



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 $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>
</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>  
</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
```

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('/');
```

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.

Daftar Barang

Home / Barang

Daftar Barang

Import Barang

Tambah Data

Tambah Data (Ajax)

Filter

Semua

Kategori Barang

Show 10 entries

Search:

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori	Aksi
1	BRG002	Mie Goreng	9.000	14.000	Makanan	Detail Edit Hapus
2	BRG003	Ayam Bakar	12.000	18.000	Makanan	Detail Edit Hapus
3	BRG004	Kaos Polos	30.000	45.000	Pakaian	Detail Edit Hapus
4	BRG005	Celana Jeans	70.000	100.000	Pakaian	Detail Edit Hapus
5	BRG006	Keripik Kentang	8.000	12.000	Snack	Detail Edit Hapus
6	BRG007	Biskuit Coklat	5.000	9.000	Snack	Detail Edit Hapus
7	BRG008	Apel Fuji	15.000	20.000	Buah-buahan	Detail Edit Hapus
8	BRG009	Jeruk Manis	10.000	15.000	Buah-buahan	Detail Edit Hapus
9	BRG010	Wortel Segar	7.000	10.000	Sayuran	Detail Edit Hapus
10	BRG011	Bayam Hijau	5.000	8.000	Sayuran	Detail Edit Hapus

Import Data Barang

Download Template

Download

Pilih File

Choose File

template_barang.xlsx

Batal

Upload

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori	Aksi
11	BRG012	Bubur Ayam	10.000	13.000	Makanan	Detail Edit Hapus
12	BRG013	Sarung	20.000	30.000	Pakaian	Detail Edit Hapus
13	BRG014	Mie Ayam	8.000	12.000	Makanan	Detail Edit Hapus
14	BRG015	Chocolatos	5.000	7.000	Snack	Detail Edit Hapus
15	BRG016	Semangka	25.000	30.000	Buah-buahan	Detail Edit Hapus



Tugas 1 – Implementasi File Upload untuk Import Data:

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

Tabel m_user

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

public function importAjax(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);

        $insert = [];
        if (count($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('/');
}
```



Screenshot of a web application interface for user management.

Daftar User

Buttons: Import User, Tambah Data, Tambah Data Ulang

Filter:

Show: 10 entries

No	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Pelanggan	Staff/Kasir	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
7	manager56	Manager55	Manager	Detail Edit Hapus
8	manager12	Manager11	Manager	Detail Edit Hapus

Showing 1 to 8 of 8 entries

Import Data User

Download Template [Download](#)

Pilih File

template_user.xlsx

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

9	Admin2	BurYam	Administrator	Detail Edit Hapus
10	Manager969	Lionel Audric	Staff/Kasir	Detail Edit Hapus



Table m_level

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

```
public function import_ajax(Request $request)  
{  
    if ($request->ajax() || $request->wantsJson()) {  
        $rules = [  
            'file_level' => ['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_level');  
  
        $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[] = [  
                        'level_kode' => $value['A'],  
                        'level_nama' => $value['B'],  
                        'created_at' => now(),  
                    ];  
                }  
            }  
        }  
  
        if (count($insert) > 0) {  
            LevelModel::insertOrIgnore($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('/');  
}
```

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

Screenshot of the application interface showing the 'Daftar Level' (Level List) page. The page displays a table with columns: No, Kode Level, Nama Level, and Aksi. The table contains three entries: ADM (Administrator), MNG (Manager), and STF (Staff Kasir). The 'Aksi' column contains buttons for 'Detail', 'Edit', and 'Hapus' (Delete). The page also includes a search bar and a 'Tambah Data (Ajax)' button.

No	Kode Level	Nama Level	Aksi
1	ADM	Administrator	Detail Edit Hapus
2	MNG	Manager	Detail Edit Hapus
3	STF	Staff Kasir	Detail Edit Hapus

Screenshot of the 'Import Data Level' form. The form includes a 'Download Template' button, a 'Pilih File' section with a 'Choose File' button and a text input field containing 'template_level.xlsx', and 'Batal' (Cancel) and 'Upload' buttons.

Import Data Level

Download Template [Download](#)

Pilih File

template_level.xlsx

4	CEO	Chief Executive Officer	Detail Edit Hapus
5	SPV	Supervisor	Detail Edit Hapus



Tabel m_kategori

```
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'  
        ]);  
    }  
}
```

localhost/PWL_2023/Minggu08/jobsheet8/public/kategori

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Login

Home / Kategori

Daftar Kategori

Import Kategori | Tambah Data | Tambah Data (Ajax)

Show 23 entries

No	Kode Kategori	Nama Kategori	Aksi
1	MKN	Makanan	Detail Edit Hapus
2	PKN	Pakaian	Detail Edit Hapus
3	SNCK	Snack	Detail Edit Hapus
4	BHBN	Buah-buahan	Detail Edit Hapus
5	SVR	Sayuran	Detail Edit Hapus
6	MNR	Minuman Ringan	Detail Edit Hapus

Showing 1 to 6 of 6 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Import Data Kategori

Download Template [Download](#)

Pilih File

[Choose File](#) template_kategori.xlsx

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

7	LPTP	Laptop	Detail Edit Hapus
8	SPT	Sepatu	Detail Edit Hapus



Tabel t_stok

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

```
public function importAjax(Request $request)  
{  
    if ($request->ajax() || $request->wantsJson()) {  
        $rules = [];  
        'file_stok' => ['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_stok');  
    $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[] = [  
                    'barang_id' => $value['A'],  
                    'stok_tanggal' => $value['B'],  
                    'stok_jumlah' => $value['C'],  
                    'created_at' => now(),  
                ];  
            }  
        }  
    }  
    if (count($insert) > 0) {  
        StokModel::insertOrIgnore($insert);  
    }  
    return response()->json([  
        'status' => true,  
        'message' => 'Data stok berhasil diimport'  
    ]);  
}
```

No	Nama Barang	Tanggal	Jumlah Stok	Aksi
1	Mie Goreng	2025-03-02 17:48:40	30	Detail Edit Hapus
2	Ayam Bakar	2025-03-02 17:48:40	25	Detail Edit Hapus
3	Kaos Polos	2025-03-02 17:48:40	20	Detail Edit Hapus
4	Celana Jeans	2025-03-02 17:48:40	15	Detail Edit Hapus
5	Keripik Kentang	2025-03-02 17:48:40	10	Detail Edit Hapus
6	Biskuit Coklat	2025-03-02 17:48:40	50	Detail Edit Hapus
7	Apel Fuji	2025-03-02 17:48:40	40	Detail Edit Hapus
8	Jeruk Manis	2025-03-02 17:48:40	35	Detail Edit Hapus
9	Wortel Segar	2025-03-02 17:48:40	45	Detail Edit Hapus
10	Bayam Hijau	2005-04-09 00:00:00	20	Detail Edit Hapus

Import Data Stok

Download Template [Download](#)

Pilih File

Choose File

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

ID	Nama Barang	Nama User	Tanggal Stok	Jumlah Stok	Aksi
11	Mie Goreng	Manager	0000-00-00 00:00:00	30	Detail Edit Hapus
12	Apel Fuji	Administrator	0000-00-00 00:00:00	25	Detail Edit Hapus



2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
 - Membuat view import.blade.php untuk form upload dan download template
 - Memodifikasi rute untuk memproses upload
 - Menginstall library phpoffice/phpspreadsheet untuk membaca dan memproses file excel ke Laravel
 - Memodifikasi BarangController untuk memproses data dari file excel
3. Submit kode untuk impementasi prakktikum 1 pada repository github kalian.



B. Export Excel di Laravel

Fitur export ke Excel pada Laravel merupakan alat yang sangat berguna dalam aplikasi web yang membutuhkan pengolahan dan penyajian data secara profesional. Fitur ini memungkinkan pengguna untuk 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
```

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();
```

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
```

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
```

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	Kebutuhan Bayi
11	BAY-002	Popok Mammy Poko	56000	58000	Kebutuhan Bayi
12	BAY-003	Baju Bayi 2th	89000	92500	Kebutuhan 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	Aksi
1	BRG002	Mie Goreng	9.000	14.000	Makanan	Detail Edit Hapus
2	BRG003	Ayam Bakar	12.000	18.000	Makanan	Detail Edit Hapus
3	BRG004	Kaos Polos	30.000	45.000	Pakaian	Detail Edit Hapus
4	BRG005	Celana Jeans	70.000	100.000	Pakaian	Detail Edit Hapus
5	BRG006	Keripik Kentang	8.000	12.000	Snack	Detail Edit Hapus
6	BRG007	Biskuit Coklat	5.000	9.000	Snack	Detail Edit Hapus
7	BRG008	Apel Fuji	15.000	20.000	Buah-buahan	Detail Edit Hapus
8	BRG009	Jeruk Manis	10.000	15.000	Buah-buahan	Detail Edit Hapus
9	BRG010	Wortel Segar	7.000	10.000	Sayuran	Detail Edit Hapus
10	BRG011	Bayam Hijau	5.000	8.000	Sayuran	Detail Edit Hapus



Tugas 2 – Implementasi File Export Excel:

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

Tabel m_user

```
358 public function export_ajax()  
359 {  
360     $user = UserModel::select('level_id', 'username', 'nama', 'password')  
361         ->orderBy('level_id')  
362         ->with('level')  
363         ->get();  
364  
365     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();  
366     $sheet = $spreadsheet->getActiveSheet();  
367  
368     $sheet->setCellValue('A1', 'No');  
369     $sheet->setCellValue('B1', 'Username');  
370     $sheet->setCellValue('C1', 'Nama');  
371     $sheet->setCellValue('D1', 'Password');  
372     $sheet->setCellValue('E1', 'Level Pengguna');  
373  
374     $sheet->getStyle('A1:E1')->getFont()->setBold(true);  
375  
376     $no = 1;  
377     $baris = 2;  
378     foreach ($user as $key => $value) {  
379         $sheet->setCellValue('A' . $baris, $no);  
380         $sheet->setCellValue('B' . $baris, $value->username);  
381         $sheet->setCellValue('C' . $baris, $value->nama);  
382         $sheet->setCellValue('D' . $baris, $value->password);  
383         $sheet->setCellValue('E' . $baris, $value->level->level_nama);  
384         $baris++;  
385         $no++;  
386     }  
387  
388     foreach (range('A', 'E') as $columnID) {  
389         $sheet->getColumnDimension($columnID)->setAutoSize(true);  
390     }  
391  
392     $sheet->setTitle('Data User');  
393  
394     $writer = \PhpOffice\PhpSpreadsheet\Writer\Xlsx::create($spreadsheet, 'Xlsx');  
395     $filename = 'Data User ' . date('Y-m-d H:i:s') . '.xlsx';  
396  
397     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');  
398     header('Content-Disposition: attachment; filename="' . $filename . '"');  
399     header('Cache-Control: max-age=0');  
400     header('Cache-Control: max-age=1');  
401     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');  
402     header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');  
403     header('Cache-Control: cache, must-revalidate');  
404     header('Pragma: public');  
405  
406     $writer->save('php://output');  
407     exit;  
408 }
```

	A	B	C	D	E
1	No	Username	Nama	Password	Level Pengguna
2	1	admin	Administrator	\$2y\$12\$6IMxCVbtJaEoU.dBz8bz6OeN2hMMZ3N0.ngV/a6IKsiZF10QUIC.	Administrator
3	2	Admin2	BurYam	\$2y\$12\$d0JbMIQGEFK.RAHLSyorj.M5UqDzcZBBu0u3XZOUgD1zFvylqyupS	Administrator
4	3	manager	Manager	\$2y\$12\$5i4qri0q872pzo97So.BPeFG8PFUIyxCSLDzCjA1EXE1ozrWQMe1q	Manager
5	4	manager_dua	Manager 2	\$2y\$12\$CwIXzqsPbf5OEvgZ.vVIMu83x443ynikWceeC7q53NLezn5vPWkai	Manager
6	5	manager33	Manager Tiga Tiga	\$2y\$12\$WYayb/nZ4SxclHG58NguBZ.twWJuUZ.WBSVhFyCk8dDY2WhHdaQLmK	Manager
7	6	manager56	Manager55	\$2y\$12\$KjsPUqFXM38ee8Ghck5KLO2vju6ABqyoE2buGgiViXjp0C26Ft5e	Manager
8	7	manager12	Manager11	\$2y\$12\$UHB93VME7bBHK5uUn7KfcuGdutEP5QFztzmEVm90d8ZZNdVz1DWrs	Manager
9	8	staff	Staff/Kasir	\$2y\$12\$AgRZBp9AganZ9kvDxsCn0.tprRedA41hrdAqrqIM.IiFyOp8xD6NS	Staff/Kasir
10	9	customer -1	Pelanggan	\$2y\$12\$/7Rv4Rz.gW1EP2kw3wxD.PiAz78o.lcu6cTEHff4RiOsrCUXRi.W	Staff/Kasir
11	10	Manager969	Lionel Audric	\$2y\$12\$foeMGzU0r7EihnlZ4bulXHBgWajhJo.j8mY0XrGz6OQpAZPV9W	Staff/Kasir



Tabel m_level

```
330 public function export_ajax()  
331 {  
332     $level = LevelModel::select('level_kode', 'level_nama')  
333         ->orderBy('level_kode')  
334         ->get();  
335  
336     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();  
337     $sheet = $spreadsheet->getActiveSheet();  
338  
339     $sheet->setCellValue('A1', 'No');  
340     $sheet->setCellValue('B1', 'Kode Level');  
341     $sheet->setCellValue('C1', 'Nama Level');  
342  
343     $sheet->getStyle('A1:C1')->getFont()->setBold(true);  
344  
345     $no = 1;  
346     $baris = 2;  
347     foreach ($level as $key => $value) {  
348         $sheet->setCellValue('A' . $baris, $no);  
349         $sheet->setCellValue('B' . $baris, $value->level_kode);  
350         $sheet->setCellValue('C' . $baris, $value->level_nama);  
351         $baris++;  
352         $no++;  
353     }  
354  
355     foreach (range('A', 'C') as $columnID) {  
356         $sheet->getColumnDimension($columnID)->setAutoSize(true);  
357     }  
358  
359     $sheet->setTitle('Data Level');  
360  
361     $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');  
362     $filename = 'Data Level ' . date('Y-m-d H:i:s') . '.xlsx';  
363  
364     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');  
365     header('Content-Disposition: attachment; filename="' . $filename . '");  
366     header('Cache-Control: max-age=0');  
367     header('Cache-Control: max-age=1');  
368     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');  
369     header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');  
370     header('Cache-Control: cache, must-revalidate');  
371     header('Pragma: public');  
372  
373     $writer->save('php://output');  
374     exit;  
375 }
```

No	Kode Level	Nama Level
1	ADM	Administrator
2	MNG	Manager
3	STF	Staff/Kasir
4	CEO	Chief Executive Officer
5	SPV	Supervisor

	A	B	C
1	No	Kode Level	Nama Level
2	1	ADM	Administrator
3	2	CEO	Chief Executive Officer
4	3	MNG	Manager
5	4	SPV	Supervisor
6	5	STF	Staff/Kasir



Tabel m_kategori

```
341 public function export_ajax()
342 {
343     $kategori = KategoriModel::select('kategori_kode', 'kategori_nama')
344         ->orderBy('kategori_kode')
345         ->get();
346
347     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
348     $sheet = $spreadsheet->getActiveSheet();
349
350     $sheet->setCellValue('A1', 'No');
351     $sheet->setCellValue('B1', 'Kode Kategori');
352     $sheet->setCellValue('C1', 'Nama Kategori');
353
354     $sheet->getStyle('A1:C1')->getFont()->setBold(true);
355
356     $no = 1;
357     $baris = 2;
358     foreach ($kategori as $key => $value) {
359         $sheet->setCellValue('A' . $baris, $no);
360         $sheet->setCellValue('B' . $baris, $value->kategori_kode);
361         $sheet->setCellValue('C' . $baris, $value->kategori_nama);
362         $baris++;
363         $no++;
364     }
365
366     foreach (range('A', 'C') as $columnID) {
367         $sheet->getColumnDimension($columnID)->setAutoSize(true);
368     }
369
370     $sheet->setTitle('Data Kategori');
371
372     $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
373     $filename = 'Data Kategori ' . date('Y-m-d H:i:s') . '.xlsx';
374
375     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
376     header('Content-Disposition: attachment; filename="' . $filename . '"');
377     header('Cache-Control: max-age=0');
378     header('Cache-Control: max-age=1');
379     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
380     header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
381     header('Cache-Control: cache, must-revalidate');
382     header('Pragma: public');
383
384     $writer->save('php://output');
385     exit;
386 }
```

	A	B	C
1	No	Kode Kategori	Nama Kategori
2	1	BHBH	Buah-buahan
3	2	LPTP	Laptop
4	3	MKN	Makanan
5	4	MNR	Minuman Ringan
6	5	PKN	Pakaian
7	6	SNCK	Snack
8	7	SPT	Sepatu
9	8	SYR	Sayuran



Table t_stok

```
350 public function export_ajax()  
351 {  
352     $stok = StokModel::select('barang_id', 'user_id', 'stok_tanggal', 'stok_jumlah')  
353         ->orderBy('user_id')  
354         ->with('user')  
355         ->get();  
356  
357     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();  
358     $sheet = $spreadsheet->getActiveSheet();  
359  
360     $sheet->setCellValue('A1', 'No');  
361     $sheet->setCellValue('B1', 'Nama Barang');  
362     $sheet->setCellValue('C1', 'Nama User');  
363     $sheet->setCellValue('D1', 'Tanggal');  
364     $sheet->setCellValue('E1', 'Jumlah Stok');  
365  
366     $sheet->getStyle('A1:E1')->getFont()->setBold(true);  
367  
368     $no = 1;  
369     $baris = 2;  
370     foreach ($stok as $key => $value) {  
371         $sheet->setCellValue('A' . $baris, $no);  
372         $sheet->setCellValue('B' . $baris, $value->barang->barang_nama);  
373         $sheet->setCellValue('C' . $baris, $value->user->nama);  
374         $sheet->setCellValue('D' . $baris, $value->stok_tanggal);  
375         $sheet->setCellValue('E' . $baris, $value->stok_jumlah);  
376         $baris++;  
377         $no++;  
378     }  
379  
380     foreach (range('A', 'E') as $columnID) {  
381         $sheet->getColumnDimension($columnID)->setAutoSize(true);  
382     }  
383  
384     $sheet->setTitle('Data Stok');  
385  
386     $writer = IOFactory::createWriter($spreadsheet, 'xlsx');  
387     $filename = 'Data Stok ' . date('Y-m-d H:i:s') . '.xlsx';  
388  
389     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');  
390     header('Content-Disposition: attachment;filename="'. $filename .'"');  
391     header('Cache-Control: max-age=0');  
392     header('Cache-Control: max-age=1');  
393     header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');  
394     header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');  
395     header('Cache-Control: cache, must-revalidate');  
396     header('Pragma: public');  
397  
398     $writer->save('php://output');  
399     exit;  
400 }
```

	A	B	C	D	E
1	No	Nama Barang	Nama User	Tanggal	Jumlah Stok
2	1	Mie Goreng	Administrator	2025-03-02 17:48:40	30
3	2	Ayam Bakar	Administrator	2025-03-02 17:48:40	25
4	3	Wortel Segar	Administrator	2025-03-02 17:48:40	45
5	4	Apel Fuji	Administrator	0000-00-00 00:00:00	25
6	5	Kaos Polos	Manager	2025-03-02 17:48:40	20
7	6	Celana Jeans	Manager	2025-03-02 17:48:40	15
8	7	Keripik Kentang	Manager	2025-03-02 17:48:40	10
9	8	Mie Goreng	Manager	0000-00-00 00:00:00	30
10	9	Biskuit Coklat	Staff/Kasir	2025-03-02 17:48:40	50
11	10	Apel Fuji	Staff/Kasir	2025-03-02 17:48:40	40
12	11	Jeruk Manis	Staff/Kasir	2025-03-02 17:48:40	35
13	12	Bayam Hijau	Staff/Kasir	2005-04-09 00:00:00	20



2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
 - Memodifikasi view index untuk menambahkan button export excel
 - Menambahkan rute untuk memproses export excel
 - Menambahkan function export_ajax pada controller untuk memproses export data
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
```

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');
}
```

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`

```
<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>
        <tr>
            <th>Kategori</th>
            <th>Kode</th>
            <th>Nama</th>
            <th>Harga Beli</th>
            <th>Harga Jual</th>
        </tr>
        <tr>
            <td>
        <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>
```



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

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

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori	Aksi
1	BRG002	Mie Goreng	9.000	14.000	Makanan	Detail Edit Hapus
2	BRG003	Ayam Bakar	12.000	18.000	Makanan	Detail Edit Hapus
3	BRG004	Kaos Polos	30.000	45.000	Pakaian	Detail Edit Hapus
4	BRG005	Celana Jeans	70.000	100.000	Pakaian	Detail Edit Hapus
5	BRG006	Keripik Kentang	8.000	12.000	Snack	Detail Edit Hapus
6	BRG007	Biskuit Coklat	5.000	9.000	Snack	Detail Edit Hapus
7	BRG008	Apel Fuji	15.000	20.000	Buah-buahan	Detail Edit Hapus
8	BRG009	Jeruk Manis	10.000	15.000	Buah-buahan	Detail Edit Hapus
9	BRG010	Wortel Segar	7.000	10.000	Sayuran	Detail Edit Hapus
10	BRG011	Bayam Hijau	5.000	8.000	Sayuran	Detail Edit Hapus

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
Jl. Soekarno-Hatta No. 9 Malang 65141
Telepon (0341) 404424 Pex. (031) 105. 0341- 404425 Fax. (0341) 404420
Laman: www.polinema.ac.id

LAPORAN DATA BARANG

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	BRG002	Mie Goreng	9.000	14.000	Makanan
2	BRG003	Ayam Bakar	12.000	18.000	Makanan
3	BRG012	Bubur Ayam	10.000	13.000	Makanan
4	BRG014	Mie Ayam	8.000	12.000	Makanan
5	BRG004	Kaos Polos	30.000	45.000	Pakaian
6	BRG005	Celana Jeans	70.000	100.000	Pakaian
7	BRG013	Sarung	20.000	30.000	Pakaian
8	BRG006	Keripik Kentang	8.000	12.000	Snack
9	BRG007	Biskuit Coklat	5.000	9.000	Snack
10	BRG015	Chocolatos	5.000	7.000	Snack
11	BRG008	Apel Fuji	15.000	20.000	Buah-buahan
12	BRG009	Jeruk Manis	10.000	15.000	Buah-buahan
13	BRG016	Semangka	25.000	30.000	Buah-buahan
14	BRG010	Wortel Segar	7.000	10.000	Sayuran
15	BRG011	Bayam Hijau	5.000	8.000	Sayuran



Tugas 3 – Implementasi Export PDF pada Laravel :

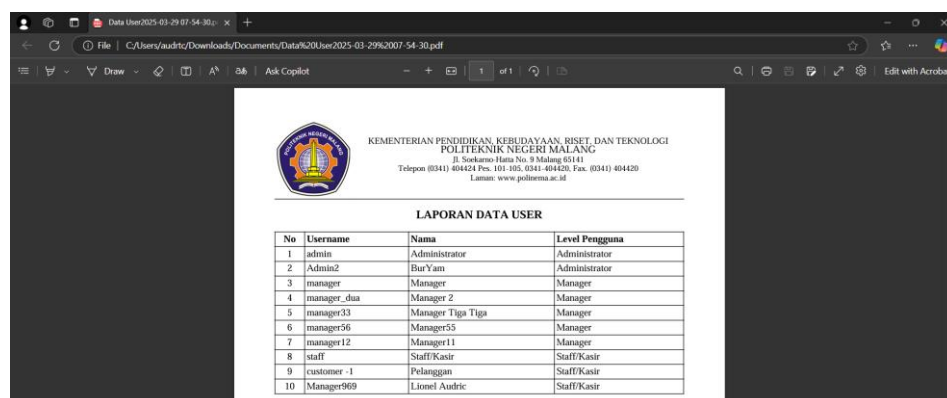
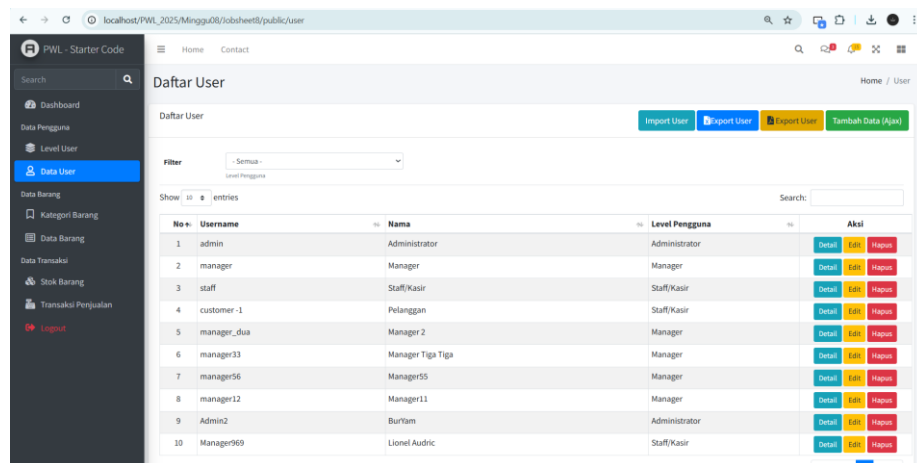
1. Silahkan implementasikan export pdf pada project kalian masing-masing untuk semua menu

Tabel m_user

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

    // use Barryvdh\DomPDF\Facade\Pdf;
    $pdf = Pdf::loadView('user.export_pdf', ['user' => $user]);
    $pdf->setPaper('a4', 'portrait');
    $pdf->setOption("isRemoteEnabled", true);
    $pdf->render();

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





Tabel m_level

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

    // use Barryvdh\DomPDF\Facade\Pdf;
    $pdf = Pdf::loadView(['level.export_pdf', ['level' => $level]]);
    $pdf->setPaper('a4', 'portrait');
    $pdf->setOption("isRemoteEnabled", true);
    $pdf->render();

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

Screenshot of a web application interface showing a table of user levels.

Daftar Level

Showing 1 to 5 of 5 entries

No	Kode Level	Nama Level	Aksi
1	ADM	Administrator	Detail Edit Hapus
2	MNG	Manager	Detail Edit Hapus
3	STF	Staff/Kasir	Detail Edit Hapus
4	CEO	Chief Executive Officer	Detail Edit Hapus
5	SPV	Supervisor	Detail Edit Hapus

Showing 1 to 5 of 5 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Screenshot of a PDF document titled "LAPORAN DATA LEVEL".

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

LAPORAN DATA LEVEL

No	Kode Level	Nama Level
1	ADM	Administrator
2	CEO	Chief Executive Officer
3	MNG	Manager
4	SPV	Supervisor
5	STF	Staff/Kasir



Table m_kategori

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

localhost/PWL_2025/Minggu08/JobSheet8/public/kategori

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Logout

Home Contact

Daftar Kategori

Import Kategori Export Kategori Export Kategori Tambah Data (Ajax)

Show 10 entries

Search:

No	Kode Kategori	Nama Kategori	Aksi
1	MKN	Makanan	Detail Edit Hapus
2	PKN	Pakaian	Detail Edit Hapus
3	SNCK	Snack	Detail Edit Hapus
4	BHBH	Buah-buahan	Detail Edit Hapus
5	SYR	Sayuran	Detail Edit Hapus
6	MNR	Minuman Ringan	Detail Edit Hapus
7	LPTP	Laptop	Detail Edit Hapus
8	SPT	Sepatu	Detail Edit Hapus

Showing 1 to 8 of 8 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Data kategori 2023-03-29 08:01

C:\Users\audrtz\Downloads\Data\20\kategori\202025-03-29\2008-09-52.pdf

File Draw A⁺ Ask Copilot

1 of 1

Edit with Acrobat

No	Kode Kategori	Nama Kategori
1	BHBH	Buah-buahan
2	LPTP	Laptop
3	MKN	Makanan
4	MNR	Minuman Ringan
5	PKN	Pakaian
6	SNCK	Snack
7	SPT	Sepatu
8	SYR	Sayuran



Tabel t_stok

```
public function export_pdf()
{
    $stok = StokModel::select('barang_id', 'user_id', 'stok_tanggal', 'stok_jumlah')
        ->orderBy('user_id')
        ->with('barang', 'user')
        ->get();

    // use Barryvdh\DomPDF\Facade\Pdf;
    $pdf = Pdf::loadView('stok.export_pdf', ['stok' => $stok]);
    $pdf->setPaper('a4', 'portrait');
    $pdf->setOption("isRemoteEnabled", true);
    $pdf->render();

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

ID	Nama Barang	Nama User	Tanggal Stok	Jumlah Stok	Aksi
1	Mie Goreng	Administrator	2025-03-02 17:48:40	30	Detail Edit Hapus
2	Ayam Bakar	Administrator	2025-03-02 17:48:40	25	Detail Edit Hapus
3	Kaos Polos	Manager	2025-03-02 17:48:40	20	Detail Edit Hapus
4	Celana Jeans	Manager	2025-03-02 17:48:40	15	Detail Edit Hapus
5	Kertipik Kentang	Manager	2025-03-02 17:48:40	10	Detail Edit Hapus
6	Biskuit Coklat	Staff/Kasir	2025-03-02 17:48:40	50	Detail Edit Hapus
7	Apel Fuji	Staff/Kasir	2025-03-02 17:48:40	40	Detail Edit Hapus
8	Jeruk Manis	Staff/Kasir	2025-03-02 17:48:40	35	Detail Edit Hapus
9	Wortel Segar	Administrator	2025-03-02 17:48:40	45	Detail Edit Hapus
10	Bayam Hijau	Staff/Kasir	2025-04-09 00:00:00	20	Detail Edit Hapus

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

LAPORAN DATA STOK

No	Nama Barang	Nama User	Tanggal Stok	Jumlah Stok
1	Mie Goreng	admin	2025-03-02 17:48:40	30
2	Ayam Bakar	admin	2025-03-02 17:48:40	25
3	Apel Fuji	admin	0000-00-00 00:00:00	25
4	Wortel Segar	admin	2025-03-02 17:48:40	45
5	Mie Goreng	manager	0000-00-00 00:00:00	30
6	Kaos Polos	manager	2025-03-02 17:48:40	20
7	Celana Jeans	manager	2025-03-02 17:48:40	15
8	Kertipik Kentang	manager	2025-03-02 17:48:40	10
9	Biskuit Coklat	staff	2025-03-02 17:48:40	50
10	Apel Fuji	staff	2025-03-02 17:48:40	40
11	Jeruk Manis	staff	2025-03-02 17:48:40	35
12	Bayam Hijau	staff	2025-04-09 00:00:00	20

2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
 - Instalasi library dompdf untuk export pdf
 - Menambahkan button export pdf pada view index
 - Menambahkan rute untuk memproses export pdf
 - Menambahkan function export_pdf untuk memproses export data
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

Menambahkan kolom photo_profile di tabel m_user

```
database > migrations > 2025_04_05_135330_photo_profile.php > class > up > Closure
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  };
29
30
31 protected $table = 'm_user';
32 protected $primaryKey = 'user_id';
33
34 protected $fillable = ['username', 'password', 'nama', 'level_id', 'photo_profile'];
```

Membuat function profile dan update_photo di UserController

```
public function profile()
{
    $user = auth()->user();

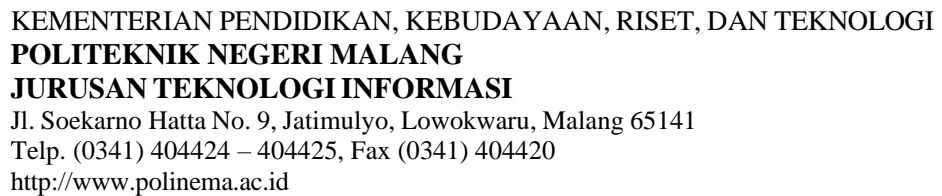
    if (!$user) {
        return redirect('/login')->with('error', 'Silahkan login terlebih dahulu');
    }

    $breadcrumb = (object) [
        'title' => 'Profile User',
        'list' => ['Home', 'Profile']
    ];

    $page = (object) [
        'title' => 'Profil Pengguna'
    ];

    $activeMenu = 'profile';

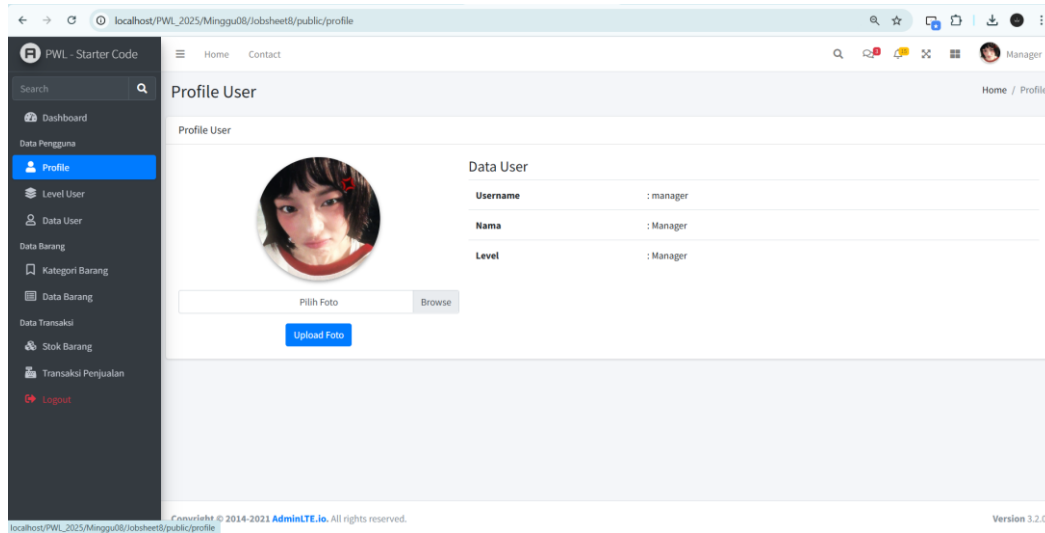
    return view('user.photo_profile', compact('user', 'breadcrumb', 'page', 'activeMenu'));
}
```



Membuat rute baru

Views profile.blade.php

Jobsheet 08 – PWL 2023/2024



2. Jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
 - Menambahkan kolom di tabel m_user
 - Menambahkan function profile dan update_photo di UserController
 - Menambahkan rute baru untuk memproses photo profile
 - Membuat tampilan baru untuk menu profile
 - Memodifikasi di bagian sidebar dan header layouts
3. Submit kode untuk impementasi export pdf pada repository github kalian.

*** Sekian, dan selamat belajar ***