

## Bekerja dengan Session

**Session PHP** adalah sebuah variabel yang dapat digunakan untuk menyimpan informasi atau nilai tertentu. Session PHP juga mendukung sistem data berlapis atau array sehingga memungkinkan untuk menampung beragam informasi dalam satu nama session. Masing-masing session yang sudah dibuat hanya berlaku untuk satu pengguna yang sedang aktif, session ini bersifat global dan dapat dipanggil setiap saat pada semua halaman php yang digunakan.

Setiap pengguna yang menggunakan halaman php yang ada variabel sessionnya, secara otomatis memiliki kode unik session yang akan otomatis hilang setelah pengguna menutup/meninggalkan halaman php.

Session sangat berperan penting dalam penanganan hak akses terhadap user dalam suatu sistem web. Session digunakan untuk menyimpan suatu informasi antar proses request, baik request dalam bentuk POST atau GET.

Kita ambil sebuah contoh dari penggunaan session misalnya pada saat proses login. Dalam proses login, user memasukkan username nya melalui form login. Setelah login berhasil, user tersebut dihadapkan pada link menu navigasi yang menuju ke beberapa halaman web. Untuk membatasi hak akses user terhadap halaman web tertentu, maka username tadi haruslah disimpan dalam session.

### 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 dan jangan ada 1 karakter pun ter-echo-kan ke browser sebelum session\_start().

```
1 <?php
2 // Membuat data session
3 session_start();
4
5 echo "<br>Selamat datang di halaman session...<br>";
6 echo "ID session Anda adalah : ".session_id();
7 ?>
```



ID session tersebut akan hilang atau terhapus jika browser ditutup. Dan akan dibuat lagi ketika kita membuat browser lagi dengan ID session yang berbeda.

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.

**Catatan:**

Untuk saat ini penggunaan session sudah di handle variable SUPERGLOBAL:

**`$_SESSION['nama_variable_session'] = nilai_variable;`**

Jadi penggunaan `session_register();` dapat diganti dengan `$_SESSION`. Dan jika memakai fungsi-fungsi session selain `session_start()` dan `session_destroy()` (mgkn msh ada lagi) tidak boleh di barengi dengan pemakaian `$_SESSION`.

**Contoh script:**

```
1 <?php
2     session_start();
3     //session_register("nama");
4     //session_register("alamat");
5
6     $_SESSION['nama']="Tukul Arwana";
7     $_SESSION['alamat']="JL. Bukan Empat Mata";
8
9 >?
10 <html>
11 <head><title>Membuat Session</title></head>
12 <body>
13     SESSION ANDA TELAH DIBUAT!<br>
14     Data Session Anda adalah :<br>
15     Nama      :<b><?php echo$_SESSION['nama'];?></b><br>
16     Alamat    :<b><?php echo$_SESSION['alamat'];?></b><br>
17 </body>
18 </html>
```



SESSION ANDA TELAH DIBUAT!  
Data Session Anda adalah :  
Nama :Tukul Arwana  
Alamat :JL. Bukan Empat Mata

**TUGAS :**

Buatlah form untuk membuat session yang berisi informasi username, password dan level

**Menghapus Session**

Menghapus session biasanya digunakan pada saat proses Logout. Proses ini akan menghapus isi session yang sudah terdaftar di browser.

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()**.

Untuk menghapus elemen-elemen dari session, tetapi tanpa membuang atau mengakhiri session itu sendiri, bisa digunakan fungsi **session\_unset()**.

Jika ingin menghapus seluruhnya (file session), gunakan **session\_destroy()**.

**Session\_destroy()** berguna untuk menghapus dan mengakhiri session, sekaligus menghapus semua elemen yang ada. Jika browser telah ditutup, secara otomatis session akan diakhiri walaupun fungsi **session\_destroy()** tidak dituliskan secara eksplisit.

**SESSION\_ID()**

Untuk mendapatkan atau memberi nilai id pada saat session saat setiap kali pengunjung membuka website, akan diberikan identifikasi session yang bersifat unik atau berbeda di setiap id session yang terbentuk.

## SESSION\_NAME()

Fungsi `session_name()` adalah untuk memperoleh atau memberikan nilai terhadap sebuah session. Nama session secara standar adalah `PHPSESSID`. jika diinginkan nama lain, bisa ditulis dengan pendeklarasian `session_name()`.

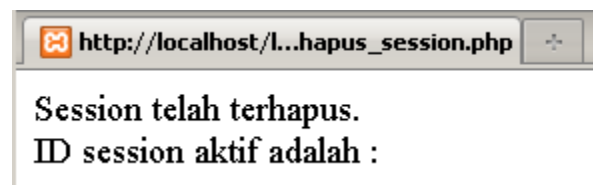
## SESSION\_REGISTER()

Fungsi `session_register()` adalah untuk mendaftarkan nilai session terhadap setiap variabel session yang akan menyimpan nilai tersebut.

### Contoh :

Buat sebuah file `hapus_session.php`

```
1  <?php
2  // Memulai session
3  session_start();
4  // Menghapus semua variabel session
5  session_unset();
6  // Menghapus seluruh file session
7  session_destroy();
8  echo "Session telah terhapus.<br>";
9  echo "ID session aktif adalah : ".session_id();
10 ?>
```



Setiap session akan berakhir secara otomatis jika halaman berpindah ke alamat server lain atau browser ditutup. Jika kita ingin mengakhiri session secara manual (seperti proses logout) maka kita bisa gunakan kode berikut :

```
1 <?php
2     // menghapus session tertentu
3     unset($_SESSION['leveluser']);
4     // menghapus semua session yang aktif
5     session_destroy();
6 >
```

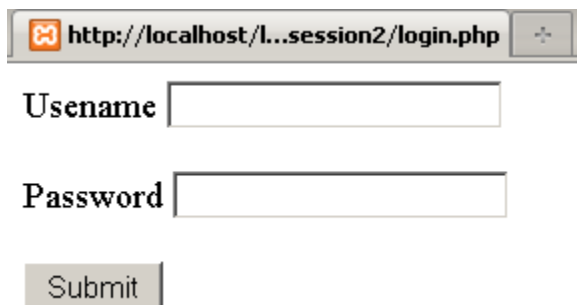
### Contoh penggunaan Session :

Kali ini kita akan mencoba membuat simulasi bagaimana Session bekerja pada sebuah halaman Login User.

Langkah pertama kita buat form login terlebih dahulu :

#### Login.php

```
1 <form method="post" action="submit.php">
2     Username <input type="text" name="username"><br><br>
3     Password <input type="password" name="password"><br><br>
4     <input type="submit" name="submit" value="Submit">
5 </form>
```



http://localhost/...session2/login.php

Username

Password

Dalam simulasi ini kita belum menggunakan database untuk menyimpan variable username dan password. Saya akan memfokuskan dulu untuk membahas cara kerja session. Jadi untuk simulasi ini, kita anggap semua proses login dianggap sukses. Untuk materi login dengan database ada di bab modul berikutnya.

Berikutnya kita buat file submit.php untuk menjalankan action proses dari form login.php

**Submit.php**

```
1 <?php
2 session_start();
3
4 $namauser = $_POST['username'];
5 $password = $_POST['password'];
6
7 if ('login sukses')
8 {
9     $_SESSION['namauser'] = $namauser;
10
11     echo "<p>Selamat datang <b>".$_SESSION['namauser']."</b></p>";
12     echo "<p>Berikut ini menu navigasi Anda</p>";
13     echo "<p><a href='hal1.php'>Menu 1</a><br>";
14     echo "<a href='hal2.php'>Menu 2</a><br>";
15     echo "<a href='hal3.php'>Menu 3</a></p>";
16 }
17 ?>
```

Hasil di browser seperti berikut :



Untuk hal1, hal2, dan hal3 juga diberikan perintah session\_start();.

**hal1.php**

```
1 <?php
2 session_start();
3
4 echo "<h1>Ini halaman pertama</h1>";
5 echo "<p>Anda login sebagai <b>".$_SESSION['namauser']."</b></p>";
6 echo "<p>Berikut ini menu navigasi Anda</p>";
7 echo "<p><a href='hal1.php'>Menu 1</a><br>";
8 echo "<a href='hal2.php'>Menu 2</a><br>";
9 echo "<a href='hal3.php'>Menu 3</a></p>";
10 echo "<p><a href='logout.php'>Logout</a></p>";
11 ?>
```

Tampilan hal1 di browser seperti berikut



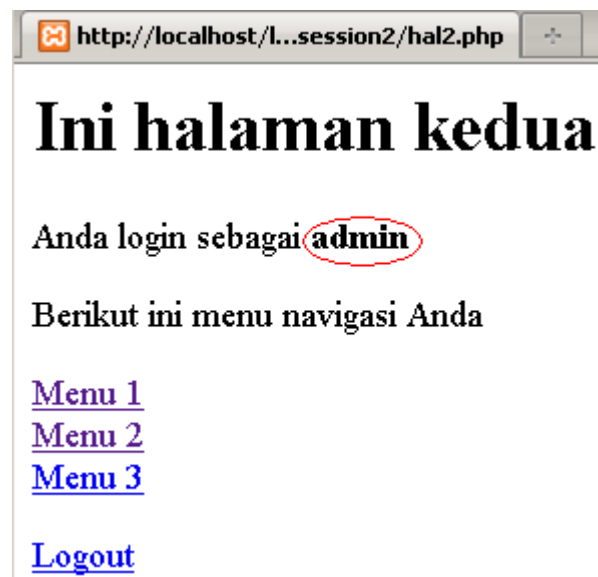
Pada halaman pertama session masih berjalan yaitu sebagai admin. Begitu juga pada halaman dua dan halaman tiga.



## hal2.php

```
1 <?php
2 session_start();
3
4 echo "<h1>Ini halaman kedua</h1>";
5 echo "<p>Anda login sebagai <b>".$_SESSION['namauser']."</b></p>";
6 echo "<p>Berikut ini menu navigasi Anda</p>";
7 echo "<p><a href='hal1.php'>Menu 1</a><br>";
8 echo "<a href='hal2.php'>Menu 2</a><br>";
9 echo "<a href='hal3.php'>Menu 3</a></p>";
10 echo "<p><a href='logout.php'>Logout</a></p>";
11 ?>
```

Tampilan halaman dua di browser sebagai berikut



## hal3.php

```
1 <?php
2 session_start();
3
4 echo "<h1>Ini halaman ketiga</h1>";
5 echo "<p>Anda login sebagai <b>".$_SESSION['namauser']."</b></p>";
6 echo "<p>Berikut ini menu navigasi Anda</p>";
7 echo "<p><a href='hal1.php'>Menu 1</a><br>";
8 echo "<a href='hal2.php'>Menu 2</a><br>";
9 echo "<a href='hal3.php'>Menu 3</a></p>";
10 echo "<p><a href='logout.php'>Logout</a></p>";
11 ?>
```

Tampilan halaman tiga pada broser sebagai berikut



**Ingat!!.** setiap akan menampilkan nilai session perintah `session_start()` harus diberikan terlebih dahulu di paling awal kode program php.

Sekarang kita akan mencoba untuk menghapus session. Bagaimana caranya??. Untuk menghapus session gunakan perintah `session_destroy()` atau `unset($_SESSION['nama session'])`. Beda dari kedua perintah tersebut adalah `session_destroy()` digunakan untuk menghapus semua session sedangkan `unset($_SESSION['nama session'])` digunakan untuk menghapus session tertentu saja.

Penerapan penghapusan nilai session ini biasanya untuk proses logout. Prinsipnya proses logout akan menghapus nilai session yang tersimpan pada browser dalam hal ini adalah username. Berikut contoh script nya:

```
1 <?php
2 session_start();
3
4 unset($_SESSION['namauser']);
5 echo "Anda telah logout";
6 ?>
```

Jika script logout di atas dijalankan, maka sername yang telah tersimpan tadi tidak akan muncul lagi di halaman-halaman yang ada.

Hasilnya akan Nampak di browser sebagai berikut. Nama uses tidak akan muncul lagi pada halaman.



## Penerapan Session pada setiap halaman

Manfaat session yang lain adalah dapat mencegah user mengakses halaman-halaman tertentu yang sifatnya private tanpa melakukan login (*by pass*). Dalam contoh di atas, Anda akan dapat mengakses halaman 1, halaman 2 dan 3 secara langsung tanpa proses login terlebih dahulu. Kita dapat membuat ketiga halaman tersebut tidak bisa diakses oleh user yang masuk tanpa proses login. Caranya aalah dengan mendeteksi session username. Pendeteksian ini dilakukan di ketiga halaman tersebut. Bila terdeteksi nilai session username ini masih kosong, maka dianggap user yang mengakses tersebut tidak melakukan login terlebih dahulu, sehingga akses harus diblok.

Berikut ini script untuk mendeteksi session username yang masih kosong.

## Cek.php

```
1 <?php
2 session_start();
3
4 if (!isset($_SESSION['namauser']))
5 {
6     echo "Anda belum login";
7     exit;
8 }
9
10 ?>
```

Script di atas nantinya akan disisipkan ke ketiga halaman private menggunakan perintah **include()**. Perintah ini disisipkan sebelum menampilkan konten yang ada pada halaman tersebut. Berikut ini contoh menyisipkan script cek.php ke halaman pertama. Untuk halaman yang lain, caranya sama.

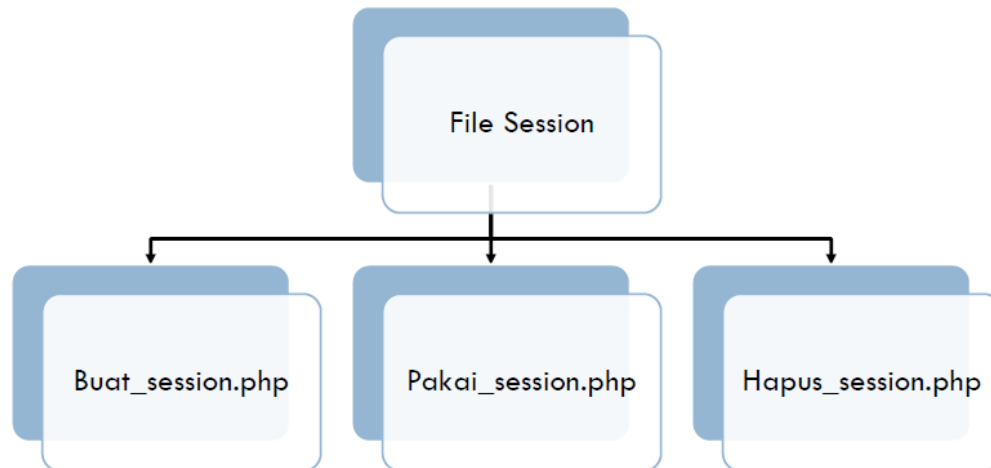
```
1 <?php
2 include "cek.php";
3
4 echo "<h1>Ini halaman pertama</h1>";
5 echo "<p>Anda login sebagai <b>".$_SESSION['namauser']. "</b></p>";
6 echo "<p>Berikut ini menu navigasi Anda</p>";
7 echo "<p><a href='hal1.php'>Menu 1</a><br>";
8 echo "<a href='hal2.php'>Menu 2</a><br>";
9 echo "<a href='hal3.php'>Menu 3</a></p>";
10 echo "<p><a href='logout.php'>Logout</a></p>";
11 ?>
```

Begitu terdeteksi user yang mengakses halaman tersebut tanpa login, maka akan muncul '**Anda belum login**', sedangkan konten halaman aslinya tidak akan muncul. Hal ini karena efek dari perintah 'exit'.

File cek.php basa kita letakkan pada setiap halaman yang akan kita batasi hak aksesnya.

## Kesimpulan dalam penggunaan Session PHP

1. 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.



2. Aplikasi yang menggunakan session salah satunya adalah situs yang menggunakan login. Setelah seseorang melakukan login, maka nama user dan status loginnya dapat dibaca di setiap 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".
3. Tempatkan fungsi `session_start()` dibagian paling atas coding php, tidak boleh didahului oleh coding apapun.
4. 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 textfield, textarea, list/menu dll).
5. Setiap kali akan menggunakan variable session, maka harus dituliskan `session_start()` dibagian paling atas coding.
6. Untuk membatasi hak akses pada setiap halaman bias kita batasi dengan menggunakan file cek.php.