



Nama	: Nimas Septiandini
Presensi/NIM	: 17/2341760087
Kelas	: SIB-2B

Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis
Semester : 4 (empat) / 5 (lima)
Pertemuan ke- : 6 (enam)

JOBSHEET 06

Ajax Form (AdminLTE) dan Client Validation

Proses pembuatan form CRUD (Create, Read, Update, Delete) dengan validasi di Laravel 10 menggunakan jQuery Validation melibatkan beberapa langkah penting yang mencakup pengaturan database, pembuatan model dan migrasi, pengembangan controller, penulisan view, dan penambahan validasi form di sisi klien. *Client side form validation* lebih dilakukan disisi browser dan bukan untuk tujuan keamanan, tetapi lebih ke kenyamanan pengguna. Sedangkan *server side validation* dilakukan di sisi server dengan tujuan keamanan dengan *filter* semua *request* yang masuk sebelum akhirnya diproses lanjutan.

Salah satu cara yang populer untuk melakukan validasi di sisi klien adalah dengan menggunakan plugin jQuery Validation. Plugin **jQuery Validation** digunakan untuk menambahkan validasi sisi klien pada form. Misalnya, Kita bisa mengatur agar suatu input wajib diisi dan tidak boleh lebih dari 255 karakter. Validasi ini membantu dalam memberikan umpan balik langsung kepada pengguna tentang kesalahan input tanpa perlu memuat ulang halaman ataupun mengirim *request* ke server.

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

A. AJAX form

AJAX (Asynchronous JavaScript and XML) adalah sebuah teknik atau metode dalam



pengembangan web yang memungkinkan aplikasi web untuk mengirim dan menerima data dari server secara asinkron (tanpa memuat ulang seluruh halaman). Dengan AJAX, interaksi antara

klien dan server menjadi lebih dinamis dan responsif, karena pengguna dapat berinteraksi dengan halaman web dan melihat perubahan langsung tanpa harus melakukan refresh halaman. *Ajax form* adalah teknik di mana sebuah form HTML dikirim ke server secara asinkron menggunakan AJAX, tanpa memuat ulang seluruh halaman web. Dengan AJAX form, Kita bisa mengirim data ke server dan menampilkan respons secara dinamis di halaman, sehingga meningkatkan pengalaman pengguna dengan membuat interaksi lebih cepat dan lebih responsif.

Mengapa Menggunakan AJAX Form?

1. **Response Instan:** AJAX memungkinkan Kita untuk mengirim data dan menerima respons dari server tanpa perlu memuat ulang halaman.
2. **Pengalaman Pengguna yang Lebih Baik:** Karena tidak ada pemuatan ulang halaman, aplikasi terasa lebih cepat dan lebih interaktif, mirip dengan aplikasi desktop.

Pengurangan Beban Server: Dengan mengirim hanya data yang diperlukan, AJAX dapat mengurangi penggunaan bandwidth dan beban di server.



B. Validasi Sisi Client

Validasi di sisi klien adalah proses pemeriksaan data yang dimasukkan oleh pengguna pada form web sebelum data tersebut dikirim ke server. Validasi ini dilakukan menggunakan kode yang berjalan di browser pengguna, seperti JavaScript, dan bertujuan untuk memastikan bahwa data yang dimasukkan sesuai dengan aturan tertentu, seperti format email yang benar, panjang karakter yang sesuai, atau tidak adanya kolom kosong yang wajib diisi.

Tujuan dan Manfaat Validasi di Sisi Klien

1. Umpan Balik Instan

Pengguna mendapatkan umpan balik segera setelah mereka memasukkan data yang tidak valid, seperti kesalahan format email atau kolom yang tidak diisi. Ini meningkatkan pengalaman pengguna (*user experience*) karena mereka tidak perlu menunggu respon dari server untuk mengetahui apakah input mereka benar atau salah.

2. Mengurangi Beban Server

Dengan melakukan validasi di sisi klien, kesalahan dapat diidentifikasi dan diperbaiki sebelum data dikirim ke server, sehingga server tidak perlu memproses permintaan yang tidak valid. Ini dapat mengurangi beban kerja server dan meningkatkan kinerja aplikasi.

3. Meningkatkan Efisiensi



Validasi di sisi klien membantu mencegah pengiriman data yang tidak valid, sehingga mengurangi jumlah permintaan HTTP yang perlu diproses oleh server. Hal ini menghemat bandwidth dan waktu pemrosesan, membuat aplikasi lebih efisien.

4. **Memastikan Integritas Data**

Dengan validasi sisi klien, banyak kesalahan input yang dapat dicegah sebelum data mencapai server. Misalnya, memastikan bahwa nomor telepon hanya berisi angka atau alamat email mengikuti format yang benar.

5. **Menyederhanakan Proses Pengembangan**

Dengan validasi di sisi klien, pengembang dapat menangani banyak potensi kesalahan input di awal, yang menyederhanakan logika pemrosesan di sisi server. Ini memungkinkan pengembang untuk fokus pada validasi yang lebih kompleks atau logika bisnis lainnya di server.

6. **Meningkatkan Keamanan**

Meskipun validasi sisi klien tidak bisa menggantikan validasi di sisi server (karena dapat dengan mudah diabaikan atau dimanipulasi oleh pengguna yang berpengalaman), validasi ini tetap dapat membantu dalam mengurangi jumlah data yang tidak valid yang mencapai server. Ini berfungsi sebagai lapisan pertama pertahanan, mencegah beberapa jenis input yang tidak diinginkan.

7. **Memberikan Panduan Pengguna**

Validasi sisi klien memungkinkan pengembang untuk memberikan panduan dan instruksi yang lebih baik kepada pengguna tentang cara memasukkan data dengan benar. Misalnya, pesan kesalahan bisa ditampilkan di bawah kolom yang salah, memberikan petunjuk spesifik kepada pengguna

Bagaimana Validasi di Sisi Klien Bekerja?

Validasi di sisi klien biasanya dilakukan menggunakan JavaScript atau framework JavaScript seperti jQuery. Berikut adalah contoh sederhana validasi form di sisi klien:

```
<!DOCTYPE html>
<html>
<head>
  <title>Client-Side Validation Example</title>
</head>
<body>
  <form name="myForm" onsubmit="return validateForm()" method="post">
    Name: <input type="text" name="name"><br><br>
    Email: <input type="text" name="email"><br><br>
    <input type="submit" value="Submit">
  </form>
  <script>
    function validateForm() {
      var email = document.forms["myForm"]["email"].value;
      var name = document.forms["myForm"]["name"].value;
```



```
if (name == "") {  
    alert("Name must be filled out");  
    return false;  
}  
  
var emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/;  
if (!emailPattern.test(email)) {  
    alert("Please enter a valid email address");  
    return false;  
}  
return true;  
}  
</script>  
</body>  
</html>
```

Keuntungan Validasi di Sisi Klien

1. Responsif

Pengguna mendapatkan respons cepat terhadap input mereka tanpa perlu menunggu interaksi dengan server.

2. Interaktif

Dapat memberikan instruksi tambahan dan lebih kontekstual kepada pengguna untuk memperbaiki kesalahan.

3. Penghematan Sumber Daya

Mengurangi jumlah permintaan ke server yang tidak perlu, sehingga menghemat bandwidth dan sumber daya server.

Keterbatasan Validasi di Sisi Klien

1. Tidak Mengganti Validasi di Sisi Server

Validasi di sisi klien dapat dilewati oleh pengguna berpengalaman atau perangkat otomatis. Oleh karena itu, validasi di sisi klien harus selalu dilengkapi dengan validasi di sisi server untuk memastikan keamanan dan integritas data.

2. Ketergantungan pada JavaScript

Jika pengguna menonaktifkan JavaScript di browser mereka, validasi di sisi klien tidak akan berfungsi.

Validasi di sisi klien merupakan komponen penting dalam pengembangan aplikasi web modern, karena meningkatkan pengalaman pengguna dan efisiensi aplikasi. Namun, ini harus selalu digunakan bersama dengan validasi di sisi server untuk menjaga keamanan dan memastikan data yang diterima oleh aplikasi adalah valid dan sesuai dengan aturan bisnis.



C. jQuery Validation

Salah satu cara yang populer untuk melakukan validasi di sisi klien adalah dengan menggunakan plugin jQuery Validation. **jQuery Validation** adalah plugin jQuery yang digunakan untuk memvalidasi form HTML di sisi klien secara efisien dan interaktif. Plugin ini memudahkan pengembang untuk menambahkan logika validasi pada form dengan cara yang mudah dan dapat disesuaikan, memberikan umpan balik langsung kepada pengguna mengenai kesalahan input mereka sebelum data dikirim ke server.

Fitur Utama jQuery Validation

1. Kemudahan Penggunaan

jQuery Validation dirancang untuk memudahkan integrasi dan penggunaan. Dengan beberapa baris kode, Kita dapat menambahkan validasi ke form HTML tanpa perlu menulis logika validasi dari awal.

2. Validasi Real-Time

Plugin ini memvalidasi input form secara real-time saat pengguna mengetik atau setelah mereka pindah dari satu field ke field lainnya. Ini memberikan umpan balik langsung kepada pengguna mengenai kesalahan input mereka.

3. Aturan Validasi yang Siap Pakai:

jQuery Validation menyediakan berbagai aturan validasi yang siap digunakan, seperti:

- *required*: Memastikan bahwa field tidak kosong.
- *email*: Memastikan bahwa input berformat alamat email yang valid.
- *url*: Memastikan bahwa input berformat URL yang valid.
- *minlength* dan *maxlength*: Membatasi jumlah karakter minimum atau maksimum dalam input.
- *number*: Memastikan bahwa input hanya berisi angka.

4. Pesan Kesalahan Kustom

Kita dapat menyesuaikan pesan kesalahan yang ditampilkan kepada pengguna. Misalnya, Kita dapat mengubah pesan default seperti "This field is required" menjadi sesuatu yang lebih spesifik atau sesuai dengan konteks aplikasi Kita.

5. Integrasi dengan jQuery UI

jQuery Validation dapat dengan mudah diintegrasikan dengan jQuery UI untuk menampilkan pesan kesalahan dalam format yang lebih menarik, seperti menggunakan tooltip atau dialog box.

6. Validasi Multi-Field



Plugin ini mendukung validasi yang melibatkan lebih dari satu field. Misalnya, Kita bisa memastikan bahwa dua field password dan konfirmasi password memiliki nilai yang sama.

7. Plugin dan Ekstensi

jQuery Validation memiliki ekosistem plugin dan ekstensi yang memungkinkan Kita menambahkan aturan validasi kustom atau mengubah perilaku default.

Cara Menggunakan jQuery Validation

Berikut adalah contoh sederhana bagaimana jQuery Validation digunakan untuk memvalidasi form:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>jQuery Validation Example</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-
  ui.css">
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
  validate/1.19.3/jquery.validate.min.js"></script>
</head>
<body>
  <form id="myForm">
    <label for="name">Name:</label><input type="text" name="name" id="name"><br>
    <label for="email">Email:</label><input type="text" name="email" id="email"><br>
    <input type="submit" value="Submit">
  </form>

  <script>
    $(document).ready(function() {
      $("#myForm").validate({
        rules: {
          name: "required",
          email: {
            required: true,
            email: true
          }
        },
        messages: {
          name: "Please enter your name",
          email: "Please enter a valid email address"
        }
      });
    });
  </script>
</body>
</html>
```

Penjelasan:

- **rules**: mendefinisikan aturan validasi untuk setiap field. Dalam contoh di atas:
 - *Field name* harus diisi (required).
 - *Field email* harus diisi dan harus berformat email yang valid (email).



- **messages:** mendefinisikan pesan kesalahan yang akan ditampilkan jika aturan validasi tidak terpenuhi.

Keuntungan Menggunakan jQuery Validation

1. Pengalaman Pengguna yang Lebih Baik

Pengguna mendapatkan umpan balik langsung, yang membantu mereka memperbaiki kesalahan input dengan cepat.

2. Pengurangan Beban Server

Validasi di sisi klien mengurangi jumlah permintaan yang tidak valid yang dikirim ke server, menghemat sumber daya server.

3. Fleksibilitas dan Kustomisasi

Plugin ini sangat fleksibel dan dapat dikustomisasi sesuai kebutuhan aplikasi, dari aturan validasi hingga pesan kesalahan yang ditampilkan.

4. Kompatibilitas dengan Semua Browser Modern

jQuery Validation kompatibel dengan hampir semua browser modern, sehingga dapat digunakan di berbagai lingkungan pengguna.

jQuery Validation memiliki berbagai metode bawaan yang sangat berguna untuk memvalidasi form di sisi klien. Selain metode standar seperti `required`, `email`, dan `number`, Kita juga dapat menambahkan metode validasi kustom menggunakan `addMethod`. Ini memungkinkan Kita untuk membuat aturan validasi yang lebih spesifik sesuai kebutuhan aplikasi Kita.

D. Method jQuery Validation

jQuery Validation menyediakan beberapa metode bawaan (built-in methods) yang dapat digunakan untuk memvalidasi form dengan berbagai jenis aturan. Selain itu, jQuery Validation juga memungkinkan pengembang untuk menambahkan metode kustom dengan `addMethod`, seperti yang telah dijelaskan sebelumnya. Berikut adalah beberapa metode tambahan yang tersedia dalam jQuery Validation

No	Method	Deskripsi
1	<code>required</code>	<ul style="list-style-type: none">• Memastikan bahwa field tidak kosong.• Contoh: <code>required: true</code>
2	<code>email</code>	<ul style="list-style-type: none">• Memastikan bahwa input berformat alamat email yang valid.• Contoh: <code>email: true</code>
3	<code>Url</code>	<ul style="list-style-type: none">• Memastikan bahwa input berformat URL yang valid.• Contoh: <code>url: true</code>



4	<i>date</i>	<ul style="list-style-type: none">Memastikan bahwa input berformat tanggal yang valid (berdasarkan pengaturan regional)Contoh: <code>date: true</code>
5	<i>dateISO</i>	<ul style="list-style-type: none">Memastikan bahwa input berformat tanggal yang valid dalam format ISO (YYYY-MM-DD)Contoh: <code>dateISO: true</code>
6	<i>number</i>	<ul style="list-style-type: none">Memastikan bahwa input hanya berisi angka (integer atau desimal).Contoh: <code>number: true</code>
7	<i>digits</i>	<ul style="list-style-type: none">Memastikan bahwa input hanya berisi angka (tanpa desimal).Contoh: <code>digits: true</code>
8	<i>creditcard</i>	<ul style="list-style-type: none">Memastikan bahwa input berformat nomor kartu kredit yang valid.Contoh: <code>creditcard: true</code>
9	<i>equalTo</i>	<ul style="list-style-type: none">Memastikan bahwa nilai elemen form sama dengan elemen lain (misalnya, untuk konfirmasi password).Contoh: <code>equalTo: "#password"</code>
10	<i>maxlength</i>	<ul style="list-style-type: none">Memastikan bahwa input tidak melebihi jumlah karakter tertentu.Contoh: <code>maxlength: 10</code>
11	<i>minlength</i>	<ul style="list-style-type: none">Memastikan bahwa input memiliki minimal jumlah karakter tertentu.Contoh: <code>minlength: 5</code>
12	<i>rangelength</i>	<ul style="list-style-type: none">Memastikan bahwa panjang input berada dalam rentang karakter tertentu.Contoh: <code>rangelength: [5, 10]</code>
13	<i>range</i>	<ul style="list-style-type: none">Memastikan bahwa nilai input berada dalam rentang tertentu (misalnya, angka 1 sampai 100)Contoh: <code>range: [1, 100]</code>
14	<i>max</i>	<ul style="list-style-type: none">Memastikan bahwa nilai input tidak melebihi angka maksimum tertentu.Contoh: <code>max: 100</code>
15	<i>min</i>	<ul style="list-style-type: none">Memastikan bahwa nilai input tidak kurang dari angka minimum tertentu.Contoh: <code>min: 1</code>
16	<i>remote</i>	<ul style="list-style-type: none">Memvalidasi nilai dengan mengirimkan permintaan ke server untuk memeriksa apakah nilai tersebut valid atau tersedia (misalnya, memeriksa ketersediaan username)Contoh<pre>remote: { url: "/check-username", type: "post" }</pre>
17	<i>step</i>	<ul style="list-style-type: none">Memastikan bahwa nilai input adalah kelipatan dari angka tertentu (berguna untuk validasi angka desimal).Contoh: <code>step: 10</code>
18	<i>phoneUS</i>	<ul style="list-style-type: none">Memastikan bahwa input berformat nomor telepon yang valid di AS.Contoh: <code>phoneUS: true</code>



19	<i>extension</i>	<ul style="list-style-type: none">Memastikan bahwa file yang diupload memiliki ekstensi tertentu.Contoh: <code>extension: "jpg png gif"</code>
20	<i>accept</i>	<ul style="list-style-type: none">Memastikan bahwa file yang diupload memiliki jenis MIME tertentu.Contoh: <code>accept: "image/*"</code>
21	<i>maxlength</i>	<ul style="list-style-type: none">Memastikan bahwa input hanya berisi karakter yang panjangnya sama persis dengan ketentuan.Contoh: <code>maxlength: 10</code>

jQuery Validation adalah alat yang sangat berguna untuk memastikan data yang dimasukkan ke dalam form web valid dan sesuai dengan aturan yang ditetapkan sebelum data tersebut dikirim ke server. Ini meningkatkan pengalaman pengguna, mengurangi kesalahan, dan mempermudah pengelolaan validasi form di sisi klien dalam pengembangan aplikasi web.

E. Praktikum Jobsheet

Langsung saja kita praktikkan untuk menggunakan Ajax form dan validasi disisi client.

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.

```
template.blade.php X
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>AdminLTE 3 | Blank Page</title>
7
8 <meta name="csrf-token" content="{{ csrf_token() }}">
9
10 <!-- Font Awesome -->
11 <link rel="stylesheet" href="{{ asset('plugins/fontawesome-free/css/all.min.css') }}">
12 <!-- Datatables -->
13 <link rel="stylesheet" href="{{ asset('plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}">
14 <link rel="stylesheet" href="{{ asset('plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}">
15 <link rel="stylesheet" href="{{ asset('plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}">
16 <!-- SweetAlert2 -->
17 <link rel="stylesheet" href="{{ asset('plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
18 <!-- Theme style -->
19 <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">
20
21 @stack('css')
22 </head>
```

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > aside.main-sidebar.sidebar-dar
2 <html lang="en">
3 <head>
14 <!-- Datatables -->
15 <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}">
16 <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}">
17 <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}">
18 <!-- SweetAlert2 -->
19 <link rel="stylesheet" href="{{ asset('plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
```



```
template.blade.php X
65 <script src="{{ asset('plugins/pdfmake/pdfmake.min.js') }}"></script>
66 <script src="{{ asset('plugins/pdfmake/vfs_fonts.js') }}"></script>
67 <script src="{{ asset('plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>
68 <script src="{{ asset('plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
69 <script src="{{ asset('plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>
70
71 <!-- jquery-validation -->
72 <script src="{{ asset('plugins/jquery-validation/jquery.validate.min.js') }}"></script>
73 <script src="{{ asset('plugins/jquery-validation/additional-methods.min.js') }}"></script>
74
75 <!-- SweetAlert2 -->
76 <script src="{{ asset('plugins/sweetalert2/sweetalert2.min.js') }}"></script>
77
78 <!-- AdminLTE App -->
79 <script src="{{ asset('dist/js/adminlte.min.js') }}"></script>
80 <script>
81     $.ajaxSetup({
82         headers: {
83             'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
84         }
85     });
86 </script>
87
88 @stack('js')
89 </body>
90 </html>
```

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini
2 <html lang="en">
25 <body class="hold-transition sidebar-mini">
79
80 <!-- jquery-validation -->
81 <script src="{{ asset('plugins/jquery-validation/jquery.validate.min.js') }}"></script>
82 <script src="{{ asset('plugins/jquery-validation/additional-methods.min.js') }}"></script>
83
84 <!-- SweetAlert2 -->
85 <script src="{{ asset('plugins/sweetalert2/sweetalert2.min.js') }}"></script>
86
```

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



```
@extends('layouts.template')

@section('content')
<div class="card card-outline card-primary">
  <div class="card-header">
    <h3 class="card-title">{{ $page->title }}</h3>
    <div class="card-tools">
      <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}">Tambah</a>
      <button onclick="modalAction('{{ url('user/create_ajax') }}')" class="btn btn-sm btn-success mt-1">Tambah Ajax</button>
    </div>
  </div>
  <div class="card-body">
    @if (@session('success'))
      <div class="alert alert-success">{{ session('success')}}</div>
    @endif
    @if (session('error'))
      <div class="alert alert-danger">{{ session('error')}}</div>
    @endif
    <div class="row">
      <div class="col-md-12">
        <div class="form-group row">
          <label class="col-1 control-label col-form-label">Filter:</label>
          <div class="col-3">
            <select class="form-control" id="level_id" name="level_id" required>
              <option value="">- Semua -</option>
              @foreach ($level as $item)
                <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
              @endforeach
            </select>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
resources > views > user > index.blade.php > div.card.card-outline.card-primary > div.card-header > div.card-to
4 <div class="card card-outline card-primary">
5 <div class="card-header">
6 <h3 class="card-title">{{ $page->title }}</h3>
7 <div class="card-tools">
8 <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}"
  ">Tambah</a>
9 <button onclick="modalAction('{{ url('user/create_ajax') }}')" class="btn
  btn-sm btn-success mt-1">Tambah Ajax</button>
10 </div>
```

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>
```

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>
```

```
resources > views > user > index.blade.php > div#myModal.modal.fade.animate.shake
4 <div class="card card-outline card-primary">
48 </div>
49 <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>
50 @endsection
```

5. 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

```
<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

@push('css')
@endpush

@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,
        ajax: {
            "url": "{{ url('/user/list') }}",
            "dataType": "json",
            "type": "POST",
            "data": function (d){
                d.level_id = $('#level_id').val();
            }
        },
        columns: [ {
```

Ubah seperti ini



```
resources > views > user > index.blade.php > script > <function>
55 @push('js')
56 <script>
57 function modalAction(url = ''){
58     $('#myModal').load(url,function(){
59         $('#myModal').modal('show');
60     });
61 }
62
63 var dataUser;
64 $(document).ready(function() {
65     dataUser = $('#table_user').DataTable({
66         // serverSide: true, jika ingin menggunakan server side processing
```

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

```
Route::group(['prefix' => 'user'], function () {
    Route::get('/', [UserController::class, 'index']); // Menampilkan halaman awal user
    Route::post('/list', [UserController::class, 'list']); // Menampilkan data user dalam bentuk json untuk datatables
    Route::get('/create', [UserController::class, 'create']); // Menampilkan halaman form tambah user
    Route::post('/', [UserController::class, 'store']); // Menyimpan data user baru
    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::get('/{id}', [UserController::class, 'show']); // Menampilkan detail user
    Route::get('/{id}/edit', [UserController::class, 'edit']); // Menampilkan halaman form edit user
    Route::put('/{id}', [UserController::class, 'update']); // Menyimpan perubahan data user
    Route::delete('/{id}', [UserController::class, 'destroy']); // Menghapus data user
});
```

```
42 Route::group(['prefix' => 'user'], function () {
43     Route::get('/', [UserController::class, 'index']); // menampilkan halaman awal user
44     Route::post('/list', [UserController::class, 'list']); // menampilkan data user dalam bentuk json un
45     Route::get('/create', [UserController::class, 'create']); // menampilkan halaman form tambah user
46     Route::post('/', [UserController::class, 'store']); // menyimpan data user baru
47     Route::get('/create_ajax', [UserController::class, 'create_ajax']); // Menampilkan halaman form ta
48     Route::post('/ajax', [UserController::class, 'store_ajax']); // Menyimpan data user baru Ajax
49     Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
50     Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
51     Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
52     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
53 });
```



7. 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);  
}
```

```
app > Http > Controllers > UserController.php  
11  
---  
194 public function create_ajax ()  
195     $level = LevelModel::select('level_id', 'level_nama')->get();  
196  
197     return view('user.create_ajax')  
198     ->with('level', $level);  
199  
200
```

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="form-  
control" required>  
                <small id="error-password" class="error-text form-text text-danger"></small>  
            </div>  
        </div>  
    </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>
</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`

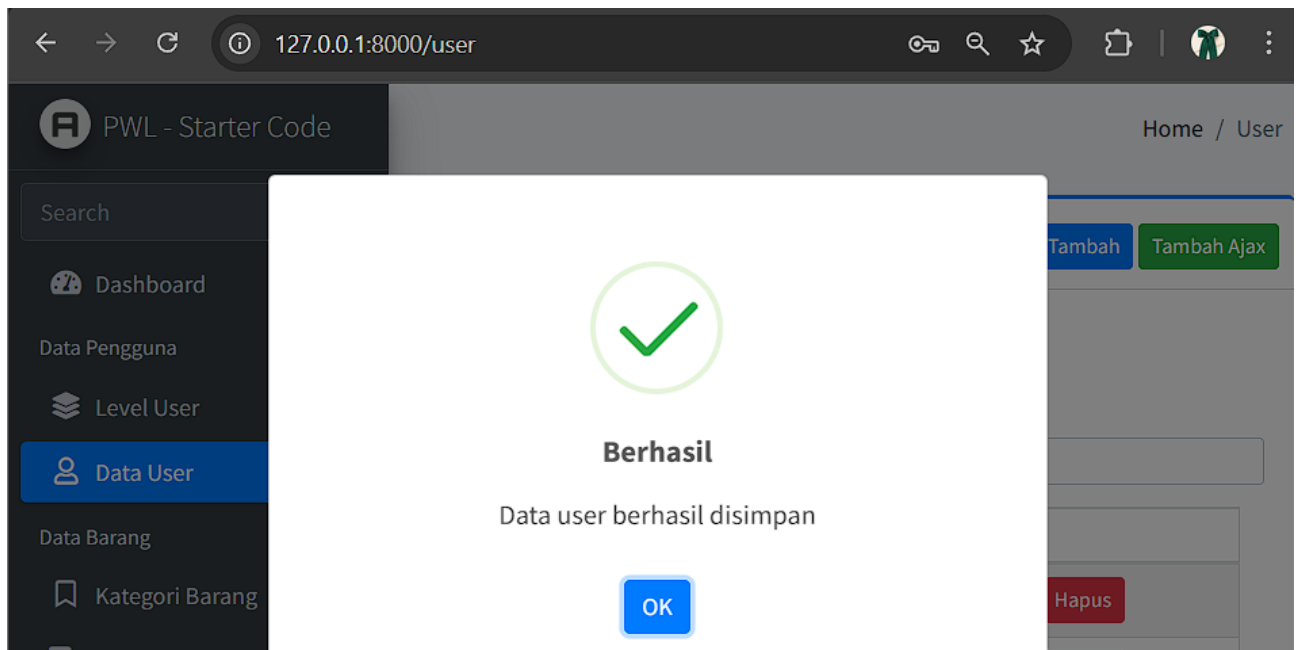
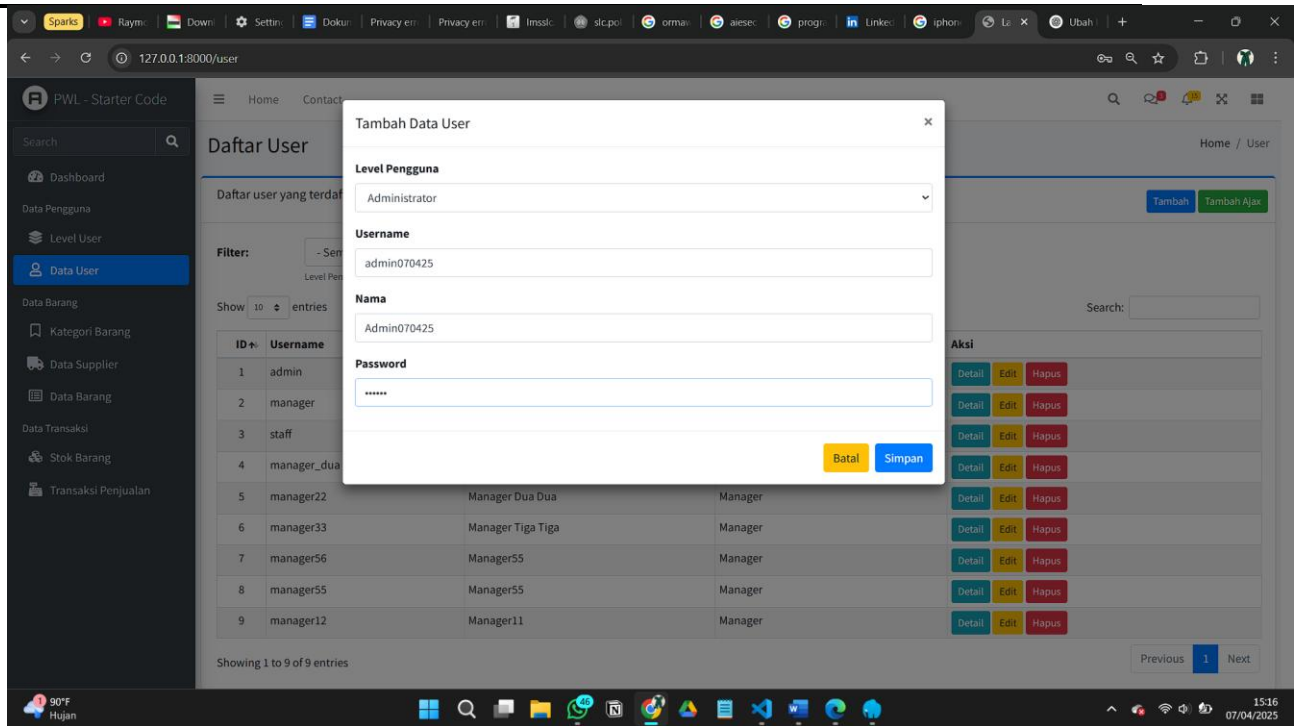


```
public function store_ajax(Request $request) {  
    // cek apakah request berupa ajax  
    if($request->ajax() || $request->wantsJson()){  
        $rules = [  
            'level_id' => 'required|integer',  
            'username' => 'required|string|min:3|unique:m_user,username',  
            'nama' => 'required|string|max:100',  
            'password' => 'required|min:6'  
        ];  
  
        // 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('/');  
}
```

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!!!

The screenshot shows a web application interface. On the left is a sidebar with a search bar and a list of menu items: Dashboard, Data Pengguna, Level User, Data User (highlighted), Data Barang, Kategori Barang, Data Supplier, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled 'Daftar User' and shows a list of registered users. The list has columns for ID, Username, Nama, Level Pengguna, and Aksi. There are 9 entries in the list. At the bottom right, there are 'Previous' and 'Next' navigation buttons.

ID	Username	Nama	Level Pengguna	Aksi
1	admin	admin_admin		Detail Edit Hapus
2	manager	Manager		Detail Edit Hapus
3	staff	Staff/Kasir		Detail Edit Hapus
4	manager_dua	Manager 2		Detail Edit Hapus
5	manager22	Manager Dua Dua		Detail Edit Hapus
6	manager33	Manager Tiga Tiga		Detail Edit Hapus
7	manager56	Manager55		Detail Edit Hapus
8	manager55	Manager55		Detail Edit Hapus
9	manager12	Manager11		Detail Edit Hapus



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



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

```
// 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 = '<a href="'.url('/user/' . $user->user_id).'>Detail</a>' ;
$btn .= '<a href="'.url('/user/' . $user->user_id . '/edit') . '>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') . '>' . '>Detail</button>' ;
$btn .= '<button onclick="modalAction(\'' . url('/user/' . $user->user_id .
'/edit_ajax') . '>' . '>Edit</button>' ;
$btn .= '<button onclick="modalAction(\'' . url('/user/' . $user->user_id .
'/delete_ajax') . '>' . '>Hapus</button>' ;

return $btn;
})
->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
->make(true);
}
```

```
app > Http > Controllers > UserController.php
14 {
15     public function list (Request $request)
16     {
17         $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
18             ->with('level');
19
20         // Filter data user berdasarkan level_id
21         if ($request->level_id) {
22             $users->where('level_id', $request->level_id);
23         }
24
25         return DataTables::of($users)
26             // menambahkan kolom index / no urut (default nama kolom: DF_RowIndex)
27             ->addIndexColumn()
28             ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
29                 // $btn = '<a href="'.url('/user/' . $user->user_id).'>Detail</a>' ;
30                 // $btn .= '<a href="'.url('/user/' . $user->user_id . '/edit') . '>Edit</a>' ;
31                 // $btn .= '<form class="d-inline-block" method="POST" action="'.url('/user/' . $user->user_id).'>'
32                 //         . csrf_field() . method_field('DELETE') .
33                 //         '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\'' . 'Apakah Anda yakin menghapus data ini?\' . '>Hapus</button></form>';
34                 $btn = '<button onclick="modalAction(\'' . url('/user/' . $user->user_id . '/show_ajax') . '>' . '>Detail</button>' ;
35                 $btn .= '<button onclick="modalAction(\'' . url('/user/' . $user->user_id . '/edit_ajax') . '>' . '>Edit</button>' ;
36                 $btn .= '<button onclick="modalAction(\'' . url('/user/' . $user->user_id . '/delete_ajax') . '>' . '>Hapus</button>' ;
37                 return $btn;
38             })
39             ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
40             ->make(true);
41     }
42 }
```

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

```
Route::group(['prefix' => 'user'], function () {
    Route::get('/', [UserController::class, 'index']); // Menampilkan halaman awal user
    Route::post('/list', [UserController::class, 'list']); // Menampilkan data user dalam bentuk json untuk datatables
    Route::get('/create', [UserController::class, 'create']); // Menampilkan halaman form tambah user
    Route::post('/', [UserController::class, 'store']); // Menyimpan data user baru
    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::get('/{id}', [UserController::class, 'show']); // Menampilkan detail user
    Route::get('/{id}/edit', [UserController::class, 'edit']); // Menampilkan halaman form edit user
    Route::put('/{id}', [UserController::class, 'update']); // Menyimpan perubahan data user
    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
    Route::delete('/{id}', [UserController::class, 'destroy']); // Menghapus data user
});
```



```
Route::group(['prefix' => 'user'], function () {
    Route::get('/', [UserController::class, 'index']); // menampilkan halaman awal user
    Route::post('/list', [UserController::class, 'list']); // menampilkan data user dalam bentuk json untuk datatables
    Route::get('/create', [UserController::class, 'create']); // menampilkan halaman form tambah user
    Route::post('/', [UserController::class, 'store']); // menyimpan data user baru
    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::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
    Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
    Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
    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
    Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
});
```

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]);
}
```

```
app > Http > Controllers > UserController.php
141 // Menyimpan perubahan data user

222 public function edit_ajax(string $id) {
223     $user = UserModel::find($id);
224     $level = LevelModel::select('level_id', 'level_name')->get();
225
226     return view('user.edit_ajax', ['user' => $user, 'level' => $level]);
227 }
228 }
```

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
```

```
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</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>
```



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->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'  
        ];  
    }  
}
```



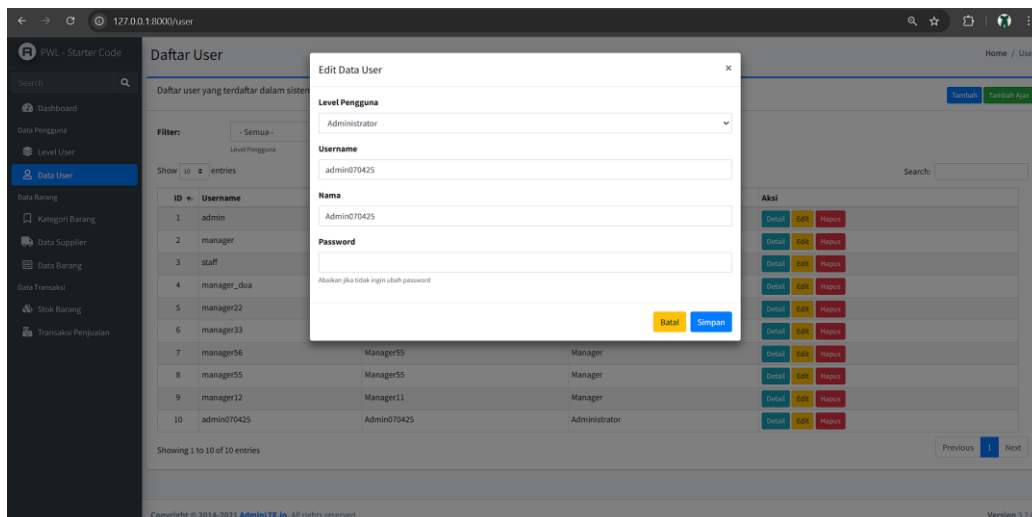

```
// use Illuminate\Support\Facades\Validator;
$validator = Validator::make($request->all(), $rules);

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('/');
}
```

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





Daftar User

Daftar user yang terdaftar dalam sistem

Filter: Level Pengguna

Show 10 entries

ID	Username	Nama	Aksi
1	admin	admin	Detail Edit Hapus
2	manager	Manag	Detail Edit Hapus
3	staff	Staff	Detail Edit Hapus
4	manager_dua	Manag	Detail Edit Hapus
5	manager22	Manag	Detail Edit Hapus
6	manager33	Manag	Detail Edit Hapus
7	manager56	Manag	Detail Edit Hapus
8	manager55	Manager55	Detail Edit Hapus
9	manager12	Manager11	Detail Edit Hapus
10	admin070425	Admin070425	Detail Edit Hapus

Showing 1 to 10 of 10 entries

10	admin070425	Admin070425_edit	Administrator	Detail Edit Hapus
----	-------------	------------------	---------------	---

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::group(['prefix' => 'user'], function () {  
    Route::get('/', [UserController::class, 'index']); // Menampilkan halaman awal user  
    Route::post('/list', [UserController::class, 'list']); // Menampilkan data user dalam bentuk json untuk datatables  
    Route::get('/create', [UserController::class, 'create']); // Menampilkan halaman form tambah user  
    Route::post('/', [UserController::class, 'store']); // Menyimpan data user baru  
    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::get('/{id}', [UserController::class, 'show']); // Menampilkan detail user  
    Route::get('/{id}/edit', [UserController::class, 'edit']); // Menampilkan halaman form edit user  
    Route::put('/{id}', [UserController::class, 'update']); // Menyimpan perubahan data user  
    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  
    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  
});
```

```
routes > web.php  
42 Route::group(['prefix' => 'user'], function () {  
43     Route::get('/', [UserController::class, 'index']); // menampilkan halaman awal user  
44     Route::post('/list', [UserController::class, 'list']); // menampilkan data user dalam bentuk json untuk datatabl  
45     Route::get('/create', [UserController::class, 'create']); // menampilkan halaman form tambah user  
46     Route::post('/', [UserController::class, 'store']); // menyimpan data user baru  
47     Route::get('/create_ajax', [UserController::class, 'create_ajax']); // Menampilkan halaman form tambah user Ajax  
48     Route::post('/ajax', [UserController::class, 'store_ajax']); // Menyimpan data user baru Ajax  
49     Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user  
50     Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user  
51     Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user  
52     Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); // Menampilkan halaman form edit user Ajax  
53     Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']); // Menyimpan perubahan data user Ajax  
54     Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); // Untuk tampilkan form confirm delete  
55     Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); // Untuk hapus data user Ajax  
56     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user  
57 });
```



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]);  
}
```

```
app > Http > Controllers > UserController.php  
141 // Menyimpan perubahan data user  
  
267 public function confirm_ajax(string $id){  
268     $user = UserModel::find($id);  
269  
270     return view('user.confirm_ajax', ['user' => $user]);  
271 }  
272 }  
273 }
```



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>
    </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>
@endempty
```

```
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</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>
```

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.

ID	Username	Nama	Level Pengguna	Aksi
1	admin	admin_admin	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	manager_dua	Manager 2	Manager	Detail Edit Hapus
5	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
6	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
7	manager56	Manager55	Manager	Detail Edit Hapus
8	manager55	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus
10	admin070425	Admin070425_edit	Administrator	Detail Edit Hapus

Hapus Data User

 **Konfirmasi !!!**

Apakah Anda ingin menghapus data seperti di bawah ini?

Level Pengguna :	Administrator
Username :	admin070425
Nama :	Admin070425_edit

[Batal](#) [Ya, Hapus](#)



ID	Username	Nama	Level Pengguna	Aksi
1	admin	admin_admin	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	manager_dua	Manager 2	Manager	Detail Edit Hapus
5	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
6	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
7	manager56	Manager55	Manager	Detail Edit Hapus
8	manager55	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus

Showing 1 to 9 of 9 entries

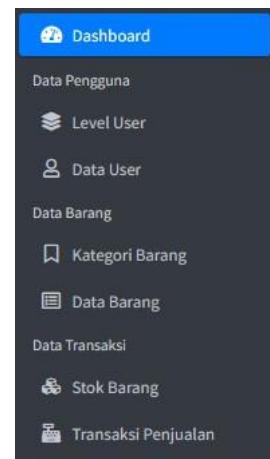
7. Jangan lupa laporkan ke laporan jobsheet dan lakukan *commit* pada *repository git* kalian.!!!

F. 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.





1. Tabel m_level

Add

Tambah Data Level

Level Kode
DRT

Level Nama
Direktur

Batal Simpan

Berhasil

Data level berhasil disimpan

OK

Edit

Show 10 entries					Search
No	Kode Level	Nama Level	Aksi		
1	ADM	Administrator	Detail	Edit	Hapus
2	MNG	Manager	Detail	Edit	Hapus
3	STF	Staff/Kasir	Detail	Edit	Hapus
4	DRT	Direktur Edit	Detail	Edit	Hapus

Delete



Level Pengguna

Show 10 entries

No	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 entries

2. Tabel m_kategori

Add

127.0.0.1:8000/kategori

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Supplier

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Daftar Kategori

Home / Kategori

Tambah Kategori

Tambah Ajax

Filter: - Semua -

Kategori Produk

Show 10 entries

Search:

No	Kode Kategori	Nama Kategori	Aksi
1	KAT001	Makanan	Detail Edit Hapus
2	KAT002	Elektronik	Detail Edit Hapus
3	KAT003	Meubel	Detail Edit Hapus
4	KAT004	Minuman	Detail Edit Hapus
5	KAT005	Perabotan Pecah Belah	Detail Edit Hapus
6	CML	Cemilan	Detail Edit Hapus
7	MNR	Minuman Ringan	Detail Edit Hapus

Showing 1 to 7 of 7 entries

Previous 1 Next

Daftar Kategori

Home / Kategori

Tambah Kategori

Tambah Ajax

Filter: - Semua -

Kategori Produk

Show 10 entries

Search:

No	Kode Kategori	Nama Kategori	Aksi
1	KAT001	Makanan	Detail Edit Hapus
2	KAT002	Elektronik	Detail Edit Hapus
3	KAT003	Meubel	Detail Edit Hapus
4	KAT004	Minuman	Detail Edit Hapus
5	KAT005	Perabotan Pecah Belah	Detail Edit Hapus
6	CML	Cemilan	Detail Edit Hapus
7	MNR	Minuman Ringan	Detail Edit Hapus

Showing 1 to 7 of 7 entries

Previous 1 Next

Berhasil

Kategori berhasil ditambahkan

OK



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

Show10entriesSearch:

No	Kode Kategori	Nama Kategori	Aksi		
1	KAT001	Makanan	Detail	Edit	Hapus
2	KAT002	Elektronik	Detail	Edit	Hapus
3	KAT003	Meubel	Detail	Edit	Hapus
4	KAT004	Minuman	Detail	Edit	Hapus
5	KAT005	Perabotan Pecah Belah	Detail	Edit	Hapus
6	CML	Cemilan	Detail	Edit	Hapus
7	MNR	Minuman Ringan	Detail	Edit	Hapus
8	KSH	Kesehatan	Detail	Edit	Hapus

Edit

Show10entriesSearch:

No	Kode Kategori	Nama Kategori	Aksi
1	KAT001	Makanan	Detail Edit Hapus
2	KAT002	Elektronik	Detail Edit Hapus
3	KAT003	Meubel	Detail Edit Hapus
4	KAT004	Minuman	Detail Edit Hapus
5	KAT005	Perabotan Pecah Belah	Detail Edit Hapus
6	CML	Cemilan	Detail Edit Hapus
7	MNR	Minuman Ringan	Detail Edit Hapus
8	KSH	Kesehatan Edit	Detail Edit Hapus

Delete

No	Kode Kategori	Nama Kategori	Aksi		
1	KAT001	Makanan	Detail	Edit	Hapus
2	KAT002	Elektronik	Detail	Edit	Hapus
3	KAT003	Meubel	Detail	Edit	Hapus
4	KAT004	Minuman	Detail	Edit	Hapus
5	KAT005	Perabotan Pecah Belah	Detail	Edit	Hapus
6	CML	Cemilan	Detail	Edit	Hapus
7	MNR	Minuman Ringan	Detail	Edit	Hapus

3. Tabel m_supplier

Add



Daftar Supplier Home / Supplier

Daftar Supplier Tambah Supplier Tambah Ajax

Filter: - Semua Supplier -
Pilih supplier untuk filter

Show 10 entries Search:

No	Nama Supplier	Kontak	Alamat	Aksi
1	PT Sumber Makmur	081234567890	Jl. Soekarno Hatta, Malang	Detail Edit Hapus
2	PT Mayora Indah	081987654321	Jl. Kenari, Surabaya	Detail Edit Hapus

Showing 1 to 2 of 2 entries Previous 1 Next

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

Daftar Supplier Home / Supplier

Daftar Supplier Tambah Supplier Tambah Ajax

Filter: - Semua Supplier -
Pilih supplier untuk filter

Show 10 entries Search:

No	Nama Supplier	Kontak	Alamat	Aksi
1	PT Sumber Makmur	081234567890	Jl. Soekarno Hatta, Malang	Detail Edit Hapus
2	PT Mayora Indah	081987654321	Jl. Kenari, Surabaya	Detail Edit Hapus

Showing 1 to 2 of 2 entries Previous 1 Next

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

Tambah Data Supplier

Supplier Kode

Supplier Nama

Supplier Telp

Supplier Alamat

Batal Simpan

Daftar Supplier Home / Supplier

Daftar Supplier Tambah Supplier Tambah Ajax

Filter: - Semua Supplier -
Pilih supplier untuk filter

Show 10 entries Search:

No	Nama Supplier	Kontak	Alamat	Aksi
1	PT Sumber Makmur	081234567890	Jl. Soekarno Hatta, Malang	Detail Edit Hapus
2	PT Mayora Indah Tbk.	081987654321	Jl. Kenari, Surabaya	Detail Edit Hapus

Showing 1 to 2 of 2 entries Previous 1 Next

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

Tambah Data Supplier

Supplier Kode

Supplier Nama

Supplier Telp

Supplier Alamat

Batal Simpan

Berhasil

Supplier berhasil ditambahkan

OK



Daftar Supplier

Home / Supplier

Daftar Supplier

Tambah Supplier

Tambah Ajax

Filter:

- Semua Supplier -

Pilih supplier untuk filter

Show

10

entries

Search:

No	Nama Supplier	Kontak	Alamat	Aksi
1	PT Sumber Makmur	081234567890	Jl. Soekarno Hatta, Malang	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	PT Mayora Indah Tbk.	081987654321	Jl. Kenari, Surabaya	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	PT Amerta Indah	085123123123	Kejayan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

Showing 1 to 3 of 3 entries

Previous

1

Next

Edit

Daftar Supplier

Home / Supplier

Daftar Supplier

Tambah Supplier

Tambah Ajax

Filter:

- Semua Supplier -

Pilih supplier untuk filter

Show

10

entries

Search:

No	Nama Supplier	Kontak	Alamat	Aksi
1	PT Sumber Makmur	081234567890	Jl. Soekarno Hatta, Malang	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	PT Mayora Indah Tbk.	081987654321	Jl. Kenari, Surabaya	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	PT Amerta Indah	085123123123	Kejayan, Pasuruan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

Showing 1 to 3 of 3 entries

Previous

1

Next

Edit Data Supplier

Kode Supplier

SUP003

Nama Supplier

PT Amerta Indah

Kontak

085123123123

Alamat

Kejayan, Pasuruan

Simpan

Berhasil

Data berhasil diperbarui

OK

Daftar Supplier

Home / Supplier

Daftar Supplier

Tambah Supplier

Filter:

- Semua Supplier -

Pilih supplier untuk filter

Show

10

entries

Search:

No	Nama Supplier	Kontak	Alamat	Aksi
1	PT Sumber Makmur	081234567890	Jl. Soekarno Hatta, Malang	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	PT Mayora Indah Tbk.	081987654321	Jl. Kenari, Surabaya	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	PT Amerta Indah	085123123123	Kejayan, Pasuruan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

Showing 1 to 3 of 3 entries

Previous

Delete



Daftar Supplier						Home /
Daftar Supplier						Tambah Supplier Tambah
Filter: - Semua Supplier -						
Pilih supplier untuk filter						
Show 10 entries						Search:
No	Nama Supplier	Kontak	Alamat	Aksi		
1	PT Sumber Makmur	081234567890	Jl. Soekarno Hatta, Malang	Detail	Edit	Hapus
2	PT Mayora Indah Tbk.	081987654321	Jl. Kenari, Surabaya	Detail	Edit	Hapus
Showing 1 to 2 of 2 entries						Previous 1

4. Tabel m_barang

Add

127.0.0.1:8000/barang

PWL - Starter Code

Search

Dashboard

Data Pengguna

- Level User
- Data User

Data Barang

- Kategori Barang
- Data Supplier
- Data Barang**
- Stok Barang
- Transaksi Penjualan

Daftar Barang

Show 10 entries

No	Kode Barang
1	B001
2	B002
3	B003
4	B004
5	B005
6	B006
7	B007
8	B008
9	B009
10	B010

Showing 1 to 10 of 11 entries

Tambah Data Barang

Kategori: Makanan

Kode Barang: B011

Nama Barang: Seblak Instan

Harga Beli: 6000

Harga Jual: 7000

✓

Berhasil

Data barang berhasil disimpan

OK

[Detail](#) [Edit](#) [Hapus](#)

[Simpan](#)

Daftar Barang

Show 10 entries

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Aksi
11	B011	Seblak Instan	6000	7000	Detail Edit Hapus

Showing 11 to 11 of 11 entries

[Previous](#) [1](#) [2](#) [Next](#)

Daftar Barang						Home / Barang
Daftar Barang						Tambah Barang Tambah Ajax
Show 10 entries						Search:
No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Aksi	
11	B011	Seblak Instan	6000	7000	Detail Edit Hapus	
Showing 11 to 11 of 11 entries						Previous 1 2 Next

Edit



Edit Data Barang

Kategori

Makanan

Kode Barang

B011

Nama Barang

Seblak Instan_Ed

Harga Beli

6000

Harga Jual

7000

Detail

Simpan

Berhasil

Data berhasil diupdate

OK

Daftar Barang						
Show 10 entries						
No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Aksi	
11	B011	Seblak Instan_Ed	6000	7000	Detail	Edit Hapus
Showing 11 to 11 of 11 entries						

Delete

Daftar Barang					
Daftar Barang					
Show 10 entries					
No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Aksi
1	B001	Laptop	5000000	6000000	Detail Edit Hapus
2	B002	Handphone	3000000	3500000	Detail Edit Hapus
3	B003	Polo	50000	75000	
4	B004	Pants	150000	200000	
5	B005	Nasi Bungkus	25000	35000	
6	B006	Air AQUA	5000	10000	
7	B007	Kipas Angin	200000	300000	
8	B008	Rice Cooker	400000	500000	
9	B009	Speaker Bluetooth	100000	150000	
10	B010	Teh Botol	7000	12000	
Showing 1 to 10 of 10 entries					

Berhasil

Data berhasil dihapus

OK