|  |  |
| --- | --- |
|  | Jurusan Teknologi Informasi Politeknik Negeri Malang  **Jobsheet-5: Multi Level USer**  **Mata Kuliah Pemrograman Web Lanjut**  **Pengampu:** Tim Ajar Pemrograman Web Lanjut  *Februari 2020* |

**Topik**

Membuat sebuah login dengan beberapa level user.

**Tujuan**

Mahasiswa diharapkan dapat:

1. Memahami cara membuat halaman login
2. Memahami cara membuat halaman login dengan beberapa level user yang berbeda

**Praktikum – Bagian 1: Membuat Login dengan Multi Level User**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat tampilan login (views/login/index.php), dan tambahkan code berikut pada views/login/index.php: |
| 2 | Buatlah sebuah ***controller*** dengan nama ***login*** (**controller/login**) |
| 3 | Tambahkan sebuah *code* pada **views**/**template** dengan nama **header**\_**login (views/template/header\_login)** |
| 4 | Ketika dijalankan di browser, hasilnya akan seperti gambar berikut: |
| 5 | Setelah berhasil membuat sebuah form login maka selanjutnya kita akan membuat logic nya. Logic ini digunakan untuk proses validasi data antara inputan user dengan database yang sudah ada. |
| 6 | Sebelum membuat *logic*-nya terlebih dahulu buat tabelnya pada **database Kuliah**\_**CI** (yang pernah dibuat sebelumnya) dengan nama **tabel user** dan tambahkan 2 data, sebagai contoh yaituadmin dan *user* |
| 7 | Setelah kita berhasil membuat sebuah tabel untuk menyimpan *username*, *password* dan level maka selanjutnya kita akan membuat model dengan nama **login\_model (models/login\_model)**    Keterangan :   1. *Line* 6 sebuah function *login* dimana terdapat 2 parameter yaitu $username dan $password 2. *Line* 11 untuk melakukan *select* mana saja yang akan digunakan. Jika kita lihat terdapat 3 parameter yaitu *username*, *password*, dan level -> untuk penulisan ini semua harus sesuai dengan nama *field* yang ada pada *tabel*-nya (**tabel user**). 3. *Line* 12 adalah nama *tabel*-nya (**tabel user**) 4. *Line* 13 adalah selama inputan *user* yang tersimpan pada $username sama dengan *username* pada tabel *user* 5. *Line* 14 adalah selama inputan *user* yang tersimpan pada $password sama dengan *password* pada tabel user 6. *Line* 15 adalah data yang diambil cuma 1 7. *Line* 16 adalah sebuah variabel baru dengan nama $query yang digunakan untuk mengambil semua data 8. *Line* 18 – 24 adalah sebuah pengkondisian dimana jika $query menemukan 1 *row* pada tabel atau dengan kata lain data berhasil ditemukan maka nilai pengembalian querynya akan mencetak isi dari $query($query->result()) dan jika tidak menemukan maka nilai kembaliannya kita set menjadi *false* |
| 8 | Langkah berikutnya tambahkan **form**\_**open** dan **form**\_**close** pada *code* **views/login/index.php**    Keterangan:   1. *Line* 2 artinya adalah *form* yang digunakan untuk membuka halaman proses *login* pada *controller login* 2. *Line* 44 artinya adalah untuk menutup *form* dikarenakan tadi terdapat perintah form\_open |
| 9 | Buatlah sebuah *function* dengan nama **proses**\_**login** pada ***controller login*.**Tapi sebelum kita membuat ***function* proses**\_**login** maka terlebih dahulu kita tambahkan sebuah ***function* construct**    Keterangan : Jika pada config/autoload sudah kalian setting maka kita tidak perlu menambahkan lagi seperti pada line 11. Dikarenakan pada config/autoload/library sudah diset maka pada *function* construct tidak perlu ditambahkan lagi. |
| 10 | Langkah berikutnya menambahkan **function proses\_login** pada **controller login**    Keterangan :   1. *Line* 25 dan 26 adalah sebuah variabel baru dimana isinya adalah inputan dari *user*. *User* akan mengirimkan data yang terkirim dengan metode **POST** yang tersimpan sementara pada **uname1** dan **pwd1** (**uname1** dan **pwd1** adalah **name** pada inputan *username* dan *password* pada page login/index pada *tag input*). 2. *Line* 28 adalah variabel baru(**$cek\_login**) isinya adalah sebuah model(**login\_model**) dimana model(**login\_model**) tersebut akan menjalan sebuah *function*(**function login**) dengan nama login(yang telah kita buat sebelumnya) dan mempunyai 2 paramater yaitu $username dan $password dimana isinya adalah isi dari *line* 25 dan 26, yang nanti akan dikirimkan ke *function* login pada login\_model. 3. *Line* 30-51 adalah sebuah pengkondisian jika $cek\_login bernilai benar maka cek\_login akan dipecah menjadi beberapa *row* ($cek\_login as $row). setelah itu kita set\_userdata menggunakan *session* dengan nama variabel barunya adalah *user* sedangkan isinya adalah *username* (*line* 34) dan kita juga set\_userdata menggunakan session dengan nama variabelnya level (*line* 35) sedangkan isinya adalah level. Setelah kita pecah menjadi beberapa row dan mempunyai isi yang berbeda-beda maka langkah berikutnya kita buat pengkondisian jika *session* userdata(‘level’) berisi admin maka akan me-*redirect* pada halaman mahasiswa/index(line 38-41). Apabila levelnya berupa user maka akan meredirect ke halaman *user* (*line* 42-45). Jika $cek\_login bernilai salah maka akan me-*redirect* ke halaman *login* kembali (*line* 47 - 51). |
| 11 | Pada *line* 25 dan 26 tambahkan **htmlspesialchar** agar *user* tidak dapat memasukkan *char* yang special. |
| 12 | Agar halaman *login* yang pertama kali di-*load* maka kita harus ubah konfigurasinya pada **application/config/routes** |
| 13 | Hasilnya jika kita masuk menggunakan admin |
| 14 | Hasilnya jika kita masuk menggunakan *user* (jika kita lihat maka hasilnya *page not found* hal ini dikarenakan kita belum membuat halaman tersebut) |
| 15 | Hasilnya jika kita salah memasukkan *password* dan *username* (jika salah maka ***redirect*** ke halaman login tanpa ada pemberitahuan jika *username* dan *password* salah) |
| 16 | Kita tambahkan ***alert*** agar pada saat ***user*** salah memasukkan ***username*** dan ***password***, maka akan muncul pemberitahuan bahwa ***username*** dan ***password*** anda salah. Tambahkan c*ode* berikut pada **login/index**    dan pada bagian **controller/login/proses\_login** kita ubah pada *else*-nya    Sehingga hasilnya akan tampak seperti berikut |
| 17 | Jika kita lihat pada percobaan sebelumnya, pada saat kita *login* dengan menggunakan level *user* maka halaman tidak akan tampil atau *page not found*. Untuk kita akan membuat halaman *user* sederhana |
| 18 | Jika dilihat pada ***function* proses**\_**login** terdapat *code* sebagai berikut:    **Untuk lebih mudahnya pada line 44 ganti *redirect* (‘mahasiswa/index’) menjadi redirect(‘user’);** |
| 19 | Setelah mengganti ***redirect,*** maka yang harus dilakukan adalah membuat sebuah ***view*** dengan nama ***user*** pada ***folder* mahasiswa**. sebelum kita membuat ***view*** dengan nama **user** kita terlebih dahulu membuat sebuah ***controller*** dengan nama **user**. Buat *controller* user dengan *code* sebagai berikut: |
| 20 | Membuat tampilan pada *view* dengan nama *user*    Sehingga hasilnya jika kita *login* pada level *user* maka tampilan akan seperti gambar berikut |
| 21 | Setelah berhasil membuat halaman *user,* maka langkah berikutnya kita membuat menu *logout*. Tambahkan *code* berikut pada **views/template/header line 36**    *line* 33 diubah menjadi seperti pada gambar |
| 22 | Tambahkan ***function logout* pada *controller login*** *code*-nya sebagai berikut |
| 23 | Hasilnya jika kita klik *logout* maka akan muncul halaman *login.* Pada halaman *user* kita juga tambahkan c*ode* untuk proses *logout* sederhana |
| 24 | Jika kita lakukan perubahan langsung pada url  <http://localhost/12.codeIgniter/code-keempat/login/> menjadi <http://localhost/12.codeIgniter/codekeempat/mahasiswa/> seperti pada gambar dibawah ini:      maka kita akan dapat masuk secara langsung tanpa perlu *login* terlebih dahulu. Untuk itu kita harus menambahkan beberapa *code* pada *controller*/mahasiswa seperti pada gambar dibawah ini (*line* 15-19)    Keterangan : *Line* 15-17 adalah pengkondisian dimana jika levelnya bukan admin maka halaman akan me-*redirect* ke halaman *login* |
| 25 | Lakukan hal yang sama pada ***user*** (***controller* user**) |
| 26 | Lakukan hal yang sama pada ***home*** (***controller* home**) |
| 27 | Tambahkan *code* berikut agar pada halaman admin tampil level *user*nya (**views/home/index**) |
| 28 | ***Comment* beberapa *code* pada *controller home* (*line* 12 dan hapus parameter name pada index)**    Sehingga hasilnya adalah sebagai berikut: |
| 29 | Lakukan hal yang sama pada *user* |

***-- Selamat Mengerjakan –***