

# MODUL PERTEMUAN 16

## SESSION DAN COOKIES PADA PHP

---

Pemrograman Web

2024

Jurusan Teknologi Informasi  
**Politeknik Negeri Malang**

## **Modul 16 : Session dan Cookies pada PHP**

---

### **Pendahuluan**

- Membuat session dan cookies pada php
- Mengimplementasikan session dan cookies pada login PHP

### **Tujuan**

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Memahami membuat session dan cookies pada PHP
2. Memahami session dan cookies pada PHP.
3. Menggunakan session dan cookies pada PHP

### **Alat dan Bahan**

1. PC atau Laptop
2. Web Browser
3. Teks editor
4. XAMPP

### **Teori Pendukung**

#### **A. 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.

**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.

**Session** adalah cara untuk menyimpan value pada variabel aktif yang dibawa terus hingga session tersebut dinonaktifkan. Variabel ini disimpan di server agar dapat digunakan pada beberapa halaman dan termasuk halaman dia sendiri. Selama session itu aktif, maka variabel-variabel yang memiliki value akan dapat diakses secara global.

#### **Fungsi-fungsi session :**

- `session_start()`, untuk memulai session.
- `session_destroy()`, untuk mengakhiri session.
- `session_id()`, untuk mengambil atau menentukan identitas sebuah session (session id).
- `session_register()`, untuk mendaftarkan variabel ke sebuah session.

Script untuk mengaktifkan session adalah `session_start()`; sebaliknya untuk menghapus session, dengan menggunakan `session_destroy()`; dan `unset_session()`; Jika tidak mendestroy session, maka halaman yang sebelumnya dibuka oleh pengguna akan masih bisa diakses. Hal ini mengakibatkan bocornya data, keterbukaan kerahasiaan yang bisa terjadi oleh pengguna lain. Unset Session hanya menghapus variable tertentu, sedangkan `session_destroy()` akan menghancurkan semua data sesi untuk pengguna tersebut.

**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 dibawah 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>";
    }
?>
```

## **B. User Authentication**

Autentifikasi user adalah suatu mekanisme untuk memastikan apakah suatu user itu berhak masuk ke dalam sistem atau bukan. Implementasinya adalah berupa login. Aplikasi : Pendaftaran user dan proses autentikasi:

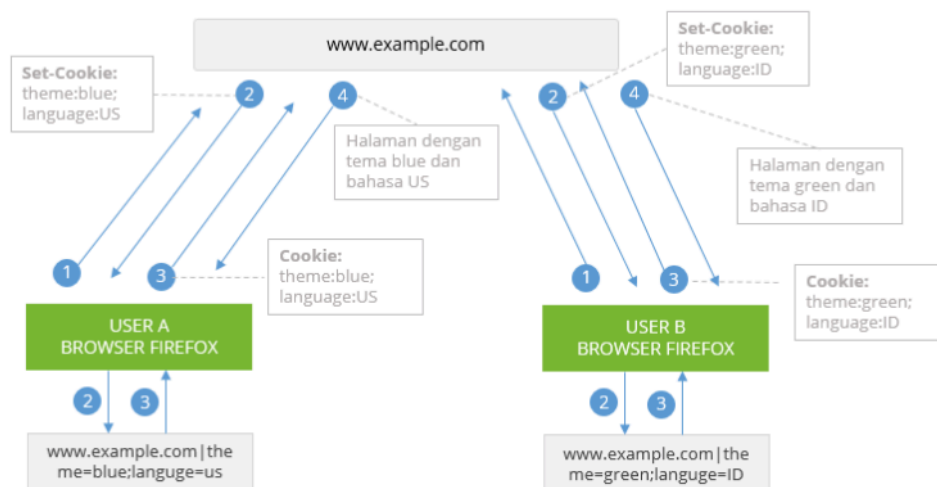
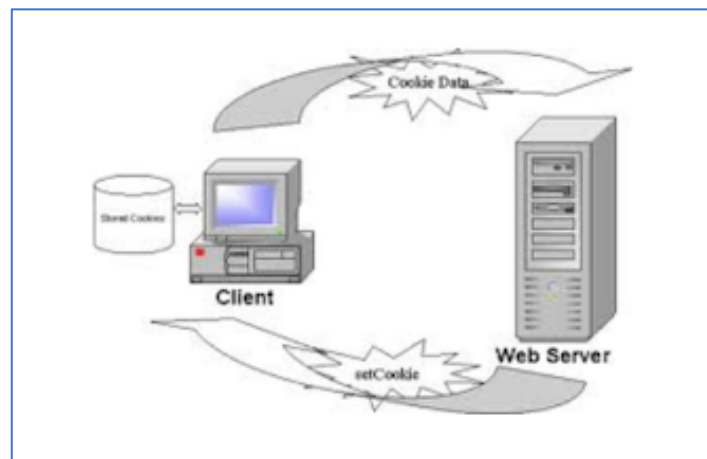
- Pendaftaran User:
  - User mengisi form pendaftaran sebagai legal user (user mengisi username dan password).
  - Sistem akan mengecek apakah username yang didaftarkan sudah ada yang memiliki atau belum.
  - Jika sudah ada, user diminta mengisi kembali username yang lain beserta passwordnya. Sedangkan jika belum ada, data user ini akan disimpan dlm database.
- Proses Autentikasi
  - User yang akan mengakses sistem diminta memasukkan username dan password (asli).
  - Sistem akan mencocokkan username dan password dengan yang tersimpan dalam database berdasarkan username yang terdaftar.
  - Jika account tersebut ada dalam database, maka user tadi bisa masuk ke dalam sistem. Jika tidak sama, maka user tadi tidak berhak masuk ke sistem.

## **C. Cookies**

Berbeda dengan session dimana informasi disimpan di server sedangkan **cookies** adalah informasi yang disimpan di sisi client. Seperti halnya session, cookies juga merupakan sebuah konsep penyimpanan informasi user. Jika session tempat penyimpanan berada di server 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 shopping cart. Biasa

digunakan untuk menyimpan sementara produk – produk yang dipilih oleh pengunjung pada saat berbelanja.

### Cara kerja cookies :



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 menelusuri 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.

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

Jenis	Menyimpan Informasi	Lokasi Penyimpanan	Dapat dipanggil	Bersifat Global	Dapat dipanggil Lain Waktu	Dibatasi Browser	Dapat diedit User
Cookies	Yes	Client	Yes	Yes	Yes	Yes	Yes
Session	Yes	Server	Yes	Yes	No	No	No

### Contoh membuat 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);
```

~ Selamat belajar ~