

# ***Pemrograman Web***

## ***PHP***

### ***Perulangan, Array, Function***

# Tujuan

- Mampu menerapkan struktur kondisi Perulangan dalam pemrograman PHP
- Mampu mengoperasikan jenis-jenis Array
- Mampu menjalankan perintah Function dalam pemrograman PHP

# Perulangan

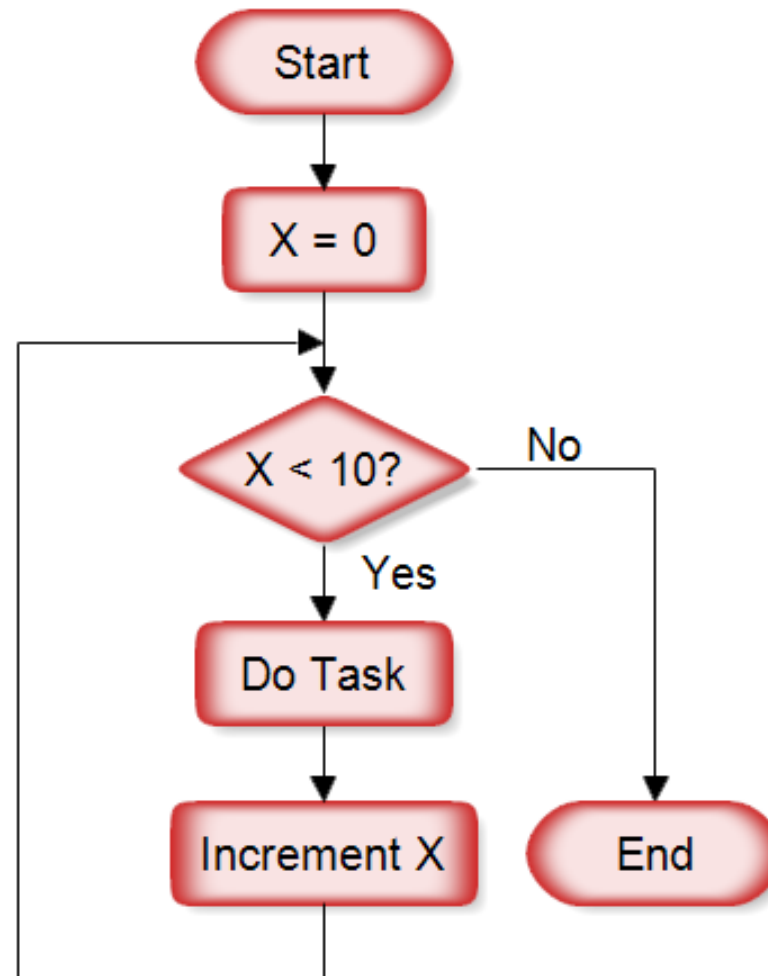


# For

- Bentuk perulangan paling sederhana
- Menjalankan operasi secara berulang-ulang sesuai dengan jumlah perulangan yang telah ditentukan
- Digunakan ketika telah diketahui sampai berapa kali suatu statement berulang untuk dieksekusi

```
for (initialization; condition; increment){  
    code to be executed;  
}
```

# Flowchart



# Contoh For

```
<?PHP
```

```
    for ($a=1; $a<=5; $a++) {  
        echo "Baris ke-$a <br />";  
    }
```

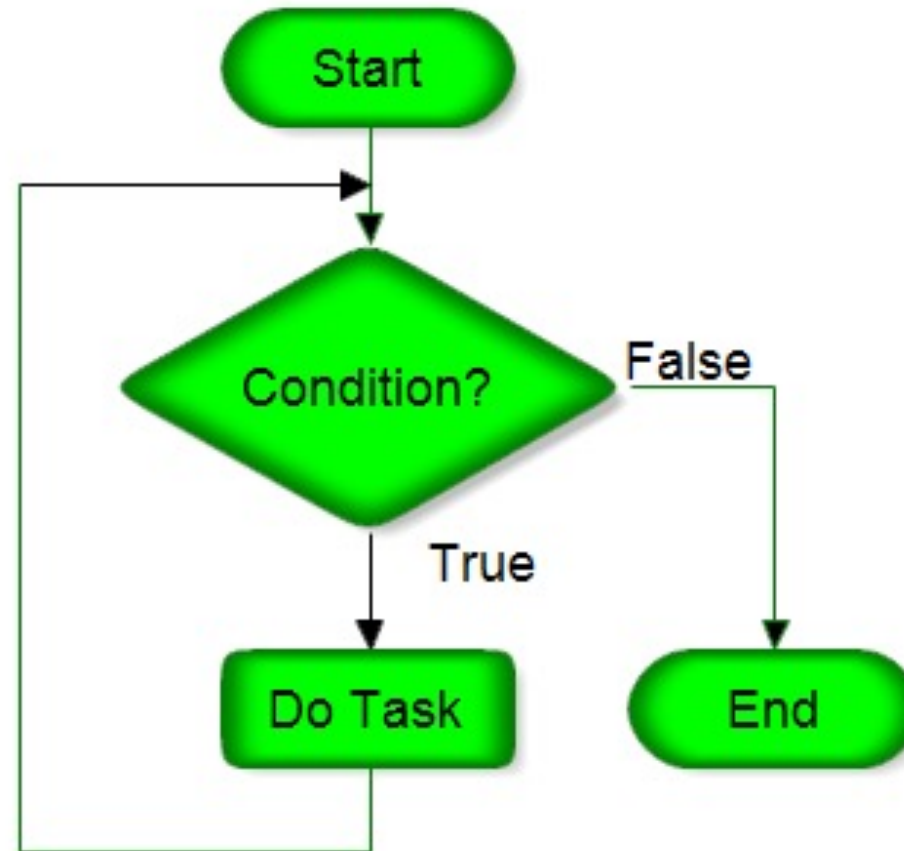
```
?>
```

# While

- Fungsi dari *while* ialah untuk menjalankan operasi secara berulang-ulang hingga ditemukan kondisi untuk menghentikannya
- Pemeriksaan kondisi di awal

```
while (condition) {  
    code to be executed;  
}
```

# Flowchart While





# Contoh While

```
<?PHP
```

```
    $awal = 1;
```

```
    while ($awal <= 5) {
```

```
        echo "Nilai Sekarang adalah $awal <br>";
```

```
        $awal++;
```

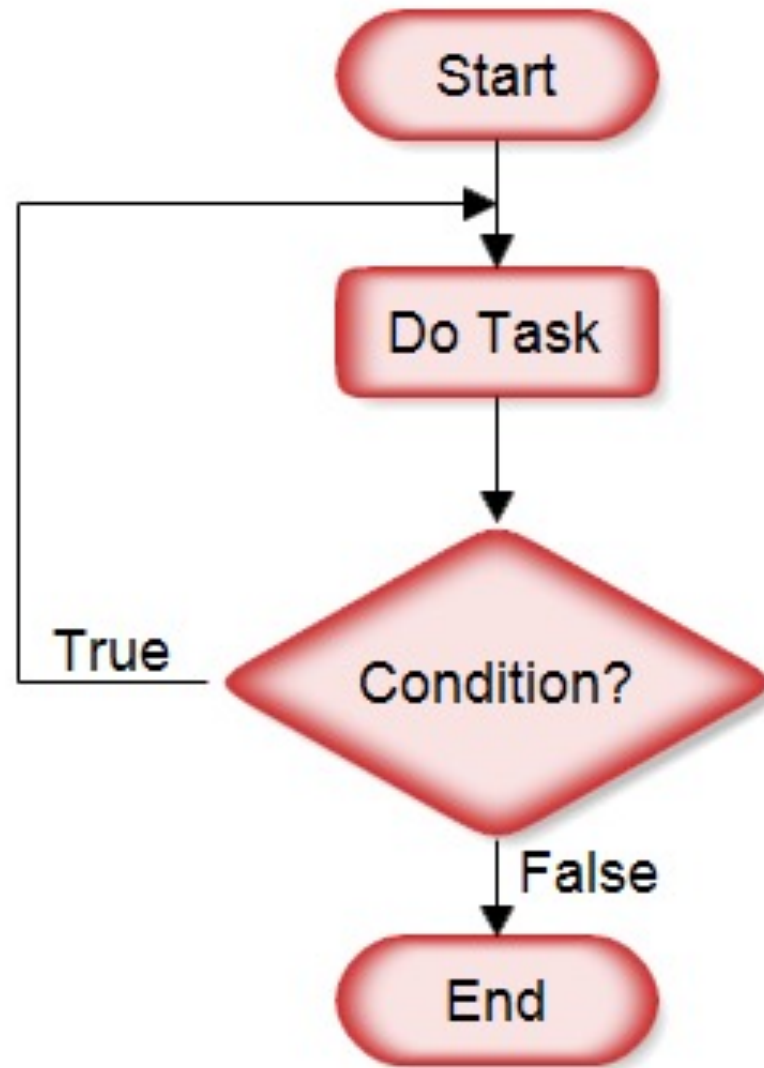
```
    }
```

```
?>
```

# Do-While

- Fungsi dari perulangan *do-while* hampir sama dengan perulangan *while*, perbedaan antara keduanya terletak pada urutan prosesnya saja
- Pemeriksaan kondisi di akhir

# Flowchart



# Contoh Do-While

```
<?PHP
```

```
    $d = 1;
```

```
    do {
```

```
        echo "Perulangan ke-$d <br>";
```

```
        $d++;
```

```
    } while ($d <= 5) ;
```

```
?>
```

# Foreach

- Pengulangan pada saat mengakses elemen-elemen dalam suatu array
- Bentuk umum sintaks:

```
foreach ($umur as $data) {  
    //statement  
}
```

# Contoh Foreach

```
<?php
$array = array( 1, 2, 3, 4, 5);
foreach($array as $value) {
    echo "Value is $value <br />"; }

?>
```

# Foreach

- Jika array yang akan diakses berupa array asosiatif dan ingin mengakses *key* dan *value* dari masing-masing element maka sintaks foreach seperti berikut:

```
foreach ($data as $kunciEl => $nilaiEl) {  
    //statement  
}
```

# Statement Kontrol

## Break

- Menghentikan proses pengulangan untuk berhenti, meskipun kondisi yang di definisikan belum bernilai *false*



# Contoh Break

```
<?php
    for ($i=0; $i<10; $i++) {
        if ($i==6) break;
        echo $i." ";
    }
?>
```

# Statement Kontrol

## Continue

- Digunakan untuk memaksa pengulangan agar melanjutkan proses ke indeks berikutnya
- Statemen ini mengabaikan semua statemen yang ditulis di bawah *continue*

# Contoh Continue

```
<?php
    for ($i=0; $i<10; $i++) {
        if ($i % 2==0) continue;
        echo $i." ";
    }
?>
```

# Statement Kontrol

## Exit

- Digunakan untuk menghentikan eksekusi kode di dalam suatu file/skrip tertentu
- Semua statemen yang ditulis di bawah *exit* tidak akan di eksekusi

# Contoh Exit

```
<?php
    for ($i=0; $i<10; $i++) {
        if ($i==6) exit();
        echo $i." ";
    }
    echo "Perulangan selesai";
?>
```

# Statement Kontrol

## Return

- Digunakan untuk mengembalikan eksekusi program dari dalam fungsi ke baris pemanggil
- Sering digunakan dalam pendefinisian fungsi

# Contoh Return

```
<?php
    function kali ($a, $b) {
        return $a*$b;
    }
    $hasil = kali ($a, $b) ;
?>
```

# Array



# Array

- Suatu koleksi nilai yang saling berkaitan
- Nilai yang ditampung di dalamnya bisa banyak
- Tersusun atas elemen-elemen
- Setiap elemen terdiri dari kunci (*key*) dan nilai (*value*)
- Tanda panah ( $\Rightarrow$ ) untuk memberikan nilai kepada *key*

# Deklarasi Array

```
$name = array();           # create
$name = array(value0, value1, ..., valueN);

$name[index]              # get element value
$name[index] = value;     # set element value
$name[] = value;          # append
```

PHP

```
$a = array();              # empty array (length 0)
$a[0] = 23;                # stores 23 at index 0 (length 1)
$a2 = array("some", "strings", "in", "an", "array");
$a2[] = "Ooh!";            # add string to end (at index 5)
```

PHP

# Jenis Array

- Array Berindeks (*indexed array*)
- Array Asosiatif (*associative array*)

# Array Berindeks (*indexed array*)

- Array yang diindeks menggunakan angka maupun karakter yang sifatnya berurutan
- Indeks array dalam PHP dimulai dari 0, bukan 1
- Ukuran array dihitung berdasarkan jumlah elemen yang ada
- `Sizeof()`

98	87	655	44	6
\$A[0]	\$A[1]	\$A[2]	\$A[3]	\$A[4]

# Array Berindeks (*indexed array*)

```
$mobil = array("Volvo", "BMW", "Toyota");
```

```
$mobil[0] = "Volvo";
```

```
$mobil[1] = "BMW";
```

```
$mobil[2] = "Toyota";
```

# Array Asosiatif (*associative array*)

- Array yang diindeks berdasarkan nama tertentu
- Perbedaan dengan array berindeks hanya terletak pada penamaan indeksnya saja

98	87	655	44	6
\$A["no"]	\$A["satu"]	\$A["dua"]	\$A["tiga"]	\$A["empat"]

# Array Asosiatif (*associative array*)

```
$umur =array("Jorge"=>"29","Jennifer"=>"28");
```

```
$umur['MangJorge'] = "29";
```

```
$umur['Jennifer'] = "28";
```

# Mengakses Array

- Cara pengaksesan sederhana
- Menempatkan nilai yang diakses ke dalam suatu variable atau dengan langsung diproses dalam perhitungan maupun langsung ditampilkan
- Contoh:

```
$asal = $A[0];
```

```
$dirut = $pegawai["dirut"];
```

```
echo "Nama Dirut ".$pegawai["dirut"];
```



# Contoh

```
<html>
<head>
    <title>Akses
    Array</title>
</head>
<body>
<?php
$pegawai = array(
    "dirut" => "Arya",
    "wadir" => "Komang",
    "manager" => "Ketut");
```

```
echo "Pegawai
Perusahaan<br/>";
echo "<b>Dirut:</b>
".$pegawai["dirut"]."<br/>";
echo "<b>Wadir:</b>
".$pegawai["wadir"]."<br/>";
echo "<b>Manager:</b>
".$pegawai["manager"]."<br/>";
;
?>
</body>
</html>
```

# Akses array menggunakan perulangan

- Akses array tidak efisien jika akses satu per satu
- Pengaksesan array dapat dilakukan dengan menggunakan struktur pengulangan
- Dapat menggunakan semua pengulangan (for, while, do-while, foreach)
- Tapi untuk jenis asosiatif biasanya menggunakan foreach

# Contoh

```
<html>
<head>
    <title>Akses Array</title>
</head>
<body>
<?php
$pegawai = array(
    "dirut" => "Arya",
    "wadir" => "Komang",
    "manager" => "Ketut");
$kelas= array("A","BB","BH");
For($i=0;$i<sizeof($kelas);$i++){
    echo "Kelas ".$kelas[$i]."<br/>";
}
```

```
echo "Pegawai Perusahaan<br/>";
Foreach($pegawai as $jab => $nama){
    echo "<b>$jab:</b> ".$nama."<br/>";
}
?>
</body>
</html>
```

# Memodifikasi nilai array

- Perubahan nilai array sama dengan proses pengisian nilai ke dalam array
- Nilai yang diisi ulang maka nilai lamanya akan tertimpa

```
$A[0]=10; $A[]=23;  
echo $A[0]." ".$A[1];  
$A[1]=200;  
echo $A[0]." ".$A[1];
```

# Mengurutkan Array

- PHP menyediakan beberapa fungsi untuk melakukan pengurutan terhadap element array
- Fungsi mengurutkan:
  - `sort()`
  - `asort()`
  - `ksort()`
  - `rsort()`
  - `arsort()`
  - `krsort()`

# Contoh Sort

- Mengurutkan array berdasarkan nilai elemen

```
70      $a = array(20,10,40,15,5);  
71      echo "Sebelum diurut: <br/>";  
72      print_r($a);  
73      sort($a);  
74      echo "<br/><br/>Sesudah diurut: <br/>";  
75      print_r($a);
```

Output:

Sebelum diurut:

Array ( [0] => 20 [1] => 10 [2] => 40 [3] => 15 [4] => 5 )

Sesudah diurut:

Array ( [0] => 5 [1] => 10 [2] => 15 [3] => 20 [4] => 40 )

# Menggabungkan Array

- Dua buah array dapat digabungkan dengan menggunakan “+”, tetapi penggabungan tidak berlaku jika *kunci/key* sama dengan array pertama

```
$A=[1=>10, 2=>20, 3=>30];
```

```
$B=[1=>100, 2=>200, 3=>300, 4=>56, 5=>70];
```

```
$c = $A+$B;
```

```
print_r($c);
```

# Menggabungkan Array

- Jika ingin menggabungkan array tanpa memperhatikan kunci digunakan fungsi **array\_merge()**

```
$A=[1=>10, 2=>20, 3=>30];
```

```
$B=[1=>100, 2=>200, 3=>300, 4=>56, 5=>70];
```

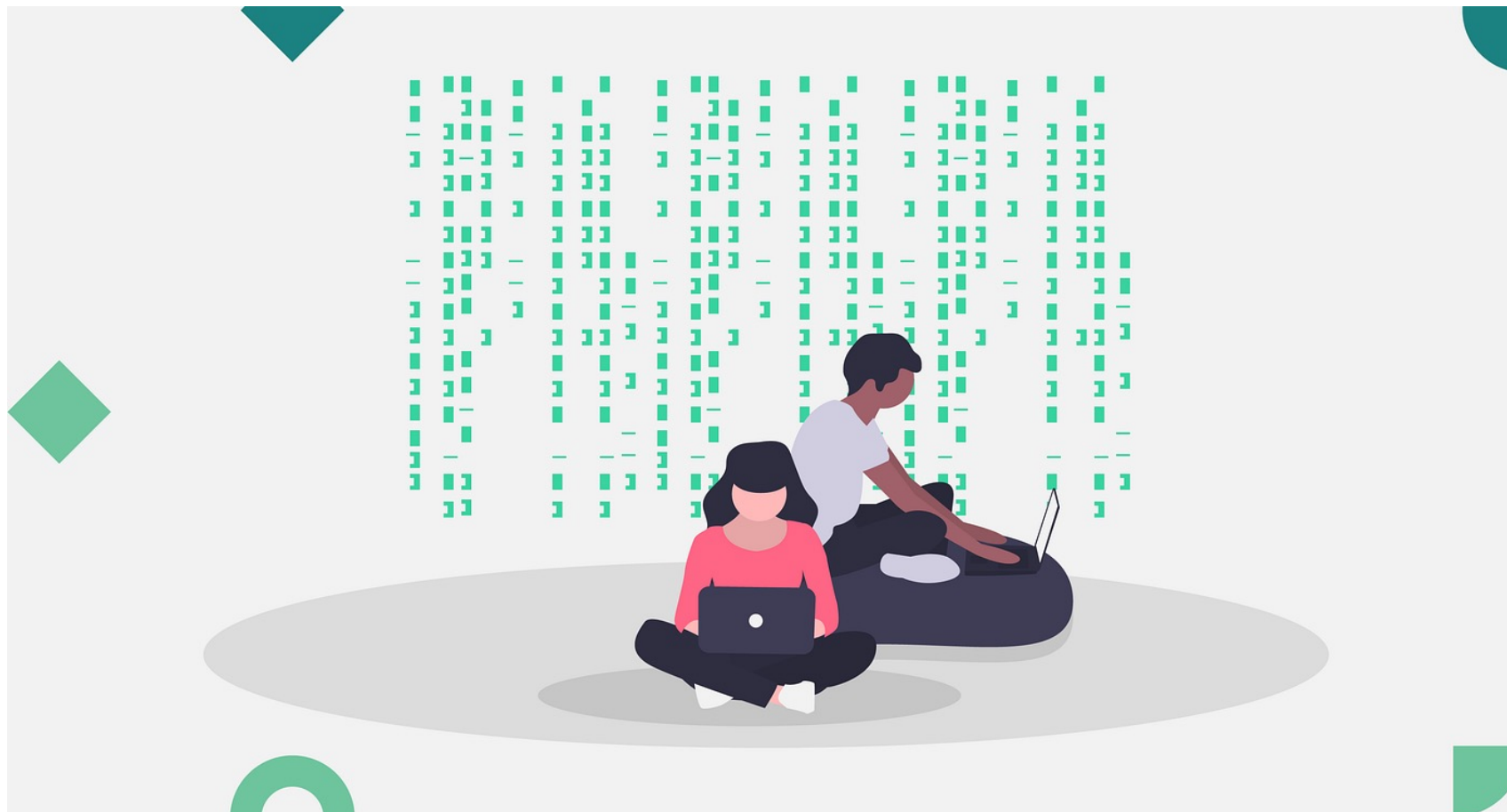
```
$c = array_merge($A, $B);
```

```
print_r($c);
```



# Fungsi Array lainnya

- `in_array()`
  - Untuk memeriksa apakah suatu nilai merupakan anggota array atau tidak
- `array_keys()`
  - Untuk mendapatkan daftar kunci yang terdapat pada suatu array
- `array_value()`
  - Untuk memperoleh daftar nilai dari suatu array
- `array_search()`
  - Mencari nilai indeks dari suatu array sesuai dengan nilai yang dicari
- `is_array()`
  - Memeriksa apakah suatu variable merupakan array atau bukan



# Latihan

a) 1 6 11 16

b) 10 9 8 7 6 5 4 3 2 1 0

c) 30 27 24 21 18 15 12 9 6 3 0

d) 10 8 6 4 2 0

- SOAL a & b KERJAKAN DENGAN WHILE
- SOAL c & d KERJAKAN DENGAN DO-WHILE

# Latihan! Kerjakan dengan Nested For

a) 1 2 3 4 5 6  
1 2 3 4 5  
1 2 3 4  
1 2 3  
1 2  
1

b) 6  
6 5  
6 5 4  
6 5 4 3  
6 5 4 3 2  
6 5 4 3 2 1

# Soal Tugas Perulangan

1) 1  
2 2  
3 3 3  
4 4 4 4  
5 5 5 5 5  
6 6 6 6 6 6

2) 1 2 3 4 5 6  
2 3 4 5 6  
3 4 5 6  
4 5 6  
5 6  
6

# Soal Tugas Perulangan

- 3) Buatlah program menggunakan pernyataan perulangan untuk menampilkan bilangan bulat 5 s/d 75 yang berkelipatan 5.
- 4) Buatlah program menggunakan pernyataan perulangan untuk mencari jumlah dan nilai rata-rata bilangan 6 s/d 80 yang merupakan bilangan ganjil.
- 5) Buatlah program menggunakan pernyataan perulangan untuk mencari banyaknya bilangan bulat mulai dari 11 s/d 127 yang merupakan kelipatan 6 atau kelipatan 10.

# Soal Tugas Array

1. Temperature suhu : 78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 75, 76, 73, 68, 62, 73, 72, 65, 74, 62, 62, 65, 64, 68, 73, 75, 79, 73

*Output :*

Rata-rata Temperature : 70.6

List 5 temperatures terendah: 60, 62, 63, 63, 64,

List 5 temperatures tertinggi: 76, 78, 79, 81, 85

# Soal Tugas Array

2. Berikut diberikan data array: `$color = array('white', 'green', 'red')`  
Buat code PHP sehingga menghasilkan output berikut

Output :

white, green, red,

- Green
- Red
- White



# Soal Tugas Array

3. Buatlah code dengan menggunakan array sehingga menghasilkan output seperti pada gambar di samping.

Gunakan asosiatif array dengan kunci/key adalah color name

## Color Table

Color Name	Hex Code
BlanchedAlmond	#ffeacd
CadetBlue	#5f9ea0
BurlyWood	#deb887
DarkOliveGreen	#556b2f
HotPink	#ff69b4
Papayawhip	#ffefd5

# Soal Tugas Array

- Gunakan array 2 dimensi untuk menampung data absensi mahasiswa. Data dimulai dari nim, nama, pertemuan ke-n dan total kehadiran. Dari array kemudian buatlah menjadi tampilan dalam bentuk table dengan border 1, spasi cell 10, spasi antar cell 5 dan header diberi warna.

Jl. Tukad Pakerisan, No.97, Denpasar - Bali, Telp. 0361-256995, Fax. 0361-246875, [www.stiki-indonesia.ac.id](http://www.stiki-indonesia.ac.id)