

PEMROGRAMAN WEB LANJUT

PERTEMUAN 6



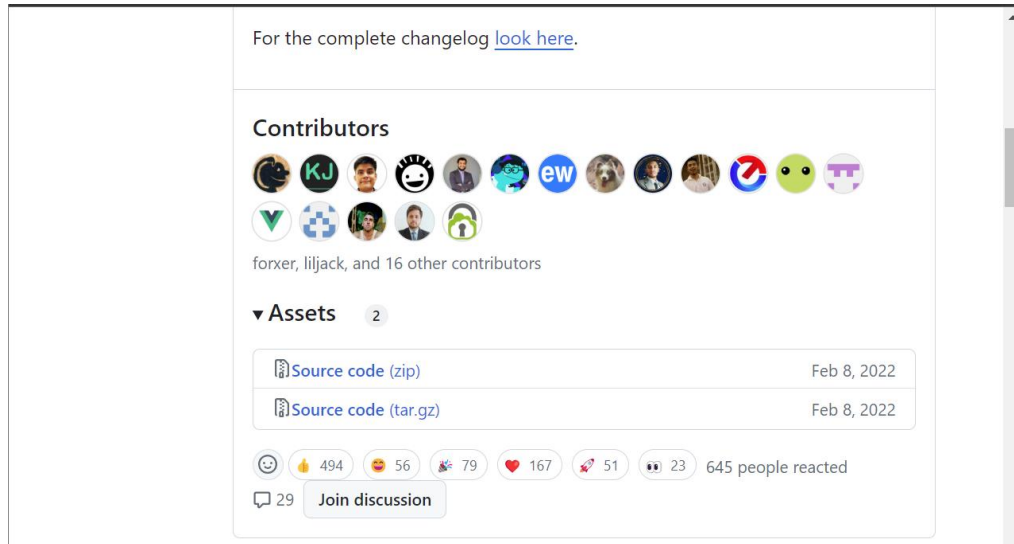
MUHAMMAD IQBAL MAKMUR AL-MUNIRI
2241720099 / 15 / TI-2C
TEKNIK INFORMATIKA
TEKNOLOGI INFORMASI

Link Github: <https://github.com/GalaAldebara/PemrogramanWebLanjut.git>

A. Template Form (AdminLTE)

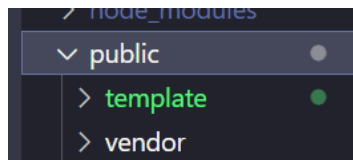
1. Penggunaan template AdminLTE Bootstrap Admin Dashboard Template dengan langkah pertama akses <https://adminlte.io/>, lalu klik download pada source code (zip)

Jawab:



2. Lakukan extracr 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.

Jawab:



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.

Jawab:

```
@section('content_body')
<div class="wrapper">
  <!-- Google Font: Source Sans Pro -->
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
  <!-- Font Awesome Icons -->
  <link rel="stylesheet" href="{{ asset('template/plugins/fontawesome-free/css/all.min.css') }}">
  <!-- overlayScrollbars -->
  <link rel="stylesheet" href="{{ asset('template/plugins/overlayScrollbars/css/OverlayScrollbars.min.css') }}">
  <!-- Theme style -->
  <link rel="stylesheet" href="{{ asset('template/dist/css/adminlte.min.css') }}">

  <!-- Preloader -->
  <div class="preloader flex-column justify-content-center align-items-center">
    
  </div>
</div>
```

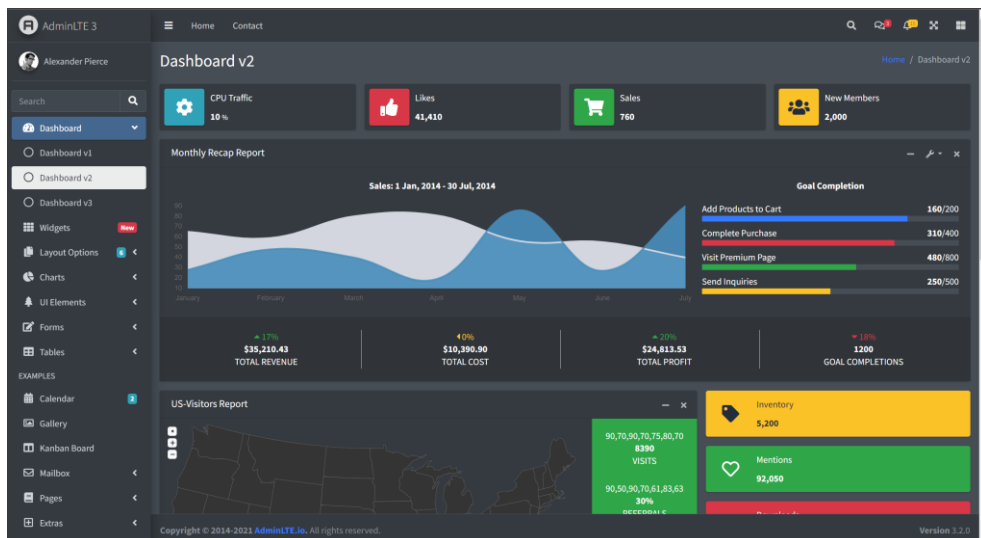
4. Scroll ke bagian paling bawah dan sesuaikan dengan kode berikut

Jawab:

```
1711 <!-- REQUIRED SCRIPTS -->
1712 <!-- jQuery -->
1713 <script src="{{ asset('template/plugins/jquery/jquery.min.js') }}"></script>
1714 <!-- Bootstrap -->
1715 <script src="{{ asset('template/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
1716 <!-- overlayScrollbars -->
1717 <script src="{{ asset('template/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js') }}"></script>
1718 <!-- AdminLTE App -->
1719 <script src="{{ asset('template/dist/js/adminlte.js') }}"></script>
1720
1721 <!-- PAGE PLUGINS -->
1722 <!-- jQuery Mapael -->
1723 <script src="{{ asset('template/plugins/jquery-mousewheel/jquery.mousewheel.js') }}"></script>
1724 <script src="{{ asset('template/plugins/raphael/raphael.min.js') }}"></script>
1725 <script src="{{ asset('template/plugins/jquery-mapael/jquery.mapael.min.js') }}"></script>
1726 <script src="{{ asset('template/plugins/jquery-mapael/maps/usa_states.min.js') }}"></script>
1727 <!-- ChartJS -->
1728 <script src="{{ asset('template/plugins/chart.js/Chart.min.js') }}"></script>
1729
1730 <!-- AdminLTE for demo purposes -->
1731 <script src="{{ asset('template/dist/js/demo.js') }}"></script>
1732 <!-- AdminLTE dashboard demo (This is only for demo purposes) -->
1733 <script src="{{ asset('template/dist/js/pages/dashboard2.js') }}"></script>
1734 </body>
1735 </html>
```

5. Jalankan pada browser

Jawab:



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

Jawab:

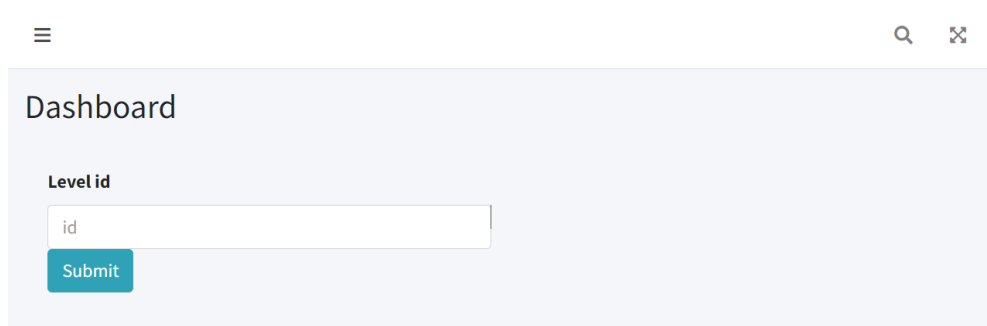
7. Buka file pada form> general cari keyword dengan 'general elements' dan coba modifikasi pada welcome.blade.php

Jawab:

```
1159 <!-- general form elements disabled -->
1160 <div class="card card-warning">
1161   <div class="card-header">
1162     <h3 class="card-title">General Elements</h3>
1163   </div>
```

8. Apa yang tampil?

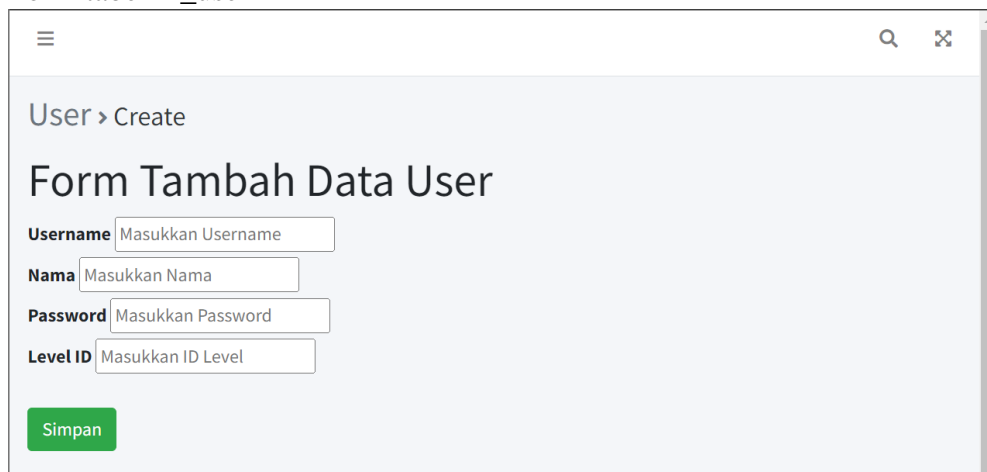
Jawab:



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

Jawab:

Form tabel m_user



Kode

```
@extends('layouts.app')
{{-- Customize layout sections --}}
@section('subtitle', 'Kategori')
@section('content_header_title', 'User')
@section('content_header_subtitle', 'Create')
{{-- Content body: main page content --}}
@section('content')
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Tambah Data User</title>
```

```

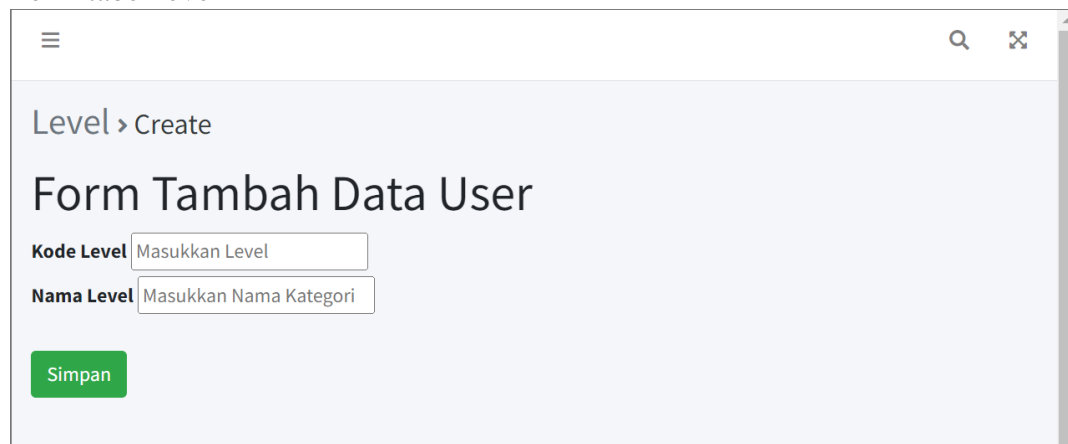
</head>
<body>
  <h1>Form Tambah Data User</h1>
  <form method="post" action="/user/tambah_simpan">

    {{-- {{ csrf_field() }} --}}

    <label>Username</label>
    <input type="text" name="username" placeholder="Masukkan
Username">
    <br>
    <label>Nama</label>
    <input type="text" name="nama" placeholder="Masukkan
Nama">
    <br>
    <label>Password</label>
    <input type="password" name="password"
placeholder="Masukkan Password">
    <br>
    <label>Level ID</label>
    <input type="number" name="level_id"
placeholder="Masukkan ID Level">
    <br><br>
    <input type="submit" class="btn btn-success"
value="Simpan">
  </form>
</body>
</html>
@endsection

```

Form tabel level



Level > Create

Form Tambah Data User

Kode Level

Nama Level

```

@extends('layouts.app')
{{-- Customize layout sections --}}
@section('subtitle', 'Kategori')
@section('content_header_title', 'Level')
@section('content_header_subtitle', 'Create')

```

```

{{-- Content body: main page content --}}
@section('content')
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <title>Tambah Data User</title>
    </head>
    <body>
        <h1>Form Tambah Data User</h1>
        <form action="../level" method="post">
            <label>Kode Level</label>
            <input type="text" name="level_kode" id="kodeKategori"
placeholder="Masukkan Level">
            <br>
            <label>Nama Level</label>
            <input type="text" name="level_nama" id="namaKategori"
placeholder="Masukkan Nama Kategori">
            <br><br>
            <input type="submit" class="btn btn-success"
value="Simpan">
        </form>
    </body>
</html>
@endsection

```

B. Validasi Pada Server

1. Buka Kembali file tugas di jobsheet 5, klik dan baca documentataion ini
Validation – Laravel 10.x – The PHP Framework For Web Artisans
Jawab:

Validation Quickstart

To learn about Laravel's powerful validation features, let's look at a complete example of validating a form and displaying the error messages back to the user. By reading this high-level overview, you'll be able to gain a good general understanding of how to validate incoming request data using Laravel:

Defining the Routes

First, let's assume we have the following routes defined in our `routes/web.php` file:

```
use App\Http\Controllers\PostController;

Route::get('/post/create', [PostController::class, 'create']);
Route::post('/post', [PostController::class, 'store']);
```

2. Pastikan sudah terdefiniskan route pada web.php

Jawab:

```
52 Route::get('/kategori/create', [KategoriController::class, 'create']);
53 Route::post('/kategori', [KategoriController::class, 'store']);
```

3. Edit pada KategoriController dengan kode:

Jawab:

```
57 public function create(): View
58 {
59     return view('kategori.create');
60 }
61
62 public function store(Request $request): RedirectResponse
63 {
64     $validated = $request->validate([
65         'kategori_kode' => 'required',
66         'kategori_nama' => 'required',
67     ]);
68     return redirect('/kategori');
69 }
70 }
```

4. Berikut ini sebagai alternatif contoh penulisan validasi dan bisa Anda sesuaikan kata-kata dengan apa yang sedang Anda kerjakan.
5. Tulis perbedaan penggunaan `validate` dengan `validateWithBag`!

Jawab:

Metode `validate()` digunakan untuk melakukan validasi langsung dalam controller dan mengembalikan respons bawaan Laravel jika validasi gagal.

Metode `validateWithBag()` juga digunakan untuk melakukan validasi data, tetapi memberikan fleksibilitas lebih dalam menangani respons jika validasi gagal.

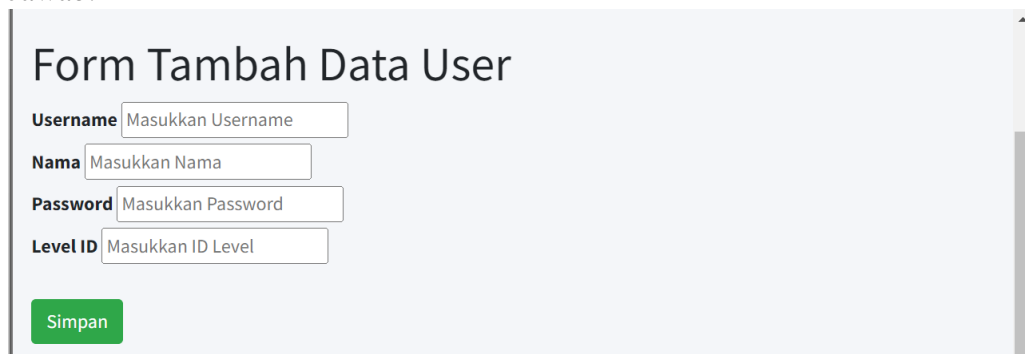
6. Terkadang Anda mungkin ingin berhenti menjalankan aturan validasi pada suatu atribut setelah kegagalan validasi pertama. Untuk melakukannya, tetapkan `bail` aturan ke atribut: coba sesuaikan dengan field pada `m_kategori`, yuk bisa.. apa yang terjadi?
- 7.
8. Pada `view/create.blade.php` tambahkan code berikut agar ketika validasi gagal, kita dapat menampilkan pesan dalam kesalahan dalam tampilan

Jawab:

```
16     @if ($errors->any())
17     <div class="alert alert-danger">
18         <ul>
19             @foreach ($errors->all() as $error)
20                 <li>{{ $error }}</li>
21             @endforeach
22         </ul>
23     </div>
```

9. Screenshot hasilnya

Jawab:



10. 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.

11. Pada view/create.blade.php tambahkan dan coba berikut:

Jawab:

```
37     <label for="kategori_kode">Kode Kategori</label>
38     <input id="kategori_kode" type="text" name="kategori_kode" class="@error('kategori_kode')
39     @error('kategori_kode')
40     <div class="alert alert-danger">{{ $message }}</div>
41     @enderror
```

C. Form Request Validation

1. Membuat permintaan form dengan menuliskan pada terminal

Jawab:

```
PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos> php artisan make:request StorePostRequest

INFO Request [D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos\app\Http\Requests\StorePostRequest.php] created successfully.

PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos>
```

2. Ketik kode berikut pada Http/request/StorePostRequest

Jawab:

```
21  * @return array<string, \Illuminate\Contracts\Validation\ValidationRule|array<mixed>>
22  */
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      $validate = $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  }
40 }
```

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.

Jawab:

```
PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos> php artisan make:controller POSController --resource

INFO Controller [D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos\app\Http\Controllers\POSController.php] created successfully.

PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos>
```

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

Jawab:

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

3. Atur pada Models/m_user

Jawab:

```
m_user.php U X 2024_03_29_093417_create_m_user_table.php U KategoriController.php M create.blade.php ...level U create.blade.php ...user U
app > Models > m_user.php > ...
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     use HasFactory;
11
12     protected $table = "m_user";
13     public $timestamps = false;
14     protected $primaryKey = 'user_id';
15
16     protected $fillable = [
17         'user_id',
18         'level_id',
19         'username',
20         'nama',
21         'password',
22     ];
23
24 }
```

4. Atur pada Migration/m_user_table

Jawab:

```
m_user.php U 2024_03_29_093417_create_m_user_table.php U KategoriController.php M create.blade.php ...level U create.blade.php ...user U StorePostRequest.php U
Database > migrations > 2024_03_29_093417_create_m_user_table.php > ...
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('user1', function (Blueprint $table) {
15             $table->id('user_id');
16             $table->unsignedInteger('level_id')->index;
17             $table->string('username', 20)->unique;
18             $table->string('nama', 100);
19             $table->string('password');
20             $table->timestamp();
21
22             $table->foreign('level_id')->references('level_id')->on('m_level');
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('user1');
32     }
33
34 }
```

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

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

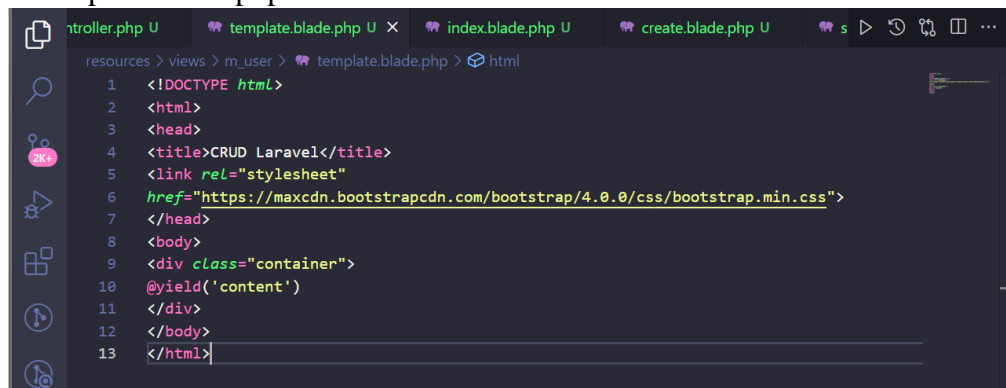
Jawab:

```
web.php M POSController.php U X
app > Http > Controllers > POSController.php > PHP Intellisense > POSController > destroy
1 use App\Models\User;
2 use Illuminate\Http\Request;
3
4 class POSController extends Controller
5 {
6     /**
7      * Display a listing of the resource.
8      */
9     public function index()
10     {
11         $user1 = m_user::all();
12         return view('m_user.index', compact('user1'))->with('1');
13     }
14
15     /**
16      * Show the form for creating a new resource.
17      */
18     public function create()
19     {
20         return view('m_user.create');
21     }
22
23     /**
24      * Store a newly created resource in storage.
25      */
26     public function store(Request $request)
27     {
28         $request->validate([
29             'user_id' => 'max:20',
30             'username' => 'required',
31             'nama' => 'required',
32         ]);
33
34         m_user::create($request->all());
35
36         return redirect()->route('m_user.index')
37             ->with('success', 'user Berhasil Ditambahkan');
38     }
39
40     /**
41      * Display the specified resource.
42      */
43     public function show(string $id)
44     {
45         $user1 = m_user::findOrFail($id);
46         return view('m_user.show', compact('user1'));
47     }
48 }
```

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

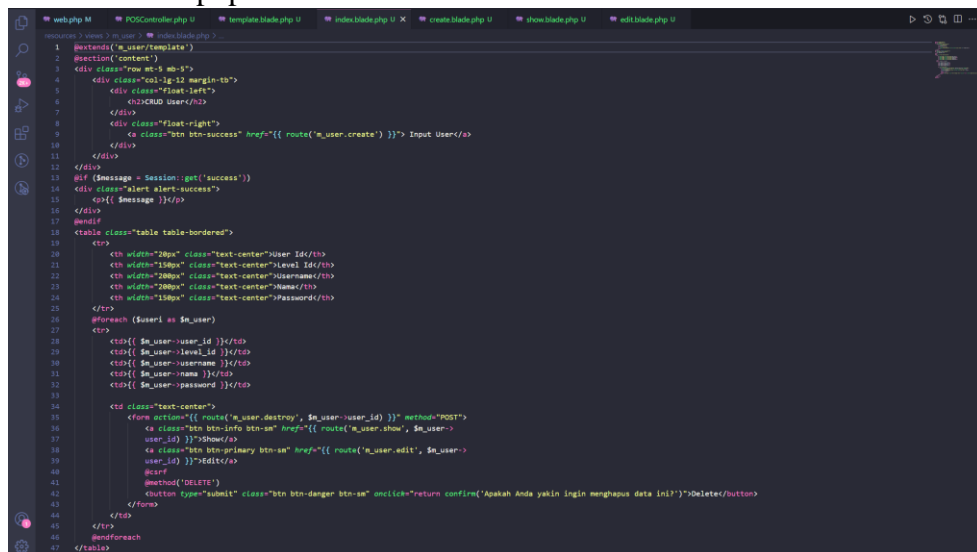
Jawab

1. template.blade.php



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

2. index.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-top">
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 User</a>
10 </div>
11 </div>
12 </div>
13 @if ($message = Session::get('success'))
14 <div class="alert alert-success">
15 <div class="alert">{{ $message }}</div>
16 </div>
17 @endif
18 <table class="table table-bordered">
19 <tr>
20 <th width="20%" class="text-center">User Id</th>
21 <th width="10%" class="text-center">Level Id</th>
22 <th width="20%" class="text-center">User Name</th>
23 <th width="20%" class="text-center">Name</th>
24 <th width="10%" class="text-center">Password</th>
25 </tr>
26 @foreach ($user as $user)
27 <tr>
28 <td{{ $user->user_id }}</td>
29 <td{{ $user->level_id }}</td>
30 <td{{ $user->username }}</td>
31 <td{{ $user->name }}</td>
32 <td{{ $user->password }}</td>
33 <td class="text-center">
34 <form action="{{ route('m_user.destroy', $user->user_id) }}" method="POST">
35 <a class="btn btn-info btn-sm" href="{{ route('m_user.show', $user->
36 user_id) }}">Show</a>
37 <a class="btn btn-primary btn-sm" href="{{ route('m_user.edit', $user->
38 user_id) }}">Edit</a>
39 <a href="#">Delete</a>
40 </td>
41 </tr>
42 </foreach>
43 </table>
44 @endforeach
45 </div>
```

3. 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') }}">
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         <div class="col-xs-12 col-sm-12 col-md-12">
27             <div class="form-group">
28                 <strong>Username</strong>
29                 <input type="text" name="username" class="form-control"
30                     placeholder="Masukkan username"></input>
31             </div>
32         </div>
33         <div class="col-xs-12 col-sm-12 col-md-12">
34             <div class="form-group">
35                 <strong>nama</strong>
36                 <input type="text" name="nama" class="form-control"
37                     placeholder="Masukkan nama"></input>
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                 <input type="password" name="password" class="form-control"
44                     placeholder="Masukkan password"></input>
45             </div>
46         </div>
47         <div class="col-xs-12 col-sm-12 col-md-12 text-center">

```

4. 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">
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                 <input type="text" value="{{ $user->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                 <input type="text" value="{{ $user->level_id }}">
25             </div>
26         </div>
27         <div class="col-xs-12 col-sm-12 col-md-12">
28             <div class="form-group">
29                 <strong>username</strong>
30                 <input type="text" value="{{ $user->username }}">
31             </div>
32         </div>
33         <div class="col-xs-12 col-sm-12 col-md-12">
34             <div class="form-group">
35                 <strong>nama</strong>
36                 <input type="text" value="{{ $user->nama }}">
37             </div>
38         </div>
39         <div class="col-xs-12 col-sm-12 col-md-12">
40             <div class="form-group">
41                 <strong>password</strong>
42                 <input type="password" value="{{ $user->password }}">
43             </div>
44         </div>
45     </div>
46 @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</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) }}"
25         method="POST">
26         @csrf
27         <method 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="user_id" value="{{ $user->user_id }}"
32                         class="form-control" placeholder="Masukkan user id">
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 }}"
39                         class="form-control" 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 Nomor username">
47                 </div>
48             </div>
49         </method>
50     </form>

```

8. Silahkan akses localhost/m_user

Jawab:

127.0.0.1:8000/mo_user

CRUD User Input User

User Id	Level Id	Username	Nama	Password	
1	1	administrator	Administrator	\$2y\$12\$jnqngAux3dFN414nbHBO11cjLOqzNcFVfre7gnUlcZm/qValNq	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$2vT8eLBqG3N.vh4Rgy0.g9/fmb44UfzkdQhXjZv4Un2RY2EkG	Show Edit Delete
3	3	staff	Staff/Kasir	\$2y\$12\$ids.iqWUZShiiQ8Vluy0.XFL63jT9OLPOvN15oD9VZgdmn/B.9zi	Show Edit Delete

TUGAS

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

Jawab:

127.0.0.1:8000/mo_user

CRUD User Input User

User Id	Level Id	Username	Nama	Password	
1	1	administrator	Administrator	\$2y\$12\$jnqngAux3dFN414nbHBO11cjLOqzNcFVfre7gnUlcZm/qValNq	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$2vT8eLBqG3N.vh4Rgy0.g9/fmb44UfzkdQhXjZv4Un2RY2EkG	Show Edit Delete
3	3	staff	Staff/Kasir	\$2y\$12\$ids.iqWUZShiiQ8Vluy0.XFL63jT9OLPOvN15oD9VZgdmn/B.9zi	Show Edit Delete

2. Modifikasi dengan tema/template kesukaan Anda

Jawab:

CRUD User Input User

User Id	Level Id	Username	Nama	Password	Actions
1	1	administrator	Administrator	\$2y\$12\$jnqngAux3dFN414nbHBO11cjLOqzNcFVfre7gnUlcZm/qValNq	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$2vT8eLBqG3N.vh4Rgy0.g9/fmb44UfzkdQhXjZv4Un2RY2EkG	Show Edit Delete
3	3	staff	Staff/Kasir	\$2y\$12\$ids.iqWUZShiiQ8Vluy0.XFL63jT9OLPOvN15oD9VZgdmn/B.9zi	Show Edit Delete

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

Jawab:

digunakan untuk melakukan validasi data yang dikirimkan melalui formulir sebelum data tersebut disimpan atau diupdate. Ini membantu memastikan bahwa data yang dimasukkan pengguna sesuai dengan aturan yang ditetapkan sebelumnya.

Ketika validasi gagal, Laravel akan secara otomatis mengarahkan kembali pengguna ke halaman sebelumnya dan menampilkan pesan kesalahan.