|  |  |
| --- | --- |
|  | Jurusan Teknologi Informasi Politeknik Negeri Malang  **Jobsheet-13: PHP (Login, Cookies, Session)**  **Mata Kuliah Desain dan Pemrograman Web**  Pengampu: Tim Ajar Desain dan Pemrograman Web  *Oktober 2023* |

**Topik**

* Pengenalan konsep login
* Multi user login
* Pengenalan konsep cookies dan session

**Tujuan**

Mahasiswa diharapkan mampu:

1. Memahami konsep login *single* *user* dan *multiuser*
2. Memahami konsep dasar *cookies* dan penggunaannya
3. Memahami konsep dasar *session* dan penggunaannya
4. Mengimplementasikan konsep PHP dan database MySQL ke dalam web dinamis

**Pendahuluan**

**Pengenalan Login**

Saat ini fitur login menjadi fitur yang banyak digunakan di dalam pengembangan aplikasi web modern. Login diartikan sebagai proses masuk ke jaringan komputer. Seorang *user* harus memasukkan identitas akun minimal terdiri dari *username*/akun pengguna dan *password* untuk mendapatkan hak akses. Kemudian sistem akan melakukan pencocokan *username* dan *password* yang diinput dengan data *username* dan *password* yang tersimpan di dalam *database*.

Di dalam praktikum ini akan dibahas mengenai login *single* dan *multiuser*. Login *multiuser* adalah proses login yang memiliki validasi level dari seorang user saat login. Pengaturan validasi level ini berkaitan dengan hak akses yang akan diberikan kepada user yang login ke dalam sistem. Di dalam login multiuser terdapat istilah level akses. Level inilah yang akan menentukan tampilan mana atau aksi apa saja yang bisa dilakukan oleh *user* tersebut.

Contoh penerapan login *multiuser* dapat kita lihat pada web lms.jti.polinema.ac.id. Terdapat beberapa level akses untuk *user* yang akan memberikan hak akses yang berbeda dan dapat melakukan aksi yang berbeda di dalam web tersebut. Berikut adalah level akses dan aksi yang dapat dilakukan setiap *user* di dalam web lms.jti.polinema.ac.id:

1. *Student*

Level akses *student* dapat melakukan beberapa aksi yaitu, mengedit profil, melihat mata kuliah*,* mengunduh (*download*) materi, mengumpulkan (*upload*) tugas, dan mengerjakan kuis.

1. *Teacher*

Level akses *teacher* dapat melakukan beberapa aksi yaitu, mengedit topik/bab pada mata kuliah, menambahkan aktivitas (*upload* materi, *jobsheet* praktikum, *submission* tugas, kuis), dan melakukan penilaian pada tugas.

1. *Administrator*

Level akses *administrator* dapat melakukan beberapa yaitu, mendaftarkan *user*, membuat dan mengatur mata kuliah, dan mengatur *grouping* tiap mata kuliah.

**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. 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.

*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.

**Pengenalan *Session***

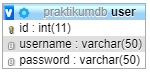
*Session* adalah salah satu fasilitas yang ada pada PHP yang digunakan untuk menyimpan data sememntara ke dalam variabel, sehingga data tadi dapat diakses oleh pengguna selama variabel *session* tersebut tidak dikosongkan atau dihilangkan. *Session* pada prinsipnya hampir sama dengan *cookies,* yang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

Tahapan pertama dalam membuat fitur login adalah membuat database. Pada praktikum kali ini, kita akan menggunakan *database* yang telah dibuat pada praktikum sebelumnya, yaitu *database* yang bernama “prakwebdb”. Selanjutnya hal yang harus dilakukan adalah membuat tabel *user*. Tabel *user* minimal terdiri dari *username* dan *password*. Berikut adalah ketentuan *username* dan *password* yang dibuat di tabel user:



* tabel bernama user
* kolom tabel terdiri dari id, *username*, dan *password*
* id memiliki tipe data integer dan di atur sebagai *primary* *key*
* *username* dan *password* memiliki tipe data varchar
* *password* akan dienkripsi menggunakan fungsi md5

MD5 adalah sebuah algoritma fungsi hash dari kriptografi yang digunakan dengan hash value 128-bit. MD5 telah digunakan untuk berbagai macam keperluan dalam membangun keamanan sebuah aplikasi, salah satunya digunakan untuk validasi login pada sistem. MD5 digunakan untuk mengubah nilai asli menjadi serangkaian nilai yang terenkripsi atau acak. Sintak MD5 di dalam PHP adalah:

|  |
| --- |
| <?php  echo md5(‘string’);  ?> |

Ikuti langkah-langkah berikut untuk melakukan persiapan dalam pembuatan tabel *user*:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buka XAMPP kemudian jalankan web server dan database dengan mengklik tombol start pada modul Apache dan MySQL |
| 2 | Buka browser kemudian ketikkan localhost/phpmyadmin |
| 3 | Berikut adalah tampilan halaman phpmyadmin |
| 4 | Buka database “prakwebdb” kemudian buat tabel baru bernama “user” dengan kolom “3”  D:\tempsnip.png |
| 6 | Berikut ini adalah kolom-kolom dalam tabel user |
| 7 | Masukkan data sebagai berikut ke dalam tabel user |
| 8 | Tunjukkan hasil *screenshot* isi tabel user (soal no 1) |
| 9 | Tuliskan *query* untuk membuat tabel user menggunakan fungsi mysqli\_query() (soal no 2) |
| 10 | Tuliskan *query* untuk memasukkan dan ke tabel user menggunakan fungsi mysqli\_query() (soal no 3) |

**Praktikum Bagian 2. Login *Single* *User***

Pada praktikum ini akan dijelaskan langkah-langkah untuk membuat form login single user. Cara kerja login single user adalah dengan mencocokan *username* dan *password* yang diinputkan pada form login dengan data *username* dan *password* yang telah disimpan pada tabel user.

Ikuti langkah-langkah berikut untuk membuat login *single user*:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Pastikan file koneksi.php dari jobsheet sebelumnya, tersimpan di dalam direktori praktik\_php. |
| 2 | Buat file dengan nama loginForm.html di dalam direktori praktik\_php, kemudian ketikkan kode program berikut ini: |
| 3 | Buat file dengan nama loginProses.html di dalam direktori praktik\_php, kemudian ketikkan kode program berikut ini: |
| 4 | Buat file dengan nama homeAdmin.html di dalam direktori praktik\_php, kemudian ketikkan kode program berikut ini: |
| 5 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/loginForm.php |
| 6 | Lakukan login dengan mengetikkan username dan password yang datanya belum tersimpan di database. Jelaskan hasil pengamatanmu (soal no 4) |
| 7 | Lakukan login dengan mengetikkan username dan password yang datanya sudah tersimpan di database. Jelaskan hasil pengamatanmu (soal no 5) |

**Praktikum Bagian 3. Menambahkan Kolom pada Tabel**

Setelah berhasil membuat login untuk *single* *user*, selanjutnya akan dibahas langkah-langkah untuk membuat login multiuser. Pertama yang perlu dipersiapkan adalah memberikan level akses di dalam *database*. Sebelumnya kita hanya memiliki kolom id, *username*, dan *password* di dalam tabel user. Untuk membuat login *multiuser* perlu ditambahkan kolom level akses. Untuk menambahkan kolom pada tabel digunakan perintah ALTER TABLE. Berikut ini adalah sintak dasar ALTER TABLE:

|  |
| --- |
| ALTER TABLE namaTabel  ADD namaKolom tipeData |

Ikuti langkah-langkah berikut untuk menambahkan kolom pada *database*

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Pada tabel user, ketikkan perintah SQL seperti berikut ini: |
| 2 | Edit data admin, ubah level menjadi “1” |
| 3 | Tambahkan data guest pada tabel user, dengan rincian data sebagai berikut: |
| 4 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 6) |
| 5 | Tunjukkan hasil *screenshot* isi tabel user (soal no 7) |

**Praktikum Bagian 4. Login *Multiuser***

Pada praktikum ini, akan dijelaskan langkah-langkah untuk membuat login *multiuser.* Berbeda dengan login sebelumnya, selain melakukan pencocokan *username* dan *password* yang diinputkan melalui form dengan data *username* dan *password* yang disimpan di *database*, login *multiuser* akan mengenali *user* yang login dan menampilkan halaman yang sesuai dengan hak aksesnya. Pada praktikum ini terdapat 2 level login, yaitu level 1 untuk admin dan level 2 untuk guest.

Ikuti langkah-langkah berikut untuk memahami cara membuat login *multiuser*:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Praktikum bagian 4 menggunakan file koneksi.php, sehingga pastikan file tersebut berada di dalam folder praktik\_php. |
| 2 | Buka file loginForm.html kemudian modifikasi kode program dengan mengubah nilai atribut action di baris ke-5 menjadi loginMultiProses.php, sehingga kode program menjadi seperti berikut ini: |
| 3 | Buat file dengan nama loginMultiProses.php di dalam direktori praktik\_php, kemudian ketikkan kode program berikut ini: |
| 4 | Buat file dengan nama homeGuest.html di dalam direktori praktik\_php, kemudian ketikkan kode program berikut ini: |
| 5 | Lakukan login dengan mengetikkan data admin. Jelaskan hasil pengamatanmu (soal no 8) |
| 6 | Lakukan login dengan mengetikkan data guest. Jelaskan hasil pengamatanmu (soal no 9) |
| 7 | Lakukan login dengan mengetikkan username dan password yang tidak tersimpan di database. Jelaskan hasil pengamatanmu (soal no 10) |

**Praktikum Bagian 5. 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);  ?> |

name, adalah nama *cookies* atau variabel yang menjadi pengenal *cookies*

value, berisi nilai yang tersimpan dalam *cookies*

expire, adalah jangka waktu *cookies* akan disimpan di dalam komputer

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[‘user’];  ?> |

Ikuti langkah berikut untuk memahami cara kerja *cookies*:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Jalankan Apache pada XAMPP. |
| 2 | Buat file baru dengan nama cookiesCreate.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Buat file baru dengan nama cookiesCall.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 4 | Buka *browser* dan jalankan kode program pada langkah ke-3 dengan mengetikkan  localhost/dasarWeb/praktik\_php/cookiesCall.php |
| 5 | Amati dan jelaskan hasil pengamatanmu (soal no 11) |
| 6 | Buka *browser* dan jalankan kode program langkah ke-2 dengan mengetikkan  localhost/dasarWeb/praktik\_php/cookiesCreate.php |
| 7 | Ulangi langkah ke-4 |
| 8 | Amati dan jelaskan hasil yang ditampilkan (soal no 12) |
| 9 | *Restart* komputer anda |
| 10 | Setelah komputer menyala, nyalakan kembali Apache pada XAMPP |
| 11 | Buka *browser* yang sama dengan sebelumnya kemudian ulangi langkah ke-4 |
| 12 | Amati dan jelaskan hasil yang ditampilkan (soal no 13) |

**Praktikum Bagian 6. Menghapus Nilai *Cookies***

Pada praktikum ini akan dibahas mengenai cara menghapus nilai *cookies.* Jika pada praktikum bagian sebelumnya *cookies* di set dengan waktu *expire* time()+3600, maka untuk menghapus nilai *cookies* adalah dengan sebagai berikut:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama cookiesDel.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 2 | Buka *browser* dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/cookiesDel.php |
| 3 | Buka *browser* dan jalankan kode program dari praktikum bagian 1 dengan mengetikkan  localhost/dasarWeb/praktik\_php/cookiesCall.php |
| 4 | Amati dan jelaskan hasil yang dihasilkan dari langkah 2 dan 3, kemudian beri kesimpulan (soal no 14) |

**Praktikum Bagian 7. 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 direktori praktik\_php, kemudian ketikkan kode berikut: |
| 2 | Buat file baru dengan nama prosesBeli.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Buat file baru dengan nama prosesBeli.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 4 | Buka *browser* dan jalankan kode program langkah ke-3 dengan mengetikkan  localhost/dasarWeb/praktik\_php/keranjangBelanja.php |
| 5 | Amati dan jelaskan hasil yang ditampilkan (soal no 15) |
| 6 | Jalankan kode program langkah ke-1 dengan mengetikkan  localhost/dasarWeb/praktik\_php/formBeli.html |
| 7 | Isikan jumlah novel dan buku teks yang akan dibeli kemudian klik tombol “submit” |
| 8 | Amati dan jelaskan hasil yang ditampilkan (soal no 16) |
| 9 | Tutup *browser* kemudian buka kembali *browser* kemudian jalankan kembali kode program langkah ke-3 dengan mengetikkan  localhost/dasarWeb/praktik\_php/keranjangBelanja.php |
| 10 | Amati dan jelaskan hasil yang ditampilkan (soal no 17) |

**Praktikum Bagian 8. 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[‘namaVariabel’];  ?> |

Ikuti langkah berikut untuk memahami cara kerja *session*:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama sessionCreate.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 2 | Buat file baru dengan nama sessionCall.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Buka *browser* dan jalankan kode program pada langkah ke-2 dengan mengetikkan  localhost/dasarWeb/praktik\_php/sessionCall.php |
| 4 | Buka *browser* dan jalankan kode program langkah ke-1 dengan mengetikkan  localhost/dasarWeb/praktik\_php/sessionCreate.php |
| 5 | Ulangi langkah ke-3 |
| 6 | Amati dan jelaskan hasil yang ditampilkan (soal no 18) |

**Praktikum Bagian 9. Menghapus Nilai *Session***

Pada praktikum ini akan dibahas mengenai cara menghapus *session.* PHP menyediakan fungsi session\_destroy() yang dapat digunakan untuk menghapus *session*. Berikut adalah langkah-langkah untuk menghapus *session*:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama sessionDel.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 2 | Buka *browser* dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/sessionDel.php |
| 3 | Buka *browser* dan jalankan kode program dari praktikum bagian 1 dengan mengetikkan  localhost/dasarWeb/praktik\_php/sessionCall.php |
| 4 | Amati dan jelaskan hasil yang ditampilkan (soal no 19) |

**Praktikum Bagian 10. Penerapan *Session* pada Fitur Login**

Ikuti langkah-langkah berikut untuk mengetahui penerapan *session* pada fitur login:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Jalankan MySQL pada XAMPP |
| 2 | Pastikan file koneksi.php yang telah dibuat pada praktikum jobsheet sebelumnya ada di dalam direktori praktik\_php |
| 3 | Buat file baru dengan nama sessionLoginForm.html di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 4 | Buat file baru dengan nama sessionLoginProses.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 5 | Buat file baru dengan nama homeSession.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 6 | Buat file baru dengan nama sessionLogout.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 7 | Buka *browser* dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/sessionLoginForm.html |
| 8 | Login menggunakan username admin. |
| 9 | Amati dan jelaskan hasil yang ditampilkan (soal no 20) |
| 10 | Jelaskan urutan proses dari login hingga logout (sebutkan juga urutan file yang diproses) (soal no 21) |

**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