



Nama : Avicenna Mumtaza

Kelas/Absen : TI-2H/07

NIM : 2241720112

Mata Kuliah : Pemrograman Web Lanjut (PWL)

Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis

Semester : 4 (empat) / 6 (enam)

Pertemuan ke- : 1 (satu)

JOBSHEET 06

Template Form (AdminLTE), Server Validation, Client Validation, CRUD

Admin LTE adalah salah satu template yang sering digunakan oleh web developer sebagai template backend pada proyek yang sering dikerjakan. Jadi admin LTE ini adalah sebuah dashboard Administrator dibuat menggunakan bootstrap yang merupakan framework css yang paling banyak digunakan.

Dalam pertemuang kali ini kita akan memahami tentang bagaimana cara mengoperasikan form pada AdminLTE, Client & server validation, dan interaksi CRUD.

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

Laravel menyediakan beberapa pendekatan berbeda untuk memvalidasi data masuk aplikasi. Cara paling umum adalah menggunakan *validate method* yang tersedia pada semua permintaan HTTP yang masuk. Namun, kami juga akan membahas pendekatan validasi lainnya.

Proses validasi client side atau *client side validation* penerjemahannya lebih dilakukan oleh web browser sebagai client-nya, dimana didalam web browser sudah terdapat penggunaan



library yang mampu menerjemahkan perintah yang dijalankan di halaman web menggunakan client side scripting.

Contoh dari client side scripting ini antara lain seperti :

- HTML
- XHTML
- CSS
- JavaScript
- XML
- jQuery

Proses validasi dari *server side* dilakukan di sisi server, setelah data dikirimkan oleh browser dan berhasil diterima oleh server, selanjutnya di dalam server terdapat modul yang dilakukan untuk memvalidasi data, sebelum akhirnya proses diteruskan.

Proses validasi di sisi server side dilakukan dengan menggunakan bahasa pemrograman seperti PHP atau SQL dan lainnya.

Contoh dari server side scripting antara lain:

- ASP (Active Server Pages)
- PHP Hypertext Processor
- JSP (Java Server Pages)
- DII

Client side validation lebih dilakukan di sisi 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.

A. Template Form (AdminLTE)

1. Penggunaan template AdminLTE Bootstrap Admin Dashboard Template dengan langkah pertama akses <https://adminlte.io/>, lalu klik download pada source code (zip)



2. Lakukan extract All dan ubah nama folder *AdminLTE-3.2.0* menjadi **template**, folder template pindahkan ke *laragon/www/PWL_POS/public*. Buka folder public pada



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>

VSCode maka akan tampil folder baru bernama template



3. Copy isi public/template/index2.html pada welcome.blade.php lakukan edit agar dapat diload, edit href="{{ asset('template/.....') }}" dan src="{{ asset('template/.....') }}" sesuai gambar berikut

```
<!-- Google Font: Source Sans Pro -->
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
<!-- Font Awesome Icons -->
<link rel="stylesheet" href="{{ asset('template/plugins/fontawesome-free/css/all.min.css') }}">
<!-- overlayScrollbars -->
<link rel="stylesheet" href="{{ asset('template/plugins/overlayScrollbars/css/OverlayScrollbars.min.css') }}">
<!-- Theme style -->
<link rel="stylesheet" href="{{ asset('template/dist/css/adminlte.min.css') }}">
</head>
<body class="hold-transition dark-mode sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">
<div class="wrapper">

<!-- Preloader -->
<div class="preloader flex-column justify-content-center align-items-center">

</div>
```

4. Scroll ke bagian paling bawah dan sesuaikan dengan kode berikut

```
1713 <!-- REQUIRED SCRIPTS -->
1714 <!-- jQuery -->
1715 <script src="{{ asset('template/plugins/jquery/jquery.min.js') }}"></script>
1716 <!-- Bootstrap -->
1717 <script src="{{ asset('template/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
1718 <!-- overlayScrollbars -->
1719 <script src="{{ asset('template/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js') }}"></script>
1720 <!-- AdminLTE App -->
1721 <script src="{{ asset('template/dist/js/adminlte.js') }}"></script>
1722
1723 <!-- PAGE PLUGINS -->
1724 <!-- jQuery Mapael -->
1725 <script src="{{ asset('template/plugins/jquery-mousewheel/jquery.mousewheel.js') }}"></script>
1726 <script src="{{ asset('template/plugins/raphael/raphael.min.js') }}"></script>
1727 <script src="{{ asset('template/plugins/jquery-mapael/jquery.mapael.min.js') }}"></script>
1728 <script src="{{ asset('template/plugins/jquery-mapael/maps/usa_states.min.js') }}"></script>
1729 <!-- ChartJS -->
1730 <script src="{{ asset('template/plugins/chart.js/Chart.min.js') }}"></script>
1731
1732 <!-- AdminLTE for demo purposes -->
1733 <script src="{{ asset('template/dist/js/demo.js') }}"></script>
1734 <!-- AdminLTE dashboard demo (This is only for demo purposes) -->
1735 <script src="{{ asset('template/dist/js/pages/dashboard2.js') }}"></script>
1736 </body>
1737 </html>
1738
```

Note: apabila gambar pada dist/img belum dapat diload maka lakukan penyesuaian dengan asset ``

5. Jalankan pada browser



*note : template ini load untuk semuanya, hapus jika tidak diperlukan. Visual pada pada admin LTE dapat mempermudah untuk pengambilan kode.

6. Pada bagian menu terdapat pilihan Form kita khususnya dulu pada Form yang disediakan oleh Admin LTE

- General Elements
- Advanced Elements
- Editor
- Validation

Cara menggunakan template bisa dilihat nama formnya contoh judulnya 'General Elements', agar dapat digunakan dapat melakukan akses pada folder yang terdapat template adminLTE contoh : C:\laragon\www\PWL_2024\public\template\pages\forms

7. Buka file pada form> general cari keyword dengan ‘general elements’ dan coba modifikasi pada welcome.blade.php

```

2      <html lang="en">
15     <body class="hold-transition sidebar-mini">
16     <div class="wrapper">
831      <div class="content-wrapper">
850        <section class="content">
1153          </div>
1154        </div>
1155        <!-- /.card-body -->
1156      </div>
1157      <!-- /.card -->
1158
1159      <!-- general form elements disabled -->
1160      <div class="card card-warning">
1161        <div class="card-header">
1162          <h3 class="card-title">General Elements</h3>
1163        </div>
1164        <!-- /.card-header -->
1165        <div class="card-body">
1166          <form>
1167            <div class="row">
1168              <div class="col-sm-6">
1169                <!-- text input -->
1170                <div class="form-group">
1171                  <label>Text</label>
1172                  <input type="text" class="form-control" placeholder="Enter ...">
1173                </div>
1174              </div>

```



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>

```
@extends('adminlte::page')

@section('title', 'Dashboard')

@section('content_header')
    <h1>Dashboard</h1>
@stop

@section('content')

    <div class="card-body">
        <form>
            <div class="row">
                <div class="col-sm-6">
                    <!-- text input -->
                    <div class="form-group">
                        <label>Level id</label><input type="text" class="form-control" placeholder="id">
                        <div>

                    </div>
                </div>
                <button type = "submit" class ="btn btn-info">Submit </button>
            </div>
        </div>

    @stop

    @section('css')
        { {-- Add here extra stylesheets --} }
        { {-- <link rel="stylesheet" href="/css/admin_custom.css"> --} }
    @stop

    @section('js')
        <script> console.log("Hi, I'm using the Laravel-AdminLTE package!"); </script>
    @stop
```

8. Apa yang tampil?



The screenshot shows the AdminLTE Dashboard. On the left is a dark sidebar with a search bar and a menu containing 'Pages', 'Manage Roles', 'ACCOUNT SETTINGS', 'Profile', 'Change Password', 'Multi Level', 'TABLES', 'Important', 'Warning', and 'Information'. The main content area is titled 'Dashboard' and contains a form with a label 'Level id', a text input field, and a blue 'Submit' button.

9. Eksplorasi jenis form pada adminLTE dan coba terapkan yang sesuai untuk studi kasus POS PWL, buatlah form untuk tabel `m_user` dan `m_level`

The screenshot shows the 'General Form' in AdminLTE. It has a yellow header bar with the text 'Form di Level'. Below this are four form fields: 'Select Level' (a dropdown menu), 'Username' (a text input field with a placeholder 'Masukkan Username'), 'Name' (a text input field with a placeholder 'Masukkan Nama User'), and 'Password' (a text input field with a placeholder 'Masukkan Password'). A blue 'Simpan' button is at the bottom of the form.



B. VALIDASI PADA SERVER

1. Buka kembali file tugas di jobsheet 05, klik dan baca documentation ini [Validation - Laravel 10.x - The PHP Framework For Web Artisans](#)
2. Pastikan sudah terdefiniskan route pada web.php :

```
Route::get('/kategori/create', [KategoriController::class, 'create']);  
Route::post('/kategori', [KategoriController::class, 'store']);
```

```
33  
34 Route::get('/kategori/create', [KategoriController::class, 'create']);  
35 Route::post('/kategori', [KategoriController::class, 'store']);  
36
```

3. Edit pada KategoriKontroller dengan kode:



```
app > Http > Controllers > KategoriController.php > PHP > App\Http\Controllers\KategoriController
1
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\RedirectResponse;
6 use Illuminate\Http\Request;
7 use Illuminate\View\View;
8
9 3 references | 0 implementations
10 class KategoriController extends Controller
11 {
12     /**
13      * Show the form to create a new post.
14      */
15     0 references | 0 overrides
16     public function create(): View
17     {
18         return view('kategori.create');
19     }
20
21     /**
22      * Store a new post.
23      */
24     0 references | 0 overrides
25     public function store(Request $request): RedirectResponse
26     {
27         $validated = $request->validate([
28             'kategori_kode' => 'required',
29             'kategori_nama' => 'required',
30         ]);
31
32         // The post is valid...
33
34         return redirect('/kategori');
35     }
36 }
```

4. Berikut ini sebagai alternatif contoh penulisan validasi dan bisa Anda sesuaikan katakata dengan apa yang sedang Anda kerjakan,

Alternatively, validation rules may be specified as arrays of rules instead of a single | delimited string:

```
$validatedData = $request->validate([
    'title' => ['required', 'unique:posts', 'max:255'],
    'body' => ['required'],
]);
```

In addition, you may use the `validateWithBag` method to validate a request and store any error messages within a named error bag:

```
$validatedData = $request->validateWithBag('post', [
    'title' => ['required', 'unique:posts', 'max:255'],
    'body' => ['required'],
]);
```

5. Tulis perbedaan penggunaan `validate` dengan `validateWithBag`!



Perbedaan `validate` dan `validateWithBag` adalah, hasil validasi `validateWithBag` akan disimpan dalam “bag” yang dapat disesuaikan (misalnya, “email”), sedangkan Hasil validasi `validate` akan disimpan dalam Session secara default.

6. Terkadang Anda mungkin ingin berhenti menjalankan aturan validasi pada suatu atribut setelah kegagalan validasi pertama. Untuk melakukannya, tetapkan **bail** aturan ke atribut: coba sesuaikan dengan field pada `m_kategori`, yuk bisa.. apa yang terjadi?

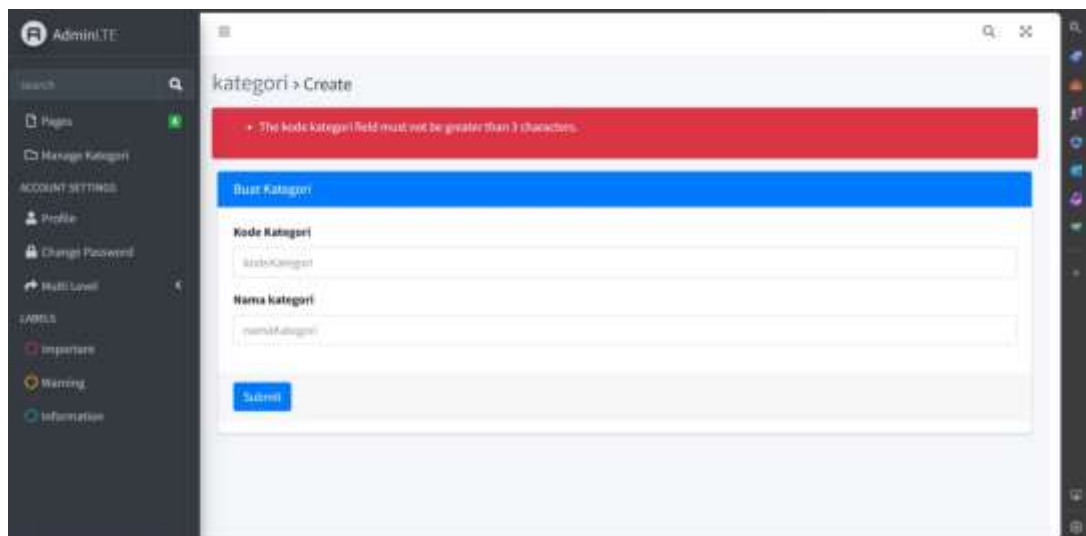
```
$request->validate([  
    'title' => 'bail|required|unique:posts|max:255',  
    'body' => 'required',  
]);
```

7. Pada `view/create.blade.php` tambahkan code berikut agar ketika validasi gagal, kita dapat menampilkan pesan kesalahan dalam tampilan:



```
@if ($errors->any())  
    <div class="alert alert-danger">  
        <ul>  
            @foreach ($errors->all() as $error)  
                <li>{{ $error }}</li>  
            @endforeach  
        </ul>  
    </div>  
@endif
```

8. Screenshot hasilnya.



9. Catatan : [Menyesuaikan Pesan Kesalahan](#)

Aturan validasi bawaan Laravel masing-masing memiliki pesan kesalahan yang terletak di `lang/en/validation.php` file aplikasi Anda. Jika aplikasi Anda tidak memiliki `lang` direktori, Anda dapat menginstruksikan Laravel untuk membuatnya menggunakan `lang:publish` perintah Artisan. Di dalam `lang/en/validation.php` file tersebut, Anda akan menemukan entri terjemahan untuk setiap aturan validasi. Anda bebas mengubah atau memodifikasi pesan-pesan ini berdasarkan kebutuhan aplikasi Anda.

10. Pada `view/create.blade.php` tambahkan dan coba running code berikut :



```
<label for="kategori_kode">Kode kategori</label>

<input id="kategori_kode"
  type="text"
  name="kategori_kode"
  class="@error('kategori_kode') is-invalid @enderror">

@error('kategori_kode')
  <div class="alert alert-danger">{{ $message }}</div>
@enderror
```

C. FORM REQUEST VALIDATION

1. Membuat permintaan form dengan menuliskan pada terminal

```
php artisan make:request StorePostRequest
```

kelas permintaan formulir yang dihasilkan akan ditempatkan di `app/Http/Requests` direktori. Jika direktori ini tidak ada, maka akan dibuat saat Anda menjalankan `make:request` perintah. Setiap permintaan formulir yang dihasilkan oleh Laravel memiliki dua metode: `authorize` dan `rules`.

```
PS C:\laragon\www\PWL_POS> php artisan make:request StorePostRequest

INFO Request [C:\laragon\www\PWL_POS\app\Http\Requests\StorePostRequest.php]
created successfully.

PS C:\laragon\www\PWL_POS>
```

2. Ketik kode berikut pada `Http/request/StorePostRequest`



```
/**
 * Get the validation rules that apply to the request.
 *
 * @return array<string, \Illuminate\Contracts\Validation\ValidationRule|array<mixed>|string>
 */
0 references | 0 overrides
public function rules(): array
{
    return [
        'kategori_kode' => 'required',
        'kategori_nama' => 'required',
    ];
}

0 references | 0 overrides
public function store(StorePostRequest $request): RedirectResponse
{
    // The incoming request is valid...

    // Retrieve the validated input data...
    $validated = $request->validated();

    // Retrieve a portion of the validated input data...
    $validated = $request->safe()->only(['kategori_kode', 'kategori_nama']);
    $validated = $request->safe()->except(['kategori_kode', 'kategori_nama']);

    // Store the post...

    return redirect('/kategori');
}
```

Note : Jika validasi gagal, respons pengalihan akan dihasilkan untuk mengirim pengguna kembali ke lokasi sebelumnya. Kesalahan juga akan ditampilkan ke sesi sehingga tersedia untuk ditampilkan. Jika permintaannya adalah permintaan XHR, respons HTTP dengan kode status 422 akan dikembalikan ke pengguna.

3. Terapkan validasi juga pada tabel m_user dan m_level.



level > Create

Buat Level

Kode Level

Masukkan Kode Level

The kode level field is required.

Nama level

Masukkan Nama Level

The nama level field is required.

Submit

Edit Level Form

- The kode level field is required.
- The nama level field is required.

Edit Level

Level ID

4

Kode Level

CUS

The kode level field is required.

Nama Level

Pelanggan

The nama level field is required.

Koreksi **Update**



D. CRUD(create, read, update, delete)

1. Buat POSController lengkap dengan resourcenya, Membuat Resource Controller dan Route yang berfungsi untuk route CRUD sehingga tidak perlu repot-repot membuat masing-masing route seperti post, get, delete dan update.

```
php artisan make:controller POSController --resource
```



```
PS C:\laragon\www\PWL_POS2> php artisan make:controller POSController --resource  
  
INFO Controller [C:\laragon\www\PWL_POS2\app\Http\Controllers\POSController.php] created successfully.  
PS C:\laragon\www\PWL_POS2> |
```

2. Berikut perintahnya: sekarang tambahkan kode pada route/web.php

```
Route::resource('m_user', POSController::class);
```

3. Atur pada Models/m_user
4. Atur pada Migration/m_user_table

```
<?php use App\Models\m_level; use  
Illuminate\Database\Migrations\Migration;
```



```
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up():
    void
    {
        Schema::create('useri', function (Blueprint $table) {
            $table-> id('user_id');
            $table -> unsignedBigInteger('level_id')->index;
            $table ->string ('username', 20)->unique();
            $table ->string ('nama', 100);
            $table ->string ('password');
            $table ->timestamps();

            $table->foreign('level_id')-> references('level_id')->on
('m_level');
        });
    }
    /**
     * Reverse the migrations.
     */
    public function down():
    void
    {
        Schema::dropIfExists('useri');
    }
};
```

5. Pada file Controller yang kita dapat secara otomatis akan terdapat 7 fungsi berikut yang dapat kita gunakan untuk membuat operasi CRUD.
- index()
 - create()
 - store()
 - show()



- edit()
- update()
- destroy()

Tugas:

1. Coba tampilkan level_id pada halaman web tersebut dimana field ini merupakan foreign key

User id	Level id	username	nama	password	Action
1	1	admin	Administrator	\$2y\$12\$2yeMg8LTgWPOvKaM42ctaBUngFuQch7oqwsld5AOF6uQgHOGokI	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$T/GcmUjHLezpMc8FDHLeihVR.CeWM2DVC37xM35vuUzXXkGJke	Show Edit Delete
3	3	staff	Staff	\$2y\$12\$NwOTQpRAjnlufudu.qQqH.6bJa1aVa3B7qD099QpkL1ychcujJlm	Show Edit Delete
6	4	customer-1	Pelanggan	\$2y\$12\$1wMXPMg/Ca6U0X6DH4Qg.r65nPORd2vgoAZRCvCVH9mnhgjr63i	Show

2. Modifikasi dengan tema/ template kesukaan Anda

User ID	Level ID	Username	Nama	Password	Action
1	1	admin	Administrator	\$2y\$12\$2yeMg8LTgWPOvKaM42ctaBUngFuQch7oqwsld5AOF6uQgHOGokI	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$T/GcmUjHLezpMc8FDHLeihVR.CeWM2DVC37xM35vuUzXXkGJke	Show Edit Delete
3	3	staff	Staff	\$2y\$12\$NwOTQpRAjnlufudu.qQqH.6bJa1aVa3B7qD099QpkL1ychcujJlm	Show Edit Delete
6	4	customer-1	Pelanggan Pertama	\$2y\$12\$1wMXPMg/Ca6U0X6DH4Qg.r65nPORd2vgoAZRCvCVH9mnhgjr63i	Show Edit Delete
7	2	manager_dua	Manager 2	\$2y\$12\$g7IPIfb57K.IEt9pxFAEOxdNkpsYGAhnoVKIO0.Xo.0BpuV36	Show Edit Delete
8	2	manager22	Manager Dua Dua	\$2y\$12\$9Doo3vSpTIGXQR1yjahPnu7ENFoE53ozVy87eccOn4m48/qP707G	Show Edit Delete
9	2	manager33	Manager Tiga Tiga	\$2y\$12\$GWh4Pg6CIZhHQ8ue7PLJau1FevHvZhmBG5DGvFabn6UBwb5xT6.tW	Show Edit Delete
10	2	manager56	Manager55	\$2y\$12\$VYdf1lu/cilBdzohcemGuQ2rnhaiWoe5LcZQnQK1dUeHzJGHCSNe	Show Edit Delete

3. Apa fungsi \$request->validate, \$error dan alert yang ada pada halaman CRUD tersebut



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>

- `$request->validate`: `$request->validate` adalah metode yang digunakan untuk melakukan validasi pada data yang dikirim melalui HTTP request. Ketika kita menerima data dari form (misalnya saat menyimpan data melalui form input), kita dapat menggunakan `$request->validate` untuk memastikan bahwa data tersebut memenuhi kriteria validasi yang telah ditentukan. Jika validasi gagal, Laravel akan secara otomatis mengarahkan pengguna kembali ke halaman form dengan pesan kesalahan yang relevan.
- `$errors`: `$errors` digunakan untuk memberikan informasi tentang kesalahan validasi yang terjadi. Setelah validasi gagal, kita dapat mengakses `$errors` untuk menampilkan pesan kesalahan secara spesifik.
- `alert`: `alert` digunakan untuk menampilkan pesan kepada pengguna. Ketika kita ingin memberikan informasi atau peringatan kepada pengguna (misalnya setelah berhasil atau gagal menyimpan data), kita dapat menggunakan `alert`.

**** Sekian, dan selamat belajar ****