

MAKALAH
JARINGAN KOMPUTER
“Session Model OSI”



Nama Kelompok : Agung P D (10110303)
Hendri Gunawan (13110222)
Johanes (13110766)
Wahyu Hargi Atmanto (18110425)
Willy Badawi Hardjono (18110519)
Yosep Wira Nugroho (18110687)

Kelas : 3KA07

Jurusan : Sistem Informasi

Fakultas : Ilmu Komputer

Dosen : Missa Lamsani

UNIVERSITAS GUNADARMA
DEPOK
2013

DAFTAR ISI

COVER	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 TUJUAN PENULISAN	1
BAB II PEMBAHASAN	2
2.1 PENGERTIAN SESSION.....	2
2.2 KONSEP DASAR SESSION	2
2.3 PERINTAH-PERINTAH DALAM SESSION	4
BAB III KESIMPULAN	10
DAFTAR PUSTAKA	12

BAB I

PENDAHULUAN

1.1 Latar Belakang

Model referensi jaringan terbuka OSI atau OSI Reference Model for open networking adalah sebuah model arsitektural jaringan yang dikembangkan oleh badan International Organization for Standardization (ISO) di Eropa pada tahun 1977. OSI sendiri merupakan singkatan dari Open System Interconnection. Model ini disebut juga dengan model "Model tujuh lapis OSI" (OSI seven layer model). Model Referensi OSI adalah Model konvensional yang dapat digunakan untuk membandingkan rangkaian, dan karakteristik umum rangkaian protokol jaringan adalah model referensi OSI.

Terdapat 7 layer pada model OSI. Setiap layer bertanggung jawab secara khusus pada proses komunikasi data. Misal, satu layer bertanggungjawab untuk membentuk koneksi antar perangkat, sementara layer lainnya bertanggung jawab untuk mengoreksi terjadinya "error" selama proses transfer data berlangsung.

1.2 Tujuan Penulisan

Tujuan dari penulisan ini adalah agar memberitahukan kepada pembaca tentang fungsi dari Layer Session

BAB II PEMBAHASAN

2.1 Pengertian Session

Session adalah suatu cara agar suatu variable dapat diakses dibanyak halaman web. Session biasanya berupa file yang tersimpan di server, berbeda dengan Cookies yang disimpan di client.

Session bersifat unqi dari setiap pengunjung. jadi seandainya pengunjung bernama "lioniel" maka session yang dimiliki lioniel adalah 1 dan selanjutnya begitu. unqi id akan dikirim dan disimpan pada cookies atau pada URL.

Secara umum, session digunakan untuk menyimpan suatu informasi antar proses request, baik request dalam bentuk POST atau GET. Session adalah variable yang dapat mempertahankan isinya dalam periode waktu tertentu (bisa diset) dan disimpan diserver. Meskipun berpindah-pindah halaman (dalam 1 website tersebut) variable masih bertahan. Jadi session hanyalah sebuah variable dengan karakteristik tertentu.

PHP mempunyai session (catatan aktivitas) yang digunakan untuk menjaga / memelihara informasi akses dari seorang pengakses / pemakai aplikasi web. Session memungkinkan pelacakan akses pemakai, pengaturan pemakaian aplikasi oleh pemakai dan meningkatkan layanan situs web. Setiap pengunjung akan diberi sebuah id yang unik, yang disebut dengan id session (session_id). ID ini dapat disimpan dalam suatu cookie pada sisi user atau disertakan dalam URL. Session koneksi antara klien dan server akan hilang atau putus apabila browser ditutup. Apabila browser dijalankan kembali dan koneksi ke server dilakukan maka dianggap sebagai koneksi baru.

Salah satu contoh yang menggambarkan penggunaan session adalah proses login. Dalam hal ini user akan memasukkan usernamenya melalui form login. Setelah login berhasil, user tersebut dihadapkan pada link menu navigasi yang menuju ke beberapa halaman web. apabila user memasukkan username maka akan selalu tampil atau tercatat di halaman-halaman web tersebut, maka username haruslah disimpan dalam session.

2.2 Konsep Dasar Session

Session bekerja seperti tempat penitipan barang di supermarket. Pembeli datang, menitipkan barang bawaan, dan petugas akan memberikan tanda pengenal. Pada saat pembeli

akan mengambil barang, pembeli memberikan tanda pada petugas, dan petugas akan dapat mengambil barang yang dititipkan tanpa tertukar dengan barang orang lain. Permasalahan yang terjadi adalah, bagaimana “tanda pengenal” tersebut akan disimpan pada setiap halaman yang diakses user, sehingga dapat diakses oleh sistem setiap kali user mengklik link.

Ada tiga alternatif yang dapat digunakan untuk menyimpan data session :

- Cookie
- Hidden Form
- URL Embedding

Secara umum, cara kerja session dapat dideskripsikan seperti Saat pengunjung mengakses suatu halaman, pengunjung tersebut memulai session (digunakan fungsi `session_start()`). Bila dalam konfigurasi php, `session.auto_start` diset ke 1 maka php akan melakukan start session secara otomatis.

Pengunjung tadi kemudian diberi pengenal berupa session id, Session id ini adalah rangkaian karakter yang unik, misalnya 12345678xx.. Bersamaan dengan itu, pada sisi server juga dibuat file `ses_12345678xx` yang berisi variabel-variabel session milik si pengunjung. Variabel yang diregister dengan fungsi `session_register()` akan disimpan dalam file ini.

Metode di atas adalah default. Tetapi bisa jug menentukan lokasi penampungan variabel session sendiri, misalnya di database. Saat pengunjung memulai atau melanjutkan session (melalui `session_start()`) maka pengunjung mengirim session id miliknya untuk memperoleh kembali variabel-variabel session miliknya yang telah disimpan di server.

Pengiriman session id ke server bisa melalui 2 cara. Yang pertama, yaitu cara default, dikirim melalui cookies. Session id berupa cookie `PHPSESSID` (default). Ada kalanya dengan alasan tertentu, pengunjung menonaktifkan cookies (bisa dilakukan melalui setting browsernya). Bila hal ini terjadi, maka pengiriman session id dilakukan dengan cara kedua, yaitu melalui query string (string yang ditambahkan ke URL, seperti contoh di awal artikel).

Session akan berakhir saat fungsi `session_destroy()` dipanggil atau sewaktu user menutup web browsernya. Dari cara kerja session tadi, terlihat bahwa yang menjadi kunci pegangan si pengunjung (di sisi client) hanyalah variabel session id, sedangkan variabel-variabel lain miliknya di server. Adanya fasilitas session handling di PHP menjadikan proses diatas menjadi sederhana dan mudah karena PHP melakukannya otomatis.

Jadi, 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. Misalnya saja, untuk melihat halaman kotak surat pada email, kita harus login terlebih dahulu. Dalam proses login antara lain 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 dipersilahkan membuka halaman kotak surat, namun jika salah maka user tidak bisa membuka halaman kotak surat dan biasanya akan diminta untuk login terlebih dahulu. Itulah sebabnya, user tidak bisa mengakses halaman kotak surat secara langsung tanpa melakukan login.

Dalam penanganan session terdapat beberapa proses yang perlu diperhatikan :

- Proses pembuatan session
- Proses pemeriksaan session
- Proses penghapusan session

2.3 Perintah- perintah dalam session

Berikut ini adalah perintah-perintah yang terdapat dalam session :

1. `Session_start()`, berfungsi untuk memulai / mengaktifkan session.
2. `Session_register()`, berfungsi untuk mendaftarkan suatu variable ke dalam session. Parameter dari fungsi ini adalah nama variable yang akan disimpan di session.
3. `Session_unregister()`, berfungsi untuk menghapus suatu variable yang disimpan di session.
4. `Session_is_registered()`, berfungsi untuk memeriksa apakah suatu variable ada / terdaftar dalam session.
5. `Session_unset()`, berfungsi untuk menghapus semua variable yang terdaftar dalam session.

6. `Session_destroy()`, berfungsi untuk menutup / menghapus session beserta file sessionnya.

Setelah sedikit penjelasan diatas maka akan dijelaskan lebih lanjut tentang perintah-perintah dari session ini.

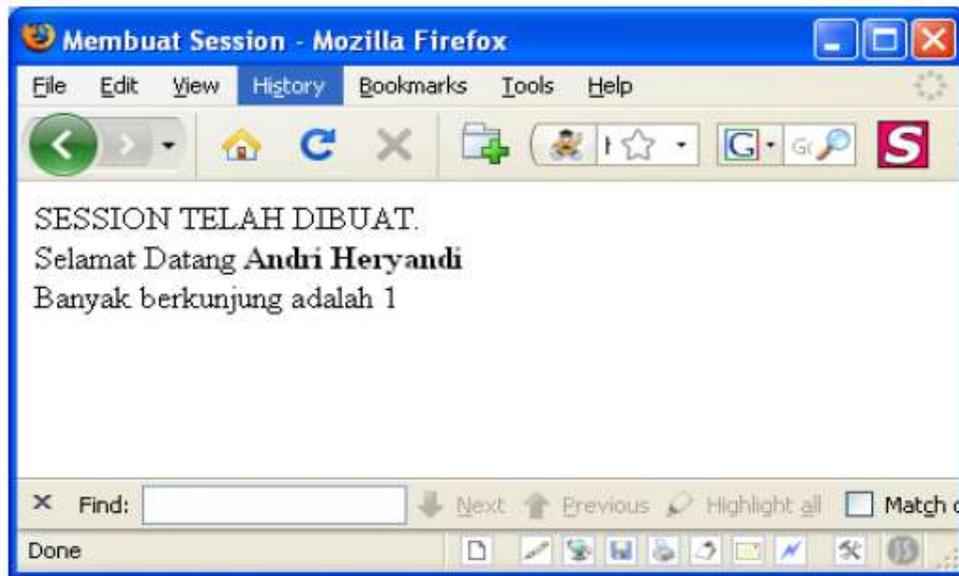
1. `Session_start()`

`Session_Start()` hanya dapat dilakukan di awal file (jangan ada 1 karakter pun ter-echo-kan ke browser sebelum `session_start()`).

Contoh :

Contoh Benar	Contoh Salah
<pre>1 <?php 2 session_start(); 3 ?></pre>	<pre>1 <html> 2 <?php 3 session_start(); 4 ?></pre> <p>Ada <html> sebelum session_start()</p> <pre>1 2 <?php 3 session_start(); 4 ?></pre> <p>Ada baris (enter) sebelum session_start()</p>

```
<?php
    session_start();
    // Buat session (jika belum ada) atau Buka Session (jika sudah ada)
    session_register("nama", "banyakberkunjung");
    // Daftarkan 2 buah variable ke session yaitu nama dan banyakberkunjung
    $_SESSION['nama']="Andri Heryandi";
    $_SESSION['banyakberkunjung']=1;
?>
<html>
<head><title>Membuat Session</title></head>
<body>
SESSION TELAH DIBUAT. <br>
Selamat Datang <b><?php echo $_SESSION['nama'];?></b><br>
Banyak berkunjung adalah <?php echo $_SESSION['banyakberkunjung'];?>
</body>
</html>
```



2. Session_register()

Pembuatan session dengan session start akan menghasilkan file session dengan nama sess_ diikuti oleh nilai session_id. Session dapat juga dibuat dengan fungsi session_register();

Penggunaan session_register(); memungkinkan penyimpanan variable dan nilainya dalam file.

Contoh script:

nama file tamu.html

```
<html>
```

```
<head>
```

```
<title>Buku tamu</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset:iso-8859-1">
```

```
</head>
```

```
<body>
```

```
<h1>Selamat datang</h1>
```

```
Silakan isi buku tamu:<br/>
```

```
<form name="tamu" method="post" action="sesreg.php">
```

```
<pre>
Nama: <input type="text" name="nama">
Email: <input type="text" name="email">
<input type="submit" value="kirim">
</pre>
</form>
</body>
</html>
```

nama file sesreg.php

```
<?php
session_start();

session_register("nama", "email", "waktu");

?>

<html>
<head>
<title>Penggunaan session register</title>
</head>
<body>
<?php
$waktu = date("H : i");

echo("<h1>Hallo $_POST['nama']</h1>");

echo("<h1>Email anda $_POST['email']</h1>");

echo("<h1>Terima kasih anda telah mengisi buku tamu kami pada jam:
$waktu</h1>");

?>
```

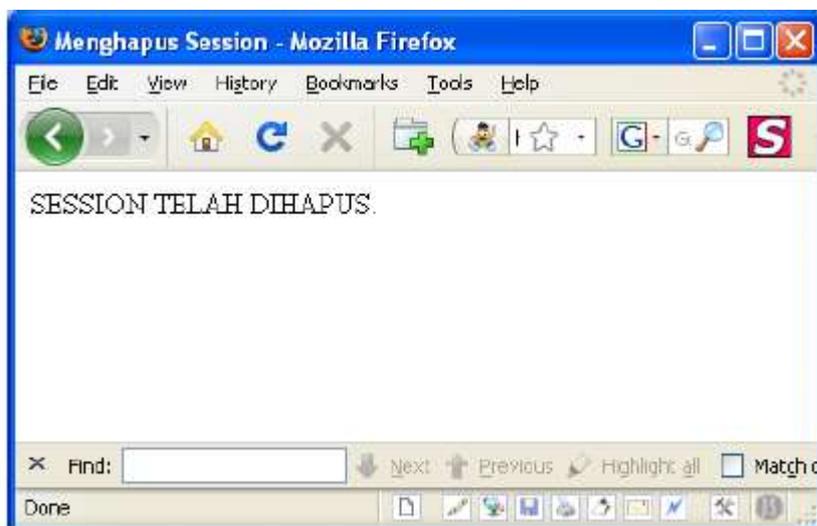
Untuk menghapus session, gunakan Session_unregister, Session_unset, atau Session_destroy. Untuk menghapus session, maka session harus diaktifkan dulu dengan

menggunakan `session_start()`. **Jika ingin hanya menghapus sebuah variable session, gunakan `Session_unregister` (“namavariabel”).** Jika ingin menghapus seluruh variable session, gunakan `session_unset()`. *Jika ingin menghapus seluruhnya (file session), gunakan `session_destroy()`.*

Contoh :

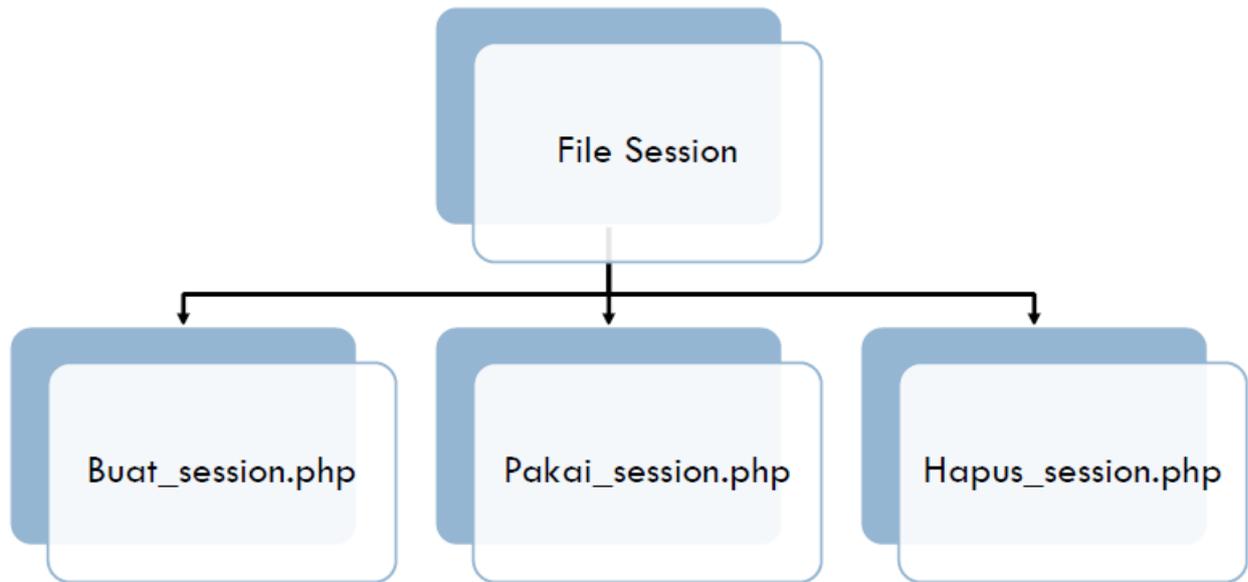
Buat sebuah file `hapus_session.php`

```
<?php
    session_start();// aktifkan session
    session_destroy();// hapus file session
?>
<html>
<head><title>Menghapus Session</title></head>
<body>
SESSION TELAH DIHAPUS.<br>
</body>
</html>
```



Struktur pembuatan, pemakaian, dan penghapusan session

Secara garis besar, session dapat dibagi menjadi tiga bagian besar, yaitu pembuatan session, pemakaian dan penghapusan session. Untuk lebih jelas dapat dilihat pada gambar dibawah ini.



Aplikasi yang menggunakan session salah satunya adalah situs yang menggunakan login. Setelah seseorang melakukan login, maka nama user dan status loginnya dapat dibaca disetiap halaman web. Jika suatu halaman diakses, tetapi sessionnya tidak memiliki status login maka dianggap belum login. Sistem harus menampilkan layar “Anda harus login terlebih dahulu untuk mengakses situs ini”.

BAB III

KESIMPULAN

1. Tempatkan fungsi `session_start()` dibagian paling atas coding php, tidak boleh didahului oleh coding apapun.
2. Perhatikan penamaan variable session ketika `session_register()` huruf besar kecilnya (case sensitive), karena ketika penggunaan variable session tersebut harus sama jenis hurufnya. (Sebenarnya bukan hanya variable session saja yang bersifat case sensitive tapi juga penamaan variable lainnya di PHP sifatnya case sensitive, termasuk penamaan komponen misalnya name textfiled, textarea, list/menu dll).
3. Setiap kali akan menggunakan variable session, maka harus dituliskan `session_start()` dibagian paling atas coding.
4. Ketika metode pengiriman variable menggunakan metode post, maka ketika menerima/membaca variable nya pun harus post. Begitu pula ketika metode nya get, maka menerimanya juga harus get.

Misalkan : formlogin.php

```
.. form action="login.php" method="post" name="form1 " ..
```

```
.....
```

```
.. input name="txtuser" type="text" class="LongText" ..
```

```
.. input name="txtpass" type="text" class="LongText" ..
```

```
.....
```

login.php

```
session_start();
```

```
if ( (! empty(trim($HTTP_POST_VARS['txtuser'])) &&  
(!empty(trim($HTTP_POST_VARS['txtpass']))) )
```

```
{
```

```
session_register('SES_USER');
```

```
session_register('SES_PASS');  
  
$HTTP_SESSION_VARS['SES_USER'] = $HTTP_POST_VARS['txtuser'];  
  
$HTTP_SESSION_VARS['SES_PASS'] = $HTTP_POST_VARS['txtpass'];  
  
}
```

.....

Keterangan :

Karena di atas kita menggunakan method Post, maka ketika menerima/membaca variabelnya kita menggunakan \$HTTP_POST_VARS[], jika di atas anda menggunakan method Get, maka anda harus menggunakan \$HTTP_GET_VARS[].

DAFTAR PUSTAKA

- <http://mafika.forumotion.net/t208-sedikit-tentang-session>
- [https://docs.google.com/viewer?a=v&q=cache:hZA7H7C_RdQJ:blog.unsri.ac.id/userfiles/09071003056\(3\).doc+session+adalah&hl=en&pid=bl&srcid=ADGEEsjZ76AftAljYC9A_Ryg-cm6NeLVSV3OiW7I52YI0ryQPGELQEr1obogbpsG3gUHU0QGzxMsEVoZWrbK1ADTHoxe75dj0vqGuYOUNVcRWMsr6LKCd9ufYkwcklxSjWJ9ChAqULil&sig=AHIEtbSgOKkLDKrmboDgPL9EETO OXGaEnQ](https://docs.google.com/viewer?a=v&q=cache:hZA7H7C_RdQJ:blog.unsri.ac.id/userfiles/09071003056(3).doc+session+adalah&hl=en&pid=bl&srcid=ADGEEsjZ76AftAljYC9A_Ryg-cm6NeLVSV3OiW7I52YI0ryQPGELQEr1obogbpsG3gUHU0QGzxMsEVoZWrbK1ADTHoxe75dj0vqGuYOUNVcRWMsr6LKCd9ufYkwcklxSjWJ9ChAqULil&sig=AHIEtbSgOKkLDKrmboDgPL9EETO OXGaEnQ)