TUGAS

Analisis Desain

PERTEMUAN 6

JS 6 Ajax Form

OLEH

MUHAMMAD NUR AZIZ NIM. 2341720237

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

20 Maret 2025

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.

```
{{-- SweetAlert2 --}}
<link rel="stylesheet" href="{{ asset('adminlte/plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
```

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

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

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

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>
endsection
```

5. Kemudian kita tambahkan kode berikut pada awal @push(‘js’) pada view user/index.blade.php

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({
```

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

```
Route::get('/create_ajax', [UserController::class, 'create_ajax']); // Menampilkan hala
Route::post('/ajax', [UserController::class, 'store_ajax']); // Menyimpan data user bar
```

7. Kemudian Kita tambahkan fungsi create_ajax() pada file UserController.php

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

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

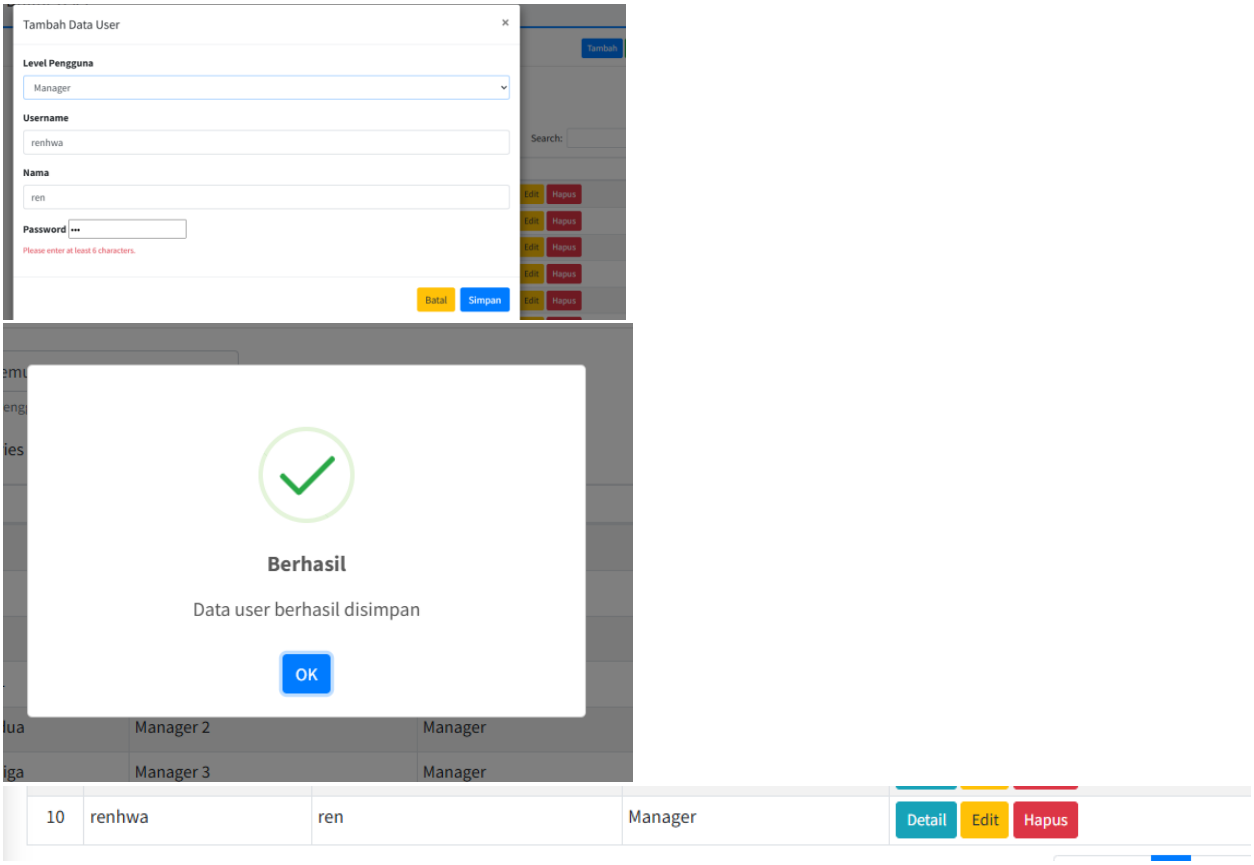
8. Setelah itu, kita buat view baru user/create_ajax.blade.php untuk menampilkan form dengan ajax

```
Lanjut > Week6 > JS6 > resources > views > user > create_ajax.blade.php > 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-title" id="exampleModalLabel">Tambah Data User</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close"
                        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"></sm
                </div>
                <div class="form-group">
                    <label>Username</label>
```

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

```
Codeium: Refactor | Explain | Generate Function Comment | X
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: ber
                'message' => 'Validasi Gagal',
                'msgField' => $validator->errors() // pesan error validasi
            ]);
        }

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

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

10. OK, sekarang kita coba melakukan proses tambah data user menggunakan form ajax. Amati apa yang terjadi dan laporankan pada laporan jobsheet dan commit di github kalian!!!



Ketika menambah user baru menggunakan ajax ver, tidak perlu memuat halaman baru. User dapat dibuat langsung saat itu juga dengan popup dari ajax. Dan Ketika user tidak memilih level atau tidak mengisi form user tidak akan bisa dibuat

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

```
// 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 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('/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 adalah html
->make(true);
```

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

```php
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
```

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

```php
Codeium: Refactor | Explain | X
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]);
}
```

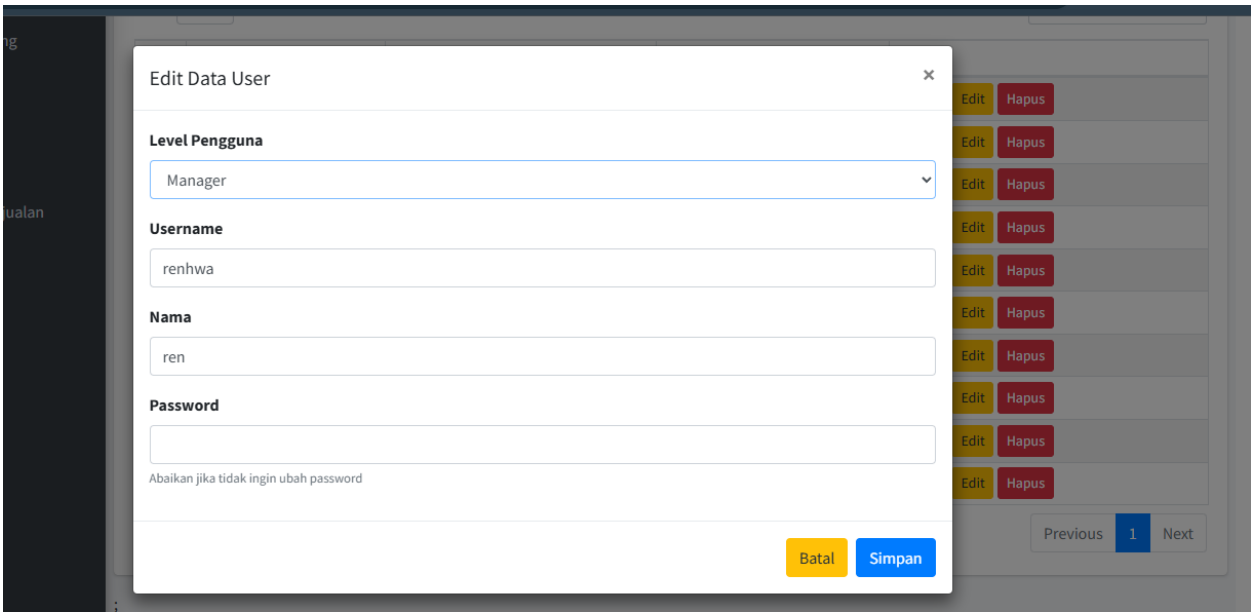
5. Kita buat view baru pada user/edit_ajax.blade.php untuk menampilkan form view ajax

```
bLanjut > Week6 > JS6 > resources > views > user > edit_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>
@else
    <form action="{{ url('/user/' . $user->user_id . '/update_ajax') }}" method="POST" id="form-edit">
        @csrf
```

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

```php
Codeium: Refactor | Explain | Generate Function Comment | X
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
            ]);
        }
```

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



Ketika mengedit data user tidak perlu memuat halaman edit lagi, bisa langsung melalui popup

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::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

```
Codeium: Refactor | Explain | Generate Function Comment | X
public function confirm_ajax(string $id)
{
    $user = UserModel::find($id);

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

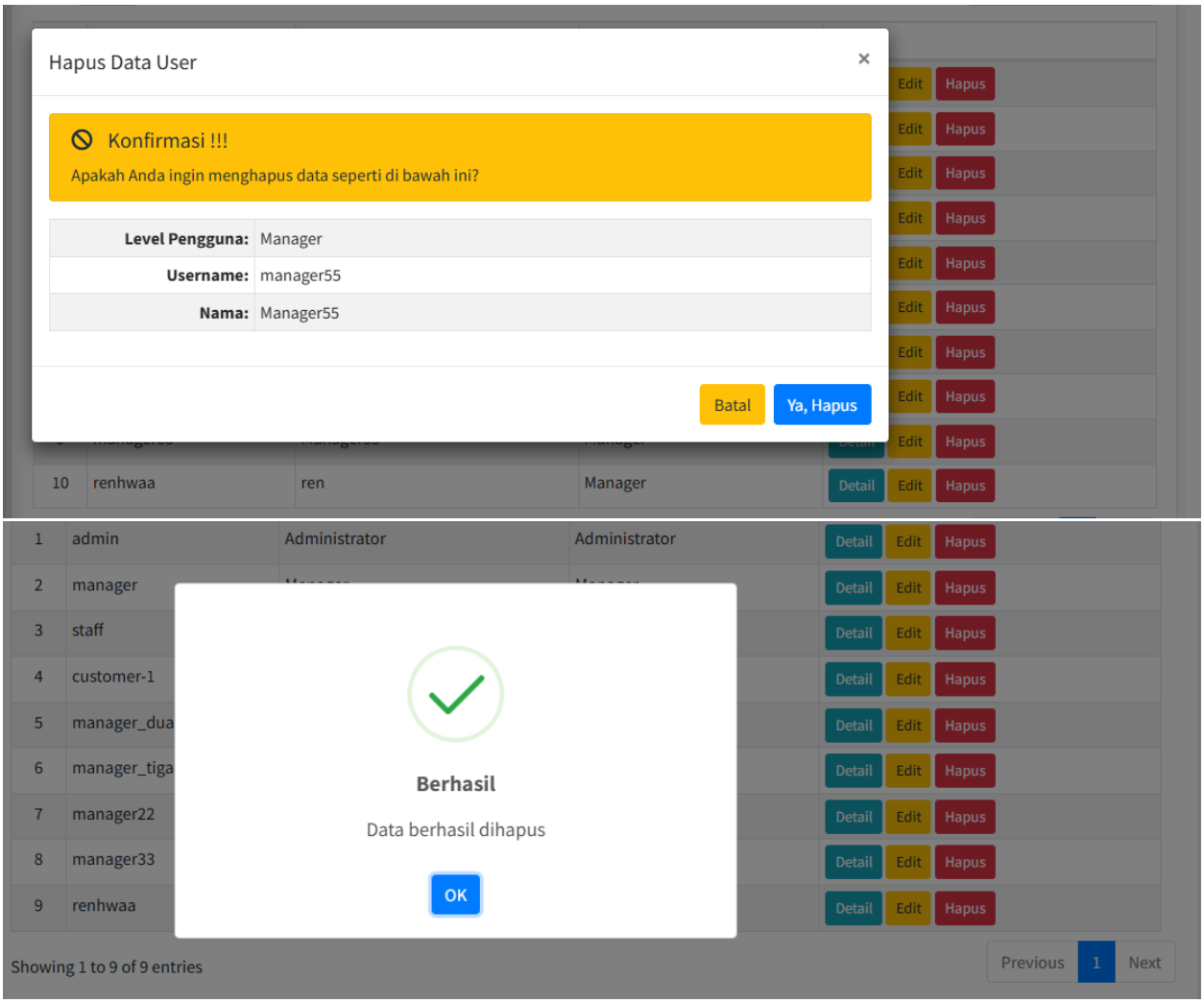
4. Selanjutnya kita view untuk konfirmasi hapus data dengan nama user/confirm_ajax.blade.php

```
ebLanjut > Week6 > JS6 > resources > 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>
                </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 . '/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">
```

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

```
Codeium: Refactor | Explain | Generate Function Comment | ×
public function delete_ajax(Request $request, $id)
{
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantsJson()) {
        $user = UserModel::find($id);
        if ($user) {
            try {
                $user->delete();
                return response()->json([
                    'status' => true,
                    'message' => 'Data berhasil dihapus'
                ]);
            } catch (\Illuminate\Database\QueryException $e) {
                return response()->json([
                    'status' => false,
                    'message' => 'Data tidak bisa 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.



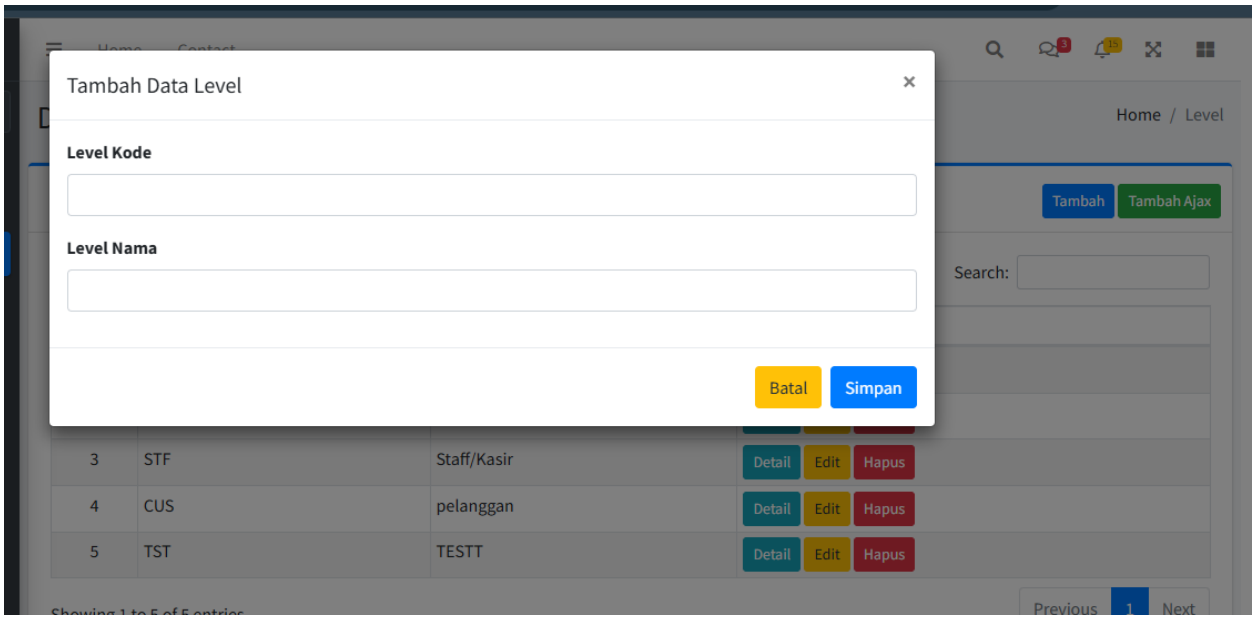
Data berhasil dihapus melalui popup ajax

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

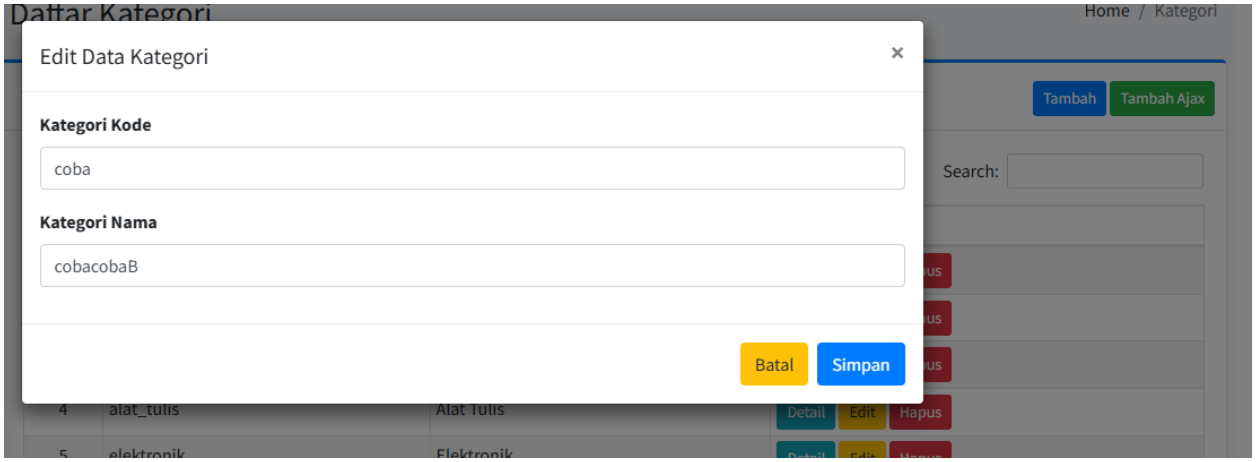
TUGAS

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

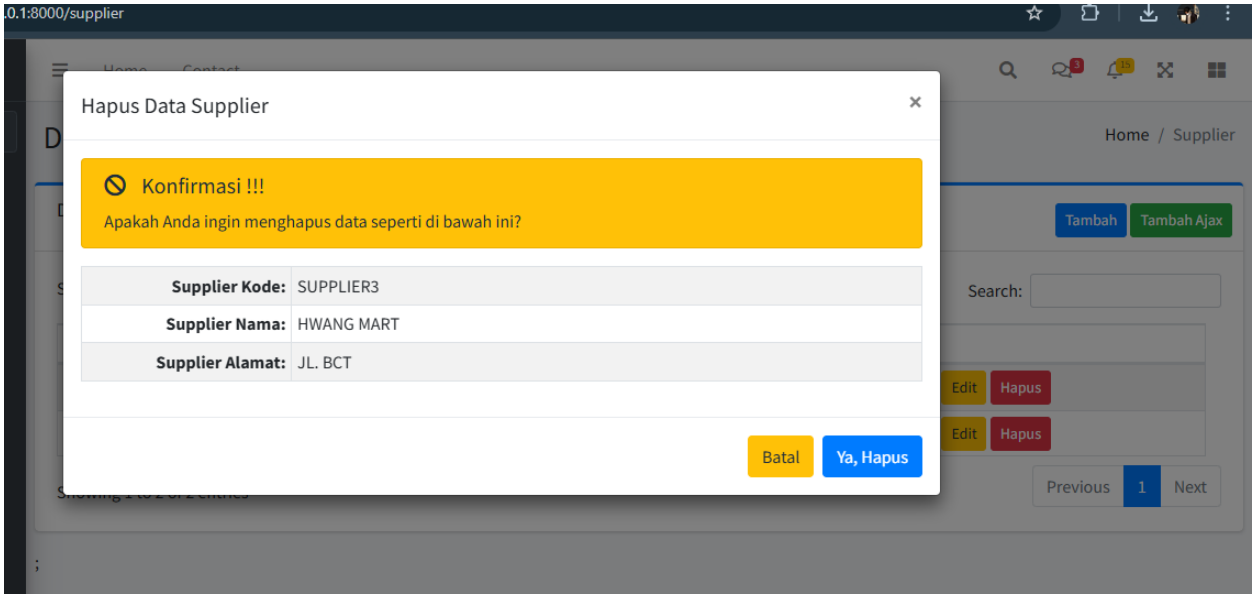
✓ Tabel m_level



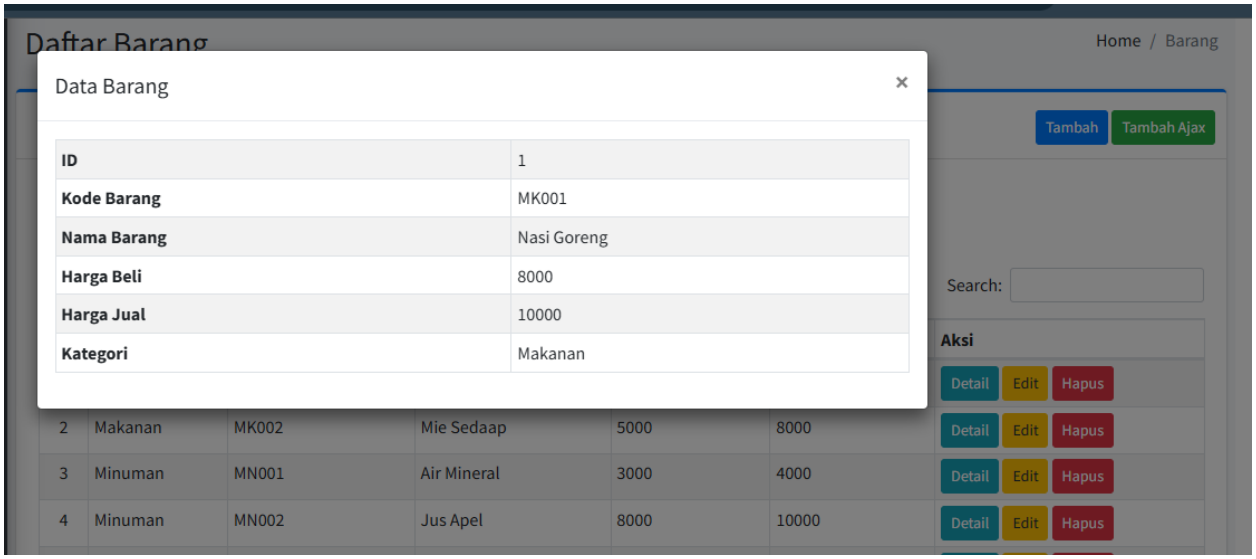
✓ Tabel m_kategori



✓ Tabel m_supplier



✓ Tabel m_barang



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