📚 DOKUMENTASI LENGKAP - OWN YOUR CAREER

Sistem Informasi Lowongan Magang/Kerja

📖 DAFTAR ISI

---

1. OVERVIEW PROJECT

🎯 Tujuan

Sistem untuk menghubungkan **Perusahaan** yang menyediakan lowongan dengan **Mahasiswa** yang mencari magang/kerja, dengan **Admin** sebagai verifikator.

👥 Pengguna Sistem

- **Admin**: Verifikasi perusahaan & lowongan, lihat laporan
- **Perusahaan**: Post lowongan, lihat pelamar, download CV
- **Mahasiswa**: Cari lowongan, upload CV, kirim lamaran

🛠️ Tech Stack

- **Framework**: Laravel 11/12 (PHP 8.4)
- **Database**: SQLite (gratis, tidak perlu XAMPP MySQL)
- **Frontend**: Blade Template + CSS inline
- **Storage**: Local storage untuk file CV

---

2. REQUIREMENTS & SETUP

A. Software yang Dibutuhkan

- PHP 8.4+
- Composer
- Visual Studio Code
- SQLite (sudah built-in di Laravel)

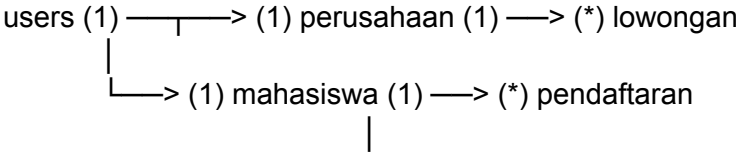
B. Langkah Setup Awal

```
# 1. Install Laravel
cd C:\xampp\htdocs  # atau folder lain
composer create-project laravel/laravel own-your-career
cd own-your-career

# 2. Setup Database
# Buat file database.sqlite
New-Item database/database.sqlite -ItemType File

# 3. Edit .env
# Ubah DB_CONNECTION menjadi:
DB_CONNECTION=sqlite
# Comment/hapus DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, DB_PASSWORD

# 4. Generate Key
php artisan key:generate

# 5. Test Server
php artisan serve
# Buka: http://localhost:8000
```

---

3. STRUKTUR DATABASE

ERD Tabel



lowongan (*) <——————————┘

A. Tabel: `users`

Kolom	Tipe	Keterangan
id	PK	Auto increment
name	VARCHAR	Nama user
email	VARCHAR	Email (unique)
password	VARCHAR	Hashed password
peran	ENUM	'admin', 'perusahaan', 'mahasiswa'
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

B. Tabel: `perusahaan`

Kolom	Tipe	Keterangan
id	PK	
user_id	FK	→ users.id
nama_perusahaan	VARCHAR	
alamat	TEXT	
no_telp	VARCHAR	
deskripsi	TEXT	Nullable
status_verifikasi	ENUM	'menunggu', 'disetujui', 'ditolak'

C. Tabel: `mahasiswa`

Kolom	Tipe	Keterangan
id	PK	

user_id	FK	→ users.id
nim	VARCHAR	Unique
nama_lengkap	VARCHAR	
jurusan	VARCHAR	
no_telp	VARCHAR	
alamat	TEXT	Nullable

D. Tabel: `lowongan`

Kolom	Tipe	Keterangan
id	PK	
perusahaan_id	FK	→ perusahaan.id
posisi	VARCHAR	
deskripsi	TEXT	
persyaratan	TEXT	
lokasi	VARCHAR	
tipe	ENUM	'magang', 'kerja'
batas_akhir	DATE	
status	ENUM	'menunggu', 'aktif', 'nonaktif'

E. Tabel: `pendaftaran`

Kolom	Tipe	Keterangan
id	PK	
lowongan_id	FK	→ lowongan.id
mahasiswa_id	FK	→ mahasiswa.id
jalur_cv	VARCHAR	Path file CV

status	ENUM	'menunggu', 'diterima', 'ditolak'
tanggal_daftar	TIMESTAMP	
UNIQUE	(lowongan_id, mahasiswa_id)	Cegah double apply

---

4. STRUKTUR FOLDER PROJECT

```
own-your-career/
├── app/
│   ├── Http/
│   │   ├── Controllers/
│   │   │   ├── Auth/
│   │   │   │   ├── DaftarController.php
│   │   │   │   ├── LoginAdminController.php
│   │   │   │   ├── LoginPerusahaanController.php
│   │   │   │   └── LoginMahasiswaController.php
│   │   │   ├── Admin/
│   │   │   │   ├── DashboardAdminController.php
│   │   │   │   ├── VerifikasiPerusahaanController.php
│   │   │   │   ├── VerifikasiLowonganController.php
│   │   │   │   └── LaporanController.php
│   │   │   ├── Perusahaan/
│   │   │   │   ├── DashboardPerusahaanController.php
│   │   │   │   ├── LowonganController.php
│   │   │   │   └── PelamarController.php
│   │   │   └── Mahasiswa/
│   │   │       ├── DashboardMahasiswaController.php
│   │   │       ├── CariLowonganController.php
│   │   │       ├── PendaftaranController.php
│   │   │       └── LamaranController.php
│   │   └── Middleware/
│   │       ├── AdminMiddleware.php
│   │       ├── PerusahaanMiddleware.php
│   │       └── MahasiswaMiddleware.php
│   └── Models/
│       ├── User.php
│       ├── Perusahaan.php
│       ├── Mahasiswa.php
│       ├── Lowongan.php
│       └── Pendaftaran.php
├── database/
```

```
│   ├── migrations/
│   │   ├── xxxx_tambah_peran_ke_tabel_users.php
│   │   ├── xxxx_buat_tabel_perusahaan.php
│   │   ├── xxxx_buat_tabel_mahasiswa.php
│   │   ├── xxxx_buat_tabel_lowongan.php
│   │   └── xxxx_buat_tabel_pendaftaran.php
│   └── seeders/
│       └── AdminSeeder.php
├── resources/
│   └── views/
│       ├── beranda.blade.php
│       ├── auth/
│       │   ├── pilih-peran.blade.php
│       │   ├── masuk/ (admin, perusahaan, mahasiswa)
│       │   └── daftar/ (perusahaan, mahasiswa)
│       ├── admin/
│       │   ├── dashboard.blade.php
│       │   ├── perusahaan/
│       │   ├── lowongan/
│       │   └── laporan/
│       ├── perusahaan/
│       │   ├── dashboard.blade.php
│       │   ├── lowongan/
│       │   └── pelamar/
│       └── mahasiswa/
│           ├── dashboard.blade.php
│           ├── lowongan/
│           └── lamaran/
└── routes/
    └── web.php
```

---

5. LANGKAH IMPLEMENTASI

FASE 1: Setup Database & Models

Step 1: Buat Migration

```
php artisan make:migration tambah_peran_ke_tabel_users
php artisan make:migration buat_tabel_perusahaan
php artisan make:migration buat_tabel_mahasiswa
php artisan make:migration buat_tabel_lowongan
php artisan make:migration buat_tabel_pendaftaran
```

Contoh Migration: `tambah_peran_ke_tabel_users`

```
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->enum('peran', ['admin', 'perusahaan', 'mahasiswa'])->after('password');
    });
}
```

Jalankan Migration:

```
php artisan migrate
```

Step 2: Buat Models

```
php artisan make:model Perusahaan
php artisan make:model Mahasiswa
php artisan make:model Lowongan
php artisan make:model Pendaftaran
```

Contoh Model: `User.php`

```
class User extends Authenticatable
{
    protected $fillable = ['name', 'email', 'password', 'peran'];

    public function perusahaan()
    {
        return $this->hasOne(Perusahaan::class);
    }

    public function mahasiswa()
    {
        return $this->hasOne(Mahasiswa::class);
    }

    public function adalahAdmin()
    {
        return $this->peran === 'admin';
    }
}
```

Step 3: Buat Admin Default (Seeder)

php artisan make:seeder AdminSeeder

```
use Illuminate\Support\Facades\Hash;

User::create([
    'name' => 'Administrator',
    'email' => 'admin@ownyourcareer.com',
    'password' => Hash::make('admin123'),
    'peran' => 'admin',
]);
```

php artisan db:seed --class=AdminSeeder

FASE 2: Setup Authentication

Step 1: Buat Middleware

php artisan make:middleware AdminMiddleware
php artisan make:middleware PerusahaanMiddleware
php artisan make:middleware MahasiswaMiddleware

Contoh: `AdminMiddleware.php`

```
public function handle(Request $request, Closure $next)
{
    if (!auth()->check()) {
        return redirect()->route('admin.masuk')
            ->with('error', 'Silakan login terlebih dahulu.');
    }

    if (!auth()->user()->adalahAdmin()) {
        return redirect()->route('pilih.peran')
            ->with('error', 'Anda tidak memiliki akses.');
    }

    return $next($request);
}
```

Daftarkan di `bootstrap/app.php`:

```
->withMiddleware(function (Middleware $middleware) {
    $middleware->alias([
        'admin' => \App\Http\Middleware\AdminMiddleware::class,
        'perusahaan' => \App\Http\Middleware\PerusahaanMiddleware::class,
        'mahasiswa' => \App\Http\Middleware\MahasiswaMiddleware::class,
    ]);
})
```

Step 2: Buat Login Controllers

```
php artisan make:controller Auth/DaftarController
php artisan make:controller Auth/LoginAdminController
php artisan make:controller Auth/LoginPerusahaanController
php artisan make:controller Auth/LoginMahasiswaController
```

Contoh: `LoginAdminController.php`

```
public function masuk(Request $request)
{
    $kredensial = $request->validate([
        'email' => 'required|email',
        'password' => 'required',
    ]);

    if (Auth::attempt(array_merge($kredensial, ['peran' => 'admin']))) {
        $request->session()->regenerate();
        return redirect()->route('admin.dashboard');
    }

    return back()->with('error', 'Email atau password salah.');
}
```

FASE 3: Setup Routes

File: `routes/web.php`

```
// Landing & Role Selection
Route::get('/', function () {
    return view('beranda');
})->name('beranda');
```

```
Route::get('/masuk', function () {
    return view('auth.pilih-peran');
})->name('pilih.peran');

// Admin Routes
Route::prefix('admin')->name('admin.')->group(function () {
    Route::get('/masuk', [LoginAdminController::class, 'tampilkanFormMasuk'])->name('masuk');
    Route::post('/masuk', [LoginAdminController::class, 'masuk']);

    Route::middleware(['auth', 'admin'])->group(function () {
        Route::get('/dashboard', [DashboardAdminController::class, 'index'])->name('dashboard');
        // ... routes lainnya
    });
});

// Perusahaan Routes (sama seperti admin)
// Mahasiswa Routes (sama seperti admin)
```

FASE 4: Buat Views

Struktur View Blade:

```
resources/views/
├── beranda.blade.php (landing page)
├── auth/
│   ├── pilih-peran.blade.php (3 card: Admin, Perusahaan, Mahasiswa)
│   ├── masuk/ (form login per role)
│   └── daftar/ (form register perusahaan & mahasiswa)
├── admin/ (dashboard + verifikasi)
├── perusahaan/ (dashboard + CRUD lowongan + pelamar)
└── mahasiswa/ (dashboard + cari lowongan + lamaran)
```

Contoh: `pilih-peran.blade.php`

```
<div class="card-container">
    <div class="card admin">
        <h2>Admin</h2>
        <a href="{{ route('admin.masuk') }}">Masuk Admin</a>
    </div>

    <div class="card perusahaan">
```

```
        <h2>Perusahaan</h2>
        <a href="{{ route('perusahaan.masuk') }}">Masuk</a>
        <a href="{{ route('perusahaan.daftar') }}">Daftar</a>
    </div>

    <div class="card mahasiswa">
        <h2>Mahasiswa</h2>
        <a href="{{ route('mahasiswa.masuk') }}">Masuk</a>
        <a href="{{ route('mahasiswa.daftar') }}">Daftar</a>
    </div>
</div>
```

FASE 5: Implementasi Fitur

A. Register Perusahaan & Mahasiswa

Controller: `DaftarController.php`

```
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\DB;

public function daftarPerusahaan(Request $request)
{
    $request->validate([
        'email' => 'required|email|unique:users,email',
        'password' => 'required|min:8|confirmed',
        'nama_perusahaan' => 'required',
        // ... validasi lainnya
    ]);

    DB::beginTransaction();
    try {
        $user = User::create([
            'name' => $request->nama_perusahaan,
            'email' => $request->email,
            'password' => Hash::make($request->password),
            'peran' => 'perusahaan',
        ]);

        Perusahaan::create([
            'user_id' => $user->id,
            'nama_perusahaan' => $request->nama_perusahaan,
```

```
            // ... data lainnya
        ]);

        DB::commit();
        return redirect()->route('perusahaan.masuk')->with('success', 'Registrasi berhasil!');
    } catch (\Exception $e) {
        DB::rollback();
        return back()->with('error', 'Terjadi kesalahan');
    }
}
```

B. CRUD Lowongan (Perusahaan)

Controller: `LowonganController.php`

```
public function simpan(Request $request)
{
    $request->validate([
        'posisi' => 'required',
        'deskripsi' => 'required',
        'batas_akhir' => 'required|date|after:today',
        // ... validasi lainnya
    ]);

    $perusahaan = auth()->user()->perusahaan;

    Lowongan::create([
        'perusahaan_id' => $perusahaan->id,
        'posisi' => $request->posisi,
        // ... data lainnya
        'status' => 'menunggu', // menunggu approval admin
    ]);

    return redirect()->route('perusahaan.lowongan.index')
        ->with('success', 'Lowongan berhasil dibuat!');
}
```

C. Upload CV & Daftar Lowongan (Mahasiswa)

Controller: `PendaftaranController.php`

```
public function daftar(Request $request, $lowongan_id)
{
```

```
    $request->validate([
        'cv' => 'required|file|mimes:pdf|max:2048', // Max 2MB
    ]);

    $mahasiswa = auth()->user()->mahasiswa;

    // Upload CV
    $file = $request->file('cv');
    $namaFile = 'cv_' . $mahasiswa->nim . '_' . time() . '.pdf';
    $jalurCV = $file->storeAs('cv', $namaFile, 'public');

    // Simpan pendaftaran
    Pendaftaran::create([
        'lowongan_id' => $lowongan_id,
        'mahasiswa_id' => $mahasiswa->id,
        'jalur_cv' => $jalurCV,
        'status' => 'menunggu',
    ]);

    return redirect()->route('mahasiswa.lamaran.index')
        ->with('success', 'Lamaran berhasil dikirim!');
}
```

D. Storage Link untuk File Upload

```
php artisan storage:link
mkdir storage/app/public/cv
```

---

6. FITUR-FITUR APLIKASI

🔐 ADMIN

1. **Login** dengan email & password
2. **Dashboard** dengan statistik (total perusahaan, mahasiswa, lowongan, lamaran)
3. **Verifikasi Perusahaan** (Setujui/Tolak pendaftaran perusahaan)
4. **Verifikasi Lowongan** (Setujui/Tolak lowongan yang dibuat perusahaan)
5. **Laporan Rekap Pendaftar** (Statistik lamaran per lowongan, per status)

🏢 PERUSAHAAN

1. **Register & Login**

2. **Dashboard** dengan statistik lowongan
3. **CRUD Lowongan** (Create, Read, Update, Delete)
4. **Lihat Daftar Pelamar** per lowongan
5. **Download CV** pelamar (format PDF)
6. **Ubah Status Pelamar** (Terima/Tolak)
7. **Status Verifikasi** dari admin (Menunggu/Disetujui/Ditolak)

🎓 MAHASISWA

1. **Register & Login**
2. **Dashboard** dengan statistik lamaran
3. **Cari Lowongan** (dengan filter tipe: magang/kerja)
4. **Lihat Detail Lowongan**
5. **Upload CV & Daftar Lowongan** (PDF max 2MB)
6. **Riwayat Lamaran** (tracking status)
7. **Lihat Status Lamaran** (Menunggu/Diterima/Ditolak)

---

7. TESTING & PENGGUNAAN

A. Jalankan Aplikasi

```
# Start server Laravel
php artisan serve

# Buka browser
http://localhost:8000
```

B. Login Credentials Default

Role	Email	Password
Admin	admin@ownyourcareer.com	admin123
Perusahaan	(daftar sendiri)	-
Mahasiswa	(daftar sendiri)	-

C. Skenario Testing Lengkap

Scenario 1: Register Perusahaan

1. Buka http://localhost:8000
2. Klik **"Masuk Sekarang"**
3. Pilih **"Perusahaan"**
4. Klik **"Belum punya akun? Daftar"**
5. Isi form registrasi
6. Klik **"Daftar Sekarang"**
7. Login dengan email & password yang baru dibuat

Scenario 2: Admin Approve Perusahaan

1. Login sebagai **Admin**
2. Klik **"Verifikasi Perusahaan"**
3. Klik **"✓ Setujui"** pada perusahaan yang mendaftar

Scenario 3: Perusahaan Buat Lowongan

1. Login sebagai **Perusahaan** (yang sudah disetujui)
2. Klik **"Buat Lowongan Baru"**
3. Isi form:
    - Posisi: "Backend Developer"
    - Tipe: "Kerja"
    - Lokasi: "Jakarta"
    - Batas Akhir: (pilih tanggal)
    - Deskripsi: (jelaskan pekerjaan)
    - Persyaratan: (tulis requirements)
4. Klik **"Simpan Lowongan"**
5. Status: **Menunggu Verifikasi Admin**

Scenario 4: Admin Approve Lowongan

1. Login sebagai **Admin**
2. Klik **"Verifikasi Lowongan"**
3. Klik **"Detail"** pada lowongan
4. Klik **"✓ Setujui Lowongan"**
5. Status berubah jadi **Aktif**

Scenario 5: Mahasiswa Daftar Lowongan

1. Daftar akun **Mahasiswa** baru
2. Login sebagai Mahasiswa
3. Klik **"Cari Lowongan"**
4. Klik **"Lihat Detail & Lamar"** pada lowongan Backend Developer
5. Upload file CV (PDF, max 2MB)
6. Klik **"Kirim Lamaran"**

7. Redirect ke **"Riwayat Lamaran"**

Scenario 6: Perusahaan Lihat & Review Pelamar

1. Login sebagai **Perusahaan**
2. Klik **"Lowongan Saya"**
3. Klik **"Lihat Pelamar"** pada lowongan Backend Developer
4. Lihat daftar pelamar
5. Klik **"📄 Download CV"** untuk unduh CV
6. Klik **"✓ Terima"** atau **"✗ Tolak"**

Scenario 7: Mahasiswa Cek Status

1. Login sebagai **Mahasiswa**
2. Klik **"Lamaran Saya"**
3. Lihat status berubah: **Diterima** atau **Ditolak**

---

8. TROUBLESHOOTING

A. Error Umum & Solusi

1. Route [login] not defined

Penyebab: Middleware redirect ke route yang salah

Solusi:

```
# Pastikan middleware redirect ke route yang benar
# AdminMiddleware → admin.masuk
# PerusahaanMiddleware → perusahaan.masuk
# MahasiswaMiddleware → mahasiswa.masuk
```

2. View [nama.view] not found

Penyebab: File blade belum dibuat atau salah path

Solusi:

```
# Cek struktur folder views/
# Pastikan nama file sesuai dengan yang dipanggil di controller
```

3. SQLSTATE: no such table

Penyebab: Migration belum dijalankan

Solusi:

```
php artisan migrate:reset
php artisan migrate
php artisan db:seed --class=AdminSeeder
```

4. Class "App\Http\Controllers..." not found

Penyebab: Controller belum dibuat atau namespace salah

Solusi:

```
# Buat controller yang hilang
php artisan make:controller NamaController

# Atau pastikan namespace sesuai dengan folder
```

5. Storage link error

Penyebab: Symbolic link belum dibuat

Solusi:

```
php artisan storage:link
mkdir storage/app/public/cv
```

B. Clear Cache (Solusi Universal)

```
php artisan route:clear
php artisan config:clear
php artisan cache:clear
php artisan view:clear
composer dump-autoload
```

---

9. TIPS PENGEMBANGAN

A. Best Practices

1. **Gunakan DB Transaction** untuk operasi yang melibatkan multiple tables

2. **Validasi Input** di semua form
3. **Hash Password** selalu dengan `Hash::make()`
4. **Proteksi Route** dengan middleware
5. **Handle Error** dengan try-catch

B. Naming Convention

- **Controller**: PascalCase + `Controller` (contoh: `DaftarController`)
- **Model**: PascalCase singular (contoh: `Lowongan`)
- **Route Name**: lowercase + dot notation (contoh: `admin.lowongan.index`)
- **View**: lowercase + dash (contoh: `pilih-peran.blade.php`)
- **Method**: camelCase (contoh: `tampilkanFormMasuk`)

C. Git Workflow (Opsional)

```
# Initialize git
git init
git add .
git commit -m "Initial commit: Own Your Career"

# Buat branch untuk setiap fitur
git checkout -b feature/authentication
git checkout -b feature/crud-lowongan
```

---

10. RESOURCES & LINKS

Dokumentasi Resmi

- Laravel: https://laravel.com/docs
- SQLite: https://www.sqlite.org/docs.html
- Blade Templates: https://laravel.com/docs/blade

Tools Recommended

- **DB Browser for SQLite**: https://sqlitebrowser.org/
- **DBeaver**: https://dbeaver.io/
- **Postman**: Untuk testing API (jika ada)

📝 CHECKLIST FINAL

Sebelum Presentasi:

- [ ] Semua migration berhasil dijalankan
- [ ] Admin seeder sudah dibuat
- [ ] Storage link sudah dibuat
- [ ] Semua routes terdaftar (`php artisan route:list`)
- [ ] Test 3 role bisa login
- [ ] Test alur lengkap (register → approve → buat lowongan → lamar)
- [ ] Screenshot untuk dokumentasi laporan
- [ ] Database backup (copy file `database.sqlite`)

Saat Presentasi:

- [ ] Jalankan `php artisan serve`
- [ ] Siapkan 3 browser/tab untuk demo 3 role
- [ ] Siapkan file CV dummy untuk testing upload
- [ ] Buat flowchart/diagram untuk menjelaskan alur

🎉 SELAMAT!

Aplikasi **Own Your Career** sudah siap digunakan dan dipresentasikan!

Dibuat oleh:

- Ani Nuri Azizah (K23524023)
- Khoirul Lathifah Oktaviani (K3524027)
- Anisa Ramadhani (K3524045)
- Natasya Luthfia Ramadhani (K3524063)

Prodi: Pendidikan Teknik Informatika dan Komputer
Fakultas: Keguruan dan Ilmu Pendidikan
Universitas: Sebelas Maret
Tahun: 2025

📌 **Catatan Penting:**

- Dokumentasi ini bisa dijadikan lampiran laporan
- Semua kode sudah menggunakan **Bahasa Indonesia** untuk konsistensi
- Struktur database sesuai dengan ERD di laporan praktikum
- Aplikasi menggunakan **SQLite** (gratis, tidak perlu MySQL)

Jika ada pertanyaan atau error, cek bagian Troubleshooting atau tanya ke anggota kelompok yang sudah paham! 💪