

## **Tugas Pertemuan 6 Pemrograman Web Lanjut**



**Muhammad Farrel Caesarian**  
**TI-2D/16**  
**2341720053**  
**POLITEKNIK NEGERI MALANG**

## 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

```
<!-- Sweet Alert 2 -->
<link rel="stylesheet" href="{{ asset('adminlte/plugins/sweetalert2-theme-bootstrap-4/bootstrap4.min.css') }}">
<!-- Theme style -->
```

```
<!-- 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>
```

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

```
<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>
```

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`

```
<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: [

```

Selanjutnya Kita modifikasi route/web.php untuk mengakomodir 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
Route::post('/ajax', [UserController::class, 'destroy_ajax']); // menghapus data user Ajax
```

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);
}
```

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

```
resources > views > user > create_ajax.blade.php > script > <function>
1
2 <form action="{{ url('user/ajax') }}" method="POST" id="form-tambah">
3     @csrf
4     <div id="modal-master" class="modal-dialog modal-lg" role="document">
5         <div class="modal-content">
6             <div class="modal-header">
7                 <h5 class="modal-title" id="exampleModallabel">Tambah
8                 <button type="button" class="close" data-dismiss="modal"
9                     aria-hidden="true">&times;</span></button>
10             </div>
11             <div class="modal-body">
```

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

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

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

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'
        ];

        $validator = Validator::make($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('/');
}
```

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!!!

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Tes Edit	Administrator	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	manager	Manager	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	staff	Staff/Kasir	Staff/Kasir	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	customer 1	Pelanggan Pertama	Pelanggan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	manager_dikuk	Manager 2	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	manager22	Manager Dua Dua	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	manager33	Manager Tiga Tiga	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	manager12	Manager11	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	prince_casual_	Fanrel Casuaran	Administrator	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Tambah Data User ✕

Level Pengguna

Administrator ▼

Username

cadsewq

Nama

123434

Password

\*\*\*\*\*

Batal

Simpan

Show 10 of 11 entries

Search

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Ten Edit	Administrator	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	manager	Manager	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	staff	StaffKaur	StaffKaur	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	customer1	Pelanggan Pertama		<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	manager_dua	Manager 2		<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	manager22	Manager Dua Dua		<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	manager33	Manager Tiga Tiga		<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	manager112	Manager 11		<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	petrol-caesar...	Farrel Caesarian		<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
10	qweert	lulu		<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 10 of 11 entries

Previous

1

2

Next

Praktikum 1

FarrelCaesarian • just now

- ## Praktikum 2. Modal Ajax Edit Data (Data User)

Pada Praktikum 2 ini, kita akan melakukan koding untuk proses edit menggunakan ajax

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');

    // Filter data user berdasarkan 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) { // menambahkan kolom aksi
            // $btn = '<a href="' . url('/user/' . $user->user_id) . '" class="btn btn-info">Detail</button> ';
            // $btn .= '<a href="' . url('/user/' . $user->user_id . '/edit') . '" class="btn btn-warning">Edit</button> ';
            // $btn .= '<form class="d-inline-block" method="POST" action="' . url('/user/' . $user->user_id . '/delete') . '">';
            // $btn .= '<input type="submit" value="Hapus" class="btn btn-danger btn-sm"></form>';
            // JB 6
            $btn = '<button onclick="modalAction(\'' . url('/user/' . $user->user_id . '/show_ajax') . '\')"' class="btn btn-info btn-sm">Detail</button> ';
            $btn .= '<button onclick="modalAction(\'' . url('/user/' . $user->user_id . '/edit_ajax') . '\')"' class="btn btn-warning btn-sm">Edit</button> ';
            $btn .= '<button onclick="modalAction(\'' . url('/user/' . $user->user_id . '/delete_ajax') . '\')"' class="btn btn-danger btn-sm">Hapus</button> ';
            return $btn;
        })
        ->rawColumns(['aksi']) // Memberitahu bahwa kolom aksi berisi HTML
        ->make(true);
}

```

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
```

Kemudian, kita buat fungsi edit\_ajax() pada UserController.php

```
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]);
}
```

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

```
resources > views > user > edit_ajax.blade.php > ...
1 @empty($user)
2 <div id="modal-master" class="modal-dialog modal-lg" role="document">
3     <div class="modal-content">
4         <div class="modal-header">
5             <h5 class="modal-title" id="exampleModalLabel">Kesalahan
6             <button type="button" class="close" data-dismiss="modal">
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
```

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

```
public function update_ajax(Request $request, $id)
{
    // Mengecek apakah request dari ajax
    if ($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('/');
}
```

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

The screenshot displays a web application interface. At the top, there is a modal titled 'Edit Data User' with a close button (X). The form contains the following fields:

- Level Pengguna:** A dropdown menu currently showing 'Manager'.
- Username:** A text input field containing 'tes edit'.
- Nama:** A text input field containing 'tes edit'.
- Password:** A password input field with masked characters '\*\*\*\*\*'.

Below the password field, there is a note: 'Abaikan jika tidak ingin ubah password'. At the bottom right of the form are two buttons: 'Batal' (Cancel) and 'Simpan' (Save).

Below the form, a table of users is visible. A modal overlay is present in the center of the screen, showing a green checkmark and the text 'Berhasil' (Successful), followed by 'Data berhasil diupdate' (Data successfully updated) and an 'OK' button. The table in the background lists users with columns for ID, Username, Name, Level, and Action buttons (Edit, Delete, Password).

### 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']); // menampilkan halaman form Delete user Ajax
Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); // menghapus 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`

```
resources > views > 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
6             <button type="button" class="close" data-dismiss="modal">
7         </div>
8     </div>
9     <div class="modal-body">
10        <div class="alert alert-danger">
11            <h5><i class="fas fa-ban"></i> Kesalahan!!!</h5>
12            Data yang anda cari tidak ditemukan</div>
13        <a href="{{ url('/user') }}" class="btn btn-warning">K
14    </div>

```

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

```
public function delete_ajax(request $request, $id)
{
    if ($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('/');
}

```

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

Show  entries

ID	Username	Nama
1	admin	Tes Edit
2	manager	Manager
3	staff	Staff/Kasir
4	customer-1	Pelanggan Pertama
5	manager_dua	Manager 2
6	manager22	Manager Dua Dua
7	manager33	Manager Tiga Tiga
8	manager12	Manager11
9	prince.caesar_	Farrel Caesarian
10	tes edit 2	tes edit 2

Showing 1 to 10 of 12 entries

### Hapus Data User

#### Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Level Pengguna :	Manager
Username :	tes edit 2
Nama :	tes edit 2

Batal

Ya, Hapus



**Berhasil**

Data berhasil dihapus

OK

### Hapus Data User

#### Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Level Pengguna :	Administrator
Username :	cadsewq
Nama :	123434

Batal

Ya, Hapus

Show 10 entries

ID	Username
1	admin
2	manager
3	staff
4	customer-1
5	manager_dua
6	manager22
7	manager33
8	manager12
9	prince.caesar_
10	asdsad

Showing 1 to 10 of 10 entries

### Praktikum 3

FarrelCaesarian • just now





4. **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.

5. **Jawab:**  
**Level**
- Tambah Data Level

Kode Level

Tes Level

Nama Level

Tambah Ajax

Batal

Simpan

5

Tes Level

Tambah Ajax

Detail

Edit

Hapus
- Edit Data Level

Kode Level

Tes Level

Nama Level

Edit Ajax

Batal

Simpan

5

Tes Level

Edit Ajax

Detail

Edit

Hapus
- Hapus Data Level

Konfirmasi!!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Kode Level: Tes Level

Nama Level: Edit Ajax

Batal

Ya, Hapus

ID	↕	Kode Level
1		ADM
2		MNG
3		STF
4		CUS

# Kategori

Tambah Data Kategori

Kode Kategori

Tes Ajax

Nama Kategori

Tambah Ajax

Batal

Simpan

9	Tes Ajax	Tambah Ajax	<div>Simpan</div> <div>Hapus</div>
---	----------	-------------	------------------------------------

Edit Data Kategori

Kategori Kode

Tes Ajax

Kategori Nama

Tambah Ajax Edit

Batal

Simpan

9	Tes Ajax	Tambah Ajax Edit	<div>Simpan</div> <div>Hapus</div>
---	----------	------------------	------------------------------------

Hapus Data Kategori

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Supplier Kode:

Tes Ajax

Supplier Nama:

Tambah Ajax Edit

Batal

Ya, Hapus

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

# Supplier

Tambah Data Supplier

Supplier Kode

TesAjax

Supplier Nama

Tes Tambah

Supplier Alamat

Tes Ajax

Batal

Simpan

TesAjax

Tes Edit

Supplier Kode

TesAjax

Supplier Nama

Tes Edit

Supplier Alamat

Tes Ajax

Batal

Simpan

4	TesAjax	Tes Edit
---	---------	----------

Hapus Data Supplier

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Supplier Kode:	TesAjax
Supplier Nama:	Tes Edit
Supplier Alamat:	Tes Ajax

Batal

Ya, Hapus

ID	Kode
1	SUP0
2	SUP0
3	SUP0

Showing 1 to 3 of 3 entries

# Barang

Tambah Data Barang

Kategori

Makanan Berat

Kode Barang

B001

Nama Barang

Tes Ajax

Harga Beli

12321321

Harga Jual

12312332

Batal

Simpan

No	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
11	B001	Tes Ajax	Makanan Berat	12321321	12312332	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 11 to 11 of 11 entries

Previous1Next

Edit Data Barang

Kategori

Makanan Berat

Kode Barang

B001

Nama Barang

Tes Ajax Edit

Harga Beli

12321321

Harga Jual

12312332

Batal

Simpan

No	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
11	B001	Tes Ajax Edit	Makanan Berat	12321321	12312332	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Hapus Data Barang

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Barang Kode:

B001

Barang Nama:

Tes Ajax Edit

Harga Beli:

12321321

Harga Jual:

12312332

Kategori:

Makanan Berat

Batal

Ya, Hapus

