

# **PRAKTIKUM PENGEMBANGAN APLIKASI WEB**

## **MODUL 16 Laravel Factory & File Storage**



**Sistem Informasi**  
Telkom University  
Surabaya

Disusun Oleh:  
Aditya Wahyu Hidayatullah – 1204230082 – IS-06-01

**PROGRAM STUDI S1 SISTEM INFORMASI  
FAKULTAS REKAYASA INDUSTRI  
TELKOM UNIVERSITY SURABAYA  
2024**

# DAFTAR ISI

DAFTAR ISI	2
1. Membuat dan Mendefinisikan Factory	3
2. Menjalankan Factory via Seeder	4
3. Meletakkan File pada Local Disk	4
4. Meletakkan File pada Public Disk	4
5. Menampilkan Isi File	5
6. Mendownload File	5
7. Menampilkan URL, Path dan Size dari File	6
8. Menyimpan File via Form	6
9. Menghapus File pada Storage	7
10. Menerapkan Fitur Upload pada Web Data Master Employee	7
11. Latihan	12
12. Tugas	12

# MODUL 15

## LARAVEL FACTORY & FILE STORAGE

Pada praktikum kali ini kita akan melanjutkan belajar tentang Laravel Factory dan File Storage. Mahasiswa akan mencoba menerapkan fitur Laravel Factory untuk men-generate data dummy ke database dan menerapkan Laravel File Storage untuk mengelola asset file pada project laravel yang dimiliki.

### 1. Membuat dan Mendefinisikan Factory

- Buat file Factory untuk tabel **positions** dengan command artisan sebagai berikut:

```
PS D:\xampp\htdocs\Modul 16> php artisan make:factory PositionFactory  
INFO Factory [D:\xampp\htdocs\Modul 16\database\factories\PositionFactory.php] created successfully.
```

- Definisikan value setiap kolom yang dimiliki tabel **positions** dengan bantuan faker untuk men-generate data dummy. Sesuaikan isi function **definition()** seperti kode program di bawah ini.

```
public function definition(): array  
{  
    return [  
        'code' => fake()->stateAbbr(),  
        'name' => fake()->jobTitle(),  
        'description' => fake()->sentence(),  
    ];  
}
```

- Buat file Factory untuk tabel **employees** dengan command artisan sebagai berikut:

```
PS D:\xampp\htdocs\Modul 16> php artisan make:factory EmployeeFactory
```

- Definisikan value setiap kolom yang dimiliki tabel **employees** dengan bantuan faker untuk men-generate data dummy. Sesuaikan isi function **definition()** seperti kode program di bawah ini.

```
public function definition(): array  
{  
    return [  
        'firstname' => fake()->firstName(),  
        'lastname' => fake()->lastName(),  
        'email' => fake()->email(),  
        'age' => fake()->numberBetween(25, 50),  
        'position_id' => Position::factory()  
    ];  
}
```

## 2. Menjalankan Factory via Seeder

- Buat file **PositionSeeder.php** kemudian **generate 5 data dummy** untuk tabel **positions** dengan menuliskan kode program seperti di bawah ini.

```
Position::factory()->count(5)->create();
```

- Buat file **EmployeeSeeder.php** kemudian **generate 10 data dummy** untuk tabel **employees** dengan menuliskan kode program seperti di bawah ini.

```
Employee::factory()->count(10)->create(); // Menggunakan factory
```

- Jalankan seeder yang telah disesuaikan dengan command di bawah ini.

```
PS D:\xampp\htdocs\Modul 16> php artisan db:seed

INFO Seeding database.

Database\Seeders\PositionSeeder ..... RUNNING
Database\Seeders\PositionSeeder ..... 7,964 ms DONE

Database\Seeders\EmployeeSeeder ..... RUNNING

Illuminate\Database\UniqueConstraintViolationException

SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'purnama.anaking@gmail.com' for key 'employees_email unique' (Connection: mysql, SQL: insert into `employees` (`age`, `email`, `firstname`, `lastname`, `position_id`) values (20, purnama.anaking@gmail.com, Purnama, Anaking, 1), (25, adzanil.rachmadhi@gmail.com, Adzanil, Rachmadhi, 2), (23, berlian.rahmy@gmail.com, Berlian, Rahmy, 3))

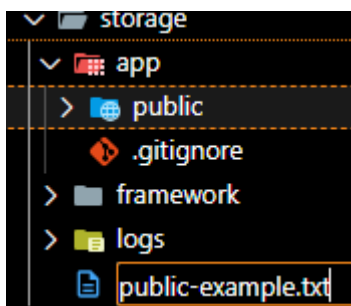
at vendor\laravel\framework\src\Illuminate\Database\Connection.php:824
  820 // message to include the bindings with SQL, which will make this exception a
  821 // lot more helpful to the developer instead of just the database's errors.
  822 catch (Exception $e) {
```

## 3. Meletakkan File pada Local Disk

- Buka file **routes/web.php** kemudian buat Route baru dengan kode program seperti di bawah ini. Kemudian akses Route tersebut via Browser.

```
Route::get('/local-disk', function () {
    Storage::disk('local')->put('local-example.txt', 'This is local example content');
    return asset('storage/local-example.txt');
});
```

- Jika berhasil akan terbuat file dengan nama **local-example.txt** di folder **/storage/app** pada project laravel anda.



#### 4. Meletakkan File pada Public Disk

- Buat **Symbolic Link** yang menghubungkan antara direktori **/storage/app/public** dengan direktori **/public/storage** via command di bawah ini:

```
php artisan storage:link
```

- Buka file **routes/web.php** kemudian buat Route baru dengan kode program seperti di bawah ini. Kemudian akses Route tersebut via Browser.

```
Route::get('/public-disk', function () {  
    Storage::disk('public')->put('public-example.txt', 'This is public example content');  
    return asset('storage/public-example.txt');  
});
```

- Jika berhasil akan terbuat file dengan nama **public-example.txt** di folder **/storage/app/public** pada project laravel anda.
- Anda dapat mengakses file yang sudah terbuat tersebut via browser dengan URL:

```
http://localhost:8000/storage/public-example.txt
```

- URL di atas mengakses file melalui folder **/public/storage** yang memiliki “**symbolic link**” ke direktori **/storage/app/public**.

#### 5. Menampilkan Isi File

- Menampilkan isi file **local** dengan cara buat Route baru dengan kode program di bawah ini.

```
Route::get('/retrieve-local-file', function () {  
    if (Storage::disk('local')->exists('local-example.txt')) {  
        $contents = Storage::disk('local')->get('local-example.txt');  
    } else {  
        $contents = 'File does not exist';  
    }  
  
    return $contents;  
});
```

- Menampilkan isi file **public** dengan cara buat Route baru dengan kode program di bawah ini.

```
Route::get('/retrieve-public-file', function () {
    if (Storage::disk('public')->exists('public-example.txt')) {
        $contents = Storage::disk('public')->get('public-example.txt');
    } else {
        $contents = 'File does not exist';
    }

    return $contents;
});
```

- Akses route-route tersebut via browser lalu analisis apa yang ditampilkan pada browser.

## 6. Mendownload File

- Mendownload file **local** dengan cara buat Route baru dengan kode program di bawah ini.

```
Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/download-local-file', function () {
    return Storage::download('local-example.txt', 'local file');
});
```

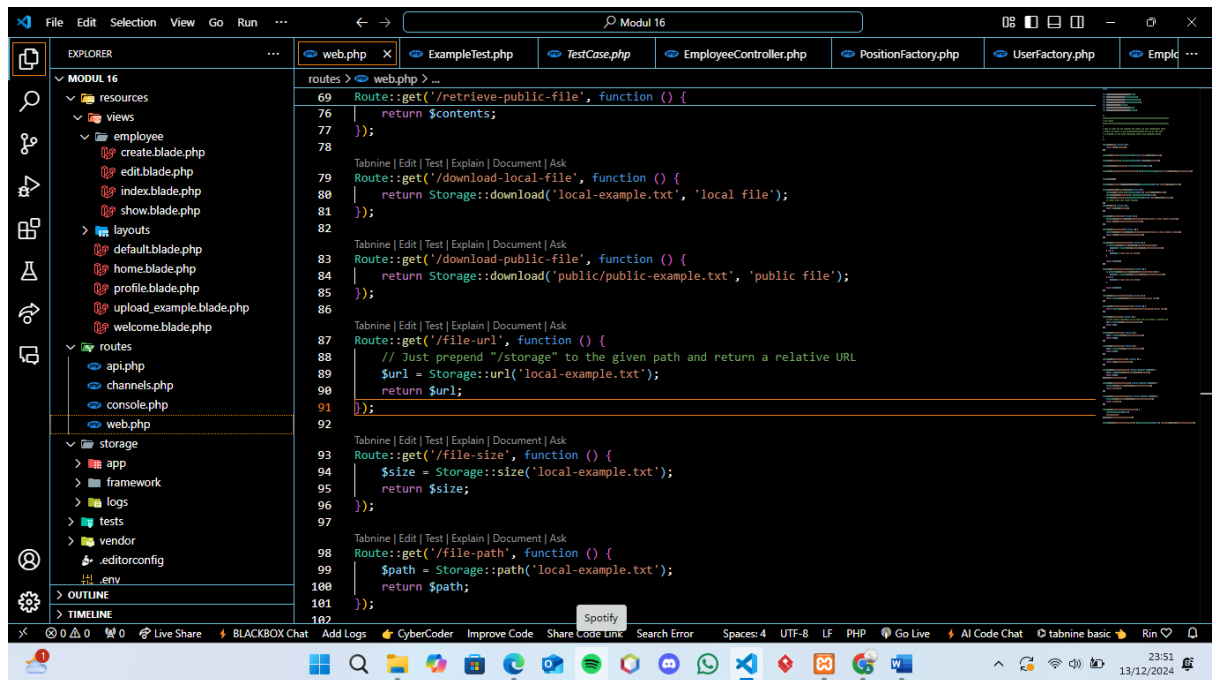
- Mendownload file **public** dengan cara buat Route baru dengan kode program di bawah ini.

```
Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/download-public-file', function () {
    return Storage::download('public/public-example.txt', 'public file');
});
```

- Akses route-route tersebut via browser lalu analisis apa yang ditampilkan pada browser.

## 7. Menampilkan URL, Path dan Size dari File

- Buat route baru dengan kode program di bawah ini.



- Akses route-route tersebut via browser lalu analisis apa yang ditampilkan pada browser.

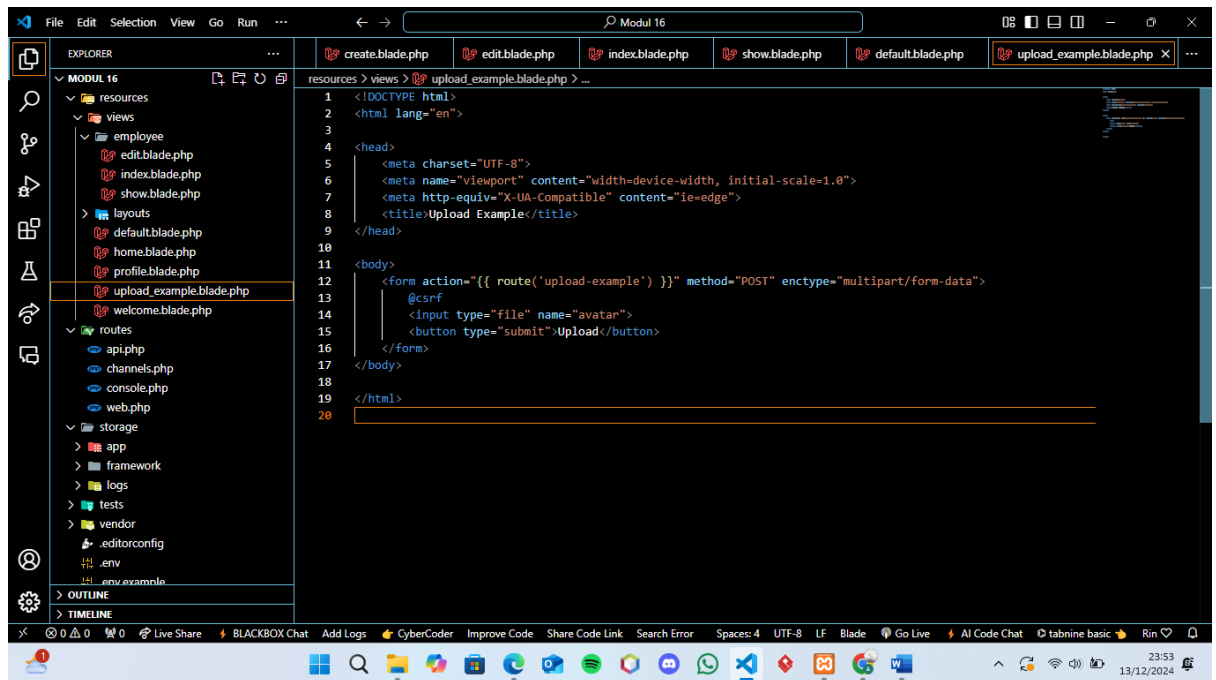
## 8. Menyimpan File via Form

- Buat route baru dengan kode program di bawah ini.

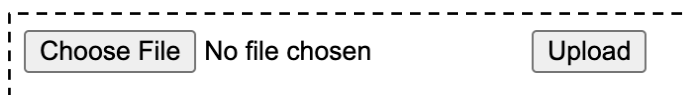
```
Route::get('/upload-example', function () {
    return view('upload_example');
});

Route::post('/upload-example', function (Request $request) {
    $path = $request->file('avatar')->store('public');
    return $path;
})->name('upload-example');
```

- Buat file view dengan nama **upload\_example.blade.php**. Tulis kode program seperti di bawah ini.



- Akses URL **/upload-example** via browser, pilih salah satu file, lalu klik button **Upload**.



- Jika berhasil maka file yang diupload akan tersimpan pada direktori **/storage/app/public**.

## 9. Menghapus File pada Storage

- Buat route baru dengan kode program di bawah ini.

```
Route::get('/delete-local-file', function(Request $request) {
    Storage::disk('local')->delete('local-example.txt');
    return 'Deleted';
});

Route::get('/delete-public-file', function(Request $request) {
    Storage::disk('public')->delete('public-example.txt');
    return 'Deleted';
});
```

- Akses route-route tersebut via browser lalu analisis apa yang terjadi.
- Jika berhasil maka file yang bersangkutan akan terhapus pada project laravel anda.

## 10. Menerapkan Fitur Upload pada Web Data Master Employee

- Buat file migration baru untuk menambahkan kolom yang digunakan menyimpan data nama file **original** dan yang **terenkripsi**. Tulis command di bawah ini:

```
php artisan make:migration add_filename_column_into_employees_table
```



- Buka file migration yang baru saja ter-generate. Tulis kode program di bawah ini untuk menambahkan 2 kolom baru dengan nama **“original\_filename”** dan **“encrypted\_filename”**.


```
/**
 * Run the migrations.
 */
public function up(): void
{
    Schema::table('employees', function (Blueprint $table) {
        $table->string('original_filename')->after('position_id')->nullable();
        $table->string('encrypted_filename')->after('original_filename')->
        nullable();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::table('employees', function (Blueprint $table) {
        $table->dropColumn('encrypted_filename');
        $table->dropColumn('original_filename');
    });
}
```

- Jalankan **“php artisan migrate”** untuk meng-eksekusi file migration yang baru saja kita buat. Jika berhasil maka akan muncul 2 kolom baru pada tabel **employees** di database anda.
- Buka file **/views/employee/create.blade.php**, sesuaikan fitur **Form Create Employee** dengan atribut **enctype** pada tag **<form>**.

```
<form action="{{ route('employees.store') }}" method="POST"
enctype="multipart/form-data">
```

- Tambahkan kode program di bawah ini setelah bagian yang menampilkan pilihan **“Position”** pada form.



### Create Employee

---

First Name

Last Name

Email

Age

Position

MD - Sociology Teacher

Curriculum Vitae (CV)

Choose File

No file chosen

---

⏪ Cancel

✓ Save

- Buka file **EmployeeController.php** kemudian sesuaikan function **store()** seperti di bawah ini.

```

1 public function store(Request $request)
2 {
3     $messages = [
4         'required' => ':Attribute harus diisi.',
5         'email' => 'Isi :attribute dengan format yang benar',
6         'numeric' => 'Isi :attribute dengan angka'
7     ];
8
9     $validator = Validator::make($request->all(), [
10         'firstName' => 'required',
11         'lastName' => 'required',
12         'email' => 'required|email',
13         'age' => 'required|numeric',
14     ], $messages);
15
16     if ($validator->fails()) {
17         return redirect()->back()->withErrors($validator)->withInput();
18     }
19
20     // Get File
21     $file = $request->file('cv');
22
23     if ($file != null) {
24         $originalFilename = $file->getClientOriginalName();
25         $encryptedFilename = $file->hashName();
26
27         // Store File
28         $file->store('public/files');
29     }
30
31     // ELOQUENT
32     $employee = new Employee;
33     $employee->firstname = $request->firstName;
34     $employee->lastname = $request->lastName;
35     $employee->email = $request->email;
36     $employee->age = $request->age;
37     $employee->position_id = $request->position;
38
39     if ($file != null) {
40         $employee->original_filename = $originalFilename;
41         $employee->encrypted_filename = $encryptedFilename;
42     }
43
44     $employee->save();
45
46     return redirect()->route('employees.index');
47 }

```

- Jalankan fitur **Form Create Employee** via browser, jika berhasil maka data employee akan masuk ke database dan file CV akan tersimpan di direktori **/storage/app/public/files** pada project laravel anda.

- Buat route baru pada file **/routes/web.php** dengan menambahkan kode program seperti di bawah ini untuk operasi **download file**.

```
Route::get('download-file/{employeeId}', [
    EmployeeController::class,
    'downloadFile'
])->name('employees.downloadFile');
```

- Buka file **/views/employee/show.blade.php**, sesuaikan kode program seperti di bawah ini.

```
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container-sm my-5">
5         <div class="row justify-content-center">
6             <div class="p-5 bg-light rounded-3 col-xl-4 border">
7                 <div class="mb-3 text-center">
8                     <i class="bi-person-circle fs-1"></i>
9                     <h4>Detail Employee</h4>
10                </div>
11                <hr>
12                <div class="row">
13                    <div class="col-md-4 mb-3">
14                        <label for="firstName" class="form-label">First Name</label>
15                        <h5>{{ $employee->firstname }}</h5>
16                    </div>
17                    <div class="col-md-4 mb-3">
18                        <label for="lastName" class="form-label">Last Name</label>
19                        <h5>{{ $employee->lastname }}</h5>
20                    </div>
21                    <div class="col-md-4 mb-3">
22                        <label for="age" class="form-label">Age</label>
23                        <h5>{{ $employee->age }}</h5>
24                    </div>
25                    <div class="col-md-12 mb-3">
26                        <label for="email" class="form-label">Email</label>
27                        <h5>{{ $employee->email }}</h5>
28                    </div>
29                    <div class="col-md-12 mb-3">
30                        <label for="age" class="form-label">Position</label>
31                        <h5>{{ $employee->position->name }}</h5>
32                    </div>
33                    <div class="col-md-12 mb-3">
34                        <label for="age" class="form-label">Curriculum Vitae (CV)</label>
35                        @if ($employee->original_filename)
36                            <h5>{{ $employee->original_filename }}</h5>
37                            <a href="{{ route('employees.downloadFile', ['employeeId' => $employee->id]) }}"
38                                class="btn btn-primary btn-sm mt-2">
39                                <i class="bi bi-download me-1"></i> Download CV
40                            </a>
41                        @else
42                            <h5>Tidak ada</h5>
43                        @endif
44                    </div>
45                </div>
46                <hr>
47                <div class="row">
48                    <div class="col-md-12 d-grid">
49                        <a href="{{ route('employees.index') }}" class="btn btn-outline-dark btn-lg mt-3"><i
50                            class="bi-arrow-left-circle me-2"></i> Back</a>
51                    </div>
52                </div>
53            </div>
54        </div>
55    </div>
56 @endsection
```

- Buka file **EmployeeController.php** lalu tambahkan function **downloadFile()** seperti di bawah ini.

```

tabnine | Edit | Test | Explain | Document | Ask
public function downloadFile($employeeId)
{
    $employee = Employee::find($employeeId);
    $encryptedFilename = 'public/files/' . $employee->encrypted_filename;
    $downloadFilename = Str::lower($employee->firstname . '_' . $employee->lastname . '_cv.pdf');

    if (Storage::exists($encryptedFilename)) {
        return Storage::download($encryptedFilename, $downloadFilename);
    }
}

```

- Jika berhasil maka ketika anda mengakses halaman employee detail, maka akan tampilan button **“download”** CV dari employee. Dan CV employee dapat di download dengan melakukan klik pada button tersebut.

**Detail Employee**

First Name	Last Name	Age
Purnama	Anaking	30

Email  
purnama@gmail.com

Position  
**Sociology Teacher**


Curriculum Vitae (CV)  
11 - PUR - PBO - Interface & Abstract Class (Modul Praktikum).pdf

[Download CV](#)

[Back](#)

## 11. Latihan

- Lanjutkan **Fitur Edit Employee**
  - Edit data employee di database
  - Ubah file CV nya dengan cara: file CV yang sebelumnya, dihapus dari project laravel anda dan diganti dengan file CV yang baru.



### Edit Employee

First Name

Purnama

Last Name

Anaking

Email

purnama@gmail.com

Age

20

Position

MD - Sociology Teacher

Curriculum Vitae (CV)

Surat Pengantar Dosen.pdf


Download

Choose File

No file chosen

Cancel

Edit



### Edit Employee

First Name

Purnama

Last Name

Anaking

Email

purnama.anaking@gmail.com

Age

20

Position

FE - Front End Developer

Curriculum Vitae (CV)

Choose File

laravel.png

Cancel

Edit

- Lanjutkan **Fitur Hapus Employee**
  - Hapus data employee dari database
  - Hapus file CV nya dari project laravel anda

## 12. Tugas

Link github: <https://github.com/AdityaWahyuH/Modul-16-Paw>