

Nama : Adinda Ivanka Maysanda Putri

Kelas : SIB 2B

NIM : 2341760058

JOBSHEET 6

Ajax Form (AdminLTE) dan Client Validation

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') }}">
<!-- Theme style -->
<link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
```

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

```
<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>
```

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>
```

5. Kemudian kita tambahkan kode berikut pada awal @push('js') pada view user/index.blade.php Sehingga tampilan kode program akan seperti berikut

```
function modalAction(url = "") {
    $('#myModal').load(url, function () {
        $('#myModal').modal('show');
    });
}

var dataUser;
$(document).ready(function () {
    dataUser = $('#table_user').DataTable({
```

6. 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
```

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

```

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

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

```

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

```

<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"><span
                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"></small>
            </div>
            <div class="form-group">
                <label>Username</label>
                <input value="" type="text" name="username" id="username" class="form-control" required>
                <small id="error-username" class="error-text form-text text-danger"></small>
            </div>
            <div class="form-group">
                <label>Nama</label>
                <input value="" type="text" name="nama" id="nama" class="form-control" required>
                <small id="error-nama" class="error-text form-text text-danger"></small>
            </div>
            <div class="form-group">
                <label>Password</label>
                <input value="" type="password" name="password" id="password" class="formcontrol" required>
                <small id="error-password" class="error-text form-text text-danger"></small>
            </div>
        </div>
        <div class="modal-footer">
            <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
            <button type="submit" class="btn btn-primary">Simpan</button>
        </div>
    </div>
</div>

```

```

</form>
<script>
$(document).ready(function () {
    $("#form-tambah").validate({
        rules: {
            level_id: { required: true, number: true },
            username: { required: true, minlength: 3, maxlength: 20 }, nama: { required: true, minlength: 3, maxlength: 100 }, password:
{ required: true, minlength: 6, maxlength: 20 }
        },
        submitHandler: function (form) {
            $.ajax({
                url: form.action, type: form.method, data: $(form).serialize(), success: function (response) {
                    if (response.status) {
                        $('#myModal').modal('hide'); Swal.fire({
                            icon: 'success', title: 'Berhasil', text: response.message
                        });
                        dataUser.ajax.reload();
                    } else {
                        $('.error-text').text("");
                        $.each(response.msgField, function (prefix, val) {
                            $('#error-' + prefix).text(val[0]);
                        });
                        Swal.fire({
                            icon: 'error',
                            title: 'Terjadi Kesalahan', text: response.message
                        });
                    }
                }
            });
            return false;
        },
        errorElement: 'span',
        errorPlacement: function (error, element) {
            error.addClass('invalid-feedback');
            element.closest('.form-group').append(error);
        }, highlight: function (element, errorClass, validClass) {
            $(element).addClass('is-invalid');
        }, unhighlight: function (element, errorClass, validClass) {
            $(element).removeClass('is-invalid');
        }
    });
});
</script>

```

9. 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->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);

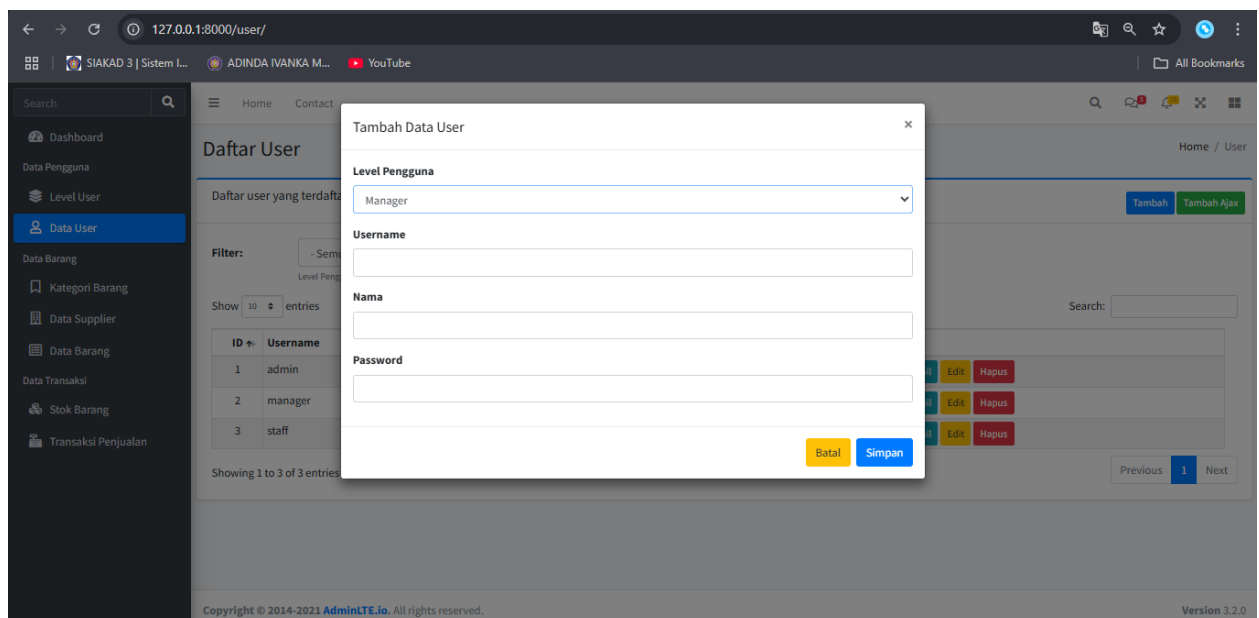
    if ($validator->fails()) {
        return response()->json([
            'status' => false, // response status, false: error/gagal, true: berhasil
            'message' => 'Validasi Gagal',
            'msgField' => $validator->errors(), // pesan error validasi
        ]);
    }

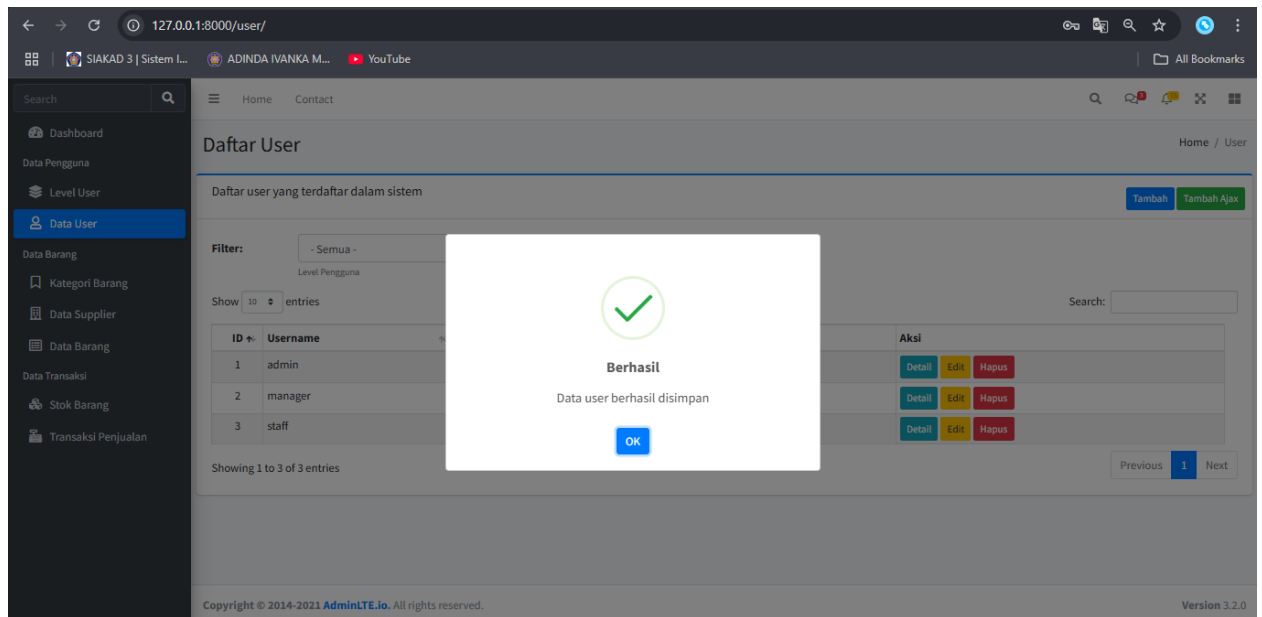
    UserModel::create($request->all());
    return response()->json([
        'status' => true,
        'message' => 'Data user berhasil disimpan'
    ]);
}

redirect('/');
}

```

10. 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!!!





Ketika tombol "Tambah Ajax" diaktifkan, sebuah modal formulir berbasis Bootstrap akan ditampilkan secara dinamis tanpa memerlukan pemuatan ulang halaman. Formulir ini menyediakan kolom input untuk Level Pengguna, Username, Nama, dan Password, serta dilengkapi dengan mekanisme validasi otomatis di sisi pengguna menggunakan pustaka jQuery Validation untuk menampilkan pesan kesalahan jika terdeteksi adanya input yang tidak valid. Apabila seluruh data pada formulir telah memenuhi kriteria validasi, pengiriman data akan dilakukan secara asinkron melalui mekanisme Ajax. Setelah proses penyimpanan data berhasil, modal formulir akan tertutup secara otomatis, notifikasi keberhasilan akan ditampilkan menggunakan SweetAlert, dan tabel data pengguna akan diperbarui secara instan tanpa memerlukan penyegaran halaman secara manual.

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

```
// 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($users)
        ->addIndexColumn() // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
        ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
```

```

        $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

```

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

```

// Menampilkan halaman form edit user ajax
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

```

@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
    @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>
  </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 $l)
        <option {{ ($l->level_id == $user->level_id)? 'selected' : '' }} value="{{ $l->level_id }}">{{ $l->level_nama
      }}</option>
      @endforeach
    </select>
    <small id="error-level_id" class="error-text form-text text-danger"></small>
  </div>
  <div class="form-group">
    <label>Username</label>
    <input value="{{ $user->username }}" type="text" name="username" id="username" class="form-control" required>
    <small id="error-username" class="error-text form-text text-danger"></small>
  </div>
  <div class="form-group">
    <label>Nama</label>
    <input value="{{ $user->nama }}" type="text" name="nama" id="nama" class="form-control" required>
    <small id="error-nama" class="error-text form-text text-danger"></small>
  </div>
  <div class="form-group">
    <label>Password</label>
    <input value="" type="password" name="password" id="password" class="form-control">
    <small class="form-text text-muted">Abaikan jika tidak ingin ubah password</small>
    <small id="error-password" class="error-text form-text text-danger"></small>
  </div>
</div>
<div class="modal-footer">
  <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
  <button type="submit" class="btn btn-primary">Simpan</button>
</div>
</div>
</div>
</form>
<script>
$(document).ready(function() {
  $("#form-edit").validate({
    rules: {
      level_id: {required: true, number: true},
      username: {required: true, minlength: 3, maxlength: 20},
      nama: {required: true, minlength: 3, maxlength: 100},
      password: {minlength: 6, maxlength: 20}
    }
  });
});

```

```

    },
    submitHandler: function(form) {
        $.ajax({
            url: form.action,
            type: form.method,
            data: $(form).serialize(),
            success: function(response) {
                if(response.status){
                    $('#myModal').modal('hide');
                    Swal.fire({
                        icon: 'success',
                        title: 'Berhasil',
                        text: response.message
                    });
                    dataUser.ajax.reload();
                } else {
                    $('.error-text').text('');
                    $.each(response.msgField, function(prefix, val) {
                        $('#error-'+prefix).text(val[0]);
                    });
                    Swal.fire({
                        icon: 'error',
                        title: 'Terjadi Kesalahan',
                        text: response.message
                    });
                }
            }
        });
        return false;
    },
    errorElement: 'span',
    errorPlacement: function (error, element) {
        error.addClass('invalid-feedback');
        element.closest('.form-group').append(error);
    },
    highlight: function (element, errorClass, validClass) {
        $(element).addClass('is-invalid');
    },
    unhighlight: function (element, errorClass, validClass) {
        $(element).removeClass('is-invalid');
    }
});
});
</script>
@endempty

```

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

```

public function update_ajax(Request $request, $id) {
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantJson()) {

```



```

$rules = [
    'level_id' => 'required|integer',
    'username' => 'required|max:20|unique:m_user,username,'.$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
    ]);
}

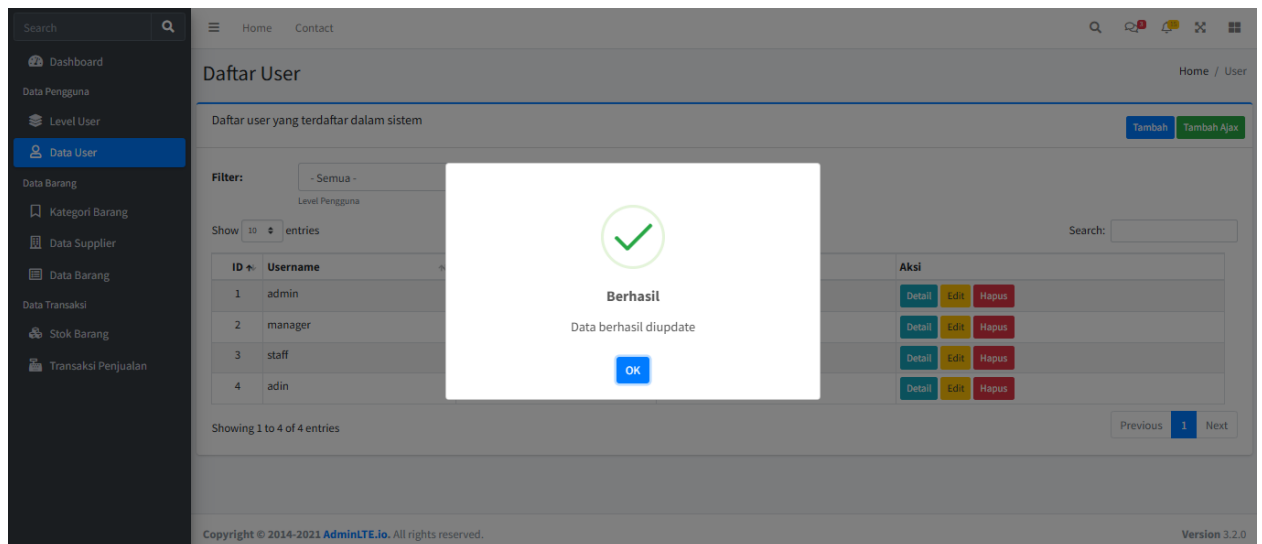
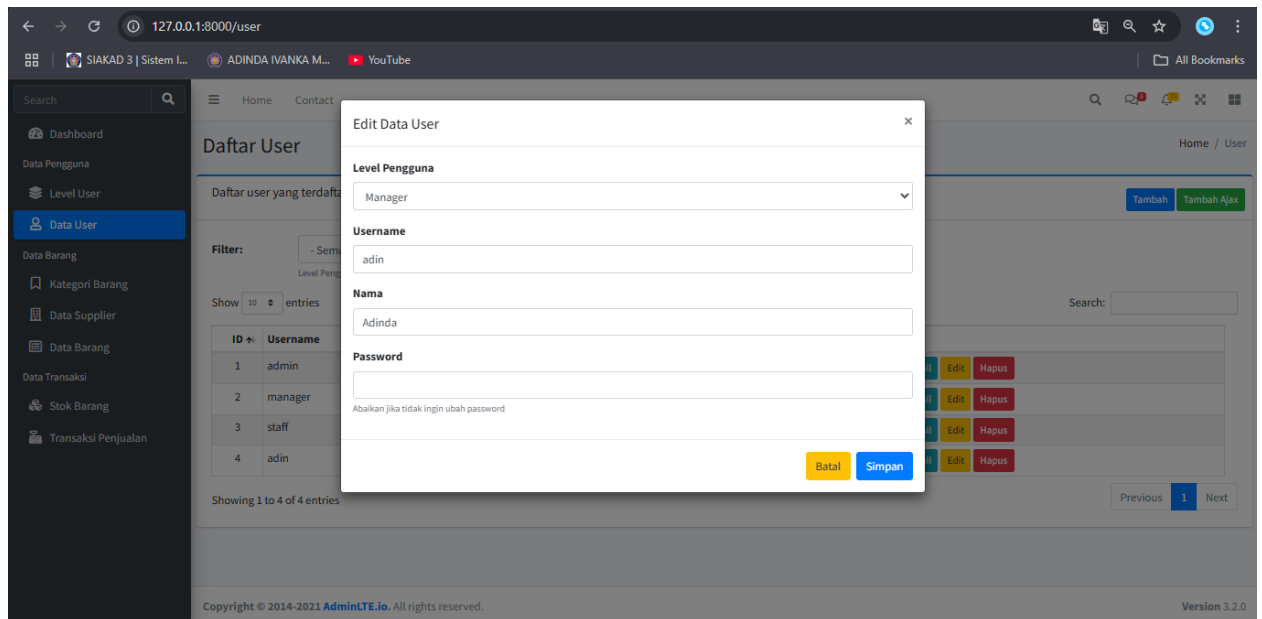
$check = UserModel::find($id);
if ($check) {
    // jika password tidak diisi, maka hapus dari request
    if (!$request->filled('password')) {
        $request->request->remove('password');
    }

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

return redirect('/');
}

```

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



Aktivasi tombol "Edit Ajax" akan memicu tampilan modal formulir Bootstrap secara dinamis tanpa memerlukan pemuatan ulang halaman. Formulir ini menyediakan kolom input untuk Level Pengguna, Username, Nama, dan Password, serta mengimplementasikan validasi otomatis di sisi pengguna melalui pustaka jQuery Validation untuk menampilkan indikasi kesalahan input. Apabila data yang dimasukkan telah memenuhi kriteria validasi, proses pengiriman data akan dilakukan secara asinkron menggunakan mekanisme Ajax. Setelah pembaruan data berhasil, modal formulir akan tertutup secara otomatis, notifikasi keberhasilan akan disampaikan melalui SweetAlert, dan tabel data pengguna akan diperbarui seketika tanpa memerlukan penyegaran halaman secara manual.

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
```

3. 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]);
}
```

4. Selanjutnya kita view untuk konfirmasi hapus data dengan nama 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">
            <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: { },
        submitHandler: function(form) {
            $.ajax({
                url: form.action,
                type: form.method,
                data: $(form).serialize(),
                success: function(response) {
                    if (response.status) {
                        $('#myModal').modal('hide');
                        Swal.fire({
                            icon: 'success',
                            title: 'Berhasil',
                            text: response.message
                        });
                        dataUser.ajax.reload();
                    } else {
                        $('.error-text').text("");
                        $.each(response.msgField, function(prefix, val) {
                            $('#error-' + prefix).text(val[0]);
                        });
                        Swal.fire({
                            icon: 'error',
                            title: 'Terjadi Kesalahan',
                            text: response.message
                        });
                    }
                }
            })
        }
    })
}
}

```

```

    });
    return false;
},
errorElement: 'span',
errorPlacement: function(error, element) {
    error.addClass('invalid-feedback');
    element.closest('.form-group').append(error);
},
highlight: function(element, errorClass, validClass) {
    $(element).addClass('is-invalid');
},
unhighlight: function(element, errorClass, validClass) {
    $(element).removeClass('is-invalid');
}
});
});
</script>
@empty

```

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

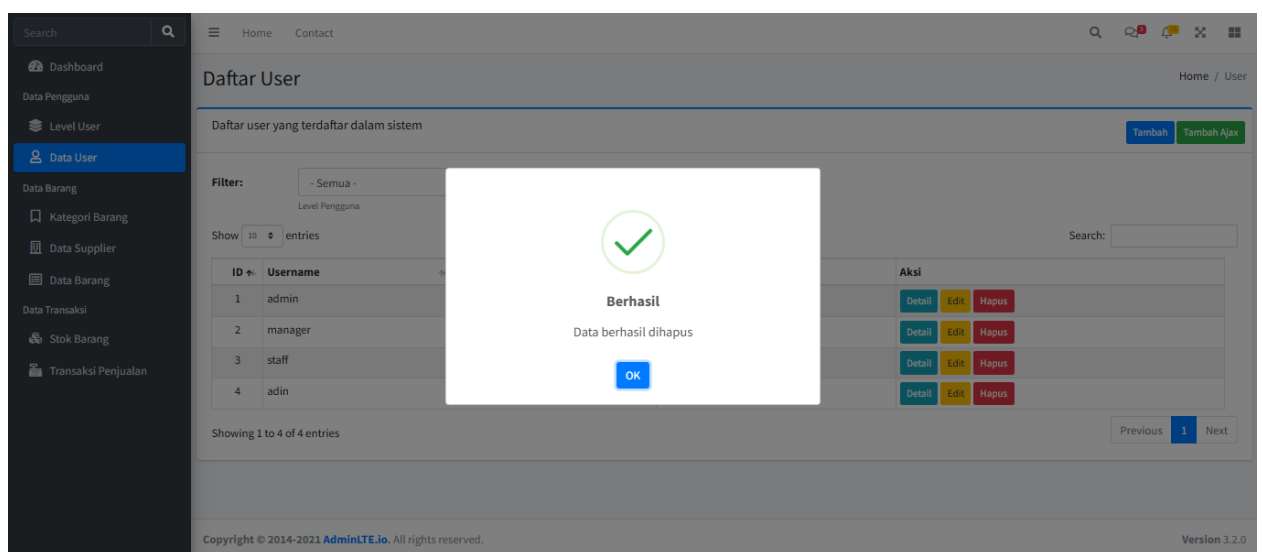
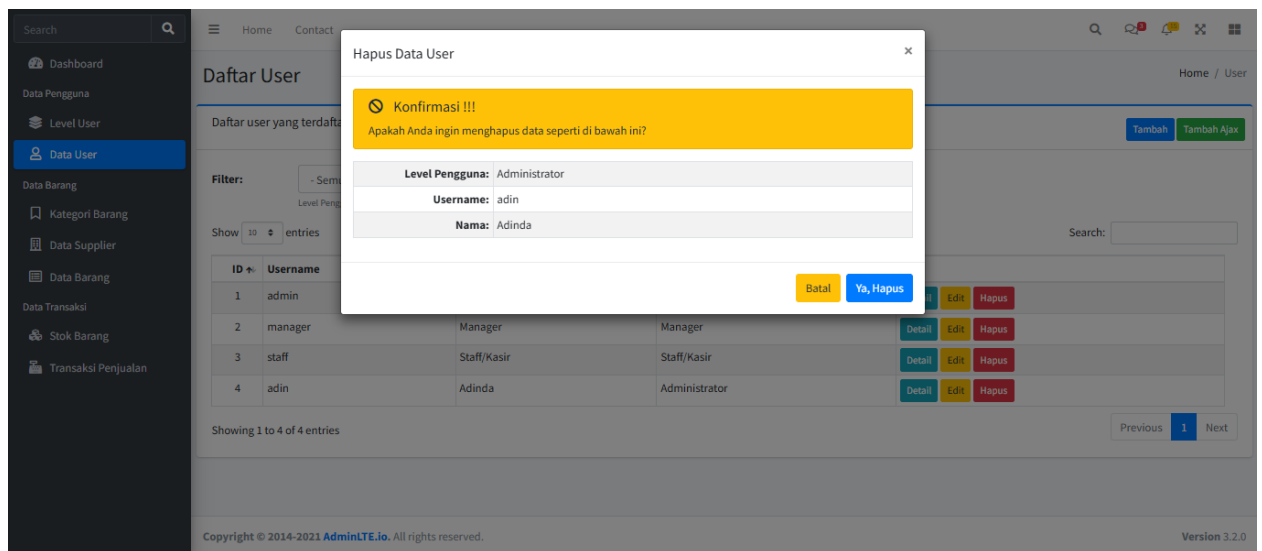
```

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

```

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.!!!



Ketika tombol "Hapus" ditekan, sebuah modal konfirmasi akan muncul untuk memastikan tindakan pengguna. Proses penghapusan data dilakukan tanpa memuat ulang halaman (reload). Setelah data berhasil dihapus, modal konfirmasi akan tertutup secara otomatis, dan notifikasi keberhasilan akan ditampilkan menggunakan SweetAlert. Selain itu, tabel data pengguna akan diperbarui secara langsung untuk mencerminkan perubahan terbaru.

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.

m_level

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Supplier

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

HomeContact

Daftar Level

Home / Level

Daftar level yang terdaftar dalam sistem

TambahTambah Ajax

Show 10 entriesSearch:

ID	Kode Level	Nama Level	Aksi
1	ADM	Administrator	Detail Edit Hapus
2	MNG	Manager	Detail Edit Hapus
3	STF	Staff/Kasir	Detail Edit Hapus

Showing 1 to 3 of 3 entriesPrevious1Next

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

m_kategori

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Supplier

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

HomeContact

Daftar Kategori

Home / Kategori

Daftar kategori yang terdaftar dalam sistem

TambahTambah Ajax

Show 10 entriesSearch:

ID	Kode Kategori	Nama Kategori	Aksi
1	EL	Elektronik	Detail Edit Hapus
2	PK	Pakaian	Detail Edit Hapus
3	MK	Makanan	Detail Edit Hapus
4	MI	Minuman	Detail Edit Hapus
5	ATK	Alat Tulis Kantor	Detail Edit Hapus

Showing 1 to 5 of 5 entriesPrevious1Next

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

m_supplier

Search

Q

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Supplier

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Home

Contact

Q

🗨️

🔔

⌵

☰

Daftar Supplier

Home / Supplier

Daftar supplier yang terdaftar dalam sistem

TambahTambah Ajax

Show10entries

Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	001	PT. Permata Indah	Jl.Kertosono No.12, Malang	Detail Edit Hapus
2	002	PT. Naga Emas	Jl.Sartika No.7, Surabaya	Detail Edit Hapus

Showing 1 to 2 of 2 entries

Previous1Next

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

Version 3.2.0

m_barang

Search

Q

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Supplier

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Home

Contact

Q

🗨️

🔔

⌵

☰

Daftar Barang

Home / Barang

Daftar barang yang terdaftar dalam sistem

TambahTambah Ajax

Filter:

- Semua -

Kategori Barang

Show10entries

Search:

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	TV001	Televisi	2000000	2500000	Elektronik	Detail Edit Hapus
2	HP002	Handphone	3000000	3500000	Elektronik	Detail Edit Hapus
3	KEM03	Kemeja	80000	100000	Pakaian	Detail Edit Hapus
4	JKT04	Jaket Jeans	200000	250000	Pakaian	Detail Edit Hapus
5	BRD05	Roti Tawar	15000	20000	Makanan	Detail Edit Hapus
6	MSD06	Mie Instan	2500	3500	Makanan	Detail Edit Hapus
7	UHT07	Susu UHT	5000	7500	Minuman	Detail Edit Hapus
8	JUS08	Jus Jeruk	5000	8000	Minuman	Detail Edit Hapus
9	PUL09	Pulpen	3000	4000	Alat Tulis Kantor	Detail Edit Hapus