

BAB 8

BEKERJA DENGAN ARRAY

8.1 Pengertian Array

Array digunakan untuk mengidentifikasi variabel yang memiliki jumlah data yang lebih dari satu, namun mempunyai nama variabel yang sama. Untuk mengidentifikasi variabel \$hari yang mempunyai beberapa nilai, seperti : Minggu, Senin, Selasa, Rabu, Kamis, Jumat dan Sabtu, yaitu dengan cara memberi nilai indeks di belakang variabel. Dengan menggunakan angka indeks, array akan memberikan nilai sesuai dengan urutan dalam nilai array tersebut. Angka indeks secara otomatis akan dimulai dengan angka 0. Berikut ini adalah contoh penulisan dari suatu array :

```
$buah  
=array("minggu","senin","selasa","rabu","kamis","jumat","sabtu");
```

Contoh penulisan array apabila dideklarasikan berdasarkan nilai indeks adalah:

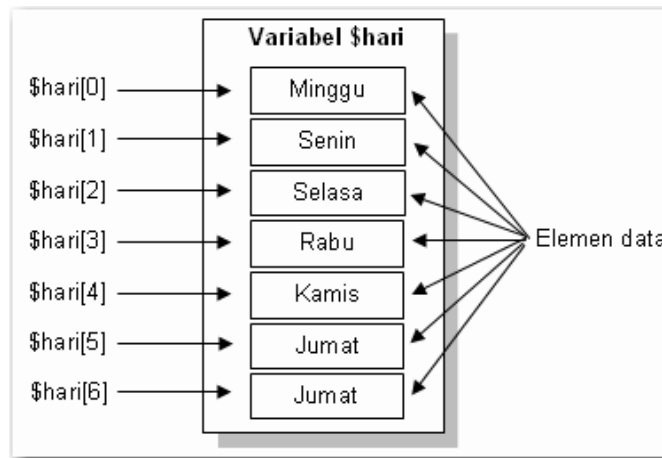
```
$hari[0] = "minggu";  
$hari[1] = "senin";  
$hari[2] = "selasa";  
$hari[3] = "rabu";
```

\$hari[4]="kamis";

\$hari[5]="jumat";

\$hari[6]="sabtu";

Apabila digambarkan ke dalam bentuk diagram dapat Anda lihat sebagai berikut:



Gambar 8.1 Diagram struktur data array

Angka indeks pada array di atas, secara otomatis dimulai dengan angka 0 dan angka berikutnya akan dituliskan secara berurutan hingga data array yang terakhir.

Fungsi-fungsi yang berkaitan dengan penentuan posisi pointer dalam suatu array yang dapat Anda gunakan antara lain :

Fungsi	Keterangan
Key()	Fungsi yang digunakan untuk mengetahui angka indeks pada posisi pointer aktif saat diakses.
Current()	Fungsi yang digunakan untuk mengetahui nilai array pada posisi pointer aktif.
Next()	Fungsi yang digunakan untuk menaikkan pointer ke data array berikutnya.
Prev()	Fungsi yang digunakan untuk kembali ke data array sebelumnya.

Fungsi	Keterangan
End()	Fungsi yang digunakan untuk meletakkan pointer pada posisi akhir array.
Reset()	Fungsi yang digunakan untuk meletakkan pointer pada posisi awal array.

8.2 Menciptakan dan Membaca Data Array

Dalam membuat array dalam program PHP, ada beberapa cara yang dapat Anda gunakan, yang pertama adalah memasukkan nilai dari array itu secara langsung. Perhatikan contoh berikut :

```
<?php
//array1.php
$hewan[0]="Kucing";
$hewan[1]="Anjing";
$hewan[2]="Ayam";
$hewan[3]="Kerbau";
echo "Data array<hr>";
for($a=0;$a<count($hewan);$a++){
    echo "Data $a adalah $hewan[$a].<br>";
}
?>
```

Adapun cara lain untuk mendeklarasikan sebuah array adalah dengan menggunakan fungsi **Array()**. Contoh penulisan dalam suatu program adalah:

```
<?
// array2.php
$nama_hari = array("Minggu", "Senin", "Selasa", "Rabu", "Kamis",
"Jumat", "Sabtu");
$hari_ini = $nama_hari[date("w")];
Echo "Sekarang Hari <b>$hari_ini</b>";
?>
```

8.3 Array Multidimensi

Array multi dimensi adalah array yang ada dalam array. Dalam array tersebut dapat berisi beberapa array lagi. Dengan adanya array multi dimensi ini dapat memudahkan dalam membuat program, karena dapat membuat beberapa array sekaligus sehingga meringkas beberapa perintah operasi. Berikut contoh penulisan array multi dimensi dalam sebuah perintah program. :

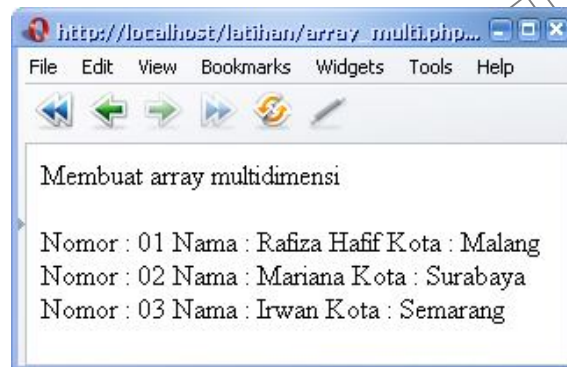
```
<?
//arraymulti.php
```

```

$data = @array(
    Array ( Nomor =>"01 ", Nama=>"Rafiza Hafif ", Kota =>
"Malang", ),
    Array (Nomor =>"02 ", Nama => "Mariana ", Kota =>
"Surabaya", ),
    Array (Nomor =>"03 ", Nama =>"Irwan ", Kota => "Semarang", )
);
Echo"Membuat array multidimensi<br><br>";
Foreach ($data as $val){
    Foreach($val as $key =>$data2)
    {
        Echo"$key : $data2";
    }
}
Echo"<br>";
}
?>

```

Hasilnya apabila dijalankan pada browser akan tampak seperti pada Gambar 8.2 berikut :



Gambar 8.2 Tampilan array yang telah dimanipulasi

Data yang ada pada sebuah array bersifat dinamis, artinya dapat ditambah, digabung dengan array yang lain, dihapus maupun diurutkan sesuai pola tertentu. Untuk memanipulasi data array, Anda dapat menggunakan beberapa fungsi yang akan digunakan dan dijelaskan pada bab ini.

8.4 Fungsi Pengolah Array

Fungsi yang khusus digunakan untuk mengolah string banyak sekali. Pada bagian ini akan membahas fungsi-fungsi tersebut dengan menggunakan

contoh script dalam mengaplikasikan fungsi tersebut. Berikut adalah fungsi-fungsi tersebut:

8.4.1 Menambah Data Array

Ada kalanya pada sebuah program memerlukan aplikasi penambahan data pada sebuah array. Untuk menambahkan data pada array dapat menggunakan fungsi berikut:

Array_push(data baru, data lama);

Keterangan :

Data baru adalah data yang ditambahkan pada variabel array.

Data lama adalah data yang sudah pada pada variabel array.

Contoh:

```
<?php
// array_push.php
// Menambah data array
$tambahan = array("Minggu", "Senin", "Selasa", "Rabu");
array_push($tambahan, "Kamis", "Jumat", "Sabtu");
print_r($tambahan);
?>
```

8.4.2 Menggabung Data Array

Selain dalam operasi penambahan, data array juga dapat digabung dengan array yang lain. Untuk menggabung array, Anda dapat menggunakan fungsi **Array_merge()**. Dengan menggunakan fungsi ini, Anda juga dapat menggabung beberapa array sekaligus. Bentuknya adalah :

Array_merge(array1, array2, array3);

Keterangan :

Array1, array2 dan **array3** adalah data array yang akan digabung menjadi satu array.

Cara penulisan fungsi untuk menggabung array, seperti yang tampak pada contoh berikut:

```
<?
$array1 = array("Jambu", "Jeruk", "Semangka");
$array2 = array("Merah", "Kuning", "Hijau");
$array3 = array("Manis", "Asam", "Segar");
```

```

$gabung = array_merge($array1, $array2,$array3);
Echo"<b>Penggabungan Array</b><br>";
Foreach($gabung as $val)
{
Echo"$val<br>";
}
?>

```

8.4.3 Mengurutkan Data Array

Data array tidak selalu dikondisikan dalam pola data yang berurutan, misalnya data tersusun sesuai urutan abjad A hingga Z. Apabila suatu kondisi meminta suatu data array harus dalam kondisi urut, maka program PHP memiliki fungsi untuk menyelesaikan pekerjaan ini dengan mudah. Bentuk syntaknya adalah sebagai berikut:

Sort(data array);

Berikut ini adalah contoh program untuk mengurutkan elemen array:

```

<?
//urut.php
$data = array("Merah", "Kuning", "Biru", "Hijau", "Abu abu");
echo"<b>Sebelum Diurutkan :</b>";
for($i=0;$i<=count($data);$i++){
echo"<br>$data[$i]";
}
sort($data);
echo"<b>Setelah Diurutkan :</b>";
for($i=0;$i<=count($data);$i++){
echo"<br>$data[$i]";
}
?>

```

Untuk mengurutkan data berdasarkan indexnya, Anda dapat menggunakan fungsi krsort dan ksort.

Krsort(data array);

Ksort(data array);

Keterangan :

Fungsi ksort digunakan untuk mengurutkan data berdasarkan abjad **a-z**, namun untuk fungsi krsort digunakan untuk pengurutan dari **z ke a**.

Contoh:

```

<?php
$buah = array("d"=>"jeruk", "a"=>"mangga", "b"=>"pepaya",
"c"=>"apel");
echo"Pengurutan data menggunakan krsort : <br>";

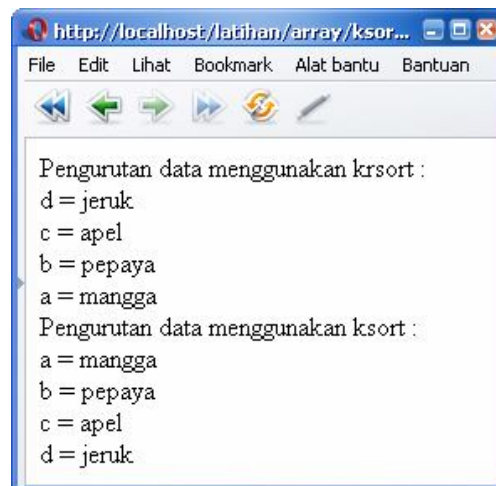
```

```

krsort($buah);
reset($buah);
while (list($key, $val) = each($buah)) {
    echo "$key = $val<br>";
}
echo "Pengurutan data menggunakan krsort : <br>";
ksort($buah);
reset($buah);
while (list($key, $val) = each($buah)) {
    echo "$key = $val<br>";
}
?>

```

Hasilnya apabila dijalankan pada browser akan tampak seperti pada Gambar 8.3 berikut:



Gambar 8.3 Tampilan pengurutan data array berdasarkan indexnya

8.4.4 Menghapus Data Array

Selain dalam operasi penambahan dan penggabungan, data array juga dapat dihapus. Untuk menghapus data array pertama, Anda dapat menggunakan fungsi **Array_shift()** bentuknya syntaksnya sebagai berikut:

Array_shift(data array);

Contoh penggunaan dari penulisan fungsi **Array_shift()** dalam suatu program adalah :

```

<?
// hapus_array.php
$hapus_array = array("Arman", "Firman", "Atika", "Farida");
while(count($hapus_array))
{
$val = array_shift($hapus_array);
echo"$val<br>";
echo"Data Array ada : ".count($hapus_array)."pada
\\$hapus_array<br>";
}
?>

```

8.4.5 Membandingkan Data Antar Array

Pada program PHP juga dapat membandingkan data antar array, hasilnya adalah data array yang memiliki kesamaan data saja. Berikut adalah beberapa fungsi yang dapat digunakan:

Array_intersect(array1, array2);

Keterangan :

Array1 dan **array2** adalah variabel dengan data array yang dibandingkan datanya.

Berikut adalah contoh penggunaan fungsi `array_intersect` pada sebuah script program:

```

<?php
//array_intersect.php
echo"Membandingkan data array dengan
<b>Array_intersect</b><br>";
$array1 = array("merah", "kuning", "biru");
$array2 = array("merah", "coklat", "kuning");
$hasil = array_intersect($array1, $array2);
for($a=0;$a<count($hasil);$a++){
echo "Hasil $a : $hasil[$a]<br>";
}
?>

```

8.4.6 Memotong Data Array

Untuk mengambil potongan dari sebuah data array dapat menggunakan fungsi `array_slice()`. Dengan menggunakan fungsi ini, Anda dapat mengambil potongan atau sebagian teks dari sebuah data array pada posisi manapun, baik yang berada di awal data array, di tengah data array maupun

di akhir data array. Selain itu, Anda juga dapat menentukan jumlah potongan yang akan diambil. Bentuk sintaks dari fungsi `array_slice()` ini adalah:

`Array_slice(data array, awal pengambilan, posisi pengambilan);`

Keterangan :

Data array adalah data array awal sebagai data sumber.

Awal pengambilan adalah bernilai angka baik bernilai positif maupun negatif yang menunjukkan posisi dari awal pengambilan data array. Apabila angka bernilai positif maka pengambilan dimulai dari kanan dan diawali dengan angka **0**. Apabila angka bernilai negatif, maka pengambilan karakter akan dimulai dari kiri dan dihitung mulai dari angka **1**.

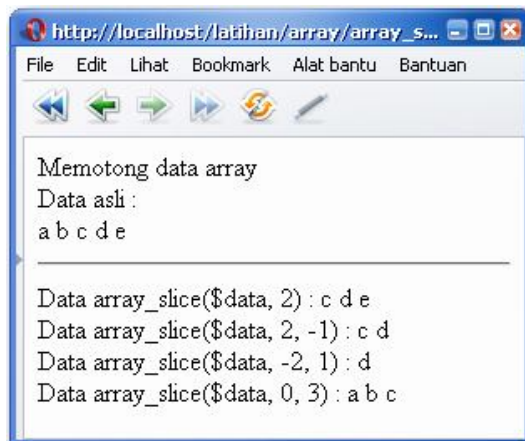
Posisi pengambilan adalah jumlah data array yang akan diambil dalam string. Posisi_pengambilan ini bernilai angka, baik bernilai positif maupun bernilai negatif.

Contoh dalam sebuah script adalah sebagai berikut:

```
<?php
//array_slice.php
// memotong data array
echo "Memotong data array<br>";
$data = array("a", "b", "c", "d", "e");
echo "Data asli :<br>";
for($a=0;$a<count($data);$a++){
    echo "$data[$a] ";
}
echo "<hr>";
echo "Data array_slice(\$data, 2) : ";
$hasil = array_slice($data, 2);
for($a=0;$a<count($hasil);$a++){
    echo "$hasil[$a] ";
}
echo "<br>Data array_slice(\$data, 2, -1) : ";
$hasil = array_slice($data, 2, -1);
for($a=0;$a<count($hasil);$a++){
    echo "$hasil[$a] ";
}
echo "<br>Data array_slice(\$data, -2, 1) : ";
$hasil = array_slice($data, -2, 1);
for($a=0;$a<count($hasil);$a++){
    echo "$hasil[$a] ";
}
echo "<br>Data array_slice(\$data, 0, 3) : ";
$hasil = array_slice($data, 0, 3);
for($a=0;$a<count($hasil);$a++){
    echo "$hasil[$a] ";
}
```

```
}  
?>
```

Hasilnya apabila dijalankan pada browser akan tampak seperti pada Gambar 8.5 berikut:



Gambar 8.5 Tampilan data array yang potong

8.4.7 Menjumlahkan Data Angka Array

Data array yang berjenis angka dapat dijumlahkan menggunakan fungsi **array_sum()**. Bentuknya sintaksnya adalah:

Array_sum(data array);

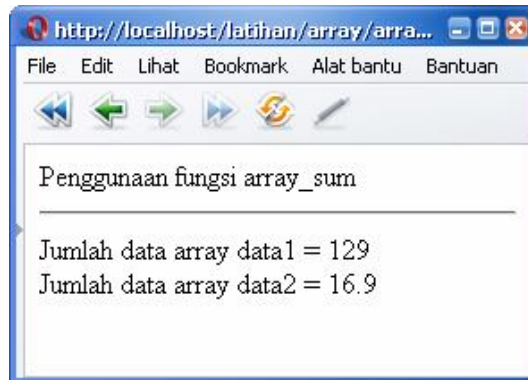
Keterangan :

Data array adalah data array berjenis angka yang akan dijumlahkan hasilnya.

Berikut adalah contoh penggunaan fungsi **array_sum** tersebut:

```
<?  
// array_sum.php  
echo "Penggunaan fungsi array_sum<br>";  
$data1 = array(34, 21, 16, 58);  
echo "Jumlah data array data1 = ".array_sum($data1)."<br>";  
$data2 = array("a" => 4.7, "b" => 7.3, "c" => 4.9);  
echo "Jumlah data array data2 = ".array_sum($data2)."<br>";  
?>
```

Hasilnya apabila dijalankan pada browser akan tampak seperti pada Gambar 8.6 berikut:



Gambar 8.6 Tampilan data array yang potong

8.4.8 Mendeklarasikan data array ke dalam variabel

Data array dapat dideklarasikan dalam variabel-variabel tertentu yang diciptakan sendiri. Perintah ini menggunakan fungsi list. Bentuknya syntaksnya adalah sebagai berikut:

List(variabel1, variabel2, variabel2) = \$array;

Keterangan :

Variabel1, 2 dan 3 adalah variabel yang diciptakan sendiri.

\$array adalah dat array yang diambil datanya dan dimasukkan pada variabel baru.

Berikut adalah contoh dalam script:

```
<?php
//Yist.php
$info = array('Rafiza Hafif', '28 tahun', 'web
programming');
echo "Penggunaan fungsi List<hr>";
// Listing all the variables
list($nama, $umur, $skill) = $info;
echo "Nama : $nama <br> Umur : $umur <br>Keahlian :
$skill.";
?>
```

Hasilnya akan tampak seperti gambar 8.7 berikut:



Gambar 8.7 Tampilan data array dengan fungsi List