

LAPORAN JOBSHEET 6
PEMROGRAMAN WEB LANJUT
Ajax Form (AdminLTE) dan Client Validation

Dosen pengajar: Dimas Wahyu Wibowo, ST., MT



Disusun Oleh:

Nama : M. Fatih Al Ghifary

NIM : 2341720194

Kelas : TI-2A

No : 16

PRODI TEKNIK INFORMATIKA D-IV
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2025

Praktikum 1 - Modal Ajax Tambah Data (Data User)

1. Membuat form tambah data dengan modal dan proses ajax.
2. Menambahkan library jQuery Validation dan SweetAlert.

```
<!-- SweetAlert2 -->
<link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
<!-- Theme styles -->
<!-- jquery-validation -->
<script src="{{ asset(path: 'adminlte/plugins/jquery-validation/jquery.validate.min.js') }}"></script>
<script src="{{ asset(path: 'adminlte/plugins/jquery-validation/additional-methods.min.js') }}"></script>
<!-- SweetAlert2 -->
<script src="{{ asset(path: 'adminlte/plugins/sweetalert2/sweetalert2.min.js') }}"></script>
```

3. Memodifikasi view user/index.blade.php untuk tombol modal.

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

4. Menambahkan modal form tambah data user dengan ajax.

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

5. Memodifikasi route/web.php untuk 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
    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']);
    Route::post('/ajax', [UserController::class, 'store_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
});
```

6. Menambahkan fungsi create_ajax() pada UserController.php.

```
public function create_ajax(): mixed|View
{
    $level = LevelModel::select('level_id', 'level_nama')->get();

    return view('user.create_ajax')->with('level', $level);
}
```

7. Membuat view baru user/create_ajax.blade.php.

```

1 <form action="{{ url(path: '/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" required>
24                <small id="error-username" class="error-text form-text text-danger"></small>
25            </div>
26            <div class="form-group">
27                <label>Nama</label>
28                <input value="" type="text" name="nama" id="nama" class="form-control" required>
29                <small id="error-nama" class="error-text form-text text-danger"></small>
30            </div>
31            <div class="form-group">
32                <label>Password</label>
33                <input value="" type="password" name="password" id="password" class="form-control" required>
34                <small id="error-password" class="error-text form-text text-danger"></small>
35            </div>
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">Simpan</button>
40        </div>
41    </div>
42 </div>
43 </form>
44 <script>
45     $(document).ready(function () {
46         $("#form-tambah").validate({
47             rules: {
48                 level_id: { required: true, number: true },
49                 username: { required: true, minlength: 3, maxlength: 20 },
50                 nama: { required: true, minlength: 3, maxlength: 100 },
51                 password: { required: true, minlength: 6, maxlength: 20 }
52             },
53             submitHandler: function (form) {
54                 $.ajax({
55                     url: form.action,
56                     type: form.method,
57                     data: $(form).serialize(),
58                     success: function (response) {
59                         if (response.status) {
60                             $('#myModal').modal('hide');
61                             Swal.fire({
62                                 icon: 'success',
63                                 title: 'Berhasil',
64                                 text: response.message
65                             });
66                         }
67                     }
68                 });
69             }
70         });
71     });
72 </script>

```

8. Membuat fungsi store_ajax() pada UserController.php.

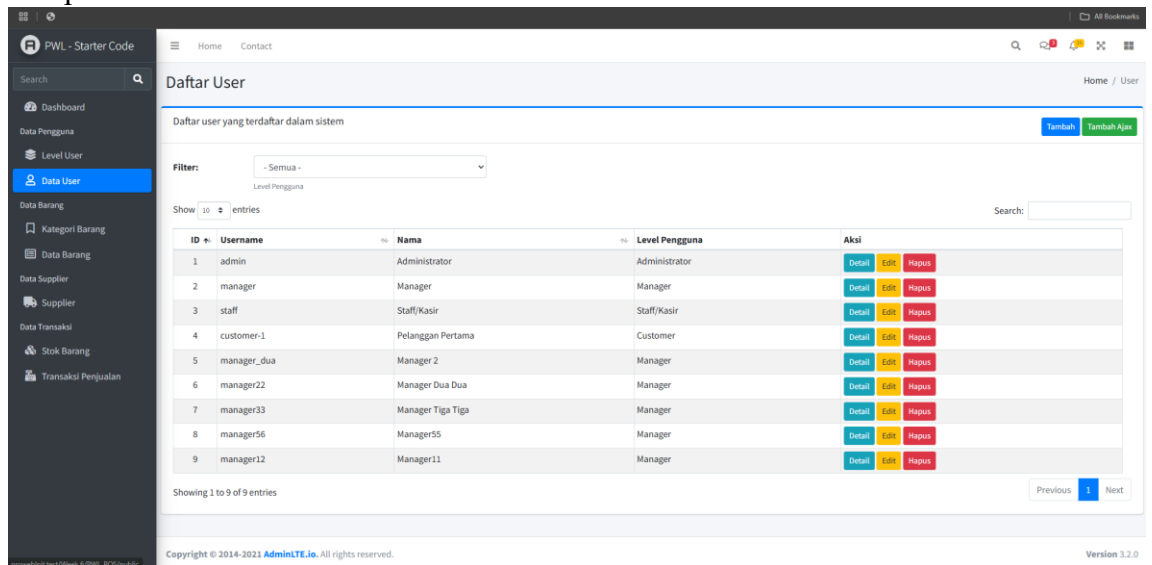
```

1 public function store_ajax(Request $request): JsonResponse|mixed
2 {
3     // cek apakah request berupa ajax
4     if ($request->ajax() || $request->wantsJson()) {
5         $rules = [
6             'level_id' => 'required|integer',
7             'username' => 'required|string|min:3|unique:m_user,username',
8             'nama' => 'required|string|max:100',
9             'password' => 'required|min:6'
10        ];
11
12        $validator = Validator::make($request->all(), $rules);
13
14        if ($validator->fails()) {
15            return response()->json([
16                'status' => false,
17                'message' => 'Validasi Gagal',
18                'msgField' => $validator->errors(),
19            ]);
20        }
21
22        UserModel::create($request->all());
23        return response()->json([
24            'status' => true,
25            'message' => 'Data user berhasil disimpan'
26        ]);
27    }
28    redirect(to: '/');
29 }

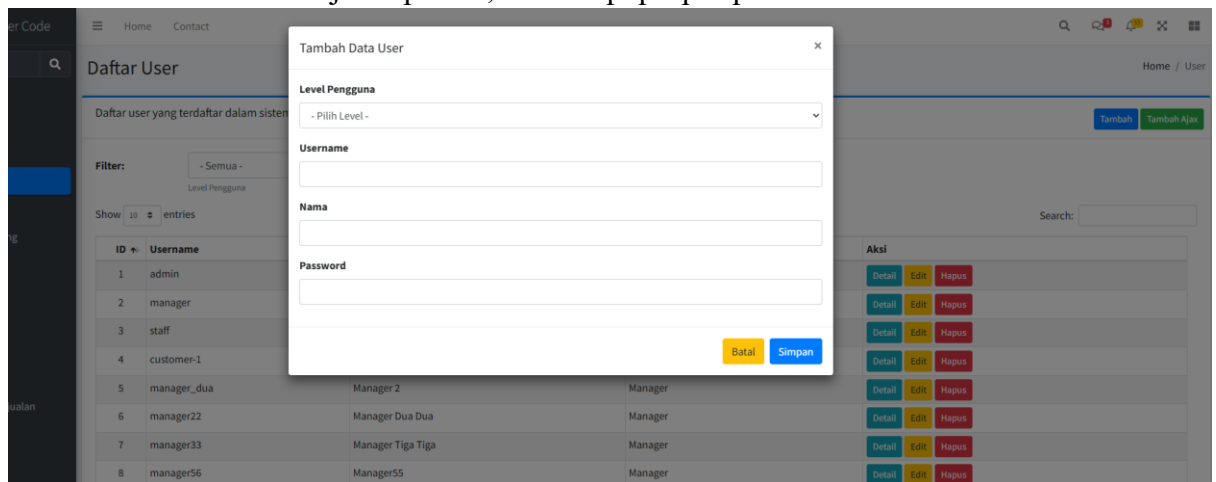
```

9. Melakukan pengujian tambah data user menggunakan form ajax.

- Tampilan awal



- Ketika tombol Tambah Ajax dipencet, muncul pop-up seperti ini



- Mengisikan data

Tambah Data User ×

Level Pengguna

Customer

Username

Wowo

Nama

Sikowokowo

Password

Batal

Simpan

- Klik tombol simpan, maka akan berhasil disimpan

Daftar User

Daftar user yang terdaftar dalam sistem

Filter: Level Pengguna

how 10 entries

Search:

ID	Username	Nama	Aksi
1	admin	Admin	Detail Edit Hapus
2	manager	Manag	Detail Edit Hapus
3	staff	Staff/K	Detail Edit Hapus
4	customer-1	Pelangu	Detail Edit Hapus
5	manager_dua	Manag	Detail Edit Hapus
6	manager22	Manag	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Detail Edit Hapus
8	manager56	Manager55	Detail Edit Hapus
9	manager12	Manager11	Detail Edit Hapus
10	Wowo	Sikowokowo	Detail Edit Hapus

Showing 1 to 10 of 10 entries

Previous 1

Praktikum 2 - Modal Ajax Edit Data (Data User)

1. Menyesuaikan fungsi list() pada UserController.php.

```
public function list(Request $request): JsonResponse
{
    $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
        ->with('level');
    // Filter data user by level_id
    if ($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): string { // Menambahkan kolom aksi
            // $btn = '<a href="' . url('/user/' . $user->user_id) . '" class="btn btn-info btn-sm">Detail</a>';
            // $btn .= '<a href="' . url('/user/' . $user->user_id) . '/edit' . '" 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(path: '/user/' . $user->user_id) . '/show_ajax') . '\'" class="btn btn-info btn-sm">Detail</button>';
            $btn .= '<button onclick="modalAction(\'' . url(path: '/user/' . $user->user_id) . '/edit_ajax') . '\'" class="btn btn-warning btn-sm">Edit</button>';
            $btn .= '<button onclick="modalAction(\'' . url(path: '/user/' . $user->user_id) . '/delete_ajax') . '\'" class="btn btn-danger btn-sm">Hapus</button>';
            return $btn;
        });
    ->rawColumns(['aksi']) // Memberitahu bahwa kolom aksi mengandung HTML
    ->make(true);
}
```

2. Memodifikasi routes/web.php untuk operasi edit ajax.

```
Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']);
Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']);
Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
```

3. Membuat fungsi edit_ajax() pada UserController.php.

```
1 reference|0 overrides
public function edit_ajax($id): Factory|View
{
    $user = UserModel::find($id);
    $level = LevelModel::select('level_id', 'level_nama')->get();

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

4. Membuat view baru 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
13 </div>
14 <a href="{{ url(path: '/user') }}" class="btn btn-warning">Kembali</a>
15 </div>
16 </div>
17 </div>
18 @else
19 <form action="{{ url(path: '/user/' . $user->user_id . '/update_ajax') }}" method="POST" id="form-edit">
20 @csrf
21 @method('PUT')
22 <div id="modal-master" class="modal-dialog modal-lg" role="document">
23 <div class="modal-content">
24 <div class="modal-header">
25 <h5 class="modal-title" id="exampleModalLabel">Edit Data User</h5>
26 <button type="button" class="close" data-dismiss="modal" aria
27 label="Close"><span aria-hidden="true">&times;</span></button>
28 </div>
29 <div class="modal-body">
30 <div class="form-group">
31 <label>Level Pengguna</label>
32 <select name="level_id" id="level_id" class="form-control" required>
33 <option value="">- Pilih level -</option>
34 @foreach($level as $l)
35 <option {{ ($l->level_id == $user->level_id)? 'selected' : '' }}
36 value="{{ ($l->level_id) }}">{{ ($l->level_nama) }}</option>
37 @endforeach
38 </select>
39 <small id="error-level_id" class="error-text form-text text-danger"></small>
40 </div>
41 <div class="form-group">
42 <label>Username</label>
43 <input value="{{ $user->username }}" type="text" name="username"
44 id="username" class="form-control" required>
45 <small id="error-username" class="error-text form-text text-danger"></small>
46 </div>
47 <div class="form-group">
48 <label>Nama</label>
49 <input type="text" name="nama" class="form-control" required>
50 <small id="error-nama" class="error-text form-text text-danger"></small>
51 </div>
52 </div>
53 </div>
54 </div>
```

5. Membuat fungsi update_ajax() pada UserController.php.

```
public function update_ajax(Request $request, $id): JsonResponse|mixed|Redirector|RedirectRes...
{
    // cek apakah request dari ajax
    if ($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'
        ];

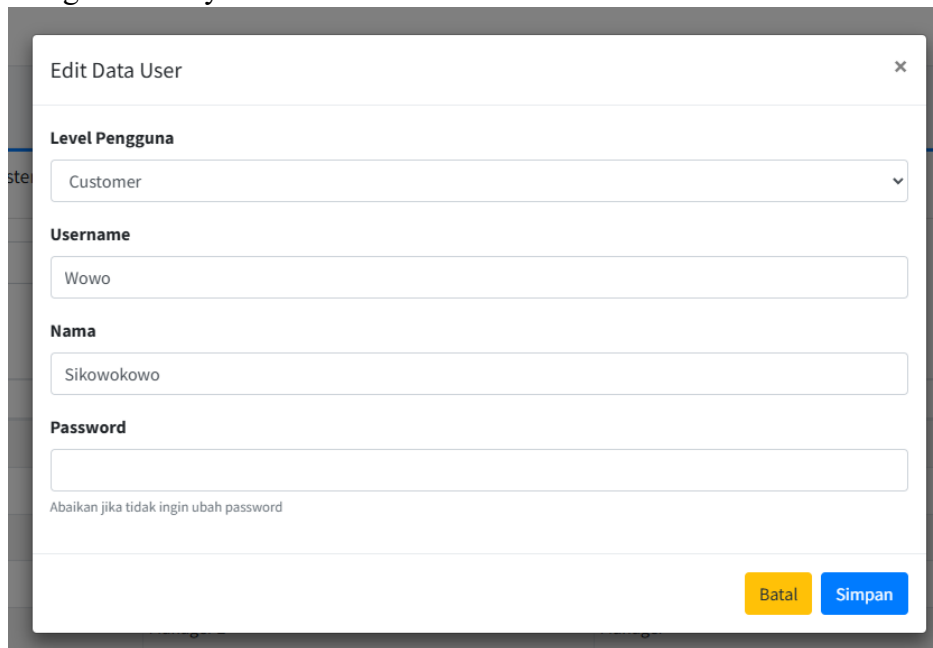
        $validator = Validator::make($request->all(), $rules);

        if ($validator->fails()) {
            return response()->json([
                'status' => false,
                'message' => 'Validasi gagal.',
                'msgField' => $validator->errors()
            ]);
        }
        $check = UserModel::find($id);
        if ($check) {
            if (!$request->filled('password')) {
                $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(to: '/user');
}
```

6. Menguji proses edit user dengan ajax.

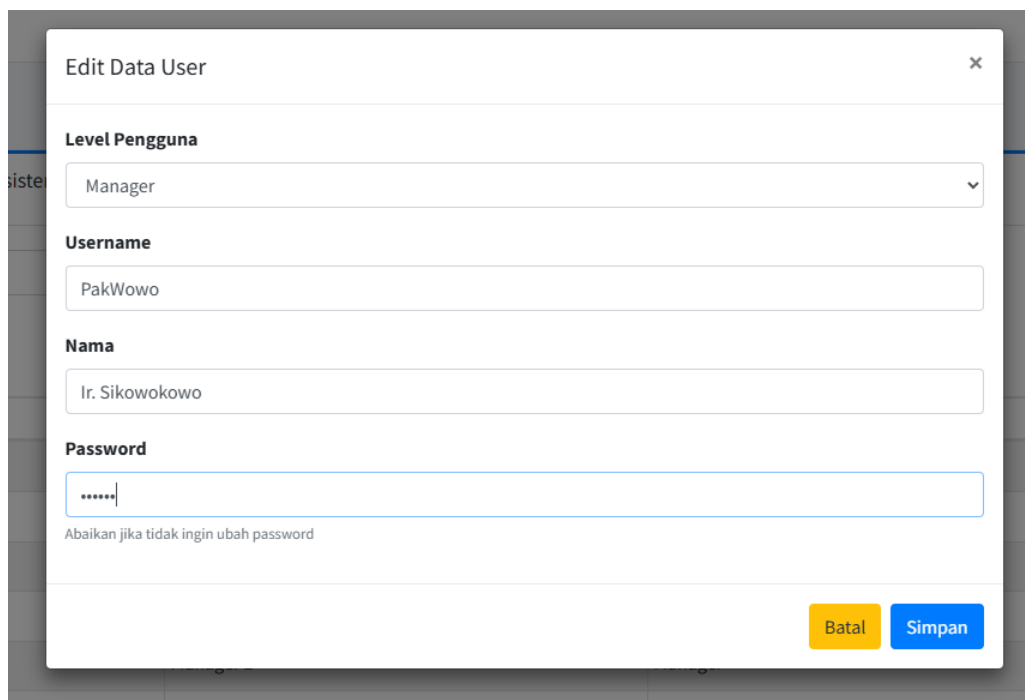
- Ketika menekan tombol edit muncul pop up seperti ini, dan disini saya mencoba mengedit datanya



The screenshot shows a modal window titled "Edit Data User" with a close button (X) in the top right corner. The form contains the following fields:

- Level Pengguna:** A dropdown menu currently showing "Customer".
- Username:** A text input field containing "Wowo".
- Nama:** A text input field containing "Sikowokowo".
- Password:** An empty text input field.

Below the password field, there is a small text label: "Abaikan jika tidak ingin ubah password". At the bottom right of the modal, there are two buttons: "Batal" (yellow) and "Simpan" (blue).

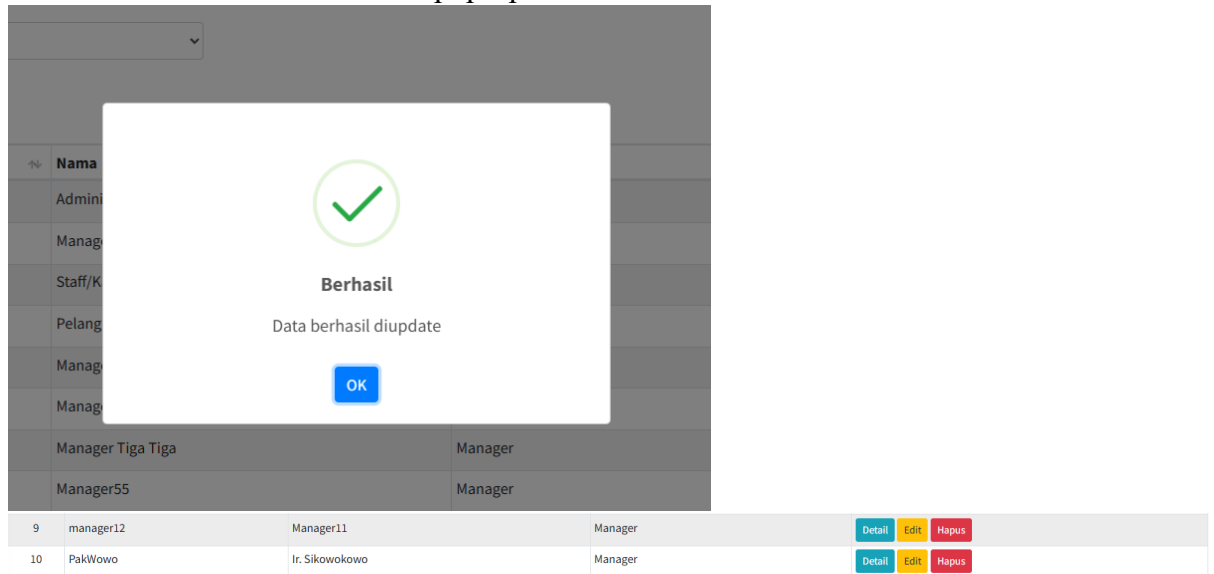


The screenshot shows the same "Edit Data User" modal window, but with the following edited values:

- Level Pengguna:** The dropdown menu now shows "Manager".
- Username:** The text input field now contains "PakWowo".
- Nama:** The text input field now contains "Ir. Sikowokowo".
- Password:** The text input field now contains six dots ".....".

The "Abaikan jika tidak ingin ubah password" label and the "Batal" and "Simpan" buttons remain the same as in the previous screenshot.

- Jika berhasil akan memunculkan pop-up berhasil dan data akan berubah



Praktikum 3. Modal Ajax Hapus Data (Data User)

1. Memodifikasi routes/web.php untuk konfirmasi hapus data.

```
Route::get('/{id}/show_ajax', [UserController::class, 'show_ajax']);
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']);
Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']);
Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']);
Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']);
Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
```

2. Membuat fungsi confirm_ajax() pada UserController.php.

```
public function confirm_ajax(string $id): Factory|View
{
    $user = UserModel::find($id);

    return view('user.confirm_ajax', data: ['user' => $user]);
}
```


3. Membuat view user/confirm_ajax.blade.php untuk konfirmasi hapus.

```
@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(path: '/user') }}" class="btn btn-warning">Kembali</a>
</div>
</div>
</div>
@else
<form action="{{ url(path: '/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: {},

```

4. Membuat fungsi delete_ajax() pada UserController.php.

```
public function delete_ajax(request $request, $id): JsonResponse|mixed|Redirector|RedirectRes...
{
    if ($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(to: '/');
}
```

5. Menguji proses hapus data user dengan ajax.

- Disini saya mencoba menghaspus data dengan ID 10

- Muncul pop-up konfirmasi data user seperti ini

Hapus Data User

Konfirmasi !!!

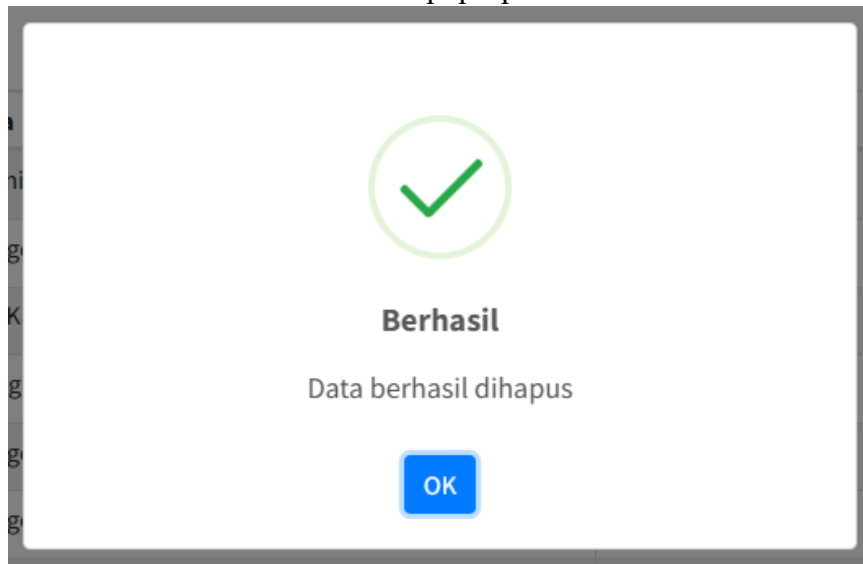
Apakah Anda ingin menghapus data seperti di bawah ini?

Level Pengguna :	Manager
Username :	PakWowo
Nama :	Ir. Sikowokowo

Batal

Ya, Hapus

- Jika berhasil maka akan muncul pop-up berhasil



- Dan disini dapat dilihat bahwa data sudah terhapus

Show 10 entries

Search:

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Pelanggan Pertama	Customer	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus

Showing 1 to 9 of 9 entries

Previous

1

Next