

# Laporan Modul 4: Form Submission & Data Validation

---

**Mata Kuliah:** Workshop Web Lanjut

**Nama:** Maisha Zahrani **NIM:** 2024573010052 **Kelas:** TI-2C

---

## Abstrak

Form submission dan data validation merupakan bagian penting dalam pengembangan aplikasi web. Laravel menyediakan fitur lengkap untuk mengatur input pengguna, melakukan validasi data, serta memberikan respon yang sesuai. Dengan keamanan seperti CSRF protection dan sistem validasi yang fleksibel, Laravel membantu pengembang membangun aplikasi yang aman, efisien, dan ramah pengguna.

---

## 1. Dasar Teori

Apa itu Form Submission dan Data Validation? Form submission dan validasi data merupakan komponen krusial dalam pengembangan aplikasi web modern. Laravel menyediakan sistem yang robust dan elegant untuk menangani input pengguna, memvalidasi data, dan memberikan feedback yang informatif. Sistem ini dirancang dengan prinsip keamanan, kemudahan penggunaan, dan fleksibilitas.

Proses Form Submission di Laravel mengikuti konsep MVC (Model-View-Controller):

1. View berfungsi menampilkan form HTML kepada pengguna.
2. Controller menerima data dari form dan memprosesnya.
3. Model berinteraksi dengan database untuk menyimpan atau mengambil data.
4. Request membawa data dari client ke server, sedangkan Response mengembalikan hasil pemrosesan ke pengguna.

Laravel mendukung beberapa HTTP Method seperti POST, PUT, PATCH, DELETE, dan GET. Karena HTML hanya mendukung GET dan POST, Laravel menggunakan method spoofing dengan direktif `@method('PUT')` agar dapat meniru method lain.

Alur Form Submission secara umum:

1. Pengguna mengisi form dan menekan tombol submit.
2. Browser mengirimkan request ke route yang sudah ditentukan.
3. Middleware Laravel memproses request (termasuk pengecekan CSRF token).
4. Controller melakukan validasi data.
5. Jika validasi berhasil, data akan disimpan; jika gagal, Laravel mengembalikan pesan error.
6. Hasilnya dikirim kembali ke pengguna melalui response.

CSRF Protection (Cross-Site Request Forgery) berfungsi mencegah pengguna tanpa sadar mengirim permintaan berbahaya. Laravel melindungi setiap form dengan CSRF token unik yang disertakan melalui `@csrf`. Token ini akan dibandingkan dengan token di session, dan jika tidak cocok maka request akan ditolak (error 419).

Validasi Data penting untuk menjaga keamanan dan kualitas aplikasi. Validasi memastikan data yang dikirim pengguna sesuai aturan, mencegah kesalahan logika, dan melindungi aplikasi dari serangan seperti SQL Injection atau XSS. Laravel menyediakan beberapa cara untuk melakukan validasi:

1. Controller Validation menggunakan `$request->validate([...])` langsung di controller.
2. Form Request Validation menggunakan class khusus untuk memisahkan aturan validasi.
3. Manual Validation menggunakan `Validator::make()` untuk kontrol yang lebih detail.

Kategori Aturan Validasi di Laravel cukup beragam:

1. Basic Rules: required, nullable, filled, sometimes
2. Tipe Data: string, numeric, integer, boolean, file, array
3. Ukuran: min, max, between, size
4. Database: unique, exists
5. Format: email, url, ip, date, regex
6. Perbandingan: confirmed, same, different
7. Kondisional: required\_if, required\_with, required\_unless, dan lain-lain

Saat validasi gagal, Laravel secara otomatis membuat Error Bag berisi pesan kesalahan untuk tiap field. Pesan error ini bisa ditampilkan di view dengan:

1. `@error('field')` untuk menampilkan error per field.
2. `$errors->all()` untuk menampilkan semua pesan error sekaligus.
3. `$errors->has('field')` untuk mengecek error tertentu.

Laravel juga menyediakan fungsi `old()` agar input sebelumnya tetap muncul setelah validasi gagal. Ini membantu pengguna tidak perlu mengetik ulang data yang sudah diisi sebelumnya. Pesan error bawaan Laravel bisa diganti dengan custom message agar lebih mudah dimengerti pengguna. Selain itu, developer juga bisa membuat aturan validasi kustom, baik dengan class rule sendiri (misal Uppercase) maupun dengan closure validation langsung di controller.

---

## 2. Langkah-Langkah Praktikum

Tuliskan langkah-langkah yang sudah dilakukan, sertakan potongan kode dan screenshot hasil.

### 2.1 Praktikum 1 – Menangani Request dan Response View di Laravel 12

1. Buat dan Buka Proyek Laravel `laravel new form-app` cd `form-app` code .
2. Definisikan Rute, edit `routes/web.php`

projects > form-app > routes > web.php > ...

```

1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\FormController;
5  use App\Http\Controllers\RegisterController;
6
7  Route::get('/register', [RegisterController::class, 'showForm'])->name('register.show');
8  Route::post('/register', [RegisterController::class, 'handleForm'])->name('register.handle');
9  Route::get('/form', [FormController::class, 'showForm'])->name('form.show');
10 Route::post('/form', [FormController::class, 'handleForm'])->name('form.handle');
11 Route::get('/result', [FormController::class, 'showResult'])->name('form.result');
12 Route::get('/', function () {
13     return view('welcome');
14 });
15

```

3. Buat controller untuk menangani logika form. php artisan make:controller FormController Ini akan membuat file controller baru di app/Http/Controllers/FormController.php. Buka file ini dan tambahkan metode berikut:

projects > form-app > app > Http > Controllers > FormController.php > ...

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class FormController extends Controller
8  {
9      public function showForm()
10     {
11         return view('form');
12     }
13
14     public function handleForm(Request $request)
15     {
16         $validated = $request->validate([
17             'name' => 'required|string|max:255',
18             'email' => 'required|email',
19             'age' => 'required|integer|min:1',
20             'password' => 'required|min:6',
21             'gender' => 'required',
22             'role' => 'required',
23             'bio' => 'required',
24             'confirm' => 'accepted',
25         ]);
26
27         return redirect()->route('form.result')->with('data', $validated);
28     }
29
30     public function showResult()
31     {
32         $data = session('data');
33         return view('result', compact('data'));
34     }
35 }

```

4. Buat View Form Buat file view Blade baru untuk form. Buat direktori bernama form di dalam resources/views dan buat file bernama form.blade.php di dalamnya.

```

1 <DOCTYPE html>
2 <html>
3 <head>
4 <title>Pengiriman Form</title>
5 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
6 </head>
7 <body class="container mt-5">
8 <h2>Form Registrasi</h2>
9 <form method="POST" action="{{ route('form.handle') }}">
10 @csrf
11 <div class="mb-3">
12 <label>Nama</label>
13 <input type="text" name="name" class="form-control" value="{{ old('name') }}">
14 @error('name') <div class="text-danger">{{ $message }}</div> @enderror
15 </div>
16 <div class="mb-3">
17 <label>Email</label>
18 <input type="email" name="email" class="form-control" value="{{ old('email') }}">
19 @error('email') <div class="text-danger">{{ $message }}</div> @enderror
20 </div>
21 <div class="mb-3">
22 <label>Umur</label>
23 <input type="number" name="age" class="form-control" value="{{ old('age') }}">
24 @error('age') <div class="text-danger">{{ $message }}</div> @enderror
25 </div>
26 <div class="mb-3">
27 <label>Password</label>
28 <input type="password" name="password" class="form-control">
29 @error('password') <div class="text-danger">{{ $message }}</div> @enderror
30 </div>
31 <div class="mb-3">
32 <label>Jenis Kelamin</label><br>
33 <div class="form-check form-check-inline">
34 <input class="form-check-input" type="radio" name="gender" value="male"> laki-laki
35 </div>
36 <div class="form-check form-check-inline">
37 <input class="form-check-input" type="radio" name="gender" value="female"> Perempuan
38 </div>
39 @error('gender') <div class="text-danger">{{ $message }}</div> @enderror
40 </div>
41 <div class="mb-3">
42 <label>Role</label>
43 <select name="role" class="form-select">
44 <option value="">-- Pilih Role --</option>
45 <option value="user">User</option>
46 <option value="admin">Admin</option>
47 </select>
48 @error('role') <div class="text-danger">{{ $message }}</div> @enderror
49 </div>
50 <div class="mb-3">
51 <label>Bio</label>
52 <textarea name="bio" class="form-control">{{ old('bio') }}</textarea>
53 @error('bio') <div class="text-danger">{{ $message }}</div> @enderror
54 </div>
55 <div class="form-check mb-3">
56 <input type="checkbox" id="confirm" name="confirm" class="form-check-input">
57 <label class="form-check-label" for="confirm">Saya konfirmasi informasi sudah benar</label>
58 @error('confirm') <div class="text-danger">{{ $message }}</div> @enderror
59 </div>
60 <button type="submit" class="btn btn-primary" id="submitBtn" disabled>Submit</button>
61 </form>
62
63 <script>
64 document.getElementById('confirm').addEventListener('change', function() {
65 document.getElementById('submitBtn').disabled = !this.checked;
66 });
67 </script>
68 </body>
69 </html>
70

```

Buat View Hasil Buat file view Blade baru untuk menampilkan hasil. Buat file bernama result.blade.php di dalam direktori resources/views dan isikan kode berikut:

```

projects > form-app > resources > views > result.blade.php > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Hasil Form</title>
5 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
6 </head>
7 <body class="container mt-5">
8 <h2>Data Form yang Dikirim</h2>
9 @if ($data)
10 <ul class="list-group">
11 @foreach ($data as $key => $value)
12 <li class="list-group-item"><strong>{{ ucfirst($key) }}:</strong> {{ $value }}</li>
13 @endforeach
14 </ul>
15 @else
16 <p>Tidak ada data tersedia.</p>
17 @endif
18 </body>
19 </html>

```

## 5. Jalankan Aplikasi: php artisan serve

The screenshot shows a web browser at the address 127.0.0.1:8000/form. The page displays a registration form with the following fields and controls:

- Nama**: Text input field.
- Email**: Text input field containing "zahranimaisha@gmail.com".
- Umur**: Text input field.
- Password**: Password input field with masked characters.
- Jenis Kelamin**: Radio buttons for "Laki-laki" and "Perempuan".
- Role**: Dropdown menu with the text "-- Pilih Role --".
- Bio**: Text input field.
- Confirmation**: A checkbox labeled "Saya konfirmasi informasi sudah benar".
- Submit**: A blue button to submit the form.

## 2.2 Praktikum 2 – Validasi Kustom dan Pesan Error di Laravel 12

1. Mendefinisikan Route Buka project praktikum 1 sebelumnya, kemudian tambahkan route baru berikut:

```

2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\FormController;
5 use App\Http\Controllers\RegisterController;
6
7 Route::get('/register', [RegisterController::class, 'showForm'])->name('register.show');
8 Route::post('/register', [RegisterController::class, 'handleForm'])->name('register.handle');

```

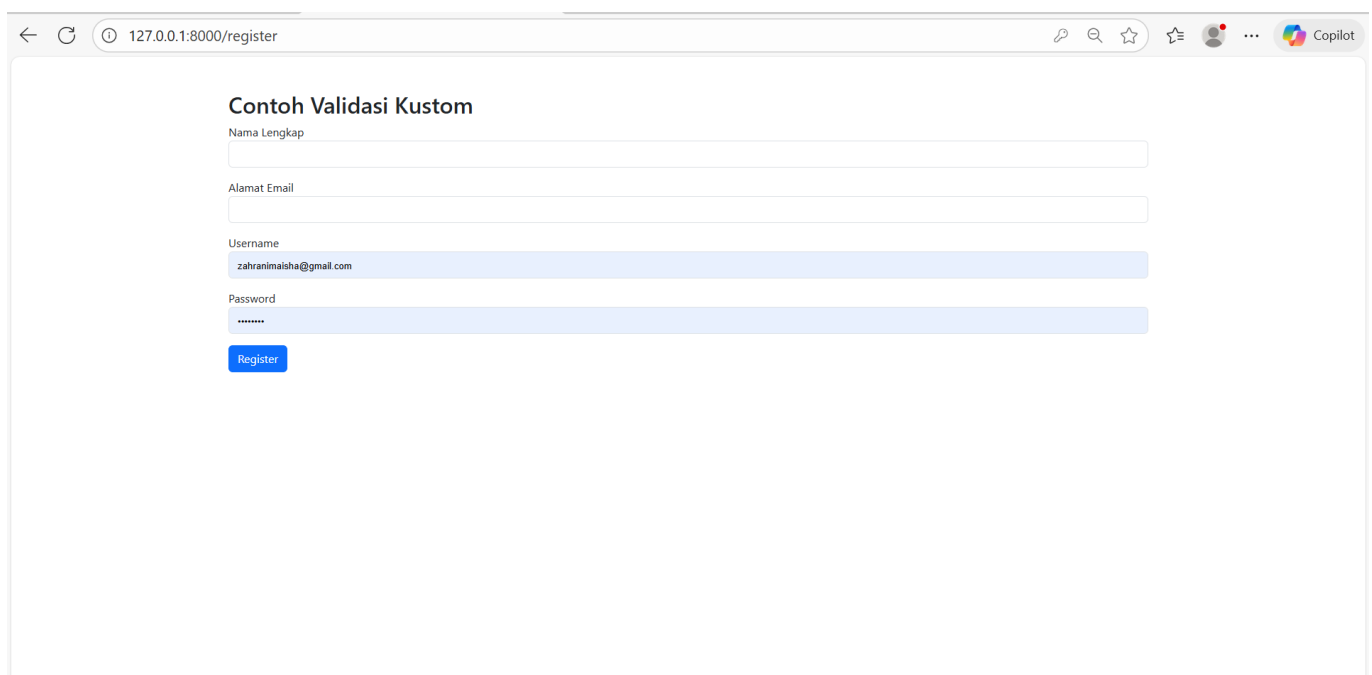
2. Buat sebuah Controller baru: php artisan make:controller RegisterController Lihat file controller baru di app/Http/Controllers/RegisterController.php. Buka file ini dan tambahkan metode berikut:

```
projects > form-app > app > Http > Controllers > RegisterController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class RegisterController extends Controller
8  {
9      public function showForm()
10     {
11         return view('register');
12     }
13
14     public function handleForm(Request $request)
15     {
16         $customMessages = [
17             'name.required' => 'Kami perlu tahu nama Anda!',
18             'email.required' => 'Email Anda penting bagi kami.',
19             'email.email' => 'Hmm... itu tidak terlihat seperti email yang valid.',
20             'password.required' => 'Jangan lupa untuk set password.',
21             'password.min' => 'Password harus minimal :min karakter.',
22             'username.regex' => 'Username hanya boleh berisi huruf dan angka.',
23         ];
24
25         $request->validate([
26             'name' => 'required|string|max:100',
27             'email' => 'required|email',
28             'username' => ['required', 'regex:/^[a-zA-Z0-9]+$/' ],
29             'password' => 'required|min:6',
30         ], $customMessages);
31
32         return redirect()->route('register.show')->with('success', 'Registrasi berhasil!');
33     }
34 }
35
```

3. Buat View Buat file view Blade baru untuk form. Buat file bernama register.blade.php di dalam direktori resources/views dan isikan kode berikut:

```
projects > form-app > resources > views > register.blade.php > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Form Register</title>
5   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
6 </head>
7 <body class="container mt-5">
8   <h2>Contoh Validasi Kustom</h2>
9
10  @if (session('success'))
11    <div class="alert alert-success">
12      {{ session('success') }}
13    </div>
14  @endif
15
16  <form method="POST" action="{{ route('register.handle') }}">
17    @csrf
18    <div class="mb-3">
19      <label for="name">Nama Lengkap</label>
20      <input name="name" class="form-control" value="{{ old('name') }}">
21      @error('name') <div class="text-danger">{{ $message }}</div> @enderror
22    </div>
23    <div class="mb-3">
24      <label for="email">Alamat Email</label>
25      <input name="email" class="form-control" value="{{ old('email') }}">
26      @error('email') <div class="text-danger">{{ $message }}</div> @enderror
27    </div>
28    <div class="mb-3">
29      <label for="username">Username</label>
30      <input name="username" class="form-control" value="{{ old('username') }}">
31      @error('username') <div class="text-danger">{{ $message }}</div> @enderror
32    </div>
33    <div class="mb-3">
34      <label for="password">Password</label>
35      @error('password') <div class="text-danger">{{ $message }}</div> @enderror
36      <input type="password" name="password" class="form-control">
37    </div>
38    <button type="submit" class="btn btn-primary">Register</button>
39  </form>
40 </body>
41 </html>
```

#### 4. Jalankan Aplikasi



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/register". The page title is "Contoh Validasi Kustom". The form contains the following fields and elements:

- Nama Lengkap**: A text input field.
- Alamat Email**: A text input field.
- Username**: A text input field containing the value "zahranihalsha@gmail.com".
- Password**: A password input field with masked characters (dots).
- Register**: A blue button to submit the form.

## 2.3 Praktikum 3 – Multi-Step Form Submission dengan Session Data

1. Buat dan Buka Proyek Laravel laravel new multistep-form-app cd multistep-form-app code .
2. Buat file layout di resources/views/layouts/app.blade.php:

```

1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>@yield('title', 'Multi-Step Form')</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8   <style>
9     .step-progress {
10       margin-bottom: 30px;
11     }
12     .step-item {
13       text-align: center;
14       position: relative;
15     }
16     .step-item .step-number {
17       width: 40px;
18       height: 40px;
19       border-radius: 50%;
20       background: #e9ecef;
21       display: flex;
22       align-items: center;
23       justify-content: center;
24       margin: 0 auto 10px;
25       font-weight: bold;
26     }
27     .step-item.active .step-number {
28       background: #007bff;
29       color: white;
30     }
31     .step-item.completed .step-number {
32       background: #28a745;
33       color: white;
34     }
35     .step-line {
36       position: absolute;
37       top: 20px;
38       left: 50%;
39       width: 100%;
40       height: 2px;
41       background: #e9ecef;
42       z-index: -1;
43     }
44     .form-step {
45       display: none;
46     }
47     .form-step.active {
48       display: block;
49     }
50     .card {
51       box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.15);
52     }
53   </style>
54 </head>
55 <body>
56   <nav class="navbar navbar-dark bg-dark">
57     <div class="container">
58       <span class="navbar-brand mb-0 h1">Multi-Step Form Demo</span>
59     </div>
60   </nav>
61
62   <div class="container mt-4">
63     @yield('content')
64   </div>
65
66   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
67 </body>
68 </html>

```

3. Buat Routes



```
1 <?php
2
3 use App\Http\Controllers\MultiStepFormController;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/multistep', [MultiStepFormController::class, 'showStep1'])->name('multistep.step1');
7 Route::post('/multistep/step1', [MultiStepFormController::class, 'storeStep1'])->name('multistep.storeStep1');
8 Route::get('/multistep/step2', [MultiStepFormController::class, 'showStep2'])->name('multistep.step2');
9 Route::post('/multistep/step2', [MultiStepFormController::class, 'storeStep2'])->name('multistep.storeStep2');
10 Route::get('/multistep/step3', [MultiStepFormController::class, 'showStep3'])->name('multistep.step3');
11 Route::post('/multistep/step3', [MultiStepFormController::class, 'storeStep3'])->name('multistep.storeStep3');
12 Route::get('/multistep/summary', [MultiStepFormController::class, 'showSummary'])->name('multistep.summary');
13 Route::post('/multistep/complete', [MultiStepFormController::class, 'complete'])->name('multistep.complete');
14
15 Route::get('/', function () {
16 |     return view('welcome');
17 | });
18
```

4. Buat controller untuk menangani logika multi-step form: php artisan make:controller MultiStepFormController Buka file app/Http/Controllers/MultiStepFormController.php dan tambahkan metode berikut:

```

project > multi-step-app > app > http > Controllers > MultiStepFormController.php > --
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class MultiStepFormController extends Controller
8  {
9      // Step 1 - Informasi Pribadi
10     public function showsteps()
11     {
12         return view('multistep.steps', [
13             'step' => 1,
14             'progress' => 0
15         ]);
16     }
17
18     public function storesteps(Request $request)
19     {
20         $validated = $request->validate([
21             'full_name' => 'required|string|max:100',
22             'email' => 'required|email',
23             'phone' => 'required|string|max:15',
24             'address' => 'required|string|max:200',
25         ]);
26
27         // Simpan data ke session
28         session(['steps_data' => $validated]);
29
30         return redirect()->route('multistep.steps');
31     }
32
33     // Step 2 - Informasi Pendidikan
34     public function showsteps()
35     {
36         if (session('steps_data')) {
37             return redirect()->route('multistep.steps');
38         }
39
40         return view('multistep.steps', [
41             'step' => 2,
42             'progress' => 25
43         ]);
44     }
45
46     public function storesteps(Request $request)
47     {
48         $validated = $request->validate([
49             'education' => 'required|string|max:50',
50             'institution' => 'required|string|max:100',
51             'graduation_year' => 'required|integer|min:1990|max:' . date('Y'),
52             'major' => 'required|string|max:100',
53         ]);
54
55         session(['steps_data' => $validated]);
56
57         return redirect()->route('multistep.steps');
58     }
59
60     // Step 3 - Pengalaman Kerja
61     public function showsteps()
62     {
63         if (session('steps_data') || !session('steps_data')) {
64             return redirect()->route('multistep.steps');
65         }
66
67         return view('multistep.steps', [
68             'step' => 3,
69             'progress' => 50
70         ]);
71     }
72
73     public function storesteps(Request $request)
74     {
75         $validated = $request->validate([
76             'current_job' => 'nullable|string|max:100',
77             'company' => 'nullable|string|max:100',
78             'experience_years' => 'required|integer|min:0|max:50',
79             'skills' => 'required|string|max:200',
80         ]);
81
82         session(['steps_data' => $validated]);
83
84         return redirect()->route('multistep.summary');
85     }
86
87     // Summary - Ringkasan Data
88     public function showsummary()
89     {
90         $step1data = session('steps_data');
91         $step2data = session('steps_data');
92         $step3data = session('steps_data');
93
94         if (!$step1data || !$step2data || !$step3data) {
95             return redirect()->route('multistep.steps');
96         }
97
98         return view('multistep.summary', [
99             'step' => 4,
100             'progress' => 100,
101             'step1data' => $step1data,
102             'step2data' => $step2data,
103             'step3data' => $step3data
104         ]);
105     }
106
107     // Complete - Proses Final
108     public function complete(Request $request)
109     {
110         // Di sini Anda bisa menyimpan data ke database
111         // Untuk demo, kita hanya akan menampilkan pesan sukses
112
113         $allData = [
114             'personal' => session('steps_data'),
115             'education' => session('steps_data'),
116             'experience' => session('steps_data')
117         ];
118
119         // hapus session data
120         $request->session()->forget(['steps_data', 'step1_data', 'step2_data']);
121
122         return view('multistep.complete', [
123             'data' => $allData
124         ]);
125     }
126 }

```

5. Buat Views untuk Setiap Step Buat direktori resources/views/multistep/ Step 1 - Informasi Pribadi: resources/views/multistep/step1.blade.php

```

@section('title', 'Step 1 - Informasi Pribadi')

@section('content')
<div class="row justify-content-center">
    <div class="col-md-8">
        <!-- Progress Bar -->
        <div class="step-progress">
            <div class="row">
                <div class="col-3 step-item active">
                    <div class="step-number">1</div>
                    <div>Informasi Pribadi</div>
                </div>
                <div class="col-3 step-item">
                    <div class="step-number">2</div>
                    <div>Pendidikan</div>
                </div>
                <div class="col-3 step-item">
                    <div class="step-number">3</div>
                    <div>Pengalaman</div>
                </div>
                <div class="col-3 step-item">
                    <div class="step-number">4</div>
                    <div>Ringkasan</div>
                </div>
            </div>
        </div>
        <div class="card">
            <div class="card-header">
                <h4 class="mb-0">Step 1: Informasi Pribadi</h4>
            </div>
            <div class="card-body">
                <form method="POST" action="{{ route('multistep.storeStep1') }}">
                    @csrf

                    <div class="row">
                        <div class="col-md-6 mb-3">
                            <label for="full_name" class="form-label">Nama Lengkap *</label>
                            <input type="text" class="form-control @error('full_name') is-invalid @enderror"
                                id="full_name" name="full_name"
                                value="{{ old('full_name', session('step1_data.full_name')) }}"
                                required>
                            @error('full_name')
                                <div class="invalid-feedback">{{ $message }}</div>
                            @enderror
                        </div>

                        <div class="col-md-6 mb-3">
                            <label for="email" class="form-label">Email *</label>
                            <input type="email" class="form-control @error('email') is-invalid @enderror"
                                id="email" name="email"
                                value="{{ old('email', session('step1_data.email')) }}"
                                required>
                            @error('email')
                                <div class="invalid-feedback">{{ $message }}</div>
                            @enderror
                        </div>
                    </div>

                    <div class="row">
                        <div class="col-md-6 mb-3">
                            <label for="phone" class="form-label">Nomor Telepon *</label>
                            <input type="text" class="form-control @error('phone') is-invalid @enderror"
                                id="phone" name="phone"
                                value="{{ old('phone', session('step1_data.phone')) }}"
                                required>
                            @error('phone')
                                <div class="invalid-feedback">{{ $message }}</div>
                            @enderror
                        </div>

                        <div class="col-md-6 mb-3">
                            <label for="address" class="form-label">Alamat *</label>
                            <input type="text" class="form-control @error('address') is-invalid @enderror"
                                id="address" name="address"
                                value="{{ old('address', session('step1_data.address')) }}"
                                required>
                            @error('address')
                                <div class="invalid-feedback">{{ $message }}</div>
                            @enderror
                        </div>
                    </div>

                    <div class="d-flex justify-content-between">
                        <div></div>
                        <div>
                            <button type="submit" class="btn btn-primary">
                                Lanjut ke Step 2 <i class="fas fa-arrow-right ms-2"></i>
                            </button>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>
@endsession

```

Step 2 - Informasi Pendidikan: resources/views/multistep/step2.blade.php

Step 3 - Pengalaman Kerja: [resources/views/multistep/step3.blade.php](#)

```

resources > multistep-form-app > resources > views > multistep > step3.blade.php > ...
1 @extends('layouts.app')
2
3 @section('title', 'Step 3 - Pengalaman Kerja')
4
5 @section('content')
6 <div class="row justify-content-center">
7   <div class="col-md-8">
8     <!-- Progress Bar -->
9     <div class="step-progress">
10       <div class="row">
11         <div class="col-3 step-item completed">
12           <div class="step-number">1</div>
13           Informasi Pribadi</div>
14         </div>
15         <div class="col-3 step-item completed">
16           <div class="step-number">2</div>
17           Pendidikan</div>
18         </div>
19         <div class="col-3 step-item active">
20           <div class="step-number">3</div>
21           Pengalaman</div>
22         </div>
23         <div class="col-3 step-item">
24           <div class="step-number">4</div>
25           Ringkasan</div>
26         </div>
27       </div>
28     </div>
29
30     <div class="card">
31       <div class="card-header">
32         <h4 class="mb-0">Step 3: Pengalaman Kerja</h4>
33       </div>
34       <div class="card-body">
35         <form method="POST" action="{{ route('multistep.storeStep3') }}">
36           @csrf
37
38           <div class="row">
39             <div class="col-md-6 mb-3">
40               <label for="current_job" class="form-label">Pekerjaan Saat Ini</label>
41               <input type="text" class="form-control @error('current_job') is-invalid @enderror"
42                 id="current_job" name="current_job"
43                 value="{{ old('current_job', session('step3_data.current_job')) }}">
44               @error('current_job')
45                 <div class="invalid-feedback">{{ $message }}</div>
46               @enderror
47             </div>
48
49             <div class="col-md-6 mb-3">
50               <label for="company" class="form-label">Perusahaan</label>
51               <input type="text" class="form-control @error('company') is-invalid @enderror"
52                 id="company" name="company"
53                 value="{{ old('company', session('step3_data.company')) }}">
54               @error('company')
55                 <div class="invalid-feedback">{{ $message }}</div>
56               @enderror
57             </div>
58           </div>
59
60           <div class="row">
61             <div class="col-md-6 mb-3">
62               <label for="experience_years" class="form-label">Pengalaman Kerja (Tahun) *</label>
63               <input type="number" class="form-control @error('experience_years') is-invalid @enderror"
64                 id="experience_years" name="experience_years"
65                 value="{{ old('experience_years', session('step3_data.experience_years')) }}"
66                 min="0" max="50" required>
67               @error('experience_years')
68                 <div class="invalid-feedback">{{ $message }}</div>
69               @enderror
70             </div>
71           </div>
72
73           <div class="mb-3">
74             <label for="skills" class="form-label">Keahlian/Keterampilan *</label>
75             <textarea class="form-control @error('skills') is-invalid @enderror"
76               id="skills" name="skills" rows="3"
77               placeholder="Contoh: PHP, Laravel, JavaScript, MySQL..."
78               required>{{ old('skills', session('step3_data.skills')) }}</textarea>
79             @error('skills')
80               <div class="invalid-feedback">{{ $message }}</div>
81             @enderror
82             <div class="form-text">Pisahkan setiap keahlian dengan koma</div>
83           </div>
84
85           <div class="d-flex justify-content-between">
86             <a href="{{ route('multistep.step2') }}" class="btn btn-secondary">
87               <i class="fas fa-arrow-left ms-2"></i> Kembali
88             </a>
89             <button type="submit" class="btn btn-primary">
90               Lihat Ringkasan <i class="fas fa-arrow-right ms-2"></i>
91             </button>
92           </div>
93         </form>
94       </div>
95     </div>
96   </div>
97 </div>

```

Summary - Ringkasan: resources/views/multistep/summary.blade.php

Complete - Selesai: [resources/views/multistep/complete.blade.php](#)

```

ects > multistep-form-app > resources > views > multistep > complete.blade.php > ...
@extends('layouts.app')

@section('title', 'Pendaftaran Selesai')

@section('content')


## Pendaftaran Berhasil!



Terima kasih telah melengkapi formulir pendaftaran. Data Anda telah berhasil disimpan.



###### fas fa-envelope me-2



- Anda akan menerima email konfirmasi dalam 24 jam
- Proses verifikasi membutuhkan waktu 2-3 hari kerja
- Tim kami akan menghubungi Anda untuk langkah selanjutnya



fas fa-plus me-2
Daftar Lagi


fas fa-home me-2
Kembali ke Home



###### Debug Data (bisa dihapus di production)



```
{{ json_encode($data, JSON_PRETTY_PRINT) }}
```


```

5. Jalankan Aplikasi: php artisan serve

←↻📄127.0.0.1:8000/multistep

🔍🌟👤⋮🎨Copilot

Multi-Step Form Demo

1

Informasi Pribadi

2

Pendidikan

3

Pengalaman

4

Ringkasan

Step 1: Informasi Pribadi

Nama Lengkap \*

maisha zahrani

Email \*

zahrani.maisha@gmail.com

Nomor Telepon \*

081361925792

Alamat \*

Lhokseumawe

Lanjut ke Step 2

←↻📄127.0.0.1:8000/multistep

🔍🌟👤⋮🎨Copilot

Multi-Step Form Demo

✓1

Informasi Pribadi

2

Pendidikan

3

Pengalaman

4

Ringkasan

Step 2: Informasi Pendidikan

Tingkat Pendidikan \*

Pilih Tingkat Pendidikan

Nama Institusi \*

Politeknik Negeri Lhokseumawe

Tahun Lulus \*

2025

Jurusan \*

TIK

Kembali

Lanjut ke Step 3

←↻📄127.0.0.1:8000/multistep/step3

🔍🌟👤⋮🎨Copilot

Multi-Step Form Demo

✓1

Informasi Pribadi

✓2

Pendidikan

3

Pengalaman

4

Ringkasan

Step 3: Pengalaman Kerja

Pekerjaan Saat Ini

Senior Data Analyst

Perusahaan

Microsoft

Pengalaman Kerja (Tahun) \*

6

Keahlian/Keterampilan \*

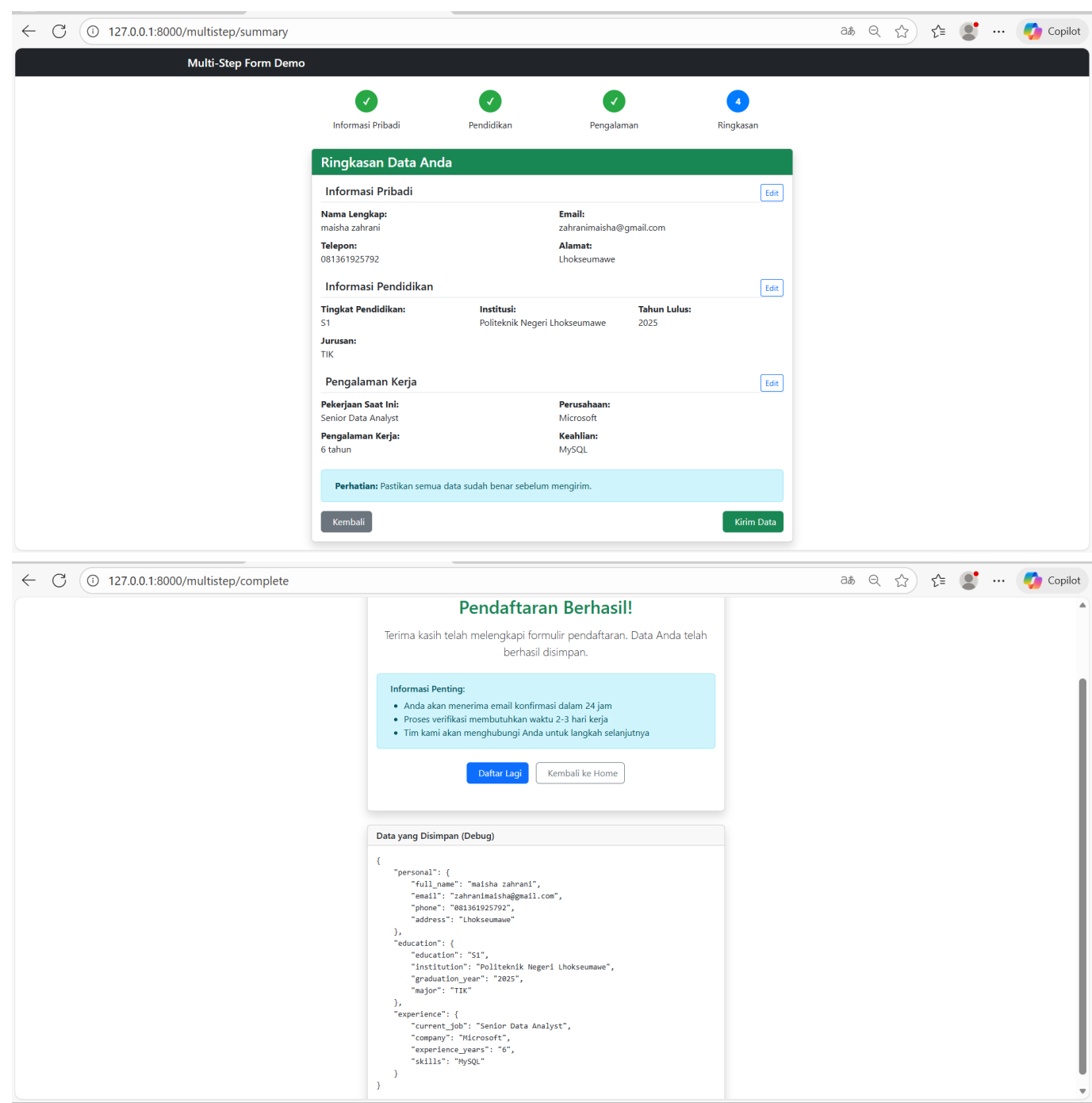
MySQL

Pisahkan setiap keahlian dengan koma

Kembali

Lihat Ringkasan





### 3. Hasil dan Pembahasan

Pada percobaan pertama, form bisa menampilkan data yang dikirim pengguna lewat controller dan view sesuai alur kerja Laravel.

Pada percobaan kedua, validasi berhasil mengecek data dengan benar dan menampilkan pesan error saat ada input yang salah. Data yang sudah diketik juga tetap muncul setelah validasi gagal, jadi pengguna tidak perlu mengisi ulang.

Pada percobaan ketiga, form bertahap (multi-step) bisa dijalankan dengan lancar. Data dari setiap langkah tersimpan di session dan ditampilkan lagi di bagian ringkasan sebelum dikirim.

### 4. Kesimpulan

Laravel memiliki sistem form submission dan data validation yang lengkap, aman, dan mudah digunakan. Melalui konsep MVC, Laravel mampu memisahkan proses input, validasi, dan penyimpanan data dengan rapi. Fitur seperti CSRF Protection, error handling, dan custom validation menjadikan Laravel sangat efisien untuk membangun aplikasi web yang aman dan terstruktur. Dengan memanfaatkan fitur ini, developer dapat memastikan setiap data yang masuk ke aplikasi sudah tervalidasi dengan baik dan bebas dari kesalahan.

---

## 5. Referensi

1. <https://hackmd.io/@mohdrzu/r1AIUzWpII>
  2. <https://chatgpt.com>
-