



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

Muhammad Fairuz Daffa Athallah  
SIB 2B / 15  
2341760079

## **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 Brame 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 Mammy 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.



kategori_id	barang_kode	barang_nama	harga_beli	harga_jual
1	SNK-003	Telur Omega (10 butir)	22000	25000
2	SNK-003	Sari Roti	11500	12500
3	MND-003	Shampoo Pantene	17500	18500
4	BAY-003	Baju Bayi 2th	89000	92500
5	MNM-003	Cleo 500ml	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.

- 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 $1)
                                        <option value="{{ $1->kategori_id }}">{{ $1->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>

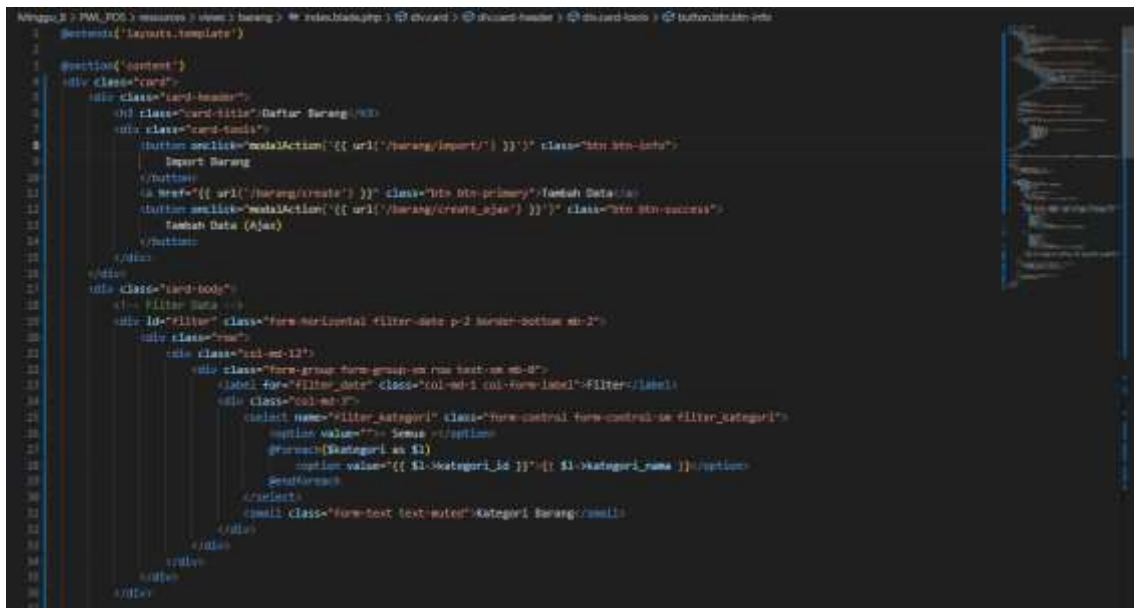
@@endsection

@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%",
```



```
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`

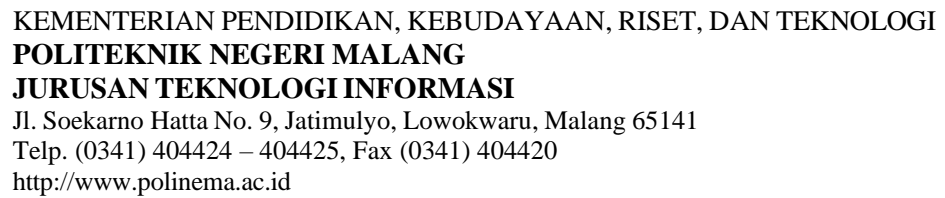


```
<form action="{{ url('/barang/import_ajax') }}" method="POST" id="form-import"
enctype="multipart/form-data">
  @csrf
  <div id="modal-master" class="modal-dialog modal-lg" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Import 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

23 Route::get('/import',[BarangController::class,'import']); // ajax form upload excel
24 Route::post('/import_ajax',[BarangController::class,'import_ajax']); // ajax form import excel
25 });
```

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
```

```
PHP packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

Warning: PHP Startup: Unable to load dynamic library 'php_sqlsrv.dll' (tried: E:/Semester3/Programan_Meb/laragon/bin/php/php-8.3.17-Win32-vs16-x64/ext/php_sqlsrv.dll (The specified module could not be found), E:/Semester3/Programan_Meb/laragon/bin/php/php-8.3.17-Win32-vs16-x64/ext/utmp_php_sqlsrv.dll.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'php_pdo_sqlsrv.dll' (tried: E:/Semester3/Programan_Meb/laragon/bin/php/php-8.3.17-Win32-vs16-x64/ext/php_pdo_sqlsrv.dll (The specified module could not be found), E:/Semester3/Programan_Meb/laragon/bin/php/php-8.3.17-Win32-vs16-x64/ext/utmp_php_pdo_sqlsrv.dll.dll (The specified module could not be found)) in Unknown on line 0

[INFO] No publishable resources for tag [laravel-assets].

Found 1 security vulnerability advisory affecting 1 package.
Run composer audit for a full list of advisories.
PS E:/Semester3/Programan_Meb/laragon/www/PM_2025_Duffa/Vilnggu_B/PM_005>

Found 1 security vulnerability advisory affecting 1 package.
Run composer audit for a full list of advisories.

Found 1 security vulnerability advisory affecting 1 package.
Run composer audit for a full list of advisories.
```

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);
        }

        return DataTables::of($barang)
            ->addIndexColumn()
            ->addColumn('aksi', function ($barang) { // menambahkan kolom aksi

```



```
/*$btn = '<a href="'.url('/barang/' . $barang->barang_id).'>' class="btn btn-  
info btn-sm">Detail</a>';  
$btn .= '<a href="'.url('/barang/' . $barang->barang_id .  
'/edit')>' class="btn btn-warning btn-sm">Edit</a>';  
$btn .= '<form class="d-inline-block" method="POST" action="'.  
url('/barang/' . $barang->barang_id).'>' .  
. csrf_field() . method_field('DELETE') .  
<button type="submit" class="btn btn-danger btn-sm" onclick="return  
confirm('\Apakah Kita 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('/');
```



```
17 public function index()
18 {
19     $breadcrumb = (object){
20         'title' => 'Daftar Barang',
21         'list' => ['Home', 'Barang'];
22     };
23
24     $page = (object){
25         'title' => 'Daftar barang yang terdaftar dalam sistem'
26     };
27
28     $activeMenu = 'barang';
29     $kategori = KategoriModel::all(); // Ambil data kategori untuk filter
30
31     return view('barang.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'kategori' => $kategori, 'activeMenu' => $activeMenu]);
32 }
33
```

```
274 public function import()
275 {
276     return view('barang.import');
277 }
278
279 public function import_ajax(Request $request)
280 {
281     if($request->ajax() || $request->wantsJson()){
282         $rules = [
283             // validasi file harus xls atauxlsx, max 1MB
284             'file_barang' => ['required', 'mimes:xlsx', 'max:1024']
285         ];
286
287         $validator = Validator::make($request->all(), $rules);
288         if($validator->fails()){
289             return response()->json([
290                 'status' => false,
291                 'message' => 'Validasi Gagal',
292                 'msgField' => $validator->errors()
293             ]);
294         }
295         $file = $request->file('file_barang'); // ambil file dari request
296
297         $reader = IOFactory::createReader('Xlsx'); // load reader file excel
298         $reader->setReadDataOnly(true); // hanya membaca data
299         $spreadsheet = $reader->load($file->getRealPath()); // load file excel
300         $sheet = $spreadsheet->getActiveSheet(); // ambil sheet yang aktif
301
302         $data = $sheet->toArray(null, false, true, true); // ambil data excel
303
304         $insert = [];
305         if(count($data) > 1){ // jika data lebih dari 1 baris
306             foreach ($data as $baris => $value) {
307                 if($baris > 1){ // baris ke 1 adalah header, maka lewati
308                     $insert[] = [
309                         'kategori_id' => $value['A'].

```



```
304     $insert = [];  
305     if(count($data) > 1){  
306         // jika data lebih dari 1 baris  
307         foreach ($data as $baris => $value) {  
308             if($baris > 1){  
309                 // baris ke 1 adalah header, maka lewati  
310                 $insert[] = [  
311                     'kategori_id' => $value['A'],  
312                     'barang_kode' => $value['B'],  
313                     'barang_nama' => $value['C'],  
314                     'harga_beli' => $value['D'],  
315                     'harga_jual' => $value['E'],  
316                     'created_at' => now(),  
317                 ];  
318             }  
319         }  
320         if(count($insert) > 0){  
321             // insert data ke database, jika data sudah ada, maka diabaikan  
322             BarangModel::insertOrIgnore($insert);  
323         }  
324         return response()->json([  
325             'status' => true,  
326             'message' => 'Data berhasil diimport'  
327         ]);  
328     }else{  
329         return response()->json([  
330             'status' => false,  
331             'message' => 'Tidak ada data yang diimport'  
332         ]);  
333     }  
334 }  
335 return redirect('/');  
336 }  
337 }
```

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





Import Data Barang

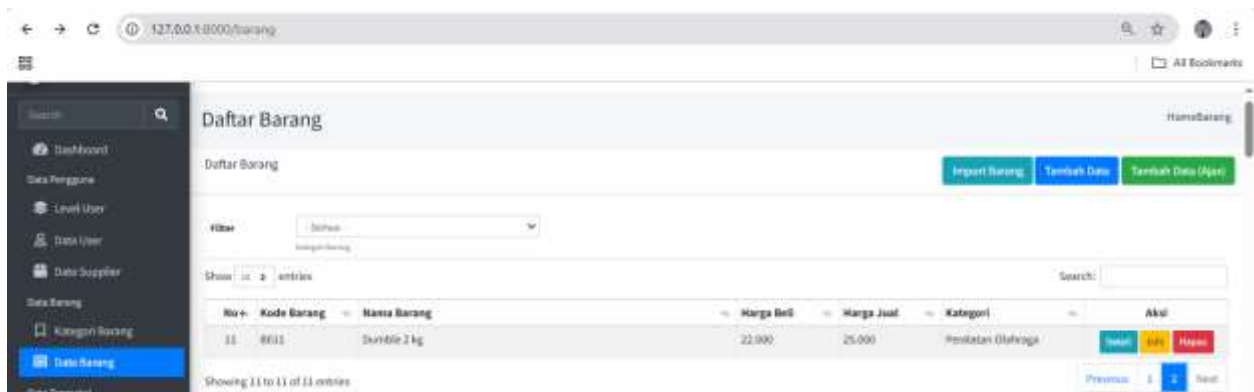
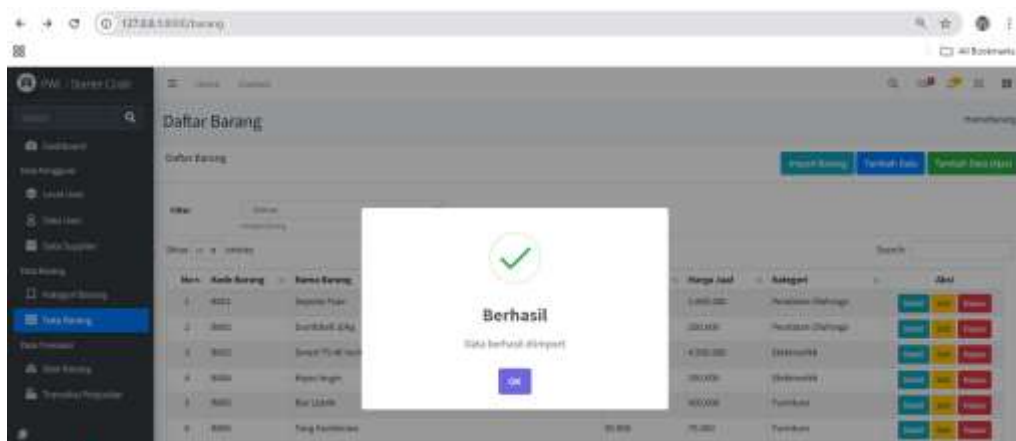
Download Template [Download](#)

Pilih File

[Browse...](#) No file selected.

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

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





## 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 prakkttikum 1 pada repository github kalian.

User

Route

```
36 Route::group(['prefix' => 'user'], function() {
37     Route::get('/', [UserController::class, 'index']); // menampilkan halaman user
38     Route::post('/list', [UserController::class, 'list']); // menampilkan data user untuk form / datatables
39     Route::get('/create', [UserController::class, 'create']); // menampilkan bentuk form untuk tambah user
40     Route::post('/', [UserController::class, 'store']); // menyimpan user data baru
41     Route::get('/create_ajax', [UserController::class, 'create_ajax']); // menampilkan bentuk form untuk tambah user ajax js &
42     Route::post('/ajax', [UserController::class, 'store_ajax']); // menyimpan user data baru ajax js &
43     Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
44     Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
45     Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
46     Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); // menampilkan halaman form edit user ajax js &
47     Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']); // menyimpan perubahan data user ajax js &
48     Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); // untuk tampilan form confirm delete ajax js &
49     Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); // untuk hapus data ajax js &
50     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
51     Route::get('/import', [UserController::class, 'import']); // ajax form upload excel
52     Route::post('/import_ajax', [UserController::class, 'import_ajax']); // ajax form import excel
53 });
```

UserController

```
100 public function import()
101 {
102     return view('user.import');
103 }
104
105 public function import_ajax(Request $request)
106 {
107     if ($request->isAjax() || $request->isMethod('POST')) {
108         $rules = [
109             'file_user' => ['required', 'mimes:xls', 'max:1000']
110         ];
111
112         $validator = Validator::make($request->all(), $rules);
113         if ($validator->fails()) {
114             return response()->json([
115                 'status' => false,
116                 'message' => 'Validasi gagal',
117                 'errors' => $validator->errors()
118             ]);
119         }
120
121         $file = $request->file('file_user');
122
123         $reader = \PhpOffice\PhpSpreadsheet\Reader::create('Xls');
124         $spreadsheet = $reader->load($file->getRealPath());
125         $sheet = $spreadsheet->getActiveSheet();
126
127         $data = $sheet->toArray(null, true, true, true);
128
129         $insert = [];
130         if (count($data) > 1) {
131             foreach ($data as $baris => $baris) {
132                 $insert[] = [
133                     'level_id' => $baris['A'],
134                     'username' => $baris['B'],
135                     'name' => $baris['C'],
136                     'password' => bcrypt($baris['D']),
137                     'created_at' => now(),
138                 ];
139             }
140
141             if (count($insert) > 0) {
142                 DB::table('users')->ignore($insert);
143             }
144
145             return response()->json([
146                 'status' => true,
147                 'message' => 'Data user berhasil diimport'
148             ]);
149         } else {
150             return response()->json([
151                 'status' => false,
152                 'message' => 'Tidak ada data yang diimport'
153             ]);
154         }
155     }
156
157     return redirect('/');
158 }
```



## Import.blade

```
Minggu, 8 > PWA_P05 > resources > views > user > importblade.php > script > <function>
1 <form action="{ url('/user/import_ajax') }" method="POST" id="form-import" enctype="multipart/form-data">
2 </form>
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 User</div>
7 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
8 <span aria-hidden="true"></span></button>
9 </div>
10 </div>
11 <div class="modal-body">
12 <div class="form-group">
13 <label>Download Template</label>
14 <a href="{ asset('template/user.xlsx') }" class="btn btn-info btn-sm" download">
15 <i class="fa fa-file-excel"></i> Download
16 </a>
17 <small id="error-kategori_id" class="error-text form-text text-danger"></small>
18 </div>
19 <div class="form-group">
20 <label>Pilih File</label>
21 <input type="file" name="file_user" id="file_user" class="form-control" required>
22 <small id="error-file_user" class="error-text form-text text-danger"></small>
23 </div>
24 </div>
25 <div class="modal-footer">
26 <button type="button" data-dismiss="modal" class="btn btn-warning" Batal</button>
27 <button type="submit" class="btn btn-primary" Upload</button>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 <script>
34 $(document).ready(function() {
35 <!-- form-import -->
36 <!-- rules -->
37 rules: {
38 file_user: {
39 required: true,
40 extension: "xlsx"
41 }
42 },
43 submitHandler: function(form) {
44 var formData = new FormData(form);
45 $.ajax({
46 url: form.action,
47 type: form.method,
48 data: formData,
49 processData: false,
50 contentType: false,
51 success: function(response) {
52 if (response.status) {
53 $('#myModal').modal('hide');
54 Swal.fire({
55 icon: 'success',
56 title: 'Berhasil',
57 text: response.message
58 });
59 tableUser.ajax.reload();
60 } else {
61 $('#error-text').text('');
62 $.each(response.msgField, function(prefix, val) {
63 $('#error-' + prefix).text(val[0]);
64 });
65 Swal.fire({
66 icon: 'error',
67 title: 'Terjadi Kesalahan',
68 text: response.message
69 });
70 }
71 },
72 error: function(xhr, status, error) {
73 Swal.fire({
74 icon: 'error',
75 title: 'Error AJAX',
76 text: 'Terjadi kesalahan: ' + xhr.status + ' - ' + error
77 }
78 )
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
```

## Import

	A	B	C	D
1	level_id	username	nama	password
2	1	RajaKoma	C.Koma	12345



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>

← → ↻ 127.0.0.1:8000/user

☰

PWL - Starter Code

Search

- Dashboard
- Data Pengguna
- Level User
- Data User**
- Data Supplier

### Daftar User

Daftar user yang terdaftar

Filter:

Show 4 entries

#### Import Data User

Download Template [Download](#)

Pilih File

template\_user.xlsx

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

11	RajaKoma	C.Koma	Administrator	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
----	----------	--------	---------------	------------------------	----------------------	-----------------------



Level

Route

```
75 Route::middleware(['authorize:ADM'])->prefix('level')->group(function () {
76     Route::get('/',[LevelController::class,'index']);//menampilkan halaman awal
77     Route::post('/list',[LevelController::class,'list']);//menampilkan data user bentuk json / datatables
78     Route::get('/create',[LevelController::class,'create']);// menampilkan bentuk form untuk tambah user
79     Route::post('/',[LevelController::class,'store']);//menyimpan user data baru
80     Route::get('/create_ajax',[LevelController::class,'create_ajax']);// menampilkan bentuk form untuk tambah user ajax js 6
81     Route::post('/ajax',[LevelController::class,'store_ajax']);//menyimpan user data baru ajax js 6
82     Route::get('/{id}',[LevelController::class,'show']); // menampilkan detil user
83     Route::get('/{id}/edit',[LevelController::class,'edit']);// menampilkan halaman form edit user
84     Route::put('/{id}',[LevelController::class,'update']);// menyimpan perubahan data user
85     Route::get('/{id}/edit_ajax',[LevelController::class,'edit_ajax']);// menampilkan halaman form edit user ajax js 6
86     Route::put('/{id}/update_ajax',[LevelController::class,'update_ajax']);// menyimpan perubahan data user ajax js 6
87     Route::get('/{id}/delete_ajax',[LevelController::class,'confirm_ajax']);// menghapus data user ajax js 6
88     Route::delete('/{id}/delete_ajax',[LevelController::class,'delete_ajax']);// menghapus data user ajax js 6
89     Route::delete('/{id}',[LevelController::class,'destroy']);// menghapus data user
90     Route::get('/import',[LevelController::class,'import']); // ajax form upload excel
91     Route::post('/import_ajax',[LevelController::class,'import_ajax']); // ajax form import excel
92 });
```

LevelController

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





## Import.blade

```
Mingggo.S > PWL_P03 > resources > view > level > @ importblade.php > @ script
1 <form action="{{ url('/level/import_ajax') }}" method="POST" id="Form-import" enctype="multipart/form-data">
2 @csrf
3 <div id="modal-import" class="modal-dialog modal-lg" role="document">
4 <div class="modal-content">
5 <div class="modal-header">
6 <div class="modal-title">Import Data level</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/level.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 level" id="File level" class="form-control" required>
21 <small id="error-file_level" 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 level: {
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 error: function(xhr, status, error) {
72 Swal.fire({
73 icon: 'error',
74 title: 'Error AJAX',
75 text: 'Terjadi Kesalahan: ' + xhr.status + ' - ' + error
76 });
77 }
78 });
79 }
80 </script>
```

## Import

	A	B	C	
1	level_kod	level_nama		
2	OWN	Owner		



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>

The screenshot shows a web application interface. At the top, there is a browser address bar with the URL "127.0.0.1:8000/level". Below the browser, the application has a dark sidebar on the left with a search bar and several menu items: "Daftar Level", "Data Pengguna", "Level User", "Data User", and "Data Koneksi". The main content area is titled "Daftar Level" and shows a table of levels. A modal window titled "Import Data Level" is open in the center, featuring a "Download Template" button, a "Pilih File" section with a "Choose File" button and a file name "template\_level.xlsx", and "Batal" and "Upload" buttons. Below the modal, a table is visible with one row containing the number "5", the text "OWN", and the text "Owner". To the right of this row are three buttons: "Detail", "Edit", and "Hapus".

ID	Level	Level Kode	Aksi
5	OWN	Owner	<button>Detail</button> <button>Edit</button> <button>Hapus</button>





## Kategori

### Route

```
94 Router::middleware(['auth:admin'])->prefix('kategori')->group(function () {
95     Route::get('/', [KategoriController::class, 'index']); // menampilkan halaman awal
96     Route::post('/list', [KategoriController::class, 'list']); // menampilkan data user bentuk json / datatable
97     Route::get('/create', [KategoriController::class, 'create']); // menampilkan bentuk form untuk tambah user
98     Route::post('/', [KategoriController::class, 'store']); // menyimpan user data baru
99     Route::get('/create_ajax', [KategoriController::class, 'create_ajax']); // menampilkan bentuk form untuk tambah user ajax js &
100     Route::post('/ajax', [KategoriController::class, 'store_ajax']); // menyimpan user data baru ajax js &
101     Route::get('/{id}', [KategoriController::class, 'show']); // menampilkan detail user
102     Route::get('/{id}/edit', [KategoriController::class, 'edit']); // menampilkan halaman form edit user
103     Route::put('/{id}', [KategoriController::class, 'update']); // menyimpan perubahan data user
104     Route::get('/{id}/edit_ajax', [KategoriController::class, 'edit_ajax']); // menampilkan halaman form edit user ajax js &
105     Route::put('/{id}/update_ajax', [KategoriController::class, 'update_ajax']); // menyimpan perubahan data user ajax js &
106     Route::get('/{id}/delete_ajax', [KategoriController::class, 'confirm_ajax']); // menghapus data user ajax js &
107     Route::delete('/{id}/delete_ajax', [KategoriController::class, 'delete_ajax']); // menghapus data user ajax js &
108     Route::delete('/{id}', [KategoriController::class, 'destroy']); // menghapus data user
109     Route::get('/import', [KategoriController::class, 'import']); // ajax form upload excel
110     Route::post('/import_ajax', [KategoriController::class, 'import_ajax']); // ajax form import excel
111 });
```

### KategoriController

```
239 public function import()
240 {
241     return view('kategori.import');
242 }
243
244 public function import_ajax(Request $request)
245 {
246     if ($request->ajax() || $request->wantsJson()) {
247         $rules = [
248             'file_kategori' => ['required', 'mimes:xlsx', 'max:1024']
249         ];
250
251         $validator = Validator::make($request->all(), $rules);
252         if ($validator->fails()) {
253             return response()->json([
254                 'status' => false,
255                 'message' => 'Validasi Gagal',
256                 'msgField' => $validator->errors()
257             ]);
258         }
259
260         $file = $request->file('file_kategori');
261
262         $reader = IOFactory::createReader('Xlsx');
263         $reader->setReadDataOnly(true);
264         $spreadsheet = $reader->load($file->getRealPath());
265         $sheet = $spreadsheet->getActiveSheet();
266         $data = $sheet->toArray(null, false, true, true);
267
268         $insert = [];
269         if (count($data) > 1) {
270             foreach ($data as $baris => $value) {
271                 if ($baris > 1) {
272                     $insert[] = [
273                         'kategori_kode' => $value['A'],
274                         'kategori_nama' => $value['B'],
275                         'created_at' => now(),
276                     ];
277                 }
278             }
279
280             if (count($insert) > 0) {
281                 KategoriModel::insertOrIgnore($insert);
282             }
283
284             return response()->json([
285                 'status' => true,
286                 'message' => 'Data kategori berhasil diimport'
287             ]);
288         } else {
289             return response()->json([
290                 'status' => false,
291                 'message' => 'Tidak ada data yang diimport'
292             ]);
293         }
294     }
295     return redirect('/');
296 }
297
298 public function index()
```



## Import.blade

### Import

	A	B	C
1	kategori_	kategori_nama	
2	FDB	Food and Beverage	
3	AKS	Aksesoris	



Daftar Kategori yang terdaftar dalam sistem

Show 10 entries

ID	KategoriKode	KategoriNama	Aksi
11	FDB	Food and Beverage	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
12	AKS	Aksesoris	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>



## Supplier

### Route

```
133 Route::middleware(['authorize:MNG'])->prefix('supplier')->group(function () {
134     Route::get('/',[SupplierController::class,'index']);//menampilkan halaman awal
135     Route::post('/list',[SupplierController::class,'list']);//menampilkan data user bentuk json / datatables
136     Route::get('/create',[SupplierController::class,'create']);// menampilkan bentuk form untuk tambah user
137     Route::post('/',[SupplierController::class,'store']);//menyimpan user data baru
138     Route::get('/create_ajax',[SupplierController::class,'create_ajax']);// menampilkan bentuk form untuk tambah user
139     Route::post('/ajax',[SupplierController::class,'store_ajax']);//menyimpan user data baru
140     Route::get('/{id}',[SupplierController::class,'show']); // menampilkan detail user
141     Route::get('/{id}/edit',[SupplierController::class,'edit']);// menampilkan halaman form edit user
142     Route::put('/{id}',[SupplierController::class,'update']);// menyimpan perubahan data user
143     Route::get('/{id}/edit_ajax',[SupplierController::class,'edit_ajax']);// menampilkan halaman form edit user
144     Route::put('/{id}/update_ajax',[SupplierController::class,'update_ajax']);// menyimpan perubahan data user
145     Route::get('/{id}/delete_ajax',[SupplierController::class,'confirm_ajax']);// menghapus data user
146     Route::delete('/{id}/delete_ajax',[SupplierController::class,'delete_ajax']);// menghapus data user
147     Route::delete('/{id}',[SupplierController::class,'destroy']);// menghapus data user
148     Route::get('/import',[SupplierController::class,'import']); // ajax form upload excel
149     Route::post('/import_ajax',[SupplierController::class,'import_ajax']); // ajax form import excel
150 });
```

### SupplierController

```
239 public function import()
240 {
241     return view('supplier.import');
242 }
243
244 public function import_ajax(Request $request)
245 {
246     if ($request->ajax() || $request->wantsJson()) {
247         $rules = [
248             'file_supplier' => ['required', 'mimes:xlsx', 'max:1024']
249         ];
250
251         $validator = Validator::make($request->all(), $rules);
252         if ($validator->fails()) {
253             return response()->json([
254                 'status' => false,
255                 'message' => 'Validasi Gagal',
256                 'msgField' => $validator->errors()
257             ]);
258         }
259
260         $file = $request->file('file_supplier');
261
262         $reader = IOFactory::createReader('Xlsx');
263         $reader->setReadDataOnly(true);
264         $spreadsheet = $reader->load($file->getRealPath());
265         $sheet = $spreadsheet->getActiveSheet();
266         $data = $sheet->toArray(null, false, true, true);
267
268         $insert = [];
269         if (count($data) > 1) {
270             foreach ($data as $baris => $value) {
271                 if ($baris > 1) {
272                     $insert[] = [
273                         'supplier_kode' => $value['A'],
274                         'supplier_nama' => $value['B'],
275                         'created_at' => now(),
276                     ];
277                 }
278             }
279
280             if (count($insert) > 0) {
281                 SupplierModel::insertOrIgnore($insert);
282             }
283
284             return response()->json([
285                 'status' => true,
286                 'message' => 'Data level berhasil diimport'
287             ]);
288         } else {
289             return response()->json([
290                 'status' => false,
291                 'message' => 'Tidak ada data yang diimport'
292             ]);
293         }
294     }
295     return redirect('/');
296 }
297
```

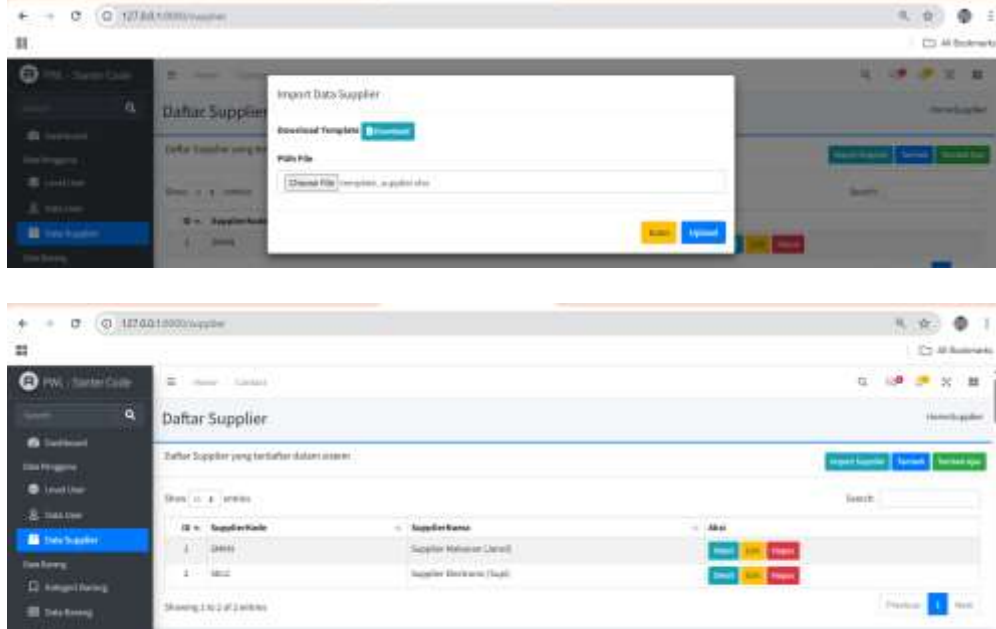


## Import.blade

```
239 public function import()
240 {
241     return view('supplier.import');
242 }
243
244 public function import_ajax(Request $request)
245 {
246     if ($request->ajax() || $request->wantsJson()) {
247         $rules = [
248             'file_supplier' => ['required', 'mimes:xlsx', 'max:1024']
249         ];
250
251         $validator = Validator::make($request->all(), $rules);
252         if ($validator->fails()) {
253             return response()->json([
254                 'status' => false,
255                 'message' => 'Validasi Gagal',
256                 'msgfield' => $validator->errors()
257             ]);
258         }
259
260         $file = $request->file('file_supplier');
261
262         $reader = IOFactory::createReader('Xlsx');
263         $reader->setReadDataOnly(true);
264         $spreadsheet = $reader->load($file->getRealPath());
265         $sheet = $spreadsheet->getActiveSheet();
266         $data = $sheet->toArray(null, false, true, true);
267
268         $insert = [];
269         if (count($data) > 1) {
270             foreach ($data as $baris => $value) {
271                 if ($baris > 1) {
272                     $insert[] = [
273                         'supplier_kode' => $value['A'],
274                         'supplier_nama' => $value['B'],
275                         'created_at' => now(),
276                     ];
277                 }
278             }
279
280             if (count($insert) > 0) {
281                 SupplierModel::insertOrIgnore($insert);
282             }
283
284             return response()->json([
285                 'status' => true,
286                 'message' => 'Data telah berhasil diimport'
287             ]);
288         } else {
289             return response()->json([
290                 'status' => false,
291                 'message' => 'Tidak ada data yang diimport'
292             ]);
293         }
294     }
295     return redirect('/');
296 }
297 }
```

## Import

	A	B	C	D
1	supplier_kode	supplier_nama		
2	SELC	Supplier Electronic (Supi)		
3				



## 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>
```







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
```

```
129 Route::get('/import',[BarangController::class,'import']); // ajax form upload excel
130 Route::post('/import_ajax',[BarangController::class,'import_ajax']); // ajax form import excel
131 Route::get('/barang/export_excel',[BarangController::class,'export_excel']); // export excel
132 ];
133
```

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();
239
240     public function export_excel()
241     {
242         $barang = BarangModel::select('kategori_id','barang_kode','barang_nama','harga_beli','harga_jual')
243             ->orderBy('kategori_id')
244             ->with('kategori')
245             ->get();
246     }
247 }
```

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
```

```
344 $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
345 $sheet = $spreadsheet->getActiveSheet();
346
347 $sheet->setCellValue('A1','No');
348 $sheet->setCellValue('B1','Kode Barang');
349 $sheet->setCellValue('C1','Nama Barang');
350 $sheet->setCellValue('D1','Harga Beli');
351 $sheet->setCellValue('E1','Harga Jual');
352 $sheet->setCellValue('F1','Kategori');
353
354 $sheet->getStyle('A1:F1')->getFont()->setBold(true); //bold header
355
```



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 }
```

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 }
```





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
```

```
339 public function export_excel()
340 {
341     // Ambil data barang yang akan di export
342     $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
343         ->orderBy('kategori_id')
344         ->with('kategori')
345         ->get();
346
347     // load library excel
348     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
349     $sheet = $spreadsheet->getActiveSheet(); // Ambil sheet yang aktif
350
351     $sheet->setCellValue('A1', 'No');
352     $sheet->setCellValue('B1', 'Kode Barang');
353     $sheet->setCellValue('C1', 'Nama Barang');
354     $sheet->setCellValue('D1', 'Harga Beli');
355     $sheet->setCellValue('E1', 'Harga Jual');
356     $sheet->setCellValue('F1', 'Kategori');
357
358     $sheet->getStyle('A1:F1')->getFont()->setBold(true); // Bold header
359
360     $no = 1; // nomor data dimulai dari 1
361     $baris = 2; // baris data dimulai dari baris ke 2
362     foreach ($barang as $key => $value) {
363         $sheet->setCellValue('A'.$baris, $no);
364         $sheet->setCellValue('B'.$baris, $value->barang_kode);
365         $sheet->setCellValue('C'.$baris, $value->barang_nama);
366         $sheet->setCellValue('D'.$baris, $value->harga_beli);
367         $sheet->setCellValue('E'.$baris, $value->harga_jual);
368         $sheet->setCellValue('F'.$baris, $value->kategori->kategori_nama); // ambil nama kategori
369         $baris++;
370         $no++;
371     }
372
373     foreach(range('A', 'F') as $columnID){
374         $sheet->getColumnDimension($columnID)->setAutoSize(true); // Set autosize untuk kolom
375     }
376
377     $sheet->setTitle('Data Barang'); // set title sheet
378
379     $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
380     $filename = 'Data Barang '.date('Y-m-d H:i:s').'.xlsx';
381
382     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
383     header('Content-Disposition: attachment;filename="'.$filename.'"');
384     header('Cache-Control: max-age=0');
385     header('Cache-Control: max-age=1');
386     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
387     header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
388     header('Cache-Control: cache, must-revalidate');
389     header('Pragma: public');
390
391     $writer->save('php://output');
392     exit;
393 }
```

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 Mammy 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

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	B001	Sepeda Fixie	1500000	1800000	Peralatan Olahraga
2	B002	Dumbbell 10kg	200000	250000	Peralatan Olahraga
3	B011	Dumple 2 kg	22000	25000	Peralatan Olahraga
4	B003	Smart TV 40 Inch	4000000	4500000	Elektronikk
5	B004	Kipas Angin	300000	350000	Elektronikk
6	B005	Bor Listrik	500000	600000	Furniture
7	B006	Tang Kombinasi	50000	75000	Furniture
8	B007	Jaket Hoodie	120000	150000	Pakaian
9	B008	Celana Jeans	200000	250000	Pakaian
10	B009	Meja Kerja	800000	1000000	Makanan
11	B010	Rak Buku Kayu	500000	650000	Makanan

## 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 User UserController



```
352 public function export_excel()
353
354     $user = UserModel::select('level_id', 'username', 'nama', 'password')
355         ->orderBy('level_id')
356         ->with('level')
357         ->get();
358
359     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
360     $sheet = $spreadsheet->getActiveSheet();
361
362     $sheet->setCellValue('A1', 'No');
363     $sheet->setCellValue('B1', 'Username');
364     $sheet->setCellValue('C1', 'Nama');
365     $sheet->setCellValue('D1', 'Password');
366     $sheet->setCellValue('E1', 'Level Pengguna');
367
368     $sheet->getStyle('A1:E1')->getFont()->setBold(true);
369
370     $no = 1;
371     $baris = 2;
372     foreach ($user as $key => $value) {
373         $sheet->setCellValue('A' . $baris, $no);
374         $sheet->setCellValue('B' . $baris, $value->username);
375         $sheet->setCellValue('C' . $baris, $value->nama);
376         $sheet->setCellValue('D' . $baris, $value->password);
377         $sheet->setCellValue('E' . $baris, $value->level->level_nama);
378         $baris++;
379         $no++;
380     }
381
382     foreach (range('A', 'E') as $columnID) {
383         $sheet->getColumnDimension($columnID)->setAutoSize(true);
384     }
385
386     $sheet->setTitle('Data User');
387
388     $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
389     $filename = 'Data User ' . date('Y-m-d H:i:s') . '.xlsx';
390
391     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
392     header('Content-Disposition: attachment; filename="'.$filename.'"');
393     header('Cache-Control: max-age=0');
394     header('Cache-Control: max-age=1');
395     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
396     header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
397     header('Cache-Control: cache, must-revalidate');
398     header('Pragma: public');
399
400     $writer->save('php://output');
401     exit;
402 }
```

## Export

No	Username	Nama	Password	Level Pengguna
1	admin	Administrator	\$2y\$12\$TuaBsZgX17z.v9Abo.eN/OfaLMggImVH35.dp13/aebBc2OihajC	Administrator
2	admin11	admin11	123456	Administrator
3	Daffaaaaa	Daffaaaaa	\$2y\$12\$NpeqSEf8jLJuefDnj9AcOdAWFhl4unlkixgm/jH7XRag4eEqwBUi	Administrator
4	DaffaF	DaffaF	\$2y\$12\$/VMA0P.yqviG3AazEqgTc.RifIE80O8R1g5Jquour7FifZVOWebci	Administrator
5	RajaKoma	C.Koma	\$2y\$12\$h1Sr0TrfJUNxtaJp6G5uugLmgu53pQq0TQgnK9C1Hoa.tnX89Ui	Administrator
6	manager	Manager	\$2y\$12\$3jlcRsdTEBaOdRXL87scuyjfbD21D0S7RDy8TslvxhvidUwKa6O	Manager
7	manager_dua	Manager 2	\$2y\$12\$HDsv1/NMfhIbA3HHIPaQJeXvNJfla7ILiv3H74pxK6p9mnaITDxu	Manager
8	manager22	Manager Dua Dua	\$2y\$12\$xE7YPCX1faV0MbN4mMHtj.eoHd/9FAng5n0b4N6oWnyMCVv2UI/ze	Manager
9	Manager33	Manager33	123456	Manager
10	staff	Staff/Kasir	\$2y\$12\$ye8zwhrKkrUUEclYuumBuuhnPW4tnx5NCnVigrnzbpIzhrQPbd7HK	Staff/Kasir
11	customer-1	Pelanggan Pertama	\$2y\$12\$.bTsVE2Oc/1a.d6UqQmqmO4.cfBtl3EUcEOEPkj0ozFrcZMQpR1W6	Pelanggan





## Level LevelController

```
383
384 public function export_excel()
385
386     $level = LevelModel::select('level_kode', 'level_nama')
387         ->orderBy('level_kode')
388         ->get();
389
390     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
391     $sheet = $spreadsheet->getActiveSheet();
392
393     $sheet->setCellValue('A1', 'No');
394     $sheet->setCellValue('B1', 'Kode Level');
395     $sheet->setCellValue('C1', 'Nama Level');
396
397     $sheet->getStyle('A1:C1')->getFont()->setBold(true);
398
399     $no = 1;
400     $baris = 2;
401     foreach ($level as $key => $value) {
402         $sheet->setCellValue('A'.$baris, $no);
403         $sheet->setCellValue('B'.$baris, $value->level_kode);
404         $sheet->setCellValue('C'.$baris, $value->level_nama);
405         $baris++;
406         $no++;
407     }
408
409     foreach (range('A', 'C') as $columnID) {
410         $sheet->getColumnDimension($columnID)->setAutoSize(true);
411     }
412
413     $sheet->setTitle('Data Level');
414
415     $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
416     $filename = 'Data Level '.date('Y-m-d H:i:s').'.xlsx';
417
418     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
419     header('Content-Disposition: attachment;filename="'.$filename.'"');
420     header('Cache-Control: max-age=0');
421     header('Cache-Control: max-age=1');
422     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
423     header('Last-Modified: '.gmdate('D, d M Y H:i:s'). ' GMT');
424     header('Cache-Control: cache, must-revalidate');
425     header('Pragma: public');
426
427     $writer->save('php://output');
428     exit;
429
```

## Export



	A	B	C
1	No	Kode Level	Nama Level
2	1	ADM	Administrator
3	2	CUS	Pelanggan
4	3	MNG	Manager
5	4	OWN	Owner
6	5	STF	Staff/Kasir
7			



## Kategori KategoriController

```
298 public function export_excel()
299 {
300     $kategori = KategoriModel::select('kategori_kode', 'kategori_nama')
301         ->orderBy('kategori_kode')
302         ->get();
303
304     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
305     $sheet = $spreadsheet->getActiveSheet();
306
307     $sheet->setCellValue('A1', 'No');
308     $sheet->setCellValue('B1', 'Kode Kategori');
309     $sheet->setCellValue('C1', 'Nama Kategori');
310
311     $sheet->getStyle('A1:C1')->getFont()->setBold(true);
312
313     $no = 1;
314     $baris = 2;
315     foreach ($kategori as $key => $value) {
316         $sheet->setCellValue('A'.$baris, $no);
317         $sheet->setCellValue('B'.$baris, $value->kategori_kode);
318         $sheet->setCellValue('C'.$baris, $value->kategori_nama);
319         $baris++;
320         $no++;
321     }
322
323     foreach (range('A', 'C') as $columnID) {
324         $sheet->getColumnDimension($columnID)->setAutoSize(true);
325     }
326
327     $sheet->setTitle('Data Kategori');
328
329     $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
330     $filename = 'Data Kategori '.date('Y-m-d H:i:s').'.xlsx';
331
332     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
333     header('Content-Disposition: attachment;filename="'.$filename.'"');
334     header('Cache-Control: max-age=0');
335     header('Cache-Control: max-age=1');
336     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
337     header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
338     header('Cache-Control: cache, must-revalidate');
339     header('Pragma: public');
340
341     $writer->save('php://output');
342     exit;
343 }
```

## Export

	A	B	C
1	No	Kode Kategori	Nama Kategori
2	1	AKS	Aksesoris
3	2	CLOTH	Pakaian
4	3	CML	Cemilan
5	4	ELECC	Elektronikk
6	5	FDB	Food and Beverage
7	6	FRN	Furniture
8	7	KK	Kebutuhan Keluarga
9	8	KM	Kebutuhan Mandi
10	9	MKN	Makanan
11	10	MNR	Minuman Ringan
12	11	SPORT	Peralatan Olahraga
13	12	SPT	Sepatu



Supplier

SupplierController

```
298 public function export_excel()
299
300     $supplier = SupplierModel::select('supplier_kode', 'supplier_nama')
301                             ->orderBy('supplier_kode')
302                             ->get();
303
304     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
305     $sheet = $spreadsheet->getActiveSheet();
306
307     $sheet->setCellValue('A1', 'No');
308     $sheet->setCellValue('B1', 'Kode supplier');
309     $sheet->setCellValue('C1', 'Nama supplier');
310
311     $sheet->getStyle('A1:C1')->getFont()->setBold(true);
312
313     $sno = 1;
314     $baris = 2;
315     foreach ($supplier as $key => $value) {
316         $sheet->setCellValue('A'.$baris, $value->supplier_kode);
317         $sheet->setCellValue('B'.$baris, $value->supplier_nama);
318         $sheet->setCellValue('C'.$baris, $value->supplier_nama);
319         $baris++;
320         $sno++;
321     }
322
323     foreach (range('A', 'C') as $columnID) {
324         $sheet->getColumnDimension($columnID)->setAutoSize(true);
325     }
326
327     $sheet->setTitle('Data supplier');
328
329     $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
330     $filename = 'Data supplier '.date('Y-m-d H:i:s').'.xlsx';
331
332     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
333     header('Content-Disposition: attachment;filename="'.$filename.'");
334     header('Cache-Control: max-age=0');
335     header('Cache-Control: max-age=1');
336     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
337     header('Last-Modified: '.gmdate('D, d M Y H:i:s'). ' GMT');
338     header('Cache-Control: cache, must-revalidate');
339     header('Pragma: public');
340
341     $writer->save('php://output');
342     exit;
343
```

Export

	A	B	C
1	No	Kode supplier	Nama supplier
2	1	SELC	Supplier Electronic (Supi)
3	2	SMKN	Supplier Makanan (Jarod)
4			

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
```

```
90 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

Warning: PHP Startup: Unable to load dynamic library 'php_sqlsrv.dll' (tried: E:/Semester3/Pemrograman_Web/laragon/bin/php/php-8.3.17-Win32-vs16-x64/ext/php_php_sqlsrv.d
Warning: PHP Startup: Unable to load dynamic library 'php_pdo_sqlsrv.dll' (tried: E:/Semester3/Pemrograman_Web/laragon/bin/php/php-8.3.17-Win32-vs16-x64/ext/php_php_
), E:/Semester3/Pemrograman_Web/laragon/bin/php/php-8.3.17-Win32-vs16-x64/ext/php_php_

[INFO] No publishable resources for tag [laravel-assets].

Found 1 security vulnerability advisory affecting 1 package.
Run composer audit for a full list of advisories.
```

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');
}
```

```
396 public function export_pdf(){
397     $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
398         ->orderBy('kategori_id')
399         ->orderBy('barang_kode')
400         ->with('kategori')
401         ->get();
402
403     $pdf = Pdf::loadView('barang.export_pdf', ['barang' => $barang]);
404     $pdf->setPaper('a4', 'portrait');
405     $pdf->setOption("isRemoteEnable", true);
406     $pdf->render();
407
408     return $pdf->stream('Data Barang '.date('Y-m-d H:i:s').'.pdf');
409 }
```

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 `dompdf` 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;
    }
  </style>
</head>
<body>
  <table border="1">
    <tr>
      <td>
        <div class="d-block">
          <img alt="Logo of Politeknik Negeri Malang" data-bbox="125 53 245 142"/>
        </div>
      </td>
      <td>
        KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
        <br>
        POLITEKNIK NEGERI MALANG
        <br>
        JURUSAN TEKNOLOGI INFORMASI
        <br>
        Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
        <br>
        Telp. (0341) 404424 – 404425, Fax (0341) 404420
        <br>
        http://www.polinema.ac.id
      </td>
    </tr>
  </table>
</body>
</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%">
                <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...!!!

The screenshot shows the 'Daftar Barang' page in the PWL - Starter Code application. The page has a sidebar with 'PWL - Starter Code', 'Search', 'Dashboard', 'Data Pengguna', and 'Logout User'. The main content area shows 'Daftar Barang' with a search bar and a table of items. A sidebar on the right contains 'Home', 'Contact', and 'Daftar Barang'. At the bottom right, there are buttons: 'Import Barang', 'Export Barang', 'Export Barang PDF', and 'Tambah Data (Aja)'. Below this, a PDF report titled 'LAPORAN DATA BARANG' is displayed. The report includes the Politeknik Negeri Malang logo and contact information, followed by a table of items.

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	B001	Sepeda Fixie	1.500.000	1.800.000	Peralatan Olahraga
2	B002	Dumbbell 10kg	200.000	250.000	Peralatan Olahraga
3	B011	Dumbell 2 kg	22.000	25.000	Peralatan Olahraga
4	B003	Smart TV 40 Inch	4.000.000	4.500.000	Elektronik
5	B004	Kipas Angin	300.000	350.000	Elektronik
6	B005	Bor Listrik	500.000	600.000	Furniture
7	B006	Tang Kombinasi	50.000	75.000	Furniture
8	B007	Jaket Hoodie	120.000	150.000	Pakaian
9	B008	Celana Jeans	200.000	250.000	Pakaian
10	B009	Meja Kayu	800.000	1.000.000	Makanan
11	B010	Rak Buku Kayu	500.000	650.000	Makanan



### 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 User Controller

```
405 public function export_pdf()  
406 {  
407     $user = UserModel::select('level_id', 'user_id', 'username', 'nama')  
408         ->orderBy('level_id')  
409         ->orderBy('user_id')  
410         ->with('level')  
411         ->get();  
412     $pdf = Pdf::loadView('user.export_pdf', ['user' => $user]);  
413     $pdf->setPaper('a4', 'portrait');  
414     $pdf->setOption('isRemoteEnable', true);  
415     $pdf->render();  
416  
417     return $pdf->stream('Data User'.date('Y-m-d H:i:s').'.pdf');  
418 }
```

#### View





## Level Controller

```
352 public function export_pdf()  
353     $level = LevelModel::select('level_id', 'level_kode', 'level_nama')  
354         ->orderBy('level_id')  
355         ->get();  
356  
357     $pdf = Pdf::loadView('level.export_pdf', ['level' => $level]);  
358     $pdf->setPaper('a4', 'portrait');  
359     $pdf->setOption("isRemoteEnable", true);  
360     $pdf->render();  
361  
362     return $pdf->stream('Data Level'.date('Y-m-d H:i:s').'.pdf');  
363
```

## View

The screenshot shows two parts of the application. The top part is a web browser displaying the 'Daftar Level' page at the URL 127.0.0.1:8000/level. The page has a sidebar with 'Level User' selected and a main content area with a table of levels. The bottom part is a PDF report titled 'Laporan Data LEVEL' generated from the application. The report includes the Polinema logo and contact information, followed by a table of levels.

No	Level ID	Level Kode	Level Nama
1	1	ADM	Administrator
2	2	MNG	Manager
3	3	STF	Staff Kasir
4	6	CUS	Pelanggan
5	16	OWN	Owner

## Kategori Controller

```
346 public function export_pdf()  
347     $kategori = KategoriModel::select('kategori_id', 'kategori_kode', 'kategori_nama')  
348         ->orderBy('kategori_id')  
349         ->get();  
350  
351     $pdf = Pdf::loadView('kategori.export_pdf', ['kategori' => $kategori]);  
352     $pdf->setPaper('a4', 'portrait');  
353     $pdf->setOption("isRemoteEnable", true);  
354     $pdf->render();  
355  
356     return $pdf->stream('Data Kategori'.date('Y-m-d H:i:s').'.pdf');  
357
```





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>

## View

The screenshot shows two parts of the application. The top part is a web browser view of the 'Daftar kategori' page. It has a sidebar with 'PWL - Starter Code', 'Dashboard', 'Data Pengguna', and 'Level User'. The main content area shows 'Daftar kategori yang terdaftar dalam sistem' with buttons for 'Import Kategori', 'Export Kategori', 'Export Kategori PDF', and 'Tambah Baru'. Below this is a search bar and a 'Show' dropdown. The bottom part is a PDF report titled 'LAPORAN DATA KATEGORI' from 'Lingu Polinema'. It contains a table with 12 rows of category data.

No	Kategori ID	Kategori Kode	Kategori Nama
1	1	SPORT	Peralatan Olahraga
2	2	ELECo	Elektronik
3	3	FRN	Furniture
4	4	CLOTH	Pakaian
5	5	MKN	Makanan
6	6	CML	Camilan
7	7	MNR	Minuman Ringan
8	8	SPT	Sepatu
9	12	KK	Kebutuhan Keluarga
10	14	KM	Kebutuhan Mandi
11	21	FDB	Food and Beverage
12	22	AKS	Aksesori

## Supplier Controller

```
346 public function export_pdf()  
347 {  
348     $supplier = SupplierModel::select('supplier_id', 'supplier_kode', 'supplier_nama')  
349         ->orderBy('supplier_id')  
350         ->get();  
351  
352     $pdf = Pdf::loadView('supplier.export_pdf', ['supplier' => $supplier]);  
353     $pdf->setPaper('a4', 'portrait');  
354     $pdf->setOption('isRemoteEnable', true);  
355     $pdf->render();  
356  
357     return $pdf->stream('Data Supplier'.date('Y-m-d H:i:s').'.pdf');
```

## View

The screenshot shows a web browser view of the 'Daftar Supplier' page. It has a sidebar with 'PWL - Starter Code', 'Dashboard', 'Data Pengguna', and 'Level User'. The main content area shows 'Daftar Supplier yang terdaftar dalam sistem' with buttons for 'Import Supplier', 'Export Supplier', 'Export Supplier PDF', and 'Tambah Baru'. Below this is a search bar and a 'Show' dropdown.



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>



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

Foto profile

Migration

```
Minggu_8 > PWL_POS > database > migrations > 2025_04_08_125318_add_profile_picture_to_m_user_table.php > class > down
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

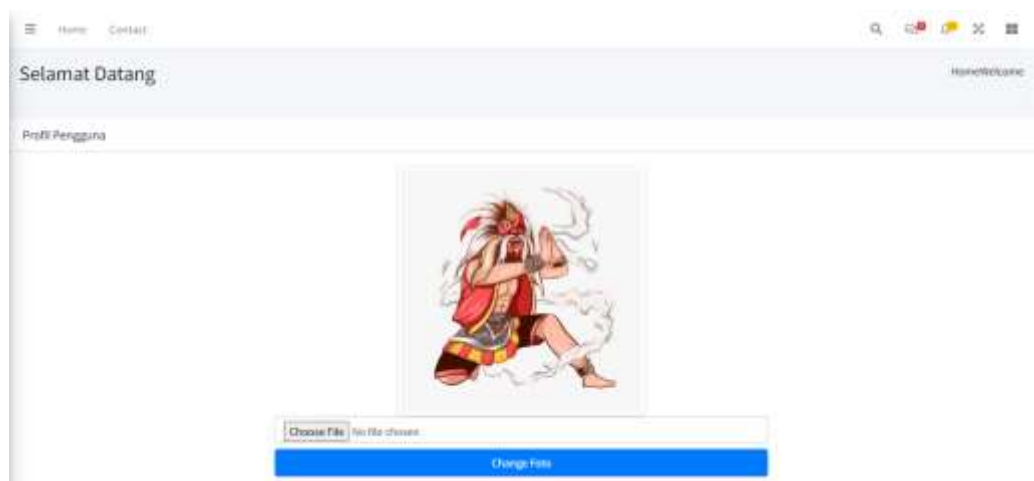
```
57 Route::group(['prefix'=>'user'], function(){
58     Route::get('/', [UserController::class, 'index']); //menampilkan halaman awal
59     Route::post('/list', [UserController::class, 'list']); //menampilkan data user bentuk json / datatables
60     Route::get('/create', [UserController::class, 'create']); // menampilkan bentuk form untuk tambah user
61     Route::post('/', [UserController::class, 'store']); // menyimpan user data baru
62     Route::get('/create_ajax', [UserController::class, 'create_ajax']); // menampilkan bentuk form untuk tambah user ajax js 6
63     Route::post('/ajax', [UserController::class, 'store_ajax']); // menyimpan user data baru ajax js 6
64     Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
65     Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
66     Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
67     Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); // menampilkan halaman form edit user ajax js 6
68     Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']); // menyimpan perubahan data user ajax js 6
69     Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); // untuk tampilan form confirm delete ajax js 6
70     Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); // untuk hapus data ajax js 6
71     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
72     Route::get('/import', [UserController::class, 'import']); // ajax form upload excel
73     Route::post('/import_ajax', [UserController::class, 'import_ajax']); // ajax form import excel
74     Route::get('/export_excel', [UserController::class, 'export_excel']); // export excel
75     Route::get('/export_pdf', [UserController::class, 'export_pdf']); // export pdf
76     Route::post('/update-photo', [UserController::class, 'update_photo']); // upload foto
77     Route::post('/delete-photo', [UserController::class, 'delete_photo']);
78 });
```

UserController



```
421 public function update_photo(Request $request)
422 {
423     // validasi file
424     $request->validate([
425         'photo_profile' => 'required|image|size:1024,2048|max:2048',
426     ]);
427
428     try {
429         $user = auth()->user();
430
431         if (!$user) {
432             return redirect('/login')->with('error', 'Silahkan login terlebih dahulu');
433         }
434
435         $userId = $user->id;
436         $userModel = UserModel::find($userId);
437
438         if (!$userModel) {
439             return redirect('/login')->with('error', 'User tidak ditemukan');
440         }
441
442         // hapus foto jika sudah ada
443         if ($userModel->photo_profile && file_exists(storage_path('app/public/' . $userModel->photo_profile))) {
444             Storage::disk('public')->delete($userModel->photo_profile);
445         }
446
447         $fileName = 'profile_' . $userId . '-' . time() . '.' . $request->photo_profile->extension();
448         $path = $request->photo_profile->storeAs('profiles', $fileName, 'public');
449
450         $userModel->where('user_id', $userId)->update([
451             'photo_profile' => $path
452         ]);
453
454         return redirect()->back()->with('success', 'Foto profil berhasil diupdate');
455     } catch (\Exception $e) {
456         return redirect()->back()->with('error', 'Gagal mengupload foto: ' . $e->getMessage());
457     }
458 }
459
460 public function delete_photo()
461 {
462     try {
463         $user = auth()->user();
464
465         if (!$user) {
466             return redirect('/login')->with('error', 'Silahkan login terlebih dahulu');
467         }
468
469         $userModel = UserModel::find($user->id);
470
471         if (!$userModel || !$userModel->photo_profile) {
472             return redirect()->back()->with('error', 'Tidak ada foto yang dapat dihapus');
473         }
474
475         if (Storage::disk('public')->exists($userModel->photo_profile)) {
476             Storage::disk('public')->delete($userModel->photo_profile);
477         }
478
479         $userModel->update(['photo_profile' => null]);
480
481         return redirect()->back()->with('success', 'Foto profil berhasil dihapus');
482     } catch (\Exception $e) {
483         return redirect()->back()->with('error', 'Gagal menghapus foto: ' . $e->getMessage());
484     }
485 }
```

View



2. Jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
3. Submit kode untuk impementasi export pdf pada repository github kalian.



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>

---

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