

LAPORAN PRAKTIKUM JOBSHEET 6

**MATA KULIAH PEMROGRAMAN WEB LANJUT
AJAX FORM (AdminLTE) dan CLIENT VALIDATION**

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Di Susun Oleh :

Louise Nazarossa (2341760117)

SIB-2A/18

**PROGRAM STUDI D4 SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025**

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 ke template.blade.php, library sudah disediakan oleh adminLTE.

```
18 <!--SweetAlert2-->
19 <link rel="stylesheet" href="{{ asset('adminlte/plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
80 <!--jquery-validation-->
81 <script src="{{ asset('adminlte/plugins/jquery-validation/jquery.validate.min.js') }}"></script>
82 <script src="{{ asset('adminlte/plugins/jquery-validation/additional-methods.min.js') }}"></script>
83 <!--SweetAlert2-->
84 <script src="{{ asset('adminlte/plugins/sweetalert2/sweetalert2.min.js') }}"></script>
```

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

```
6 <h3 class="card-title">{{ $page->title }}</h3>
7 <div class="card-tools">
8 <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}">Tambah</a>
9 <button onclick="modalAction('{{ url('user/create_ajax') }}')" class="btn btn-sm btn-success mt-1">Tambah Ajax</button>
10 </div>
```

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

```
41 @endsection
42 <div id="myModal" class="modal fade animate shake" tabindex="-1" role="dialog" data-backdrop="static"
43 data-keyboard="false" data-width="75%" aria-hidden="true"></div>
```

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

```
47 @push('js')
48 <script>
49     function modalAction(url = ''){
50         $('#myModal').load(url,function(){
51             $('#myModal').modal('show');
52         });
53     }
54
55     $(document).ready(function() {
56         var dataUser = $('#table_user').DataTable({
```

- Selanjutnya Kita modifikasi route/web.php untuk mengakomodir operasi ajax

```
22 Route::get('/create_ajax', [UserController::class, 'create_ajax']);
23 Route::post('/ajax', [UserController::class, 'store_ajax']);
```

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

```
325 public function create_ajax()
326 {
327     $level = LevelModel::select('level_id', 'level_nama')->get();
328
329     return view('user.create_ajax')
330         ->with('level', $level);
331 }
```

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

```
1 <form action="{{ url('/user/ajax') }}" method="POST" id="form-tambah">
2     @csrf
3     <div id="modal-master" class="modal-dialog modal-lg" role="document">
4         <div class="modal-content">
5             <div class="modal-header">
6                 <h5 class="modal-title" id="exampleModalLabel">Tambah Data User</h5>
7                 <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
8                 aria-hidden="true">&times;</span></button>
9             </div>
10            <div class="modal-body">
11                <div class="form-group">
12                    <label>Level Pengguna</label>
13                    <select name="level_id" id="level_id" class="form-control" required>
14                        <option value="">- Pilih Level -</option>
15                        @foreach($level as $l)
16                            <option value="{{ $l->level_id }}">{{ $l->level_nama }}</option>
17                        @endforeach
18                    </select>
19                    <small id="error-level_id" class="error-text form-text text-danger"></small>
20                </div>
21                <div class="form-group">
22                    <label>Username</label>
23                    <input value="" type="text" name="username" id="username" class="form-control"
24                    required>
25                    <small id="error-username" class="error-text form-text text-danger"></small>
26                </div>
27                <div class="form-group">
28                    <label>Nama</label>
29                    <input value="" type="text" name="nama" id="nama" class="form-control"
30                    required>
31                    <small id="error-nama" class="error-text form-text text-danger"></small>
32                </div>
33                <div class="form-group">
34                    <label>Password</label>
35                    <input value="" type="password" name="password" id="password" class="form
36                    control" required>
37                    <small id="error-password" class="error-text form-text text-danger"></small>
38                </div>
39            </div>
40            <div class="modal-footer">
41                <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
42                <button type="submit" class="btn btn-primary">Simpan</button>
43            </div>
44        </div>
45    </div>
46 </form>
47 <script>
```

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

```

332     public function store_ajax(Request $request)
333     {
334         if ($request->ajax() || $request->wantsJson()) {
335
336             $rules = [
337                 'level_id' => 'required|integer',
338                 'username' => 'required|string|min:3|unique:m_user,username',
339                 'nama'     => 'required|string|max:100',
340                 'password' => 'required|min:6'
341             ];
342
343             $validator = Validator::make($request->all(), $rules);
344
345             if ($validator->fails()) {
346                 return response()->json([
347                     'status' => false,
348                     'message' => 'Validasi Gagal',
349                     'msgField' => $validator->errors(),
350                 ]);
351             }
352
353             UserModel::create([
354                 'username' => $request->username,
355                 'nama'     => $request->nama,
356                 'password' => bcrypt($request->password),
357                 'level_id' => $request->level_id
358             ]);
359
360             return response()->json([
361                 'status' => true,
362                 'message' => 'Data user berhasil disimpan'
363             ]);
364         }
365         redirect('/');
366     }

```

PRAKTIKUM 2: Modal Ajax Edit Data (Data User)

- a.) Pada Praktikum 2 ini, kita akan melakukan koding untuk proses edit menggunakan ajax.
- b.) Pertama-tama, kita ubah dulu fungsi list() pada UserController.php untuk mengganti tombol edit untuk bisa menggunakan modal

```

153 public function list(Request $request)
154 {
155     $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
156         ->with('level');
157
158     // Filter data user berdasarkan level_id
159     if ($request->level_id) {
160         $users->where('level_id', $request->level_id);
161     }
162     return DataTables::of($users)
163         // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
164         ->addIndexColumn()
165         ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
166             $btn = '<button onclick="modalAction(\''.url('/user/' . $user->user_id .
167                 '/show_ajax').'\')"' class="btn btn-info btn-sm">Detail</button> ';
168             $btn .= '<button onclick="modalAction(\''.url('/user/' . $user->user_id .
169                 '/edit_ajax').'\')"' class="btn btn-warning btn-sm">Edit</button> ';
170             $btn .= '<button onclick="modalAction(\''.url('/user/' . $user->user_id .
171                 '/delete_ajax').'\')"' class="btn btn-danger btn-sm">Hapus</button> ';

```

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

```

27 Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); // m
28 Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']);

```

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

```

367 public function edit_ajax(string $id)
368 {
369     $user = UserModel::find($id);
370     $level = LevelModel::select('level_id', 'level_nama')->get();
371
372     return view('user.edit_ajax', ['user' => $user, 'level' => $level]);
373 }

```

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

```

resources > views > user > edit_ajax.blade.php > ...
1 @empty($user)
2     <div id="modal-master" class="modal-dialog modal-lg" role="document">
3         <div class="modal-content">
4             <div class="modal-header">
5                 <h5 class="modal-title" id="exampleModalLabel">Kesalahan</h5>
6                 <button type="button" class="close" data-dismiss="modal" aria
7 label="Close"><span aria-hidden="true">&times;</span></button>
8             </div>
9             <div class="modal-body">
10                <div class="alert alert-danger">
11                    <h5><i class="icon fas fa-ban"></i> Kesalahan!!!</h5>
12                    Data yang anda cari tidak ditemukan</div>
13                    <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
14                </div>
15            </div>
16        </div>
17    @else
18        <form action="{{ url('/user/' . $user->user_id . '/update_ajax') }}" method="POST" id="form
19 edit">
20        @csrf
21        @method('PUT')
22        <div id="modal-master" class="modal-dialog modal-lg" role="document">
23            <div class="modal-content">
24                <div class="modal-header">
25                    <h5 class="modal-title" id="exampleModalLabel">Edit Data User</h5>
26                    <button type="button" class="close" data-dismiss="modal" aria
27 label="Close"><span aria-hidden="true">&times;</span></button>
28            </div>

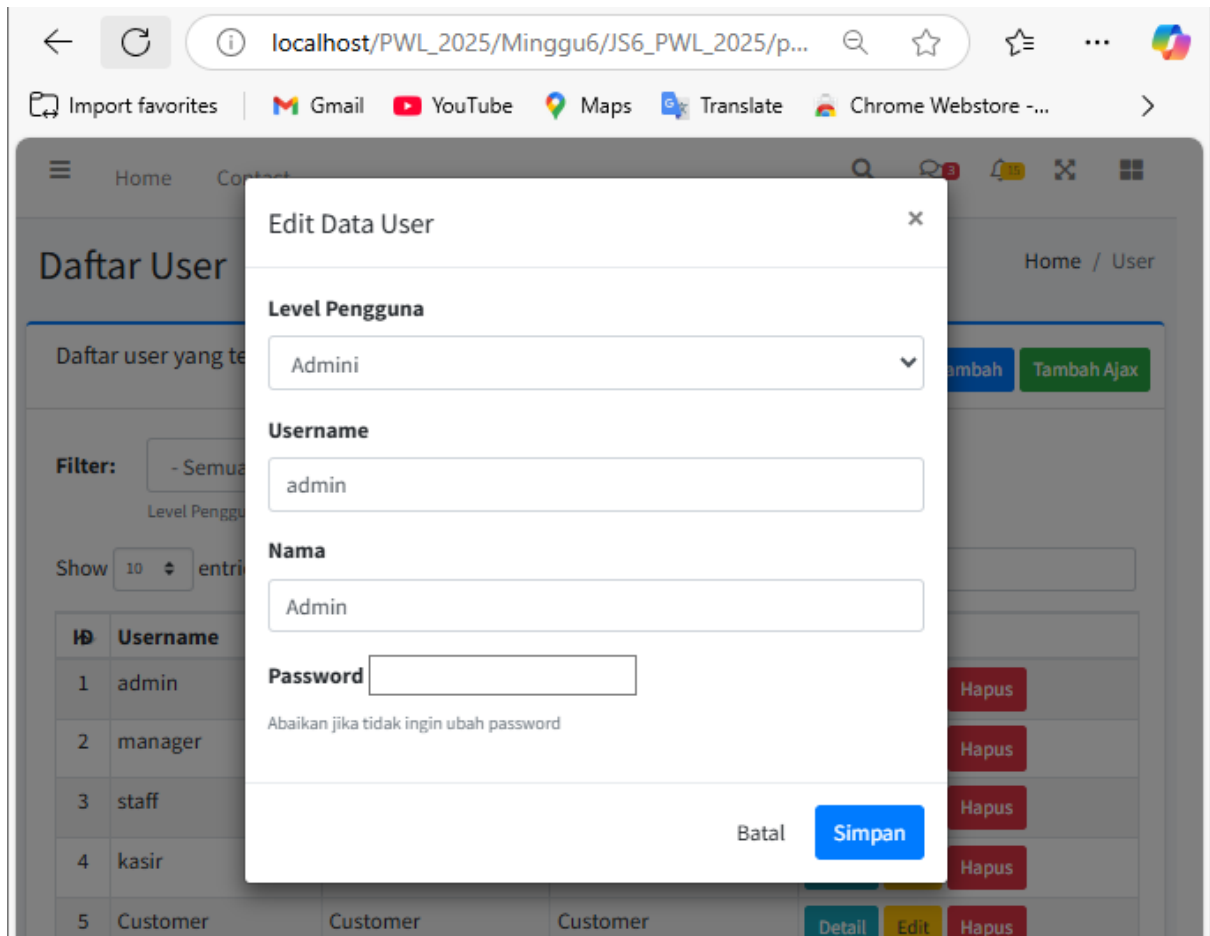
```

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

```
374     public function update_ajax(Request $request, $id){
375         // cek apakah request dari ajax
376         if ($request->ajax() || $request->wantsJson()) {
377             $rules = [
378                 'level_id' => 'required|integer',
379                 'username' => 'required|max:20|unique:m_user,username, '.$id.',user_id',
380                 'nama'     => 'required|max:100',
381                 'password' => 'nullable|min:6|max:20'
382             ];
383
384             // use Illuminate\Support\Facades\Validator;
385             $validator = Validator::make($request->all(), $rules);
386
387             if ($validator->fails()) {
388                 return response()->json([
389                     'status' => false,    // respon json, true: berhasil, false: gagal
390                     'message' => 'Validasi gagal.',
391                     'msgField' => $validator->errors() // menunjukkan field mana yang error
392                 ]);
393             }
394         }
```

```
395         $check = UserModel::find($id);
396         if ($check) {
397             if (!$request->filled('password')) { // jika pas
398                 $request->request->remove('password');
399             }
400
401             $check->update($request->all());
402             return response()->json([
403                 'status' => true,
404                 'message' => 'Data berhasil diupdate'
405             ]);
406         } else{
407             return response()->json([
408                 'status' => false,
409                 'message' => 'Data tidak ditemukan'
410             ]);
411         }
412     }
413     return redirect('/');
414 }
```

- g.) Sekarang kita coba bagian edit user, amati proses nya. Jangan lupa laporkan dan commit ke repository git kalian !



PRAKTIKUM 3: Modal Ajax Hapus Data (Data User)

- a.) Pada Praktikum 3 ini, kita akan melakukan koding untuk proses hapus menggunakan ajax.
- b.) Pertama-tama, kita ubah dulu routes/web.php untuk mengakomodir request halaman konfirmasi untuk menghapus data

```
29 Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); //
30 Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']);
```

- c.) Kemudian kita buat fungsi confirm_ajax() pada UserController.php

```
415 public function confirm_ajax(string $id)
416 {
417     $user = UserModel::find($id);
418
419     return view('user.confirm_ajax', ['user' => $user]);
420 }
```

d.) Selanjutnya kita view untuk konfirmasi hapus data dengan nama user/confirm_ajax.blade.php

```
resources > views > user > confirm_ajax.blade.php > ...
1  @empty($user)
2      <div id="modal-master" class="modal-dialog modal-lg" role="document">
3          <div class="modal-content">
4              <div class="modal-header">
5                  <h5 class="modal-title" id="exampleModalLabel">Kesalahan</h5>
6                  <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
7              </div>
8              <div class="modal-body">
9                  <div class="alert alert-danger">
10                     <h5><i class="icon fas fa-ban"></i> Kesalahan!!!</h5>
11                     Data yang anda cari tidak ditemukan</div>
12                     <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
13                 </div>
14             </div>
15         </div>
16     @else
17         <form action="{{ url('/user/' . $user->user_id . '/delete_ajax') }}" method="POST" id="form-delete">
18             @csrf
19             @method('DELETE')
20             <div id="modal-master" class="modal-dialog modal-lg" role="document">
21                 <div class="modal-content">
22                     <div class="modal-header">
23                         <h5 class="modal-title" id="exampleModalLabel">Hapus Data User</h5>
24                         <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
25                     </div>
26                     <div class="modal-body">
27                         <div class="alert alert-warning">
28                             <h5><i class="icon fas fa-ban"></i> Konfirmasi !!!</h5>
29                             Apakah Anda ingin menghapus data seperti di bawah ini?
30                         </div>
31                         <table class="table table-sm table-bordered table-striped">
32                             <tr><th class="text-right col-3">Level Pengguna :</th><td class="col-9">{{ $user->level->level_nama }}</td></tr>
33                             <tr><th class="text-right col-3">Username :</th><td class="col-9">{{ $user->username }}</td></tr>
34                             <tr><th class="text-right col-3">Nama :</th><td class="col-9">{{ $user->nama }}</td></tr>
35                         </table>
36                     </div>
37                     <div class="modal-footer">
38                         <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
39                         <button type="submit" class="btn btn-primary">Ya, Hapus</button>
40                     </div>
41                 </div>
42             </div>
43         </form>
44     @endif
45 </div>
```



```

43     </form>
44     <script>
45         $(document).ready(function() {
46             $("#form-delete").validate({
47                 rules: {},
48                 submitHandler: function(form) {
49                     $.ajax({
50                         url: form.action,
51                         type: form.method,
52                         data: $(form).serialize(),
53                         success: function(response) {
54                             if(response.status){
55                                 $('#myModal').modal('hide');
56                                 Swal.fire({
57                                     icon: 'success',
58                                     title: 'Berhasil',
59                                     text: response.message
60                                 });
61                                 dataUser.ajax.reload();
62                             }else{
63                                 $('.error-text').text('');
64                                 $.each(response.msgField, function(prefix, val) {
65                                     $('#error-'+prefix).text(val[0]);
66                                 });
67                                 Swal.fire({
68                                     icon: 'error',
69                                     title: 'Terjadi Kesalahan',

```

```

71                                     });
72                                 }
73                             }
74                         });
75                         return false;
76                     },
77                     errorElement: 'span',
78                     errorPlacement: function (error, element) {
79                         error.addClass('invalid-feedback');
80                         element.closest('.form-group').append(error);
81                     },
82                     highlight: function (element, errorClass, validClass) {
83                         $(element).addClass('is-invalid');
84                     },
85                     unhighlight: function (element, errorClass, validClass) {
86                         $(element).removeClass('is-invalid');
87                     }
88                 });
89             });
90         </script>
91         @endempty

```

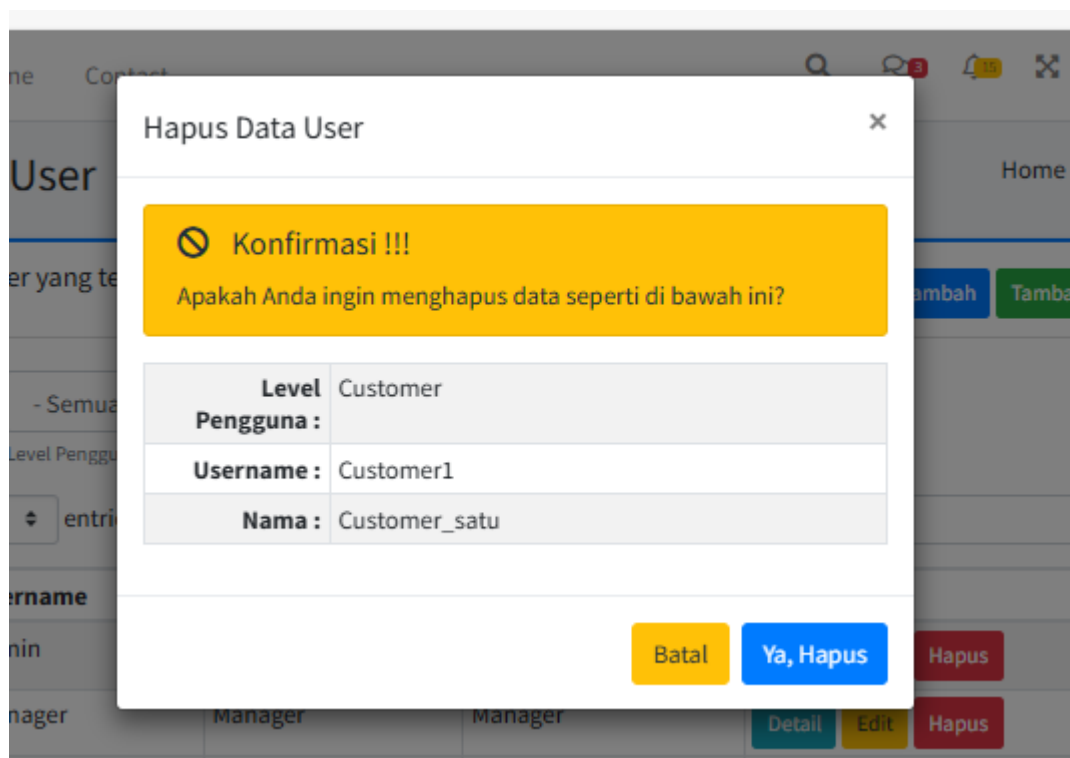
- e.) Kemudian kita buat fungsi delete_ajax() pada UserController.php untuk mengakomodir request hapus data user

```

421 public function delete_ajax(Request $request, $id)
422 {
423     // cek apakah request dari ajax
424     if ($request->ajax() || $request->wantsJson()) {
425         $user = UserModel::find($id);
426         if ($user) {
427             try {
428                 $user->delete();
429                 return response()->json([
430                     'status' => true,
431                     'message' => 'Data berhasil dihapus'
432                 ]);
433             } catch (\Illuminate\Database\QueryException ) {
434                 return response()->json([
435                     'status' => false,
436                     'message' => 'Data gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini'
437                 ]);
438             }
439         } else {
440             return response()->json([
441                 'status' => false,
442                 'message' => 'Data tidak ditemukan'
443             ]);
444         }
445     }
446     return redirect('/');

```

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



TUGAS

