# Laporan Praktikum Pemrograman Web Lanjut

Jobsheet 6: Ajax Form (AdminLTE) dan Client Validation .



# Oleh:

Danica Nasywa Putriniar (2341760122) Kelas SIB 2B / 05

# 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

#### A. 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.

3. Selanjutnya kita modifikasi view user/index.blade.php

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

```
cript>
  function modalAction(url){
      $('#myModal').load(url, function(){
          $('#myModal').modal('show');
      });
 var dataUser;
 $(document).ready(function () {
      dataUser = $('#table_user').DataTable({
          // serverSide: true, jika ingi menggunakan server
          serverSide: true,
          ajax: {
              url: "{{ url('user/list') }}",
              "dataType": "json",
              "type": "POST",
              "data": function (d) {
                  d.level_id = $('#level_id').val();
```

6. Selanjutnya kita modifikasi route/web.php untuk mengakomodir oprasi ajax

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

7. Kemudian kita tambahkan fungsi create\_ajax() pada file UserController.php

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

```
resources > views > user > 🦏 create_ajax.blade.php > ᢒ form#form-tambah
      <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-tittle" id="exampleModalLabel">Tambah Data User</h5>
                   <button type="button" class="close" data-dismiss="modal" aria-label="Close"><sr</pre>
               <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 $1)
                               <option value=" {{ 1->level_id }}">{{ <math>1->level_nama }}</option>
                       <small id="error-level id" class="error-text form-text text-danger"></small</pre>
                   <div class="form-group">
                       <label>Username</label>
                       <input value="" type="text" name="username" id="username" class="form-contr</pre>
                       <small id="error-username" class="error-text form-text text-danger"></small</pre>
                   <div class="form-group">
                       <label>Nama</label>
                       <input value="" type="text" name="nama" id="nama" class="form-control" requ</pre>
                       <small id="error-nama" class="error-text form-text text-danger"></small>
                   <div class="form-group">
                       <label>Password</label>
                       <input value="" type="password" name="password" id="password" class="form-</pre>
                       <small id="error-password" class="error-text form-text text-danger"></small
</pre>
               <div class="modal footer">
                   <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button</pre>
                   <button type="submit" class="btn btn-primary">Simpan</button>
42
       (script)
           $(document).ready(function(){
```

9. Kemudian untuk mengakomodir proses simpan data melalui ajax, kita buat fungsi

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

#### 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

```
public function list(Request $request)
    $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
        ->with('level');
    if ($request->level id) {
        $users->where('level_id', $request->level_id);
    return DataTables::of($users)
        ->addIndexColumn()
        ->addColumn('aksi', function ($user) {
            $btn = '<button onclick="modalAction(\''.url('/user/' . $user->user_id . '/s
            $btn .= '<button onclick="modalAction(\''.url('/user/' . $user->user_id .
            $btn .= '<button onclick="modalAction(\''.url('/user/' . $user->user_id .
            // $btn .= '<form class="d-inline-block" method="POST" action="'.url('/user,
                   '<button type="submit" class="btn btn-danger btn-sm" onclick="return
            return $btn;
        ->rawColumns(['aksi'])
        ->make(true);
```

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

```
Route::put( /{id} , [UserController::class, update ]); // menyimpan per
Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); // men
Route::put('/{id}upcate_ajax', [UserController::class, 'update_ajax']);
```

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

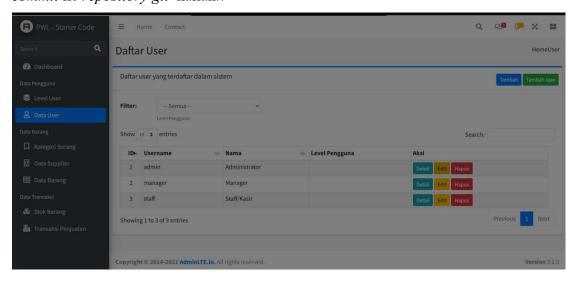
5. Kita buat view baru pada user.edit\_ajax.blade.php untuk menampilkan form view ajax

```
@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"><</pre>
                        aria-hidden="true">×</span></button>
            <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
                <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
@else
<form action="{{ url('/user/' . $user->user_id . '/update_ajax') }}" method="POST" id="form-e
    @method('PUT')
    <div id="modal-master" class="modal-dialog modal-lg" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Edit Data User</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
            <div class="modal-body">
                <div class="form-group">
                    <label>Level Pengguna</label>
                    <select name="level_id" id="level_id" class="form-control" required>
                        <option value="">- Pilih Level -</option>
                        @foreach($level as $1)
                        <option {{ ($1->level_id == $user->level_id) ? 'selected' : '' }}
                           value="{{ $1->level_id }}">{{ $1->level_nama }}
                        @endforeach
                    <small id="error-level_id" class="error-text form-text textdanger"></small
</pre>
                <div class="form-group">
```

6. Selanjurnya, kita buat fungsi update\_ajax() pada UserController.php untuk mengakomodir request update data user malalui ajax

```
public function update_ajax(Request $request, $id){
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'level_id' => 'required|integer',
            'username' => 'required|max:20|unique:m_user,username,'.$id.',user_id',
            'nama' => 'required|max:100',
        ];
        $validator = Validator::make($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
                 1); }
                 $check = UserModel::find($id);
                 if ($check) { if(!$request->filled('password') ){ // jika password tidak diis
                     $request->request->remove('password');
                 $check->update($request->all());
                 return response()->json([
                     'status' => true,
'message' => 'Data berhasil diupdate'
            else{ return response()->json([
    return redirect('/');
```

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



Hasilnya error

#### 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 untuk menghapus data

```
Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); // menampilkan
Route::put('/{id}delete_ajax', [UserController::class, 'delete_ajax']); // menyimpan
```

3. Kemudian kuta buat fungsi confirm ajax() pada UserController.php

```
public function confirm_ajax(String $id){
    $user = UserModel::find($id);

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

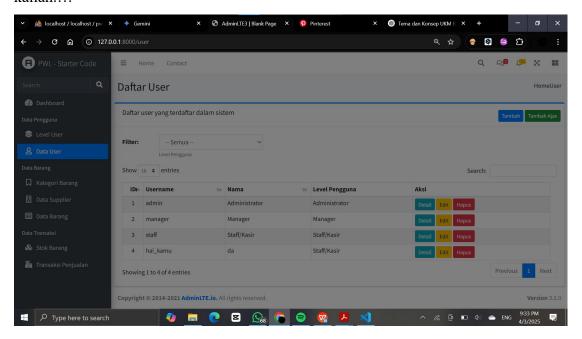
4. Selanjutnya kita review untuk konfirmasi hapus data dengan nama user/confirm ajax.blade.php

```
> views > 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>
                <div class="alert alert-danger">
                    <h5><i class="icon fas fa-ban"></i> Kesalahan!!!</h5>
                    Data yang anda cari tidak ditemukan. <a href="{{ url('/user') }}" class="
@else
    <form action="{{ url('/user/' . $user->user_id . '/delete_ajax') }}" method="POST" id="fo
        @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="Clos</pre>
                        <span aria-hidden="true">&times;</span>
                <div class="modal-body">
                        <h5><i class="icon fas fa-exclamation-triangle"></i> Konfirmasi !!!</
                        Apakah Anda ingin menghapus data seperti di bawah ini?
```

5. Kemudian kita buat fungsi delete\_ajax() pada UserController.php untuk mengakomodir *request* hapus data user.

```
public function delete_ajax(Request $request, $id)
   //cek apakah rewuest dari ajax
    if ($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('/');
```

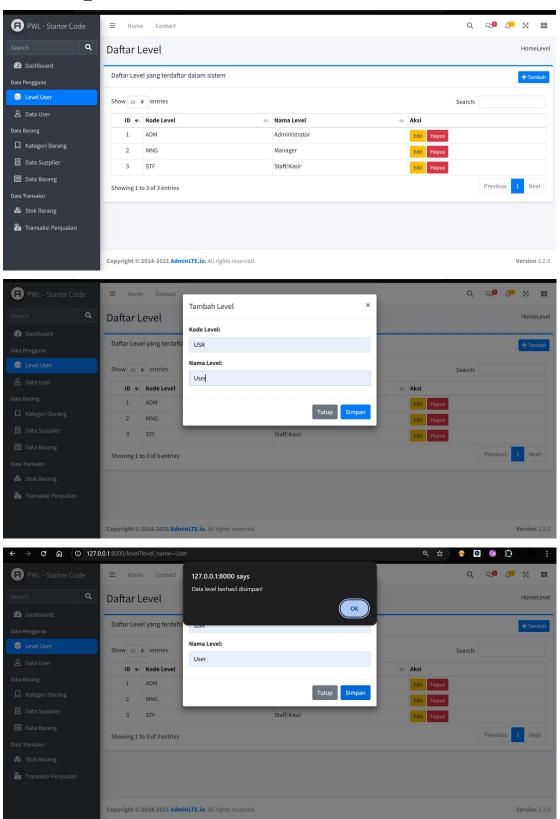
- 6. Setelah semua selesai, mari kita coba untuk melakukan percobaan dari koding yang telah kita lakukan
- 7. Jangan lupa laporkan ke laporan *jobsheet* dan lakukan *commit* pada *repository git* kalian.!!!

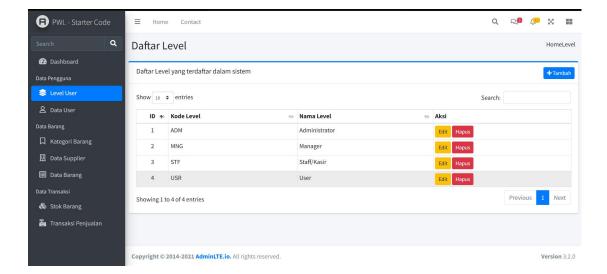


# **Tugas jobsheet**

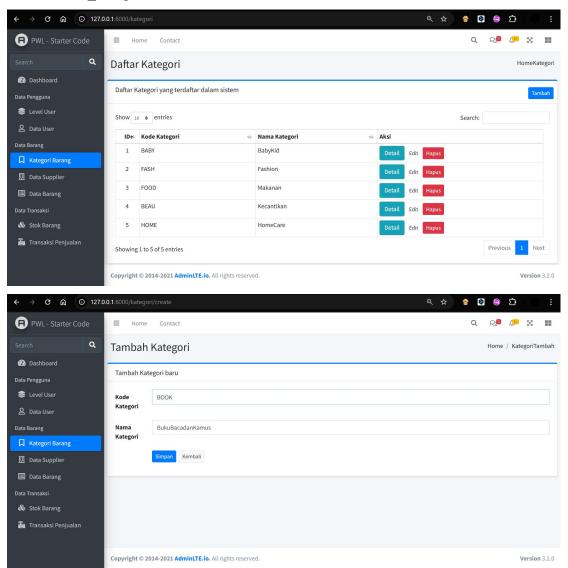
Implementasikan koding untuk ajax form dan client validation dengan jQuery Validation pada menu berikut ini

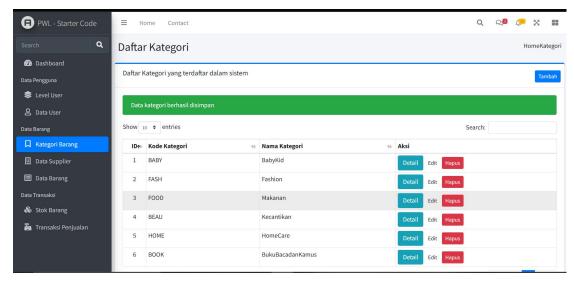
• Tabel m level



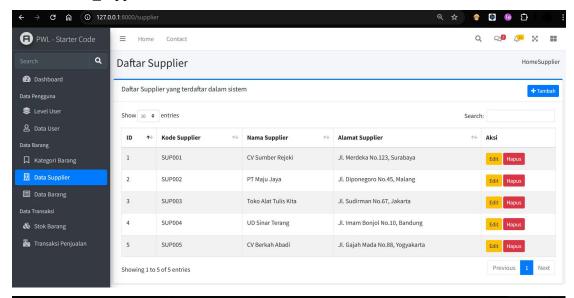


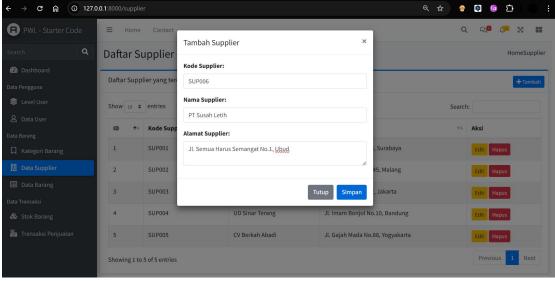
# Tabel m\_kategori

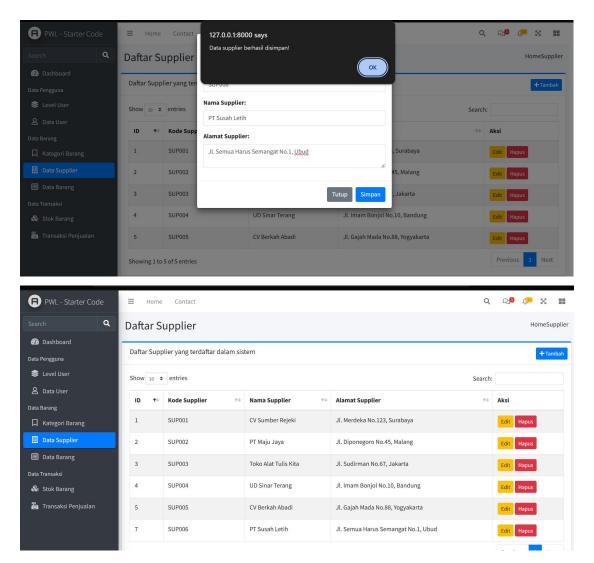




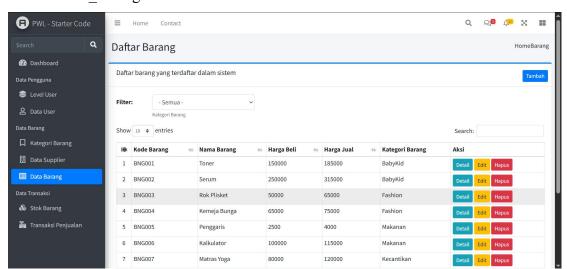
# • Tabel m supplier

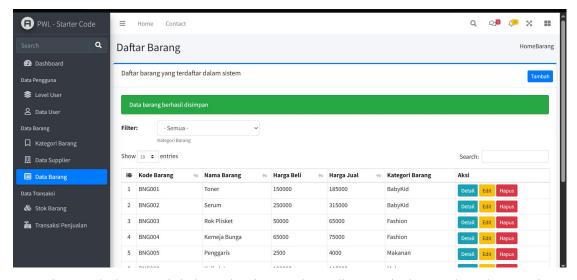






# • Tabel m barang





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