



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 dihalaman web menggunakan client side scripting. Contoh dari client side scripting ini antara lain seperti :

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



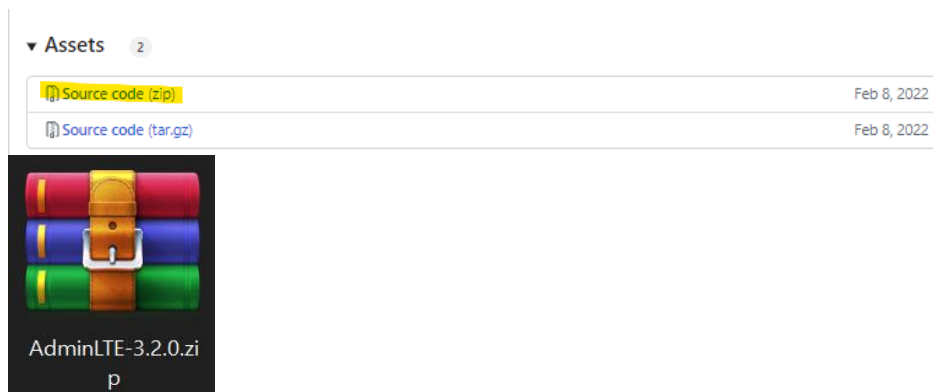
Proses validasi dari *server side* dilakukan di sisi server, setelah data dikirimkan oleh browser dan berhasil diterima oleh server, selanjutnya didalam server terdapat modul yang dilakukan untuk memvalidasi data, sebelum akhirnya proses diteruskan. Proses validasi di sisi server side dilakukan dengan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

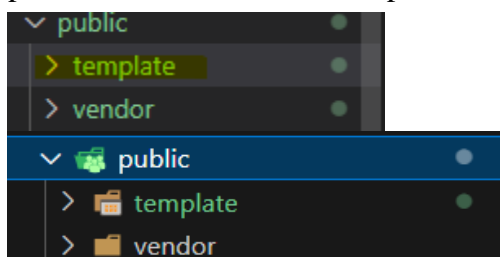
Client side validation lebih dilakukan disisi browser dan bukan untuk tujuan keamanan, tetapi lebih ke kenyamanan pengguna. Sedangkan *server side validation* dilakukan di sisi server dengan tujuan keamanan dengan *filter* semua *request* yang masuk sebelum akhirnya diproses lanjutan.

#### 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**, foldertemplate pindahkan ke laragon/www/PWL\_2024/public. Buka folder public pada 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>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>AdminLTE 3 | Dashboard 2</title>

  <!-- 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
```

```
<!-- REQUIRED SCRIPTS -->
<!-- jQuery -->
<script src="{{ asset('template/plugins/jquery/jquery.min.js') }}"></script>
<!-- Bootstrap -->
<script src="{{ asset('template/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<!-- overlayScrollbars -->
<script src="{{ asset('template/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js') }}"></script>
<!-- AdminLTE App -->
<script src="{{ asset('template/dist/js/adminlte.js') }}"></script>

<!-- PAGE PLUGINS -->
<!-- jQuery Mapael -->
<script src="{{ asset('template/plugins/jquery-mousewheel/jquery.mousewheel.js') }}"></script>
<script src="{{ asset('template/plugins/raphael/raphael.min.js') }}"></script>
<script src="{{ asset('template/plugins/jquery-mapael/jquery.mapael.min.js') }}"></script>
<script src="{{ asset('template/plugins/jquery-mapael/maps/usa_states.min.js') }}"></script>
<!-- ChartJS -->
<script src="{{ asset('template/plugins/chart.js/Chart.min.js') }}"></script>

<!-- AdminLTE for demo purposes -->
<script src="{{ asset('template/dist/js/demo.js') }}"></script>
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script src="{{ asset('template/dist/js/pages/dashboard2.js') }}"></script>
</body>
</html>
```

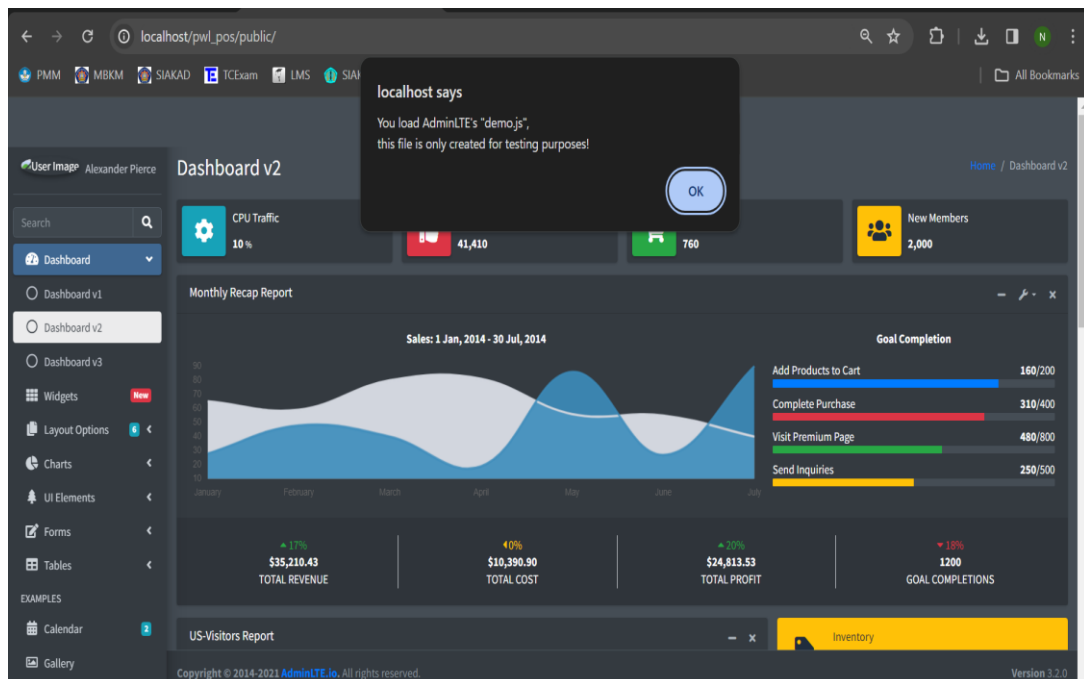
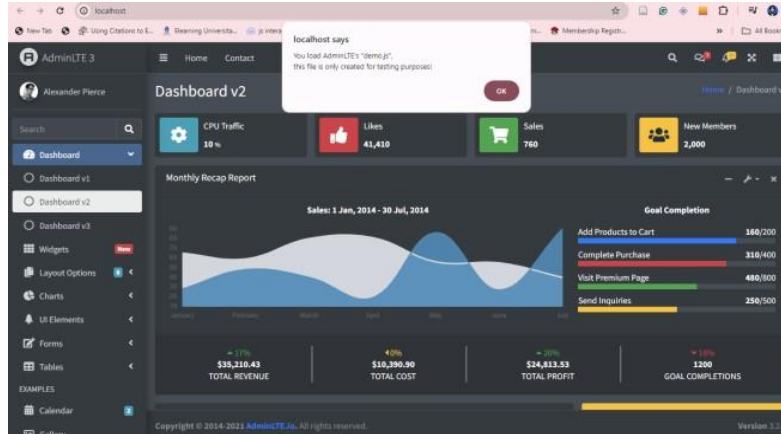
Note: apabila gambar pada dist/img belum dapat diload maka lakukan penyesuaian dengan asset.

```

```



## 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 khususkan 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>
```



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

@section('title', 'Dashboard')

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

@section('content')

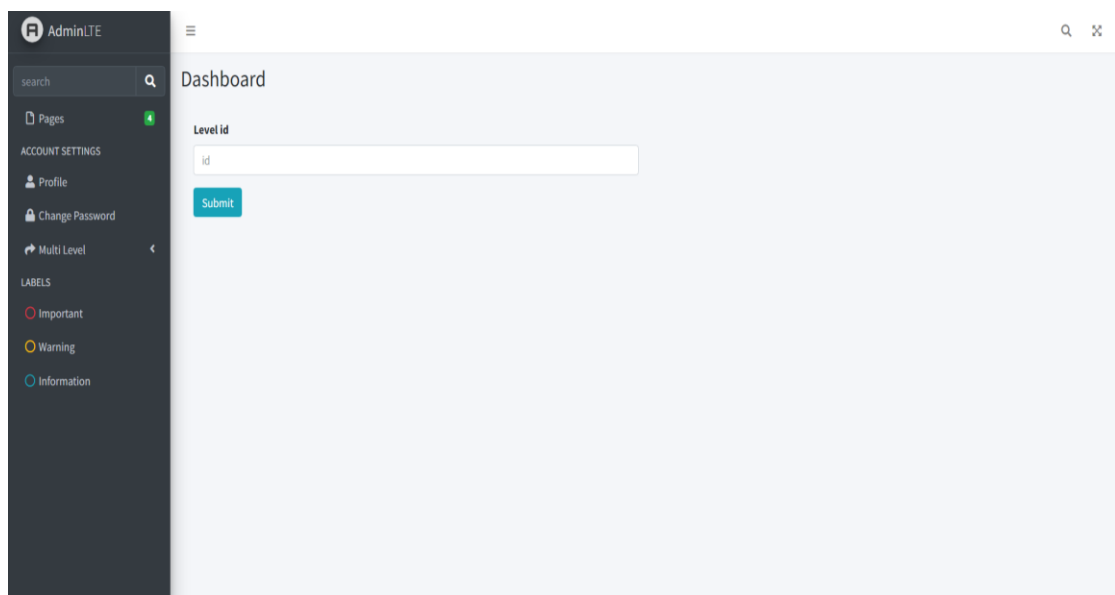
    <div class="card-body">
        <form>
            <div class="row">
                <div class="col-sm-6">
                    <!-- text input -->
                    <div class="form-group">
                        <label>Level id</label><input type="text" class="form-control" placeholder="id">
  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





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

m\_user

The screenshot shows the 'Tambah Pengguna' (Add User) form in the AdminLTE interface. The form is located on the right side of the dashboard, with a sidebar on the left containing navigation links. The form fields are as follows:

- Nama Pengguna**: A text input field with a placeholder 'Masukkan nama pengguna'.
- Email**: A text input field with a placeholder 'Masukkan email'.
- Kata Sandi**: A text input field with a placeholder 'Masukkan kata sandi'.
- Level Pengguna**: A dropdown menu with a placeholder 'Pilih level pengguna'.
- Simpan**: A blue button to save the new user.

m\_level

The screenshot shows the 'Tambah Level' (Add Level) form in the AdminLTE interface. The form is located on the right side of the dashboard, with a sidebar on the left containing navigation links. The form fields are as follows:

- Nama Level**: A text input field with a placeholder 'Masukkan nama level'.
- Deskripsi**: A text input field with a placeholder 'Masukkan deskripsi'.
- Simpan**: A blue button to save the new level.





## 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']);
```

3. Edit pada KategoriController dengan kode:

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



4. Berikut ini sebagai alternatif contoh penulisan validasi dan bisa Anda sesuaikan kata-kata 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`!  
`validate()` secara otomatis mengembalikan respons dengan pesan kesalahan validasi jika validasi gagal, sedangkan `validateWithBag()` memberikan kontrol lebih besar atas respons yang dihasilkan.
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',  
]);  
  
$validated = $request->validate([  
    'kategori_kode' => 'bail|required|unique:m_kategori|max:255',  
    'kategori_nama' => '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

@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 langdirektori, 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.

```
PS C:\laragon\www\PWL_POS> php artisan lang:publish
INFO Language files published successfully.
```

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

<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
```

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

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

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`.

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');
}
```



```
public function rules(): array
{
    return [
        'kategori_kode' => 'required',
        'kategori_nama' => 'required',
    ];
}

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.
  - m\_user

## Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	<a href="#">Ubah</a>   <a href="#">Hapus</a>
2	manager	Manager	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
3	staff	Staff/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
23	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
24	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
25	manager_tiga	Manager 3	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
26	manager22	Manager Dua Dua	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
27	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
31	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
32	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>



AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

User > Tambah

Create User

Nama

Masukkan Nama

Level ID

Masukkan Level ID

Username

Masukkan Username

Password

Masukkan Password

Simpan

- m\_level

## Data Level Pengguna

[+ Tambah Level](#)

ID	Kode Level	Nama Level
1	ADM	Administrator
2	MNG	Manager
3	STF	Staff/Kasir
4	CUS	Pelanggan

AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

Level > Tambah

Create Level

Nama Level

Masukkan Nama

Level Kode

Masukkan Level Kode

Simpan





#### 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_POS> php artisan make:controller POSController --resource
```

```
INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\POSController.php] created successfully.
```

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

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

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

3. Atur pada Models/m\_user

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory; use
Illuminate\Database\Eloquent\Model;

class m_user extends Model
{
    protected $table = "m_user";
    public $timestamps = false;
    protected $primaryKey = 'user_id';

    protected $fillable = [
        'user_id',
        'level_id',
        'username',
        'nama',
        'password',
    ];
}

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class m_user extends Model
{
    use HasFactory;
    protected $table = 'm_user';
    public $timestamps = false;
    protected $primaryKey = 'user_id';

    protected $fillable = [
        'user_id',
        'level_id',
        'username',
        'nama',
        'password'
    ];
}
```



#### 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');
    }
};
```

```
<?php

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()
6. Selanjutnya bukalah `app/Http/Controllers/POSController.php` kemudian ketikkan, kodenya seperti berikut ini:

```
<?php

namespace App\Http\Controllers;

use App\Models\m_user;
use Illuminate\Http\Request;

class POSController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        //fungsi eloquent menampilkan data menggunakan pagination
        $useri = m_user::all(); // Mengambil semua isi tabel
        return view('m_user.index', compact('useri'))->with('i');
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('m_user.create');
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        //melakukan validasi data
        $request->validate([
            'user_id' => 'max 20',
            'username' => 'required',
            'nama' => 'required',
        ]);
        //fungsi eloquent untuk menambah data
    }
}
```



```
m_user::create($request->all());

return redirect()->route('m_user.index')
->with('success', 'user Berhasil Ditambahkan');
}

/**
 * Display the specified resource.
 */
public function show(string $id, m_user $useri)
{
    $useri = m_user::findOrFail($id);
    return view('m_user.show', compact('useri'));
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(string $id)
{
    $useri = m_user::find($id);
    return view('m_user.edit', compact('useri'));
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    $request->validate([
        'username' => 'required',
        'nama' => 'required',
        'password' => 'required',
    ]);
    //fungsi eloquent untuk mengupdate data inputan kita
    m_user::find($id)->update($request->all());
    //jika data berhasil diupdate, akan kembali ke halaman utama
    return redirect()->route('m_user.index')
    ->with('success', 'Data Berhasil Diupdate');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    $useri= m_user::findOrFail($id)->delete();
    return \redirect()->route('m_user.index')

    -> with('success', 'data Berhasil Dihapus');
}
}
```



```
<?php

namespace App\Http\Controllers;

use App\Models\m_user;
use Illuminate\Http\Request;

class POSController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        // Fungsi eloquent menampilkan data menggunakan pagination
        $useri = m_user::all(); // Mengambil semua isi tabel
        return view('m_user.index', compact('useri'))->with('i');
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('m_user.create');
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        // Melakukan validasi data
        $request->validate([
            'user_id' => 'max 20',
            'username' => 'required',
            'nama' => 'required',
        ]);
        // Fungsi eloquent untuk menambah data
        m_user::create($request->all());

        return redirect()->route('m_user.index')
            ->with('success', 'User Berhasil Ditambahkan');
    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        $useri = m_user::findOrFail($id);
        return view('m_user.show', compact('useri'));
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $useri = m_user::find($id);
        return view('m_user.edit', compact('useri'));
    }
}
```



```
/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    $request->validate([
        'username' => 'required',
        'nama' => 'required',
        'password' => 'required',
    ]);
    // Fungsi eloquent untuk mengupdate data inputan kita
    m_user::find($id)->update($request->all());
    // Jika data berhasil diupdate, akan kembali ke halaman utama
    return redirect()->route('m_user.index')
    ->with('success', 'Data Berhasil Diubah');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    $useri= m_user::findOrFail($id)->delete();
    return \redirect()->route('m_user.index')

    -> with('success', 'Data Berhasil Dihapus');
}
}
```





7. Buatlah folder di Resources/Views/**m\_user** dengan beberapa blade dan isian kode berikut:

1. template.blade.php

```
<!DOCTYPE html>
<html>
<head>
<title>CRUD Laravel</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
</head>
<body>
<div class="container">
@yield('content')
</div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>CRUD Laravel</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
</head>
<body>
<div class="container"> @yield('content') </div>
</body>
</html>
```

2. index.blade.php

```
@extends('m_user/template')
@section('content')
<div class="row mt-5 mb-5">
<div class="col-lg-12 margin-tb">
<div class="float-left">
<h2>CRUD user</h2>
</div>
<div class="float-right">
<a class="btn btn-success" href="{{ route('m_user.create') }}"> Input
User</a>
</div>
</div>
</div>
```



```
</div>
</div>
</div>
@if ($message = Session::get('success'))
<div class="alert alert-success">
<p>{{ $message }}</p>
</div>
@endif
<table class="table table-bordered">
<tr>
<th width="20px" class="text-center">User id</th>
<th width="150px" class="text-center">Level id</th>
<th width="200px" class="text-center">username</th>
<th width="200px" class="text-center">nama</th>
<th width="150px" class="text-center">password</th>
</tr>
@foreach ($useri as $m_user)
<tr>

<td>{{ $m_user->user_id }}</td>
<td>{{ $m_user->level_id }}</td>
<td>{{ $m_user->username }}</td>
<td>{{ $m_user->nama }}</td>
<td>{{ $m_user->password }}</td>

<td class="text-center">
<form action="{{ route('m_user.destroy',$m_user->user_id) }}"
method="POST">
<a class="btn btn-info btn-sm" href="{{ route('m_user.show',$m_user->user_id) }}">Show</a>
<a class="btn btn-primary btn-sm" href="{{ route('m_user.edit',$m_user->user_id) }}">Edit</a>
@csrf
@method('DELETE')
<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda yakin ingin menghapus data ini?')">Delete</button>
</form>
</td>
</tr>
@endforeach
</table>
@endsection
```



```
@extends('m_user/template')
@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2>CRUD user</h2>
            </div>
            <div class="float-right">
                <a class="btn btn-success" href="{{ route('m_user.create') }}"> Input User</a>
            </div>
        </div>
    </div>
    @if ($message = Session::get('success'))
        <div class="alert alert-success">
            <p>{{ $message }}</p>
        </div>
    @endif
    <table class="table table-bordered">
        <tr>
            <th width="20px" class="text-center">User id</th>
            <th width="150px" class="text-center">Level id</th>
            <th width="200px" class="text-center">username</th>
            <th width="200px" class="text-center">nama</th>
            <th width="150px" class="text-center">password</th>
        </tr>
        @foreach ($useri as $m_user)
            <tr>
                <td>{{ $m_user->user_id }}</td>
                <td>{{ $m_user->level_id }}</td>
                <td>{{ $m_user->username }}</td>
                <td>{{ $m_user->nama }}</td>
                <td>{{ $m_user->password }}</td>
                <td class="text-center">
                    <form action="{{ route('m_user.destroy', $m_user->user_id) }}" method="POST">
                        <a class="btn btn-info btn-sm" href="{{ route('m_user.show', $m_user->user_id) }}">Show</a>
                        <a class="btn btn-primary btn-sm" href="{{ route('m_user.edit', $m_user->user_id) }}">Edit</a>
                        @csrf
                        @method('DELETE')
                        <button type="submit" class="btn btn-danger btn-sm"
                            onclick="return confirm('Apakah Anda yakin ingin menghapus data ini?')">Delete</button>
                    </form>
                </td>
            </tr>
        @endforeach
    </table>
@endsection
```



### 3. create.blade.php

```
@extends('m_user/template')
@section('content')
<div class="row mt-5 mb-5">
<div class="col-lg-12 margin-tb">
<div class="float-left">
<h2>Membuat Daftar User</h2>
</div>
<div class="float-right">
<a class="btn btn-secondary" href="{{ route('m_user.index') }}">
Kembali</a>
</div>
</div>
</div>
</div>
@if ($errors->any())
<div class="alert alert-danger">
<strong>Ops</strong> Input gagal<br><br>
<ul>
@foreach ($errors->all() as $error)
<li>{{ $error }}</li>
@endforeach
</ul>
</div>
@endif
<form action="{{ route('m_user.store') }}" method="POST">
@csrf

<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Username:</strong>
<input type="text" name="username" class="form-control"
placeholder="Masukkan username"></input>
</div>
</div>

<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>nama:</strong>
<input type="text" name="nama" class="form-control"
placeholder="Masukkan nama"></input>
</div>
</div>
```



```
</div>

<div class="col-xs-12 col-sm-12 col-md-12">
  <div class="form-group">
    <strong>Password:</strong>
    <input type="password" name="password" class="form-control"
placeholder="Masukkan password"></input>
  </div>
</div>

<div class="col-xs-12 col-sm-12 col-md-12 text-center">
<button type="submit" class="btn btn-primary">Submit</button>
</div>
</div>
</form>
@endsection
```

```
@extends('m_user/template')
@section('content')
  <div class="row mt-5 mb-5">
    <div class="col-lg-12 margin-tb">
      <div class="float-left">
        <h2>Membuat Daftar User</h2>
      </div>
      <div class="float-right">
        <a class="btn btn-secondary" href="{{ route('m_user.index') }}"> Kembali</a>
      </div>
    </div>
  </div>
  @if ($errors->any())
    <div class="alert alert-danger">
      <strong>Ops</strong> Input gagal<br><br>
      <ul>
        @foreach ($errors->all() as $error)
          <li>{{ $error }}</li>
        @endforeach
      </ul>
    </div>
  @endif
  <form action="{{ route('m_user.store') }}" method="POST">
    @csrf
    <div class="col-xs-12 col-sm-12 col-md-12">
      <div class="form-group">
        <strong>Username:</strong>
        <input type="text" name="username" class="form-control" placeholder="Masukkan username"></input>
      </div>
    </div>

    <div class="col-xs-12 col-sm-12 col-md-12">
      <div class="form-group">
        <strong>nama:</strong>
        <input type="text" name="nama" class="form-control" placeholder="Masukkan nama"></input>
      </div>
    </div>

    <div class="col-xs-12 col-sm-12 col-md-12">
      <div class="form-group">
        <strong>Password:</strong>
        <input type="password" name="password" class="form-control" placeholder="Masukkan password"></input>
      </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12 text-center">
      <button type="submit" class="btn btn-primary">Submit</button>
    </div>
  </div>
</form>
@endsection
```



#### 4. show.blade.php

```
@extends('m_user/template')
@section('content')
<div class="row mt-5 mb-5">
<div class="col-lg-12 margin-tb">
<div class="float-left">
<h2> Show User</h2>
</div>
<div class="float-right">
<a class="btn btn-secondary" href="{{ route('m_user.index') }}">
Kembali</a>
</div>
</div>
</div>
</div>
<div class="row">
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>User_id:</strong>
{{ $useri->user_id }}
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Level_id:</strong>
{{ $useri->level_id }}
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Username:</strong>
{{ $useri->username }}
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Nama:</strong>
{{ $useri->nama }}
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Password:</strong>
{{ $useri->password }}
</div>
</div>
</div>
@endsection
```





```
@extends('m_user/template')
@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2> Show User</h2>
            </div>
            <div class="float-right">
                <a class="btn btn-secondary" href="{{ route('m_user.index') }}"> Kembali</a>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>User_id:</strong>
                {{ $useri->user_id }}
            </div>
        </div>

        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Level_id:</strong>
                {{ $useri->level_id }}
            </div>
        </div>

        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Username:</strong>
                {{ $useri->username }}
            </div>
        </div>

        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Nama:</strong>
                {{ $useri->nama }}
            </div>
        </div>

        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Password:</strong>
                {{ $useri->password }}
            </div>
        </div>
    </div>
@endsection
```



## 5. edit.blade.php

```
@extends('m_user/template')
@section('content')
<div class="row mt-5 mb-5">
<div class="col-lg-12 margin-tb">
<div class="float-left">
<h2>Edit User</h2>
</div>
<div class="float-right">
<a class="btn btn-secondary" href="{{ route('m_user.index') }}">
Kembali</a>
</div>
</div>
</div>
</div>
@if ($errors->any())
<div class="alert alert-danger">
<strong>Ops!</strong> Error <br><br>
<ul>
@foreach ($errors->all() as $error)
<li>{{ $error }}</li>
@endforeach
</ul>
</div>
@endif
<form action="{{ route('m_user.update', $useri->user_id) }}"
method="POST">
@csrf
@method('PUT')
<div class="row">
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>User_id:</strong>
<input type="text" name="userid" value="{{ $useri->user_id }}"
class="form-control" placeholder="Masukkan user id">
</div>
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Level_id:</strong>
<input type="text" name="levelid" value="{{ $useri->level_id }}"
class="form-control" placeholder="Masukkan level">
</div>
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Username:</strong>
<input type="text" value="{{ $useri->username }}" class="form-
control" name="username" placeholder="Masukkan Nomor username">
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>nama:</strong>
<input type="text" value="{{ $useri->nama }}" name="nama"
class="form-control" placeholder="Masukkan nama"></input>
</div>
</div>
</div>
```



```
<div class="col-xs-12 col-sm-12 col-md-12">
  <div class="form-group">
    <strong>Password:</strong>
    <input type="password" value= "{{ $useri->password
  }}"name="password" class="form-control" placeholder="Masukkan
  password"></input>
  </div>
</div>

<div class="col-xs-12 col-sm-12 col-md-12 text-center">
<button type="submit" class="btn btn-primary">Update</button>
</div>
</div>
</form>
@endsection
```

```
@extends('m_user/template')
@section('content')
  <div class="row mt-5 mb-5">
    <div class="col-lg-12 margin-tb">
      <div class="float-left">
        <h2>Edit User</h2>
      </div>
      <div class="float-right">
        <a class="btn btn-secondary" href="{{ route('m_user.index') }}"> Kembali</a>
      </div>
    </div>
    <div>
      @if ($errors->any())
        <div class="alert alert-danger">
          <strong>Ops!</strong> Error <br><br>
          <ul>
            @foreach ($errors->all() as $error)
              <li>{{ $error }}</li>
            @endforeach
          </ul>
        </div>
      @endif
      <form action="{{ route('m_user.update', $useri->user_id) }}" method="POST">
        @csrf @method('PUT')
        <div class="row">
          <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
              <strong>User_id:</strong>
              <input type="text" name="userid" value="{{ $useri->user_id }}" class="form-control"
                placeholder="Masukkan user id">
            </div>
          </div>
          <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
              <strong>Level_id:</strong>
              <input type="text" name="levelid" value="{{ $useri->level_id }}" class="form-control"
                placeholder="Masukkan level">
            </div>
          </div>
          <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
              <strong>Username:</strong>
              <input type="text" value= "{{ $useri->username }}" class="form- control" name="username"
                placeholder="Masukkan Nomor username">
            </div>
          </div>
          <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
              <strong>nama:</strong>
              <input type="text" value= "{{ $useri->nama }}"name="nama" class="form-control"
                placeholder="Masukkan nama"></input>
            </div>
          </div>
          <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
              <strong>Password:</strong>
              <input type="password" value= "{{ $useri->password }}"name="password" class="form-control"
                placeholder="Masukkan password"></input>
            </div>
          </div>
          <div class="col-xs-12 col-sm-12 col-md-12 text-center">
            <button type="submit" class="btn btn-primary">Update</button>
          </div>
        </div>
      </form>
    </div>
  @endsection
```



Silahkan akses localhost/m\_user

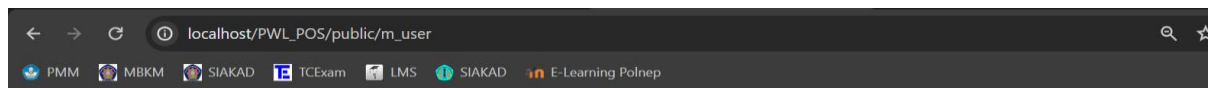


## CRUD user

Input User

Data Berhasil Diupdate

User id	Level id	username	nama	password	
11		pwl	pwl	122345	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a>



## CRUD user

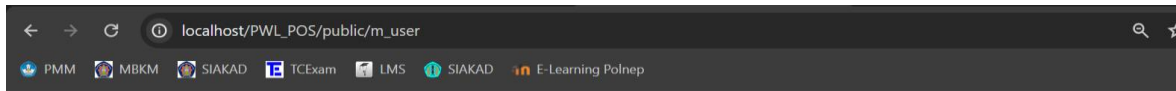
Input User

User id	Level id	username	nama	password	
1	1	admin	Administrator	\$2y\$12\$Q3sMfqwmPOB1V2sza2bQedvyzPJ0dwi1/YdRL8k5KYKAMi9m.dDO	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	2	manager	Manager	\$2y\$12\$hQB8Q6Fhola/O0KLacyXler76u6xj8lvEEFTw7SkilgEdNBnfOWK	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a>



Tugas:

1. Coba tampilkan level\_id pada halaman web tersebut dimana field ini merupakan foreign key



### CRUD user

Input User

User id	Level id	username	nama	password	
1	1	admin	Administrator	\$2y\$12\$Q3sMfxqwmPOB1V2sza2bQedvzPJ0dwi1/YdRL8k5KYKAMi9m.dDO	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$hQBzQ6Fhola/OOKLacyXler76u6xJ8lveEFETw7SkIlgEdNBnfOWK	Show Edit Delete

2. Modifikasi dengan tema/ template kesukaan Anda

### CRUD User

Input User

Daftar User						
User ID	Level Nama	Username	Nama	Password	Aksi	
1	Administrator	admin	Administrator	\$2y\$12\$Q3sMfxqwmPOB1V2sza2bQedvzPJ0dwi1/YdRL8k5KYKAMi9m.dDO	Show Edit Delete	
2	Manager	manager	Manager	\$2y\$12\$hQBzQ6Fhola/OOKLacyXler76u6xJ8lveEFETw7SkIlgEdNBnfOWK	Show Edit Delete	

3. Apa fungsi `$request->validate`, `$error` dan `alert` yang ada pada halaman CRUD tersebut  
`$request->validate` digunakan untuk melakukan validasi data yang diterima dari permintaan HTTP, `$errors` adalah variabel yang berisi pesan kesalahan validasi, dan `<div class="alert">` digunakan untuk menampilkan pesan tersebut kepada pengguna dalam halaman CRUD.

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