

PEMROGRAMAN FRAMEWORK

LAPORAN PRAKTIKUM MODUL 5

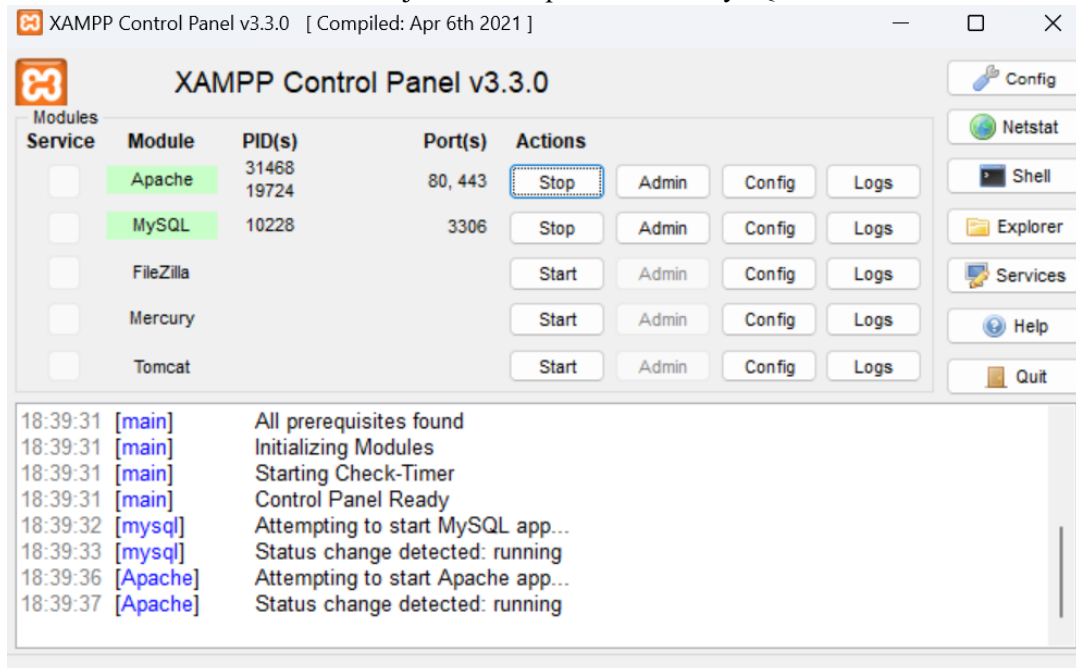


Disusun Oleh:
Ferdi Zahi Robbani
1204220104

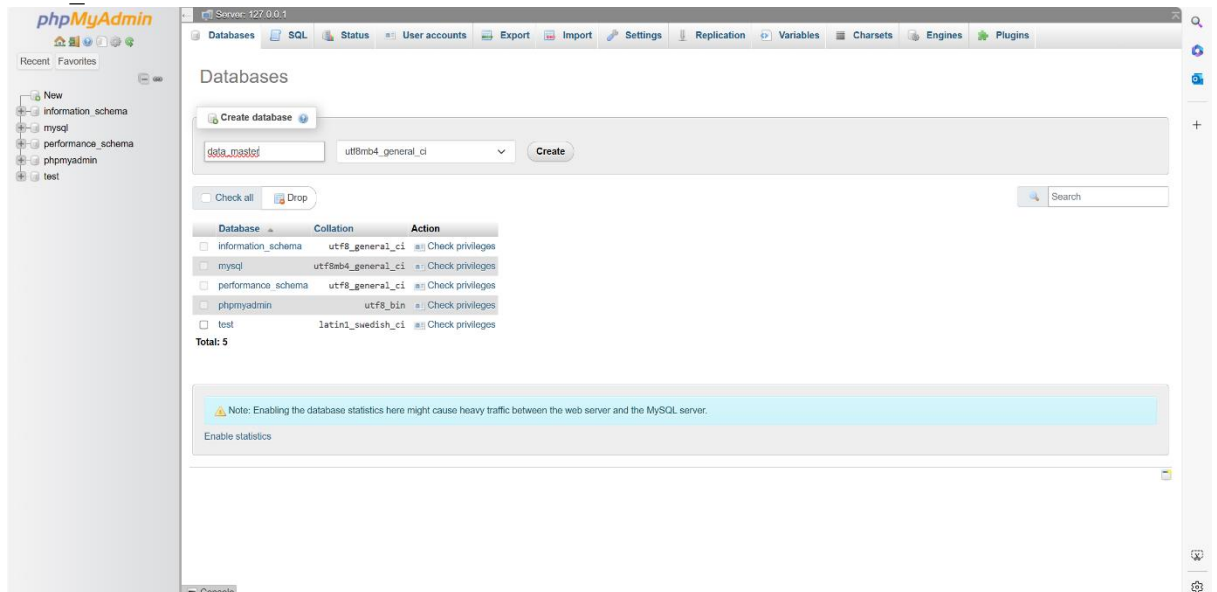
PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
TELKOM UNIVERSITY SURABAYA
2024

1. Aktifkan Service MySQL dan PhpMyAdmin

- Buka XAMPP Control Panel dan jalankan “Apache” dan “MySQL”.



- Buka “PhpMy Admin” pada browser kemudian buat database baru dengan nama “data_master”.

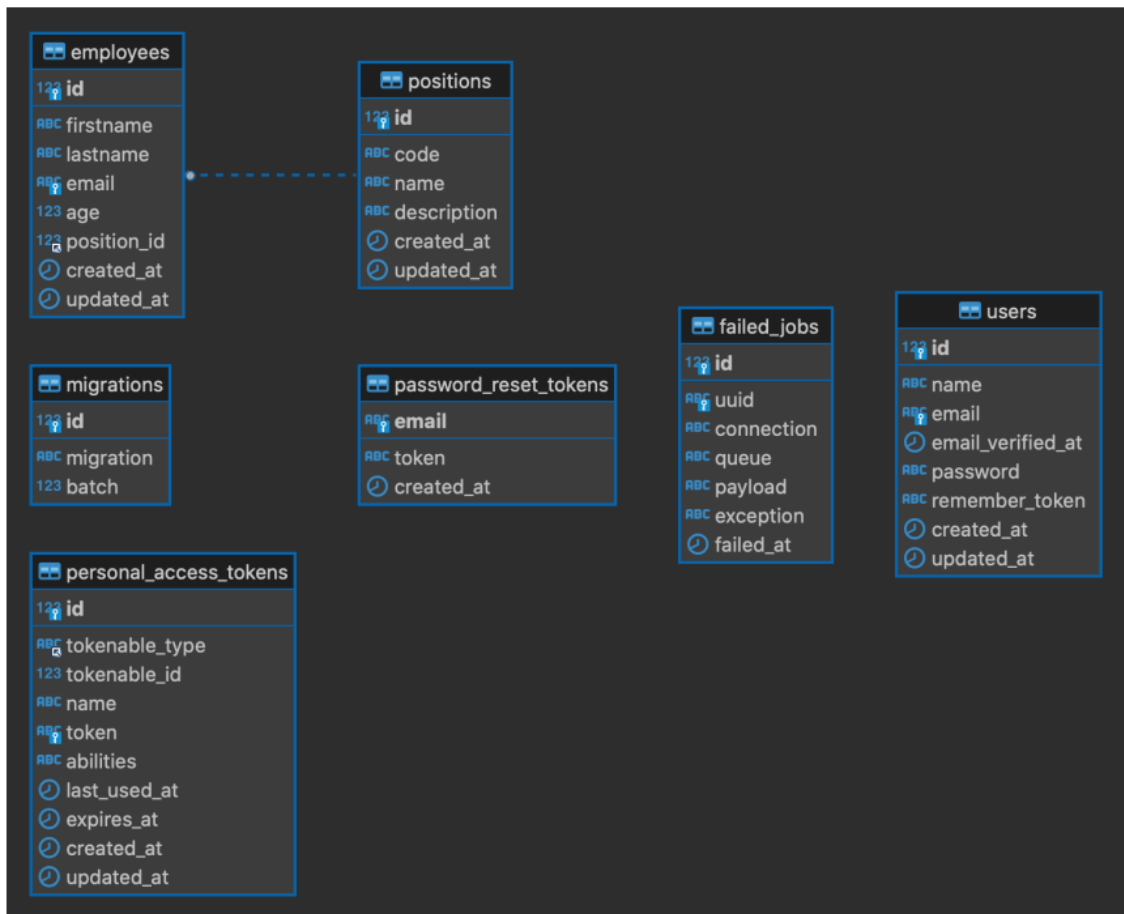


2. Setup Database Config

- Buka file “.env” pada project laravel yang telah dibuat sebelumnya. Kemudian set variable “DB_DATABASE” dengan nama databsae yang telah dibuat sebelumnya (“data_master”).

```
22 DB_CONNECTION=sqlite
23 DB_HOST=127.0.0.1
24 DB_PORT=3306
25 DB_DATABASE=data_master
26 DB_USERNAME=root
27 DB_PASSWORD=
```

3. Skema Database



4. Membuat Skema Database Menggunakan Migration

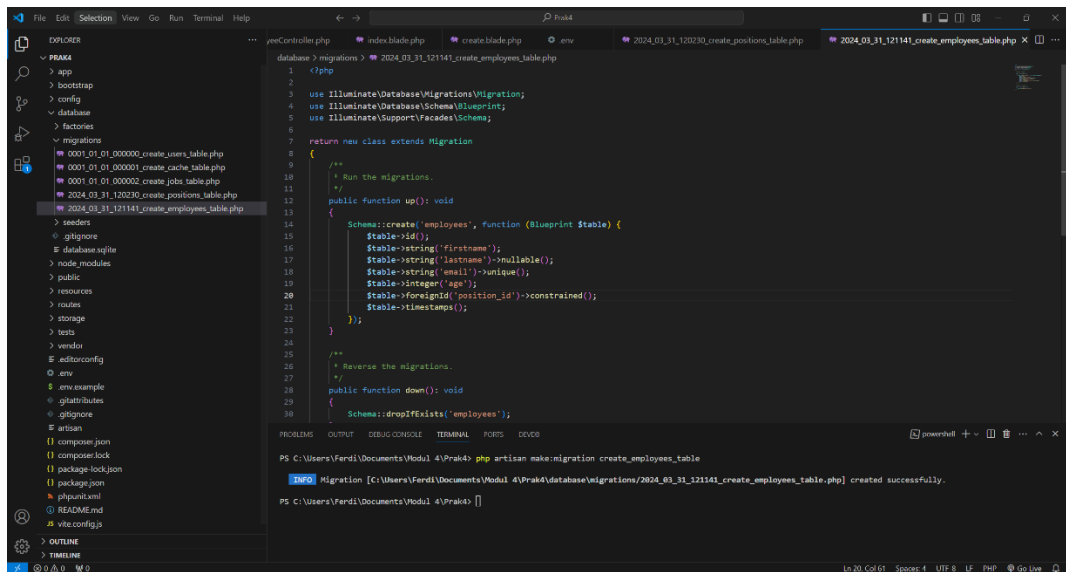
- Generate file migration untuk tabel Positions menggunakan artisan.

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('positions', function (Blueprint $table) {
15             $table->id();
16             $table->timestamps();
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::dropIfExists('positions');
26     }
27
28 };
29
```

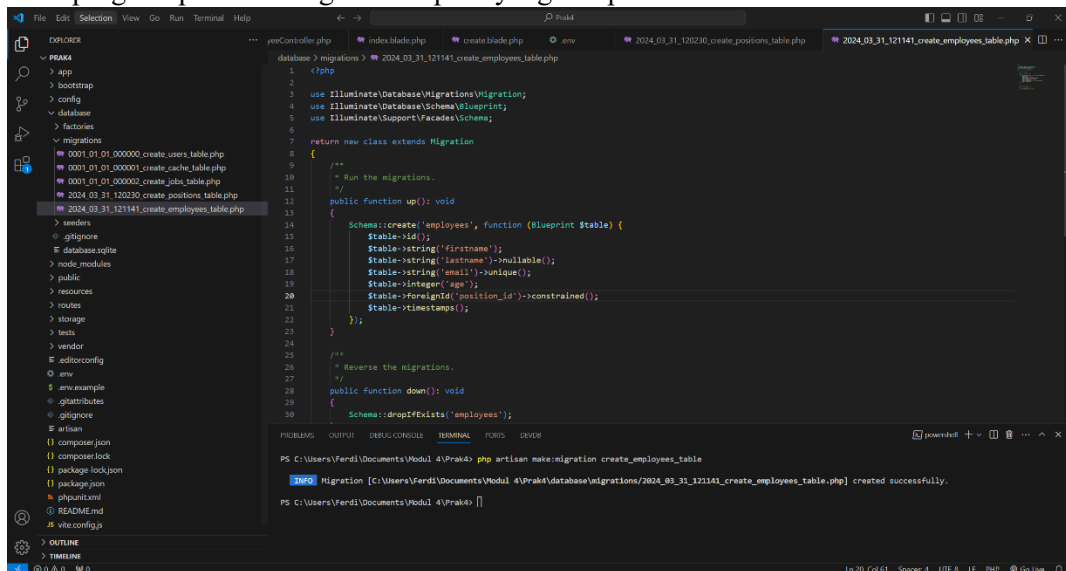
Terminal output:

```
PS C:\Users\Ferd\Documents\Modul 4\Prak4> php artisan make:migration create_positions_table
INFO: Migration [C:\Users\Ferd\Documents\Modul 4\Prak4\database\migrations\2024_03_31_120230_create_positions_table.php] created successfully.
PS C:\Users\Ferd\Documents\Modul 4\Prak4>
```

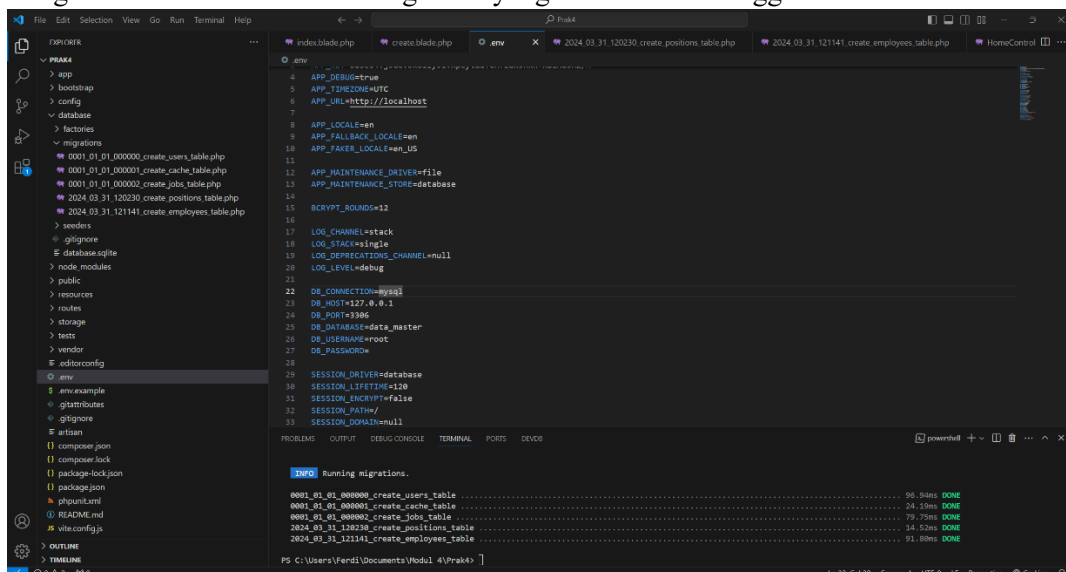
- Buat code program pada file migration seperti yang ada pada modul.



- Lalu, generate file migration untuk tabel Employees menggunakan artisan, kemudian buat kode program pada file migration seperti yang ada pada modul.



- Langkah terakhir eksekusi file migration yang telah dibuat menggunakan artisan.

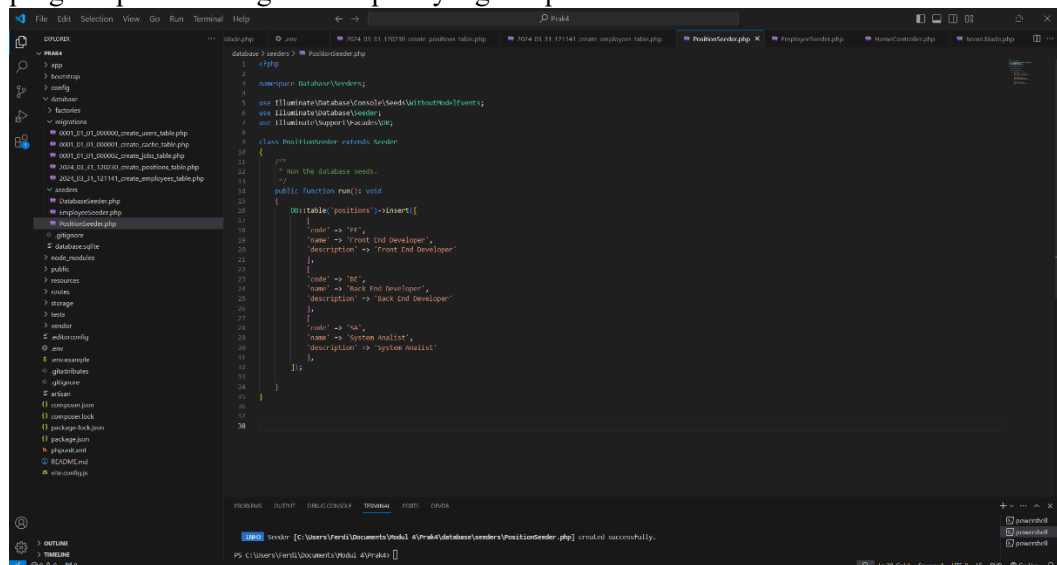


Penjelasan:

Pada laporan modul saya saat ini, saya menggunakan Laravel 11 dimana “DB_CONNECTION” yang digunakan adalah “sqlite”, dan disini saya menggantinya menjadi “mysql” dan “meng-uncomment” valuenya lalu di set sesuai yang diperlukan dan setelah di eksekusi pada file migrationnya akan menghasilkan output seperti gambar di atas.

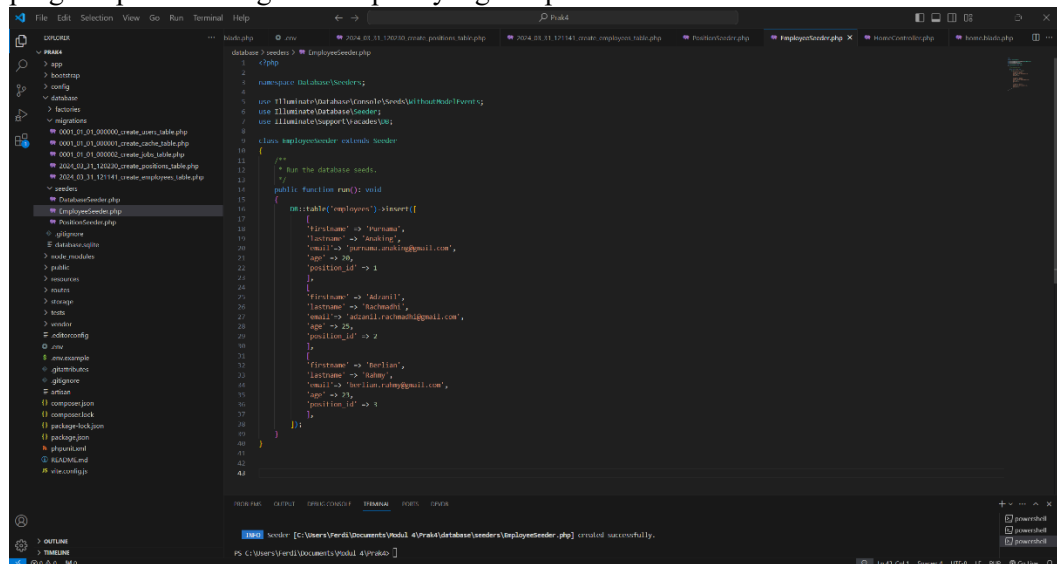
5. Membuat Data Dummy Menggunakan Seeder

- Generate file seeder untuk tabel Positions menggunakan artisan, kemudian buat kode program pada file migration seperti yang ada pada modul.



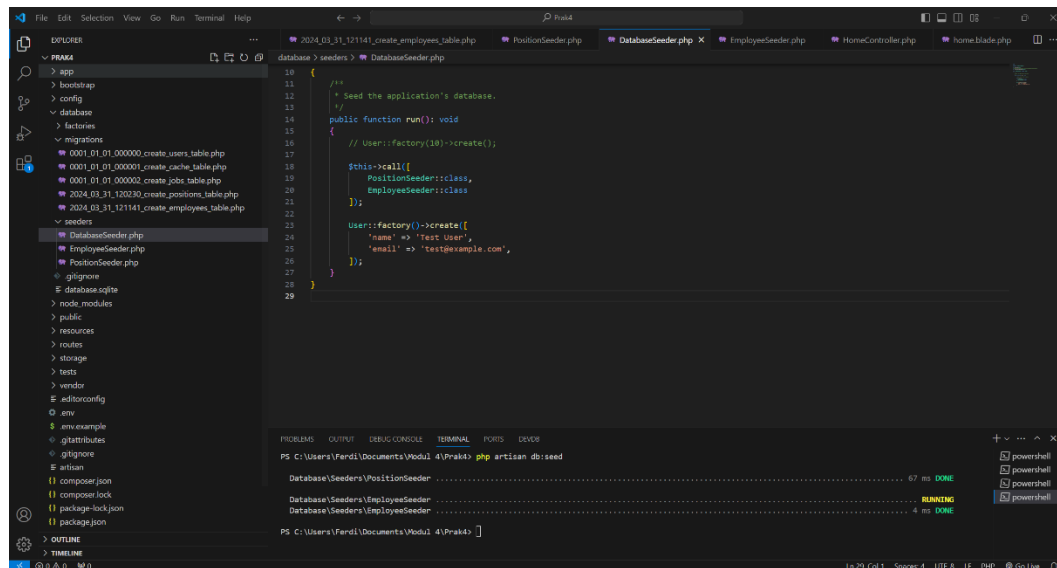
```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console/SeedsWithoutEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Str;
8
9 class PositionSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         DB::table('positions')->insert([
17             [
18                 'code' => 'FE',
19                 'name' => 'Front End Developer',
20                 'description' => 'Front End Developer'
21             ],
22             [
23                 'code' => 'BE',
24                 'name' => 'Back End Developer',
25                 'description' => 'Back End Developer'
26             ],
27             [
28                 'code' => 'SA',
29                 'name' => 'System Analyst',
30                 'description' => 'System Analyst'
31             ],
32         ]);
33     }
34 }
```

- Generate file seeder untuk tabel Employees menggunakan artisan, kemudian buat kode program pada file migration seperti yang ada pada modul.



```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console/SeedsWithoutEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Str;
8
9 class EmployeeSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         DB::table('employees')->insert([
17             [
18                 'first_name' => 'Purnama',
19                 'last_name' => 'Anaking',
20                 'email' => 'purnama.anaking@gmail.com',
21                 'age' => 20,
22                 'position_id' => 1
23             ],
24             [
25                 'first_name' => 'Adinil',
26                 'last_name' => 'Rachmadhigmail',
27                 'email' => 'adinil.rachmadhigmail.com',
28                 'age' => 25,
29                 'position_id' => 2
30             ],
31             [
32                 'first_name' => 'Berlim',
33                 'last_name' => 'Namy',
34                 'email' => 'berlim.namy@gmail.com',
35                 'age' => 21,
36                 'position_id' => 3
37             ],
38             [
39                 'first_name' => 'Purnama',
40                 'last_name' => 'Anaking',
41                 'email' => 'purnama.anaking@gmail.com',
42                 'age' => 20,
43                 'position_id' => 1
44             ],
45             [
46                 'first_name' => 'Adinil',
47                 'last_name' => 'Rachmadhigmail',
48                 'email' => 'adinil.rachmadhigmail.com',
49                 'age' => 25,
50                 'position_id' => 2
51             ],
52             [
53                 'first_name' => 'Berlim',
54                 'last_name' => 'Namy',
55                 'email' => 'berlim.namy@gmail.com',
56                 'age' => 21,
57                 'position_id' => 3
58             ],
59         ]);
60     }
61 }
```

- Lalu definisikan file seeder yang akan dieksekusi pada fuction “run()” pada file “DatabaseSedeer.php” seperti yang ada pada modul, kemudian eksekusi file seeder menggunakan artisan.



6. Menampilkan List Data dari Database

- Buat Raw SQL Query di method “index()” pada file EmployeeController dan passing data melalui Employee dari Controller ke View, kemudian Facade DB di bagian file Controller.

```

1  <?php
2
3  namespace App\Http\Controllers;
4  use Illuminate\Support\Facades\Validator;
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\DB;
7
8  class EmployeeController extends Controller
9  {
10
11      /**
12       * Display a listing of the resource.
13       */
14      public function index()
15      {
16          $pageTitle = 'Employee List';
17
18          // RAW SQL QUERY
19          $employees = DB::select('
20              select *, employees.id as employee_id, positions.name as
21              position_name
22              from employees
23              left join positions on employees.position_id = positions.id
24              ');
25
26          return view('employee.index', [
27              'pageTitle' => $pageTitle,
28              'employees' => $employees
29          ]);
30      }

```

- Tampilkan data Employee pada file View di “index.blade.php”, dan sesuaikan kode program seperti yang ada pada modul.

```

63 <tbody>
64     @foreach ($employees as $employee)
65     <tr>
66         <td>{{ $employee->firstname }}</td>
67         <td>{{ $employee->lastname }}</td>
68         <td>{{ $employee->email }}</td>
69         <td>{{ $employee->age }}</td>
70         <td>{{ $employee->position_name }}</td>
71         <td>
72             <div class="d-flex">
73                 <a href="{{ route('employees.show', ['employee' => $employee->employee_id]) }}"
74                     class="btn btn-outline-dark btn-sm
75                 me-2"><i>
76                     class="bi-person-lines-fill"></i></a>
77                 <a href="{{ route('employees.edit', ['employee' => $employee->employee_id]) }}"
78                     class="btn btn-outline-dark btn-sm
79                 me-2"><i>
80                     class="bi-pencil-square"></i></a>
81                 <div>
82                     <form
83                         action="{{ route('employees.destroy', ['employee' => $employee->employee_id]) }}"
84                         method="POST">
85                         @csrf
86                         @method('delete')
87                         <button type="submit"
88                             class="btn
89                         btn-outline-dark btn-sm me-2"><i>
90                             class="bi-trash"></i></button>
91                     </form>
92                 </div>
93             </div>
94         </td>
95     </tr>
96     @endforeach
97 </tbody>

```

7. Input Data ke Database

- Buat Raw SQL Query di method “create()” pada file EmployeeController untuk pilihan “Position” pada Form Create Employee dan passing data melalui Employee dari Controller ke View.

```

34 public function create()
35 {
36     $pageTitle = 'Create Employee';
37     // RAW SQL Query
38     $positions = DB::select('select * from positions');
39
40     return view('employee.create', compact('pageTitle', 'positions'));
41 }

```

- Tampilkan data Position pada file View di “create.blade.php”, dan sesuaikan kode program seperti yang ada pada modul.

```

108 <div class="col-md-12 mb-3">
109     <label for="position" class="form-label">Position</label>
110     <select name="position" id="position" class="form-select">
111         @foreach ($positions as $position)
112             <option value="{{ $position->id }}"
113                 {{ old('position') == $position->id ? 'selected' : '' }}>
114                 {{ $position->code .
115                     ' - ' .
116                     $position->name }}</option>
117         @endforeach
118     </select>
119     @error('position')
120     <div class="text-danger"><small>{{ $message }}</small></div>
121     @enderror
122

```

- Hasil akhir dari Form Create Employee akan seperti gambar dibawah ini.

The screenshot shows a web browser at the URL `localhost:8000/employees/create`. The application has a blue header with the text "Data Master" and navigation links for "Home" and "Employee List". A "My Profile" button is also present. The main content area is titled "Create Employee" and contains a form with the following fields:

- First Name:** Input field with placeholder "Enter First Name".
- Last Name:** Input field with placeholder "Enter Last Name".
- Email:** Input field with placeholder "Enter Email".
- Age:** Input field with placeholder "Enter Age".
- Position:** A dropdown menu currently showing "FE - Front End Developer".

At the bottom of the form are two buttons: a "Cancel" button with a left arrow icon and a "Save" button with a checkmark icon.

- Buat Raw SQL Query di method “store()” pada file EmployeeController dan lakukan Redirect route ke halaman Employee List.

```

46 public function store(Request $request)
47 {
48     $messages = [
49         'required' => ':Attribute harus diisi.',
50         'email' => 'Isi :attribute dengan format yang benar',
51         'numeric' => 'Isi :attribute dengan angka'
52     ];
53
54     $validator = Validator::make($request->all(), [
55         'firstName' => 'required',
56         'lastName' => 'required',
57         'email' => 'required|email',
58         'age' => 'required|numeric',
59     ], $messages);
60
61     if ($validator->fails()) {
62         return redirect()->back()->withErrors($validator)->withInput();
63     }
64
65     // INSERT QUERY
66     DB::table('employees')->insert([
67         'firstname' => $request->firstName,
68         'lastname' => $request->lastName,
69         'email' => $request->email,
70         'age' => $request->age,
71         'position_id' => $request->position,
72     ]);
73
74     return redirect()->route('employees.index');
75 }

```


8. Menampilkan Detail Data dari Database

- Buat Raw SQL Query di method “show()” pada file EmployeeController dan passing data Employee dari Controller ke View.

```
80 public function show(string $id)
81 {
82     $pageTitle = 'Employee Detail';
83
84     // RAW SQL QUERY
85     $employee = collect(DB::select('
86     select *, employees.id as employee_id, positions.name as
87     position_name
88     from employees
89     left join positions on employees.position_id = positions.id
90     where employees.id = ?
91     ', [$id]))->first();
92
93     return view('employee.show', compact('pageTitle', 'employee'));
94 }
```

- Buat file baru pada folder Employee di dalam folder Views dengan nama “show.blade.php”, tampilkan data Employee pada file View seperti pada modul.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>{{ $pageTitle }}</title>
    @vite('resources/sass/app.scss')
</head>
<body>
    <nav class="navbar navbar-expand-md navbar-dark bg-primary">
        <div class="container">
            <a href="{{ route('home') }}" class="navbar-brand mb-0 h1"><i
class="bi-hexagon-fill me-2"></i> Data
Master</a>
            <button type="button" class="navbar-toggler" data-bs-
toggle="collapse"
                data-bs-target="#navbarSupportedContent">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse"
id="navbarSupportedContent">
                <hr class="d-lg-none text-white-50">
                <ul class="navbar-nav flex-row flex-wrap">
                    <li class="nav-item col-2 col-md-auto"><a href="{{
route('home') }}" class="nav-link">Home</a></li>
                    <li class="nav-item col-2 col-md-auto"><a href="{{
route('employees.index') }}"
```

```

                class="nav-link active">Employee
                List</a></li>
            </ul>
            <hr class="d-lg-none text-white-50">
            <a href="{{ route('profile') }}" class="btn
btn-outline-light my-2 ms-md-auto"><i
                class="bi-person-circle me-1"></i>
                My Profile</a>
        </div>
    </div>
</nav>
<div class="container-sm my-5">
    <div class="row justify-content-center">
        <div class="p-5 bg-light rounded-3 col-xl-4 border">
            <div class="mb-3 text-center">
                <i class="bi-person-circle fs-1"></i>
                <h4>Detail Employee</h4>
            </div>
            <hr>
            <div class="row">
                <div class="col-md-12 mb-3">
                    <label for="firstName" class="form-label">First
                        Name</label>
                    <h5>{{ $employee->firstname }}</h5>
                </div>
                <div class="col-md-12 mb-3">
                    <label for="lastName" class="form-label">Last
                        Name</label>
                    <h5>{{ $employee->lastname }}</h5>
                </div>
                <div class="col-md-12 mb-3">
                    <label for="email" class="form-
label">Email</label>
                    <h5>{{ $employee->email }}</h5>
                </div>
                <div class="col-md-12 mb-3">
                    <label for="age" class="form-label">Age</label>
                    <h5>{{ $employee->age }}</h5>
                </div>
                <div class="col-md-12 mb-3">
                    <label for="age" class="form-
label">Position</label>
                    <h5>{{ $employee->position_name }}</h5>
                </div>
            </div>
            <hr>
            <div class="row">
                <div class="col-md-12 d-grid">

```

```

        <a href="{ route('employees.index') }"
class="btn btn-outline-dark btn-lg mt-3"><i
        class="bi-arrow-left-circle
me-2"></i> Back</a>
    </div>
</div>
</div>
</div>
</div>
    @vite('resources/js/app.js')
</body>
</html>

```

9. Menghapus Data dari Database

Buat Builder Query pada method “destroy()” dalam file EmployeeController kemudian Redirect ke halaman Employee List.

```

115     public function destroy(string $id)
116     {
117         // QUERY BUILDER
118         DB::table('employees')
119             ->where('id', $id)
120             ->delete();
121
122         return redirect()->route('employees.index');
123     }
124 }

```

10. Tugas

- Buat Fitur Edit untuk melengkapi hasil akhir dari praktikum ini.
- Ubah semua query yang ditulis dengan RAW SQL QUERY menjadi dengan pendekatan QUERY BUILDER.

Link Github:

<https://github.com/FerdiZ12/Modul-5.git>