

BAB 9

FUNGSI STRING

9.1 Memanipulasi Tanda Slash “\” Pada Teks

Banyak sekali fungsi string yang digunakan untuk mengolah tanda slash “\” pada karakter tertentu. Berikut adalah beberapa fungsi yang sering digunakan untuk membuat sebuah website:

9.1.1 Menambah Tanda Slash “\” Di depan Karakter Tertentu

Fungsi yang digunakan untuk menambah tanda “\” pada awal karakter tertentu adalah `addslashes()`. Bentuk syntaksnya adalah:

`Addslashes(string, parameter);`

Keterangan :

String adalah teks sumber yang dijadikan sebagai data sumber.

Parameter adalah menunjukkan teks yang akan diberi tanda “\” di depan karakter tersebut.

Contoh:

```
<?php
//addslashes1.php
```

Oleh : Yeni Cahyono- www.nicdesain.net 59

```

$data="Alamat rumah saya : Jl. Pahlawan No. 34 Surabaya
Timur.";
$hasil1=addslashes($data, 'a..z');
$hasil2=addslashes($data, 'A..Z');
$hasil3=addslashes($data, '0..9');
$hasil4=addslashes($data, 'a');
$hasil5=addslashes($data, 'P,S');
echo"Menggunakan parameter \"<b>a..z</b>\"
adalah:<br><b>$hasil1</b><br>";
echo"Menggunakan parameter \"<b>A..Z</b>\"
adalah:<br><b>$hasil2</b><br>";
echo"Menggunakan parameter \"<b>0..9</b>\"
adalah:<br><b>$hasil3</b><br>";
echo"Menggunakan parameter \"<b>a</b>\"
adalah:<br><b>$hasil4</b><br>";
echo"Menggunakan parameter \"<b>P,S</b>\"
adalah:<br><b>$hasil5</b><br>";
?>

```

Penjelasan script:

Yang perlu Anda perhatikan dalam fungsi **addslashes()** di atas adalah bagian parameternya. Pada fungsi addslashes yang pertama memiliki parameter **a..z** dimaksudkan akan memberikan tanda slash diawal semua karakter huruf kecil. Untuk parameter **A..Z** akan memberikan tanda slash diawal teks huruf besar atau katalog. Parameter **0..9** digunakan untuk memberikan tanda slash pada karakter angka saja, kemudian parameter a digunakan untuk memberikan tanda slash hanya pada huruf a saja. Apabila ingin memberikan tanda slash pada beberapa karakter yang berbeda, Anda dapat menuliskan karakter tersebut dengan dipisahkan tanda koma, misalkan P,S,R,e dan seterusnya. Hasilnya dapat Anda lihat pada gambar berikut:

```

http://localhost/latihan/string/addslashes1.php - Opera
File Edit View Bookmarks Widgets Tools Help
ing/addslashes1.php
Menggunakan parameter "a..z" adalah:
A\l\am\at\ r\um\ah\ s\ay\ia : J\l. P\ah\l\aw\an No. 34
S\ur\ab\ay\ia T\im\ur.
Menggunakan parameter "A..Z" adalah:
\A\l\am\at\ r\um\ah\ s\ay\ia : \J\l. \P\ah\l\aw\an \No. 34 \S\ur\ab\ay\ia \T\im\ur.
Menggunakan parameter "0..9" adalah:
A\l\am\at\ r\um\ah\ s\ay\ia : J\l. P\ah\l\aw\an No. \3\4 Surabaya Timur.
Menggunakan parameter "a" adalah:
A\l\am\at\ r\um\ah\ s\ay\ia : J\l. P\ah\l\aw\an No. 34 S\ur\ab\ay\ia T\im\ur.
Menggunakan parameter "P,S" adalah:
A\l\am\at\ r\um\ah\ s\ay\ia : J\l. \P\ah\l\aw\an No. 34 \S\ur\ab\ay\ia T\im\ur.

```

Gambar 9.1 Tampilan teks dengan penambahan tanda slesh

9.1.2 Menghilangkan Tanda Slesh “\”

Untuk menghapus tanda slash pada data string dapat menggunakan fungsi `stripslashes()`. Tanda slash yang ada pada sebuah data string akan dihilangkan. Bentuk syntaksnya adalah:

Stripslashes(“string”);

Keterangan :

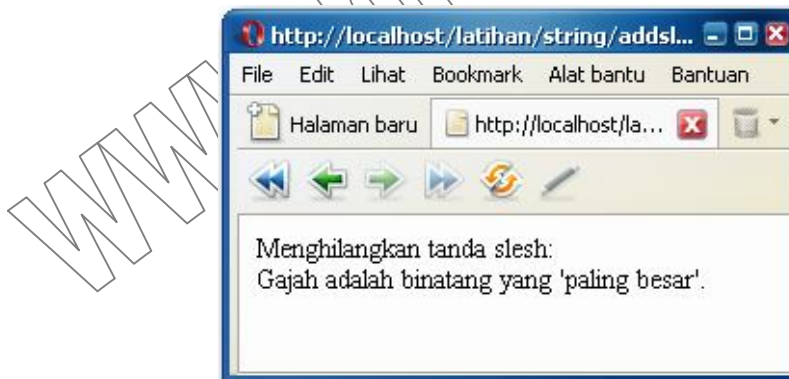
String adalah data string yang akan dihapus tanda sleshnya, jika data string tersebut memiliki tanda slesh.

Berikut adalah contoh dalam sebuah script:

```
<?php
//stripslashes.php
echo"Menghilangkan tanda slesh:<br>";
$str = "\\Gajah adalah binatang yang 'paling besar'.";
echo stripslashes($str);
?>
```

Penjelasan script:

Pada contoh tersebut, string diambil dari nilai variabel `$str` yang berisikan nilai teks “\\Gajah adalah binatang yang 'paling besar'.". Apabila dijalankan pada browser maka tanda slesh akan dihilangkan. Perhatikan pada gambar berikut:



Gambar 9.2 Tampilan string yang sudah dihilangkan tanda sleshnya

9.2 Fungsi Untuk Memanipulasi Kode HTML

Banyak sekali fungsi PHP yang digunakan untuk memanipulasi teks menjadi kode HTML. Hal ini dilakukan bertujuan untuk memudahkan program dalam berpindah dari Script PHP ke HTML atau sebaliknya. Ada beberapa fungsi PHP yang digunakan untuk memanipulasi kode HTML.

9.2.1 Fungsi nl2br

Fungsi nl2br digunakan untuk mengkonversi script baris baru (`\n`) pada script PHP menjadi kode HTML `
` (baris baru). Fungsi nl2br mulai dapat berjalan pada modul PHP versi 4.0.5. Bentuk syntaksnya adalah:

nl2br("string");

Keterangan :

String adalah teks atau data yang diperiksa, apabila memiliki kode `\n` akan dikonversi menjadi `
` sebagai garis baru pada kode HTML.

Contoh:

```
<?php
// nl2br.php
$data="selamat datang \nDi website Saya";
$fungsi=nl2br($data);
echo "Data Awal: $data";
echo "<br>Data baru : $fungsi";
?>
```

Penjelasan script:

Pada contoh penggunaan fungsi **nl2br** di atas memeriksa data yang berada pada variabel **\$data**, kemudian pada data tersebut akan diperiksa keberadaan script (`\n`) dan kemudian diubah menjadi `
` yang berarti baris baru pada kode HTML.

9.2.2 Fungsi strip_tags

Fungsi strip_tags digunakan untuk menghilangkan kode-kode PHP dan HTML. Ada dua bentuk fungsi pada fungsi ini, yang pertama Strip_tags(teks) dan strip_tags(teks, kode yang diabaikan). Bentuk sintaksnya adalah:

Strip_tags(string);

Strip_tags(string, kode yang diabaikan);

Keterangan :

String adalah data teks yang dihilangkan kode HTML maupun PHPnya.

Kode yang diabaikan adalah kode PHP atau HTML yang khusus diabaikan atau tidak dihilangkan oleh fungsi strip_tags(). Apabila bagian ini tidak diisi, maka semua kode PHP dan HTML akan dihilangkan.

Contoh:

```
<?php
// strip_tags.php
$data = "<p>Selamat Datang Di website kami.</p>
<p>Hari ini hari Senin. tanggal 27 Nopember 2005</p>
<br>Nama saya Rafiza Hafif.";
$stripl=strip_tags($data);
echo $stripl;
echo "<br>-----<br>";
// abaikan kode <p>
$strip2=strip_tags($data, '<p>');
echo $strip2;
?>
```

Hasilnya apabila dijalankan pada browser seperti pada gambar berikut:



Gambar 9.3 Tampilan string yang dihilangkan kode HTML-nya

Apabila halaman tersebut dilihat kode HTML-nya tampak seperti dibawah ini:

```
Selamat Datang Di website kami.
Hari ini hari Senin. tanggal 27 Nopember
2005
Nama saya Rafiza Hafif.<br>-----
<br><p>Selamat Datang Di website kami.</p>
<p>Hari ini hari Senin. tanggal 27
Nopember 2005</p>
Nama saya Rafiza Hafif.
```

Gambar 9.4 Tampilan kode HTML pada halaman strip_tags.php

9.2.3 Fungsi htmlspecialchars()

Fungsi ini digunakan untuk mengkonversikan karakter-karakter tertentu menjadi kode-kode HTML. Beberapa karakter tersebut adalah:

Karakter	Kode HTML
& (tanda dan)	& (mencetak tanda & pada browser)
“ (tanda petik ganda)	" (Mencetak tanda petik pada browser)

' (petik tunggal)	> (mencetak tanda petik tunggal pada browser)
< (lebih kecil)	< (mencetak tanda lebih kecil pada browser)
> (lebih besar)	> (mencetak tanda lebih besar pada browser)

Bentuk syntaksnya adalah :

htmlspecialchars(string, mode);

Keterangan :

String adalah data teks yang menjadi data sumber.

Mode adalah format karakter yang dijadikan sebagai acuan konversi data.

Contoh:

```
<?php
// htmlspecialchars.php
$data = htmlspecialchars("<a href='halaman dua'>halaman
dua</a>", ENT_QUOTES);
echo $data;
?>
```

Hasilnya pada karakter khusus seperti < (lebih kecil), ' (tanda petik tunggal) dan > (tanda lebih besar) akan diubah menjadi kode-kode HTML yang berfungsi menampilkan karakter-karakter tersebut pada browser. Script tersebut apabila dijalankan pada browser dan dilihat source codenya akan tampak seperti berikut:

```
1 &lt;a href='&#039;halaman dua&#039;&gt;halaman dua&lt;/a&gt;;
2
```

Gambar 9.5 Tampilan kode HTML pada halaman htmlspecialchars.php

9.2.4 Fungsi quotemeta

Fungsi ini digunakan untuk memberikan tanda slesh di depan karakter tertentu, sehingga karakter khusus tersebut menjadi sebuah data string. Bentuk syntaksnya adalah:

Quotemeta(string);

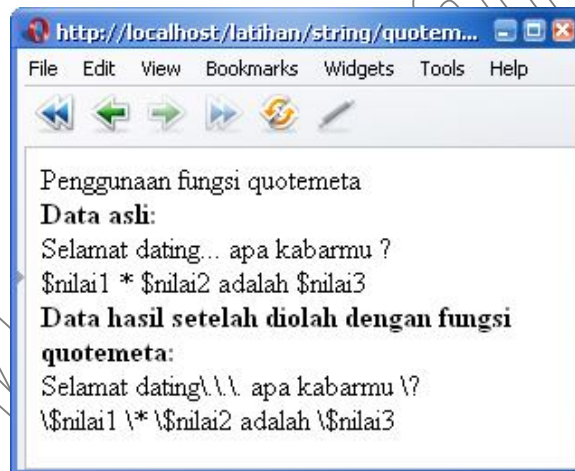
Keterangan :

String adalah data teks yang dijadikan data sumber. Apabila data teks tersebut memiliki karakter khusus akan ditambahkan tanda slesh di depannya. Karakter yang akan ditambahkan tanda slesh di depannya adalah: .(tanda titik) \ (slash) + (tanda positif) * (asterisk) ? (tanda Tanya) [^] dan (\$).

Contoh:

```
<?php
// quotemeta.php
echo"Penggunaan fungsi quotemeta<br>";
$data = " Selamat datang... apa kabarmu ?<br>
\nilai1 * \nilai2 adalah \nilai3";
echo"<b>Data asli:</b><br>$data";
echo"<br><b>Data hasil setelah diolah dengan fungsi
quotemeta:</b><br>";
$hasil=quotemeta($data);
Echo $hasil;
?>
```

Hasilnya dapat Anda lihat sebagai berikut:



Gambar 9.6 Tampilan hasil penggunaan fungsi qoutemeta

9.2.5 Fungsi htmlentities dan html_entity_decode

Fungsi `htmlspecialchars` digunakan untuk mengkonversikan karakter khusus yang sering digunakan pada kode html menjadi kode html, contohnya `<` menjadi `<`, tanda `>` menjadi `>`; dan lain sebagainya. Sedangkan fungsi `htmlspecialchars_decode` digunakan untuk mengembalikan kode html tersebut menjadi karakter semula. Bentuk sintaksnya adalah:

`htmlspecialchars("string");`

`htmlspecialchars_decode("string");`

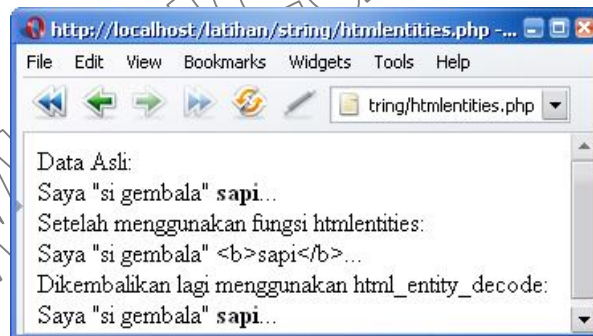
Keterangan :

String adalah data teks yang menjadi data sumber.

Contoh:

```
<?php
//htmlspecialchars.php
$data_asli="Saya \"si gembala\" <b>sapi</b>...";
$hasil1=htmlspecialchars($data_asli);
$hasil2=htmlspecialchars_decode($hasil1);
echo "Data Asli:<br>$data_asli<br>";
echo "Setelah menggunakan fungsi
htmlspecialchars:<br>$hasil1<br>";
echo "Dikembalikan lagi menggunakan
htmlspecialchars_decode:<br>$hasil2"/;
?>
```

Jalankan script di atas pada browser. Hasilnya dapat Anda lihat pada gambar berikut ini:



Gambar 9.7 Tampilan penggunaan fungsi `htmlspecialchars` dan `htmlspecialchars_decode`

9.3 Memotong dan Menggabung String

Ada beberapa fungsi yang digunakan untuk memotong dan menggabung kembali data string. Berikut ini adalah beberapa fungsi yang digunakan untuk melakukan tugas tersebut dan sering digunakan pada sebuah aplikasi pembuata website:

9.3.1 Fungsi Explode

Fungsi ini digunakan untuk memotong data string menjadi beberapa bagian berdasarkan mode tertentu. Bentuk syntaksnya adalah:

Explode(mode, data string, jumlah potongan);

Keterangan :

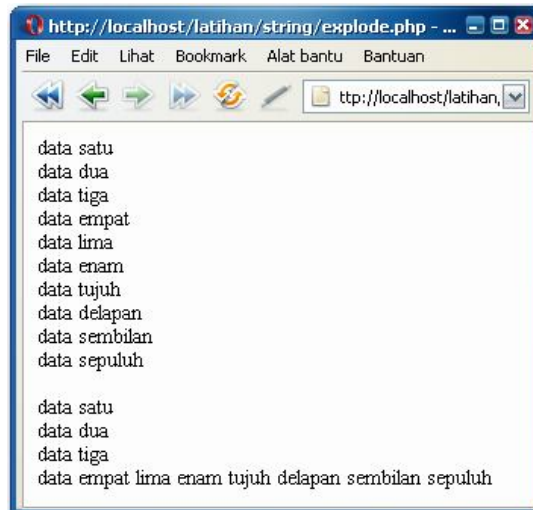
Mode adalah karakter tertentu yang ditetapkan menjadi parameter pemisah.

Data string adalah data sumber yang dipotong.

Jumlah potongan adalah sebuah angka yang menyatakan jumlah maksimal hasil pemotongan. Apabila bagian ini tidak dideklarasikan, maka data string dipotong sebanyak mode atau parameter yang ada pada data string tersebut. Berikut adalah contoh penggunaan fungsi tersebut dalam sebuah script:

```
<?php
//explode.php
$data="satu dua tiga empat lima enam tujuh delapan sembilan
sepuluh";
$potong1=explode(" ", $data);
for($a=0;$a<count($potong1);$a++){
    echo "data $potong1[$a]<br>";
}
echo"<br>";
$potong2=explode(" ", $data, 4);
for($a=0;$a<count($potong2);$a++){
    echo "data $potong2[$a]<br>";
}
?>
```

Simpan script tersebut dengan nama file **explode.php** dan jalankan pada browser. Hasilnya dapat Anda lihat pada gambar berikut ini:



Gambar 9.8 Tampilan penggunaan fungsi explode

9.3.2 Fungsi Split

Fungsi split memiliki penggunaan yang sama dengan fungsi **explode**, yaitu memotong data string berdasarkan mode tertentu. Bentuk syntaksnya adalah:

Split(mode, data string);

Keterangan :

Mode adalah karakter tertentu yang ditetapkan menjadi parameter pemisah.

Data string adalah data sumber yang dipotong.

Berikut contoh dalam sebuah script:

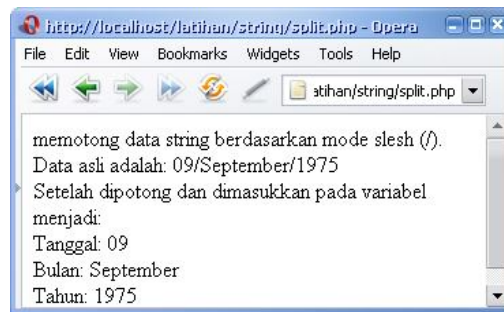
```
<?php
// split.php
$data = "09/September/1975";
list($hari, $bulan, $tahun) = split('[/.-]', $data);
echo "memotong data string berdasarkan mode slash (/).<br>";
echo "Data asli adalah: $data<br>";
echo "Setelah dipotong dan dimasukkan pada variabel
menjadi:<br>";
```

Oleh : Yeni Cahyono- www.nicdesain.net 69

```
echo "Tanggal: $hari<br>Bulan: $bulan<br>Tahun: $tahun<br
/>\n" ;
?>
```

Pada script di atas menggunakan fungsi **split** yang memotong string data tanggal menggunakan mode / parameter **slesh (/)**. Hasil dari fungsi split ini akan dimasukkan pada variabel-variabel tersendiri yang sudah dideklarasikan pada fungsi list. Berikutnya akan dicetak menggunakan perintah **echo**.

Simpan script tersebut dengan nama file **explode.php** dan jalankan pada browser. Hasilnya dapat Anda lihat pada gambar berikut ini:



Gambar 9.9 Tampilan penggunaan fungsi *split*

9.3.3 Fungsi Implode

Fungsi implode digunakan untuk menggabung data yang sudah dipotong dengan fungsi explode maupun split. Bentuk sintaksnya adalah:

implode(pemisah, string);

Keterangan :

Pemisah adalah karakter tertentu yang dijadikan pemisah antara bagian-bagian data yang digabung.

String adalah data array yang akan digabung.

Contoh:

70 Oleh : *Yeni Cahyono- www.nicdesain.net*

```

<?php
//implode.php
// menggabung data.
$data = array('Nama : Rafiza Hafif', 'Tanggal Lahir : 9
September 1975', 'Email : rafiza_hafif@yahoo.com');
$digabung = implode("<br>", $data);
echo"<b>Data yang sudah digabung:</b><br>";
echo"$digabung";
?>

```

Pada script di atas berfungsi menggabung data menggunakan fungsi implode dan pemisahannya menggunakan **
, sehingga pada masing-masing data akan dicetak di bawahnya. Simpan file tersebut dengan nama **implode.php dan panggil melalui browser untuk melihat hasilnya.

9.3.4 Fungsi join

Fungsi join digunakan untuk menggabung data, sama seperti fungsi implode. Bentuk syntaksnya adalah :

join(pemisah, string);

Keterangan :

Pemisah adalah karakter tertentu yang dijadikan pemisah antara bagian-bagian data yang digabung.

String adalah data array yang akan digabung.

9.3.5 Fungsi strtok

Fungsi strtok digunakan untuk memotong data string menjadi beberapa bagian berdasarkan mode tertentu. Bentuk syntaksnya adalah:

Strtok(string1, string2);

Keterangan :

String1 adalah data teks yang dijadikan data sumber.

String2 adalah data yang berupa karakter-karakter tertentu yang menjadi mode pemotongan data string1.

Contoh:

```

<?php
//strtok.php
$data = "Contoh data\nyang\tdipisah.";
echo "Data awal: <b>".$data."</b><br>";
$hasil = strtok($data, " \n\t");
while ($hasil) {
    echo "Kata = $hasil<br>";
    $hasil = strtok("\n\t");
}
?>

```

9.4 Membandingkan dan Mengganti String

9.4.1 Fungsi strstr

Fungsi strstr digunakan untuk mengambil potongan string yang diawali dengan karakter tertentu yang sudah ditentukan oleh mode. Fungsi strstr ini membedakan huruf besar dan huruf kecil. Apabila anda ingin mengabaikan huruf besar atau huruf kecil, maka anda dapat menggunakan fungsi **stristr()**. Bentuk maupun susunan fungsi dari keduanya adalah sama. Bentuk sintaksnya adalah:

Strstr(String, mode);

Keterangan :

String adalah data sumber yang dijadikan data awal.

Mode adalah karkter tertentu yang dijadikan awal pengambilan atau pemotongan data string.

Contoh:

```

<?php
// strstr.php
echo "<b>Penggunaan Fungsi strstr:<br></b>";
$email = 'rafiza_hafif@yahoo.com';
$domain = strstr($email, '@');
echo "Data awal adalah : $email.<br>";
echo "Data hasil yang diambil setelah tanda @ adalah :
$domain";
?>

```

Pada script di atas memiliki fungsi mengambil potongan string yang diawali dengan karakter "@". Apabila dijalankan menggunakan browser, hasilnya dapat Anda lihat pada gambar berikut:



Gambar 9.10 Tampilan penggunaan fungsi strpos

9.4.2 Fungsi strpos

Fungsi strpos digunakan untuk memeriksa posisi karakter tertentu pada sebuah data string. Huruf besar dan huruf kecil dibedakan. Bentuk sintaksnya adalah:

Strpos(string, mode);

Keterangan :

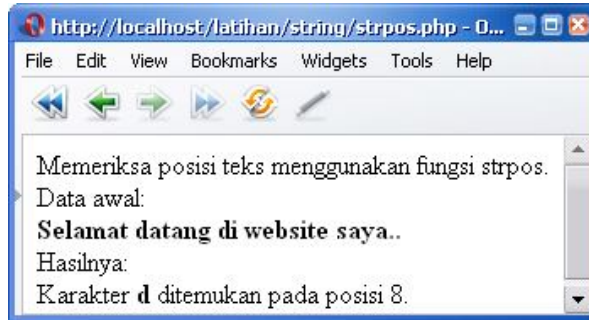
String adalah data teks yang dijadikan data sumber pencarian.

Mode adalah karakter yang dijadikan sebagai data pencari.

Contoh:

```
<?php
// strpos.php
// mencari posisi karakter tertentu pada
// sebuah data string.
$data="Selamat datang di website saya..";
$pencari="d";
$hasil=strpos($data, $pencari);
echo "Memeriksa posisi teks menggunakan fungsi strpos.<br>";
echo "Data awal:<br>";
echo "<b>$data</b><br>";
echo "Hasilnya:<br>";
echo "Karakter <b>$pencari</b> ditemukan pada posisi
$hasil.";
?>
```

Simpan script di atas dengan nama **strpos.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar berikut:



Gambar 9.11 Tampilan penggunaan fungsi strpos

9.4.3 Fungsi strpos

Fungsi **strpos** digunakan untuk memeriksa posisi teks pada sebuah string tanpa membedakan huruf besar maupun huruf kecil. Bentuk sintaksnya adalah:

Stripos(string, mode);

Keterangan :

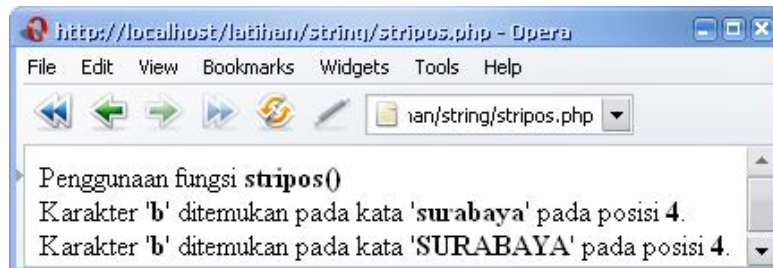
String adalah data teks yang dijadikan data sumber / yang diperiksa.

Mode adalah teks atau huruf yang diperiksa posisinya dalam data string.

Contoh:

```
<?php
// strpos.php
// memeriksa keberadaan teks tanpa
// membedakan huruf besar dan kecil.
$ pencari = 'b';
$string1 = 'surabaya';
$string2 = 'SURABAYA';
$hasil1 = strpos($string1, $pencari);
$hasil2 = strpos($string2, $pencari);
echo "Penggunaan fungsi <b>strpos()</b><br>";
echo "Karakter '<b>$pencari</b>' ditemukan pada kata
'<b>$string1</b>' pada posisi <b>$hasil1</b>.<br>";
echo "Karakter '<b>$pencari</b>' ditemukan pada kata
'<b>$string2</b>' pada posisi <b>$hasil2</b>.";
?>
```

Simpan script di atas dengan nama **strpos.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar berikut:



Gambar 9.12 Tampilan penggunaan fungsi stripos

9.4.4 Fungsi count_chars

Fungsi **count_chars** digunakan untuk menghitung jumlah karakter pada sebuah data string. Bentuk adalah:

Count_chars(string, mode);

Keterangan :

String adalah data sumber yang dihitung huruf dan jumlahnya.

Mode adalah parameter yang dijadikan bentuk pemeriksaan dan penghitungan karakter.

Berikut adalah contoh penggunaan fungsi count_chars() dalam sebuah script:

```
<?php
// count_chars.php
// Menghitung karakter pada sebuah string
$data = "Selamat datang di website Saya..";
$hasil = count_chars($data, 0);
echo "Data String:<br>
<b>$data</b><br>";
for ($i=0; $i < count($hasil); $i++) {
if ($hasil[$i] != 0)
echo "Ditemukan karakter \"", chr($i) , "\" sebanyak
$hasil[$i] buah.<br>";
}
?>
```

9.4.5 Fungsi strlen

Fungsi **strlen** digunakan untuk menghitung jumlah keseluruhan karakter dari sebuah data string. Bentuk syntaksnya adalah:

Strlen(string);

Keterangan :

String adalah data teks yang dihitung jumlah karakternya.

Contoh:

```
<?php
//strlen.php
// menghitung jumlah karakter
$data = 'Nama saya Rafiza Hafif';
echo "Data String:<br>$data<br>";
echo "Jumlah karakternya adalah: ",strlen($data);
?>
```

Simpan script di atas dengan nama **strlen.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar berikut.



Gambar 9.13 Menghitung jumlah karakter dari sebuah string

9.4.6 Fungsi Strspn

Fungsi **strspn** digunakan untuk memeriksa jumlah karakter di awal string dengan menggunakan data pembanding tertentu. Bentuk syntaksnya adalah:

Strspn(string1, string2);

Keterangan :

String1 adalah data string yang dijadikan data sumber.

String2 adalah data string yang dijadikan pembanding atau mode.

Contoh:

76 Oleh : *Yeni Cahyono*- www.nicdesain.net

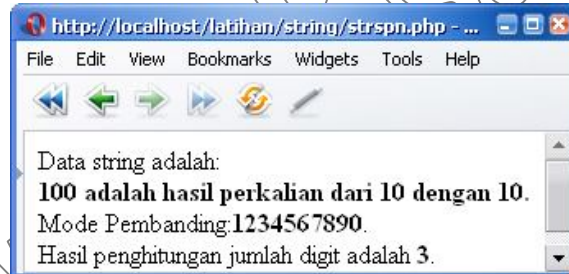
```

<?php
// strspn.php
// menghitung karakter pertama berdasarkan mode.
$data="100 adalah hasil perkalian dari 10 dengan 10.";
$pembanding="1234567890";
$hasil = strspn($data, $pembanding);
echo "Data string adalah:<br><b>$data</b><br>";
echo "Mode Pembanding:<b>$pembanding</b>.<br>";
echo "Hasil penghitungan jumlah digit adalah
<b>$hasil</b>.";
?>

```

Variabel data pada script di atas adalah sebagai data yang dijadikan data sumber. Kemudian pada variabel **\$pembanding** memiliki data pembanding yang berisi data "1234567890". Jadi apabila data awal dari variabel data tersebut merupakan angka maka akan dihitung sampai data angka tersebut habis diperiksa. Pembanding dari fungsi strspn ini juga dapat diisi karakter atau huruf tertentu, misalkan "abc". Apabila pada awal data yang dimiliki oleh variabel **\$data** mengandung huruf **a**, **b** atau **c** maka akan dihitung.

Simpan script di atas dengan nama **strspn.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar 9.14 berikut:



Gambar 9.14 Menghitung jumlah karakter di awal string menggunakan fungsi strspn

9.4.7 Fungsi strrev

Fungsi **strrev** digunakan untuk membalik pembacaan data string., sehingga data string akan dicetak dari belakang ke depan. Bentuk syntaksnya adalah:

Strrev(string);

Keterangan :

String adalah data teks yang dijadikan data sumber.

Contoh:

```
<?php
// strrev.php
// Membaca data dari belakang
$data="Selamat Datang...";
$hasil=strrev($data);
echo"Membalik pembacaan data dengan strrev.<br>";
echo"Data awal:<b>\"$data\"</b><br>";
echo"Setelah diubah:<b>\"$hasil\"</b>";
?>
```

Simpan script di atas dengan nama strrev.php dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar 9.15 berikut:



Gambar 9.15 Membaca data dari belakang menggunakan fungsi strrev

9.4.8 Fungsi Substr_count

Fungsi **substr_count** digunakan untuk menghitung jumlah karakter tertentu pada sebuah string. Bentuk syntaksnya adalah:

Substr_count(string, mode);

Keterangan :

String adalah data awal yang dicari karakter khususnya.

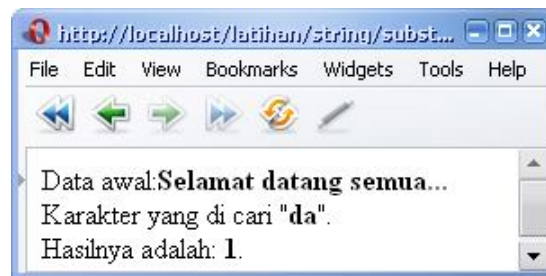
Mode adalah karakter tertentu yang dicari jumlahnya pada data string.

78 Oleh : Yeni Cahyono- www.nicdesain.net

Contoh:

```
<?php
//substr_count.php
$data="Selamat datang semua...";
$cari=substr_count($data, "da");
echo "Data awal:<b>$data</b><br>";
echo "Karakter yang di cari \"<b>da</b>\".<br>";
echo "Hasilnya adalah: <b>$cari</b>.";
?>
```

Simpan script di atas dengan nama **substr_count.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar 9.16 berikut:



Gambar 9.16 Menghitung jumlah karakter menggunakan fungsi **substr_count**

9.4.9 Fungsi **ereg**, **eregi**, **stristr**

Untuk mencari atau mencocokkan bagian dari suatu string yang cocok dengan string atau karakter lain dapat menggunakan beberapa fungsi, diantaranya adalah **Ereg()**, **Eregi()** dan **Stristr()**. Bentuk syntaksnya adalah:

Ereg(Parameter, String);

Eregi(Parameter, String);

Stristr(String, Parameter);

Keterangan :

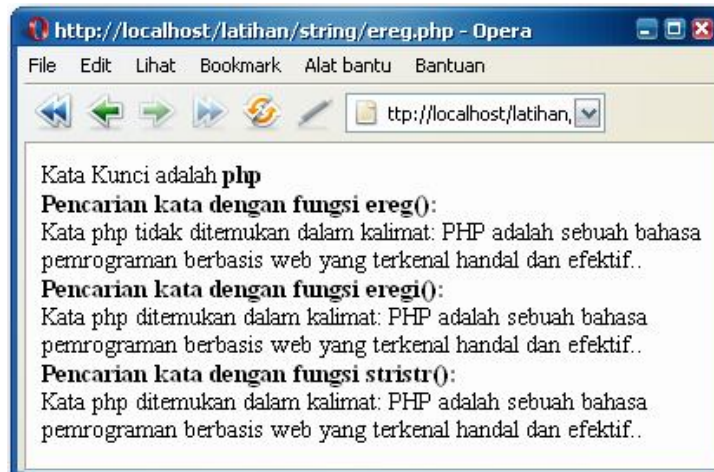
String adalah argumen yang menunjukkan data sumber yang akan dicari kesamaannya.

Parameter adalah argumen yang menunjukkan data pembanding

Contoh:

```
<?
// ereg.php
$data="PHP adalah sebuah bahasa pemrograman berbasis web
yang terkenal handal dan efektif.";
$cari="php";
$ereg=Ereg($cari, $data);
$eregi=Eregi($cari, $data);
$stristr=Stristr($data, $cari);
Echo"Kata Kunci adalah <b>$cari</b><br>";
Echo"<b>Pencarian kata dengan fungsi ereg():</b><br>";
If($ereg){
Echo"Kata $cari ditemukan dalam kalimat: $data. <br>";
}else{
Echo"Kata $cari tidak ditemukan dalam kalimat: $data.
<br>";
}
Echo" <b>Pencarian kata dengan fungsi eregi():</b><br>";
If($eregi){
Echo"Kata $cari ditemukan dalam kalimat: $data.<br>";
}else{
Echo"Kata $cari tidak ditemukan dalam kalimat: $data.
<br>";
}
Echo" <b>Pencarian kata dengan fungsi stristr():</b><br>";
If($stristr){
Echo"Kata $cari ditemukan dalam kalimat: $data.<br>";
}else{
Echo"Kata $cari tidak ditemukan dalam kalimat: $data.
<br>";
}
?>
```

Hasil program apabila dijalankan pada browser akan tampak seperti pada gambar berikut :



Gambar 9.17 Membandingkan potongan string dengan string lain

Program PHP mengenal beberapa pola karakter yang dapat dijadikan sebagai kunci pencarian atau pencocokan data. Sebagai contoh, melakukan proses pencarian berdasarkan data angka saja, pencocokan data berdasarkan huruf vocal saja, dan sebagainya.

Misalnya memeriksa apakah suatu string mengandung karakter vocal (a, i, u, e, o) atau tidak. Dalam hal ini, Anda dapat menggunakan pola penulisan karakter sebagai berikut:

Ereg("[aiueo]", "ytghkd");

Bentuk penulisan fungsi di atas akan menghasilkan nilai False.

Selanjutnya, perhatikan bentuk penulisan fungsi berikut :

Ereg("[aiueo]", "surabaya");

Ereg("[aiueo]", "pemrograman");

Ereg("[aiueo]", "falsafah");

Fungsi-fungsi di atas akan menghasilkan nilai True, karena teks pada argumen kedua mengandung huruf yang terdapat pada tanda kurung siku dari argumen pertama.

Perhatikan pola penulisan program berikut:

[aiueo]

Pola tersebut mempunyai pengertian bahwa salah satu karakter yang berbeda dalam antara tanda [dan].

Untuk mendeklarasikan karakter antara, Anda dapat menggunakan tanda “-“ yang berarti “sampai dengan”.

Pola Karakter	Keterangan
[a-z]	Karakter antara huruf kecil a sampai z.
[A-Z]	Karakter antara huruf kapital A sampai Z.
[0-9]	Karakter antara angka 0 sampai 9.

Berikut ini beberapa contoh penggunaan pola karakter yang digunakan untuk mencocokkan dengan sebuah string.

Fungsi	Hasil	Keterangan
Ereg(“[abc]”, “harimau”)	True	Mengandung karakter “a”
Ereg(“[abc]”, “gunung”)	False	Tidak mengandung karakter “a” atau “b” atau “c”
Ereg(“[ab][cd]”, “kucing”)	False	Tidak mengandung karakter “a” atau “b” dan “c” atau “d”.
Ereg(“[ab][cd]”, “kancing”)	True	Mengandung “a” dan “c”.
Ereg(“[ab][cd]”, “badai”)	True	Mengandung “a” dan “d”.
Ereg(“[ab][cd]”, “kucing”)	False	Tidak mengandung “a” atau “b”.
Ereg(“[ab][cd]”, “acong”)	True	Mengandung “a” dan “c”.
Ereg(“[ab][cd]\$, “abad”)	True	Mengandung “a” dan “d” diakhir string.
Ereg(“[ab][cd]\$, “acong”)	False	Karakter diakhir string bukan “ac” atau “ad” atau “bc” atau “bd”.
Ereg(“^[ab][cd]”, “pacar”)	False	Mengandung karakter “a”.

Tanda \$ di belakang tanda] mempunyai pengertian bahwa akan bernilai True jika string diakhiri dengan karakter yang cocok dengan pola yang sudah ditentukan. Tanda ^ di awal karakter pada pola mempunyai

pengertian bahwa akan bernilai True jika tidak cocok dengan karakter setelah tanda ^. Sebagai contoh, penulisan pola [^A-Z] mempunyai pengertian selain huruf kapital A sampai Z.

Pada penulisan pola karakter tertentu, ada beberapa hal yang perlu diperhatikan. Sebagai contoh, Anda akan mencocokkan tanda titik, baris baru dan tab, yaitu untuk mendeklarasikan tanda titik, Anda dapat memberikan tanda \ sebelum titik (.), sehingga berbentuk [\.]. Sedangkan untuk baris baru atau tab Anda dapat menuliskannya dengan bentuk penulisan pola [\n] dan [\t].

Selain itu, PHP juga menyediakan bentuk pencocokan string dengan pola pengulangan karakter. Misalnya adalah string yang cocok dengan dua karakter "a" dan berurutan. Pendeklarasian pola ini adalah dengan menggunakan tanda { }. Perhatikan contoh pada tabel berikut :

Pola	Keterangan	Contoh True	Contoh False
b{2}	Mencocokkan 2 huruf b yang posisinya berurutan.	"abba" "babba" "abba"	"bobo" "babi"
b{2,4}	Mencocokkan 2 huruf b atau 4 huruf b yang posisinya berurutan	"babba" "baju" "bawang"	"merah" "
b{2,}	Mencocokkan 2 huruf b atau lebih.	"abba" "abbbba"	"baba" "
e{2}\$	Mencocokkan 2 huruf e yang posisinya terletak di akhir string.	"babee" "adee"	"adeek" "eenak" "
.{2}	Mencocokkan 2 huruf apa saja yang posisinya berurutan	"maaf" "aa4"	"a4e"
[A-Z]{2}	Mencocokkan 2 huruf kapital A sampai Z yang posisinya berurutan	"AA4" "bILa"	"Bila" "semu" "A"
b{0,1}	Mencocokkan 1 huruf b atau yang tidak mengandung huruf b	"maksud" "badan"	

9.4.10 Fungsi ereg_replace, eregi_replace, str_replace

Adapun beberapa fungsi yang dapat digunakan untuk mencari dan mengganti bagian string tertentu adalah fungsi `Ereg_Replace()`, `Eregi_Replace()`, dan `Str_Replace()`. Ketiga fungsi tersebut mempunyai kegunaan yang sama yaitu untuk mencari pola tertentu pada sebuah data string dan menggantinya dengan string yang baru. Bentuk syntaksnya adalah:

`Ereg_Replace(pola awal, pola pengganti, string);`

`Eregi_Replace(pola awal, pola pengganti, string);`

`Str_Replace(pola pengganti, pola awal, string);`

Keterangan :

Pola awal adalah argumen yang menyatakan data tertentu yang akan diubah.

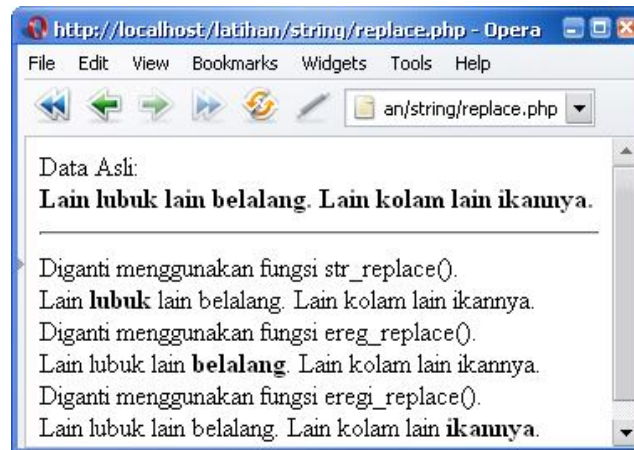
Pola pengganti adalah argumen yang menyatakan data tertentu yang akan dijadikan pengganti.

String adalah argumen yang menyatakan data sumber baik berupa huruf maupun angka yang akan diubah.

Contoh:

```
<?
// replace.php
$data="Lain lubuk lain belalang. Lain kolam lain ikannya.";
echo "Data Asli:<br><b>$data</b><hr>";
$str="lubuk";
$str2=str_replace($str,"<b>$str</b>", $data);
Echo " Diganti menggunakan fungsi str_replace().";
Echo "<br>$str2<br>";
$ereg="belalang";
$ereg2=ereg_replace($ereg,"<b>$ereg</b>", $data);
Echo " Diganti menggunakan fungsi ereg_replace().";
Echo "<br>$ereg2<br>";
$eregi="ikannya";
$eregi2=eregi_replace($eregi,"<b>$eregi</b>", $data);
Echo " Diganti menggunakan fungsi eregi_replace().";
Echo "<br>$eregi2<br>";
?>
```

Simpan script di atas dengan nama **replace.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar 9.18 berikut:



Gambar 9.18 Mencari dan mengganti potongan string

9.4.11 Fungsi str_ireplace

Fungsi ini digunakan untuk mengganti potongan string dengan string baru dan digabung menjadi sebuah data string. Fungsi **str_ireplace** mulai dapat berjalan pada PHP versi 5.0. Jadi kalau PHP Anda masih menggunakan versi 4 atau di bawahnya, pasti tidak akan jalan dech. Bentuk syntaksnya adalah:

Str_ireplace(potongan string, string pengganti, string hasil);

Keterangan :

Potongan string adalah bagian string yang ada di string hasil yang akan diganti.

String pengganti adalah data teks sebagai pengganti potongan string.

String hasil adalah string sumber dan juga sebagai string hasil dari penggantian potongan string tersebut.

Contoh:

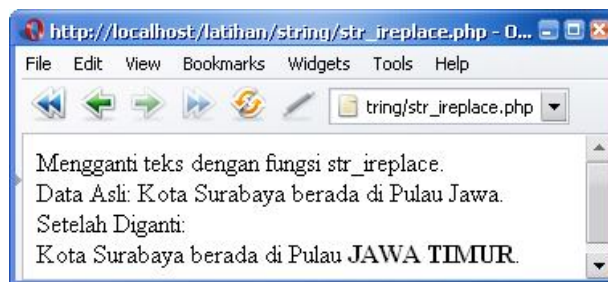
```
<?php
//str_ireplace.php
//mengganti teks dengan pola tertentu.
$string="Kota Surabaya berada di Pulau Jawa.";
$encari="Jawa";
```

```

$pengganti="<b>JAWA TIMUR</b>";
$hasil = str_ireplace($pencari, $pengganti, $string);
echo"Mengganti teks dengan fungsi str_ireplace.<br>";
echo"Data Asli: $string<br>";
echo"Setelah Diganti:<br>$hasil";
?>

```

Simpan script di atas dengan nama **str_ireplace.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar 9.19 berikut:



Gambar 9.19 Mencari dan mengganti potongan string

9.4.12 Fungsi **str_pad**

Fungsi **str_pad** digunakan untuk menambahkan karakter tertentu pada sebuah string. Jumlah karakter yang ditambahkan akan diulang-ulang dan ditambah string awal sehingga jumlah karakter sesuai dengan jumlah yang ditentukan. Bentuk sintaksnya adalah:

Str_pad(string, jumlah karakter, string tambahan, mode penambahan);

Keterangan :

String adalah data awal atau data sumber.

Jumlah karakter adalah banyaknya karakter yang akan dimasukkan pada string hasil.

String tambahan adalah string yang ditambahkan. Apabila jumlah karakter string awal ditambah string tambahan kurang dari ketentuan jumlah karakter, maka string tambahan akan diulang sehingga jumlah karakter hasil sesuai dengan jumlah karakter yang ditentukan.

Mode penambahan adalah ketentuan posisi penambahan karakter / string, di depan, dibelakang atau kedua-duanya.

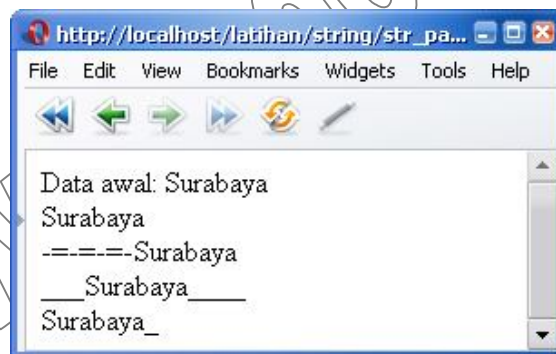
Beberapa mode tersebut adalah **STR_PAD_RIGHT** digunakan untuk menambahkan potongan string di sebelah kanan string awal, **STR_PAD_LEFT** digunakan untuk menambahkan potongan string disebelah kiri string awal, atau **STR_PAD_BOTH** digunakan untuk menambahkan potongan string di sebelah kiri dan kanan dari string awal. Apabila mode tidak dituliskan, maka penambahan string di sebelah kanan string awal.

Apabila string tambahan dan mode penambahan tidak dideklarasikan, maka string hasil akan ditambah spasi kosong di belakang string awal sehingga sejumlah karakter yang sudah ditentukan.

Contoh:

```
<?php
//srt_pad.php
$data = "Surabaya";
echo "Data awal: $data<br>";
echo str_pad($data, 15). "<br>";
echo str_pad($data, 15, "--", STR_PAD_LEFT). "<br>";
echo str_pad($data, 15, "_", STR_PAD_BOTH). "<br>";
echo str_pad($data, 9, "_");
?>
```

Simpan script di atas dengan nama **str_pad.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar 9.20 berikut:



Gambar 9.20 Menambah karakter menggunakan fungsi **str_pad**

9.4.13 Fungsi **Str_word_count**

Fungsi **str_word_count** digunakan untuk menghitung jumlah kata pada sebuah data string. Bentuk syntaksnya adalah:

Str_word_count(string, mode);

Keterangan :

String adalah data teks yang dihitung jumlah katanya.

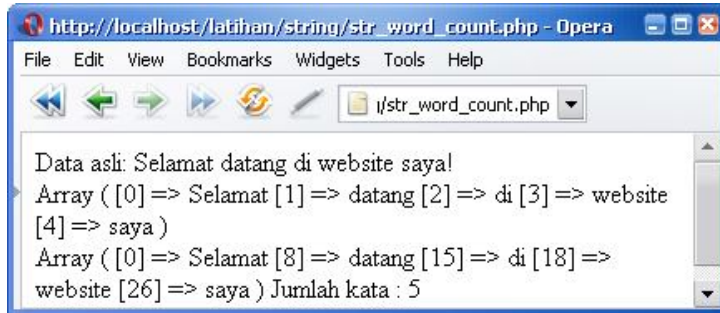
Mode adalah angka yang dijadikan sebagai mode angka index penghitungan karakter. Apabila mode tidak dideklarasikan, maka hasil dari fungsi ini adalah angka yang menyatakan jumlah kata dari data string tersebut.

Ada dua mode, yang pertama adalah 1, adalah mode yang digunakan untuk mendeklarasikan angka index dari hasil penghitungan kata berdasarkan nomor kata. Yang kedua adalah 2, adalah mode yang digunakan untuk mendeklarasikan angka index hasil dari penghitungan berdasarkan angka karakter.

Contoh:

```
<?php
//str_word_count.php
//Menghitung jumlah kata.
$data = "Selamat datang di website saya!";
$a = str_word_count($data, 1);
$b = str_word_count($data, 2);
$c = str_word_count($data);
echo "Data asli: $data<br>";
print_r($a);
echo "<br>";
print_r($b);
echo "<br>";
echo "Jumlah kata : $c";
?>
```

Simpan script di atas dengan nama **str_word_count.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar 9.21 berikut:



Gambar 9.21 Menghitung kata menggunakan fungsi `str_word_count`

9.4.14 Fungsi `substr_replace`

Fungsi yang digunakan untuk mengganti potongan string dengan string baru adalah menggunakan fungsi `Substr_replace()`. Bentuk syntaksnya adalah:

`$data=substr_replace(string_awal, string_pengganti, [awal_penggantian], [panjang_pengambilan]);`

Keterangan :

String_awal pada bentuk fungsi di atas adalah string atau variabel yang berisikan data teks sebelum diubah.

String_pengganti adalah string atau variabel yang mempunyai nilai string dan berfungsi sebagai pengganti.

Awal_penggantian adalah angka yang menyatakan posisi awal pengambilan karakter dari string awal.

Panjang_pengambilan adalah bernilai angka yang menyatakan jumlah karakter string awal yang akan diganti.

Contoh:

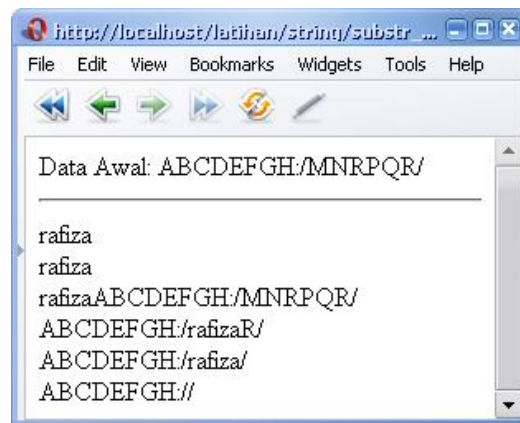
```
<?
// substr_replace.php
// fungsi mengganti potongan string dengan string yang baru
$var = 'ABCDEFGH:/MNRQR/';
echo "Data Awal: $var<hr>\n";
/* Dua contoh ini menggantikan semua dari variabel $var
dengan 'rafiza'. */
echo substr_replace($var, 'rafiza', 0) . "<br>\n";
```

```

echo substr_replace($var, 'rafiza', 0, strlen($var)) .
"<br>\n";
/* Masukkan 'irzha' pada kanan dari nilai dari variabel
$var. */
echo substr_replace($var, 'rafiza', 0, 0) . "<br>\n";
/* Berikutnya ini dua contoh mengganti 'MNRPQR' pada
variabel $ var dengan ' rafiza'. */
echo substr_replace($var, 'rafiza', 10, 5) . "<br>\n";
echo substr_replace($var, 'rafiza', -7, -1) . "<br>\n";
/* Menghapus 'MNRPQR' dari variabel $var. */
echo substr_replace($var, '', 10, -1) . "<br>\n";
?>

```

Simpan script di atas dengan nama **substr_replace.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar berikut:



Gambar 9.22 Mengganti potongan string menggunakan fungsi `str_replace`

9.4.15 Fungsi `str_repeat`

Fungsi ini digunakan untuk mencetak ulang data string tertentu pada sebuah browser. Bentuk syntaksnya adalah:

`Str_repeat(string, nilai pengulangan);`

Keterangan :

String adalah data string yang diulang dalam pencetakannya.

Nilai pengulangan adalah angka yang menyatakan jumlah pengulangan.

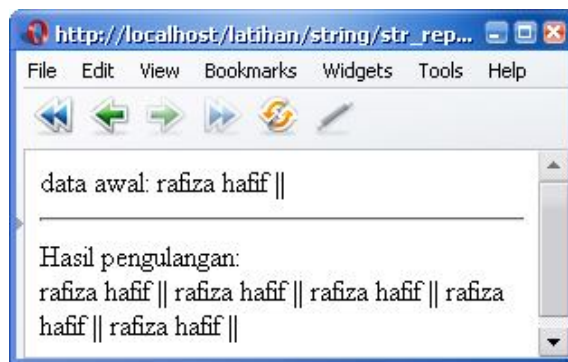
Contoh:


```

<?php
$data="rafiza hafif || ";
$hasil=str_repeat($data, 5);
echo"data awal: $data<hr>";
echo"Hasil pengulangan:<br>$hasil";
?>

```

Simpan script di atas dengan nama **str_repeat.php** dan jalankan pada browser. Hasilnya dapat anda lihat seperti pada gambar berikut:



Gambar 9.23 Mengulang data string menggunakan fungsi `str_repeat`

9.4.16 Fungsi `str_shuffle`

Fungsi `str_shuffle` ini digunakan untuk mengacak data string berdasarkan karakter yang ada pada data string tersebut. Bentuk syntaksnya adalah:

Str_shuffle(string);

Keterangan :

String adalah data sumber yang diacak karakternya.

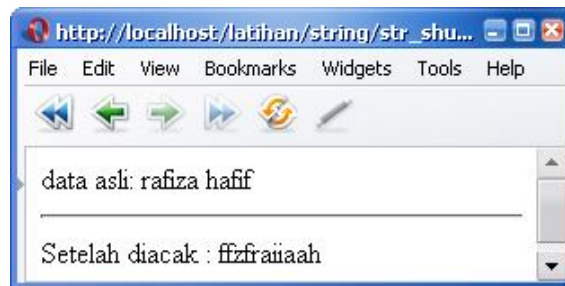
Contoh:

```

<?php
//str_shuffle.php
// mengacak data string
$data = "rafiza hafif";
$acak = str_shuffle($data);
echo"data asli: $data<hr>";
echo"Setelah diacak : $acak";
?>

```

Simpan script di atas dengan nama **str_shuffle.php** dan jalankan pada browser. Setiap browser di refresh, data yang diacak selalu berubah. Hasilnya dapat anda lihat seperti pada gambar berikut:



Gambar 9.24 Mengacak data string menggunakan fungsi **str_shuffle**

9.4.17 Fungsi Wordwrap

Fungsi ini digunakan untuk mengatur wrapping teks pada sebuah data string. Panjang wrapping dibatasi berdasarkan jumlah karkter tertentu. Bentuk syntaksnya adalah:

Wordwrap(string, panjang karakter, mode wrapping);

Keterangan :

String adalah data teks yang dijadikan data sumber.

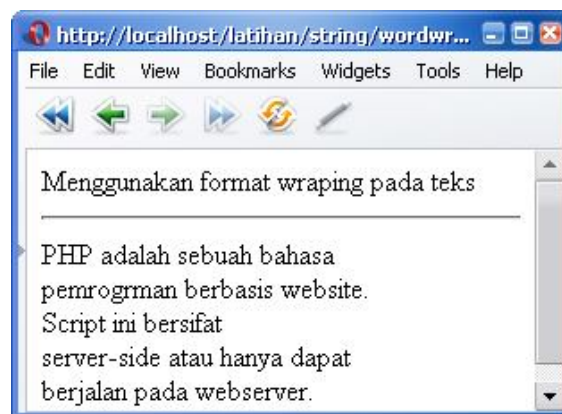
Panjang karakter adalah batas jumlah karakter yang digunakan untuk membatasi wrapping teks.

Mode wrapping adalah karakter atau kode HTML yang dijadikan batas wrapping.

Contoh:

```
<?php
//wordwrap.php
// membuat format wrapping dengan batas karakter tertentu.
$data = "PHP adalah sebuah bahasa pemrograman berbasis
website. Script ini bersifat server-side atau hanya dapat
berjalan pada webserver.";
$hasil = wordwrap($data, 30, "<br>");
echo "Menggunakan format wrapping pada teks<hr>";
echo "$hasil\n";
?>
```

Variabel **\$data** adalah variabel yang memiliki data teks yang dijadikan data sumber. Kemudian pada fungsi `wordwrap` memiliki nilai panjang **30**, artinya panjang karakter maksimal pada setiap baris teks adalah **30** karakter. Kemudian pada mode wrapping adalah sebuah tag **HTML** “`
`”, sehingga setiap baris wrapping akan diletakkan pada baris baru. Simpan script di atas dengan nama **wordwrap.php** dan jalankan pada browser.



Gambar 9.25 Penggunaan fungsi `wordwrap`

9.4.18 Fungsi mengubah Bentuk Huruf

Apabila suatu kondisi atau perintah mengharuskan Anda untuk mengubah suatu kata atau kalimat diubah bentuk penulisan hurufnya menjadi huruf kecil atau huruf kapital semua, maka program PHP menyediakan fungsi yang cukup efisien untuk digunakan.

Fungsi **Strtoupper()** adalah fungsi yang digunakan untuk mengubah huruf kecil menjadi huruf kapital.

Fungsi **Strtolower()** adalah fungsi yang digunakan untuk mengubah huruf kapital menjadi huruf kecil. Bentuk sintaksnya adalah:

Strtoupper(string);

Strtolower(string);

Keterangan :

String adalah data sumber dalam bentuk karakter yang akan diproses.

Namun pada kondisi tertentu, suatu argumen diharuskan hanya karakter pertama saja yang akan diubah menjadi huruf kapital atau setiap karakter pertama pada setiap kata pada sebuah kalimat. Adapun fungsi yang digunakan adalah :

Fungsi **Ucfirst()** adalah fungsi yang digunakan untuk mengubah karakter pertama dalam suatu kalimat menjadi huruf kapital.

Fungsi **Ucwords()** adalah fungsi yang digunakan untuk mengubah karakter dari setiap kata dalam suatu kalimat menjadi huruf kapital. Bentuk sintaksnya adalah:

Ucfirst(string);

Ucwords(string);

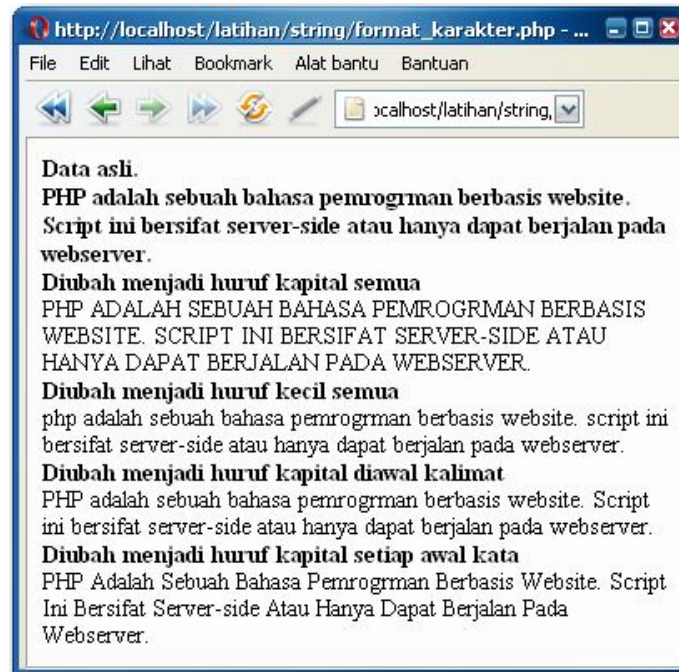
Keterangan :

String adalah data sumber dalam bentuk karakter yang akan diproses.

Contoh:

```
<?
//Mengformat karakter
//format_karakter.php
$data="PHP adalah sebuah bahasa pemrograman berbasis
website. Script ini bersifat server-side atau hanya dapat
berjalan pada webserver.";
echo "<b>Data asli.<br>$data</b><br>";
echo "<b>Diubah menjadi huruf kapital semua</b><br>";
$kap=strtoupper($data);
echo "$kap<br>";
echo "<b>Diubah menjadi huruf kecil semua</b><br>";
$kap=strtolower($data);
echo "$kap<br>";
echo "<b>Diubah menjadi huruf kapital diawal
kalimat</b><br>";
$kap=ucfirst($data);
echo "$kap<br>";
echo "<b>Diubah menjadi huruf kapital setiap awal
kata</b><br>";
$kap=ucwords($data);
echo "$kap<br>";
?>
```

Hasil program apabila dijalankan pada browser akan tampak seperti pada gambar berikut :



Gambar 9.26 Hasil penggunaan fungsi pengubah huruf

9.4.19 Fungsi Menghapus spasi pada data string

Untuk menghapus spasi pada sebuah string dapat menggunakan beberapa fungsi sesuai dengan kondisi data string.

- Fungsi **Trim()** adalah fungsi yang digunakan untuk menghilangkan spasi di bagian kiri dan kanan teks.
- Fungsi **Ltrim()** adalah fungsi yang digunakan untuk menghilangkan spasi pada di bagian kiri teks.
- Fungsi **Chop()** adalah fungsi yang digunakan menghilangkan spasi di bagian kanan dan juga menghilangkan karakter-karakter whitespace, seperti : "\n", "\r", "\t", "\v", "\0".

Bentuk adalah :

Trim(string);

Ltrim(string);

Chop(string);

Keterangan :

String adalah argumen sumber data yang apabila mengandung spasi akan dihilangkan spasinya.

Contoh:

```
<?
// Contoh menggunakan fungsi trim(), ltrim(), chop().
// chop.php
$str = " Program PHP ";
echo chop($str);
echo trim($str);
echo ltrim($str);
echo "<br>";
echo ltrim($str);
echo trim($str);
echo chop($str);
?>
```

Hasil program apabila dijalankan pada browser akan tampak seperti pada gambar berikut :



Gambar 9.27 Hasil penggunaan fungsi penghapus spasi

9.5 Fungsi Enkripsi dan Integritas Data

PHP menyediakan banyak sekali fungsi yang digunakan untuk pengamanan data, seperti pembuatan password yang dienkripsi. Fungsi ini digunakan untuk memeriksa integritas data.

9.5.1 Fungsi Crc32

Fungsi ini digunakan untuk menghitung polinomial **Crc 32** terhadap data string argumennya. Fungsi ini biasa digunakan untuk meng-validasi integritas data yang dikirim dari halaman satu dengan halaman yang lain. Bentuk syntaksnya adalah:

Crc32(string);

Keterangan :

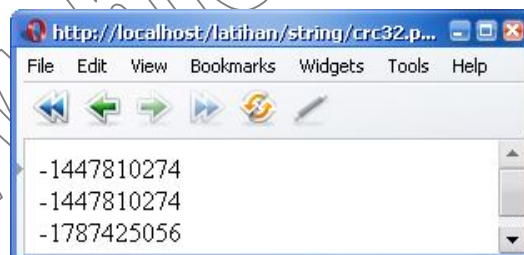
String adalah data sumber yang akan dienkripsi oleh fungsi.

Huruf besar dan huruf kecil akan dibedakan, artinya nilai yang dihasilkan berbeda apabila data string menggunakan huruf besar atau huruf kecil.

Contoh:

```
<?php
//crc32.php
// enkripsi data
$hasil=crc32("userku");
echo"$hasil<br>";
$hasil=crc32("userku");
echo"$hasil<br>";
$hasil=crc32("UserKU");
echo"$hasil<br>";
?>
```

Hasil program apabila dijalankan pada browser akan tampak seperti pada gambar berikut :



Gambar 9.28 Hasil enkripsi dengan fungsi crc32

Apabila string yang dienkripsi sama, maka hasilnya juga sama, namun apabila ada perbedaan walaupun satu huruf juga akan menghasilkan enkripsi yang berbeda.

9.5.2 Fungsi md5

Fungsi **md5** digunakan untuk mengenkripsi string menjadi 32 karakter baik angka maupun huruf dalam bentuk acak. Fungsi ini juga sering digunakan untuk menciptakan data password yang dienkripsi. Bentuk syntaksnya adalah:

Md5(string);

Keterangan :

String adalah data teks yang akan dienkripsi menjadi 32 karakter acak.

Contoh:

```
<?php
//md5.php
//enkripsi data dengan md5
$data="rafiza";
$hasil=md5($data);
echo"Hasil enkripsi dari teks \"$data\" adalah:<br>";
echo$hasil;
?>
```

Hasil program apabila dijalankan pada browser akan tampak seperti pada gambar berikut :



Gambar 9.29 Hasil enkripsi dengan fungsi md5

9.5.3 Fungsi Sha1

Fungsi **sha1** juga berfungsi untuk mengenkripsi string. Bedanya pada fungsi ini hasil enkripsi akan menghasilkan angka dan huruf acak dengan jumlah **41 karakter**. Bentuk syntaksnya adalah:

Sha1(string);

Keterangan :

String adalah data teks yang akan dienkripsi.

Contoh:

```
<?php
//sha1.php
//enkripsi data dengan sha1
$data="rafiza";
$hasil=sha1($data);
echo"Hasil enkripsi dari teks \"\$data\" adalah:<br>";
echo$hasil;
?>
```

Hasil program apabila dijalankan pada browser akan tampak seperti pada gambar berikut :



Gambar 9.30 Hasil enkripsi dengan fungsi sha1