**PERTEMUAN 9**

**PHP (*Cookies* dan *Session*)**

1. 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.

1. Materi

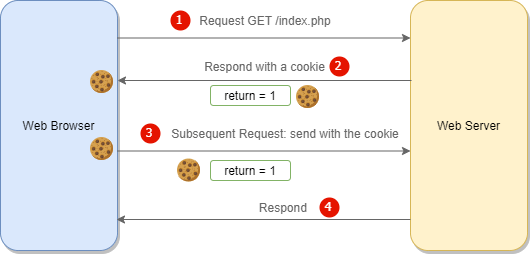
**Pengenalan *Cookies***

*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 infomasi berkaitan dengan pengguna. *Cookies* bergunauntuk 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 mengujungi 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: *cookise* 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* dugunakan 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 sememntara 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 informasilogin 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*** |
| ***Cookies*** | Ya | *Client* | Ya | Ya | Ya | Ya | Ya |
| ***Session*** | Ya | *Server* | 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 pengenal *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: |
| 2 | Buat file baru dengan nama panggilCookies.php di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Buka *browser* 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 *browser* 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): |
| 2 | Buka *browser* dan jalankan kode program pada langkah ke-1 dengan mengetikkan localhost/dasarWeb/praktik\_php/ubahCookies.php |
| 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 3) |

**Praktikum Bagian 3. Menghapus *Cookies***

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

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buka folder praktik\_php dan buat file baru dengan nama hapusCookies.php, kemudian ketikkan kode berikut: |
| 2 | Buka *browser* dan jalankan kode program pada langkah ke-1 dengan mengetikkan localhost/dasarWeb/praktik\_php/hapusCookies.php |
| 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 | Buat file baru dengan nama formBeli.html di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 2 | Buat file baru dengan nama prosesBeli.php di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Selanjutnya buat file baru dengan nama keranjangBelanja.php di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 4 | Buka *browser* 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 (soal no 5) |
| 5 | Klik tombol tambahkan ke keranjang. Berikan tangkapan layar pada browser dan jelaskan hasil pengamatan Anda (soal no 6) |

**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 penggunakan variable global $\_SESSION:

|  |
| --- |
| <?php  $\_SESSION[name];  ?> |

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama buatSession.php di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 2 | Buat file baru dengan nama panggilSession.php di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Buka *browser* 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 *browser* 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 | Buat file baru dengan nama ubahSession.php di dalam folder praktik\_php, kemudian ketikkan kode berikut (ubah sesuai makanan favorit, tidak harus seperti contoh): |
| 2 | Buka *browser* dan jalankan kode program pada langkah ke-2 dengan mengetikkan  localhost/dasarWeb/praktik\_php/ubahSession.php |
| 3 | Buka *browser* 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 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 | Buka folder praktik\_php dan buat file baru dengan nama hapusSession.php, kemudian ketikkan kode berikut: |
| 2 | Buka *browser* dan jalankan kode program pada langkah ke-1 dengan mengetikkan localhost/dasarWeb/praktik\_php/hapusSession.php |
| 3 | Selanjutnya jalankan kode pada Bagian 1 yaitu panggilCookies.php dengan mengetikkan localhost/dasarWeb/praktik\_php/panggilSession.php |
| 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 | Buat file baru dengan nama loginForm.html di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 2 | Selanjutnya buat file untuk memroses login dengan nama loginProses.php di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 3 | 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: |
| 4 | Jika sudah login, maka pengguna bisa logout. Buat file untuk logout dengan nama sessionLogout.php di dalam folder praktik\_php, kemudian ketikkan kode berikut: |
| 5 | Buka *browser* dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/loginForm.html |
| 6 | Login menggunakan username = nama Anda dan password = NIM anda  Berikan tangkapan layar hasil dari browser dan jelaskan pengamatan Anda (soal no 11) |
| 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