Tugas Pertemuan 6 Pemrograman Web Lanjut



Muhammad Farrel Caesarian TI-2D/16 2341720053 POLITEKNIK NEGERI MALANG

1. | Praktikum 1. Modal Ajax Tambah Data (Data User)

Kita buat form tambah data baru dengan menerapkan modal dan proses ajax

Pertama yang kita siapkan adalah menambahkan *library jQuery Validation* dan *Sweetalert* ke aplikasi web kita. Caranya kita tambahkan link kedua *library* tersebut ketemplate.blade.php, library sudah disediakan oleh adminLTE

Selanjutnya Kita modifikasi view *user/index.blade.php*, kita tambahkan tombol untuk membuat form popup ajax

Selanjutnya kita tambahkan kode berikut pada *akhir* @section('content') pada view user/index.blade.php

```
(d)/ 10- BYMOGB1 CLBSS= BOOB1 fade animate shake tabindex= 1 mole-dialog databackdrop= static data-keyboard= false data-kldtn= /5% aria-nidden= true ></di
```

Kemudian kita tambahkan kode berikut pada *awal @push('js')* pada view*user/index.blade.php*

Selanjutkan Kita modifikasi route/web.php untuk mengakomodir operasi ajax

```
Route::get('/create_ajax', [UserController::class, 'create_ajax']); // menampilkan halaman form tambah user Ajax Route::post('/ajax', [UserController::class, 'store_ajax']); // menyimpan data user baru Ajax
```

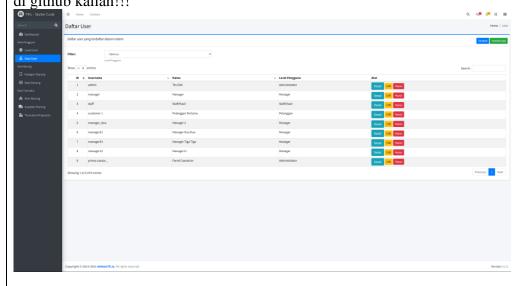
Kemudian Kita tambahkan fungsi create_ajax() pada file UserController.php

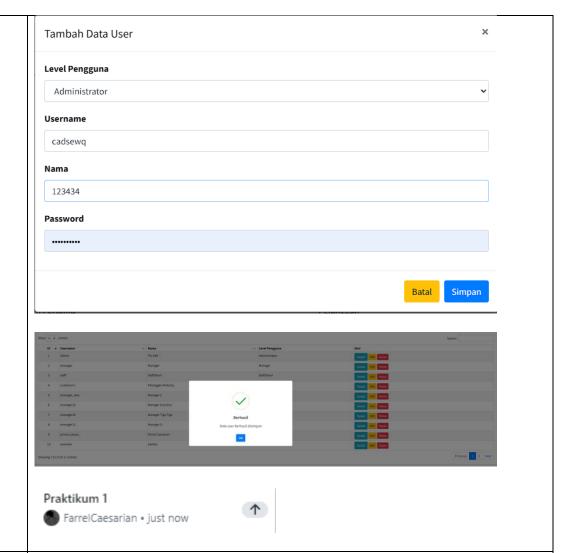
Setelah itu, kita buat **view baru** user/create_ajax.blade.php untuk menampilkan form dengan ajax

```
resources > views > user > reate_ajax.blade.php > resources > views > user > reate_ajax.blade.php > reate_ajax.bla
```

Kemudian untuk mengakomodir proses simpan data melalui ajax, kita buat fungsi store_ajax() pada UserController.php

OK, sekarang kita coba melakukan proses tambah data user menggunakan form ajax. Amati apa yang terjadi dan laporankan pada laporan *jobsheet* dan *commit* di github kalian!!!





2. | Praktikum 2. Modal Ajax Edit Data (Data User)

Pada Praktikum 2 ini, kita akan melakukan koding untuk proses edit menggunakan ajax

Pertama-tama, kita **ubah** dulu fungsi list() pada UserController.php untuk mengganti **tombol edit** untuk bisa menggunakan modal

```
public function list(Request $request)

$users = UserModel::select('user_id', 'username', 'nama', 'level_id')
    ->with('level');

// Filter data user berdasarkan level_id
    if ($request->level_id) {
        $users->where('level_id', $request->level_id);
}

return DataTables::of($users)

// menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
    ->addIndexColumn()
    ->addColumn('aksi', function ($user) { // menambahkan kolom aksi

// $btn = 'ca href=" . url('/user/' . $user->user_id) . " class="btn btn-info

// $btn = 'ca href=" . url('/user/' . $user->user_id . '/edit') . " class="btn

// $btn = 'cfonm class="d-inline-block" method="PoSt" action=" . url('/user/'

// . csrf_field() . method_field('DELETE')

// 'cbutton type="submit" class="btn btn-danger btn-sm" onclick="return cont

// 38 6

$btn = 'cbutton onclick="modalAction(\' . url('/user/' . $user->user_id .

| '/show_ajax') . \'\' class="btn btn-info btn-sm">betail</button> ';

$btn = 'cbutton onclick="modalAction(\' . url('/user/' . $user->user_id .

| '/edit_ajax') . \'\' class="btn btn-warning btn-sm">Edit/button> ';

$btn = 'cbutton onclick="modalAction(\' . url('/user/' . $user->user_id .

| '/delte_ajax') . \'\' class="btn btn-warning btn-sm">Edit/button> ';

return $btn;

})

-rawColumns(['aksi']) // Memberitahu bahwa kolom aksi berisi HTML
->make(true);
```

Selanjutnya kita modifikasi routes/web.php untuk mengakomodir request edit menggunakan ajax

```
Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); // menampilkan halaman form edit user Ajax
Route::put('/{id}update_ajax', [UserController::class, 'update_ajax']); // menyimpan perubahan data user Ajax
```

Kemudian, kita buat fungsi edit_ajax() pada UserController.php

```
public function edit_ajax(string $id){
    $user = UserModel::find($id);
    $level = LevelModel::select('level_id','level_nama')->get();

    return view('user.edit_ajax',['user' => $user, 'level' => $level]);
}
```

Kita buat **view baru** pada user/edit_ajax.blade.php untuk menampilkan form view ajax

Selanjutnya, kita buat fungsi update_ajax() pada UserController.php untuk mengakomodir request update data user melalui ajax

Sekarang kita coba bagian edit user, amati proses nya. Jangan lupa laporkan dan commit ke repository git kalian! Edit Data User × Level Pengguna Manager Username tes edit tes edit Password Abaikan jika tidak ingin ubah password Simpan Praktikum 3. Modal Ajax Hapus Data (Data User) Pada Praktikum 3 ini, kita akan melakukan koding untuk proses hapus menggunakan ajax. Pertama-tama, kita ubah dulu routes/web.php untuk mengakomodir request halaman konfirmasi untuk menghapus data Route::get('/[id]/delete_ajax', [UserController::class, 'confirm_ajax']); // menampilkan halaman form D
Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); // menghapus data user Ajax Kemudian kita buat fungsi confirm_ajax() pada UserController.php public function confirm_ajax(string \$id){ \$user = UserModel::find(\$id); return view('user.confirm_ajax', ['user' => \$user]);

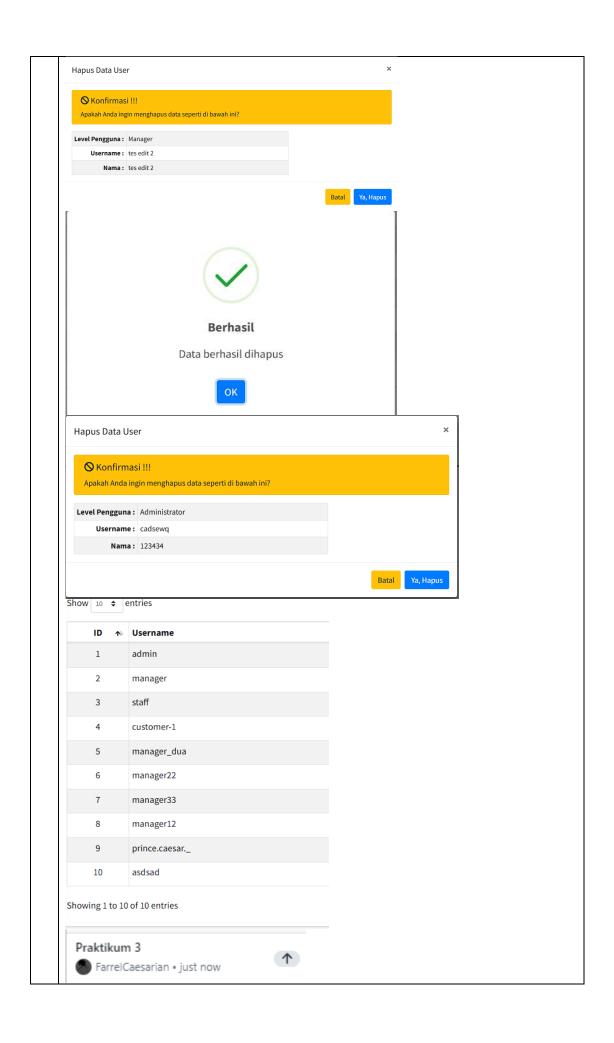
Selanjutnya kita view untuk konfirmasi hapus data dengan nama

user/confirm_ajax.blade.php

Kemudian kita buat fungsi delete_ajax() pada UserController.php untuk mengakomodir *request* hapus data user

Setelah semua selesai, mari kita coba untuk melakukan percobaan dari koding yang telah kita lakukan.

ID ↑	Username ↑	Nama	
1	admin	Tes Edit	
2	manager	Manager	
3	staff	Staff/Kasir	
4	customer-1	Pelanggan Pertama	
5	manager_dua	Manager 2	
6	manager22	Manager Dua Dua	
7	manager33	Manager Tiga Tiga	
8	manager12	Manager11	
9	prince.caesar	Farrel Caesarian	
10	tes edit 2	tes edit 2	



4. Tugas Jobsheet

Implementasikan koding untuk Ajax Form dan Client Validation dengan jQuery Validation pada pada menu berikut ini

✓ Tabel m_level

