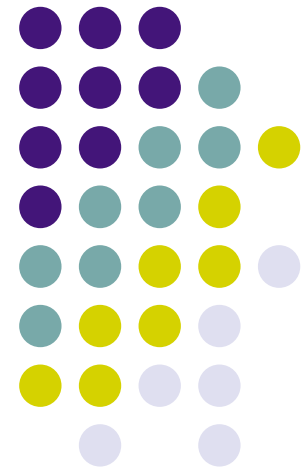


PHP – SESSION & COOKIES

Pemrograman Web II

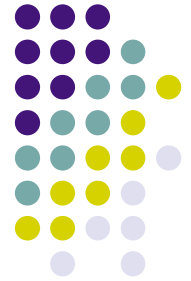
Ganjil 2013 - 2014





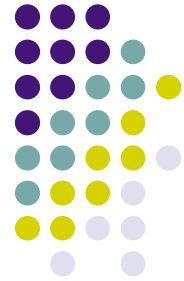
statelessness (1)

- HTTP merupakan protokol yang *stateless*
- Artinya browser cukup mengirimkan satu kali request untuk mengidentifikasi dirinya
- Jika kita mengirimkan request selanjutnya (selama browser belum di-close), maka server akan masih menganggap request itu dari sumber yang sama seperti request pertama



statelessness (2)

- Saat kita mengirimkan request pertama sampai dengan browser (bukan tab) di-close, itu disebut dengan sebuah session
- Sebuah session ditandai dengan adanya identitas unik yang disebut dengan PHPSESSID
- Tiap-tiap browser memiliki session dan PHPSESSID tersendiri yang disimpan di tempat yang berbeda di komputer kita sebagai **cookie**



PHPSESSID

X Headers Preview Response Cookies Timing

Request URL: http://localhost:8888/pw2/pertemuan-5a/
Request Method: GET
Status Code: 200 OK

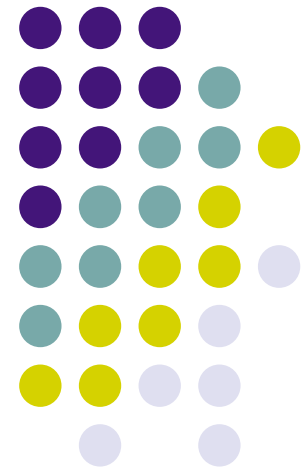
▼ **Request Headers** [view source](#)

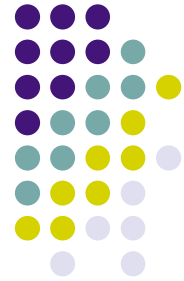
- Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
- Accept-Encoding:** gzip,deflate,sdch
- Accept-Language:** en-US,en;q=0.8
- Connection:** keep-alive
- Cookie:** PHPSESSID=416516e2713dd5cff40f52dbeed5ef4a; SQLiteManager_currentLangue=2
- Host:** localhost:8888
- Referer:** http://localhost:8888/pw2/
- User-Agent:** Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/30.0.1599.101 Safari/537.36

▼ **Response Headers** [view source](#)

- Connection:** Keep-Alive
- Content-Length:** 320
- Content-Type:** text/html; charset=ISO-8859-1
- Date:** Sun, 13 Oct 2013 07:13:45 GMT
- Keep-Alive:** timeout=5, max=100

COOKIES





COOKIE

- Cookie adalah informasi (biasanya kecil), yang dikirim oleh server kepada browser, yang kemudian dikirim kembali oleh browser ke server untuk request berikutnya
- Cookie bisa digunakan untuk:
 - Otentifikasi
 - Identifikasi user
 - Mempertahankan preferensi user (segmentasi, shopping cart)
- Data cookie berisi pasangan key & value yang disimpan di variabel superglobal **`$_COOKIE`**

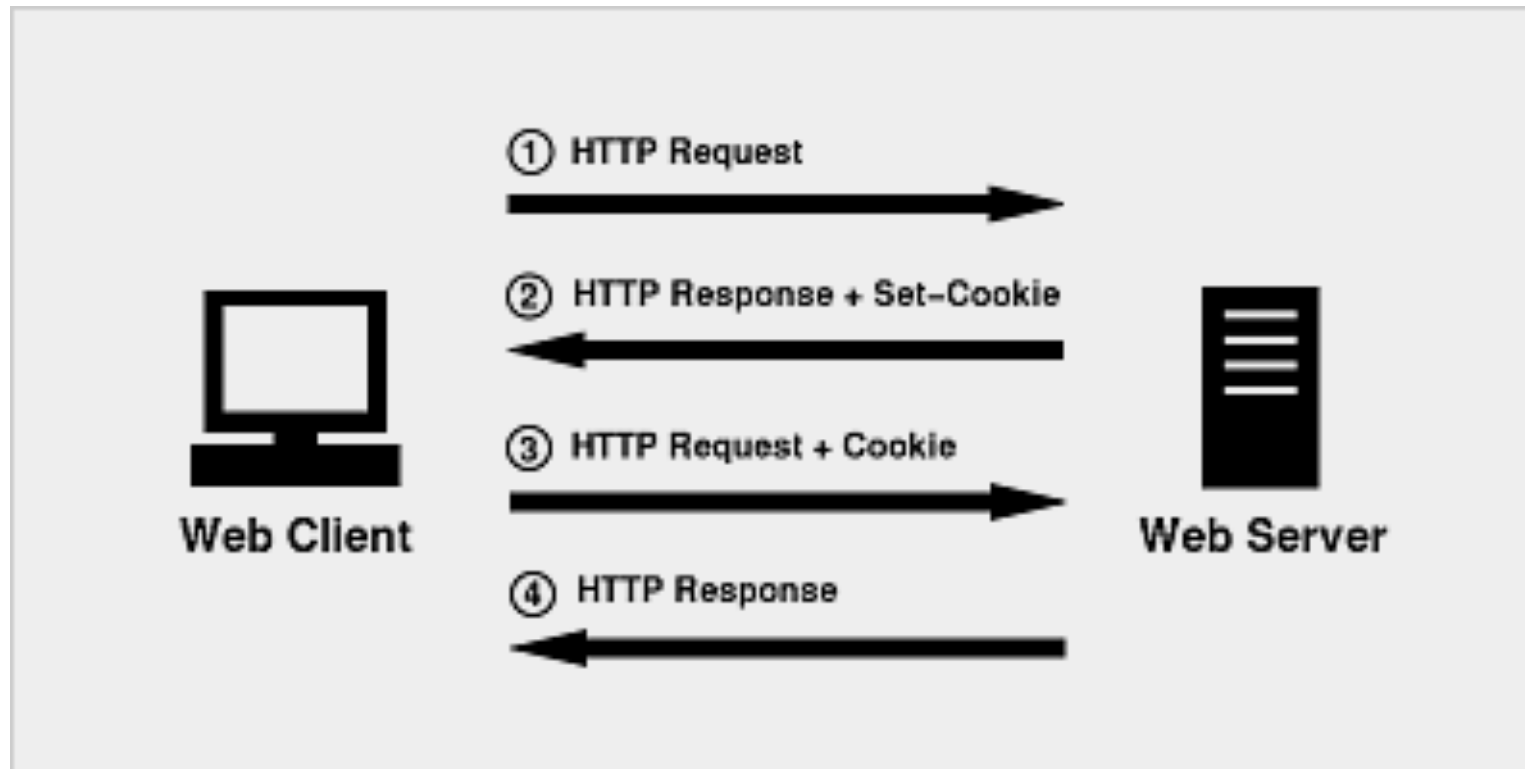


Cara kerja COOKIE (2)

1. Client / Browser mengirim HTTP Request
2. Server mengembalikan respon dengan menyertakan header **Set-Cookie** berisi pasangan key dan value yang kita tentukan
3. Client sekarang bisa mengirimkan request dengan menyertakan nilai yang sudah di-set tadi menggunakan **cookie**
4. Server mengembalikan respon



Cara kerja COOKIE (2)



Mitos seputar Cookies



- Mitos

- Cookie seperti virus yang bisa menghapus data di komputer kita
- Cookie bisa mencuri informasi kita
- Cookie digunakan untuk spam

- Fakta

- Cookie hanyalah data, bukan program
- Cookie biasanya anonim (tidak berisi data sensitif)
- Cookie bisa digunakan untuk mendeteksi preferensi / kebiasaan kita dalam mengakses situs
 - Facebook : untuk menentukan iklan yang muncul
 - Youtube : untuk menentukan video rekomendasi

Kadaluarsa Cookie (1)



- **Session Cookie** : tipe default, cookie sementara yang dapat diakses dalam satu session tertentu dan terhapus ketika browser di-close (cth: PHPSESSID)
 - Tidak bisa digunakan untuk mengakses data untuk jangka waktu yang lama
 - Lebih aman, karena tidak ada program yang bisa mengaksesnya (kecuali browser itu sendiri)
- **Persistent Cookie** : cookie yang kita buat sendiri dan kita tentukan sendiri kapan akan kadaluarsanya, dan disimpan di dalam file di komputer client
 - Bisa mengakses data untuk jangka waktu yang lama
 - Cenderung tidak aman, karena user dapat memodifikasi cookie dan merubah nilainya

n-5a/cookies

The screenshot shows the Chrome DevTools interface with the 'Resources' panel open. The 'Cookies' section is selected in the left sidebar. The main table displays the following data:

Name	Value	Domain	Path	Expires / Max-Age	Size
PHPSESSID	416516e2713dd5cff40f52dbeed5ef4a	localhost	/	Session	41
nama	sandhika	localhost	/pw2/pertemuan-5a/cookies	Mon, 14 Oct 2013 08:42:24 GMT	12

A red arrow points from the text 'Persistent Cookie' to the 'Expires' column of the second row, indicating that this cookie is persistent.

Membuat Cookie (1)



```
setcookie(name, value, expire, path, domain, secure)
```

- **name (required)**
 - Nama dari cookie yang dibuat
- **value (required)**
 - Nilai dari cookie yang dibuat
- **expire (optional)**
 - Berisi waktu kapan cookie akan expire / kadaluarsa
 - cth: `time() + 60 * 60 * 24 * 30`
(kadaluarsa dalam 1 bulan)
 - Jika parameter tidak diisi, maka cookie akan kadaluarsa dalam satu session (ketika browser di-close)

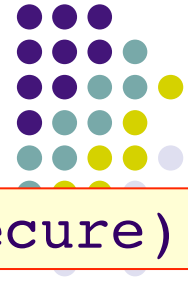
Membuat Cookie (2)



```
setcookie(name, value, expire, path, domain, secure)
```

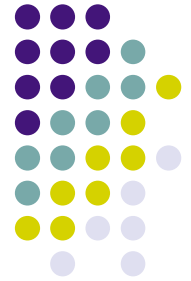
- **path (optional)**
 - Menentukan letak dimana cookie bisa digunakan di server
 - Jika diisi “ / ”, maka dapat digunakan di semua domain pada server
 - Jika diisi “ /pw2 / ”, maka cookie dapat digunakan hanya di dalam direktori pw2 beserta seluruh subdirektornya
 - Jika tidak diisi, maka cookie hanya dapat digunakan di dalam direktori file yang membuatnya saja
- **domain (optional)**
 - Menentukan domain dari cookie
 - Jika menuliskan sandhikagalih.net, maka cookie dapat digunakan oleh sandhikagalih.net beserta seluruh sub-domainnya

Membuat Cookie (3)



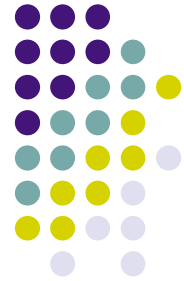
```
setcookie(name, value, expire, path, domain, secure)
```

- **secure (optional)**
 - Menentukan apakah cookie akan dikirimkan melalui SSL / HTTPS atau tidak
 - Jika diisi **true** / **1**, maka cookie hanya akan di-set apabila menggunakan HTTPS
 - Defaultnya adalah **false**



Membuat Cookie (4)

- Memanggil fungsi **setcookie()** membuat server mengirimkan cookie ke browser
- **setcookie()** ini harus dipanggil sebelum menuliskan elemen HTML apapun
- Kita bisa membuat multiple cookie (sampai dengan 50) dengan ukuran sekitar 3-4 Kb



Membaca isi Cookie

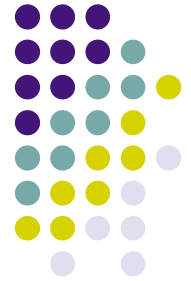
- Untuk membaca isi dari cookie yang sudah di-set sebelumnya, gunakan superglobal **\$_COOKIE**

setcookie.php

```
<?php
// set cookie selama 1 hari / 24 jam
setcookie("nama", "sandhika", time() + 60 * 60 * 24);
?>
```

printcookie.php

```
<?php
// cetak isi dari cookie nama
// yang sudah di-set sebelumnya
echo $_COOKIE["nama"];
?>
```

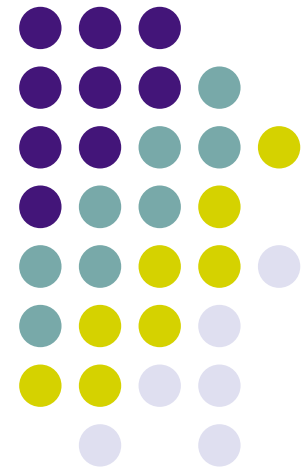
Menghapus Cookie

- Untuk menghapus cookie, set ulang cookie dengan nama yang sama
- isi valuenya dengan string kosong ("")
- set expirenya dengan waktu yang lampau
- Cth: **time()** – **3600**, expire 1 jam yang lalu

deletetecookie.php

```
<?php
// menghapus cookie
setcookie("nama", "", time() - 3600);
?>
```

SESSION





SESSION

- Session adalah konsep abstrak untuk merepresentasikan interaksi antara browser / client dan server
- Session vs. Cookie
 - Cookie menyimpan data di client
 - Session menyimpan data di server
- Session dibangun diatas cookie
 - Setiap request, client mengirimkan cookie yang berisi session ID (PHPSESSID)
 - server menggunakan session ID tersebut untuk mencari data session

SESSION



- Session adalah konsep abstrak untuk merepresentasikan interaksi antara browser / client dan server
- Session vs. Cookie
 - Cookie menyimpan data di client
 - Session menyimpan data di server
- Session dibangun diatas cookie
 - Setiap request, client mengirimkan cookie yang berisi session ID (PHPSESSID)
 - server menggunakan session ID tersebut untuk mencari data session
- Setelah ditemukan, isi data session pada server disimpan ke dalam variabel superglobal **`$_SESSION`**

Mekanisme Session



- Pertama-tama, browser mengirimkan request ke server untuk halaman yang sudah menjalankan fungsi session
- Request diterima server, kemudian server mengirimkan data session ID kepada client, disimpan oleh client sebagai cookie dengan nama PHPSESSID
- Setelah diketahui session id-nya, request selanjutnya yang dikirimkan client, baik itu berupa pengiriman data maupun permintaan data session, akan dilakukan berdasarkan session id tersebut
- Data yang dikirim dan diminta, nantinya disimpan di variabel superglobal `$_SESSION`

Bekerja dengan Session (1)



```
session_start();
```

- Panggil fungsi session_start(); sebelum mencetak elemen HTML apapun (termasuk spasi) untuk menandakan akan memulai session
- Ketika memanggil fungsi session_start(); maka:
 - Jika server belum pernah berinteraksi dengan client, maka akan dibuat session baru
 - Jika sudah pernah maka kita memiliki akses ke variabel \$_SESSION, untuk menambah / menerima data

Bekerja dengan Session (2)



startsession.php

```
<?php
// jalankan fungsi session_start(); untuk memulai session
// jalankan sebelum ada elemen HTML apapun
session_start();

// akses ke variabel superglobal $_SESSION
// mengisikan pasangan key dan value
$_SESSION["nama"] = "Sandhika";
?>
```

printsession.php

```
<?php
// sebelum bekerja dengan session,
// jalankan fungsi session_start(); terlebih dahulu
session_start();

// cetak isi variabel superglobal,
// yang sebelumnya sudah di set
echo $_SESSION["nama"];
?>
```

Bekerja dengan Session (3)



deletesession.php

```
<?php
// sebelum bekerja dengan session,
// jalankan fungsi session_start(); terlebih dahulu
session_start();

// matikan akses ke variabel $_SESSION
session_destroy();

// hapus isi array $_SESSION
$_SESSION = array();
?>
```


Terima Kasih

Pemrograman Web 2 - 20132014

sandhikagalih@unpas.ac.id

erik@unpas.ac.id

anggoro.ari.nurcahyo@unpas.ac.id

acep.hendra@unpas.ac.id

