

PRAKTIKUM PENGEMBANGAN APLIKASI WEB

MODUL 16 Laravel Factory & File Storage



Sistem Informasi
Telkom University
Surabaya

Disusun Oleh:
Dhini julia ningrum - 1204230058

**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

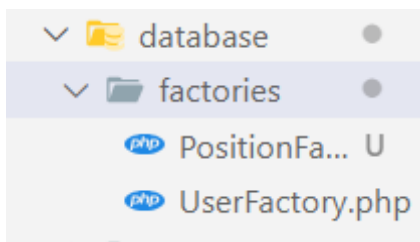
Pada praktikum kali ini kita akan melanjutkan belajar tentang Laravel Factory dan File Storage. Mahasiswa akan mencoba menerapkan fitur Laravel Factory untuk men-generate data dummy ke database dan menerapkan Laravel File Storage untuk mengelola asset file pada project laravel yang dimiliki.

1. Membuat dan Mendefinisikan Factory

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

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  DEVD
PS C:\xampp\modul16pw> php artisan make:factory PositionFactory

INFO Factory [C:\xampp\modul16pw\database\factories\PositionFactory.php] created successfully.
PS C:\xampp\modul16pw> █
```



- 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.

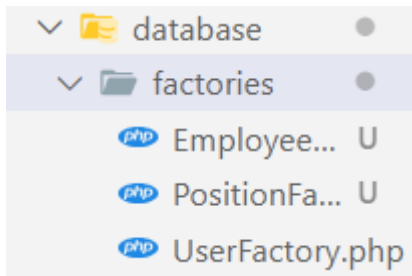
```
database > factories > PositionFactory.php
1 public function definition(): array
2 {
3     return [
4         'code' => fake()->stateAbbr(),
5         'name' => fake()->jobTitle(),
6         'description' => fake()->sentence(),
7     ];
8 }
9
```

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

• PS C:\xampp\modul16pw> `php artisan make:factory EmployeeFactory`

INFO Factory [C:\xampp\modul16pw\database\factories\EmployeeFactory.php] created successfully.

• PS C:\xampp\modul16pw> █



- 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.

```
database > factories > EmployeeFactory.php
1  public function definition(): array
2  {
3      return [
4          'firstname' => fake()->firstName(),
5          'lastname' => fake()->lastName(),
6          'email' => fake()->email(),
7          'age' => fake()->numberBetween(25, 50),
8          'position_id' => Position::factory()
9      ];
10 }
11
```

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.

```

1  public function run(): void
2  {
3      Position::factory()->count(5)->create();
4
5      DB::table('positions')->insert([
6          [
7              'position_name' => 'Software Engineer',
8              'salary' => 10000000,
9              'experience' => 5,
10             'status' => 'active'
11         ],
12         [
13             'position_name' => 'Data Analyst',
14             'salary' => 8000000,
15             'experience' => 3,
16             'status' => 'active'
17         ],
18         [
19             'position_name' => 'Product Manager',
20             'salary' => 12000000,
21             'experience' => 7,
22             'status' => 'active'
23         ],
24         [
25             'position_name' => 'Marketing Specialist',
26             'salary' => 7000000,
27             'experience' => 4,
28             'status' => 'active'
29         ],
30         [
31             'position_name' => 'Sales Representative',
32             'salary' => 6000000,
33             'experience' => 2,
34             'status' => 'active'
35         ]
36     ]);
37 }

```

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

```
public function run(): void
{
    Employee::factory()->count(10)->create();
    DB::table('employees')->insert([
```

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



```
PS C:\xampp\modul16pw> php artisan db:seed

INFO Seeding database.

Database\Seeders\PositionSeeder ..... RUNNING

Error

Class "Database\Seeders\Position" not found

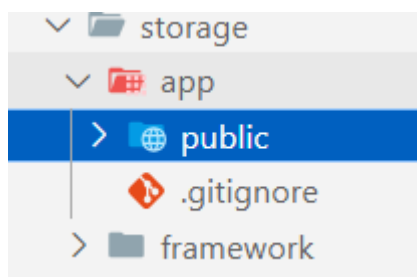
at database\seeders\PositionSeeder.php:17
13      */
14
15      public function run(): void
16      {
17          Position::factory()->count(5)->create();
18
19          DB::table('positions')->insert([
20              [
21                  'code' => 'FE',
```

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:

```
PS C:\xampp\modul16pw> php artisan storage:link
```

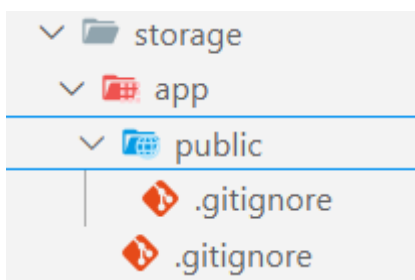
```
INFO The [C:\xampp\modul16pw\public\storage] link has been connected to [C:\xampp\modul16pw\storage\app\public].
```

```
PS C:\xampp\modul16pw>
```

- 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 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.

```

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.

```

Route::get('/download-local-file', function() {
    return Storage::download('local-example.txt', 'local file');
});
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.

```
Route::get('/file-url', function() {
    // Just prepend "/storage" to the given path and return a relative URL
    $url = Storage::url('local-example.txt');
    return $url;
});

Route::get('/file-size', function() {
    $size = Storage::size('local-example.txt');
    return $size;
});

Route::get('/file-path', function() {
    $path = Storage::path('local-example.txt');
    return $path;
});
```

- 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.

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

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

No file chosen

Create Employee

First Name

Last Name

Email

Age

Position

FE - Front End Developer

Curriculum Vitae (CV)

No file chosen

- 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:

```
PS C:\xampp\modul16pw> php artisan make:migration add_filename_column_into_employees_table
INFO Migration [C:\xampp\modul16pw\database\migrations\2024_12_24_093100_add_filename_column_into_employees_table.php] created successfully.
PS C:\xampp\modul16pw>
```

- 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->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>**.

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

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

```
<div class="col-md-12 mb-3">
    <label for="cv" class="form-label">Curriculum Vitae (CV)</label>
    <input type="file" class="form-control" name="cv" id="cv">
</div>
```

- 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 b
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',
```

- 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.

Laravel Administrator

Create Employee

First Name

Last Name

dhini

julia

Email

Age

dhinijulia@gmail.com

19

Position

FE - Front End Developer

Curriculum Vitae (CV)

Choose File Screenshot 2024-12-29 231317.png

Cancel

Save

- 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.download');
```

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

```
resources > views > employee > show.blade.php > ...
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>
34             </div>
35         </div>
36     </div>
37 @endsection
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS DEVD

PS C:\xampp\modul116pw>

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

```


49 public function downloadFile($employeeId)
50 {
51     $employee = Employee::find($employeeId);
52     $encryptedFilename = 'public/files/'.$employee->encrypted_filename;
53     $downloadFilename = Str::lower($employee->firstname.'_'.$employee->lastname.'_cv.pdf');
54
55     if(Storage::exists($encryptedFilename)) {
56         return Storage::download($encryptedFilename, $downloadFilename);
57     }
58 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS DEVDB

PS C:\xampp\modul16pw>

- Jika berhasil maka ketika anda mengakses halaman employee detail, maka akan tampilkan 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.

The image displays two versions of the 'Edit Employee' form side-by-side. Both forms have a header with a user icon and the title 'Edit Employee'. The form fields include: First Name (Purnama), Last Name (Anaking), Email (purnama@gmail.com), Age (20), and Position (MD - Sociology Teacher). The 'Curriculum Vitae (CV)' section is present in both. In the left version, the CV section shows 'Tidak ada' and a 'Download' button. In the right version, the CV section shows 'Surat Pengantar Dosen.pdf' and a 'Download' button. Both forms have a 'Choose File' button and a 'No file chosen' message. At the bottom, there are 'Cancel' and 'Edit' buttons.

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

12. Tugas

1. **Praktekan** seluruh poin **praktikum dan latihan** yang ada di atas secara **INDIVIDU**. Buat pada **1 repository** Github saja. Infokan URL Repository tersebut pada laporan praktikum.
2. Dokumentasikan seluruhnya (**screenshot output**) dalam bentuk Laporan Praktikum.
3. Kumpulkan Laporan Praktikum (**.pdf**) yang telah dibuat di dalam praktikum via **LMS** paling lambat sebelum jadwal praktikum minggu depan.

Github : <https://github.com/dhini/n/modul-16-paw.git>