

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT
CRUD LARAVEL-SEDERHANA**



Disusun Oleh :

Qusnul Diah Mawanti

2341760035

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2025



Jurusan Teknologi Informasi – Politeknik Negeri
Malang
Jobsheet 1 : Membuat CRUD Laravel Sederhana NO CSS
menggunakan Laravel dengan Eloquent ORM
Mata Kuliah : Pemrograman Web Lanjut
Kelas : 2A – SIB
Nama : Qusnul Diah Mawanti
No absen : 25
NIM : 2341760035

Februari 2025

1. Langkah-Langkah Instalasi

- Pastikan **Composer** telah terinstal pada sistem dengan menjalankan perintah composer pada **command prompt**.

```
Command Prompt
C:\Users\LENOVO>composer

Composer version 2.8.5 2025-01-21 15:23:40
```

Jika belum, unduh dan instal dari getcomposer.org.

Download Composer Latest: v2.8.5

Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run **Composer-Setup.exe** - it will install the latest composer version whenever it is executed.

- Pastikan **PHP versi 8.1** atau lebih sudah tersedia. Jika menggunakan **Laragon**, pastikan sudah dikonfigurasi dengan benar.

```
Command Prompt
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>php -v
PHP 8.2.12 (cli) (built: Oct 24 2023 21:15:15) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.2.12, Copyright (c) Zend Technologies
```

- Buka terminal atau command prompt, lalu jalankan perintah berikut untuk menginstal Laravel versi 10:

After you have installed PHP and Composer, you may create a new Laravel project via Composer's `create-project` command:

```
composer create-project "laravel/laravel:^10.0" example-app
```

- Gantilah nama_proyek dengan nama yang Anda inginkan untuk aplikasi Laravel Anda.

```
C:\laragon\www>composer create-project "laravel/laravel:^10.0" laravel
Creating a "laravel/laravel:^10.0" project at "./laravel"
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\laragon\www\laravel
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.3)
- Locking carbonphp/carbon-doctrine-types (2.1.0)

No security vulnerability advisories found.
> @php artisan key:generate --ansi

INFO Application key set successfully.
```

- Setelah instalasi selesai, masuk ke direktori proyek dengan perintah:

```
C:\laragon\www>cd laravel
```

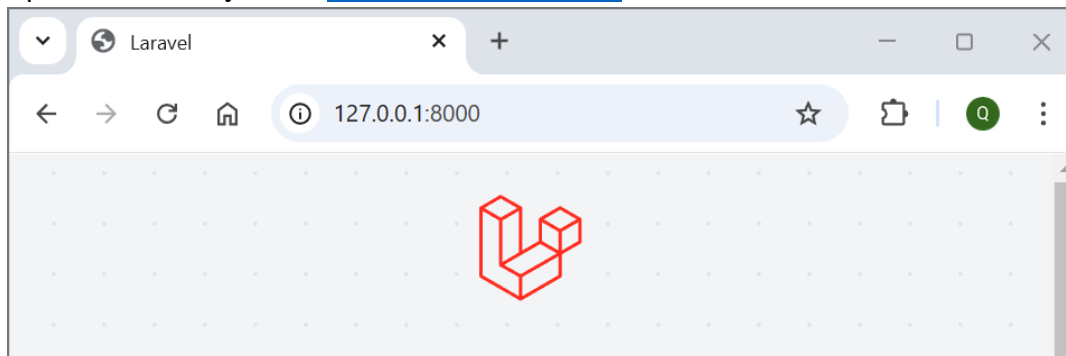
- Gunakan perintah berikut untuk menjalankan server pengembangan lokal Laravel:

```
C:\laragon\www\laravel>php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

- Aplikasi akan berjalan di <http://127.0.0.1:8000/>.



2. Persiapan Proyek CRUD

- Buat proyek Laravel dengan nama **crud-app** menggunakan perintah:

```
C:\laragon\www>composer create-project "laravel/laravel:^10.0" crud-app
Creating a "laravel/laravel:^10.0" project at "./crud-app"
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\laragon\www\crud-app
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.3)
- Locking carbonphp/carbon-doctrine-types (2.1.0)
```

- Masuk ke folder proyek:

```
C:\laragon\www>cd crud-app
```

- Jalankan server lokal Laravel:

```
C:\laragon\www\crud-app>php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

3. Buat Database

- Buat Database: Buat database baru di MySQL, misalnya **crud_db**.

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0039 seconds.)

```
CREATE DATABASE crud_db;
```

[Edit inline] [Edit] [Create PHP code]

- Buka file **.env** di root proyek Laravel dan atur konfigurasi database:

⚙️ .env	14 DB_DATABASE=crud_db
💰 .env.example	15 DB_USERNAME=root
🔑 .gitattributes	16 DB_PASSWORD=
	17

4. Buat Model Dan Migrasi

- Jalankan perintah berikut untuk membuat model **Item** beserta file migrasinya:

```
C:\laragon\www\crud-app>php artisan make:model Item -m

INFO Model [C:\laragon\www\crud-app\app\Models\Item.php] created successfully.

INFO Migration [C:\laragon\www\crud-app\database\Migrations\2025_03_01_024101_create_items_table.php] created successfully.
```

- Edit Migrasi: Buka file migrasi yang ada di **database/migrations/xxxx_xx_xx_create_items_table.php**

```
🐘 2025_02_20_081655_create_items_table.php
```

- Tambahkan kolom yang dibutuhkan dalam metode **up()**:

```
public function up(): void
{
    Schema::create('items', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->text('description');
        $table->timestamps();
    });
}
```

- Gunakan perintah berikut untuk menjalankan migrasi:

```
C:\laragon\www\crud-app>php artisan migrate

INFO Preparing database.

Creating migration table ..... 43ms DONE

INFO Running migrations.

2014_10_12_000000_create_users_table ..... 63ms DONE
2014_10_12_100000_create_password_reset_tokens_table 17ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 36ms DONE
2019_12_14_000001_create_personal_access_tokens_table 68ms DONE
2025_03_01_024101_create_items_table ..... 19ms DONE
```

5. Buat Controller

- Buat **ItemController** dengan perintah berikut:

```
C:\laragon\www\crud-app>php artisan make:controller ItemController --resource

INFO Controller [C:\laragon\www\crud-app\app\Http\Controllers\ItemController.php] created successfully.
```

- Implementasikan CRUD di Controller: Buka **app/Http/Controllers/ItemController.php** dan tambahkan kode berikut:

```
app > Http > Controllers > ItemController.php

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Item; // Mengimpor model Item agar dapat digunakan dalam controller
6  use Illuminate\Http\Request;
7
8  class ItemController extends Controller
9  {
10     // Mengambil semua data dari tabel items
11     public function index()
12     {
13         $items = Item::all();
14         return view('items.index', compact('items'));
15     }
16
17     // Mengembalikan tampilan halaman tambah item
18     public function create()
19     {
20         return view('items.create');
21     }
22 }
```

```

23 // Memvalidasi input agar name dan description wajib diisi
24 public function store(Request $request)
25 {
26     $request->validate([
27         'name' => 'required',
28         'description' => 'required',
29     ]);
30
31     //Item::create($request->all());
32     //return redirect()->route('items.index');
33
34     // Menyimpan data item baru dengan hanya atribut yang diizinkan
35     Item::create($request->only(['name', 'description']));
36     return redirect()->route('items.index')->with('succes', 'Item added successfully');
37 }
38
39 // Menampilkan halaman detail item
40 public function show(Item $item)
41 {
42     return view('items.show', compact('item'));
43 }
44
45 // Menampilkan halaman formulir untuk mengedit item yang dipilih
46 public function edit(Item $item)
47 {
48     return view('items.edit', compact('item'));
49 }
50
51 // Memperbarui item yang dipilih dengan data baru dari formulir
52 public function update(Request $request, Item $item)
53 {
54     $request->validate([
55         'name' => 'required',
56         'description' => 'required',
57     ]);
58
59     //$item->update($request->all());
60     //return redirect()->route('items.index');
61     // Hanya masukkan atribut yang diizinkan
62     $item->update($request->only(['name', 'description']));
63     return redirect()->route('items.index')->with('success', 'Item updated successfully.');
```

```

64 }
65
66 // Menghapus item yang dipilih dari database
67 public function destroy(Item $item)
68 {
69
70     // return redirect()->route('items.index');
71     $item->delete();
72     return redirect()->route('items.index')->with('success', 'Item deleted successfully.');
```

```

73 }
74 }
```

6. Buat Routing

- Buka file **routes/web.php**, lalu tambahkan rute untuk mengatur akses ke controller:

```

routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\ItemController; // Mengimpor ItemController agar dapat digunakan dalam routing
5  /*
6  |-----
7  | Web Routes
8  |-----
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider and all of them will
12 | be assigned to the "web" middleware group. Make something great!
13 |
14 */
15
16 Route::get('/', function () {
17     return view('welcome'); // Menampilkan halaman utama (welcome)
18 });
19
20 Route::resource('items', ItemController::class); // Membuat rute resource untuk CRUD item secara otomatis

```

7. Buat View

- Buat Folder Views: Buat folder **items** di **resources/views**.
- Buat File View:

i. index.blade.php

```

resources > views > items > index.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Item List</title>
5  </head>
6  <body>
7  |   <h1>Items</h1>
8
9  |   <!-- Menampilkan pesan sukses jika ada -->
10 |   @if(session('success'))
11 |       <p>{{ session('success') }}</p>
12 |   @endif
13
14 |   <!-- Link untuk menambahkan item baru -->
15 |   <a href="{{ route('items.create') }}">Add Item</a>
16 |   <ul>
17 |       <!-- Looping untuk menampilkan semua item dalam daftar -->
18 |       @foreach ($items as $item)
19 |           <li>
20 |               {{ $item->name }} -
21 |               <!-- Link untuk mengedit item -->
22 |               <a href="{{ route('items.edit', $item) }}">Edit</a>
23 |               <!-- Formulir untuk menghapus item -->
24 |               <form action="{{ route('items.destroy', $item) }}" method="POST" style="display:inline;">
25 |                   @csrf
26 |                   @method('DELETE')
27 |                   <button type="submit">Delete</button> <!-- Menggunakan metode DELETE untuk penghapusan -->
28 |               </form>
29 |           </li>
30 |       @endforeach
31 |   </ul>
32 </body>
33 </html>

```

ii. create.blade.php

```
resources > views > items > create.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Add Item</title>
5  </head>
6  <body>
7      <h1>Add Item</h1>
8      <!-- Form untuk menambahkan item baru -->
9      <form action="{{ route('items.store') }}" method="POST">
10         @csrf
11         <!-- Input untuk nama item -->
12         <label for="name">Name:</label>
13         <input type="text" name="name" required>
14         <br>
15         <!-- Input untuk deskripsi item -->
16         <label for="description">Description:</label>
17         <textarea name="description" required></textarea>
18         <br>
19         <!-- Tombol untuk mengirim form -->
20         <button type="submit">Add Item</button>
21     </form>
22     <!-- Link untuk kembali ke daftar item -->
23     <a href="{{ route('items.index') }}">Back to List</a>
24 </body>
25 </html>
```

iii. edit.blade.php

```
resources > views > items > edit.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Edit Item</title>
5  </head>
6  <body>
7      <h1>Edit Item</h1>
8      <!-- Form untuk mengedit item yang sudah ada -->
9      <form action="{{ route('items.update', $item) }}" method="POST">
10         @csrf
11         @method('PUT')
12         <!-- Input untuk nama item dengan nilai default dari data item -->
13         <label for="name">Name:</label>
14         <input type="text" name="name" value="{{ $item->name }}" required>
15         <br>
16         <!-- Input untuk deskripsi item dengan nilai default dari data item -->
17         <label for="description">Description:</label>
18         <textarea name="description" required>{{ $item->description }}</textarea>
19         <br>
20         <!-- mengirim form pembaruan -->
21         <button type="submit">Update Item</button>
22     </form>
23     <!-- Link untuk kembali ke daftar item -->
24     <a href="{{ route('items.index') }}">Back to List</a>
25 </body>
26 </html>
```


iv. show.blade.php

```
resources > views > items > show.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Item List</title>
5  </head>
6  <body>
7  |   <h1>Items</h1>
8  |   @if(session('success'))
9  |   |   <p>{{ session('success') }}</p>
10 |   @endif
11 |   <!-- Untuk menambahkan item baru -->
12 |   <a href="{{ route('items.create') }}">Add Item</a>
13 |   <ul>
14 |   |   <!-- Melakukan looping untuk menampilkan daftar item -->
15 |   |   @foreach ($items as $item)
16 |   |   |   <li>
17 |   |   |   |   <!-- Menampilkan nama item -->
18 |   |   |   |   {{ $item->name }} -
19 |   |   |   |   <!-- Link untuk mengedit item -->
20 |   |   |   |   <a href="{{ route('items.edit', $item) }}">Edit</a>
21 |   |   |   |   <!-- Form untuk menghapus item -->
22 |   |   |   |   <form action="{{ route('items.destroy', $item) }}" method="POST" style="display:inline;">
23 |   |   |   |   |   @csrf
24 |   |   |   |   |   @method('DELETE')
25 |   |   |   |   |   <button type="submit">Delete</button>
26 |   |   |   |   </form>
27 |   |   |   </li>
28 |   |   @endforeach
29 |   </ul>
30 </body>
31 </html>
```

8. Done

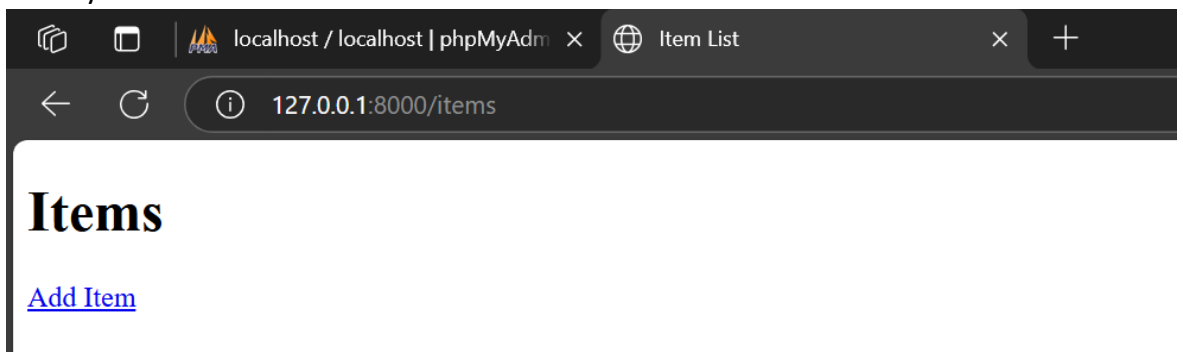
- Jalankan

```
C:\laragon\www\crud-app>php artisan serve

INFO  Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

- Hasilnya



localhost / localhost | phpMyAdmin

Add Item

127.0.0.1:8000/items/create

Add Item

Name:

Description:

[Back to List](#)

localhost / localhost | phpMyAdmin

Item List

127.0.0.1:8000/items

Items

Item added successfully.

[Add Item](#)

- RG_Multimedia - [Edit](#)