

**Membuat CRUD Laravel Sederhana NO CSS
menggunakan Laravel dengan Eloquent ORM**

1. Langkah-Langkah Instalasi

- Install Composer
- Pastikan PHP versi 8.1 sudah terinstall atau lebih (Laragon)
- buka terminal dan install laravel (versi 10)

```
bash
composer create-project --prefer-dist laravel/laravel nama_proyek "10.*"
```

- Gantilah nama_proyek dengan nama yang Anda inginkan untuk aplikasi Laravel Anda.
- Masuk ke Direktori Proyek: Setelah instalasi selesai, masuk ke direktori proyek Anda:

```
bash
cd nama_proyek
```

- Gantilah nama_proyek dengan nama yang Anda inginkan untuk aplikasi Laravel Anda. Masuk ke Direktori Proyek: Setelah instalasi selesai, masuk ke direktori proyek Anda:

```
bash
php artisan serve
```

2. Persiapan Proyek CRUD

- Buatlah proyek laravel dengan nama crud-app

```
bash
composer create-project --prefer-dist laravel/laravel crud-app
cd crud-app
```

- Jalankan server

```
bash
php artisan serve
```

3. Buat Database

- Buat Database: Buat database baru di MySQL, misalnya crud_db
- Konfigurasi Database: Edit file .env untuk mengatur koneksi database

```
env
DB_DATABASE=crud_db
DB_USERNAME=root
DB_PASSWORD=
```

4. Buat Model dan Migrasi

- Buat Model dan Migrasi

```
bash
php artisan make:model Item -m
```

- Edit Migrasi: Buka file migrasi di Database/migrations/xxxx_xx_xx_create_items_table.php dan tambahkan kolom:

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

- Jalankan Migrasi

```
bash
php artisan migrate
```

5. Buat Controller

- Buat Controller

```
bash
php artisan make:controller ItemController --resource
```

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

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Item;
6 use Illuminate\Http\Request;
7
8 class ItemController extends Controller
9 {
10     public function index()
11     {
12         $items = Item::all();
13         return view('items.index', compact('items'));
14     }
15
16     public function create()
17     {
18         return view('items.create');
19     }
20
21     public function store(Request $request)
22     {
23         $request->validate([
24             'name' => 'required',
25             'description' => 'required',
26         ]);
27
28         //Item::create($request->all());
29         //return redirect()->route('items.index');
30
31         // Hanya masukkan atribut yang diizinkan
32         Item::create($request->only(['name', 'description']));
33         return redirect()->route('items.index')->with('success', 'Item added successfully.');
```

```

35
36 public function show(Item $item)
37 {
38     return view('items.show', compact('item'));
39 }
40
41 public function edit(Item $item)
42 {
43     return view('items.edit', compact('item'));
44 }
45
46 public function update(Request $request, Item $item)
47 {
48     $request->validate([
49         'name' => 'required',
50         'description' => 'required',
51     ]);
52
53     //$item->update($request->all());
54     //return redirect()->route('items.index');
55     // Hanya masukkan atribut yang diizinkan
56     $item->update($request->only(['name', 'description']));
57     return redirect()->route('items.index')->with('success', 'Item updated successfully.');
```

```

58 }
59
60 public function destroy(Item $item)
61 {
62     // return redirect()->route('items.index');
```

```

63
64     $item->delete();
65     return redirect()->route('items.index')->with('success', 'Item deleted successfully.');
```

```

66 }
67 }
68

```

6. Buat Routing

- Edit File Routing: Buka routes/web.php dan tambahkan

```

crud-app > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\ItemController;
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');
18 });
19
20 Route::resource('items', ItemController::class);

```

7. Buat View

- Buat Folder Views: Buat folder items di resources/views.
- Buat File View:
 - I. Index.blade.php

```
crud-app > resources > views > items > index.blade.php > html
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  |   <a href="{{ route('items.create') }}">Add Item</a>
12  |   <ul>
13  |   |   @foreach ($items as $item)
14  |   |   |   <li>
15  |   |   |   |   {{ $item->name }} -
16  |   |   |   |   <a href="{{ route('items.edit', $item) }}">Edit</a>
17  |   |   |   |   <form action="{{ route('items.destroy', $item) }}" method="POST" style="display:inline;">
18  |   |   |   |   |   @csrf
19  |   |   |   |   |   @method('DELETE')
20  |   |   |   |   |   <button type="submit">Delete</button>
21  |   |   |   |   </form>
22  |   |   |   </li>
23  |   |   @endforeach
24  |   </ul>
25  </body>
26  </html>
```

II. create.blade.php

```
crud-app > resources > views > items > create.blade.php > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Add Item</title>
5  </head>
6  <body>
7  |   <h1>Add Item</h1>
8  |   <form action="{{ route('items.store') }}" method="POST">
9  |   |   @csrf
10  |   |   <label for="name">Name:</label>
11  |   |   <input type="text" name="name" required>
12  |   |   <br>
13  |   |   <label for="description">Description:</label>
14  |   |   <textarea name="description" required></textarea>
15  |   |   <br>
16  |   |   <button type="submit">Add Item</button>
17  |   </form>
18  |   <a href="{{ route('items.index') }}">Back to List</a>
19  </body>
20  </html>
```

III. edit.blade.php

```
crud-app > resources > views > items > edit.blade.php > html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 | <title>Edit Item</title>
5 </head>
6 <body>
7 | <h1>Edit Item</h1>
8 | <form action="{{ route('items.update', $item) }}" method="POST">
9 | | @csrf
10 | | @method('PUT')
11 | | <label for="name">Name:</label>
12 | | <input type="text" name="name" value="{{ $item->name }}" required>
13 | | <br>
14 | | <label for="description">Description:</label>
15 | | <textarea name="description" required>{{ $item->description }}</textarea>
16 | | <br>
17 | | <button type="submit">Update Item</button>
18 | </form>
19 | <a href="{{ route('items.index') }}">Back to List</a>
20 </body>
21 </html>
```

IV. show.blade.php

```
crud-app > resources > views > items > index.blade.php > html
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 | <a href="{{ route('items.create') }}">Add Item</a>
12 | <ul>
13 | | @foreach ($items as $item)
14 | | | <li>
15 | | | | {{ $item->name }} -
16 | | | | <a href="{{ route('items.edit', $item) }}">Edit</a>
17 | | | | <form action="{{ route('items.destroy', $item) }}" method="POST" style="display:inline;">
18 | | | | | @csrf
19 | | | | | @method('DELETE')
20 | | | | | <button type="submit">Delete</button>
21 | | | | </form>
22 | | | </li>
23 | | @endforeach
24 | </ul>
25 </body>
26 </html>
```

8. Done

- Jalankan

```
PS C:\laragon\www\PWL2025\crud-app> php artisan serve

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

Press Ctrl+C to stop the server
```

- Hasilnya



Items

[Add Item](#)

- RG_Multimedia - [Edit](#)