

Membuat Framework Class dengan OOP PHP – bagian 2

Benediktus Ardian Hersanto, SE
ardie_b@yahoo.com
<http://ardieb77.multiply.com>
<http://profiles.friendster.com/ianbenedict>

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.



Pendahuluan

Pada sesi bagian dua ini kita akan mempelajari tentang bagaimana sebuah session bekerja pada setiap pengunjung web. Pada saat web kita dikunjungi oleh para pengguna internet, secara otomatis php akan menginisialisasi identitas komputer pengunjung dengan membuat cookies di browser bernama *PHPSESSID*. Nama cookies "*PHPSESSID*" ini dapat anda rubah sendiri di file *php.ini* bagian session jika anda ingin menggunakan nama yang lain. Fungsi dari cookies ini adalah untuk proses mengidentifikasi browser pengunjung dan mensynchronize data-data input yang dikirimkan pengunjung dengan file session yang tersimpan pada server, dimana isi dari cookie *PHPSESSID* ini adalah berupa karakter hash (finger print ID) yang biasanya berupa output karakter dari hasil MD5. Sebenarnya PHP sendiri memberikan kebebasan kepada kita untuk memodifikasi fungsi session bawaan dari PHP.

Isi

Jika anda sudah mempelajari contoh framework yang ada di attachment bagian satu, anda dapat menemukan class session pada direktori *lib/php4/session.lib.php*.

Di class session tersebut anda dapat melihat 3 macam mode session yaitu dengan menggunakan default session dari PHP, file session yang dibuat sendiri, dan yang ketiga session yang diletakkan di database server (note : untuk yang didatabase server blm sempat saya buat). Pada artikel ini yang akan saya bahas adalah cara membuat session yang diletakkan pada file sendiri (mode kedua). Pertama-tama dalam membuat session, kita harus membuat tanda pada browser pengunjung dengan meletakkan cookie di browsernya. Identitas session ID ini dapat di generate dengan mengambil variable environment yang didapat dari identitas pengunjung kemudian dari kumpulan identitas tersebut kita bundle jadi satu dan kita buat string signature dengan MD5.

```
$this->directory = ROOT_DIR.SESS_TMP_DIR; // didapat dari configuration file config.ini.php
$cookie = $this->_getCookie($this->sess_name); // check apakah cookie PHPSESSID sudah ada
if(!empty($cookie)) $this->id = $cookie;
if(!USE_PHP_DEFAULT_HANDLER) { // didapat dari configuration file config.ini.php
    if(empty($cookie)&&($this->type==SESS_TYPE_FILE)) $this->generate_id();
    else $this->sess_file = $this->directory. "/sess_". $cookie;
}
// fungsi untuk membuat cookie signature identity pada browser pengunjung
function _generate_id() {
    $this->id=
    md5(microtime().getenv('HTTP_ACCEPT_CHARSET').getenv('HTTP_ACCEPT_ENCODING').getenv('HTTP_A
CCEPT_LANGUAGE').getenv('REMOTE_ADDR').getenv('HTTP_USER_AGENT').getenv('REMOTE_HOST').get
env('REMOTE_PORT'));
    return $this->id;
}
```

Fungsi `_generate_id()` diatas dapat kita lihat, semua isi dari variable environment diambil dari identitas browser pengunjung, kemudian untuk lebih uniknya kita tambahkan waktu aksesnya yaitu `microtime()`, mungkin ada yang bertanya-tanya kenapa tidak menggunakan fungsi `time()` karena fungsi `microtime` lebih detail (micro seconds). Harap diperhatikan untuk class session yang kita buat ini, fungsi `session_start()` bawaan dari PHP tidak akan digunakan karena fungsi tersebut hanya berlaku jika kita set jenis sessionnya adalah session default PHP, jadi yang kita buat diatas adalah fungsi pengganti dari `session_start()`.

Setelah kita membuat cookies pada browser pengguna langkah berikutnya adalah kita membuat file temporary yang digunakan untuk menampung variabel-variabel session dari pengunjung.

File temporary ini dibuat pada saat kita meregisterkan variable-variable session ke server, perintah register variable ini kemudian akan mengecek apakah file bernama `sess_[md5_session_id]` ada di dalam direktori tempat session temp (lihat di file config), jika ada simpan variabel-variabel session ke file tersebut.

```
function register($name) {  
    if($this->type==SESS_TYPE_FILE) {  
        $fid = fopen($this->directory . "/sess_" . $this->_getCookie($this->sess_name), "w");  
        if(!$fid) {  
            fwrite($fid, serialize($GLOBALS[$this->_varname]));  
        }  
        fclose($fid);  
    }  
}
```

Fungsi register diatas adalah versi simplenya jika kita membuat temporary file session, untuk yang ada di file *session.lib.php* fungsi register session sudah memiliki data type filtering, jadi setiap tipe data berupa array atau objek dapat di simpan ke dalam file session dengan mengkonversi data dengan perintah *serialize*. Dan yang terakhir apabila pengunjung tersebut logout maka class session akan menghapus file sess_xxxxxxxx dari temporary directory.

Dari penjelasan ini mungkin anda sudah bisa menangkap bagaimana sebuah session bekerja pada suatu website. Mungkin dari class-class yang saya jelaskan disini hanya untuk sebuah gambaran tentang bagaimana pemrograman berorientasi objek akan sangat membantu kita dalam mengurangi kegiatan coding terhadap prosedur-prosedur database yang monoton. Agar kita lebih dapat berkonsentrasi pada business processnya saja dan fungsionalitas web.

Mengenai class-class framework yang lainnya saya tidak akan mengulas lagi disini, anda dapat mendownload contoh frameworknya dari code2art.com, framework ini terbuka bagi siapa saja yang ingin menambahkan fungsi di framework classnya. Mungkin sebelumnya mohon maaf jika website tersebut masih kosong pada saat anda membaca tulisan ini, karena saya masih belum mempunyai waktu untuk membuat pagennya. Trims..... Salam Developer.

Penutup

Tidak ada gading yang tak retak, apabila anda masih belum mengerti mengenai penjelasan saya, anda dapat bertanya melalui email. Tetapi tidak ada sesuatu yang dapat dikuasai tanpa memerlukan proses pembelajaran (learning by doing), ketekunan dan kesabaran. Anda pasti bisa karena anda adalah manusia juga seperti saya, dan semuanya memerlukan suatu proses yang disebut dengan waktu. Ingatlah bahwa hidup tidak hanya didepan komputer, tetapi hidup tidak akan bervariasi tanpa pengetahuan. Selamat belajar.

Referensi

<http://www.google.com>



Biografi penulis

Benediktus Ardian Hersanto, SE. Adalah lulusan S1 Manajemen Keuangan STIE Perbanas Surabaya angkatan 1995. Berkecimpung dalam dunia IT sejak tahun 1999, tetapi sebelumnya sempat belajar elektronika digital di IEI (surabaya) pada tahun 1997-1998. Di dunia IT sendiri penulis belajar secara otodidak dari internet. Pernah bekerja sebagai Web Designer, Server Administrator, System Programmer, dan pernah mendirikan perusahaan IT pada tahun 2002 - 2005. Saat ini penulis bekerja di salah satu perusahaan IT di Jakarta sebagai R & D.

Benediktus Ardian Hersanto, SE

URL : <http://ardieb77.multiply.com>

URL : <http://profiles.friendster.com/ianbenedict>

Email & YM : ardie_b@yahoo.com