**LAPORAN PRAKTIKUM INSERT DATA**



**PEMROGRAMAN WEB 2**

Disusun Oleh :

**Dio Ananda Maulana Wijaya (4123031)**

Dosen Pengampu :

**Muhammad Miftakhul Syaikhuddin, S.Kom., M.Kom**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNVERSITAS PESANTREN TINGGI DARUL ‘ULUM**

**JOMBANG**

**2025**

1. **Pendahuluan**

Praktikum ini bertujuan untuk membuat aplikasi sederhana pengelolaan data buku menggunakan framework CodeIgniter 4. Fitur utama yang dibuat meliputi pembuatan, pembacaan, pengubahan, dan penghapusan data (CRUD) pada halaman Books. Selain itu, juga dilakukan upload gambar sampul buku dengan validasi, serta pembuatan tampilan yang responsif dan rapi.

1. **Tujuan Praktikum**
2. Mahasiswa memahami struktur MVC pada CodeIgniter 4.
3. Mahasiswa mampu membuat fitur CRUD data buku dengan validasi.
4. Mahasiswa mampu menggunakan route, controller, model, dan view.
5. Mahasiswa mampu menerapkan flashdata untuk feedback ke pengguna.
6. Mahasiswa dapat menyelesaikan tantangan unik: validasi data ganda.
7. **Tools yang digunakan**

* CodeIgniter 4
* PHP 8.2
* XAMPP (Apache & MySQL)
* VS Code
* phpMyAdmin
* Bootstrap 5 (untuk tampilan UI)

1. **Struktur Folder Proyek**

1. Model

* app/Models/BooksModel.php  
  Model untuk mengakses data tabel books dengan field yang diperbolehkan diisi.

2. Controller

* app/Controllers/Books.php  
  Controller utama yang menangani semua method CRUD (index, create, save, detail, edit, update, delete).

3. Views (app/Views/books/)

* index.php: Menampilkan daftar buku dengan gambar sampul dan tombol Detail.
* create.php: Form tambah buku dengan upload gambar sampul dan validasi input.
* detail.php: Menampilkan detail buku dengan gambar yang dapat diperbesar, serta tombol Edit dan Hapus.
* edit.php: Form edit buku dengan upload gambar baru dan validasi.

4. Routes

Di /app/Config/Routes.php terdapat routing untuk semua method CRUD pada controller Books.

5. CSS  
public/assets/css/style.css berisi styling tampilan halaman agar responsif dan menarik.

1. **Struktur Database**

Nama Database: **ci4**

Nama Tabel: **books**

Kolom:

**id** (INT, Primary Key, Auto Increment)

**judul** (VARCHAR)

**slug** (VARCHAR)

**penulis** (VARCHAR)

**penerbit** (VARCHAR)

**sampul** (VARCHAR)

**created\_at, updated\_at** (TIMESTAMP)

1. **Konfigurasi Koneksi Database**

(app/Config/Database.php)

public array $default = [

'hostname' => 'localhost',

'username' => 'root',

'password' => '',

'database' => 'ci4',

'DBDriver' => 'MySQLi',

'DBDebug' => true,

'charset' => 'utf8mb4',

'DBCollat' => 'utf8mb4\_general\_ci',

'port' => 3306,

];tu

1. **Routing**

(app/Config/Routes.php)

$routes->get('/books', 'Books::index');

$routes->get('/books/create', 'Books::create');

$routes->post('/books/save', 'Books::save');

$routes->get('/books/edit/(:segment)', 'Books::edit/$1');

$routes->post('/books/update/(:num)', 'Books::update/$1');

$routes->delete('/books/(:num)', 'Books::delete/$1');

$routes->get('/books/(:segment)', 'Books::detail/$1');

1. **Penjelasan Controller**

(app/Controllers/Books.php)

1. **index()**

public function index()

{

$data = [

'title' => 'Daftar Buku',

'buku' => $this->bukuModel->findAll()

];

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

}

**Menampilkan semua data buku dalam bentuk tabel.**

1. **create()**

public function create()

{

$data = [

'title' => 'Form Tambah Buku',

'validation' => \Config\Services::validation()

];

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

}

**Menampilkan form input data buku baru.**

1. **save()**

public function save()

{

if (!$this->validate([

'judul' => 'required|is\_unique[books.judul]',

'penulis' => 'required|is\_unique[books.penulis]',

'penerbit' => 'required|is\_unique[books.penerbit]',

])) {

return redirect()->to('/books/create')->withInput()->with('validation', $this->validator);

}

$slug = url\_title($this->request->getVar('judul'), '-', true);

$this->bukuModel->save([

'judul' => $this->request->getVar('judul'),

'slug' => $slug,

'penulis' => $this->request->getVar('penulis'),

'penerbit' => $this->request->getVar('penerbit'),

'sampul' => $this->request->getVar('sampul') ?: 'default.jpg'

]);

session()->setFlashdata('pesan', 'Data buku berhasil ditambahkan.');

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

}

**Melakukan penyimpanan data buku baru ke database.**

1. **detail($slug)**

public function detail($slug)

{

$buku = $this->bukuModel->getBuku($slug);

if (empty($buku)) {

throw new \CodeIgniter\Exceptions\PageNotFoundException("Buku $slug tidak ditemukan.");

}

$data = [

'title' => 'Detail Buku',

'buku' => $buku

];

return view('books/detail', $data);

}

**Menampilkan detail informasi buku berdasarkan slug.**

1. **edit($slug) & update($id)**

public function edit($slug)

{

$data = [

'title' => 'Form Ubah Buku',

'validation' => \Config\Services::validation(),

'buku' => $this->bukuModel->getBuku($slug)

];

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

}

public function update($id)

{

$bukuLama = $this->bukuModel->getBuku($this->request->getVar('slug'));

$judulRule = ($bukuLama['judul'] == $this->request->getVar('judul')) ? 'required' : 'required|is\_unique[books.judul]';

$penulisRule = ($bukuLama['penulis'] == $this->request->getVar('penulis')) ? 'required' : 'required|is\_unique[books.penulis]';

$penerbitRule = ($bukuLama['penerbit'] == $this->request->getVar('penerbit')) ? 'required' : 'required|is\_unique[books.penerbit]';

if (!$this->validate([

'judul' => $judulRule,

'penulis' => $penulisRule,

'penerbit' => $penerbitRule

])) {

return redirect()->to('/books/edit/' . $this->request->getVar('slug'))->withInput()->with('validation', $this->validator);

}

$slug = url\_title($this->request->getVar('judul'), '-', true);

$this->bukuModel->save([

'id' => $id,

'judul' => $this->request->getVar('judul'),

'slug' => $slug,

'penulis' => $this->request->getVar('penulis'),

'penerbit' => $this->request->getVar('penerbit'),

'sampul' => $this->request->getVar('sampul')

]);

session()->setFlashdata('pesan', 'Data buku berhasil diubah.');

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

}

**Digunakan untuk mengedit dan menyimpan perubahan data buku.**

1. **delete($id)**

public function delete($id)

{

$buku = $this->bukuModel->find($id);

if ($buku['sampul'] != 'default.jpg') {

$path = FCPATH . 'img/' . $buku['sampul'];

if (is\_file($path)) {

@unlink($path);

}

}

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

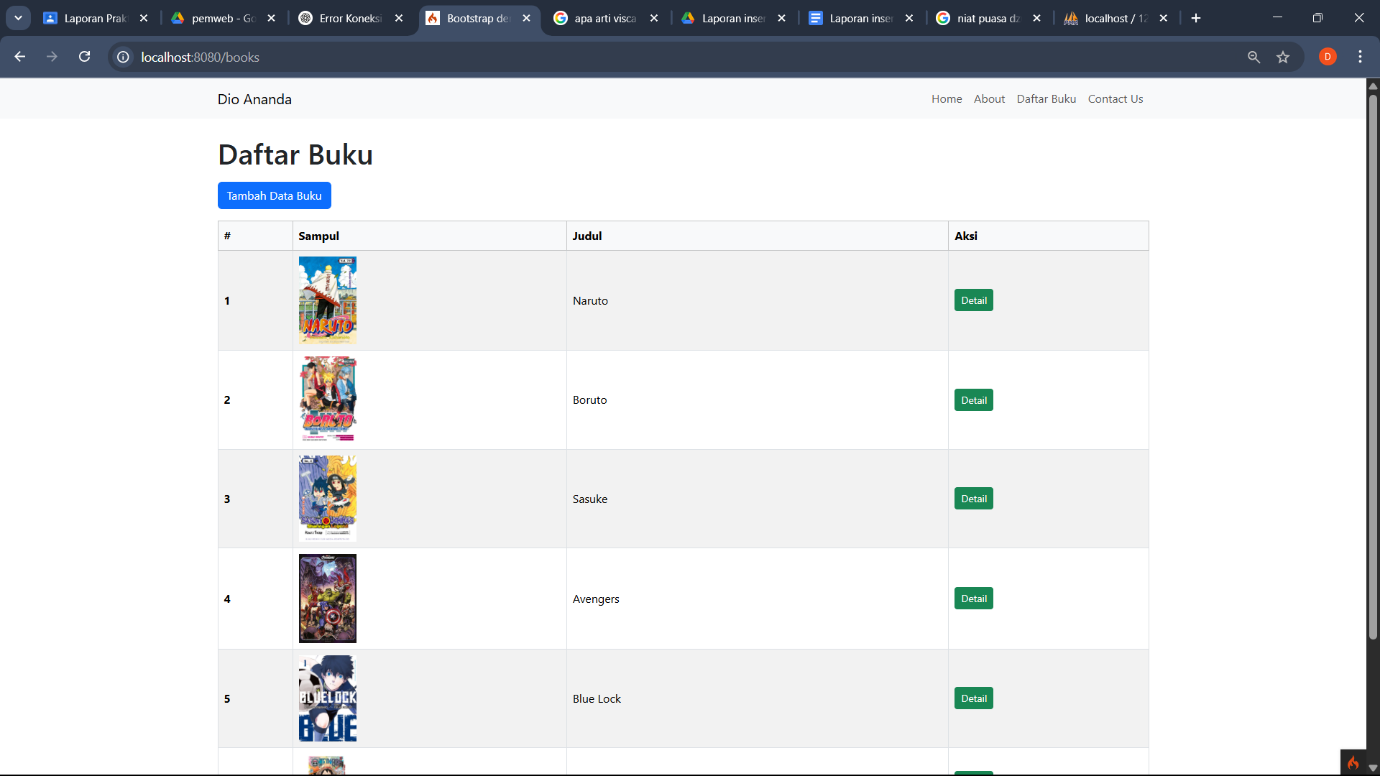
session()->setFlashdata('pesan', 'Data buku berhasil dihapus.');

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

}

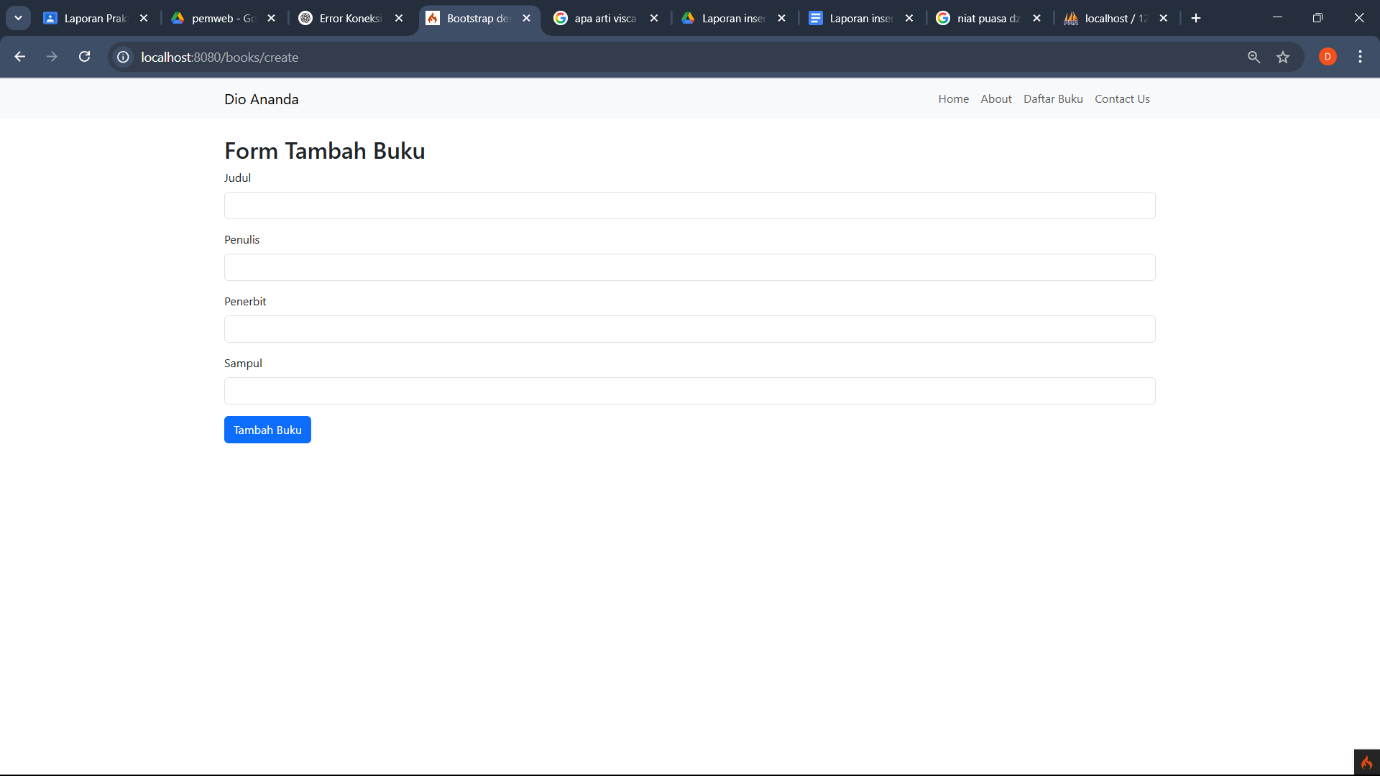
**Menghapus data buku dan juga file gambar jika bukan default.**

1. **Hasil Praktikum**
2. Halaman Daftar Buku (/books)



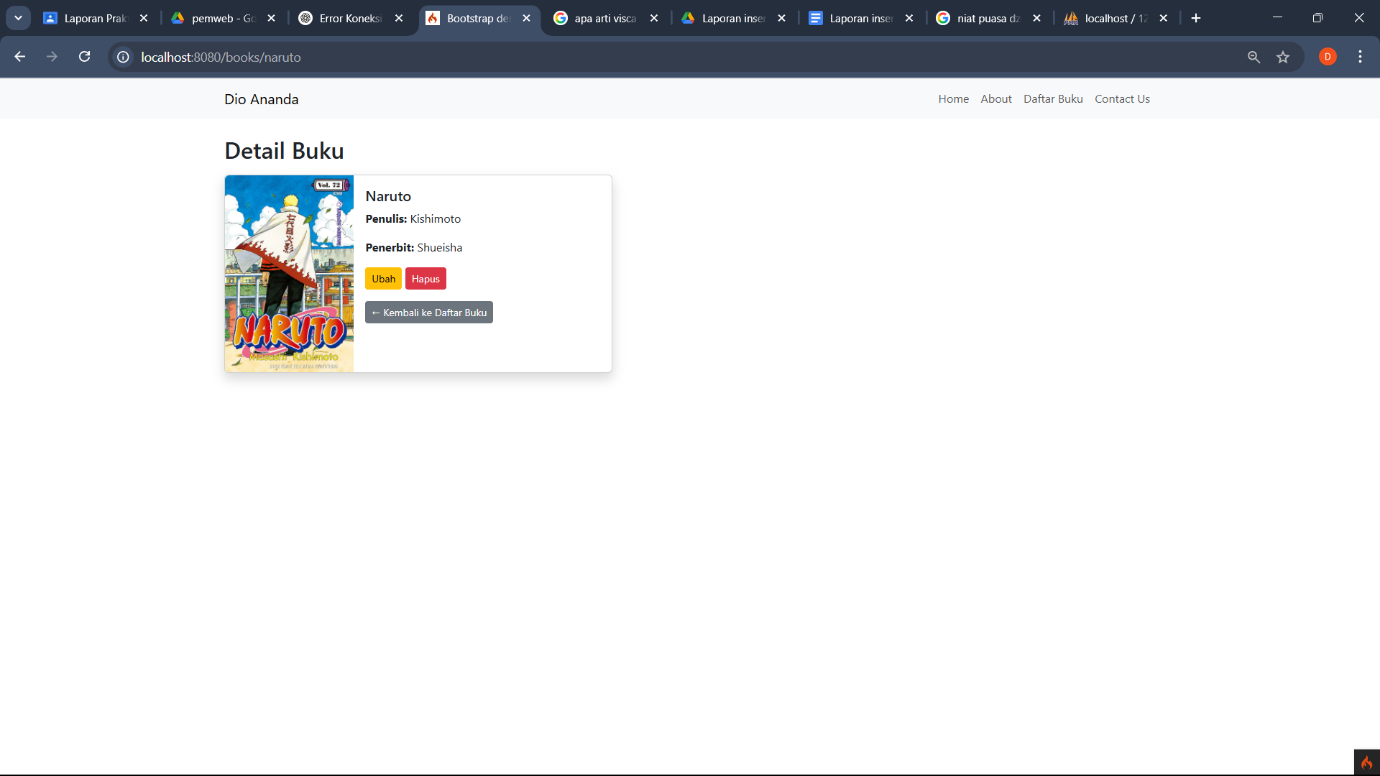
Menampilkan tabel buku dengan gambar sampul thumbnail dan tombol Detail, serta tombol Tambah Data di bawah tabel.

1. Halaman Tambah Buku (/books/create)

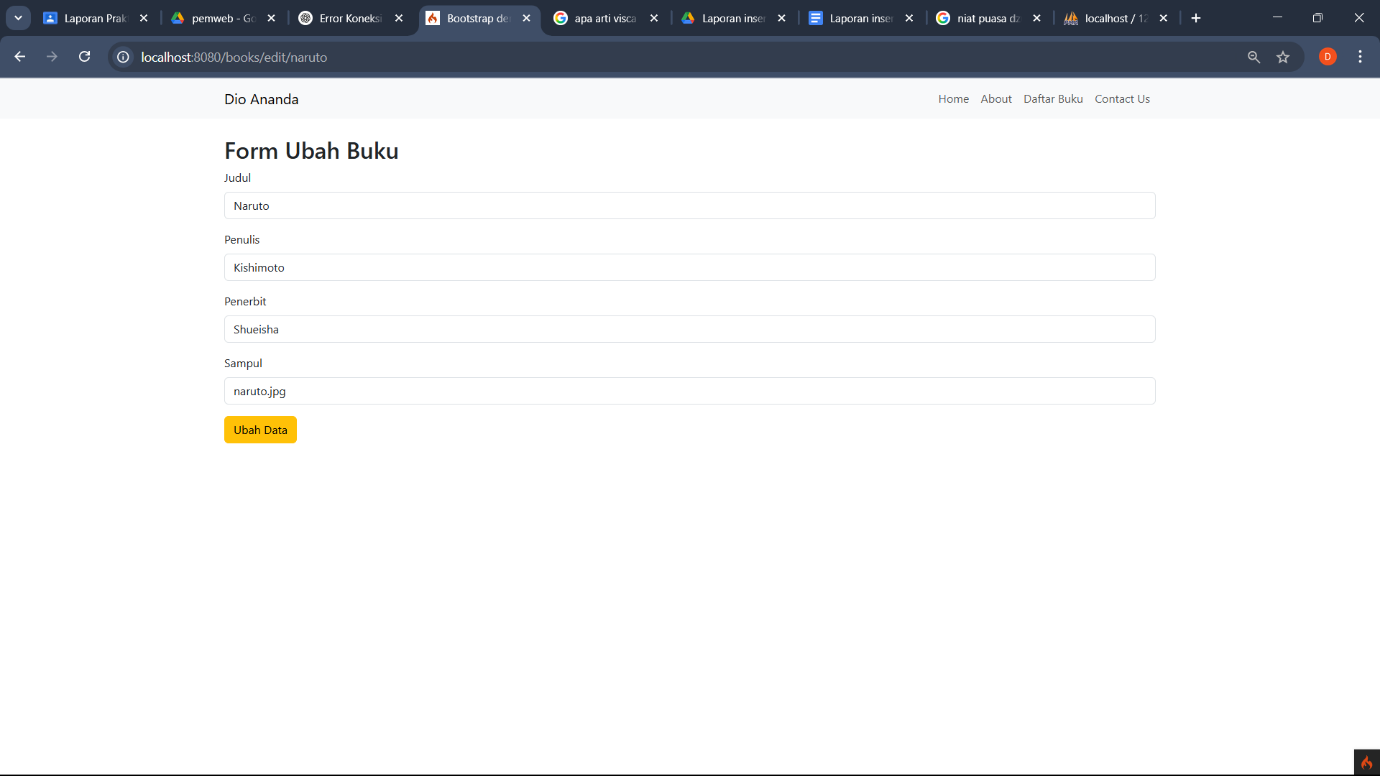


Form untuk input data buku baru dan upload gambar sampul dengan validasi.

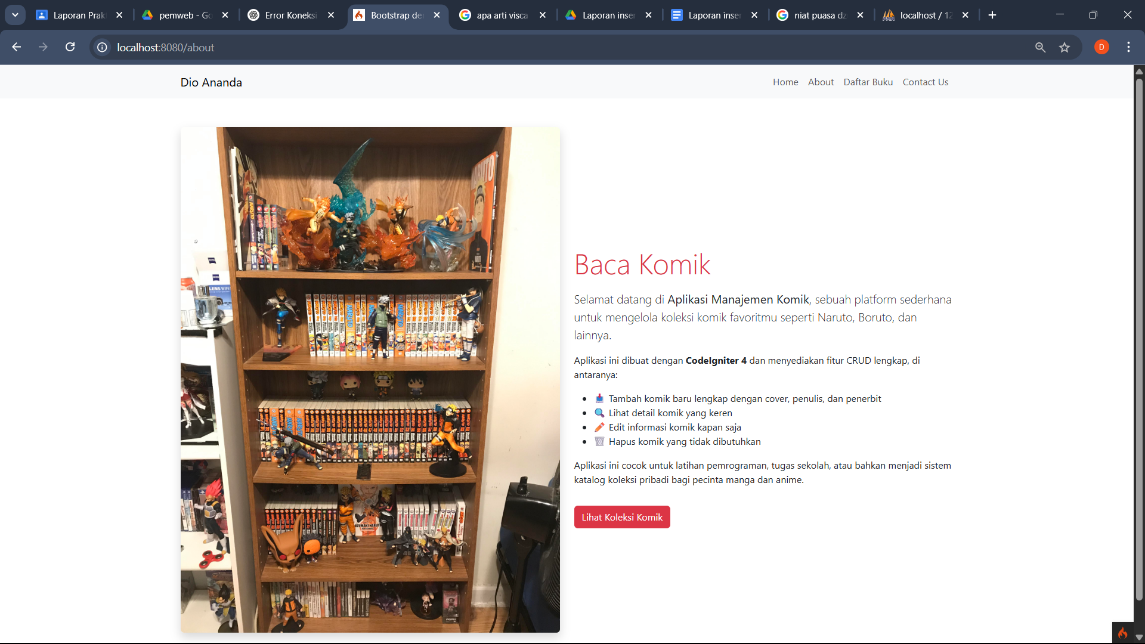
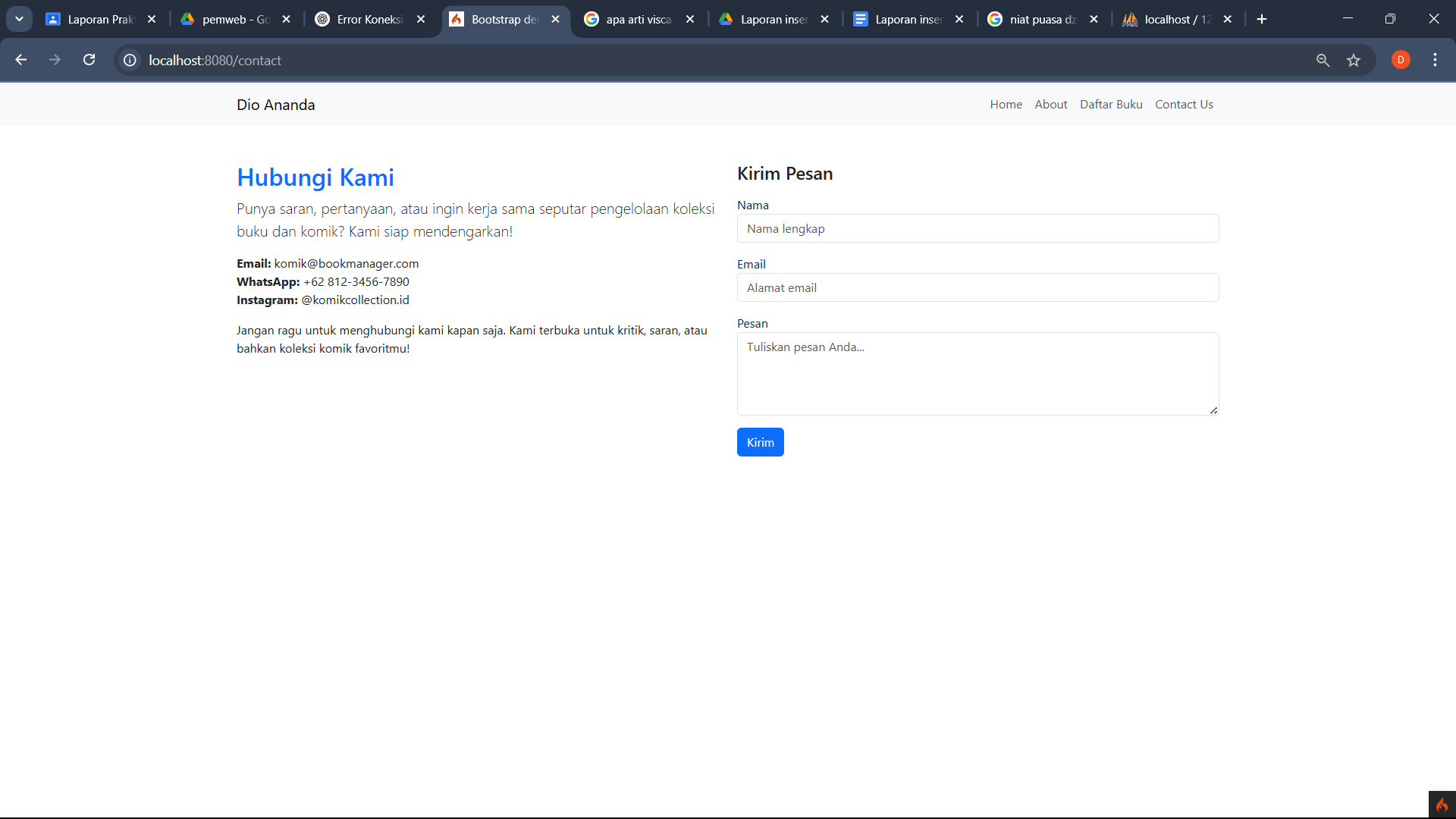
1. Halaman Detail Buku (/books/detail/slug)



Menampilkan gambar sampul besar yang bisa diperbesar dengan klik, serta info judul, penulis, penerbit, tombol Edit (kuning) dan Hapus (merah).

1. Halaman Edit Buku (/books/edit/slug)

Form edit data buku dan upload gambar sampul baru dengan validasi.

1. About
2. Contact Us
3. **Kesimpulan**

Aplikasi CRUD buku ini berhasil dibuat sesuai dengan struktur dan materi pada PPT praktikum. Semua fungsi CRUD dapat berjalan dengan baik, dilengkapi dengan validasi, flashdata, pengecekan error, dan struktur kode yang sesuai standar MVC. Soal tantangan tentang duplikasi data juga telah diselesaikan dengan validasi is unique.

Link Github :

https://github.com/dioannda/DioAnanda\_Pemweb2/tree/main/Praktikum%20Insert%20Data