

**LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT**  
**MINGGU KE VI : AJAX FORM (ADMINLTE) DAN CLIENT VALIDATION**



**OLEH :**  
**FAIZA ANATHASYA EKA FALEN**  
**2341760105 / 12**

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141

**TAHUN 2025**

Pemrograman Web Lanjut (PWL)

Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis

Semester : 4 (empat) / 5 (lima)

Pertemuan ke- : 6 (enam)

## JOBSHEET 06

### Ajax Form (AdminLTE) dan Client Validation

#### Praktikum 1. Modal Ajax Tambah Data (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.

```
template.blade.php X
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>AdminLTE 3 | Blank Page</title>
7
8     <meta name="csrf-token" content="{{ csrf_token() }}">
9
10    <!-- Font Awesome -->
11    <link rel="stylesheet" href="{{ asset('plugins/fontawesome-free/css/all.min.css') }}">
12    <!-- Datatables -->
13    <link rel="stylesheet" href="{{ asset('plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}">
14    <link rel="stylesheet" href="{{ asset('plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}">
15    <link rel="stylesheet" href="{{ asset('plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}">
16    <!-- SweetAlert2 -->
17    <link rel="stylesheet" href="{{ asset('plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
18    <!-- Theme style -->
19    <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">
20
21    @stack('css')
22 </head>
```

```

template.blade.php X
65 <script src="{{ asset('plugins/pdfmake/pdfmake.min.js') }}"></script>
66 <script src="{{ asset('plugins/pdfmake/vfs_fonts.js') }}"></script>
67 <script src="{{ asset('plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>
68 <script src="{{ asset('plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
69 <script src="{{ asset('plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>
70
71 <!-- jquery-validation -->
72 <script src="{{ asset('plugins/jquery-validation/jquery.validate.min.js') }}"></script>
73 <script src="{{ asset('plugins/jquery-validation/additional-methods.min.js') }}"></script>
74
75 <!-- SweetAlert2 -->
76 <script src="{{ asset('plugins/sweetalert2/sweetalert2.min.js') }}"></script>
77
78 <!-- AdminLTE App -->
79 <script src="{{ asset('dist/js/adminlte.min.js') }}"></script>
80 <script>
81     $.ajaxSetup({
82         headers: {
83             'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
84         }
85     });
86 </script>
87
88 @stack('js')
89 </body>
90 </html>

```

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

```

@extends('layouts.template')

@section('content')
<div class="card card-outline card-primary">
<div class="card-header">
<h3 class="card-title">{{ $page->title }}</h3>
<div class="card-tools">
<a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}">Tambah</a>
<button onclick="modalAction('{{ url('user/create_ajax') }}')" class="btn btn-sm btn-success mt-1">Tambah Ajax</button>
</div>
</div>
<div class="card-body">
@if (@session('success'))
<div class="alert alert-success">{{ session('success') }}</div>
@endif
@if (session('error'))
<div class="alert alert-danger">{{ session('error') }}</div>
@endif
<div class="row">
<div class="col-md-12">
<div class="form-group row">
<label class="col-1 control-label col-form-label">Filter:</label>
<div class="col-3">
<select class="form-control" id="level_id" name="level_id" required>
<option value="">- Semua -</option>
@foreach ($level as $item)
<option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
@endforeach

```

Kita tambahkan kode berikut, untuk membuat form modal tambah data user dengan ajax

```
<button onclick="modalAction('{{ url('/user/create_ajax') }}')" class="btn btn-sm btn-success mt-1">Tambah Ajax</button>
```

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

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

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');
    });
}
```

Sehingga tampilan kode program akan seperti berikut

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

@push('css')
@endpush

@push('js')
<script>
    function modalAction(url = ''){
        $('#myModal').load(url,function(){
            $('#myModal').modal('show');
        });
    }

    var dataUser;
    $(document).ready(function() {
        dataUser = $('#table_user').DataTable({
            // serverSide: true, jika ingin menggunakan server side processing
            serverSide: true,
            ajax: {
                "url": "{{ url('user/list') }}",
                "dataType": "json",
                "type": "POST",
                "data": function (d){
                    d.level_id = $('#level_id').val();
                }
            },
            columns: [ {
```



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

```

Route::group(['prefix' => 'user'], function () {
    Route::get('/', [UserController::class, 'index']); // Menampilkan halaman awal user
    Route::post('/list', [UserController::class, 'list']); // Menampilkan data user dalam bentuk json untuk datatables
    Route::get('/create', [UserController::class, 'create']); // Menampilkan halaman form tambah user
    Route::post('/', [UserController::class, 'store']); // Menyimpan data user baru
    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
    Route::get('/{id}', [UserController::class, 'show']); // Menampilkan detail user
    Route::get('/{id}/edit', [UserController::class, 'edit']); // Menampilkan halaman form edit user
    Route::put('/{id}', [UserController::class, 'update']); // Menyimpan perubahan data user
    Route::delete('/{id}', [UserController::class, 'destroy']); // Menghapus data user
});

```

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

```

<form action="{{ url('/user/ajax') }}" method="POST" id="form-tambah">
@csrf
<div id="modal-master" class="modal-dialog modal-lg" role="document">
    <div class="modal-content">
        <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Tambah Data User</h5>
            <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
        </div>
        <div class="modal-body">
            <div class="form-group">
                <label>Level Pengguna</label>
                <select name="level_id" id="level_id" class="form-control" required>
                    <option value="">- Pilih Level -</option>
                    @foreach($level as $l)
                        <option value="{{ $l->level_id }}">{{ $l->level_nama }}</option>
                    @endforeach
                </select>
                <small id="error-level_id" class="error-text form-text text-danger"></small>
            </div>
            <div class="form-group">
                <label>Username</label>
                <input value="" type="text" name="username" id="username" class="form-control"
required>
                <small id="error-username" class="error-text form-text text-danger"></small>
            </div>
            <div class="form-group">
                <label>Nama</label>
                <input value="" type="text" name="nama" id="nama" class="form-control"
required>
                <small id="error-nama" class="error-text form-text text-danger"></small>
            </div>
            <div class="form-group">
                <label>Password</label>
                <input value="" type="password" name="password" id="password" class="form-
control" required>
                <small id="error-password" class="error-text form-text text-danger"></small>
            </div>
        </div>
    </div>
</div>

```

```

        <div class="modal-footer">
            <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
            <button type="submit" class="btn btn-primary">Simpan</button>
        </div>
    </div>
</div>
</form>
<script>
    $(document).ready(function() {
        $("#form-tambah").validate({
            rules: {
                level_id: {required: true, number: true},
                username: {required: true, minlength: 3, maxlength: 20},
                nama: {required: true, minlength: 3, maxlength: 100},
                password: {required: true, minlength: 6, maxlength: 20}
            },
            submitHandler: function(form) {
                $.ajax({
                    url: form.action,
                    type: form.method,
                    data: $(form).serialize(),
                    success: function(response) {
                        if(response.status){
                            $('#myModal').modal('hide');
                            Swal.fire({
                                icon: 'success',
                                title: 'Berhasil',
                                text: response.message
                            });
                            dataUser.ajax.reload();
                        }else{
                            $('.error-text').text('');
                            $.each(response.msgField, function(prefix, val) {
                                $('#error-'+prefix).text(val[0]);
                            });
                            Swal.fire({
                                icon: 'error',
                                title: 'Terjadi Kesalahan',
                                text: response.message
                            });
                        }
                    }
                });
                return false;
            },
            errorElement: 'span',
            errorPlacement: function (error, element) {
                error.addClass('invalid-feedback');
                element.closest('.form-group').append(error);
            },
            highlight: function (element, errorClass, validClass) {
                $(element).addClass('is-invalid');
            },
            unhighlight: function (element, errorClass, validClass) {
                $(element).removeClass('is-invalid');
            }
        });
    });
</script>

```

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, true: berhasil
                'message' => 'Validasi Gagal',
                'msgField' => $validator->errors(), // pesan error validasi
            ]);
        }

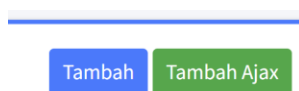
        UserModel::create($request->all());
        return response()->json([
            'status' => true,
            'message' => 'Data user berhasil disimpan'
        ]);
    }
    redirect('/');
}

```

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!!!

Setelah menambahkan code diatas terdapat beberapa perubahan pada website, yaitu :

- Terdapat button tambah ajax



- Form tambah ajax

- Pengisian form tambah ajax

- Hasil tambah data pada form ajax

11	tasya	nathasya	Manager
12	Raim	Raim Laode	Manager

## Praktikum 2. Modal Ajax Edit Data (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

```
// Ambil data user dalam bentuk json untuk datatables
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($users)
        ->addIndexColumn() // menambahkan kolom index / no urut (default nama kolom:
DT_RowIndex)
        ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
```



```

        /* $btn = '<a href="'.url('/user/' . $user->user_id).'>Detail</a>' class="btn btn-info btn-sm">Detail</a>' ;
        $btn .= '<a href="'.url('/user/' . $user->user_id . '/edit').'>Edit</a>' class="btn btn-warning btn-sm">Edit</a>' ;
        $btn .= '<form class="d-inline-block" method="POST" action="'.url('/user/' . $user->user_id).'>'
                . csrf_field() . method_field('DELETE') .
                '<button type="submit" class="btn btn-danger btn-sm" onclick="return
confirm(\''Apakah Anda yakin menghapus data ini?\'');">Hapus</button></form>' ;*/
        $btn = '<button onclick="modalAction(\'' . url('/user/' . $user->user_id .
'/show_ajax').'\'>Detail</button>' ;
        $btn .= '<button onclick="modalAction(\'' . url('/user/' . $user->user_id .
'/edit_ajax').'\'>Edit</button>' ;
        $btn .= '<button onclick="modalAction(\'' . url('/user/' . $user->user_id .
'/delete_ajax').'\'>Hapus</button>' ;

        return $btn;
    })
    ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
    ->make(true);
}

```

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

```

Route::group(['prefix' => 'user'], function () {
    Route::get('/', [UserController::class, 'index']); // Menampilkan halaman awal user
    Route::post('/list', [UserController::class, 'list']); // Menampilkan data user dalam bentuk json untuk datatables
    Route::get('/create', [UserController::class, 'create']); // Menampilkan halaman form tambah user
    Route::post('/', [UserController::class, 'store']); // Menyimpan data user baru
    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
    Route::get('/{id}', [UserController::class, 'show']); // Menampilkan detail user
    Route::get('/{id}/edit', [UserController::class, 'edit']); // Menampilkan halaman form edit user
    Route::put('/{id}', [UserController::class, 'update']); // Menyimpan perubahan data user
    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
    Route::delete('/{id}', [UserController::class, 'destroy']); // Menghapus data user
});

```

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')->get();

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

```

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

```

@empty($user)
<div id="modal-master" class="modal-dialog modal-lg" role="document">
    <div class="modal-content">
        <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Kesalahan</h5>
            <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>

```

```

        </div>
        <div class="modal-body">
            <div class="alert alert-danger">
                <h5><i class="icon fas fa-ban"></i> Kesalahan!!!</h5>
                Data yang anda cari tidak ditemukan</div>
                <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
            </div>
        </div>
    </div>
@else
    <form action="{{ url('/user/' . $user->user_id.'/update_ajax') }}" method="POST" id="form-
edit">
        @csrf
        @method('PUT')
        <div id="modal-master" class="modal-dialog modal-lg" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="exampleModallabel">Edit Data User</h5>
                    <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                </div>
                <div class="modal-body">
                    <div class="form-group">
                        <label>Level Pengguna</label>
                        <select name="level_id" id="level_id" class="form-control" required>
                            <option value="">- Pilih Level -</option>
                            @foreach($level as $l)
                                <option {{ ($l->level_id == $user->level_id)? 'selected' : '' }}
value="{{ $l->level_id }}">{{ $l->level_nama }}</option>
                            @endforeach
                        </select>
                        <small id="error-level_id" class="error-text form-text text-
danger"></small>
                    </div>
                    <div class="form-group">
                        <label>Username</label>
                        <input value="{{ $user->username }}" type="text" name="username"
id="username" class="form-control" required>
                        <small id="error-username" class="error-text form-text text-
danger"></small>
                    </div>
                    <div class="form-group">
                        <label>Nama</label>
                        <input value="{{ $user->nama }}" type="text" name="nama" id="nama"
class="form-control" required>
                        <small id="error-nama" class="error-text form-text text-danger"></small>
                    </div>
                    <div class="form-group">
                        <label>Password</label>
                        <input value="" type="password" name="password" id="password" class="form-
control">
                        <small class="form-text text-muted">Abaikan jika tidak ingin ubah
password</small>
                        <small id="error-password" class="error-text form-text text-
danger"></small>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" data-dismiss="modal" class="btn btn-
warning">Batal</button>
                    <button type="submit" class="btn btn-primary">Simpan</button>
                </div>
            </div>
        </div>
    </form>
</script>

```

```

$(document).ready(function() {
    $("#form-edit").validate({
        rules: {
            level_id: {required: true, number: true},
            username: {required: true, minlength: 3, maxlength: 20},
            nama: {required: true, minlength: 3, maxlength: 100},
            password: {minlength: 6, maxlength: 20}
        },
        submitHandler: function(form) {
            $.ajax({
                url: form.action,
                type: form.method,
                data: $(form).serialize(),
                success: function(response) {
                    if(response.status){
                        $('#myModal').modal('hide');
                        Swal.fire({
                            icon: 'success',
                            title: 'Berhasil',
                            text: response.message
                        });
                        dataUser.ajax.reload();
                    }else{
                        $('.error-text').text('');
                        $.each(response.msgField, function(prefix, val) {
                            $('#error-'+prefix).text(val[0]);
                        });
                        Swal.fire({
                            icon: 'error',
                            title: 'Terjadi Kesalahan',
                            text: response.message
                        });
                    }
                }
            });
            return false;
        },
        errorElement: 'span',
        errorPlacement: function (error, element) {
            error.addClass('invalid-feedback');
            element.closest('.form-group').append(error);
        },
        highlight: function (element, errorClass, validClass) {
            $(element).addClass('is-invalid');
        },
        unhighlight: function (element, errorClass, validClass) {
            $(element).removeClass('is-invalid');
        }
    });
});
</script>
@empty

```

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

```

public function update_ajax(Request $request, $id){
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'level_id' => 'required|integer',
            'username' => 'required|max:20|unique:m_user,username, '.$id.', user_id',
            '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: berhasil, false: gagal
        'message' => 'Validasi gagal.',
        'msgField' => $validator->errors() // menunjukkan field mana yang error
    ]);
}

$check = UserModel::find($id);
if ($check) {
    if (!$request->filled('password')) { // jika password tidak diisi, maka hapus dari
request
        $request->request->remove('password');
    }

    $check->update($request->all());
    return response()->json([
        'status' => true,
        'message' => 'Data berhasil diupdate'
    ]);
} else {
    return response()->json([
        'status' => false,
        'message' => 'Data tidak ditemukan'
    ]);
}
}
return redirect('/');
}

```

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

before :

ID	Username	Nama	Level Pengguna	Aksi
11	tasya	nathasya	Manager	Detail Edit Hapus
12	Raim	Raim Laode	Manager	Detail Edit Hapus

After :

ID	Username	Nama	Level Pengguna	Aksi
11	tasya	nathasya	Manager	Detail Edit Hapus
12	Raim	Raim	Manager	Detail Edit Hapus

### Praktikum 3. Modal Ajax Hapus Data (Data User)

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

```
Route::group(['prefix' => 'user'], function () {
    Route::get('/', [UserController::class, 'index']); // Menampilkan halaman awal user
    Route::post('/list', [UserController::class, 'list']); // Menampilkan data user dalam bentuk json untuk datatables
    Route::get('/create', [UserController::class, 'create']); // Menampilkan halaman form tambah user
    Route::post('/', [UserController::class, 'store']); // Menyimpan data user baru
    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
    Route::get('/{id}', [UserController::class, 'show']); // Menampilkan detail user
    Route::get('/{id}/edit', [UserController::class, 'edit']); // Menampilkan halaman form edit user
    Route::put('/{id}', [UserController::class, 'update']); // Menyimpan perubahan data user
    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
    Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); // Untuk tampilkan form confirm delete user Ajax
    Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); // Untuk hapus data user Ajax
    Route::delete('/{id}', [UserController::class, 'destroy']); // Menghapus data user
});
```

3. 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]);
}
```

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

```
@empty($user)
    <div id="modal-master" class="modal-dialog modal-lg" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModallabel">Kesalahan</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
            </div>
            <div class="modal-body">
                <div class="alert alert-danger">
                    <h5><i class="icon fas fa-ban"></i> Kesalahan!!!</h5>
                    Data yang anda cari tidak ditemukan</div>
                    <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
                </div>
            </div>
        </div>
    </div>
@else
    <form action="{{ url('/user/' . $user->user_id . '/delete_ajax') }}" method="POST" id="form-delete">
        @csrf
        @method('DELETE')
        <div id="modal-master" class="modal-dialog modal-lg" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="exampleModallabel">Hapus Data User</h5>
                    <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                </div>
                <div class="modal-body">
                    <div class="alert alert-warning">
                        <h5><i class="icon fas fa-ban"></i> Konfirmasi !!!</h5>
                        Apakah Anda ingin menghapus data seperti di bawah ini?
                    </div>
                    <table class="table table-sm table-bordered table-striped">
                        <tr><th class="text-right col-3">Level Pengguna :</th><td class="col-9">{{
                        $user->level->level_nama }}</td></tr>
                        <tr><th class="text-right col-3">Username :</th><td class="col-9">{{
                        $user->username }}</td></tr>
                        <tr><th class="text-right col-3">Nama :</th><td class="col-9">{{ $user-
                        >nama }}</td></tr>
                    </table>
                </div>
                <div class="modal-footer">
                    <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
                    <button type="submit" class="btn btn-primary">Ya, Hapus</button>
                </div>
            </div>
        </div>
    </form>
    <script>
        $(document).ready(function() {
            $("#form-delete").validate({
                rules: {},
                submitHandler: function(form) {
                    $.ajax({
```

```

        url: form.action,
        type: form.method,
        data: $(form).serialize(),
        success: function(response) {
            if(response.status){
                $('#myModal').modal('hide');
                Swal.fire({
                    icon: 'success',
                    title: 'Berhasil',
                    text: response.message
                });
                dataUser.ajax.reload();
            }else{
                $('#error-text').text('');
                $.each(response.msgField, function(prefix, val) {
                    $('#error-' + prefix).text(val[0]);
                });
                Swal.fire({
                    icon: 'error',
                    title: 'Terjadi Kesalahan',
                    text: response.message
                });
            }
        }
    });
    return false;
},
errorElement: 'span',
errorPlacement: function (error, element) {
    error.addClass('invalid-feedback');
    element.closest('.form-group').append(error);
},
highlight: function (element, errorClass, validClass) {
    $(element).addClass('is-invalid');
},
unhighlight: function (element, errorClass, validClass) {
    $(element).removeClass('is-invalid');
}
});
</script>
@endempty

```

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.

Sebelum di delete :

ID	Username	Nama	Level Pengguna	Aksi
11	tasya	nathasya	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
12	Raim	Raim	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
13	lita	lalita	Administratorr	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Saat klik button delete muncul tampilan seperti ini :

Hapus Data User

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Level : Administratorr

Pengguna :

Username : lita

Nama : lalita

Batal

Ya, Hapus

Setelah di delete :

ID	Username	Nama	Level Pengguna	Aksi
11	tasya	nathasya	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
12	Raim	Raim	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

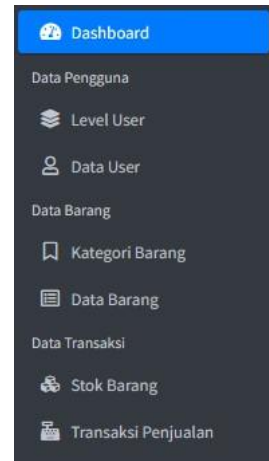
7. Jangan lupa laporkan ke laporan jobsheet dan lakukan *commit* pada *repository git* kalian.!!

## B. Tugas Jobsheet

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

- ✓ Tabel m\_level
- ✓ Tabel m\_kategori
- ✓ Tabel m\_supplier
- ✓ Tabel m\_barang

Laporkan pada laporan jobsheet dan Jangan lupa di commit dan push pada repository git kalian.



### TUGAS IMPLEMENTASI :

#### • Tabel m\_level

The 'Daftar Level' page displays a table of user levels. The table has columns for ID, Level Kode, Level Nama, and aksi. There are three entries: ADM (Administrator), MNG (Manager), and STF (Staff / Kasir). Each entry has 'Detail', 'Edit', and 'Hapus' buttons. The page also includes a filter dropdown, a search bar, and pagination controls.

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	Detail Edit Hapus
2	MNG	Manager	Detail Edit Hapus
3	STF	Staff / Kasir	Detail Edit Hapus

The 'Tambah Data Level' modal form contains two input fields: 'Kode Level' and 'Level Nama'. At the bottom, there are 'Batal' and 'Simpan' buttons.

#### • Tabel m\_kategori

The 'Daftar Kategori' page displays a table of product categories. The table has columns for ID, Kode Kategori, Nama Kategori, and Aksi. There are five entries: KAT001 (Makanan), KAT002 (Minuman), KAT003 (Obat), KAT004 (Elektronik), and KAT005 (Meubel). Each entry has 'Detail', 'Edit', and 'Hapus' buttons. The page also includes a search bar and pagination controls.

ID	Kode Kategori	Nama Kategori	Aksi
1	KAT001	Makanan	Detail Edit Hapus
2	KAT002	Minuman	Detail Edit Hapus
3	KAT003	Obat	Detail Edit Hapus
4	KAT004	Elektronik	Detail Edit Hapus
5	KAT005	Meubel	Detail Edit Hapus

The 'Tambah Data Kategori' modal form contains two input fields: 'Kode Kategori' and 'Kategori Nama'. At the bottom, there are 'Batal' and 'Simpan' buttons.

- Tabel m\_supplier

Daftar Supplier

Home / Supplier

Daftar supplier yang terdaftar dalam sistem

Tambah

Tambah Baru

Show 10 entries

Search:

#	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SUP001	PT. Indofood CBP Sukasa Makmur Tbk	Perum Pondok Blimbing Indah, Jl. Raya Tlogomas No.31, Samsan, Kec. Klagen, Kota Malang, Jawa Timur 65144	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	SUP002	Mebel Bontobudur	Jl. Kauman No.10, Kauman, Kec. Klagen, Kota Malang, Jawa Timur 65119	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	SUP003	Cimory Center Blimbing	Purwodadi, Kec. Blimbing, Kota Malang, Jawa Timur	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
4	SUP004	Apple Store MGG	Jl. Kawi No.44, Gading Kaori, Kec. Klagen, Kota Malang, Jawa Timur 65116	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
5	SUP005	PT Industri Jamu Dan Farmasi Sido Muncul Tbk.	Jl. Gajahmada No.123, Kel. Pekunden, Kec. Semarang Tengah, Semarang, 50134	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

Tambah Data supplier

Kode supplier

Supplier Nama

Alamat Supplier

Batal

Simpan

- Tabel m\_barang

Data Barang

Home / Barang

Daftar barang yang terdaftar dalam sistem

Tambah

Tambah Baru

Filter: - Semua -

Kategori Barang

Show 10 entries

Search:

#	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
1	BRG001	Rak Buku	Meubel	25000	30000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	BRG002	Container Barang	Meubel	47500	50000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	BRG003	Indomie Goreng	Makanan	2500	3000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
4	BRG004	Cimory Yogurt	Minuman	5500	8000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
5	BRG005	Milku Susu Strawberry	Minuman	4500	6000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
6	BRG006	Kopi Kapal Api	Minuman	5000	7000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
7	BRG007	Tolak Angin	Obat	3000	5000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
8	BRG008	Necozp	Obat	2500	4000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
9	BRG009	Earphone	Elektronik	125000	128000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
10	BRG010	Iphone 15	Elektronik	15000000	16000000	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

Tambah Data Barang

Kategori Barang

Pilih Kategori

Barang Kode

Barang Nama

Harga Beli

Harga Jual

Batal

Simpan