



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

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

A. 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`

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	
		1	FB	Food & Beverage	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	
		2	BH	Beauty & Health	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	
		3	HC	Home Care	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	
		4	BK	Baby & Kids	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	
		5	EL	Electronics	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	
		7	CML	Cemilan	2024-03-09 10:30:00	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	
		8	MNR	Minuman Ringan	2024-03-09 13:01:39	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	
		9	SPT	Sepatu	2025-03-17 16:07:25	2025-03-17 16:07:25

		barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		1	1	B001	Air Mineral	2000	3000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		2	1	B002	Roti Tawar	10000	12000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		3	2	B003	Sabun Mandi	5000	7000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		4	2	B004	Shampoo	15000	18000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		5	3	B005	Pembersih Lantai	10000	13000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		6	3	B006	Tisu	8000	10000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		7	4	B007	Popok Bayi	50000	55000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		8	4	B008	Susu Formula	60000	65000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		9	5	B009	Lampu LED	25000	30000	NULL	NULL
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete				
		10	5	B010	Charger HP	35000	40000	NULL	NULL

2. 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.

	A	B	C	D	E	F
1	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	
2	1	SBK-003	Telur Omega (10 butir)	22000	25000	
3	2	SNK-003	Sari Roti	11500	12500	
4	3	MND-003	Shampo Pantene	17500	18500	
5	4	BAY-003	Baju Bayi 2th	89000	92500	
6	5	MNM-003	Cleo 600ml	3750	4300	



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.

3. 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
        </div>
    </div>
```

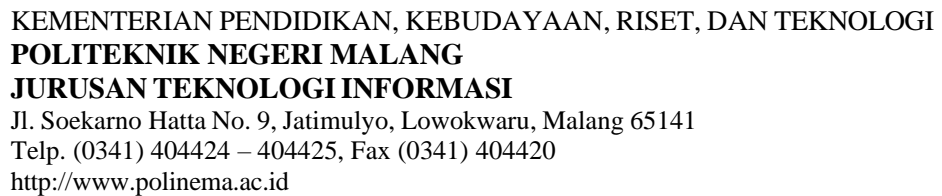


```
<table class="table table-bordered table-sm table-striped table-hover"
id="table-barang">
  <thead>
    <tr><th>No</th><th>Kode Barang</th><th>Kode Barang</th><th>Harga
Beli</th><th>Harga Jual</th><th>Kategori</th><th>Aksi</th></tr>
  </thead>
  <tbody></tbody>
</table>
</div>
</div>
<div id="myModal" 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%",
```

[illegible]



4. 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">
```

```
1 <form action="{{ url('/barang/import_ajax') }}" method="POST" id="form-import" enctype="multipart/form-data">
2     @csrf
3     <div id="modal-master" class="modal-dialog modal-lg" role="document">
4         <div class="modal-content">
5             <div class="modal-header">
6                 <h5 class="modal-title" id="exampleModalLabel">Import Data Barang</h5>
7                 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
8                     <span aria-hidden="true">&times;</span></button>
9             </div>
10            <div class="modal-body">
11                <div class="form-group">
12                    <label>Download Template</label>
13                    <a href="{{ asset('template_barang.xlsx') }}" class="btn btn-info btn-sm" download>
14                        <i class="fa fa-file-excel"></i>Download
15                    </a>
16                    <small id="error-kategori_id" class="error-text form-text text-danger"></small>
17                </div>
18                <div class="form-group">
19                    <label>Pilih File</label>
20                    <input type="file" name="file_barang" id="file_barang" class="form-control" required>
21                    <small id="error-file_barang" class="error-text form-text text-danger"></small>
22                </div>
23            </div>
24            <div class="modal-footer">
25                <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
26                <button type="submit" class="btn btn-primary">Upload</button>
27            </div>
28        </div>
29    </div>
30 </form>
31
32 <script>
33     $(document).ready(function() {
34         $("#form-import").validate({
35             rules: {
36                 file_barang: {
37                     required: true,
38                     extension: "xlsx"
39                 },
40             },
41         });
42     });
43 </script>
```

```
41 submitHandler: function(form) {
42     var formData = new FormData(form); // Jadikan form ke FormData untuk menghandle file
43
44     $.ajax({
45         url: form.action,
46         type: form.method,
47         data: formData, // Data yang dikirim berupa FormData
48         processData: false, // setting processData ke false, untuk menghandle file
49         contentType: false,
50         success: function(response) {
51             if(response.status) { // jika sukses
52                 $("#myModal").modal("hide");
53                 Swal.fire({
54                     icon: "success",
55                     title: "Berhasil",
56                     text: response.message
57                 });
58                 tableBarang.ajax.reload(); // reload datatable
59             } else { // jika error
60                 $("#error-tabel").text("");
61                 $.each(response.msgfield, function(prefix, val) {
62                     $("#error-"+prefix).text(val[0]);
63                 });
64                 Swal.fire({
65                     icon: "error",
66                     title: "Terjadi Kesalahan",
67                     text: response.message
68                 });
69             }
70         }
71     });
72     return false;
73 },
74 errorElement: "span",
75 errorPlacement: function(error, element) {
76     error.addClass("invalid-feedback");
77     element.closest(".form-group").append(error);
78 },
79 highlight: function(element, errorClass, validClass) {
80     $(element).addClass("is-invalid");
81 },
82 unhighlight: function(element, errorClass, validClass) {
83     $(element).removeClass("is-invalid");
84 },
85 });
86 </script>
```



```
<button type="button" data-dismiss="modal" class="btn btn-  
warning">Batal</button>  
<button type="submit" class="btn btn-primary">Upload</button>  
</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>
```

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



```
124 // Import Barang with Excel
125 Route::get(uri: 'import', action: [BarangController::class, 'import']); // ajax form upload excel
126 Route::post(uri: 'import_ajax', action: [BarangController::class, 'import_ajax']); // ajax import excel
127
```

6. Untuk bisa membaca/menulis file excel, maka kita butuh library untuk membaca/menulis file excel. Jadi kita bisa memakai library [phpoffice/phpspreadsheet](#). Kita ketikkan perintah di terminal/CMD

```
composer require phpoffice/phpspreadsheet
```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.26100.3624]
(c) Microsoft Corporation. All rights reserved.

C:\laragon\www\PWL_2025\Week8\JS8>composer require phpoffice/phpspreadsheet
./composer.json has been updated
Running composer update phpoffice/phpspreadsheet
Loading composer repositories with package information
Updating dependencies
Lock file operations: 5 installs, 0 updates, 0 removals
- Locking composer/pcre (3.3.2)
- Locking maennchen/zipstream-php (3.1.1)
- Locking markbaker/complex (3.0.2)
- Locking markbaker/matrix (3.0.1)
- Locking phpoffice/phpspreadsheet (4.1.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 5 installs, 0 updates, 0 removals
```

7. 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);
        }
    }
}
```




```
/*$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 yakin 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 ditemukan
            $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 atauxlsx, 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

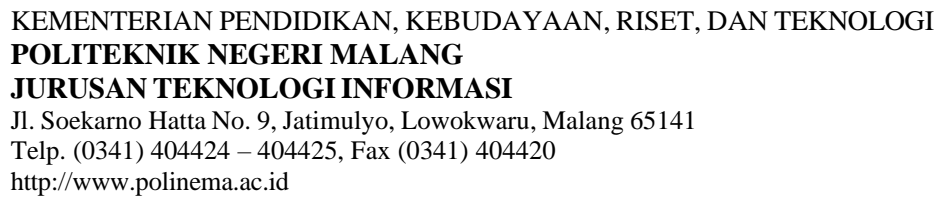
        $reader = IOFactory::createReader('Xlsx'); // load reader file excel
        $reader->setReadDataOnly(true); // hanya membaca data
        $spreadsheet = $reader->load($file->getRealPath()); // load file excel
        $sheet = $spreadsheet->getActiveSheet(); // ambil sheet yang aktif

        $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('/');
```

[illegible]

- Import Data Barang

Download Template

Download

Pilih File

Choose File

No file chosen

Batal

Upload

- Import Data Barang

Download Template

Download

Pilih File

Choose File

template_barang.xlsx

Batal

Upload

Show	10	entries	Search: <input type="text"/>			
No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori	Aksi
11	B011	Telur Omega(10 butir)	22.000	25.000	Food & Beverage	Detail Edit Hapus
12	B012	Sari Roti	11.500	12.500	Food & Beverage	Detail Edit Hapus
13	B013	Shampo Pantene	17.500	18.500	Home Care	Detail Edit Hapus
14	B014	Baju Bayi 2th	89.000	92.500	Baby & Kids	Detail Edit Hapus
15	B015	Cleo 600ml	3.750	4.300	Food & Beverage	Detail Edit Hapus



Tugas 1 – Implementasi File Upload untuk Import Data:

1. Silahkan implementasikan praktikum 1 pada project kalian masing-masing untuk semua menu

➤ Data user

	A	B	C	D
1	user_id	username	nama	level_id
2	17	kasir	icha	3
3	18	kasir1	juju	3
4	19	kasir2	bintang	3

Import Data Pengguna

Download Template

Download

Pilih File

Choose File

template_user.xlsx

Batal

Upload

➤ Data level

	A	B	C
1	level_id	level_kode	level_nama
2	7	SUP	Supervisor
3	8	FIN	Finance
4	9	MKT	Marketing

Import Data Level

Download Template

Download

Pilih File

Choose File

template_level.xlsx

Batal

Upload

➤ Data kategori

	A	B	C
1	kategori_kode	kategori_nama	
2	FSH	Fashion	
3	SPR	Sports	
4	TOY	Toys	
5	BKY	Books & Stationery	



Import Data Kategori

Download Template [Download](#)

Pilih File

[Choose File](#) template_kategori.xlsx

[Batal](#) [Upload](#)

➤ Data supplier

	A	B	C	D	E
1	supplier_kod	supplier_nama	supplier_alamat		
2	SP004	PT Mega Bersama	Jl. Sudirman No. 45, Jakarta Pusat		
3	SP005	PT Cahaya Abadi	Jl. Merdeka No. 12, Yogyakarta		
4	SP006	PT Global Sentosa	Jl. Ahmad Yani No. 99, Medan		

Import Data Supplier

Download Template [Download](#)

Pilih File

[Choose File](#) template_supplier.xlsx

[Batal](#) [Upload](#)

- Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
 - Membuat file Excel (.xlsx) berisi contoh data agar pengguna mudah mengisi dan terhindar dari kesalahan format saat mengimpor data.
 - Setiap menu yang butuh fitur import memiliki halaman khusus (import.blade.php) yang berisi form upload file Excel.
 - import: Menangani upload file secara langsung (form biasa).
 - import_ajax: Menangani upload file dengan AJAX tanpa reload halaman.
 - Fungsi import: Membaca dan menyimpan data dari file Excel ke database.
 - Fungsi import_ajax: Sama seperti import, tapi merespon dengan JSON agar bisa digunakan oleh AJAX.
- Submit kode untuk impementasi prakktikum 1 pada repository github kalian.

B. 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 mengeksport 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 dengan

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

```
9 | <a href="{{ url(path: '/barang/export_excel') }}" class="btn btn-primary"><i class="fa fa-file-excel"></i> Export Barang</a>  
10 | <button onclick="modalAction('{{ url(path: '/barang/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
```

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

2. Kemudian kita tambahkan route pada `route/web.php` untuk bisa memproses export excel

```
Route::get('/barang/import',[BarangController::class,'import']); // ajax form upload excel  
Route::post('/barang/import_ajax',[BarangController::class,'import_ajax']); // ajax import excel  
Route::get('/barang/export_excel',[BarangController::class,'export_excel']); // export excel
```

```
134 |  
135 | Route::get(uri: 'export_excel', action: [BarangController::class, 'export_excel']); //export excel  
136 | );
```

3. Selanjutnya kita tambahkan fungsi `export_excel()` pada file `BarangController.php`
Kita ambil data barang yang akan kita export ke excel (tentu dengan menyertakan relasi kategori barang)

```
232 | public function export_excel()  
233 | {  
234 |     // ambil data barang yang akan di export  
235 |     $barang = BarangModel::select('kategori_id','barang_kode','barang_nama','harga_beli','harga_jual')  
236 |         ->orderBy('kategori_id')  
237 |         ->with('kategori')  
238 |         ->get();
```

```
1 reference | 0 overrides  
233 | public function export_excel(): never  
234 |  
235 | // Ambil data barang yang akan diekspor  
236 | $barang = BarangModel::select(columns: 'kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')  
237 |     ->orderBy(column: 'kategori_id')  
238 |     ->with(relations: 'kategori')  
239 |     ->get();
```

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



```
240 // load library excel
241 $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
242 $sheet = $spreadsheet->getActiveSheet(); // ambil sheet yang aktif
243
244 $sheet->setCellValue('A1', 'No');
245 $sheet->setCellValue('B1', 'Kode Barang');
246 $sheet->setCellValue('C1', 'Nama Barang');
247 $sheet->setCellValue('D1', 'Harga Beli');
248 $sheet->setCellValue('E1', 'Harga Jual');
249 $sheet->setCellValue('F1', 'Kategori');
250
251 $sheet->getStyle('A1:F1')->getFont()->setBold(true); // bold header
```

```
241 // Load library PhpSpreadsheet
242 $spreadsheet = new Spreadsheet();
243 $sheet = $spreadsheet->getActiveSheet();
244
245 // Set header kolom
246 $sheet->setCellValue('A1', 'No');
247 $sheet->setCellValue('B1', 'Kode Barang');
248 $sheet->setCellValue('C1', 'Nama Barang');
249 $sheet->setCellValue('D1', 'Harga Beli');
250 $sheet->setCellValue('E1', 'Harga Jual');
251 $sheet->setCellValue('F1', 'Kategori');
252
253 // Format header bold
254 $sheet->getStyle('A1:F1')->getFont()->setBold(true);
```

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

```
253 $no = 1; // nomor data dimulai dari 1
254 $baris = 2; // baris data dimulai dari baris ke 2
255 foreach ($barang as $key => $value) {
256     $sheet->setCellValue('A' . $baris, $no);
257     $sheet->setCellValue('B' . $baris, $value->barang_kode);
258     $sheet->setCellValue('C' . $baris, $value->barang_nama);
259     $sheet->setCellValue('D' . $baris, $value->harga_beli);
260     $sheet->setCellValue('E' . $baris, $value->harga_jual);
261     $sheet->setCellValue('F' . $baris, $value->kategori->kategori_nama); // ambil nama kategori
262     $baris++;
263     $no++;
264 }
```

```
256 // Isi data barang
257 $no = 1;
258 $baris = 2;
259 foreach ($barang as $value) {
260     $sheet->setCellValue('A' . $baris, $no);
261     $sheet->setCellValue('B' . $baris, $value->barang_kode);
262     $sheet->setCellValue('C' . $baris, $value->barang_nama);
263     $sheet->setCellValue('D' . $baris, $value->harga_beli);
264     $sheet->setCellValue('E' . $baris, $value->harga_jual);
265     $sheet->setCellValue('F' . $baris, $value->kategori->kategori_nama);
266     $baris++;
267     $no++;
268 }
269 }
```

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

```
266 foreach(range('A','F') as $columnID) {
267     $sheet->getColumnDimension($columnID)->setAutoSize(true); // set auto size untuk kolom
268 }
```

```
270 // Set auto size untuk kolom
271 foreach (range(start: 'A', end: 'F') as $columnID) {
272     $sheet->getColumnDimension($columnID)->setAutoSize(true);
273 }
```

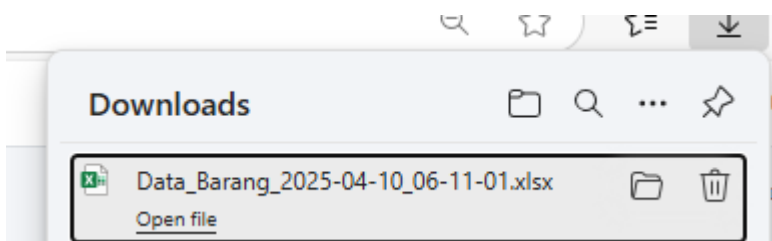


7. Bagian akhir proses export excel adalah kita set nama sheet, dan proses untuk dapat di download oleh pengguna

```
270 $sheet->setTitle('Data Barang'); // set title sheet
271
272 $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
273 $filename = 'Data Barang '.date('Y-m-d H:i:s').'.xlsx';
274
275 header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
276 header('Content-Disposition: attachment;filename="'. $filename .'"');
277 header('Cache-Control: max-age=0');
278 header('Cache-Control: max-age=1');
279 header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
280 header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
281 header('Cache-Control: cache, must-revalidate');
282 header('Pragma: public');
283
284 $writer->save('php://output');
285 exit;
286 } // end function export_excel
```

```
275 // Set title sheet
276 $sheet->setTitle('Data Barang');
277
278 // Generate filename
279 $filename = 'Data_Barang_'.date(format: 'Y-m-d_H-i-s') . '.xlsx';
280
281 // Set header untuk download file
282 header(header: 'Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
283 header(header: 'Content-Disposition: attachment;filename="'. $filename .'"');
284 header(header: 'Cache-Control: max-age=0');
285 header(header: 'Expires: Mon, 26 Jul 1997 05:00:00 GMT');
286 header(header: 'Last-Modified: ' . gmdate(format: 'D, d M Y H:i:s') . ' GMT');
287 header(header: 'Cache-Control: cache, must-revalidate');
288 header(header: 'Pragma: public');
289
290 $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
291 $writer->save('php://output');
292 exit;
293
```

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



	A	B	C	D	E	F
1	No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
2	1	B001	Air Mineral	2000	3000	Food & Beverage
3	2	B002	Roti Tawar	10000	12000	Food & Beverage
4	3	B003	Sabun Mandi	5000	7000	Beauty & Health
5	4	B004	Shampoo	15000	18000	Beauty & Health
6	5	B005	Pembersih Lantai	10000	13000	Home Care
7	6	B006	Tisu	8000	10000	Home Care
8	7	B007	Popok Bayi	50000	55000	Baby & Kids
9	8	B008	Susu Formula	60000	65000	Baby & Kids
10	9	B009	Lampu LED	25000	30000	Electronics



Tugas 2 – Implementasi File Export Excel:

1. Silahkan implementasikan praktikum 2 pada project kalian masing-masing untuk semua menu

➤ Data user

Daftar User Home / User

Daftar Pengguna Import Pengguna Export User Tambah Data (Ajax)

Downloads

Data_User_2025-04-10_06-48-47.xlsx
[Open file](#)

	A	B	C	D	E
1	No	User ID	Username	Nama	Level
2	1	1	admin	Administrator	Administrator
3	2	2	manager	Manager	Manager
4	3	3	staff	Staff/Kasir	Staff/Kasir
5	4	5	customer-1	Pelanggan Pertama	Customer

< > Data User + :

➤ Data level

Daftar Level Home / Level

Daftar level yang terdaftar dalam sistem Import Level Export Level Tambah Data (Ajax)

Downloads

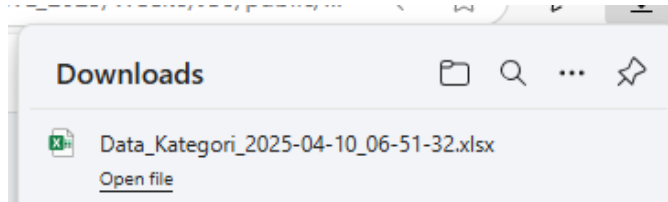
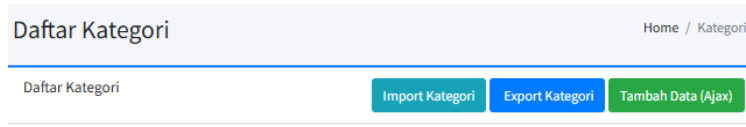
Data_Level_2025-04-10_06-21-36.xlsx
[Open file](#)

	A	B	C	D	E
1	No	ID Level	Kode Level	Nama Level	
2	1	1	ADM	Administrator	
3	2	2	MNG	Manager	
4	3	3	STF	Staff/Kasir	
5	4	4	CUS	Customer	

< > Data Level + :

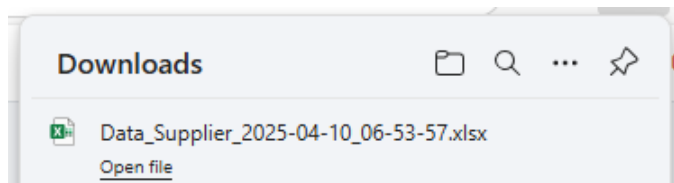
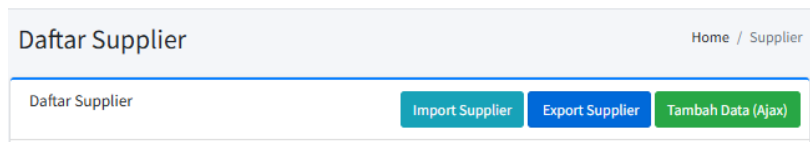


➤ Data kategori



	A	B	C	D
1	No	Kode Kategori	Nama Kategori	
2	1	FB	Food & Beverage	
3	2	BH	Beauty & Health	
4	3	HC	Home Care	
5	4	BK	Baby & Kids	

➤ Data supplier



	A	B	C	D	E
1	No	Supplier ID	Kode Supplier	Nama Supplier	Alamat Supplier
2	1		1 SP001	PT Sumber Jaya Abadi	Jl. Industri Raya No. 10, Jakarta E
3	2		2 SP002	PT Mitra Sejahtera	Jl. Pahlawan No. 25, Surabaya
4	3		3 SP003	PT Sentosa Makmur Bersama	Jl. Gatot Subroto No. 15, Bandung

2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
 - Menambahkan Tombol "Export (Menu)" di index.blade.php
Tombol ini ditambahkan di halaman utama tiap menu untuk memicu proses ekspor data ke file Excel.
 - Menambahkan Route untuk Export Excel
Setiap menu memiliki route khusus yang menghubungkan tombol export ke fungsi export_excel di controller.
 - Menambahkan Fungsi export_excel di Controller
Fungsi ini mengambil data dari database dan mengubahnya ke format Excel menggunakan library, lalu mengunduhnya ke pengguna.
3. Submit kode untuk impementasi prakktikum 2 pada repository github kalian.



C. 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
```

```
Microsoft Windows [Version 10.0.26100.3624]
(c) Microsoft Corporation. All rights reserved.

C:\laragon\www\PWL_2025\Week8\JS8>composer require barryvdh/laravel-dompdf
./composer.json has been updated
Running composer update barryvdh/laravel-dompdf
Loading composer repositories with package information
Updating dependencies
Lock file operations: 6 installs, 0 updates, 0 removals
- Locking barryvdh/laravel-dompdf (v3.1.1)
- Locking dompdf/dompdf (v3.1.0)
- Locking dompdf/php-font-lib (1.0.1)
- Locking dompdf/php-svg-lib (1.0.0)
- Locking symfony/dom-crawler (v6.4.0)
- Locking symfony/css-selector (v6.4.0)
```

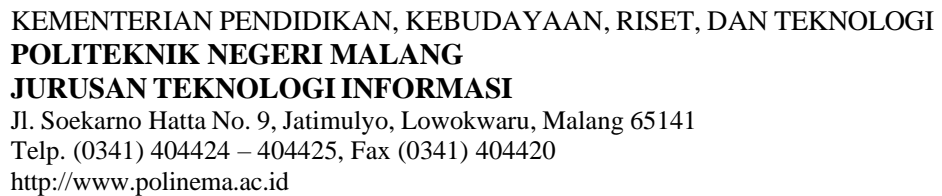
2. Setelah proses instalasi **dompdf** berhasil, selanjutnya Kita tambahkan kode berikut untuk menambahkan tombol export ke pdf pada **barang/index.blade.php**

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

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

```
// Export Barang with Pdf
Route::get(uri: 'export_pdf', action: [BarangController::class, 'export_pdf']); //export pdf
```

4. Selanjutnya, kita buat fungsi **export_pdf()** pada **BarangController.php**



5. Selanjutnya, kita buat view untuk dijadikan pdf dari layout html. File bisa kita buat dengan nama `barang/export_pdf.blade.php`

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.

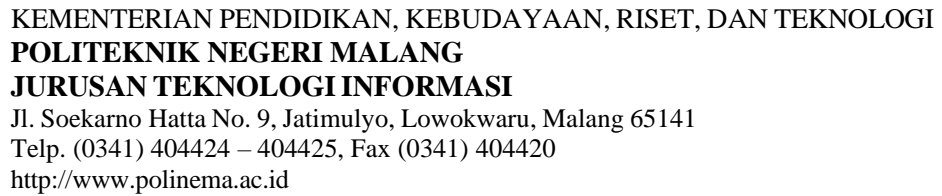
- ```
<html>
<head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
 <style>
 body{
 font-family: "Times New Roman", Times, serif;
 margin: 6px 20px 5px 20px;
 line-height: 15px;
 }
 table {
 width:100%;
 border-collapse: collapse;
 }
 td, th {
 padding: 4px 3px;
 }
 th{
 text-align: left;
 }
 .d-block{
 display: block;
 }
 img.image{
 width: auto;
 height: 80px;
 max-width: 150px;
 max-height: 150px;
 }
 .text-right {
 text-align: right;
 }
 </style>
</head>
</html>
```



```
.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"></td>
 <td width="85%">
 KEMENTERIAN
PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
 POLITEKNIK NEGERI
MALANG
 Jl. Soekarno-Hatta No. 9 Malang
65141
 Telepon (0341) 404424 Pes. 101-
105, 0341-404420, Fax. (0341) 404420
 Laman: www.polinema.ac.id
 </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>
 </tbody>
 </table>
```





```

1 @media (max-width: 480px) {
2 .html;
3 }
4
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
7 <style>
8 body {
9 font-family: "Times New Roman", Times, serif;
10 margin: 5px 20px 5px 20px;
11 line-height: 1.5px;
12 }
13
14 table {
15 width: 100%;
16 border-collapse: collapse;
17 }
18
19 td,
20 th {
21 padding: 5px 3px;
22 }
23
24 th {
25 text-align: left;
26 }
27
28 .d-block {
29 display: block;
30 }
31
32 img, image {
33 width: 100%;
34 height: 80px;
35 max-width: 100px;
36 max-height: 100px;
37 }
38
39 .text-right {
40 text-align: right;
41 }
42
43 .text-center {
44 text-align: center;
45 }
46
47 .p-1 {
48 padding: 5px 5px 5px 5px;
49 }
50
51 .font-10 {
52 font-size: 10pt;
53 }
54
55 .font-11 {
56 font-size: 10pt;
57 }
58
59 .font-12 {
60 font-size: 12pt;
61 }
62
63 .font-13 {
64 font-size: 10pt;
65 }
66
67 .border-bottom-header {
68 border-bottom: 1px solid;
69 }
70
71 .border-all,
72 .border-all td,
73 .border-all td {
74 border: 1px solid;
75 }
76 }
77 />
78
79 </style>
80 </head>
81
82 <body>
83 <table class="border-bottom-header">
84 <tr>
85 <td width="15%" class="text-center">
86
87 </td>
88 <td width="85%">
89
90 KEPENTERIAN PEMEDISIKAN, KESEHATAN, RISET, DAN TEKNOLOGI
91
92
93 POLITEKNIK NEGERI PALANG
94
95
96 Jl. Soekarno-Hatta No. 9 Malang 65144
97
98
99 Telpun (0341) 484424 Pst. 101-105, 0341-484420, Fax. (0341) 484420
100
101
102 Email: www.polinema.ac.id
103
104 </td>
105 </tr>
106 </table>
107
108 <div class="text-center">LAPORAN DATA BARANG</div>
109
110 <table class="border-all">
111 <thead>
112 <tr>
113 <th class="text-center">No</th>
114 <th class="text-center">Kode Barang</th>
115 <th class="text-center">Nama Barang</th>
116 <th class="text-right">Harga Beli</th>
117 <th class="text-right">Harga Jual</th>
118 <th class="text-right">Kategori</th>
119 </tr>
120 </thead>
121 <tbody>
122 <tr>
123 <td colspan="6">
124 <div>
125 <div>
126 <div>
127 <div>
128 <div>
129 <div>
130 <div>
131 <div>
132 <div>
133 <div>
134 <div>
135 <div>
136 <div>
137 <div>
138 <div>
139 <div>
140 <div>
141 <div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </td>
161 </tr>
162 </tbody>
163 </table>
164
165 </body>
166 </html>

```

- Data Barang

Daftar barang

Import Barang

Export Barang

Tambah Data (Ajax)

Export Barang

Sign in


localhost/localhost/pwl\_pos/

export\_pdf

localhost/PWL\_2025/Week8/JS8/public/barang/export\_pdf

1 of 1

Edit with Acrobat



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI MALANG

Jl. Soekarno-Hatta No. 9 Malang 65141

Telepon (0341) 404424 Pns. 101-105, 0341-404420, Fax. (0341) 404420

Laman: [www.polinema.ac.id](http://www.polinema.ac.id)

LAPORAN DATA BARANG

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	B001	Air Mineral	2.000	3.000	Food & Beverage
2	B002	Roti Tawar	10.000	12.000	Food & Beverage
3	B003	Sabun Mandi	5.000	7.000	Beauty & Health
4	B004	Shampoo	15.000	18.000	Beauty & Health
5	B005	Pembersih Lantai	10.000	13.000	Home Care
6	B006	Tisu	8.000	10.000	Home Care
7	B007	Popok Bayi	50.000	55.000	Baby & Kids
8	B008	Susu Formula	60.000	65.000	Baby & Kids
9	B009	Lampu LED	25.000	30.000	Electronics
10	B010	Charger HP	35.000	40.000	Electronics



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

1. Silahkan implementasikan export pdf pada project kalian masing-masing untuk semua menu

#### ➤ Data user

Daftar User Home / User

Daftar Pengguna Import Pengguna Export User Tambah Data (Ajax) Export User

Sign in localhost / localhost / pwl\_pos / - x export\_pdf

localhost/PWL\_2025/Week8/JS8/public/user/export\_pdf

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
POLITEKNIK NEGERI MALANG  
Jl. Soekarno-Hatta No. 9 Malang 65141  
Telepon (0341) 404424 Pes. 101-105, 0341-404420, Fax. (0341) 404420  
Laman: www.polinema.ac.id

**LAPORAN DATA USER**

No	ID Pengguna	Username	Nama	Level
1	1	admin	Administrator	Administrator
2	2	manager	Manager	Manager
3	3	staff	Staff/Kasir	Staff/Kasir
4	5	customer-1	Pelanggan Pertama	Customer
5	6	manager_dua	Manager 2	Manager
6	8	manager22	Manager Dua Dua	Manager
7	11	manager33	Manager Tiga Tiga	Manager
8	12	manager56	Manager55	Manager
9	16	manager new	karina	Manager

#### ➤ Data level

Daftar Level Home / Level

Daftar level yang terdaftar dalam sistem Import Level Export Level Tambah Data (Ajax) Export Level

Sign in localhost / loc x export\_pdf

localhost/PWL\_2025/Week8/JS8/public/level/e...

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
POLITEKNIK NEGERI MALANG  
Jl. Soekarno-Hatta No. 9 Malang 65141  
Telepon (0341) 404424 Pes. 101-105, 0341-404420, Fax. (0341) 404420  
Laman: www.polinema.ac.id

**LAPORAN DATA LEVEL**

No	Level Kode	Level Nama
1	ADM	Administrator
2	MNG	Manager
3	STF	Staff/Kasir
4	CUS	Customer
5	CEO	CEO



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>

➤ Data kategori

Daftar Kategori Home / Kategori

Daftar Kategori Import Kategori Export Kategori Tambah Data (Ajax) Export Kategori

Sign in localhost / loc: x export\_pdf x + - □ ×

localhost/PWL\_2025/Week8/JS8/public/katego... 1 of 1

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
POLITEKNIK NEGERI MALANG  
Jl. Soekarno-Hatta No. 9 Malang 65141  
Telepon (0341) 404424 Pes. 101-105, 0341-404420, Fax. (0341) 404420  
Laman: www.polinema.ac.id

**LAPORAN DATA KATEGORI**

No	Kode Kategori	Nama Kategori
1	FB	Food & Beverage
2	BH	Beauty & Health
3	HC	Home Care
4	BK	Baby & Kids
5	EL	Electronics
6	CML	Cemilan
7	MNR	Minuman Ringan
8	SPT	Sepatu

➤ Data supplier

Daftar Supplier Home / Supplier

Daftar Supplier Import Supplier Export Supplier Tambah Data (Ajax) Export Supplier

Sign in localhost / loc: x export\_pdf x + - □ ×

localhost/PWL\_2025/Week8/JS8/public/suppli... 1 of 1

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
POLITEKNIK NEGERI MALANG  
Jl. Soekarno-Hatta No. 9 Malang 65141  
Telepon (0341) 404424 Pes. 101-105, 0341-404420, Fax. (0341) 404420  
Laman: www.polinema.ac.id

**LAPORAN DATA SUPPLIER**

No	Kode Supplier	Nama Supplier	Alamat Supplier
1	SP001	PT Sumber Jaya Abadi	Jl. Industri Raya No. 10, Jakarta Barat
2	SP002	PT Mitra Sejahtera	Jl. Pahlawan No. 25, Surabaya
3	SP003	PT Sentosa Makmur Bersama	Il. Gatot Subroto No. 15, Bandung

2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan

- Fungsi export\_pdf di Controller  
Mengambil data dari database dan mengubahnya jadi file PDF.
- Tombol "Export PDF" di index.blade.php  
Untuk mengunduh laporan PDF langsung dari halaman utama.
- View export\_pdf.blade.php  
Template tampilan untuk isi file PDF (tabel, header, dll).
- Route Export PDF  
Menghubungkan tombol ke fungsi export\_pdf.

3. 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
2. Jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
  - Menambahkan Fitur Profil dan Ganti FotoFungsi profile di Controller: Menampilkan halaman profil dari user yang sedang login.  
Fungsi updatePhoto di Controller: Mengganti foto profil user:
  - Menghapus foto lama dari server.
  - Menyimpan foto baru ke folder yang ditentukan.
  - Update data di database.

```
457 public function profile(): Factory|mixed|RedirectResponse|View
458 {
459 $user = auth()->user();
460
461 if (!$user) {
462 return redirect(to: '/login')->with(key: 'error', value: 'Silahkan login terlebih dahulu');
463 }
464
465 $breadcrumb = (object) [
466 'title' => 'Profile User',
467 'list' => ['Home', 'Profile']
468];
469
470 $page = (object) [
471 'title' => 'Profil Pengguna'
472];
473
474 $activeMenu = 'profile';
475
476 return view(view: 'user.profile', data: compact('var_name', 'var_names', 'breadcrumb', 'page', 'activeMenu'));
477 }
478
479 //Update foto profile user
480 @reference: @csrf
481 public function updatePhoto(Request $request)
482 {
483 // Validasi file
484 $request->validate([
485 'profile_picture' => 'required|image|mimes:jpeg,png,jpg|max:2048',
486]);
487
488 try {
489 // auth user yang login
490 $user = auth()->user();
491
492 if (!$user) {
493 return redirect('/login')->with('error', 'Silahkan login terlebih dahulu');
494 }
495
496 // user id
497 $userId = $user->user_id;
498
499 $userModel = UserModel::find($userId);
500
501 if (!$userModel) {
502 return redirect('/login')->with('error', 'User tidak ditemukan');
503 }
504
505 // hapus jika sudah ada foto profile
506 if ($userModel->profile_picture && file_exists(storage_path('app/public/' . $userModel->profile_picture))) {
507 Storage::disk('public')->delete($userModel->profile_picture);
508 }
509
510 // update foto profile baru
511 $fileName = 'profile_' . $userId . '_' . time() . '.' . $request->profile_picture->extension();
512 $path = $request->profile_picture->storeAs('profiles', $fileName, 'public');
513
514 // update
515 UserModel::where('user_id', $userId)->update([
516 'profile_picture' => $path
517]);
518
519 return redirect()->back()->with('success', 'Foto profile berhasil diperbarui');
520 } catch (\Exception $e) {
521 return redirect()->back()->with('error', 'Gagal mengupload foto: ' . $e->getMessage());
522 }
523 }
524 }
```

- Tambah izin update di UserModel

```
protected $fillable = ['username', 'password', 'nama', 'level_id', 'profile_picture', 'created_at', 'updated_at'];
```



➤ Fungsi `getProfilePictureAttribute` di `UserModel`

```
53 // Menampilkan foto profil user secara otomatis.
54 // references: 0 overrides
55 public function getProfilePictureUrlAttribute(): string
56 {
57 return $this->profile_picture
58 ? asset(path: 'storage/profile_pictures/' . $this->profile_picture)
59 : asset(path: 'images/default-profile.png');
60 }
```

➤ Tombol & Tampilan Profil di `header.blade.php`

```
134 <!-- User profile dropdown -->
135 <li class="nav-item dropdown user-menu" style="position: relative;"
136 <a href="#" class="nav-link dropdown-toggle d-flex align-items-center" data-toggle="dropdown"
137 style="padding: 0.5rem 1rem;">
138
142 {{ auth()->user()->nama }}
143 <i class="fas fa-chevron-down ml-2"></i>
144
145 <ul class="dropdown-menu dropdown-menu-lg dropdown-menu-right">
146 <!-- User Image -->
147 <li class="user-header bg-primary">
148
152 <p>
153 {{ auth()->user()->nama }}
154 <small>{{ auth()->user()->level->level_nama ?? 'User' }}</small>
155 </p>
156
157 <!-- Menu Footer -->
158 <li class="user-footer">
159
160 <i class="fas fa-user mr-1"></i> Profile
161
162 <a href="#" class="btn btn-default btn-flat float-right"
163 onclick="event.preventDefault(); document.getElementById('logout-form').submit();" >
164 <i class="fas fa-sign-out-alt mr-1"></i> Logout
165
166 <form id="logout-form" action="{{ route(name: 'logout') }}" method="POST" style="display: none;">
167 @csrf
168 @form>
169 </form>
170
171
```

➤ Menambahkan button profile pada `sidebar.blade.php`

```
26 <!-- Data Pengguna -->
27 <li class="nav-header">Data Pengguna
28 <li class="nav-item">
29
30 <i class="nav-icon fas fa-user"></i>
31 <p>Profile</p>
32
33
34
```

➤ Menambahkan file baru pada `view/user/profile.blade.php` untuk membuat tampilan form pengisian

```
1 @extends('view.layouts.template')
2
3 @section('content')
4 <div class="card">
5 <div class="card-header">
6 <h3 class="card-title">Profile User</h3>
7 </div>
8 <div class="card-body">
9 <div class="row">
10 <!-- Kolom kiri: Foto profil dan form upload -->
11 <div class="col-md-4 text-center">
12 <!-- Menampilkan foto profil, jika tidak ada pakai default -->
13
16 <!-- Form upload foto -->
17 <form action="{{ route(name: 'user.updatePhoto') }}" method="POST" enctype="multipart/form-data">
18 @csrf
19 <div class="form-group">
20 <label for="profile_picture">Pilih Foto</label>
21 <input type="file" class="form-control" id="profile_picture" name="profile_picture"
22 accept="image/*">
23 </div>
24 @error('profile_picture')
25 <small class="text-danger">{{ $message }}</small>
26 @enderror
27 </div>
28 <div class="form-group">
29 <button type="submit" class="btn btn-primary mt-2">Upload Foto</button>
30 </div>
31 </div>
32 </div>
33 <!-- Kolom kanan: Data user -->
34 <div class="col-md-8">
35 <h4>Data User</h4>
36 <table class="table">
37 <tr>
38 <th width="30%">Username</th>
39 <td>{{ $user->username }}</td>
40 </tr>
41 <tr>
42 <th>Nama</th>
43 <td>{{ $user->nama }}</td>
44 </tr>
45 <tr>
46 <th>Level</th>
47 <td>{{ $user->level->level_nama ?? '' }}</td>
48 </tr>
49 </table>
```



- Buat migration baru untuk menambahkan kolom profile\_picture pada database pwl\_pos table m\_user

```
C:\laragon\www\PWL_2025\Week8\JS8>php artisan make:migration add_profile_picture_to_m_user_table --table=m_user
```

**INFO** Migration [C:\laragon\www\PWL\_2025\Week8\JS8\database\migrations\2025\_04\_10\_085408\_add\_profile\_picture\_to\_m\_user\_table.php] created successfully.

- Menambahkan kolom profile\_picture di tabel m\_user melalui migration untuk menyimpan foto profil, serta bisa dihapus saat rollback.

```
JS8 > database > migrations > 2025_04_10_085408_add_profile_picture_to_m_user_table.php ...
```

```
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 * Run the migrations.
10 */
11 public function up(): void
12 {
13 Schema::table('m_user', function (Blueprint $table) {
14 $table->string('profile_picture')->nullable()->after(column: 'level_id');
15 });
16 }
17
18 /**
19 * Reverse the migrations.
20 */
21 public function down(): void
22 {
23 Schema::table('m_user', function (Blueprint $table) {
24 $table->dropColumn('profile_picture');
25 });
26 }
27 };
```

- Menjalankan php artisan migrate

```
C:\laragon\www\PWL_2025\Week8\JS8>php artisan migrate
```

**INFO** Running migrations.

2025\_04\_10\_085408\_add\_profile\_picture\_to\_m\_user\_table ..... 734ms **DONE**

- Menjalankan php artisan storage:link untuk menyimpan foto yang diupload.

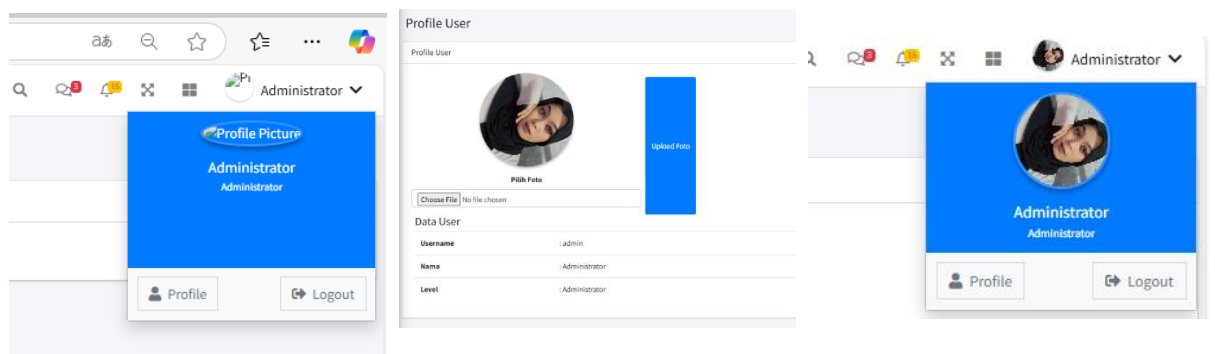
```
C:\laragon\www\PWL_2025\Week8\JS8>php artisan storage:link
```

**INFO** The [C:\laragon\www\PWL\_2025\Week8\JS8\public\storage] link has been connected to [C:\laragon\www\PWL\_2025\Week8\JS8\storage\app\public].

- Memenambahkan route pada web.php

```
28 Route::get(uri: '/profile', action: [UserController::class, 'profile']);
29 Route::post(uri: '/profile/update-photo', action: [UserController::class, 'updatePhoto'])->name(name: 'user.updatePhoto');
```

- Hasil



- Gambar akan tersimpan di public/storage/profiles & database

user_id	level_id	profile_picture	username	nama	password	created_at	updated_at
1	1	profiles/profile_1_1744270222.jpg	admin	Administrator	\$2y\$12\$T0at8tly0lvz0hg4FeRH5K47QNS0mzEoluzX...	NULL	2025-04-10 08:10:22

3. Submit kode untuk impementasi export pdf pada repository github kalian.

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