**LAPORAN TUGAS BESAR**

**PEMROGRAMAN WEB II**

**APLIKASI CLEANING SERVICE**

202311107

Rangga Ardinsa Putra

Kelas A

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS SEPULUH NOPEMBER PAPUA**

**2024**

KATA PENGANTAR

Puji dan syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga laporan pengembangan aplikasi *Cleaning Service* berbasis web menggunakan CodeIgniter 4 ini dapat diselesaikan dengan baik. Aplikasi ini dirancang dengan memanfaatkan framework CodeIgniter 4, yang mendukung metode pengembangan berbasis *Model-View-Controller* (MVC) untuk menghasilkan sistem yang efisien dan mudah dikelola.

Laporan ini disusun sebagai salah satu tugas mata kuliah yang diasuh oleh Bapak Ajenkris Yanto Kungkung, MMSI. Selain sebagai bentuk pemenuhan tugas, laporan ini diharapkan dapat memberikan kontribusi positif, baik sebagai referensi pengembangan ilmu pengetahuan maupun sebagai landasan implementasi di dunia kerja.

Dalam proses penyusunan laporan ini, penulis mendapatkan bimbingan, dukungan, serta saran yang berharga dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat, penulis menyampaikan terima kasih kepada:

1. **Bapak Ajenkris Yanto Kungkung, MMSI**, selaku dosen pengampu mata kuliah, atas ilmu, bimbingan, dan masukan yang sangat berarti selama pelaksanaan proyek ini.
2. Seluruh pihak yang telah membantu, baik secara langsung maupun tidak langsung, sehingga laporan ini dapat terselesaikan dengan baik.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan, baik dalam hal isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang.

Akhir kata, semoga laporan ini dapat memberikan manfaat bagi pembaca dan semua pihak yang berkepentingan. Kami juga berharap laporan ini dapat menjadi langkah kecil dalam mendukung pengembangan teknologi informasi, khususnya di bidang pengelolaan layanan *Cleaning Service*.

**Jayapura, 6 Desember 2024**

**Rangga Ardinsa Putra**

**DAFTAR ISI**

[A. Desain Basis Data 6](#_Toc184501648)

[**A.1.** **Tabel dan Penjelasan Kolom** 6](#_Toc184501649)

[**A.2.** **Relasi Antar Tabel** 7](#_Toc184501650)

[B. Implementasi 8](#_Toc184501651)

[**B.1.** **Admins** 8](#_Toc184501652)

[B.1.1. Model 8](#_Toc184501653)

[B.1.2. Controller dan View 9](#_Toc184501654)

[B.1.3. View 10](#_Toc184501655)

[**B.2.** **Customer** 11](#_Toc184501656)

[B.2.1. Model 11](#_Toc184501657)

[B.2.2. Controller dan View 11](#_Toc184501658)

[B.2.3. View 18](#_Toc184501659)

[**B.3.** **Employees** 19](#_Toc184501660)

[B.3.1. Model 19](#_Toc184501661)

[B.3.2. Controller dan View 20](#_Toc184501662)

[B.3.3. View 24](#_Toc184501663)

[**B.4.** **Service** 24](#_Toc184501664)

[B.4.1. Model 24](#_Toc184501665)

[B.4.2. Controller dan View 25](#_Toc184501666)

[B.4.3. View 29](#_Toc184501667)

[**B.5.** **Comment** 30](#_Toc184501668)

[B.5.1. Model 30](#_Toc184501669)

[B.5.2. Controller dan View 31](#_Toc184501670)

[B.5.3. View 39](#_Toc184501671)

[**B.6.** **Payment** 40](#_Toc184501672)

[B.6.1. Model 40](#_Toc184501673)

[B.6.2. Controller dan View 40](#_Toc184501674)

[B.6.3. View 49](#_Toc184501675)

[C. Blackbox Testing 50](#_Toc184501676)

[D. Daftar Pustaka 53](#_Toc184501677)

DAFTAR GAMBAR

[Gambar 1. 1Model Database Aplikasi Cleaning Service 6](#_Toc184503764)

[Gambar 1. 2 Auth Login 11](#_Toc184503765)

[Gambar 1. 3 Index customers 18](#_Toc184503766)

[Gambar 1. 4 Tambah 19](#_Toc184503767)

[Gambar 1. 5 Edit Task 19](#_Toc184503768)

[Gambar 1. 6 Index Employee 24](#_Toc184503769)

[Gambar 1. 7 Tambah Employee 24](#_Toc184503770)

[Gambar 1. 8 Edit Employee 24](#_Toc184503771)

[Gambar 1. 9 index service 29](#_Toc184503772)

[Gambar 1. 10 Tambah Service 30](#_Toc184503773)

[Gambar 1. 11 Edit Service 30](#_Toc184503774)

[Gambar 1. 12 Index Order 39](#_Toc184503775)

[Gambar 1. 13 Tambah order 39](#_Toc184503776)

[Gambar 1. 14 Edit order 40](#_Toc184503777)

[Gambar 1. 15 Index Payment 49](#_Toc184503778)

[Gambar 1. 16 Tambah Payment 49](#_Toc184503779)

[Gambar 1. 17 Edit Payment 49](#_Toc184503780)

DAFTAR TABEL

[Tabel 1. 1 Blackbox Testing **Error! Bookmark not defined.**](#_Toc184462950)

# Desain Basis Data

Gambar 1. Model Database Aplikasi Cleaning Service

Model database pada gambar di atas adalah desain Entity-Relationship Diagram (ERD) untuk Aplikasi Cleaning Service. Berikut adalah penjelasan setiap tabel beserta relasi antar tabel :

## **Tabel dan Penjelasan Kolom**

1. **Tabel** A**dmins**

* **Deskripsi** : Menyimpan data pengguna.
* **Kolom**:

adminID: ID unik admin

username: Nama pengguna untuk login

password: Kata sandi untuk login.

1. **Tabel customers**

* **Deskripsi** :- Menyimpan informasi pelanggan yang memanfaatkan layanan.

- Digunakan untuk mengelola data pelanggan, seperti nama, kontak, dan alamat

* **Kolom**:

id: ID unik pelanggan

name: Nama pelanggan.

email: Email pelanggan (unik).

phone: Nomor telepon pelanggan.

address: Alamat pelanggan.

1. **Tabel** **Employees**

* **Deskripsi** : Menyimpan data karyawan yang menyediakan layanan kepada pelanggan.
* **Kolom**:

id: ID unik karyawan.

name: Nama karyawan

email: Email karyawan (unik)

phone: Nomor telepon karyawan.

position: Jabatan atau posisi karyawan.

hire\_date: Tanggal karyawan mulai bekerja.

1. **Tabel services**

* **Deskripsi :** Menyimpan daftar layanan yang tersedia dalam sistem.
* **Kolom** **:**

id: ID unik layanan.

name: Nama layanan

description: Deskripsi layanan

price: Harga layanan (format desimal).

duration: Durasi layanan (dalam menit)

1. **Tabel orders**

* **Deskripsi :** Menyimpan data pesanan layanan dari pelanggan.

- Menghubungkan pelanggan, layanan, dan karyawan untuk suatu pesanan.

- Digunakan untuk melacak status pesanan, waktu, dan harga total.

**Kolom:**

id: ID unik pesananname: Nama status.

customer\_id: Referensi ke pelanggan yang memesan (relasi ke customers.id).

service\_id: Referensi ke layanan yang dipesan (relasi ke services.id).

employee\_id: Referensi ke karyawan yang menangani pesanan (relasi ke employees.id)

order\_date: Tanggal pesanan dibuat

start\_time: Waktu mulai layanan.

status: Status pesanan (pending, confirmed, in\_progress, completed, cancelled).

total\_price: Total harga pesanan.

1. **Tabel** **payments**

* **Deskripsi :** Menyimpan data pembayaran untuk pesanan tertentu.

- Digunakan untuk melacak metode pembayaran, status, dan jumlah yang dibayar.

* **Kolom**:

id: ID unik pembayaran.

order\_id: Referensi ke pesanan yang dibayar (relasi ke orders.id).

payment\_date: Tanggal pembayaran dilakukan.

payment\_method: Metode pembayaran (cash, credit\_card, bank\_transfer).

status: Status pembayaran (pending, completed, failed)

## **Relasi Antar Tabel**

1. **Relasi antara orders dan customers**

* **Jenis Relasi:** One-to-Many
* **Penjelasan:**
* Satu pelanggan (customers) dapat memiliki banyak pesanan (orders).

Setiap pesanan terkait dengan satu pelanggan yang diidentifikasi oleh kolom customer\_id di tabel orders.

1. **Relasi** **antara orders dan services**

* **Jenis** **Relasi:** One-to-Many
* **Penjelasan:**
* Satu layanan (services) dapat dipesan dalam banyak pesanan (orders).

Setiap pesanan terkait dengan satu layanan yang diidentifikasi oleh kolom service\_id di tabel orders.

1. **Relasi** **antara orders dan employees**

* **Jenis Relasi:** One-to-Many
* **Penjelasan:**
* Satu karyawan (employees) dapat menangani banyak pesanan (orders).

Setiap pesanan, jika ada karyawan yang ditugaskan, terkait dengan satu karyawan yang diidentifikasi oleh kolom employee\_id di tabel orders.

Kolom employee\_id dapat bernilai NULL jika tidak ada karyawan yang ditugaskan.

1. **Relasi** **antara payments dan orders**

* **Jenis Relasi: One-to-One**
* **Penjelasan:**
* Satu pesanan (orders) hanya dapat memiliki satu pembayaran (payments).

Setiap pembayaran terkait dengan satu pesanan yang diidentifikasi oleh kolom order\_id di tabel payments.

# Implementasi

## **Admins**

### Model

<?php

namespace App\Models;

use CodeIgniter\Model;

class AdminModel extends Model

{

protected $table = 'admins';

protected $primaryKey = 'AdminID';

protected $allowedFields = ['username', 'password'];

public function checkLogin($username, $password)

{

$user = $this->where('username', $username)->first();

if ($user && password\_verify($password, $user['password'])) {

return $user;

}

return false;

}

### Controller dan View

<?php

namespace App\Controllers;

use App\Models\UserModel;

class UserController extends BaseController

{

protected $userModel;

public function \_\_construct()

{

$this->userModel = new UserModel();

}

public function index()

{

$data['users'] = $this->userModel->findAll();

return view('users/index', $data);

}

public function create()

{

return view('users/create');

}

public function store()

{

$this->userModel->save($this->request->getPost());

return redirect()->to('/users');

}

public function edit($id)

{

$data['user'] = $this->userModel->find($id);

return view('users/edit', $data);

}

public function update($id)

{

$this->userModel->update($id, $this->request->getPost());

return redirect()->to('/users');

}

public function delete($id)

{

$this->userModel->delete($id);

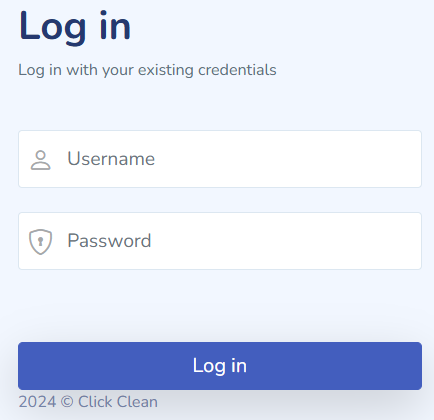
return redirect()->to('/users');

}

}

### View

1. **Index Tampilan Login Admin & Dokter**



Gambar 1. Auth Login

## **Customer**

### Model

<?php

namespace App\Models;

use CodeIgniter\Model;

class CustomersModel extends Model

{

protected $table = 'customers';

protected $primaryKey = 'id';

protected $useAutoIncrement = false;

protected $returnType = 'array';

protected $useSoftDeletes = false;

protected $protectFields = true;

protected $allowedFields = ['id','name', 'email', 'phone', 'address', 'created\_at'];

}

### Controller dan View

1. Menampilkan Data
2. Controller

<?php

namespace App\Controllers;

use App\Models\CustomersModel;

use CodeIgniter\Controller;

class CustomersController extends Controller

{

protected $customerModel;

public function \_\_construct()

{

$this->customerModel = new CustomersModel();

}

// List all customers

public function index()

{

$data['customers'] = $this->customerModel->findAll();

return view('customers/index', $data);

}

1. View

<?= $this->extend('layouts/app') ?>

<?= $this->section('content') ?>

<div class="page-heading">

<div class="page-title">

<div class="row">

<div class="col-12 col-md-6 order-md-1 order-last">

<h3>Customers List</h3>

<p class="text-subtitle text-muted">List of customers and their details</p>

</div>

<div class="col-12 col-md-6 order-md-2 order-first">

<nav aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

<ol class="breadcrumb">

<li class="breadcrumb-item"><a href="/dashboard">Dashboard</a></li>

<li class="breadcrumb-item active" aria-current="page">Customers</li>

</ol>

</nav>

</div>

</div>

</div>

<section class="section">

<div class="card">

<div class="card-header">

Customers

<a href="/customers/create" class="btn btn-primary btn-sm float-end">Create New Order</a>

</div>

<div class="card-body">

<table class="table table-striped" id="customersTable">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Phone</th>

<th>Address</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<?php foreach ($customers as $customer) : ?>

<tr>

<td><?= esc($customer['id']) ?></td>

<td><?= esc($customer['name']) ?></td>

<td><?= esc($customer['email']) ?></td>

<td><?= esc($customer['phone']) ?></td>

<td><?= esc($customer['address']) ?></td>

<td>

<!-- <a href="/customers/show/<?= esc($customer['id']) ?>" class="btn btn-info btn-sm">View</a> -->

<a href="/customers/edit/<?= esc($customer['id']) ?>" class="btn btn-warning btn-sm">Edit</a>

<a href="/customers/delete/<?= esc($customer['id']) ?>" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to delete this customer?');">Delete</a>

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

</div>

</div>

</section>

</div>

<?= $this->endSection() ?>

<?= $this->section('styles') ?>

<link rel="stylesheet" href="/assets/vendors/simple-datatables/style.css">

<?= $this->endSection() ?>

<?= $this->section('javascript') ?>

<script src="/assets/vendors/simple-datatables/simple-datatables.js"></script>

<script>

document.addEventListener('DOMContentLoaded', function () {

let customersTable = document.querySelector('#customersTable');

if (customersTable) {

new simpleDatatables.DataTable(customersTable);

}

});

</script>

<?= $this->endSection() ?>

1. Menambahkan Data
2. Controller

public function create()

{

return view('customers/create');

}

// Store a new customer

public function store()

{

$this->customerModel->save([

'id' => $this->request->getPost('id'),

'name' => $this->request->getPost('name'),

'email' => $this->request->getPost('email'),

'phone' => $this->request->getPost('phone'),

'address' => $this->request->getPost('address'),

]);

return redirect()->to('customers');

}

1. View

<?= $this->extend('layouts/app') ?>

<?= $this->section('content') ?>

<div class="page-heading">

<div class="page-title">

<div class="row">

<div class="col-12 col-md-6 order-md-1 order-last">

<h3>Create Customer</h3>

<p class="text-subtitle text-muted">Fill in the form below to add a new customer</p>

</div>

<div class="col-12 col-md-6 order-md-2 order-first">

<nav aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

<ol class="breadcrumb">

<li class="breadcrumb-item"><a href="/dashboard">Dashboard</a></li>

<li class="breadcrumb-item"><a href="/customers">Customers</a></li>

<li class="breadcrumb-item active" aria-current="page">Create</li>

</ol>

</nav>

</div>

</div>

</div>

<section class="section">

<div class="card">

<div class="card-header">

Add Customer

</div>

<div class="card-body">

<form action="/customers/store" method="post">

<?= csrf\_field() ?>

<div class="form-group">

<label for="id">ID</label>

<input type="text" id="id" name="id" class="form-control" required>

</div>

<div class="form-group">

<label for="name">Name</label>

<input type="text" id="name" name="name" class="form-control" required>

</div>

<div class="form-group">

<label for="email">Email</label>

<input type="email" id="email" name="email" class="form-control" required>

</div>

<div class="form-group">

<label for="phone">Phone</label>

<input type="text" id="phone" name="phone" class="form-control" required>

</div>

<div class="form-group">

<label for="address">Address</label>

<textarea id="address" name="address" class="form-control" rows="4" required></textarea>

</div>

<div class="d-flex justify-content-end">

<a href="/customers" class="btn btn-secondary me-2">Cancel</a>

<button type="submit" class="btn btn-primary">Save</button>

</div>

</form>

</div>

</div>

</section>

</div>

<?= $this->endSection() ?>

1. Mengubah Data
2. Controller

// Edit customer

public *function* edit($id)

{

$data['customer'] = $this->customerModel->find($id);

return view('customers/edit', $data);

}

// Update customer

public *function* update($id)

{

$this->customerModel->update($id, [

'id'    => $this->request->getPost('id'),

'name'    => $this->request->getPost('name'),

'email'   => $this->request->getPost('email'),

'phone'   => $this->request->getPost('phone'),

'address' => $this->request->getPost('address'),

]);

return redirect()->to('/customers');

}

1. View

<?= $this->extend('layouts/app') ?>

<?= $this->section('content') ?>

<div class="page-heading">

<div class="page-title">

<div class="row">

<div class="col-12 col-md-6 order-md-1 order-last">

<h3>Update Customer</h3>

<p class="text-subtitle text-muted">Fill in the form below to update customer details</p>

</div>

<div class="col-12 col-md-6 order-md-2 order-first">

<nav aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

<ol class="breadcrumb">

<li class="breadcrumb-item"><a href="/dashboard">Dashboard</a></li>

<li class="breadcrumb-item"><a href="/customers">Customers</a></li>

<li class="breadcrumb-item active" aria-current="page">Update</li>

</ol>

</nav>

</div>

</div>

</div>

<section class="section">

<div class="card">

<div class="card-header">

Update Customer

</div>

<div class="card-body">

<form action="/customers/update/<?= $customer['id'] ?>" method="post">

<?= csrf\_field() ?>

<!-- ID (Hidden) -->

<div class="form-group">

<label for="id">ID</label>

<input type="text" id="id" name="id" class="form-control" value="<?= $customer['id'] ?>" readonly>

</div>

<!-- Name -->

<div class="form-group">

<label for="name">Name</label>

<input type="text" id="name" name="name" class="form-control" value="<?= $customer['name'] ?>" required>

</div>

<!-- Email -->

<div class="form-group">

<label for="email">Email</label>

<input type="email" id="email" name="email" class="form-control" value="<?= $customer['email'] ?>" required>

</div>

<!-- Phone -->

<div class="form-group">

<label for="phone">Phone</label>

<input type="text" id="phone" name="phone" class="form-control" value="<?= $customer['phone'] ?>" required>

</div>

<!-- Address -->

<div class="form-group">

<label for="address">Address</label>

<textarea id="address" name="address" class="form-control" rows="4" required><?= $customer['address'] ?></textarea>

</div>

<div class="d-flex justify-content-end">

<a href="/customers" class="btn btn-secondary me-2">Cancel</a>

<button type="submit" class="btn btn-primary">Update</button>

</div>

</form>

</div>

</div>

</section>

</div>

<?= $this->endSection() ?>

1. Menghapus Data
2. Controller

public function delete($id)

{

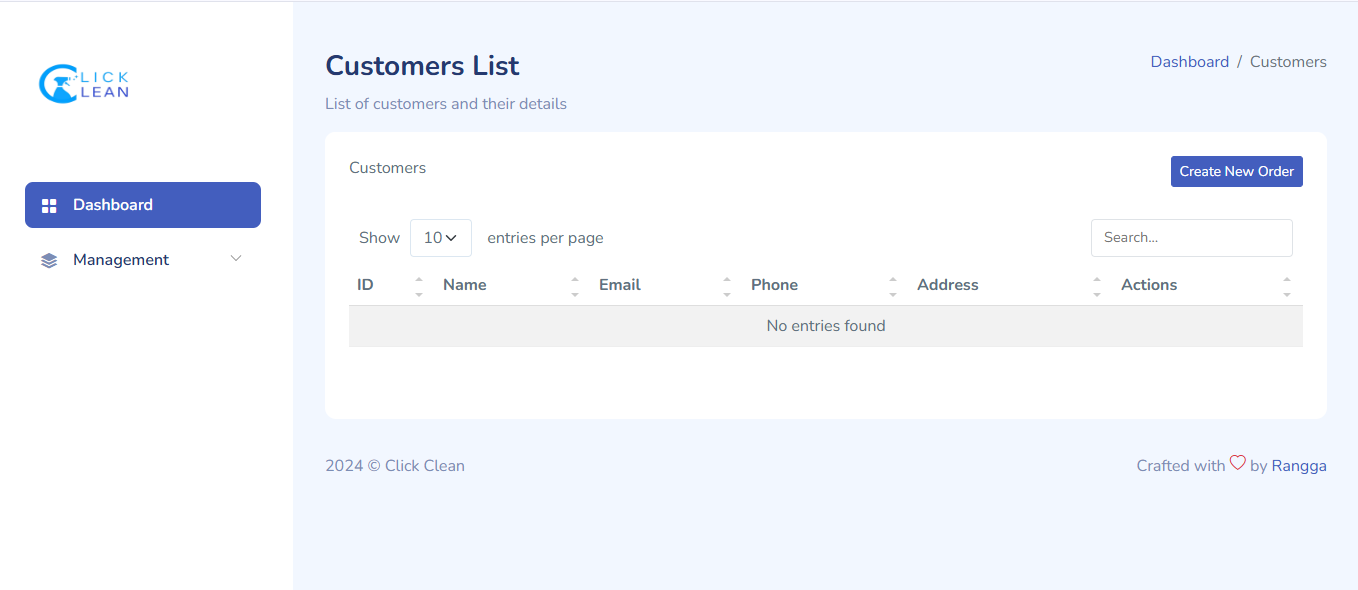
$this->customerModel->delete($id);

return redirect()->to('/customers');

}

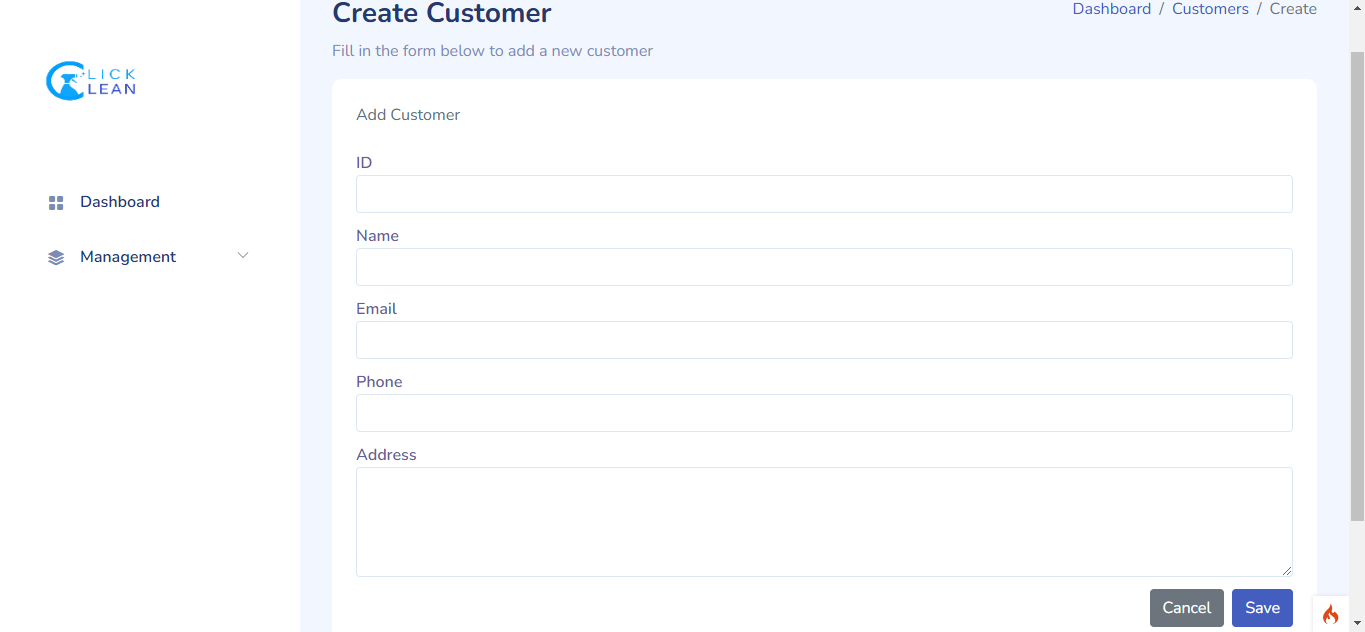
### View

1. **Index Customer**



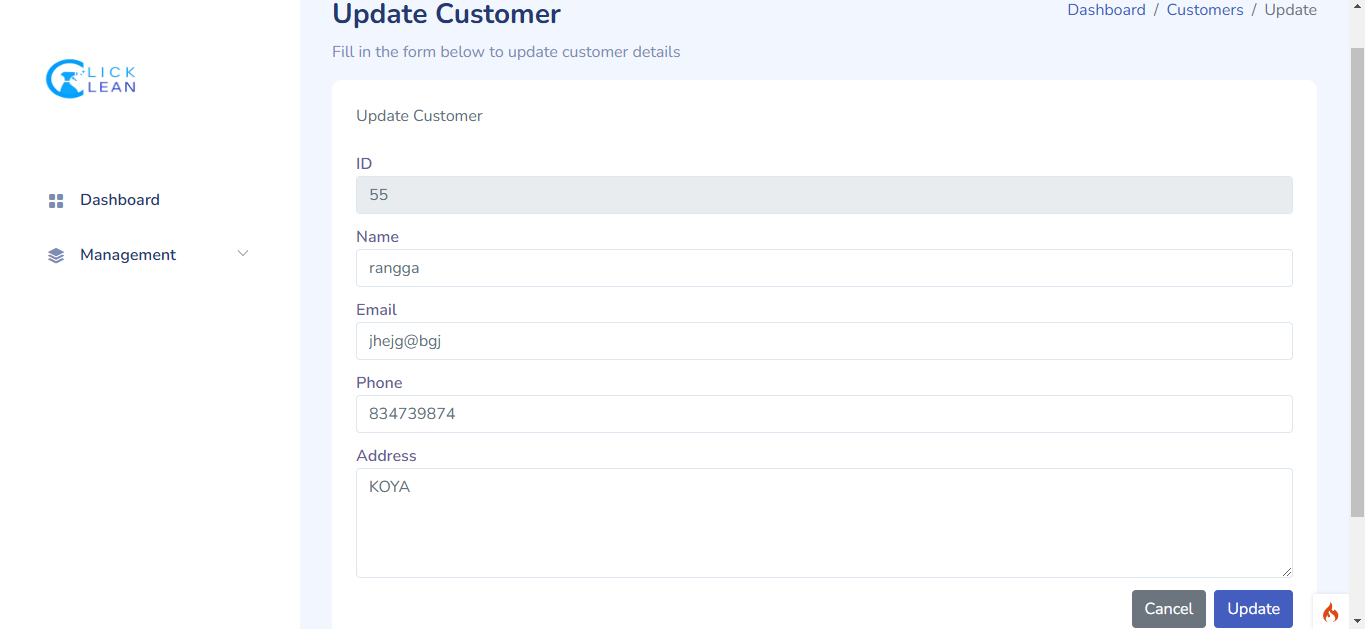
Gambar 1. Index customers

1. **Tambah Customer**



Gambar 1. Tambah

1. **edit Customer**



Gambar 1. Edit Task

## **Employees**

### Model

<?php

namespace App\Models;

use CodeIgniter\Model;

class EmployeeModel extends Model

{

protected $table = 'employees';

protected $primaryKey = 'id';

protected $useAutoIncrement = true;

protected $returnType = 'array';

protected $useSoftDeletes = false;

protected $protectFields = true;

protected $allowedFields = ['name', 'email', 'phone', 'position', 'hire\_date', 'created\_at'];

}

### Controller dan View

1. Menampilkan Data
2. Controller

public function index()

{

$data['employees'] = $this->employeeModel->findAll();

return view('Employee/index', $data);

}

1. View

<table class="table table-striped" id="employeesTable">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Phone</th>

<th>Position</th>

<th>Hire Date</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<?php if (!empty($employees) && is\_array($employees)): ?>

<?php foreach ($employees as $employee): ?>

<tr>

<td><?= esc($employee['id']) ?></td>

<td><?= esc($employee['name']) ?></td>

<td><?= esc($employee['email']) ?></td>

<td><?= esc($employee['phone']) ?></td>

<td><?= esc($employee['position']) ?></td>

<td><?= esc($employee['hire\_date']) ?></td>

<td>

<a href="/employees/edit/<?= $employee['id'] ?>" class="btn btn-warning btn-sm">Edit</a>

<a href="/employees/delete/<?= $employee['id'] ?>" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to delete this employee?');">Delete</a>

</td>

</tr>

<?php endforeach; ?>

<?php else: ?>

<tr>

<td colspan="7">No employees found.</td>

</tr>

<?php endif; ?>

</tbody>

</table>

</div>

</div>

</section>

</div>

1. Menambahkan Data
2. Controller

public function create()

{

return view('Employee/create');

}

public function store()

{

$this->employeeModel->save([

'name' => $this->request->getPost('name'),

'email' => $this->request->getPost('email'),

'phone' => $this->request->getPost('phone'),

'position' => $this->request->getPost('position'),

'hire\_date' => $this->request->getPost('hire\_date'),

]);

return redirect()->to('/employees');

}

1. View

<section class="section">

<div class="card">

<div class="card-header">

Add Employee

</div>

<div class="card-body">

<form action="/employees/store" method="post">

<?= csrf\_field() ?>

<div class="form-group">

<label for="name">Name</label>

<input type="text" id="name" name="name" class="form-control" placeholder="Enter full name" required>

</div>

<div class="form-group">

<label for="email">Email</label>

<input type="email" id="email" name="email" class="form-control" placeholder="Enter email address" required>

</div>

<div class="form-group">

<label for="phone">Phone</label>

<input type="text" id="phone" name="phone" class="form-control" placeholder="Enter phone number" required>

</div>

<div class="form-group">

<label for="position">Position</label>

<input type="text" id="position" name="position" class="form-control" placeholder="Enter job position" required>

</div>

<div class="form-group">

<label for="hire\_date">Hire Date</label>

<input type="date" id="hire\_date" name="hire\_date" class="form-control" required>

</div>

<div class="d-flex justify-content-end">

<a href="/employees" class="btn btn-secondary me-2">Cancel</a>

<button type="submit" class="btn btn-primary">Save</button>

</div>

</form>

</div>

</div>

</section>

1. Mengubah Data
2. Controller

public function edit($id)

{

$data['employee'] = $this->employeeModel->find($id);

return view('Employee/edit', $data);

}

public function update($id)

{

$this->employeeModel->update($id, [

'name' => $this->request->getPost('name'),

'email' => $this->request->getPost('email'),

'phone' => $this->request->getPost('phone'),

'position' => $this->request->getPost('position'),

'hire\_date' => $this->request->getPost('hire\_date'),

]);

return redirect()->to('/employees');

}

1. View

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Edit Kategori</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

</head>

<body>

<div class="container mt-5">

<h1>Edit Kategori</h1>

<form action="/categories/update/<?= $category['id'] ?>" method="post">

<?= csrf\_field() ?>

<div class="mb-3">

<label for="name" class="form-label">Nama Kategori</label>

<input type="text" class="form-control" id="name" name="name" value="<?= $category['name'] ?>" required>

</div>

<button type="submit" class="btn btn-primary">Perbarui</button>

<a href="/categories" class="btn btn-secondary">Batal</a>

</form>

</div>

</body>

</html>

1. Menghapus Data
2. Controller

// Delete employee

public function delete($id)

{

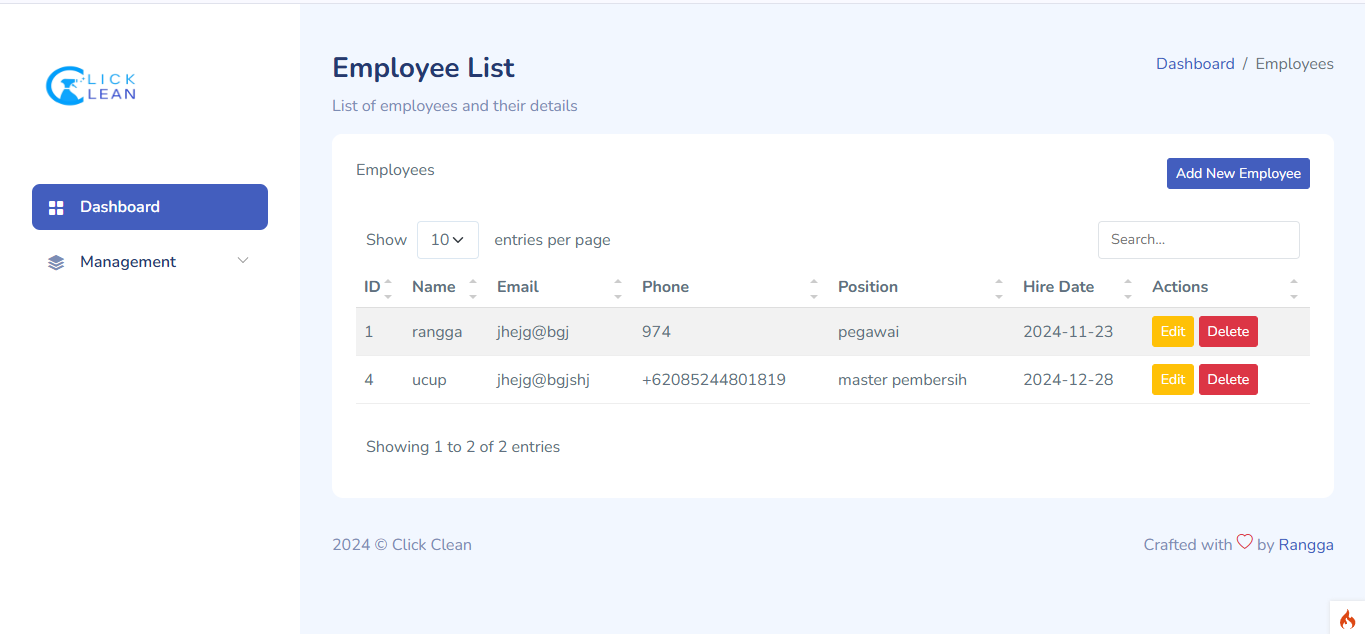
$this->employeeModel->delete($id);

return redirect()->to('/employees');

}

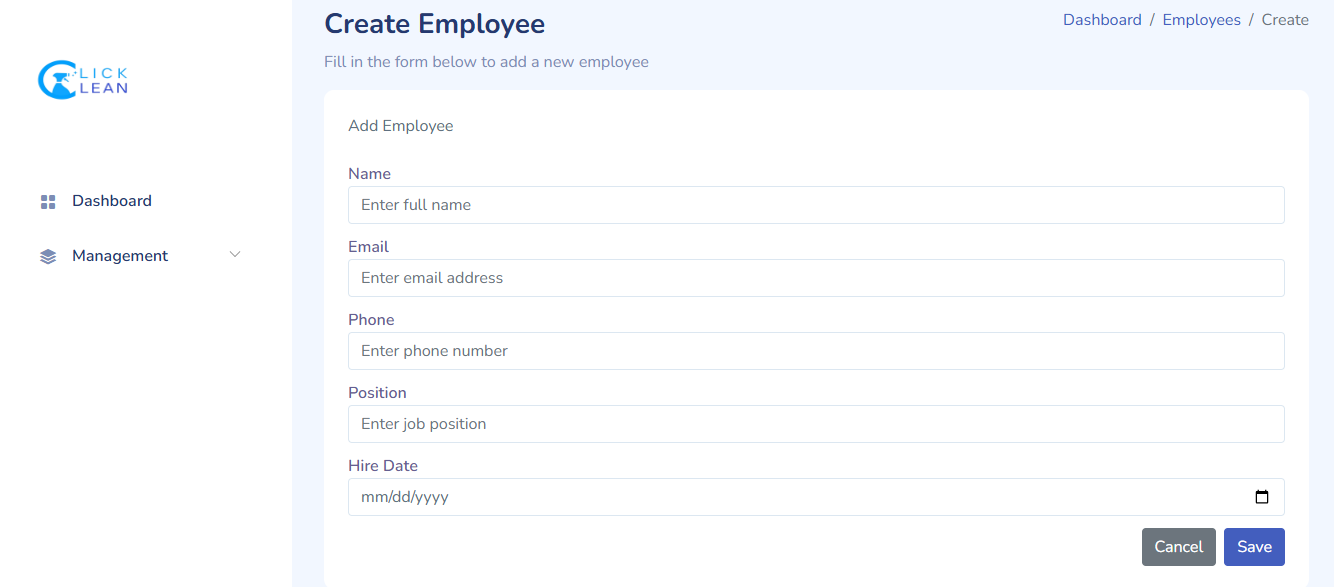
### View

1. **Index Employee**



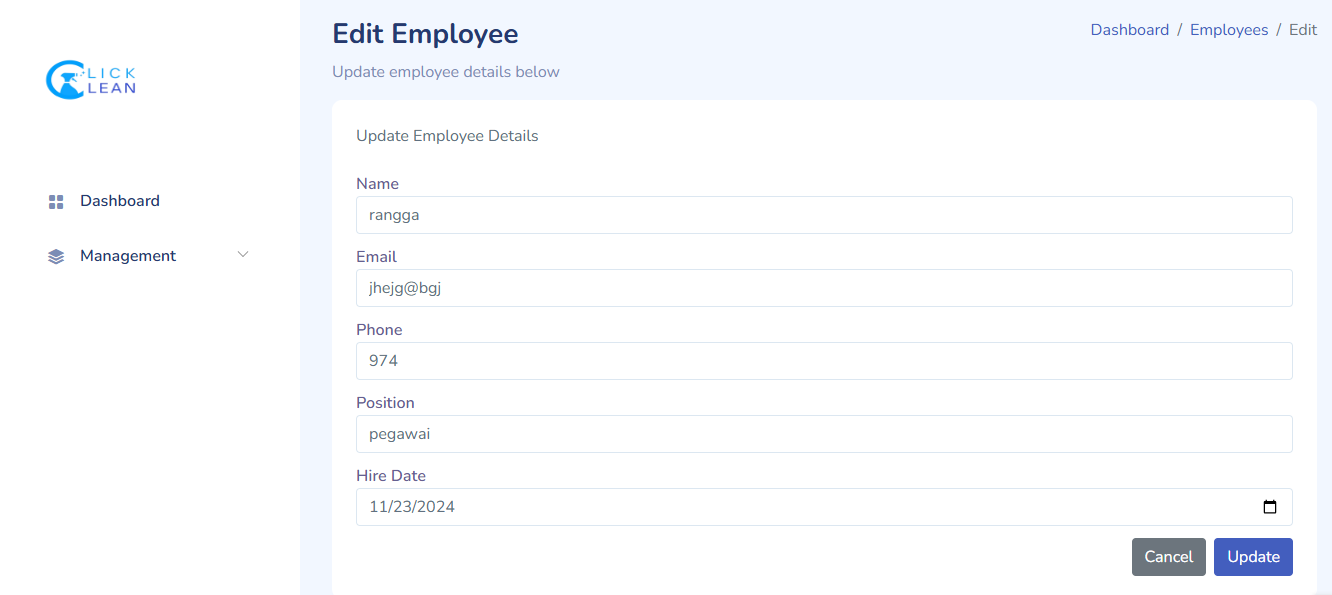
Gambar 1. Index Employee

1. **Tambah Employee**



Gambar 1. Tambah Employee

1. **Edit Employee**



Gambar 1. Edit Employee

## **Service**

### Model

<?php

namespace App\Models;

use CodeIgniter\Model;

class ServiceModel extends Model

{

protected $table = 'services';

protected $primaryKey = 'id';

protected $allowedFields = ['name', 'description', 'price', 'duration'];

}

### Controller dan View

1. Menampilkan Data
2. Controller

namespace App\Controllers;

use App\Models\ServiceModel;

class ServicesController extends BaseController

{

protected $serviceModel;

public function \_\_construct()

{

$this->serviceModel = new ServiceModel();

}

// Read: Menampilkan daftar services

public function index()

{

$data['services'] = $this->serviceModel->findAll();

return view('service/index', $data);

}

1. View

<div class="card-body">

<div class="table-responsive">

<table class="table table-striped" id="servicesTable">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Description</th>

<th>Price</th>

<th>Duration</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<?php if (!empty($services) && is\_array($services)): ?>

<?php foreach ($services as $service): ?>

<tr>

<td><?= esc($service['id']) ?></td>

<td><?= esc($service['name']) ?></td>

<td><?= esc($service['description']) ?></td>

<td>Rp <?= number\_format($service['price'], 0, ',', '.') ?></td>

<td><?= esc($service['duration']) ?> minutes</td>

<td>

<a href="/services/edit/<?= esc($service['id']) ?>" class="btn btn-warning btn-sm">Edit</a>

<a href="/services/delete/<?= esc($service['id']) ?>" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to delete this service?');">Delete</a>

</td>

</tr>

<?php endforeach; ?>

<?php else: ?>

<tr>

<td colspan="6" class="text-center">No services found.</td>

</tr>

<?php endif; ?>

</tbody>

</table>

</div>

</div>

</div>

</section>

</div>

1. Menambahkan Data
2. Controller

public function create()

{

return view('service/create');

}

// Store: Menyimpan service baru

public function store()

{

$this->serviceModel->save([

'name' => $this->request->getPost('name'),

'description' => $this->request->getPost('description'),

'price' => $this->request->getPost('price'),

'duration' => $this->request->getPost('duration'),

]);

return redirect()->to('/services');

}

// Edit: Form untuk edit service

public function edit($id)

{

$data['service'] = $this->serviceModel->find($id);

return view('service/edit', $data);

}

1. View

<section class="section">

<div class="card">

<div class="card-header">

New Service Details

</div>

<div class="card-body">

<form action="/services/store" method="post">

<?= csrf\_field() ?>

<!-- Name -->

<div class="mb-3">

<label for="name" class="form-label">Name</label>

<input type="text" id="name" name="name" class="form-control" required>

</div>

<!-- Description -->

<div class="mb-3">

<label for="description" class="form-label">Description</label>

<textarea id="description" name="description" class="form-control" rows="4"></textarea>

</div>

<!-- Price -->

<div class="mb-3">

<label for="price" class="form-label">Price</label>

<input type="number" id="price" name="price" class="form-control" required>

</div>

<!-- Duration -->

<div class="mb-3">

<label for="duration" class="form-label">Duration (in minutes)</label>

<input type="number" id="duration" name="duration" class="form-control" required>

</div>

<div class="d-flex justify-content-end">

<a href="/services" class="btn btn-secondary me-2">Cancel</a>

<button type="submit" class="btn btn-primary">Create Service</button>

</div>

</form>

</div>

</div>

</section>

</div>

<?= $this->endSection() ?>

1. Mengubah Data
2. Controller

public function edit($id)

{

$data['service'] = $this->serviceModel->find($id);

return view('service/edit', $data);

}

// Update: Menyimpan perubahan service

public function update($id)

{

$this->serviceModel->update($id, [

'name' => $this->request->getPost('name'),

'description' => $this->request->getPost('description'),

'price' => $this->request->getPost('price'),

'duration' => $this->request->getPost('duration'),

]);

return redirect()->to('/services');

}

1. View

<section class="section">

<div class="card">

<div class="card-header">

Edit Service Details

</div>

<div class="card-body">

<form action="/services/update/<?= esc($service['id']) ?>" method="post">

<?= csrf\_field() ?>

<!-- Name -->

<div class="mb-3">

<label for="name" class="form-label">Name</label>

<input type="text" id="name" name="name" class="form-control" value="<?= esc($service['name']) ?>" required>

</div>

<!-- Description -->

<div class="mb-3">

<label for="description" class="form-label">Description</label>

<textarea id="description" name="description" class="form-control" rows="4"><?= esc($service['description']) ?></textarea>

</div>

<!-- Price -->

<div class="mb-3">

<label for="price" class="form-label">Price</label>

<input type="number" id="price" name="price" class="form-control" value="<?= esc($service['price']) ?>" required>

</div>

<!-- Duration -->

<div class="mb-3">

<label for="duration" class="form-label">Duration (in minutes)</label>

<input type="number" id="duration" name="duration" class="form-control" value="<?= esc($service['duration']) ?>" required>

</div>

<div class="d-flex justify-content-end">

<a href="/services" class="btn btn-secondary me-2">Cancel</a>

<button type="submit" class="btn btn-primary">Update Service</button>

</div>

</form>

</div>

</div>

</section>

</div>

1. Menghapus Data
2. Controller

public function delete($id)

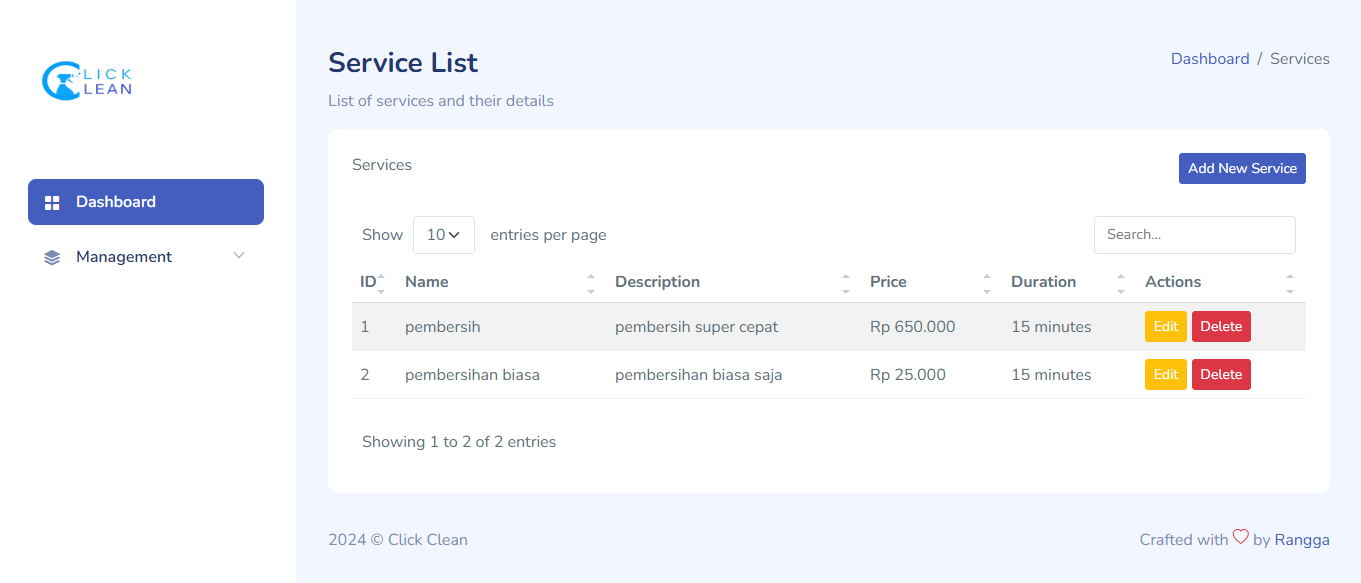
{

$this->serviceModel->delete($id);

return redirect()->to('/services');

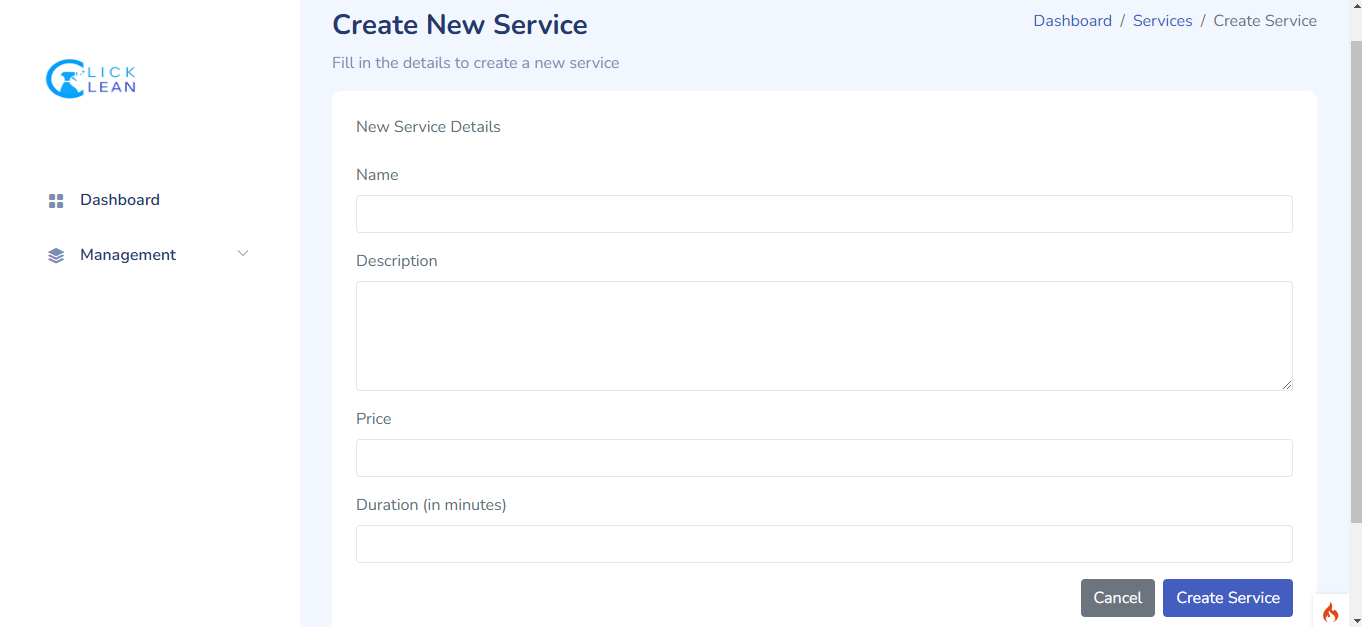
}

### View

1. **Index service**

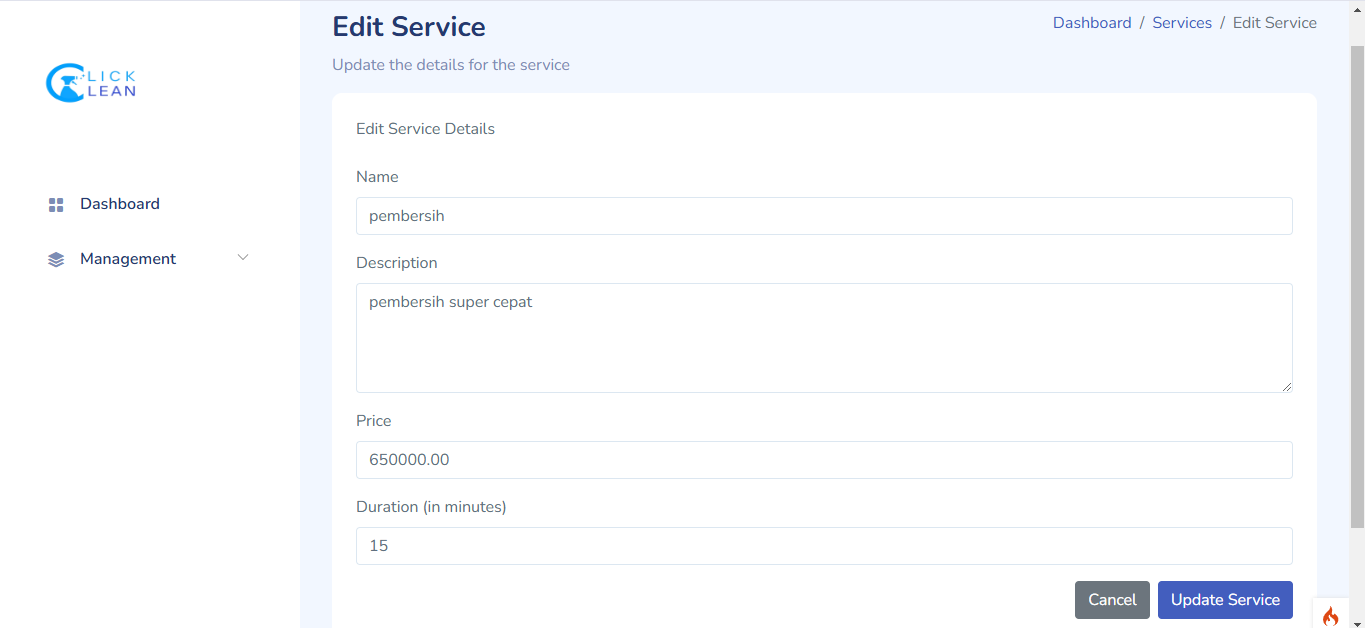
Gambar 1. index service

1. **Tambah Service**



Gambar 1. Tambah Service

1. **Edit Task Assignment**



Gambar 1. Edit Service

## **Comment**

### Model

<?php

namespace App\Models;

use CodeIgniter\*Model*;

*class* OrdersModel extends *Model*

{

protected $table = 'orders';

protected $primaryKey = 'id';

protected $useAutoIncrement = false;

protected $allowedFields = [

'customer\_id',

'service\_id',

'employee\_id',

'order\_date',

'start\_time',

'status',

'total\_price'

];

/ Mengambil satu data order berdasarkan ID dengan relasi ke tabel terkait

public function getOrder($id)

{

return $this->select('orders.\*, customers.name as customer\_name, services.price as service\_price,services.name as services\_name ,employees.name as employee\_name')

->join('customers', 'customers.id = orders.customer\_id', 'left')

->join('services', 'services.id = orders.service\_id', 'left')

->join('employees', 'employees.id = orders.employee\_id', 'left')

->find($id);

}

}

### Controller dan View

1. Menampilkan Data
2. Controller

namespace App\Controllers;

use App\Models\*OrdersModel*;

use App\Models\*CustomersModel*;

use App\Models\*ServiceModel*;

use App\Models\*EmployeeModel*;

*class* OrdersController extends *BaseController*

{

protected $orderModel;

protected $customerModel;

protected $serviceModel;

protected $employeeModel;

public *function* \_\_construct()

{

$this->orderModel = new *OrdersModel*();

$this->customerModel = new *CustomersModel*();

$this->serviceModel = new *ServiceModel*();

$this->employeeModel = new *EmployeeModel*();

}

// Read: Menampilkan semua orders

public *function* index()

{

$data['orders'] = $this->orderModel->getOrders(); // Mengambil semua order dengan relasi

$data['getStatusColor'] = [$this, 'getStatusColor']; // Fungsi untuk menentukan warna status

return view('orders/index', $data);

}

public *function* getStatusColor($status)

{

// Menentukan warna berdasarkan status

$colors = [

'pending' => 'warning',

'confirmed' => 'primary',

'in\_progress' => 'info',

'completed' => 'success',

'cancelled' => 'danger'

];

return $colors[$status] ?? 'secondary'; // Default 'secondary' jika status tidak ditemukan

}

1. View

<table class="table table-striped" id="ordersTable">

<thead>

<tr>

<th>ID</th>

<th>Customer Name</th>

<th>Service ID</th>

<th>Employee ID</th>

<th>Order Date</th>

<th>Start Time</th>

<th>Status</th>

<th>Total Price</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<?php if (!empty($orders) && is\_array($orders)): ?>

<?php foreach ($orders as $order): ?>

<tr>

<td><?= esc($order['id']) ?></td>

<td><?= esc($order['customer\_name']) ?> (ID: <?= esc($order['customer\_id']) ?>)</td>

<td><?= esc($order['services\_name']) ?></td>

<td><?= esc($order['employee\_name']) ?> (ID: <?= esc($order['customer\_id']) ?>)</td>

<td><?= esc($order['order\_date']) ?></td>

<td><?= esc($order['start\_time']) ?></td>

<td>

<span class="badge bg-<?= isset($getStatusColor) ? $getStatusColor($order['status']) : 'secondary' ?>">

<?= esc($order['status']) ?>

</span>

</td>

<td>Rp <?= number\_format($order['service\_price'], 0, ',', '.') ?></td>

<td>

<a href="/orders/edit/<?= esc($order['id']) ?>" class="btn btn-warning btn-sm">Edit</a>

<a href="/orders/delete/<?= esc($order['id']) ?>" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to delete this order?');">Delete</a>

</td>

</tr>

<?php endforeach; ?>

<?php else: ?>

<tr>

<td colspan="9" class="text-center">No orders found.</td>

</tr>

<?php endif; ?>

</tbody>

</table>

</div>

</div>

1. Menambahkan Data
2. Controller

public *function* create()

{

$customerModel = new *CustomersModel*();

$serviceModel = new *ServiceModel*();

$employeeModel = new *EmployeeModel*();

$custom = $employeeModel->findAll();

// $order =  $this->orderModel->findAll();

$dataCustome = [];

$dataCheck = [];

foreach ($custom as $key1 => $cus) {

$check = $this->orderModel

->groupStart()

->where('status', 'pending')

->orWhere('status', 'confirmed')

->orWhere('status', 'in\_progress')

->groupEnd()

->where('employee\_id', $cus['id'])

->first();

if($check == null){

$dataCustome[] = $cus;

}

}

// dd($dataCustome);

return view('orders/create', [

'customers' => $customerModel->findAll(),

'services' => $serviceModel->findAll(),

'employees' => $dataCustome,

]);

}

// Store: Menyimpan data order baru

public *function* store()

{

$this->orderModel->save([

'customer\_id' => $this->request->getPost('customer\_id'),

'id' => $this->request->getPost('id'),

'service\_id' => $this->request->getPost('service\_id'),

'employee\_id' => $this->request->getPost('employee\_id'),

'order\_date' => $this->request->getPost('order\_date'),

'start\_time' => $this->request->getPost('start\_time'),

'status' => $this->request->getPost('status'),

'total\_price' => $this->request->getPost('total\_price'),

]);

return redirect()->to('/orders')->with('message', 'Order berhasil ditambahkan.');

}

1. View

<!-- Customer ID dropdown -->

<div class="mb-3">

<label for="customer\_id" class="form-label">Customer</label>

<select id="customer\_id" name="customer\_id" class="form-select" required>

<option value="">Select a Customer</option>

<?php foreach ($customers as $customer): ?>

<option value="<?= esc($customer['id']) ?>">

<?= esc($customer['id']) ?> - <?= esc($customer['name']) ?>

</option>

<?php endforeach; ?>

</select>

</div>

<div class="mb-3">

<label for="service\_id" class="form-label">Service</label>

<select id="service\_id" name="service\_id" class="form-select" required>

<option value="">Select a Service</option>

<?php foreach ($services as $service): ?>

<option value="<?= esc($service['id']) ?>">

<?= esc($service['id']) ?> - <?= esc($service['name']) ?>

</option>

<?php endforeach; ?>

</select>

</div>

<!-- Employee ID dropdown -->

<div class="mb-3">

<label for="employee\_id" class="form-label">Employee</label>

<select id="employee\_id" name="employee\_id" class="form-select">

<option value="">Select an Employee</option>

<?php foreach ($employees as $employee): ?>

<option value="<?= esc($employee['id']) ?>">

<?= esc($employee['id']) ?> - <?= esc($employee['name']) ?>

</option>

<?php endforeach; ?>

</select>

</div>

<!-- Order Date -->

<div class="mb-3">

<label for="order\_date" class="form-label">Order Date</label>

<input type="date" id="order\_date" name="order\_date" class="form-control" required>

</div>

<!-- Start Time -->

<div class="mb-3">

<label for="start\_time" class="form-label">Start Time</label>

<input type="time" id="start\_time" name="start\_time" class="form-control" required>

</div>

<!-- Status dropdown -->

<div class="mb-3">

<label for="status" class="form-label">Status</label>

<select name="status" id="status" class="form-select">

<option value="pending">Pending</option>

<option value="confirmed">Confirmed</option>

<option value="in\_progress">In Progress</option>

<option value="completed">Completed</option>

<option value="cancelled">Cancelled</option>

</select>

</div>

<button type="submit" class="btn btn-primary">Create Order</button>

</form>

</div>

</div>

</section>

</div>

1. Mengubah Data
2. Controller

// Edit: Menampilkan form edit order

public function edit($id)

{

// Ambil data order

$data['order'] = $this->orderModel->getOrder($id);

if (!$data['order']) {

throw new \CodeIgniter\Exceptions\PageNotFoundException("Order dengan ID $id tidak ditemukan.");

}

// Ambil data pelanggan untuk dropdown

$data['customers'] = $this->customerModel->findAll();

// Ambil data layanan dan pegawai

$data['services'] = $this->serviceModel->findAll();

$data['employees'] = $this->employeeModel->findAll(); // Perbaiki nama properti di sini

// Kirim data ke view

return view('orders/edit', $data);

}

// Update: Menyimpan perubahan data order

public function update($id)

{

$this->orderModel->update($id, [

'customer\_id' => $this->request->getPost('customer\_id'),

'service\_id' => $this->request->getPost('service\_id'),

'employee\_id' => $this->request->getPost('employee\_id'),

'order\_date' => $this->request->getPost('order\_date'),

'start\_time' => $this->request->getPost('start\_time'),

'status' => $this->request->getPost('status'),

'total\_price' => $this->request->getPost('total\_price'),

]);

return redirect()->to('/orders')->with('message', 'Order berhasil diperbarui.');

}

1. View

<section class="section">

<div class="card">

<div class="card-header">

Edit Order Details

</div>

<div class="card-body">

<form action="/orders/update/<?= $order['id'] ?>" method="post">

<?= csrf\_field() ?>

<!-- Customer ID dropdown -->

<div class="mb-3">

<label for="customer\_id" class="form-label">Customer</label>

<select id="customer\_id" name="customer\_id" class="form-select" required>

<option value="">Select a Customer</option>

<?php foreach ($customers as $customer): ?>

<option value="<?= esc($customer['id']) ?>" <?= $order['customer\_id'] == $customer['id'] ? 'selected' : '' ?>>

<?= esc($customer['id']) ?> - <?= esc($customer['name']) ?>

</option>

<?php endforeach; ?>

</select>

</div>

<!-- Service ID dropdown -->

<div class="mb-3">

<label for="service\_id" class="form-label">Service</label>

<select id="service\_id" name="service\_id" class="form-select" required>

<option value="">Select a Service</option>

<?php foreach ($services as $service): ?>

<option value="<?= esc($service['id']) ?>" <?= $order['service\_id'] == $service['id'] ? 'selected' : '' ?>>

<?= esc($service['id']) ?> - <?= esc($service['name']) ?>

</option>

<?php endforeach; ?>

</select>

</div>

<!-- Employee ID dropdown -->

<div class="mb-3">

<label for="employee\_id" class="form-label">Employee</label>

<select id="employee\_id" name="employee\_id" class="form-select">

<option value="">Select an Employee</option>

<?php foreach ($employees as $employee): ?>

<option value="<?= esc($employee['id']) ?>" <?= $order['employee\_id'] == $employee['id'] ? 'selected' : '' ?>>

<?= esc($employee['id']) ?> - <?= esc($employee['name']) ?>

</option>

<?php endforeach; ?>

</select>

</div>

<!-- Order Date -->

<div class="mb-3">

<label for="order\_date" class="form-label">Order Date</label>

<input type="date" id="order\_date" name="order\_date" class="form-control" value="<?= esc($order['order\_date']) ?>" required>

</div>

<!-- Start Time -->

<div class="mb-3">

<label for="start\_time" class="form-label">Start Time</label>

<input type="time" id="start\_time" name="start\_time" class="form-control" value="<?= esc($order['start\_time']) ?>" required>

</div>

<!-- Status dropdown -->

<div class="mb-3">

<label for="status" class="form-label">Status</label>

<select name="status" id="status" class="form-select">

<option value="pending" <?= $order['status'] == 'pending' ? 'selected' : '' ?>>Pending</option>

<option value="confirmed" <?= $order['status'] == 'confirmed' ? 'selected' : '' ?>>Confirmed</option>

<option value="in\_progress" <?= $order['status'] == 'in\_progress' ? 'selected' : '' ?>>In Progress</option>

<option value="completed" <?= $order['status'] == 'completed' ? 'selected' : '' ?>>Completed</option>

<option value="cancelled" <?= $order['status'] == 'cancelled' ? 'selected' : '' ?>>Cancelled</option>

</select>

</div>

<div class="d-flex justify-content-end">

<a href="/orders" class="btn btn-secondary me-2">Cancel</a>

<button type="submit" class="btn btn-primary">Update Order</button>

</div>

</form>

</div>

</div>

</section>

</div>

1. Menghapus Data
2. Controller

public *function* delete($id)

{

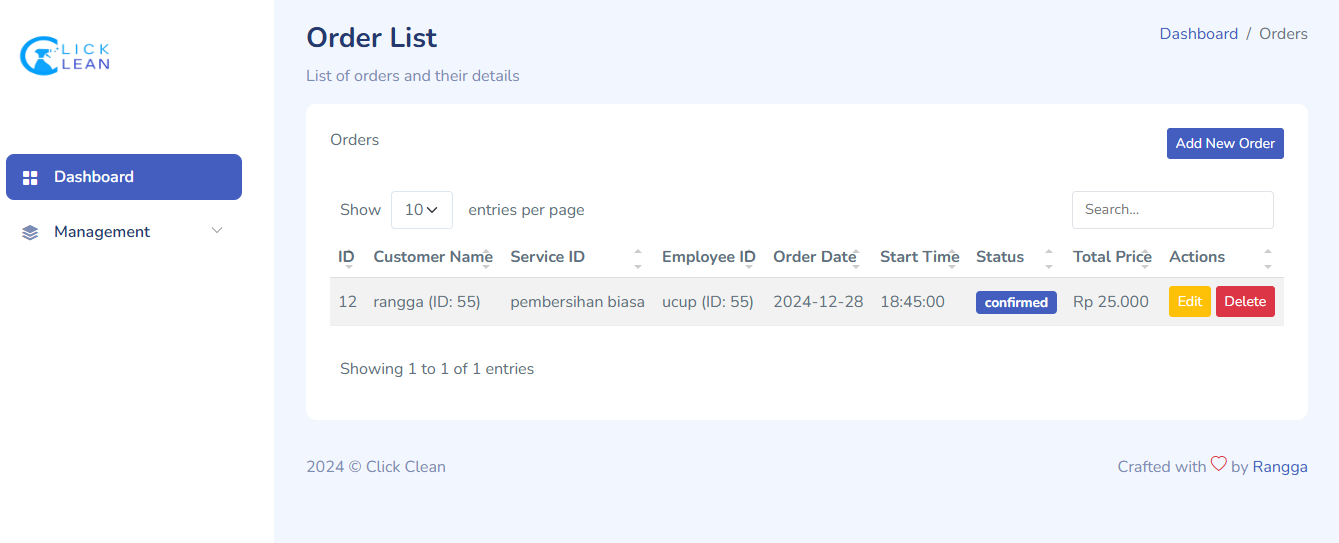
$this->orderModel->delete($id);

return redirect()->to('orders')->with('message', 'Order berhasil dihapus.');

}

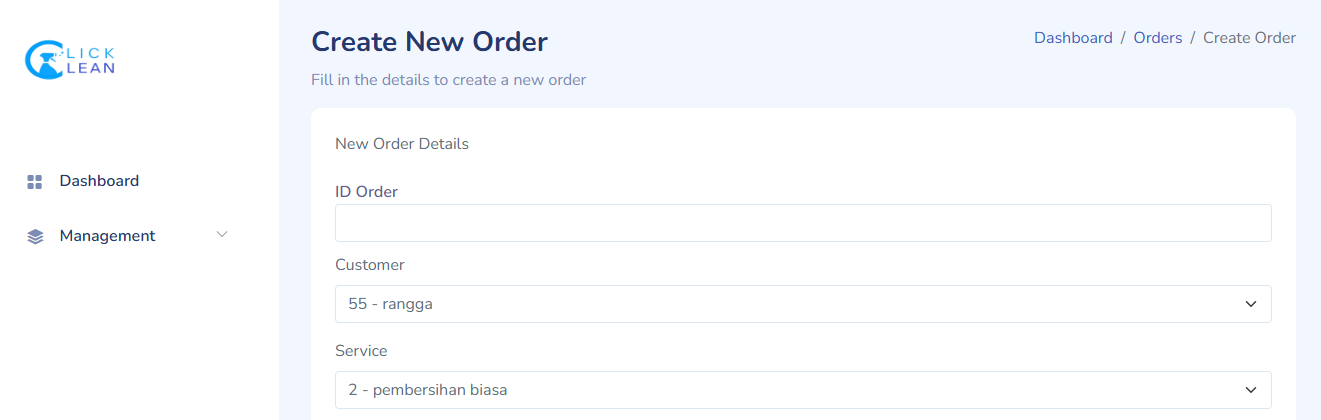
### View

1. **Index Order**



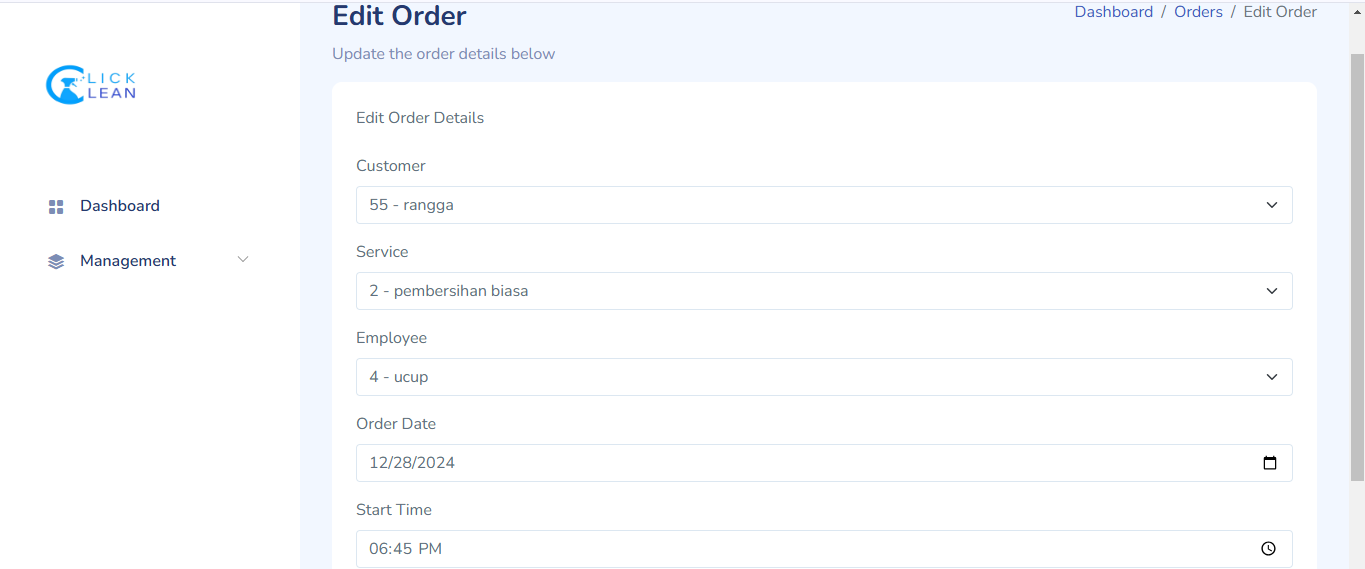
Gambar 1. Index Order

1. **Tambah Order**



Gambar 1. Tambah order

1. **Edit Order**

****

Gambar 1. Edit order

## **Payment**

### Model

<?php

namespace App\Models;

use CodeIgniter\*Model*;

*class* PaymentModel extends *Model*

{

protected $table = 'payments';

protected $primaryKey = 'id';

protected $allowedFields = ['order\_id','price', 'payment\_date', 'payment\_method', 'status'];

public *function* getPaymentById($id)

{

return $this->select('payments.\*, orders.id as order\_id, services.price as service\_price')

->join('orders', 'orders.id = payments.order\_id')

->join('services', 'services.id = orders.service\_id')

->find($id);

}

}

### Controller dan View

1. Menampilkan Data
2. Controller

<?php

namespace App\Controllers;

use App\Models\*PaymentModel*;

use App\Models\*OrdersModel*;

use App\Models\*ServiceModel*;

use CodeIgniter\*Controller*;

*class* PaymentsController extends *Controller*

{

protected $paymentModel;

protected $ordersModel;

protected $servicesModel;

public *function* \_\_construct()

{

$this->paymentModel = new *PaymentModel*();

$this->ordersModel = new *OrdersModel*();

$this->servicesModel = new *ServiceModel*();

}

// Menampilkan semua pembayaran

public *function* index()

{

// Mengambil semua data payment, termasuk informasi harga dari services

$payments = $this->paymentModel->getPayments();

$data['getStatusColor'] = [$this, 'getStatusColor'];

return view('payments/index', ['payments' => $payments], $data);

}

public *function* getStatusColor($status)

{

// Menentukan warna berdasarkan status

$colors = [

'pending' => 'warning',

'completed' => 'success',

'failed' => 'danger'

];

return $colors[$status] ?? 'secondary'; // Default 'secondary' jika status tidak ditemukan

}

1. View

<table class="table table-striped" id="paymentsTable">

<thead>

<tr>

<th>ID</th>

<th>Order ID</th>

<th>Amount</th>

<th>Payment Date</th>

<th>Payment Method</th>

<th>Status</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<?php if (!empty($payments) && is\_array($payments)): ?>

<?php foreach ($payments as $payment): ?>

<tr>

<td><?= esc($payment['id']) ?></td>

<td><?= esc($payment['order\_id']) ?></td>

<td>Rp <?= number\_format($payment['service\_price'], 0, ',', '.') ?></td>

<td><?= esc($payment['payment\_date']) ?></td>

<td><?= esc($payment['payment\_method']) ?></td>

<td>

<span class="badge bg-<?= isset($getStatusColor) ? $getStatusColor($order['status']) : 'secondary' ?>">

<?= esc($payment['status']) ?>

</span>

</td>

<td>

<a href="/payments/edit/<?= esc($payment['id']) ?>" class="btn btn-warning btn-sm">Edit</a>

<a href="/payments/delete/<?= esc($payment['id']) ?>" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to delete this payment?');">Delete</a>

</td>

</tr>

<?php endforeach; ?>

<?php else: ?>

<tr>

<td colspan="7" class="text-center">No payments found.</td>

</tr>

<?php endif; ?>

</tbody>

</table>

1. Menambahkan Data
2. Controller

public *function* create()

{

// Ambil data orders beserta harga dari tabel services

$orders = $this->ordersModel

->select('orders.id, orders.service\_id, services.name as service\_name, services.price as service\_price')

->join('services', 'services.id = orders.service\_id', 'left')

->findAll();

$data['orders'] = $orders;

return view('payments/create', $data);

}

// Menyimpan data pembayaran

public *function* store()

{

// Validasi input

$validation = $this->validate([

'order\_id'        => 'required|integer',

'payment\_date'    => 'required|valid\_date',

'payment\_method'  => 'required',

'status'          => 'required',

]);

if (!$validation) {

return redirect()->back()->withInput()->with('validation', \Config\*Services*::validation());

}

// Ambil data order berdasarkan ID

$order = $this->ordersModel->find($this->request->getPost('order\_id'));

if (!$order) {

return redirect()->back()->withInput()->with('error', 'Order tidak ditemukan.');

}

// Ambil service\_id dari order

$serviceId = $order['service\_id'];

// Ambil harga dari tabel services berdasarkan service\_id

$service = $this->servicesModel->find($serviceId);

if (!$service) {

return redirect()->back()->withInput()->with('error', 'Service tidak ditemukan.');

}

$price = $service['price']; // Harga diambil dari tabel services

// Simpan data ke tabel payments

$this->paymentModel->save([

'order\_id'        => $this->request->getPost('order\_id'),

'service\_id'      => $serviceId,

'price'           => $price, // Simpan harga dari service

'payment\_date'    => $this->request->getPost('payment\_date'),

'payment\_method'  => $this->request->getPost('payment\_method'),

'status'          => $this->request->getPost('status'),

]);

return redirect()->to('/payments')->with('success', 'Payment created successfully');

}

1. View

<section class="section">

<div class="card">

<div class="card-header">

Payment Details

</div>

<div class="card-body">

<?php if (isset($validation)): ?>

<div class="alert alert-danger">

<?= esc($validation->listErrors()); ?>

</div>

<?php endif; ?>

<form action="/payments/store" method="post">

<?= csrf\_field(); ?> <!-- Tambahkan CSRF untuk keamanan -->

<!-- Order ID -->

<select id="order\_id" name="order\_id" class="form-select" required>

<option value="" disabled selected>-- Select Order --</option>

<?php foreach ($orders as $order): ?>

<option value="<?= esc($order['id']) ?>" data-price="<?= esc($order['service\_price']) ?>">

Order ID: <?= esc($order['id']) ?> | Service: <?= esc($order['service\_name']) ?>

</option>

<?php endforeach; ?>

</select>

<!-- Amount -->

<!-- <div class="mb-3">

<label for="amount" class="form-label">Amount</label>

<input type="number" id="amount" name="amount" class="form-control" required step="0.01">

</div> -->

<!-- Payment Date -->

<div class="mb-3">

<label for="payment\_date" class="form-label">Payment Date</label>

<input type="date" id="payment\_date" name="payment\_date" class="form-control" required>

</div>

<!-- Payment Method -->

<div class="mb-3">

<label for="payment\_method" class="form-label">Payment Method</label>

<select id="payment\_method" name="payment\_method" class="form-select" required>

<option value="" disabled selected>-- Select Payment Method --</option>

<option value="cash">Cash</option>

<option value="credit\_card">Credit Card</option>

<option value="bank\_transfer">Bank Transfer</option>

</select>

</div>

<!-- Status -->

<div class="mb-3">

<label for="status" class="form-label">Status</label>

<select id="status" name="status" class="form-select" required>

<option value="" disabled selected>-- Select Status --</option>

<option value="pending">Pending</option>

<option value="completed">Completed</option>

<option value="failed">Failed</option>

</select>

</div>

<!-- Submit Button -->

<button type="submit" class="btn btn-primary">Create Payment</button>

<a href="/payments" class="btn btn-secondary">Cancel</a>

</form>

</div>

</div>

</section>

</div>

1. Mengubah Data
2. Controller

public *function* edit($id)

{

$data['payment'] = $this->paymentModel->getPaymentById($id);

$data['orders'] = $this->ordersModel

->select('orders.id, orders.service\_id, services.name as service\_name, services.price as service\_price')

->join('services', 'services.id = orders.service\_id', 'left')

->findAll();

if (empty($data['payment'])) {

throw new \CodeIgniter\Exceptions\*PageNotFoundException*('Payment not found');

}

return view('payments/edit', $data);

}

// Mengupdate data pembayaran

public *function* update($id)

{

// Validasi input

$validation = $this->validate([

'order\_id'        => 'required|integer',

'payment\_date'    => 'required|valid\_date',

'payment\_method'  => 'required',

'status'          => 'required',

]);

if (!$validation) {

return redirect()->back()->withInput()->with('validation', \Config\*Services*::validation());

}

// Ambil data order berdasarkan ID

$order = $this->ordersModel->find($this->request->getPost('order\_id'));

if (!$order) {

return redirect()->back()->withInput()->with('error', 'Order tidak ditemukan.');

}

// Ambil service\_id dari order

$serviceId = $order['service\_id'];

// Ambil harga dari tabel services berdasarkan service\_id

$service = $this->servicesModel->find($serviceId);

if (!$service) {

return redirect()->back()->withInput()->with('error', 'Service tidak ditemukan.');

}

$price = $service['price']; // Harga diambil dari tabel services

// Update data pembayaran

$this->paymentModel->update($id, [

'order\_id'        => $this->request->getPost('order\_id'),

'service\_id'      => $serviceId,

'price'           => $price, // Update harga jika perlu

'payment\_date'    => $this->request->getPost('payment\_date'),

'payment\_method'  => $this->request->getPost('payment\_method'),

'status'          => $this->request->getPost('status'),

]);

return redirect()->to('/payments')->with('success', 'Payment updated successfully');

}

1. View

<section class="section">

<div class="card">

<div class="card-header">

Payment Details

</div>

<div class="card-body">

<?php if (isset($validation)): ?>

<div class="alert alert-danger">

<?= esc($validation->listErrors()); ?>

</div>

<?php endif; ?>

<form action="/payments/update/<?= esc($payment['id']); ?>" method="post">

<?= csrf\_field(); ?> <!-- Tambahkan CSRF untuk keamanan -->

<!-- Order ID -->

<div class="mb-3">

<label for="order\_id" class="form-label">Order ID</label>

<select id="order\_id" name="order\_id" class="form-select" required>

<option value="" disabled>Select Order</option>

<?php foreach ($orders as $order): ?>

<option value="<?= esc($order['id']) ?>" <?= ($payment['order\_id'] == $order['id']) ? 'selected' : ''; ?>>

Order ID: <?= esc($order['id']) ?> | Service: <?= esc($order['service\_name']) ?>

</option>

<?php endforeach; ?>

</select>

</div>

<!-- Payment Date -->

<div class="mb-3">

<label for="payment\_date" class="form-label">Payment Date</label>

<input type="date" id="payment\_date" name="payment\_date" class="form-control" value="<?= esc($payment['payment\_date']); ?>" required>

</div>

<!-- Payment Method -->

<div class="mb-3">

<label for="payment\_method" class="form-label">Payment Method</label>

<select id="payment\_method" name="payment\_method" class="form-select" required>

<option value="cash" <?= ($payment['payment\_method'] == 'cash') ? 'selected' : ''; ?>>Cash</option>

<option value="credit\_card" <?= ($payment['payment\_method'] == 'credit\_card') ? 'selected' : ''; ?>>Credit Card</option>

<option value="bank\_transfer" <?= ($payment['payment\_method'] == 'bank\_transfer') ? 'selected' : ''; ?>>Bank Transfer</option>

</select>

</div>

<!-- Status -->

<div class="mb-3">

<label for="status" class="form-label">Status</label>

<select id="status" name="status" class="form-select" required>

<option value="pending" <?= ($payment['status'] == 'pending') ? 'selected' : ''; ?>>Pending</option>

<option value="completed" <?= ($payment['status'] == 'completed') ? 'selected' : ''; ?>>Completed</option>

<option value="failed" <?= ($payment['status'] == 'failed') ? 'selected' : ''; ?>>Failed</option>

</select>

</div>

<div class="d-flex justify-content-end">

<a href="/payments" class="btn btn-secondary me-2">Cancel</a>

<button type="submit" class="btn btn-primary">Update Payment</button>

</div>

</form>

</div>

</div>

</section>

</div>

1. Menghapus Data
2. Controller

public *function* delete($id)

{

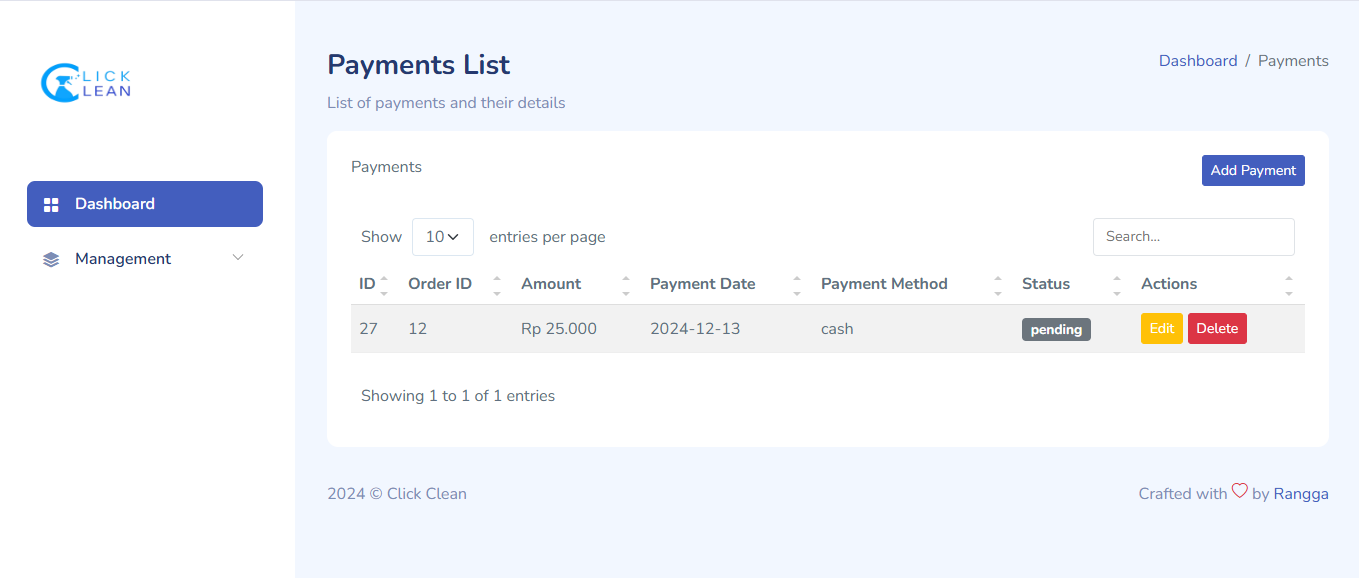
$this->paymentModel->delete($id);

return redirect()->to('/payments')->with('success', 'Payment deleted successfully');

}

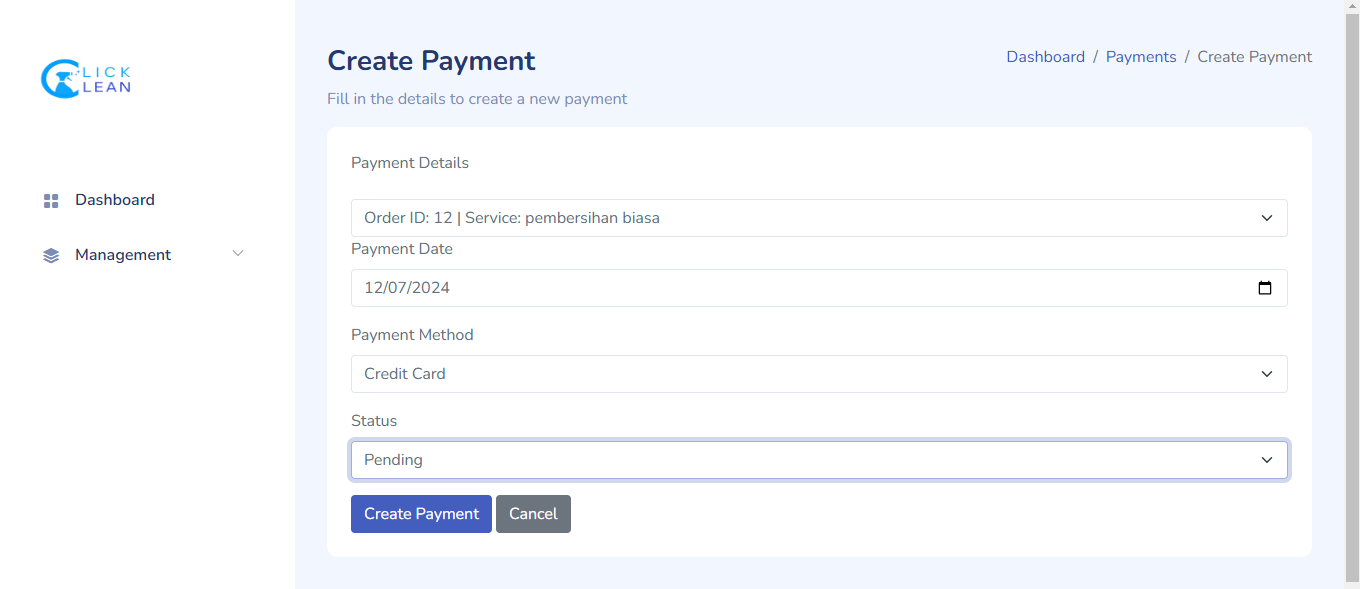
### View

1. **Index Status**



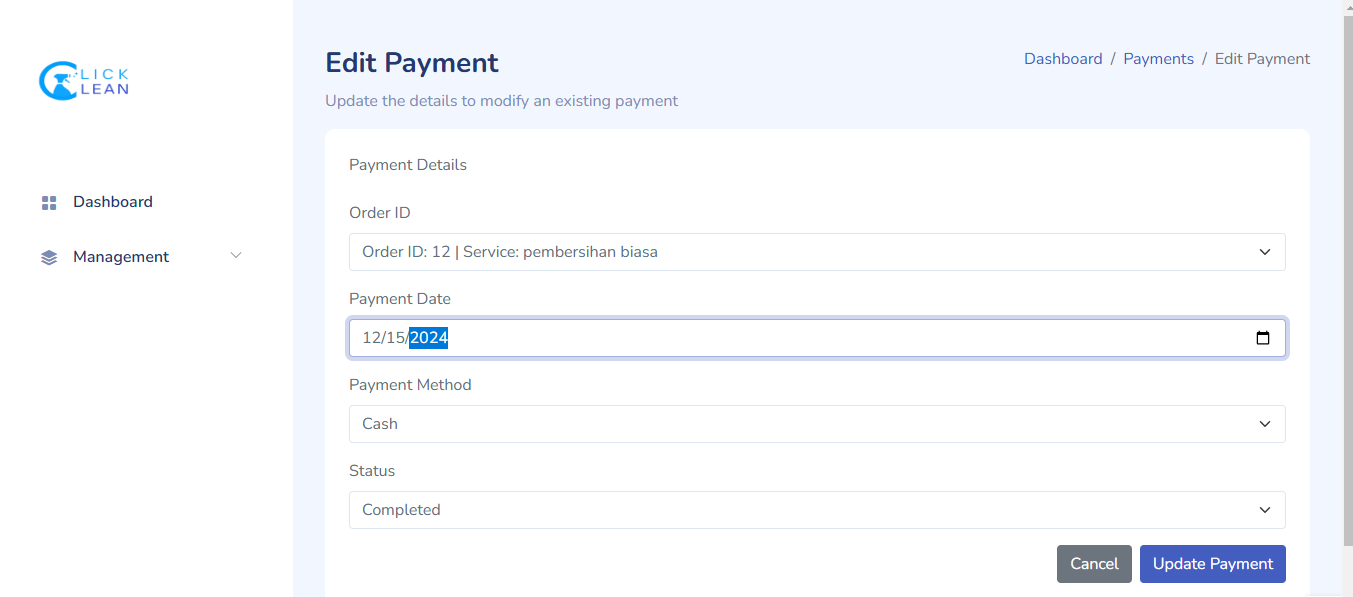
Gambar 1. Index Payment

1. **Tambah Status**



Gambar 1. Tambah Payment

1. **Edit Status**

****

Gambar 1. Edit Payment

# Blackbox Testing

Berisi testing dari fungsionalitas yang ada pada aplikasi berikut format tabelnya

| **Fungsionalitas** | **Skenario Uji** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Keterangan** |
| --- | --- | --- | --- | --- |
| **Menambah Data Pelanggan** | Tidak mengisi input apa pun, klik tombol Simpan | Muncul pesan error "Field wajib diisi". | Muncul pesan error | Valid/Sesuai |
|  | Mengisi semua input dengan data valid | Data pelanggan berhasil disimpan, muncul pesan sukses. | Data pelanggan berhasil disimpan | Valid/Sesuai |
|  | Mengisi input dengan data tidak valid (contoh: email salah format) | Muncul pesan error "Input tidak valid". | Muncul pesan error | Valid/Sesuai |
| **Menghapus Data Pelanggan** | Klik tombol Hapus pada pelanggan tertentu | Data pelanggan berhasil dihapus, muncul pesan sukses. | Data pelanggan berhasil dihapus | Valid/Sesuai |
|  | Menghapus pelanggan yang tidak ada di database | Muncul pesan error "Data tidak ditemukan". | Muncul pesan error | Valid/Sesuai |
| **Mengedit Data Pelanggan** | Mengubah data dengan input valid, klik Simpan | Perubahan disimpan, muncul pesan sukses. | Perubahan disimpan | Valid/Sesuai |
|  | Mengubah data dengan input tidak valid (contoh: email salah format) | Muncul pesan error "Input tidak valid". | Muncul pesan error | Valid/Sesuai |
| **Menambah Data Pesanan** | Tidak mengisi input apa pun, klik tombol Simpan | Muncul pesan error "Field wajib diisi". | Muncul pesan error | Valid/Sesuai |
|  | Mengisi semua input dengan data valid | Pesanan berhasil ditambahkan, muncul pesan sukses. | Pesanan berhasil ditambahkan | Valid/Sesuai |
|  | Memasukkan Customer ID yang tidak ada di database | Muncul pesan error "Customer tidak ditemukan". | Muncul pesan error | Valid/Sesuai |
| **Menghapus Data Pesanan** | Klik tombol Hapus pada pesanan tertentu | Pesanan berhasil dihapus, muncul pesan sukses. | Pesanan berhasil dihapus | Valid/Sesuai |
|  | Menghapus pesanan yang tidak ada di database | Muncul pesan error "Pesanan tidak ditemukan". | Muncul pesan error | Valid/Sesuai |
| **Menambah Pembayaran** | Tidak mengisi input apa pun, klik tombol Simpan | Muncul pesan error "Field wajib diisi". | Muncul pesan error | Valid/Sesuai |
|  | Mengisi semua input dengan data valid | Pembayaran berhasil ditambahkan, muncul pesan sukses. | Pembayaran berhasil ditambahkan | Valid/Sesuai |
|  | Memasukkan Order ID yang tidak ada di database | Muncul pesan error "Pesanan tidak ditemukan". | Muncul pesan error | Valid/Sesuai |
| **Menghapus Pembayaran** | Klik tombol Hapus pada pembayaran tertentu | Pembayaran berhasil dihapus, muncul pesan sukses. | Pembayaran berhasil dihapus | Valid/Sesuai |
|  | Menghapus pembayaran yang tidak ada di database | Muncul pesan error "Pembayaran tidak ditemukan". | Muncul pesan error | Valid/Sesuai |
| **Mengelola Status Pesanan** | Menambah status dengan nama valid | Status berhasil ditambahkan, muncul pesan sukses. | Status berhasil ditambahkan | Valid/Sesuai |
|  | Menambah status dengan nama kosong | Muncul pesan error "Field wajib diisi". | Muncul pesan error | Valid/Sesuai |
|  | Menghapus status yang sudah digunakan | Muncul pesan error "Status tidak dapat dihapus". | Muncul pesan error | Valid/Sesuai |
| **Menambah Penugasan** | Menambah penugasan dengan Order ID dan Employee ID valid | Penugasan berhasil dibuat, muncul pesan sukses. | Penugasan berhasil dibuat | Valid/Sesuai |
|  | Mengisi Order ID atau Employee ID yang tidak valid | Muncul pesan error "Data tidak valid". | Muncul pesan error | Valid/Sesuai |

# Daftar Pustaka

1. CodeIgniter Foundation. (n.d.). *CodeIgniter 4 User Guide*. <https://codeigniter.com/user_guide/>
2. PHP Group. (n.d.). *PHP Documentation*.

<https://www.php.net/docs.php>

1. Oracle Corporation. (n.d.). *MySQL Documentation*.

<https://dev.mysql.com/doc/>

1. Bootstrap. (n.d.). *Bootstrap Documentation*.

<https://getbootstrap.com/docs/>

1. GitHub. (n.d.). *GitHub Documentation*.

<https://github.com/>

1. Stack Overflow. (n.d.). *Programming Q&A Community*.

<https://stackoverflow.com/>

1. Link repository github Project

https://github.com/Rgg-hdm/ClickClean