Array

Definisi Array

- Digunakan untuk menyimpan koleksi data dalam suatu nama variable dan diakses menggunakan indeksnya.

Numeric Array

- Numeric array pada PHP bermula dari indeks 0. Dan indeks berikutnya tambah 1.
- Indeks paling terakhir adalah panjang array – 1.

Inisialisasi

Initialization via multiple statements:

```
$weekday[] = "Sunday";
$weekday[] = "Monday";
$weekday[] = "Tuesday";
```

Sama saja dengan:

```
$weekday[0] = "Sunday";
$weekday[1] = "Monday";
$weekday[2] = "Tuesday";
```

Inisialisasi

Array Initialization via a single statement:

```
$weekday = array(
   "Sunday", "Monday", "Tuesday", "Wednesday",
   "Thursday", "Friday", "Saturday" );
```

Sama saja dengan:

```
$weekday[0] = "Sunday";
$weekday[1] = "Monday";
$weekday[2] = "Tuesday";
...
```

Contoh Looping Array - for

```
<?php
    $color = array( "red",
                    "green",
                    "blue" );

    // obtain array length
    $size = count( $color );

    for ( $i = 0; $i < $size; $i++ )
    {
        printf( "\$color[%d]: %s<br />",
            $i, $color[$i] );
    }
?>
```

Looping Array - foreach

- Looping foreach khusus dirancang untuk array yang berfungsi sebagai iterator yang akan menjejaki setiap unsur pada array satu per satu.
- Penggunaan:
  - Ketika setiap unsur pada array perlu diakses.
  - Ketika nilai setiap unsur diperlukan.
  - Ketika indeks tidak diperlukan.

Syntax Looping Array - foreach

```
foreach ( $array as $value )
{
    statement;
    ...
}
```

Contoh Looping Array - foreach

```
<?php
    $colors = array( "red",
                     "green",
                     "blue" );

    foreach ( $colors as $name )
        echo "$name<br />";
?>
```

Latihan

- Buat program PHP yang akan mencetak nilai ROT13 (http://en.wikipedia.org/wiki/ROT13) untuk setiap alfabet dari a – z.
- Output kira-kira seperti ini:

  a = n

  b = o

  …

  z = m
- Tips:
  - Pakai fungsi range() untuk inisialisasi array
  - Pakai fungsi ord(string $s) untuk mengembalikan nilai ASCII dari karakter
  - Pakai fungsi chr(int $nilaiascii) untuk mengembalikai karakter dari suatu nilai ASCII

Pengiriman Array ke Fungsi

```
<?php
   function printList( $list )
   {
      foreach( $list as $item )
      {
         printf( "%s<br />", $item );
      }
   }

   $groceries = array( "blueberries",
                       "strawberries",
                       "bananas" );

   printList( $groceries );
?>
```

Associative Array (or Hash)

Associative Array (or Hash)

- Associative array juga disebut hash.
- Hash menggunakan suatu string unique yang disebut “key” or “kunci” sebagai indeks untuk setiap unsur array.
- Syntax:

```
$var['key'] = $value;
```

Inisialisasi

- Via multiple statements

```
<?php
    $month['jan'] = 31;
    $month['feb'] = 28;
    $month['mar'] = 31;
    $month['apr'] = 30;
    ...

    foreach ( $month as $name => $days )
    {
        printf( "%s has %d days<br />",
                    $name, $days );
    }
?>
```

Urutan associative array tidak pasti

Inisialisasi

- Via single statement

```
<?php
   $user = array( 'mluther' => 'Martin',
                  'bgraham' => 'Billy',
                  'dlmoody' => 'D.L.',
                  'jwesley' => 'John' );

   foreach ( $user as $uname => $fname )
   {
      printf( "Username: %s  
               First: %s<br />",
                 $uname, $fname );
   }
?>
```

Looping Hash - foreach

- Looping foreach juga dapat digunakan untuk hash untuk mendapatkan key dan value untuk setiap elemen.
- Penggunaan:
  - Ketika unsur key dan value diperlukan.
  - Ketika operasi berlaku untuk setiap elemen pada hash.

Syntax Looping Hash - foreach

```
foreach ( $array as
                $key => $value )
{
    statement;
    ...
}
```

Contoh Looping Hash - foreach

```
<?php
    $color =
        array( "red" => "#FF0000",
               "green" => "#00FF00",
               "blue" => "#0000FF" );

    foreach ( $color as $key => $val )
    {
        printf( "\$color['%s']:
                  %s<br />", $key, $val );
    }
?>
```

Latihan

- Buat program PHP yang membuat hash bagi kode pos sebagai key dan nama kota sebagai value.
- Contoh:
  - 23116 => Lampineung, 23115 => Prada, dll

Variable sebagai Hash Key

```
<?php
   $year = 1960;  // leap year
   $thirty = array( 'sep', 'apr', 'jun', 'nov' );

   $thirtyOne = array( 'jan', 'mar', 'may', 'jul',
                       'aug', 'oct', 'dec' );

   foreach( $thirty as $name )
      $month[$name] = 30;  // initialize 30 day months

   foreach( $thirtyOne as $name )
      $month[$name] = 31; // initialize 31 day months

   // $month['feb'] = isLeapYear( $year ) ? 29 : 28;
   $month['feb'] = 29;  // initialize leap year month

   // how would you print out months in order?
   foreach ( $month as $name => $days )
   {
      printf( "%s has %d days<br />",
                  $name, $days );
   }
?>
```

Hash untuk Unique List

```
<?php
   $cartoon =
      array( 'Fred', 'Barney', 'Fred', 'Wilma',
             'Fred', 'Pebbles', 'Fred', 'Dino' );

   foreach ( $cartoon as $name )
      $unique[$name]++;  // count occurrences

   foreach ( $unique as $name => $occur )
      printf( "Actor %s occurred %d times<br />",
                  $name, $occur );
?>
```

Looping Hash – while each

```
<?php
   $color = array( "red" => "#FF0000",
                   "green" => "#00FF00",
                   "blue" => "#0000FF" );

   while (list($key, $val) = each($color))
   {
      printf( "\$color['%s']: %s<br />",
         $key, $val );
   }
?>
```

Multidimensi Array dan Hash

Multidimensi Array

```
<pre>
<?php
    for ( $i = 0; $i < 3; $i++ )
    {
        for ( $j = 0; $j < 3; $j++ )
        {
            // multiplication table
            $grid[$i][$j] = $i * $j;
        }
    }

    print_r( $grid );
?>
</pre>
```

Multidimensi Hash

```
<pre>
<?php
   $pop['IA']['Independence'] = 6101;
   $pop['IA']['Manchester'] = 4898;

   $pop['KS']['Independence'] = 9277;
   $pop['KS']['Manchester'] = 100;

   $pop['MO']['Independence'] = 110704;
   $pop['MO']['Manchester'] = 18657;

   printf( "print_r( \$pop );<br />" );
   print_r( $pop );

   printf( "<hr />" );
   printf( "var_dump( \$pop );<br />" );
   var_dump( $pop );

   printf( "<hr />" );
   printf( "var_export( \$pop );<br />" );
   var_export( $pop );
?>
</pre>
```

Array of Arrays

```
<pre>
<?php
   $grid =
      array( 0, array( 0 => 0 * 0,
                       1 => 0 * 1,
                       2 => 0 * 2 ),
                1, array( 0 => 1 * 0,
                       1 => 1 * 1,
                       2 => 1 * 2 ),
                2, array( 0 => 2 * 0,
                       1 => 2 * 1,
                       2 => 2 * 2 ) );
   print_r( $grid );
?>
</pre>
```

Hash of Arrays

```
<?php
   $bedrock = array(
      'Flintstone' =>
         array( 'Fred', 'Wilma',
                'Pebbles', 'Dino' ),
      'Rubble' =>
         array( 'Barney', 'Betty',
                'Bambam' ) );

   foreach ( $bedrock as $last => $family )
   {
      echo "\$last: $last<br />";

      foreach ( $family as $first )
      {
         echo "     
               \$first: $first<br />";
      }
   }
?>
```

Hash of Hashes

```
<?php
   // array can also be defined via multiple statements above
   $popByZip =
      array( 'IA' =>
         array( 'Ames' =>
                  array( 50010 => 24991,
                         50014 => 29541 ),
                'Dubuque' =>
                  array( 52001 => 44033,
                         52002 => 11539,
                         52003 => 13305 ),
                'Waterloo' =>
                  array( 50701 => 29890,
                         50702 => 19299,
                         50703 => 20978,
                         50707 => 8155 ) ) );
   PrintArray( $popByZip, 0 );
?>
```

Printing Function

Fungsi print_r

```
<pre>
<?php

   $pop['IA']['Independe
   nce'] = 6101;

   $pop['IA']['Mancheste
   r'] = 4898;


    printf( "print_r(
   \$pop );<br />" );
    print_r( $pop );
?>
</pre>
```

```
print_r( $pop );
Array ( [IA] => Array (
   [Independence] => 6101
   [Manchester] => 4898 ) )
```

Fungsi var_dump

```
<pre>
<?php

   $pop['IA']['Independence
   '] = 6101;

   $pop['IA']['Manchester']
   = 4898;

   printf( "var_dump(
   \$pop );<br />" );
   var_dump( $pop );
?>
</pre>
```

```
var_dump( $pop );
   array(1) { ["IA"]=>
   array(2) {
   ["Independence"]=>
   int(6101)
   ["Manchester"]=>
   int(4898) } }
```

Fungsi var_export

```
<pre>
<?php

   $pop['IA']['Independence
   '] = 6101;

   $pop['IA']['Manchester']
   = 4898;

   printf( "var_export(
   \$pop );<br />" );
   var_export( $pop );
?>
</pre>
```

```
var_export( $pop );
array ( 'IA' => array (
   'Independence' => 6101,
   'Manchester' => 4898, ),
   )
```

Fungsi-fungsi untuk Array

Fungsi array_push

Deskripsi: tambah satu atau lebih elemen pada ujung array

Syntax:

```
array_push( &$array, $elements );
```

Contoh:

```
<?php
   $obj = array( "rock" );

   array_push( $obj, "paper",
                     "scissors" );

   for ( $i = 0; $i < count($obj); $i++ )
      printf( "\$obj[%d]: %s<br />",
              $i, $obj[$i] );
?>
```

Fungsi array_pop

Deskripsi: menghapus elemen terakhir pada array dan mengembalikan nilainya.

Syntax:

```
$last = array_pop( &$array );
```

Contoh:

```
<?php
   $stooge = array( "Moe", "Larry",
                    "Curly" );

   $best = array_pop( $stooge );

   for ( $i = 0; $i < count($stooge); $i++ )
      printf( "\$stooge[%d]: %s<br />",
              $i, $stooge[$i] );

   printf( "<p />" );
   printf( "\$best: %s", $best );
?>
```

Fungsi array_unshift

Deskripsi: tambah satu atau lebih elemen pada awal array.

Syntax:

```
array_unshift( &$array, $elements );
```

Contoh:

```
<?php
   $stooge = array( "Moe", "Larry" );

   array_unshift( $stooge, "Shemp" );

   for ( $i = 0; $i < count($stooge); $i++ )
      printf( "\$stooge[%d]: %s<br />",
                  $i, $stooge[$i] );
?>
```

Fungsi array_shift

Deskripsi: hapus elemen pertama pada array dan mengembalikan nilainya.

Syntax:

```
$first = array_shift( &$array );
```

Contoh:

```
<?php
   $narnia = array( "Peter", "Susan", "Lucy",
                    "Edmund" );

   $highKing = array_shift( $narnia );

   for ( $i = 0; $i < count($narnia); $i++ )
      printf( "\$narnia[%d]: %s<br />",
              $i, $narnia[$i] );

   printf( "\$highKing: %s", $highKing );
?>
```

Fungsi shuffle

Deskripsi: susun anggota array dengan urutan acak.

Syntax:

```
shuffle( &$array );
```

Contoh:

```
<?php
   $deck = array( "Ace", "King", "Queen", "Jack",
                  "Joker" );

   shuffle( $deck );  // randomize array

   for ( $i = 0; $i < count($deck); $i++ )
      printf( "\$deck[%d]: %s<br />",
              $i, $deck[$i] );
?>
```

Fungsi sort

Deskripsi: urut elemen array dalam urutan abjad.

Syntax:

```
sort( &$array );
```

Contoh:

```
<?php
   $torah = array( "Genesis", "Exodus",
                   "Leviticus", "Numbers",
                   "Deuteronomy" );

   sort( $torah );  // sort array by value

   for ( $i = 0; $i < count($torah); $i++ )
      printf( "\$torah[%d]: %s<br />",
              $i, $torah[$i] );
?>
```

Fungsi array_reverse

Deskripsi: reverse/balikkan urutan elemen-elemen pada array.

Syntax:

```
$arrayNew = array_reverse( $arrayOld );
```

Contoh:

```
<?php
   $name = "rumpelstiltskin";

   // convert string to array
   $pieces = str_split( $name );

   // reverse the array order
   $reverse = array_reverse( $pieces );

   // convert array back to string
   $backward = implode( "", $reverse );

   echo $backward;
?>
```

Fungsi array_splice

Contoh:

```
<?php
   $deck = array( "Ace", "King",
                  "Queen", "Jack",
                  "Joker" );

   $max = count($deck) - 1;
   $offset = rand( 0, $max );

   // draw one random card
   $drawn = array_splice( $deck, $offset, 1 );

   printf( "Card drawn: %s<br />", $drawn[0] );
   printf( "Remaining deck:<br />" );

   for ( $i = 0; $i < count($deck); $i++ )
      printf( "\$deck[%d]: %s<br />",
              $i, $deck[$i] );
?>
```

Deskripsi: hapus sebahagian dari array dan diganti dengan sesuatu yang lain.

Syntax:

```
$aryRm =
array_splice(
&$input, $offset[,
$length=0 [,
$replacement]] );
```

Fungsi array_walk

array_walk(): fungsi yang mengirimkan pasangan value/key dari associative array atau hash ke fungsi buatan sendiri (user defined function).

Syntax:

```
<?php
    function functName( $value, $key )
    {
        ...
    }

    array_walk( $hash, 'functName' );
?>
```

Contoh array_walk

```
<?php
   function printRow( $value, $key )
   {
      printf( "
         <tr>
            <td> $key </td>
            <td> $value </td>
         </tr>\n" );
   }

   $color = array( 'Red' => '#FF0000',
                   'Green' => '#00FF00',
                   'Blue' => '#0000FF',
                   'Yellow' => '#FFFF00' );

   array_walk( $color, 'printRow' );
?>
```

Latihan

- Buat fungsi untuk mengacak urutan karakter pada string.
- Input: satu string
- Output: original string dan string yang teracak
- Tips:
  - Gunakan str_split() untuk memecah string ke karakter.
  - Gunakan shuffle() untuk membuatnya menjadi acak.
  - Gunakan implode() untuk merekatkan karakter acak tadi menjadi string kembali.

Latihan

- Buat fungsi untuk mengacak urutan kata dalam kalimat.
- Input: satu kalimat
- Output: original kalimat dan kalimat yang teracak
- Tips:
  - Gunakan explode() untuk memecah kalimat menjadi string/kata-kata yang terpisah dalam array.
  - Gunakan shuffle() untuk membuatnya menjadi acak.
  - Gunakan implode() untuk merekatkan string acak tadi menjadi kalimat kembali.