

JOBSHEET 6

LAPORAN TUGAS

**Diajukan untuk Memenuhi Tugas
Pemograman Web Lanjut**

Jurusan:

TEKNOLOGI INFORMASI

Program Studi:

TEKNIK INFORMATIKA

Disusun oleh

1. Afrizal Dwi Septian (2241720122) / 01 / 2H



**POLITEKNIK NEGERI MALANG
TEKNIK INFORMATIKA**

2023



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 pertemuan 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.

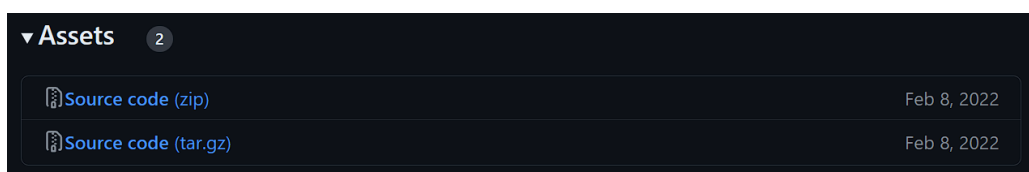
Contoh dari server side scripting antara lain:

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

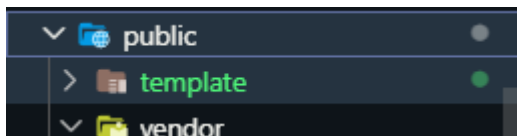
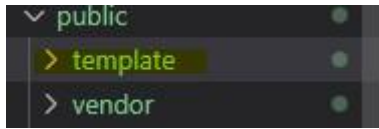
Client side validation lebih dilakukan disisi browser dan bukan untuk tujuan keamanan, tetapi lebih ke kenyamanan pengguna. Sedangkan *server side validation* dilakukan di sisi server dengan tujuan keamanan dengan *filter* semua *request* yang masuk sebelum akhirnya diproses lanjutan.

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**, folder template pindahkan ke *laragon/www/PWL_POS/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>
```

```
<!-- 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') }}">

<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>

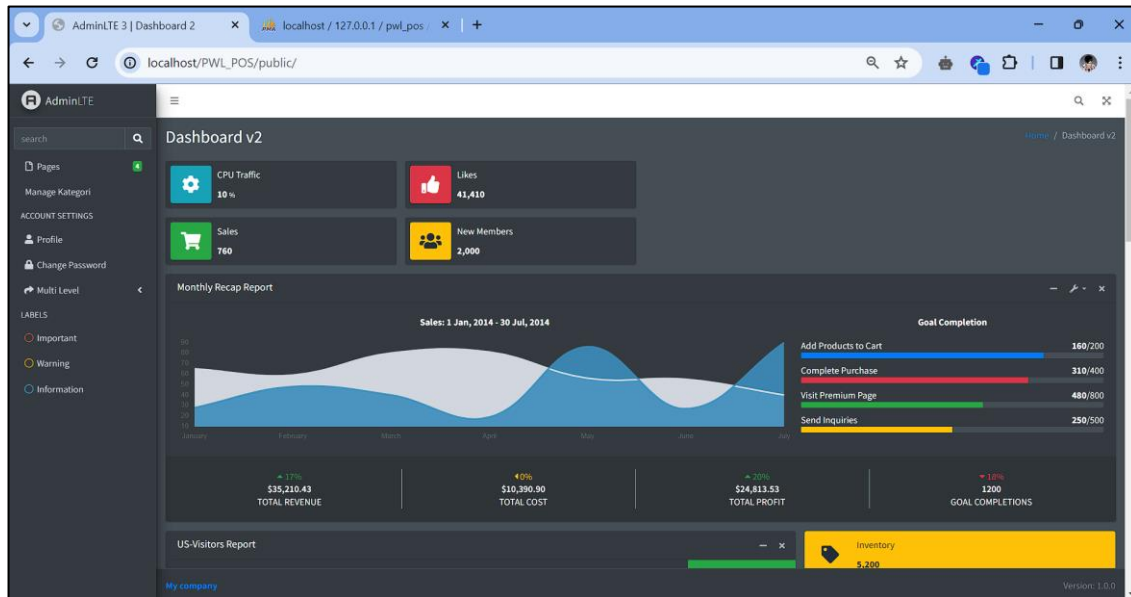
<!-- AdminLTE for demo purposes -->
<script src="{{ asset ('template/dist/js/demo.js')}}"></script>
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script src=""></script>
<script src="{{ asset ('template/dist/js/pages/dashboard2.js')}}"></script>
```

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

```
<img src="{{ asset('template/dist/img/user1-128x128.jpg')}}"
```

5. Jalankan pada browser





*note : template ini load untuk semuanya, hapus jika tidak diperlukan. Visual 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



General Elements

Text	Text Disabled
<input type="text" value="Enter ..."/>	<input type="text" value="Enter ..."/>
Textarea	Textarea Disabled
<div><div>Enter ...</div><div></div></div>	<div><div>Enter ...</div><div></div></div>
✓ Input with success	
<div><div>Enter ...</div><div></div></div>	

General Elements

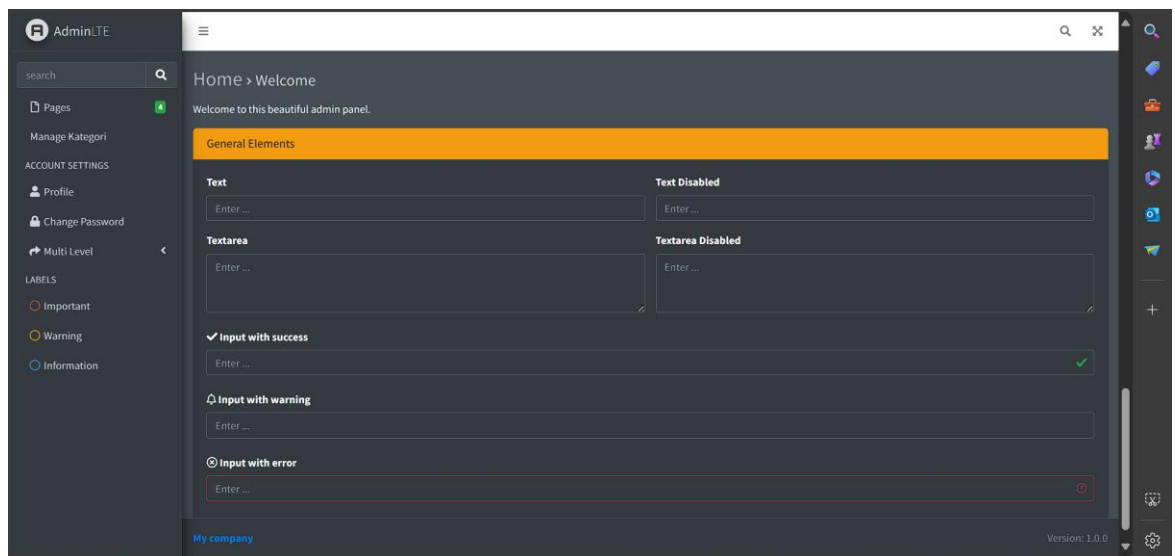
Text	Text Disabled
<input type="text" value="Enter ..."/>	<input type="text" value="Enter ..."/>
Textarea	Textarea Disabled
<div><div>Enter ...</div><div></div></div>	<div><div>Enter ...</div><div></div></div>
✓ Input with success	
<div><div>Enter ...</div><div></div></div>	



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

```
15 <body class="hold-transition sidebar-mini">
16 <div class="wrapper">
831 <div class="content-wrapper">
850 <section class="content">
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>
1175 <div class="col-sm-6">
1176 <div class="form-group">
1177 <label>Text Disabled</label>
1178 <input type="text" class="form-control" placeholder="Enter ..." disabled>
1179 </div>
1180 </div>
1181 </div>
1182 <div class="row">
1183 <div class="col-sm-6">
1184 <!-- textarea -->
1185 <div class="form-group">
1186 <label>Textarea</label>
```

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

User

```
1 @extends('adminlte::page')
2
3 @section('title', 'Dashboard')
4
5 @section('content_header')
6     <h1>form User</h1>
7 @stop
8
9 @section('content')
10     <!-- general form elements disabled -->
11     <div class="card card-warning">
12         <div class="card-header">
13             <h3 class="card-title">Add User</h3>
14         </div>
15         <!-- /.card-header -->
16         <div class="card-body">
17             <div class="row">
18                 <div class="col">
19                     <div class="form-group">
20                         <label>Select Level</label>
21                         <select class="form-control">
22                             <option>Admin 1</option>
23                             <option>Admin 2</option>
24                             <option>Admin 3</option>
25                             <option>Admin 4</option>
26                             <option>Admin 5</option>
27                         </select>
28                     </div>
29                 </div>
30             </div>
31             <div class="row">
32                 <div class="col">
33                     <div class="form-group">
34                         <label>Level</label>
35                         <input type="text" class="form-control" placeholder="Masukkan Username">
36                     </div>
37                 </div>
38             </div>
39             <div class="row">
40                 <div class="col">
41                     <div class="form-group">
42                         <label>Name</label>
43                         <input type="text" class="form-control" placeholder="Masukkan Username">
44                     </div>
45                 </div>
46             </div>
47             <div class="row">
48                 <div class="col">
49                     <div class="form-group">
50                         <label>Password</label>
51                         <input type="password" class="form-control" placeholder="Masukkan Password">
52                     </div>
53                 </div>
54             </div>
55             <div class="row">
56                 <div class="col">
57                     <div class="form-group">
58                         <button type="button" class="btn btn-success">Tambah</button>
59                     </div>
60                 </div>
61             </div>
62         </div>
63     </div>
64 </div>
```

Level

[illegible]

The screenshot displays the AdminLTE web application. On the left is a dark sidebar with a navigation menu. The main content area is white and titled 'Form Level'. It features a form with a yellow 'Add level' button at the top, followed by a 'Level Code' input field, a 'Level Name' input field, and a green 'ADD' button at the bottom.



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

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`!

Jawab: perbedaannya adalah Metode `validate` merupakan metode bawaan laravel untuk memvalidasi inputan, jika validasi gagal maka pengguna akan secara otomatis dikembalikan ke halaman sebelumnya dengan pesan kesalahan yang sesuai. Sedangkan metode `validateWithBag` memberikan fleksibilitas tambahan dengan memungkinkan pengguna untuk menentukan message bag validation yang akan digunakan. Ini memungkinkan pengguna untuk menyesuaikan dimana pesan validasi akan disimpan.

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',  
]);
```



```
public function store(Request $request)
{
    $validated = $request->validate([
        'kategori_kode' => 'bail|required|max:3|unique:m_kategori,kategori_kode',
        'kategori_nama' => 'required',
    ]);

    // Simpan data kategori
    KategoriModel::create([
        'kategori_kode' => $request->kategori_kode,
        'kategori_nama' => $request->kategori_nama,
    ]);

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

Jawab: Yang Terjadi adalah jika validasi pada kodeKategori gagal karena tidak memenuhi required atau unique maka validasi untuk namaKategori tidak akan dilakukan, kondisi ini memungkinkan pengguna untuk memberhentikan validasi awal jika terdapat kondisi yang tidak terpenuhi.

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
```

```
{{-- Content body: main page content --}}
@section('content')
    @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.

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
```

```
<input type="text" class="form-control @error('kategori_kode') is-invalid @enderror" name="kategori_kode" placeholder="Untuk Makanan, contoh : MKN" >

@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
```

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

```
INFO Request [C:\laragon\www\PWL_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(Request $request):RedirectResponse
{
    $validated = $request->validated();
    $validated = $request->safe()->only(['kategori_kode', 'kategori_nama']);
    $validated = $request->safe()->except(['kategori_kode', 'kategori_nama']);

    return redirect(to: '/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.
 - Validasi User



```
app > Http > Controllers > UserController.php > UserController
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Database\Eloquent\ModelNotFoundException;
9 use Illuminate\Http\RedirectResponse;
10 use App\DataTables\UserDataTable;
11
12 class UserController extends Controller
13 {
14     public function index(UserDataTable $dataTable)
15     {
16         return $dataTable->render('view: user.index');
17     }
18
19     public function create()
20     {
21         return view('view: user.create');
22     }
23
24     public function store(Request $request): RedirectResponse
25     {
26         //dd($request->all());
27         $validated = $request->validate($rules: [
28             'level_id' => 'bail|required',
29             'username' => 'required',
30             'name' => 'required',
31             'password' => 'required',
32         ]);
33
34         $password = Hash::make($validated['password']);
35
36         UserModel::create([
37             'level_id' => $validated['level_id'],
38             'username' => $validated['username'],
39             'name' => $validated['name'],
40             'password' => Hash::make($validated['password']),
41         ]);
42
43         return redirect($to: '/user_tes');
44     }
45
46     public function edit($id){
47         $user = UserModel::find($id);
48         return view('view: user.edit', ['data' => $user]);
49     }
50
51     public function update(Request $request, $id){
52         $user = UserModel::find($id);
53         $user->level_id = $request->level_id;
54         $user->username = $request->username;
55         $user->name = $request->name;
56         $user->password = $request->password;
57         $user->save();
58         return redirect($to: '/user');
59     }
60
61     public function delete($id) {
62         UserModel::find($id)->delete();
63
64         return redirect($to: '/user');
65     }
66 }
67
```

The screenshot shows the 'AdminLTE' interface with a sidebar menu on the left. The main content area is titled 'user > Create'. At the top, there is a red alert box with the following messages:

- The username field is required.
- The name field is required.
- The password field is required.

Below the alert box is a form titled 'Buat User'. The form contains the following fields:

- ID User:** A text input field with the value 'Admin 1'.
- Username:** A text input field with the placeholder 'Masukkan Username'. It has a red border and a red error message: 'The username field is required.'
- Nama:** A text input field with the placeholder 'Masukkan Nama'.
- Password:** A text input field with the placeholder 'Masukkan Password'. It has a red border and a red error message: 'The password field is required.'

At the bottom of the form is a blue 'Submit' button.



- Validasi Level

```
app > Http > Controllers > LevelController.php > LevelController > store
11 class LevelController extends Controller
12 {
13     public function index(LevelDataTable $dataTable)
14     {
15         return $dataTable->render(view: 'level.index');
16     }
17
18     public function create()
19     {
20         return view(view: 'level.create');
21     }
22
23     public function store(Request $request): RedirectResponse
24     {
25         //dd($request->all());
26         $validated = $request->validate([
27             'kodeLevel' => 'bail|required|unique:level,level_kode',
28             'namaLevel' => 'required',
29         ]);
30
31         LevelModel::create([
32             'level_kode' => $validated['kodeLevel'],
33             'level_nama' => $validated['namaLevel'],
34         ]);
35
36         return redirect(to: '/level');
37     }
38
39     public function edit($id){
40         $level = LevelModel::find($id);
41         return view(view: 'level.edit', data: ['data' => $level]);
42     }
43
44     public function update(Request $request, $id){
45         $level = LevelModel::find($id);
46         $level->level_kode = $request->kodeLevel;
47         $level->level_nama = $request->namaLevel;
48         $level->save();
49         return redirect(to: '/level');
50     }
51
52     public function destroy($id) {
53         LevelModel::find($id)->delete();
54
55         return redirect(to: '/level');
56     }
57 }
```

Admin UI

level > Create

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

Buat level

Kode Level

Masukkan Kode Level

The kode level field is required.

Nama level

Masukkan Nama Level

The nama level field is required.

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
```

```
C:\laragon\www\PMI_POS>php artisan make:controller POScontroller --resource
```

```
INFO Controller [C:\laragon\www\PMI_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(name: 'm_user', controller: POSController::class);
```

3. Atur pada Models/m_user

```
app > Models > m_user.php > ...
1  <?php
2  namespace App\Models;
3  use Illuminate\Database\Eloquent\Factories\HasFactory;
4  use Illuminate\Database\Eloquent\Model;
5  class m_user extends Model
6  {
7      protected $table = "m_user";
8      public $timestamps = false;
9      protected $primaryKey = 'user_id';
10     protected $fillable = [
11         'user_id',
12         'level_id',
13         'username',    'nama',
14         'password',
15     ];
16
17 }
```

4. Atur pada Migration/m_user_table 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()



5. Selanjutnya bukalah app/Http/Controllers/POSController.php kemudian ketikkan, kodenya seperti berikut ini:

```
app > http > controllers > POSController.php > create
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\m_user;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8 use App\Http\Controllers\User\LevelController;
9 use App\Models\levelModel;
10
11 class POSController extends Controller
12 {
13     /**
14      * Display a listing of the resource.
15      */
16     public function index()
17     {
18         //fungsi eloquent menampilkan data menggunakan pagination
19         $user = m_user::all();
20         // Mengambil semua isi tabel
21         return view('m_user.index', ['data' => compact('user')]);
22     }
23
24     /**
25      * Show the form for creating a new resource.
26      */
27     public function create()
28     {
29         // return view('m_user.create');
30         $level = levelModel::all(); //mendapatkan data dari tabel level
31         return view('m_user.create', ['data' => compact('level')]);
32     }
33
34     /**
35      * Store a newly created resource in storage.
36      */
37     public function store(Request $request)
38     {
39         //validasi data
40         $request->validate([
41             'user_id' => 'max:20',
42             'username' => 'required',
43             'nama' => 'required',
44         ]);
45         //fungsi eloquent untuk menambah data
46     }
47 }
```

6. Buatlah folder di Resources/Views/m_user dengan beberapa blade dan isian kode berikut:

1. template.blade.php

```
resources > views > layouts > m_user > template.blade.php > html
1 <!DOCTYPE html>
2 <html lang="en" data-bs-theme="dark">
3
4 <head>
5     <title>CRUD Laravel</title>
6     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
7     <style>
8         .container {
9             margin-top: 50px;
10         }
11     </style>
12 </head>
13
14 <body>
15     <div class="container">
16         @yield('content')
17     </div>
18     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
19 </body>
20
21
22 </html>
```

2.index.blade.php

```
resources > views > layouts > m_user > index.blade.php > html
1 @extends('m_user/template')
2 @section('content')
3     <div class="row mt-5 mb-5">
4         <div class="col-12">
5             <div class="float-left">
6                 <h2>CRUD user</h2>
7             </div>
8             <div class="float-right">
9                 <a href="{{ route('m_user.create') }}"> Input User</a>
10            </div>
11        </div>
12        @if ($message = Session::get('success'))
13            <div class="alert alert-success">
14                {{ $message }}</div>
15        @endif
16        <table class="table table-bordered">
17            <thead>
18                <tr>
19                    <th>id</th>
20                    <th>user_id</th>
21                    <th>username</th>
22                    <th>nama</th>
23                    <th>action</th>
24                </tr>
25            </thead>
26            <tbody>
27                @foreach ($user as $user)
28                    <tr>
29                        <td>{{ $user->id }}</td>
30                        <td>{{ $user->user_id }}</td>
31                        <td>{{ $user->username }}</td>
32                        <td>{{ $user->nama }}</td>
33                        <td>
34                            <a href="{{ route('m_user.destroy', $user->id) }}"> Hapus</a>
35                            <a href="{{ route('m_user.edit', $user->id) }}"> Edit</a>
36                            <a href="{{ route('m_user.create') }}"> Input User</a>
37                        </td>
38                    </tr>
39                @endforeach
40            </tbody>
41        </table>
42    </div>
43 @endsection
```

[illegible]

```

1 <?extends("w_user/register")> <section("content")>
2   <div class="row md-5">
3     <div class="col-lg-12 margin-top">
4       <div class="float-left">
5         <div Show Error</div>
6       </div>
7       <div class="float-right">
8         <div class="btn btn-secondary" href="{ route('w_user.index') }">
9           <em>Emball</em></div>
10      </div>
11    </div>
12  </div>
13  <div class="row">
14    <div class="col-xs-12 col-sm-12 col-md-12">
15      <div class="form-group">
16        <strong>User Id</strong>
17        {{ $user->user_id }}
18      </div>
19    </div>
20    <div class="col-xs-12 col-sm-12 col-md-12">
21      <div class="form-group">
22        <strong>User Level</strong>
23        {{ $user->level_id }}
24      </div>
25    </div>
26    <div class="col-xs-12 col-sm-12 col-md-12">
27      <div class="form-group">
28        <strong>User Name</strong>
29        {{ $user->username }}
30      </div>
31    </div>
32    <div class="col-xs-12 col-sm-12 col-md-12">
33      <div class="form-group">
34        <strong>User Email</strong>
35        {{ $user->email }}
36      </div>
37    </div>
38    <div class="col-xs-12 col-sm-12 col-md-12">
39      <div class="form-group">
40        <strong>User Password</strong>
41        {{ $user->password }}
42      </div>
43    </div>
44  </div>
45  <div class="row md-5">
46    <div class="col-lg-12 margin-top">
47      <div class="float-left">
48        <div Show Error</div>
49      </div>
50      <div class="float-right">
51        <div class="btn btn-secondary" href="{ route('w_user.index') }">
52          <em>Emball</em></div>
53      </div>
54    </div>
55  </div>
56  </div>
57  </div>
58  </div>
59  </div>
60  </div>
61  </div>
62  </div>
63  </div>
64  </div>
65  </div>
66  </div>
67  </div>
68  </div>
69  </div>
70  </div>
71  </div>
72  </div>
73  </div>
74  </div>
75  </div>
76  </div>
77  </div>
78  </div>
79  </div>
80  </div>
81  </div>
82  </div>
83  </div>
84  </div>
85  </div>
86  </div>
87  </div>
88  </div>
89  </div>
90  </div>
91  </div>
92  </div>
93  </div>
94  </div>
95  </div>
96  </div>
97  </div>
98  </div>
99  </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 </div>
132 </div>
133 </div>
134 </div>
135 </div>
136 </div>
137 </div>
138 </div>
139 </div>
140 </div>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>
185 </div>
186 </div>
187 </div>
188 </div>
189 </div>
190 </div>
191 </div>
192 </div>
193 </div>
194 </div>
195 </div>
196 </div>
197 </div>
198 </div>
199 </div>
200 </div>
201 </div>
202 </div>
203 </div>
204 </div>
205 </div>
206 </div>
207 </div>
208 </div>
209 </div>
210 </div>
211 </div>
212 </div>
213 </div>
214 </div>
215 </div>
216 </div>
217 </div>
218 </div>
219 </div>
220 </div>
221 </div>
222 </div>
223 </div>
224 </div>
225 </div>
226 </div>
227 </div>
228 </div>
229 </div>
230 </div>
231 </div>
232 </div>
233 </div>
234 </div>
235 </div>
236 </div>
237 </div>
238 </div>
239 </div>
240 </div>
241 </div>
242 </div>
243 </div>
244 </div>
245 </div>
246 </div>
247 </div>
248 </div>
249 </div>
250 </div>
251 </div>
252 </div>
253 </div>
254 </div>
255 </div>
256 </div>
257 </div>
258 </div>
259 </div>
260 </div>
261 </div>
262 </div>
263 </div>
264 </div>
265 </div>
266 </div>
267 </div>
268 </div>
269 </div>
270 </div>
271 </div>
272 </div>
273 </div>
274 </div>
275 </div>
276 </div>
277 </div>
278 </div>
279 </div>
280 </div>
281 </div>
282 </div>
283 </div>
284 </div>
285 </div>
286 </div>
287 </div>
288 </div>
289 </div>
290 </div>
291 </div>
292 </div>
293 </div>
294 </div>
295 </div>
296 </div>
297 </div>
298 </div>
299 </div>
300 </div>
301 </div>
302 </div>
303 </div>
304 </div>
305 </div>
306 </div>
307 </div>
308 </div>
309 </div>
310 </div>
311 </div>
312 </div>
313 </div>
314 </div>
315 </div>
316 </div>
317 </div>
318 </div>
319 </div>
320 </div>
321 </div>
322 </div>
323 </div>
324 </div>
325 </div>
326 </div>
327 </div>
328 </div>
329 </div>
330 </div>
331 </div>
332 </div>
333 </div>
334 </div>
335 </div>
336 </div>
337 </div>
338 </div>
339 </div>
340 </div>
341 </div>
342 </div>
343 </div>
344 </div>
345 </div>
346 </div>
347 </div>
348 </div>
349 </div>
350 </div>
351 </div>
352 </div>
353 </div>
354 </div>
355 </div>
356 </div>
357 </div>
358 </div>
359 </div>
360 </div>
361 </div>
362 </div>
363 </div>
364 </div>
365 </div>
366 </div>
367 </div>
368 </div>
369 </div>
370 </div>
371 </div>
372 </div>
373 </div>
374 </div>
375 </div>
376 </div>
377 </div>
378 </div>
379 </div>
380 </div>

```



```

1 <template("m_user/register")>
2 <html><html>
3   <div class="row mt-5 mb-5">
4     <div class="col-lg-12 margin-top">
5       <div class="float-left">
6         <h2Edit User/>
7       </div>
8       <div class="float-right">
9         <a class="btn btn-secondary href="{ route('m_user.index') }">
10           <small>Kembali/>
11         </a>
12       </div>
13     </div>
14     <div {errors.any()}>
15       <div class="alert alert-danger">
16         <strong></strong> Error </div>
17       <div>
18         <foreach {errors.all()} as Error>
19           <div {Error}></div>
20         </foreach>
21       </div>
22     </div>
23   </div>
24   <div>
25     <form action="{ route('m_user.update', $user->id) }" method="POST"> </div>
26     <method("PUT")>
27       <div class="row">
28         <div class="col-xs-12 col-sm-12 col-md-12">
29           <div class="form-group">
30             <strong>id</strong>
31             <input type="text" name="user_id" value="{ $user->id }" class="form-control">
32             <placeholder>"Masukkan user id">
33           </div>
34         </div>
35       </div>
36       <div class="col-xs-12 col-sm-12 col-md-12">
37         <div class="form-group">
38           <strong>level_id</strong>
39           <input type="text" name="level_id" value="{ $user->level_id }" class="form-control">
40           <placeholder>"Masukkan level">
41         </div>
42       </div>
43       <div class="col-xs-12 col-sm-12 col-md-12">
44         <div class="form-group">
45           <strong>nama</strong>
46           <input type="text" value="{ $user->nama }" class="form-control" name="username">
47           <placeholder>"Masukkan Nomor username">
48         </div>
49       </div>
50     </div>
51     <div class="col-xs-12 col-sm-12 col-md-12">
52       <div class="form-group">
53         <strong>nama</strong>
54       </div>
55     </div>

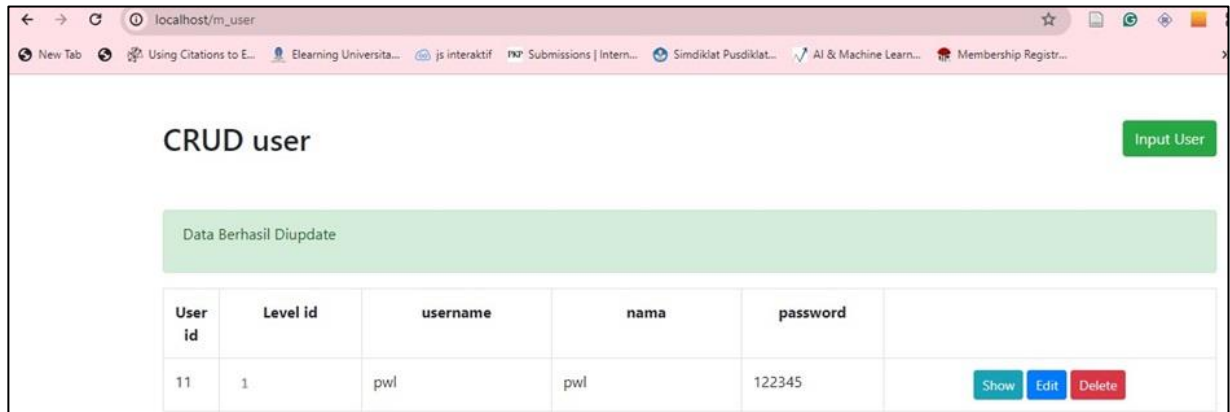
```

localhost/m_user

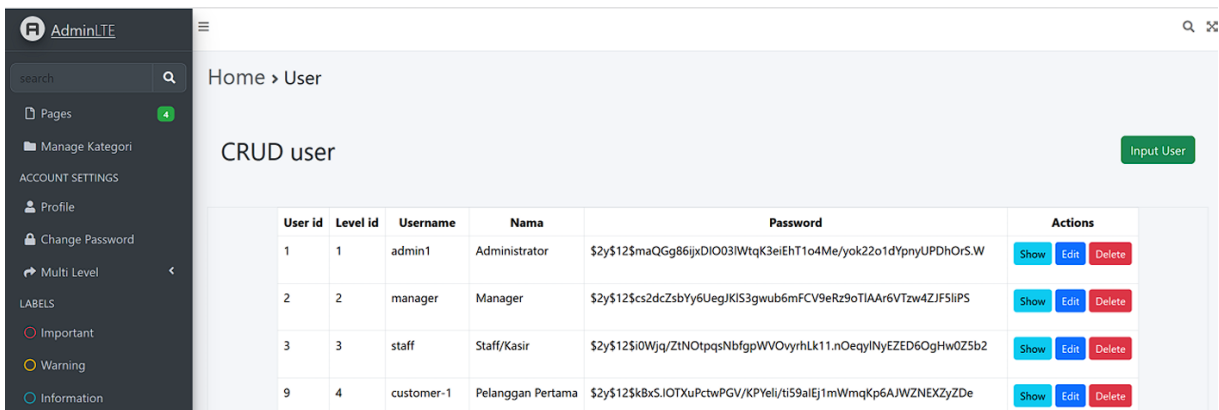
Input User

User id	Level id	username	nama	password	
11		pwl	pwl	122345	<div>ShowEditDelete</div>

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



2. Modifikasi dengan tema/ template kesukaan Anda



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

Jawab: Fungsi \$request->validate adalah metode bawaan laravel yang digunakan untuk validasi input dari pengguna sesuai dengan aturan yang telah ditentukan sebelumnya, fungsi \$error adalah untuk menampilkan pesan dari error yang dihasilkan validasi, dan fungsi alert adalah untuk menampilkan pesan error ketika pengguna melakukan kesalahan saat penginputan.

*** Sekian, dan selamat belajar ***