

PRAKTIKUM PEMROGRAMAN FRAMEWORK

MODUL 10 Laravel Third Party Packages



Disusun Oleh:
Purnama Anaking, S.Kom., M.Kom.

**PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS
INSTITUT TEKNOLOGI TELKOM SURABAYA
2023**

DAFTAR ISI

DAFTAR ISI	2
1. Penerapan Laravel DataTables	3
2. Penerapan Laravel Sweet Alert	7
3. Penerapan Laravel Excel	10
4. Penerapan Laravel DomPdf	13
5. Memindahkan dan Memisahkan Kode Javascript	16
6. Tugas	19

MODUL 10

LARAVEL THIRD PARTY PACKAGES

Pada praktikum kali ini kita akan melanjutkan belajar tentang bagaimana menerapkan package-package tambahan pada project laravel kita. Package tambahan yang diberikan pada praktikum ini adalah Laravel DataTables, Laravel Sweet Alert, Laravel Exce dan Laravel DomPdf.

Seluruh package akan diterapkan pada project laravel yang telah dibuat pada praktikum modul 9.

1. Penerapan Laravel DataTables

- Edit file **EmployeeSeeder** untuk dapat menambahkan data dummy seperti di bawah ini. Data dummy ditambahkan menjadi sebanyak 200.

```
<?php

namespace Database\Seeders;

use App\Models\Employee;
use Illuminate\Database\Seeder;

class EmployeeSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        Employee::factory()->count(200)->create();
    }
}
```

- Jalankan file seeder tersebut via Artisan seperti di bawah ini.

```
php artisan db:seed --class=EmployeeSeeder
```

1.1 Basic DataTables

- Install packages **Laravel DataTables** via **composer**

```
composer require yajra/laravel-datatables-oracle
```

- Install aset-aset **DataTables** yang dibutuhkan via **NPM**

```
npm install datatables.net-bs5 datatables.net-buttons-bs5
```

- Registrasikan aset-aset **DataTables** pada file **/resources/js/app.js** seperti yang terlihat di bawah ini.

```
import './bootstrap';
import 'datatables.net-bs5';
import 'datatables.net-buttons-bs5';
import.meta.glob(["../images/**"]);
```

- Registrasikan aset-aset **DataTables** pada file **/resources/sass/app.scss** seperti yang terlihat di bawah ini.

```
// Bootstrap
@import 'bootstrap/scss/bootstrap';
@import "bootstrap-icons/font/bootstrap-icons.css";

// DataTables
@import "datatables.net-bs5/css/dataTables.bootstrap5.css";
@import "datatables.net-buttons-bs5/css/buttons.bootstrap5.css";
```

- Import **jQuery** pada file **/resources/js/bootstrap.js** seperti yang terlihat di bawah ini.

```
import 'bootstrap';

import axios from 'axios';
window.axios = axios;
window.axios.defaults.headers.common['X-Requested-With'] = 'XMLHttpRequest';

import $ from 'jquery';
window.$ = $;
```

- Buka file base layout pada **/resources/views/layouts/app.blade.php** kemudian tambahkan blade directive **stack()** dengan nama **“scripts”** seperti di bawah ini.

```
<!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>
    @include('layouts.nav')
    @yield('content')
    @vite('resources/js/app.js')
    @stack('scripts')
</body>
</html>
```































- Buka file view pada **/resources/views/employee/index.blade.php** kemudian gunakan block blade directive **push()** untuk menambahkan kode program seperti di bawah ini.

```
@push('scripts')
<script type="module">
    $(document).ready(function() {
        $('#employeeTable').DataTable();
    });
</script>
@endpush
```

- Berikan atribut **id** dengan nilai “**employeeTable**” pada elemen tag **<table>** seperti kode program di bawah ini.

```
<table class="table table-bordered table-hover table-striped mb-0 bg-white"
id="employeeTable">
```

- Akses halaman **Employee List** via browser, jika berhasil akan muncul tampilan **DataTables** untuk data employee.

Show 10	entries	Search:			
First Name	Last Name	Email	Age	Position	
Unique	Ryan	johnston.kadin@feest.com	30	Purchasing Agent	  
Uriah	Feil	noemi.murazik@schneider.net	32	Portable Power Tool Repairer	  
Verda	Becker	kaylee.douglas@yahoo.com	35	Law Enforcement Teacher	  
Verlie	Bechtelar	mkunde@bednar.biz	49	Chiropractor	  
Vivienne	Lind	waelchi.urban@lind.org	45	Grinder OR Polisher	  
Wayne	Rutherford	kemmer.maximillian@hotmail.com	45	Segmental Paver	  
Wendell	Hermiston	xkling@witting.com	44	Artillery Crew Member	  
Willard	Pfeffer	vleuschke@thiel.com	46	Marking Machine Operator	  
Willis	Jast	brooklyn.wiegand@fay.com	39	Poet OR Lyricist	  
Zetta	Bradtke	katharina99@hotmail.com	27	Cabinetmaker	  
Showing 191 to 200 of 200 entries					Previous 1 ... 16 17 18 19 20 Next

1.2 Server-side Processing DataTables

- Buat Route baru pada **/routes/web.php** dengan URI **/getEmployees**, mengarahkan ke method **getData()** pada **EmployeeController** seperti kode program di bawah ini.

```
Route::get('getEmployees', [EmployeeController::class,
'getData'])->name('employees.getData');
```

- Buat method **getData()** pada **EmployeeController** seperti di bawah ini.

```
public function getData(Request $request)
{
    $employees = Employee::with('position');

    if ($request->ajax()) {
        return datatables()->of($employees)
            ->addIndexColumn()
            ->addColumn('actions', function($employee) {
```

```

        return view('employee.actions', compact('employee'));
    })
    ->toJson();
}
}

```

- Sesuaikan method **index()** pada **EmployeeController** seperti kode program di bawah ini. Query eloquent untuk mengambil data employee sudah diperlukan lagi karna sudah digantikan proses nya pada method **getData()**.

```

public function index()
{
    $pageTitle = 'Employee List';

    return view('employee.index', compact('pageTitle'));
}

```

- Buka file view pada **/resources/views/employee/index.blade.php** kemudian sesuaikan kode program pada bagian tag **<table>** nya seperti di bawah ini. Perulangan yang ada pada tag **<tbody>** dihapus karna data akan di load secara asynchronous dan ditangani oleh DataTable.

```

<table class="table table-bordered table-hover table-striped mb-0 bg-white"
id="employeeTable">
    <thead>
        <tr>
            <th>ID</th>
            <th>No.</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
            <th>Age</th>
            <th>Position</th>
            <th></th>
        </tr>
    </thead>
</table>

```

- Buka file view pada **/resources/views/employee/index.blade.php** kemudian sesuaikan script **DataTable** nya seperti di bawah ini.

```

@push('scripts')
    <script type="module">
        $(document).ready(function() {
            $("#employeeTable").DataTable({
                serverSide: true,
                processing: true,
                ajax: "/getEmployees",
                columns: [
                    { data: "id", name: "id", visible: false },
                    { data: "DT_RowIndex", name: "DT_RowIndex", orderable: false,

```

```





















searchable: false },
    { data: "firstname", name: "firstname" },
    { data: "lastname", name: "lastname" },
    { data: "email", name: "email" },
    { data: "age", name: "age" },
    { data: "position.name", name: "position.name" },
    { data: "actions", name: "actions", orderable: false,
searchable: false },
    ],
    order: [[0, "desc"]],
    lengthMenu: [
        [10, 25, 50, 100, -1],
        [10, 25, 50, 100, "All"],
    ],
    });
});
</script>
@endpush

```

- Akses halaman **Employee List** via browser, jika berhasil akan muncul tampilan **DataTables** untuk data employee dengan teknik server-side processing.

Showing 10 entries

Search:

First Name	Last Name	Email	Age	Position	
Unique	Ryan	johnston.kadin@feest.com	30	Purchasing Agent	 
Uriah	Feil	noemi.murazik@schneider.net	32	Portable Power Tool Repairer	 
Verda	Becker	kaylee.douglas@yahoo.com	35	Law Enforcement Teacher	 
Verlie	Bechtelar	mkunde@bednar.biz	49	Chiropractor	 
Vivienne	Lind	waelchi.urban@lind.org	45	Grinder OR Polisher	 
Wayne	Rutherford	kemmer.maximillian@hotmail.com	45	Segmental Paver	 
Wendell	Hermiston	xkling@witting.com	44	Artillery Crew Member	 
Willard	Pfeffer	vleuschke@thiel.com	46	Marking Machine Operator	 
Willis	Jast	brooklyn.wiegand@fay.com	39	Poet OR Lyricist	 
Zetta	Bradtko	katharina99@hotmail.com	27	Cabinetmaker	 

Showing 191 to 200 of 200 entries

Previous 1 ... 16 17 18 19 20 Next

2. Penerapan Laravel Sweet Alert

- Install package **SweetAlert** via **composer**

```
composer require realrashid/sweet-alert
```

- Buka file base layout pada **/resources/views/layouts/app.blade.php** kemudian tambahkan blade directive **include()** untuk menerapkan package SweetAlert seperti di bawah ini. Letakkan seteah **vite()** dan sebelum **stack()**.

```
<!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>
  @include('layouts.nav')
  @yield('content')
  @vite('resources/js/app.js')
  @include('sweetalert::alert')
  @stack('scripts')
</body>
</html>

```

- Publish aset dari SweetAlert via Artisan.

```
php artisan sweetalert:publish
```

- Buka **EmployeeController** dan letakkan SweetAlert pada method **store()**, **update()** dan **destroy()**.

```

use RealRashid\SweetAlert\Facades\Alert;

public function store(Request $request)
{
    ...

    Alert::success('Added Successfully', 'Employee Data Added Successfully.');
```

```

use RealRashid\SweetAlert\Facades\Alert;

public function update(Request $request, string $id)
{
    ...

    Alert::success('Changed Successfully', 'Employee Data Changed Successfully.');
```

```

use RealRashid\SweetAlert\Facades\Alert;

public function destroy(string $id)
{
    ...

```



```
Alert::success('Deleted Successfully', 'Employee Data Deleted Successfully.');
```

```
return redirect()->route('employees.index');
```

```
}
```

- Buka file view pada **/resources/views/employee/actions.blade.php** kemudian sesuaikan kode program pada bagian tag **<button>**. Tambahkan class **“btn-delete”** dan atribut **“data-name”** seperti di bawah ini.

```
<form action="{{ route('employees.destroy', ['employee' => $employee->id]) }}"
method="POST">
    @csrf
    @method('delete')
    <button type="submit" class="btn btn-outline-dark btn-sm me-2 btn-delete"
data-name="{{ $employee->firstname.' '.$employee->lastname }}">
        <i class="bi-trash"></i>
    </button>
</form>
```

- Buka file view pada **/resources/views/employee/index.blade.php** kemudian sesuaikan kode program pada bagian tag **<table>**. Tambahkan class **“datatable”** seperti di bawah ini.

```
<table class="table table-bordered table-hover table-striped mb-0 bg-white
datatable" id="employeeTable">
```

- Buka file view pada **/resources/views/employee/index.blade.php** kemudian tambahkan kode program di bawah ini pada block blade directive **push()** seperti di bawah ini.

```
@push('scripts')
    <script type="module">
        $(document).ready(function() {

            ...

            $(".datatable").on("click", ".btn-delete", function (e) {
                e.preventDefault();

                var form = $(this).closest("form");
                var name = $(this).data("name");

                Swal.fire({
                    title: "Are you sure want to delete\n" + name + "?",
                    text: "You won't be able to revert this!",
                    icon: "warning",
                    showCancelButton: true,
                    confirmButtonClass: "bg-primary",
                    confirmButtonText: "Yes, delete it!",
                }).then((result) => {
                    if (result.isConfirmed) {
```

```

        form.submit();
    }
    });
});
</script>
@endpush

```

- Sesuaikan method **index()** pada **EmployeeController** seperti kode program di bawah ini. Panggil method **confirmDelete()** milik SweetAlert.

```

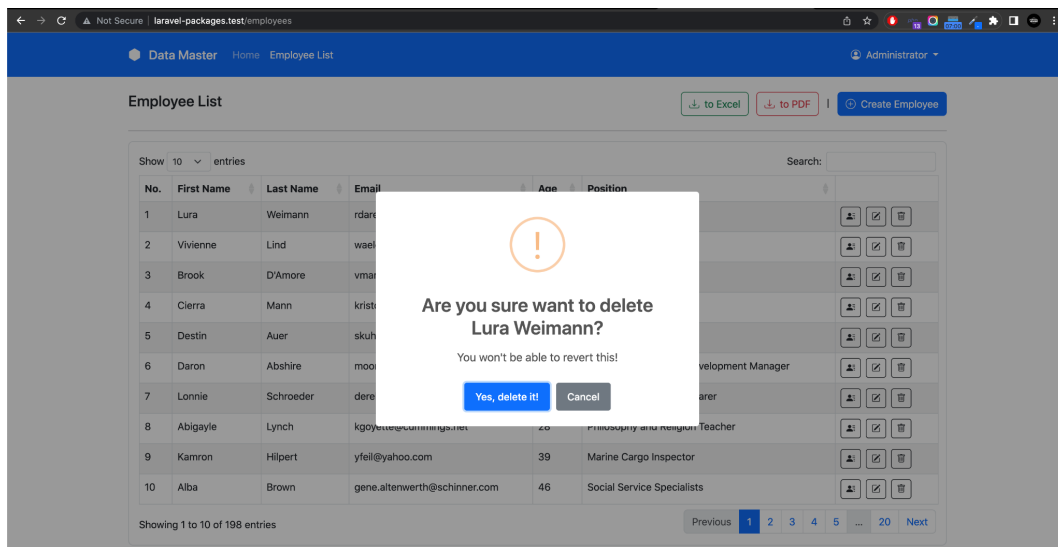
public function index()
{
    $pageTitle = 'Employee List';

    confirmDelete();

    return view('employee.index', compact('pageTitle'));
}

```

- Akses Fitur **Create**, **Edit** dan **Delete Employee** via browser, jika berhasil akan muncul tampilan popup window dari **SweetAlert** sebagai feedback sistem untuk proses yang sedang dilakukan oleh user. Contoh tampilannya seperti gambar di bawah ini.



3. Penerapan Laravel Excel

- Install **Laravel Excel** via **Composer**.

```
composer require maatwebsite/excel
```

- Publish file konfigurasi dengan menjalankan command script seperti di bawah ini.

```
php artisan vendor:publish --provider="Maatwebsite\Excel\ExcelServiceProvider" --tag=config
```

- Buka file view pada **/resources/views/employee/index.blade.php** kemudian buat **button** untuk fitur export data employee menjadi Excel. Sesuaikan bagian kode program seperti di bawah ini.

```
<div class="row mb-0">
  <div class="col-lg-9 col-xl-6">
    <h4 class="mb-3">{{ $pageTitle }}</h4>
  </div>
  <div class="col-lg-3 col-xl-6">
    <ul class="list-inline mb-0 float-end">
      <li class="list-inline-item">
        <a href="{{ route('employees.exportExcel') }}" class="btn btn-outline-success">
          <i class="bi bi-download me-1"></i> to Excel
        </a>
      </li>
      <li class="list-inline-item">|</li>
      <li class="list-inline-item">
        <a href="{{ route('employees.create') }}" class="btn btn-primary">
          <i class="bi bi-plus-circle me-1"></i> Create Employee
        </a>
      </li>
    </ul>
  </div>
</div>
```

- Buat Route baru pada **/routes/web.php** dengan URI **/exportExcel**, mengarahkan ke method **exportExcel()** pada **EmployeeController** seperti kode program di bawah ini.

```
Route::get('exportExcel', [EmployeeController::class,
'exportExcel'])->name('employees.exportExcel');
```

- Buat file **Export** dengan nama **EmployeesExport** via Artisan seperti di bawah ini. Diarahkan kepada model **Employee**.

```
php artisan make:export EmployeesExport --model=Employee
```

- Edit atau ubah file **EmployeesExport**, sesuaikan seperti di bawah ini.

```
<?php

namespace App\Exports;

use App\Models\Employee;
use Illuminate\Contracts\View\View;
use Maatwebsite\Excel\Concerns\FromView;
use Maatwebsite\Excel\Concerns\ShouldAutoSize;
use Maatwebsite\Excel\Concerns\WithStyles;
use PhpOffice\PhpSpreadsheet\Worksheet\Worksheet;

class EmployeesExport implements FromView, WithStyles, ShouldAutoSize
```

```

{
    public function styles(Worksheet $sheet)
    {
        return [
            1 => ['font' => ['bold' => true]],
        ];
    }

    public function view(): View
    {
        return view('employee.export_excel', [
            'employees' => Employee::all()
        ]);
    }
}

```

- Buat file view baru dengan nama **export_excel.blade.php** pada direktori **/resources/views/employee**. Isi dengan kode program seperti di bawah ini.

```

<table>
    <thead>
        <tr>
            <th>No.</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
            <th>Age</th>
            <th>Position</th>
        </tr>
    </thead>
    <tbody>
        @foreach ($employees as $index => $employee)
            <tr>
                <td>{{ $index + 1 }}</td>
                <td>{{ $employee->firstname }}</td>
                <td>{{ $employee->lastname }}</td>
                <td>{{ $employee->email }}</td>
                <td>{{ $employee->age }}</td>
                <td>{{ $employee->position->name }}</td>
            </tr>
        @endforeach
    </tbody>
</table>

```

- Buat method **exportExcel()** pada **EmployeeController** seperti kode program di bawah ini.

```

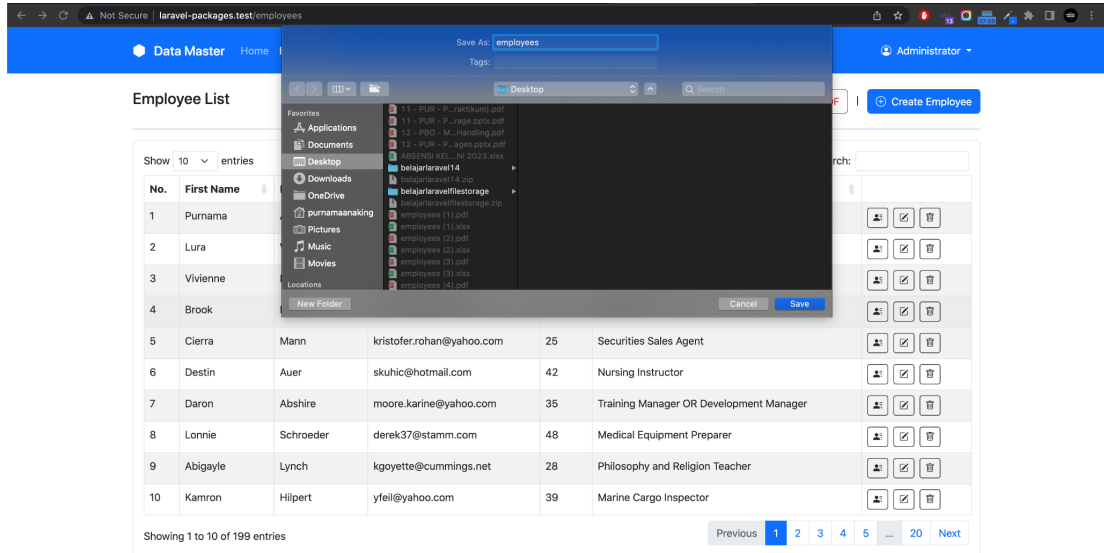
use Maatwebsite\Excel\Facades\Excel;
use App\Exports\EmployeesExport;

public function exportExcel()
{
    return Excel::download(new EmployeesExport, 'employees.xlsx');
}

```

```
}
```

- Akses halaman **Employee List** via browser, kemudian klik button “**to Excel**” untuk meng-export data employee ke dalam bentuk file Excel.



No.	First Name	Last Name	Email	Age	Position
1	Estell	Moer	rutherford.lonie@hotmail.com	40	Kindergarten Teacher
2	Braulio	Kautzer	dhessel@keebler.com	34	Automotive Mechanic
3	Maximus	Bogisich	kihn.magdalen@murray.com	30	Extraction Worker
4	Monique	Stanton	hickle.wanda@yahoo.com	33	Materials Scientist
5	Emerald	Ondricka	rweimann@gmail.com	31	Gaming Surveillance Officer
6	Alfredo	Carter	leora.buckridge@yahoo.com	34	Pharmacy Technician
7	Jody	West	stroman.lonnie@hotmail.com	27	Plating Operator OR Coating Machine Operator
8	Gussie	Block	graham.gaston@kuhlman.org	37	Police Identification OR Records Officer
9	Chaim	Mohr	trace20@hotmail.com	29	Psychologist
10	Rosetta	Franecki	mariaela.cartwright@gmail.com	42	Waiter
11	Gerhard	Bailey	claud.crist@yahoo.com	39	Courier
12	Dovie	Lemke	zmayer@zemlak.com	43	Pipelayer
13	Dock	Stehr	virgie.hickle@zemlak.com	45	Dental Laboratory Technician
14	Serena	Tremblay	leann72@yahoo.com	38	Copy Machine Operator
15	Gwendolyn	Ondricka	vwil@gmail.com	31	ccc
16	Marilyne	Zulauf	kaliy53@hotmail.com	30	Forest Fire Inspector
17	Wendell	Hermiston	xking@witting.com	44	Artillery Crew Member
18	Adella	Breitenberg	luella13@hotmail.com	35	City Planning Aide
19	Kraig	Strosin	araceli99@gmail.com	35	Motorcycle Mechanic
20	Colin	Reichel	santos.feest@yahoo.com	47	Septic Tank Servicer
21	Raphaelle	Beatty	estell41@hirth.net	27	Choreographer
22	Sylvia	Waters	annalise51@yahoo.com	27	Forest Fire Inspector
23	Mortimer	Barton	laron34@yahoo.com	43	Microbiologist
24	Jayme	Adams	greenfelder.laura@gmail.com	46	Aircraft Assembler
25	Sabina	Ondricka	jarod29@yahoo.com	44	Ambulance Driver
26	Meredith	Kuhlman	haley.kayden@batz.com	33	Philosophy and Religion Teacher
27	Meagan	Johnson	miller.jerrold@pacocho.org	39	Computer Programmer
28	Derrick	Kunde	dawson81@becker.com	28	Plant and System Operator
29	Ruby	Grady	albertha05@gmail.com	28	Substance Abuse Social Worker
30	Jessyca	Weimann	nwalker@hotmail.com	42	Motorboat Operator
31	Chloe	Bartell	alfreda44@kusikowski.com	43	Power Generating Plant Operator
32	Leonardo	O'Kon	giovani.larkin@marks.com	39	Elevator Installer and Repairer
33	Abdul	Green	tborer@nitzsche.biz	39	Sculptor
34	Janet	Marks	west.bert@hettinger.org	30	Admin
35	Marianne	Mills	hleuschke@thompson.com	34	Dishwasher
36	Kelly	Kirlin	abigale.tillman@yahoo.com	37	Set and Exhibit Designer
37	Marge	Stokes	justus.koepp@hotmail.com	37	Personal Care Worker
38	Zetta	Bradtko	katharina99@hotmail.com	27	Cabinetmaker
39	Fabiola	Shields	lexie43@bechtelar.info	43	Refractory Materials Repairer
40	Christ	Schneider	xroob@yahoo.com	32	Oral Surgeon
41	Ewald	Wilkinson	missouri07@anderson.com	27	Compliance Officers

4. Penerapan Laravel DomPdf

- Install **Laravel DomPdf** via **Composer**.

```
composer require barryvdh/laravel-dompdf
```

- Publish file konfigurasi dengan menjalankan command script seperti di bawah ini.

```
php artisan vendor:publish --provider="Barryvdh\DomPDF\ServiceProvider"
```

- Buka file view pada **/resources/views/employee/index.blade.php** kemudian buat **button** untuk fitur export data employee menjadi PDF. Sesuaikan bagian kode program seperti di bawah ini.

```
<div class="row mb-0">
  <div class="col-lg-9 col-xl-6">
    <h4 class="mb-3">{{ $pageTitle }}</h4>
  </div>
  <div class="col-lg-3 col-xl-6">
    <ul class="list-inline mb-0 float-end">
      <li class="list-inline-item">
        <a href="{{ route('employees.exportExcel') }}" class="btn btn-outline-success">
          <i class="bi bi-download me-1"></i> to Excel
        </a>
      </li>
      <li class="list-inline-item">
        <a href="{{ route('employees.exportPdf') }}" class="btn btn-outline-danger">
          <i class="bi bi-download me-1"></i> to PDF
        </a>
      </li>
      <li class="list-inline-item"></li>
      <li class="list-inline-item">
        <a href="{{ route('employees.create') }}" class="btn btn-primary">
          <i class="bi bi-plus-circle me-1"></i> Create Employee
        </a>
      </li>
    </ul>
  </div>
</div>
```

- Buat Route baru pada **/routes/web.php** dengan URI **/exportPdf**, mengarahkan ke method **exportPdf()** pada **EmployeeController** seperti kode program di bawah ini.

```
Route::get('exportPdf', [EmployeeController::class,
'exportPdf'])->name('employees.exportPdf');
```

- Buat file view baru dengan nama **export_pdf.blade.php** pada direktori **/resources/views/employee**. Isi dengan kode program seperti di bawah ini.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Employee List</title>
  <style>
    html {
```

```

        font-size: 12px;
    }

    .table {
        border-collapse: collapse !important;
        width: 100%;
    }

    .table-bordered th,
    .table-bordered td {
        padding: 0.5rem;
        border: 1px solid black !important;
    }
</style>
</head>
<body>
    <h1>Employee List</h1>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>No.</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Email</th>
                <th>Age</th>
                <th>Position</th>
            </tr>
        </thead>
        <tbody>
            @foreach ($employees as $index => $employee)
                <tr>
                    <td align="center">{{ $index + 1 }}</td>
                    <td>{{ $employee->firstname }}</td>
                    <td>{{ $employee->lastname }}</td>
                    <td>{{ $employee->email }}</td>
                    <td align="center">{{ $employee->age }}</td>
                    <td>{{ $employee->position->name }}</td>
                </tr>
            @endforeach
        </tbody>
    </table>
</body>
</html>

```

- Buat method **exportPdf()** pada **EmployeeController** seperti kode program di bawah ini.

```

use PDF;
use App\Models\Employee;

public function exportPdf()
{
    $employees = Employee::all();
}

```

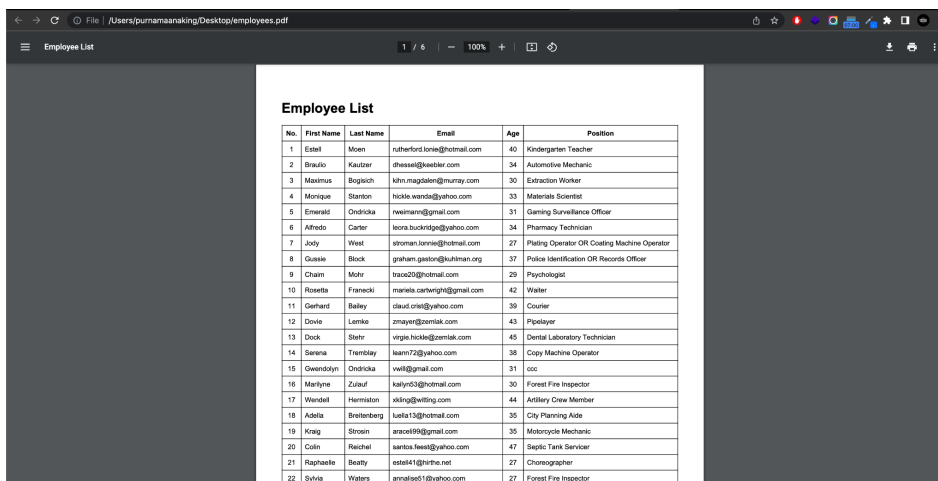
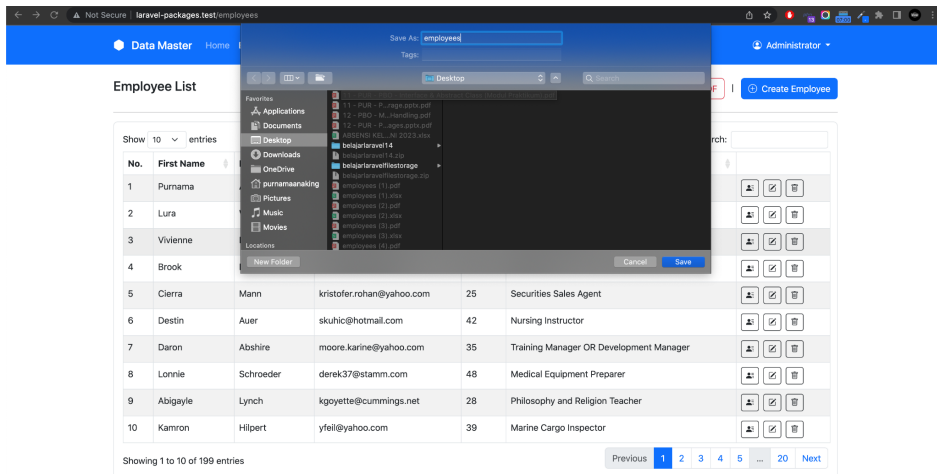
```

$pdf = PDF::loadView('employee.export_pdf', compact('employees'));

return $pdf->download('employees.pdf');
}

```

- Akses halaman **Employee List** via browser, kemudian klik button “to PDF” untuk meng-export data employee ke dalam bentuk file PDF.



5. Memindahkan dan Memisahkan Kode Javascript

- Kita dapat memindahkan kode javascript yang ada pada file view kita dan kita kelompokkan dengan file-file javascript pada project laravel kita.
- Buat file **employee.js** pada direktori **/resources/js/**. Hapus blok kode yang ada pada blade directive **push()** pada file **/resources/views/index.blade.php** dan pindahkan kode javascript yang ada ke file **employee.js**. Kode program pada file **index.blade.php** akan menjadi seperti di bawah ini.

```

@extends('layouts.app')

@section('content')
    <div class="container mt-4">

```



```

<div class="row mb-0">
  <div class="col-lg-9 col-xl-6">
    <h4 class="mb-3">{{ $pageTitle }}</h4>
  </div>
  <div class="col-lg-3 col-xl-6">
    <ul class="list-inline mb-0 float-end">
      <li class="list-inline-item">
        <a href="{{ route('employees.exportExcel') }}" class="btn
btn-outline-success">
          <i class="bi bi-download me-1"></i> to Excel
        </a>
      </li>
      <li class="list-inline-item">
        <a href="{{ route('employees.exportPdf') }}" class="btn
btn-outline-danger">
          <i class="bi bi-download me-1"></i> to PDF
        </a>
      </li>
      <li class="list-inline-item"></li>
      <li class="list-inline-item">
        <a href="{{ route('employees.create') }}" class="btn
btn-primary">
          <i class="bi bi-plus-circle me-1"></i> Create
Employee
        </a>
      </li>
    </ul>
  </div>
</div>
<hr>
<div class="table-responsive border p-3 rounded-3 mb-5">
  <table class="table table-bordered table-hover table-striped mb-0
bg-white datatable" id="employeeTable">
    <thead>
      <tr>
        <th>ID</th>
        <th>No.</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
        <th>Age</th>
        <th>Position</th>
        <th></th>
      </tr>
    </thead>
  </table>
</div>
</div>
@endsection

```

- Kode program pada file **employee.js** akan menjadi seperti di bawah ini.

```

$(function() {
  $("#employeeTable").DataTable({
    serverSide: true,

```

```

        processing: true,
        ajax: "/getEmployees",
        columns: [
            { data: "id", name: "id", visible: false },
            { data: "DT_RowIndex", name: "DT_RowIndex", orderable: false,
searchable: false },
            { data: "firstname", name: "firstname" },
            { data: "lastname", name: "lastname" },
            { data: "email", name: "email" },
            { data: "age", name: "age" },
            { data: "position.name", name: "position.name" },
            { data: "actions", name: "actions", orderable: false, searchable:
false },
        ],
        order: [[0, "desc"]],
        lengthMenu: [
            [10, 25, 50, 100, -1],
            [10, 25, 50, 100, "All"],
        ],
    });

$(".datatable").on("click", ".btn-delete", function (e) {
    e.preventDefault();

    var form = $(this).closest("form");
    var name = $(this).data("name");

    Swal.fire({
        title: "Are you sure want to delete\n" + name + "?",
        text: "You won't be able to revert this!",
        icon: "warning",
        showCancelButton: true,
        confirmButtonClass: "bg-primary",
        confirmButtonText: "Yes, delete it!",
    }).then((result) => {
        if (result.isConfirmed) {
            form.submit();
        }
    });
});
});
});

```

- Buka file `/resources/js/app.js` kemudian import file `employee.js`.

```

import './bootstrap';
import 'datatables.net-bs5';
import 'datatables.net-buttons-bs5';
import './employee';
import.meta.glob(["../images/**"]);

```

- Akses halaman **Employee List** via browser, jika berhasil seluruh fitur tetap berjalan dengan baik.

6. Tugas

1. **Praktekkan** dan **analisis** seluruh poin **praktikum dan latihan** yang ada di atas secara **INDIVIDU**. Infokan URL Repository Github pada laporan praktikum.
2. Dokumentasikan seluruhnya (**screenshot output**) dalam bentuk Laporan Praktikum.
3. Kumpulkan Laporan Praktikum (**.pdf**) yang telah dibuat di dalam praktikum via **E-Learning** paling lambat sebelum jadwal praktikum minggu depan.