

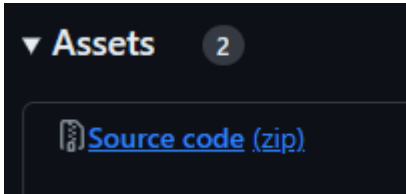
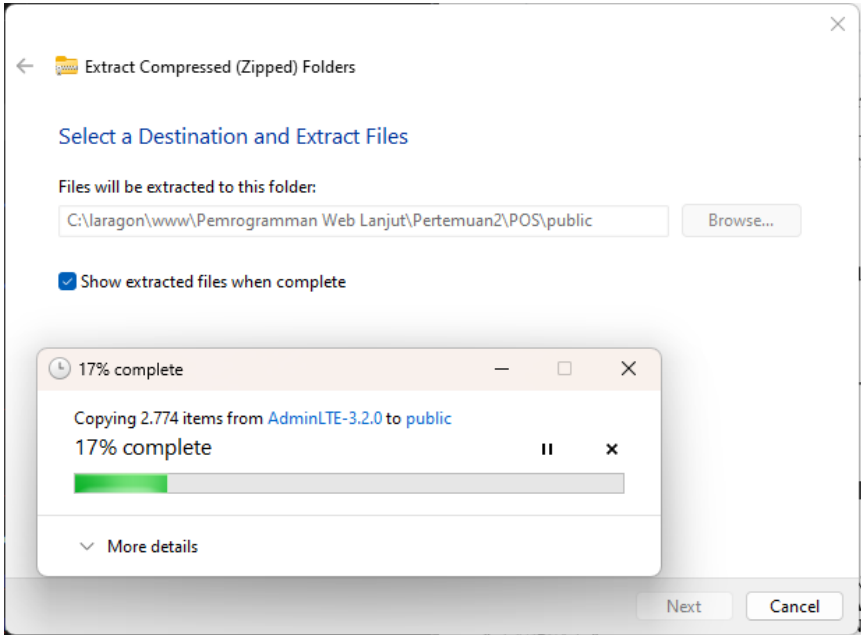
JOBSHEET 06



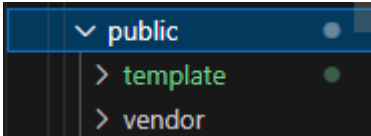





Template Form (AdminLTE), Server Validation, Client Validation, CRUD

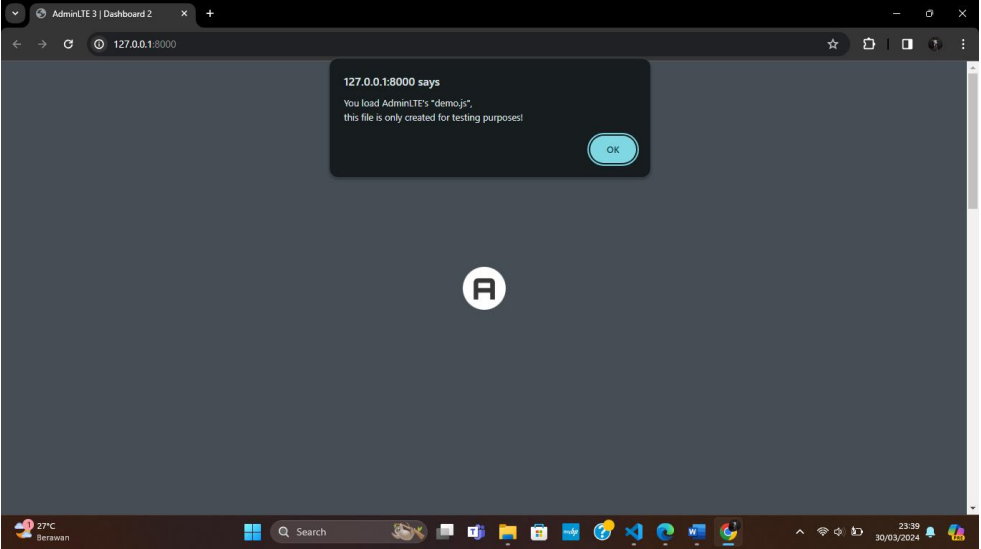
Nama : Pattricya Sonnya Fridayanti

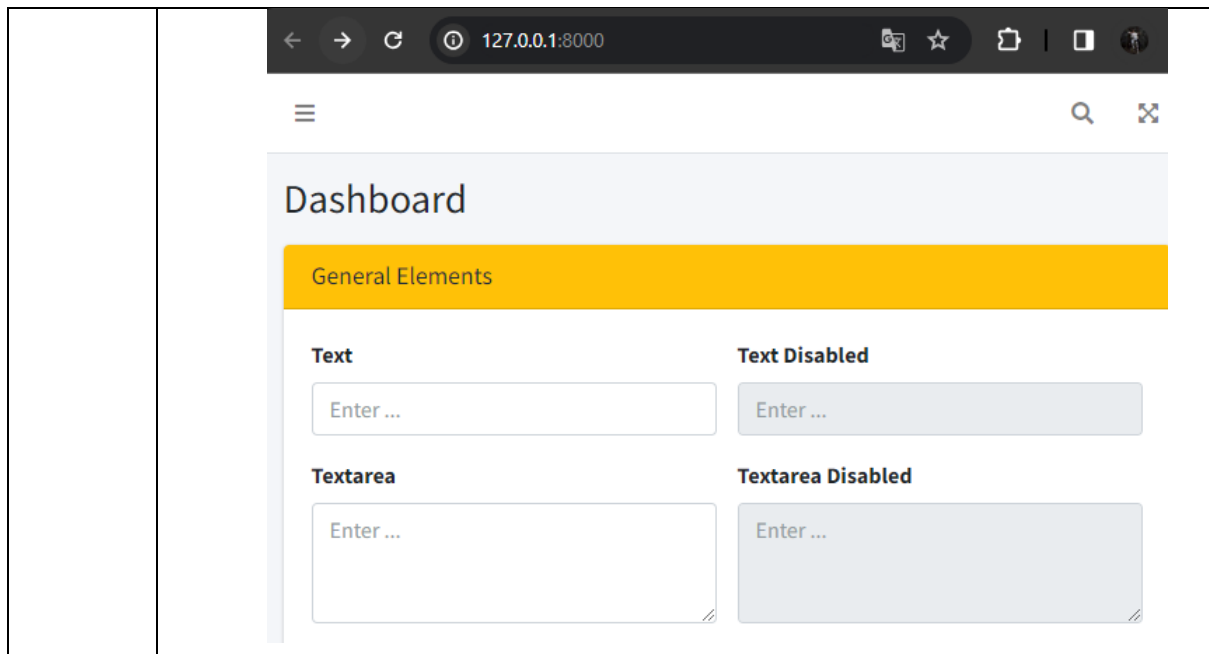
NIM : 2141762060

Kelas : SIB3C

No	Keterangan
A. Template Form (AdminLTE)	
1.	<p>Penggunaan template AdminLTE Bootstrap Admin Dashboard Template dengan langkah pertama akses https://adminlte.io/ , lalu klik download pada source code (zip)</p> <p>Link : Releases · ColorlibHQ/AdminLTE (github.com)</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> 

	<table><thead><tr><th>Name</th><th>Date modified</th><th>Type</th></tr></thead><tbody><tr><td> template</td><td>24/03/2024 12:31</td><td>File folder</td></tr><tr><td> vendor</td><td>22/03/2024 11:14</td><td>File folder</td></tr></tbody></table> 	Name	Date modified	Type	 template	24/03/2024 12:31	File folder	 vendor	22/03/2024 11:14	File folder
Name	Date modified	Type								
 template	24/03/2024 12:31	File folder								
 vendor	22/03/2024 11:14	File folder								
3.	<p>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</p> 									
4.	<p>Scroll ke bagian paling bawah dan sesuaikan dengan kode berikut</p>  <p>Note: apabila gambar pada dist/img belum dapat diload maka lakukan penyesuaian dengan asset</p>									
5.	Jalankan pada browser									

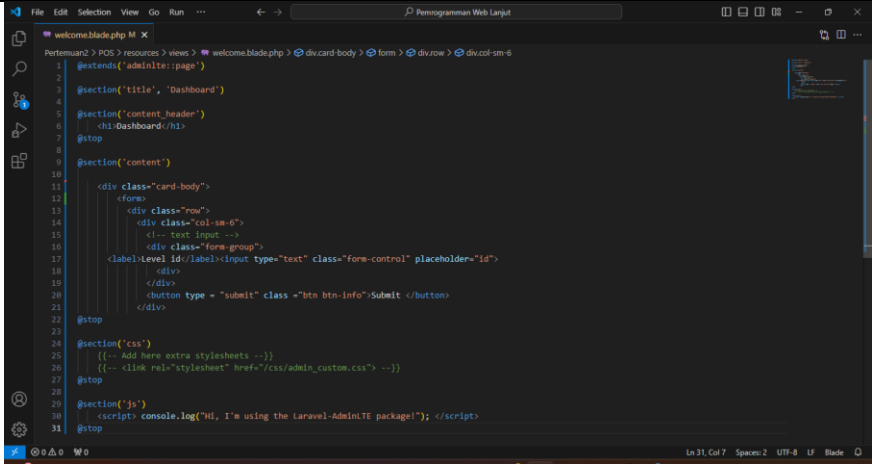
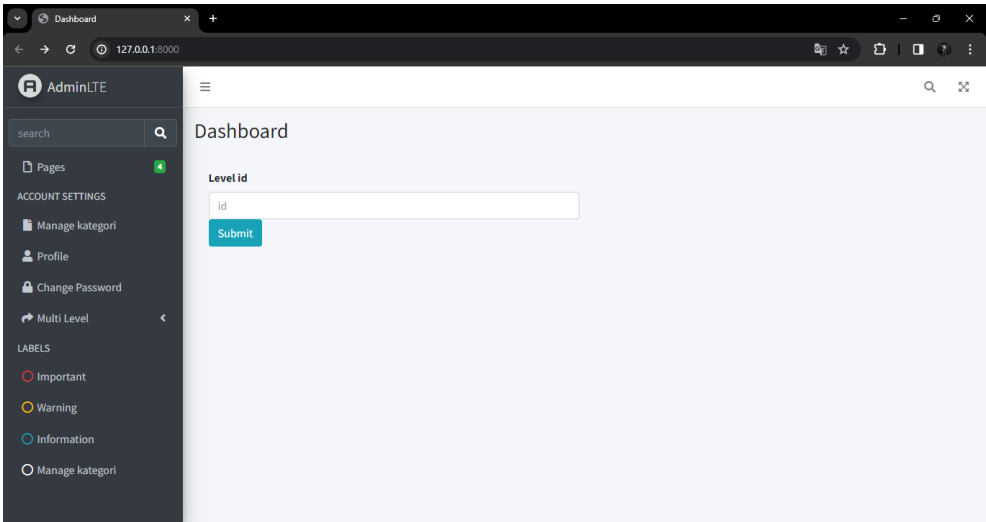
	 <p>note : template ini load untuk semuanya, hapus jika tidak diperlukan. Visual pada pada admin LTE dapat mempermudah untuk pengambilan kode.</p>
6.	<p>Pada bagian menu terdapat pilihan Form kita khususkan dulu pada Form yang disediakan oleh Admin LTE</p> <ul style="list-style-type: none"> • General Elements • Advanced Elements • Editor • Validation <p>Cara menggunakan template bisa dilihat nama formnya contoh judulnya 'General Elements', agar dapat digunakan dapat melakukan akses pada folder yang terdapat template adminLTE contoh :</p> <p>C:\laragon\www\PWL_2024\public\template\pages\forms</p> <ul style="list-style-type: none"> • Menerapkan general elements pada filde welcome.blade.php



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

- Form general

- Melakukan edit pada welcome.blade.php

	 <pre> 1 @extends('adminlte::page') 2 3 @section('title', 'Dashboard') 4 5 @section('content_header') 6 <h1>Dashboard</h1> 7 @stop 8 9 @section('content') 10 11 <div class="card-body"> 12 <form> 13 <div class="row"> 14 <div class="col-sm-6"> 15 <!-- text input --> 16 <div class="form-group"> 17 <label>level id</label><input type="text" class="form-control" placeholder="id"> 18 </div> 19 </div> 20 <div class="col-sm-6"> 21 <button type="submit" class="btn btn-info">Submit</button> 22 </div> 23 </div> 24 </form> 25 </div> 26 27 @section('css') 28 <!-- Add here extra stylesheets --> 29 <!-- link rel="stylesheet" href="/css/admin_custom.css" --> 30 @stop 31 32 @section('js') 33 <script> console.log("Hi, I'm using the Laravel AdminLTE package!"); </script> 34 @stop </pre>
8.	<p>Apa yang tampil?</p> 
9.	<p>Eksplorasi jenis form pada adminLTE dan coba terapkan yang sesuai untuk studi kasus POS PWL, buatlah form untuk tabel m_user dan m_level</p> <ul style="list-style-type: none"> • Script form m_user <ul style="list-style-type: none"> - Halaman index.blade.php

```

Pertemuan2 > POS > resources > views > user > index.blade.php > div.container >
1  @extends('layout.app')
2
3  {{-- Customize layout sections --}}
4
5  @section('subtitle', 'User')
6  @section('content_header_title', 'Home')
7  @section('content_header_subtitle', 'User')
8
9  @section('content')
10 <div class="container">
11     <div class="card">
12         <div class="card-header">Manage User</div>
13         <div class="card-header">
14             <a href="/user/create_user/">
15                 <button class="btn btn-primary" href="">Add +
16             </button>
17         </a>
18     </div>
19 </div>
20     <div class="card-body">
21         <h1>Data Pengguna</h1>
22         <table border='1' cellpadding='2' cellspacing='0'>
23             <tr>
24                 <th>ID</th>
25                 <th>Username</th>
26                 <th>Nama Level</th>
27                 <th>Action</th>
28             </tr>
29             @foreach ($data as $d)
30                 <tr>
31                     <td>{{ $d->user_id }}</td>
32                     <td>{{ $d->username }}</td>
33                     <td>{{ $d->level->level_nama }}</td>
34                     <td><a href="/user/edit/{{ $d->user_id }}">Edit</a>
35                     <a href="/user/delete/{{ $d->user_id }}">Delete</a></td>
36                 </tr>
37             @endforeach
38         </table>
39     </div>
40 </div>
41 </div>
42 @endsection

```

- Halaman UserController

```

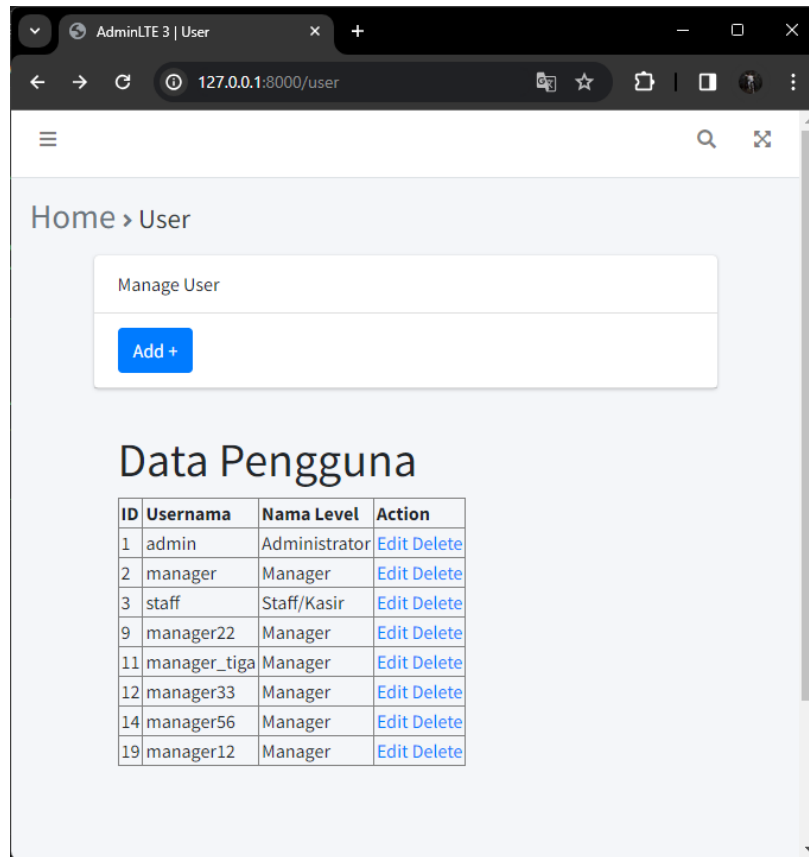
Pertemuan2 > POS > app > Http > Controllers > UserController.php > UserController > store
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Psy\TabCompletion\Matcher\FunctionsMatcher;
9
10 class UserController extends Controller
11 {
12     /*public function index(){
13         $user = UserModel::with('level')->get();
14         return view('user', ['data' => $user]);
15     }*/
16     public function index(){
17         $user = UserModel::all();
18         return view('user.index', ['data' => $user]);
19     }

```

- Halaman web.php

```
72 | Route::get('/level', [LevelController::class, 'index']);
```

- Hasil



- Script form m_level
 - Halaman index.blade.php

```

Pertemuan2 > POS > resources > views > level > index.blade.php > div.container > div.card-body > t
1  @extends('layout.app')
2
3  {{-- Customize layout sections --}}
4
5  @section('subtitle', 'Level')
6  @section('content_header_title', 'Home')
7  @section('content_header_subtitle', 'Level')
8
9  @section('content')
10 <div class="container">
11   <div class="card">
12     <div class="card-header">Manage Level</div>
13     <div class="card-header">
14       <a href="/level/create_level/">
15         <button class="btn btn-primary" href="">Add +
16       </button>
17     </a>
18   </div>
19 </div>
20 <div class="card-body">
21   <h1>Data Level Pengguna</h1>
22   <table border="1" cellpadding="2" cellspacing="0">
23     <tr>
24       <th>ID</th>
25       <th>Kode Level</th>
26       <th>Nama Level</th>
27       <th>Action</th>
28     </tr>
29     @foreach ($data as $d)
30     <tr>
31       <td>{{ $d->level_id }}</td>
32       <td>{{ $d->level_kode }}</td>
33       <td>{{ $d->level_nama }}</td>
34       <td><a href="/level/edit/{{ $d->level_id }}">Edit</a>
35         <a href="/level/delete/{{ $d->level_id }}">Delete</a></td>
36     </tr>
37   @endforeach
38 </table>
39 </div>
40 </div>
41 </div>
42 @endsection

```

- Halaman LevelController

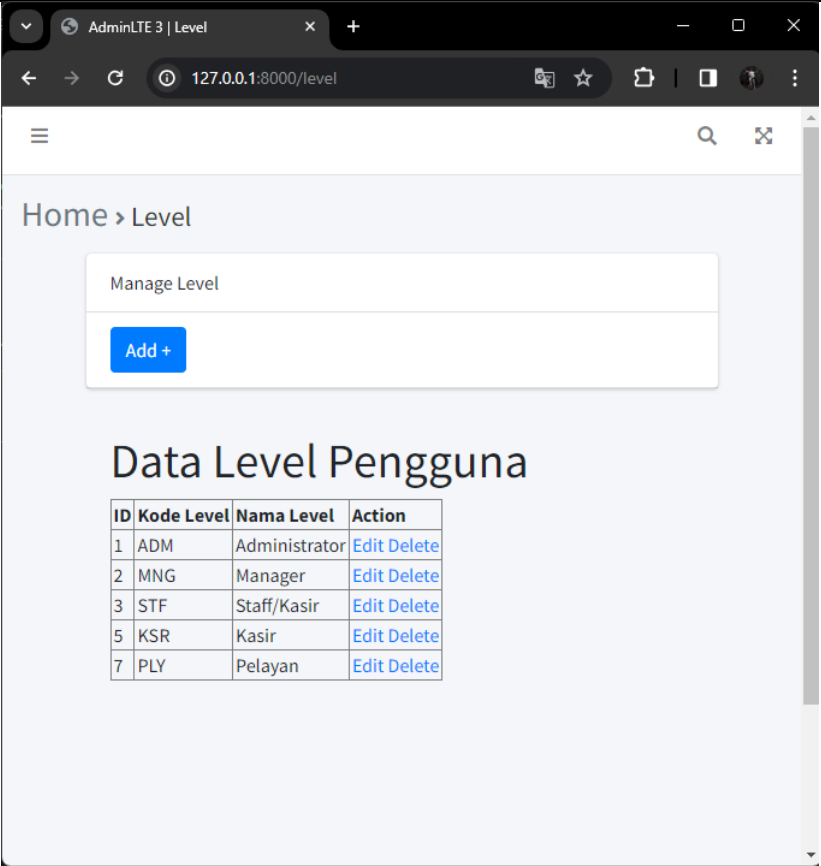
Pertemuan2 > POS > app > Http > Controllers > LevelController.php > LevelController

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\LevelModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\DB;
8  use Monolog\Level;
9
10 class LevelController extends Controller
11 {
12
13     public function index(){
14         $data = LevelModel::all();
15
16         return view('level.index', ['data' => $data]);
17     }
18
19     public function create(){
20         return view('level.create_level');
21     }
22
23     public function store(Request $request){
24         LevelModel::create([
25             'level_kode' => $request->kodeLevel,
26             'level_nama' => $request->namaLevel,
27         ]);
28         return redirect('/level');
29     }
29 }
```

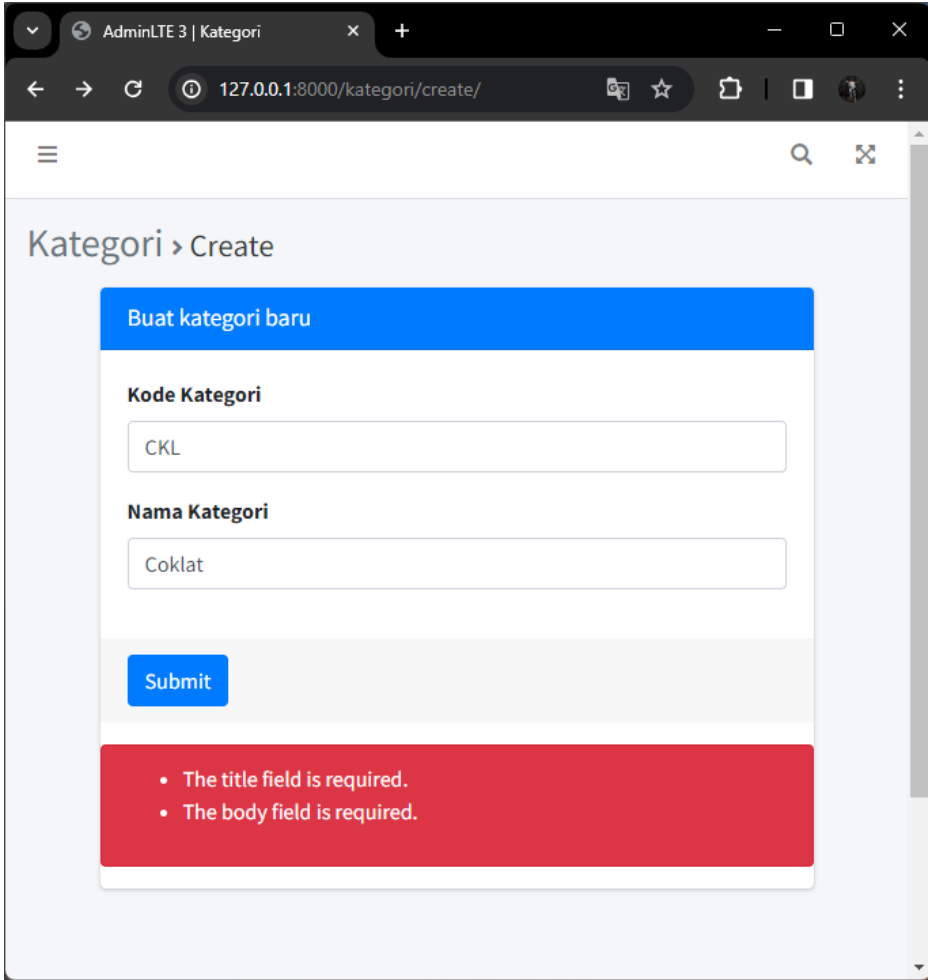
- Halaman web.php

```
76 Route::get('/user', [UserController::class, 'index']);
```

- Hasil

	
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 : <pre> 63 Route::get('/kategori/create', [KategoriController::class, 'create']); 64 Route::post('/kategori', [KategoriController::class, 'store']); </pre>
3.	Edit pada KategoriKontroller dengan kode:

	<pre> 12 class KategoriController extends Controller 13 { 14 15 public function create(): View 16 { 17 return view('category.create'); 18 } 19 20 public function store(Request \$request) : RedirectResponse { 21 \$validate = \$request->validate([22 'kategori_kode' => 'required', 23 'kategori_nama' => 'required', 24]); 25 26 return redirect('/kategori'); 27 } </pre>
4.	<p>Berikut ini sebagai alternatif contoh penulisan validasi dan bisa Anda sesuaikan kata kata dengan apa yang sedang Anda kerjakan,</p> <ul style="list-style-type: none"> Alt 1 <pre> 31 \$validateData = \$request->validate([32 'title' => ['required', 'unique:posts', 'max:255'], 33 'body' => ['required'], 34]); </pre> Alt 2 <pre> 36 \$validateData = \$request->validateWithBag('post', [37 'title' => ['required', 'unique:posts', 'max:255'], 38 'body' => ['required'], 39]); </pre>
5.	Tulis perbedaan penggunaan validate dengan validateWithBag!
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> 26 \$request->validate([27 'title'=> 'bail required unique:posts max:255', 28 'body'=> 'required', 29]); </pre>
7.	Pada view/create.blade.php tambahkan code berikut agar ketika validasi gagal, kita dapat menampilkan pesan kesalahan dalam tampilan:

	<pre> 32 {{--MENAMBAHKAN ERROR YANG TERJADI --}} 33 @if (\$errors->any()) 34 <div class="alert alert-danger"> 35 36 @foreach (\$errors->all() as \$error) 37 {{\$error}} 38 @endforeach 39 40 </div> 41 @endif </pre>
8.	<p>Screenshot hasilnya.</p> 
9.	<p>Catatan : Menyesuaikan Pesan Kesalahan</p> <p>Aturan validasi bawaan Laravel masing-masing memiliki pesan kesalahan yang terletak di lang/en/validation.php file aplikasi Anda. Jika aplikasi Anda tidak memiliki lang/direktori, Anda dapat menginstruksikan Laravel untuk membuatnya menggunakan lang:publish perintah Artisan. Di dalam lang/en/validation.php file tersebut, Anda akan menemukan entri terjemahan</p>

	untuk setiap aturan validasi. Anda bebas mengubah atau memodifikasi pesan-pesan ini berdasarkan kebutuhan aplikasi Anda.
10.	<p>Pada view/create.blade.php tambahkan dan coba running code berikut :</p> <pre> 16 <div class="card-body"> 17 <div class="form-group"> 18 <label for="kodeKategori">Kode Kategori</label> 19 <input 20 type="text" 21 class="@error('kodeKategori') is-invalid @enderror" 22 id="kodeKategori" 23 name="kodeKategori" 24 placeholder="untuk makanan, contoh: MKN"> 25 @error('kodeKategori') 26 <div class="alert alert-danger">{{\$message}}</div> 27 @enderror </pre>

C. FORM REQUEST VALIDATION

1.	<p>Membuat permintaan form dengan menuliskan pada terminal</p> <pre> PS C:\laragon\www\Pemrogramman Web Lanjut\Pertemuan2\POS> php artisan make:request StorePostRequest </pre> <p>INFO Request [C:\laragon\www\Pemrogramman Web Lanjut\Pertemuan2\POS\app\Http\Requests\StorePostRequest.php] created successfully.</p> <p>kelas permintaan formulir yang dihasilkan akan ditempatkan di app/Http/Requestsdirektori. 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.</p>
2.	<p>Ketik kode berikut pada Http/request/StorePostRequest</p> <pre> 24 public function rules(): array 25 { 26 return [27 'kodeKategori' => 'required', 28 'namaKategori' => 'required', 29]; 30 } </pre> <ul style="list-style-type: none"> • Memodifikasi pada kategorikontroller

```

21 public function store(StorePostRequest $request): RedirectResponse{
22
23     // The incoming request is valid
24
25     //Retrive the validated input data
26     $validated = $request->validate();
27
28     //Retrive a portion of the validated input data
29     $validated = $request->safe()->only(['kodeKategori', 'namaKategori']);
30     $validated = $request->safe()->except(['kodeKategori', 'namaKategori']);
31
32     // store the post
33
34     return redirect('/kategori');
35
36 }

```

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.

- Tambahan validate ke m_user

```

29 public function store(Request $request) : RedirectResponse {
30     $validate = $request->validate([
31         'username' => 'required',
32         'namaLevel' => 'required',
33     ]);
34
35     $request->validate([
36         'title' => 'bail|required|unique:posts|max:255',
37         'body' => 'required',
38     ]);
39
40     /*$validateData = $request->validate([
41         'title' => ['required', 'unique:posts', 'max:255'],
42         'body' => ['required'],
43     ]);*/
44
45     /*$validateData = $request->validateWithBag('post', [
46         'title' => ['required', 'unique:posts', 'max:255'],
47         'body' => ['required'],
48     ]);*/
49
50     UserModel::create([
51         'level_kode' => $request->kodeLevel,
52         'level_nama' => $request->namaLevel,
53     ]);
54     return redirect('/user');
55 }

```

- Menambahkan keterangan error

```
Pertemuan2 > POS > resources > views > user > create_user.blade.php > div.container >
8      <div class="container">
9      <div class="card card-primary">
32      {{--MENAMBAHKAN ERROR YANG TERJADI --}}
33      @if ($errors->any())
34      <div class="alert alert-danger">
35      <ul>
36          @foreach ($errors->all() as $error)
37              <li>{{$error}}</li>
38          @endforeach
39      </ul>
40      </div>
41      @endif
```

- Tambahan validate ke m_level

```
26      public function store(Request $request) : RedirectResponse {
27          $validate = $request->validate([
28              'kodeLevel' => 'required',
29              'namaLevel' => 'required',
30          ]);
31
32          $request->validate([
33              'title' => 'bail|required|unique:posts|max:255',
34              'body' => 'required',
35          ]);
36
37          /*$validateData = $request->validate([
38              'title' => ['required', 'unique:posts', 'max:255'],
39              'body' => ['required'],
40          ]);*/
41
42          /*$validateData = $request->validateWithBag('post', [
43              'title' => ['required', 'unique:posts', 'max:255'],
44              'body' => ['required'],
45          ]);*/
46
47          LevelModel::create([
48              'level_kode' => $request->kodeLevel,
49              'level_nama' => $request->namaLevel,
50          ]);
51          return redirect('/level');
52      }
```

- Menambahkan keterangan error

```
Pertemuan2 > POS > resources > views > level > create_level.blade.php > div.container >
8      <div class="container">
9      <div class="card card-primary">
32      {{--MENAMBAHKAN ERROR YANG TERJADI --}}
33      @if ($errors->any())
34      <div class="alert alert-danger">
35      <ul>
36          @foreach ($errors->all() as $error)
37              <li>{{$error}}</li>
38          @endforeach
39      </ul>
40      </div>
41      @endif
```

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

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\Pemrogramman Web Lanjut\Pertemuan2\POS> php artisan make:controller POSController --resource</pre> <p>INFO Controller [C:\laragon\www\Pemrogramman Web Lanjut\Pertemuan2\POS\app\Http\Controllers\POSController.php] created successfully.</p>
2.	<p>Berikut perintahnya: sekarang tambahkan kode pada route/web.php</p> <pre>81 Route::resource('m_user', POSController::class);</pre>
3.	<p>Atur pada Models/m_user (UserModel)</p> <pre> 9 class UserModel extends Model 10 { 11 use HasFactory; 12 13 protected \$table = 'm_user'; 14 public \$timestamps = false; 15 protected \$primaryKey = 'user_id'; 16 17 protected \$fillable = [18 'user_id', 19 'level_id', 20 'username', 21 'nama', 22 'password']; </pre>
4.	<p>Atur pada Migration/m_user_table</p>

	<pre> n2 > POS > database > migrations > 2024_03_03_011002_create_m_user_table.php > cla 7 rn new class extends Migration 9 /** 10 * Run the migrations. 11 */ 12 public function up(): void 13 { 14 Schema::create('m_user', function (Blueprint \$table) { 15 \$table->id('user_id'); 16 \$table->unsignedBigInteger('level_id')->index(); 17 \$table->string('username', 20)->unique(); 18 \$table->string('nama', 100); 19 \$table->string('password'); 20 \$table->timestamps(); 21 22 \$table->foreign('level_id')->references('level_id') 23 ->on('m_level'); 24 }); 25 } 26 27 /** 28 * Reverse the migrations. 29 */ 30 public function down(): void 31 { 32 Schema::dropIfExists('useri'); 33 } </pre>
5.	<p>Pada file Controller yang kita dapat secara otomatis akan terdapat 7 fungsi berikut yang dapat kita gunakan untuk membuat operasi CRUD.</p> <ul style="list-style-type: none"> • index() • create() • store() • show() • edit() • update() • destroy()
6.	<p>Selanjutnya bukalah app/Http/Controllers/POSController.php kemudian ketikkan, kodenya seperti berikut ini:</p>

Pertemuan2 > POS > app > Http > Controllers > POSController.php > POSController > destroy

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\User;
6  use App\Models\UserModel;
7  use Illuminate\Http\Request;
8
9  use function PHPUnit\Framework\returnSelf;
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 = UserModel::all(); //Mengambil semua isi tabel
20         return view('user.index',
21             compact('useri'))->with('i');
22     }
23
24     /**
25      * Show the form for creating a new resource.
26      */
27     public function create()
28     {
29         return view('user.create_user');
30     }
31
32     /**
```

```

33      * Store a newly created resource in storage.
34      */
35      public function store(Request $request)
36      {
37          //Melakukan validasi
38          $request->validate([
39              'user_id' => 'max 20',
40              'username' => 'required',
41              'nama' => 'required',
42          ]);
43
44          //fungsi eloquent untuk menambah data
45          UserModel::create($request->all());
46
47          return redirect()->route('user.index')
48              ->with('succes', 'user berhasil ditambahkan');
49      }
50
51      /**
52       * Display the specified resource.
53       */
54      public function show(string $id, UserModel $useri)
55      {
56          $useri = UserModel::findOrFail($id);
57          return view('user.show', compact('useri'));
58      }
59
60      /**
61       * Show the form for editing the specified resource.
62       */
63      public function edit(string $id)

```

```

63      public function edit(string $id)
64      {
65          $useri = UserModel::find($id);
66          return view('user.edit', compact('useri'));
67      }
68
69      /**
70       * Update the specified resource in storage.
71       */
72      public function update(Request $request, string $id)
73      {
74          $request->validate([
75              'username' => 'required',
76              'nama' => 'required',
77              'password' => 'required',
78          ]);
79
80          //fungsi eloquent untuk mengupdate data inputan kita
81          UserModel::find($id)->update($request->all());
82
83          //Jika data berhasil di update, maka kembali ke halaman utama
84          return redirect()->route('user.index')
85              ->with('success', 'Data Berhasil Di Update');
86      }
87
88      /**
89       * Remove the specified resource from storage.
90       */
91      public function destroy(string $id)
92      {
93          $useri = UserModel::findOrFail($id)->delete();

```

	<pre> 94 return redirect()->route('user.index') 95 ->with('success', 'Data Berhasil Di Hapus'); 96 } 97 } 98 </pre>
7.	<p>Buatlah folder di Resources/Views/m_user dengan beberapa blade dan isian kode berikut:</p> <ol style="list-style-type: none"> template.blade.php <div data-bbox="450 542 1342 788" data-label="Text"> <pre> Pertemuan2 > POS > resources > views > user > template.blade.php > html > body > div.container 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <title>CRUD Laravel</title> 5 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"> 6 </head> 7 <body> 8 <div class="container"> 9 @yield('content') 10 </div> 11 </body> 12 </html> </pre> </div> index.blade.php

Pertemuan2 > POS > resources > views > m_user > index.blade.php > table.table-bordered > tr > th

```
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>CRUD user</h2>
7  </div>
8  <div class="float-right">
9  <a class="btn btn-success" href="{{ route('m_user.create') }}"> Input
10 User</a>
11 </div>
12 </div>
13 </div>
14 @if ($message = Session::get('success'))
15 <div class="alert alert-success">
16 <p>{{ $message }}</p>
17 </div>
18 @endif
19 <table class="table table-bordered">
20 <tr>
21 <th width="20px" class="text-center">User id</th>
22 <th width="15px" class="text-center">Level id</th>
23 <th width="20px" class="text-center">username</th>
24 <th width="20px" class="text-center">nama</th>
25 <th width="15px" class="text-center">password</th>
26 </tr>
27 @foreach ($useri as $m_user)
28 <tr>
29
30 <td>{{ $m_user->user_id }}</td>
31 <td>{{ $m_user->level_id }}</td>
32 <td>{{ $m_user->username }}</td>
33 <td>{{ $m_user->nama }}</td>
34 <td>{{ $m_user->password }}</td>
35
36 <td class="text-center">
37 <form action="{{ route('m_user.destroy',$m_user->user_id) }}"
38 method="POST">
39 <a class="btn btn-info btn-sm" href="{{ route('m_user.show',$m_user->user_id) }}">Show</a>
40 <a class="btn btn-primary btn-sm" href="{{
41 route('m_user.edit',$m_user->user_id) }}">Edit</a>
42 @csrf
43 @method('DELETE')
44 <button type="submit" class="btn btn-danger btn-sm onclick="return
45 confirm('Apakah Anda yakin ingin menghapus data
46 ini?')">Delete</button>
47 </form>
48 </td>
49 </tr>
50 @endforeach
51 </table>
52 @endsection
```

3. create.blade.php

Pertemuan2 > POS > resources > views > m_user > create.blade.php > div.row.mt-5.mb-5

```
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') }}">
10 Kembali</a>
11 </div>
12 </div>
13 </div>
14 @if ($errors->any())
15 <div class="alert alert-danger">
16 <strong>Ops</strong> Input gagal<br><br>
17 <ul>
18 @foreach ($errors->all() as $error)
19 <li>{{ $error }}</li>
20 @endforeach
21 </ul>
22 </div>
23 @endif
24 <form action="{{ route('m_user.store') }}" method="POST">
25 @csrf
26
27
28 <div class="col-xs-12 col-sm-12 col-md-12">
29 <div class="form-group">
30 <strong>Username:</strong>
31 <input type="text" name="username" class="form-control"
32 placeholder="Masukkan username"></input>
33 </div>
34 </div>
35
36 <div class="col-xs-12 col-sm-12 col-md-12">
37 <div class="form-group">
38 <strong>nama:</strong>
39 <input type="text" name="nama" class="form-control"
40 placeholder="Masukkan nama"></input>
41 </div>
42 </div>
43
44
45 <div class="col-xs-12 col-sm-12 col-md-12">
46 <div class="form-group">
47 <strong>Password:</strong>
48 <input type="password" name="password" class="form-control"
49 placeholder="Masukkan password"></input>
50 </div>
51 </div>
52 <div class="col-xs-12 col-sm-12 col-md-12 text-center">
53 <button type="submit" class="btn btn-primary">Submit</button>
54 </div>
55 </div>
56 </form>
57 @endsection
```

4. show.blade.php

Pertemuan2 > POS > resources > views > m_user > show.blade.php > div.row > div.col-xs-12.col-sm-12

```
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> Show 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 <div class="row">
15 <div class="col-xs-12 col-sm-12 col-md-12">
16 <div class="form-group">
17 <strong>User_id:</strong>
18 {{ $useri->user_id }}
19 </div>
20 </div>
21 <div class="col-xs-12 col-sm-12 col-md-12">
22 <div class="form-group">
23 <strong>Level_id:</strong>
24 {{ $useri->level_id }}
25 </div>
26 </div>
27
28 <div class="col-xs-12 col-sm-12 col-md-12">
29 <div class="form-group">
30 <strong>Username:</strong>
31 {{ $useri->username }}
32 </div>
```

```
33 </div>
34
35 <div class="col-xs-12 col-sm-12 col-md-12">
36 <div class="form-group">
37 <strong>Nama:</strong>
38 {{ $useri->nama }}
39 </div>
40 </div>
41
42 <div class="col-xs-12 col-sm-12 col-md-12">
43 <div class="form-group">
44 <strong>Password:</strong>
45 {{ $useri->password }}
46 </div>
47 </div>
48
49 </div>
50 @endsection
```

5. edit.blade.php

	<pre> Pertemuan2 > POS > resources > views > m_user > edit.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 10 Kembali 11 </div> 12 </div> 13 </div> 14 @if (\$errors->any()) 15 <div class="alert alert-danger"> 16 Ops! Error

 17 18 @foreach (\$errors->all() as \$error) 19 {{ \$error }} 20 @endforeach 21 22 </div> 23 @endif 24 <form action="{{ route('m_user.update', \$user->user_id) }}" 25 method="POST"> 26 @csrf 27 @method('PUT') 28 <div class="row"> 29 <div class="col-xs-12 col-sm-12 col-md-12"> 30 <div class="form-group"> 31 User_id: 32 <input type="text" name="userid" value="{{ \$user->user_id }}" 33 class="form-control" placeholder="Masukkan user id"> 34 </div> 35 </div> 36 37 <div class="col-xs-12 col-sm-12 col-md-12"> 38 <div class="form-group"> 39 Level_id: 40 <input type="text" name="levelid" value="{{ \$user->level_id }}" 41 class="form-control" placeholder="Masukkan level"> 42 </div> 43 </div> 44 45 <div class="col-xs-12 col-sm-12 col-md-12"> 46 <div class="form-group"> 47 Username: 48 <input type="text" value= "{{ \$user->username }}" class="form 49 control" name="username" placeholder="Masukkan Nomor username"> 50 </div> 51 </div> 52 53 <div class="col-xs-12 col-sm-12 col-md-12"> 54 <div class="form-group"> 55 nama: 56 <input type="text" value= "{{ \$user->nama }}" name="nama" 57 class="form-control" placeholder="Masukkan nama"></input> 58 </div> 59 </div> 60 61 62 <div class="col-xs-12 col-sm-12 col-md-12"> 63 <div class="form-group"> 64 Password: 65 <input type="password" value= "{{ \$user->password 66 }}" name="password" class="form-control" placeholder="Masukkan 67 password"></input> 68 </div> 69 </div> 70 <div class="col-xs-12 col-sm-12 col-md-12 text-center"> 71 <button type="submit" class="btn btn-primary">Update</button> 72 </div> 73 </div> 74 </form> 75 @endsection </pre>
8.	Silahkan akses localhost/m_user

	<div><div>CRUD Laravel</div><div><div>127.0.0.1:8000/m_user</div><div>☆</div><div></div><div></div><div></div><div></div></div></div> <div><div>CRUD user</div><div>Input User</div><table><tr><th>User Id</th><th>Level Id</th><th>username</th><th>nama</th><th>password</th><th></th></tr><tr><td>1</td><td>1</td><td>admin</td><td>admin ubah</td><td>\$2y\$12\$Sjs.WqcOrQZkutru.1SROvQ7oLsdSVXOX7C6GBEB5f11zE5qNm</td><td>Show Edit Delete</td></tr><tr><td>2</td><td>2</td><td>manager</td><td>Manager</td><td>\$2y\$12\$I92xlwKH5bna6v8y2Vg7WuFJGenEkClkWIDwHfaNvGEy.2vv4D3rG</td><td>Show Edit Delete</td></tr><tr><td>3</td><td>3</td><td>staff</td><td>Staff/Kasir</td><td>\$2y\$12\$XWUV37KLIdItIuhUGnMhzuPyEj51XH0Kqve/12IFC56IUU3qqLLPy</td><td>Show Edit Delete</td></tr><tr><td>9</td><td>2</td><td>manager22</td><td>Manager Dua Dua</td><td>\$2y\$12\$75B.lhzoKkVZjCMI5D0b7eqVuHiiu7eymDm8Cwdbi56Q1vH522uom</td><td>Show Edit Delete</td></tr><tr><td>11</td><td>2</td><td>manager_tiga</td><td>Manager 3</td><td>\$2y\$12\$.5tWSr52geegaQiYt8RqUOFTSbkeXzVS.MIT9X/xOWDsgxR81DWu.</td><td>Show Edit Delete</td></tr><tr><td>12</td><td>2</td><td>manager33</td><td>Manager Tiga Tiga</td><td>\$2y\$12\$t5qZVqUfVWAQdr7h.FvDyuT5TctIWpNRjD8dW0DuDm5wxnfV5T6HG</td><td>Show Edit Delete</td></tr><tr><td>14</td><td>2</td><td>manaer56</td><td>Manaer55</td><td>\$2v\$12\$WH63DBIIUaR2Hr0/7bZ7//rD3//l7xxMKii1k6.ZR962oeVnwKUi</td><td>Show Edit Delete</td></tr></table></div>	User Id	Level Id	username	nama	password		1	1	admin	admin ubah	\$2y\$12\$Sjs.WqcOrQZkutru.1SROvQ7oLsdSVXOX7C6GBEB5f11zE5qNm	Show Edit Delete	2	2	manager	Manager	\$2y\$12\$I92xlwKH5bna6v8y2Vg7WuFJGenEkClkWIDwHfaNvGEy.2vv4D3rG	Show Edit Delete	3	3	staff	Staff/Kasir	\$2y\$12\$XWUV37KLIdItIuhUGnMhzuPyEj51XH0Kqve/12IFC56IUU3qqLLPy	Show Edit Delete	9	2	manager22	Manager Dua Dua	\$2y\$12\$75B.lhzoKkVZjCMI5D0b7eqVuHiiu7eymDm8Cwdbi56Q1vH522uom	Show Edit Delete	11	2	manager_tiga	Manager 3	\$2y\$12\$.5tWSr52geegaQiYt8RqUOFTSbkeXzVS.MIT9X/xOWDsgxR81DWu.	Show Edit Delete	12	2	manager33	Manager Tiga Tiga	\$2y\$12\$t5qZVqUfVWAQdr7h.FvDyuT5TctIWpNRjD8dW0DuDm5wxnfV5T6HG	Show Edit Delete	14	2	manaer56	Manaer55	\$2v\$12\$WH63DBIIUaR2Hr0/7bZ7//rD3//l7xxMKii1k6.ZR962oeVnwKUi	Show Edit Delete
User Id	Level Id	username	nama	password																																													
1	1	admin	admin ubah	\$2y\$12\$Sjs.WqcOrQZkutru.1SROvQ7oLsdSVXOX7C6GBEB5f11zE5qNm	Show Edit Delete																																												
2	2	manager	Manager	\$2y\$12\$I92xlwKH5bna6v8y2Vg7WuFJGenEkClkWIDwHfaNvGEy.2vv4D3rG	Show Edit Delete																																												
3	3	staff	Staff/Kasir	\$2y\$12\$XWUV37KLIdItIuhUGnMhzuPyEj51XH0Kqve/12IFC56IUU3qqLLPy	Show Edit Delete																																												
9	2	manager22	Manager Dua Dua	\$2y\$12\$75B.lhzoKkVZjCMI5D0b7eqVuHiiu7eymDm8Cwdbi56Q1vH522uom	Show Edit Delete																																												
11	2	manager_tiga	Manager 3	\$2y\$12\$.5tWSr52geegaQiYt8RqUOFTSbkeXzVS.MIT9X/xOWDsgxR81DWu.	Show Edit Delete																																												
12	2	manager33	Manager Tiga Tiga	\$2y\$12\$t5qZVqUfVWAQdr7h.FvDyuT5TctIWpNRjD8dW0DuDm5wxnfV5T6HG	Show Edit Delete																																												
14	2	manaer56	Manaer55	\$2v\$12\$WH63DBIIUaR2Hr0/7bZ7//rD3//l7xxMKii1k6.ZR962oeVnwKUi	Show Edit Delete																																												
	Tugas																																																
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																																																