

BAB XI

OPERASI STRING

strpos

Function strpos() digunakan untuk menentukan posisi suatu substring dari sederetan string. Function ini akan mereturn bilangan integer yang merupakan urutan posisi substring tersebut.

Contoh:

```
<?
$stringku = "1234567890";

$posisi = strpos($stringku, "5");
echo "Posisi dari karakter 5 dalam string adalah $posisi";
?>
```

Script di atas akan menghasilkan posisi karakter '5' adalah di 4.

NB:

Ingat bahwa index dari array dalam PHP dimulai dari index ke – 0.

Kelemahan dari strpos() adalah bahwa function tersebut hanya dapat menentukan posisi suatu substring pada pemunculan pertama. Pada pemunculan substring pada posisi yang berikutnya tidak akan terdeteksi. Meskipun demikian, Anda masih tetap menggunakannya untuk mencari posisi yang lain dari suatu substring dengan sedikit melakukan pemrograman.

Contoh:

```
<?
$stringku = "1234567890123456789012345678901234567890";
$offset = 0;
$countner = 1;

while($offset = strpos($numberedString, "5", $offset + 1)){
    $countner++;
    echo "<br>Karakter 5 ke-$countner ada posisi - $offset";
}
?>
```

str_replace

Function ini memiliki peran yang sama seperti ketika kita menggunakan fasilitas Replace All pada MS. Word. Function ini akan menggantikan suatu string dengan string yang lain.

Contoh:

```
$stringawal = "selamat datang di halaman web ini";
$ubahstring = str_replace("web", "website", $stringawal);
echo "$ubahstring";
```

Function `str_replace()` juga dapat menggantikan beberapa string secara simultan sekaligus. Berikut ini contohnya:

```
<?
$rawstring = "Welcome Birmingham parent! <br>
    Your offspring is a pleasure to have!
    We believe pronoun is learning a lot.<br>
    The faculty simple adores pronoun2 and you can often hear
    them say \"Attah sex!\"<br>";

$placeholders = array('offspring', 'pronoun', 'pronoun2', 'sex');
$malevals = array('son', 'he', 'him', 'boy');
$femalevals = array('daughter', 'she', 'her', 'girl');

$malestr = str_replace($placeholders, $malevals, $rawstring);
$femalestr = str_replace($placeholders, $femalevals, $rawstring);

echo "Son: ". $malestr . "<br>";
echo "Daughter: ". $femalestr;
?>
```

Pada script di atas, array `$placeholder` berisi string-string dari `$rawstring` yang akan diganti.

Dan tampilan script di atas adalah sbb:

```
Son: Welcome Birmingham parent!
Your son is a pleasure to have! We believe he is learning a lot.
The faculty simple adores he2 and you can often hear them say "Attah
boy!"

Daughter: Welcome Birmingham parent!
Your daughter is a pleasure to have! We believe she is learning a lot.
The faculty simple adores she2 and you can often hear them say "Attah
girl!"
```

Perhatikan string yang dicetak merah pada tampilan script tersebut. Seharusnya string `pronoun2` akan diganti dengan `him` dan `her`. Hal ini dikarenakan efek `str_replace` dari string `pronoun` yang merupakan substring dari `pronoun2`. Untuk menghindari efek kesalahan seperti itu, hindari peletakan substring di depan string lain dalam `str_replace`.

Berikut ini hasil modifikasinya:

```
<?
$rawstring = "Welcome Birmingham parent! <br>
    Your offspring is a pleasure to have!
    We believe pronoun is learning a lot.<br>
    The faculty simple adores pronoun2 and you can often hear
    them say \"Attah sex!\"<br>";

$placeholders = array('offspring', 'pronoun2', 'pronoun', 'sex');
$malevals = array('son', 'him', 'he', 'boy');
$femalevals = array('daughter', 'her', 'she', 'girl');
```

```
$malestr = str_replace($placeholders, $malevals, $rawstring);  
  
$femalestr = str_replace($placeholders, $femalevals, $rawstring);  
  
echo "Son: ". $malestr . "<br>";  
echo "Daughter: ". $femalestr;  
?>
```

dan tampilannya adalah

```
Son: Welcome Birmingham parent!  
Your son is a pleasure to have! We believe he is learning a lot.  
The faculty simple adores him and you can often hear them say "Attah  
boy!"
```

```
Daughter: Welcome Birmingham parent!  
Your daughter is a pleasure to have! We believe she is learning a lot.  
The faculty simple adores her and you can often hear them say "Attah  
girl!"
```

strtoupper

Function ini digunakan untuk mengubah semua karakter huruf dari suatu string menjadi kapital.

Contoh:

```
<?  
$originalString = "String Capitalization 1234";  
  
$upperCase = strtoupper($originalString);  
echo "Old string - $originalString <br>";  
echo "New String - $upperCase";  
?>
```

Hasilnya adalah

```
Old string - String Capitalization 1234  
New String - STRING CAPITALIZATION 1234
```

strtolower

Kebalikan dari strtoupper(), function ini mengubah semua karakter huruf dari string menjadi huruf kecil.

Contoh:

```
<?  
$originalString = "String Capitalization 1234";  
  
$lowerCase = strtolower($originalString);  
echo "Old string - $originalString <br>";
```

```
echo "New String - $lowerCase";  
?>
```

Hasilnya adalah

```
Old string - String Capitalization 1234  
New String - string capitalization 1234
```

ucwords

Karakter huruf pertama dari suatu kata dalam string juga dapat diubah menjadi huruf kapital menggunakan function ini.

Contoh:

```
<?  
$titleString = "a title that could use some hELP";  
  
$ucTitleString = ucwords($titleString);  
echo "Old title - $titleString <br>";  
echo "New title - $ucTitleString";  
?>
```

Hasilnya adalah:

```
Old title - a title that could use some hELP  
New title - A Title That Could Use Some HELP
```

Bagaimana cara mengubah HELP menjadi Help (pada tampilan outputnya)? Perhatikan script modifikasi berikut ini

```
<?  
$titleString = "a title that could use some hELP";  
  
$lowercaseTitle = strtolower($titleString);  
$ucTitleString = ucwords($lowercaseTitle);  
echo "Old title - $titleString <br />";  
echo "New title - $ucTitleString";  
?>
```

yaitu dengan cara mengubahnya ("hELP") ke huruf kecil semua terlebih dahulu ("help"), kemudian huruf pertama dari "help" diberikan perintah `ucwords()`.

explode

Sesuai namanya "explode", fungsi ini digunakan untuk meledakkan/memecah suatu string menjadi potongan-potongan string yang kecil. Selanjutnya potongan-potongan string ini akan disimpan dalam suatu array. Perhatikan contoh berikut ini

```
<?
$PhoneNumber = "800-555-5555";

$hasil = explode("-", $PhoneNumber);
echo " Phone Number = $PhoneNumber <br>";
echo "Pecahan 1 = $hasil[0]<br>";
echo "Pecahan 2 = $hasil[1]<br>";
echo "Pecahan 3 = $hasil[2]";
?>
```

Hasilnya adalah:

```
Phone Number = 800-555-5555
Pecahan 1 = 800
Pecahan 2 = 555
Pecahan 3 = 5555
```

Pada perintah `explode("-", $PhoneNumber);` di atas, karakter "-" dapat diibaratkan sebagai dinamitnya. Dinamit ini apabila diledakkan akan memecah string dari `$PhoneNumber`. Karakter yang akan digunakan sebagai dinamit dapat ditentukan sendiri oleh programmer, dapat berupa spasi, koma dsb.

Jumlah pecahan string dari hasil ledakan dapat dibatasi dengan menambahkan jumlah batas ledakan sebagai paramater ketiga dari function `explode()`.

Contoh:

```
<?
$PhoneNumber = "800-555-5555";

$hasil = explode("-", $PhoneNumber, 2);
echo " Phone Number = $PhoneNumber <br>";
echo "Pecahan 1 = $hasil[0]<br>";
echo "Pecahan 2 = $hasil[1]<br>";
echo "Pecahan 3 = $hasil[2]";
?>
```

Hasilnya adalah:

```
Phone Number = 800-555-5555
Pecahan 1 = 800
Pecahan 2 = 555-5555
Pecahan 3 =
```

Perintah `explode("-", $PhoneNumber, 2);` di atas membatasi 2 buah pecahan string dari hasil ledakan. Dengan demikian, tampak pada hasil bahwa pecahan ketiga tidak ada.

implode

Kebalikan dari `explode()`, function `implode()` digunakan untuk menyatukan pecahan-pecahan string menjadi satu kesatuan string.

Contoh:

```
<?
$pecahan = array("Hello", "World,", "I", "am", "Here!");

$disatukandenganspasi = implode(" ", $pecahan);
$disatukandengandash = implode("-", $pecahan);

echo "$disatukandenganspasi <br>";
echo "$disatukandengandash ";
?>
```

Hasilnya adalah:

```
Hello World, I am Here!
Hello-World,-I-am-Here!
```