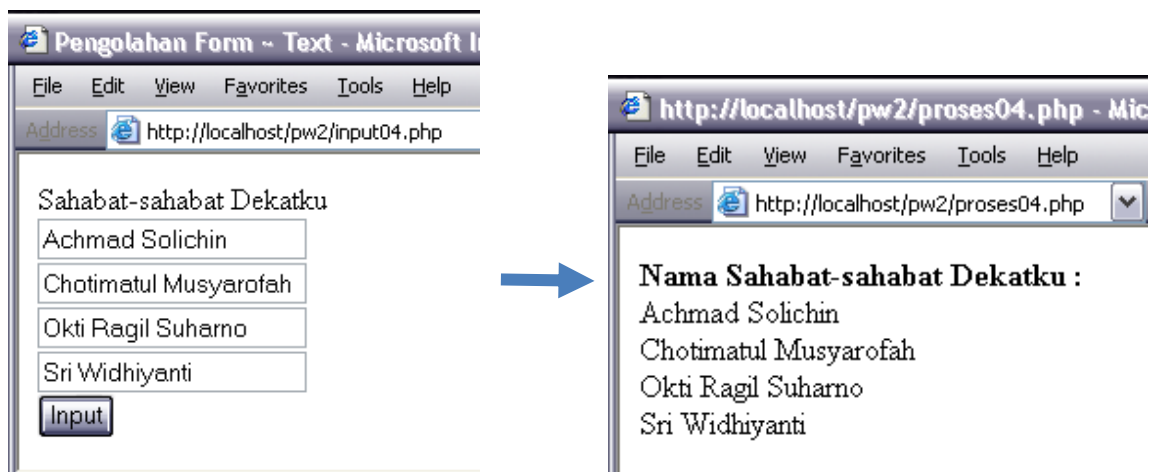
Tampilan Program akan sebagai berikut.



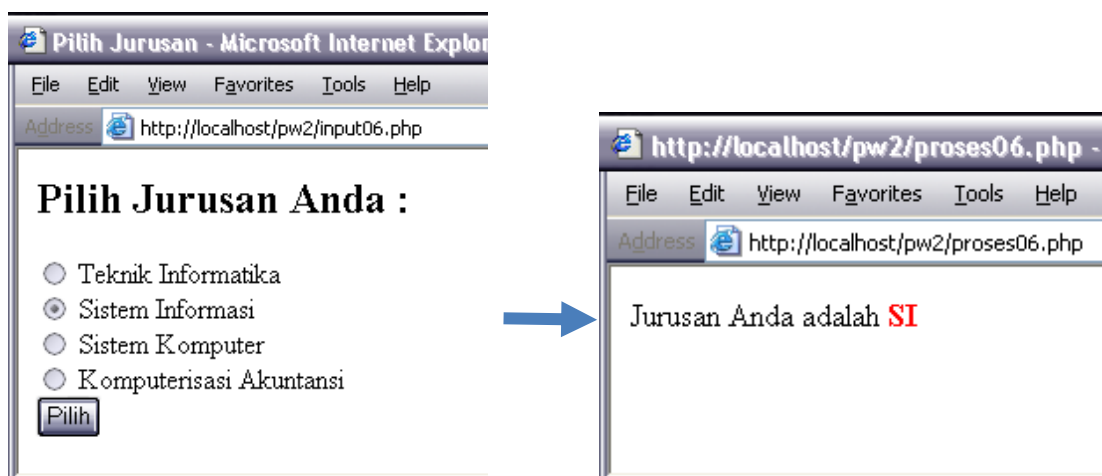
TUGAS LATIHAN PHP

Buatlah program seperti dibawah ini!

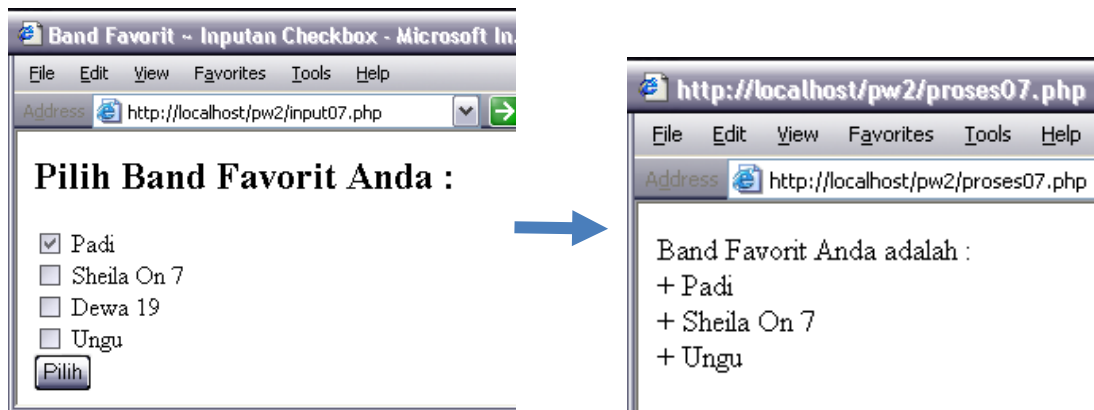
1. Form Input Type TEXT



2. Form Input Type RADIO

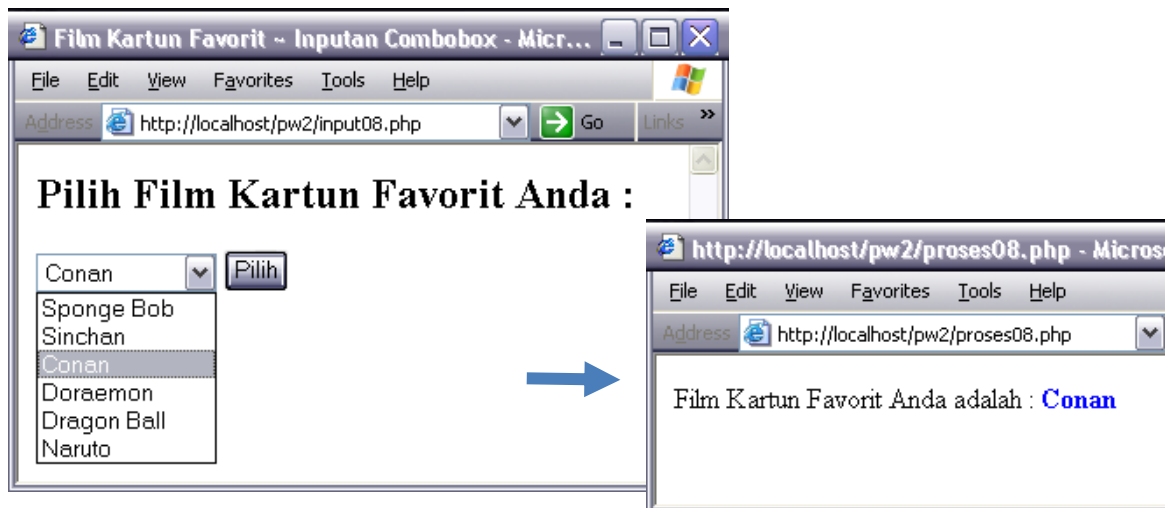


3. Form Input Type CHECKBOX

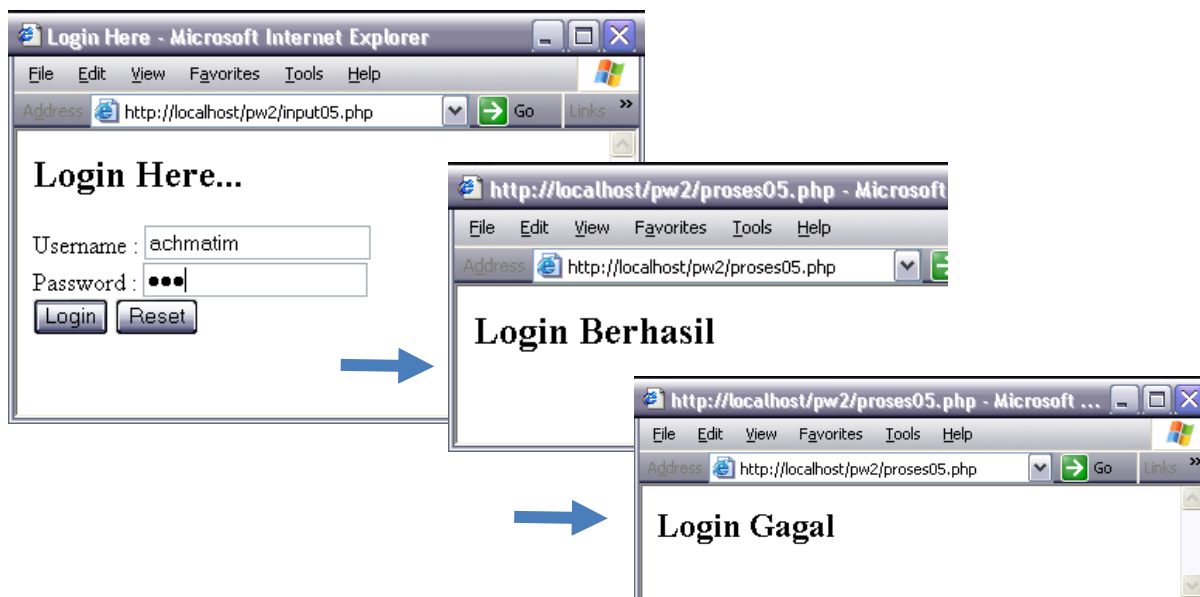


4. Form Input Type SELECT

Buatlah program seperti dibawah ini.



5. Form Input Type PASSWORD



TUGAS PRAKTIKUM PHP-1

Dengan melibatkan elemen form (HTML), buatlah program menggunakan kode PHP diantaranya dibawah ini:

1. Program (IF – ELSEIF) Menggunakan Inputan dari FORM
2. Program (SWITCH - CASE) Menggunakan Inputan dari FORM
3. Program (FOR) Menggunakan Inputan dari FORM

Array dan Fungsi

- ❖ Pengantar Array
- ❖ Mendeklarasikan dan Menampilkan Array
- ❖ Fungsi-fungsi Array dalam PHP
- ❖ Fungsi dalam PHP

Pengantar Array

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Bagian yang menyusun array disebut elemen array, yang masing-masing elemen dapat diakses tersendiri melalui index array. Index array dapat berupa bilangan integer atau string.

Mendeklarasikan dan Menampilkan Array

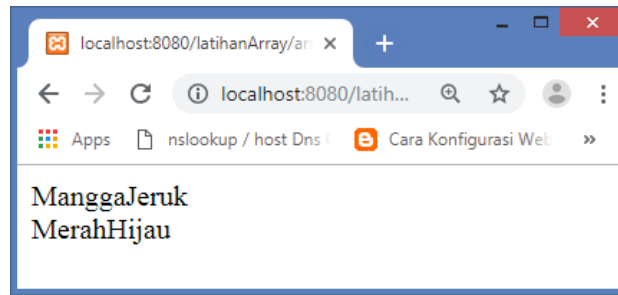
Program 1

Deskripsi : Program mendeklarasikan array dengan index numeric.

```
1 <?php
2 $arrBuah=array("Mangga", "Apel", "Pisang", "Jeruk");
3 echo $arrBuah[0]; //Mangga
4 echo $arrBuah[3]; //Jeruk
5 echo "</br>";
6
7 $arrWarna=array();
8 $arrWarna[]="Merah";
9 $arrWarna[]="Biru";
10 $arrWarna[]="Hijau";
11 $arrWarna[]="Putih";
12 echo $arrWarna[0]; //Merah
13 echo $arrWarna[2]; //Hijau
14 ?>
```

Penjelasan Program 1

Untuk mendeklarasikan atau mendefinisikan sebuah array di PHP bisa menggunakan keyword **array()**. Pada program 1, terdapat pendeklarasian dan pendefinisian array pada baris ke-2. Jumlah elemen array tidak perlu disebutkan saat deklarasi. Sedangkan untuk menampilkan isi array pada elemen tertentu, cukup dengan menyebutkan nama array beserta index array-nya (lihat baris ke-3 dan ke-4). Pendeklarasian dan pengisian array juga bisa dilakukan seperti pada baris ke-7 sampai baris ke-11. Dengan pendefinisian array seperti pada dua cara di atas, index (key) array secara otomatis akan diisi dengan bilangan bulat integer yang dimulai dari 0.



Gambar. Tampilan Program 1

Program 2

Deskripsi : Program mendeklarasikan array dengan index string (array assosiatif).

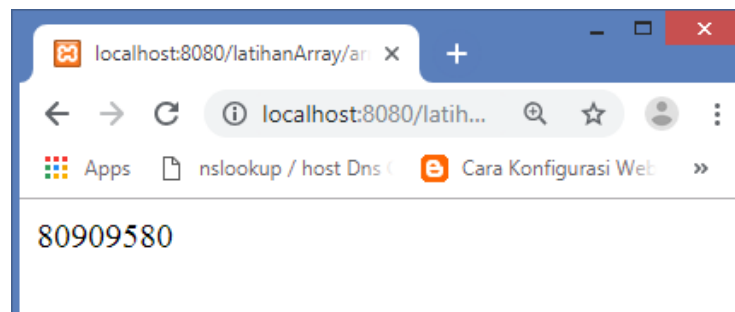
```

1 <?php
2 $arrNilai=array("Ani"=>80, "Otim"=>90, "Ana"=>75, "Budi"=>85);
3 echo $arrNilai['Ani']; //80
4 echo $arrNilai['Otim']; //90
5
6 $arrNilai=array();
7 $arrNilai['Ami']=80;
8 $arrNilai['Asma']=95;
9 $arrNilai['Sri']=77;
10 echo $arrNilai['Asma']; //95
11 echo $arrNilai['Ami']; //80
12 ?>

```

Penjelasan Program 2

Untuk mendeklarasikan atau mendefinisikan sebuah array *assosiatif* (array yang menggunakan index selain integer), dapat dengan cara menyebutkan index-nya terlebih dahulu diikuti operator `=>` dan diikuti value atau nilai elemennya. Lihat program baris ke-2! Sedangkan untuk menampilkan value atau nilai dari suatu array, bisa dengan menyebutkan nama array yang diikuti index arraynya (lihat baris ke-3 dan ke-4). Pendefinisian array *assosiatif* juga bisa dilakukan seperti pada baris ke-6 sampai baris ke-9.



Gambar. Tampilan Program 2

Program 3

Deskripsi : Program menampilkan seluruh isi array dengan index numerik.

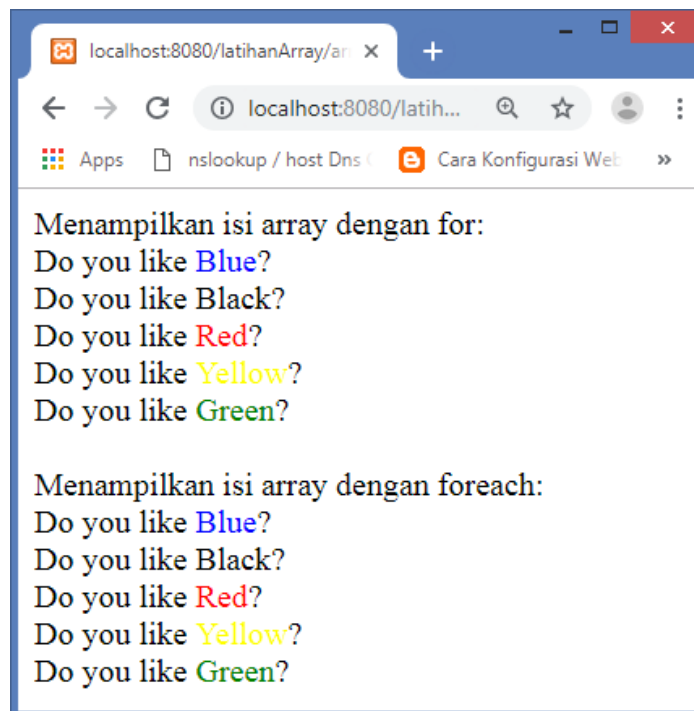
```

1 <?php
2 $arrWarna=array("Blue","Black","Red","Yellow","Green");
3
4 echo "Menampilkan isi array dengan for: <br>";
5 for($i=0; $i<count($arrWarna); $i++){
6     echo "Do you like <font color=$arrWarna[$i]>". $arrWarna[$i]. "</font>?</br>";
7 }
8
9 echo "<br>Menampilkan isi array dengan foreach:<br>";
10 foreach($arrWarna as $warna){
11     echo "Do you like <font color=$warna>". $warna. "</font>?</br>";
12 }
13 ?>

```

Penjelasan Program 3

Untuk menampilkan keseluruhan isi array bisa dilakukan dengan perulangan for, while, do..while atau foreach. Baris 5 – 7 program di atas merupakan contoh menampilkan array dengan perulangan for. Fungsi **count()** pada baris ke-5 berfungsi untuk mencari jumlah elemen dari array \$arrWarna. Untuk menampilkan array juga bisa menggunakan struktur **foreach()**. Struktur foreach akan mengambil elemen array mulai dari posisi pointer sampai akhir pointer. Pada contoh program di atas (baris ke-10) bisa dinyatakan dengan pernyataan bahwa "untuk setiap elemen array \$arrWarna, masukkan ke variabel \$warna".



Gambar. Tampilan Program 3

Program 4

Deskripsi : Program menampilkan seluruh isi array asosiatif.

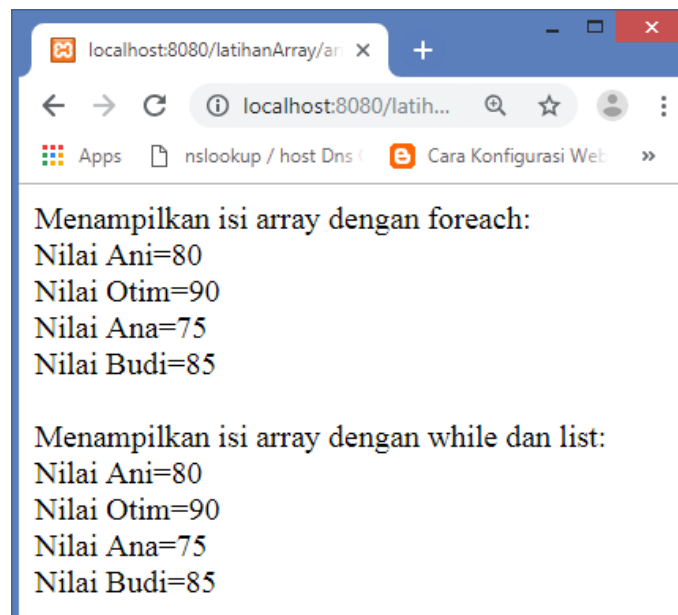
```

1 <?php
2 $arrNilai=array("Ani"=>80, "Otim"=>90, "Ana"=>75, "Budi"=>85);
3
4 echo "Menampilkan isi array dengan foreach:<br>";
5
6 foreach($arrNilai as $nama=>$nilai){
7     echo "Nilai $nama=$nilai<br>";
8 }
9 reset ($arrNilai);
10 echo "<br>Menampilkan isi array dengan while dan list:<br>";
11 while(list($nama, $nilai)=each($arrNilai)){
12     echo "Nilai $nama=$nilai<br>";
13 }
14 ?>

```

Penjelasan Program 4

Untuk menampilkan array *asosiatif* bisa menggunakan perulangan `foreach` dan `while` dengan fungsi `list()` dan `each()`. Baris ke-5 dari program 4. di atas bisa dinyatakan dengan pernyataan "untuk setiap elemen array `$arrNilai`, masukkan key/index-nya ke variabel `$nama` dan masukkan value-nya ke variabel `$nilai`". Pada baris ke-8 terdapat fungsi **`reset()`** yang berfungsi mengembalikan pointer array ke elemen pertama. Pada baris ke-10, array asosiatif ditampilkan dengan struktur `while` yang menggunakan fungsi **`list()`** dan **`each()`**.



Gambar. Tampilan Program 4

Fungsi-fungsi Array dalam PHP

PHP menyediakan lebih dari 70 fungsi untuk manipulasi array yang dapat dilihat di alamat <http://ca.php.net/manual/en/ref.array.php>.

Fungsi Pengurutan Array

- **arsort()** – Pengurutan array berdasarkan **value** secara **descending**
- **asort()** – Pengurutan array berdasarkan **value** secara **ascending**
- **krsort()** – Pengurutan array berdasarkan **index/key** secara **descending**
- **ksort()** – Pengurutan array berdasarkan **index/key** secara **ascending**
- **rsort()** – Pengurutan array berdasarkan **value** secara **descending** dengan mengubah **index/key**
- **sort()** – Pengurutan array berdasarkan **value** secara **ascending** dengan mengubah **index/key**
- **shuffle()** – Random pengurutan array

Fungsi Pengaturan Pointer Array

- **current()** – Mendapatkan elemen array yang ditunjuk oleh pointer
- **end()** – Pointer menunjuk pada elemen array terakhir
- **key()** – Mendapatkan key yang ditunjuk oleh pointer
- **next()** – Pointer menunjuk pada elemen selanjutnya
- **prev()** – Pointer menunjuk pada elemen sebelumnya
- **reset()** – Memindahkan pointer array ke awal (elemen pertama)
- **count()** – Menghitung jumlah elemen array

Fungsi Pencarian pada Array

- **array_search()** – Mencari posisi (key) dari suatu value dalam array
- **array_key_exists()** – Memeriksa suatu key ada dalam array atau tidak
- **in_array()** – Memeriksa suatu elemen ada dalam array atau tidak

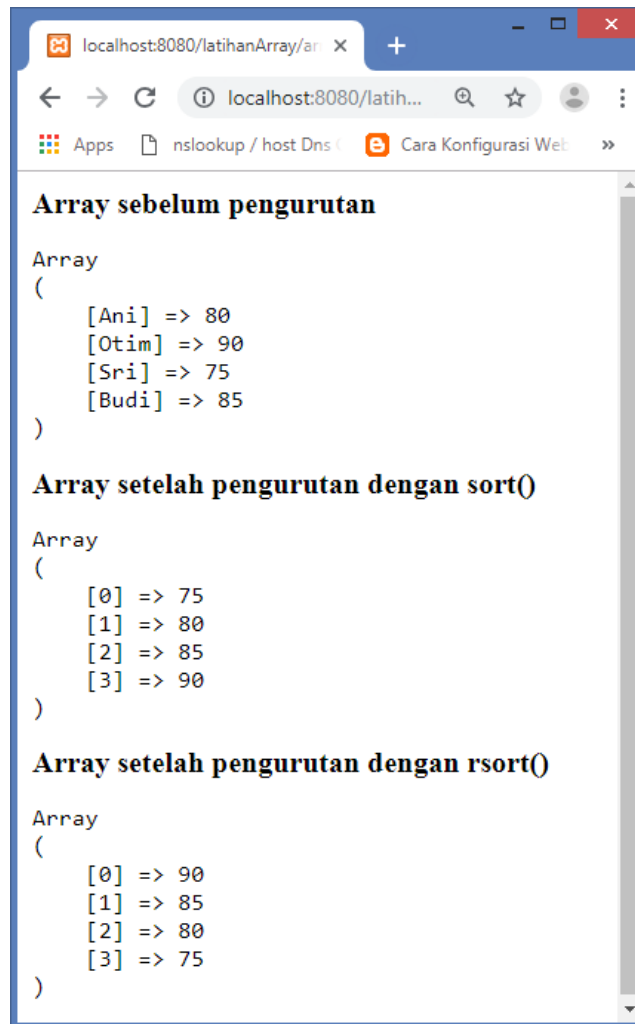
Program 5

Deskripsi : Program mengurutkan array dengan sort() dan rsort().

```

1 <?php
2 $arrNilai=array("Ani"=>80, "Otim"=>90, "Sri"=>75, "Budi"=>85);
3 echo "<b>Array sebelum pengurutan</b>";
4 echo "<pre>";
5 print_r($arrNilai);
6 echo "</pre>";
7
8 sort($arrNilai);
9 reset($arrNilai);
10
11 echo "<b>Array setelah pengurutan dengan sort()</b>";
12 echo "<pre>";
13 print_r($arrNilai);
14 echo "</pre>";
15
16 rsort($arrNilai);
17 reset($arrNilai);
18
19 echo "<b>Array setelah pengurutan dengan rsort()</b>";
20 echo "<pre>";
21 print_r($arrNilai);
22 echo "</pre>";
23 ?>

```



Gambar. Tampilan Program 5

FUNGSI DALAM PHP

Bentuk umum pendefinisian fungsi dalam PHP

```
function nama_fungsi(parameter1,...,n) {
    statement2;
}
```

Program 6

Deskripsi : Program sederhana mendefinisikan dan memanggil fungsi tanpa parameter dan tanpa return value.

```
1 <?php
2 function cetak_ganjil() {
3     for($i=0; $i<20; $i++) {
4         if($i%2 == 1) {
5             echo "$i </br>";
6         }
7     }
8 }
9 //pemanggilan fungsi
10 cetak_ganjil();
11 ?>
```

Program 7

Deskripsi : Program fungsi dengan parameter.

```
1 <?php
2 function cetak_ganjil($awal,$akhir){
3     for($i=$awal; $i<$akhir; $i++){
4         if ($i%2 == 1){
5             echo "$i ";
6         }
7     }
8 }
9 //pemanggilan fungsi
10 $a=10;
11 $b=50;
12 echo "<b>Bilangan ganjil dari $a sampai $b:</b><br>";
13 cetak_ganjil($a,$b);
14 ?>
```

Program 8

Deskripsi : Program fungsi yang mengembalikan nilai.

```
1 <?php
2 function luas_lingkaran($jari){
3     return 3.14 * $jari * $jari;
4 }
5 //pemanggilan fungsi
6 $r = 10;
7 echo "Luas lingkaran dengan jari-jari $r = ";
8 echo luas_lingkaran($r);
9 ?>
```

Program 9

Deskripsi : Program passing by value dalam fungsi.

```
1 <?php
2 function tambah_string($str){
3     $str = $str.", Jakarta";
4     return $str;
5 }
6 //
7 $str = "Universitas Budi Luhur";
8 echo "\$str = $str<br>";
9 echo tambah_string($str)."<br>";
10 echo "\$str = $str<br>";
11 ?>
```

Program 10

Deskripsi : Program passing by reference dalam fungsi.

```
1 <?php
2 function tambah_string(&$str) {
3     $str = $str.", Jakarta";
4     return $str;
5 }
6 //
7 $str = "Universitas Budi Luhur";
8 echo "\$str = $str<br>";
9 echo tambah_string($str). "<br>";
10 echo "\$str = $str<br>";
11 ?>
```

TUGAS PRAKTIKUM PHP-2

4. Buatlah program menampilkan data array 1 dimensi menggunakan *looping*!
5. Analisa dan jelaskan perbedaan jalannya program yang menggunakan *function passing by value* dengan *function passing by reference*!

Penanganan String dan Tanggal

- ❖ Pengantar String
- ❖ Fungsi-fungsi String dalam PHP
- ❖ Fungsi-fungsi Operasi Tanggal dalam PHP

Pengantar String

String merupakan kumpulan dari karakter. Dalam PHP, karakter sama dengan byte, dimana terdapat 256 karakter. PHP tidak mendukung native-unicode. Untuk menuliskan sebuah string dalam PHP, bisa menggunakan 3 (tiga) cara, yaitu dengan :

1. single quoted – kutip tunggal (')
2. double quoted – kutip ganda (")
3. heredoc sintaks

Fungsi-fungsi String dalam PHP

PHP menyediakan lebih dari 90 fungsi untuk manipulasi string.

Beberapa fungsi manipulasi string yang sering digunakan, antara lain :

- **addslashes()** – menambahkan backslashes (\) di setiap tanda kutip (quote) dalam string.
- **chr()** – menghasilkan karakter dari bilangan ASCII
- **crypt()** – menghasilkan string terenkripsi searah.
- **echo()** – menampilkan satu atau beberapa string.
- **explode()** – memecah string berdasarkan suatu delimiter (separator).
- **htmlentities()** – mengubah semua tag html menjadi HTML entities.
- **htmlspecialchars()** – mengubah semua karakter khusus menjadi HTML entities
- **implode()** – menggabungkan elemen array menjadi string dengan suatu delimiter (separator).
- **join()** – sama dengan implode()
- **ltrim()** – menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di awal string.
- **md5()** – menghasilkan string terenkripsi searah.
- **nl2br()** – menambahkan HTML line break (
) sebelum semua baris baru dalam string.
- **number_format()** – mengatur format angka.
- **ord()** – menghasilkan kode ASCII dari suatu karakter.
- **rtrim()** – menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di akhir string.
- **split()** – sama dengan explode
- **str_repeat()** – mengulang string
- **str_replace()** – mengganti semua string dalam pola menjadi suatu string.
- **strip_tags()** – mengabaikan HTML dan PHP tag dalam string.
- **stripslashes()** – menghilangkan backslashes (\) dalam string.
- **strlen()** – menghitung panjang string.
- **strpos()** – mencari posisi pertama sebuah string dalam string.
- **strrchr()** – mencari posisi terakhir sebuah karakter dalam string.
- **strrpos()** – mencari posisi terakhir sebuah string dalam string.
- **strrev()** – membalik string.
- **strstr()** – mencari posisi pertama sebuah karakter dalam string.
- **strtolower()** – mengubah string menjadi huruf kecil (lower-case).
- **strtoupper()** – mengubah string menjadi huruf kapital (upper-case)
- **substr()** – memotong string
- **trim()** – menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di akhir dan awal string.
- **ucfirst()** – mengubah huruf pertama dari semua string menjadi Upper-case.
- **ucwords()** – mengubah huruf pertama tiap kata dalam string menjadi upper- case.
- **wordwrap()** – memotong sejumlah karakter dalam string dengan *string- break-character*.

Program

Deskripsi : Program penggunaan fungsi `strtolower`, `strtoupper`, `ucfirst`, `ucwords`, `strrev`, dan `strlen` dalam string.

```
1 <?php
2 $str="Everything i do, i do it for YOU";
3
4 echo "<b>Ini String asli</b>: $str";
5 echo "<br><b>Ini strtolower()</b>: ".strtolower($str);
6 echo "<br><b>Ini strtoupper()</b>: ".strtoupper($str);
7 echo "<br><b>Ini ucfirst()</b>: ".ucfirst($str);
8 echo "<br><b>Ini ucwords()</b>: ".ucwords($str);
9 echo "<br><b>Ini strrev()</b>: ".strrev($str);
10 echo "<br><b>Jumlah karakter</b>: ".strlen($str);
11 ?>
```

Penjelasan Program

- `strtolower`, untuk mengubah string menjadi lower-case.
- `strtoupper`, untuk mengubah string menjadi upper-case.
- `ucfirst`, untuk mengubah huruf pertama string menjadi upper-case.
- `ucwords`, untuk mengubah huruf pertama tiap kata menjadi upper-case.
- `strrev`, untuk membalik string.
- `strlen`, untuk menghitung panjang string.

```
Ini String asli: Everything i do, i do it for YOU
Ini strtolower(): everything i do, i do it for you
Ini strtoupper(): EVERYTHING I DO, I DO IT FOR YOU
Ini ucfirst(): Everything i do, i do it for YOU
Ini ucwords(): Everything I Do, I Do It For YOU
Ini strrev(): UOY rof ti od i ,od i gnihtyrevE
Jumlah karakter: 32
```

Program

Deskripsi : Program penggunaan fungsi `number_format` untuk mengatur format tampilan angka.

```
1 <?php
2 $number=1234.56;
3
4 //english notation (default)
5 $english_format_number=number_format($number);
6 echo $english_format_number; //1,235
7
8 //French notation
9 $nombre_format_francais=number_format($number,2,',','');
10 echo "<br>".$nombre_format_francais; //1 234,56
11
12 //indonesian notation
13 $format_indonesia=number_format($number,2,',','');
14 echo "<br>".$format_indonesia; //1.234,56
15
16 $number=1234.5678;
17 //english notation without thousands separator
18 $english_format_number=number_format($number,2,'.','');
19 echo "<br>".$english_format_number; //1234.57
20 ?>
```

Penjelasan Program

Program di atas merupakan contoh penggunaan fungsi `number_format` untuk mengatur tampilan angka. Secara default, fungsi `number_format` akan memformat angka dengan model Inggris yaitu setiap tiga digit dipisahkan dengan koma (,) dan tanpa pecahan. Pada baris ke-9, fungsi `number_format` akan mengubah format angka dengan 2 digit pecahan (parameter kedua), pecahan dipisahkan dengan koma (parameter ketiga) dan setiap tiga digit dipisahkan dengan spasi (parameter keempat).

Program

Deskripsi : Program penggunaan fungsi `explode` untuk memecah string menjadi array.

```
1 <?php
2 //Contoh 1
3 $buah="mangga pisang rambutan apel jeruk kedondong";
4 $buahan=explode(" ", $buah);
5 echo $buahan[0]. "<br>"; //mangga
6 echo $buahan[3]. "<br><br>"; //apel
7
8 // Contoh 2
9 $data="foo*:1023:1000::/home/foo:/bin/sh";
10 list($user,$pass,$uid,$gid,$gecos,$home,$shell)=explode(":", $data);
11 echo $user. "<br>"; //foo
12 echo $pass; // *
13 ?>
```

Penjelasan Program

Program di atas merupakan contoh penggunaan fungsi `explode` untuk memecah string berdasarkan pola (aturan) tertentu. Pada baris ke-4, dengan fungsi `explode()` akan memecah string `$buah` berdasarkan spasi dan masing-masing akan menjadi elemen array `$buahan`.

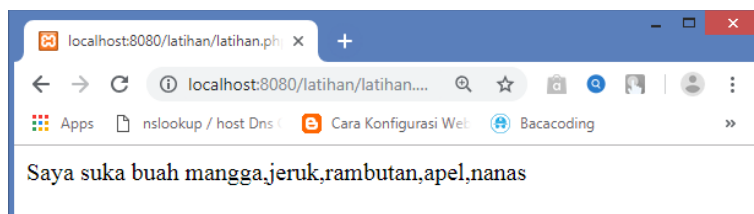
Program

Deskripsi : Program penggunaan fungsi `implode` untuk menggabungkan array menjadi string.

```
1 <?php
2 $buahan=array('mangga','jeruk','rambutan','apel','nanas');
3 $buah=implode(",",$buahan);
4 echo "Saya suka buah ".$buah;
5 //mangga, jeruk, rambutan, apel, nanas
6 ?>
```

Penjelasan Program

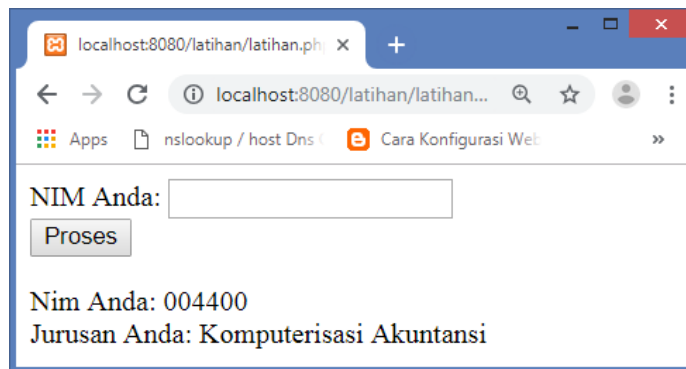
Program di atas merupakan contoh penggunaan fungsi `implode()` untuk menggabungkan semua isi array menjadi satu string. Fungsi `implode` merupakan kebalikan dari fungsi `explode`. Pada baris ke-3, dengan fungsi `implode()` akan menggabungkan setiap elemen array `$buahan` menjadi string `$buah` dengan pemisah koma di antara tiap elemen.



Program

Deskripsi : Program penggunaan fungsi substr untuk memotong string.

```
1 <form action="" method="post">
2 NIM Anda: <input type="text" name="txtnim"/><br/>
3 <input type="submit" name="submit" value="Proses"/>
4 </form>
5
6 <?php
7 if(isset($_POST['submit'])){
8     $nim=$_POST['txtnim'];
9     $a=substr($nim,2,2);
10    switch($a){
11        case '11':
12            $jurusan="Teknik Informatika";
13            break;
14        case '22':
15            $jurusan="Sistem Informasi";
16            break;
17        case '33':
18            $jurusan="Sistem Komputer";
19            break;
20        case '44':
21            $jurusan="Komputerisasi Akuntansi";
22            break;
23        default:
24            $jurusan="Salah jurusan";
25    }
26    echo "Nim Anda: $nim<br>";
27    echo "Jurusan Anda: $jurusan";
28 }
29 ?>
```



Penjelasan Program

Program di atas merupakan contoh penggunaan fungsi substr (baris ke-9) untuk memotong string. Pada baris ke-9 tersebut, string \$nim akan dipotong mulai karakter index ke-2 (index dimulai dari 0) sepanjang 2 karakter.

Fungsi-fungsi Operasi Tanggal dalam PHP

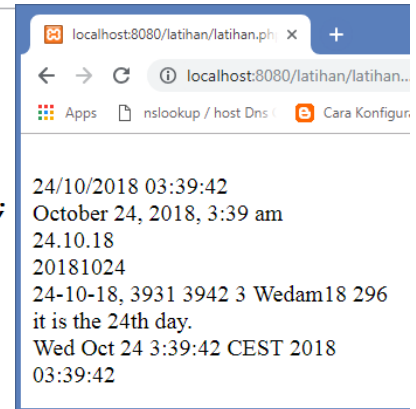
Fungsi dalam Operasi tanggal dalam PHP yang utama dan sering digunakan adalah fungsi date(). Fungsi ini akan menghasilkan tanggal dan waktu server sekarang. Beberapa pilihan parameter dari fungsi date() ini dapat dilihat pada tabel berikut ini :

parameter	Keterangan	Contoh nilai
Hari		
d	Tanggal, 2 digit dengan nol	01 s/d 31
D	Tiga digit nama hari dalam seminggu	Mon s/d Sun
j	Tanggal tanpa nol	1 s/d 31
l (lowercase 'l')	Nama hari lengkap dalam seminggu	Sunday through Saturday
N	Urutan hari dalam seminggu	1 (untuk Monday) s/d 7 (untuk Sunday)
S	Akhiran angka English untuk tanggal, 2 karakter	st, nd, rd atau th.
w	Urutan hari dalam seminggu	0 (untuk Sunday) s/d 6 (untuk Saturday)
z	Urutan hari dalam setahun	0 s/d 365
Minggu		
W	Urutan minggu dalam setahun	Contoh: 42 (minggu ke-42 dalam tahun ini)
Bulan		
F	Nama bulan lengkap	January s/d December
m	Urutan bulan dalam setahun dengan nol	01 s/d 12
M	Tiga digit nama bulan dalam setahun	Jan s/d Dec
n	Urutan bulan dalam setahun,	1 s/d 12
Tahun		
t	tanpa nol Jumlah hari dalam tiap bulan	28 s/d 31
Y	4 (empat) digit tahun	Contoh: 1999 atau 2006
y	2 (dua) digit tahun	Contoh: 99 or 06
Waktu		
a	Lowercase Ante meridiem dan Post meridiem	am atau pm
A	Uppercase Ante meridiem dan Post meridiem	AM atau PM
g	Jam format 12 tanpa nol	1 s/d 12
G	Jam format 24 tanpa nol	0 s/d 23
h	Jam format 12 dengan nol	01 s/d 12
H	Jam format 24 dengan nol	00 s/d 23
i	Menit dengan nol	00 s/d 59
s	Detik dengan nol	00 s/d 59

Program

Deskripsi : Program menampilkan date/time dengan berbagai parameter.

```
1 <?php
2 echo "<br>".date("d/m/Y H:i:s");
3 echo "<br>".date("F j, Y, g:i a");
4 echo "<br>".date("d.m.y");
5 echo "<br>".date("Ymd");
6 echo "<br>".date('j-m-y, it is w Day z ');
7 echo "<br>".date('\i\t \i\s \t\h\e jS \d\a\y. ');
8 echo "<br>".date("D M j G:i:s T Y");
9 echo "<br>".date("H:i:s");
10 ?>
```



Upload File

PHP, sebagai server-side-scripting, sangat memungkinkan untuk menangani upload file ke server. Ada beberapa hal yang perlu diperhatikan, yaitu :

1. Pada Form HTML harus ditambahkan atribut :
ENCTYPE="multipart/form-data"
2. Form inputan upload file dapat menggunakan tag <input> dengan value atribut TYPE="FILE".
3. Untuk menangani inputan, PHP menyediakan sebuah variabel array global yaitu \$_FILES. Index dari variabel ini antara lain :
 - \$_FILES['file']['name'] : Nama asli dari file yang diupload
 - \$_FILES['file']['tmp_name'] : Nama temporary file yang diupload
 - \$_FILES['file']['size'] : Ukuran file asli (dalam byte)
 - \$_FILES['file']['type'] : MIME type file yang diupload
4. Destination folder file upload harus bisa writable (bisa diakses), biasanya dengan permission 777 atau 775.

Program 1

Nama File : form_upload.php

Deskripsi : Program menampilkan form upload file.

```
1 <html>
2 <head>
3     <title>Upload File</title>
4 </head>
5 <body>
6 <form action="upload.php" method="post" enctype="multipart/form-data">
7     Upload File: <input type="file" name="file"/><br/>
8     <input type="submit" name="Upload" value="Upload"/>
9 </form>
10 </body>
11 </html>
```

Program 2

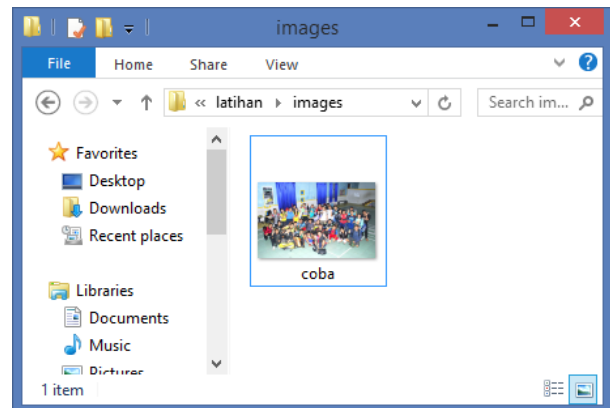
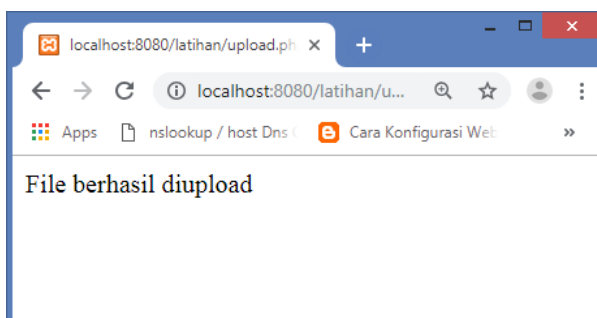
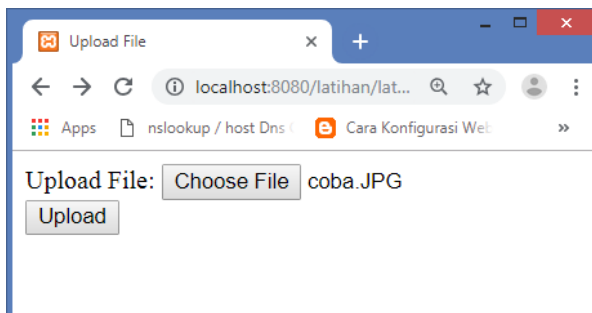
Nama File : upload.php

Deskripsi : Program proses upload file.

```
1 <?php
2 if(isset($_POST['Upload'])) {
3     $dir_upload="images/";
4     $nama_file=$_FILES['file']['name'];
5     //
6     if(is_uploaded_file($_FILES['file']['tmp_name'])) {
7         $cek=move_uploaded_file($_FILES['file']['tmp_name'],$dir_upload.$nama_file);
8         if($cek) {
9             die("File berhasil diupload");
10        }else{
11            die("File gagal diupload");
12        }
13    }
14 }
15 ?>
```

Penjelasan Program

Program di atas merupakan program sederhana penanganan upload file. Fungsi **is_uploaded_file()** pada baris ke-6 akan mengupload file yang telah dipilih melalui form pada program 7.1 ke folder temporary (sementara). Selanjutnya pada baris ke-7, file yang sudah diupload ke folder temporary akan dipindahkan (move) ke direktori yang diinginkan menggunakan fungsi **move_uploaded_file()**. Lihat baris ke-7!



Session dan Cookies

- ❖ Session
- ❖ Cookies

Session

Session merupakan hal yang cukup penting dalam aplikasi berbasis web. Dengan session memungkinkan programmer menyimpan informasi user secara semi-permanen, artinya selama masa tertentu informasi akan tersimpan. Penyimpanan isi variabel session berada di server, jadi tidak bisa diakses secara langsung oleh client.

Dalam aplikasi berbasis web, session banyak digunakan sebagai autentifikasi login. Dengan session memungkinkan programmer mengatur siapa saja yang bisa mengakses suatu halaman. Misalnya saja, untuk melihat halaman kotak surat pada email, kita harus login terlebih dahulu. Dalam proses login antara lain akan terjadi pembuatan suatu session yang akan dibawa oleh user di setiap halaman. Di halaman kotak surat, session tersebut akan diperiksa. Jika session benar maka user dipersilahkan membuka halaman kotak surat, namun jika salah maka user tidak bisa membuka halaman kotak surat dan biasanya akan diminta untuk login terlebih dahulu. Itulah sebabnya, user tidak bisa mengakses halaman kotak surat secara langsung tanpa melakukan login.

Dalam penanganan session terdapat beberapa proses yang perlu diperhatikan :

- Proses pembuatan session
- Proses pemeriksaan session
- Proses penghapusan session

Selanjutnya bagaimana session itu sendiri dijalankan? Agar proses penyimpanan dalam session berjalan, PHP melakukan beberapa hal berikut ini :

1. PHP meng-generate (membentuk) sebuah ID session.
ID session ini merupakan sejumlah deret angka random yang unik untuk setiap user dan hampir tidak mungkin bisa ditebak. ID session disimpan oleh PHP di dalam variabel sistem PHP dengan nama PHPSESSID
2. PHP menyimpan nilai yang akan Anda simpan dalam session di dalam file yang berada di server.
Nama file tempat penyimpanan session tersebut sesuai (sama) dengan ID session. File disimpan dalam suatu direktori yang ditunjukkan oleh session.save_path dalam file php.ini.
3. PHP melempar ID session ke setiap halaman.
4. PHP mengambil nilai session dari file session untuk setiap halaman session.

Program 1

Nama File : session01.php

Deskripsi : Program menciptakan session.

```
1 <?php
2 session_start();
3 if(isset($_POST['login'])){
4     $user=$_POST['user'];
5     $pass=$_POST['pass'];
6
7     //periksa login
8     if($user=="budi" && $pass=="123"){
9         //menciptakan session
10        $_SESSION['login']=$user;
11        //menuju ke halaman pemeriksaan session
12        echo "<h1>Anda berhasil LOGIN</h1>";
```



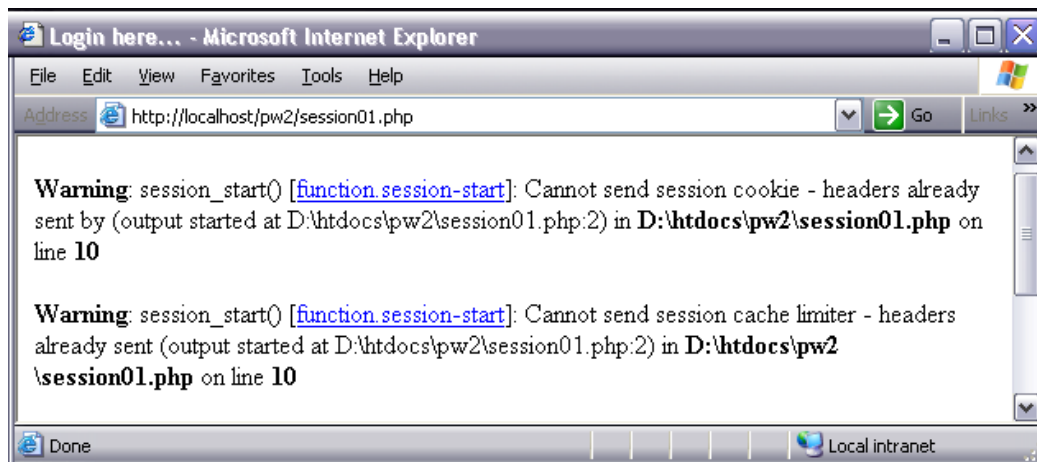
```

13         echo "<h2>Klik <a href='session02.php'>di sini(session02.php)</a>
14         untuk menuju ke halaman pemeriksaan session";
15     }else{
16         header("location: latihan.php");
17     }
18 }else{
19 ?>
20 <html>
21 <head>
22     <title>Login here...</title>
23 </head>
24 <body>
25     <form action="" method="post">
26     <h2>Login Here...</h2>
27     Username: <input type="text" name="user"/><br/>
28     Password: <input type="password" name="pass"/><br/>
29     <input type="submit" name="login" value="Log In"/>
30     </form>
31 </body>
32 </html>
33 <?php
34 }
35 ?>

```

Penjelasan Program 1

Pada program di atas terdapat fungsi **session_start()** yang berada pada baris ke-9. Fungsi session_start() berfungsi untuk memulai sebuah session. Fungsi harus dipanggil pertama kali dalam suatu halaman PHP, artinya fungsi session_start() harus dipanggil sebelum sesuatu tampil di layar, walaupun hal itu hanya berupa spasi. Jika sebelum session_start() terdapat output yang ditampilkan maka akan terjadi error seperti pada gambar berikut ini :



Gambar Error yang terjadi karena sebelum session_start() sudah ada output

Selanjutnya, program 8.1 akan menampilkan sebuah form login sederhana di layar. Form terdiri dari form inputan username, password dan sebuah tombol login. Jika tombol login ditekan maka pemeriksaan kondisi pada baris 3 akan bernilai TRUE dan perintah yang ada di blok **if** akan dieksekusi. Baris 4 dan 5 merupakan perintah untuk mengambil nilai pada form inputan username dan password. Nilai username dan password tersebut diperiksa pada baris 8. Pemeriksaan username dan password ini, untuk aplikasi web yang sudah menggunakan database, bisa diganti dengan pemeriksaan username dan password ke tabel database.

Pada baris 10, terdapat perintah sebagai berikut :

```
$_SESSION['login'] = $user;
```

Baris perintah tersebut merupakan perintah untuk membuat session baru (*create session*) dimana nama session adalah "login" dan isi dari session adalah \$user. \$_SESSION merupakan sebuah variabel array global yang didefinisikan oleh PHP, jadi variabel ini harus ditulis dengan huruf kapital (*upper-case*).

Selanjutnya, pada baris 12-14 akan ditampilkan pesan bahwa login berhasil dan juga sebuah link menuju ke halaman pemeriksaan session (session02.php, program 8.2).

Program 2

Nama File : session02.php

Deskripsi : Program pemeriksaan session.

```
1 <?php
2 session_start();
3 //pemeriksaan session
4 if(isset($_SESSION['login'])){ //jika sudah login
5     //menampilkan isi session
6     echo "<h1>Selamat Datang ".$_SESSION['login']."</h1>";
7     echo "<h2>Halaman ini hanya bisa diakses jika Anda sudah login</h2>";
8     echo "<h2>Klik <a href='session03.php'>di sini (session03.php)</a>
9         untuk LOGOUT</h2>";
10 }else{
11     //session belum ada, artinya belum login
12     die("Anda belum login! Anda tidak berhak masuk ke halaman ini.
13     Silahkan login <a href='session01.php'>di sini</a>");
14 }
15 ?>
```

Penjelasan Program 2

Program 2 di atas diawali dengan pemanggilan fungsi **session_start()**. Hal ini berarti dalam halaman program ini terdapat operasi session. Pada baris 4 program di atas, terdapat pemeriksaan keberadaan variabel \$_SESSION['login'] dengan menggunakan fungsi **isset()**. Pemeriksaan kondisi ini akan bernilai TRUE jika \$_SESSION['login'] sudah terbentuk. Jika bernilai TRUE maka blok program baris 6-9 yang akan dieksekusi, dan sebaliknya, jika FALSE maka blok program baris 12-13 yang akan dijalankan. Blok program baris 6-9 hanya bisa diakses saat user sudah login dengan benar. Dengan kata lain, bagian tersebut merupakan bagian yang bersifat private (terbatas).

Untuk menampilkan isi variabel session dapat dilakukan seperti pada baris 6 program di atas. Cukup dengan meng-echo variabel \$_SESSION['login'].

Jika user belum login (tidak melewati halaman login) atau langsung mengakses halaman ini, maka akan ditampilkan pesan bahwa user tidak berhak masuk ke halaman ini. Fungsi **die()** pada baris 12 digunakan untuk menampilkan suatu pesan sekaligus men-*terminate* program, artinya perintah selain die() akan diabaikan.

Program 3

Nama File : session03.php

Deskripsi : Program menghapus session.

```
1 <?php
2 session_start();
3 if(isset($_SESSION['login'])) {
4     unset($_SESSION);
5     session_destroy();
6
7 echo "<h1>Anda sudah berhasil LOGOUT</h1>";
8 echo "<h2>Klik <a href='session01.php'>di sini</a>
9     untuk LOGIN kembali</h2>";
10 echo "<h2>Anda sekarang tidak bisa masuk ke halaman
11     <a href='session02.php'>session02.php</a> lagi</h2>";
12 }
13 ?>
```

Penjelasan Program 3

Program 3 merupakan contoh program penghapusan session. Penghapusan session biasanya digunakan ketika user melakukan logout. Proses penghapusan session dilakukan dengan memanggil fungsi `unset()` dan fungsi `session_destroy()`. Fungsi **`unset()`** digunakan untuk menghapus atau menghilangkan suatu variabel. Sedangkan **`session_destroy()`** digunakan untuk menghapus semua data yang berhubungan dengan session tersebut.

Cookies

Seperti halnya session, cookies juga merupakan sebuah konsep penyimpanan informasi user. Hanya saja, jika session tempat penyimpanan berada di server, cookies berada di client. Oleh karena itu, konsep cookies sebaiknya jangan digunakan untuk menyimpan informasi login user seperti username, password dsb. Selain user bisa melihat informasi yang disimpan, user juga bisa men-disable cookies itu sendiri. Jika cookies di-disable, maka program yang memanfaatkan cookies tentunya tidak akan berjalan dengan baik.

Cookies sendiri biasanya dipakai dalam aplikasi shopping cart. Biasa digunakan untuk menyimpan sementara, produk-produk yang dipilih oleh pengunjung pada saat berbelanja.

TUGAS PRAKTIKUM PHP-3

6. Buatlah form login yang menerapkan session, sehingga ketika berhasil login maka session terbentuk, dan ketika logout session destroy.