



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

#### 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
1	SBK	Sembako	2024-08-16 02:57:53	NULL
2	SNK	Makanan Ringan	2024-08-16 02:57:53	NULL
3	MND	Peralatan Mandi	2024-08-16 02:57:53	NULL
4	BAY	Keperluan Bayi	2024-08-16 02:57:53	NULL
5	MNM	Minuman	2024-08-16 02:57:53	NULL

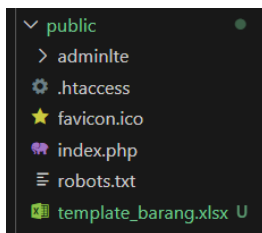
barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
1	1	SBK-001	Beras Cap Jago (5kg)	65000	68000	2024-08-19 20:07:06	NULL
2	1	SBK-002	Beras Bramo Cap Lele	80000	83000	2024-08-19 20:07:06	NULL
3	2	SNK-001	Happy Toss	10500	11000	2024-08-19 20:07:06	NULL
4	2	SNK-002	Oreo	7200	7800	2024-08-19 20:07:06	NULL
5	3	MND-001	Sabun Lifebouy	4250	5000	2024-08-19 20:07:06	NULL
6	3	MND-002	Pasta Gigi Pepsoden	6750	7500	2024-08-19 20:07:06	NULL
7	4	BAY-001	Susu SGM Coklat 900gr	92500	95000	2024-08-19 20:07:06	NULL
8	4	BAY-002	Popok Mamy Poko	56000	58000	2024-08-19 20:07:06	NULL
9	5	MNM-001	Aqua 600ml	3700	4500	2024-08-19 20:07:06	NULL
10	5	MNM-002	Le Mineral	3500	4000	2024-08-19 20:07:06	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	
7						

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.



- 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>
</section>
```



```
<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
```

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">
```



```
<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>
```

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





```
Route::get('/barang',[BarangController::class,'index']);
Route::post('/barang/list',[BarangController::class,'list']);
Route::get('/barang/create_ajax',[BarangController::class,'create_ajax']); // ajax form create
Route::post('/barang_ajax',[BarangController::class,'store_ajax']); // ajax store
Route::get('/barang/{id}/edit_ajax',[BarangController::class,'edit_ajax']); // ajax form edit
Route::put('/barang/{id}/update_ajax',[BarangController::class,'update_ajax']); // ajax update
Route::get('/barang/{id}/delete_ajax',[BarangController::class,'confirm_ajax']); // ajax form confirm
Route::delete('/barang/{id}/delete_ajax',[BarangController::class,'delete_ajax']); // ajax delete
Route::get('/barang/import',[BarangController::class,'import']); // ajax form upload excel
Route::post('/barang/import_ajax',[BarangController::class,'import_ajax']); // ajax import excel
```

6. 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
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\laragon\www\PWL_2025\Week8\PWL_POS> composer require phpoffice/phpspreadsheet
```

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
        ]);
    }
}
```





```
/*$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('/');
```

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



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

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori	Aksi
11	A003	Rendang	20.000	25.000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
12	B003	Milkshake	35.000	40.000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
13	C003	Majalah	75.000	100.000	Buku	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
14	D003	Sepatu Basket	700.000	850.000	Olahraga	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
15	E003	Lemari Kayu	750.000	900.000	Furniture	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 11 to 15 of 15 entries

Previous 1 2 Next

## 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. Submit kode untuk impementasi prakktikum 1 pada repository github kalian.

User Controller :

```
public function import()
{
    return view('user.import');
}

public function import_ajax(Request $request)
{
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'file_user' => ['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_user');

        $reader = IOFactory::createReader('xlsx');
        $reader->setReadDataOnly(true);
        $spreadsheet = $reader->load($file->getRealPath());
        $sheet = $spreadsheet->getActiveSheet();

        $data = $sheet->toArray(null, false, true, true);
        $insert = [];
```

```
$insert = [];
if (count($data) > 1) {
    foreach ($data as $baris => $value) {
        if ($baris > 1) {
            $insert[] = [
                'level_id' => $value['A'],
                'username' => $value['B'],
                'nama' => $value['C'],
                'password' => bcrypt($value['D']),
                'created_at' => now(),
            ];
        }
    }

    if (count($insert) > 0) {
        UserModel::insertOrIgnore($insert);
    }

    return response()->json([
        'status' => true,
        'message' => 'Data user berhasil diimport'
    ]);
} else {
    return response()->json([
        'status' => false,
        'message' => 'Tidak ada data yang diimport'
    ]);
}

return redirect('/');
```



## Import.blade :

```
Week8 > PWL_POS > resources > views > user > import.blade.php > @script
1 <form action="{{ url('/user/import_ajax') }}" method="POST" id="form-import" enctype="multipart/form-data">
2 <div id="modal-master" class="modal-dialog modal-lg" role="document">
3 <div class="modal-content">
4 <div class="modal-header">
5 <div class="modal-title">Import Data User</div>
6 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
7 <span aria-hidden="true"></span></button>
8 </div>
9 <div class="modal-body">
10 <div class="form-group">
11 <label>Download Template</label>
12 <a href="{{ asset('template_user.xlsx') }}" class="btn btn-info btn-sm" download>
13 <i class="fa fa-file-excel"></i> Download
14 </a>
15 <small id="error-kategori_id" class="error-text form-text text-danger"></small>
16 </div>
17 <div class="form-group">
18 <input type="file" name="file_user" id="file_user" class="form-control" required>
19 <small id="error-file_user" class="error-text form-text text-danger"></small>
20 </div>
21 </div>
22 <div class="modal-footer">
23 <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
24 <button type="submit" class="btn btn-primary">Upload</button>
25 </div>
26 </div>
27 </form>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 <script>
34 $(document).ready(function() {
35     $('#form-import').validate({
36         rules: {
37             file_user: {
38                 required: true,
39                 extension: "xlsx"
40             },
41             submitHandler: function(form) {
42                 var formData = new FormData(form);
43                 $.ajax({
44                     url: form.action,
45                     type: form.method,
46                     data: formData,
47                     processData: false,
48                     contentType: false,
49                     success: function(response) {
50                         if (response.status) {
51                             $('#myModal').modal('hide');
52                             Swal.fire({
53                                 icon: 'success',
54                                 title: 'Berhasil',
55                                 text: response.message
56                             });
57                             tableUser.ajax.reload();
58                         } else {
59                             $('#error-text').text('');
60                             $.each(response.msgfield, function(prefix, val) {
61                                 $('#error-' + prefix).text(val[0]);
62                             });
63                             Swal.fire({
64                                 icon: 'error',
65                                 title: 'Terjadi Kesalahan',
66                                 text: response.message
67                             });
68                         }
69                     }
70                 });
71             }
72         }
73     });
74     error: function(xhr, status, error) {
75         Swal.fire({
76             icon: 'error',
77             title: 'Error Ajax',
78             text: 'Terjadi kesalahan: ' + xhr.status + ' - ' + error
79         });
80         console.log('Error: ', xhr.responseText);
81     }
82     return false;
83     errorElement: 'span',
84     errorPlacement: function(error, element) {
85         error.addClass('invalid-feedback');
86         element.closest('.form-group').append(error);
87     },
88     highlight: function(element) {
89         $(element).addClass('is-invalid');
90     },
91     unhighlight: function(element) {
92         $(element).removeClass('is-invalid');
93     }
94 });
95 </script>
96 </div>
```

## Routes :

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

## Import:

level_id	username	nama	password
1	admin3	Toro	123456

Import Data User

Download Template [Download](#)

Pilih File

template\_user.xlsx



**Berhasil**

Data user berhasil diimport

6	admin3	Toro	Administrator	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
---	--------	------	---------------	------------------------	----------------------	-----------------------



## Level Controller:

```
247 public function import()
248 {
249     return view('level.import');
250 }
251
252 public function import_ajax(Request $request)
253 {
254     if ($request->ajax() || $request->wantsJson()) {
255         $rules = [
256             'file_level' => ['required', 'mimes:xlsx', 'max:1024']
257         ];
258
259         $validator = Validator::make($request->all(), $rules);
260         if ($validator->fails()) {
261             return response()->json([
262                 'status' => false,
263                 'message' => 'Validasi gagal',
264                 'msgfield' => $validator->errors()
265             ]);
266         }
267
268         $file = $request->file('file_level');
269
270         $reader = IOFactory::createReader('Xlsx');
271         $reader->setReadDataOnly(true);
272         $spreadsheet = $reader->load($file->getRealPath());
273         $sheet = $spreadsheet->getActiveSheet();
274         $data = $sheet->toArray(null, false, true, true);
275
276         $insert = [];
277         if (count($data) > 1) {
278             foreach ($data as $baris => $value) {
279                 if ($baris > 1) {
280                     $insert[] = [
281                         'level_kode' => $value['A'],
282                         'level_nama' => $value['B'],
283                         'created_at' => now(),
284                     ];
285                 }
286             }
287
288             if (count($insert) > 0) {
289                 LevelModel::insertOrIgnore($insert);
290             }
291
292             return response()->json([
293                 'status' => true,
294                 'message' => 'Data level berhasil diimport'
295             ]);
296         } else {
297             return response()->json([
298                 'status' => false,
299                 'message' => 'Tidak ada data yang diimport'
300             ]);
301         }
302     }
303     return redirect('/');
304 }
```

## Import Blade:

```
1 <form action="{{ url('/level/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">Import Data Level</h5>
7 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
8 <span aria-hidden="true"></span></button>
9 </div>
10 <div class="modal-body">
11 <div class="form-group">
12 <label>Download Template</label>
13 <a href="{{ asset('template_level.xlsx') }}" class="btn btn-info btn-sm">Download</a>
14 <div class="fa fa-file-excel"></div> Download
15 </div>
16 <div class="form-group">
17 <label>Pilih File</label>
18 <input type="file" name="file_level" id="file_level" class="form-control" required>
19 <small id="error-file_level" class="error-text form-text text-danger"></small>
20 </div>
21 <div class="modal-footer">
22 <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
23 <button type="submit" class="btn btn-primary">Upload</button>
24 </div>
25 </div>
26 </form>
27
28 <script>
29 $(document).ready(function() {
30     $('#form-import').validate({
31         rules: {
32             file_level: {
33                 required: true,
34             },
35         },
36         error: function(xhr, status, error) {
37             Swal.fire({
38                 icon: 'error',
39                 title: 'Error AJAX',
40                 text: 'Terjadi kesalahan: ' + xhr.status + ' - ' + error
41             });
42             console.log('Error:', xhr.responseText);
43         },
44         errorElement: 'span',
45         errorPlacement: function(error, element) {
46             error.addClass('invalid-feedback');
47             element.closest('.form-group').append(error);
48         },
49         highlight: function(element) {
50             $(element).addClass('is-invalid');
51         },
52         unhighlight: function(element) {
53             $(element).removeClass('is-invalid');
54         },
55     });
56 });
57 </script>
```

```
37     required: true,
38     extension: "xlsx"
39 },
40 },
41 submitHandler: function(form) {
42     var formData = new FormData(form);
43
44     $.ajax({
45         url: form.action,
46         type: form.method,
47         data: formData,
48         processData: false,
49         contentType: false,
50         success: function(response) {
51             if (response.status) {
52                 $('#myModal').modal('hide');
53                 Swal.fire({
54                     icon: 'success',
55                     title: 'Berhasil',
56                     text: response.message
57                 });
58                 tablelevel.ajax.reload();
59             } else {
60                 $('#error-text').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 }
```

## Routes:

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



## Import :

level_kode	level_nama
AGT	Anggota

## Output:

Import Data Level

Download Template [Download](#)

Pilih File

[Choose File](#) No file chosen

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

### Berhasil

Data level berhasil diimport

[OK](#)

5	AGT	Anggota	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
---	-----	---------	---

## Kategori Controller :

```
public function import()
{
    return view('kategori.import');
}

public function import_ajax(Request $request)
{
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'file_kategori' => ['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_kategori');

        $reader = IOFactory::createReader('xlsx');
        $reader->setReadDataOnly(true);
        $spreadsheet = $reader->load($file->getRealPath());
        $sheet = $spreadsheet->getActiveSheet();
        $data = $sheet->toArray(null, false, true, true);

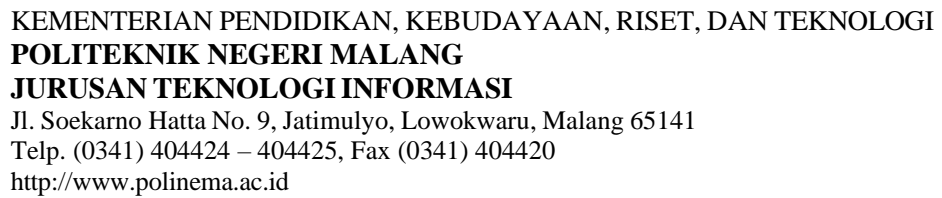
        $insert = [];
        if (count($data) > 1) {
            foreach ($data as $baris => $value) {
                if ($baris > 1) {
                    $insert[] = [
                        'kategori_kode' => $value['A'],
                        'kategori_nama' => $value['B'],
                        'created_at' => now(),
                    ];
                }
            }

            if (count($insert) > 0) {
                KategoriModel::insertOrIgnore($insert);
            }

            return response()->json([
                'status' => true,
                'message' => 'Data kategori berhasil diimport'
            ]);
        } else {
            return response()->json([
                'status' => false,
                'message' => 'Tidak ada data yang diimport'
            ]);
        }
    }

    return redirect('/');
}
```



[illegible]

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

kategori_kode	kategori_nama
MSC	Music
ELC	Elektronik

Import Data Kategori

Download Template

Download

Pilih File

Choose File

No file chosen

Batal

Upload

Berhasil

Data kategori berhasil diimport

OK

6	MSC	Music	<div><div>Detail</div><div>Edit</div><div>Hapus</div></div>
7	ELC	Elektronik	<div><div>Detail</div><div>Edit</div><div>Hapus</div></div>



## Supplier Controller:

```
public function import()
{
    return view('supplier.import');
}

public function import_ajax(Request $request)
{
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'file_supplier' => ['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_supplier');

        $reader = IOFactory::createReader('Xlsx');
        $reader->setReadDataOnly(true);
        $spreadsheet = $reader->load($file->getRealPath());
        $sheet = $spreadsheet->getActiveSheet();
        $data = $sheet->toArray(null, false, true, true);

        $insert = [];
        if (count($data) > 1) {
            foreach ($data as $baris => $value) {
                if ($baris > 1) {
                    $insert[] = [
                        'supplier_kode' => $value['A'],
                        'supplier_nama' => $value['B'],
                        'created_at' => now(),
                    ];
                }
            }
        }

        if (count($insert) > 0) {
            SupplierModel::insertOrIgnore($insert);
        }

        return response()->json([
            'status' => true,
            'message' => 'Data level berhasil diimport'
        ]);
    } else {
        return response()->json([
            'status' => false,
            'message' => 'Tidak ada data yang diimport'
        ]);
    }
}

return redirect('/');
```

## Import.blade:

```
Week8 > PWL_POS > resources > views > supplier > importblade.php > @script
1 <form action="{{ url('/supplier/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 <div class="modal-title">Import Data Supplier</div>
7 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
8 <span aria-hidden="true"></span>
9 </button>
10 </div>
11 <div class="modal-body">
12 <div class="form-group">
13 <label>Download Template</label>
14 <a href="{{ asset('template_supplier.xlsx') }}" class="btn btn-info btn-sm" download>
15 <i class="fa fa-file-excel"></i> Download
16 </a>
17 </div>
18 <div class="form-group">
19 <label>Pilih File</label>
20 <input type="file" name="file_supplier" id="file_supplier" class="form-control" required>
21 <small id="error-file_supplier" 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_supplier: {
37                 required: true,
38                 extension: "xlsx"
39             }
40         },
41         submitHandler: function(form) {
42             var formData = new FormData(form);
43
44             $.ajax({
45                 url: form.action,
46                 type: form.method,
47                 data: formData,
48                 processData: false,
49                 contentType: false,
50                 success: function(response) {
51                     if (response.status) {
52                         $('#myModal').modal('hide');
53                         Swal.fire({
54                             icon: 'success',
55                             title: 'Berhasil',
56                             text: response.message
57                         });
58                         tableSupplier.ajax.reload();
59                     } else {
60                         $('#error-text').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                 error: function(xhr, status, error) {
72                     Swal.fire({
73                         icon: 'error',
74                         title: 'Gagal',
75                         text: error
76                     });
77                 }
78             });
79         }
80     });
81 }
82 </script>
```



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>

```
73         icon: 'error',
74         title: 'Error AJAX',
75         text: 'Terjadi kesalahan: ' + xhr.status + ' - ' + error
76     });
77     console.log('Error:', xhr.responseText);
78     }
79     });
80     return false;
81 },
82 errorElement: 'span',
83 errorPlacement: function(error, element) {
84     error.addClass('invalid-feedback');
85     element.closest('.form-group').append(error);
86 },
87 highlight: function(element) {
88     $(element).addClass('is-invalid');
89 },
90 unhighlight: function(element) {
91     $(element).removeClass('is-invalid');
92 }
93 });
94 });
95 </script>
```

Routes :

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

Import :

supplier_kode	supplier_nama
321	Putera

Output:

Import Data Supplier

Download Template [Download](#)

Pilih File

Choose File

No file chosen

Batal

Upload

✓

**Berhasil**

Data level berhasil diimport

OK

2	321	Putera	<div><div>Detail</div><div>Edit</div><div>Hapus</div></div>
---	-----	--------	---

## 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 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 dengan

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

```
<button onclick="modalAction('{{ url('/barang/import') }}')" class="btn btn-info">Import Barang</button>
<a href="{{ url('/barang/export_excel') }}" class="btn btn-primary"><i class="fa fa-file-excel"></i> Export Barang</a>|
<button onclick="modalAction('{{ url('/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
```

```
Route::get('/export_excel',[BarangController::class,'export_excel']); // export excel
```

3. Selanjutnya kita tambahkan fungsi `export_excel()` pada file `BarangController.php`
4. 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 public function export_excel()
2 {
3     // Ambil data barang yang akan di export
4     $barang = BarangModel::select('kategori_id','barang_kode','barang_nama','harga_beli','harga_jual')
5         ->orderBy('kategori_id')
6         ->with('kategori')
7         ->get();
8 }
```

5. 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
```

```
// Load library excel
$spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
$sheet = $spreadsheet->getActiveSheet(); // Ambil sheet yang aktif

$sheet->setCellValue('A1','No');
$sheet->setCellValue('B1','Kode Barang');
$sheet->setCellValue('C1','Nama Barang');
$sheet->setCellValue('D1','Harga Beli');
$sheet->setCellValue('E1','Harga Jual');
$sheet->setCellValue('F1','Kategori');

$sheet->getStyle('A1:F1')->getFont()->setBold(true); // Bold header
```



6. 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 }
```

```
$no = 1; // nomor data dimulai dari 1
$baris = 2; // baris data dimulai dari baris ke 2
foreach ($barang as $key => $value) {
    $sheet->setCellValue('A'.$baris, $no);
    $sheet->setCellValue('B'.$baris, $value->barang_kode);
    $sheet->setCellValue('C'.$baris, $value->barang_nama);
    $sheet->setCellValue('D'.$baris, $value->harga_beli);
    $sheet->setCellValue('E'.$baris, $value->harga_jual);
    $sheet->setCellValue('F'.$baris, $value->kategori->kategori_nama); // ambil nama kategori
    $baris++;
    $no++;
}
```

7. 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 }
```

```
foreach(range('A','F') as $columnID){
    $sheet->getColumnDimension($columnID)->setAutoSize(true); // Set autosize untuk kolom
}
```





8. 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
```

```
$sheet->setTitle('Data Barang'); // set title sheet

$writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
$filename = 'Data Barang '.date('Y-m-d H:i:s').'.xlsx';

header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
header('Content-Disposition: attachment;filename="'.$filename.'"');
header('Cache-Control: max-age=0');
header('Cache-Control: max-age=1');
header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: cache, must-revalidate');
header('Pragma: public');

$writer->save('php://output');
exit;
```

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

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	SBK-001	Beras Cap Jago (5kg)	65000	68000	Sembako
2	SBK-002	Beras Bramo Cap Lele	80000	83000	Sembako
3	SBK-003	Telur Omega (10 butir)	22000	25000	Sembako
4	SNK-001	Happy Toss	10500	11000	Makanan Ringan
5	SNK-002	Oreo	7200	7800	Makanan Ringan
6	SNK-003	Sari Roti	11500	12500	Makanan Ringan
7	MND-001	Sabun Lifebouy	4250	5000	Peralatan Mandi
8	MND-002	Pasta Gigi Pepsoden	6750	7500	Peralatan Mandi
9	MND-003	Shampo Pantene	17500	18500	Peralatan Mandi
10	BAY-001	Susu SGM Coklat 900gr	92500	95000	Keperluan Bayi
11	BAY-002	Popok Mamy Poko	56000	58000	Keperluan Bayi
12	BAY-003	Baju Bayi 2th	89000	92500	Keperluan Bayi
13	MNM-001	Aqua 600ml	3700	4500	Minuman
14	MNM-002	Le Mineral	3500	4000	Minuman
15	MNM-003	Cleo 600ml	3750	4300	Minuman





Data Barang 2025-04-09 14_28_25 [Protected View] - Excel									
PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.									
A1 No Kode Barang Nama Barang Harga Beli Harga Jual Kategori									
1	A001	Roti Gandum	15000	20000	Makanan				
2	A002	Nasi Goreng Instan	20000	25000	Makanan				
3	A003	Rendang	20000	25000	Makanan				
4	B001	Kopi Arabika	50000	75000	Minuman				
5	B002	Teh Hijau	40000	50000	Minuman				
6	B003	Milkshake	35000	40000	Minuman				
7	C001	Novel Fiksi	75000	100000	Buku				
8	C002	Buku Sejarah	55000	150000	Buku				
9	C003	Majalah	75000	100000	Buku				
10	D001	Bola Basket	200000	250000	Olahraga				
11	D002	Sepatu Lari	500000	650000	Olahraga				
12	D003	Sepatu Basket	700000	850000	Olahraga				
13	E001	Meja Kayu	700000	850000	Furniture				
14	E002	Kursi Kayu	750000	900000	Furniture				
15	E003	Lemari Kayu	750000	900000	Furniture				

## 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. Submit kode untuk impementasi prakktikum 2 pada repository github kalian.

### User Controller :

```
public function export_excel()
{
    $user = User::select('level_id', 'username', 'nama', 'password')
        ->orderBy('level_id')
        ->with('level')
        ->get();

    $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
    $sheet = $spreadsheet->getActiveSheet();

    $sheet->setCellValue('A1', 'No');
    $sheet->setCellValue('B1', 'Username');
    $sheet->setCellValue('C1', 'Nama');
    $sheet->setCellValue('D1', 'Password');
    $sheet->setCellValue('E1', 'Level Pengguna');

    $sheet->setStyle('A1:E1')->getFont()->setBold(true);

    $no = 1;
    $baris = 2;
    foreach ($user as $key => $value) {
        $sheet->setCellValue('A' . $baris, $no);
        $sheet->setCellValue('B' . $baris, $value->username);
        $sheet->setCellValue('C' . $baris, $value->nama);
        $sheet->setCellValue('D' . $baris, $value->password);
        $sheet->setCellValue('E' . $baris, $value->level->level_nama);
        $baris++;
        $no++;
    }

    foreach (range('A', 'E') as $columnID) {
        $sheet->getColumnDimension($columnID)->setAutoSize(true);
    }
}
```

```
foreach (range('A', 'E') as $columnID) {
    $sheet->getColumnDimension($columnID)->setAutoSize(true);
}

$sheet->setTitle('Data User');

$writer = \IOFactory::createWriter($spreadsheet, 'Xlsx');
$filename = 'Data User_' . date('Y-m-d H:i:s') . '.xlsx';

header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
header('Content-Disposition: attachment; filename="' . $filename . '"');
header('Cache-Control: max-age=0');
header('Cache-Control: max-age=1');
header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: cache, must-revalidate');
header('Pragma: public');

$writer->save('php://output');
exit;
```

### Export:

Data User 2025-04-09 14_32_13 - Excel									
Putra Bhag									
F4									
A	B	C	D	E					
1	No	Username	Nama	Password	Level Pengguna				
2	1	admin	Administrator	\$2y\$12\$yAmNEIV8tD6Fk3rzcCSC.dwfpXlRn3ckP6u6/ZhPp9AwLyHhNa	Administrator				
3	2	admin2	Bhagas	\$2y\$12\$ix9k.HQnRr7FMv82y6s2uHRYRjMzKC21VCGRVGH7pe8MDvK2cpK	Administrator				
4	3	admin3	Toro	\$2y\$12\$R6PBwWu79NadNanR77KAu012Y7XFOHbA67yWXTbbaRlRBUzr6ym	Administrator				
5	4	manager	Manager	\$2y\$12\$0iA2v3Ytnkjh7icCnkfNNOqNpl68UQEfeFPLoJrcedborvhl4xj.	Manager				
6	5	manager2	Putera	\$2y\$12\$zsbMzK7py38sWYFDgnSKu0lfZedqH6XwRHERs7MEWjehHscff.sMG	Manager				
7	6	bhagas	Staff/Kasir	\$2y\$12\$/hqU2ZgkQe2/vTVdvzFN2un9OXIBYHugLyh9rpK8RIKayd1IMsDe	Staff/Kasir				



## Level Controller:

```
public function export_excel()
{
    $level = LevelModel::select('level_kode', 'level_nama')
        ->orderBy('level_kode')
        ->get();

    $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
    $sheet = $spreadsheet->getActiveSheet();

    $sheet->setCellValue('A1', 'No');
    $sheet->setCellValue('B1', 'Code Level');
    $sheet->setCellValue('C1', 'Nama Level');

    $sheet->getStyle('A1:C1')->getFont()->setBold(true);

    $no = 1;
    $baris = 2;
    foreach ($level as $key => $value) {
        $sheet->setCellValue('A.' . $baris, $no);
        $sheet->setCellValue('B.' . $baris, $value->level_kode);
        $sheet->setCellValue('C.' . $baris, $value->level_nama);
        $baris++;
        $no++;
    }

    foreach (range('A', 'C') as $columnID) {
        $sheet->getColumnDimension($columnID)->setAutoSize(true);
    }

    $sheet->setTitle('Data Level');

    $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
    $filename = 'Data Level ' . date('Y-m-d H:i:s') . '.xlsx';

    header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
```

```
140 header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
141 header('Content-Disposition: attachment;filename="'. $filename . '");
142 header('Cache-Control: max-age=0');
143 header('Cache-Control: max-age=1');
144 header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
145 header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
146 header('Cache-Control: cache, must-revalidate');
147 header('Pragma: public');
148
149 $writer->save('php://output');
150 exit;
151
```

## Export :

No	Code Level	Nama Level
1	ADM	Administrator
2	AGT	Anggota
3	KPL	Kepala
4	MNG	Manager
5	STF	Staff/Kasir

## Kategori Controller:

```
public function export_excel()
{
    $kategori = KategoriModel::select('kategori_kode', 'kategori_nama')
        ->orderBy('kategori_kode')
        ->get();

    $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
    $sheet = $spreadsheet->getActiveSheet();

    $sheet->setCellValue('A1', 'No');
    $sheet->setCellValue('B1', 'Code Kategori');
    $sheet->setCellValue('C1', 'Nama Kategori');

    $sheet->getStyle('A1:C1')->getFont()->setBold(true);

    $no = 1;
    $baris = 2;
    foreach ($kategori as $key => $value) {
        $sheet->setCellValue('A.' . $baris, $no);
        $sheet->setCellValue('B.' . $baris, $value->kategori_kode);
        $sheet->setCellValue('C.' . $baris, $value->kategori_nama);
        $baris++;
        $no++;
    }

    foreach (range('A', 'C') as $columnID) {
        $sheet->getColumnDimension($columnID)->setAutoSize(true);
    }

    $sheet->setTitle('Data Kategori');

    $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
    $filename = 'Data Kategori ' . date('Y-m-d H:i:s') . '.xlsx';

    header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
    header('Content-Disposition: attachment;filename="'. $filename . '");
```

```
334 header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
335 header('Content-Disposition: attachment;filename="'. $filename . '");
336 header('Cache-Control: max-age=0');
337 header('Cache-Control: max-age=1');
338 header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
339 header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
340 header('Cache-Control: cache, must-revalidate');
341 header('Pragma: public');
342
343 $writer->save('php://output');
344 exit;
345
```



## Export:

The screenshot shows an Excel spreadsheet with the following data:

No	Kode Kategori	Nama Kategori
1	BEV	Minuman
2	BOOK	Buku
3	ELC	Elektronik
4	FOOD	Makanan
5	FURN	Furniture
6	MSC	Music
7	SPORT	Olahraga

## Supplier Controller:

```
public function export_excel()
{
    $supplier = SupplierModel::select('supplier_kode', 'supplier_nama')
        >orderBy('supplier_kode')
        >get();

    $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
    $sheet = $spreadsheet->getActiveSheet();

    $sheet->setCellValue('A1', 'No');
    $sheet->setCellValue('B1', 'Kode supplier');
    $sheet->setCellValue('C1', 'Nama supplier');

    $sheet->getStyle('A1:C1')->getFont()->setBold(true);

    $no = 1;
    $baris = 2;
    foreach ($supplier as $key => $value) {
        $sheet->setCellValue('A', $baris, $no);
        $sheet->setCellValue('B', $baris, $value->supplier_kode);
        $sheet->setCellValue('C', $baris, $value->supplier_nama);
        $baris++;
        $no++;
    }

    foreach (range('A', 'C') as $columnID) {
        $sheet->getColumnDimension($columnID)->setAutoSize(true);
    }

    $sheet->setTitle('Data supplier');

    $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
    $filename = 'Data supplier ' . date('Y-m-d H:i:s') . '.xlsx';
    header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
    header('Content-Disposition: attachment; filename="'.$filename.'");
```

```
header('Cache-Control: max-age=1');
header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
header('Cache-Control: cache, must-revalidate');
header('Pragma: public');

$writer->save('php://output');
exit;
}
```

## Export:

The screenshot shows an Excel spreadsheet with the following data:

No	Kode supplier	Nama supplier
1	123	Bhagas
2	321	Putera



### 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
```

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

```
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
Route::get('/barang/export_pdf',[BarangController::class,'export_pdf']); // export pdf
```

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



```
public function export_pdf()
{
    $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
        ->orderBy('kategori_id')
        ->orderBy('barang_kode')
        ->with('kategori')
        ->get();

    // use Barryvdh\DomPDF\Facade\Pdf;
    $pdf = Pdf::loadView('barang.export_pdf', ['barang' => $barang]);
    $pdf->setPaper('a4', 'portrait'); // set ukuran kertas dan orientasi
    $pdf->setOption("isRemoteEnabled", true); // set true jika ada gambar dari url
    $pdf->render();

    return $pdf->stream('Data Barang '.date('Y-m-d H:i:s').'.pdf');
}
```

```
public function export_pdf(){
    $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
        ->orderBy('kategori_id')
        ->orderBy('barang_kode')
        ->with('kategori')
        ->get();

    $pdf = Pdf::loadView('barang.export_pdf', ['barang' => $barang]);
    $pdf->setPaper('a4', 'portrait');
    $pdf->setOption("isRemoteEnabled", true);
    $pdf->render();

    return $pdf->stream('Data Barang '.date('Y-m-d H:i:s').'.pdf');
}
```

5. 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 `dmpdf` tidak support semua fitur styling yang ada di framework CSS.

6. Selanjutnya kita buat view untuk me-generate html untuk tampilan pdf yang akan kita sajikan. View berada di `barang/export_pdf.blade.php`



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>

```
<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">
                <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</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>
            </tbody>
        </table>
```





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>

```
@endforeach
</tbody>
</table>
</body>
</html>
```

7. Selanjutnya, Kita coba untuk melakukan proses download export pdf. Amati dan pelajari...!!!

Data Barang

Daftar barang

[Import Barang](#) [Export Barang](#) [Export Barang PDF](#) [Tambah Data \(Ajax\)](#)

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

Logo Polinema

LAPORAN DATA BARANG

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	A001	Roti Gandum	15.000	20.000	Makanan
2	A002	Nasi Goreng Instan	20.000	25.000	Makanan
3	A003	Rendang	20.000	25.000	Makanan
4	B001	Kopi Arabika	50.000	75.000	Minuman
5	B002	Teh Hijau	40.000	50.000	Minuman
6	B003	Milkshake	35.000	40.000	Minuman
7	C001	Novel Fiksi	75.000	100.000	Buku
8	C002	Buku Sejarah	55.000	150.000	Buku
9	C003	Majalah	75.000	100.000	Buku
10	D001	Bola Basket	200.000	250.000	Olahraga
11	D002	Sepatu Lari	500.000	650.000	Olahraga
12	D003	Sepatu Basket	700.000	850.000	Olahraga
13	E001	Meja Kayu	700.000	850.000	Furniture
14	E002	Kursi Kayu	750.000	900.000	Furniture
15	E003	Lemari Kayu	750.000	900.000	Furniture

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

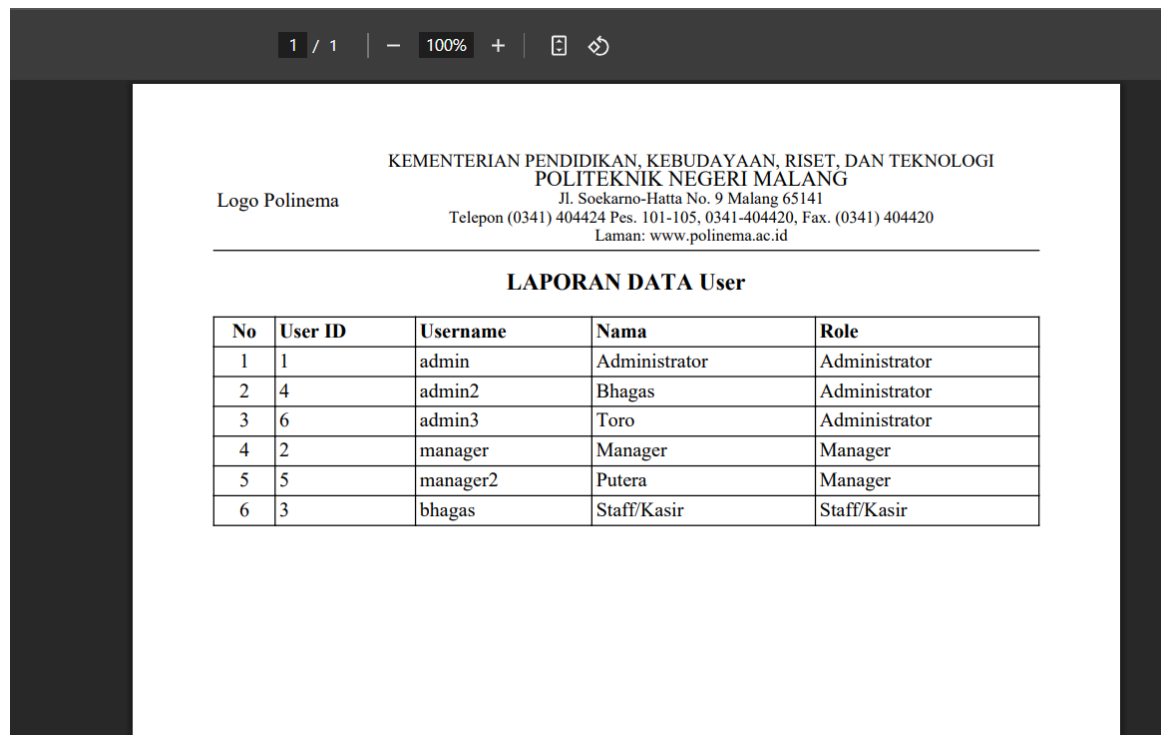
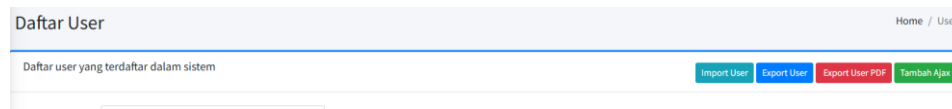


### User Controller:

```
public function export_pdf(){
    $user = UserModel::select('level_id', 'user_id', 'username', 'nama')
        ->orderBy('level_id')
        ->orderBy('user_id')
        ->with('level')
        ->get();

    $pdf = Pdf::loadView('user.export_pdf', ['user' => $user]);
    $pdf->setPaper('a4', 'portrait');
    $pdf->setOption("isRemoteEnable", true);
    $pdf->render();

    return $pdf->stream('Data User'.date('Y-m-d H:i:s').'.pdf');
```



### Level Controller:

```
public function export_pdf(){
    $level = LevelModel::select('level_id', 'level_kode', 'level_nama')
        ->orderBy('level_id')
        ->get();

    $pdf = Pdf::loadView('level.export_pdf', ['level' => $level]);
    $pdf->setPaper('a4', 'portrait');
    $pdf->setOption("isRemoteEnable", true);
    $pdf->render();

    return $pdf->stream('Data Level'.date('Y-m-d H:i:s').'.pdf');
```



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>

## Daftar Level

Home / Level

Daftar Level

Import Level

Export Level PDF

Export Level

Tambah Ajax

1 / 1 | - 100% + | [ ] [ ]

Logo Polinema

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

### LAPORAN DATA LEVEL

No	Level ID	Level Kode	Level Nama
1	1	ADM	Administrator
2	2	MNG	Manager
3	3	STF	Staff/Kasir
4	4	KPL	Kepala
5	5	AGT	Anggota

## Kategori Controller :

```
public function export_pdf()  
{  
    $kategori = KategoriModel::select('kategori_id', 'kategori_kode', 'kategori_nama')  
        ->orderBy('kategori_id')  
        ->get();  
  
    $pdf = Pdf::loadView('kategori.export_pdf', ['kategori' => $kategori]);  
    $pdf->setPaper('a4', 'portrait');  
    $pdf->setOption("isRemoteEnable", true);  
    $pdf->render();  
  
    return $pdf->stream('Data Kategori'.date('Y-m-d H:i:s').'.pdf');  
}
```

## Daftar kategori

Home / Kategori

Daftar Kategori

Import Kategori

Export Kategori

Export Kategori PDF

Tambah Ajax

1 / 1 | - 100% + | [ ] [ ]

Logo Polinema

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

### LAPORAN DATA KATEGORI

No	Kategori ID	Kategori Kode	Kategori Nama
1	1	FOOD	Makanan
2	2	BEV	Minuman
3	3	BOOK	Buku
4	4	SPORT	Olahraga
5	5	FURN	Furniture
6	7	MSC	Music
7	8	ELC	Elektronik



Supplier Controller :

```
public function export_pdf() {  
    $supplier = SupplierModel::select('supplier_id', 'supplier_kode', 'supplier_nama')  
        ->orderBy('supplier_id')  
        ->get();  
  
    $pdf = Pdf::loadView('supplier.export_pdf', ['supplier' => $supplier]);  
    $pdf->setPaper('a4', 'portrait');  
    $pdf->setOption("isRemoteEnable", true);  
    $pdf->render();  
  
    return $pdf->stream('Data Supplier'.date('Y-m-d H:i:s').'.pdf');  
}
```

Daftar Supplier Home / Supplier

---

Daftar Supplier yang terdaftar dalam sistem

[Import Supplier](#) [Export Supplier](#) [Export Supplier PDF](#) [Tambah Ajax](#)

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

---

**LAPORAN DATA SUPPLIER**

No	Supplier ID	Supplier Kode	Supplier Nama
1	1	123	Bhagas
2	3	321	Putera

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



## Migration :

```
Weeks > PWL_PCS > database > migrations > 2025_04_10_084602_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     /**  
10      * Run the migrations.  
11      */  
12     public function up(): void  
13     {  
14         Schema::table('m_user', function (Blueprint $table) {  
15             $table->string('photo_profile')->nullable();  
16         });  
17     }  
18  
19     /**  
20      * Reverse the migrations.  
21      */  
22     public function down(): void  
23     {  
24         Schema::table('m_user', function (Blueprint $table) {  
25             $table->dropColumn('photo_profile');  
26         });  
27     }  
28 };
```

## Web.php

```
Route::post('/update-photo', [UserController::class, 'update_photo']); // upload foto  
Route::post('/delete-photo', [UserController::class, 'delete_photo']);
```

## User Controller:

```
public function update_photo(Request $request)  
{  
    // Validasi file  
    $request->validate([  
        'photo_profile' => 'required|image|mimes:jpeg,png,jpg|max:2048',  
    ]);  
  
    try {  
        $user = auth()->user();  
  
        if (!$user) {  
            return redirect('/login')->with('error', 'Silahkan login terlebih dahulu');  
        }  
  
        $userId = $user->user_id;  
        $userModel = UserModel::find($userId);  
  
        if (!$userModel) {  
            return redirect('/login')->with('error', 'User tidak ditemukan');  
        }  
  
        // Menghapus foto jika sudah ada  
        if ($userModel->photo_profile && file_exists(storage_path('app/public/' . $userModel->photo_profile))) {  
            Storage::disk('public')->delete($userModel->photo_profile);  
        }  
  
        $fileName = 'profile_' . $userId . '_' . time() . '.' . $request->photo_profile->extension();  
        $path = $request->photo_profile->storeAs('profiles', $fileName, 'public');  
  
        UserModel::where('user_id', $userId)->update([  
            'photo_profile' => $path  
        ]);  
  
        return redirect()->back()->with('success', 'Foto profile berhasil diperbarui');  
    } catch (\Exception $e) {  
        return redirect()->back()->with('error', 'Gagal mengupload foto: ' . $e->getMessage());  
    }  
}  
  
public function delete_photo()  
{  
    try {  
        $user = auth()->user();  
  
        if (!$user) {  
            return redirect('/login')->with('error', 'Silahkan login terlebih dahulu');  
        }  
  
        $userModel = UserModel::find($user->user_id);  
  
        if (!$userModel || !$userModel->photo_profile) {  
            return redirect()->back()->with('error', 'Tidak ada foto yang dapat dihapus');  
        }  
  
        if (Storage::disk('public')->exists($userModel->photo_profile)) {  
            Storage::disk('public')->delete($userModel->photo_profile);  
        }  
  
        $userModel->update(['photo_profile' => null]);  
  
        return redirect()->back()->with('success', 'Foto profil berhasil dihapus');  
    } catch (\Exception $e) {  
        return redirect()->back()->with('error', 'Gagal menghapus foto: ' . $e->getMessage());  
    }  
}
```




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>

---


Selamat Datang Home / Welcome

Profil Pengguna

 Foto Profil

No file chosen

Profil Pengguna



No file chosen

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