

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT
AJAX FORM (ADMINLTE) DAN CLIENT VALIDATION



Disusun Oleh :

Qusnul Diah Mawanti

2341760035

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2025



Jurusan Teknologi Informasi – Politeknik Negeri
Malang
Jobsheet 6 : Ajax Form (AdminLTE) dan Client Validation
Mata Kuliah : Pemrograman Web Lanjut
Kelas : 2A – SIB
Nama : Qusnul Diah Mawanti
No absen : 25
NIM : 2341760035

Maret 2025

PERSIAPAN

View merupakan tempat bagi kita untuk meletakkan kode-kode HTML. Sehingga dalam laravel tidak menggunakan lagi file berformat .html. Kita masih menggunakan tagging HTML, tapi kita perlu handle tampilan dengan lebih canggih. Menampilkan data yang diberikan oleh controller. Untuk itu kita akan menggunakan templating engine, yaitu Blade.

Blade merupakan templating engine bawaan Laravel. Berguna untuk mempermudah dalam menulis kode tampilan. Dan juga memberikan fitur tambahan untuk memanipulasi data di view yang dilempar dari controller. Blade juga memungkinkan penggunaan plain PHP pada kode View.

Sesuai dengan **Studi Kasus PWL.pdf**.

Jadi project Laravel 10 kita masih sama dengan menggunakan repositori **PWL_POS**.

Project PWL_POS akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

```
C:\laragon\www>composer create-project "laravel/laravel:^10.0" PWL_POS
Creating a "laravel/laravel:^10.0" project at "./PWL_POS"
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\laragon\www\PWL_POS
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.3)
```

DASAR TEORI

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 ke [template.blade.php](#), library sudah disediakan oleh adminLTE.

```
<!-- Font Awesome -->
<link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
<!-- DataTables -->
<link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}">
<link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}">
<link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}">
<!-- SweetAlert2 -->
<link rel="stylesheet" href="{{ asset('plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
<!-- Theme style -->
<link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">

<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>

<!-- jquery-validation -->
<script src="{{ asset('plugins/jquery-validation/jquery.validate.min.js') }}"></script>
<script src="{{ asset('plugins/jquery-validation/additional-methods.min.js') }}"></script>

<!-- SweetAlert2 -->
<script src="{{ asset('plugins/sweetalert2/sweetalert2.min.js') }}"></script>

<!-- AdminLTE App -->
<script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
```

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

```
@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">
```

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>
```

- 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>
```

- 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

```
@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,
```

- f. Selanjutnya Kita modifikasi [route/web.php](#) untuk mengkomodir 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
```

- g. 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);
}
```

- h. Setelah itu, kita buat **view baru** [user/create_ajax.blade.php](#) untuk menampilkan form dengan ajax

```
resources > views > user > create_ajax.blade.php > form#form-tambah
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">
8 <span aria-hidden="true">&times;</span>
9 </button>
10 </div>
11 <div class="modal-body">
12 <div class="form-group">
13 <label>Level Pengguna</label>
14 <select name="level_id" id="level_id" class="form-control" required>
15 <option value="">Pilih Level</option>
16 @foreach ($level as $l)
17 <option value="{{ $l->level_id }}">{{ $l->level_nama }}</option>
18 @endforeach
19 </select>
20 <small id="error-level_id" class="error-text form-text text-danger"></small>
21 </div>
22 <div class="form-group">
23 <label>Username</label>
24 <input type="text" name="username" id="username" class="form-control" 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 type="text" name="nama" id="nama" class="form-control" required>
```

```

30         <small id="error-nama" class="error-text form-text text-danger"></small>
31     </div>
32     <div class="form-group">
33         <label>Password</label>
34         <input type="password" name="password" id="password" class="form-control" required>
35         <small id="error-password" class="error-text form-text text-danger"></small>
36     </div>
37 </div>
38 <div class="modal-footer">
39     <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
40     <button type="submit" class="btn btn-primary">Simpan</button>
41 </div>
42 </div>
43 </div>
44 </form>
45 <script>
46 $(document).ready(function() {
47     $("#form-tambah").validate({
48         rules: {
49             level_id: { required: true, number: true },
50             username: { required: true, minlength: 3, maxlength: 20 },
51             nama: { required: true, minlength: 3, maxlength: 100 },
52             password: { required: true, minlength: 6, maxlength: 20 }
53         },
54         submitHandler: function(form) {
55             $.ajax({
56                 url: form.action,
57                 type: form.method,
58                 data: $(form).serialize(),
59                 success: function(response) {
60                     if (response.status) {
61                         $('#myModal').modal('hide');
62                         Swal.fire({
63                             icon: 'success',
64                             title: 'Berhasil',
65                             text: response.message
66                         });
67                         dataUser.ajax.reload();
68                     } else {
69                         $('.error-text').text('');
70                         $.each(response.msgField, function(prefix, val) {
71                             $('#error-' + prefix).text(val[0]);
72                         });
73                         Swal.fire({
74                             icon: 'error',
75                             title: 'Terjadi Kesalahan',
76                             text: response.message
77                         });
78                     }
79                 }
80             });
81             return false;
82         },
83         errorElement: 'span',
84         errorPlacement: function(error, element) {
85             error.addClass('invalid-feedback');
86             element.closest('.form-group').append(error);
87         },
88         highlight: function(element) {
89             $(element).addClass('is-invalid');
90         },
91         unhighlight: function(element) {
92             $(element).removeClass('is-invalid');
93         }
94     });
95 });
96 </script>

```

- i. 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->wantJson()) {
        $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);

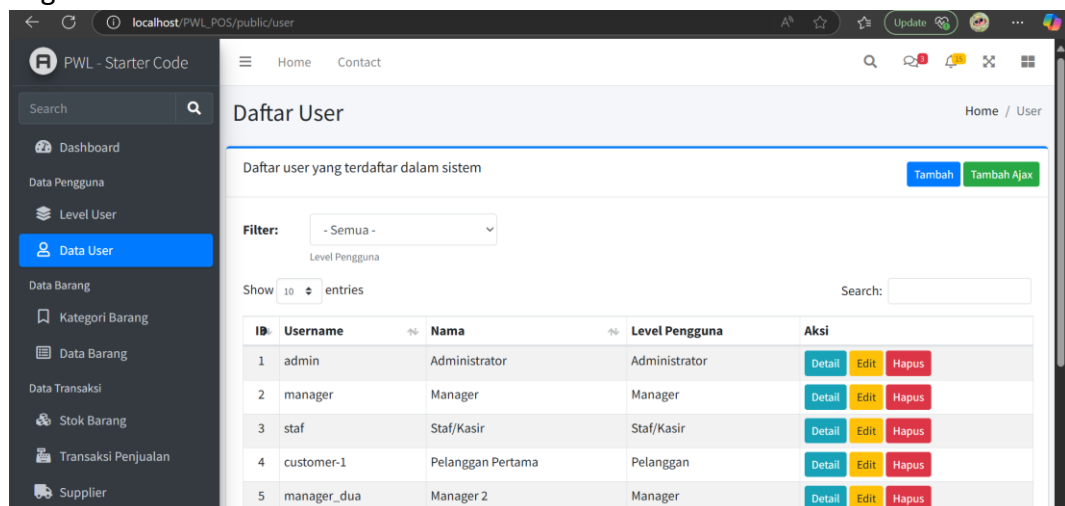
        if($validator->fails()) {
            return response()->json([
                'status' => false, // response status, false: error/gagal, true: berhasil
                'message' => 'Validasi gagal',
                'msgField' => $validator->errors() // pesan error validasi
            ]);
        }

        UserModel::create($request->all());

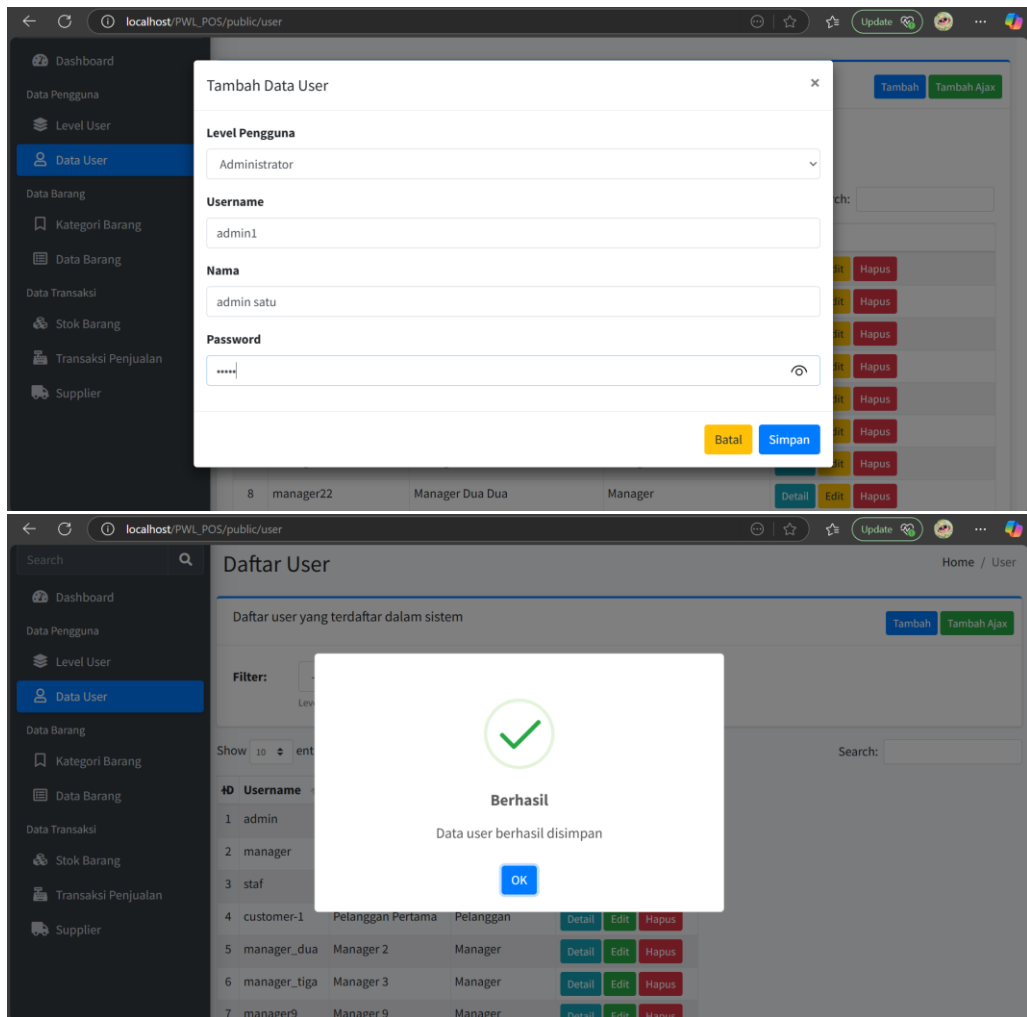
        return response()->json([
            'status' => true,
            'message' => 'Data user berhasil disimpan'
        ]);
    }

    redirect('/');
}
```

- j. 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!!!



Tambah Data User Menggunakan Ajax



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

- Pada Praktikum 2 ini, kita akan melakukan coding untuk proses edit menggunakan ajax.
- 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)
        // Menambahkan kolom index / nomor urut (default nama kolom: DT_RowIndex)
        ->addIndexColumn()

        // Menambahkan kolom aksi
        ->addColumn('aksi', function ($user) {
            $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-success 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>';

            return $btn;
        });
}
```

```

// Memberitahu bahwa kolom aksi berisi HTML
->rawColumns(['aksi'])

->make(true);
}

```

- c. 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

```

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

```

// Menampilkan halaman form edit user
public function edit(string $id)
{
    $user = UserModel::find($id);
    $level = LevelModel::all();

    $breadcrumb = (object) ['title' => 'Edit User', 'list' => ['Home', 'User', 'Edit']];

    $page = (object) ['title' => 'Edit user'];

    $activeMenu = 'user';

    return view('user.edit', [
        'breadcrumb' => $breadcrumb,
        'page' => $page,
        'user' => $user,
        'level' => $level,
        'activeMenu' => $activeMenu
    ]);
}

```

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

```

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">
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> Kesalahan!!!</h5>
13                     Data yang anda cari tidak ditemukan
14                 </div>
15                 <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
16             </div>
17         </div>
18     </div>
19 @else
20     <form action="{{ url('/user/' . $user->user_id . '/update_ajax') }}" method="POST" id="form-edit">
21         @csrf
22         @method('PUT')
23
24         <div id="modal-master" class="modal-dialog modal-lg" role="document">
25             <div class="modal-content">
26                 <div class="modal-header">
27                     <h5 class="modal-title" id="exampleModalLabel">Edit Data User</h5>
28                     <button type="button" class="close" data-dismiss="modal" aria-label="Close">
29                         <span aria-hidden="true">&times;</span>

```



```

30         </button>
31     </div>
32
33     <div class="modal-body">
34         <div class="form-group">
35             <label>Level Pengguna</label>
36             <select name="level_id" id="level_id" class="form-control" required>
37                 <option value="">- Pilih Level -</option>
38                 @foreach($level as $l)
39                     <option value="{{ $l->level_id }}" {{ $l->level_id == $user->level_id ? 'selected' : '' }}>
40                         {{ $l->level_nama }}
41                     </option>
42                 @endforeach
43             </select>
44             <small id="error-level_id" class="error-text form-text text-danger"></small>
45         </div>
46
47         <div class="form-group">
48             <label>Username</label>
49             <input value="{{ $user->username }}" type="text" name="username" id="username" class="form-control" required>
50             <small id="error-username" class="error-text form-text text-danger"></small>
51         </div>
52
53         <div class="form-group">
54             <label>Nama</label>
55             <input value="{{ $user->nama }}" type="text" name="nama" id="nama" class="form-control" required>
56             <small id="error-nama" class="error-text form-text text-danger"></small>
57         </div>
58
59         <div class="form-group">
60             <label>Password</label>
61             <input type="password" name="password" id="password" class="form-control">
62             <small class="form-text text-muted">Abaikan jika tidak ingin mengubah password</small>
63             <small id="error-password" class="error-text form-text text-danger"></small>
64         </div>
65     </div>
66
67     <div class="modal-footer">
68         <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
69         <button type="submit" class="btn btn-primary">Simpan</button>
70     </div>
71 </div>
72 </div>
73 </form>
74
75 <script>
76     $(document).ready(function() {
77         $("#form-edit").validate({
78             rules: {
79                 level_id: { required: true, number: true },
80                 username: { required: true, minlength: 3, maxlength: 20 },
81                 nama: { required: true, minlength: 3, maxlength: 100 },
82                 password: { minlength: 6, maxlength: 20 }
83             },
84             submitHandler: function(form) {
85                 $.ajax({
86                     url: form.action,
87                     type: form.method,
88                     data: $(form).serialize(),
89                     success: function(response) {
90                         if(response.status) {
91                             $('#myModal').modal('hide');
92                             Swal.fire({
93                                 icon: 'success',
94                                 title: 'Berhasil',
95                                 text: response.message
96                             });
97                             dataUser.ajax.reload();
98                         } else {
99                             $('.error-text').text('');
100                             $.each(response.msgField, function(prefix, val) {
101                                 $('#error-' + prefix).text(val[0]);
102                             });
103                             Swal.fire({
104                                 icon: 'error',
105                                 title: 'Terjadi Kesalahan',

```

```

106         text: response.message
107     });
108 }
109 }
110 });
111 return false;
112 },
113 errorElement: 'span',
114 errorPlacement: function(error, element) {
115     error.addClass('invalid-feedback');
116     element.closest('.form-group').append(error);
117 },
118 highlight: function(element, errorClass, validClass) {
119     $(element).addClass('is-invalid');
120 },
121 unhighlight: function(element, errorClass, validClass) {
122     $(element).removeClass('is-invalid');
123 }
124 });
125 });
126 </script>
127 @endempty

```

- f. 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,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
            ]);
        }

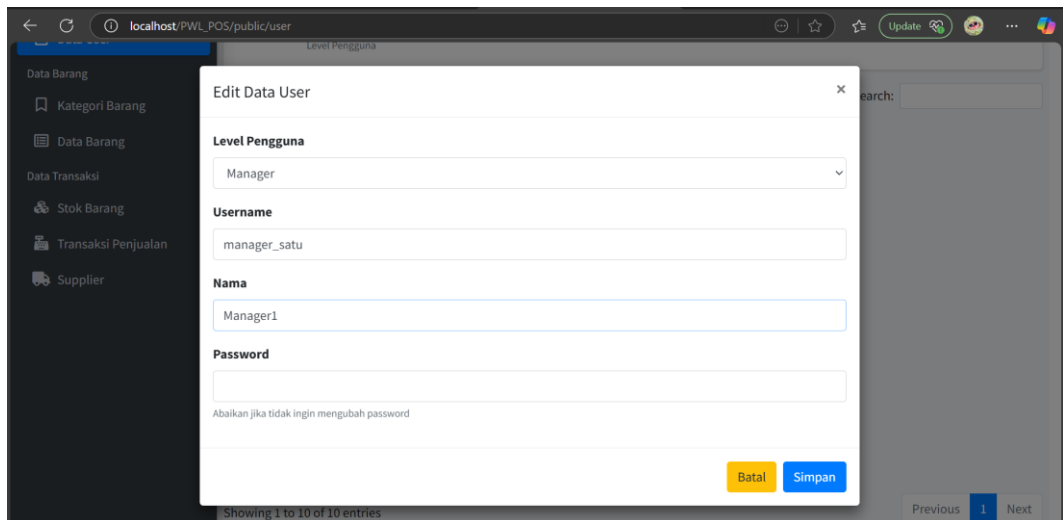
        $check = UserModel::find($id);
        if ($check) {
            // Jika password tidak diisi, hapus dari request
            if (!$request->filled('password')) {
                $request->request->remove('password');
            }

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

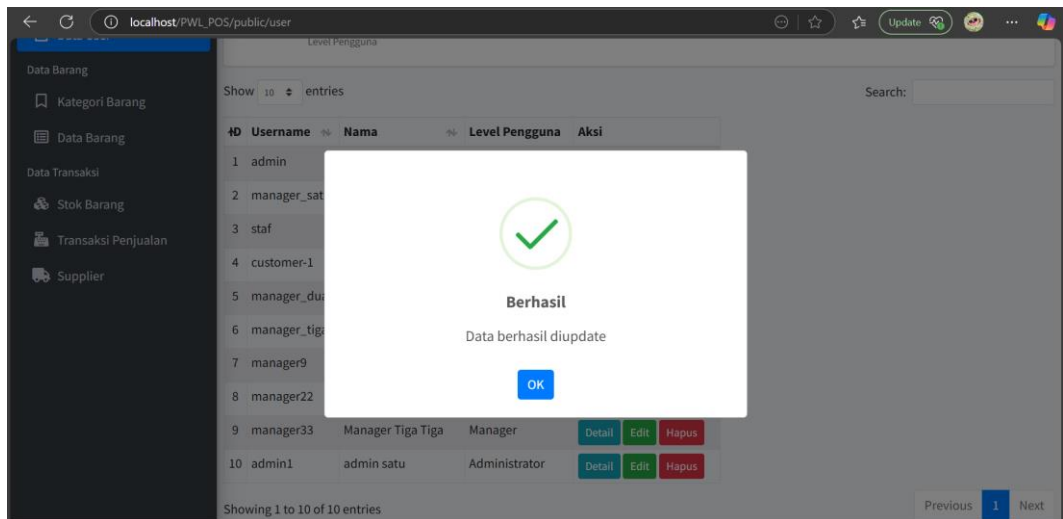
    return redirect('/');
}

```

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



Data user berhasil diubah



3. 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']); // Untuk tampilkan form confirm delete user Ajax
Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); // Untuk hapus 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`

```

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">
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> Kesalahan!!!</h5>
13                 Data yang Anda cari tidak ditemukan
14             </div>
15             <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
16         </div>
17     </div>
18 </div>
19 @else
20 <form action="{{ url('/user/' . $user->user_id . '/delete_ajax') }}" method="POST" id="form-delete">
21     @csrf
22     @method('DELETE')
23     <div id="modal-master" class="modal-dialog modal-lg" role="document">
24         <div class="modal-content">
25             <div class="modal-header">
26                 <h5 class="modal-title" id="exampleModallabel">Hapus Data User</h5>
27                 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
28                     <span aria-hidden="true">&times;</span>
29                 </button>
30             </div>
31             <div class="modal-body">
32                 <div class="alert alert-warning">
33                     <h5><i class="icon fas fa-ban"></i> Konfirmasi !!!</h5>
34                     Apakah Anda ingin menghapus data seperti di bawah ini?
35                 </div>
36                 <table class="table table-sm table-bordered table-striped">
37                     <tr>
38                         <th class="text-right col-3">Level Pengguna :</th>
39                         <td class="col-9">{{ $user->level->level_nama }}</td>
40                     </tr>
41                     <tr>
42                         <th class="text-right col-3">Username :</th>
43                         <td class="col-9">{{ $user->username }}</td>
44                     </tr>
45                     <tr>
46                         <th class="text-right col-3">Nama :</th>
47                         <td class="col-9">{{ $user->nama }}</td>
48                     </tr>
49                 </table>
50             </div>
51             <div class="modal-footer">
52                 <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
53                 <button type="submit" class="btn btn-success">Ya, Hapus</button>
54             </div>
55         </div>
56     </div>
57 </form>
58 <script>
59     $(document).ready(function() {
60         $("#form-delete").validate({
61             rules: {},
62             submitHandler: function(form) {
63                 $.ajax({
64                     url: form.action,
65                     type: form.method,
66                     data: $(form).serialize(),
67                     success: function(response) {
68                         if (response.status) {
69                             $('#myModal').modal('hide');
70                             Swal.fire({
71                                 icon: 'success',
72                                 title: 'Berhasil',
73                                 text: response.message
74                             });
75                             dataUser.ajax.reload();
76                         } else {
77                             $('.error-text').text('');
78                             $.each(response.msgField, function(prefix, val) {
79                                 $('#error-' + prefix).text(val[0]);
80                             });
81                         }
82                     }
83                 });
84             }
85         });
86     });
87 </script>

```

```

81     Swal.fire({
82         icon: 'error',
83         title: 'Terjadi Kesalahan',
84         text: response.message
85     });
86     }
87     }
88     });
89     return false;
90 },
91 errorElement: 'span',
92 errorPlacement: function(error, element) {
93     error.addClass('invalid-feedback');
94     element.closest('.form-group').append(error);
95 },
96 highlight: function(element) {
97     $(element).addClass('is-invalid');
98 },
99 unhighlight: function(element) {
100     $(element).removeClass('is-invalid');
101 }
102 });
103 </script>
104 @endempty

```

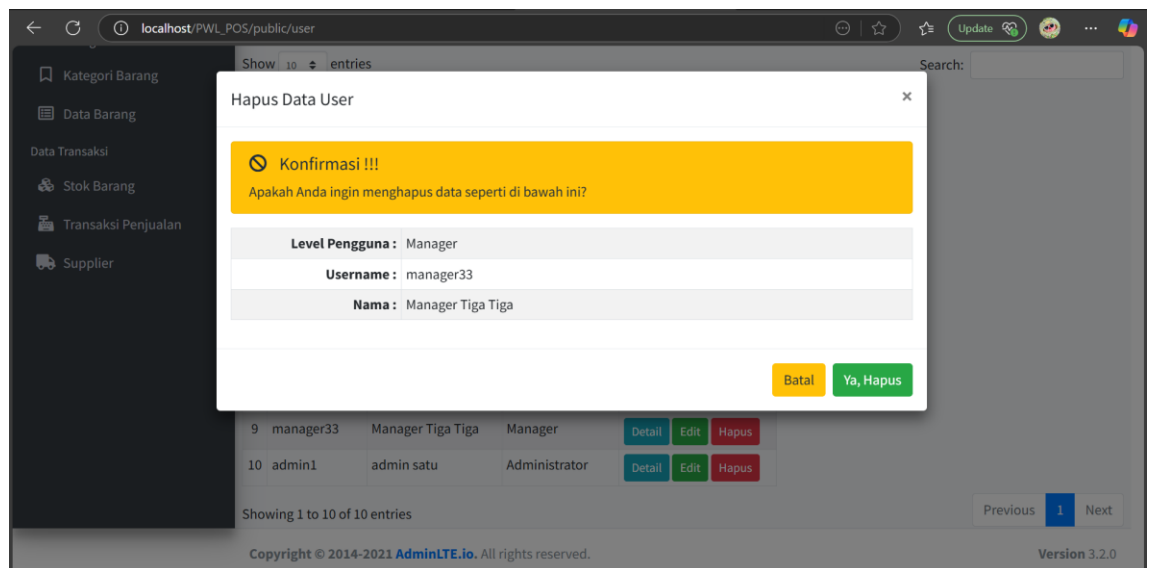
- e. Kemudian kita buat fungsi `delete_ajax()` pada `UserController.php` untuk mengakomodir request hapus data user

```

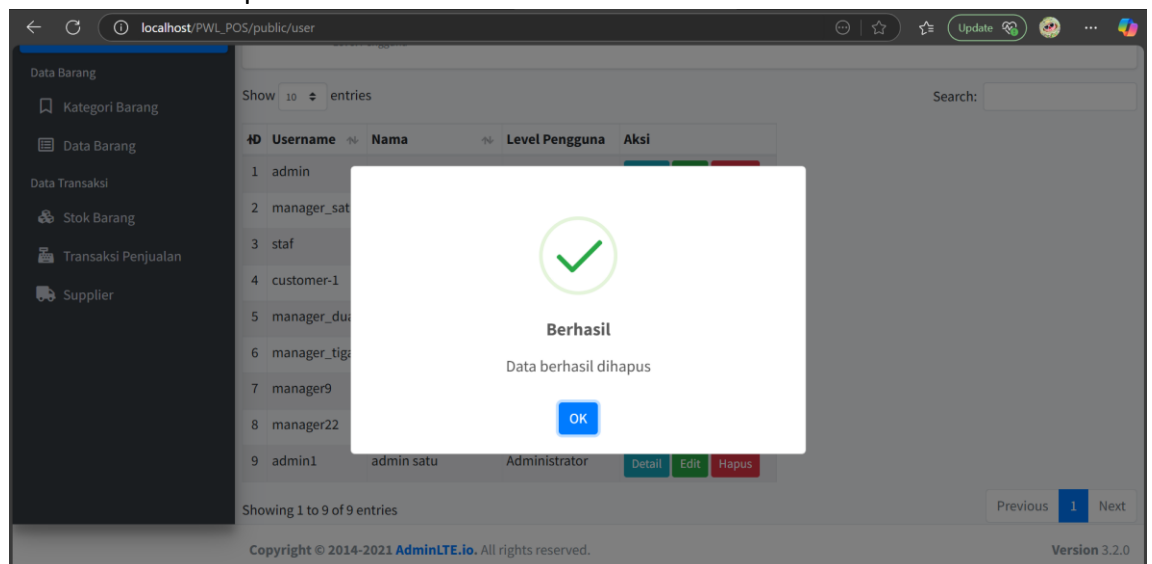
public function delete_ajax(Request $request, $id)
{
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantsJson()) {
        $user = UserModel::find($id);
        if ($user) {
            $user->delete();
            return response()->json([
                'status' => true,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

```

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



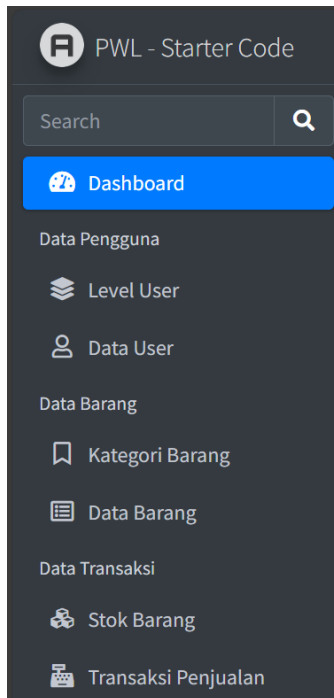
Data berhasil dihapus



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

TUGAS

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



✓ 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. The data is as follows:

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	Detail Edit Hapus
2	MNG	Manager	Detail Edit Hapus
3	STF	Staf/Kasir	Detail Edit Hapus
4	PLG	Pelanggan	Detail Edit Hapus

Additional page elements include a filter dropdown set to '- Semua -', a search bar, and pagination showing 'Showing 1 to 4 of 4 entries' with 'Previous', '1', and 'Next' links.

✓ Tabel m_kategori

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Tambah Tambah Ajax

Show 10 entries Search:

ID	Kode Kategori	Nama Kategori	Aksi
1	ELK	Elektronik	Detail Edit Hapus
2	FAS	Fashion	Detail Edit Hapus
3	MKN	Makanan	Detail Edit Hapus
4	ATK	Alat Tulis	Detail Edit Hapus
5	SPRT	Sport	Detail Edit Hapus
6	CML	Cemilan	Detail Edit Hapus
7	MNR	Minuman Ringan	Detail Edit Hapus

✓ Tabel m_supplier

Daftar Supplier

Daftar supplier yang terdaftar dalam sistem

Tambah Tambah Ajax

Show 10 entries Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP1	Toko Elektronik Jaya	JL. Merdeka No. 45, Bandung	Detail Edit Hapus
2	SP2	Sentral Komputer	JL. Ahmad Yani No. 10, Semarang	Detail Edit Hapus
3	SP3	Toko Grosir Makmur	JL. Raya Sudirman No. 99, Yogyakarta	Detail Edit Hapus

Showing 1 to 3 of 3 entries Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

✓ Tabel m_barang

Daftar barang yang terdaftar dalam sistem

Filter: Kategori Barang

Show entries Search:

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
1	HP001	Handphone	Elektronik	3000000	3500000	Detail Edit Hapus
2	SW001	Sweater	Fashion	150000	200000	Detail Edit Hapus
3	JNS001	Jeans	Fashion	200000	250000	Detail Edit Hapus
4	MSG001	Mie Instan	Makanan	2500	3500	Detail Edit Hapus
5	SUS001	Susu Kotak	Makanan	5000	7000	Detail Edit Hapus

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