

MODUL 7

PEMROGRAMAN WEB

SESSION DAN COOKIE

A. TUJUAN PERKULIAHAN

- a. Mahasiswa mengenali konsep session pada bahasa PHP
- b. Mahasiswa mengenali konsep cookie pada bahasa PHP
- c. Mahasiswa mampu mengimplementasikan session dan cookie pada aplikasi web berbasis PHP

B. ALOKASI WAKTU

3 x 50 menit

C. DASAR TEORI

1. SESSION

a. Definisi

Dalam PHP, session merupakan data yang disimpan dalam suatu server yang dapat digunakan secara global di server tersebut, dimana data tersebut spesifik merujuk ke user/client tertentu. Contoh penggunaan session adalah ketika user telah login di halaman tertentu, maka ketika membuka halaman lain, PHP akan mengingat bahwa user tersebut telah login.

Contoh lain ketika kita telah login pada Google, maka setiap kali kita membuka layanan Google seperti GMail, Google Drive, dll di tab berbeda, kita akan selalu dalam keadaan login, kecuali kita buka dengan browser yang berbeda.

b. Cara kerja session

Ketika memulai session dengan menjalankan perintah `session_start()` maka PHP akan menjalankan perintah baik pada server maupun pada client/user.

- **Sisi Server**

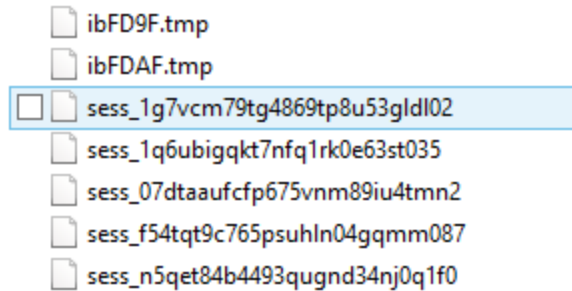
Dari sisi server, PHP akan membuat file yang disebut file session yang diawali dengan `sess_` dan diikuti dengan ID session, dimana ID Session tersebut mereferensikan id session yang ada di browser client. nilai ID session adalah:

- Jika browser mengizinkan penggunaan cookie, maka ID session berupa angka acak sebanyak 26 karakter hexadecimal, contoh: `1g7vcm79tg4869tp8u53gldl02`
- Jika browser tidak mengizinkan penggunaan cookie, ada dua kemungkinan. Jika nama session ada di url, maka nilai ID Session sama dengan id yang ada di url, misal **`https://example.com?page=login&PHPSESSID=1234`**, maka ID Session yang digunakan adalah 1234, sehingga nama file session yang dibuat **`sess_1234`**, Namun jika ID Session tidak ada di URL, maka PHP akan membuat file dengan ID Session sama seperti sebelumnya yaitu 26 karakter hexadecimal. File ini akan dibuat setiap kali perintah `session_start()` dijalankan, dan tidak ada referensi ke user/client.

File Session secara default disimpan di dalam direktori temporary dimana letak direktori ini tergantung dari masing-masing sistem operasi yang digunakan. untuk mengetahui dimana php menyimpan file session, dapat menggunakan perintah: `session_save_path()` Misal ada file dengan kode berikut:

```
<?php echo session_save_path(); ?>
```

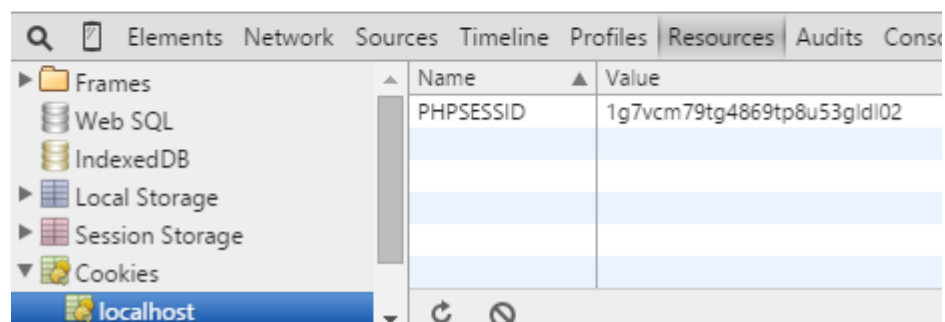
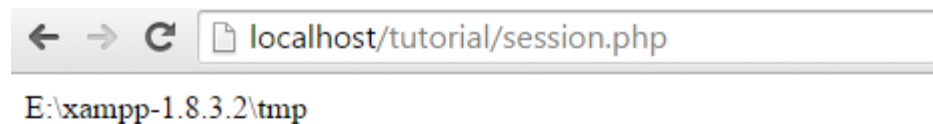
Ketika dijalankan file tersebut di komputer yang menggunakan sistem operasi windows, contoh output yang dihasilkan adalah `C:\xampp\tmp`, jika kita buka direktori tersebut, disana kita temukan banyak file `.tmp`, cari nama dengan awalan `sess_`, file tersebut merupakan file session.



- **Sisi User/Client**

Selanjutnya dari sisi client/user, karakteristik session PHP:

- Jika browser mengijinkan penggunaan cookie, maka browser akan membuat file cookie dengan nama default yaitu PHPSESSID dengan nilai (value) ID session. Lokasi penyimpanan file tersebut berbeda beda tergantung jenis browser yang digunakan, untuk google chrome lokasi cookie berada di: **C:\Users\<NAMA USER>\AppData\Local\Google\Chrome\User Data\Default** dengan nama file Cookie, namun kita tidak dapat membukanya secara langsung, melainkan harus menggunakan SQLite. Kita dapat membaca isinya melalui Developer Tools, yaitu pada tab Resources bagian Cookies. Contoh seperti gambar berikut:



Jika kita bandingkan antara gambar 1 dan 2 terdapat kesamaan nilai yaitu `1g7vcm79tg4869tp8u53g1d102` yang merupakan ID Session. Melalui ID ini, setiap kali user mengunjungi situs yang sama dan dengan browser yang sama, maka server akan selalu menggunakan data-data yang ada di file `sess_1g7vcm79tg4869tp8u53g1d102`.

- Jika browser tidak mengizinkan penggunaan cookie, ada dua kemungkinan. Jika nama session berada di url, maka PHP akan membuat referensi ID Session sesuai dengan yang ada di URL. Misal: `https://example.com?page=login&PHPSESSID=1234`, maka php akan menggunakan data session pada file session `sess_1234`. Jika ID session tidak ada di URL maka data session hanya dapat digunakan di halaman tersebut saja.

c. Isi File Session Pada PHP

Isi file session pada PHP berupa array yang di *serialize* nilainya akan berubah ubah sesuai dengan manipulasi data yang kita lakukan, misal file session.php kita ubah menjadi:

```
<?php
echo session_save_path();
session_start();
$_SESSION['user'] = 'ptikuns';
$_SESSION['auth'] = 1;
```

ketika kita jalankan PHP akan menyimpan data session dengan key dan value sesuai dengan yang kita tentukan, jika kita buka file session tadi menggunakan notepad, maka akan berisi `user|s:7:"ptikuns";auth|b:1;`, format: nama key diikuti tanda | kemudian data yang di *serialize* (tipe data : panjang data(jika ada) : nilai data) dengan tanda pemisah antara data satu dengan yang lain menggunakan titik koma. contoh diatas berarti bahwa s merupakan tipe data string, 6 panjang data, sedangkan b berarti tipe data boelan.

d. Menjalankan Session Pada PHP

Untuk mengawali menjalankan session, kita gunakan perintah `session_start()` yang akan memberitahu server bahwa kita akan menggunakan session, selanjutnya server akan mengecek apakah pengunjung telah memiliki session ID:

1. jika ya, maka server akan mengambil data sesuai dengan ID session yang di minta oleh browser (misal di file a.php kita telah menjalankan perintah `session_start()` kemudian di file b.php kita jalankan lagi perintah `session_start()`, maka ketika menjalankan file b.php browser akan mengirimkan ID Session, sehingga data session yang terbentuk ketika menjalankan file a.php akan dipakai lagi ketika kita menjalankan file b.php),
2. jika tidak, maka server akan membuat file session dengan nama `sess_ID` dan mengirimkan id tersebut ke browser untuk dibuatkan cookie.

Karena penggunaan data session harus didahului dengan perintah `session_start()` maka perintah ini sebaiknya ditulis di baris paling awal di program kita.

e. Menambahkan Data Session Pada PHP

Semua data session disimpan dalam bentuk array superglobal dengan nama `$_SESSION`, sehingga seperti array pada umumnya setiap variabel session disimpan dalam hubungan key dan value, untuk menambahkan data kedalamnya sama dengan ketika kita menambahkan data di array biasa, namun bedanya variabel `$_SESSION` akan tetap dapat kita gunakan di file PHP manapun (dalam satu server) hingga kita menakhirinya dengan perintah `session_destroy()`

Contoh untuk menambahkan data session:

```
<?php
session_start();
$_SESSION['user'] = 'ptikuns';
$_SESSION['role'] = 'admin';
$_SESSION['login_time'] = date('Y-m-d');
print_r($_SESSION);
```

```
/*
Output
Array
(
    [user] => ptikuns
    [role] => admin
    [login_time] => 2021-09-05
)
*/
```

f. Membaca Data Session Pada PHP

Setelah berhasil menyimpan data pada session, data tersebut langsung dapat digunakan. Untuk memanggil data session pada PHP dapat dilakukan seperti memanggil data pada array pada umumnya, yaitu dengan key nya. contoh:

```
<?php
    $_SESSION['user'] = ' ptikuns ';
    echo $_SESSION['user']; // ptikuns
?>
```

Note: Session yang telah kita isi datanya dapat langsung kita gunakan

g. Menghapus Data Session Pada PHP

Untuk menghapus data session pada PHP, sama seperti ketika kita menghapus variabel, yaitu menggunakan perintah `unset()` . contoh:

```
<?php
unset($_SESSION['user']);
print_r($_SESSION);

/*
Output
Array
(
    [role] => admin
    [login_time] => 2015-09-05
)
*/
```

Di samping itu kita juga dapat menggunakan perintah `session_unset()` untuk menghapus semua data pada `$_SESSION`, contoh:

```
<?php
session_start();
$_SESSION['user'] = 'ptikuns';
$_SESSION['role'] = 'Admin';
session_unset();
print_r($_SESSION);

/*
Output
Array ()
*/
```

perlu diperhatikan bahwa jika menggunakan perintah `unset()` untuk menghapus data session, pilihlah data secara spesifik, jangan menghapus `$_SESSION` itu sendiri (`unset($_SESSION)`), karena akan menghapus semua isi data session, dan menghapus variabel `$_SESSION` itu sendiri, sehingga jika kita panggil akan memunculkan pesan error bahwa variabel `$_SESSION` undefined. contoh:

```
<?php
session_start();
$_SESSION['user'] = 'agusph';
$_SESSION['role'] = 'admin';
$_SESSION['login_time'] = date('Y-m-d');
unset($_SESSION);
print_r($_SESSION);

// Notice: Undefined variable: _SESSION in
E:\xampp\htdocs\pemweb\session.php on line
```

h. Mengakhiri Session Pada PHP

Untuk mengakhiri session pada PHP, kita gunakan perintah `session_destroy()`, dengan perintah ini maka file session akan dihapus dari server. contohnya ketika user logout, maka session akan berakhir dan user diminta untuk login kembali.

2. COOKIE

a. Definisi

Cookie adalah data teks yang disimpan pada komputer user oleh suatu website. Penyimpanan data ini umumnya melalui media browser. Adapun karakteristik cookies adalah sebagai berikut:

- Cookie ini disimpan khusus oleh masing-masing browser, artinya masing-masing browser memiliki data cookie tersendiri.
- Data cookie hanya merujuk ke website tertentu, sehingga ketika browser mengirim data cookie ke webserver, hanya data cookie website tersebut yang akan dikirim.
- Ketika merequest alamat website tertentu melalui browser, maka browser juga akan mengirim data cookie yang ada di komputer ke website tersebut.
- Data cookie yang disimpan umumnya maksimal 4096 byte atau 4Kb dan maksimal 180 cookie per domain (chrome), sehingga kita tidak bisa menyimpan data dalam jumlah besar pada cookie.

Masing masing browser menyimpan cookie di tempat yang berbeda, contoh: browser seperti Google Chrome menyimpan data cookie dengan nama file **Cookies** pada path berikut

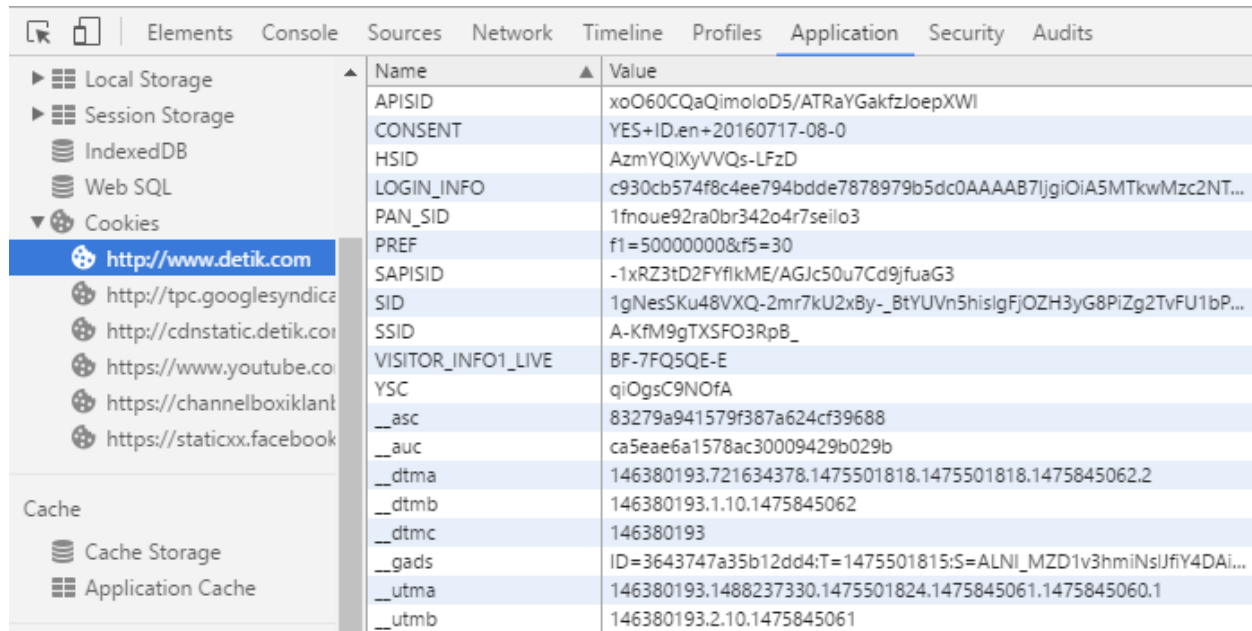
```
C:\Users\<username>\AppData\Local\Google\Chrome\User_Data\Default
```

File cookie ini umumnya berbentuk database, seperti sqlite, sehingga kita tidak bisa langsung membukanya dengan code editor/text viewer seperti notepad. Cara yang paling mudah untuk melihat data cookies adalah menggunakan browser.

Pada browser, kita dapat melihat data cookie melalui **Web Developer Tools**. Karena data cookie ini disimpan per website. Maka untuk melihatnya kita harus membuka website tersebut.

Contoh: pada browser chrome, untuk melihat data cookies, buka Developer Tools (F12) kemudian pilih tab Application, selanjutnya pilih bagian cookies

Berikut ini contoh cookie dari **detik.com**



The screenshot shows the 'Application' tab in a web browser's developer tools. The left sidebar lists storage areas: Local Storage, Session Storage, IndexedDB, Web SQL, and Cookies. Under Cookies, 'http://www.detik.com' is selected. The main pane displays a table of cookies for this domain.

Name	Value
APISID	xoO60CQaQimoloD5/ATRaYGakfzJoepXWl
CONSENT	YES+ID.en+20160717-08-0
HSID	AzmYQIXyVVQs-LFzD
LOGIN_INFO	c930cb574f8c4ee794bdde7878979b5dc0AAAAB7ljgiOiA5MTkwMzc2NT...
PAN_SID	1fnoue92ra0br342o4r7seilo3
PREF	f1=50000000&f5=30
SAPISID	-1xRZ3tD2FYfikME/AGJc50u7Cd9jfuaG3
SID	1gNes5Ku48VXQ-2mr7kU2xBy-_BtYUVn5hisgFjOZH3yG8PiZg2TvFU1bP...
SSID	A-KfM9gTXSFO3RpB_
VISITOR_INFO1_LIVE	BF-7FQ5QE-E
YSC	qiOgsC9NoFA
__asc	83279a941579f387a624cf39688
__auc	ca5eae6a1578ac30009429b029b
__dtma	146380193.721634378.1475501818.1475501818.1475845062.2
__dtmb	146380193.1.10.1475845062
__dtmc	146380193
__gads	ID=3643747a35b12dd4:T=1475501815:S=ALNI_MZD1v3hmiNsJfY4DAi...
__utma	146380193.1488237330.1475501824.1475845061.1475845060.1
__utmb	146380193.2.10.1475845061

b. Cara Kerja Cookie

Pada saat kita meload halaman website atau melakukan pengaturan tertentu (misal pengaturan bahasa atau timezone), website tersebut akan memerintahkan browser untuk menyimpan data tertentu ke dalam file cookies.

Selanjutnya, ketika kita meload kembali website tersebut, maka data cookie tersebut juga akan terkirim bersama dengan pengiriman HTTP Header. Data cookie ini akan digunakan website untuk berbagai keperluan, salah satunya melakukan pengaturan bahasa atau timezone sehingga content yang dikirim sesuai dengan preferensi user.

Cara kerja cookie dapat dimisalkan sebagai berikut:

Pertama-tama, browser meminta webserver (misal pada **www.example.com**) untuk memberikan data halaman tertentu, ketika mengirim data, webserver memerintahkan browser untuk menyimpan data ke file cookie.

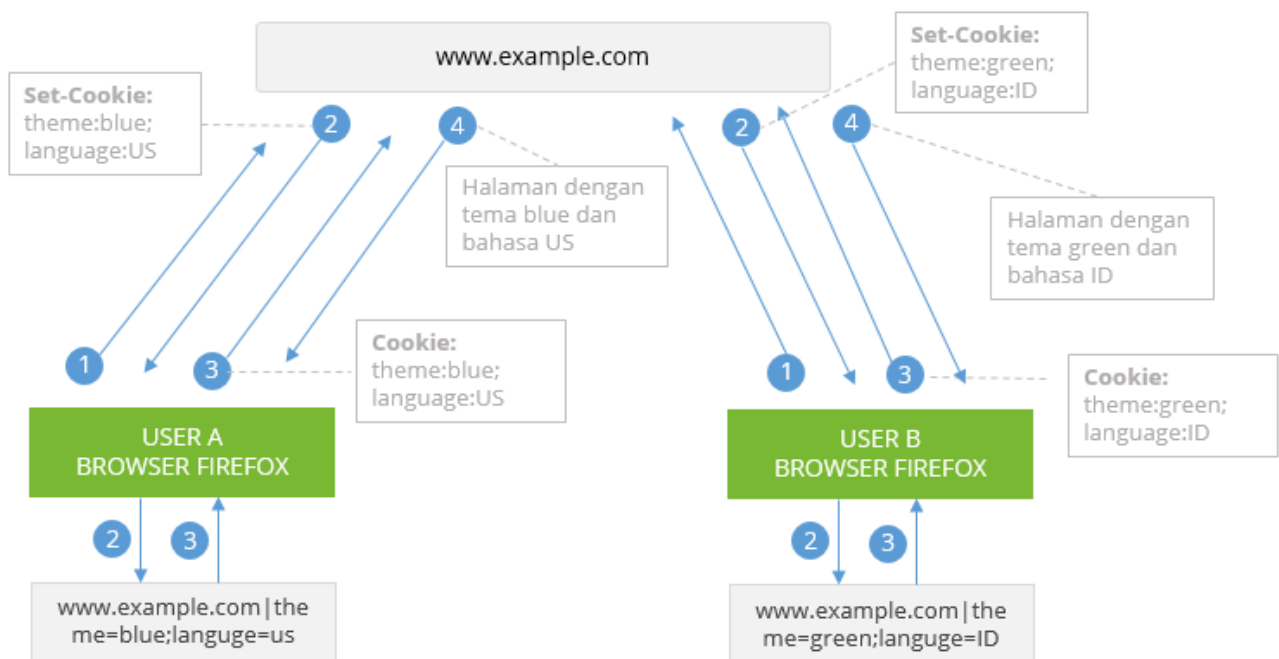
Selanjutnya, ketika browser kembali mengakses website tersebut, browser akan sekaligus mengirim data cookie yang disimpan tadi.

Data cookie tersebut digunakan oleh webserver untuk berbagai keperluan, termasuk menerapkan setting tertentu pada data yang akan dikirim kembali ke browser. Misal:

- User A dengan browser Firefox mengunjungi website www.example.com kemudian mengakses menu setting dan menerapkan tema biru dan bahasa antar muka Bahasa Inggris.
- Di belahan negara lain, user B dengan browser Firefox menerapkan tema hijau dengan bahasa antar muka Bahasa Indonesia.

Agar server www.example.com mengetahui tema dan bahasa yang telah diterapkan oleh masing-masing user pada masing masing browser, maka www.example.com memerintahkan browser untuk menyimpan data setting tersebut.

Selanjutnya ketika browser tersebut kembali mengunjungi www.example.com, data cookie tersebut ikut dikirim sehingga server tahu tema dan bahasa apa yang akan diterapkan kepada browser tersebut. Untuk lebih jelasnya, perhatikan ilustrasi berikut:



c. Manfaat Menggunakan Cookie

1. **Fitur “Remember me” pada login.** Fitur ini mengizinkan user yang sudah login pada suatu website (tidak logout) untuk tidak perlu login kembali maka ketika user tersebut mengakses lagi ke website tersebut.
2. **Fitur shopping chart.** fitur ini dapat dicoba pada web e-commerce. Buka salah satu e-commerce dan masukkan beberapa item ke dalam keranjang. Selanjutnya kita dapat mengakses kembali item-item tersebut di dalam keranjang meski kita sudah menutup website atau browser kita.
3. **Google Account.** Google akan merekam account yang kita gunakan, sehingga akan memudahkan kita untuk berpindah dari account satu ke account yang lain.
4. **Iklan.** Dengan menggunakan cookie, webserver dapat menampilkan iklan sesuai dengan preferensi produk apa saja yang kita sukai.
5. **Pemilihan bahasa, tema warna, font** dan lain sebagainya pada sebuah website

d. Membuat Cookies Pada PHP

Untuk membuat cookies pada PHP, gunakan fungsi **setcookie()**. Fungsi ini memiliki enam argumen, yaitu:

1. **Nama cookie**, berisi nama variabel cookie.
2. **Nilai cookie**, data yang akan disimpan yang merujuk pada nama cookie (nomor 1)
3. **Expire.** Kapan cookie akan berakhir. Format waktu berupa *Unix timestamp*, yang dibuat dengan menggunakan fungsi `time()`. Jika nilai expire dikosongkan atau bernilai 0, maka data cookie akan expire/dihapus ketika browser ditutup (ketika session berakhir)
4. **Path.** Path/Lokasi pada server di mana cookie dapat digunakan. Jika diisi tanda slash '/', maka cookie dapat digunakan diseluruh bagian website. Jika diisi nilai tertentu, misal `‘/blog/’`, maka cookie hanya dapat digunakan pada direktori blog dan subdirektornya, misal: `‘/blog/archive/’`.

Jika nilai dikosongkan, maka lokasi akan merujuk ke direktori dimana cookie di buat.

5. **Domain.** subdomain dimana cookie dapat digunakan.

Misal: jika kita isi (**uns.ac.id**), maka cookie akan dapat digunakan pada semua subdomain seperti: **fkip.uns.ac.id**, **ptik.fkip.uns.ac.id**, dll.

Jika nilainya kosong, maka cookie akan berlaku pada seluruh bagian domain namun tidak pada subdomainnya.

6. **Secure.** Default false. Jika true, browser akan mengirim cookie ke webserver hanya jika koneksi berbentuk HTTPS.

7. **Httponly.** Default false. Cookie hanya dapat diakses hanya melalui protokol http.

Kita tidak bisa mengosongkan waktu expire, dengan demikian, kita tidak bisa membuat cookie dengan umur yang tidak terbatas, solusinya, kita buat waktu expired cookie sangat lama, misal 5 tahun kedepan

Note: Dari ketujuh argumen tersebut, yang paling sering digunakan adalah empat pertama. Khusus untuk argumen domain, isinya (jika ada) harus menggunakan dot (.)

• Menghitung Waktu Expired

Pada argumen ke tiga, kita menentukan kapan cookie akan berakhir (expired), karena nilai yang diisikan berupa timestamp, maka cara yang paling mudah untuk menentukan waktu expire adalah menggunakan hitungan mundur dari detik, menit, jam, hari, dst...

Contoh untuk membuat waktu expire cookie 30 hari kedepan, maka kita isikan `time()` + `(60 * 60 * 24 * 30)` yang artinya:

- `time()` yang berarti waktu ketika script dieksekusi (dalam unix timestamp).
- 60 pertama yang berarti 60 detik (1 menit = 60 detik);
- 60 kedua yang berarti 60 menit (1 jam = 60 menit)
- 24 yang berarti 24 jam (1 hari = 24 jam)
- 30 yang berarti 30 hari

Cara ini jauh lebih mudah dibaca dan di-maintenance daripada kita menggunakan `time() + 2595000`

Cara lain yang lebih mudah adalah dengan menggunakan format tanggal relatif ([Relative Date Format](#)) yaitu dengan menggunakan kata-kata day, month, weeks, year, dsb.

Contoh penggunaan relative date format pada cookie:

```
// Expire 7 hari kedepan
setcookie('background-color', '#75b14a', strtotime('+7
days'), '/');
// Expire 3 bulan kedepan
setcookie('background-color', '#75b14a', strtotime('+3
months'), '/');
// Expire 1 tahun kedepan
setcookie('background-color', '#75b14a', strtotime('+1
year'), '/');
```

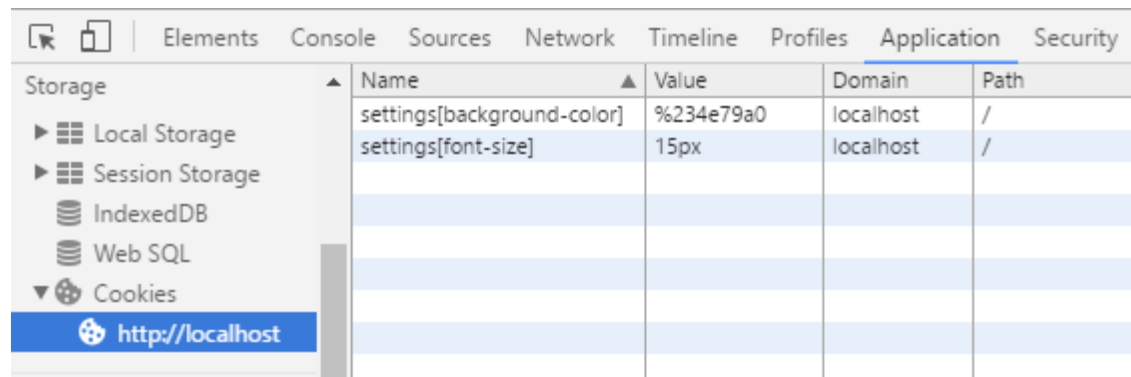
Note: kita dapat menggunakan bentuk tunggal maupun bentuk jamak (dengan tambahan s). Misal: penggunaan `day` atau `days` akan menghasilkan nilai yang sama.

- **Url Encode**

Apabila kita perhatikan, nilai dari cookie bisa berbeda dari nilai yang kita set. Perhatikan contoh berikut :

```
setcookie('settings[background-color]', '#4e79a0', time()
+ (60 * 60), '/');
setcookie('settings[font-size]', '15px', time() + (60 *
60), '/');
```

Jika kita lihat di developer tools, maka kita dapatkan seperti ini:



	Name	Value	Domain	Path
	settings[background-color]	%234e79a0	localhost	/
	settings[font-size]	15px	localhost	/

Pada contoh diatas, kita set cookie background-color dengan nilai #4e79a0, namun ketika kita lihat pada developer tools, nilainya %2375b14a

Hal ini disebabkan karena PHP mengencode nilai tersebut menjadi format standar url (fungsi `url_encode`). Hal ini untukantisipasi, ketika user tidak menggunakan cookie, maka nilai cookie dapat disimpan pada url.

Jika kita menginginkan nilai cookie apa adanya (sama seperti nilai yang kita set) maka kita dapat menggunakan fungsi `setrawcookie()`. Namun `setrawcookie` tidak bisa menggunakan karakter tertentu seperti titik koma dan spasi.

e. Membaca Data Cookie Pada PHP

Untuk membaca data cookie, kita gunakan variabel bawaan PHP `$_COOKIE`, jika tidak ada cookie sama sekali maka variabel ini tetap ada namun bernilai kosong. Untuk mengambil data cookie tertentu, gunakan variabel `$_COOKIE` dengan index nama cookie tersebut, misal: untuk nama cookie theme, kita panggil menggunakan `$_COOKIE['theme']`

Untuk menampilkan semua data cookie, kita gunakan fungsi `print_r()`, contoh: `echo '<pre>'; print_r($_COOKIE)`

```
echo '<pre>'; print_r($_COOKIE);  
/* Hasil:  
Array  
(  
    [PHPSESSID] => vprkeu3jsk7qmkohbhcu0fcto7  
    [settings] => Array  
        (  
            [background-color] => #4e79a0  
            [font-size] => 15px  
        )  
    )  
) */
```

Note: Pada PHP, cookie yang dikirim oleh browser disimpan dalam variabel global `$_COOKIE`. Hal ini berarti bahwa kita tidak dapat langsung mengambil nilai cookie ketika baru saja membuatnya, karena, data tersebut belum dikirim kembali ke server.

f. Menghapus Cookie Pada PHP

Untuk menghapus cookie pada PHP, kita cukup mengisi nilai waktu pada masa sebelum saat ini. Misal kita ingin hapus data cookie "user":

```
setcookie("user", "", time() - 3600);
```

Pada contoh di atas, kita mengisi variabel cookie "user" dengan waktu satu jam (3600 detik) sebelum saat perintah setcookie() tersebut dijalankan. Karena karakteristik cookie hanya dapat bekerja dengan menghitung masa waktu yang tersisa, sehingga jika diisi nilai minus variabel cookie akan terhapus secara otomatis.

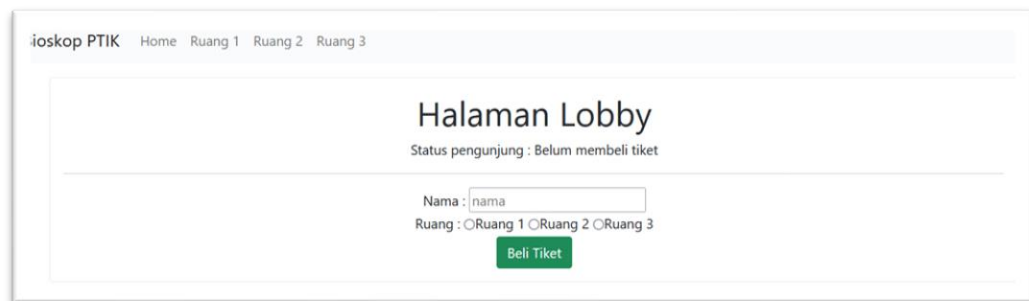
D. PRAKTIKUM

1. KEGIATAN 1 : Membuat Tiket Bioskop dengan Session

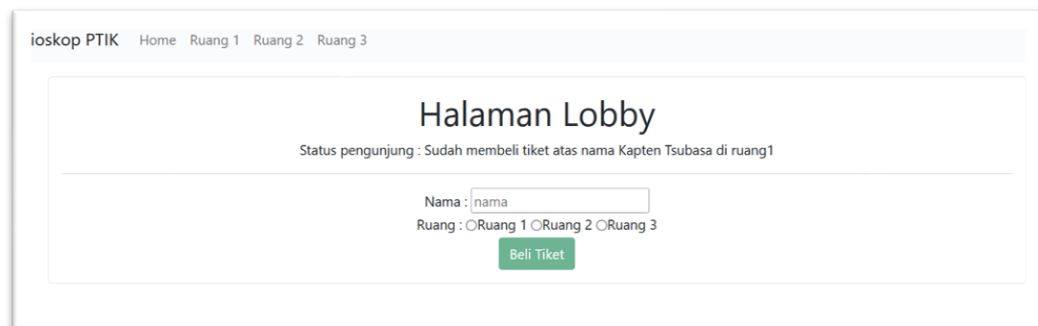
Pada kegiatan 1, kita akan membuat sebuah web untuk melihat video dengan konsep seperti bioskop, yakni pengunjung harus mendapatkan tiket terlebih dahulu agar dapat masuk ke dalam ruang bioskop.

a. Halaman lobby

- Halaman lobby berisi status sesi saat ini dan form pembelian tiket sederhana
- Form pembelian perlu input nama, ruangan dan sebuah tombol submit yang kemudian mengarah ke generate session.
- Session yang terbentuk adalah **nama** dan **ruang**. Kebutuhan variabel lain bebas.



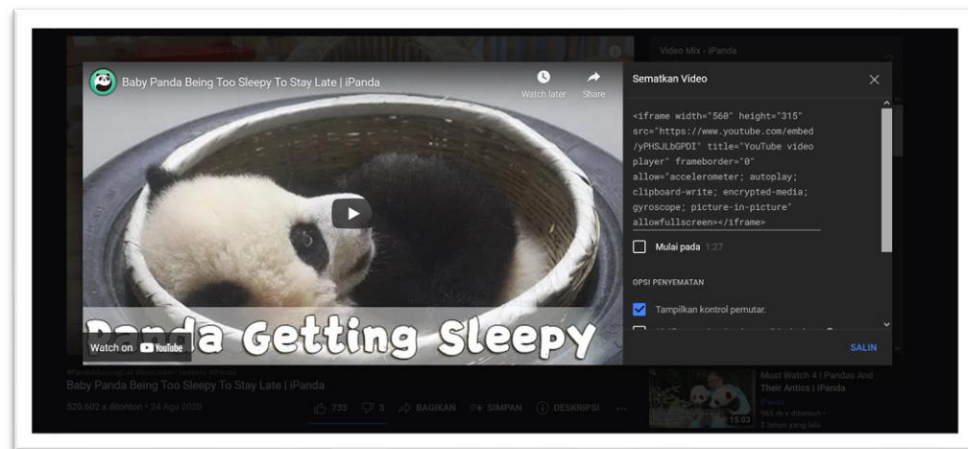
- Apabila pengunjung sudah membeli tiket, **status pengunjung akan berubah** dan **tombol submit akan disabled**.



b. Halaman ruang bioskop

- Sistem memiliki tiga ruang bioskop.

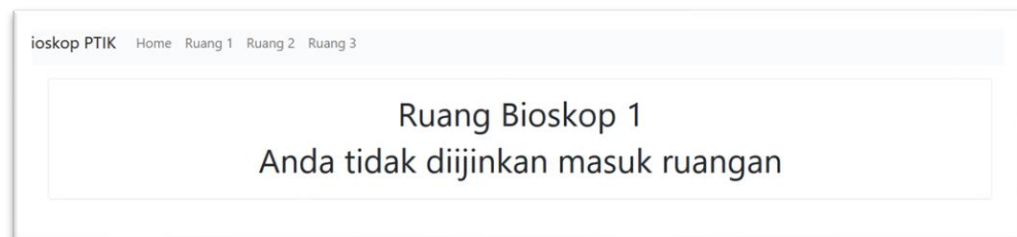
- Masing-masing ruang bioskop berisi sebuah video yang berasal dari youtube. Diambil dengan menggunakan teknik embed video.



- Ruang bioskop hanya dapat diakses apabila pengunjung masih memiliki sesi pembelian tiket dengan kode ruang yang sesuai.



- Jika halaman **diakses tanpa sesi** pembelian tiket yang sesuai, maka akan muncul notifikasi



- Tombol keluar digunakan untuk menghapus session yang sudah terbentuk dan keluar menuju halaman lobby.

Ketentuan Submission: Buat sistem secara utuh dengan mengimplementasikan variabel global \$_SESSION untuk validasi akses ruangan bioskop. Simpan file dengan nama berikut :

- Lobby : p70101.php
- Ruang 1: p70102.php
- Ruang 2: p70103.php
- Ruang 3: p70104.php

..... Silakan lanjutkan apabila ada file tambahan.

Template yang tersedia (Bootstrap) :

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstra
p.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpUuCOMLASjC"
crossorigin="anonymous">
    <title>Bioskop PTIK</title>
  </head>
  <body>
    <header>
      <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fluid">
          <a class="navbar-brand" href="#">Bioskop PTIK</a>
          <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
```

```

        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse"
id="navbarNavAltMarkup">
        <div class="navbar-nav">
            <a class="nav-link" href="#">Home</a>
            <a class="nav-link" href="#">Ruang 1</a>
            <a class="nav-link" href="#">Ruang 2</a>
            <a class="nav-link" href="#">Ruang 3</a>
        </div>
    </div>
</div>
</nav>
</header>
<main>
    <div class="container">
        <div class="row mt-3">
            <div class="card p-3">
                <center>
                    // ISI DENGAN KODEMU
                </center>
            </div>
        </div>
    </main>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.
bundle.min.js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsPlUyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
    </body>
</html>

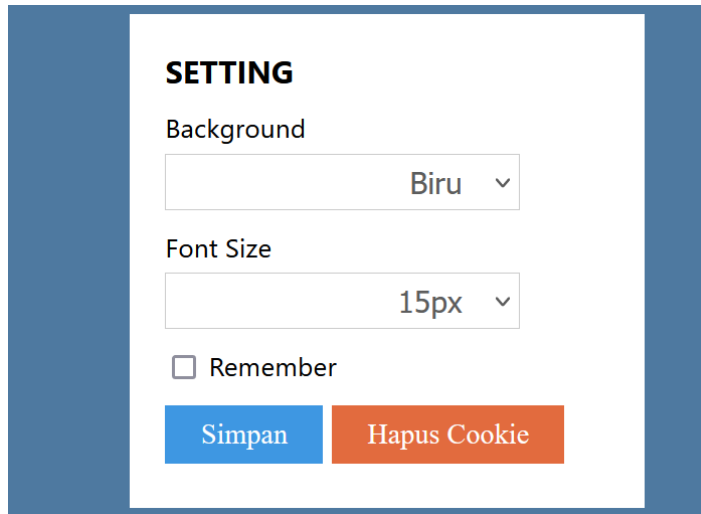
```

2. KEGIATAN 2 : Membuat Tema Web dengan Cookie

Pada kegiatan 2, kita akan membuat panel yang berfungsi untuk mengubah preferensi tampilan dari web yang dibentuk dengan menggunakan cookie. Cookie akan menyimpan variabel background dan font sehingga ketika web ditutup dan diakses kembali, tema yang dipilih terakhir akan tetap digunakan. Adapun dua buah file yang digunakan adalah :

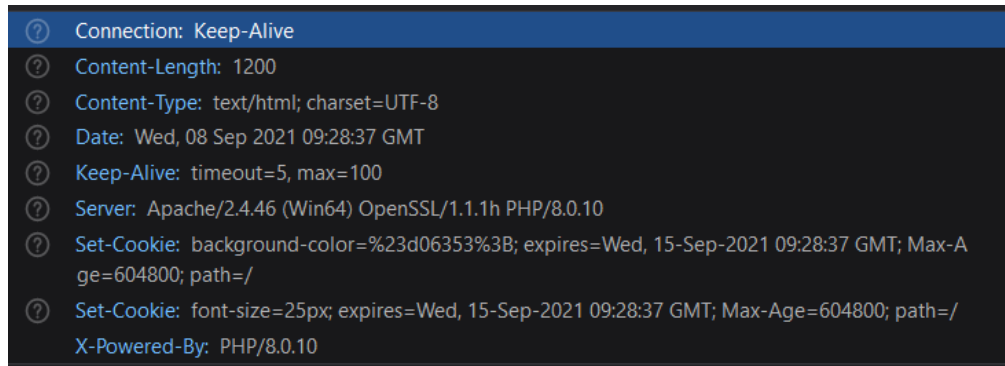
a. Halaman utama

- Halaman utama berisi form setting untuk style web.
- Elemen yang dapat diubah antara lain **ukuran font** dan **warna background**

A screenshot of a web settings panel titled "SETTING". It contains two dropdown menus: "Background" with "Biru" selected and "Font Size" with "15px" selected. Below these is a checkbox labeled "Remember" which is unchecked. At the bottom are two buttons: "Simpan" (blue) and "Hapus Cookie" (orange). The panel is set against a blue background.

- Background yang tersedia adalah : **biru, hijau dan merah.**
- Font size yang tersedia adalah : **15px, 17px, 20px dan 25px.**
- Form setting menyediakan **checkbox** untuk mengingat pilihan style dengan memanfaatkan **cookie**
- **Cookie bertahan selama 1 hari.** Artinya apabila pengunjung keluar dari web dan kemudian masuk lagi ke dalam web tersebut, pengaturan style terakhir masih akan diterapkan selama cookie masih ada yang akan expired dalam waktu 1 hari.
- Terdapat button **Hapus Cookie** untuk **menghapus cookie secara instan.** Sehingga pilihan style terakhir tidak akan tersimpan lagi apabila pengunjung keluar web.

- o Cookie yang terbentuk



b. Style Eksternal

- o File style.css berisi variabel pengaturan gaya dari elemen yang ada di halaman utama.
- o File style.css bersifat eksternal css dengan menggunakan passing parameter **var(--parameter)**.
- o Terapkan passing parameter pada tag **<body>**

Ketentuan Submission: Buat sistem secara utuh dengan mengimplementasikan variabel global `$_COOKIE` untuk menyimpan pilihan style terakhir. Simpan file dengan nama berikut :

- o Halaman utama : p70201.php
- o Style: style.css
- o Silakan lanjutkan apabila ada file tambahan.

Template PHP (native) :

```
<!doctype html>
<html lang="en">
<head>
  <title>Bermain dengan Tema</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

```

<body>
<div class="container">
    <form method="post" action="">
        <h3>SETTING</h3>
        <div>Background</div>
        <select name="background_color">
            <?php
                $colors = array('#4e79a0' => 'Biru', '#75b14a' =>
'Hijau', '#d06353;' => 'Merah');
                foreach ($colors as $name => $value) {
                    $selected = $name == $_POST['background_color']
? 'SELECTED="SELECTED"' : '';
                    echo '<option
value="'. $name. '"'. $selected. '>' . $value. '</option>';
                }
            ?>
        </select>
        <div>Font Size</div>
        <select name="font_size">
            <?php
                $font_sizes = array('15px', '17px', '20px', '25px');
                foreach ($font_sizes as $value) {
                    $selected = $value == $_POST['font_size'] ?
'SELECTED="SELECTED"' : '';
                    echo '<option
value="'. $value. '"'. $selected. '>' . $value. '</option>';
                }
            ?>
        </select>
        <div class="remember">
            <input type="checkbox" id="remember"
name="remember"/>
            <label for="remember"> Remember</label>
        </div>
        <div class="tombol">

```

```

        <input type="submit" class="button blue"
name="submit" value="Simpan"/>

        <input type="submit" class="button red"
name="hapus_cookie" value="Hapus Cookie"/>

    </div>

</form>
</div>
</body>
</html>

```

Template CSS (native) :

```

body {
    font: var(--font_size) "open sans", "segoe ui", tahoma;
    background-color: var(--background_color);
}
h3 {
    margin-top: 0;
    margin-bottom: 10px;
}
div {
    margin-bottom: 5px;
}
select {
    padding: 5px 10px;
    font-size: 17px;
    border: 1px solid #CCCCCC;
    color: #5d5d5d;
    text-align: right;
    width: 200px;
    margin-bottom: 10px;
}
form {
    margin: 0;
}
.container {
    width: 250px;
    margin: auto;
    margin-top: 15px;
    border: 0;
    padding: 20px;
    background-color: #FFFFFF;
}
.button {
    border: 0;
    padding: 7px 20px;
    float: left;
    font-family: "open sans";
    color: #FFFFFF;
    font-size: 16px;

```

```
        margin-right: 5px;
        cursor: pointer;
    }
    .blue {
        background-color: #3e97e2;
    }
    .red {
        background-color: #e26b3e;
    }
    .remember {
        margin-bottom: 12px;
    }
    .tombol::before,
    .tombol::after
    {
        content: "";
        float: none;
        clear: both;
        display: block;
    }
```