

# **TUGAS**

## **CRUD**

*Makalah Ini Dibuat Untuk Memenuhi Tugas Kuliah*

*Dosen : Dicky Yudha Pratama, M.Kom*



Di susun Oleh ;

Nama : Dicky Dirgantara

Nim : G.231.23.0165

Mata Kuliah : Permmrograman Berorientasi Objeck

Praktikum : III - IV

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI**

**UNIVERSITAS SEMARANG**

**2024/2025**

# Membuat CRUD pada Laravel

## Pendahuluan

**CRUD** adalah akronim yang digunakan untuk menggambarkan empat operasi dasar yang dapat dilakukan pada data dalam sistem basis data atau aplikasi. Keempat operasi ini sangat penting karena merupakan tindakan utama dalam interaksi antara pengguna dan aplikasi, terutama yang berkaitan dengan pengelolaan data. Berikut penjelasan masing-masing operasi:

1. **Create** (Membuat).

Operasi ini digunakan untuk menambahkan data baru ke dalam sistem atau basis data. Biasanya, ini dilakukan melalui form input, API, atau alat lain yang memungkinkan pengguna memasukkan data baru.

2. **Read** (Membaca).

Operasi ini digunakan untuk mengambil atau membaca data yang sudah ada dalam sistem atau basis data. Ini dapat dilakukan untuk menampilkan data kepada pengguna atau untuk tujuan lainnya, seperti analisis atau laporan.

3. **Update** (Memperbarui).

Operasi ini digunakan untuk mengubah data yang sudah ada dalam basis data. Ini bisa meliputi pengeditan data seperti nama, alamat, atau status lainnya yang sudah tersimpan.

4. **Delete** (Menghapus).

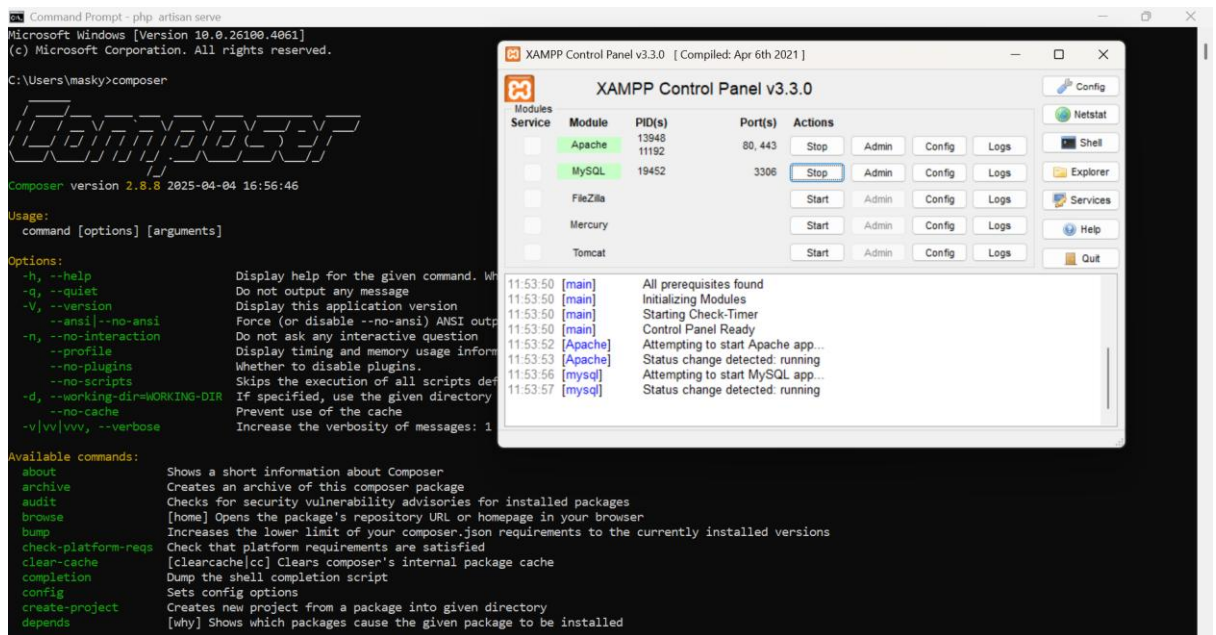
Operasi ini digunakan untuk menghapus data yang sudah ada dalam sistem atau basis data. Proses ini akan menghilangkan data secara permanen dari sistem.

## Langkah – Langkah Membuat Project CRUD

1. **Mengaktifkan XAMPP**

Jalankan Xampp :

- Buka Xampp Control Panel.
- Klik Start pada bagian Apache dan MySQL untuk mengaktifkan server local dan database



## 2. Membuat Database di php MyAdmin

- Akses phpMyAdmin.
  - Buka browser dan Ketikan <http://localhost/phpmyadmin/> untuk mengakses phpMyAdmin.
  - Login username root dan kosongkan password.
- Buat Database :
  - Klik tab Databases.
  - Buat database baru, misalnya data\_pegawai.
- Buat Tabel Pegawai :
  - Pilih database yang baru dibuat, lalu klik Create table.
  - Nama table :pegawai
  - Buat kolom berikut :
    - pegawai\_id Int Primary key.
    - pegawai\_nama Varchar,
    - pegawai\_jabatan Varchar,
    - pegawai\_umur INT,
    - pegawai\_alamat Varchar.

Table structure

Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	pegawai_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	pegawai_nama	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	pegawai_jabatan	varchar(20)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	pegawai_umur	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	5	pegawai_alamat	text	utf8mb4_general_ci		No	None			Change  Drop  More

Document was last saved: Just now

### 3. Membuat Proyek Laravel

- Install Laravel :

Pastikan Anda sudah menginstal **Composer**. Buka terminal/command prompt, lalu jalankan perintah:

```
composer create-project --prefer-dist laravel/laravel data-pegawai
```

- Mengatur .env File :

Buka file .env di direktori utama proyek dan ubah pengaturan database agar sesuai dengan konfigurasi XAMPP:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=data-pegawai
DB_USERNAME=root
DB_PASSWORD=
```

### 4. Membuat Model dan Migration

- Buat Model dan Migration

Jalankan perintah artisan untuk membuat model Pegawai dan migration untuk tabel pegawai:

```
PS C:\Users\masky> php artisan make:model Pegawai -m
```

- Edit Migration :

Buka file migration yang baru saja dibuat di database/migrations/xxxx\_xx\_xx\_create\_pegawai\_table.php dan sesuaikan dengan struktur tabel yang ada di phpMyAdmin:

```

public function up(): void
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}

```

- Migrasi Database :

Jalankan perintah untuk melakukan migrasi:

```
PS C:\Users\masky> php artisan migrate
```

## 5. Membuat Controller dan Route

- Buat Controller :

Jalankan perintah untuk membuat controller PegawaiController:

```
PS C:\Users\masky> php artisan make:controller PegawaiController
```

- Mendefinisikan Method di Controller :

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class PegawaiController extends Controller
{
    public function index()
    {
        //
        $pegawai = DB::table('pegawai')->get();

        //
        return view('index',['pegawai' => $pegawai]);
    }

    //
    public function tambah()
    {
        // memanggil view tambah
        return view('tambah');
    }
}

```

```

// method untuk insert data ke table pegawai
public function store(Request $request)
{
    // insert data ke table pegawai
    DB::table('pegawai')->insert([
        'pegawai_nama' => $request->nama,
        'pegawai_jabatan'=> $request->jabatan,
        'pegawai_umur' => $request->umur,
        'pegawai_alamat' => $request->alamat
    ]);

    // alihkan halaman ke halaman pegawai
    return redirect('/pegawai');
}

// method untuk edit data pegawai
public function edit($id)
{
    // mengambil data pegawai berdasarkan id yang dipilih
    $pegawai = DB::table('pegawai')->where('pegawai_id', $id)->get();

    // passing data pegawai yang didapat ke view edit.blade.php
    return view('edit', ['pegawai' => $pegawai]);
}

public function update(Request $request)
{
    // update data pegawai
    DB::table('pegawai')->where('pegawai_id', $request->id)->update([
        'pegawai_nama' => $request->nama,
        'pegawai_jabatan'=> $request->jabatan,
        'pegawai_umur' => $request->umur,
        'pegawai_alamat' => $request->alamat
    ]);

    // alihkan halaman ke halaman pegawai
    return redirect('/pegawai');
}

// In app/Http/Controllers/PegawaiController.php
public function hapus($id)
{
    // Menghapus data pegawai berdasarkan id
    DB::table('pegawai')->where('pegawai_id', $id)->delete();

    // Redirect kembali ke halaman pegawai
    return redirect('/pegawai');
}

```

- Medefinisikan Route :

Tambahkan route untuk setiap aksi (tambah, edit, hapus, dll) di file routes/web.php:

```
Route::get('/', function () {
    return view('welcome');
});

//
Route::get('/pegawai', [PegawaiController::class, 'index']);

//route CRUD Tambah Pegawai
Route::get('/pegawai/tambah', [PegawaiController::class, 'tambah']);

//route CRUD pegawai --> store
Route::post('/pegawai/store', [PegawaiController::class, 'store']);

// route CRUD Edit Pegawai
Route::get('/pegawai/edit/{id}', [PegawaiController::class, 'edit']);

// route CRUD Update Pegawai
Route::post('/pegawai/update', [PegawaiController::class, 'update']);

// In routes/web.php
Route::get('/pegawai/hapus/{id}', [PegawaiController::class, 'hapus']);
```

## 6. Membuat Blade Views untuk Menampilkan Tabel.

- Buat Folder dan File Views :

Buat folder resources/views/pegawai, kemudian buat file index.blade.php untuk menampilkan daftar pegawai dan form create.blade.php serta edit.blade.php untuk menambah dan mengedit data pegawai.

```
<h3>Data Pegawai</h3>

<a href="/pegawai/tambah"> + Tambah Pegawai Baru</a>

<br/>
<br/>

<table border="1">
    <tr>
        <th>Nama</th>
        <th>Jabatan</th>
        <th>Umur</th>
        <th>Alamat</th>
        <th>Opsional</th>
    </tr>

    @foreach($pegawai as $p)
    <tr>
        <td>{{ $p->pegawai_nama }}</td>
        <td>{{ $p->pegawai_jabatan }}</td>
        <td>{{ $p->pegawai_umur }}</td>
        <td>{{ $p->pegawai_alamat }}</td>
        <td>
            <a href="/pegawai/edit/{{ $p->pegawai_id }}">Edit</a>
            |
            <a href="/pegawai/hapus/{{ $p->pegawai_id }}">Hapus</a>
        </td>
    </tr>
</table>
```

- Form Input untuk Menambah dan Mengedit Data:
  - Form tambah pegawai (tambah.blade.php):

```
<a href="/pegawai">Kembali</a>

<br/><br/>

<form action="/pegawai/store" method="post">
    {{ csrf_field() }}
    Nama <input type="text" name="nama" required="required"> <br/>
    Jabatan <input type="text" name="jabatan" required="required"> <br/>
    Umur <input type="number" name="umur" required="required"> <br/>
    Alamat <textarea name="alamat" required="required"></textarea> <br/>
    <input type="submit" value="Simpan Data">
</form>
```



- Form edit pegawai (edit.blade.php):

```
<h2><a href="https://www.usm.ac.id">www.usm.ac.id</a></h2>
<h3>Edit Pegawai</h3>

<a href="/pegawai">Kembali</a>
<br/><br/>

@foreach($pegawai as $p)
<form action="/pegawai/update" method="post">
    {{ csrf_field() }}
    <input type="hidden" name="id" value="{{ $p->pegawai_id }}">

    Nama: <input type="text" required="required" name="nama" value="{{ $p->pegawai_nama }}"> <br/>
    Jabatan: <input type="text" required="required" name="jabatan" value="{{ $p->pegawai_jabatan }}"> <br/>
    Umur: <input type="number" required="required" name="umur" value="{{ $p->pegawai_umur }}"> <br/>
    Alamat: <textarea required="required" name="alamat">{{ $p->pegawai_alamat }}</textarea> <br/>

    <input type="submit" value="Simpan Data">
</form>
@endforeach
```

## 7. Menjalankan Aplikasi yang sudah dibuat

Jalankan aplikasi menggunakan php artisan serve:

```
C:\xampp\htdocs\data-pegawai>php artisan serve

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

Press Ctrl+C to stop the server

2025-06-03 11:55:49 / ..... ~ 5s
2025-06-03 11:55:49 /favicon.ico ..... ~ 5s
2025-06-03 11:56:13 /pegawai ..... ~ 0.77ms
2025-06-03 11:56:13 /css/style.css ..... ~ 0.15ms
2025-06-03 11:56:17 /favicon.ico ..... ~ 0.73ms
2025-06-03 11:56:17 ..... ~ 1m 24s
```

Akses aplikasi melalui <http://localhost:8000/pegawai> untuk melihat daftar pegawai dan fitur tambah, edit, serta hapus.

8. Menginput Hasil  
Tambah

[www.usm.ac.id](http://www.usm.ac.id)

## Data Pegawai

[Kembali](#)

Nama	<input type="text"/>
Jabatan	<input type="text"/>
Umur	<input type="text"/>
Alamat	<input type="text"/>
<input type="button" value="Simpan Data"/>	

Edit

[www.usm.ac.id](http://www.usm.ac.id)

## Edit Pegawai

[Kembali](#)

Nama:	<input type="text" value="Dicky Dirgantara"/>
Jabatan:	<input type="text" value="professor"/>
Umur:	<input type="text" value="12"/>
Alamat:	<input type="text" value="jdhxgs"/>
<input type="button" value="Simpan Data"/>	

9. Output/hasil dari pemrograman tersebut:

Hasil Awal

Data Pegawai

+ [Tambah Pegawai Baru](#)

Nama	Jabatan	Umur	Alamat	Opsi	
Dicky Dirgantara	professor	12	jdhxgs	<a href="#">Edit</a>	<a href="#">Hapus</a>
Akmal Fuadi	TNI	67	Semarang	<a href="#">Edit</a>	<a href="#">Hapus</a>
Putri Ayu Verawati	Dokter	27	Jakarta	<a href="#">Edit</a>	<a href="#">Hapus</a>
Nela Muna Oktovia	SPG	22	Genuk	<a href="#">Edit</a>	<a href="#">Hapus</a>
Rudi Tabuti	Dosen	34	Bandungan	<a href="#">Edit</a>	<a href="#">Hapus</a>
Aulia Rahmawati	Wiraswasta	24	Ungaran	<a href="#">Edit</a>	<a href="#">Hapus</a>

Tambah Peagawai Baru

(Fikky Agih Imawan/Mentri/45/Sampangan)

Data Pegawai

+ [Tambah Pegawai Baru](#)

Nama	Jabatan	Umur	Alamat	Opsi	
Dicky Dirgantara	professor	12	jdhxgs	<a href="#">Edit</a>	<a href="#">Hapus</a>
Akmal Fuadi	TNI	67	Semarang	<a href="#">Edit</a>	<a href="#">Hapus</a>
Putri Ayu Verawati	Dokter	27	Jakarta	<a href="#">Edit</a>	<a href="#">Hapus</a>
Nela Muna Oktovia	SPG	22	Genuk	<a href="#">Edit</a>	<a href="#">Hapus</a>
Rudi Tabuti	Dosen	34	Bandungan	<a href="#">Edit</a>	<a href="#">Hapus</a>
Aulia Rahmawati	Wiraswasta	24	Ungaran	<a href="#">Edit</a>	<a href="#">Hapus</a>
Fikky Agih Imawan	Mentri	45	Sampangan	<a href="#">Edit</a>	<a href="#">Hapus</a>

Edit Data Pegawai

(Akmal Fuadi/Guru/54/Pedurungan)

Data Pegawai

+ [Tambah Pegawai Baru](#)

Nama	Jabatan	Umur	Alamat	Opsi	
Dicky Dirgantara	professor	12	jdhxgs	<a href="#">Edit</a>	<a href="#">Hapus</a>
Akmal Fuadi	Guru	54	Pedurungan	<a href="#">Edit</a>	<a href="#">Hapus</a>
Putri Ayu Verawati	Dokter	27	Jakarta	<a href="#">Edit</a>	<a href="#">Hapus</a>
Nela Muna Oktovia	SPG	22	Genuk	<a href="#">Edit</a>	<a href="#">Hapus</a>
Rudi Tabuti	Dosen	34	Bandungan	<a href="#">Edit</a>	<a href="#">Hapus</a>
Aulia Rahmawati	Wiraswasta	24	Ungaran	<a href="#">Edit</a>	<a href="#">Hapus</a>
Fikky Agih Imawan	Mentri	45	Sampangan	<a href="#">Edit</a>	<a href="#">Hapus</a>

# Hapus Data Pegawai

(Rudi Tabuti/Dosen/34/Bandungan)

Data Pegawai

+ Tambah Pegawai Baru

Nama	Jabatan	Umur	Alamat	Opsi	
Dicky Dirgantara	professor	12	jdhxgs	Edit	Hapus
Akmal Fuadi	Guru	54	Pedurungan	Edit	Hapus
Putri Ayu Verawati	Dokter	27	Jakarta	Edit	Hapus
Nela Muna Oktovia	SPG	22	Genuk	Edit	Hapus
Aulia Rahmawati	Wiraswasta	24	Ungaran	Edit	Hapus
Fikky Agih Imawan	Mentri	45	Sampangan	Edit	Hapus