

PERTEMUAN 9

PHP (*Cookies dan Session*)

A. Tujuan Pembelajaran

Pada pertemuan ini akan dijelaskan penggunaan *Cookies* dan *Session* pada PHP. Setelah selesai materi pada pertemuan ini, mahasiswa mampu memahami konsep php (*cookies* dan *session*), dan mengimplementasikan konsep php ke dalam web dinamis.

B. Materi

Pengenalan *Cookies*

Cookies adalah sebuah nilai yang dikirimkan dan ditanamkan *server* pada komputer *client*. *Cookies* berupa berkas berukuran kecil umumnya kurang dari 100 bytes, sehingga tidak akan mempengaruhi kecepatan *browsing*. *Cookies* menampung informasi berkaitan dengan pengguna. *Cookies* berguna untuk memudahkan pengguna, seperti "mengingat" pengguna setiap kali mengunjungi website yang sama.

Informasi-informasi yang disimpan dalam *cookies* adalah data-data sederhana tentang kebiasaan pengguna bukan merupakan data penting. Contoh sederhananya saat pengguna mengunjungi sebuah website yang menawarkan tampilan dengan versi bahasa Inggris dan bahasa Indonesia. Apabila pengguna memilih bahasa Indonesia, *cookies* akan secara otomatis menyimpan konfigurasi tersebut. Konfigurasi tersebut akan diingat, sehingga apabila pengguna mengunjungi kembali website tersebut, pengguna tidak perlu memilih jenis bahasa lagi.

Berikut adalah cara kerja *cookie*:

1. Web browser mengirimkan permintaan ke web server. Server web tidak memiliki informasi apa pun tentang web browser. Web server membuat *cookie* dengan mengembalikan nama dan nilai 1 dan melampirkan *cookie* ke *header* respons HTTP.
2. Web browser menyimpan *cookie*.
3. Web browser mengirimkan permintaan kedua dengan *cookie* yang disimpan di *header* permintaan HTTP ke web server. Di web server, PHP dapat mengakses *cookie* dan melakukan sesuatu yang sesuai permintaan.
4. Web server merespons dengan konten permintaan. Biasanya, ia merespons web browser dengan konten berdasarkan nilai *cookie*.



Gambar 1. Cara Kerja Cookie (Sumber: <https://www.phptutorial.net>)

Cookies mempunyai umur, maksudnya setiap data yang disimpan dalam komputer pengguna suatu saat bisa hilang atau musnah. Berdasarkan umurnya *cookies* dibagi menjadi dua jenis, yaitu *session cookies* dan *persistent cookies*. *Session cookies* adalah jenis *cookies* yang tersimpan sementara dan akan hilang saat pengguna menutup *browser*. *Session cookies* ini biasanya digunakan pada fitur “keranjang belanja” di situs belanja online. *Persistent cookies* adalah jenis *cookies* yang tersimpan secara permanen di dalam komputer sampai pengguna memutuskan untuk menghapusnya. *Persistent cookies* umumnya digunakan pada fitur “remember me” saat login.

Tujuan menggunakan *cookies* antara lain:

- Manajemen sesi: *cookies* memungkinkan situs web untuk mengingat pengguna dan informasi login mereka atau apa pun yang harus diingat oleh web server.
- Personalisasi: *cookies* dapat menyimpan preferensi pengguna, tema, dan pengaturan lainnya.
- Pelacakan: *cookies* menyimpan perilaku pengguna. Misalnya, di situs web e-commerce, *cookies* digunakan untuk merekam produk yang dilihat pengguna sebelumnya. Nanti, informasi ini dipakai untuk merekomendasikan produk terkait yang mungkin diminati pengguna.

Pengenalan *Session*

Session adalah salah satu fasilitas yang ada pada PHP yang digunakan untuk menyimpan data sementara ke dalam variabel, sehingga data tadi dapat diakses oleh pengguna selama variabel *session* tersebut tidak dikosongkan atau dihilangkan. *Session* pada prinsipnya hampir sama dengan *cookies*, yang membedakan adalah posisi

penyimpanan data. Jika *cookies* menyimpan data pada *client*, data *session* akan disimpan pada sisi *server*. Jadi *session* relatif lebih aman digunakan untuk menyimpan variabel nilai yang bersifat rahasia seperti *username* dan *password* pada saat login. Contoh penggunaan *session* diantaranya untuk menyimpan informasi login yang berlaku hanya dalam satu sesi dan menyimpan catatan pemesanan barang dalam sistem *e-commerce*/transaksi online.

Ketika kita menggunakan aplikasi di komputer, kita dapat membuka, melakukan perubahan, dan menutupnya. Dan komputer tahu detail pengguna yang menggunakan aplikasi tersebut. Tetapi pada penggunaan internet, terdapat masalah yaitu web server tidak tahu informasi pengguna dan apa yang pengguna lakukan, karena alamat HTTP tidak mempertahankan status. *Session* memecahkan masalah ini dengan menyimpan informasi pengguna untuk digunakan di beberapa halaman (misalnya nama pengguna, tema, dll). Secara *default*, variabel *session* berlangsung hingga pengguna menutup browser. Sehingga variabel *session* menyimpan informasi tentang satu pengguna tunggal, dan tersedia untuk semua halaman dalam satu aplikasi.

Tabel berikut menjelaskan perbandingan *cookies* dan *session*:

Tabel 1 Perbandingan *Cookies* dan *Session*

Jenis	Menyimpan Informasi	Lokasi Penyimpanan	Dapat dipanggil	Bersifat Global	Dapat dipanggil Lain Waktu	Dibatasi Browser	Dapat diedit User
<i>Cookies</i>	Ya	<i>Client</i>	Ya	Ya	Ya	Ya	Ya
<i>Session</i>	Ya	<i>Server</i>	Ya	Ya	Tidak	Tidak	Tidak

Praktikum Bagian 1. Membuat *Cookies*

PHP memiliki fungsi yang dapat digunakan untuk membuat dan mengambil nilai *cookies*. *Cookies* harus dideklarasikan sebelum halaman ditampilkan, yang artinya dituliskan sebelum tag <html>. *Cookies* dibuat menggunakan fungsi `setcookie()`. Berikut adalah sintaks dasar fungsi `setcookie()`:

```
<?php
    setcookie(name, value, expire, path, domain, secure, httponly);
?>
```

- `name`, adalah nama *cookies* atau variabel yang menjadi pengenalan *cookies*. Variabel ini wajib diisi, sedangkan lainnya adalah opsional.
- `value`, berisi nilai yang tersimpan dalam *cookies*
- `expire`, adalah jangka waktu *cookies* akan disimpan di dalam computer. Jika tidak diatur atau bernilai 0, *cookies* akan hilang saat browser ditutup.
- `path`, adalah path web server dimana *cookies* akan tersedia
- `domain`, domain dimana *cookies* tersedia
- `secure`, jika bernilai TRUE, maka *cookies* dikirim melalui koneksi *secure* HTTP (HTTPS) dari web browser.
- `httponly`, jika bernilai TRUE, maka *cookies* hanya bisa diakses melalui HTTP protokol dan bukan Javascript.

Setelah mengetahui cara membuat *cookies*, selanjutnya nilai *cookies* yang telah di buat dapat diambil dengan menggunakan variabel `$_COOKIE`. Berikut ini adalah sintaks untuk mengambil nilai *cookies*:

```
<?php
    $_COOKIE[name];
?>
```

Langkah	Keterangan
1	Buka folder praktik_php dan buat file baru dengan nama buatCookies.php, kemudian ketikkan kode berikut:

	<pre> 1 <?php 2 \$name = "mahasiswa"; 3 \$value = "Zia Zidani"; 4 setcookie(\$name, \$value, time()+600); 5 6 setcookie("kelas","MI-1H", time()+3600); 7 ?> 8 <html> 9 <body> 10 <?php 11 echo "Cookies telah dibuat"; 12 ?> 13 </body> 14 </html> </pre>
2	<p>Buat file baru dengan nama panggilCookies.php di dalam folder praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 echo \$_COOKIE["mahasiswa"]; 3 echo "
"; 4 echo \$_COOKIE["kelas"]; 5 ?> </pre>
3	Buka <i>browser</i> dan jalankan kode program pada langkah ke-2 dengan mengetikkan localhost/dasarWeb/praktik_php/panggilCookies.php
4	Berikan tangkapan layar pada browser dan jelaskan hasil pengamatan Anda (soal no 1)
5	Buka <i>browser</i> dan jalankan kode program langkah ke-1 dengan mengetikkan localhost/dasarWeb/praktik_php/buatCookies.php
6	Ulangi langkah ke-3
7	Berikan tangkapan layar hasil pada browser dan jelaskan hasil pengamatan Anda (soal no 2)

Praktikum Bagian 2. Mengubah Nilai Cookies

Langkah	Keterangan
1	Buka folder praktik_php dan buat file baru dengan nama ubahCookies.php, kemudian ketikkan kode berikut (isi nama dan kelas kalian sendiri):

	<pre> 1 <?php 2 \$name = "mahasiswa"; 3 \$value = "Tulis nama kalian"; 4 setcookie(\$name, \$value, time()+600); 5 6 setcookie("kelas", "tulis kelas kalian", time()+3600); 7 ?> 8 <html> 9 <body> 10 <?php 11 echo "Cookies telah diubah"; 12 ?> 13 </body> 14 </html> </pre>
2	Buka <i>browser</i> dan jalankan kode program pada langkah ke-1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/ubahCookies.php</code>
3	Selanjutnya jalankan kode pada Bagian 1 yaitu <code>panggilCookies.php</code> dengan mengetikkan <code>localhost/dasarWeb/praktik_php/panggilCookies.php</code>
4	Berikan tangkapan layar pada browser dan jelaskan hasil pengamatan Anda (soal no 3)

Praktikum Bagian 3. Menghapus *Cookies*

Untuk menghapus *cookie*, gunakan fungsi `setcookie()` dengan waktu kedaluwarsa.

Langkah	Keterangan
1	<p>Buka folder <code>praktik_php</code> dan buat file baru dengan nama <code>hapusCookies.php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 setcookie("mahasiswa", "", time()-600); 3 setcookie("kelas", "", time()-3600); 4 ?> 5 <html> 6 <body> 7 <?php 8 echo "Cookies mahasiswa telah dihapus
"; 9 echo "Cookies kelas telah dihapus"; 10 ?> 11 </body> 12 </html> </pre>
2	Buka <i>browser</i> dan jalankan kode program pada langkah ke-1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/hapusCookies.php</code>

3	Selanjutnya jalankan kode pada Bagian 1 yaitu panggilCookies.php dengan mengetikkan localhost/dasarWeb/praktik_php/panggilCookies.php
4	Berikan tangkapan layar pada browser dan jelaskan hasil pengamatan Anda (soal no 4)

Praktikum Bagian 4. Penerapan Cookies pada Fitur Keranjang Belanja

Salah satu contoh pemanfaatan *cookies* adalah pada fitur “keranjang belanja” pada aplikasi web toko online. Keranjang belanja berisi barang-barang yang akan dibeli oleh user. *Cookies* digunakan untuk mengingat jumlah barang yang dipilih user. Berikut adalah contoh penggunaan *cookies* pada fitur keranjang belanja:

Langkah	Keterangan
1	<p>Buat file baru dengan nama formBeli.html di dalam folder praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 <title>Form Pembelian</title> 4 </head> 5 <body> 6 <form action="prosesBeli.php" method="POST"> 7 <label>Jumlah gula yang dibeli: </label> 8 <input type="number" name="gula"> 9
 10 <label>Jumlah minyak yang dibeli: </label> 11 <input type="number" name="minyak"> 12
 13 <input type="submit" value="Tambahkan ke keranjang"> 14 </form> 15 </body> 16 </html> </pre>
2	<p>Buat file baru dengan nama prosesBeli.php di dalam folder praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 if(isset(\$_POST["gula"]) && isset(\$_POST["minyak"])){ 3 setcookie("jumlahgula", \$_POST["gula"]); 4 setcookie("jumlahminyak", \$_POST["minyak"]); 5 header("location:keranjang.php"); 6 } 7 > </pre>
3	Selanjutnya buat file baru dengan nama keranjangBelanja.php di dalam folder praktik_php, kemudian ketikkan kode berikut:

	<pre> 1 <html> 2 <head> 3 <title>Keranjang Pembelian</title> 4 </head> 5 <body> 6 <h2>Jumlah yang dibeli</h2> 7 <?php 8 \$jmlGula = \$_COOKIE["jumlahgula"]; 9 \$jmlMinyak = \$_COOKIE["jumlahminyak"]; 10 11 echo "Jumlah gula yang dibeli \$jmlGula
"; 12 echo "Jumlah minyak yang dibeli \$jmlMinyak"; 13 ?> 14 </body> 15 </html> </pre>
4	<p>Buka <i>browser</i> dan jalankan kode program formBeli.html dengan mengetikkan localhost/dasarWeb/praktik_php/panggilCookies.php Isilah form dengan memasukkan angka sesuai jumlah yang ingin dibeli. Berikan tangkapan layar pada browser (<i>soal no 5</i>)</p>
5	<p>Klik tombol tambahkan ke keranjang. Berikan tangkapan layar pada browser dan jelaskan hasil pengamatan Anda (<i>soal no 6</i>)</p>

Praktikum Bagian 5. Membuat Session

Untuk membuat *session* dalam PHP digunakan fungsi `session_start()`. Berikut adalah sintaks dasar untuk memulai *session*:

```

<?php
    session_start();
?>

```

Setelah *session* dimulai, maka variabel *session* sudah dapat mulai digunakan. Variabel yang dimaksud adalah PHP global variable `$_SESSION`. Berikut adalah sintaks penggunaan variable global `$_SESSION`:

```

<?php
    $_SESSION[name];
?>

```


Langkah	Keterangan
1	<p>Buat file baru dengan nama buatSession.php di dalam folder praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 ?> 4 <html> 5 <body> 6 <?php 7 \$_SESSION["makananfav"] = "Rujak"; 8 \$_SESSION["minumanfav"] = "Es Teler"; 9 echo "Session telah diset"; 10 ?> 11 </body> 12 </html> </pre>
2	<p>Buat file baru dengan nama panggilSession.php di dalam folder praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start() 3 ?> 4 <html> 5 <body> 6 <?php 7 echo "Makanan favorit saya adalah " . \$_SESSION["makananfav"] . "
"; 8 echo "Minuman favorit saya adalah " . \$_SESSION["minumanfav"] . "
"; 9 10 print_r(\$_SESSION); 11 ?> 12 </body> 13 </html> </pre>
3	Buka <i>browser</i> dan jalankan kode program pada langkah ke-2 dengan mengetikkan localhost/dasarWeb/praktik_php/panggilSession.php
4	Berikan tangkapan layar pada browser dan jelaskan hasil pengamatan Anda (soal no 7)
5	Buka <i>browser</i> dan jalankan kode program langkah ke-1 dengan mengetikkan localhost/dasarWeb/praktik_php/buatSession.php
6	Ulangi langkah ke-3
7	Berikan tangkapan layar hasil pada browser dan jelaskan hasil pengamatan Anda (soal no 8)

Praktikum Bagian 6. Mengubah Nilai Session

Langkah	Keterangan
1	<p>Buat file baru dengan nama ubahSession.php di dalam folder praktik_php, kemudian ketikkan kode berikut (ubah sesuai makanan favorit, tidak harus seperti contoh):</p> <pre> 1 <?php 2 session_start(); 3 ?> 4 <html> 5 <body> 6 <?php 7 \$_SESSION["makananfav"] = "Sate Padang"; 8 9 echo "Session makananfav telah diubah"; 10 ?> 11 </body> 12 </html> </pre>
2	Buka <i>browser</i> dan jalankan kode program pada langkah ke-2 dengan menetikkan localhost/dasarWeb/praktik_php/ubahSession.php
3	Buka <i>browser</i> dan jalankan kode program pada langkah ke-2 dengan menetikkan localhost/dasarWeb/praktik_php/panggilSession.php
4	Berikan tangkapan layar pada browser dan jelaskan hasil pengamatan Anda (soal no 9)

Praktikum Bagian 7. Menghapus Session

Pada bagian ini akan dibahas mengenai cara menghapus session. PHP menyediakan fungsi `session_unset()` untuk menghapus semua variable session. Sedangkan `session_destroy()` dapat digunakan untuk menghapus session. Berikut adalah langkah-langkah untuk menghapus session:

Langkah	Keterangan
1	<p>Buka folder praktik_php dan buat file baru dengan nama hapusSession.php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 ?> 4 <html> 5 <body> 6 <?php 7 // menghapus semua variabel session 8 session_unset(); 9 10 // menghapus session 11 session_destroy(); 12 13 echo "Session telah dihapus"; 14 ?> 15 16 </body> 17 </html> </pre>

2	Buka <i>browser</i> dan jalankan kode program pada langkah ke-1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/hapusSession.php</code>
3	Selanjutnya jalankan kode pada Bagian 1 yaitu <code>panggilCookies.php</code> dengan mengetikkan <code>localhost/dasarWeb/praktik_php/panggilSession.php</code>
4	Berikan tangkapan layar pada browser dan jelaskan hasil pengamatan Anda (soal no 10)

Praktikum Bagian 8. Penerapan Session Pada Fitur Login

Pada bagian ini, kita akan menerapkan penggunaan session untuk fitur login. Form login akan memeriksa email dan password pengguna. Karena kita belum menggunakan basis data, maka ditentukan bahwa untuk login digunakan `username="admin"` dan `password="1234"`

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>loginForm.html</code> di dalam folder <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <!DOCTYPE HTML> 2 <html> 3 <head> 4 <title>Login Page</title> 5 </head> 6 <body> 7 <form action="loginProses.php" method="POST"> 8 <table> 9 <tr> 10 <td>Username:</td> 11 <td><input type="text" name="username" size="20"></td> 12 </tr> 13 <tr> 14 <td>Password:</td> 15 <td><input type="password" name="password" size="20"></td> 16 </tr> 17 <tr> 18 <td>&nbsp;</td> 19 <td><input type="submit" value="Login"></td> 20 </tr> 21 </table> 22 </form> 23 </body> 24 </html> </pre>
2	Selanjutnya buat file untuk memproses login dengan nama <code>loginProses.php</code> di dalam folder <code>praktik_php</code> , kemudian ketikkan kode berikut:

	<pre> 1 <?php 2 \$user = \$_POST["username"]; 3 \$pass = \$_POST["password"]; 4 5 if(\$user=="admin" && \$pass=="1234"){ 6 session_start(); 7 \$_SESSION["username"] = \$user; 8 \$_SESSION["role"] = "Administrator"; 9 header("Location:homeSession.php"); 10 } 11 else{ 12 echo "Username atau password tidak sesuai
"; 13 echo 'Kembali ke Halaman Login'; 14 } 15 ?> </pre>
3	<p>Jika berhasil login, maka akan masuk ke halaman homeSession.php. Buat file dengan nama homeSession.php di dalam folder praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 ?> 4 <html> 5 <head> 6 <title>Home Page</title> 7 </head> 8 <body> 9 <?php 10 if(isset(\$_SESSION["username"])){ 11 echo "Selamat datang ". \$_SESSION["username"] . "
"; 12 echo "Anda sebagai ". \$_SESSION["role"] . "
"; 13 echo 'Logout'; 14 } 15 else{ 16 echo "Anda belum Login
"; 17 echo 'Login'; 18 } 19 ?> 20 </body> 21 </html> </pre>
4	<p>Jika sudah login, maka pengguna bisa logout. Buat file untuk logout dengan nama sessionLogout.php di dalam folder praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 session_destroy(); 4 5 echo "You have successfully logged out" 6 ?> </pre>
5	<p>Buka <i>browser</i> dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/loginForm.html</p>
6	<p>Login menggunakan username = nama Anda dan password = NIM anda Berikan tangkapan layar hasil dari browser dan jelaskan pengamatan Anda (soal no 11)</p>

7	Kembalilah ke halaman login form, kemudian lakukan login menggunakan username = admin dan password = 1234
8	Berikan tangkapan layar hasil pada browser dan jelaskan hasil pengamatan Anda (soal no 12)
9	Selanjutnya lakukan logout. Kemudian buka homeSession dengan mengetikkan localhost/dasarWeb/praktik_php/homeSession.php Berikan tangkapan layar hasil pada browser dan jelaskan hasil pengamatan Anda (soal no 13)
10	Jelaskan urutan proses dari login hingga logout (sebutkan juga urutan file yang diproses) (soal no 14)

Referensi :

- 1) Nixon, Robin. (2018). Learning PHP, MySQL, JavaScript, CSS & HTML: A Step-by-step Guide to Creating Dynamic Websites, 5th Edition. O'Reilly Media, Inc.
- 2) Forbes, Alan. (2012). The Joy of PHP: A Beginners's Guide to Programming Interactive Web Applications with PHP and MySQL, 5th Edition. Plum Island Publishing
- 3) <https://www.phptutorial.net/php-tutorial/php-cookies/>
- 4) https://www.w3schools.com/php/php_sessions.asp