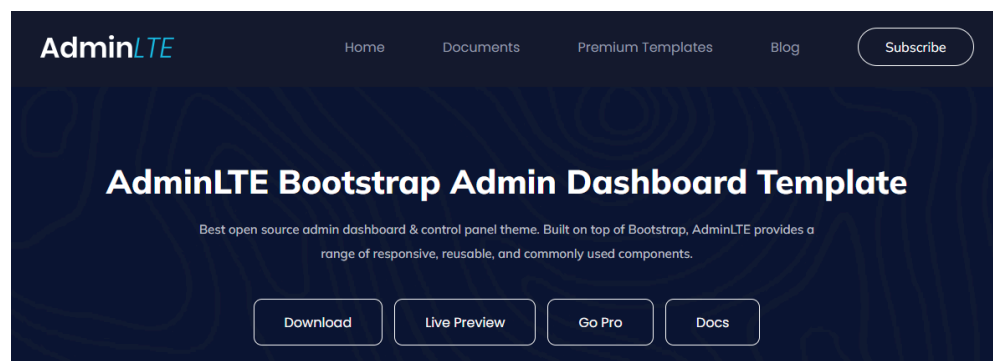


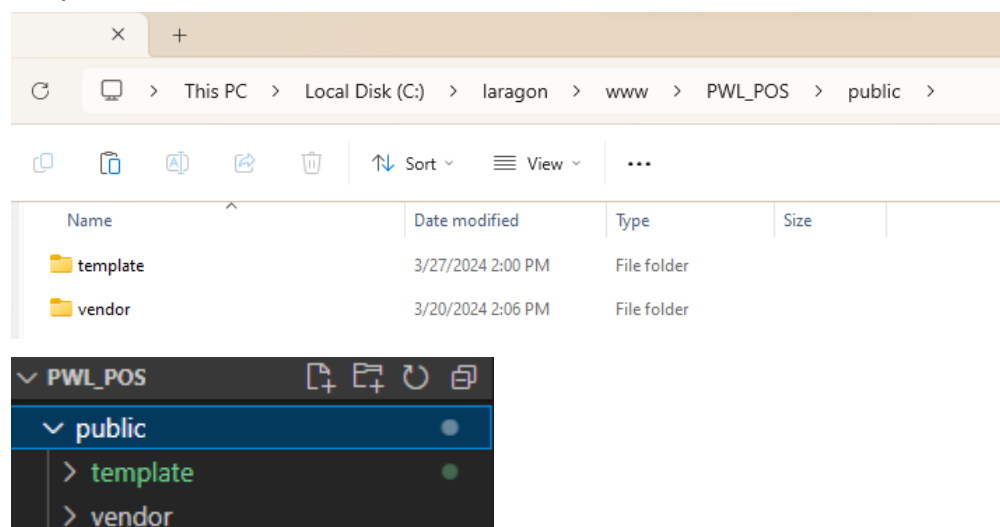
NIM : 2241720095
Nama : Ana Bellatus Mustaqfiro
Kelas : D4 TI - 2H
No. Urut : 04
Mata Kuliah : Pemrograman Web Lanjut Jobsheet 6 (Template Form, Server Validation, Client Validation, CRUD)

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_2024/public. Buka folder public pada VSCode maka akan tampil folder baru bernama template



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

```
resources > views > welcome.blade.php > html > head
31 <html lang="en">
32 <head>
33
34 <!-- Google Font: Source Sans Pro -->
35 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
36 <!-- Font Awesome Icons -->
37 <link rel="stylesheet" href="{{ asset('template/plugins/fontawesome-free/css/all.min.css') }}">
38 <!-- overlayScrollbars -->
39 <link rel="stylesheet" href="{{ asset('template/plugins/overlayScrollbars/css/OverlayScrollbars.min.css') }}">
40 <!-- Theme style -->
41 <link rel="stylesheet" href="{{ asset('template/dist/css/adminlte.min.css') }}">
42 </head>
43 <body class="hold-transition dark-mode sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">
44 <div class="wrapper">
45
46 <!-- Preloader -->
47 <div class="preloader flex-column justify-content-center align-items-center">
48 
49 </div>
```

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

```
resources > views > welcome.blade.php > ...
31 <html lang="en">
46 <body class="hold-transition dark-mode sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">
1740 <!-- ./wrapper -->
1741
1742 <!-- REQUIRED SCRIPTS -->
1743 <!-- jQuery -->
1744 <script src="{{ asset('template/plugins/jquery/jquery.min.js') }}"></script>
1745 <!-- Bootstrap -->
1746 <script src="{{ asset('template/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
1747 <!-- overlayScrollbars -->
1748 <script src="{{ asset('template/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js') }}"></script>
1749 <!-- AdminLTE App -->
1750 <script src="{{ asset('template/dist/js/adminlte.js') }}"></script>
1751
1752 <!-- PAGE PLUGINS -->
1753 <!-- jQuery Mapael -->
1754 <script src="{{ asset('template/plugins/jquery-mousewheel/jquery.mousewheel.js') }}"></script>
1755 <script src="{{ asset('template/plugins/raphael/raphael.min.js') }}"></script>
1756 <script src="{{ asset('template/plugins/jquery-mapael/jquery.mapael.min.js') }}"></script>
1757 <script src="{{ asset('template/plugins/jquery-mapael/maps/usa_states.min.js') }}"></script>
1758 <!-- ChartJS -->
1759 <script src="{{ asset('template/plugins/chart.js/Chart.min.js') }}"></script>
1760
1761 <!-- AdminLTE for demo purposes -->
1762 <script src="{{ asset('template/dist/js/demo.js') }}"></script>
1763 <!-- AdminLTE dashboard demo (This is only for demo purposes) -->
1764 <script src="{{ asset('template/dist/js/pages/dashboard2.js') }}"></script>
```

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

```
103 
```

5. Jalankan pada browser



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

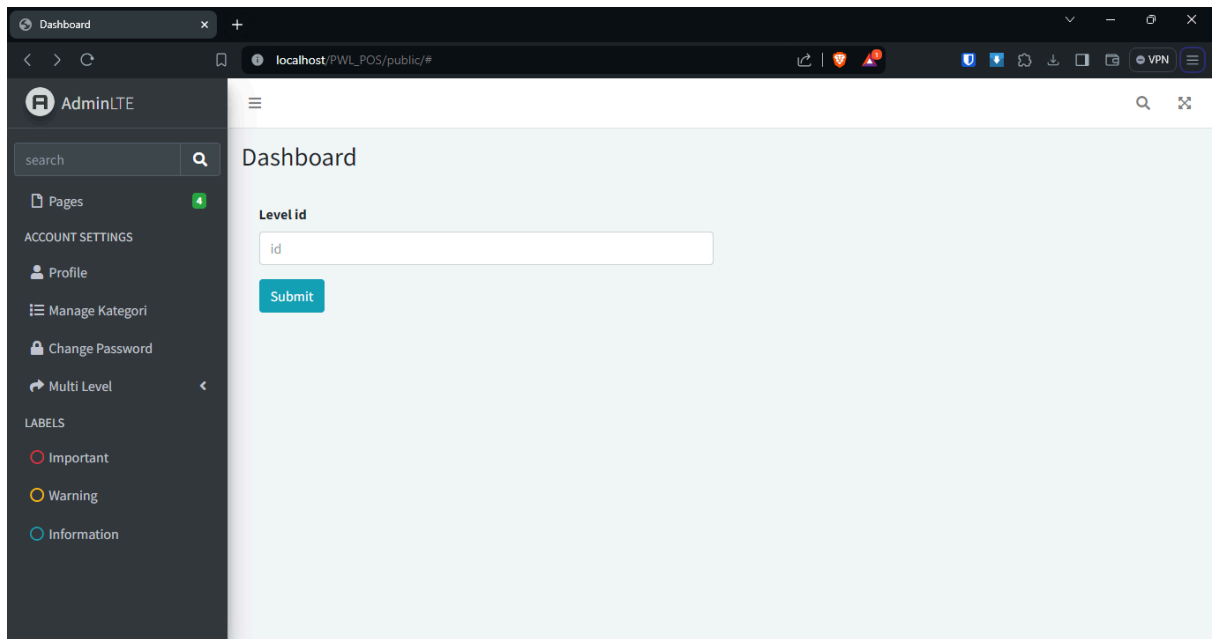
- General Elements
- Advanced Elements
- Editor
- Validation

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

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

```
welcome.blade.php M X
resources > views > welcome.blade.php > div.card-body > form > button.btn.btn-info
1  @extends('adminlte::page')
2
3  @section('title', 'Dashboard')
4
5  @section('content_header')
6      <h1>Dashboard</h1>
7  @stop
8  @section('content')
9      <div class="card-body">
10         <form>
11             <div class="row">
12                 <div class="col-sm-6">
13                     <!-- text input -->
14                     <div class="form-group">
15                         <label>Level id</label>
16                         <input type="text" class="form-control" placeholder="id">
17                     </div>
18                 </div>
19             </div>
20             <button type = "submit" class = "btn btn-info">Submit </button>
21         </form>
22     </div>
23 @stop
24
25 @section('css')
26     {{-- Add here extra stylesheets --}}
27     {{-- <link rel="stylesheet" href="/css/admin_custom.css"> --}}
28 @stop
29
30 @section('js')
31     <script>
32         console.log("Hi, I'm using the Laravel-AdminLTE package!");
33     </script>
34 @stop
```

8. Apa yang tampil?



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

a. User

```
config > adminlte.php
3   return [
296       'menu' => [
318           [
324               'submenu' => [
329                   [
330                       'text' => 'User',
331                       'url' => '/user',
332                   ],

```

```
PS C:\laragon\www\PWL_POS> php artisan datatables:make User
```

```
INFO DataTable [C:\laragon\www\PWL_POS\app\DataTables\UserDataTable.php] created successfully.
```

```
app > DataTables > UserDataTable.php > ...
13 class UserDataTable extends DataTable
14 {
15     /**
16      * Build the DataTable class.
17      *
18      * @param QueryBuilder $query Results from query() method.
19      */
20     public function dataTable(QueryBuilder $query): EloquentDataTable
21     {
22         return (new EloquentDataTable($query))
23             ->setRowId('id')
24             ->addColumn('action', function ($row) {
25                 $editButton = '<a href="" . route('user.edit', $row->user_id) . "" class="btn btn-primary btn-sm"><i class="fas
fa-edit"></i></a>&nbsp';
26
27                 $deleteButton = '<a href="" . route('user.delete', $row->user_id) . "" class="btn btn-danger btn-sm" onclick="return
confirm('\Apakah Anda yakin ingin menghapus user ini?\'")><i class="fa fa-trash"></i></a>';
28
29                 return $editButton . ' ' . $deleteButton;
30             });
31     }

```

```

app > DataTables > UserDataTable.php > ...
44     public function html(): HtmlBuilder
45     {
46         return $this->builder()
47             ->setTableId('user-table')
48             ->columns($this->getColumns())
49             ->minifiedAjax()
50             //->dom('Bfrtip')
51             ->orderBy(0, 'asc')
52             ->selectStyleSingle()
53             ->buttons([
54                 Button::make('excel'),
55                 Button::make('csv'),
56                 Button::make('pdf'),
57                 Button::make('print'),
58                 Button::make('reset'),
59                 Button::make('reload')
60             ]);
61     }

```

```

app > DataTables > UserDataTable.php > ...
66     public function getColumns(): array
67     {
68         return [
69             // Column::computed('action')
70             //     ->exportable(false)
71             //     ->printable(false)
72             //     ->width(60)
73             //     ->addClass('text-center'),
74             Column::make('user_id'),
75             Column::make('level_id'),
76             Column::make('username'),
77             Column::make('nama'),
78             Column::make('created_at'),
79             Column::make('updated_at'),
80             Column::make('action'),
81         ];
82     }

```

```

app > Models > UserModel.php > ...
9     class UserModel extends Model
10     {
11         use HasFactory;
12
13         protected $table = 'm_user';
14         protected $primaryKey = 'user_id';
15
16         protected $fillable = ['level_id', 'username', 'password', 'nama'];
17
18         public function level(): BelongsTo
19         {
20             return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
21         }
22     }

```

```

app > Http > Controllers > UserController.php > ...
11 class UserController extends Controller
12 {
13     public function index(UserDataTable $dataTable)
14     {
15         return $dataTable->render('user.index');
16     }
17
18     public function create()
19     {
20         return view('user.create');
21     }
22
23     public function store(Request $request)
24     {
25         UserModel::create([
26             'nama' => $request->nama,
27             'level_id' => $request->levelId,
28             'username' => $request->username,
29             'password' => Hash::make($request->password),
30         ]);
31
32         return redirect('/user');
33     }
34
35     public function edit($id)
36     {
37         $user = UserModel::find($id);
38         $levels = LevelModel::all();
39         return view('user.edit', compact('user', 'levels'));
40     }

```

```

app > Http > Controllers > UserController.php > ...
42     public function update(Request $request, $id)
43     {
44         $user = UserModel::find($id);
45
46         $user->nama = $request->nama;
47         $user->level_id = $request->levelId;
48         $user->username = $request->username;
49         $user->password = Hash::make($request->password);
50         $user->save();
51
52         return redirect('/user')->with('success', 'User berhasil diupdate');
53     }
54
55     public function delete($id)
56     {
57         $user = UserModel::find($id);
58         if (!$user) {
59             abort(404);
60         }
61
62         $user->delete();
63
64         return redirect('/user')->with('success', 'User berhasil dihapus');
65     }
66
67     public function getLevel(){
68         $levels = LevelModel::all();
69         return view('user.create', ['levels' => $levels]);
70     }

```

```
resources > views > user > create.blade.php > ...
1  @extends('layouts.app')
2
3  @section('subtitle', 'Create User')
4  @section('content_header_title', 'User')
5  @section('content_header_subtitle', 'Create')
6
7  @section('content')
8      <div class="container">
9          <div class="card card-primary">
10             <div class="card-header">
11                 <h3 class="card-title">Buat user baru</h3>
12             </div>
13
14             <form method="post" action=".../user">
15                 <div class="card-body">
16                     <div class="form-group">
17                         <label for="nama">Nama</label>
18                         <input type="text" class="form-control" id="nama" name="nama" placeholder=" Nama">
19                     </div>
20
21                     <div class="form-group">
22                         <label for="levelId">Level Id</label>
23                         <select class="form-control" id="levelId" name="levelId">
24                             @foreach ($levels as $level)
25                                 <option value="{{ $level->level_id }}">{{ $level->level_nama }}</option>
26                             @endforeach
27                         </select>
28                     </div>

```

```

29                 <div class="form-group">
30                     <label for="username">Username</label>
31                     <input type="text" class="form-control" id="username" name="username" placeholder=" Username">
32                 </div>
33
34                 <div class="form-group">
35                     <label for="password">Password</label>
36                     <input type="password" class="form-control" id="password" name="password" placeholder=" Password">
37                 </div>
38             </div>
39             <div class="card-footer">
40                 <button type="submit" class="btn btn-primary">Submit</button>
41             </div>
42         </form>
43     </div>
44 </div>
45 @endsection
46

```

```
resources > views > user > edit.blade.php > ...
1  @extends('layouts.app')
2
3  @section('subtitle', 'User')
4  @section('content_header_title', 'User')
5  @section('content_header_subtitle', 'Edit')
6
7  @section('content')
8      <div class="container">
9          <div class="card card-primary">
10             <div class="card-header">
11                 <h3 class="card-title">Edit User</h3>
12             </div>
13
14             <form method="post" action="{{ route('user.update', $user->user_id) }}">
15                 @csrf
16                 @method('PUT')
17             </form>

```

```

24             <div class="form-group">
25                 <label for="namaKategori">level Id</label>
26                 <select class="form-control" id="levelId" name="levelId">
27                     @foreach ($levels as $level)
28                         <option value="{{ $level->level_id }}" @if($user->level_id == $level->level_id) selected @endif>{{ $level->level_nama }}</option>
29                     @endforeach
30                 </select>
31             </div>
32
33             <div class="form-group">
34                 <label for="kodeKategori">Username</label>
35                 <input type="text" class="form-control" id="username" name="username" value="{{ $user->username }}">
36             </div>
37
38             <div class="form-group">
39                 <label for="password">Password</label>
40                 <input type="password" class="form-control" id="password" name="password" value="{{ $user->password }}">
41             </div>
42         </div>
43         <div class="card-footer">
44             <button type="submit" class="btn btn-primary">Submit</button>
45         </div>
46     </form>
47 </div>
48 </div>
49 @endsection
50

```

```

routes > web.php > ...
28 Route::get('/user', [UserController::class, 'index']);
29 Route::get('/user/create', [UserController::class, 'create'])->name('user.create');
30 Route::get('/user/create', [UserController::class, 'getLevel'])->name('user.create');
31 Route::post('/user', [UserController::class, 'store']);
32 Route::get('/user/edit/{id}', [UserController::class, 'edit'])->name('user.edit');
33 Route::put('/user/update/{id}', [UserController::class, 'update'])->name('user.update');
34 Route::get('/user/delete/{id}', [UserController::class, 'delete'])->name('user.delete');

```

AdminLTE 3 | Create User

localhost/PWL_POS/public/user/create

AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

User > Create

Buat user baru

Nama

Nama

Level Id

Administrator

Username

Username

Password

Password

Submit

AdminLTE 3 | User

localhost/PWL_POS/public/user/edit/19

AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

User > Edit

Edit User

Nama

Prof. Beaulah Boyer V

Level Id

Manager

Username

boyerv

Password

Submit

b. Level

```
config > adminlte.php
3   return [
296       'menu' => [
318           [
324               'submenu' => [
332                   ],
333                   [
334                       'text' => 'Level',
335                       'url' => '/ level',
336                   ],

```

```
PS C:\laragon\www\PWL_POS> php artisan datatables:make Level
```

INFO DataTable [C:\laragon\www\PWL_POS\app\DataTables\LevelDataTable.php] created successfully.

```
app > DataTables > LevelDataTable.php > ...
14 class LevelDataTable extends DataTable
15 {
16     /**
17      * Build the DataTable class.
18      *
19      * @param QueryBuilder $query Results from query() method.
20      */
21     public function dataTable(QueryBuilder $query): EloquentDataTable
22     {
23         return (new EloquentDataTable($query))
24             ->setRowId('id')
25             ->addColumn('action', function ($row) {
26                 $editButton = '<a href="' . route('level.edit', $row->level_id) . '" class="btn btn-primary btn-sm"><i class="fas fa-edit"></i> </a>&nbsp;<a href="' . route('level.delete', $row->level_id) . '" class="btn btn-danger btn-sm" onclick="return confirm('\Apakah Anda yakin ingin menghapus level ini?')"><i class="fa fa-trash"></i></a>';
27                 $deleteButton = '<a href="' . route('level.delete', $row->level_id) . '" class="btn btn-danger btn-sm" onclick="return confirm('\Apakah Anda yakin ingin menghapus level ini?')"><i class="fa fa-trash"></i></a>';
28                 return $editButton . ' ' . $deleteButton;
29             });
30     }
31 }
32 }
```

```
app > DataTables > LevelDataTable.php > ...
66     /**
67     public function getColumns(): array
68     {
69         return [
70             // Column::computed('action')
71             //     ->exportable(false)
72             //     ->printable(false)
73             //     ->width(60)
74             //     ->addClass('text-center'),
75             Column::make('level_id'),
76             Column::make('level_nama'),
77             Column::make('created_at'),
78             Column::make('updated_at'),
79             Column::make('action'),
80         ];
81     }
```

```
app > Models > LevelModel.php > LevelModel
14
15     protected $fillable = ['level_kode', 'level_nama'];
16 }
```

```

app > Http > Controllers > LevelController.php > ...
9
10 class LevelController extends Controller
11 {
12     public function index(LevelDataTable $dataTable)
13     {
14         return $dataTable->render('level.index');
15     }
16
17     public function create()
18     {
19         return view('level.create');
20     }
21
22     public function store(Request $request)
23     {
24         LevelModel::create([
25             'level_kode' => $request->levelKode,
26             'level_nama' => $request->levelNama,
27         ]);
28
29         return redirect('/level');
30     }
31
32     public function edit($id)
33     {
34         $level = LevelModel::find($id);
35         return view('level.edit', compact('level'));
36     }

```

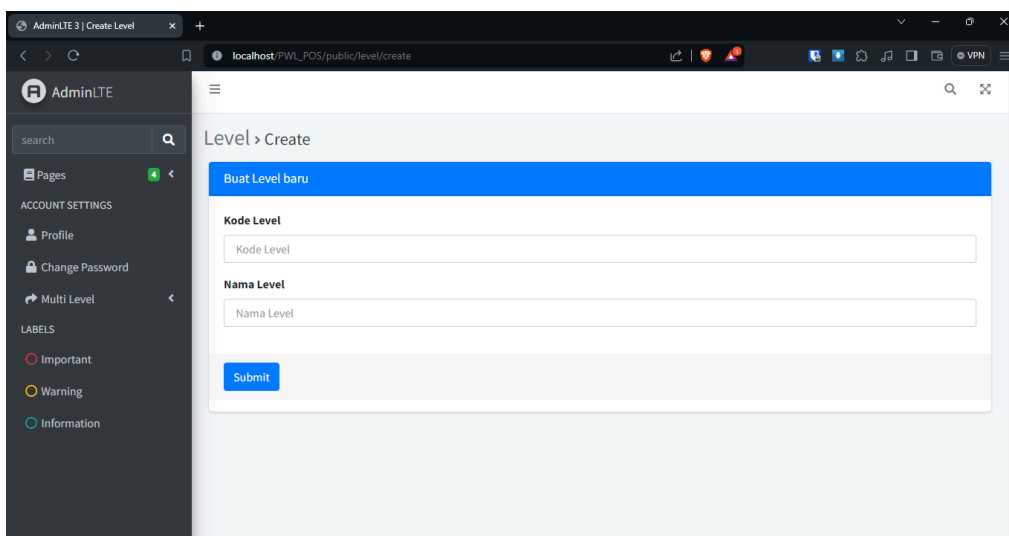
```

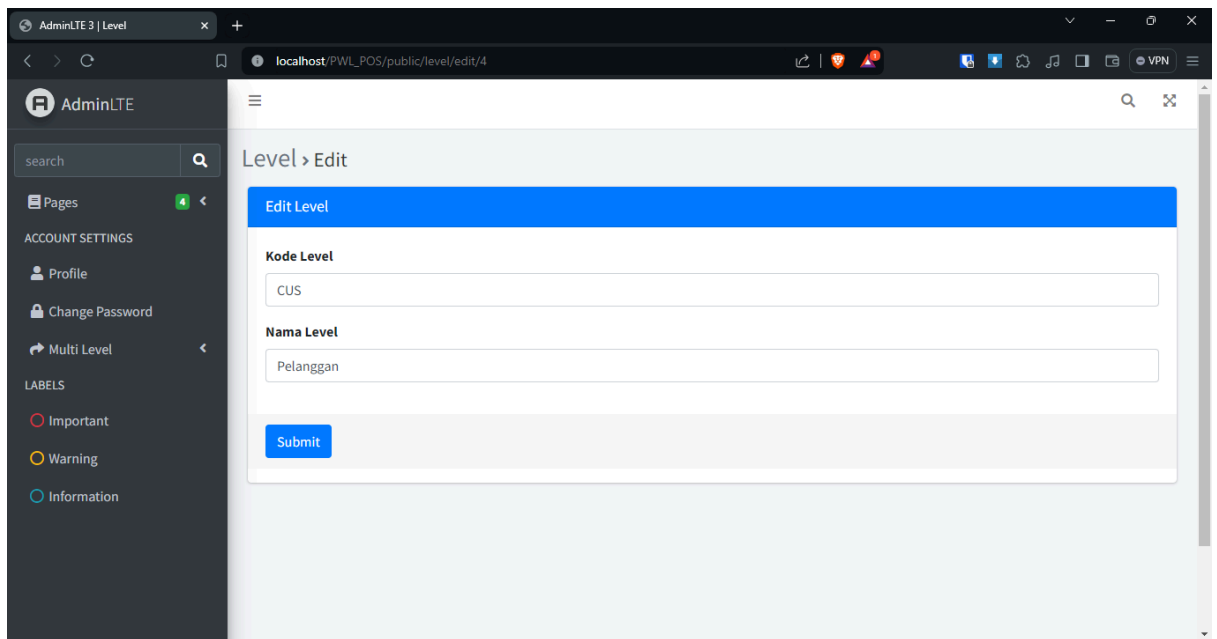
app > Http > Controllers > LevelController.php > ...
37
38     public function update(Request $request, $id)
39     {
40         $level = LevelModel::find($id);
41
42         $level->level_kode = $request->levelKode;
43         $level->level_nama = $request->levelNama;
44         $level->save();
45
46         return redirect('/level')->with('success', 'Level berhasil diupdate');
47     }
48
49     public function delete($id)
50     {
51         $level = LevelModel::find($id);
52         if (!$level) {
53             abort(404);
54         }
55
56         DB::table('m_user')->where('level_id', $id)->update(['level_id' => null]);
57         $level->delete();
58
59         return redirect('/level')->with('success', 'Level berhasil dihapus');
60     }
61 }

```

```
resources > views > level > create.blade.php > ...
1 @extends('layouts.app')
2
3 @section('subtitle', 'Create Level')
4 @section('content_header_title', 'Level')
5 @section('content_header_subtitle', 'Create')
6
7 @section('content')
8     <div class="container">
9         <div class="card card-primary">
10             <div class="card-header">
11                 <h3 class="card-title">Buat Level baru</h3>
12             </div>
13
14             <form method="post" action="..../level">
15                 <div class="card-body">
16                     <div class="form-group">
17                         <label for="levelKode">Kode Level</label>
18                         <input type="text" class="form-control" id="levelKode" name="levelKode" placeholder=" Kode Level">
19                     </div>
20
21                     <div class="form-group">
22                         <label for="levelNama">Nama Level</label>
23                         <input type="text" class="form-control" id="levelNama" name="levelNama"
24                             placeholder=" Nama Level">
25                     </div>
26                 </div>
27                 <div class="card-footer">
28                     <button type="submit" class="btn btn-primary">Submit</button>
29                 </div>
30             </form>
31         </div>
32     </div>
33 @endsection
```

```
resources > views > level > edit.blade.php > div.container > div.card.card-primary > form > div.card-body > div.form-group > input#levelNama.form-control
1 @extends('layouts.app')
2
3 @section('subtitle', 'Level')
4 @section('content_header_title', 'Level')
5 @section('content_header_subtitle', 'Edit')
6
7 @section('content')
8     <div class="container">
9         <div class="card card-primary">
10             <div class="card-header">
11                 <h3 class="card-title">Edit Level</h3>
12             </div>
13
14             <form method="post" action="{{ route('level.update', $level->level_id) }}">
15                 @csrf
16                 @method('PUT')
17
18                 <div class="card-body">
19                     <div class="form-group">
20                         <label for="levelKode">Kode Level</label>
21                         <input type="text" class="form-control" id="levelKode" name="levelKode" value="{{ $level->level_kode }}">
22                     </div>
23
24                     <div class="form-group">
25                         <label for="levelNama">Nama Level</label>
26                         <input type="text" class="form-control" id="levelNama" name="levelNama"
27                             value="{{ $level->level_nama }}">
28                     </div>
29                 </div>
30                 <div class="card-footer">
31                     <button type="submit" class="btn btn-primary">Submit</button>
32                 </div>
33             </form>
34         </div>
35     </div>
```





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 :

```
routes > web.php > ...
42 Route::get('/kategori/create', [KategoriController::class, 'create'])->name('kategori.create');
43 Route::post('/kategori', [KategoriController::class, 'store']);
```

3. Edit pada KategoriKontroller dengan kode:

```
app > Http > Controllers > KategoriController.php > KategoriController > edit
12 class KategoriController extends Controller
13 {
14     public function index(KategoriDataTable $dataTable)
15     {
16         return $dataTable->render('kategori.index');
17     }
18
19     public function create(): View
20     {
21         return view('kategori.create');
22     }
23
24     public function store(Request $request): RedirectResponse
25     {
26         $validated = $request->validate([
27             'kategori_kode' => 'required',
28             'kategori_nama' => 'required',
29         ]);
30
31         return redirect('/kategori');
32     }
}
```

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

```
app > Http > Controllers > KategoriController.php > KategoriController > store
24 public function store(Request $request): RedirectResponse
25 {
26     $validated = $request->validate([
27         'kategori_kode' => 'required|max:10',
28         'kategori_nama' => 'required|max:100',
29     ]);
30
31     return redirect('/kategori');
32 }
```

5. Tulis perbedaan penggunaan validate dengan validateWithBag
validate() → langsung menangani respon validasi dan memberikan pesan error yang sesuai
validateWithBag() → memungkinkan untuk menyimpan pesan error respon validasi yang bisa digunakan di tempat lain.
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?

```
app > Http > Controllers > KategoriController.php > KategoriController
24 public function store(Request $request): RedirectResponse
25 {
26     $validated = $request->validate([
27         'kategori_kode' => 'bail|required|max:10',
28         'kategori_nama' => 'required|max:100',
29     ]);
```

jika kategori_kode gagal validasi, maka validasi untuk fields selanjutnya dihentikan atau tidak dilakukan validasi

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 > ...
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
```

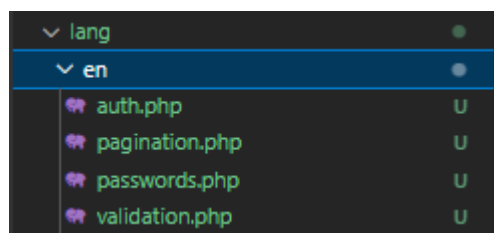
8. Screenshot hasilnya.

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

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

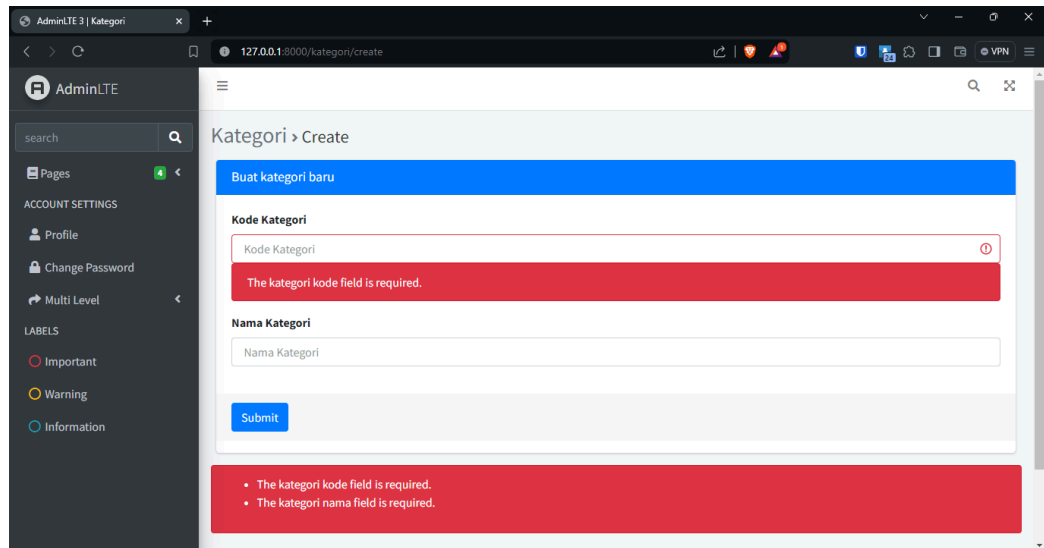
```
PS C:\laragon\www\PWL_POS> php artisan lang:publish
```

```
INFO Language files published successfully.
```



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

```
resources > views > kategori > create.blade.php > div.container > div.card.card-primary > form > div.card-body > div.form-group
17 | <label for="kategori_kode">Kode Kategori</label>
18 | <input type="text" class="@error('kategori_kode') is-invalid @enderror form-control" id="kategori_kode"
19 |     name="kategori_kode" placeholder=" Kode Kategori">
20 |
21 | @error('kategori_kode')
22 |     <div class="alert alert-danger">{{ $message }}</div>
23 | @enderror
24 | </div>
```



C. Form Request Validation

1. Membuat permintaan form dengan menuliskan pada terminal

```
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.
```

2. Ketik kode berikut pada Http/request/StorePostRequest

```
app > Http > Requests > StorePostRequest.php > ...
23 public function rules(): array
24 {
25     return [
26         'kategori_kode' => 'required',
27         'kategori_nama' => 'required',
28     ];
29 }
30
31 public function store(StorePostRequest $request): RedirectResponse
32 {
33     $validated = $request->validated();
34
35     $validated = request()->safe()->only(['kategori_kode', 'kategori_nama']);
36     $validated = $request->safe()->except(['kategori_kode', 'kategori_nama']);
37
38     return redirect('/kategori');
39 }
```

3. Terapkan validasi juga pada tabel m_user dan m_level
User

```
app > Http > Requests > StorePostRequest.php > ...
23 public function rules(): array
24 {
25     return [
26         'kategori_kode' => 'required',
27         'kategori_nama' => 'required',
28         'username' => 'required',
29         'level_id' => 'required',
30         'nama' => 'required',
31         'password' => 'required',
32     ];
33 }
```

```

35     public function store(StorePostRequest $request): RedirectResponse
36     {
37         $validated = $request->validated();
38
39         $validated = request()->safe()->only(['kategori_kode', 'kategori_nama']);
40         $validated = $request->safe()->except(['kategori_kode', 'kategori_nama']);
41         $validated = $request->safe()->only(['username', 'level_id', 'nama', 'password']);
42         $validated = $request->safe()->except(['username', 'level_id', 'nama', 'password']);
43
44         return redirect('/kategori');
45         return redirect('/user');
46     }

```

```

app > Http > Controllers > UserController.php > UserController > update
24     public function store(Request $request) : RedirectResponse
25     {
26         $validated = $request->validate([
27             'nama' => 'bail|required|max:100',
28             'level_id' => 'bail|required',
29             'username' => 'bail|required|max:20',
30             'password' => 'bail|required|max:255',
31         ]);
32
33         UserModel::create([
34             'nama' => $request->nama,
35             'level_id' => $request->levelId,
36             'username' => $request->username,
37             'password' => Hash::make($request->password),
38         ]);
39
40         return redirect('/user');
41     }

```

```

resources > views > user > create.blade.php > div.container
17     <label for="nama">Nama</label>
18     <input type="text" class="@error('nama') is-invalid @enderror form-control" id="nama"
19         name="nama" placeholder="Nama">
20     @error('nama')
21     <div class="alert alert-danger">{{ $message }}</div>
22     @enderror
23 </div>
24
25     <div class="form-group">
26         <label for="level_id">Level Id</label>
27         <select class="@error('level_id') is-invalid @enderror form-control" id="level_id" name="level_id">
28             @foreach ($levels as $level)
29                 <option value="{{ $level->level_id }}">{{ $level->level_nama }}</option>
30             @endforeach
31         </select>
32         @error('level_id')
33         <div class="alert alert-danger">{{ $message }}</div>
34         @enderror
35     </div>
36
37     <div class="form-group">
38         <label for="username">Username</label>
39         <input type="text" class="@error('username') is-invalid @enderror form-control" id="username"
40             name="username" placeholder=" Username">
41         @error('username')
42         <div class="alert alert-danger">{{ $message }}</div>
43         @enderror
44     </div>

```



```

61     @if ($errors->any())
62         <div class="alert alert-danger">
63             <ul>
64                 @foreach ($errors->all() as $error)
65                     <li>{{ $error }}</li>
66                 @endforeach
67             </ul>
68         </div>
69     @endif
70 @endsection

```

AdminLTE

search

PAGES 4

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

User > Create

Buat user baru

Nama

Nama

The nama field is required.

Level Id

Administrator

Username

Username

The username field is required.

Password

Password

The password field is required.

Level

```

app > Http > Requests > StorePostRequest.php > ...
23     public function rules(): array
24     {
25         return [
26             'kategori_kode' => 'required',
27             'kategori_nama' => 'required',
28             'username' => 'required',
29             'level_id' => 'required',
30             'nama' => 'required',
31             'password' => 'required',
32             'level_kode' => 'required',
33             'level_nama' => 'required',
34         ];
35     }
36
37     public function store(StorePostRequest $request): RedirectResponse
38     {
39         $validated = $request->validated();
40
41         $validated = request()->safe()->only(['kategori_kode', 'kategori_nama']);
42         $validated = $request->safe()->except(['kategori_kode', 'kategori_nama']);
43         $validated = $request->safe()->only(['username', 'level_id', 'nama', 'password']);
44         $validated = $request->safe()->except(['username', 'level_id', 'nama', 'password']);
45         $validated = $request->safe()->only(['level_kode', 'level_nama']);
46         $validated = $request->safe()->except(['level_kode', 'level_nama']);
47
48         return redirect('/kategori');
49         return redirect('/user');
50         return redirect('/level');
51     }

```

```

app > Http > Controllers > LevelController.php > LevelController

24 public function store(StorePostRequest $request): RedirectResponse
25 {
26     $validated = $request->validate([
27         'level_kode' => 'bail|required|max:10',
28         'level_nama' => 'bail|required|max:100',
29     ]);
30
31     LevelModel::create([
32         'level_kode' => $request->level_kode,
33         'level_nama' => $request->level_nama,
34     ]);
35
36     return redirect('/level');
37 }

```

```

resources > views > level > create.blade.php > ...

16 <div class="form-group">
17     <label for="level_kode">Kode Level</label>
18     <input type="text" class="@error('level_kode') is-invalid @enderror form-control" id="level_kode"
19         name="level_kode" placeholder=" Kode Level">
20     @error('level_kode')
21     <div class="alert alert-danger">{{ $message }}</div>
22     @enderror
23 </div>
24
25 <div class="form-group">
26     <label for="level_nama">Nama Level</label>
27     <input type="text" class="@error('level_nama') is-invalid @enderror form-control" id="level_nama"
28         name="level_nama" placeholder=" Nama Level">
29     @error('level_nama')
30     <div class="alert alert-danger">{{ $message }}</div>
31     @enderror
32 </div>
33 </div>
34 <div class="card-footer">
35     <button type="submit" class="btn btn-primary">Submit</button>
36 </div>
37 </form>
38 </div>
39 </div>
40 @if ($errors->any())
41 <div class="alert alert-danger">
42     <ul>
43         @foreach ($errors->all() as $error)
44             <li>{{ $error }}</li>
45         @endforeach
46     </ul>
47 </div>
48 @endif

```

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.

```

PS C:\laragon\www\PWL_POS> php artisan make:controller POSController --resource

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

```

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

```

routes > web.php > ...

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

```

3. Atur pada Models/m_user

```
app > Models > UserModel.php > UserModel > $fillable

 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',
23      ];
```

4. Atur pada Migration/m_user_table

```
database > migrations > 2024_03_06_084958_create_m_user_table.php > class > down

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', 255);
20          $table->timestamps();
21
22          $table->foreign('level_id')->references('level_id')->on(
23              'm_level');
24      });
25  }
26
27  /**
28   * Reverse the migrations.
29   */
30  public function down(): void
31  {
32      Schema::dropIfExists('m_user');
33  }
```

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

- index()
- create()
- store()
- show()
- edit()
- update()
- destroy()

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

```
app > Http > Controllers > POSController.php > POSController > destroy
14     public function index()
15     {
16         $user = UserModel::all();
17         return view('m_user.index', compact('user'))->with('i');
18     }
19
20     /**
21      * Show the form for creating a new resource.
22      */
23     public function create()
24     {
25         return view('m_user.create');
26     }
```

```
app > Http > Controllers > POSController.php > POSController > destroy
31     public function store(Request $request)
32     {
33         $request->validate([
34             'user_id' => 'max:20',
35             'username' => 'required',
36             'nama' => 'required',
37         ]);
38
39         UserModel::create($request->all());
40         return redirect()->route('m_user.index')->with('success', 'User berhasil ditambahkan');
41     }
42
43     /**
44      * Display the specified resource.
45      */
46     public function show(string $id, UserModel $user)
47     {
48         $user = UserModel::findOrFail($id);
49         return view('m_user.show', compact('user'));
50     }
```

```
app > Http > Controllers > POSController.php > POSController > destroy
54     /
55     public function edit(string $id)
56     {
57         $user = UserModel::find($id);
58         return view('m_user.edit', compact('user'));
59     }
```

```
app > Http > Controllers > POSController.php > POSController > destroy
64     public function update(Request $request, string $id)
65     {
66         $request->validate([
67             'username' => 'required',
68             'nama' => 'required',
69             'password' => 'required',
70         ]);
71
72         UserModel::find($id)->update($request->all());
73         return redirect()->route('m_user.index')->with('success', 'User berhasil diupdate');
74     }
```

```

app > Http > Controllers > POSController.php > POSController > destroy
79     public function destroy(string $id)
80     {
81         $user = UserModel::findOrFail($id)->delete();
82         return redirect()->route('m_user.index')->with('success', 'User berhasil dihapus');
83     }

```

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

1) template.blade.php

```

resources > views > m_user > template.blade.php > html > head > link
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
8  <body>
9      <div class="container">
10         @yield('content')
11     </div>
12 </body>
13
14 </html>

```

2) index.blade.php

```

resources > views > m_user > show.blade.php > div.row.mt-5.mb-5 > div.col-lg-12.margin-tb > div.float-right
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
11                 </a>
12             </div>
13         </div>
14     </div>
15     <div class="row">
16         <div class="col-xs-12 col-sm-12 col-md-12">
17             <div class="form-group">
18                 <strong>User_id:</strong>
19                 {{ $user->user_id }}
20             </div>
21         </div>
22         <div class="col-xs-12 col-sm-12 col-md-12">
23             <div class="form-group">
24                 <strong>Level_id:</strong>
25                 {{ $user->level_id }}
26             </div>
27         </div>

```

```
resources > views > m_user > show.blade.php > div.row.mt-5.mb-5 > div.col-lg-12.margin-tb > div.float-right
28     <div class="col-xs-12 col-sm-12 col-md-12">
29         <div class="form-group">
30             <strong>Username:</strong>
31             {{ $user->username }}
32         </div>
33     </div>
34     <div class="col-xs-12 col-sm-12 col-md-12">
35         <div class="form-group">
36             <strong>Nama:</strong>
37             {{ $user->nama }}
38         </div>
39     </div>
40     <div class="col-xs-12 col-sm-12 col-md-12">
41         <div class="form-group">
42             <strong>Password:</strong>
43             {{ $user->password }}
44         </div>
45     </div>
46 </div>
47 @endsection
```

3) create.blade.php

```
resources > views > m_user > create.blade.php > div.alert.alert-danger > br
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
11                  </a>
12              </div>
13          </div>
14      </div>
15      @if ($errors->any())
16          <div class="alert alert-danger">
17              <strong>Ops</strong>
18              Input gagal
19              <br><br>
20              <ul>
21                  @foreach ($errors->all() as $error)
22                      <li>{{ $error }}</li>
23                  @endforeach
24              </ul>
25          </div>
26      @endif
27      <form action="{{ route('m_user.store') }}" method="POST">
28          @csrf
29          <div class="col-xs-12 col-sm-12 col-md-12">
30              <div class="form-group">
31                  <strong>Username:</strong>
32                  <input type="text" name="username" class="form-control" placeholder="Masukkan username"></input>
33              </div>
34          </div>
```

```

resources > views > m_user > create.blade.php > div.alert.alert-danger > br
35         <div class="col-xs-12 col-sm-12 col-md-12">
36             <div class="form-group">
37                 <strong>nama:</strong>
38                 <input type="text" name="nama" class="form-control" placeholder="Masukkan nama"></input>
39             </div>
40         </div>
41         <div class="col-xs-12 col-sm-12 col-md-12">
42             <div class="form-group">
43                 <strong>Password:</strong>
44                 <input type="password" name="password" class="form-control" placeholder="Masukkan password"></input>
45             </div>
46         </div>
47         <div class="col-xs-12 col-sm-12 col-md-12 text-center">
48             <button type="submit" class="btn btn-primary">Submit</button>
49         </div>
50     </div>
51 </form>
52 @endsection

```

4) show.blade.php

```

resources > views > m_user > show.blade.php > div.row.mt-5.mb-5 > div.col-lg-12.margin-tb > div.float-right
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
11                 </a>
12             </div>
13         </div>
14     </div>
15     <div class="row">
16         <div class="col-xs-12 col-sm-12 col-md-12">
17             <div class="form-group">
18                 <strong>User_id:</strong>
19                 {{ $user->user_id }}
20             </div>
21         </div>
22         <div class="col-xs-12 col-sm-12 col-md-12">
23             <div class="form-group">
24                 <strong>Level_id:</strong>
25                 {{ $user->level_id }}
26             </div>
27         </div>
28         <div class="col-xs-12 col-sm-12 col-md-12">
29             <div class="form-group">
30                 <strong>Username:</strong>
31                 {{ $user->username }}
32             </div>
33         </div>

```

```

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

```

5) edit.blade.php

```

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                 <a class="btn btn-secondary" href="{{ route('m_user.index') }}">
10                     Kembali
11                 </a>
12             </div>
13         </div>
14     </div>
15     @if ($errors->any())
16         <div class="alert alert-danger">
17             <strong>Ops!</strong>
18             Error
19             <br><br>
20             <ul>
21                 @foreach ($errors->all() as $error)
22                     <li>{{ $error }}</li>
23                 @endforeach
24             </ul>
25         </div>
26     @endif

```

```

resources > views > m_user > edit.blade.php > ...
27 <form action="{{ route('m_user.update', $user->user_id) }}" method="POST">
28     @csrf
29     @method('PUT')
30     <div class="row">
31         <div class="col-xs-12 col-sm-12 col-md-12">
32             <div class="form-group">
33                 <strong>User_id:</strong>
34                 <input type="text" name="userd" value="{{ $user->user_id }}" class="form-control"
35                     placeholder="Masukkan user id">
36             </div>
37         </div>
38         <div class="col-xs-12 col-sm-12 col-md-12">
39             <div class="form-group">
40                 <strong>Level_id:</strong>
41                 <input type="text" name="levelid" value="{{ $user->level_id }}" class="form-control"
42                     placeholder="Masukkan level">
43             </div>
44         </div>
45         <div class="col-xs-12 col-sm-12 col-md-12">
46             <div class="form-group">
47                 <strong>Username:</strong>
48                 <input type="text" value="{{ $user->username }}" class="formcontrol" name="username"
49                     placeholder="Masukkan Nomor username">
50             </div>
51         </div>

```



```
resources > views > m_user > edit.blade.php > ...
52         <div class="col-xs-12 col-sm-12 col-md-12">
53             <div class="form-group">
54                 <strong>nama:</strong>
55                 <input type="text" value= "{{ $user->nama }}" name="nama" class="form-control"
56                     placeholder="Masukkan nama"></input>
57             </div>
58         </div>
59         <div class="col-xs-12 col-sm-12 col-md-12">
60             <div class="form-group">
61                 <strong>Password:</strong>
62                 <input type="password" value= "{{ $user->password }}" name="password" class="form-control"
63                     placeholder="Masukkan password"></input>
64             </div>
65         </div>
66         <div class="col-xs-12 col-sm-12 col-md-12 text-center">
67             <button type="submit" class="btn btn-primary">Update</button>
68         </div>
69     </div>
70 </form>
71 @endsection
```

Silahkan akses localhost/PWL_POS/public/m_user

localhost/PWL_POS/public/m_user

CRUD user Input User

User id	Level id	username	nama	password	
1	1	admin	Administrator	\$2y\$12\$6G0ZleXmDBBddPP35GNZpe2RjLxss3iwtUY3DSA/M/Ku87qY9hf6	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$qYFZLkc3OV8mDMOen63p0.tRs0uzlQN1KgZB6C.NsXtOlhc0dD6.	Show Edit Delete
3	3	staff	Staff/Kasir	\$2y\$12\$87W6e6ZhT1WaulXGzRSjuEnd8WG44OCXo09evRGkd3NRIuNOz3N.	Show Edit Delete
6	4	customer-1	Pelanggan Pertama	\$2y\$12\$NnusYn8g42kg98jHH04C3e4SGAvDCPCzQ8Z5tL4yJ0E3gHKlaKpgK	Show Edit Delete
9	2	manager_dua	Manager 2	\$2y\$12\$88GZULsKt7XtDSOip6Y/IOr0SvFYhNKbuEexjoypJ3c/CcelQwx.	Show Edit Delete
10	2	manager22	Manager Dua Dua	\$2y\$12\$t4Fzde0B45Gfcg2EuGMNYOmNbq8/0F.xlxK8MMXSiG8DEhK59Bzh.	Show Edit Delete
11	2	manager33	Manager Tiga Tiga	\$2y\$12\$iklJ4CgcGogH9j/jBESK0mWFLemISVrhLAO/PDre/srmQsF/QTKS	Show Edit Delete

Tugas

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

```
app > Http > Controllers > POSController.php > POSController
31 public function store(Request $request)
32 {
33     $request->validate([
34         'user_id' => 'max:20',
35         'username' => 'required',
36         'nama' => 'required',
37         'level_id' => 'required',
38     ]);
39
40     UserModel::create($request->all());
41     return redirect()->route('m_user.index')->with('success', 'User berhasil ditambahkan');
42 }
```

```
resources > views > m_user > create.blade.php > form > div.col-xs-12.col-sm-12.col-md-12 > div.form-group > input.form-control
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" class="form-control" placeholder="Masukkan Level Id"></input>
39     </div>
40 </div>
```

localhost/PWL_POS/public/m_user/create

Membuat Daftar User

[Kembali](#)

Username:

Level ID:

nama:

Password:

[Submit](#)

2. Modifikasi dengan tema/template kesukaan anda

```
resources > views > m_user > index.blade.php > table.table-bordered > tr > th
1  {{-- @extends('m_user/template') --}}
2  @extends('layouts.app')
3
4  @section('subtitle', 'User')
5  @section('content_header_title', 'Home')
6  @section('content_header_subtitle', 'User')
7
8  @section('content')
9  <div class="row mt-1 mb-1">
10 <div class="col-lg-12 margin-tb">
11 <div class="float-left">
12 <h2>CRUD user</h2>
13 </div>
14 <div class="float-right">
15 <a class="btn btn-success" href="{{ route('m_user.create') }}">Input User</a>
16 </div>
17 </div>
18 </div>
```

localhost/PWL_POS/public/m_user

AdminLTE

search

Pages 4

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

Home > User

[Input User](#)

User id	Level id	username	nama	password	Actions
1	1	admin	Administrator	\$2y\$12\$6G0ZleXmDBBddPP35GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$qYFZLkc3OVBMDOen63p0.tRs0uzlQN1KgZB6C.NsXtOlhcX0dD6.	Show Edit Delete
3	3	staff	Staff/Kasir	\$2y\$12\$87W6eZhT1WaulXGzRSojuEnd8WG44OCXo9evRGkD3NRiuNO23N.	Show Edit Delete
6	4	customer-1	Pelanggan Pertama	\$2y\$12\$NnusYn8g42kg98jHH04C3e4SGAvDCPCzQ8Z5tL4yJ0E3ghKlaKpgK	Show Edit Delete
9	2	manager_dua	Manager 2	\$2y\$12\$B8GZULsKt7XtDSOip6Y/IOrSvfYyhNKbuEexjoypJ3c/CcelQwx.	Show Edit Delete
10	2	manager22	Manager Dua	\$2y\$12\$t4Fzde0B45Gfcg2EuGMNYOmNbq8/0F.xlxK8MMXSiG8DEhK59Bzh.	Show Edit Delete

AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

localhost/PWL_POS/public/m_user/1

User > Detail

Show User

Kembali

User_id: 1

Level_id: 1

Username: admin

Nama: Administrator

Password: \$2y\$12\$6G0ZleXmDBBddPP35GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6

AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

localhost/PWL_POS/public/m_user/create

User > Create

Membuat Daftar User

Kembali

Username:
Masukkan username

Level ID:
Masukkan Level Id

nama:
Masukkan nama

Password:
Masukkan password

Submit

AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

localhost/PWL_POS/public/m_user/1/edit

User > Edit

Edit User

Kembali

User_id:
1

Level_id:
1

Username:
admin

nama:
Administrator

Password:
.....

Update

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

a. \$request→validate

digunakan untuk melakukan validasi terhadap data yang diinputkan pengguna sebelum data tersebut diproses lebih lanjut (update/add) ke dalam database. Validasi dilakukan berdasarkan rules yang telah ditentukan di dalam controller atau request class. Jika validasi gagal, maka secara otomatis pengguna akan diarahkan kembali ke halaman sebelumnya dengan data yang telah diinputkan serta pesan error yang sesuai.

b. \$error

digunakan untuk menyimpan pesan-pesan error hasil dari proses validasi jika terdapat kesalahan dalam input data, kemudian dapat diakses melalui tampilan blade

c. alert

elemen HTML digunakan untuk memberikan tampilan pesan error pada user, setelah proses validasi.