**Manipulasi String, Session dan Cookies pada PHP**

**Session**

Session merupakan hal yang cukup penting dalam aplikasi berbasis web. Dengan session memungkinkan programmer menyimpan informasi user secara semi permanen, artinya selama masa tertentu informasi akan tersimpan. Penyimpanan isi variabel session berada di server, jadi tidak bisa diakses secara langsung oleh client.

Dalam aplikasi berbasis web, session banyak digunakan sebagai autentifikasi login. Dengan session memungkinkan programmer mengatur siapa saja yang bisa mengakses suatu halaman. Misalkan untuk melihat halaman kotak surat pada email, harus terlebih dahulu login. Dalam proses login diantaranya akan terjadi pembuatan suatu session yang akan dibawa oleh user di setiap halaman. Di halaman kotak surat, session tersebut akan diperiksa. Jika session benar maka user dipersilakan untuk membuka halaman kotak surat, namun jika salah maka user tidak bisa membuka halaman kotak surat dan biasanya akan diminta login terlebih dahulu.

Kegunaan session, diantaranya adalah :

1. Menyimpan informasi login yang berlaku hanya dalam satu sesi
2. Menyimpan catatan order barang dalam sistem e-commerce / transaksi online

**Deklarasi Session**

Untuk menggunakan session, harus mengaktifkan dengan cara mendeklarasikan pada bagian awal – awal header dokumen sebelum <html>, berikut penulisan sintaks session :

<?php

session\_start();

?>

Setiap halaman yang menggunakan variabel session harus mendeklarasikan sintaks diatas. Sehingga bisa digunakan, setelah itu baru bisa membuat nama session, berikut sintaks pembuatan nama session.

<?php

$\_SESSION[‘Coba’] = “OKE”;

?>

Session dengan nama coba akan dapat digunakan dalam semua halaman, dengan cara seperti dibawha ini :

<?php

echo $\_SESSION[‘Coba’] ;

?>

Dan jika tidak memerlukannya dapat dihapus dengan cara, sebagai berikut :

<?php

unset ($\_SESSION[‘coba’]); //ini untuk menghapus session tertentu

//jika menghapus semua session yang digunakan oleh user yang

//digunakan

session\_destroy(); //menutup session

?>

Contoh session :

* Proses pembuatan session

<?php

session\_start(); //perintah ini harus ada pada setiap halaman yang

berhubungan dengan session

$\_SESSION[‘seslogin’] = $user;

?>

* Proses pemeriksaan session

<?php

session\_start();

if(isset ($\_SESSION [‘seslogin’]))

{

echo “<h1> selamat datang” .$\_SESSION [‘seslogin’] . “</h1>”;

echo “<h2> halaman ini hanya bisa diakses jika anda sudah login

</h2>”;

echo “<h2> klik <a href = ‘session03.php’> disini (session03.php) </a>

Untuk logout </h2>”;

}

?>

* Proses penghapusan session

<?php

session\_start();

if(isset ($\_SESSION [‘seslogin’]))

{

unset ($\_SESSION);

session\_destroy();

echo “<h1> Anda sudah berhasil LOGOUT</h1>”;

echo “<h2> klik <a href = ‘session01.php’> disini </a> untuk LOGIN

kembali </h2>”;

echo “<h2> anda sekarang tidak bisa masuk ke halaman <a href =

‘session02.php’ </a> lagi </h2>”;

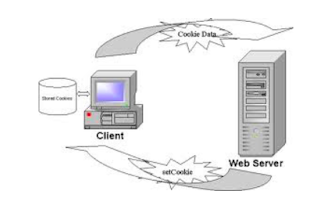
}

?>

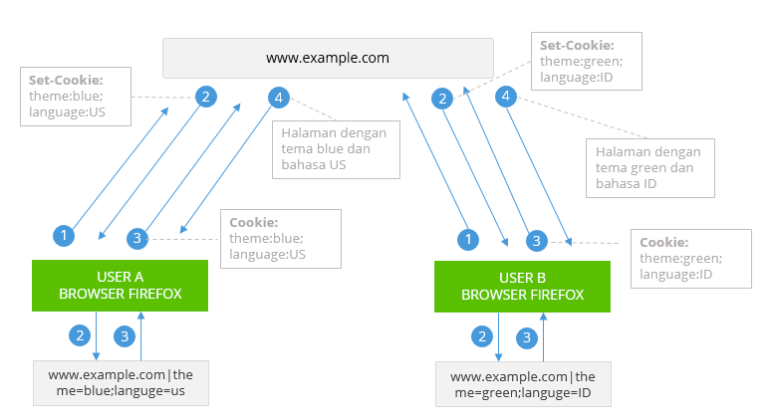
**Cookies**

Seperti halnya session, cookies juga merupakan sebuah konsep penyimpanan informasi user. Jika session tempat penyimpanan berada diserver sedangkan cookies berada di client. Oleh karena itu, konsep cookies sebaiknya jangan digunakan untuk menyimpan informasi login user seperti username dan password dan sebagainya. Selain user bisa melihat informasi yang disimpan, user juga bisa men-disable atau menonaktifkan cookies itu sendiri. Jika cookies di-disable maka program yang memanfaatkan cookies tentunya tidak akan berjalan dengan baik.

Cookies sendiri biasanya dipakai dalam aplikasi shooping cart. Biasa digunakan untuk menyimpan sementara produk – produk yang dipilih oleh pengunjung pada saat berbelanja.



Gambar 1 Cara Kerja Cookies



Gambar 2 Ilustrasi cara kerja cookies

Dalam penanganan cookies juga terdapat beberapa proses yang perlu diperhatikan :

* Proses pembuatan cookies. Contoh :

$value = “A-Zoel”;

$value2 = “Zulkarnaen NS”;

Setcookie (“username”,$value);

Setcookie (“namalengkap”,$value2, time() +3600); //expire in 1 hour

* Proses pemeriksaan cookies. Contoh :

if(isset ($\_COOKIE[‘username’]))

{

echo ”<h1> cookie username ada isinya “ . $\_COOKIE[‘username’] .

“ < / h1> “;

}

else

{

echo “<h1> cookie yang bernama username TIDAK ADA </h1>”;

}

* Proses penghapus cookies. Contoh :

setcookie (“username”,” “, time() -3600);

setcookie (“NamaLengkap”, “ “, time() -3600);

Selain menggunakan script PHP seperti diatas untuk menghilangkan cookie juga dapat langsung dari browser yang sedang digunakan. Jika menggunakan internet explorer, pilih ***tools > internet options***, kemudian tekan delete cookie, maka cookie akan terhapus. Jika menggunakan firefox pilih menu ***tool > clear private data*** atau dengan menggunakan tombol ***ctrl + shift + del***. Kemudian centang cookie, kemudian klik ***clear private data now***.

**Kegunaan cookies, diantaranya adalah :**

1. Menyimpan username dan password login agar pengguna tidak selalu harus mengisikannya pada saat membuka halaman. Biasanya untuk memberitahukan menggunakan checkbox “remember me” atau “ingat saya” yang diletakkan pada layar tampilan login
2. Untuk mencatat konfigurasi yang dilakukan oleh pengguna, seperti warna tema, jenis huruf, pilihan bahasa, dan lain – lain
3. Untuk mengetahui apakah pengunjung pernah datang atau belum ke halaman web yang sedang dibuka (seperti digunakan dalam hit counter / penghitung pengunjung).

**Jenis – jenis cookies, diantaranya :**

1. **Non persistent (session) cookies** : suatu cookie yang akan hilang sewaktu user menutup browser dan biasanya digunakan pada shopping carts di toko belanja online untuk menelurusi item – item yang dibeli.
2. **Persistent cookies** : diatur oleh situs – situs portal, banner / media iklan situs dan lainnya yang ingin tahu ketika user kembali mengunjungi site mereka. Misalkan dengan cara memberikan opsi “remember me” saat login). File – file ini tersimpan di hardisk user.

**Mana yang harus digunakan session atau cookies ?**

Session atau cookie adalah pilihan menggunakan teknologi untuk menyimpan data. Dengan menggunakan cookie berarti data akan disimpan di komputer client, berarti komputer client bisa memodifikasi data tersebut (jika tidak menggunakan enkripsi) dan keamanan server akan terancam. Tapi jika menggunakan enkripsi untuk menyamarkan data, hal itu tidak menjadi masalah selama enkripsi yang digunakan masih ampuh. Contoh: google sudah mulai menggunakan cookie yang menyimpan data login.

Kelebihan cookie dibanding session, yaitu bisa diakses oleh javascript. Jadi selama ini beranggapan bahwa javascript tidak bisa berhubungan dengan PHP adalah kurang tepat karena cookie bisa menjadi penghubung antara PHP dan javascript. Sedangkan session disimpan di server, artinya semakin banyak session yang diset, maka semakin banyak juga resource yang dibutuhkan. Mungkin hal ini juga yang menjadi pertimbangan google untuk memutuskan menggunakan cookie. Sebagai ganti dari resource tersebut, session lebih aman dibanding cookie, karena letaknya di server.

**Perbedaan session dan cookies**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Menyimpan informasi | Tempat menyimpan | Bisa dipanggil | Bersifat global | Dipanggil lain waktu | Dibatasi browser | Dapat diedit user |
| Session | Ya | Server | Ya | Ya | Tidak | Tidak | Tidak |
| Cookies | Ya | Klien | Ya | Ya | Ya | Ya | Ya |

**Manipulasi String Menggunakan Fungsi di PHP**

**Pengantar string**

String merupakan kumpulan dari karakter. Dalam PHP, karakter sama dengan byte, dimana terdapat 256 karakter. PHP tidak mendukung native – Unicode. Untuk menuliskan sebuah string dalam PHP, bisa menggunakan 3 (tiga) cara yaitu sebagai berikut :

1. Single quoted – kutip tunggal ( ‘ )
2. Double quoted – kutip ganda ( “ )
3. Heredoc sintaks

**Fungsi – fungsi string dalam PHP**

Php menyediakan lebih dari 90 fungsi untuk manipulasi string. Berikut fungsi – fungsi manipulasi string yang digunakan, antara lain :

* **addslashes**() – menambahkan backslashes (\) di setiap tanda kutip (quote) dalam string.
* **chr**() – menghasilkan karakter dari bilangan ASCII
* **crypt**() – menghasilkan string terenkripsi searah.
* **echo**() – menampilkan satu atau beberapa string.
* **explode**() – memecah string berdasarkan suatu delimiter (separator).
* **htmlentities** () – mengubah semua tag html menjadi HTML entities.
* **htmlspecialchars**() – mengubah semua karakter khusus menjadi HTML entities
* **implode**() – menggabungkan elemen array menjadi string dengan suatu delimiter (separator).
* **join**() – sama dengan implode()
* **ltrim**() – menghapus karakter tertentu (“ “, \t, \n, \r, \0, \x0B) di awal string.
* **md5**() – menghasilkan string terenkripsi searah.
* **nl2br**() – menambahkan HTML line break (<BR>) sebelum semua baris baru dalam string.
* **number\_format**() – mengatur format angka.
* **ord**() – menghasilkan kode ASCII dari suatu karakter.
* **rtrim**() - menghapus karakter tertentu (“ “, \t, \n, \r, \0, \x0B) di akhir string.
* **split**() – sama dengan explode
* **str\_repeat**() – mengulang string
* **str\_replace**() – mengganti semua string dalam pola menjadi suatu string.
* **strip\_tags**() – mengabaikan HTML dan PHP tag dalam string.
* **stripslashes**() – menghilangkan backslashes (\) dalam string.
* **strlen**() – menghitung panjang string.
* **strpos**() – mencari posisi pertama sebuah string dalam string.
* **strrchr**() – mencari posisi terakhir sebuah karakter dalam string.
* **strrpos**() – mencari posisi terakhir sebuah string dalam string.
* **strrev**() – membalik string.
* **strstr**() – mencari posisi pertama sebuah karakter dalam string.
* **strtolower**() – mengubah string menjadi huruf kecil (lower-case).
* **strtoupper**() – mengubah string menjadi huruf kapital (upper-case)
* **substr**() – memotong string
* **trim**() - menghapus karakter tertentu (“ “, \t, \n, \r, \0, \x0B) di akhir dan akhir string.
* **ucfirst**() – mengubah huruf pertama dari semua string menjadi Upper-case.
* **ucwords**() – mengubah huruf pertama tiap kata dalam string menjadi upper- case.
* **wordwrap**() – memotong sejumlah karakter dalam string dengan *break-character.*

Terdapat beberapa perintah manipulasi string yang terdapat di PHP, diantaranya :

1. Menghitung jumlah karakter dengan strlen()

Strlen() digunakan untuk menghitung mencari jumlah string atau karakter. Dalam fungsi ini dapat mengetahui jumlah karakter pada suatu kalimat atau kata dan spasi juga dapat dihitung sebagai karakter.

Contoh:



1. Menghitung jumlah kata pada string dengan str\_word\_count()

Str\_word\_count() adalah fungsi untuk menghitung banyaknya kata dalam sebuah string. Fungsi ini hanya menghitung jumlah kata yang terpisah oleh spasi dalam sebuah string.

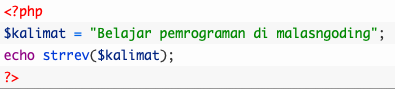
Contoh :



1. Membalik string dengan strrev()

Strrev() merupakan fungsi untuk membalikkan kata pada string.

Contoh:



1. Mencari kata tertentu dalam string dengan strpos()

Strops() adalah fungsi untuk memcari kata dalam string, fungsi ini juga sering disebug dengan fungsi pencarian, akan tetapi return valuenya berupa sebuah angka tempat karakter itu berada dalam sebuah string atau text.

Contoh :



1. Memfilter dan mengganti string dengan str\_replace()

Str\_replace() adalah fungsi yang digunakan untuk mengganti text dan sering disebut juga sebagai replace function.

Contoh:



1. Mengubah string menjadi lower – case dengan strtolower()

Strtolower adalah perintah PHP untuk memanipulasi sebuah string menjadi huruf kecil. Sehingga dapat membantu apabila dalam sebuah string terdapat huruf besar dan kecil yang tidak sesuai. Maka dengan perintah ini bisa mengubah menjadi huruf kecil.

Contoh :



1. Mengubah string menjadi upper-case dengan strtoupper

Fungsi strtoupper adalah suatu perintah yang ada pada PHP untuk membuat suatu string menjadi huruf kapital. Strtoupper ini merupakan kebalikan dari perintah strtolower.

Contoh :



1. Mengubah huruf pertama string menjadi uppercase dengan ucfirst

UC yang berarti uppercase atau dalam Bahasa Indonesia Teks tebal, yang bersifat hanya melihat awal karakter dari setiap kata yang telah di input. Fungsi ini sama dengan perintah ucwords, namun hasil akhirnya saja yang berbeda. Jika pada ucwords merubah setiap awal kata pada string, sedangkan ucfirst ini merubah awal kata pada kata pertama saja di dalam string.

Contoh :



1. Mengubah huruf pertama tiap kata menjadi uppercase dengan ucwords

Ucwords adalah suatu fungsi php yang bertugas untuk memanipulasi string. Setiap awalan kata pada string tersebut akan berubah menjadi huruf kapital.

Contoh :



1. Mengubah huruf pertama tiap kata menjadi lowercase dengan lcfirst()

Lcfirst adalah fungsi php yang dapat memanipulasi string pada umumnya. Jika ucword untuk membuat string menjadi huruf besar pada setiap awal kata dan ucfirst merubah huruf awal kata pertama saja yang menjadi besar. Sehingga, lcfirst adalah kebaliknya yaitu mengubah string pada awal kata menjadi huruf kecil.

Contoh :



**Fungsi – fungsi operasi tanggal dalam PHP**

Fungsi dalam operasi tanggal dalam PHP yang utama dan sering digunakan adalah fungsi date(). Fungsi ini menghasilkan tanggal dan waktu server sekarang. Beberapa pilihan parameter dari fungsi date() ini adalah sebagai berikut :

