

WEB PROGRAMMING :

ARRAY, FUNGSI & MODULARISASI

Tujuan :

1. Memahami tentang Array di PHP
2. Memahami penggunaan Fungsi dalam PHP
3. Memahami tentang Modularisasi dalam PHP

I. ARRAY

Array merupakan variabel yang berisi lebih dari 1 variabel, atau dapat dikatakan sebagai variabel jamak.

Sebagai analogi, ketika kita mempunyai banyak nama buah (kita anggap ini sebagai variabel), maka daripada kita menyatakan tiap nama buah dengan variabel \$buah1, \$buah2, \$buah3, ...dst. Maka akan lebih baik dan efisien jika kita menggunakan variabel \$buah yang berupa array.

Variabel array pada PHP, kadang tidak perlu kita definisikan. Namun untuk indeks array, kita menggunakan indeks mulai dari 0 hingga sejumlah elemen array dikurang 1.

Assignment Array

Ada beberapa cara untuk memberi nilai pada suatu variabel array, antara lain :

Contoh 1 :

```
$buah[0] = "nanas";  
$buah[1] = "mangga";  
$buah[2] = "jambu";
```

Contoh 2 :

```
$buah[] = "nanas";  
$buah[] = "mangga";  
$buah[] = "jambu";
```

Pada contoh ini, PHP secara otomatis akan memberi indeks sesuai nilai indeks terakhir dari array. Maka pada variabel \$buah indeks 0 berisi "nanas", indeks 1 berisi "mangga" dan indeks 2 akan berisi "jambu".

Contoh 3 :

```
$buah = array("nanas", "mangga", "jambu");
```

Pada contoh ini, suatu array dideklarasikan dengan memberi nilai secara langsung. Hasilnya akan sama dengan contoh-contoh sebelumnya.

Array Asosiatif

Array asosiatif tidak menggunakan integer sebagai indeks, namun menggunakan string sebagai indeks.

Contoh 1:

```
$hobi["nanang"] = "renang";  
$hobi["eko"] = "merokok";
```

Contoh 2 :

```
$hobi = array("nanang" => "renang", "eko" => "merokok");
```

II. FUNGSI

Fungsi terbagi menjadi 3 :

1. Built-in
2. User Defined Function (UDF)
3. External

Fungsi Built-in

Fungsi yang sudah disediakan oleh PHP dan pemrogram dapat langsung menggunakannya.

Contoh :

Fungsi Matematika : sin, cos, tan, asin, deg2rad, dll.

Fungsi Tanggal & waktu : date, checkdate

Fungsi String : strlen, strpos, strtolower, strtoupper, substr, dll.

Fungsi UDF

Fungsi yang dibuat sendiri oleh pemrogram. Dapat diletakkan dimana saja dalam script PHP, tapi sebaiknya diletakkan paling atas agar memudahkan dalam proses debugging dalam tahap pembuatan dan pengembangan program.

Deklarasi UDF :

```
function namafungsi([parameter]) {  
    Statement;  
    Statement;  
    ....  
}
```

Pemanggilan UDF

Sintaks :

```
$varhasil = &namafungsi ([parameter]);
```

Penggunaan tanda & tidak mutlak, karena PHP secara otomatis akan menganggap sebagai fungsi karena tidak didahului dengan tanda \$.

Fungsi External (Tidak dibahas dalam bab ini)

Daftar fungsi yang belum diaktifkan dalam keseluruhan modul PHP, untuk mengaktifkannya secara otomatis dengan meregistrasikan pada file PHP.INI.

III. MODULARISASI

Modularisasi dalam pemrograman umum dilakukan dan sangat diperlukan untuk mempermudah debugging dan pengembangan program. Modularisasi berarti melakukan pembuatan program berdasarkan modul-modul. Setiap modul dikembangkan untuk tujuan atau fungsi khusus. Modul dibuat secara general (umum). Modul dapat berupa fungsi atau prosedur.

Require

Statement Require digunakan untuk membaca nilai variable dan fungsi-fungsi dari sebuah file lain.

Cara penulisan statement Require adalah:

```
require(namafile);
```

Teknik ini cocok untuk membuat template (pola) yang memudahkan proses pengembangan aplikasi dengan menggunakan pola tampilan. Misal dalam membuat design web yang dilakukan dengan kerja tim, pola dan gambar-gambar yang dibutuhkan dapat dibuat dengan teknik ini.

Statement Require ini tidak dapat dimasukkan didalam suatu struktur looping misalnya while atau for. Karena hanya memperbolehkan pemanggilan file yang sama tersebut hanya sekali saja.

Include

Statement Include akan menyertakan isi suatu file tertentu. Include dapat diletakkan di dalam suatu looping misalkan dalam statement for atau while.

Syntax penulisan :

include (namafile);

LATIHAN – LATIHAN :

1. Menampilkan Array dengan menunjuk nomor indeksnya

```
<?php
$anak[0] = "Faruq";
$anak[1] = "Alya";
$anak[2] = "Zahro";
echo "Isi array \ $anak[0] adalah $anak[0]";
echo "<br>";
echo "Isi array \ $anak[1] adalah $anak[1]";
echo "<br>";
echo "Isi array \ $anak[2] adalah $anak[2]";
echo "<br>";
?>
```

2. Menampilkan Array tanpa menunjuk nomor indeksnya

```
<?php
$nama[] = "Muhammad";
$nama[] = "Faruq";
$nama[] = "Zakiuddin";
echo "Hallo";
echo "Isi array \ $anak[0] adalah $nama[0]";
echo "<br>";
echo "Isi array \ $anak[1] adalah $nama[1]";
echo "<br>";
echo "Isi array \ $anak[2] adalah $nama[2]";
echo "<br>";
?>
```

3. Menampilkan Array dengan foreach

```
<?php
$anak[0] = "Faruq";
$anak[1] = "Alya";
$anak[2] = "Zahro";
foreach ($anak as $value) {
    echo "Nama anak : $value";
    echo "<br>";
}
?>
```

4. Menampilkan Array secara Asosiatif

```
<?php
$telpon["Farid"] = "5982345";
$telpon["Fikri"] = "7782345";
$telpon["Nida"] = "4562345";
echo "Telpon Farid : ".$telpon ['Farid'];
echo "<br>";
echo "Telpon Fikri : ".$telpon ['Fikri'];
echo "<br>";
echo "Telpon Nida : ".$telpon ['Nida'];
echo "<br>";
?>
```

5. Pemakaian Fungsi Built-in : Matematika

```
<?php
$a = pow(2,10);      //Fungsi perpangkatan
$b = sqrt(100);      //Fungsi akar
$c = ceil(4.25);     //Pembulatan keatas
$d = floor(4.25);    //Pembulatan kebawah
echo "2 pangkat 10 = $a <br>";
echo "akar 100 = $b <br>";
echo "ceil(4.25) = $c <br>";
echo "floor(4.25) = $d <br>";
?>
```

6. Pemakaian Fungsi Built-in : Date & Time

```
<?php
$skr = date("d/m/Y");
echo "Sekarang adalah $skr <br>";
$waktu = date("h:i:s A"); //A menunjukkan AM atau PM
echo "Jam menunjukkan pukul : $waktu";
?>
```

7. Pemakaian Fungsi Built-in : String

```
<?php
$str = "Belajar PHP ternyata Menyenangkan";
echo strtolower($str); //Ubah huruf ke kecil semua
echo "<br>";
echo strtoupper($str); //Ubah huruf ke besar semua
echo "<br>";
echo str_replace("Menyenangkan", "mudah lho", $str);
//Mengganti string
?>
```

8. Pemakaian Fungsi UDF : Tanpa Nilai Balik

```
<?php
function psgpjg ($pjg, $lbr) {
    $luas = $pjg * $lbr;
    echo "$luas <br>";
}
$bil1 = 5;
$bil2 = 3;
echo "Luas persegi panjang dengan pjg 5 dan lebar 3 =";
psgpjg($bil1, $bil2);
?>
```

8. Pemakaian Fungsi UDF : Berisi Nilai Balik → menggunakan return

```
<?php
function psgpjg ($pjg, $lbr) {
    $luas = $pjg * $lbr;
    return $luas;
}
$bil1 = 5;
$bil2 = 3;
echo "Luas persegi panjang dengan pjg 5 dan lebar 3 =";
echo psgpjg($bil1, $bil2);
?>
```

9. Pemakaian Fungsi : Tanpa nilai parameter

```
<?php
function garis() {
    echo "<hr>";
}
echo "Ini contoh fungsi yang tanpa parameter <br>";
garis();
echo "Lihat perbedaan dengan fungsi yang dengan parameter <br>";
garis();
?>
```

10. Modularisasi menggunakan require

a. Simpan file berikut dengan nama : contoh_require.php

```
<?php
$a="Saya sedang belajar PHP";
function tulistebal($teks)
{
    echo "<b>$teks</b>";
}
?>
```

b. Buat file untuk memanggil file php diatas.

```
<?php
require("contoh_require.php"); //Akan dipanggil 1x saja
                                //dalam file php ini
tulistebal("Ini adalah tulisan tebal");
echo "<br>";
echo $a; //Mengambil nilai dari require
?>
```

11. Modularisasi menggunakan include

a. Simpan file berikut dengan nama : contoh_include.php

```
<?php
echo("----- <br>");
echo("Belajar PHP Dasar <br>"); echo("----- <br>");
echo("<br>");
?>
```

b. Buat file untuk memanggil file php diatas.

```
<?php
for ($b=1; $b<5; $b++)
{
    include("contoh_include.php");
    //Include bisa dipanggil lbh dari 1x
}
?>
```

TUGAS :

1. Buatlah program dengan menggunakan array untuk menghitung penjumlahan antara 2 matriks dimensi 2x2.
2. Buatlah program untuk mencari nilai max dari 3 buah nilai, gunakan fungsi dengan 3 nilai tsb dikirimkan sebagai parameter ke fungsi.
3. Modifikasi program no 2, jadikan fungsi yang telah anda buat menjadi sebuah file dan panggil dengan file yang lain (pakai require).