

File Upload

Tujuan

1. Mahasiswa memahami konsep **CRUD (Create, Read, Update, Delete)** di Laravel.
2. Mahasiswa bisa membuat tabel **jobs** di database.
3. Mahasiswa bisa menampilkan daftar lowongan.
4. Mahasiswa bisa menambahkan lowongan dengan **upload logo perusahaan**.
5. Mahasiswa bisa mengedit dan menghapus lowongan.

Dasar Teori

Setiap aplikasi web pada dasarnya dibangun dari sebuah siklus yang disebut **CRUD**, singkatan dari **Create, Read, Update, Delete**. Empat operasi ini adalah pondasi dasar dari hampir semua sistem:

- **Create (Tambah data)** → menambahkan data baru, misalnya Admin menambahkan lowongan kerja baru.
- **Read (Lihat data)** → menampilkan data yang sudah ada, misalnya user melihat daftar lowongan yang tersedia.
- **Update (Edit data)** → memperbarui data lama, misalnya Admin mengubah gaji atau deskripsi pekerjaan.
- **Delete (Hapus data)** → menghapus data yang tidak diperlukan lagi, misalnya Admin menghapus lowongan yang sudah ditutup.

Dalam Laravel, proses CRUD biasanya diatur menggunakan beberapa komponen utama:

- **Migration** → untuk membuat tabel di database.
- **Model** → sebagai penghubung langsung antara tabel database dan kode Laravel.

- **Controller** → untuk mengatur logika CRUD, misalnya apa yang terjadi ketika tombol “Tambah” ditekan.
- **Blade View** → untuk menampilkan data atau membuat form di halaman web.

Selain CRUD teks biasa (seperti judul, deskripsi, atau angka), sering kali aplikasi juga membutuhkan **form untuk upload file**. Fitur ini sangat penting dalam banyak kasus nyata:

- Pada aplikasi Portal Lowongan Kerja, Admin mungkin ingin mengunggah **logo perusahaan** saat menambahkan lowongan.
- Sementara Job Seeker perlu mengunggah **CV dalam bentuk PDF** saat melamar pekerjaan.

Laravel sudah menyediakan cara mudah untuk membuat form upload file melalui Blade View, dan file yang diunggah akan disimpan di folder khusus (`storage/public`). Dengan fitur ini, aplikasi terasa lebih **hidup dan realistik**, karena tidak hanya menyimpan data teks, tetapi juga dokumen atau gambar yang mendukung.

Perintah `php artisan storage:link` dapat digunakan untuk membuat symbolic link (tautan) antara folder storage Laravel dengan folder public, agar file yang disimpan lewat Laravel bisa diakses lewat URL di browser.

Maka dari itu, di pertemuan ini kita akan belajar bagaimana cara:

1. Membuat **CRUD sederhana** dengan migration, model, controller, dan view.
2. Menampilkan data lowongan kerja dalam bentuk tabel.
3. Menambahkan fitur **form upload file** untuk melengkapi data lowongan.

Dengan memahami dan menguasai materi ini, kita akan memiliki keterampilan dasar yang bisa diterapkan di hampir semua aplikasi web, dari sistem akademik, e-commerce, hingga portal lowongan kerja seperti yang kita bangun bersama.

Langkah Praktikum

1. Membuat Migration & Model Job

1. Jalankan perintah berikut:

```
php artisan make:model JobVacancy -m
```

2. Buka file migration `database/migrations/xxxx_create_jobs_table.php` dan isi:

```
public function up()
{
    Schema::create('job_vacancies', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->text('description');
        $table->string('location');
        $table->string('company');
        $table->string('logo')->nullable();
        $table->integer('salary')->nullable();
        $table->timestamps();
    });
}
```

3. Jalankan migrasi:

```
php artisan migrate
```

4. Di model `app/Models/JobVacancy.php`, tambahkan property tabel agar eksplisit

```
use HasFactory;

protected $table = 'job_vacancies';
protected $fillable = [
    'title',
    'description',
    'location',
    'company',
    'logo',
    'salary',
];
```

2. Membuat Controller

1. Buat controller:

```
php artisan make:controller JobController --resource
```

2. Laravel otomatis membuat file `app/Http/Controllers/JobController.php` dengan fungsi CRUD (index, create, store, edit, update, destroy).

3. Menambahkan Route CRUD

Buka `routes/web.php` dan tambahkan:

```
use App\Http\Controllers\JobController;

...

Route::resource('jobs',
JobController::class)->middleware(['auth', 'isAdmin']);
```

Artinya: hanya **Admin** yang bisa mengelola lowongan kerja.

4. Membuat Form Tambah Lowongan

Buka file resources/views/jobs/create.blade.php:

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

@section('content')


<h2>Tambah Lowongan</h2>
    <form action="{{ route('jobs.store') }}" method="POST"
enctype="multipart/form-data">
        @csrf
        <input type="text" name="title" placeholder="Judul
Lowongan" class="form-control mb-2">
        <textarea name="description"
placeholder="Deskripsi" class="form-control
mb-2"></textarea>
        <input type="text" name="location"
placeholder="Lokasi" class="form-control mb-2">
        <input type="text" name="company"
placeholder="Nama Perusahaan" class="form-control mb-2">
        <input type="number" name="salary"
placeholder="Gaji" class="form-control mb-2">
        <input type="file" name="logo" class="form-control
mb-2">
        <button type="submit" class="btn
btn-primary">Simpan</button>
    </form>


```

```
</div>  
@endsection
```

5. Menyimpan Data & Upload File

Edit `JobController.php` → method `store`:

```
use App\Models\JobVacancy as Job;  
  
...  
  
public function store(Request $request)  
{  
    $request->validate([  
        'title' => 'required',  
        'description' => 'required',  
        'location' => 'required',  
        'company' => 'required',  
        'logo' => 'image|mimes:jpg,png,jpeg|max:2048'  
    ]);  
  
    $logoPath = null;  
    if ($request->hasFile('logo')) {  
        $logoPath = $request->file('logo')->store('logos',  
        'public');  
    }  
  
    Job::create([
```

```
        'title' => $request->title,
        'description' => $request->description,
        'location' => $request->location,
        'company' => $request->company,
        'salary' => $request->salary,
        'logo' => $logoPath
    ]);

    return
redirect()->route('jobs.index')->with('success', 'Lowongan
berhasil ditambahkan');
}
```

6. Menampilkan Daftar Lowongan

Buat file resources/views/jobs/index.blade.php:

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

@section('content')


## Daftar Lowongan

Tambah Lowongan

| Judul | Perusahaan | Lokasi |
|-------|------------|--------|
|-------|------------|--------|


```

```

<th>Gaji</th>
<th>Logo</th>
<th>Aksi</th>
</tr>
@foreach($jobs as $job)
<tr>
<td>{{ $job->title }}</td>
<td>{{ $job->company }}</td>
<td>{{ $job->location }}</td>
<td>{{ $job->salary }}</td>
<td>
@if($job->logo)

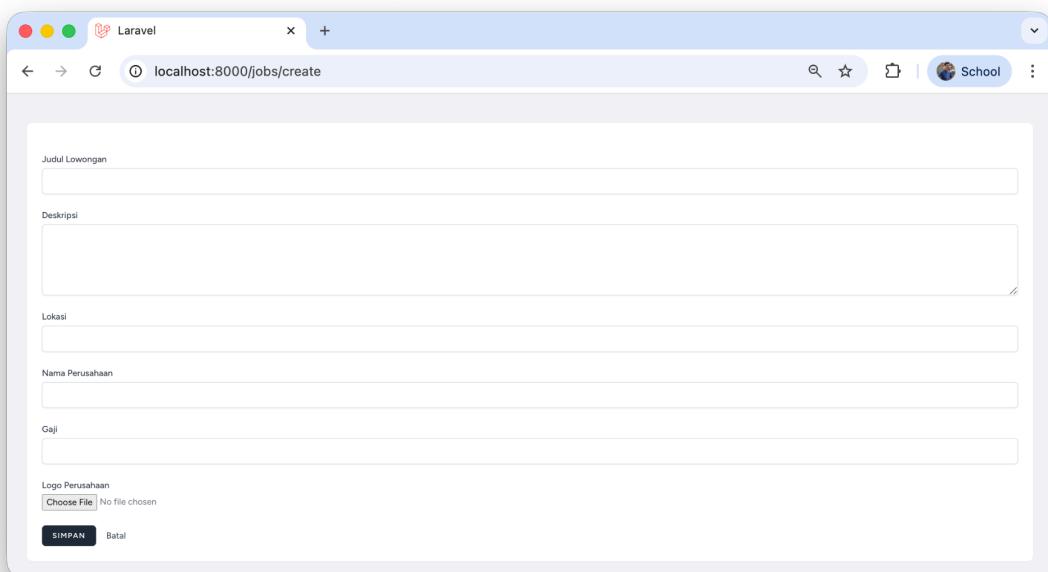
@endif
</td>
<td>
<a href="{{ route('jobs.edit', $job->id) }}"
class="btn btn-warning btn-sm">Edit</a>
<form action="{{ route('jobs.destroy',
$job->id) }}" method="POST" style="display:inline;">
@csrf @method('DELETE')
<button class="btn btn-danger btn-sm"
onclick="return confirm('Hapus data?')>Hapus</button>
</form>
</td>
</tr>
@endforeach
</table>

```

```
</div>  
@endsection
```

Edit `JobController.php` → method `index` dan `store`:

```
public function index()  
{  
    $jobs = Job::all();  
    return view('jobs.index', compact('jobs'));  
}  
  
public function create()  
{  
    return view('jobs.create');  
}
```



7. Edit & Hapus Lowongan

Laravel sudah menyiapkan method `edit`, `update`, dan `destroy`.

- `edit` → menampilkan form edit data.
- `update` → menyimpan perubahan.
- `destroy` → menghapus data.

Latihan

1. Tambahkan kolom **Jenis Pekerjaan (Full-time/Part-time)** di tabel `jobs`.
2. Ubah tampilan daftar lowongan agar lebih menarik (gunakan Bootstrap card).
3. Tambahkan fitur `edit`, `update`, dan `destroy`