

# **LAPORAN PRAKTIKUM PEMROGRAMAN WEB 2**

## **PERTEMUAN 6**

### **CRUD**



Disusun Oleh:

Wildan Dzaky Ramadhani

22/505766/SV/21917

Dosen Pengampu:

Dinar Nugroho Pratomo, S.Kom., M.IM., M.Cs.

Faza Maula Azif, S.Kom., M.Eng.

**PROGRAM STUDI D4 TEKNOLOGI REKAYASA PERANGKAT LUNAK**

**DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS GADJAH MADA**

**YOGYAKARTA**

**2023**

## DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	3
1.1. Latar Belakang.....	3
1.2. Rumusan Masalah.....	3
BAB II.....	4
2.1. CRUD.....	4
BAB III.....	5
3.1. Menambah Data Buku.....	5
3.2. Membuat Tombol Tambah Buku.....	5
3.3. Membuat Form Tambah Buku.....	6
3.4. Menangani Proses Simpan Data Buku.....	7
3.5. Membuat Route Hapus Buku.....	9
3.6. Membuat Tombol Hapus.....	9
3.7. Menangani Proses Hapus Data Buku.....	10
4.1. Tugas Praktikum.....	11
BAB IV.....	14
4.1. Kesimpulan.....	14

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Dalam pengembangan aplikasi web, konsep CRUD (Create, Read, Update, Delete) adalah dasar yang sangat penting karena memungkinkan pengelolaan data yang efisien. CRUD melibatkan operasi dasar seperti membuat data baru, membaca data yang ada, memperbarui data, dan menghapus data dari database. Di dunia pengembangan aplikasi web, Laravel sering menjadi pilihan utama karena menyediakan alat-alat yang kuat dan efisien untuk menerapkan operasi CRUD dengan mudah.

Laporan ini akan menjelaskan bagaimana konsep CRUD diintegrasikan dalam pengembangan aplikasi web berbasis Laravel. Kami akan memfokuskan perhatian pada peran kelas Eloquent sebagai model dalam Laravel, yang sangat penting dalam mengelola data. Dengan bantuan Eloquent, kami akan menunjukkan bagaimana Kita dapat membuat, membaca, memperbarui, dan menghapus data dalam aplikasi web Kita. Kami juga akan menunjukkan bagaimana data ini dapat disajikan kepada pengguna melalui antarmuka yang responsif.

Praktikum ini bertujuan untuk memberikan pengalamazan praktis dalam menerapkan konsep CRUD dalam pengembangan aplikasi web. Melalui langkah-langkah yang diajarkan, pembaca akan mendapatkan pemahaman yang kuat tentang penggunaan Laravel dan peran penting CRUD dalam mengelola data dalam aplikasi. Hasilnya adalah kemampuan untuk membangun aplikasi web yang lebih efisien dan responsif dalam berbagai konteks pengembangan.

### **1.2. Rumusan Masalah**

- A. Apa yang dimaksud dengan CRUD?
- B. Bagaimana cara penerapan CRUD pada suatu project?

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. CRUD**

CRUD merupakan akronim dari "Create, Read, Update, Delete," dan merupakan konsep yang mendasar dalam pengelolaan data dalam pengembangan perangkat lunak. Konsep CRUD merinci empat operasi dasar yang digunakan untuk mengelola data dalam sebuah aplikasi:

##### **1. Create**

Operasi Create adalah langkah pertama dalam konsep CRUD. Ini berkaitan dengan pembuatan data baru yang akan disimpan dalam database. Create memungkinkan pengguna atau sistem untuk menambahkan informasi baru ke dalam aplikasi. Sebagai contoh, dalam aplikasi e-commerce, operasi Create digunakan untuk menambahkan produk baru ke dalam daftar produk yang tersedia.

##### **2. Read**

Operasi Read adalah langkah kedua dalam konsep CRUD. Ini melibatkan pengambilan data yang sudah ada dari database. Read penting karena memungkinkan pengguna atau aplikasi untuk melihat dan mengakses data yang telah diinput sebelumnya. Sebagai contoh, dalam aplikasi media sosial, operasi Read memungkinkan pengguna untuk melihat kiriman teman mereka.

##### **3. Update**

Operasi Update adalah langkah ketiga dalam konsep CRUD. Ini digunakan untuk mengubah data yang sudah ada dalam database. Update sangat penting untuk memungkinkan aplikasi mengikuti perubahan. Misalnya, dalam aplikasi perbankan, operasi Update digunakan untuk mengubah saldo akun pengguna setelah mereka melakukan transaksi.

##### **4. Delete**

Operasi Delete adalah langkah terakhir dalam konsep CRUD. Ini digunakan untuk menghapus data yang sudah ada dalam database. Data yang dihapus akan dihilangkan secara permanen dari sistem. Delete digunakan untuk menghapus data yang sudah tidak relevan atau diinginkan dalam aplikasi. Dalam aplikasi e-commerce, operasi Delete digunakan untuk menghapus produk yang sudah tidak dijual lagi.

## BAB III

### DOKUMENTASI PRAKTIKUM

#### 3.1. Menambah Data Buku

Dalam melakukan *Create*, Langkah pertama yang dapat dilakukan adalah dengan membuat Route terlebih dahulu pada folder Laravel yang telah dibuat sebelumnya. Masukkan Langkah untuk menambahkan route berikut pada file web.php untuk menangani tambah buku:

```
Route::get('/buku', [BukuController::class, 'index']);  
Route::get('/buku/create', [BukuController::class, 'create'])->name('buku.create');  
Route::post('/buku', [BukuController::class, 'store'])->name('buku.store');
```

**Route::get('/buku', [BukuController::class, 'index']);** Route ini mengarahkan ke metode index dalam BukuController, dimana route ini digunakan untuk menampilkan daftar buku dan biasanya terkait dengan tampilan indeks (list) buku.

**Route::get('/buku/create', [BukuController::class, 'create'])->name('buku.create');** Route ini juga mengarahkan ke metode create dalam BukuController, dimana route ini digunakan untuk menampilkan formulir pembuatan buku baru dan diberi nama route dengan alias “buku.create” untuk kemudahan penggunaan.

**Route::post('/buku', [BukuController::class, 'store'])->name('buku.store');** Route ini juga mengarahkan ke metode store dalam BukuController, dimana route ini digunakan untuk mengirimkan data formulir buku baru yang dibuat oleh pengguna. Nama route “buku.store” digunakan untuk memproses penyimpanan data buku yang dikirim melalui formulir.

Dengan pengaturan route seperti ini, kita dapat mengakses halaman untuk menampilkan daftar buku, membuat buku baru, dan menyimpan data buku baru dalam aplikasi web kita.

#### 3.2. Membuat Tombol Tambah Buku

Tombol Tambah Buku yang akan kita buat merupakan sebuah tombol yang apabila di klik akan mengarahkan kita ke form tambah buku. Berikut merupakan implementasinya, pertama kita buka file **buku.blade.php** lalu masukkan kode berikut:

```
<p align="left"><a href="{{ route('buku.create') }}">Tambah Buku</a></p>
```

Kode diatas digunakan untuk membuat tautan dengan teks “Tambah buku” ke dalam halaman web. Ketika pengguna mengklik tautan ini, mereka akan diarahkan ke route dengan alias “buku.create” dalam aplikasi ini. Langkah selanjutnya adalah kita buka file **BukuController.php**, tambahkan function index() seperti pada gambar berikut:

```
1 reference | 0 overrides
public function create(){
    return view('create');
}
```

Fungsi create() digunakan untuk menampilkan tampilan yang disebut create kepada pengguna. Tampilan ini biaanya berisi formular yang memungkinkan pengguna untuk menginput informasi yang diperlukan untuk membuat data baru. Fungsi ini bertanggung jawab untuk menampilkan halaman pembuatan data baru kepada pengguna Ketika mereka mengaksesnya melalui route. Tambahkan perintah tersebut di bawah fungsi index() pada BukuController. Fungsi ini akan mengarahkan pengguna ke tampilan create.blade.php.

### 3.3. Membuat Form Tambah Buku

Pada Langkah sebelumnya telah disebutkan bahwa file create.blade.php digunakan untuk menampilkan tampilan tombol tambah buku. Perintah yang dituliskan adalah sebagai berikut:

```
@extends('layouts.main')

@section('content')
    <div class="container">
        <h4>Tambah Buku</h4>
        <form method="post" action="{{ route('buku.store') }}">
            @csrf
            <div>Judul <input type="text" name="judul"></div>
            <div>Penulis <input type="text" name="penulis"></div>
            <div>Harga <input type="number" name="harga"></div>
            <div>Tgl. Terbit <input type="date" name="tgl_terbit"></div>
            <div><button type="submit">Simpan</button></div>
            <a href="/buku">Batal</a>
        </form>
    </div>
@endsection
```

Kode diatas digunakan untuk membuat buku baru. Halaman ini berisi sebuah formulir yang memungkinkan pengguna untuk menginputkan informasi seperti judul, penulis, harga, dan tanggal terbit buku. Ketika pengguna mengisi formulir dan mengklik tombol "Simpan", data yang dimasukkan akan dikirimkan ke rute dengan nama "buku.store" untuk diproses dan disimpan dalam database. Juga, terdapat tautan "Batal" yang memungkinkan pengguna kembali ke halaman daftar buku jika mereka memutuskan untuk tidak menyimpan data baru.

Halaman ini mengextend layout utama layouts.main. Formulir ini dijaga juga oleh token CSRF yang dihasilkan oleh @csrf untuk menjaga keamanan dalam mengirim data ke server. Pastikan sebelumnya telah membuat file **main.blade.php** yang digunakan sebagai tempat penulisan kode pembuatan header dan footer. Berikut adalah isi file pada **main.blade.php**:

```
<!DOCTYPE html>
<html>
<head>
|   <title>@yield('title')</title>
</head>
<body>
|   <header>
|   |   <h1>Data Buku</h1>
|   </header>
|
|   <nav>
|   |   <ul>
|   |   |   <li><a href="/">Beranda</a></li>
|   |   |   <li><a href="/tentang">Buku</a></li>
|   |   |   <li><a href="/kontak">Pengurus</a></li>
|   |   </ul>
|   </nav>
|
|   <main>
|   |   @yield('content')
|   </main>
|
|   <footer>
|   |   &copy; {{ date('Y') }} perpustakaanbuku.co
|   </footer>
</body>
</html>
```

### 3.4. Menangani Proses Simpan Data Buku

Pada bab 3.1 sudah dituliskan route untuk function store(). Store digunakan untuk tempat penyimpanan data yang telah dituliskan saat menambahkan data buku. Sama seperti dengan Create, perlu adanya penambahan perintah pada bagian BukuController untuk memproses fungsi store(). Tambahkan perintah berikut dibawah function create():

```

1 reference | 0 overrides
public function store(Request $request) {
    $buku = new Buku;
    $buku->judul = $request->judul;
    $buku->penulis = $request->penulis;
    $buku->harga = $request->harga;
    $buku->tgl_terbit = $request->tgl_terbit;
    $buku->save();
    return redirect('/buku');
}

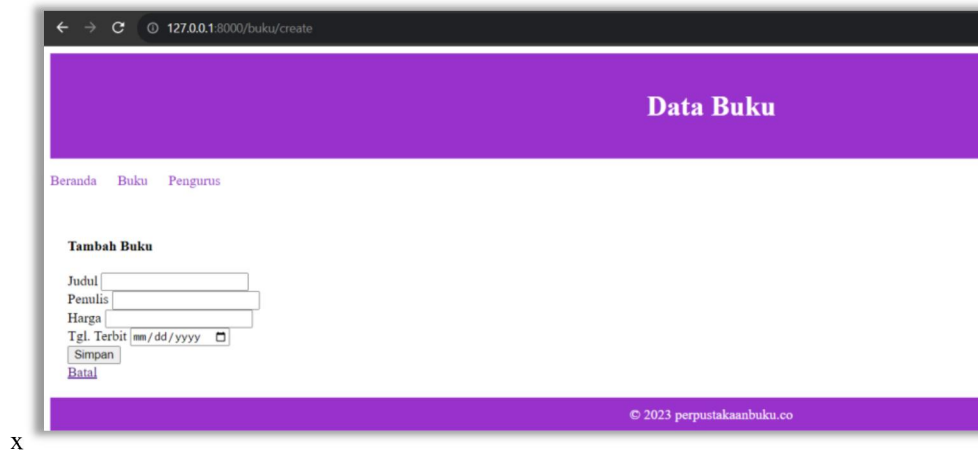
```

Fungsi `store` memiliki peran penting dalam proses pengelolaan data. Ketika pengguna mengisi formulir dengan informasi buku seperti judul, penulis, harga, dan tanggal terbit, fungsi ini berfungsi untuk mengambil data tersebut dari objek `Request` yang mewakili permintaan pengguna. Kemudian, data tersebut diassign ke properti-properti objek model "Buku" yang mewakili tabel database yang sesuai. Setelah data buku berhasil diisi, perintah `\$buku->save()` digunakan untuk menyimpannya ke dalam database. Akhirnya, pengguna diarahkan kembali ke halaman daftar buku menggunakan fungsi `redirect()` sehingga mereka dapat melihat buku yang baru saja ditambahkan. Berikut adalah tampilan Ketika kode-kode diatas telah berhasil untuk dijalankan:

3	Maiores voluptatem eos quae quod ut.	Lyla Quigley
4	Omnis repellendus sint qui dolor id praesentium.	Prof. Keely Terry
5	Adipisci eius est rerum perferendis.	Mohammad Bradtke
6	Mollitia qui qui suscipit sed.	Joyce Torp DDS
7	Quas dolorem dolorum necessitatibus at.	Garret Bruen Jr.
8	Deserunt beatae minima ut consequatur.	Mr. Garry Ernser
9	Suscipit vel sit aliquid excepturi tempore.	Lisette Beier
10	Aspernatur magnam ut cupiditate beatae qui repellat.	Nyasia Fisher

Jumlah data = 10  
Total harga = Rp 614429,00  
Tambah Buku





### 3.5. Membuat Route Hapus Buku

Buka pada file **web.php**, kemudian tambahkan route yang akan mengelola proses penghapusan buku “delete” ini, kita perlu menyertakan parameter id karena penghapusan data dilakukan berdasarkan identifikasi id. Sebagai contoh, jika kita ingin menghapus data buku dengan id 7, alamat URL yang digunakan akan menjadi dengan <http://localhost:8000/buku/delete/7>.

```
Route::post('/buku/delete/{id}', [BukuController::class, 'destroy'])->name('buku.destroy');
```

Route diatas digunakan untuk menangani permintaan URL `/buku/delete/{id}` adalah parameter dinamis yang akan menyertakan indentifikasi unik buku yang akan dihapus. Ketika route ini diakses, akan dipanggil method `destroy` yang terdapat dalam `BukuController`. Dengan menggunakan route ini, pengguna dapat menghapus buku dengan mengirimkan permintaan yang breisi indentifikasi id buku yang akan dihapus.

### 3.6. Membuat Tombol Hapus

Tombol hapus akan berbentuk sebuah tombol yang disisipkan ke dalam formular di halaman **buku.blade.php**, karena operasi delete data harus diimplementasikan melalui sebuah formular. Untuk menjalankan ini, kita perlu mengedit tampilan **buku.blade.php** dan menambahkan tombol hapus. Tambahkan kode seperti pada gambar di bawah:

```
<td>
    <form action="{{ route('buku.destroy', $buku->id) }}" method="post">
        @csrf
        <button onClick="return confirm('Yakin mau dihapus?')">Hapus</button>
    </form>
    <a href="{{ route('buku.edit', $buku->id) }}">Edit</a>
</td>
```

Kode diatas akan menampilkan tombol hapus, tombol ini memungkinkan pengguna untuk menghapus data buku yang sesuai dengan baris tertentu dalam tabel. Ketika tombol ini diklik, sebuah formular dengan metoe POST akan dikirimkan ke Route **buku.destroy** dengan menyertakan id buku yang akan dihapus. Tombol ini akan memunculkan konfirmasi peringatan kepada pengguna untuk memastikan mereka ingin melanjutkan dengan penghapusan data tersebut sebelum Tindakan penghapusan dilakukan. Dengan demikian, tombol ini memberikann cara yang aman untuk melakukan penghapusan data dalam aplikasi.

### 3.7. Menangani Proses Hapus Data Buku

Sama seperti dengan Create, Delete juga perlu adanya pengembalian data ke tampilan awal Ketika telah berhasil diubah dengan cara tambahkan function destroy() di bawah function store() pada file BukuController.

```
1 reference | 0 overrides
public function destroy($id){
    $buku = Buku::find($id);
    $buku->delete();
    return redirect('/buku');
}
```

Kode diatas merupakan bagian dari controller yang mengatur penghapusan data buku berdasarkan id yang diterima sebagai parameter. Ketika method destroy() diakses melalui route yang sesuai, system akan mencari buku dengan id yang sesuai dalam database menggunakan Buku::find(\$id). Setelah itu, buku tersebut akan dihapus dari database dengan menggunakan metode delete(). Kemudian, pengguna akan diarahkan Kembali ke halaman indeks daftar buku dengan menggunakan fungsi redirect(), sehingga data buku yang dihapus tidak lagi muncul dalam daftar buku yang ada. Berikut merupakan tampilan Ketika kode diatas dijalankan:

Id	Judul Buku	Penulis	Harga	Tgl. Terbit	Aksi
1	Reven consequatur sit ut ipa magna.	Dr. Kiana Gutman	Rp 37,801.00	17/09/2023	Hapus
2	Autem sit esqne ducimus perspiciatis eveniet exercitationem.	Megan Douglas	Rp 54,463.00	09/09/2023	Hapus
3	Mauris volupatem eoz que quod ut.	Lyla Quigley	Rp 39,722.00	11/08/2023	Hapus
4	Omnis repellendus sint qui dolor id praesentium.	Prof. Keely Terry	Rp 64,234.00	04/09/2023	Hapus
5	Adipisci etno est remus perferendis.	Mohammad Ibrahim	Rp 92,493.00	02/09/2023	Hapus
6	Mollitia qui qui nuncipit sed.	Joyce Torp DDS	Rp 72,752.00	09/09/2023	Hapus
7	Qua dolorem dolorem necessitatibus sit.	Garret Brown Jr.	Rp 70,714.00	31/08/2023	Hapus
8	Deserunt beatae minima ut consequatur.	Mr. Gary Erner	Rp 45,408.00	01/09/2023	Hapus
9	Suscipit vel sit aliquam exoptati tempore.	Luanne Borer	Rp 40,988.00	14/09/2023	Hapus
10	Aspernatur magnam ut cupiditate beatae qui repellat.	Nysia Fisher	Rp 75,854.00	15/09/2023	Hapus

Jumlah data = 10  
Total harga = Rp 614429.00  
Tambah Buku

id	Judul Buku	Harga	Tgl. Terbit	Aksi
1	Rerum consequatur sit in ipsa magni.	Rp 37.801,00	17/09/2023	Hapus
2	Autem sit eaque ducimus perspiciatis eveniet ex.	Rp 54.463,00	09/09/2023	Hapus
3	Maiores voluptatem eos quae quod ut.	Rp 39.722,00	31/08/2023	Hapus
4	Omnis repellendus sint qui dolor id praesentium.	Rp 64.234,00	04/09/2023	Hapus
5	Adipisci eius est rerum perferendis.	Rp 92.493,00	02/09/2023	Hapus
6	Mollitia qui qui suscipit sed.	Rp 72.752,00	09/09/2023	Hapus
7	Quos dolorem dolorem necessitatibus at.	Rp 70.714,00	31/08/2023	Hapus
8	Deserunt beatae minima ut consequatur.	Rp 45.408,00	01/09/2023	Hapus
9	Suscipit vel sit aliquid exceptum tempore.	Rp 69.988,00	16/09/2023	Hapus
10	Aspernatur magnam ut cupiditate beatae qui repellat.	Rp 75.854,00	15/09/2023	Hapus

Jumlah data = 10  
Total harga = Rp 614429,00  
Tambah Buku

#### 4.1. Tugas Praktikum

Silahkan buat sebuah fitur update pada project yang kita miliki. Ketentuan atau informasi yang dapat kalian gunakan adalah sebagai berikut:

- Tambah tombol untuk setiap data yang muncul di table (seperti tombol delete)
- Potongan kode form yang ditambahkan pada View adalah:

```
<form method="post" action="{{ route('buku.update', $buku->id) }}">
```

- Tampilkan data yang dipilih dengan menerapkan potongan kode berikut kedalam form edit buku

```
<div>Judul <input type="text" name="judul" value="{{ $buku->judul }}"></div>
```

LINK GITHUB : <https://github.com/wildandr/pertemuan5>

#### Web.php

```
Route::get('/buku/edit/{id}', [BukuController::class, 'edit'])->name('buku.edit');
Route::post('/buku/update/{id}', [BukuController::class, 'update'])->name('buku.update');
```

Route pertama **buku.edit** menggunakan metode GET untuk mengarahkan pengguna ke halaman yang memungkinkan pengeditan data buku tertentu berdasarkan id yang disertakan dalam URL. Route kedua yaitu **buku.update** menggunakan metode POST untuk mengelola pembaruan data buku setelah pengguna mengirimkan formulir yang telah diubah. Dengan route ini, aplikasi web dapat mengizinkan pengguna untuk mengedit dan menyimpan perubahan pada data buku yang ada.

## Buku.blade.php

```
<td>
    <form action="{{ route('buku.destroy', $buku->id) }}" method="post">
        @csrf
        <button onClick="return confirm('Yakin mau dihapus?')">Hapus</button>
    </form>
    <a href="{{ route('buku.edit', $buku->id) }}">Edit</a>
</td>
```

## Edit.blade.php

```
@extends('layouts.main')

@section('content')
    <div class="container">
        <h4>Edit Buku</h4>
        <form method="post" action="{{ route('buku.update', $buku->id) }}">
            @csrf
            <div>Judul <input type="text" name="judul" value="{{ $buku->judul }}"></div>
            <div>Penulis <input type="text" name="penulis" value="{{ $buku->penulis }}"></div>
            <div>Harga <input type="number" name="harga" value="{{ $buku->harga }}"></div>
            <div>Tgl. Terbit <input type="date" name="tgl_terbit" value="{{ $buku->tgl_terbit }}"></div>
            {{-- <div>Judul <input type="text" name="judul"></div>
            <div>Penulis <input type="text" name="penulis"></div>
            <div>Harga <input type="number" name="harga"></div>
            <div>Tgl. Terbit <input type="date" name="tgl_terbit"></div> --}}
            <div><button type="submit">Simpan</button></div>
            <a href="/buku">Batal</a>
        </form>
    </div>
@endsection
```

## BukuController.php

```
1 reference | 0 overrides
public function edit($id){
    $buku = Buku::find($id);
    return view('edit', compact('buku'));
}

1 reference | 0 overrides
public function update(Request $request, $id){
    $buku = Buku::find($id);
    $buku->judul = $request->judul;
    $buku->penulis = $request->penulis;
    $buku->harga = $request->harga;
    $buku->tgl_terbit = $request->tgl_terbit;
    $buku->save();
    return redirect("/buku");
}
```

Kode diatas terdapat dua method yang terdapat pada controller. Method pertama yaitu **edit(\$id)** dimana method tersebut mengambil buku berdasarkan id yang diberikan kemudian mengarahkan pengguna ke tampilan edit dengan mengirimkan data buku tersebut. Method kedua yaitu **update(Request \$request, \$id)**, mengelola proses pembaruan data buku berdasarkan id yang diberikan dalam permintaan. Data yang diubah

oleh pengguna dalam formulir diambil melalui objek “Request” kemudian digunakan untuk memperbarui atribut-atribut buku yang sesuai. Setelah perubahan dilakukan, data buku disimpan ke dalam database dan pengguna diarahkan Kembali ke halaman indeks daftar buku. Dengan demikian, pengguna dapat mengedit dan menyimpan perubahan pada data buku dengan menggunakan formulir yang disediakan dalam aplikasi web.

## Tampilan website

Id	Judul Buku	Penulis	Harga	Tgl. Terbit	Aksi
1	Enims consequatur ut in qua magna	Dr. Kianan Gutman	Rp 87.865,00	11/09/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
2	Amet ut magna ducimus peripetatis et nislit exercitabitur	Morgan Douglas	Rp 54.463,00	09/09/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
3	Mauris volutpatibus ari qua quat ut	Lila Quaglin	Rp 88.722,00	31/08/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
4	Quam repellenfus iust qui dolor ut praesentiam	Prof. Emily Terry	Rp 64.234,00	04/09/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
5	Adipisci enim est verum perferendis	Mohammad Ibrahim	Rp 90.499,00	02/09/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
6	Modula qui qua suscipit sed	Joyce Tang DDI	Rp 72.752,00	09/09/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
7	Quam dolorem dolorem suscipiendibus et	Garet Brown Jr.	Rp 70.734,00	31/08/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
8	Deserunt beatae minima ut consequatur	Mr. Garry Knauer	Rp 45.408,00	01/09/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
9	Tenueque vel ut aliquam suscipere tempore	Liamie Breen	Rp 80.988,00	18/08/2023	<a href="#">Hapus</a> <a href="#">Edit</a>
10	Aperiamque magnam ut cupiditate beatae qui repellat	Plyana Fisher	Rp 73.874,00	15/09/2023	<a href="#">Hapus</a> <a href="#">Edit</a>

Jumlah data = 10  
Total harga = Rp 614429,00  
[Tambah Buku](#)

### Data Buku

[Beranda](#) [Buku](#) [Pengurus](#)

#### Edit Buku

Judul   
 Penulis   
 Harga   
 Tgl. Terbit

© 2023 perpustakaanbuku.co

### Data Buku

[Beranda](#) [Buku](#) [Pengurus](#)

#### Tambah Buku

Judul   
 Penulis   
 Harga   
 Tgl. Terbit

© 2023 perpustakaanbuku.co

## **BAB IV**

### **KESIMPULAN**

#### **4.1. Kesimpulan**

Laporan praktikum ini menjelaskan penerapan operasi CRUD (Create, Read, Update, Delete) dalam kerangka kerja Laravel, dengan fokus pada kemampuan kerangka kerja ini dalam menyederhanakan pengembangan aplikasi web. Kami memahami prinsip dasar Model-View-Controller (MVC), pembuatan model, pengaturan rute, dan bagaimana menghubungkannya dengan Controller. Selain itu, kami juga memahami cara membuat formulir untuk menambahkan, mengambil, mengedit, dan menghapus data dalam basis data. Laporan ini juga menekankan pentingnya penggunaan token CSRF untuk menjaga keamanan formulir. Dengan memanfaatkan Laravel, pengembangan aplikasi web dapat dilakukan dengan efisien, sehingga memungkinkan pengelolaan data yang aman dan user-friendly.

## DAFTAR PUSTAKA

<https://appmaster.io/blog/crud-operations-what-is-crud>

<https://github.com/wildandr/pertemuan5>