

Nama : Raffy Jamil Octavialdy

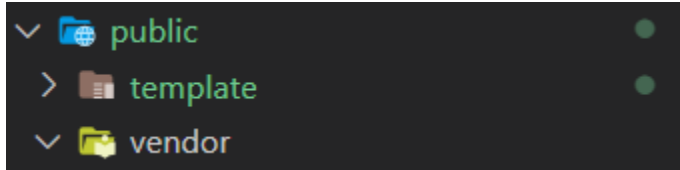
NIM : 2241720082

## JOBSHEET 06

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

#### A. Template Form (AdminLTE)

1. Download Admin Dashboard Template AdminLTE Bootstrap, extract All dan ubah nama folder AdminLTE-3.2.0 menjadi template.



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

```
<!-- 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') }}">
</head>
<body class="hold-transition dark-mode sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">
<div class="wrapper">

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

    <!-- REQUIRED SCRIPTS -->
    <!-- jQuery -->
    <script src="{{ asset('template/plugins/jquery/jquery.min.js') }}"></script>
    <!-- Bootstrap -->
    <script src="{{ asset('template/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
    <!-- overlayScrollbars -->
    <script src="{{ asset('template/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js') }}"></script>
    <!-- AdminLTE App -->
    <script src="{{ asset('template/dist/js/adminlte.js') }}"></script>

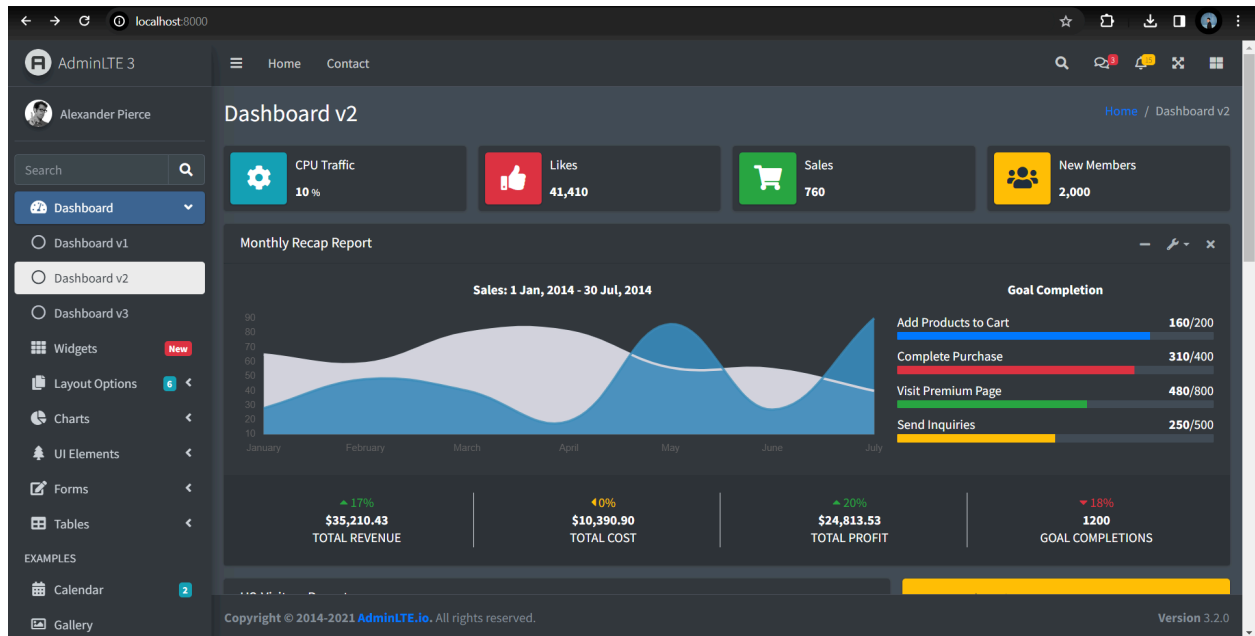
    <!-- PAGE PLUGINS -->
    <!-- jQuery Mapael -->
    <script src="{{ asset('template/plugins/jquery-mousewheel/jquery.mousewheel.js') }}"></script>
    <script src="{{ asset('template/plugins/raphael/raphael.min.js') }}"></script>
    <script src="{{ asset('template/plugins/jquery-mapael/jquery.mapael.min.js') }}"></script>
    <script src="{{ asset('template/plugins/jquery-mapael/maps/usa_states.min.js') }}"></script>
    <!-- ChartJS -->
    <script src="{{ asset('template/plugins/chart.js/Chart.min.js') }}"></script>

    <!-- AdminLTE for demo purposes -->
    <script src="{{ asset('template/dist/js/demo.js') }}"></script>
    <!-- AdminLTE dashboard demo (This is only for demo purposes) -->
    <script src="{{ asset('template/dist/js/pages/dashboard2.js') }}"></script>
</body>
</html>
```

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

### 3. Jalankan pada browser



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

The screenshot shows the Raffy Store Dashboard. The left sidebar contains a menu with 'User', 'Level', and 'Kategori' options. The main content area is titled 'Dashboard' and contains a form with the label 'Level id'. The form has an input field for 'id' and a 'Submit' button.

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

5. Eksplorasi jenis form pada adminLTE dan coba terapkan yang sesuai untuk studi kasus POS PWL, buatlah form untuk tabel m\_user dan m\_level

The screenshot shows the 'Form User' in the 'Raffy Store' admin interface. The browser address bar indicates 'localhost:8000/user'. The left sidebar contains a search bar and a menu with 'User', 'Level', and 'Kategori'. The 'User' menu item is selected. The main content area is titled 'Dashboard' and contains the 'Form User'. The form has the following fields: 'Nama' (text input), 'Level' (dropdown menu with 'Admin' selected), 'Username' (text input), 'Password' (text input), and a 'Submit' button.

The screenshot shows the 'Form Level' in the 'Raffy Store' admin interface. The browser address bar indicates 'localhost:8000/level'. The left sidebar contains a search bar and a menu with 'User', 'Level', and 'Kategori'. The 'Level' menu item is selected. The main content area is titled 'Dashboard' and contains the 'Form Level'. The form has the following fields: 'Kode' (text input), 'Nama' (text input), and a 'Submit' button.

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

## B. VALIDASI PADA SERVER

### 1. Edit pada KategoriKontroller

```
public function store(Request $request): RedirectResponse
{
    $validate = $request->validate([
        'kategori_kode' => 'bail|required',
        'kategori_nama' => 'required',
    ]);

    return redirect('/kategori');
}
```

### 2. Tulis perbedaan penggunaan validate dengan validateWithBag!

- Metode validate() biasanya digunakan di dalam controller. Jika validasi gagal, Laravel secara otomatis akan mengarahkan kembali pengguna ke halaman sebelumnya dengan pesan error yang sesuai. Pesan error validasi akan dikirim kembali ke tampilan dan dapat diakses menggunakan fungsi bantuan errors().
- Metode validateWithBag() memberikan fleksibilitas yang lebih besar dalam menangani pesan error validasi. Dengan menggunakan metode ini, dapat mengontrol di mana pesan error validasi disimpan dan bagaimana pesan tersebut ditampilkan kepada pengguna.

3. Menggunakan bail untuk menghentikan validasi pada field setelah kegagalan validasi pertama, Sehingga, jika validasi untuk kode\_kategori gagal, maka Laravel akan menghentikan validasi dan tidak mengevaluasi aturan validasi untuk nama\_kategori.

```
$validate = $request->validate([
    'kategori_kode' => 'bail|required',
    'kategori_nama' => 'required',
]);
```

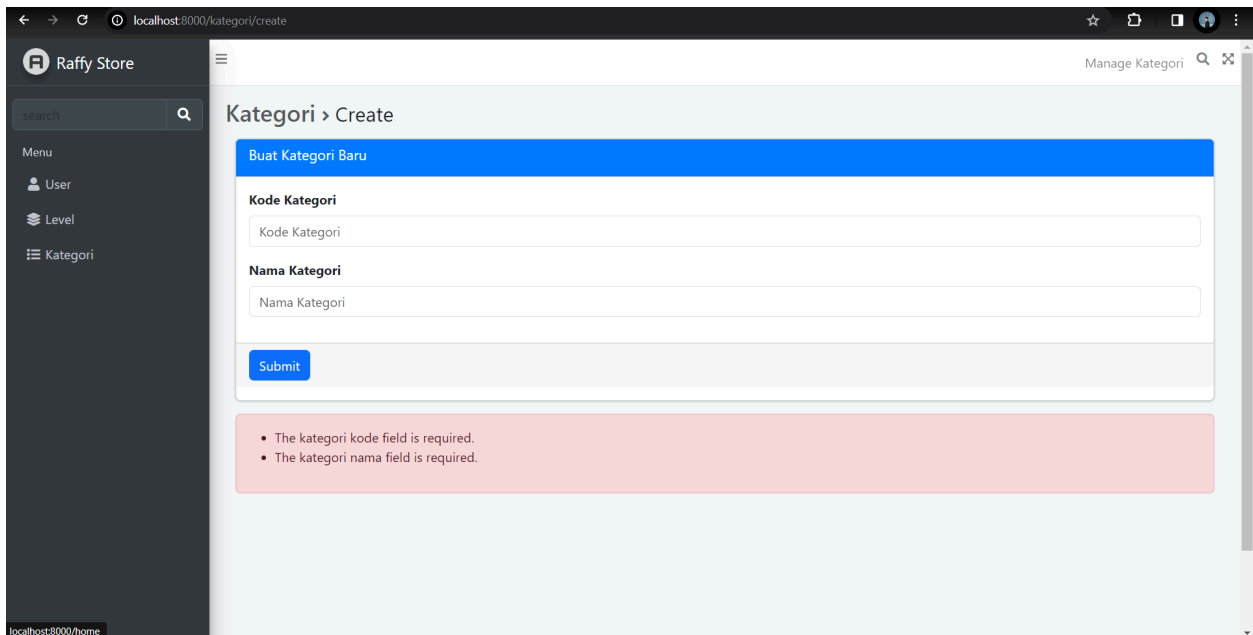
Nama : Raffy Jamil Octavialdy

NIM : 2241720082

4. Pada view/create.blade.php tambahkan code berikut agar ketika validasi gagal, kita dapat menampilkan pesan kesalahan dalam tampilan

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

5. Screenshot hasilnya



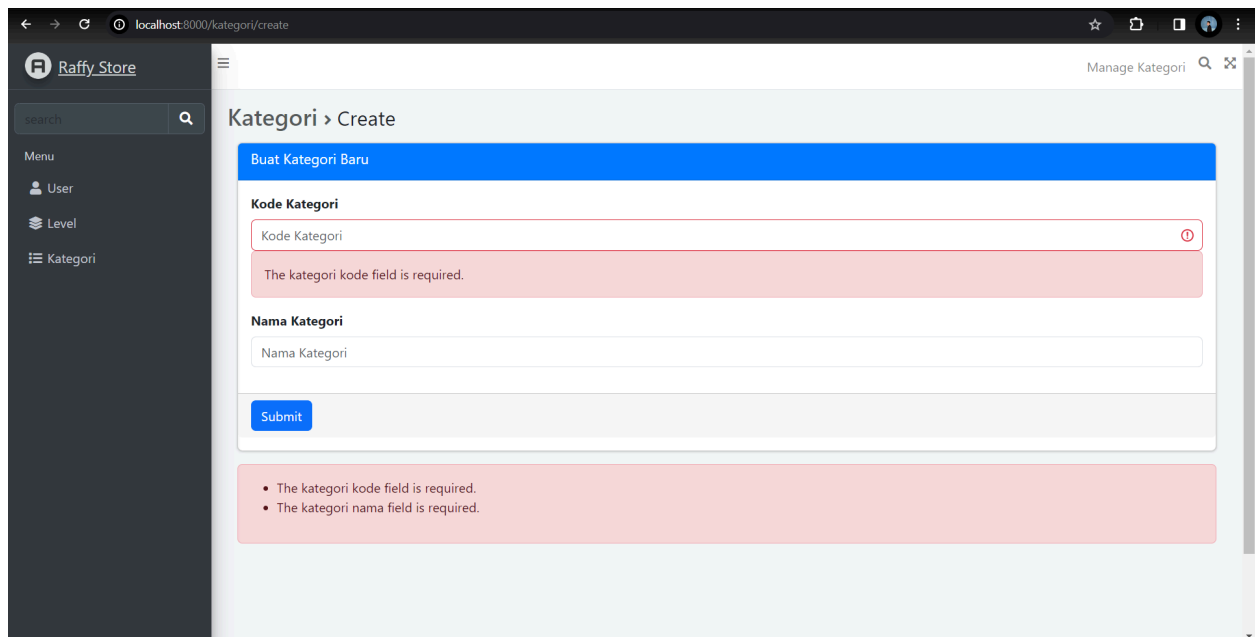
Nama : Raffy Jamil Octavialdy

NIM : 2241720082

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

```
<label for="kategori_kode">Kode Kategori</label>
<input type="text"
    name="kategori_kode"
    id="kategori_kode"
    class="@error('kategori_kode') is-invalid @enderror form-control"
    placeholder="Kode Kategori">

@error('kategori_kode')
    <div class="alert alert-danger">{{ $message }}</div>
@enderror
```



## C. FORM REQUEST VALIDATION

1. Membuat permintaan form dengan menuliskan pada terminal

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> php artisan make:request StorePostRequest
INFO Request [D:\Semester 4\Pemrograman Web Lanjut\PWL_POS\app\Http\Requests\StorePostRequest.php] created successfully.
```

2. Ketik code berikut pada Http/request/StorePostRequest

```
public function rules(): array
{
    return [
        'kategori_kode' => 'required',
        'kategori_nama' => 'required',
    ];
}
```

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

```
public function store(StorePostRequest $request): RedirectResponse
{
    // $validate = $request->validate([
    //     'kategori_kode' => 'bail|required',
    //     'kategori_nama' => 'required',
    // ]);

    // Retrieve the validated input data...
    $validated = $request->validated();

    // Retrieve a portion of the validated input data...
    $validated = $request->safe()->only(['kategori_kode', 'kategori_nama']);
    $validated = $request->safe()->except(['kategori_kode', 'kategori_nama']);

    return redirect('/kategori');
}
```

3. Terapkan validasi juga pada tabel m\_user dan m\_level.

The screenshot shows the 'Form User' page in the 'Raffy Store' application. The page has a sidebar menu with 'User' selected. The main content area is titled 'Dashboard' and contains a 'Form User' section. The form has four fields: 'Nama' (text input), 'Level' (dropdown menu with 'Admin' selected), 'Username' (text input), and 'Password' (text input). A blue 'Submit' button is at the bottom of the form. Below the form, a red error message box displays the following messages: 'The nama field is required.', 'The username field is required.', and 'The password field is required.'

The screenshot shows the 'Form Level' page in the 'Raffy Store' application. The page has a sidebar menu with 'Level' selected. The main content area is titled 'Dashboard' and contains a 'Form Level' section. The form has two fields: 'Kode' (text input) and 'Nama' (text input). A blue 'Submit' button is at the bottom of the form. Below the form, a red error message box displays the following messages: 'The level kode field is required.' and 'The level nama field is required.'

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

## 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 D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> php artisan make:controller POSController --resource
INFO Controller [D:\Semester 4\Pemrograman Web Lanjut\PWL_POS\app\Http\Controllers\POSController.php] created successfully.
```

2. Tambahkan kode pada route/web.php

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

3. Atur pada Models

```
class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user'; // mendefinisikan nama tabel yang digunakan oleh model ini
    public $timestamps = false;
    protected $primaryKey = 'user_id'; // mendefinisikan primary key dari tabel yang digunakan

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

4. Atur pada Migration

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('m_user', function (Blueprint $table) {
            $table->id('user_id');
            $table->unsignedBigInteger('level_id')->index(); // indexing untuk ForeignKey
            $table->string('username', 20)->unique(); // unique untuk memastikan tidak ada username yang sama
            $table->string('nama', 100);
            $table->string('password');
            $table->timestamps();

            // mendefinisikan Foreign Key pada kolom level_id mengacu pada kolom level_id di tabel m_level
            $table->foreign('level_id')->references('level_id')->on('m_level');
        });
    }
}
```



Nama : Raffy Jamil Octavialdy

NIM : 2241720082

5. Buka app/Http/Controllers/POSController.php kemudian ketikkan, kodenya seperti berikut ini

```
class POSController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        // fungsi eloquent menampilkan data menggunakan pagination
        $useri = UserModel::all(); // Mengambil semua isi tabel
        return view('m_user.index', compact('useri'))->with('i');
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('m_user.create');
    }
}
```

6. Buatlah folder di Resources/Views/m\_user dengan beberapa blade dan isian kode berikut

1. template.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>CRUD Laravel</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
</head>
<body>
    <div class="container">
        @yield('content')
    </div>
</body>
</html>
```

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

## 2. index.blade.php

```
@extends('m_user/template')

@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2>CRUD user</h2>
            </div>
            <div class="float-right">
                <a class="btn btn-success" href="{{ route('m_user.create') }}">Input User</a>
            </div>
        </div>
    </div>

    @if ($message = Session::get('success'))
        <div class="alert alert-success">
            <p>{{ $message }}</p>
        </div>
    @endif

    <table class="table table-bordered">
        <tr>
            <th width="20px" class="text-center">User id</th>
            <th width="150px" class="text-center">Level id</th>
            <th width="200px" class="text-center">Username</th>
            <th width="200px" class="text-center">Nama</th>
        </tr>
    </table>
</section>
```

## 3. create.blade.php

```
@extends('m_user/template')

@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2>Membuat Daftar User</h2>
            </div>
            <div class="float-right">
                <a class="btn btn-secondary" href="{{ route('m_user.index') }}">Kembali</a>
            </div>
        </div>
    </div>

    @if ($errors->any())
        <div class="alert alert-danger">
            <strong>Ops</strong> Input gagal<br><br>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('m_user.store') }}" method="POST">
    </form>
</section>
```

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

#### 4. show.blade.php

```
@extends('m_user/template')

@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2>Show User</h2>
            </div>
            <div class="float-right">
                <a class="btn btn-secondary" href="{{ route('m_user.index') }}">Kembali</a>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>User_id:</strong>
                {{ $useri->user_id }}
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Level_id:</strong>
                {{ $useri->level_id }}
            </div>
        </div>
    </div>
</div>
```

#### 5. edit.blade.php

```
@extends('m_user/template')

@section('content')
    <div class="row mt-5 mb-5">
        <div class="col-lg-12 margin-tb">
            <div class="float-left">
                <h2>Edit User</h2>
            </div>
            <div class="float-right">
                <a class="btn btn-secondary" href="{{ route('m_user.index') }}">Kembali</a>
            </div>
        </div>
    </div>

    @if ($errors->any())
        <div class="alert alert-danger">
            <strong>Ops!</strong> Error <br><br>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('m_user.update', $useri->user_id) }}" method="POST">
```

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

Silahkan akses localhost/m\_user



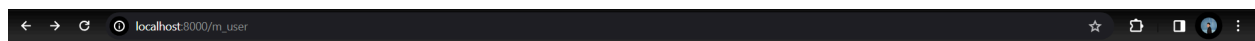
## CRUD user

Input User

| User id | Level id | Username     | Nama              | Password         |  |
|---------|----------|--------------|-------------------|------------------|--|
| 1       | 1        | admin        | Administrator     | \$2y\$12\$AQZ... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 2       | 2        | manager      | Manager           | \$2y\$12\$TnN... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 3       | 3        | staff        | Staff/Kasir       | \$2y\$12\$A4Q... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 5       | 4        | customer-1   | Pelanggan Pertama | \$2y\$12\$sy5... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 6       | 2        | Manager_dua  | Manager 2         | \$2y\$12\$bca... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 9       | 2        | Manager_tiga | Manager 3         | \$2y\$12\$Z31... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 13      | 2        | manager22    | Manager Dua Dua   | \$2y\$12\$1OQ... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 18      | 2        | manager56    | Manager55         | \$2y\$12\$EWS... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |

Tugas:

1. Coba tampilkan level\_id pada halaman web tersebut dimana field ini merupakan foreign key



## CRUD user

Input User

| User id | Level id | Username     | Nama              | Password         |  |
|---------|----------|--------------|-------------------|------------------|--|
| 1       | 1        | admin        | Administrator     | \$2y\$12\$AQZ... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 2       | 2        | manager      | Manager           | \$2y\$12\$TnN... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 3       | 3        | staff        | Staff/Kasir       | \$2y\$12\$A4Q... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 5       | 4        | customer-1   | Pelanggan Pertama | \$2y\$12\$sy5... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 6       | 2        | Manager_dua  | Manager 2         | \$2y\$12\$bca... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 9       | 2        | Manager_tiga | Manager 3         | \$2y\$12\$Z31... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 13      | 2        | manager22    | Manager Dua Dua   | \$2y\$12\$1OQ... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| 18      | 2        | manager56    | Manager55         | \$2y\$12\$EWS... | <a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a> |

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

## 2. Modifikasi dengan tema/template kesukaan Anda

The image displays two screenshots of a web application interface for 'RAFFY STORE', showing user management functionality.

**Top Screenshot: Users List**

The browser address bar shows `localhost:8000/users`. The left sidebar menu includes 'Dashboard', 'General Dashboard', 'Users', 'Level', and 'Kategori'. The main content area is titled 'Users' and includes an 'Add New' button. Below the title, there is a description: 'You can manage all Users, such as editing, deleting and more.' A section titled 'All Users' contains a table with columns: Nama, Level ID, Username, Password, and Action. The table lists four users, each with 'Edit' and 'Delete' buttons.

| Nama            | Level ID | Username     | Password  | Action                                      |
|-----------------|----------|--------------|---|---|
| Manager55       | 2        | manager56    | \$2y\$12\$EWSR.OTIH20r.UohjKMaz.eDbFx8frsOU91v2otcurlsMhZ66Hz/m | <a href="#">Edit</a> <a href="#">Delete</a> |
| Manager Dua Dua | 2        | manager22    | \$2y\$12\$10QbzeogU3vzhcLjxubYmuLMQeLc0yZEwdkaW0dFEUwsDBURr256O | <a href="#">Edit</a> <a href="#">Delete</a> |
| Manager 3       | 2        | Manager_tiga | \$2y\$12\$Z31yRUH.I6SleH0EUMe6dOKW.WZvg/dhFyHhwK1uw6tMs0vkzZTK2 | <a href="#">Edit</a> <a href="#">Delete</a> |
| Manager 2       | 2        | Manager_dua  | \$2y\$12\$bcafqf8wd4xMczl2OeCkr.lrkpB5QOMbqbokcQwVDUBEcTZ5UalPe | <a href="#">Edit</a> <a href="#">Delete</a> |

**Bottom Screenshot: Add User Form**

The browser address bar shows `localhost:8000/users/create`. The left sidebar menu is the same. The main content area is titled 'Tambah User' and includes a description: 'You can manage all Users, such as editing, deleting and more.' A section titled 'Input Text' contains form fields for 'Nama', 'Username', and 'Password'. Below these fields is a 'Level' selection dropdown menu with options: 'Admin', 'Manager', and 'Staff'.

Nama : Raffy Jamil Octavialdy

NIM : 2241720082

RAFFY STORE

DASHBOARD

Dashboard >

Edit Users

Dashboard / Forms / Users

Users

Input Text

Nama

Manager

Username

manager

Password

Level

Admin Manager Staff

Submit

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

**Jawab:**

- `$request->validate`: Ini adalah metode dari objek `$request` yang digunakan untuk melakukan validasi input yang diterima dari pengguna. Validasi dilakukan berdasarkan aturan yang didefinisikan di dalam metode `rules()` pada file Form Request atau secara langsung di dalam controller. Jika validasi gagal, maka metode ini akan mengembalikan pesan kesalahan.
- `$errors`: Variabel ini digunakan untuk menyimpan pesan-pesan kesalahan yang dihasilkan dari validasi input. Jika validasi gagal, maka pesan kesalahan akan disimpan di dalam variabel ini. Ini dapat diakses di dalam tampilan Blade untuk menampilkan pesan kesalahan kepada pengguna.
- `alert`: Ini adalah elemen HTML yang digunakan untuk menampilkan pesan kesalahan kepada pengguna. Dalam konteks halaman CRUD, pesan kesalahan biasanya ditampilkan dalam elemen `alert` untuk memberi tahu pengguna tentang kesalahan yang terjadi selama proses validasi atau operasi CRUD.

Dengan menggunakan fungsi `$request->validate` untuk validasi input dan variabel `$errors` untuk menangani pesan kesalahan, serta menggunakan elemen `alert` di dalam tampilan Blade.