

**PRAKTIKUM MODUL 09**  
**PEMROGRAMAN FRAMEWORK**  
**LARAVEL FACTORY & FILE STORAGE**



**DISUSUN OLEH:**

**1204200081   RAFAEL FILLAH FIRMANSYAH**

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

## Membuat dan Mendefinisikan Factory

Link Github: <https://github.com/elfill03/Pemrograman-Framework>

### 1. Membuat sebuah file Factory untuk tabel positions

```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba> php artisan make:factory PositionFactory
```

```
INFO Factory [C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba\database\factories\PositionFactory.php] created successfully.
```

### 2. Kemudian menambahkan pendefinisian penerapan faker untuk code, name, dan description

```
PositionFactory.php X
database > factories > PositionFactory.php

10 class PositionFactory extends Factory
11 {
12     /**
13      * Define the model's default state.
14      *
15      * @return array<string, mixed>
16      */
17     public function definition(): array
18     {
19         return [
20             'code' => fake()->stateAbbr(),
21             'name' => fake()->jobTitle(),
22             'description' => fake()->sentence(),
23         ];
24     }
25 }
```

### 3. Membuat sebuah file Factory untuk tabel Employee

```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba> php artisan make:factory EmployeeFactory
```

```
INFO Factory [C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba\database\factories\EmployeeFactory.php] created successfully.
```

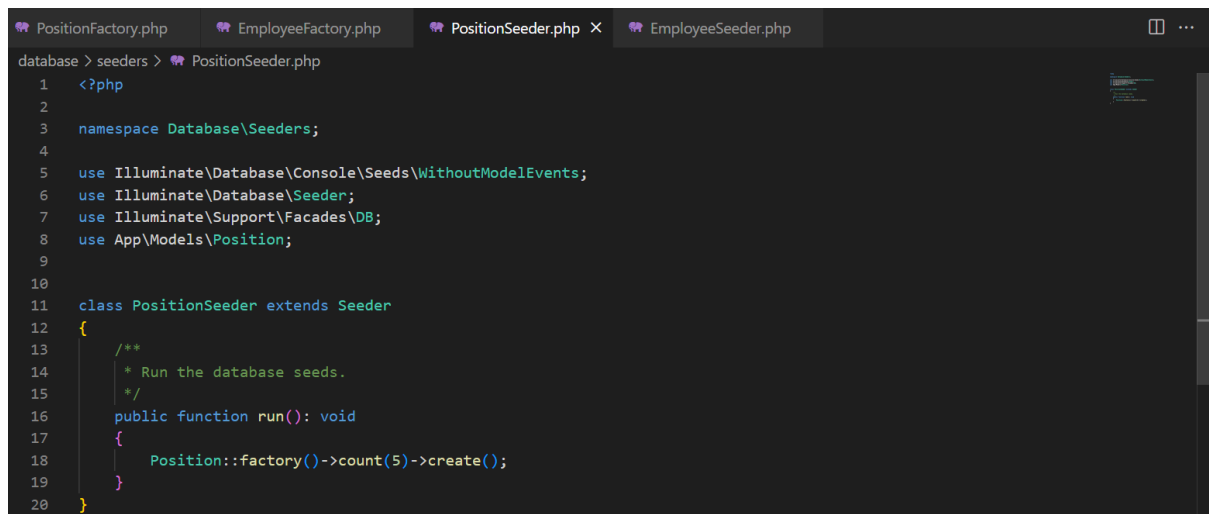
### 4. Kemudian menambahkan pendefinisian penerapan faker untuk firstname, lastname, email, age, position\_id

```
PositionFactory.php EmployeeFactory.php X PositionSeeder.php EmployeeSeeder.php
database > factories > EmployeeFactory.php

10 class EmployeeFactory extends Factory
11 {
12     /**
13      * Define the model's default state.
14      *
15      * @return array<string, mixed>
16      */
17     public function definition(): array
18     {
19         return [
20             'firstname' => fake()->firstName(),
21             'lastname' => fake()->lastName(),
22             'email' => fake()->email(),
23             'age' => fake()->numberBetween(25, 50),
24             'position_id' => rand(1, 5)
25         ];
26     }
27 }
```

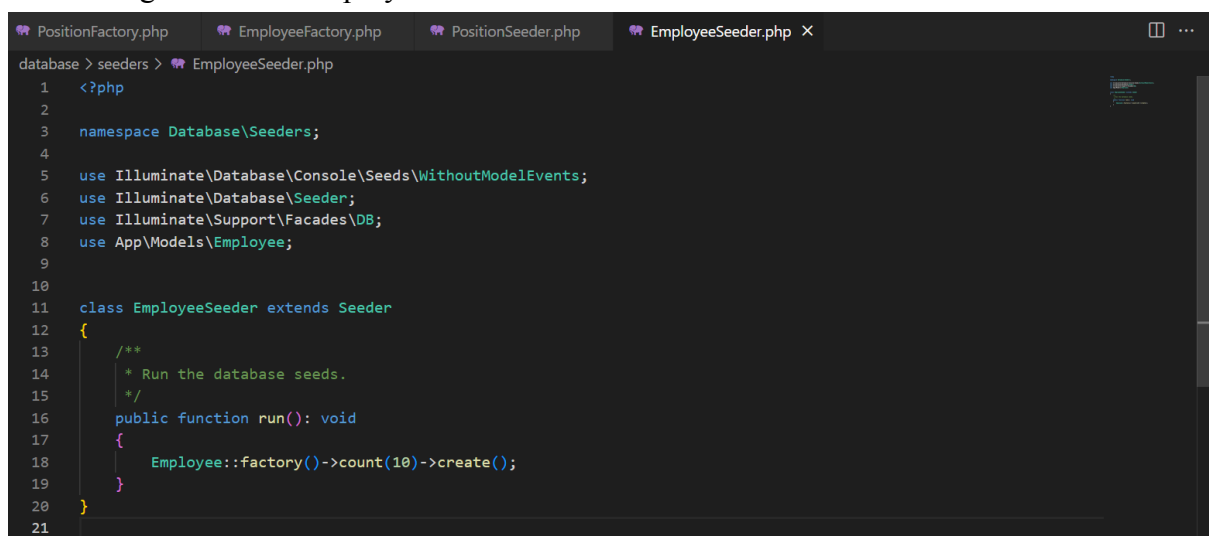
## Menjalankan Factory via Seeder

1. Mengubah PositionSeeder yang awalnya masih manual menjadi menggunakan factory untuk mengenerate 5 position



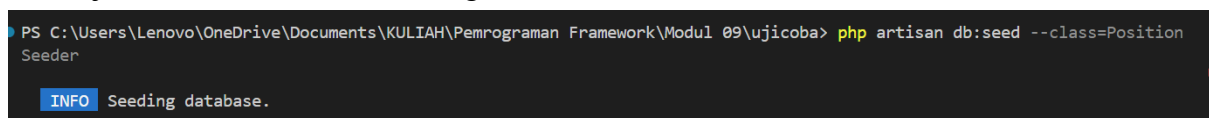
```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Facades\DB;
8 use App\Models\Position;
9
10
11 class PositionSeeder extends Seeder
12 {
13     /**
14      * Run the database seeds.
15      */
16     public function run(): void
17     {
18         Position::factory()->count(5)->create();
19     }
20 }
```

2. Mengubah EmployeeSeeder yang awalnya masih manual menjadi menggunakan factory untuk mengenerate 10 employee



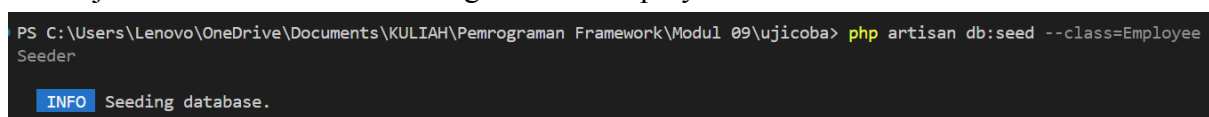
```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Facades\DB;
8 use App\Models\Employee;
9
10
11 class EmployeeSeeder extends Seeder
12 {
13     /**
14      * Run the database seeds.
15      */
16     public function run(): void
17     {
18         Employee::factory()->count(10)->create();
19     }
20 }
21
```

3. Menjalankan seeder via class dengan nama PositionSeeder



```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba> php artisan db:seed --class=PositionSeeder
INFO Seeding database.
```

4. Menjalankan seeder via class dengan nama EmployeeSeeder

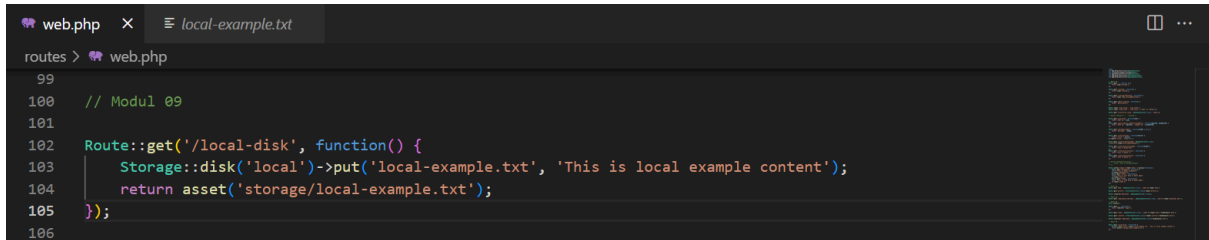


```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba> php artisan db:seed --class=EmployeeSeeder
INFO Seeding database.
```

- 5.

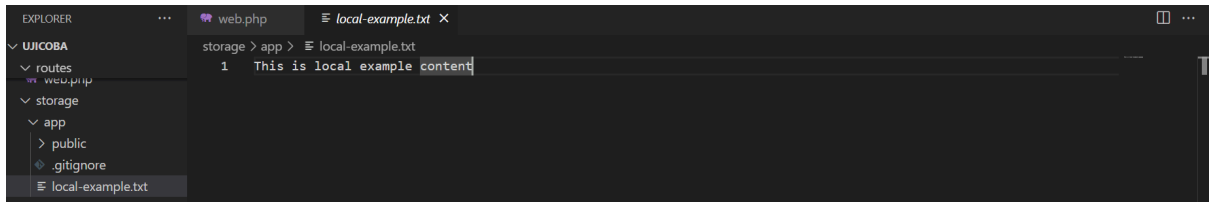
## Meletakkan File pada Local Disk

1. Membuat sebuah route baru untuk local-disk yang dimana apabila diakses pada browser maka akan men generate file local-example.txt pada folder storage



```
web.php x local-example.txt
routes > web.php
99
100 // Modul 09
101
102 Route::get('/local-disk', function() {
103     Storage::disk('local')->put('local-example.txt', 'This is local example content');
104     return asset('storage/local-example.txt');
105 });
106
```

2. Hasil generate file



```
EXPLORER
UJICOBA
  routes
  web.php
  storage
    app
      public
      .gitignore
      local-example.txt
  local-example.txt

storage > app > local-example.txt
1 This is local example content
```

## Meletakkan File pada Public Disk

1. Buat Symbolic Link yang menghubungkan antara direktori /storage/app/public dengan direktori /public/storage

```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba> php artisan storage:link

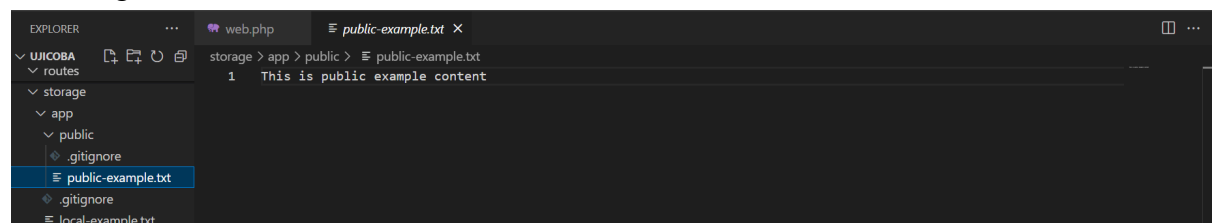
INFO The [C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba\public\storage] link has been connected to [C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba\storage\app\public].
```

2. Membuat sebuah route baru untuk public-disk yang dimana apabila diakses pada browser maka akan generate file public-example.txt pada folder public



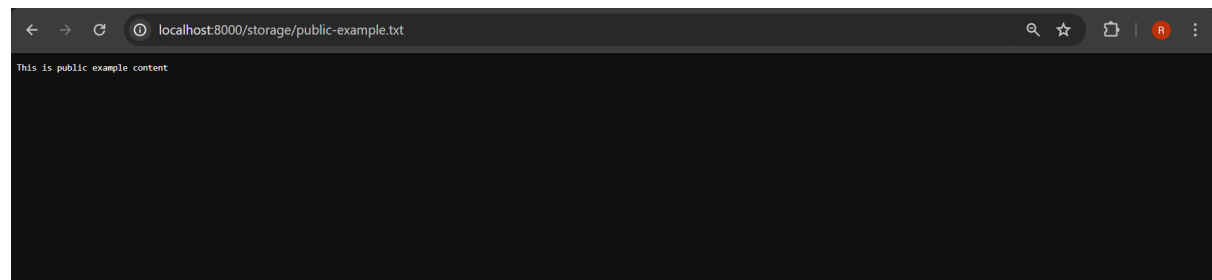
```
99
100 // Modul 09
101
102 Route::get('/local-disk', function() {
103     Storage::disk('local')->put('local-example.txt', 'This is local example content');
104     return asset('storage/local-example.txt');
105 });
106
107 Route::get('/public-disk', function() {
108     Storage::disk('public')->put('public-example.txt', 'This is public example content');
109     return asset('storage/public-example.txt');
110 });
111
```

3. Hasil generate file



```
storage > app > public > public-example.txt
1 This is public example content
```

4. Setelah diakses di browser



## Menampilkan Isi File

1. Menampilkan isi file local dengan cara buat Route baru pada web.php untuk local file dan public



```
web.php x
routes > web.php
111
112 Route::get('/retrieve-local-file', function() {
113     if (Storage::disk('local')->exists('local-example.txt')) {
114         $contents = Storage::disk('local')->get('local-example.txt');
115     } else {
116         $contents = 'File does not exist';
117     }
118
119     return $contents;
120 });
121
122
123 Route::get('/retrieve-public-file', function() {
124     if (Storage::disk('public')->exists('public-example.txt')) {
125         $contents = Storage::disk('public')->get('public-example.txt');
126     } else {
127         $contents = 'File does not exist';
128     }
129
130     return $contents;
131 });
```

2. Untuk /retrieve-local-file apabila diakses akan menampilkan isi dari local file, apabila tidak ada maka akan menampilkan “File does not exist”



3. Untuk /retrieve-public-file apabila diakses akan menampilkan isi dari public file, apabila tidak ada maka akan menampilkan “File does not exist”



## Mendownload File

1. Membuat route baru untuk download file lokal dan public



```
web.php x index.blade.php
routes > web.php > ...
147 Route::get('/download-local-file', function () {
148     return Storage::download('local-example.txt', 'local file');
149 });
150
151 Route::get('/download-public-file', function () {
152     return Storage::download('public/public-example.txt', 'public file');
153 });
```

2. Apabila diakses urlnya maka akan mendownload file tergantung url lokal atau public

# Menampilkan URL, Path dan Size dari File

## 1. Membuat route baru untuk file url, file size, dan file path



```
web.php x index.blade.php
routes > web.php > Closure
154
155 Route::get('/file-url', function () {
156     $url = Storage::url('local-example.txt');
157     return $url;
158 });
159 Route::get('/file-size', function () {
160     $size = Storage::size('local-example.txt');
161     return $size;
162 });
163 Route::get('/file-path', function () {
164     $path = Storage::path('local-example.txt');
165     return $path;
166 });
```

## 2. Apabila mengakses file url akan menampilkan url dari file local



## 3. Apabila mengakses file size akan menampilkan size dari file local



## 4. Apabila mengakses file path akan menampilkan path dari file local





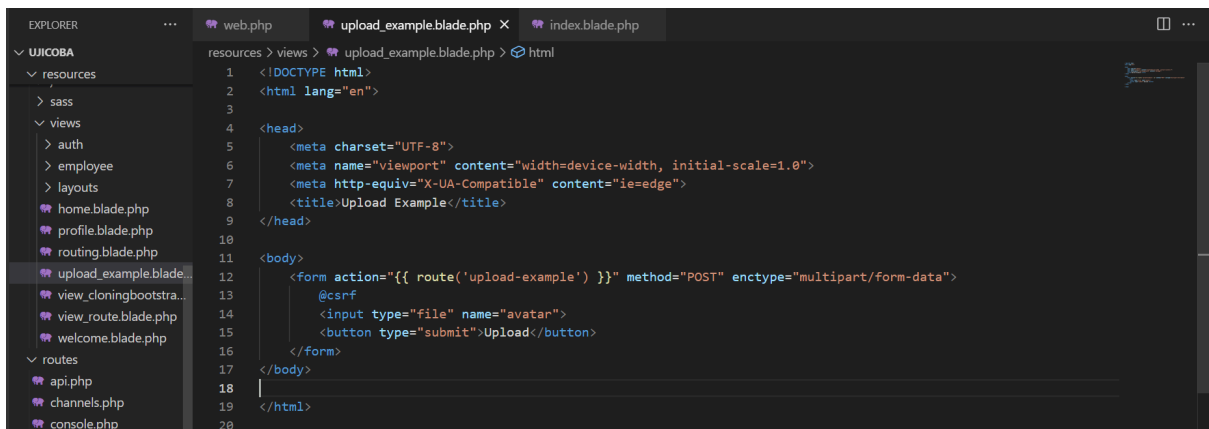
# Menyimpan File via Form

## 1. Membuat sebuah route untuk mengupload file



```
routes > web.php > ...
168
169 Route::get('/upload-example', function () {
170     return view('upload_example');
171 });
172 Route::post('/upload-example', function (Request $request) {
173     $path = $request->file('avatar')->store('public');
174     return $path;
175 })->name('upload-example');
176
```

## 2. Kemudian membuat file view baru yang berisi form untuk upload file



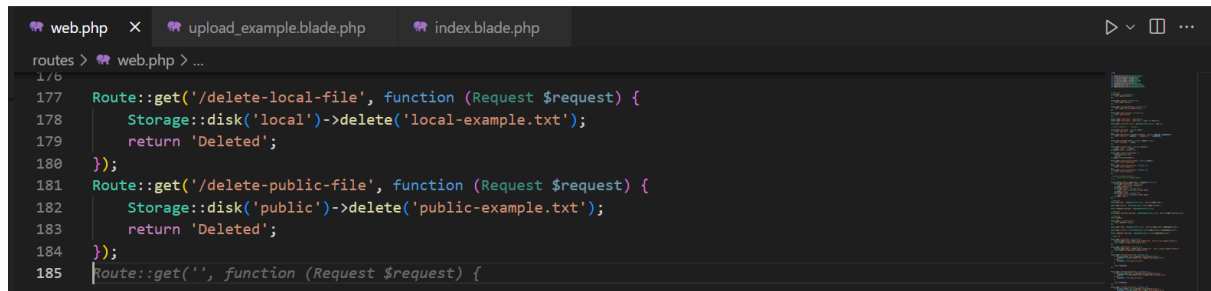
```
EXPLORER
resources > views > upload_example.blade.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <meta http-equiv="X-UA-Compatible" content="ie=edge">
8     <title>Upload Example</title>
9 </head>
10
11 <body>
12     <form action="{{ route('upload-example') }}" method="POST" enctype="multipart/form-data">
13         @csrf
14         <input type="file" name="avatar">
15         <button type="submit">Upload</button>
16     </form>
17 </body>
18
19 </html>
20
```

## 3. Apabila url di akses maka akan muncul form seperti berikut



## Menghapus File pada Storage

1. Buat route baru untuk melakukan hapus file yang ada pada storage



```
routes > web.php > ...  
177 Route::get('/delete-local-file', function (Request $request) {  
178     Storage::disk('local')->delete('local-example.txt');  
179     return 'Deleted';  
180 });  
181 Route::get('/delete-public-file', function (Request $request) {  
182     Storage::disk('public')->delete('public-example.txt');  
183     return 'Deleted';  
184 });  
185 Route::get('', function (Request $request) {
```

2. Apabila url tersebut diakses maka akan menghapus file pada storage

## Menerapkan Fitur Upload pada Web Data Master Employee

1. Buat sebuah file migration baru file migration baru untuk menambahkan kolom yang digunakan menyimpan data nama file original dan yang terenkripsi.

```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba> php artisan make:migration add_filename_column_into_employees_table

INFO Migration [C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba\database\Migrations\2024_05_17_080206_add_filename_column_into_employees_table.php] created successfully.
```

2. Kemudian definisikan skema tabel seperti berikut untuk original\_filename dan encrypted\_filename

```
web.php 2024_05_17_080206_add_filename_column_into_employees_table.php EmployeeSeeder.php upload_example.blade.php
database > migrations > 2024_05_17_080206_add_filename_column_into_employees_table.php > class > down
7 return new class extends Migration
11
12 public function up(): void
13 {
14     Schema::table('employees', function (Blueprint $table) {
15         $table->string('original_filename')->after('position_id')->nullable();
16         $table->string('encrypted_filename')->after('original_filename')->nullable();
17     });
18 }
19
20 /**
21  * Reverse the migrations.
22  */
23 public function down(): void
24 {
25     Schema::table('employees', function (Blueprint $table) {
26         $table->dropColumn('encrypted_filename');
27         $table->dropColumn('original_filename');
28     });
29 }
30 };
```

3. Kemudian jalankan dengan php artisan migrate atau php artisan migrate:fresh

```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba> php artisan migrate

INFO Preparing database.

Creating migration table ..... 100ms DONE

INFO Running migrations.

2014_10_12_000000_create_users_table ..... 70ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 5ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 44ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 27ms DONE
2024_04_01_043045_create_positions_table ..... 5ms DONE
2024_04_01_043507_create_employees_table ..... 78ms DONE
2024_05_17_080206_add_filename_column_into_employees_table ..... 4ms DONE

PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 09\ujicoba> php artisan db:seed

INFO Seeding database.

Database\Seeders\UserSeeder ..... RUNNING
Database\Seeders\UserSeeder ..... 336 ms DONE

Database\Seeders\PositionSeeder ..... RUNNING
Database\Seeders\PositionSeeder ..... 620 ms DONE

Database\Seeders\EmployeeSeeder ..... RUNNING
Database\Seeders\EmployeeSeeder ..... 54 ms DONE
```

4. Kemudian sesuaikan fitur Form Create Employee dengan atribut enctype pada tag <form>. Guna dari enctype adalah untuk mengirimkan file tersebut ke server.

```
web.php create.blade.php index.blade.php
resources > views > employee > create.blade.php > div.container-sm.mt-5 > form
1 @extends('layouts.app')
2 @section('content')
3 <div class="container-sm mt-5">
4 <form action="{{ route('employees.store') }}" method="POST" enctype="multipart/form-data">
```

5. Kemudian tambahkan form input untuk file cv pada halaman create employee

```

55 <div class="col-md-12 mb-3">
56 <label for="position" class="form-label">Position</label>
57 <select name="position" id="position" class="form-select">
58     @foreach ($positions as $position)
59         <option value="{{ $position->id }}"
60             {{ old('position') == $position->id ? 'selected' : '' }}>
61             {{ $position->code . ' - ' . $position->name }}
62         </option>
63     @endforeach
64 </select>
65 </div>
66 <div class="col-md-12 mb-3">
67 <label for="cv" class="form-label">Curriculum Vitae (CV)</label>
68 <input type="file" class="form-control" name="cv" id="cv">
69 </div>

```

6. Kemudian pada EmployeeController bagian function store tambahkan logic untuk menyimpan file cv

```

web.php create.blade.php EmployeeController.php X index.blade.php
app > Http > Controllers > EmployeeController.php > PHP Intelephense > EmployeeController > store
11 class EmployeeController extends Controller
35 public function store(Request $request)
51 {
52
53     // Get File
54     $file = $request->file('cv');
55     if ($file != null) {
56         $originalFilename = $file->getClientOriginalName();
57         $encryptedFilename = $file->hashName();
58         // Store File
59         $file->store('public/files');
60     }
61
62
63     $employee = new Employee;
64     $employee->firstname = $request->firstName;
65     $employee->lastname = $request->lastName;
66     $employee->email = $request->email;
67     $employee->age = $request->age;
68     $employee->position_id = $request->position;
69     if ($file != null) {
70         $employee->original_filename = $originalFilename;
71         $employee->encrypted_filename = $encryptedFilename;
72     }
73
74     $employee->save();
75
76     return redirect()->route('employees.index');
77 }

```

7. Kemudian buat sebuah route baru untuk mendownload file cv dari employee berdasarkan id

```

web.php X create.blade.php EmployeeController.php index.blade.php
routes > web.php > ...
184
185 Route::get('download-file/{employeeId}', [
186     EmployeeController::class, 'downloadFile'
187 ]->name('employees.downloadFile'));
188

```

8. Pada halaman show.blade.php atau detail employee tambahkan informasi cv dari employee dan tombol untuk download cv

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

9. Pada EmployeeController buat function tambahan dengan nama downloadFile yang memiliki logic untuk melakukan download file cv dari employee berdasarkan id

```
web.php create.blade.php show.blade.php EmployeeController.php X index.blade.php
app > Http > Controllers > EmployeeController.php > ...
13 class EmployeeController extends Controller
181
182     1 reference | 0 overrides
    public function downloadFile($employeeId)
183     {
184         $employee = Employee::find($employeeId);
185         $encryptedFilename = 'public/files/' . $employee->encrypted_filename;
186         $downloadFilename =
187             Str::lower($employee->firstname . '_' . $employee->lastname . '_cv.pdf');
188         if (Storage::exists($encryptedFilename)) {
189             return Storage::download($encryptedFilename, $downloadFilename);
190         }
191     }
192 }
193
```

10. Terakhir yaitu tambahkan import statement dari request, storage, dan str.

```
web.php create.blade.php show.blade.php EmployeeController.php X index.blade.php
app > Http > Controllers > EmployeeController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Validator;
7 use Illuminate\Support\Facades\DB;
8 use Illuminate\Support\Facades\Storage;
9 use Illuminate\Support\Str;
10 use App\Models\Employee;
11 use App\Models\Position;
12
```

# LATIHAN

## Mengubah Fitur Edit Besert CV

### 1. Menambahkan enctype pada tag form di halaman edit employee

```
resources > views > employee > edit.blade.php > div.container-sm.mt-5 > form > div.row.justify-content-center > div.p-5.bg-light.rounded-3.border.col-xl-6 >
1 @extends('layouts.app')
2 @section('content')
3 <div class="container-sm mt-5">
4 <form action="{{ route('employees.update', $employee->id) }}" method="POST" enctype="multipart/form-data">
```

### 2. Menambahkan bagian untuk menampilkan CV pada halaman edit employee serta form untuk upload cv baru

```
68 <div class="col-md-12 mb-3">
69 <label for="age" class="form-label">Curriculum Vitae
70 (CV)</label>
71 @if ($employee->original_filename)
72 <h5>{{ $employee->original_filename }}</h5>
73 <a href="{{ route('employees.downloadFile', ['employeeId' => $employee->id]) }}"
74 class="btn btn-primary btn-sm mt-2">
75 <i class="bi bi-download me-1"></i> Download CV
76 </a>
77 @else
78 <h5>Tidak ada</h5>
79 @endif
80 </div>
81 <div class="col-md-12 mb-3">
82 <label for="cv" class="form-label">Upload New CV</label>
83 <input type="file" class="form-control" name="cv" id="cv">
84 </div>
```

### 3. Ubah function update pada EmployeeController untuk melakukan update cv, dengan logic atau alur yaitu pertama menghapus file terlebih dahulu dari storage kemudian menyimpan file cv yang baru kedalam folder public dan melakukan update pada employee dengan file terbaru.

```
web.php show.blade.php edit.blade.php EmployeeController.php X
app > Http > Controllers > EmployeeController.php > PHP > App\Http\Controllers\EmployeeController > update()
13 class EmployeeController extends Controller
109 public function update(Request $request, string $id)
110 {
136 // Update File CV
137 if ($request->hasFile('cv')) {
138 // Delete old CV
139 if ($employee->encrypted_filename) {
140 Storage::delete('public/files/' . $employee->encrypted_filename);
141 }
142
143 // Store new CV file
144 $file = $request->file('cv');
145 $originalFilename = $file->getClientOriginalName();
146 $encryptedFilename = $file->hashName();
147 $file->store('public/files');
148
149 // Update employee CV
150 $employee->original_filename = $originalFilename;
151 $employee->encrypted_filename = $encryptedFilename;
152 }
153
154 $employee->save();
155
156 return redirect()->route('employees.index');
157 }
```

### 4. Tampilan dari fitur edit employee yang telah ada cvnya

👤

Edit Employee

First Name

Last Name

rafael

fillah

Email

Age

zztes@gmail.com

22

Position

CA - Electronics Engineer ▾

Curriculum Vitae (CV)

11 - PUR - PF - Factory File Storage (Modul 09 Praktikum).pdf

📄 Download CV

Upload New CV

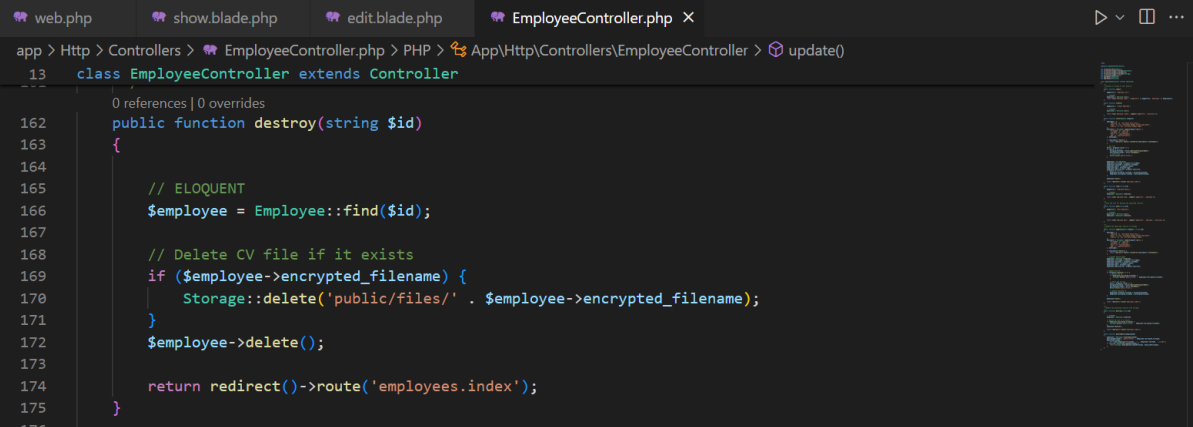
Choose File

No file chosen



## Hapus Employee Beserta CV

1. Pada EmployeeController bagian function destroy/delete ubah logic yang awalnya hanya hapus employee menjadi file cv yang telah tersimpan ikut terhapus. Pertama penggunaan eloquent untuk memilih/mencari id yang akan dipilih untuk dihapus lalu tambahkan logic untuk menghapus file cvnya yang ada pada storage.



```
13 class EmployeeController extends Controller
14 {
15     // ELOQUENT
16     public function destroy(string $id)
17     {
18         // Delete CV file if it exists
19         if ($employee->encrypted_filename) {
20             Storage::delete('public/files/' . $employee->encrypted_filename);
21         }
22         $employee->delete();
23         return redirect()->route('employees.index');
24     }
25 }
```

2.