

LAPORAN PRAKTIKUM
MATA KULIAH PEMROGRAMAN WEB LANJUT

Dosen Pengampu : Dimas Wahyu Wibowo, ST., MT.

JOBSHEET 2: MEMBUAT CRUD LARAVEL SEDERHANA NO CSS
MENGGUNAKAN LARAVEL DENGAN ELOQUENT ORM



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

INSTALASI

1. Install Composer

```
C:\Windows\System32>composer -v

Composer version 2.8.3 2024-11-17 13:13:04

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list
                             command
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi|--no-ansi         Force (or disable --no-ansi) ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  --no-scripts             Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache               Prevent use of the cache
  -v|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
                             3 for debug
```

2. Pastikan PHP versi 8.1 sudah terinstall atau lebih (Laragon)

```
C:\Windows\System32>php -v
PHP 8.3.14 (cli) (built: Nov 19 2024 15:53:22) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.3.14, Copyright (c) Zend Technologies
```

3. Buka terminal dan install laravel (versi 10)

```
C:\Windows\System32>composer create-project --prefer-dist/laravel/laravel PWL_2025 "10.*"
```

4. Jalankan Server Lokal: Anda dapat menjalankan server pengembangan lokal dengan perintah:

```
PS D:\Apps\laragon\www\PWL_2025> php artisan serve

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

PERSIAPAN PROYEK CRUDs

1. Buatlah proyek Laravel dengan nama crud-app

```
C:\Windows\System32>composer create-project --prefer-dist laravel/laravel crud-app
Creating a "laravel/laravel" project at "./crud-app"
Installing laravel/laravel (v11.6.1)
- Downloading laravel/laravel (v11.6.1)
- Installing laravel/laravel (v11.6.1): Extracting archive
Created project in C:\Windows\System32\crud-app
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 110 installs, 0 updates, 0 removals
- Locking brick/math (0.12.1)
- Locking carbonphp/carbon-doctrine-types (3.2.0)
- Locking dflydev/dot-access-data (v3.0.3)
```

2. Buat Database

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

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=crud_db
DB_USERNAME=root
DB_PASSWORD=
```

Penjelasan: File .env menyimpan konfigurasi environment seperti detail database, pengaturan cache, dan kredensial lainnya. DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, dan DB_PASSWORD mendefinisikan koneksi database, menggunakan mysql dan nama databasenya yaitu crud_db

3. Buat Model dan Migrasi

```
PS D:\Apps\laragon\www\crud-app> php artisan make:model Item-m
```

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

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

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

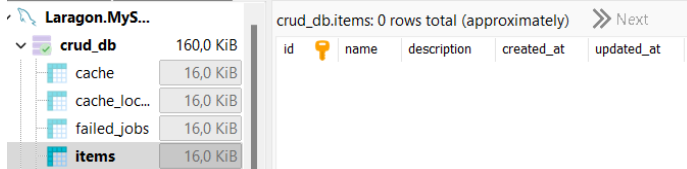
Penjelasan: Migration ini akan membuat tabel items dengan kolom id, name, description, serta menambahkan kolom timestamps.

- Jalankan Migrasi

```
PS D:\Apps\laragon\www\crud-app> php artisan migrate
```

INFO Running migrations.

2025_02_20_074638_create_items_table 33.95ms DONE



The screenshot shows the Laragon MySQL interface. On the left, a tree view shows the 'crud_db' database selected, with a list of tables: 'cache', 'cache_loc...', 'failed_jobs', and 'items'. The 'items' table is highlighted. On the right, a table structure view for 'crud_db.items' is shown, indicating 0 rows total. The table has columns: 'id' (primary key), 'name', 'description', 'created_at', and 'updated_at'.

4. Buat Controller

```
PS D:\Apps\laragon\www\crud-app> php artisan make:controller ItemController --resource
```

INFO Controller [D:\Apps\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:

```

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     // digunakan untuk menampilkan data item yang ada di database
16
17     public function create()
18     {
19         return view('items.create');
20     }
21     // digunakan untuk menampilkan form untuk menambahkan item baru
22
23     public function store(Request $request)
24     {
25         $request->validate([
26             'name' => 'required',
27             'description' => 'required',
28         ]);
29
30         // Item::create($request->all());
31         // return redirect()->route('items.index')
32
33         // Hanya masukkan atribut yang diizinkan
34         Item::create($request->only(['name', 'description']));
35         return redirect()->route('items.index');
36     }
37     // digunakan untuk menyimpan data item baru ke database
38
39     public function show(Item $items)
40     {
41         return view('items.show', compact('items'));
42     }
43     // digunakan untuk menampilkan detail item
44
45     public function edit(Item $item)
46     {
47         return view('items.edit', compact('item'));
48     }
49     // digunakan untuk menampilkan form untuk mengedit item
50
51     public function update(Request $request, Item $item)
52     {
53         $request->validate([
54             'name' => 'required',
55             'description' => 'required',
56         ]);
57
58         // $item->update($request->all());
59         // return redirect()->route('items.index');
60
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     // digunakan untuk menyimpan data item yang telah diubah ke database
66
67     public function destroy(Item $item)
68     {
69         // return redirect()->route('items.index');
70         $item->delete();
71         return redirect()->route('items.index')->with('success', 'Item deleted successfully');
72     }
73     // digunakan untuk menghapus item dari database
74 }
75

```

5. Buat Routing

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

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\ItemController;

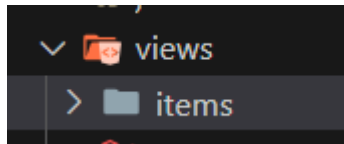
Route::get(uri: '/', action: function (): View {
    return view(view: 'welcome');
});
// digunakan untuk menampilkan halaman welcome

Route::resource(name: 'items', controller: ItemController::class);
// digunakan untuk membuat route resource untuk item
```

Penjelasan: File web.php ini mendefinisikan rute Laravel. Baris `Route::resource('items', ItemController::class)` secara otomatis membuat rute CRUD (create, read, update, delete) untuk model Item yang di-handle oleh ItemController.

6. Buat View

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



- Buat File View:
 - a. Index.blade.php

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <meta http-equiv="X-UA-Compatible" content="ie=edge">
8     <title>Item List</title>
9 </head>
10
11 <body>
12     <h1>Items</h1>
13     @if(session('success'))
14         <p>{{ session('success') }}</p>
15     @endif
16     <a href="{{route('items.create')}}">Add Item</a>
17     <ul>
18         @foreach($items as $item)
19             <li>
20                 {{ $item->name }}
21                 <a href="{{route('items.edit', $item->id)}}">Edit</a>
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
32 </html>

```

Penjelasan: Halaman ini menampilkan daftar item beserta tautan “Add Item” untuk menambahkan item baru. Setiap item memiliki link “Edit” dan tombol “Delete” untuk pengelolaan data. Jika ada pesan sukses (session('success')), pesannya akan ditampilkan di atas daftar item.

b. Create.blade.php

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Add Item</title>
8 </head>
9 <body>
10  <h1>Add Item</h1>
11  <form action="{{route('items.store')}}" method="POST">
12    @csrf
13    <label for="name">Name:</label>
14    <input type="text" name="name" id="name" required>
15    <br>
16    <label for="description">Description:</label>
17    <textarea name="description" id="description" required></textarea>
18    <br>
19    <button type="submit">Add Item</button>
20  </form>
21  <a href="{{route('items.index')}}">Back to List </a>
22 </body>
23 </html>

```

Penjelasan: Halaman ini menampilkan form untuk menambah item baru. Data yang dimasukkan (nama dan deskripsi) akan dikirim ke route items.store menggunakan metode POST setelah pengguna menekan tombol “Add Item”.

c. Edit.blade.php

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Edit Item</title>
8 </head>
9 <body>
10  <h1>Edit Item</h1>
11  <form action="{{route('items.update', $item)}}" method="POST">
12    @csrf
13    @method('PUT')
14    <label for="name">Name:</label>
15    <input type="text" name="name" id="name" value="{{ $item->name }}" required>
16    <br>
17    <label for="description">Description:</label>
18    <textarea name="description" required>{{ $item->description }}</textarea>
19    <br>
20    <button type="submit">Update Item</button>
21  </form>
22  <a href="{{route('items.index')}}">Back to List</a>
23 </body>
24 </html>

```


Penjelasan: Halaman ini menampilkan form untuk mengedit item yang sudah ada. Data item yang ada diisi ke dalam form. Setelah pengguna mengubah data dan menekan tombol “Update Item”, data dikirim ke route `items.update` menggunakan metode PUT untuk memperbarui item.

d. Show.blade.php

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Item List</title>
8 </head>
9 <body>
10  <h1>Items</h1>
11  @if(session('success'))
12    <p>{{ session('success') }}</p>
13  @endif
14  <a href="{{route('items.create')}}">Add Item</a>
15  <ul>
16    @foreach($items as $item)
17      <li>
18        {{ $item->name }}
19        <a href="{{route('items.edit', $item)}}">Edit</a>
20        <form action="{{route('items.destroy', $item)}}" method="POST" style="display: inline;">
21          @csrf
22          @method('DELETE')
23          <button type="submit">Delete</button>
24        </form>
25      </li>
26    @endforeach
27  </body>
28 </html>
```

Penjelasan: Halaman ini menampilkan detail dari satu item. Jika ada pesan sukses (`session('success')`), pesannya ditampilkan. Terdapat tautan untuk menambah item baru dan daftar item dengan opsi untuk mengedit atau menghapus setiap item.

7. Done

Add Item

Name:

Description:

[Back to List](#)

Items

[Add Item](#)

- warung_makan [Edit](#)