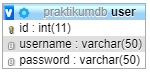
**Praktikum Bagian 1. Membuat Tabel User**

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)  CREATE TABLE user(  Id INT PRIMARY KEY,  Username VARCHAR(50),  Password VARCHAR(50)); |
| 10 | Tuliskan *query* untuk memasukkan dan ke tabel user menggunakan fungsi mysqli\_query() (soal no 3)  INSERT INTO user (id, username, password) VALUES (1, “admin”, 123); |

**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)        Jawab: dari hasil pengamatan saya login tersebut gagal di karenakan username tidak ada di dalam database yang sudah di setting |
| 7 | Lakukan login dengan mengetikkan username dan password yang datanya sudah tersimpan di database. Jelaskan hasil pengamatanmu (soal no 5)        Jawab: dari hasil pengaamatan saya berhasil login karena terdapat nama yang sudah di setting di dalam database |

**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)        Jawab: dari hasil pengamatan saya program tersebut berhasil login karena terdapat username dan password yang sudah di simpan di dalam database |
| 6 | Lakukan login dengan mengetikkan data guest. Jelaskan hasil pengamatanmu (soal no 9)        Jawab: dari hasil pengamatan saya program tersebut berhasil login karena terdapat username dan password yang sudah di simpan di dalam database |
| 7 | Lakukan login dengan mengetikkan username dan password yang tidak tersimpan di database. Jelaskan hasil pengamatanmu (soal no 10)      Jawab: dari pengamatan saya Ketika memasukkan username dan password tersebut tidak bisa login karena username dan password tidak ada di dalam database |

**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)    Jawab: Dari hasil pengamatan saya tidak terdapat cookies karena array masih tidak terdifinisikan atau tidak di buat |
| 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)    Jawab: cookies berhasil di cetak karena sudah didefinisikan atau sudah di buat |
| 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)    Jawab: cookies tersebut sudah di hapus |

**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)    Jawab: terjadi eror Ketika menjalankan program tersebut karena tidak ada yang membeli buku dan novel |
| 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)      Jawab: Dari hasil tersebut dapat di dilihat terdapat pembeli yang membeli novel sebanyak 2 dan buku sebanyak 1 karena sudah ada yang membeli atau di setting |
| 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)      Jawab: Ketika session di set akan muncul warna dan hewan yang sudah di set |

**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)    Jawab: Session tersebut di hapus dengan menggunakan sessionDel |

**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)  **1. Pengguna memasukkan username dan password.**   * Pengguna mengisi data username dan password pada formulir login (LoginForm.html). * Data yang diinputkan dikirimkan ke file sessionLoginProses.php dengan metode POST.   **2. Verifikasi data dan pengaturan sesi.**   * File sessionLoginProses.php menerima data username dan password. * Data tersebut digunakan dalam query SELECT untuk mencari data yang cocok di database. * Jika data ditemukan:   + Fungsi session\_start() dijalankan untuk memulai sesi.   + Username disimpan dalam variabel username dan disimpan dalam SESSION.   + Variabel status diset ke login dan disimpan dalam SESSION.   + Pengguna diarahkan ke homeSession.php dengan link. * Jika data tidak ditemukan:   + Pesan "Gagal Login" ditampilkan.   + Pengguna diarahkan kembali ke LoginForm.html dengan link untuk mencoba lagi.   **3. Tampilan dan logout di homeSession.php.**   * File homeSession.php menggunakan nilai username dari SESSION untuk menampilkan pesan "Selamat datang". * Pengguna dapat logout dengan mengklik link "Log Out" yang mengarahkan ke file sessionLogout.php.   **4. Penghapusan data sesi.**   * File sessionLogout.php menjalankan fungsi session\_start() dan session\_destroy() untuk menghapus data SESSION. |