Nama : Adinda Ivanka Maysanda Putri

Kelas : SIB 2B

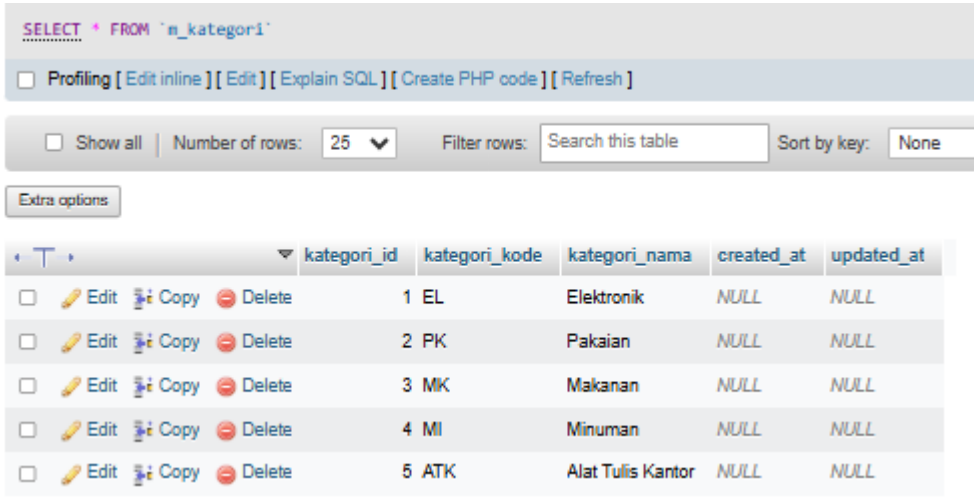
NIM : 2341760058

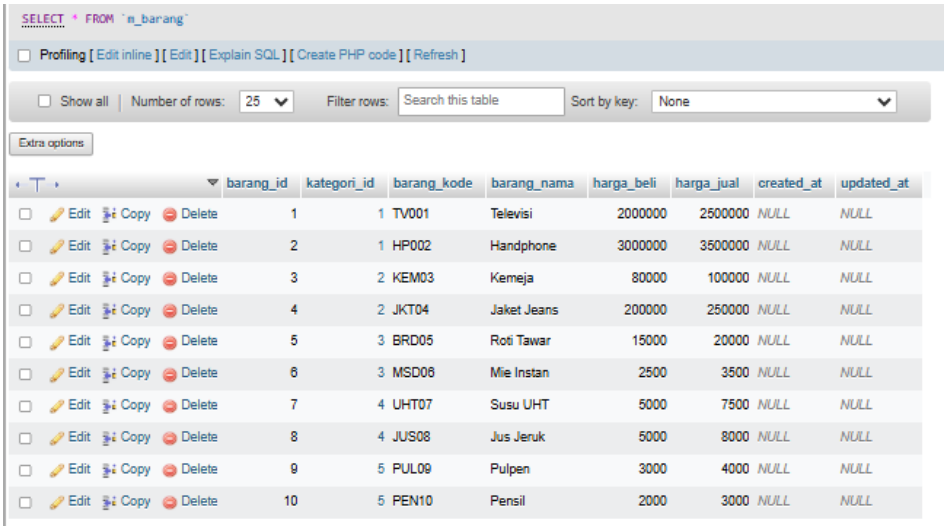
JOBSHEET 8

File Import dan Export ke PDF dan Excel pada Laravel

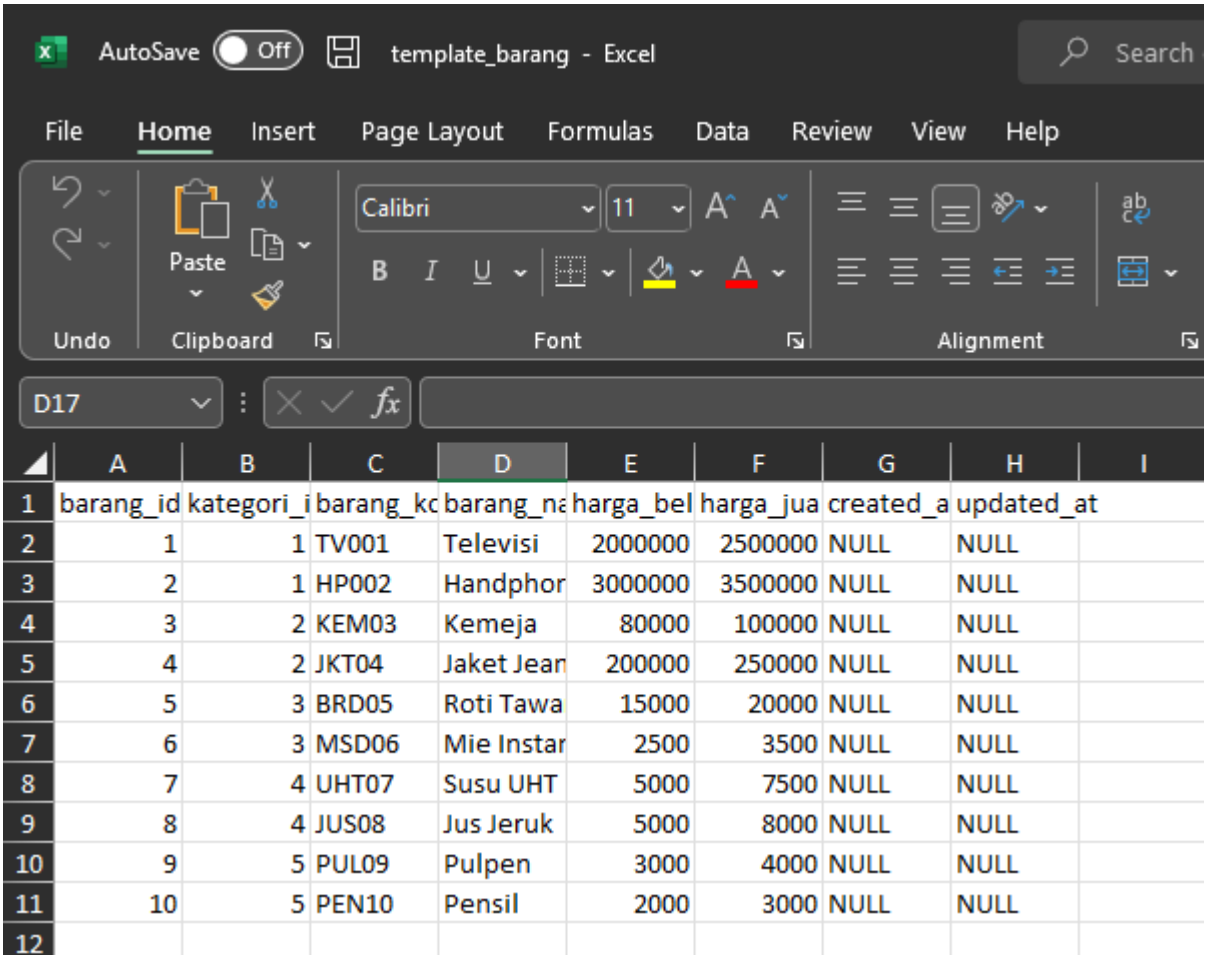
Praktikum 1 – Implementasi Upload File untuk import data:

1. Kita buka database yang sudah kita buat sebelumnya, dan kita cek tabel m_kategori dan m_barang





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	G	H	I
1	barang_id	kategori_i	barang_kc	barang_na	harga_bel	harga_jua	created_a	updated_at	
2	1	1	TV001	Televisi	2000000	2500000	NULL	NULL	
3	2	1	HP002	Handphor	3000000	3500000	NULL	NULL	
4	3	2	KEM03	Kemeja	80000	100000	NULL	NULL	
5	4	2	JKT04	Jaket Jean	200000	250000	NULL	NULL	
6	5	3	BRD05	Roti Tawa	15000	20000	NULL	NULL	
7	6	3	MSD06	Mie Instar	2500	3500	NULL	NULL	
8	7	4	UHT07	Susu UHT	5000	7500	NULL	NULL	
9	8	4	JUS08	Jus Jeruk	5000	8000	NULL	NULL	
10	9	5	PUL09	Pulpen	3000	4000	NULL	NULL	
11	10	5	PEN10	Pensil	2000	3000	NULL	NULL	
12									

3. Selanjutnya kita modifikasi view pada barang/index.blade.php untuk bisa menambahkan tombol menambah form untuk upload untuk import data barang
4. Selanjutnya kita buat view untuk form upload/import file excel dan download file template_barang.xlsx. Kita beri nama file dengan nama barang/import.blade.php

```
<form action="{{ url('/barang/import_ajax') }}" method="POST" id="form-import" enctype="multipart/form-data">
    @csrf
    <div id="modal-master" class="modal-dialog modal-lg" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Import Data Barang</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <div class="form-group">
                    <label>Download Template</label>
                    <a href="{{ asset('template_barang.xlsx') }}" class="btn btn-info btn-sm" download>
                        <i class="fa fa-file-excel"></i>Download
                    </a>
                    <small id="error-kategori_id" class="error-text form-text text-danger"></small>
                </div>
```

```
            <div class="form-group">
                <label>Pilih File</label>
                <input type="file" name="file_barang" id="file_barang" class="form-control" required>
                <small id="error-file_barang" class="error-text form-text text-danger"></small>
            </div>
        </div>
        <div class="modal-footer">
            <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
            <button type="submit" class="btn btn-primary">Upload</button>
        </div>
    </div>
  </div>
</form>
<script>
  $(document).ready(function() {
    $("#form-import").validate({
        rules: {
            file_barang: { required: true, extension: "xlsx" },
        },
        submitHandler: function(form) {
            var formData = new FormData(form); // Jadikan form ke FormData untuk menghandle file
            $.ajax({
                url: form.action,
                type: form.method,
                data: formData, // Data yang dikirim berupa FormData
                processData: false, // setting processData dan contentType ke false, untuk menghandle file
                contentType: false,
                success: function(response) {
                    if (response.status) { // jika sukses
                        $('#myModal').modal('hide');
                        Swal.fire({
                            icon: 'success',
                            title: 'Berhasil',
                            text: response.message
                        });
                        tableBarang.ajax.reload(); // reload datatable
                    } else { // jika error
                        $('.error-text').text('');
                        $.each(response.msgField, function(prefix, val) {
                            $('#error-' + prefix).text(val[0]);
                        });
                        Swal.fire({
                            icon: 'error',
                            title: 'Terjadi Kesalahan',
                            text: response.message
                        });
                    }
                }
            });
            return false;
```

```
        },
        errorElement: 'span',
        errorPlacement: function(error, element) {
            error.addClass('invalid-feedback');
            element.closest('.form-group').append(error);
        },
        highlight: function(element, errorClass, validClass) {
            $(element).addClass('is-invalid');
        },
        unhighlight: function(element, errorClass, validClass) {
            $(element).removeClass('is-invalid');
        }
    });
  });
</script>
```

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

```
Route::middleware(['authorize:ADM,MNG,STF'])->group(function () { //hanya level admin,manager,staff yang dapat mengakses menu barang
    Route::get('/barang', [BarangController::class, 'index']);              // Menampilkan halaman awal barang
    Route::post('/barang/list', [BarangController::class, 'list']);         // Menampilkan data barang dalam bentuk json untuk datatables
    Route::get('/barang/create', [BarangController::class, 'create']);      // Menampilkan halaman form tambah barang
    Route::post('/barang', [BarangController::class, 'store']);             // Menyimpan data barang baru
    Route::get('/barang/create_ajax', [BarangController::class, 'create_ajax']);    // Menampilkan halaman form tambah barang Ajax
    Route::post('/barang/ajax', [BarangController::class, 'store_ajax']);          // Menyimpan data barang baru Ajax
    Route::get('/barang/{id}', [BarangController::class, 'show']);          // Menampilkan detail barang
    Route::get('/barang/{id}/edit', [BarangController::class, 'edit']);     // Menampilkan halaman form edit barang
    Route::put('/barang/{id}', [BarangController::class, 'update']);        // Menyimpan perubahan data barang
    Route::get('/barang/{id}/show_ajax', [BarangController::class, 'show_ajax']);    // Menampilkan halaman form barang Ajax
    Route::get('/barang/{id}/edit_ajax', [BarangController::class, 'edit_ajax']);    // Menampilkan halaman form edit barang Ajax
    Route::put('/barang/{id}/update_ajax', [BarangController::class, 'update_ajax']);  // Menyimpan perubahan data barang Ajax
    Route::get('/barang/{id}/delete_ajax', [BarangController::class, 'confirm_ajax']); // Untuk tampilkan form confirm delete barang Ajax
    Route::delete('/barang/{id}/delete_ajax', [BarangController::class, 'delete_ajax']); // Untuk hapus data barang Ajax
    Route::delete('/barang/{id}', [BarangController::class, 'destroy']);    // Menghapus data barang
    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
  });
```

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

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 8> composer require phpoffice/phpspreadsheet
./composer.json has been updated
Running composer update phpoffice/phpspreadsheet
Loading composer repositories with package information
Updating dependencies
Lock file operations: 5 installs, 0 updates, 0 removals
  - Locking composer/pcre (3.3.2)
  - Locking maennchen/zipstream-php (3.1.1)
  - Locking markbaker/complex (3.0.2)
  - Locking markbaker/matrix (3.0.1)
  - Locking phpoffice/phpspreadsheet (4.1.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 5 installs, 0 updates, 0 removals
```

7. Selanjutnya kita modifikasi file BarangController.php untuk memproses data

```php
<?php

namespace App\Http\Controllers;

use App\Models\KategoriModel;
use App\Models\BarangModel;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use PhpOffice\PhpSpreadsheet\IOFactory;
use PhpOffice\PhpSpreadsheet\Spreadsheet;
use Yajra\DataTables\Facades\DataTables;

class BarangController extends Controller
{
    //Menampilkan halaman awal user
    public function index()
    {
        $breadcrumb = (object) [
            'title' => 'Daftar Barang',
            'list'  => ['Home', 'Barang']
        ];

        $page = (object) [
            'title' => 'Daftar barang yang terdaftar dalam sistem'
        ];

        $activeMenu = 'barang'; // set menu yang sedang aktif

        $kategori = KategoriModel::select('kategori_id', 'kategori_nama')->get();  //ambil data kategori untuk filter kategori

        return view('barang.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'kategori' => $kategori, 'activeMenu' =>
$activeMenu]);
    }

    // Ambil data barang dalam bentuk json untuk datatables
    public function list(Request $request)
    {
```

```
        $barang = BarangModel::select('barang_id', 'kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
            ->with(['kategori']); // Menggunakan array untuk multiple with

        // Filter data barang berdasarkan kategori_id
        if ($request->kategori_id) {
            $barang->where('kategori_id', $request->kategori_id);
        }

        return DataTables::of($barang)
            // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
            ->addIndexColumn()
            ->addColumn('aksi', function ($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 Anda 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']) // memberitahu bahwa kolom aksi adalah html
            ->make(true);
    }

    // Menampilkan halaman form tambah barang
    public function create()
    {
        $breadcrumb = (object) [
            'title' => 'Tambah Barang',
            'list'  => ['Home', 'Barang', 'Tambah']
        ];

        $page = (object) [
            'title' => 'Tambah Barang baru'
        ];

        $kategori = KategoriModel::all(); // ambil data kategori untuk ditampilkan di form
        $activeMenu = 'barang';       // set menu yang sedang aktif

        return view('barang.create', [
            'breadcrumb' => $breadcrumb,
            'page'       => $page,
            'kategori'   => $kategori,
            'activeMenu' => $activeMenu
```

```
        ]);
    }

    public function store(Request $request)
    {
        // Validasi input
        $request->validate([
            'barang_kode' => 'required|string|max:10|unique:m_barang,barang_kode',
            'barang_nama' => 'required|string|max:100',  // barang_nama harus diisi, berupa string, dan maksimal 100 karakter
            'harga_beli'  => 'required|numeric',
            'harga_jual'  => 'required|numeric',
            'kategori_id' => 'required|numeric',
        ]);

        // Menyimpan data user baru
        BarangModel::create([
            'barang_kode' => $request->barang_kode,
            'barang_nama' => $request->barang_nama,
            'harga_beli'  => $request->harga_beli,
            'harga_jual'  => $request->harga_jual,
            'kategori_id' => $request->kategori_id,
        ]);

        // Redirect ke halaman barang dengan pesan sukses
        return redirect('/barang')->with('success', 'Data barang berhasil disimpan');
    }

    // Menampilkan detail barang
    public function show(string $id)
    {
        $barang = BarangModel::with(['kategori'])->find($id);

        $breadcrumb = (object) [
            'title' => 'Detail barang',
            'list'  => ['Home', 'barang', 'Detail']
        ];

        $page = (object) [
            'title' => 'Detail barang'
        ];

        $activeMenu = 'barang'; // set menu yang sedang aktif

        return view('barang.show', [
            'breadcrumb' => $breadcrumb,
            'page' => $page,
            'barang' => $barang,
            'activeMenu' => $activeMenu
        ]);
    }
```

```php
// Menampilkan halaman form edit barang
public function edit(string $id)
{
    $barang = BarangModel::find($id);
    $kategori = KategoriModel::all();

    $breadcrumb = (object) [
        'title' => 'Edit Barang',
        'list'  => ['Home', 'Barang', 'Edit']
    ];

    $page = (object) [
        'title' => 'Edit barang'
    ];

    $activeMenu = 'barang'; // set menu yang sedang aktif

    return view('barang.edit', [
        'breadcrumb' => $breadcrumb,
        'page' => $page,
        'barang' => $barang,
        'kategori' => $kategori,
        'activeMenu' => $activeMenu
    ]);
}

// Menyimpan perubahan data user
public function update(Request $request, string $id)
{
    $request->validate([
        'barang_kode' => 'required|string|max:10|unique:m_barang,barang_kode,' . $id . ',barang_id',
        'barang_nama' => 'required|string|max:100',
        'harga_beli'  => 'required|numeric',
        'harga_jual'  => 'required|numeric',
        'kategori_id' => 'required|numeric',
    ]);

    BarangModel::find($id)->update([
        'barang_kode' => $request->barang_kode,
        'barang_nama' => $request->barang_nama,
        'harga_beli'  => $request->harga_beli,
        'harga_jual'  => $request->harga_jual,
        'kategori_id' => $request->kategori_id,
    ]);

    return redirect('/barang')->with('success', 'Data barang berhasil diubah');
}

// Menghapus data barang
```

```php
    public function destroy(string $id)
    {
        $check = BarangModel::find($id);
        if (!$check) {    // untuk mengecek apakah data barang dengan id yang dimaksud ada atau tidak
            return redirect('/barang')->with('error', 'Data barang tidak ditemukan');
        }

        try {
            BarangModel::destroy($id);    // Hapus data barang

            return redirect('/barang')->with('success', 'Data barang berhasil dihapus');
        } catch (\Illuminate\Database\QueryException $e) {
            // Jika terjadi error ketika menghapus data, redirect kembali ke halaman dengan membawa pesan error
            return redirect('/barang')->with('error', 'Data barang gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini');
        }
    }

    public function create_ajax()
    {
        $kategori = KategoriModel::select('kategori_id', 'kategori_nama')->get();
        return view('barang.create_ajax')->with('kategori', $kategori); // Mengembalikan view create_ajax.blade.php dengan membawa data kategori
    }

    public function store_ajax(Request $request)
    {
        // cek apakah request berupa ajax
        if ($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'],
                '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, // response status, false: error/gagal, true: berhasil
                    'message' => 'Validasi Gagal',
                    'msgField' => $validator->errors(), // pesan error validasi
                ]);
            }

            BarangModel::create($request->all());
            return response()->json([
```

```
            'status' => true,
            'message' => 'Data barang berhasil disimpan'
        ]);
    }
    redirect('/');
}

public function edit_ajax(string $id)
{
    $barang = BarangModel::find($id);
    $kategori = KategoriModel::select('kategori_id', 'kategori_nama')->get();
    return view('barang.edit_ajax', ['barang' => $barang, 'kategori' => $kategori]);
}

public function update_ajax(Request $request, $id)
{
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'kategori_id' => 'required|integer',
            'barang_kode' => 'required|string|min:3|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'
            ]);
        }
```

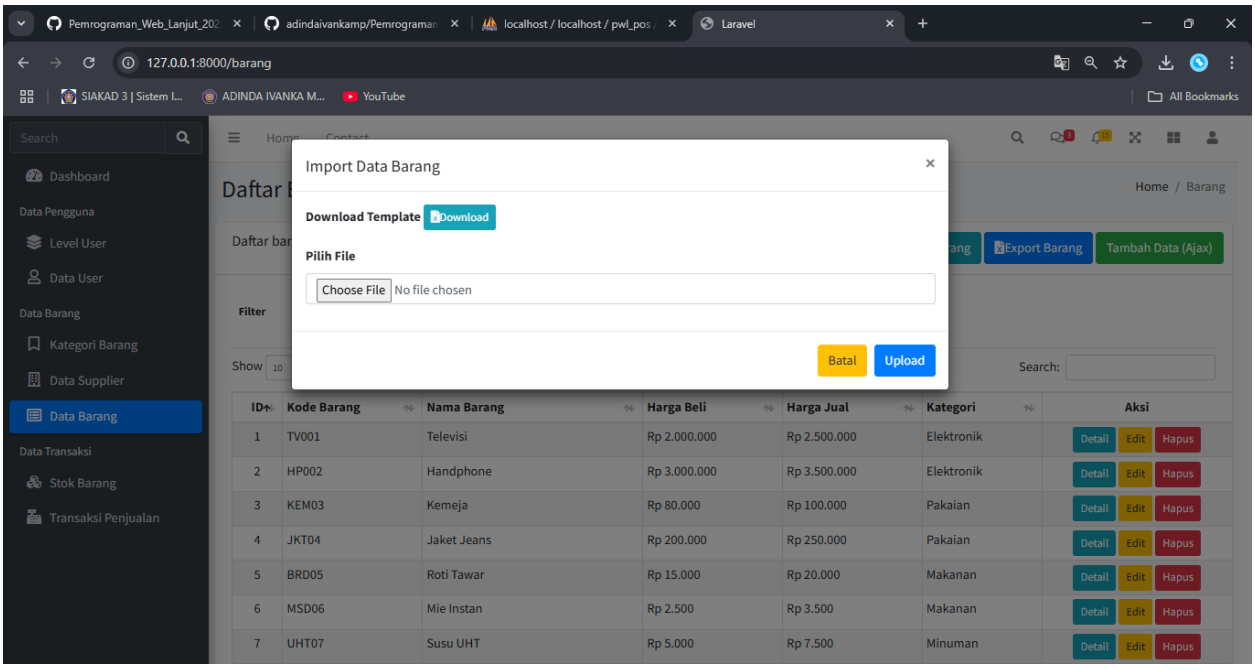
```
    }
    return redirect('/');
}

public function confirm_ajax(string $id)
{
    $barang = BarangModel::find($id);
    return view('barang.confirm_ajax', ['barang' => $barang]);
}

public function delete_ajax(Request $request, $id)
{
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantsJson()) {
        $barang = BarangModel::find($id);
        if ($barang) {
            try {
                $barang->delete();
                return response()->json([
                    'status' => true,
                    'message' => 'Data berhasil dihapus'
                ]);
            } catch (\Illuminate\Database\QueryException $e) {
                return response()->json([
                    'status' => false,
                    'message' => 'Data barang gagal dihapus karena masih terkait data lain'
                ]);
            }
        }

        return response()->json([
            'status' => false,
            'message' => 'Data tidak ditemukan'
        ]);
    }

    return redirect('/');
}

public function show_ajax(string $id)
{
    $barang = barangModel::find($id);
    $kategori = KategoriModel::select('kategori_id', 'kategori_nama')->get();
    return view('barang.show_ajax', ['barang' => $barang, 'kategori' => $kategori]);
}

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

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

        try {
            $reader = IOFactory::createReader('Xlsx');
            $reader->setReadDataOnly(true);
            $spreadsheet = $reader->load($file->getRealPath());
            $sheet = $spreadsheet->getActiveSheet();
            $data = $sheet->toArray(null, false, true, true); // Kolom dengan huruf (A, B, C, ...)

            $inserted = 0;
            foreach ($data as $key => $row) {
                // Lewati baris header (misal baris ke-1)
                if ($key === 1) continue;

                // Contoh: pastikan kolom sesuai dengan template file
                $barangKode = $row['A'] ?? null;
                $barangNama = $row['B'] ?? null;
                $hargaBeli = $row['C'] ?? null;
                $hargaJual = $row['D'] ?? null;
                $kategoriId = $row['E'] ?? null;

                // Validasi minimal sebelum insert
                if ($barangKode && $barangNama && $hargaBeli && $hargaJual && $kategoriId) {
                    // Cek duplikat berdasarkan barang_kode
                    $existing = BarangModel::where('barang_kode', $barangKode)->first();
                    if (!$existing) {
                        BarangModel::create([
                            'barang_kode' => $barangKode,
                            'barang_nama' => $barangNama,
                            'harga_beli'  => $hargaBeli,
                            'harga_jual'  => $hargaJual,
                            'kategori_id' => $kategoriId,
                        ]);
```

```
                    $inserted++;
                }
            }
        }

        return response()->json([
            'status' => true,
            'message' => "Import berhasil. $inserted data ditambahkan."
        ]);
    } catch (\Exception $e) {
        return response()->json([
            'status' => false,
            'message' => 'Terjadi kesalahan saat memproses file: ' . $e->getMessage()
        ]);
    }
  }

  return redirect('/');
}

public function export_excel()
 {
    // ambil data barang yang akan di export
    $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
        ->orderBy('kategori_id')
        ->with('kategori')
        ->get();

    // load library excel
    $spreadsheet = new Spreadsheet();
    $sheet = $spreadsheet->getActiveSheet(); // ambil sheet yang aktif

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

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

    $no = 1; // nomor data dimulai dari 1
    $baris = 2; // baris data dimulai dari baris ke 2

    foreach ($barang as $key => $value) {
        $sheet->setCellValue('A' . $baris, $no);
        $sheet->setCellValue('B' . $baris, $value->barang_kode);
        $sheet->setCellValue('C' . $baris, $value->barang_nama);
        $sheet->setCellValue('D' . $baris, $value->harga_beli);
        $sheet->setCellValue('E' . $baris, $value->harga_jual);
```

```
            $sheet->setCellValue('F' . $baris, $value->kategori->kategori_nama); // ambil nama kategori

            $baris++;
            $no++;
        }

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

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

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

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

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

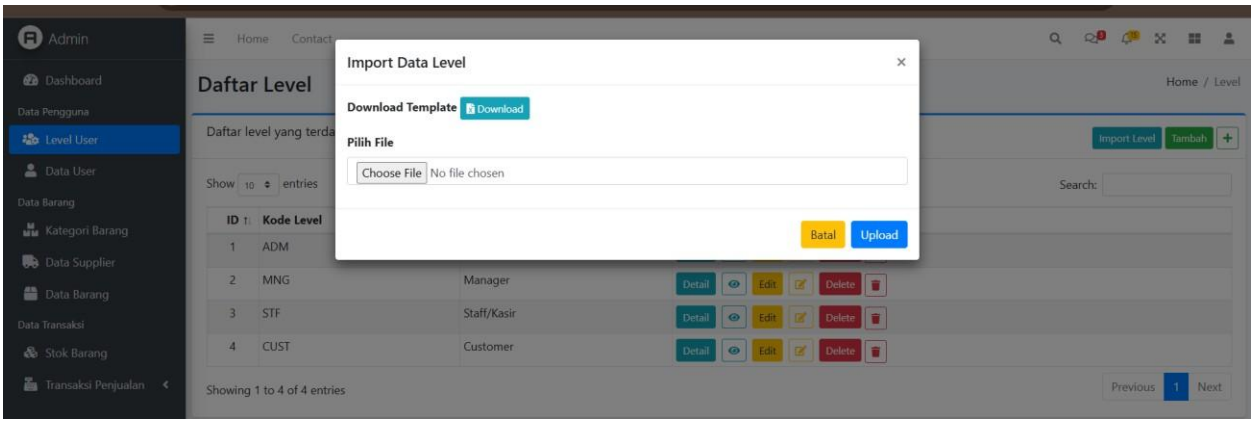
8. Sekarang kita coba jalankan browser dan klik tombol import pada menu Barang

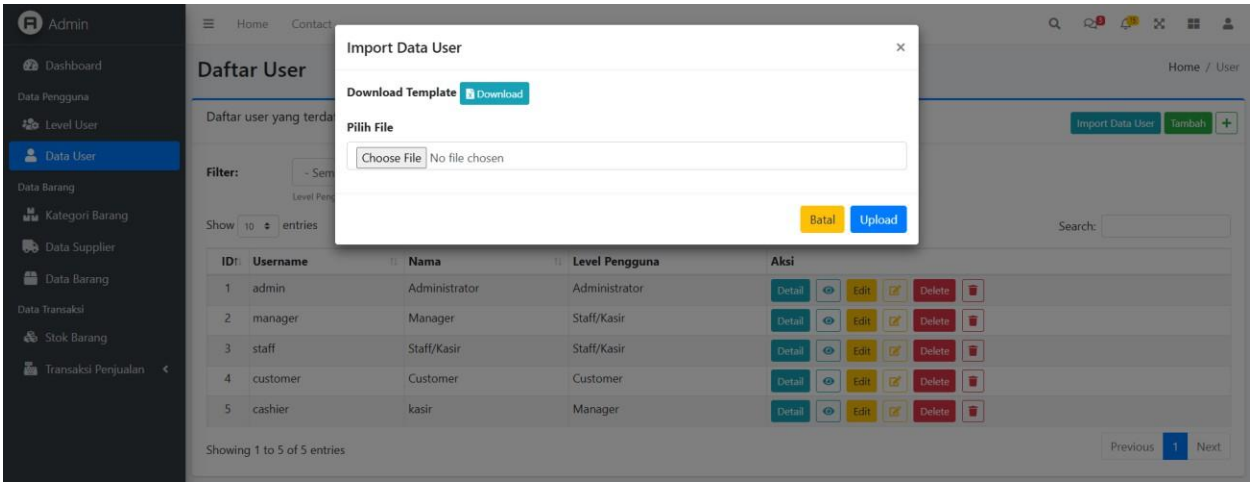


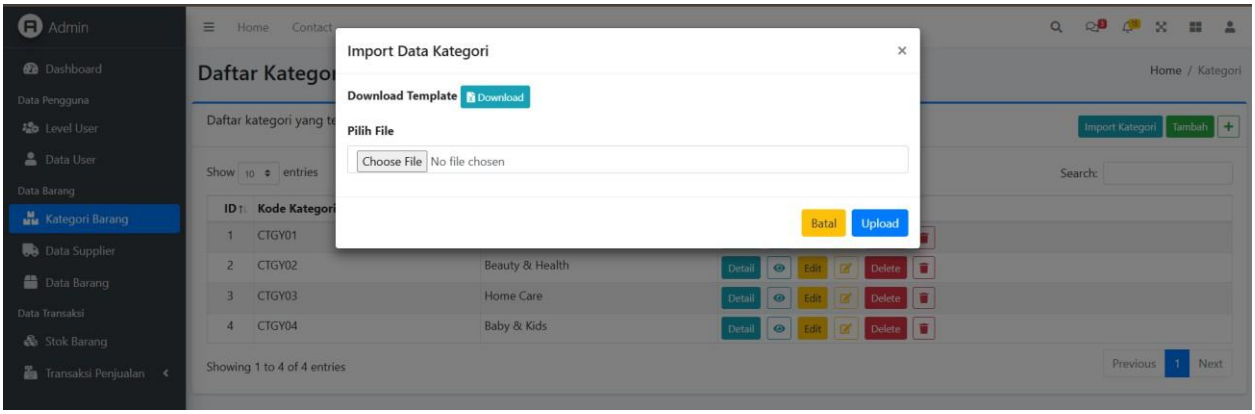
9. Kita upload template data yang sudah kita siapkan, dan amati apa yang terjadi.

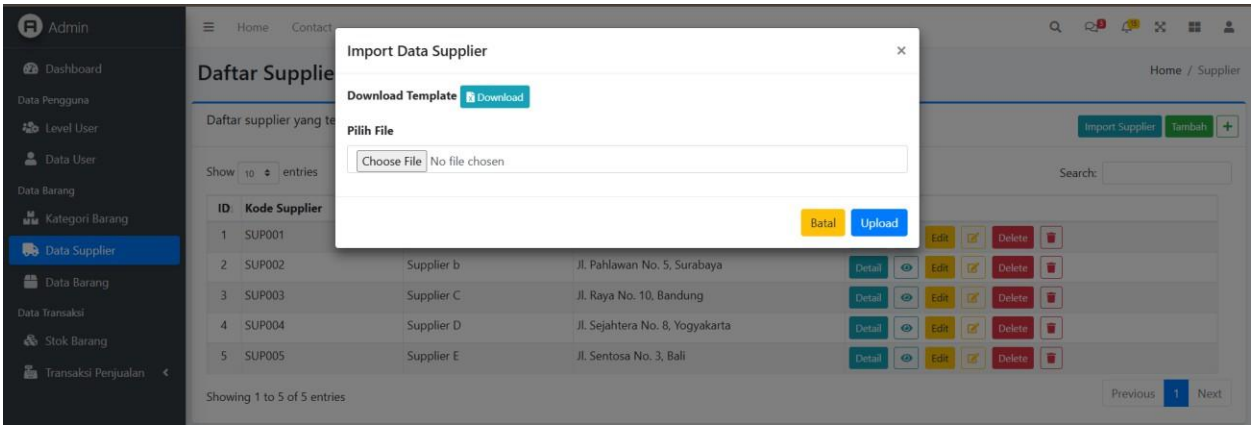
Tugas 1 – Implementasi File Upload untuk Import Data:

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









2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
    - Langkah pertama yang dilakukan adalah membuat file view untuk form import data Excel pada masing-masing menu, misalnya import.blade.php di folder views/barang. Setelah itu, file index pada setiap menu dimodifikasi dengan menambahkan tombol *Import* yang memanggil modal import via AJAX. Kemudian, dua function ditambahkan di controller, yaitu import() untuk menampilkan form, dan import_ajax() untuk memproses file yang diunggah menggunakan PhpSpreadsheet. Terakhir, instal library phpoffice/phpspreadsheet melalui perintah composer require phpoffice/phpspreadsheet, dan mendaftarkan route import dan import_ajax di file web.php untuk mengaktifkan fitur tersebut.
3. Submit kode untuk impementasi prakktikum 1 pada repository github kalian.

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

```
<button onclick="modalAction('{{ url('/barang/import') }}')" class="btn btn-info">Import Barang</button>
<a href="{{ url('/barang/export_excel') }}" class="btn btn-primary btn-sm py-1 px-2 mt-1"><i class="fa fa-fileexcel"></i> Export Barang</a>
<button onclick="modalAction('{{ url('/barang/create_ajax') }}')" class="btn btn-success">Tambah Data (Ajax)</button>
```

   Hal ini kita lakukan karena tombol Tambah Data sudah tidak kita gunakan karena kita sudah menggunakan tombol Tambah Data (Ajax).
2. Kemudian kita tambahkan route pada route/web.php untuk bisa memproses export excel

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

3. Selanjutnya kita tambahkan fungsi export_excel() pada file BarangController.php

```php
public function export_excel()
{

}
```

4. Kita ambil data barang yang akan kita export ke excel (tentu dengan menyertakan relasi kategori barang)

```php
public function export_excel()
{
    // ambil data barang yang akan di export
    $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
    ->orderBy('kategori_id')
    ->with('kategori')
    ->get();
}
```

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

```php
// load library excel
$spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
$sheet = $spreadsheet->getActiveSheet(); // ambil yang active

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

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

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

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

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

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

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

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

            $sheet->setTitle('Data Barang'); // set title sheet
            $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
            $filename = 'Data Barang ' . date('Y-m-d H:i:s') . '.xlsx';
            header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
            header('Content-Disposition: attachment;filename="' . $filename . '"');
            header('Cache-Control: max-age=0');
            header('Cache-Control: max-age=1');
            header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
            header('Last-Modified: ' . gmdate('D, d M Y H:i:s') . ' GMT');
            header('Cache-Control: cache, must-revalidate');
            header('Pragma: public');

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

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

Tugas 2 – Implementasi File Export Excel:

---

1. Silahkan implementasikan praktikum 2 pada project kalian masing-masing untuk semua menu
2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
    - Langkah pertama dimulai dengan mengedit file index pada setiap menu untuk menambahkan tombol *Export* sebagai fitur tambahan. Setelah itu, route baru ditambahkan di web.php untuk menangani proses export ke Excel. Kemudian, dibuat function export() di masing-masing controller yang bertugas mengambil data dari database dan mengubahnya menjadi file Excel menggunakan PhpSpreadsheet. Saat tombol *Export* ditekan, data langsung diunduh dalam bentuk file Excel oleh pengguna.
3. Submit kode untuk impementasi prakktikum 2 pada repository github kalian.

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

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 8 (PWL_POS)> composer require barryvdh/laravel-dompdf
./composer.json has been updated
Running composer update barryvdh/laravel-dompdf
Loading composer repositories with package information
Updating dependencies
Lock file operations: 6 installs, 0 updates, 0 removals
  - Locking barryvdh/laravel-dompdf (v3.1.1)
  - Locking dompdf/dompdf (v3.1.0)
  - Locking dompdf/php-font-lib (1.0.1)
  - Locking dompdf/php-svg-lib (1.0.0)
  - Locking masterminds/html5 (2.9.0)
  - Locking sabberworm/php-css-parser (v8.8.0)
Writing lock file
```

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

```
<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

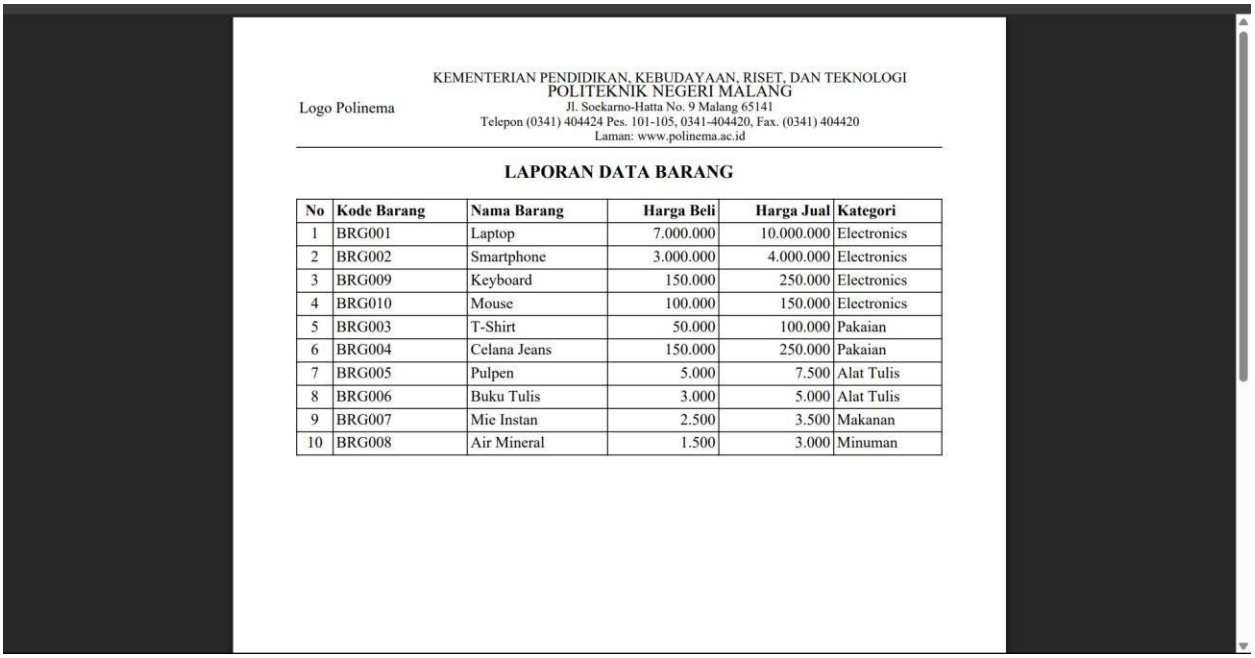
```
Route::get('/export_excel', [BarangController::class, 'export_excel']); // export excel
Route::get('/export_pdf', [BarangController::class, 'export_pdf']); // export pdf
```

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

6. Selanjutnya kita buat view untuk me-generate html untuk tampilan pdf yang akan kita sajikan. View berada di barang/export_pdf.blade.php
7. Selanjutnya, Kita coba untuk melakukan proses download export pdf. Amati dan pelajari…!!!



Logo Polinema

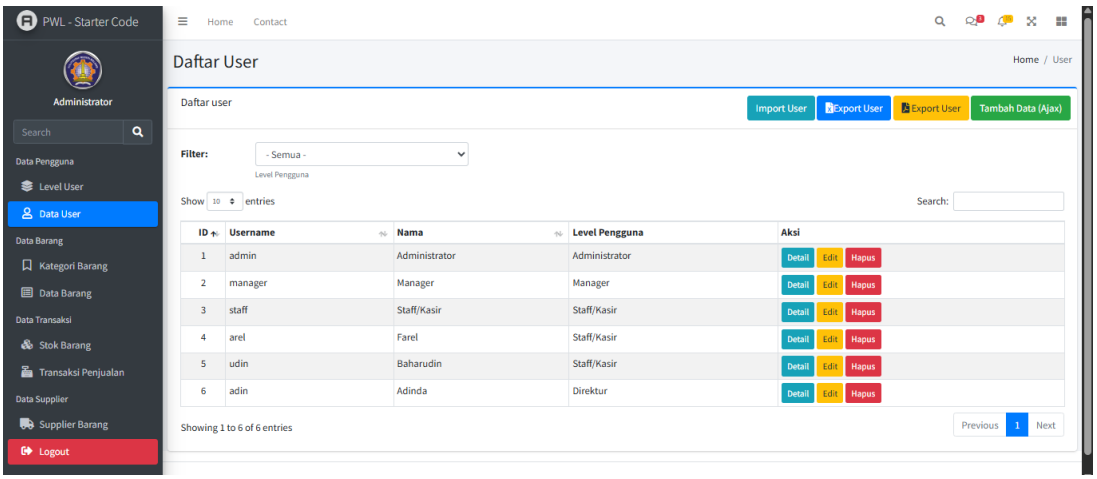
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
Jl. Soekarno-Hatta No. 9 Malang 65141
Telepon (0341) 404424 Pes. 101-105, 0341-404420, Fax. (0341) 404420
Laman: www.polinema.ac.id

LAPORAN DATA BARANG

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	BRG001	Laptop	7.000.000	10.000.000	Electronics
2	BRG002	Smartphone	3.000.000	4.000.000	Electronics
3	BRG009	Keyboard	150.000	250.000	Electronics
4	BRG010	Mouse	100.000	150.000	Electronics
5	BRG003	T-Shirt	50.000	100.000	Pakaian
6	BRG004	Celana Jeans	150.000	250.000	Pakaian
7	BRG005	Pulpen	5.000	7.500	Alat Tulis
8	BRG006	Buku Tulis	3.000	5.000	Alat Tulis
9	BRG007	Mie Instan	2.500	3.500	Makanan
10	BRG008	Air Mineral	1.500	3.000	Minuman

Tugas 3 – Implementasi Export PDF pada Laravel :

1. Silahkan implementasikan export pdf pada project kalian masing-masing untuk semua menu
   - https://github.com/adindaivankamp/Pemrograman-Web-Lanjut/tree/main/Minggu%208



2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
   - Langkah pertama adalah menambahkan tombol Export Pdf di halaman index.blade.php, biasanya diletakkan di bagian atas tabel data agar mudah diakses oleh pengguna. Setelah itu, kita mendefinisikan route baru pada web.php yang mengarah ke function export_pdf() di controller terkait. Kemudian kita membuat function export_pdf() di controller untuk mengambil data dari database dan me-render-nya ke file PDF menggunakan library seperti dompdf. Terakhir, kita membuat file view khusus untuk

PDF, misalnya pdf.blade.php, dengan desain sederhana dan styling CSS manual agar hasil PDF terlihat rapi dan sesuai kebutuhan.

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

Tugas 4 – Implementasi Upload File Gambar :

1. Silahkan implementasikan fitur upload file untuk mengubah foto profile di project web kalian

2. Jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
    - Pertama, ditambahkan menu profile di header navbar yang mengarah ke halaman profil user. Kemudian, dibuat migration untuk menambahkan kolom user_profile pada tabel users. Setelah itu, model User diperbarui dengan menambahkan field tersebut ke dalam $fillable. Selanjutnya, dibuat view profile.blade.php untuk menampilkan dan mengedit data user. Lalu, ditambahkan fungsi profile() dan updateProfile() di controller untuk menangani tampilan dan update data. Route baru ditambahkan untuk akses dan penyimpanan data profil. Terakhir, dijalankan perintah php artisan storage:link untuk membuat folder penyimpanan foto profil.

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