



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

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.

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



```
@extends('layouts.template')

@section('content')
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">Daftar barang</h3>
            <div class="card-tools">
                <button onclick="modalAction('{{ url('/barang/import') }}')" class="btn btn-info">Import Barang</button>
                <a href="{{ url('/barang/create') }}" class="btn btn-primary">Tambah
Data</a>
                <button onclick="modalAction('{{ url('/barang/create_ajax') }}')" class="btn
btn-success">Tambah Data (Ajax)</button>
            </div>
        </div>
        <div class="card-body">
            <!-- untuk Filter data -->
            <div id="filter" class="form-horizontal filter-date p-2 border-bottom mb-2">
                <div class="row">
                    <div class="col-md-12">
                        <div class="form-group form-group-sm row text-sm mb-0">
                            <label for="filter_date" class="col-md-1 col-form-
label">Filter</label>
                            <div class="col-md-3">
                                <select name="filter_kategori" class="form-control form-
control-sm filter_kategori">
                                    <option value="">- Semua -</option>
                                    @foreach($kategori as $l)
                                        <option value="{{ $l->kategori_id }}">{{ $l->
kategori_nama }}</option>
                                    @endforeach
                                </select>
                                <small class="form-text text-muted">Kategori Barang</small>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            @if(session('success'))
                <div class="alert alert-success">{{ session('success') }}</div>
            @endif
            @if(session('error'))
                <div class="alert alert-danger">{{ session('error') }}</div>
            @endif
        </div>
    </div>
</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

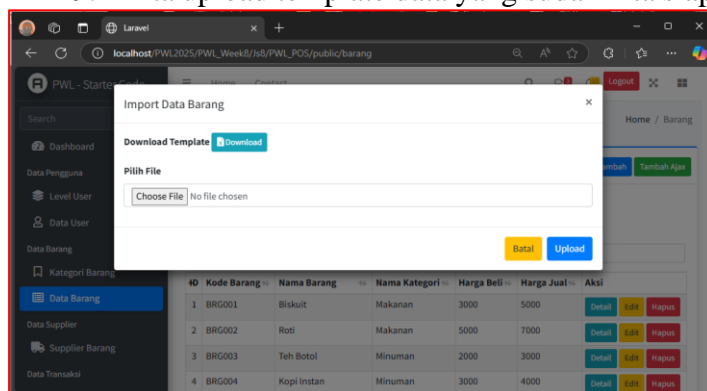
Browse...

No file selected.

Batal

Upload

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



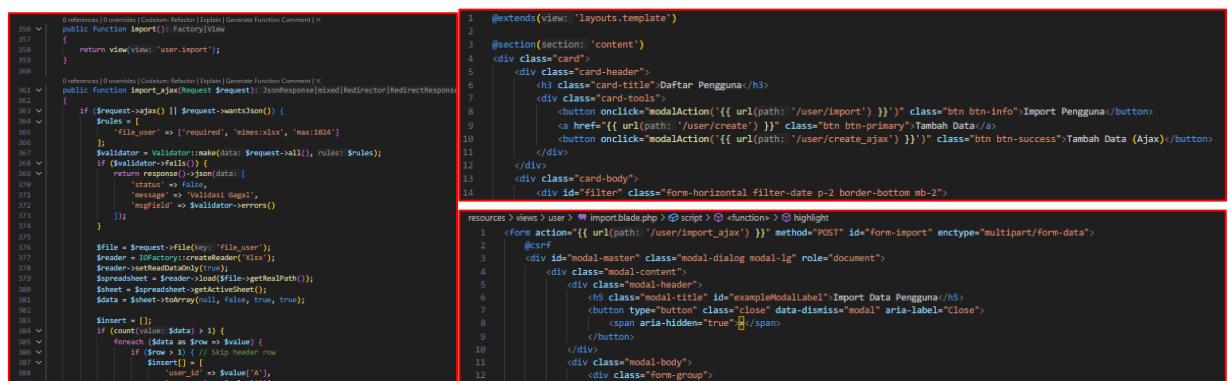
	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual
1					
2		1 BRG011	Astor	3000	5000

Tidak Terjadi apa-apa, malahan data Templatenya terupdate menjadi punya saya yang baru

Tugas 1 – Implementasi File Upload untuk Import Data:

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

➤ Menu User:



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

user_id	username	nama	level_id
17	manager78	Managertujuhapan	2

Import Data Pengguna

Download Template

Download

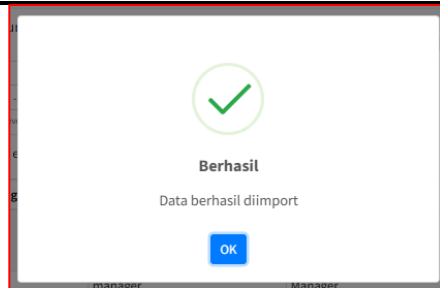
Pilih File

Choose File

import_user.xlsx

Batal

Upload

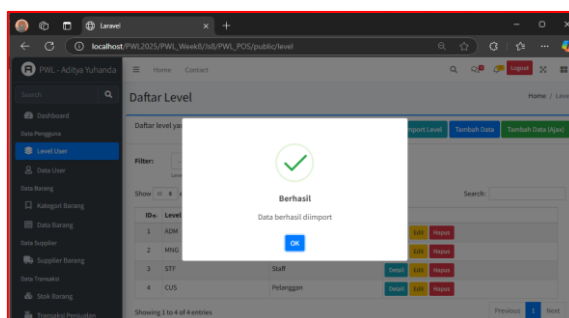
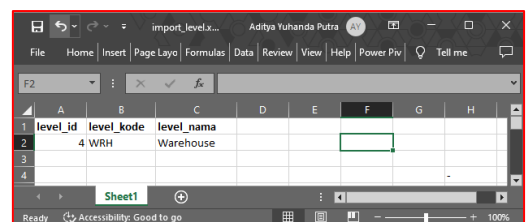
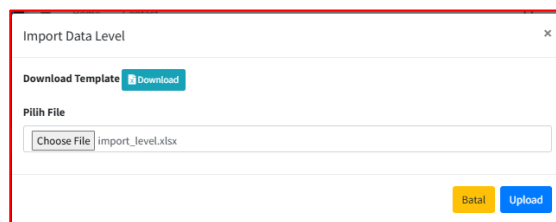


No.	ID Pengguna	Username	Nama	Level	Aksi
11	17	manager78	Managertujuhapan	Manager	Detail Edit Hapus

➤ Menu Level:

```
app > Http > Controllers > LevelController.php > PHP > LevelController
268
269 0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
270 public function import(): Factory|View
271 {
272     return view(view: 'level.import');
273 }
274
275 /**
276  * Mengimport data level dari file excel
277  */
278
279 0 references | 0 overrides | Codeium: Refactor | Explain | X
280 public function import_ajax(Request $request): JsonResponse|mixed|Redirector
281 {
282     if ($request->ajax() || $request->wantsJson()) {
283         $rules = [
284             'file_level' => ['required', 'mimes:xlsx', 'max:1024']
285         ];
286         $validator = Validator::make(data: $request->all(), rules: $rules);
287         if ($validator->fails()) {
288             return response()->json(data: [
289                 'status' => false,
290                 'message' => 'Validasi Gagal',
291                 'msgfield' => $validator->errors()
292             ]);
293         }
294     }
295 }
296
resources > views > level > index.blade.php > {} {} > div.card.card-outline.card-primary > div.card-body
1
2 @extends(view: 'layouts.template')
3
4 @section(section: 'content')
5     <div class="card card-outline card-primary">
6         <div class="card-header">
7             <h3 class="card-title">{{ $page->title }}</h3>
8         </div>
9         <div class="card-tools">
10             <button onclick="modalAction('{{ url(path: '/level/import') }}')" class="btn btn-info">Import Level</button>
11             <a href="{{ url(path: '/level/create') }}" class="btn btn-primary">Tambah Data</a>
12             <button onclick="modalAction('{{ url(path: '/level/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
13         </div>
14     </div>
15
resources > views > level > import.blade.php > @script
1
2 <form action="{{ url(path: '/level/import_ajax') }}" method="POST" id="form-import-level" enctype="multipart/form-data">
3     @csrf
4     <div id="modal-master" class="modal-dialog modal-lg" role="document">
5         <div class="modal-content">
6             <div class="modal-header">
7                 <h5 class="modal-title">Import Data Level</h5>
8                 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
9                     <span aria-hidden="true">&times;</span>
10             </div>
11             <div class="modal-body">
12                 <div class="form-group">
13                     <input type="file" class="form-control" />
14                 </div>
15                 <div class="text-center">
16                     <button type="button" class="btn btn-warning">Batal</button>
17                     <button type="button" class="btn btn-success">Upload</button>
18                 </div>
19             </div>
20         </div>
21     </div>
22 </script>
```

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



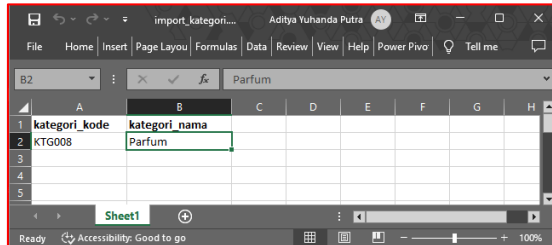
ID	Level Kode	Level Nama	Aksi
1	ADM	Administrator	Detail Edit Hapus
2	MNG	Manager	Detail Edit Hapus
3	STF	Staff	Detail Edit Hapus
4	CUS	Pelanggan	Detail Edit Hapus
5	WRH	Warehouse	Detail Edit Hapus

➤ Menu Kategori:

```
app > Http > Controllers > KategoriController.php > PHP > KategoriController
268
269 0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
270 public function import(): Factory|View
271 {
272     return view(view: 'kategori.import');
273 }
274
275 0 references | 0 overrides | Codeium: Refactor | Explain | X
276 public function import_ajax(Request $request): JsonResponse|mixed|Redirector|RedirectResponse
277 {
278     if ($request->ajax() || $request->wantsJson()) {
279         // Validasi file
280         $rules = [
281             'file_kategori' => ['required', 'mimes:xlsx', 'max:1024']
282         ];
283         $validator = Validator::make(data: $request->all(), rules: $rules);
284         if ($validator->fails()) {
285             return response()->json(data: [
286                 'status' => false,
287                 'message' => 'Validasi Gagal',
288                 'msgfield' => $validator->errors()
289             ]);
290         }
291     }
292 }
293
resources > views > kategori > index.blade.php > {} {} >
1
2 @extends(view: 'layouts.template')
3
4 @section(section: 'content')
5     <div class="card card-outline card-primary">
6         <div class="card-header">
7             <h3 class="card-title">Daftar Kategori</h3>
8         </div>
9         <div class="card-tools">
10             <button onclick="modalAction('{{ url(path: '/kategori/import') }}')" class="btn btn-info">Import Kategori</button>
11             <a href="{{ url(path: '/kategori/create') }}" class="btn btn-primary">Tambah Data</a>
12             <button onclick="modalAction('{{ url(path: '/kategori/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
13         </div>
14     </div>
15
resources > views > kategori > import.blade.php
1
2 <form action="{{ url(path: '/kategori/import_ajax') }}" method="POST" id="form-import-kategori" enctype="multipart/form-data">
3     @csrf
4     <div id="modal-master" class="modal-dialog modal-lg" role="document">
5         <div class="modal-content">
6             <div class="modal-header">
7                 <h5 class="modal-title">Import Data Kategori</h5>
8                 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
9                     <span aria-hidden="true">&times;</span>
10             </div>
11             <div class="modal-body">
12                 <div class="form-group">
13                     <input type="file" class="form-control" />
14                 </div>
15                 <div class="text-center">
16                     <button type="button" class="btn btn-warning">Batal</button>
17                     <button type="button" class="btn btn-success">Upload</button>
18                 </div>
19             </div>
20         </div>
21     </div>
22 </form>
```



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



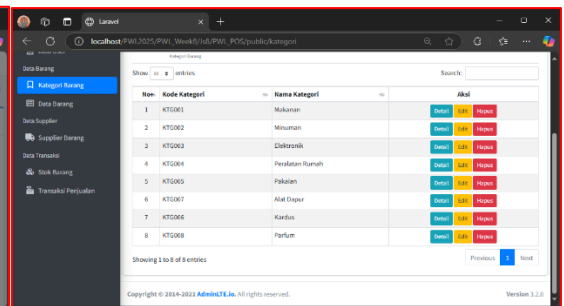
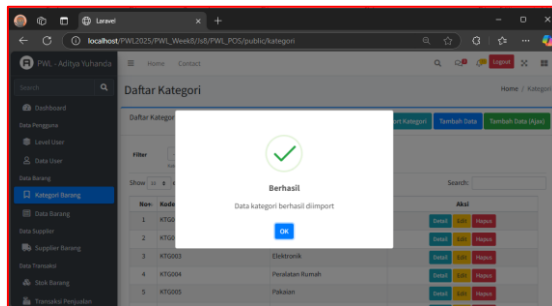
Import Data Kategori

Download Template [Download](#)

Pilih File

Choose File import_kategori.xlsx

Batal Upload

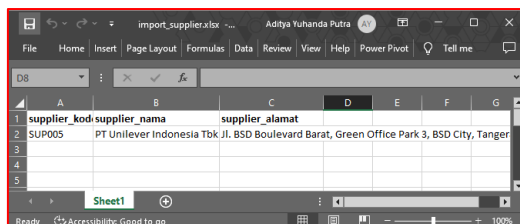


➤ Menu Supplier:

```
267 public function import(): Factory\View
268 {
269     return view('supplier.import');
270 }
271
272 1 reference [0 overrides] | Codeium: Refactor | Explain | Generate Function Comment | X
273 public function import_ajax(Request $request): JsonResponse|mixed|Redirector
274 {
275     if ($request->ajax() || $request->wantsJson()) {
276         $rules = [
277             'file_supplier' => ['required', 'mimes:xlsx,xls', 'max:1024']
278         ];
279         $validator = Validator::make($request->all(), $rules);
280         if ($validator->fails()) {
281             return response()->json($data: [
282                 'status' => false,
283                 'message' => 'Validasi Gagal',
284                 'msgField' => $validator->errors()
285             ]);
286         }
287     }
288 }
```

```
resources > view > supplier > @import.blade.php > (1) 11
2 <div class="card card-outline card-primary">
3 <div class="card-header">
4 <h3 class="card-title">Daftar Supplier</h3>
5 <div class="card-tools">
6 <button onclick="modalAction('{{ url(path: '/supplier/import') }}')" class="btn btn-info">Import Supplier</button>
7 <a href="{{ url(path: '/supplier/create') }}" class="btn btn-primary">Tambah Data</a>
8 <button onclick="modalAction('{{ url(path: '/supplier/create_ajax') }}" class="btn btn-success">Tambah Data (Ajax)</button>
9 </div>
10 </div>
```

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



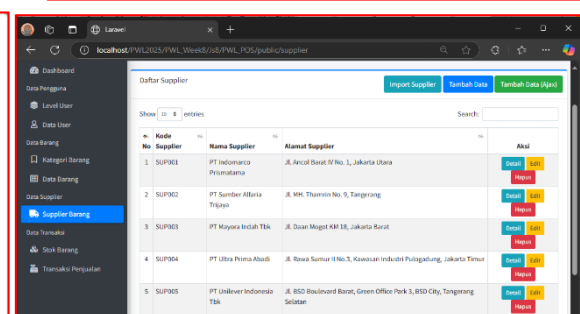
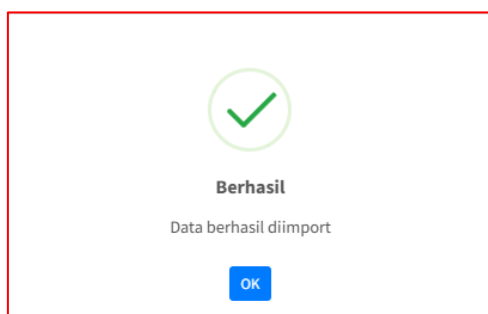
Import Data Supplier

Download Template [Download](#)

Pilih File

Choose File import_supplier.xlsx

Batal Upload



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

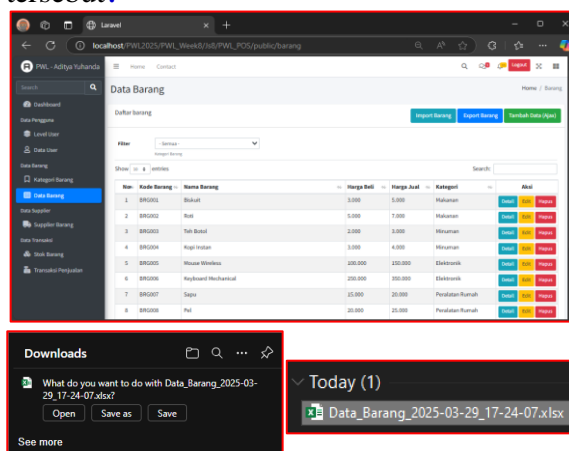
7. Kita set lebar tiap kolom di excel untuk menyesuaikan dengan panjang karakter pada masing-masing kolom

```
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	BRG001	Biskuit	3000	5000	Makanan
2	BRG002	Roti	5000	7000	Makanan
3	BRG011	Astor	3000	5000	Makanan
4	BRG003	Teh Botol	2000	3000	Minuman
5	BRG004	Kopi Instan	3000	4000	Minuman
6	BRG005	Mouse Wireless	100000	150000	Elektronik
7	BRG006	Keyboard Mechanical	250000	350000	Elektronik
8	BRG007	Sapu	15000	20000	Peralatan Rumah
9	BRG008	Pel	20000	25000	Peralatan Rumah
10	BRG009	Kaos Polos	40000	50000	Pakaian
11	BRG010	Jaket Hoodie	100000	120000	Pakaian



Tugas 2 – Implementasi File Export Excel:

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

➤ Menu User:

The screenshot shows the implementation of the 'export_excel' function in the UserController. The function selects user data and exports it to an Excel file. The browser view shows the 'Daftar User' page with buttons for 'Import Pengguna', 'Export User', and 'Tambah Data (Ajax)'. A download dialog is visible for the 'Export User' button.

```
app > Http > Controllers > UserController.php > PHP > UserController
414 public function export_excel(): never
415 {
416     // Ambil data users yang akan diekspor
417     $users = UserModel::select(columns: 'user_id', 'username', 'nama', 'level_id')
418         ->orderBy(column: 'user_id')
419         ->with(relations: 'level')
420         ->get();
421
422     // Load library PhpSpreadsheet
423     $spreadsheet = new Spreadsheet();
424     $sheet = $spreadsheet->getActiveSheet();
425
426     // Set header kolom
427     $sheet->setCellValue('A1', 'No');
428     $sheet->setCellValue('B1', 'User ID');
429     $sheet->setCellValue('C1', 'Username');
430     $sheet->setCellValue('D1', 'Nama');
431     $sheet->setCellValue('E1', 'Level');
432
433     <div class="card-tools">
434         <button onclick="modalAction('{{ url(path: '/user/import') }}')" class="btn btn-info">Import Pengguna</button>
435         <a href="{{ url(path: '/user/export_excel') }}" class="btn btn-primary"><i class="fa fa-file-excel"></i> Export User</a>
436         <button onclick="modalAction('{{ url(path: '/user/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
437     </div>
438
439     Route::post(uri: 'import_ajax', action: [UserController::class, 'import_ajax']); // ajax import excel
440     Route::get(uri: 'export_excel', action: [UserController::class, 'export_excel']); //export excel
441 }
```

No	User ID	Username	Nama	Level
1	1	admin	Admin	Administrator
2	2	manager	Manager	Manager
3	3	staff	Staff/Kasir	Staff
4	7	customer-1	Pelanggan Pertama	Pelanggan
5	8	manager_dua	Manager 2	Manager
6	9	manager22	Manager Dua Dua	Manager
7	10	manager33	Manager Tiga Tiga	Manager
8	11	manager56	Manager55	Manager
9	15	admin1	admin1	Administrator
10	16	manager21	managerdua-satu	Manager

➤ Menu Level:

The screenshot shows the implementation of the 'export_excel' function in the LevelController. The function selects level data and exports it to an Excel file. The browser view shows the 'Daftar Level' page with buttons for 'Import Level', 'Export Level', and 'Tambah Data (Ajax)'. A download dialog is visible for the 'Export Level' button.

```
app > Http > Controllers > LevelController.php > PHP > LevelController
332 public function export_excel(): never
333 {
334     // Ambil data level yang akan diekspor
335     $levels = LevelModel::select(columns: 'level_id', 'level_kode', 'level_nama')->orderBy(column: 'level_id')->get();
336
337     // Load library PhpSpreadsheet
338     $spreadsheet = new Spreadsheet();
339     $sheet = $spreadsheet->getActiveSheet();
340
341     // Set header kolom
342     $sheet->setCellValue('A1', 'No');
343     $sheet->setCellValue('B1', 'ID Level');
344     $sheet->setCellValue('C1', 'Kode Level');
345     $sheet->setCellValue('D1', 'Nama Level');
346
347     <div class="card-tools">
348         <button onclick="modalAction('{{ url(path: '/level/import') }}')" class="btn btn-info">Import Level</button>
349         <a href="{{ url(path: '/level/export_excel') }}" class="btn btn-primary"><i class="fa fa-file-excel"></i> Export Level</a>
350         <button onclick="modalAction('{{ url(path: '/level/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
351     </div>
352
353     Route::post(uri: 'import_ajax', action: [LevelController::class, 'import_ajax']); // ajax import excel
354     Route::get(uri: 'export_excel', action: [LevelController::class, 'export_excel']); //export excel
355 }
```

No	ID Level	Kode Level	Nama Level
1	1	ADM	Administrator
2	2	MNG	Manager
3	3	STF	Staff
4	4	CUS	Pelanggan
5	9	WRH	Warehouse



➤ Menu Kategori:

```
app > Http > Controllers > KategoriController.php > PHP > KategoriController
0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
324 public function export_excel(): BinaryFileResponse
325 {
326     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
327     $sheet = $spreadsheet->getActiveSheet();
328
329     // Set Header Kolom
330     $sheet->setCellValue('A1', 'No');
331     $sheet->setCellValue('B1', 'Kode Kategori');
332     $sheet->setCellValue('C1', 'Nama Kategori');
333
334     // Ambil Data dari Database
335     $kategori = KategoriModel::select(columns: 'kategori_kode', 'kategori_nama')->get();



<h3 class="card-title">Daftar Kategori</h3>
<div class="card-tools">
<button onclick="modalAction('{{ url(path: '/kategori/import') }}')" class="btn btn-info">Import Kategori</button>
<a href="{{ url(path: '/kategori/export_excel') }}" class="btn btn-primary"><i class="fa fa-file-excel"></i> Export Kategori Barang</a>
<button onclick="modalAction('{{ url(path: '/kategori/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
</div>
</div>

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


```

No	Kode Kategori	Nama Kategori
1	KTG001	Makanan
2	KTG002	Minuman
3	KTG003	Elektronik
4	KTG004	Peralatan Rumah
5	KTG005	Pakaian
6	KTG007	Alat Dapur
7	KTG006	Kardus

➤ Menu Supplier:

```
app > Http > Controllers > SupplierController.php > PHP > SupplierController > export_excel()
331 public function export_excel(): never
332 {
333     // Ambil data supplier yang akan diekspor
334     $suppliers = SupplierModel::select(columns: 'supplier_id', 'supplier_kode', 'supplier_nama', 'supplier_alamat')
335     ->orderBy(column: 'supplier_id')
336     ->get();
337
338     // Load library PhpSpreadsheet
339     $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
340     $sheet = $spreadsheet->getActiveSheet();
341
342     // Set header kolom
343     $sheet->setCellValue('A1', 'No');
344     $sheet->setCellValue('B1', 'Supplier ID');
345     $sheet->setCellValue('C1', 'Kode Supplier');
346     $sheet->setCellValue('D1', 'Nama Supplier');
347     $sheet->setCellValue('E1', 'Alamat Supplier');
348


<button onclick="modalAction('{{ url(path: '/supplier/import') }}')" class="btn btn-info">Import Supplier</button>
<a href="{{ url(path: '/supplier/export_excel') }}" class="btn btn-primary"><i class="fa fa-file-excel"></i> Export Supplier</a>
<button onclick="modalAction('{{ url(path: '/supplier/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
</div>

Route::post('import_ajax', action: [SupplierController::class, 'import_ajax']); // ajax import excel
Route::get('export_excel', [SupplierController::class, 'export_excel']); //export excel
});


```

No	Supplier ID	Kode Supplier	Nama Supplier
1	1 SUP001		PT Indomarco Prismatama
2	2 SUP002		PT Sumber Alfaria Trijaya
3	3 SUP003		PT Mayora Indah Tbk
4	5 SUP004		PT Ultra Prima Abadi
5	6 SUP005		PT Unilever Indonesia Tbk



2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
3. Submit kode untuk implementasi praktikum 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`

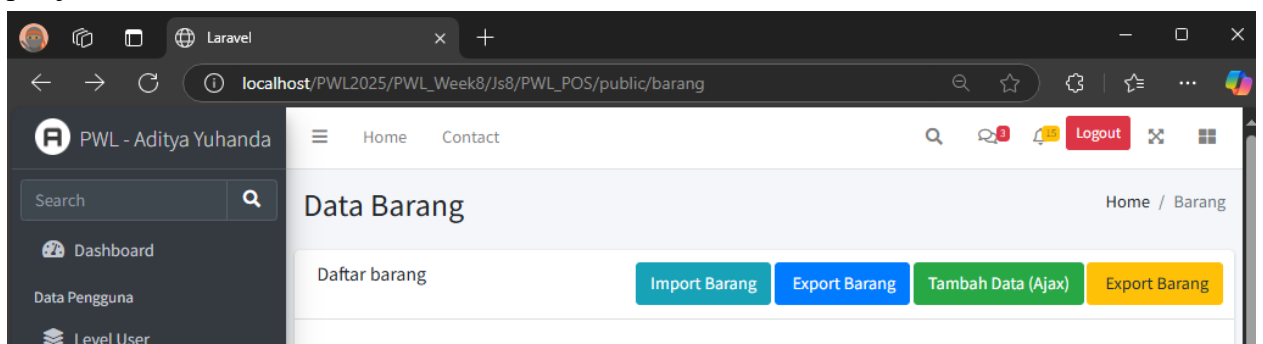
```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <style>
    body{
      font-family: "Times New Roman", Times, serif;
      margin: 6px 20px 5px 20px;
      line-height: 15px;
    }
    table {
      width:100%;
      border-collapse: collapse;
    }
    td, 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;
    }
    .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;
    }
  </style>
</head>
<body>
```



```
.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</h3>
    <table class="border-all">
        <thead>
            <tr>
                <th class="text-center">No</th>
                <th>Kode Barang</th>
                <th>Nama Barang</th>
                <th class="text-right">Harga Beli</th>
                <th class="text-right">Harga Jual</th>
                <th>Kategori</th>
            </tr>
        </thead>
        <tbody>
            @foreach($barang as $b)
                <tr>
                    <td class="text-center">{{ $loop->iteration }}</td>
                    <td>{{ $b->barang_kode }}</td>
                    <td>{{ $b->barang_nama }}</td>
                    <td class="text-right">{{ number_format($b->harga_beli, 0, ',', '.') }}</td>
                    <td class="text-right">{{ number_format($b->harga_jual, 0, ',', '.') }}</td>
                    <td>{{ $b->kategori->kategori_nama }}</td>
                </tr>
            @endforeach
        </tbody>
    </table>
</body>
</html>
```

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





No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	BRG001	Biskuit	3.000	5.000	Makanan
2	BRG002	Roti	5.000	7.000	Makanan
3	BRG011	Astor	3.000	5.000	Makanan
4	BRG003	Teh Botol	2.000	3.000	Minuman
5	BRG004	Kopi Instan	3.000	4.000	Minuman
6	BRG005	Mouse Wireless	100.000	150.000	Elektronik
7	BRG006	Keyboard Mechanical	250.000	350.000	Elektronik
8	BRG007	Sapu	15.000	20.000	Peralatan Rumah
9	BRG008	Pel	20.000	25.000	Peralatan Rumah
10	BRG009	Kaos Polos	40.000	50.000	Pakaian
11	BRG010	Jaket Hoodie	100.000	120.000	Pakaian

Tugas 3 – Implementasi Export PDF pada Laravel :

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

➤ Menu User:

```
app > Http > Controllers > UserController.php > PHP > UserController
475 public function export_pdf(): Response
476 {
477     $users = UserModel::select(columns: 'user_id', 'username', 'nama', 'level_id')
478     ->orderBy(column: 'user_id')
479     ->with(relations: 'level')
480     ->get();
481
482     // use Barryvdh/DomPDF\Facade\Pdf;
483     $pdf = Pdf::loadView('user.export_pdf', ['user' => $users]);
484     $pdf->setPaper('a4', 'portrait'); // set ukuran kertas dan orientasi
485     $pdf->setOption("isRemoteEnabled", true); // set true jika ada gambar dari url
486     $pdf->render();
487
488     return $pdf->stream('Data user ' . date(format: 'Y-m-d H:i:s') . '.pdf');
```

```
<a href="{{ url(path: '/user/export_pdf') }}" class="btn btn-warning"><i class="fa fa-file-pdf"></i> Export User</a>
</div>
```

```
Route::get(uri: 'export_pdf', action: [UserController::class, 'export_pdf']); //export pdf
```




➤ Menu Level:

```
app > Http > Controllers > LevelController.php > PHP > LevelController
0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
388 public function export_pdf(): Response
389 {
390     $level = LevelModel::select(columns: 'level_id', 'level_kode', 'level_nama')
391         ->orderBy(column: 'level_id')
392         ->get();
393
394     // use Barryvdh\DomPDF\Facade\Pdf;
395     $pdf = Pdf::loadView('level.export_pdf', ['level' => $level]);
396     $pdf->setPaper('a4', 'portrait'); // set ukuran kertas dan orientasi
397     $pdf->setOption("isRemoteEnabled", true); // set true jika ada gambar dari url
398     $pdf->render();
399
400     return $pdf->stream('Data Level ' . date(format: 'Y-m-d H:i:s') . '.pdf');
401 }
402 }
```

```
resources > views > level > export_pdf.blade.php > @html
1 <?php
2 <?php
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
4 <style>
5 {
6     font-family: "Times New Roman", Times, serif;
7     margin: 6px 20px 5px 20px;
8     line-height: 15px;
9 }
10 <table>
11 {
12     width: 100%;
13     border-collapse: collapse;
14 }
15 <tr>
16 {
17     padding: 4px 3px;
18 }
19 <th>
20 {
21     text-align: left;
22 }
23 <tbody>
24 {
25     <tr>
26     <td>
27     <td>
```

```
<button onclick="modalAction('{{ url(path: '/level/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
<a href="{{ url(path: '/level/export_pdf') }}" class="btn btn-warning"><i class="fa fa-file-pdf"></i> Export Level</a>
</div>

75
76 Route::get(uri: 'export_excel', action: [LevelController::class, 'export_excel']); //export excel
77 Route::get(uri: 'export_pdf', action: [LevelController::class, 'export_pdf']); //export pdf
```

➤ Menu Kategori:

```
app > Http > Controllers > KategoriController.php > PHP > KategoriController
374 public function export_pdf(): Response
375 {
376     $kategori = KategoriModel::select(columns: 'kategori_kode', 'kategori_nama')
377         ->orderBy(column: 'kategori_id')
378         ->get();
379
380     // use Barryvdh\DomPDF\Facade\Pdf;
381     $pdf = Pdf::loadView('kategori.export_pdf', ['kategori' => $kategori]);
382     $pdf->setPaper('a4', 'portrait'); // set ukuran kertas dan orientasi
383     $pdf->setOption("isRemoteEnabled", true); // set true jika ada gambar dari url
384     $pdf->render();
385
386     return $pdf->stream('Data Kategori ' . date(format: 'Y-m-d H:i:s') . '.pdf');
387 }
388 }
```

```
resources > views > kategori > export_pdf.blade.php > @html
1 <?php
2 <?php
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
4 <style>
5 {
6     font-family: "Times New Roman", Times, serif;
7     margin: 6px 20px 5px 20px;
8     line-height: 15px;
9 }
10 <table>
11 {
12     width: 100%;
13     border-collapse: collapse;
14 }
15 <tr>
16 {
17     padding: 4px 3px;
18 }
19 <th>
20 {
21     text-align: left;
22 }
23 <tbody>
24 {
25     <tr>
26     <td>
27     <td>
```

```
<button onclick="modalAction('{{ url(path: '/kategori/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
<a href="{{ url(path: '/kategori/export_pdf') }}" class="btn btn-warning"><i class="fa fa-file-pdf"></i> Export Kategori</a>
</div>

10
11 Route::get(uri: 'export_pdf', action: [KategoriController::class, 'export_pdf']); //export pdf
12
```




➤ Menu Supplier:

```
app > Http > Controllers > SupplierController.php > PHP Intelephense > SupplierController > export_pdf
391
392 public function export_pdf(): Response
393 {
394     $suppliers = SupplierModel::select(columns: ['supplier_id', 'supplier_kode', 'supplier_nama', 'supplier_alamat'])
395     ->orderBy(column: 'supplier_id')
396     ->get();
397
398     // use Barryvdh\DomPDF\Facade\Pdf;
399     $pdf = Pdf::loadView('supplier.export_pdf', ['supplier' => $suppliers]);
400     $pdf->setPaper('a4', 'portrait'); // set ukuran kertas dan orientasi
401     $pdf->setOption('isRemoteEnabled', true); // set true jika ada gambar dari url
402     $pdf->render();
403
404     return $pdf->stream('Data Supplier ' . date(format: 'Y-m-d H:i:s') . '.pdf');
405 }
```

```
<a href="{{ url(path: '/supplier/export_pdf') }}" class="btn btn-warning"><i class="fa fa-download"> Export Supplier</a>
</div>

Route::get(uri: 'export_pdf', action: [SupplierController::class, 'export_pdf']); //export pdf
);
```

2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
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
 - a) Pertama-tama modifikasi UserModel dengan menambahkan variable 'foto_profil' di \$fillable

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

- b) Kemudian membuat migration untuk menambahkan Field foto_profil dengan mengisi function up() down() nya

```
PS C:\laragon\www\pwl2025\pwl_week8\js8\pwl_pos> php artisan make:migration add_foto_profil_to_users_table --table=m_user
INFO Migration [C:\laragon\www\pwl2025\pwl_week8\js8\pwl_pos\database\migrations\2025_04_04_165828_add_foto_profil_to_users_table.php] created successfully.
```

```
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::table(table: 'm_user', callback: function (Blueprint $table): void {
15             $table->string(column: 'foto_profil')->nullable()->after(column: 'level_id');
16         });
17     }
18
19     /**
20      * Reverse the migrations.
21      */
22     public function down(): void
23     {
24         Schema::table(table: 'm_user', callback: function (Blueprint $table): void {
25             $table->dropColumn(columns: 'foto_profil');
26         });
27     }
28 }
```



```
PS C:\laragon\www\PWL2025\PWL_Week8\Js8\PWL_POS> php artisan migrate
```

```
INFO Running migrations.
```

```
2025_04_04_165828_add_foto_profil_to_users_table ..... 2,365ms DONE
```

c) Menambahkan symbolic link

```
PS C:\laragon\www\PWL2025\PWL_Week8\Js8\PWL_POS> php artisan storage:link
```

```
INFO The [C:\laragon\www\PWL2025\PWL_Week8\Js8\PWL_POS\public\storage] link has been connected to [C:\laragon\www\PWL2025\PWL_Week8\Js8\PWL_POS\storage\app\public].
```

d) Membuat halaman untuk Profil di views/user/profile.blade

```
resources > views > user > profile.blade.php > ...
1 @extends('layouts.template')
2
3 @section('content')
4 <div class="card">
5   <div class="card-header">
6     <h3 class="card-title">Profil Pengguna</h3>
7   </div>
8   <div class="card-body">
9     <div class="row">
10      <div class="col-md-4 text-center">
11        
13      </div>
14      <div class="col-md-8">
15        <form action="{{ url(path: '/user/update-foto') }}" method="POST" enctype="multipart/form-data" class="mt-3">
16          @csrf
17          <div class="form-group">
18            <div class="input-group">
19              <div class="custom-file">
20                <input type="file" class="custom-file-input" id="foto_profil" name="foto_profil" accept="image/*">
21                <label class="custom-file-label" for="foto_profil">Pilih Foto</label>
22              </div>
23            </div>
24            @error('foto_profil')
25              <small class="text-danger">{{ $message }}</small>
26            @enderror
27          </div>
28          <button type="submit" class="btn btn-primary">Upload Foto</button>
29        </form>
30      </div>
31    </div>
32  </div>
33 </section>
```

e) Membuat method untuk UserController profilePage() dan editPhoto()

```
401 /**
402  * Menampilkan halaman profile
403  */
404 0 references [0 overrides] | WinHttp | Refactor | Explain | X
405 public function profilePage(): Factory[mixed] | RedirectResponse | View
406 {
407   // Use laravel's auth system to get the current user
408   $user = auth()->user();
409
410   if (!$user) {
411     return redirect(to: '/login')->with(key: 'error', value: 'Silahkan login terlebih dahulu');
412   }
413
414   // Define breadcrumb as an OBJECT (note the (object) cast)
415   $breadcrumb = (object) [
416     'title' => 'Profile User',
417     'list' => ['Home', 'Profile']
418   ];
419
420   // Define page information
421   $page = (object) [
422     'title' => 'Profil Pengguna'
423   ];
424
425   // Set active menu
426   $activeMenu = 'profile';
427
428   return view(view: 'user.profile', data: compact(var_name: 'user', var_names: 'breadcrumb', 'page', 'activeMenu'));
429 }
430
431 /**
432  * Menampilkan halaman form edit photo
433  */
434 0 references [0 overrides] | WinHttp | Refactor | Explain | X
435 public function editPhoto(Request $request)
436 {
437   // Validasi file
438   $request->validate([
439     'foto_profil' => 'required|image|mimes:jpeg,png,jpg|max:2048',
440   ]);
441
442   try {
443     // Get the authenticated user
444     $user = auth()->user();
445
446     if (!$user) {
447       return redirect(to: '/login')->with('error', 'Silahkan login terlebih dahulu');
448     }
449
450     // Get the user ID
451     $userId = $user->user_id;
452
453     // Get the proper UserModel instance
454     $userModel = UserModel::find($userId);
455
456     if (!$userModel) {
457       return redirect(to: '/login')->with('error', 'User tidak ditemukan');
458     }
459
460     // Delete old photo if exists
461     if ($userModel->foto_profil && file_exists(storage_path('app/public/' . $userModel->foto_profil))) {
462       Storage::disk('public')->delete($userModel->foto_profil);
463     }
464
465     // Store new photo
466     $fileName = 'profile-' . $userId . '-' . time() . '.' . $request->foto_profil->extension();
467     $path = $request->foto_profil->storeAs('profiles', $fileName, 'public');
468
469     // Update user record using update() method
470     UserModel::where('user_id', $userId)->update([
471       'foto_profil' => $path
472     ]);
473
474     return redirect()->back()->with('success', 'Foto profile berhasil diperbarui');
475   } catch (\Exception $e) {
476     return redirect()->back()->with('error', 'Gagal mengupload foto: ' . $e->getMessage());
477   }
478 }
```



f) Selanjutnya tambahkan route pada web.php untuk fungsi foto profil ini

```
24 Route::middleware(middleware: ['auth'])->group(callback: function (): void {  
25     Route::get(uri: '/', action: [WelcomeController::class, 'index']);  
26     Route::get(uri: '/profile', action: [UserController::class, 'profile']);  
27     Route::post(uri: '/user/update-photo', action: [UserController::class, 'updatePhoto']);  
});
```

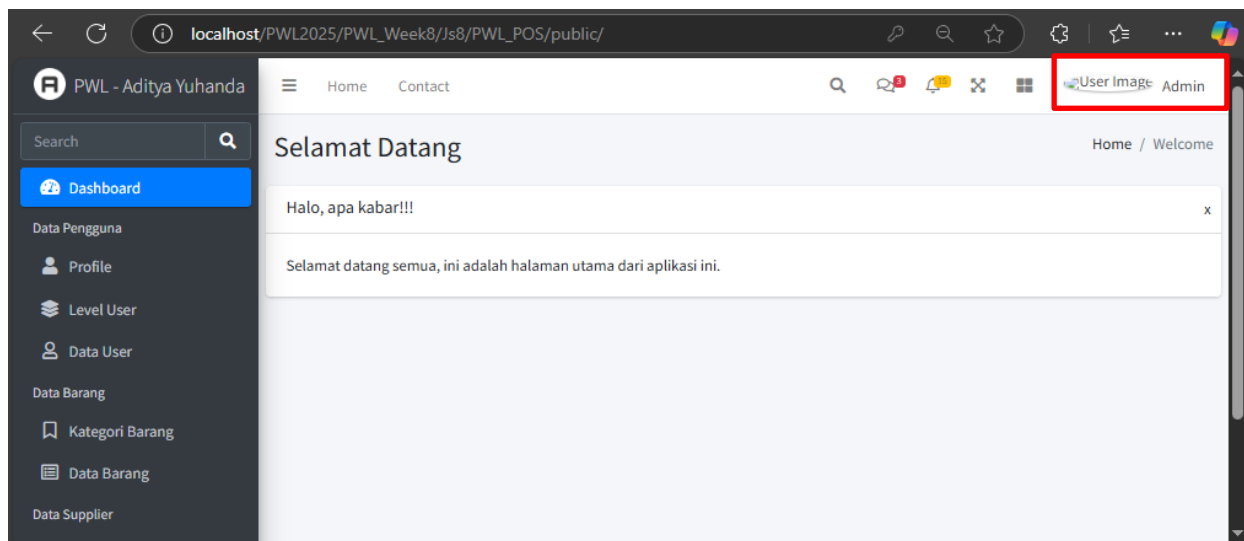
g) Selanjutnya modifikasi tampilan di /layouts/sidebar.blade.php untuk menambahkan element button profile

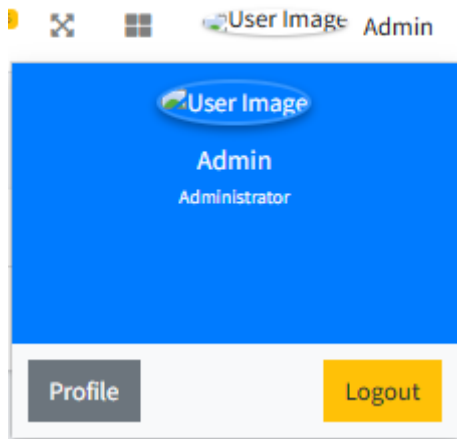
```
<li class="nav-item">  
    <a href="{{ url(path: '/profile') }}" class="nav-link {{ ($activeMenu == 'profile') ? 'active' : '' }}">  
        <i class="nav-icon fas fa-user"></i>  
        <p>Profile</p>  
    </a>  
</li>
```

h) Selanjutnya modifikasi header.blade.php untuk menampilkan foto profil, nama, dan level nya

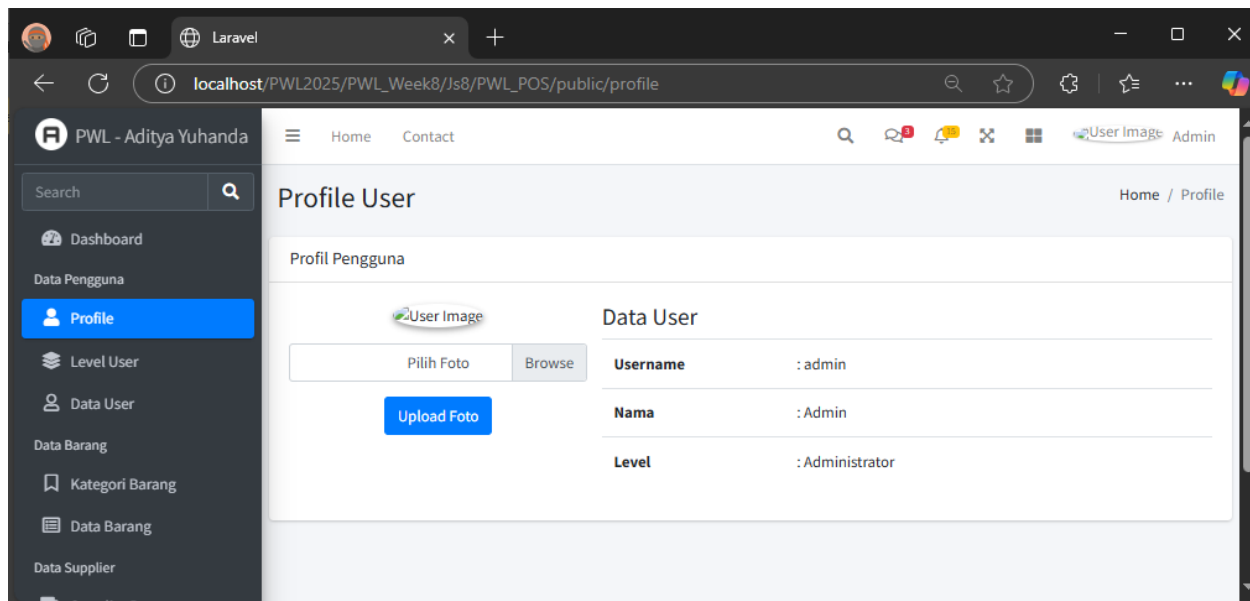
```
135 <!-- User profile dropdown -->  
136 <li class="nav-item dropdown user-menu">  
137     <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown">  
138           
141         <span class="d-none d-md-inline">{{ auth()->user()->nama }}</span>  
142     </a>  
143     <ul class="dropdown-menu dropdown-menu-lg dropdown-menu-right">  
144         <!-- User image -->  
145         <li class="user-header bg-primary">  
146               
149             <p>  
150                 {{ auth()->user()->nama }}  
151                 <small>{{ auth()->user()->level->level_nama ?? 'User' }}</small>  
152             </p>  
153         </li>  
154         <!-- Menu Footer -->  
155         <li class="user-footer">  
156             <a href="{{ url(path: '/profile') }}" class="btn btn-default btn-flat">Profile</a>  
157             <a href="#" class="btn btn-default btn-flat float-right" >  
158                 onclick="event.preventDefault(); document.getElementById('logout-form').submit();" >  
159                 Sign out  
160             </a>  
161             <form id="logout-form" action="{{ route(name: 'logout') }}" method="POST" style="display: none;">  
162                 @csrf  
163             </form>  
164         </li>  
165     </ul>  
166 </li>
```

i) Kemudian hasil tampilannya login sebagai admin akan seperti :

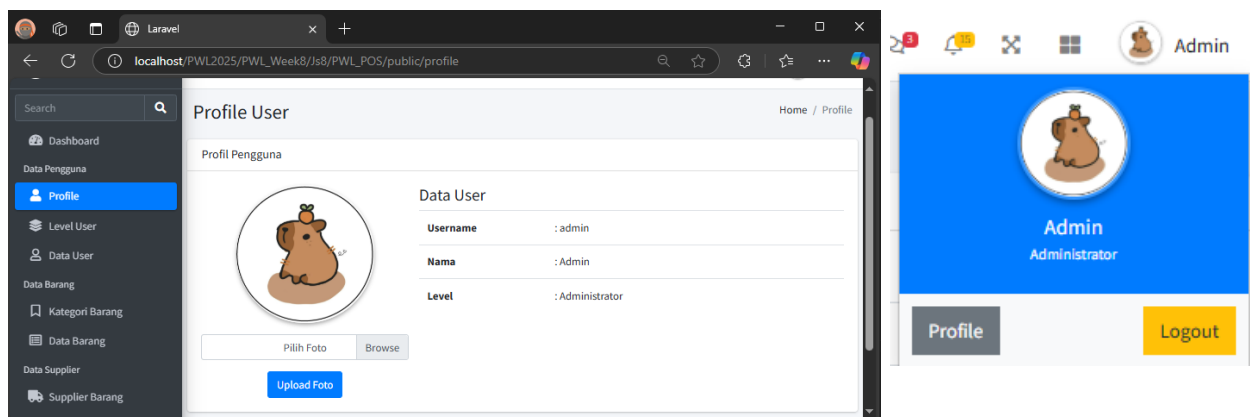




j) Tampilan menu profile :



Upload Foto Capybara :



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 ****