Nama	:	AL BYAN AGUNG SHAFIQRI
Nim	:	2241720080
Kelas	:	TI 2H

LEMBAR JAWABAN JOBSHEET-6

Soal No	Jawaban
1	Mengekstrak Admin LTE v3 pada POS_PWL/public dan merename nya menjadi template
	menyalin isi index2.html pada template dan memindahkanya ke welcome.blade.php lalu merubah href dan src

```
1713  <!-- REQUIRED SCRIPTS -->
1714  <!-- jQuery -->
1715  <script src="{{ asset('template/plugins/jquery/jquery.min.js')}}"></script>
1716  <!-- Bootstrap -->
1717  <script src="{{ asset('template/plugins/bootstrap/js/bootstrap.bundle.min.js')}}"></script>
1718  <!-- overlayScrollbars -->
1719  <script src="{{ asset('template/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js')}}"></script>
1720  <!-- AdminLTE App -->
1721  <script src="{{ asset('template/dist/js/adminlte.js')}}"></script>
1722
1723  <!-- PAGE PLUGINS -->
1724  <!-- jQuery Mapael -->
1725  <script src="{{ asset('template/plugins/jquery-mousewheel/jquery.mousewheel.js')}}"></script>
1726  <script src="{{ asset('template/plugins/raphael/raphael.min.js')}}"></script>
1727  <script src="{{ asset('template/plugins/jquery-mapael/jquery.mapael.min.js')}}"></script>
1728  <script src="{{ asset('template/plugins/jquery-mapael/maps/usa_states.min.js')}}"></script>
1729  <!-- ChartJS -->
1730  <script src="{{ asset('template/plugins/chart.js/Chart.min.js')}}"></script>
1731
1732  <!-- AdminLTE for demo purposes -->
1733  <script src="{{ asset('template/dist/js/demo.js')}}"></script>
1734  <!-- AdminLTE dashboard demo (This is only for demo purposes) -->
1735  <script src="{{ asset('template/dist/js/pages/dashboard2.js')}}"></script>
1736  </body>
1737  </html>
```


Menjalankan ke web

m_user form

m_level form

2	
	``php
54   Route::get('/kategori/create', [KategoriController::class, 'create']);
55   Route::post('/kategori', [KategoriController::class, 'store']);
```

Route Ke Kategori |
| | ```php
30    public function create(){
31        return view('kategori.create');
32    }
33
34    public function store(Request $req): RedirectResponse{
35        $validated = $req->validate([
36            'kategori_kode' => 'required',
37            'kategori_nama' => 'required'
38        ]);
39
40        return redirect('/kategori');
41    }
```

Memperbaiki KategoriController agar memvalidasi data terlebih dahulu |

Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet Minggu-6: Template Form (AdminLTE), Server Validation,
Client Validation, CRUD
Mata Kuliah Pemprograman Web Lanjut

```html
<form action="/kategori" method="post">
    @if ($errors->any())
    <div class="alert alert-danger">
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
    @endif
```

Mengembalikan error apabila tidak tervalidasi

Buat Kategori Baru

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

Kode Kategori [_____]

Nama Kategori [_____]

[submit]

```html
<div class="form-group">
    <label for="kategori_kode">Kode Kategori</label>
    <input
        type="text"
        name="kategori_kode"
        id="kategori_kode"
        class="@error('kategori_kode') is-invalid @enderror">
    @error('kategori_kode')
        <div class="alert alert-danger">{{ $message }}</div>
    @enderror
</div>
```

Menambah Pesan Error dibawah field

	Buat Kategori Baru • The kategori kode field is required. • The kategori nama field is required. **Kode Kategori** [_____] The kategori kode field is required. **Nama Kategori** [_____] [submit]
| 3 | ```
$ php artisan make:request StorePostRequest
PHP Warning:  Module "zip" is already loaded in Unknown on line 0

Warning: Module "zip" is already loaded in Unknown on line 0

   INFO  Request [D:\Tugas Dokumen Kuliah\Tugas\Learn\Laravel PWL\POS\app\Http\R
equests\StorePostRequest.php] created successfully.
```<br>membuat StorePostRequest |
| | ```
22      public function rules(): array
23      {
24          return [
25              'kategori_kode' => 'required',
26              'kategori_nama' => 'required'
27          ];
28      }
29  }
```<br>Membuat rule pada StorePostRequest |
| | ```
public function store(StorePostRequest $req): RedirectResponse{
    $validated = $req->validated();

    $validated = $req->safe()->only(['kategori_kode', 'kategori_nama']);
    $validated = $req->safe()->except(['kategori_kode', 'kategori_nama']);

    return redirect('/kategori');
}
```<br>menyesuaikan store pada KategoriController |

Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet Minggu-6: Template Form (AdminLTE), Server Validation,
Client Validation, CRUD
Mata Kuliah Pemprograman Web Lanjut

	Home › Create
	Buat Kategori Baru
	• The kategori kode field is required.
	• The kategori nama field is required.
	Kode Kategori
	The kategori kode field is required.
	Nama Kategori
	submit
| 4 | ```
$ php artisan make:controller POSController --resource
PHP Warning:  Module "zip" is already loaded in Unknown on line 0

Warning: Module "zip" is already loaded in Unknown on line 0

   INFO  Controller [D:\Tugas Dokumen Kuliah\Tugas\Learn\Laravel PWL\POS\app\Http\Controllers\POSController.php] created successfully.
```<br><br>Membuat controller POSController |
| | ```
74 │    Route::resource('user', POSController::class);
```<br><br>Membuat resource route |
| | ```
 8 ∨ class m_user extends Model{
 9
10     protected $table = "m_user";
11     public $timestamps = false;
12     protected $primaryKey = 'user_id';
13
14 ∨   protected $fillable = [
15         'user_id',
16         'level_id',
17         'username',
18         'nama',
19         'password'
20     ];
21 }
```<br><br>m_user model |

```php
 7    return new class extends Migration
 8    {
 9        /**
10         * Run the migrations.
11         */
12        public function up(): void
13        {
14            Schema::create('useri', 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')->on('m_level');
23            });
24        }
25
26        /**
27         * Reverse the migrations.
28         */
29        public function down(): void
30        {
31            Schema::dropIfExists('m_level');
32        }
33    };
```

Migration m_user

PosController.php

```php
public function index()
{
    $useri = m_user::all();
    return view('user.index', compact('useri'))->with('i');
}
```

```php
22        public function create()
23        {
24            return view('user.create');
25        }
```

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

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

```php
public function show(string $id, m_user $useri)
{
    $useri = m_user::findOrFail($id);
    return view('user.show', compact('useri'));
}
```

```php
    public function edit(string $id)
    {
        $useri = m_user::find($id);
        return view('user.edit', compact('useri'));
    }
```

```php
public function update(Request $request, string $id)
{
    $request->validate([
        'username' => 'required',
        'nama' => 'required',
        'password' => 'required'
    ]);

    m_user::find($id)->update($request->all());
    return redirect()->route('user.index')->with('success', 'Data Berhasil DiUpdate');
}
```

```php
public function destroy(string $id)
{
    $useri = m_user::findOrFail($id)->delete();
    return redirect()->route('user.index')->with('success', 'Data Berhasil DiHapus');
}
```

```php
resources > views > user > template.blade.php > html
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <title>CRUD Laravel</title>
5       <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
6   </head>
7   <body>
8       <div class="container">
9           @yield('content')
10      </div>
11  </body>
12  </html>
```

template.blade.php

```php
resources > views > user > index.blade.php > table.table.table-bordered > tr > th.text-center
1   @extends('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 href="{{ route('user.create') }}" class="btn btn-success">Input User</a>
10          </div>
11      </div>
12  </div>
13  @if ($message = Session::get('success'))
14  <div class="alert alert-success">
15      <p>{{ $message }}</p>
16  </div>
17  @endif
18  <table class="table table-bordered">
19      <tr>
20          <th class="text-center" width="20px">User Id</th>
21          <th class="text-center" width="150px">Level Id</th>
22          <th class="text-center" width="200px">Username</th>
23          <th class="text-center" width="200px">Nama</th>
24          <th class="text-center" width="150px">Password</th>
25      </tr>
26      @foreach ( $useri as $m_user )
27      <tr>
28          <td>{{ $m_user->user_id }}</td>
29          <td>{{ $m_user->level_id }}</td>
30          <td>{{ $m_user->username }}</td>
31          <td>{{ $m_user->nama }}</td>
32          <td>{{ $m_user->password }}</td>
33          <td>
34              <form method="POST" action="{{ route('user.destroy', $m_user->user_id)}}">
35                  <a class="btn btn-info btn-sm" href="{{ route('user.show', $m_user->user_id )}}">Show</a>
36                  <a class="btn btn-primary btn-sm" href="{{ route('user.edit', $m_user->user_id )}}">Edit</a>
37                  @csrf
38                  @method('DELETE')
```

index.blade.php

```php
resources > views > user > create.blade.php > form
1   @extends('user/template')
2   @section('content')
3   <div class="row mt-5 bm-5">
4       <div class="com-lg-12 margin-tb">
5           <div class="float-left">
6               <h2>Membuat Daftar User</h2>
7           </div>
8           <div class="float-right">
9               <a href="{{ route('user.index') }}" class="btn btn-secondary">Kembali</a>
10          </div>
11      </div>
12  </div>
13  @if($errors->any())
14  <div class="alert alert-danger">
15      <strong>Ops</strong> Input gagal <br><br>
16      <ul>
17          @foreach($errors->all() as $error)
18          <li>{{ $error }}</li>
19          @endforeach
20      </ul>
21  </div>
22  @endif
23  <form action="{{ route('user.store') }}" method="POST">
24      @csrf
25      <div class="col-xs-12 col-sm-12 sol-md-12">
26          <div class="form-group">
27              <strong>Username:</strong>
28              <input type="text" name="username" class="form-control" placeholder="Masukkan username"></input>
29          </div>
30      </div>
31
32      <div class="col-xs-12 col-sm-12 col-md-12">
33          <div class="form-group">
34              <strong>Nama:</strong>
35              <input type="text" name="nama" class="form-control" placeholder="Masukkan nama"></input>
36          </div>
37      </div>
```

create.blade.php

```php
resources > views > user > edit.blade.php > ...
1   @extends('user/template')
2   @section('content')
3   <div class="row mt-5 bm-5">
4       <div class="com-lg-12 margin-tb">
5           <div class="float-left">
6               <h2>Edit User</h2>
7           </div>
8           <div class="float-right">
9               <a href="{{ route('user.index') }}" class="btn btn-secondary">Kembali</a>
10          </div>
11      </div>
12  </div>
13  @if($errors->any())
14  <div class="alert alert-danger">
15      <strong>Ops</strong> Input gagal <br><br>
16      <ul>
17          @foreach($errors->all() as $error)
18          <li>{{ $error }}</li>
19          @endforeach
20      </ul>
21  </div>
22  @endif
23  <form action="{{ route('user.update', $useri->user_id) }}" method="POST">
24      @csrf
25      @method('PUT')
26      <div class="col-xs-12 col-sm-12 sol-md-12">
27          <div class="form-group">
28              <strong>User_id:</strong>
29              <input type="text" name="userid" class="form-control" placeholder="Masukkan username" value="{{ $useri->user_id }}"></input>
30          </div>
31      </div>
32
33      <div class="col-xs-12 col-sm-12 sol-md-12">
34          <div class="form-group">
35              <strong>Level_id:</strong>
36              <input type="text" name="levelid" class="form-control" placeholder="Masukkan username" value="{{ $useri->level_id }}"></input>
37          </div>
38      </div>
```

edit.blade.php

show.blade.php

Halaman utama crud untuk m_user

Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet Minggu-6: Template Form (AdminLTE), Server Validation,
Client Validation, CRUD
Mata Kuliah Pemprograman Web Lanjut

	Halaman untuk menambahkan data user baru
	Show User Kembali **User_id:** 1 **Level_id:** 1 **Username:** admin **Nama:** Administrator **Password:** $2y$12$9lmWA2DR7Y08RIVDyMXvr.G3G8uyBlyYegK6dR9883XFjK1bkIghO Halaman untuk menunjukan data 1 per satu

	Edit User
	Kembali
	User_id:
	`1`
	Level_id:
	`1`
	Username:
	`admin`
	Nama:
	`Administrator`
	Password:
	`••••••••••••••••••••••••••••••••••••••••••••`
	Update
	
	Halaman untuk mengedit data user

1.	(code below)

```php
<?php

namespace App\Models;

use Illuminate\database\Eloquent\Factories\HasFactory;
use Illuminate\database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasOne;

class m_user extends Model{

    protected $table = "m_user";
    public $timestamps = false;
    protected $primaryKey = 'user_id';

    protected $fillable = [
        'user_id',
        'level_id',
        'username',
        'nama',
        'password'
    ];

    public function barang(): HasOne {
        return $this->hasOne(LevelModel::class, 'level_id', 'level_id');
    }
}
```

User Id	Level Id	Username	Nama	Password	
1	1	admin	Administrator	$2y$12$9lmWA2DR7Y08RIVDyMXvr.G3G8uyBlyYegK6dR9883XFjK1bkIghO	Show Edit Delete
2	2	manager	Manager	$2y$12$RH9Yv6ENZfID7p3s6C9LN.wFejucAyKK2vcgIOcf.7YEB6RCYifUy	Show Edit Delete
3	3	staff	Staff/Kasir	$2y$12$WytmAYZ090oCFNcrQl8ZkusXAaJtHIYRKNgIDOpGui.roTRLnF.aK	Show Edit Delete
4	4	customer-1	Pelanggan Pertama	$2y$12$JuR0HGapijN.U6yHfDZRC.zYTLOSN3Srbhbxhr7HcMLem8TrxLymq	Show Edit Delete
5	2	manager-dua	Manager 2	$2y$12$6VbkA.7Ebc7fKnpXeuT20edlS0/LQxbEKKGpeAizD2y3xVLh8Ztk6	Show Edit Delete
6	2	manager22	Manager Dua Dua	$2y$12$Yo5NnBg5c4SVoKGz8hnBN.AvPGHNH9O1UQ3NHL4UXRG0qNJ.0ofdO	Show Edit Delete

2	
3	- **$request->validate**: Metode ini digunakan untuk memvalidasi data yang dikirimkan melalui permintaan HTTP. Laravel memiliki fitur built-in untuk validasi data, dan $request->validate adalah salah satu cara untuk menggunakannya. - **$error**: Variabel ini digunakan untuk menampilkan pesan kesalahan yang terkait dengan validasi. Setelah melakukan validasi menggunakan **$request->validate**, jika ada kesalahan, Laravel akan secara otomatis menempatkan pesan kesalahan ke dalam variabel **$error**