

Jobsheet 06: Template Form (AdminLTE), Server Validation, Client Validation, CRUD

Dosen Pengampu Mata Kuliah :

M. Hasyim Ratsanjani, S.Kom., M.Kom.



Dibuat Oleh :

Daffa Yudisa Akbar : TI-2H / 10 : 2241720008

Politeknik Negeri Malang

Jurusan Teknologi Informasi

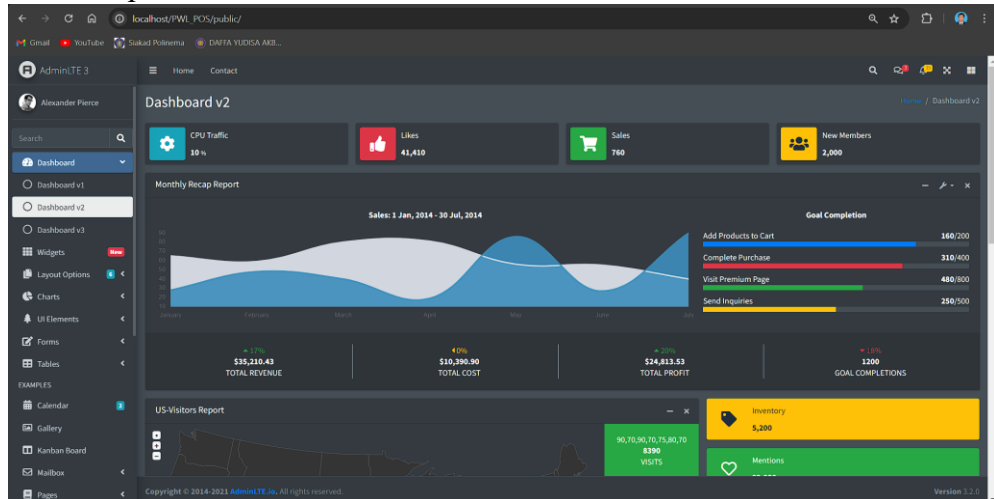
D-IV Teknik Informatika

2024

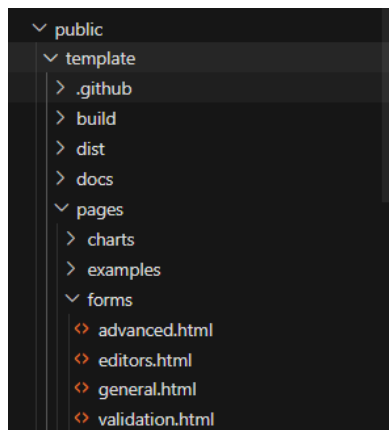
A. Template Form (AdminLTE)

No.	Langkah Praktikum
1.	<p>Penggunaan template AdminLTE Bootstrap Admin Dashboard Template dengan langkah pertama akses https://adminlte.io/ , lalu klik download pada source code (zip)</p> 
2.	<p>Lakukan extract All dan ubah nama folder AdminLTE-3.2.0 menjadi template, folder template pindahkan ke laragon/www/PWL_2024/public. Buka folder public pada VSCode maka akan tampil folder baru bernama template</p> 
3.	<p>Copy isi public/template/index2.html pada welcome.blade.php lakukan edit agar dapat di load, edit href = "{{ asset ('template/.....') }}" dan src= " {{ asset ('template/.....') }}" sesuai gambar berikut</p> 
4.	<p>Scroll ke bagian paling bawah dan sesuaikan dengan kode berikut</p> 

5. Jalankan pada browser



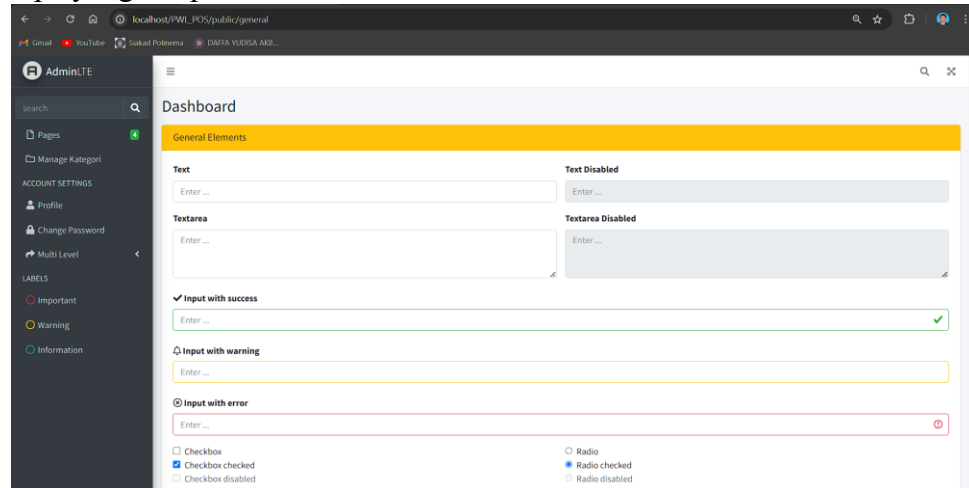
6. Pada bagian menu terdapat pilihan Form kita khususkan dulu pada Form yang disediakan oleh AdminLTE



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

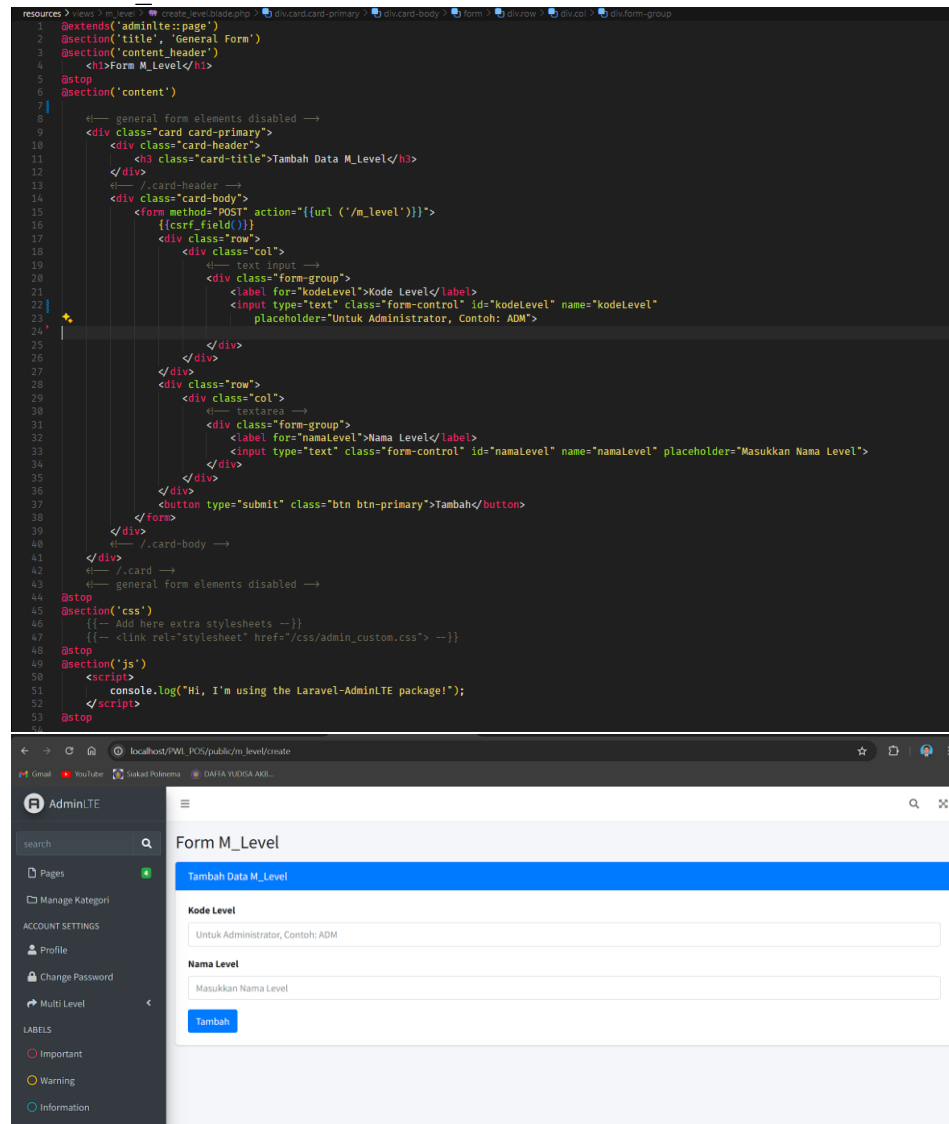
```
<html lang="en">
<body class="hold-transition sidebar-mini">
<div class="wrapper">
  <div class="content-wrapper">
    <section class="content">
      <!-- general form elements disabled -->
      <div class="card card-warning">
        <div class="card-header">
          <h3 class="card-title">General Elements</h3>
        </div>
        <div class="card-body">
          <form>
            <div class="row">
              <div class="col-sm-6">
                <!-- text input -->
                <div class="form-group">
                  <label>Text</label>
                  <input type="text" class="form-control" placeholder="Enter ..." />
                </div>
              </div>
              <div class="col-sm-6">
                <div class="form-group">
                  <label>Text Disabled</label>
                  <input type="text" class="form-control" placeholder="Enter ..." disabled />
                </div>
              </div>
            </div>
            <div class="row">
              <div class="col-sm-6">
                <!-- textarea -->
                <div class="form-group">
                  <label>Textarea</label>
                  <textarea class="form-control" rows="3" placeholder="Enter ..." />
                </div>
              </div>
              <div class="col-sm-6">
                <div class="form-group">
                  <label>Textarea Disabled</label>
                  <textarea class="form-control" rows="3" placeholder="Enter ..." disabled />
                </div>
              </div>
            </div>
          </form>
        </div>
      </div>
    </section>
  </div>
</div>
```

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_Level



• M_User

```

resources > views > user_tes > create_user.blade.php > div.card.card-primary > div.card-body > form > div.row > div.col > div.form-group > input#username.form-control
1 @extends('adminlte::page')
2 @section('title', 'Form User')
3 @section('content_header')
4 <h1>Form M_User</h1>
5 @stop
6 @section('content')
7
8 <!-- general form elements disabled -->
9 <div class="card card-primary">
10 <div class="card-header">
11 <h3 class="card-title">Tambah Data M_User</h3>
12 </div>
13 <!-- /.card-header -->
14 <div class="card-body">
15 <form method="POST" action="{{url ('/user_tes')}}">
16 {{csrf_field()}}
17 <div class="row">
18 <div class="col">
19 <!-- select -->
20 <div class="form-group">
21 <label>Level ID</label>
22 <select class="form-control" name="level_id">
23 <option value=1>Administrator</option>
24 <option value=2>Manager</option>
25 <option value=3>Staff/Kasir</option>
26 <option value=4>Customer</option>
27 </select>
28 </div>
29 </div>
30 </div>
31 <div class="row">
32 <div class="col">
33 <!-- text input -->
34 <div class="form-group">
35 <label>Username</label>
36 <input type="text" class="form-control" placeholder="Masukkan Username" name="username" id="username">
37 </div>
38 </div>
39 </div>
40 <div class="row">
41 <div class="col">
42 <!-- textarea -->
43 <div class="form-group">
44 <label>Nama</label>
45 <input type="text" class="form-control" placeholder="Masukkan Nama" name="name" id="name">
46 </div>
47 </div>
48 </div>
49 <div class="row">
50 <div class="col">
51 <!-- textarea -->
52 <div class="form-group">
53 <label>Password</label>
54 <input type="password" class="form-control" placeholder="Masukkan Password" name="password" id="password">
55 </div>
56 </div>
57 </div>
58 <button type="submit" class="btn btn-primary">Tambah</button>
59 </form>
60 </div>
61 <!-- /.card-body -->
62 </div>
63 <!-- /.card -->
64 <!-- general form elements disabled -->
65 @stop
66 @section('css')
67 <!-- Add here extra stylesheets -->
68 <!-- <link rel="stylesheet" href="/css/admin_custom.css" --> -->
69 @stop
70 @section('js')
71 <script>
72 console.log("Hi, I'm using the Laravel-AdminLTE package!");
73 </script>
74 @stop
75
76

```

localhost/PWL_POJ/public/user_tes/create

AdminLTE

search

Pages

Manage Kategori

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

Form M_User

Tambah Data M_User

Level ID

Administrator

Username

Masukkan Username

Nama

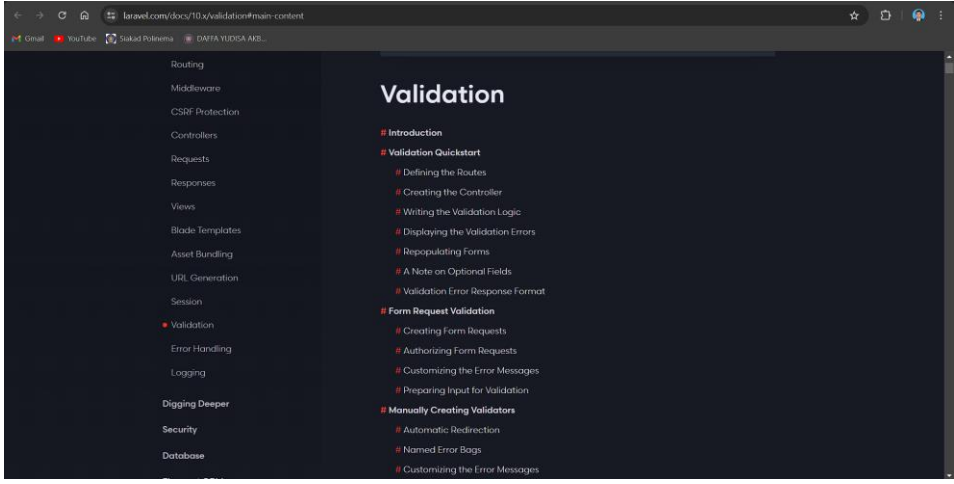
Masukkan Nama

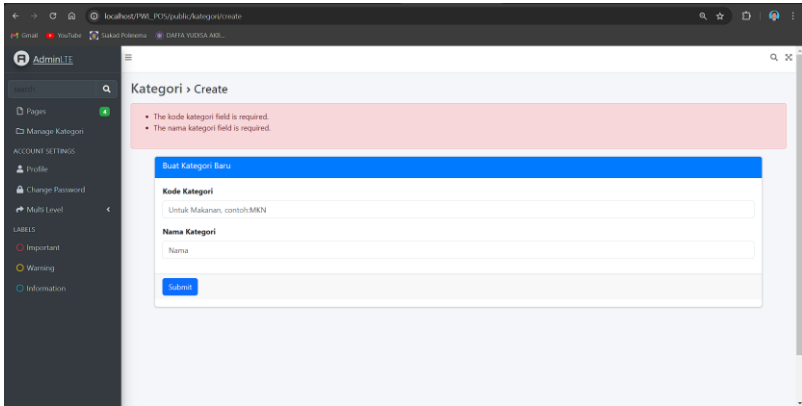
Password

Masukkan Password

Tambah

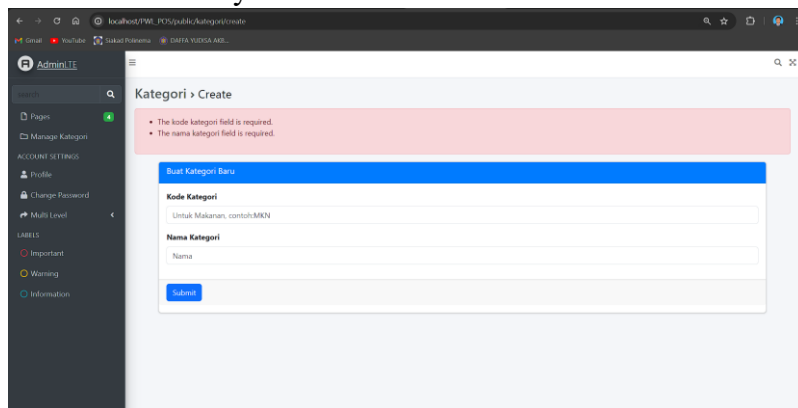
B. Validasi Pada Server

No.	Langkah Praktikum
1.	<p>Buka kembali file tugas di jobsheet 05, klik dan baca documentation ini Validation - Laravel 10.x - The PHP Framework For Web Artisans</p> 
2.	<p>Pastikan sudah terdefiniskan route pada web.php :</p> <pre data-bbox="331 925 1289 1010"> 41 Route::get('/kategori/create', [KategoriController::class, 'create']); 42 Route::post('/kategori', [KategoriController::class, 'store']); 43 Route::get('/kategori/edit/{id}', [KategoriController::class, 'edit']); </pre>
3.	<p>Edit pada KategoriKontroller dengan kode:</p> <pre data-bbox="331 1077 1289 1760"> class KategoriController extends Controller { public function index(KategoriDataTable \$dataTable) { return \$dataTable->render('kategori.index'); } public function create(): View { return view('kategori.create'); } public function store(Request \$request): RedirectResponse { //dd(\$request->all()); // KategoriModel::create \$validated = \$request->validate(['kategori_kode' => 'required', 'kategori_nama' => 'required',]); return redirect('/kategori'); } } </pre>
4.	<p>Berikut ini sebagai alternatif contoh penulisan validasi dan bisa Anda sesuaikan katakata dengan apa yang sedang Anda kerjakan</p>
5.	<p>Tulis perbedaan penggunaan validate dengan validateWithBag!</p> <ul style="list-style-type: none"> Perbedaan penggunaan validate dengan validateWithBag adalah jika validate digunakan untuk melakukan validasi secara global sedangkan validateWithBag digunakan ketika ada beberapa form di halaman yang

	sama dan pesan kesalahan perlu dipisahkan dengan jelas untuk setiap form tersebut
6.	<p>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?</p> <pre> public function create() { return view('kategori.create'); } public function store(Request \$request): RedirectResponse { //dd(\$request->all()); // KategoriModel::create \$validated = \$request->validate(['kodeKategori' => 'bail required max:3 unique:m_kategori,kategori_kode', 'namaKategori' => 'required',]); KategoriModel::create(['kategori_kode' => \$validated['kodeKategori'], 'kategori_nama' => \$validated['namaKategori'],]); return redirect('/kategori'); } </pre>  <ul style="list-style-type: none"> • Fungsi dari rule bail dalam controller adalah untuk menghentikan proses validasi segera setelah aturan validasi yang ditetapkan di dalamnya gagal. Dengan menggunakan bail, jika validasi untuk 'kodeKategori' gagal, proses validasi akan dihentikan dan tidak akan mengevaluasi aturan validasi untuk 'namaKategori'. Ini bertujuan untuk menghindari validasi yang tidak perlu jika input 'kodeKategori' tidak memenuhi syarat yang ditentukan (required : tidak boleh kosong, max 3: maksimum input adalah 3 karakter, unique:m_kategori,kategori_kode: Memastikan bahwa nilai input untuk harus unik dalam tabel m_kategori di kolom kategori_kode).
7.	Pada view/create.blade.php tambahkan code berikut agar ketika validasi gagal, kita dapat menampilkan pesan kesalahan dalam tampilan:

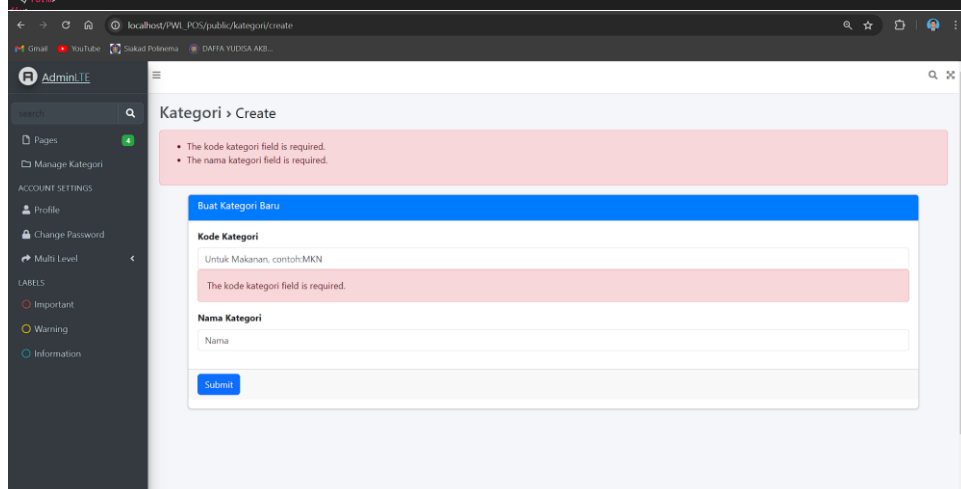
```
resources > views > kategori > create.blade.php > ...
1 @extends('layouts.app')
2 {{-- Customize layout sections --}}
3 @section('subtitle', 'Kategori')
4 @section('content_header_title', 'Kategori')
5 @section('content_header_subtitle', 'Create')
6 {{-- Content body: main page content --}}
7 @section('content')
8     @if ($errors->any())
9         <div class="alert alert-danger">
10             <ul>
11                 @foreach ($errors->all() as $error)
12                     <li>{{ $error }}</li>
13                 @endforeach
14             </ul>
15         </div>
16     @endif
```

8. Screenshot hasilnya.



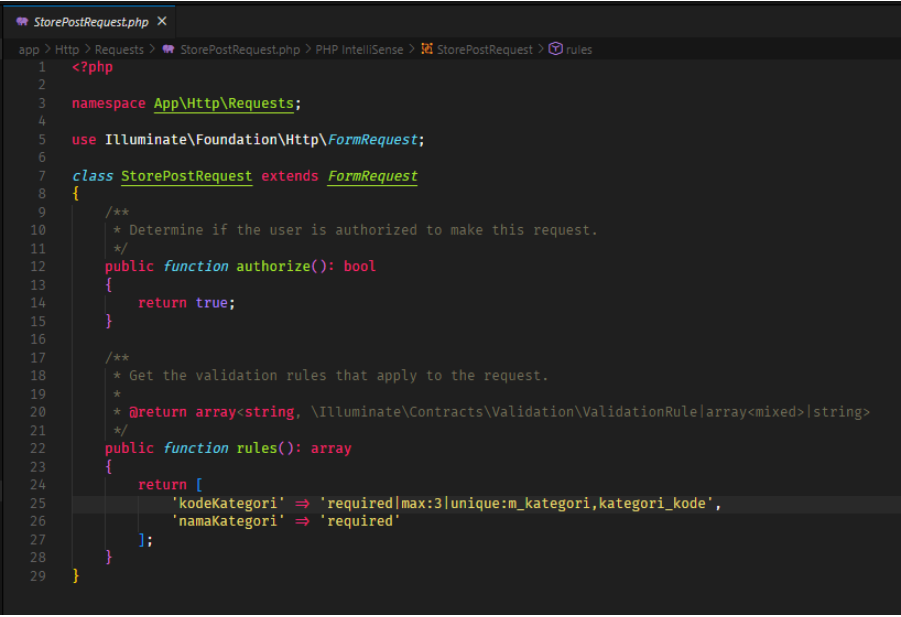
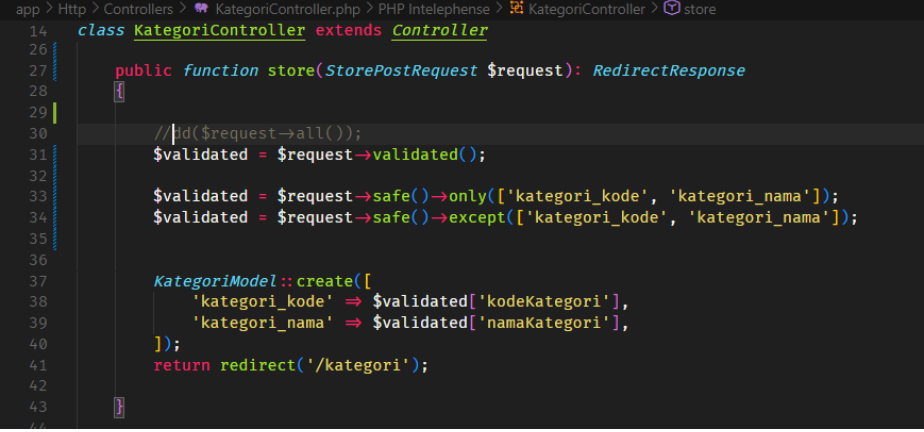
9. Pada view/create.blade.php tambahkan dan coba running code berikut :

```
<form method="POST" action="..../kategori">
<div class="card-body">
<div class="form-group">
<label for="kodeKategori">Kode Kategori</label>
<input type="text" class="form-control" id="kodeKategori" name="kodeKategori" placeholder="Untuk Makanan, contoh: MKN">
<div class="text-danger">
<div class="alert alert-danger">{{ $message }}</div>
</div>
</div>
<div class="form-group">
<label for="namaKategori">Nama Kategori</label>
<input type="text" class="form-control" id="namaKategori" name="namaKategori" placeholder="Nama">
</div>
<div class="card-footer">
<button type="submit" class="btn btn-primary">Submit</button>
</div>
</form>
```



- Kode tersebut bertujuan untuk menampilkan pesan error pada input field secara spesifik.

C. Form Request Validation

No.	Langkah Praktikum
1.	<p>Membuat permintaan form dengan menuliskan pada terminal</p> <pre> 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> </pre>
2.	<p>Ketik kode berikut pada Http/request/StorePostRequest</p>  <pre> 1 <?php 2 3 namespace App\Http\Requests; 4 5 use Illuminate\Foundation\Http\FormRequest; 6 7 class StorePostRequest extends FormRequest 8 { 9 /** 10 * Determine if the user is authorized to make this request. 11 */ 12 public function authorize(): bool 13 { 14 return true; 15 } 16 17 /** 18 * Get the validation rules that apply to the request. 19 * 20 * @return array<string, \Illuminate\Contracts\Validation\ValidationRule array<mixed> string> 21 */ 22 public function rules(): array 23 { 24 return [25 'kodeKategori' => 'required max:3 unique:m_kategori,kategori_kode', 26 'namaKategori' => 'required' 27]; 28 } 29 } </pre>  <pre> 14 class KategoriController extends Controller 15 { 16 public function store(StorePostRequest \$request): RedirectResponse 17 { 18 //id(\$request->all()); 19 \$validated = \$request->validated(); 20 21 \$validated = \$request->safe()->only(['kategori_kode', 'kategori_nama']); 22 \$validated = \$request->safe()->except(['kategori_kode', 'kategori_nama']); 23 24 KategoriModel::create([25 'kategori_kode' => \$validated['kodeKategori'], 26 'kategori_nama' => \$validated['namaKategori'], 27]); 28 return redirect('/kategori'); 29 } 30 } </pre>
3.	<p>Terapkan validasi juga pada tabel m_user dan m_level.</p> <ul style="list-style-type: none"> Validasi pada tabel m_level

```

UserController.php
app > Http > Controllers > UserController.php > PHP Intelephense > UserController > store
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
15     public function index(UserDataTable $dataTable)
16     {
17         return $dataTable->render('m_user.index');
18     }
19
20     public function create()
21     {
22         return view('m_user.create_user');
23     }
24
25     public function store(Request $request): RedirectResponse
26     {
27         //dd($request->all());
28         $validated = $request->validate([
29             'level_id' => 'bail|required',
30             'username' => 'required',
31             'name' => 'required',
32             'password' => 'required',
33         ]);
34
35         $password = Hash::make($validated['password']);
36
37         UserModel::create([
38             'level_id' => $validated['level_id'],
39             'username' => $validated['username'],
40             'name' => $validated['name'],
41             'password' => Hash::make($validated['password']),
42         ]);
43
44         return redirect('/m_user');
45     }
46
47     public function edit($id){
48         $user = UserModel::find($id);
49         return view('m_user.edit_user', ['data' => $user]);
50     }
51
52     public function update(Request $request, $id){
53         $user = UserModel::find($id);
54         $user->level_id = $request->level_id;
55         $user->username = $request->username;
56         $user->name = $request->name;
57         $user->password = $request->password;
58         $user->save();
59         return redirect('/m_user');
60     }
61
62     public function destroy($id) {
63         UserModel::find($id)->delete();
64
65         return redirect('/m_user');
66     }
67

```

Form M_Level

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

Tambah Data M_Level

Kode Level

Untuk Administrator, Contoh: ADM

The kode level field is required.

Nama Level

Masukkan Nama Level

Tambah

- Validasi pada tabel m_user

LevelController.php

app > Http > Controllers > LevelController.php > PHP Intelephense > LevelController > destroy

```
10
11 class LevelController extends Controller
12 {
13     public function index(LevelDataTable $dataTable)
14     {
15         return $dataTable->render('m_level.index');
16         // DB::insert('insert into m_level(level_kode, level_nama, created_at)
17         // values (?, ?, ?)', ['CUS', 'Pelanggan', now()]);
18
19         // return 'Insert Data Baru Berhasil!';
20
21         // $row = DB::update('update m_level set level_nama = ? where level_kode = ?',
22         // ['Customer', 'CUS']);
23         // return 'Update Data Berhasil Jumlah Data yang Diupdate: '.$row.' baris';
24
25         // $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
26         // return 'Delete Data Berhasil. Jumlah Data yang Dihapus: '.$row.' baris';\
27
28         // $data = DB::select('select * from m_level');
29         // return view('level', ['data' => $data]);
30     }
31
32     public function create()
33     {
34         return view('m_level.create_level');
35     }
36
37     public function store(Request $request): RedirectResponse
38     {
39         //dd($request->all());
40         $validated = $request->validate([
41             'kodeLevel' => 'bail|required|unique:m_level,level_kode',
42             'namaLevel' => 'required',
43         ]);
44
45         LevelModel::create([
46             'level_kode' => $validated['kodeLevel'],
47             'level_nama' => $validated['namaLevel'],
48         ]);
49
50         return redirect('/m_level');
51     }
52
53     public function edit($id){
54         $level = LevelModel::find($id);
55         return view('m_level.edit_level', ['data' => $level]);
56     }
57
58     public function update(Request $request, $id){
59         $level = LevelModel::find($id);
60         $level->level_kode = $request->kodeLevel;
61         $level->level_nama = $request->namaLevel;
62         $level->save();
63         return redirect('/m_level');
64     }
65
66     public function destroy($id) {
67         LevelModel::find($id)->delete();
68
69         return redirect('/m_level');
70     }
71 }
72
```

localhost/PWL_PDS/public/user_level/create

Gmail YouTube Instagram Pinterest DATA YUGA AND...

search

Pages Manage Kategori ACCOUNT SETTINGS Profile Change Password Multi Level LABELS Important Warning Information

Form M_User

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

Tambah Data M_User

Level ID Administrator

Username Masukkan Username The username field is required.

Nama Masukkan Nama

Password Masukkan Password

Tambah

D. CRUD (Create, Read, Update Delete)

No.	Langkah Praktikum
1.	<p>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.</p> <pre>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. PS C:\laragon\www\PWL_POS> </pre>
2.	<p>Berikut perintahnya: sekarang tambahkan kode pada route/web.php</p> <pre>Route::resource('m_user', POSController::class); </pre>
3.	<p>Atur pada Models/m_user</p> <pre>m_user.php U X app > Models > m_user.php > PHP Intelephense > m_user > \$fillable 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Factories\HasFactory; 6 use Illuminate\Database\Eloquent\Model; 7 8 class m_user extends Model 9 { 10 protected \$table = "m_user"; 11 public \$timestamps = false; 12 protected \$primaryKey = 'user_id'; 13 14 protected \$fillable = [15 'user_id', 16 'level_id', 17 'username', 18 'name', 19 'password', 20]; 21 } 22</pre>
4.	<p>Atur pada Migration/m_user_table</p> <pre>m_user_table.php X database > migrations > m_user_table.php > class > down 1 <?php 2 3 use App\Models\m_level; 4 use Illuminate\Database\Migrations\Migration; 5 use Illuminate\Database\Schema\Blueprint; 6 use Illuminate\Support\Facades\Schema; 7 8 return new class extends Migration 9 { 10 /** 11 * Run the migrations. 12 */ 13 public function up(): void 14 { 15 Schema::create('useri', function (Blueprint \$table) { 16 \$table->id('user_id'); 17 \$table->unsignedBigInteger('level_id')->index; 18 \$table->string('username', 20)->unique(); 19 \$table->string('name', 100); 20 \$table->string('password'); 21 \$table->timestamps(); 22 23 \$table->foreign('level_id')->references('level_id')->on('m_level'); 24 }); 25 } 26 /** 27 * Reverse the migrations. 28 */ 29 public function down(): void 30 { 31 Schema::dropIfExists('useri'); 32 } 33 } 34</pre>

5. Pada file Controller yang kita dapat secara otomatis akan terdapat 7 fungsi berikut yang dapat kita gunakan untuk membuat operasi CRUD.

```
app > Http > Controllers > POSController.php > PHP Intelephense > POSController > store
13 class POSController extends Controller
14 {
15     /**
16      * Display a listing of the resource.
17      */
18     public function index()
19     {
20     }
21 }
22
23 /**
24  * Show the form for creating a new resource.
25  */
26 public function create()
27 {
28 }
29
30 /**
31  * Store a newly created resource in storage.
32  */
33 public function store()
34 {
35 }
36
37 /**
38  * Display the specified resource.
39  */
40 public function show(string $id)
41 {
42 }
43
44 /**
45  * Show the form for editing the specified resource.
46  */
47 public function edit(string $id)
48 {
49 }
50
51 /**
52  * Update the specified resource in storage.
53  */
54 public function update(Request $request, string $id)
55 {
56 }
57
58 /**
59  * Remove the specified resource from storage.
60  */
61 public function destroy(string $id)
62 {
63 }
64
65
66
67
68
69
70
71
72
73
74
```

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

```
app > Http > Controllers > POSController.php > PHP Intelephense > POSController
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
9
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         $useri = m_user::all(); // Mengambil semua isi tabel
20         return view('m_user.index', compact('useri'))->with('i');
21     }
22
23     /**
24      * Show the form for creating a new resource.
25      */
26     public function create()
27     {
28         return view('m_user.create');
29     }
30
31     /**
32      * Store a newly created resource in storage.
33      */
34     public function store(Request $request)
35     {
36         $request->validate([
37             'user_id' => 'max:20',
38             'level_id' => 'required',
39             'username' => 'required',
40             'name' => 'required',
41         ]);
42
43         $userData = $request->all();
44         $userData['password'] = Hash::make($request->password);
45
46         m_user::create($userData);
47         return redirect()->route('m_user.index')
48             ->with('success', 'user Berhasil Ditambahkan');
49     }
50 }
```

```

11  > Controllers > POSController.php > PHP Inteprephse > POSController
12  class POSController extends Controller
13  {
14      /**
15       * Display the specified resource.
16       */
17      public function show(string $id)
18      {
19          $user = m_user::findOrFail($id);
20          return view('m_user.show', compact('user'));
21      }
22
23      /**
24       * Show the form for editing the specified resource.
25       */
26      public function edit(string $id)
27      {
28          $user = m_user::find($id);
29          return view('m_user.edit', compact('user'));
30      }
31
32      /**
33       * Update the specified resource in storage.
34       */
35      public function update(Request $request, string $id)
36      {
37          $request->validate([
38              'level_id' => 'required',
39              'username' => 'required',
40              'name' => 'required',
41              'password' => 'required',
42          ]);
43          //fungsi eloquent untuk mengupdate data inputan kita
44          m_user::find($id)->update($request->all());
45          //jika data berhasil diupdate, akan kembali ke halaman utama
46          return redirect()->route('m_user.index')
47              ->with('success', 'Data Berhasil Diupdate');
48      }
49
50      /**
51       * Remove the specified resource from storage.
52       */
53      public function destroy(string $id)
54      {
55          $user = m_user::findOrFail($id)->delete();
56          return \redirect()->route('m_user.index')
57              -> with('success', 'data Berhasil Dihapus');
58      }
59  }

```

- | | |
|----|---|
| 7. | Buatlah folder di Resources/Views/m_user dengan beberapa blade dan isian kode berikut:
a. template.blade.php |
|----|---|

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <title>CRUD Laravel</title>
6 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
7 </head>
8
9
10 <body>
11 <div class="container">
12 @yield('content')
13 </div>
14 </body>
15
16 </html>

```

b. index.blade.php

```
resources > views > user > # Create User Page | # Edit User Page | # Delete User Page | # Show User Page | # Index User Page  
1 @section('content')  
2     <div class="row mt-5">  
3         <div class="col-lg-12 margin-top d-flex justify-content-between w-full">  
4             <div class="w-half">  
5                 <@if auth::check()>  
6                     </div>  
7                 <div class="">  
8                     <div class="btn btn-success"><a href="{{ route('m_user.create')}}"> Input User</a>  
9                 </div>  
10            </div>  
11        </div>  
12    </div>  
13    @if ($message == Session::get('success'))  
14        <div class="alert alert-success">  
15            <div class="text-center"><div class="font-weight-bold">{{ $message }}</div>  
16        </div>  
17    </div>  
18    <table border="1" class="table table-bordered">  
19        <thead>  
20            <tr>  
21                <th width="20%">No</th>  
22                <th width="15%">Nama</th>  
23                <th width="15%">Email</th>  
24                <th width="15%">Username</th>  
25                <th width="15%">Password</th>  
26                <th width="15%">>Action</th>  
27            </tr>  
28        </thead>  
29        <tbody>  
30            <tr>  
31                <td>{{ $m_user->id }}</td>  
32                <td>{{ $m_user->level_id }}</td>  
33                <td>{{ $m_user->nama }}</td>  
34                <td>{{ $m_user->username }}</td>  
35                <td>{{ $m_user->password }}</td>  
36                <td>{{ $m_user->action }}</td>  
37            </tr>  
38        </tbody>  
39    </table>  
40    <div class="text-center">  
41        <div class="btn btn-danger">Hapus Data</div>  
42        <div class="btn btn-primary">Edit Data</div>  
43    </div>  
44    </div>  
45    </div>  
46    </div>  
47    </div>  
48    </div>  
49    </div>  
50    </div>  
51    </div>  
52    </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>
```

c. create.blade.php

```
resources > views > m_user > create.blade.php > ...
1 @extends('m_user/template')
2 @section('content')
3     <div class="row mt-5 mb-5">
4         <div class="col-lg-12 margin-tb">
5             <div class="float-left">
6                 <h2>Membuat Daftar User</h2>
7             </div>
8             <div class="float-right">
9                 <a class="btn btn-secondary" href="{{ route('m_user.index') }}">Kembali</a>
10            </div>
11        </div>
12    </div>
13    @if ($errors->any())
14        <div class="alert alert-danger">
15            <strong>Ops</strong> Input gagal<br><br>
16            <ul>
17                @foreach ($errors->all() as $error)
18                    <li>{{ $error }}</li>
19                @endforeach
20            </ul>
21        </div>
22    @endif
23    <form action="{{ route('m_user.store') }}" method="POST">
24        @csrf
25        <div class="col-xs-12 col-sm-12 col-md-12">
26            <div class="form-group">
27                <strong>Level_id:</strong>
28                <input type="text" name="level_id" class="form-control" placeholder="Masukkan level">
29            </div>
30        </div>
31        <div class="col-xs-12 col-sm-12 col-md-12">
32            <div class="form-group">
33                <strong>Username:</strong>
34                <input type="text" name="username" class="form-control" placeholder="Masukkan username">
35            </div>
36        </div>
37        <div class="col-xs-12 col-sm-12 col-md-12">
38            <div class="form-group">
39                <strong>Nama:</strong>
40                <input type="text" name="name" class="form-control" placeholder="Masukkan nama">
41            </div>
42        </div>
43        <div class="col-xs-12 col-sm-12 col-md-12">
44            <div class="form-group">
45                <strong>Password:</strong>
46                <input type="password" name="password" class="form-control" placeholder="Masukkan password">
47            </div>
48        </div>
49        <div class="col-xs-12 col-sm-12 col-md-12 text-center">
50            <button type="submit" class="btn btn-primary">Submit</button>
51        </div>
52    </form>
53 @endsection
```

d. show.blade.php

```
resources > views > m_user > show.blade.php > ...
1 @extends('m_user/template')
2 @section('content')
3     <div class="row mt-5 mb-5">
4         <div class="col-lg-12 margin-tb d-flex justify-content-between w-full">
5             <div class="w-full">
6                 <h2>Show User</h2>
7             </div>
8             <div class="">
9                 <a class="btn btn-secondary" href="{{ route('m_user.index') }}">Kembali</a>
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                {{ $useri->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>Level_id:</strong>
23                {{ $useri->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>Username:</strong>
29                {{ $useri->username }}
30            </div>
31        </div>
32        <div class="col-xs-12 col-sm-12 col-md-12">
33            <div class="form-group">
34                <strong>Nama:</strong>
35                {{ $useri->name }}
36            </div>
37        </div>
38        <div class="col-xs-12 col-sm-12 col-md-12">
39            <div class="form-group">
40                <strong>Password:</strong>
41                {{ $useri->password }}
42            </div>
43        </div>
44    </div>
45 @endsection
46
```

e. edit.blade.php

```
resources > views > m_user > create.blade.php > ...
1 @extends('m_user/template')
2 @section('content')
3     <div class="row mt-5 mb-5">
4         <div class="col-lg-12 margin-tb">
5             <div class="float-left">
6                 <h2>Edit User</h2>
7             </div>
8             <div class="float-right">
9                 <a class="btn btn-secondary" href="{{ route('m_user.index') }}">
10                     Kembali</a>
11             </div>
12         </div>
13     </div>
14     @if ($errors->any())
15         <div class="alert alert-danger">
16             <strong>Ops!</strong> Error <br><br>
17             <ul>
18                 @foreach ($errors->all() as $error)
19                     <li>{{ $error }}</li>
20                 @endforeach
21             </ul>
22         </div>
23     @endif
24     <form action="{{ route('m_user.update', $user->user_id) }}" method="POST">
25         @csrf
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>User id:</strong>
31                     <input type="text" name="userid" value="{{ $user->user_id }}" class="form-control"
32                         placeholder="Masukkan user id" disabled>
33                 </div>
34             </div>
35             <div class="col-xs-12 col-sm-12 col-md-12">
36                 <div class="form-group">
37                     <strong>Level id:</strong>
38                     <input type="text" name="level_id" value="{{ $user->level_id }}" class="form-control"
39                         placeholder="Masukkan level">
40                 </div>
41             </div>
42             <div class="col-xs-12 col-sm-12 col-md-12">
43                 <div class="form-group">
44                     <strong>Username:</strong>
45                     <input type="text" value="{{ $user->username }}" class="form-control" name="username"
46                         placeholder="Masukkan username">
47                 </div>
48             </div>
49             <div class="col-xs-12 col-sm-12 col-md-12">
50                 <div class="form-group">
51                     <strong>nama:</strong>
52                     <input type="text" value="{{ $user->name }}" name="name" class="form-control"
53                         placeholder="Masukkan nama">
54                 </div>
55             </div>
56             <div class="col-xs-12 col-sm-12 col-md-12">
57                 <div class="form-group">
58                     <strong>Password:</strong>
59                     <input type="password" value="{{ $user->password }}" name="password" class="form-control"
60                         placeholder="Masukkan password">
61                 </div>
62             </div>
63             <div class="col-xs-12 col-sm-12 col-md-12 text-center">
64                 <button type="submit" class="btn btn-primary">Update</button>
65             </div>
66         </div>
67     </form>
68 @endsection
```

8. Output yang dihasilkan

localhost/PWL_POS/public/m_user

CRUD user

Input User

User id	Level id	username	nama	password	
1	1	admin	Administrator	\$2y\$12\$VnmYtriKex0xv4yWiwbdU7Losoqu1RtYv2XoGU/T35B6..TIX6	<a>Show <a>Edit <a>Delete
2	2	manager	Manager	\$2y\$12\$AdJcPCCMd950O4T48cJEekLX1cSOTG.78qnUlv/FYz6j.eihagem	<a>Show <a>Edit <a>Delete
3	3	staff	Staff/Kasir	\$2y\$12\$RIUamUW6nnxCU/Pg6updOQ2fDhxu6l/in1J4FURzQhwxTr6V0B6	<a>Show <a>Edit <a>Delete
11	4	customer-1	Pelanggan Pertama	\$2y\$12\$MGwLf02r3yKBpq.4i48D/QQh.HMoMA78OZW3bkZyv5jpqakh1mfy	<a>Show <a>Edit <a>Delete

Membuat Daftar User

[Kembali](#)

Level_id:

Username:

Nama:

Password:

Show User

[Kembali](#)

User_id: 1

Level_id: 1

Username: admin

Nama: Administrator

Password: \$2y\$12\$VnmytriKeoxv4yWiwbdU7LosoqrU1RtY.v2XoGU/T35B6.TX6

Edit User

[Kembali](#)

User_id:

Level_id:

Username:

nama:

Password:

2	2	manager	Manager	\$2y\$12\$R/UamUk6nnxCU/Pg6updOQ2fDhxu6/in1J4FURzQhwxTr6V0B6	Show Edit Delete
3	3	staff	Staff/Kasir	\$2y\$12\$R/UamUk6nnxCU/Pg6updOQ2fDhxu6/in1J4FURzQhwxTr6V0B6	Show Edit Delete
11	4	customer-1	Pelanggan Pertama	\$2y\$12\$MGwL02r3yKBpq.4H4BD/QQh.HMoMA780ZW3bkZYv5jppakh1mFy	Show Edit Delete
12	2	manager_dua	Manager 2	\$2y\$12\$WzD7o3gd5nqKisLDpkKrugA44tqZAwUVzvUBMhievcb2DlkwcA2	Show Edit Delete
13	2	manager22	Manager Dua Dua	\$2y\$12\$4jJNUL3GBAARCH0dsij506eODf8toTcoTT0JSAILikd2EFcZCZufi	Show Edit Delete
22	4	dxo2err	Daffa Yudisa Akbar	\$2y\$12\$2.D1HnT5cv7ZW9ZumiNv6Oz20c6hOCjdWU1GyKjHqHnsISTVFN7.6	Show Edit Delete

localhost says

Apakah Anda yakin ingin menghapus data ini?

Tugas

No.	Jawaban																																																
1.	<p>Coba tampilkan level_id pada halaman web tersebut dimana field ini merupakan foreign key</p> <div><p>localhost/PWL_POS/public/m_user</p><p>CRUD user Input User</p><table><tr><th>User Id</th><th>Level Id</th><th>Username</th><th>Nama</th><th>Password</th><th>Action</th></tr><tr><td>1</td><td>1</td><td>admin</td><td>Administrator</td><td>\$2y\$12\$VnmytriKeoxJvx4yWiwbdu7Losoqru1RtY.v2XoGU/T35B6.TIX6</td><td>Show Edit Delete</td></tr><tr><td>2</td><td>2</td><td>manager</td><td>Manager</td><td>\$2y\$12\$JA4JcPCCMd95004T4BcdEekLX1cSOTG.78qnUlv/FYz6j.eihagem</td><td>Show Edit Delete</td></tr><tr><td>3</td><td>3</td><td>staff</td><td>Staff/Kasir</td><td>\$2y\$12\$RIUamUKj6nnxCU/Pg6updOQ2fDhxux6/in1J4FURzQhwx7i6V0B6</td><td>Show Edit Delete</td></tr><tr><td>11</td><td>4</td><td>customer-1</td><td>Pelanggan Pertama</td><td>\$2y\$12\$MGwLf02r3yKBpq.4H4BD/QQh.HMoMA78OZW3bkZyV5jpqakh1mfy</td><td>Show Edit Delete</td></tr><tr><td>12</td><td>2</td><td>manager_dua</td><td>Manager 2</td><td>\$2y\$12\$WzD7o3gd5nqXisLDpkKrugA4tUqZAwUVzvUBMhievcb2DikwcA2</td><td>Show Edit Delete</td></tr><tr><td>13</td><td>2</td><td>manager22</td><td>Manager Dua Dua</td><td>\$2y\$12\$4jNUL3GBAARCH0dsj506eODf8toTcoTT0ISAILkDZEFcCZzUfi</td><td>Show Edit Delete</td></tr><tr><td>22</td><td>4</td><td>doxZerr</td><td>Daffa Yudisa Akbar</td><td>\$2y\$12\$2.D1HnT5cv7ZW9ZumiNv6OZ0c6hOCJdWU1GyKyHqHNsl5TVFN7.6</td><td>Show Edit Delete</td></tr></table></div> <pre>public function create() { \$level = LevelModel::all(); //mendapatkan data dari tabel level return view('m_user.create', compact('level')); }</pre> <div><p>localhost/PWL_POS/public/m_user/create</p><p>Membuat Daftar User Kembali</p><p>Level id: <input type="text" value="Administrator"/></p><p>Username: <input type="text" value="Masukkan username"/></p><p>Nama: <input type="text" value="Masukkan nama"/></p><p>Password: <input type="text" value="Masukkan password"/></p><p>Submit</p></div>	User Id	Level Id	Username	Nama	Password	Action	1	1	admin	Administrator	\$2y\$12\$VnmytriKeoxJvx4yWiwbdu7Losoqru1RtY.v2XoGU/T35B6.TIX6	Show Edit Delete	2	2	manager	Manager	\$2y\$12\$JA4JcPCCMd95004T4BcdEekLX1cSOTG.78qnUlv/FYz6j.eihagem	Show Edit Delete	3	3	staff	Staff/Kasir	\$2y\$12\$RIUamUKj6nnxCU/Pg6updOQ2fDhxux6/in1J4FURzQhwx7i6V0B6	Show Edit Delete	11	4	customer-1	Pelanggan Pertama	\$2y\$12\$MGwLf02r3yKBpq.4H4BD/QQh.HMoMA78OZW3bkZyV5jpqakh1mfy	Show Edit Delete	12	2	manager_dua	Manager 2	\$2y\$12\$WzD7o3gd5nqXisLDpkKrugA4tUqZAwUVzvUBMhievcb2DikwcA2	Show Edit Delete	13	2	manager22	Manager Dua Dua	\$2y\$12\$4jNUL3GBAARCH0dsj506eODf8toTcoTT0ISAILkDZEFcCZzUfi	Show Edit Delete	22	4	doxZerr	Daffa Yudisa Akbar	\$2y\$12\$2.D1HnT5cv7ZW9ZumiNv6OZ0c6hOCJdWU1GyKyHqHNsl5TVFN7.6	Show Edit Delete
User Id	Level Id	Username	Nama	Password	Action																																												
1	1	admin	Administrator	\$2y\$12\$VnmytriKeoxJvx4yWiwbdu7Losoqru1RtY.v2XoGU/T35B6.TIX6	Show Edit Delete																																												
2	2	manager	Manager	\$2y\$12\$JA4JcPCCMd95004T4BcdEekLX1cSOTG.78qnUlv/FYz6j.eihagem	Show Edit Delete																																												
3	3	staff	Staff/Kasir	\$2y\$12\$RIUamUKj6nnxCU/Pg6updOQ2fDhxux6/in1J4FURzQhwx7i6V0B6	Show Edit Delete																																												
11	4	customer-1	Pelanggan Pertama	\$2y\$12\$MGwLf02r3yKBpq.4H4BD/QQh.HMoMA78OZW3bkZyV5jpqakh1mfy	Show Edit Delete																																												
12	2	manager_dua	Manager 2	\$2y\$12\$WzD7o3gd5nqXisLDpkKrugA4tUqZAwUVzvUBMhievcb2DikwcA2	Show Edit Delete																																												
13	2	manager22	Manager Dua Dua	\$2y\$12\$4jNUL3GBAARCH0dsj506eODf8toTcoTT0ISAILkDZEFcCZzUfi	Show Edit Delete																																												
22	4	doxZerr	Daffa Yudisa Akbar	\$2y\$12\$2.D1HnT5cv7ZW9ZumiNv6OZ0c6hOCJdWU1GyKyHqHNsl5TVFN7.6	Show Edit Delete																																												

```

    */
    public function edit(string $id)
    {
        $level = LevelModel::all(); //mendapatkan data dari tabel level
        $useri = m_user::find($id);
        return view('m_user.edit', compact('useri', 'level'));
    }
}

</div>
<div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
        <strong>Level_id:</strong>
        <select name="level_id" class="form-control">
            @foreach ($level as $lvl)
                <option value="{{ $lvl->level_id }}" @if($lvl->level_id == $useri->level_id) selected @endif>{{ $lvl->level_nama }}</option>
            @endforeach
        </select>
    </div>
</div>

```

localhost/PWL_PC5/public/m_user/1/edit

Edit User

Kembali

User_id:
1

Level_id:
Administrator

Username:
admin

nama:
Administrator

Password:

Update

2. Modifikasi dengan tema/ template kesukaan Anda

```

resources > views > m_user > template.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en" data-bs-theme="dark">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>CRUD Laravel</title>
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
9 <style>
10 /* Ganti warna teks menjadi putih */
11 body {
12     color: white !important;
13 }
14
15 /* Ganti warna latar belakang menjadi biru langit */
16 /* body { */
17 /* background-color: #87CEEB !important; */
18 /* } */
19
20 /* Ganti warna latar belakang container menjadi putih */
21 .container {
22     background-color: white !important;
23     color: #333 !important;
24     border-radius: 10px;
25     padding: 20px;
26     margin-top: 50px;
27 }
28 </style>
29 </head>
30
31 <body>
32 <div class="container">
33     @yield('content')
34 </div>
35
36 {{-- <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script> --}}
37 {{-- <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script> --}}
38 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
39 </body>
40
41 <html>
42

```

← → ↻ 🏠 localhost/PWL_POS/public/m_user

📧 Gmail 📺 YouTube 🖼️ Sakad Polinema 🌟 DAFFA YUDISA AKB...

CRUD user

Input User

User Id	Level Id	Username	Nama	Password	Action
1	1	admin	Administrator	\$2y\$12\$VnmytriKeouxv4yWiwbdu7Losoqru1RtY.v2XoGU/T35B6.TIX6	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$AdjCpCCMd95004T48cJEkLX1cSOTG.78qnUlv/FYz6j.elhagem	Show Edit Delete
3	3	staff	Staff/Kasir	\$2y\$12\$RiUamUk6nrxCU/Pg6updOQ2fDhxux6l/in14FURzQhwaTr6V0B6	Show Edit Delete
11	4	customer-1	Pelanggan Pertama	\$2y\$12\$MgwLf02-3yK8pq.4I4BD/.QQh.HMoMA78OZW3bkZY5j.pqakh1mfy	Show Edit Delete
12	2	manager_dua	Manager 2	\$2y\$12\$WzD7o3gd5nqXisLDpkKrugAI4UqZAwUVzvUBMhievcb2DlkwcA2	Show Edit Delete
13	2	manager22	Manager Dua Dua	\$2y\$12\$4jNUL3GBAARCH0dsj506eODF8toTcoTT0JSAiLfkD2EfCZCZufi	Show Edit Delete
22	4	dx2err	Daffa Yudisa Akbar	\$2y\$12\$2.D1HnT5cv7ZW92umINv6OzZ0c6hOCIdWU1GykyHqHNi5TVFN7.6	Show Edit Delete

← → ↻ 🏠 localhost/PWL_POS/public/m_user/create

📧 Gmail 📺 YouTube 🖼️ Sakad Polinema 🌟 DAFFA YUDISA AKB...

Membuat Daftar User

Kembali

Level id:

Administrator

Username:

Masukkan username

Nama:

Masukkan nama

Password:

Masukkan password

Submit

← → ↻ 🏠 localhost/PWL_POS/public/m_user/1

📧 Gmail 📺 YouTube 🖼️ Sakad Polinema 🌟 DAFFA YUDISA AKB...

Show User

Kembali

User id: 1

Level id: 1

Username: admin

Nama: Administrator

Password: \$2y\$12\$VnmytriKeouxv4yWiwbdu7Losoqru1RtY.v2XoGU/T35B6.TIX6

← → ↻ 🏠 localhost/PWL_POS/public/m_user/1/edit

📧 Gmail 📺 YouTube 🖼️ Sakad Polinema 🌟 DAFFA YUDISA AKB...

Edit User

Kembali

User id:

1

Level id:

Administrator

Username:

admin

nama:

Administrator

Password:

.....

Update

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

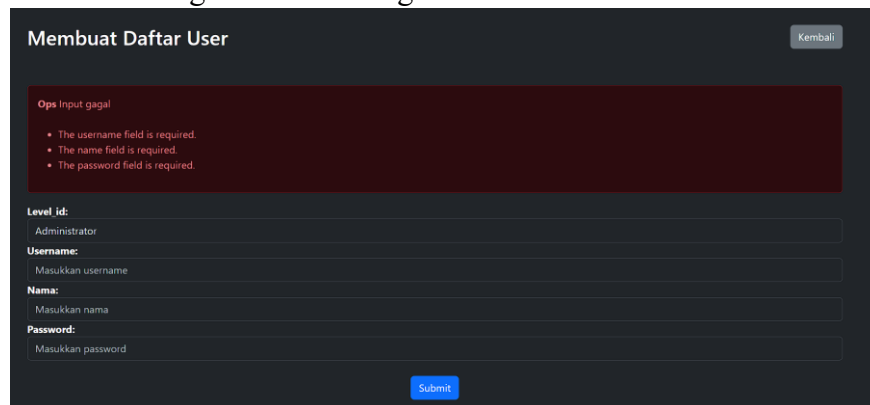
- a. **`$request->validate`** adalah metode yang digunakan untuk memvalidasi input yang diterima dari pengguna sebelum data tersebut disimpan ke dalam database. Fungsi ini memastikan bahwa data yang dimasukkan oleh pengguna sesuai dengan aturan yang ditetapkan sebelumnya, seperti aturan validasi yang telah ditentukan. Contoh:

```
public function store(Request $request)
{
    $request->validate([
        'user_id' => 'max 20',
        'level_id' => 'required',
        'username' => 'required',
        'name' => 'required',
        'password' => 'required',
    ]);

    $userData = $request->all();
    $userData['password'] = Hash::make($request->password);

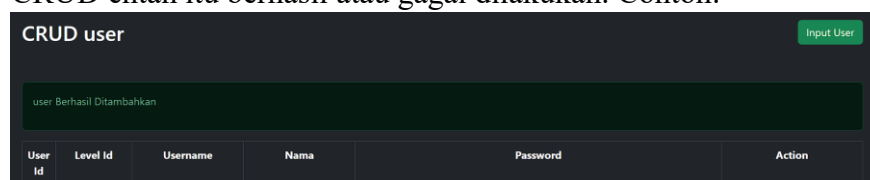
    m_user::create($userData);
    return redirect()->route('m_user.index')
        ->with('success', 'user Berhasil Ditambahkan');
}
```

- b. **`$errors`** akan berisi informasi tentang kesalahan validasi yang terjadi. Informasi ini kemudian dapat digunakan untuk memberikan feedback kepada pengguna, sehingga mereka dapat memperbaiki masalah sebelum mencoba mengirimkan data lagi. Contoh:



The screenshot shows a web form titled "Membuat Daftar User" with a "Kembali" button in the top right. A red error box at the top contains the message "Ops Input gagal" and a list of three errors: "The username field is required.", "The name field is required.", and "The password field is required." Below the error box, the form has four input fields: "Level id:" with a dropdown menu showing "Administrator", "Username:" with placeholder text "Masukkan username", "Nama:" with placeholder text "Masukkan nama", and "Password:" with placeholder text "Masukkan password". A blue "Submit" button is at the bottom right of the form.

- c. **`Alert`** yang ada pada halaman CRUD dapat digunakan untuk memberikan umpan balik atau konfirmasi kepada pengguna setelah dilakukan operasi CRUD entah itu berhasil atau gagal dilakukan. Contoh:



The screenshot shows a web page titled "CRUD user" with an "Input User" button in the top right. A green success alert box at the top contains the message "user Berhasil Ditambahkan". Below the alert box is a table with the following columns: "User id", "Level id", "Username", "Nama", "Password", and "Action". The table is currently empty.