**LAPORAN PRAKTIKUM**

**PEMROGRAMAN WEB LANJUT**

**JOBSHEET - 8 : File Import dan Export ke PDF dan Excel pada Laravel**



**Disusun Oleh :**

**Ghoffar Abdul Ja’far**

**2341720035/TI2F**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024/2025**

|  |  |  |
| --- | --- | --- |
| Mata Kuliah | **:** | Pemrograman Web Lanjut (PWL) |
| Program Studi | **:** | D4 – Teknik Informatika **/** D4 – Sistem Informasi Bisnis |
| Semester | **:** | 4 (empat) / 5 (lima) |
| Pertemuan ke- | **:** | 8 (delapan) |

**JOBSHEET 08**

**File Import dan Export ke PDF dan Excel pada Laravel**

Sesuai dengan **Studi Kasus PWL.pdf**.

Jadi project Laravel 10 kita masih sama dengan menggunakan repositori **PWL\_POS.**

*Project* **PWL\_POS** akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

1. **File Upload (import data) pada Laravel**

Mengunggah file merupakan salah satu fitur esensial yang sering kali dijumpai di berbagai proyek website. Fitur ini memberikan kemampuan bagi pengguna untuk memindahkan file dari perangkat komputer lokal mereka ke dalam lingkungan server aplikasi web. Proses ini melibatkan pengguna dalam memilih file dari sistem lokal mereka, seperti gambar, dokumen, atau media lainnya, yang kemudian diunggah melalui formulir yang disediakan di website. Setelah file tersebut berhasil diunggah, server akan menyimpan file tersebut di direktori tertentu yang telah ditentukan sebelumnya oleh pengembang. Penyimpanan ini dapat berada di server lokal, penyimpanan awan (*cloud storage*), atau bahkan dalam jaringan distribusi konten (CDN) untuk memastikan kinerja dan ketersediaan yang optimal. Dengan fitur unggah file ini, pengguna dapat memperkaya konten website secara dinamis, sementara pengembang dapat memastikan bahwa file yang diunggah dikelola dengan baik, aman, dan sesuai dengan kebijakan aplikasi.

Mengunggah file bukan hanya digunakan untuk menambahkan media atau dokumen ke sebuah website, tetapi juga memiliki peran penting dalam proses pengelolaan data, khususnya untuk *import* data ke database dalam aplikasi berbasis Laravel. Fitur ini memungkinkan

pengguna untuk mengunggah file berisi data dalam format seperti CSV atau Excel, yang kemudian diproses oleh aplikasi untuk dimasukkan ke dalam database.

Proses ini biasanya dimulai dengan pengguna memilih file yang berisi data dari komputer lokal mereka dan mengunggahnya ke server melalui formulir yang disediakan di aplikasi web. Setelah file berhasil diunggah, Laravel akan memproses file tersebut menggunakan paket seperti Maatwebsite Excel, yang dirancang khusus untuk menangani impor data dari file Excel atau CSV. Dalam jobsheet ini, kita akan mempelajari bagaimana cara mengunggah file yang digunakan untuk proses input data ke dalam database (import) menggunakan Laravel.

**Praktikum 1** – Implementasi Upload File untuk import data:

* 1. Kita buka database yang sudah kita buat sebelumnya, dan kita cek tabel m\_kategori dan

m\_barang



* 1. Disini kita akan mencoba untuk memasukkan data ke dalam sistem kita secara banyak. Kita bisa menggunakan file excel untuk mencoba memasukkan data barang ke dalam sistem kita. Kita buat template file excel yang akan kita gunakan untuk import data barang.



Contoh kita ingin mengimport 5 data barang sekaligus. Kita simpan ke file excel dengan nama template\_barang.xlsx, dan kita simpan pada folder **public** di project web kita.

* 1. Selanjutnya kita modifikasi view pada barang/index.blade.php untuk bisa menambahkan tombol menambah form untuk upload untuk import data barang

@extends('layouts.template')

@section('content')

<div class="card">

<div class="card-header">

<h3 class="card-title">Daftar barang</h3>

<div class="card-tools">

<button onclick="modalAction('{{ url('/barang/import') }}')" class="btn btn- info">Import Barang</button>

<a href="{{ url('/barang/create') }}" class="btn btn-primary">Tambah

Data</a>

<button onclick="modalAction('{{ url('/barang/create\_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>

</div>

</div>

<div class="card-body">

<!-- untuk Filter data -->

<div id="filter" class="form-horizontal filter-date p-2 border-bottom mb-2">

<div class="row">

<div class="col-md-12">

<div class="form-group form-group-sm row text-sm mb-0">

<label for="filter\_date" class="col-md-1 col-form-

label">Filter</label>

<div class="col-md-3">

<select name="filter\_kategori" class="form-control form-

control-sm filter\_kategori">

<option value="">- Semua -</option> @foreach($kategori as $l)

<option value="{{ $l->kategori\_id }}">{{ $l-

>kategori\_nama }}</option>

@endforeach

</select>

<small class="form-text text-muted">Kategori Barang</small>

</div>

</div>

</div>

</div>

</div> @if(session('success'))

<div class="alert alert-success">{{ session('success') }}</div> @endif

@if(session('error'))

<div class="alert alert-danger">{{ session('error') }}</div> @endif

<table class="table table-bordered table-sm table-striped table-hover" id="table-barang">

<thead>

<tr><th>No</th><th>Kode Barang</th><th>Kode Barang</th><th>Harga Beli</th><th>Harga Jual</th><th>Kategori</th><th>Aksi</th></tr>

</thead>

<tbody></tbody>

</table>

</div>

</div>

<div id="myModal" class="modal fade animate shake" tabindex="-1" data-backdrop="static" data-keyboard="false" data-width="75%"></div>

@endsection @push('js')

<script>

function modalAction(url = ''){

$('#myModal').load(url,function(){

$('#myModal').modal('show');

});

}

var tableBarang;

$(document).ready(function(){

tableBarang = $('#table-barang').DataTable({ processing: true,

serverSide: true, ajax: {

"url": "{{ url('barang/list') }}", "dataType": "json",

"type": "POST",

"data": function (d) {

d.filter\_kategori = $('.filter\_kategori').val();

}

},

columns: [{

data: "No\_Urut", className: "text-center", width: "5%",

orderable: false, searchable: false

},{

data: "barang\_kode", className: "",

width: "10%", orderable: true, searchable: true

},{

data: "barang\_nama", className: "",

width: "37%", orderable: true, searchable: true,

},{

data: "harga\_beli", className: "",

width: "10%", orderable: true, searchable: false,

render: function(data, type, row){

return new Intl.NumberFormat('id-ID').format(data);

}

},{

data: "harga\_jual", className: "",

width: "10%",

orderable: true, searchable: false,

render: function(data, type, row){

return new Intl.NumberFormat('id-ID').format(data);

}

},{

data: "kategori.kategori\_nama", className: "",

width: "14%", orderable: true, searchable: false

},{

data: "aksi",

className: "text-center", width: "14%",

orderable: false, searchable: false

}

]

});

$('#table-barang\_filter input').unbind().bind().on('keyup', function(e){ if(e.keyCode == 13){ // enter key

tableBarang.search(this.value).draw();

}

});

$('.filter\_kategori').change(function(){ tableBarang.draw();

});

});

</script> @endpush

* 1. Selanjutnya kita buat view untuk form upload/import file excel dan download file

template\_barang.xlsx. Kita beri nama file dengan nama barang/import.blade.php

<form action="{{ url('/barang/import\_ajax') }}" method="POST" id="form-import" enctype="multipart/form-data">

@csrf

<div id="modal-master" class="modal-dialog modal-lg" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Import Data Barang</h5>

<button type="button" class="close" data-dismiss="modal" aria- label="Close"><span aria-hidden="true">&times;</span></button>

</div>

<div class="modal-body">

<div class="form-group">

<label>Download Template</label>

<a href="{{ asset('template\_barang.xlsx') }}" class="btn btn-info btn- sm" download><i class="fa fa-file-excel"></i>Download</a>

<small id="error-kategori\_id" class="error-text form-text text-

danger"></small>

</div>

<div class="form-group">

<label>Pilih File</label>

<input type="file" name="file\_barang" id="file\_barang" class="form-

control" required>

<small id="error-file\_barang" class="error-text form-text text-

danger"></small>

</div>

</div>

<div class="modal-footer">

<button type="button" data-dismiss="modal" class="btn btn- warning">Batal</button>

<button type="submit" class="btn btn-primary">Upload</button>

</div>

</div>

</div>

</form>

<script>

$(document).ready(function() {

$("#form-import").validate({ rules: {

file\_barang: {required: true, extension: "xlsx"},

},

submitHandler: function(form) {

var formData = new FormData(form); // Jadikan form ke FormData untuk

menghandle file

$.ajax({

url: form.action, type: form.method,

data: formData, // Data yang dikirim berupa FormData processData: false, // setting processData dan contentType ke false,

untuk menghandle file

contentType: false,

success: function(response) { if(response.status){ // jika sukses

$('#myModal').modal('hide'); Swal.fire({

icon: 'success', title: 'Berhasil', text: response.message

});

tableBarang.ajax.reload(); // reload datatable

}else{ // jika error

$('.error-text').text('');

$.each(response.msgField, function(prefix, val) {

$('#error-'+prefix).text(val[0]);

});

Swal.fire({

icon: 'error',

title: 'Terjadi Kesalahan', text: response.message

});

}

}

});

return false;

},

errorElement: 'span',

errorPlacement: function (error, element) { error.addClass('invalid-feedback'); element.closest('.form-group').append(error);

},

highlight: function (element, errorClass, validClass) {

$(element).addClass('is-invalid');

},

unhighlight: function (element, errorClass, validClass) {

$(element).removeClass('is-invalid');

}

});

});

</script>

* 1. Kemudian kita modifikasi route/web.php untuk mengakomodir proses upload file pada menu barang



* 1. Untuk bisa membaca/menulis file excel, maka kita butuhkan library untuk membaca/menulis file excel. Jadi kita bisa memakai library phpoffice/phpspreadsheet. Kita ketikkan perintah di terminal/CMD

composer require phpoffice/phpspreadsheet

* 1. Selanjutnya kita modifikasi file BarangController.php untuk memproses data

<?php

namespace App\Http\Controllers;

use App\Models\LevelModel; use App\Models\BarangModel; use App\Models\KategoriModel; use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Validator; use PhpOffice\PhpSpreadsheet\IOFactory; use Yajra\DataTables\Facades\DataTables;

class BarangController extends Controller

{

public function index (){

$activeMenu = 'barang';

$breadcrumb = (object) [ 'title' => 'Data Barang', 'list' => ['Home', 'Barang']

];

$kategori = KategoriModel::select('kategori\_id', 'kategori\_nama')->get(); return view('barang.index', [

'activeMenu' => $activeMenu, 'breadcrumb' => $breadcrumb, 'kategori' => $kategori

]);

}

public function list(Request $request)

{

$barang = BarangModel::select('barang\_id', 'barang\_kode', 'barang\_nama', 'harga\_beli', 'harga\_jual', 'kategori\_id')->with('kategori');

$kategori\_id = $request->input('filter\_kategori'); if(!empty($kategori\_id)){

$barang->where('kategori\_id', $kategori\_id);

}

return DataTables::of($barang)

->addIndexColumn()

->addColumn('aksi', function ($barang) { // menambahkan kolom aksi

/\*$btn = '<a href="'.url('/barang/' . $barang->barang\_id).'" class="btn btn- info btn-sm">Detail</a> ';

$btn .= '<a href="'.url('/barang/' . $barang->barang\_id . '/edit').'"class="btn btn-warning btn-sm">Edit</a> ';

$btn .= '<form class="d-inline-block" method="POST" action="'. url('/barang/'.$barang->barang\_id).'">'

. csrf\_field() . method\_field('DELETE') .

'<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\'Apakah Kita yakit menghapus data ini?\');">Hapus</button></form>';\*/

$btn = '<button onclick="modalAction(\''.url('/barang/' . $barang->barang\_id

. '/show\_ajax').'\')" class="btn btn-info btn-sm">Detail</button> ';

$btn .= '<button onclick="modalAction(\''.url('/barang/' . $barang-

>barang\_id . '/edit\_ajax').'\')" class="btn btn-warning btn-sm">Edit</button> ';

$btn .= '<button onclick="modalAction(\''.url('/barang/' . $barang-

>barang\_id . '/delete\_ajax').'\')" class="btn btn-danger btn-sm">Hapus</button> '; return $btn;

})

->rawColumns(['aksi']) // ada teks html

->make(true);

}

public function create\_ajax()

{

$kategori = KategoriModel::select('kategori\_id', 'kategori\_nama')->get(); return view('barang.create\_ajax')->with('kategori', $kategori);

}

public function store\_ajax(Request $request)

{

if($request->ajax() || $request->wantsJson()){

$rules = [

'kategori\_id' => ['required', 'integer', 'exists:m\_kategori,kategori\_id'], 'barang\_kode' => ['required', 'min:3', 'max:20',

'unique:m\_barang,barang\_kode'],

'barang\_nama' => ['required', 'string', 'max:100'], 'harga\_beli' => ['required', 'numeric'], 'harga\_jual' => ['required', 'numeric'],

];

$validator = Validator::make($request->all(), $rules); if($validator->fails()){

return response()->json([ 'status' => false,

'message' => 'Validasi Gagal', 'msgField' => $validator->errors()

]);

}

BarangModel::create($request->all()); return response()->json([

'status' => true,

'message' => 'Data berhasil disimpan'

]);

}

redirect('/');

}

public function edit\_ajax($id)

{

$barang = BarangModel::find($id);

$level = LevelModel::select('level\_id', 'level\_nama')->get();

return view('barang.edit\_ajax', ['barang' => $barang, 'level' => $level]);

}

public function update\_ajax(Request $request, $id)

{

// cek apakah request dari ajax

if ($request->ajax() || $request->wantsJson()) {

$rules = [

'kategori\_id' => ['required', 'integer', 'exists:m\_kategori,kategori\_id'], 'barang\_kode' => ['required', 'min:3', 'max:20',

'unique:m\_barang,barang\_kode, '. $id .',barang\_id'],

'barang\_nama' => ['required', 'string', 'max:100'], 'harga\_beli' => ['required', 'numeric'], 'harga\_jual' => ['required', 'numeric'],

];

// use Illuminate\Support\Facades\Validator;

$validator = Validator::make($request->all(), $rules); if ($validator->fails()) {

return response()->json([

'status' => false, // respon json, true: berhasil, false: gagal 'message' => 'Validasi gagal.',

'msgField' => $validator->errors() // menunjukkan field mana yang error

]);

}

$check = BarangModel::find($id); if ($check) {

$check->update($request->all()); return response()->json([

'status' => true,

'message' => 'Data berhasil diupdate'

]);

} else{

return response()->json([ 'status' => false,

'message' => 'Data tidak ditemukan'

]);

}

}

return redirect('/');

}

public function confirm\_ajax($id)

{

$barang = BarangModel::find($id);

return view('barang.confirm\_ajax', ['barang' => $barang]);

}

public function delete\_ajax(Request $request, $id)

{

if($request->ajax() || $request->wantsJson()){

$barang = BarangModel::find($id); if($barang){ // jika sudah ditemuikan

$barang->delete(); // barang di hapus return response()->json([

'status' => true,

'message' => 'Data berhasil dihapus'

]);

}else{

return response()->json([ 'status' => false,

'message' => 'Data tidak ditemukan'

]);

}

}

return redirect('/');

}

public function import()

{

return view('barang.import');

}

public function import\_ajax(Request $request)

{

if($request->ajax() || $request->wantsJson()){

$rules = [

// validasi file harus xls atau xlsx, max 1MB 'file\_barang' => ['required', 'mimes:xlsx', 'max:1024']

];

$validator = Validator::make($request->all(), $rules); if($validator->fails()){

return response()->json([ 'status' => false,

'message' => 'Validasi Gagal', 'msgField' => $validator->errors()

]);

}

$file = $request->file('file\_barang'); // ambil file dari request

$reader = IOFactory::createReader('Xlsx'); // load reader file excel

$reader->setReadDataOnly(true); // hanya membaca data

$spreadsheet = $reader->load($file->getRealPath()); // load file excel

$sheet = $spreadsheet->getActiveSheet(); // ambil sheet yang aktif

$data = $sheet->toArray(null, false, true, true); // ambil data excel

$insert = [];

if(count($data) > 1){ // jika data lebih dari 1 baris foreach ($data as $baris => $value) {

if($baris > 1){ // baris ke 1 adalah header, maka lewati

$insert[] = [

'kategori\_id' => $value['A'], 'barang\_kode' => $value['B'], 'barang\_nama' => $value['C'], 'harga\_beli' => $value['D'], 'harga\_jual' => $value['E'], 'created\_at' => now(),

];

}

}

if(count($insert) > 0){

// insert data ke database, jika data sudah ada, maka diabaikan BarangModel::insertOrIgnore($insert);

}

return response()->json([ 'status' => true,

'message' => 'Data berhasil diimport'

]);

}else{

return response()->json([ 'status' => false,

'message' => 'Tidak ada data yang diimport'

]);

}

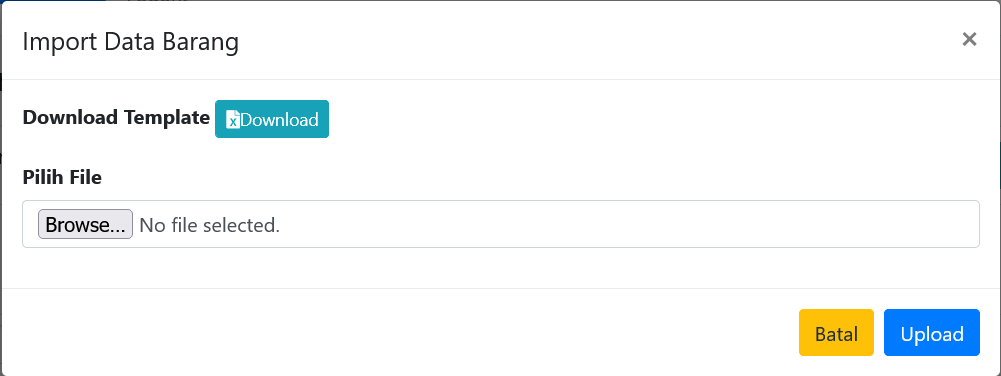
}

return redirect('/');

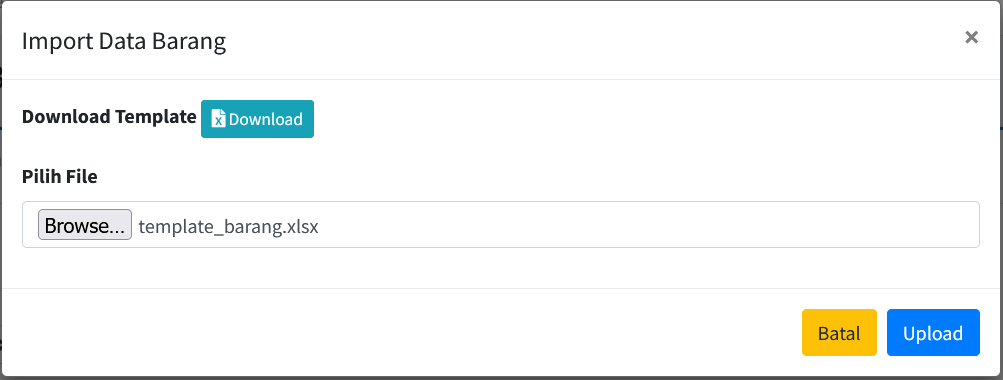
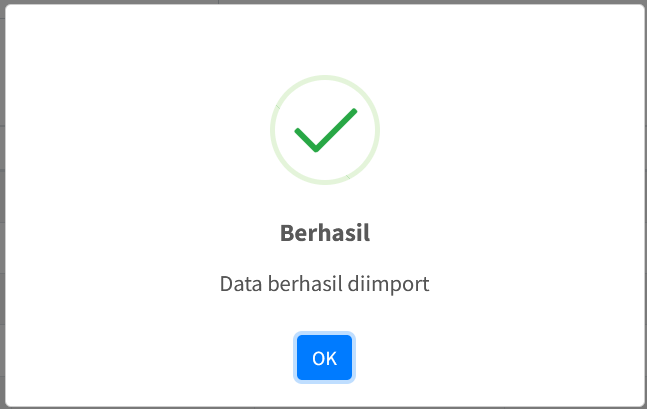
}

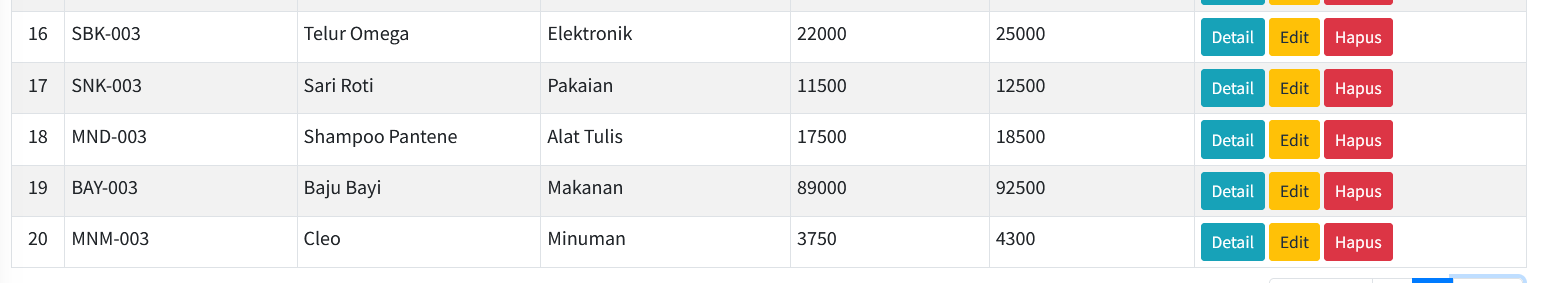
}

* 1. Sekarang kita coba jalankan browser dan klik tombol import pada menu Barang



* 1. Kita upload template data yang sudah kita siapkan, dan amati apa yang terjadi.

x

**Tugas 1** – Implementasi File Upload untuk Import Data:

1. Silahkan implementasikan praktikum 1 pada project kalian masing-masing untuk semua menu
2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
3. Menambahkan tombol untuk membuka view upload/import
4. Membuat view upload/import dan membuat template xlsx untuk import barang
5. Menambahkan library phpoffice/phpspreadsheet
6. Menambahkan method import dan import\_ajax pada BarangController
7. Menambahkan route pada route/web.php untuk melakukan impor
8. Submit kode untuk impementasi prakktikum 1 pada repository github kalian.
9. **Export Excel di Laravel**

Fitur export ke Excel pada Laravel merupakan alat yang sangat berguna dalam aplikasi web yang membutuhkan pengolahan dan penyajian data secara profesional. Fitur ini memungkinkan pengguna untuk mengekspor data dari database atau hasil pemrosesan lainnya ke dalam format file Excel (.xlsx atau .xls) yang bisa diunduh dan digunakan untuk berbagai tujuan, seperti pelaporan, analisis data, atau penyimpanan arsip.

Fitur export ke Excel di Laravel memungkinkan Kita untuk menyediakan data dalam format yang lebih fleksibel dan mudah digunakan oleh pengguna akhir.

**Praktikum 2** – Export Data ke Excel

Kita akan menerapkan export data dari database ke file excel pada project Laravel dengan menggunakan library yang sama dengan praktikum 1. Langkah-langkah yang kita kerjakan sebagai berikut:

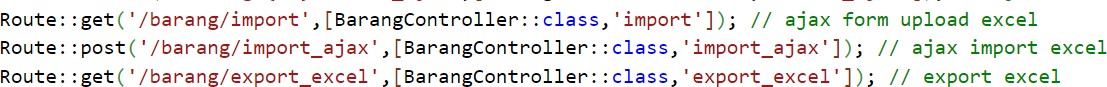
* 1. Kita modifikasi barang/index.blade.php dengan mengganti kode berikut

<a href="{{ url('/barang/create') }}" class="btn btn-primary">Tambah Data</a>

Kita ubah denganHal ini kita lakukan karena tombol **Tambah Data** sudah tidak kita gunakan karena kita sudah menggunakan tombol **Tambah Data (Ajax)**.

<a href="{{ url('/barang/export\_excel') }}" class="btn btn-primary"><i class="fa fa-file- excel"></i> Export Barang</a>

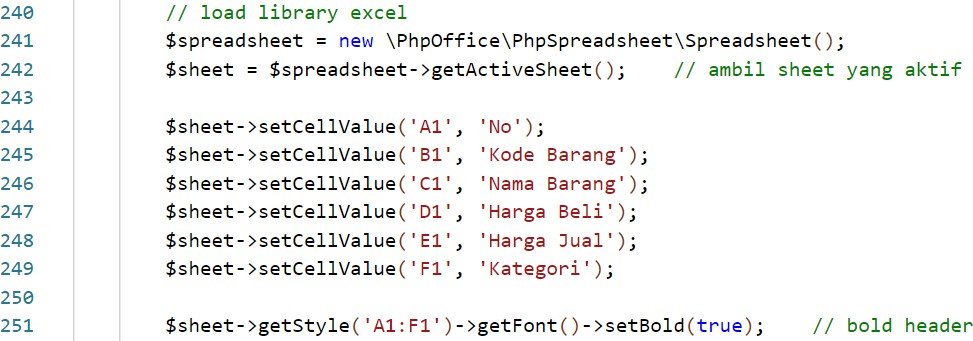
* 1. Kemudian kita tambahkan route pada route/web.php untuk bisa memproses export excel



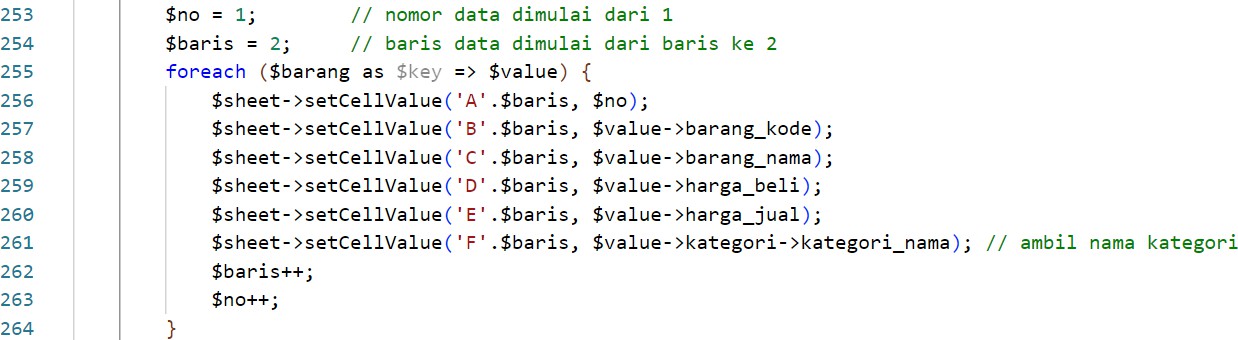
* 1. Selanjutnya kita tambahkan fungsi export\_excel() pada file BarangController.php
  2. Kita ambil data barang yang akan kita export ke excel (tentu dengan menyertakan relasi kategori barang)



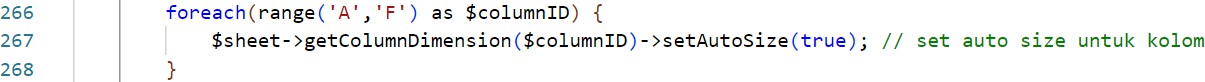
* 1. Kemudian kita load library Spreadsheet dan kita tentukan header data pada baris pertama di excel



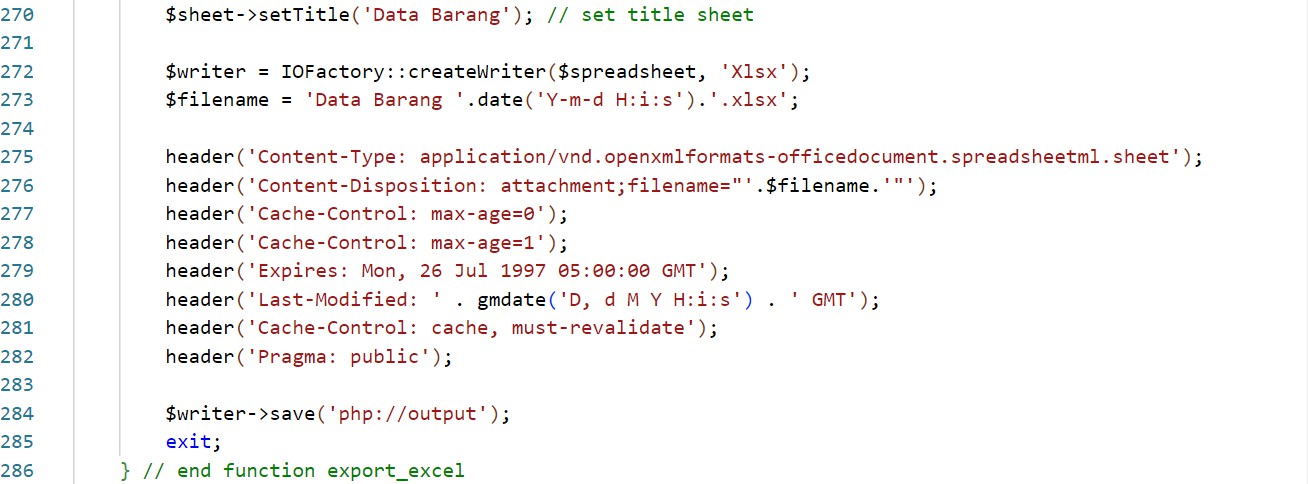
* 1. Selanjutnya, kita looping data yang telah kita dapatkan dari database, kemudian kita masukkan ke dalam cell excel



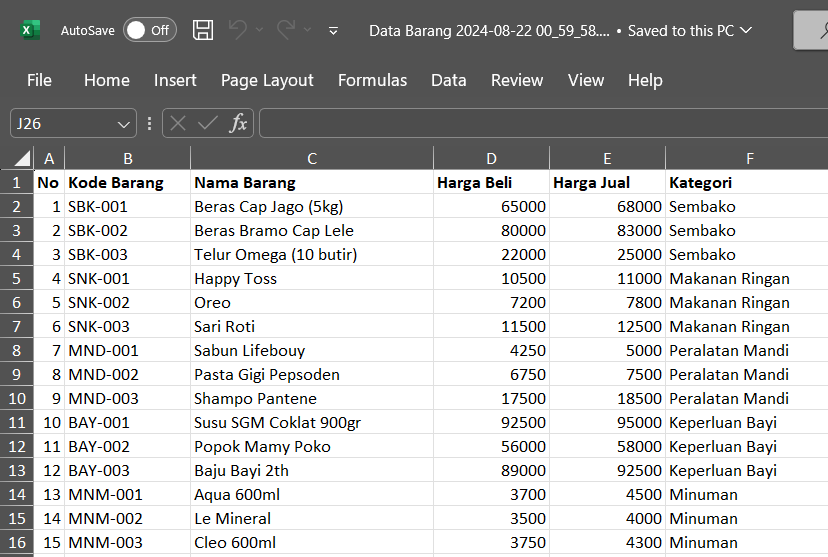
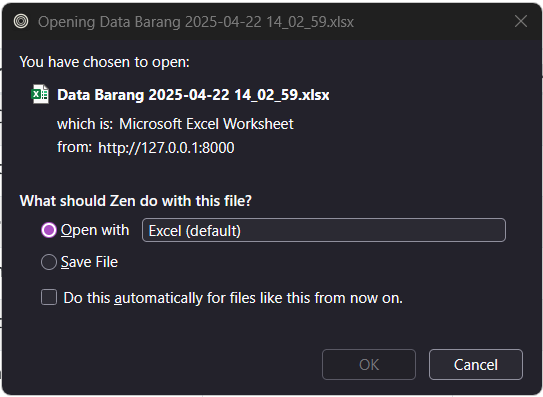
* 1. Kita set lebar tiap kolom di excel untuk menyesuaikan dengan panjang karakter pada masing-masing kolom



* 1. Bagian akhir proses export excel adalah kita set nama sheet, dan proses untuk dapat di download oleh pengguna



* 1. Jika sudah selesai diimplementasikan, kita coba untuk melakukan Download file Export tersebut.



**Tugas 2** – Implementasi File Export Excel:

1. Silahkan implementasikan praktikum 2 pada project kalian masing-masing untuk semua menu
2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
3. Menambahkan tombol untuk melakukan export ke excel
4. Menambahkan route pada route/web.php untuk bisa memproses export excel
5. Menambahkan method export\_excel() pada file BarangController
6. Submit kode untuk impementasi prakktikum 2 pada repository github kalian.
7. **Export PDF di Laravel**

Fitur **export ke PDF** pada Laravel sangat berguna dalam situasi di mana Kita perlu menghasilkan dokumen yang terstruktur, terformat dengan baik, dan mudah dibagikan. Dokumen PDF (Portable Document Format) adalah salah satu format yang paling sering digunakan untuk laporan, faktur, tiket, dan berbagai dokumen resmi lainnya karena stabilitasnya dan kemampuan untuk mempertahankan tata letak yang konsisten di berbagai perangkat.

Untuk mengimplementasikan export ke PDF di Laravel, salah satu paket yang paling populer adalah **dompdf**. Dengan menggunakan paket dompdf, Kita dapat dengan mudah mengubah tampilan HTML menjadi PDF yang dapat disesuaikan sesuai kebutuhan. Fitur ini sangat bermanfaat dalam berbagai konteks, seperti pembuatan faktur, laporan, sertifikat, dan dokumen-dokumen penting lainnya.

**Praktikum 3** – Implementasi Export PDF di Laravel dengan dompdf

Kita akan menerapkan export data ke file pdf pada project Laravel dengan menggunakan library dompdf. Langkah-langkah yang kita kerjakan sebagai berikut:

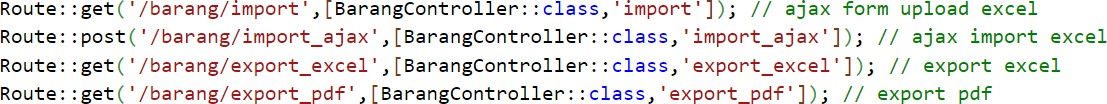
* 1. Kita lakukan proses instalasi library dompdf terlebih dahulu dengan mengetikkan perintah pada terminal/CMD

composer require barryvdh/laravel-dompdf

* 1. Setelah proses instalasi dompdf berhasil, selanjutnya Kita tambahkan kode berikut untuk menambahkan tombol export ke pdf pada barang/index.blade.php

<a href="{{ url('/barang/export\_pdf') }}" class="btn btn-warning"><i class="fa fa-file- pdf"></i> Export Barang</a>

* 1. Setelah itu tinggal kita perbaiki route/web.php untuk proses export pdf



* 1. Selanjutnya, kita buat fungsi export\_pdf() pada BarangController.php



* 1. Selanjutnya, kita buat view untuk dijadikan pdf dari layout html. File bisa kita buat dengan nama barang/export\_pdf.blade.php

**INFO**

Perlu diingat, pada export\_pdf, kita gunakan style CSS secara manual, dan tidak menggunakan framework CSS seperti bootstrap. Karena library dompdf tidak support semua fitur styling yang ada di framework CSS.

* 1. Selanjutnya kita buat view untuk me-generate html untuk tampilan pdf yang akan kita sajikan. View berada di barang/export\_pdf.blade.php

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<style>

body{

font-family: "Times New Roman", Times, serif; margin: 6px 20px 5px 20px;

line-height: 15px;

}

table {

width:100%;

border-collapse: collapse;

}

td, th {

padding: 4px 3px;

}

th{

text-align: left;

}

.d-block{

display: block;

}

img.image{

width: auto; height: 80px;

max-width: 150px; max-height: 150px;

}

.text-right {

text-align: right;

}

.text-center {

text-align: center;

}

.p-1{

padding: 5px 1px 5px 1px;

}

.font-10{

font-size: 10pt;

}

.font-11{

font-size: 11pt;

}

.font-12{

font-size: 12pt;

}

.font-13{

font-size: 13pt;

}

.border-bottom-header{ border-bottom: 1px solid;

}

.border-all, .border-all th, .border-all td{ border: 1px solid;

}

</style>

</head>

<body>

<table class="border-bottom-header">

<tr>

<td width="15%" class="text-center"><img src="{{ asset('polinema-bw.png')

}}"></td>

<td width="85%">

<span class="text-center d-block font-11 font-bold mb-1">KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI</span>

<span class="text-center d-block font-13 font-bold mb-1">POLITEKNIK NEGERI

MALANG</span>

<span class="text-center d-block font-10">Jl. Soekarno-Hatta No. 9 Malang

65141</span>

<span class="text-center d-block font-10">Telepon (0341) 404424 Pes. 101-

105, 0341-404420, Fax. (0341) 404420</span>

<span class="text-center d-block font-10">Laman: [www.polinema.ac.id](http://www.polinema.ac.id/)</span>

</td>

</tr>

</table>

<h3 class="text-center">LAPORAN DATA BARANG</h4>

<table class="border-all">

<thead>

<tr>

<th class="text-center">No</th>

<th>Kode Barang</th>

<th>Nama Barang</th>

<th class="text-right">Harga Beli</th>

<th class="text-right">Harga Jual</th>

<th>Kategori</th>

</tr>

</thead>

<tbody>

@foreach($barang as $b)

<tr>

<td class="text-center">{{ $loop->iteration }}</td>

<td>{{ $b->barang\_kode }}</td>

<td>{{ $b->barang\_nama }}</td>

<td class="text-right">{{ number\_format($b->harga\_beli, 0, ',', '.') }}</td>

<td class="text-right">{{ number\_format($b->harga\_jual, 0, ',', '.') }}</td>

<td>{{ $b->kategori->kategori\_nama }}</td>

</tr>

@endforeach

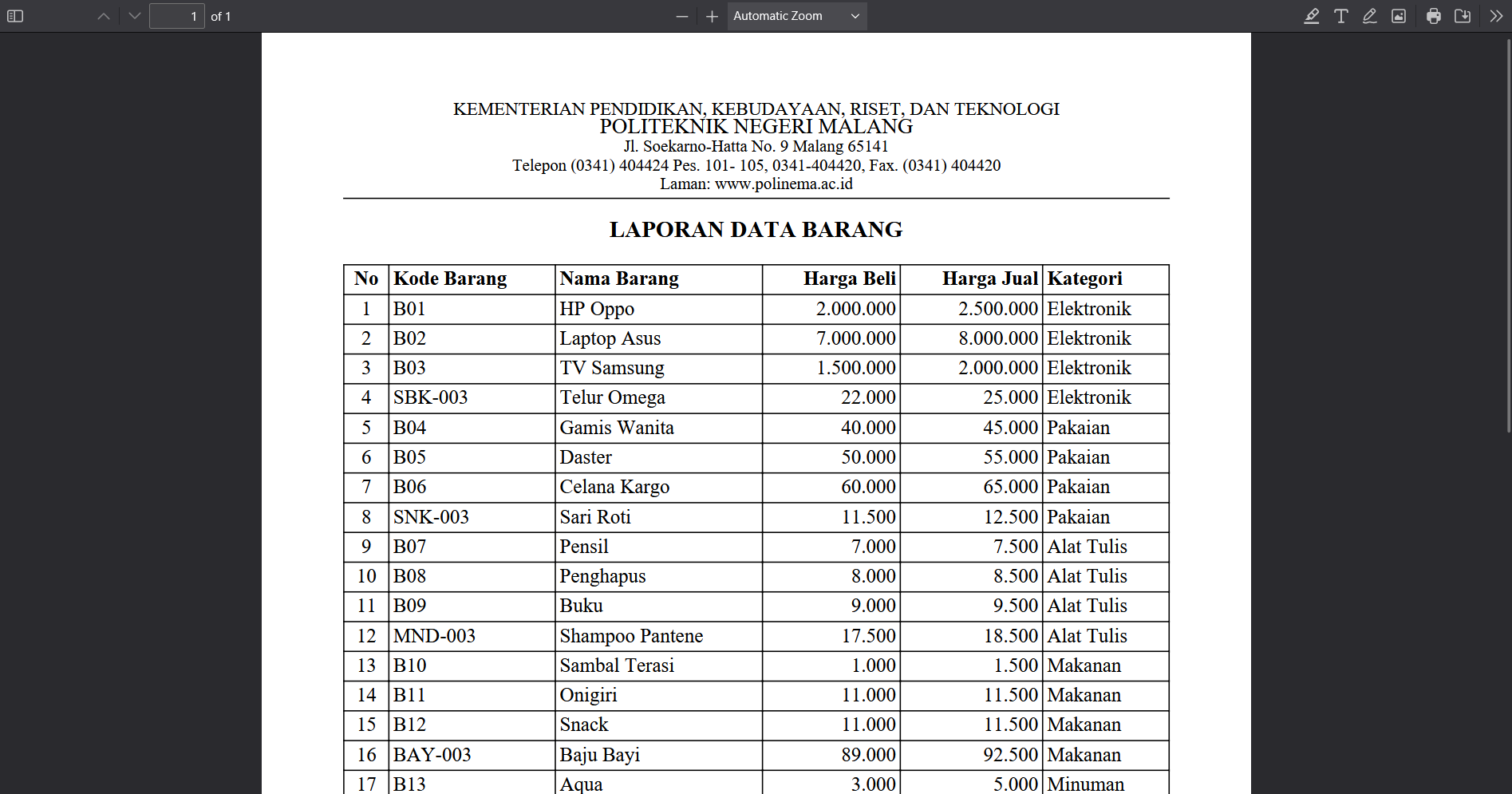
</tbody>

</table>

</body>

</html>

* 1. Selanjutnya, Kita coba untuk melakukan proses download export pdf. Amati dan pelajari…!!!

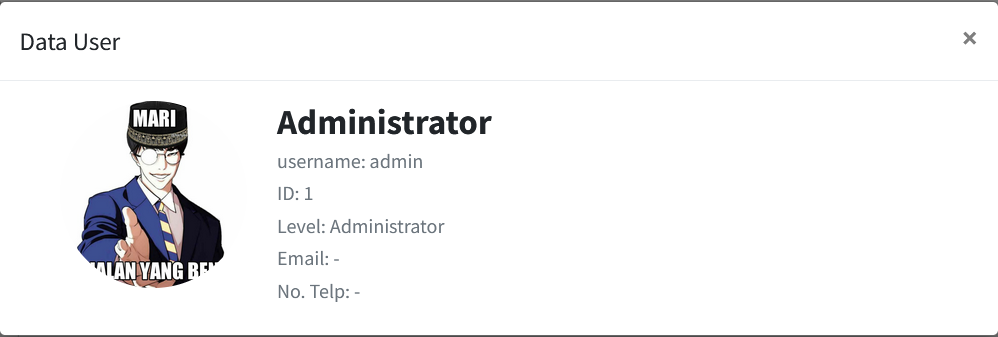
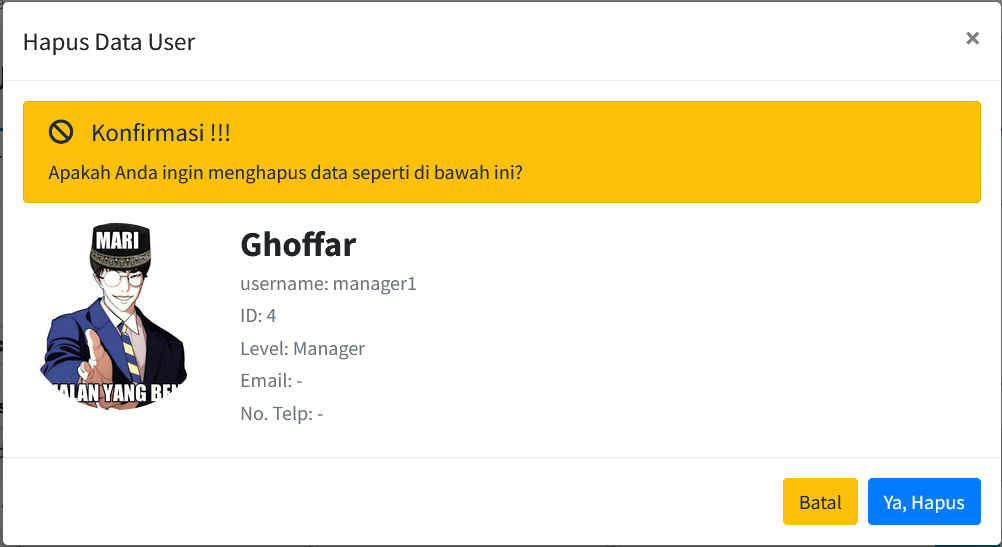


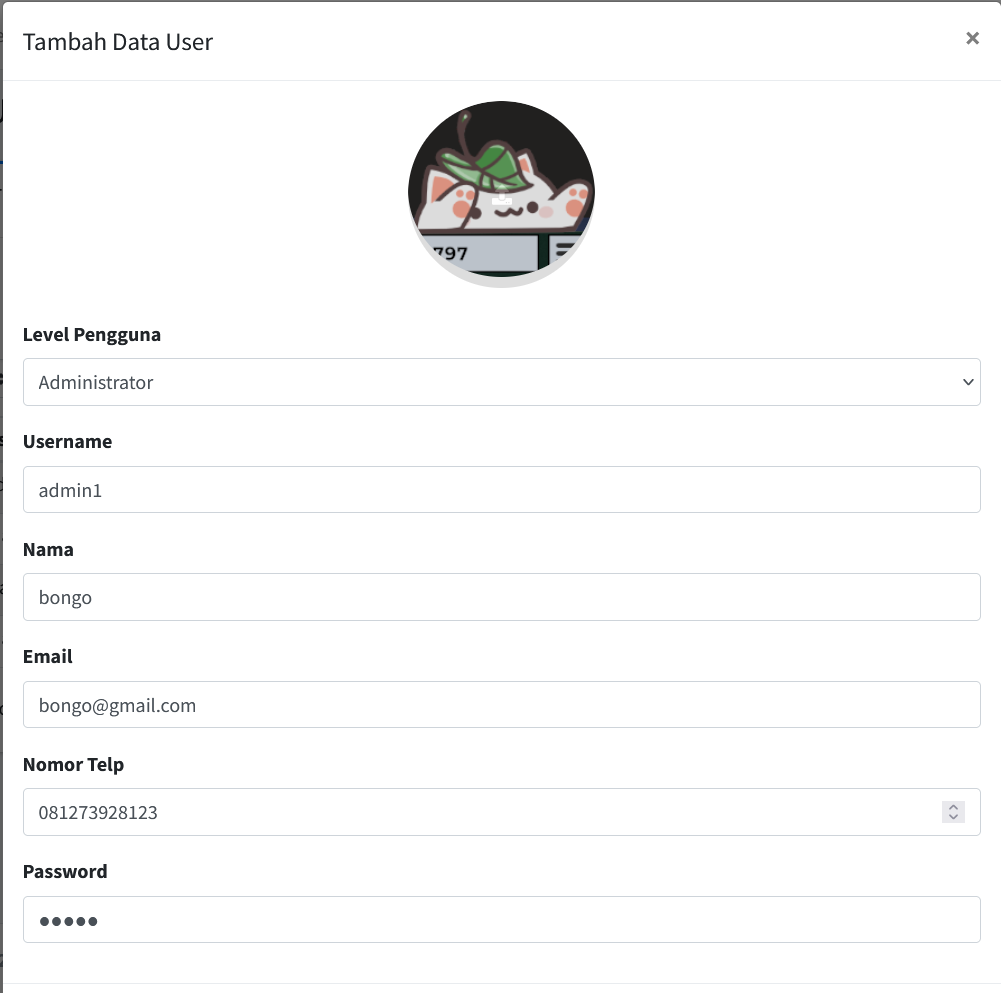
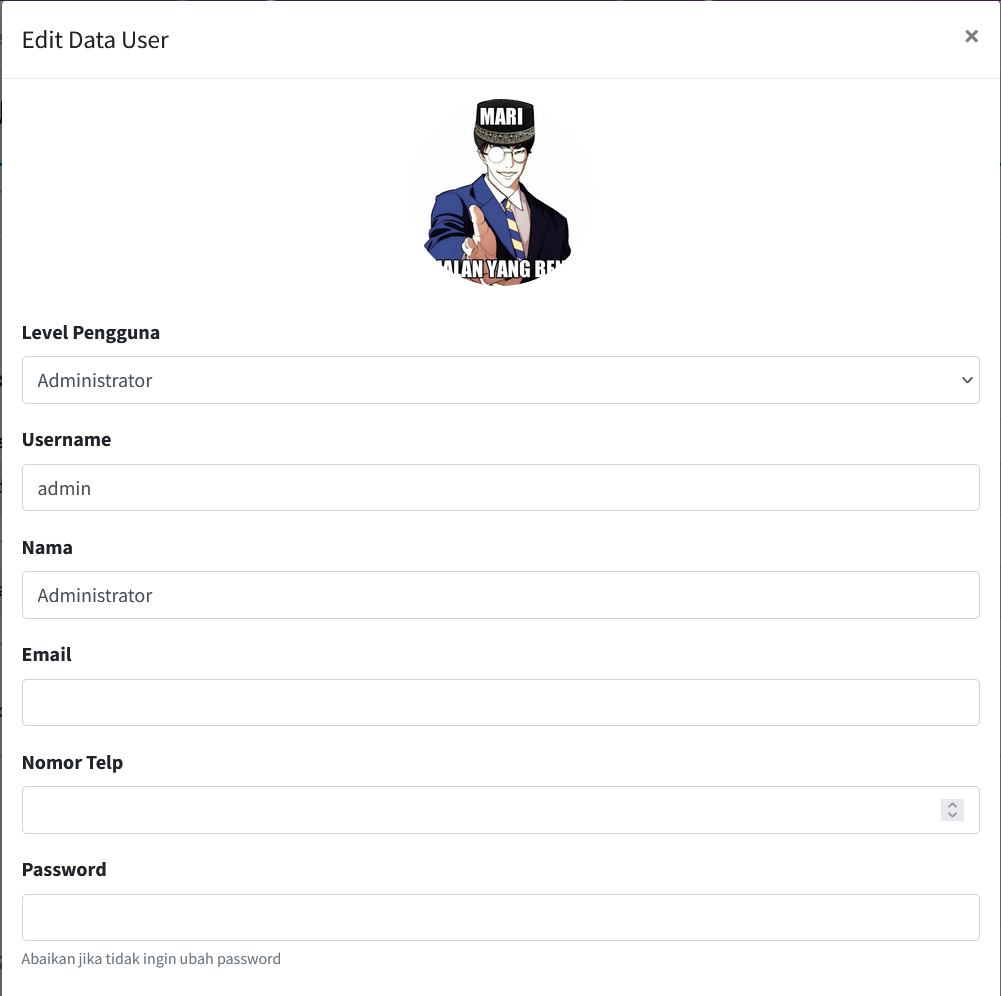
**Tugas 3** – Implementasi Export PDF pada Laravel :

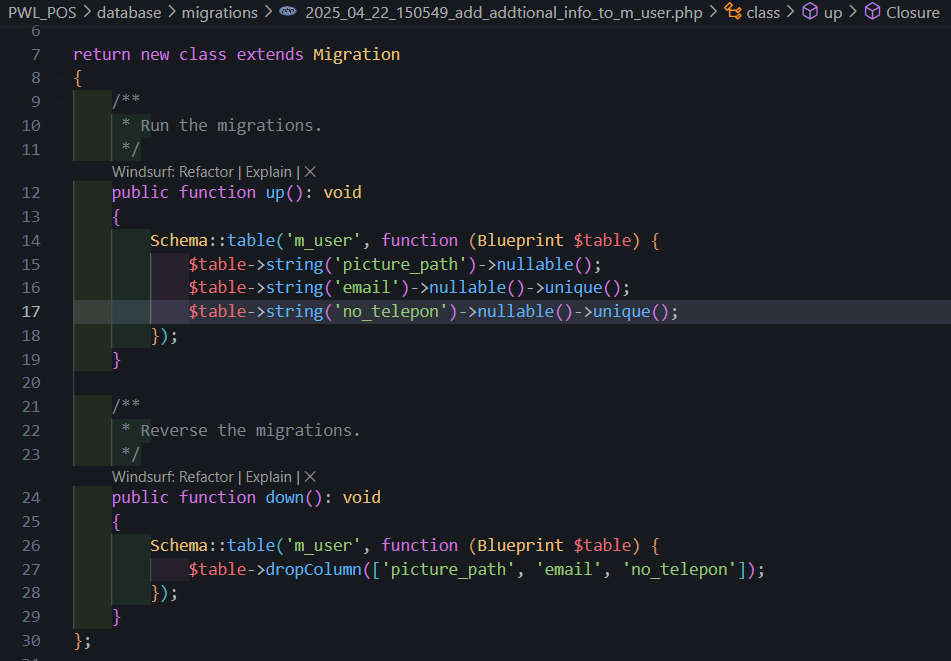
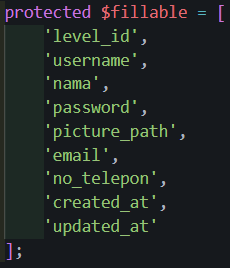
1. Silahkan implementasikan export pdf pada project kalian masing-masing untuk semua menu
2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
3. Menambahkan library barryvdh/laravel-dompdf
4. Menambahkan tombol untuk melakukan export ke pdf
5. Menambahkan route pada route/web.php untuk bisa memproses export pdf
6. Menambahkan method export\_pdf() pada BarangController
7. Membuat view untuk tampilan pdf
8. Submit kode untuk impementasi export pdf pada repository github kalian.

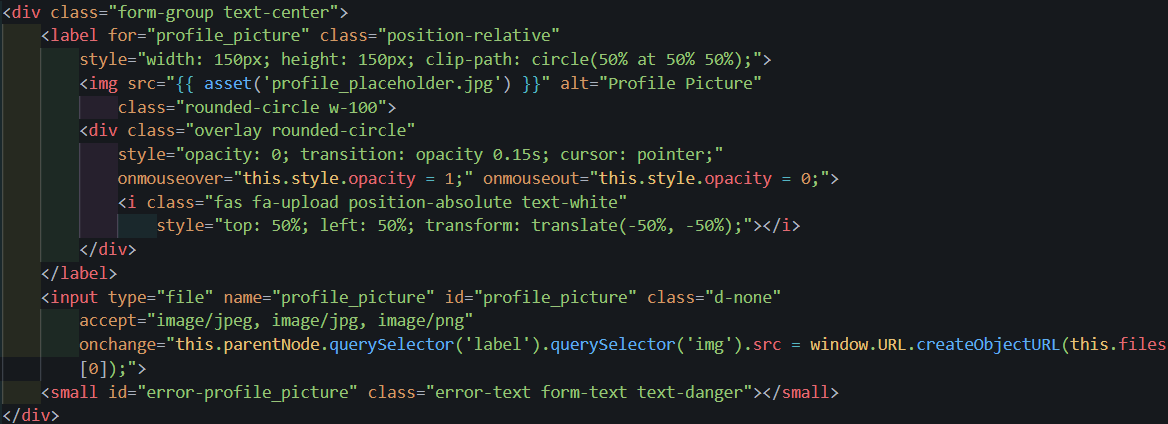
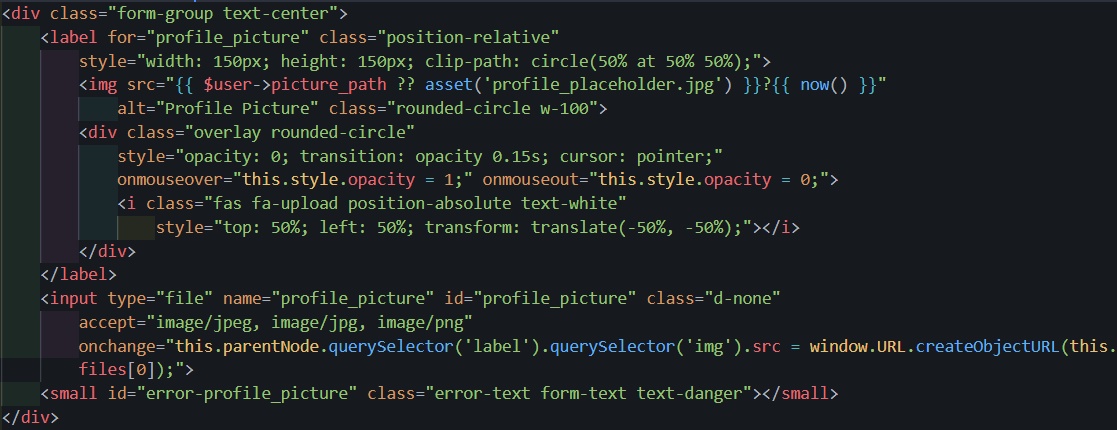
**Tugas 4** – Implementasi Upload File Gambar :

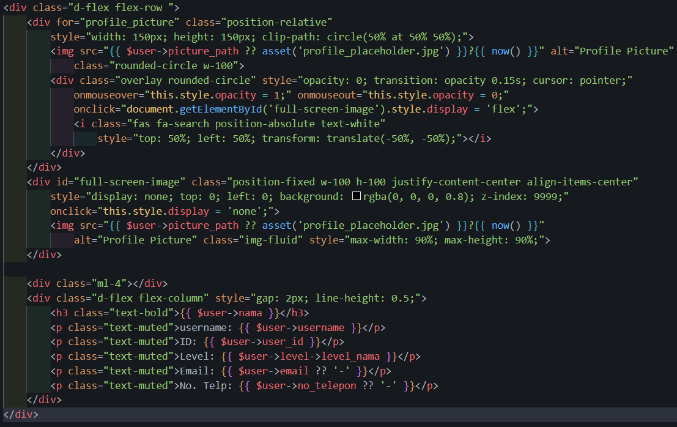
1. Silahkan implementasikan fitur **upload file** untuk mengubah **foto profile** di project web kalian

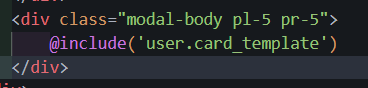
 

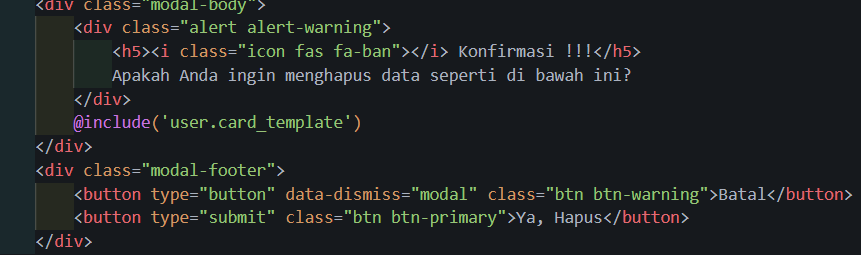
 

1. Jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
2. Membuat migration baru untuk menambahkan tempat menyimpan path dari gambar foto profil pada tabel m\_user
3. Pada UserModel tambahkan atribut baru yang sudah dibuat pada fillable agar dapat diisi
4. Mengubah tipe form data dengan menambahkan enctype multipart/form-data
5. Mengedit view user: create, edit, show, dan confirm. Pada bagian create dan edit, menambahkan field untuk mengisi/mengedit atribut yang tadi ditambahkan

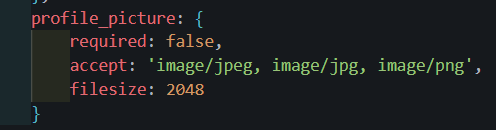
* Create – form gambar
* Edit – form gambar

Pada bagian show dan confirm dirubah agar dapat menampilkan foto profil, dengan membuat view card template baru agar dapat digunakan kembali pada 2 view yang berbeda, yaitu:

Dan dapat digunakan pada view show:

dan confirm:

1. Menambahkan validasi untuk ajax dan pada sisi server

Ajax:

Sisi server:

1. Melakukan penyimpanan pada gambar, dengan menyimpan gambar pada folder storage/app/public/profile\_pictures dan menyimpan path untuk gambarnya pada tabel m\_user (“picture\_path”)
2. Submit kode untuk impementasi export pdf pada repository github kalian.

*\*\*\* Sekian, dan selamat belajar \*\*\**