

LAPORAN JOBSHEET 6

AJAX FORM (AdminLTE) dan CLIENT VALIDATION

MATA KULIAH PEMROGRAMAN WEB LANJUT

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



Disusun oleh :

Dahniar Davina SIB-2A / 2341760023

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

POLITEKNIK NEGERI MALANG

2025

A. PRAKTIKUM 1 : Modal AJAX Tambah Data User

1. Kita buat form tambah data baru dengan menerapkan modal dan proses ajax.
2. 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.

```
<!-- SweetAlert2 -->
<link rel="stylesheet" href="{{ asset('plugins/sweetalert2-theme-bootstrap-4/sweetalert2-theme-bootstrap-4.css') }}" />
<!-- Theme style -->

81 <!-- jquery-validation -->
82 <script src="{{ asset('plugins/jquery-validation/jquery.validate.min.js') }}"></script>
83 <script src="{{ asset('plugins/jquery-validation/additional-methods.min.js') }}"></script>
84
85 <!-- SweetAlert2 -->
86 <script src="{{ asset('plugins/sweetalert2/sweetalert2.min.js') }}"></script>
```

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

```
<a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}">
<button onclick="modalAction('{{ url('user/create_ajax') }}')" class="btn btn-sm btn-primary mt-1">
```

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

```
47 <div id="myModal" class="modal fade animate shake" tabindex="-1" role="dialog">
```

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

```
function modalAction(url = '') {
    $('#myModal').load(url, function() {
        $('#myModal').modal('show');
    });

    var dataUser;
    $(document).ready(function() {
        dataUser = $('#table_user').DataTable({
```

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

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

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

```
public function create_ajax()
{
    $level = LevelModel::select('level_id', 'level_nama')->get();
    return view('user.create_ajax')
        ->with('level', $level);
}
```

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

```
resources > views > user > create_ajax.blade.php > script
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
7 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
8 <span aria-hidden="true">&times;</span></button>
9 </div>
10 </div>
11 <div class="modal-body">
12 <div class="form-group">
13 <label>Pengguna</label>
14 <select name="level_id" id="level_id" class="form-control">
15 <option value="">Pilih Level</option>
16 @foreach ($level as $l)
17 <option value="{{ $l->level_id }}">{{ $l->level_name }}</option>
18 @endforeach
19 </select>
20 <input type="text" name="username" class="form-control">
21 <input type="password" name="password" class="form-control">
22 <input type="button" value="Simpan" class="btn btn-primary">
23 </div>
24 </div>
25 </div>
26 </form>
```

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

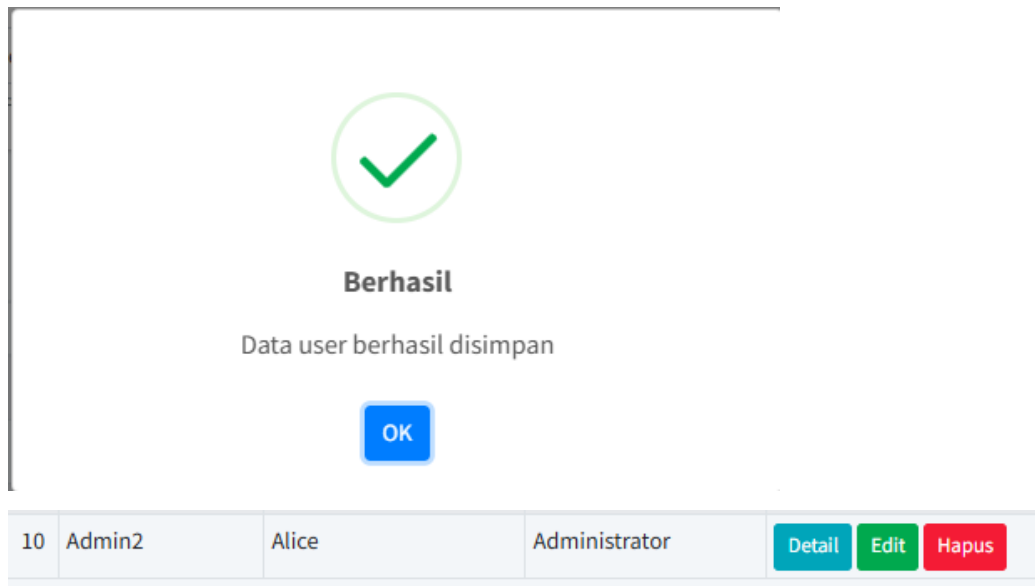
```
public function store_ajax(Request $request) {
    // cek apakah request berupa ajax
    if($request->ajax() || $request->wantsJson()) {
        $rules = [
            'level_id' => 'required|integer',
            'username' => 'required|string|min:3|unique:m_user,username',
            'nama' => 'required|string|max:100',
            'password' => 'required|min:6'
        ];
        // use Illuminate\Support\Facades\Validator;
        $validator = Validator::make($request->all(), $rules);
        if($validator->fails()) {
            return response()->json([
                'status' => false, // response status, false: error/gagal
                'message' => 'Validasi gagal',
                'msgField' => $validator->errors() // pesan error validasi
            ]);
        }
        UserModel::create($request->all());
        return response()->json([
            'status' => true,
            'message' => 'Data user berhasil disimpan'
        ]);
    }
    redirect('/');
}
```

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

Daftar User Home / User

Daftar user yang terdaftar dalam sistem

Filter: -- Semua -- Tambah Tambah Ajax



B. PRAKTIKUM 2 : Modal AJAX Edit Data User

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

```

233 // Ambil data user dalam bentuk JSON untuk DataTables
234 public function list(Request $request)
235 {
236     $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
237     ->with('level');
238
239     // Filter data user berdasarkan level_id
240     if ($request->level_id) {
241         $users->where('level_id', $request->level_id);
242     }
243
244     return DataTables::of($users)
245         ->addIndexColumn() // Menambahkan kolom index / no urut (default nama k
246         ->addColumn('aksi', function ($user) { // Menambahkan kolom aksi
247             $btn = '<button onclick="modalAction(\''.url('/user/' . $user->user
248                 class="btn btn-info btn-sm">Detail</button> ';
249
250             $btn .= '<button onclick="modalAction(\''.url('/user/' . $user->us
251                 class="btn btn-warning btn-sm">Edit</button> ';
252
253             $btn .= '<button onclick="modalAction(\''.url('/user/' . $user->us
254                 class="btn btn-danger btn-sm">Hapus</button> ';
255
256             return $btn;
257         });

```

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

```

Route::get('/{id}/edit_ajax', [UserController::class, 'e
Route::put('/{id}/update_ajax', [UserController::class,

```

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

```
// Menampilkan halaman form edit user ajax
public function edit_ajax(string $id)
{
    $user = UserModel::find($id);
    $level = LevelModel::select('level_id', 'level_nama'

    return view('user.edit_ajax', ['user' => $user, 'lev
}
```

5. 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" re
3         <div class="modal-content">
4             <div class="modal-header">
5                 <h5 class="modal-title" id="exampleModallat
6                 <button type="button" class="close" data-di
7                     <span aria-hidden="true">&times;</span>
8                 </button>
9             </div>
10            <div class="modal-body">
11                <div class="alert alert-danger">
12                    <h5><i class="icon fas fa-ban"></i> Kes
13                    Data yang anda cari tidak ditemukan
14                </div>
15                <a href="{{ url('/user') }}" class="btn btr
16            </div>
17        </div>
18    </div>
19 @else
20    <form action="{{ url('/user/' . $user->user_id . '/upda
```

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

```
public function update_ajax(Request $request, $id)
{
    // Cek apakah request berasal dari AJAX
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'level_id' => 'required|integer',
            'username' => 'required|max:20|unique:m_user,us
            'nama' => 'required|max:100',
            'password' => 'nullable|min:6|max:20'
        ];

        // use Illuminate\Support\Facades\Validator;
        $validator = Validator::make($request->all(), $rules

        if ($validator->fails()) {
            return response()->json([
                'status' => false, // respon json, true:
                'message' => 'Validasi gagal.',
                'msgField' => $validator->errors() // menunj
            ]);
        }

        $check = UserModel::find($id);
        if ($check) {
```

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

Berhasil

OK

C. PRAKTIKUM 3 : Modal AJAX Hapus Data User

```
Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']);
Route::get('/{id}/delete_ajax', [UserController::class, 'delete_ajax']);
Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']);
```

3. Kemudian kita buat fungsi `confirm_ajax()` pada `UserController.php`


4. Selanjutnya kita view untuk konfirmasi hapus data dengan nama `user/confirm_ajax.blade.php`

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

```
public function delete_ajax(Request $request, $id)
{
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantsJson()) {
        $user = UserModel::find($id);
        if ($user) {
            $user->delete();
            return response()->json([
                'status' => true,
                'message' => 'Data berhasil dihapus'
            ]);
        } else {
            return response()->json([
                'status' => false,
                'message' => 'Data tidak ditemukan'
            ]);
        }
    }
    return redirect('/');
}
```

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

Hapus Data User ×

 **Konfirmasi !!!**

Apakah Anda ingin menghapus data seperti di bawah ini?

Level	Administrator
Pengguna :	
Username :	Admin2
Nama :	Alice Noorine

Batal

Ya, Hapus

7. Jangan lupa laporkan ke laporan jobsheet dan lakukan commit pada repository git kalian.!!!

D. TUGAS

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

✓ Tabel m_level

Daftar Level Home / Level

Daftar level yang terdaftar dalam sistem Tambah Tambah Ajax

Filter:

- Semua -

Level Pengguna

Show

10

 entries

Search:

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	MNG	Manager	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	STF	Staff/Kasir	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
4	CUS	Pelanggan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>



Berhasil

Data berhasil diupdate

OK

5	MBR	Member	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
---	-----	--------	--

Edit Data Level ×

Level Kode

MBR

Level Nama

Member

Batal

Simpan

Hapus Data level ×

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

level Kode : MBR

level Nama : Member

Batal

Ya, Hapus

✓ Tabel m_kategori

Daftar KategoriHome / Kategori

Daftar kategori yang terdaftar dalam sistem

Tambah

Tambah Ajax

Show 10 entries

Search:

ID	Kode Kategori	Nama Kategori	Aksi
1	KTG001	Elektronik	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	KTG002	Pakaian	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	KTG003	Makanan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
4	KTG004	Minuman	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
5	KTG005	Obat-obatan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
6	KTG006	Camilan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
7	KTG007	Minuman Kaleng	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
8	KTG008	Sepatu	<div>Detail</div> <div>Edit</div> <div>Hapus</div>



Berhasil

Data kategori berhasil disimpan

OK

9	KTG009	Kosmetik	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
---	--------	----------	--

Edit Data Kategori

Kategori Kode

KTG009

Kategori Nama

Kosmetik

Batal

Simpan

Hapus Data Kategori

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Kategori Kode

:

KTG009

Kategori Nama

:

Kosmetik

Batal

Ya, Hapus

✓ Tabel m_supplier

Tambah Data supplier

×

Kode supplier

SP4

Supplier Nama

PT Paragon Innovation

Alamat Supplier

JL. Letjen Sunandar Priyo Sudarmo no 36, Malang



Berhasil

Data supplier berhasil disimpan

Batal

Simpan

OK

4	SP4	PT Paragon Innovation	JL. Letjen Sunandar Priyo Sudarmo no 36, Malang	Detail	Edit
				Hapus	

Edit Data upplier

×

upplier Kode

SP4

Supplier Nama

PT Paragon Innovation

Supplier Alamat

JL. Letjen Sunandar Priyo Sudarmo no 36, Malang



Berhasil

Data supplier berhasil diubah

Batal

Simpan

OK

Hapus Data Supplier

×

⛔ Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

supplier Kode :	SP4
supplier Nama :	PT Paragon Innovation
supplier Alamat :	JL. Letjen Sunandar Priyo Sudarmo no 36, Malang

Batal

Ya, Hapus

✓ Tabel m_barang

Tambah Data Barang

Kategori Barang

Kosmetik

Barang Kode

BRG016

The barang kode has already been taken.

Barang Nama

Wardah Lightening Serum:

Harga Beli

50000

Harga Jual

68000

Batal

Simpan

OK



Berhasil

Data barang berhasil disimpan

16	BRG016	Wardah Lightening Serum:	Kosmetik	50000	68000	<div>Detail</div> <div>Hapus</div>	<div>Edit</div>
----	--------	--------------------------------	----------	-------	-------	------------------------------------	-----------------

Detail Barang

Home / Barang / Detail

Detail barang

Kesalahan!

Data yang Anda cari tidak ditemukan.

Kembali

Hapus Data Barang

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Kode Barang :

Kategori Barang :	Elektronik
BRG001	
Nama Barang :	Laptop ASUS
Harga Beli :	16000000
Harga Jual :	17500000

Batal

Ya, Hapus