

Laporan Praktikum Pemrograman Web Lanjut
Jobsheet 8: Authentication dan *Authorization* di Laravel



Oleh :

Danica Nasywa Putriniair (2341760122)

Kelas SIB 2B / 05

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS JURUSAN
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur

65141

Praktikum 1 - Implementasi Upload File untuk inport data

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

←T→		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	SKC	Skincare	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	PKN	Pakaian	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	ATK	Alat Tulis Kantor	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	AOR	Alat Olahraga	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	FNB	Food and Beverage	NULL	NULL

←T→		barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual
<input type="checkbox"/>	Edit Copy Delete	1	1	BNG001	Toner	150000	185000
<input type="checkbox"/>	Edit Copy Delete	2	1	BNG002	Serum	250000	315000
<input type="checkbox"/>	Edit Copy Delete	3	2	BNG003	Rok Plisket	50000	65000
<input type="checkbox"/>	Edit Copy Delete	4	2	BNG004	Kemeja Bunga	65000	75000
<input type="checkbox"/>	Edit Copy Delete	5	3	BNG005	Penggaris	2500	4000
<input type="checkbox"/>	Edit Copy Delete	6	3	BNG006	Kalkulator	100000	115000
<input type="checkbox"/>	Edit Copy Delete	7	4	BNG007	Matras Yoga	80000	120000
<input type="checkbox"/>	Edit Copy Delete	8	5	BNG008	Waffer Coklat	15000	18500
<input type="checkbox"/>	Edit Copy Delete	9	5	BNG009	Yogurt	10000	10500
<input type="checkbox"/>	Edit Copy Delete	10	5	BNG010	Kopi Susu	10000	15000

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.

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')


### Daftar barang</h3> <button onclick="modalAction('{{ url('/barang/import/') }}')" class="btn btn-info">Import Data</b <a href="{{ url('/barang/create') }}" class="btn btn-primary">Tambah Data</a> <button onclick="modalAction('{{ url('/barang/create_ajax/') }}')" class="btn btn-success">Tambah </div> </div> <!-- untuk Filter data --> <label for="filter_date" class="col-md-1 col-form-label">Filter</label> <select name="filter_kategori" class="form-control form-control-sm filter_kategor <option value="">- Semua -</option> @foreach($kategori as $l) <option value="{{ $l->kategori_id }}">{{ $l->kategori_nama }}</option> </foreach> </select> <small class="form-text text-muted">Kategori Barang</small> </div> <small class="form-text text-muted">Kategori Barang</small> </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 <table class="table table-bordered table-sm table-striped table-hover" id="table-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


```

```

resources/views/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 menhandle file

            $.ajax({
                url: form.action,
                type: form.method,
                data: formData,
                processData: false, // setting processData dan contentType ke false, untuk menhandle file
                contentType: false,
                success: function (response) {
                    if (response.status) { // jika sukses
                        $("#myModal").modal('hide');
                        Swal.fire({

```

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

```

// ADMIN & MANAGER (bisa CRUD)
Route::middleware(['authorize:ADM,MNG'])->prefix('barang')->group(function () {
    Route::get('/create', [BarangController::class, 'create']);
    Route::post('/', [BarangController::class, 'store']);
    Route::get('/create_ajax', [BarangController::class, 'create_ajax']);
    Route::post('/ajax', [BarangController::class, 'store_ajax']);
    Route::get("/{id}/edit", [BarangController::class, 'edit']);
    Route::put("/{id}", [BarangController::class, 'update']);
    Route::get("/{id}/edit_ajax", [BarangController::class, 'edit_ajax']);
    Route::put("/{id}/update_ajax", [BarangController::class, 'update_ajax']);
    Route::get("/{id}/delete_ajax", [BarangController::class, 'confirm_ajax']);
    Route::delete("/{id}/delete_ajax", [BarangController::class, 'delete_ajax']);
    Route::delete('/import', [BarangController::class, 'import']);
    Route::delete('/import_ajax', [BarangController::class, 'import_ajax']);
    Route::delete("/{id}", [BarangController::class, 'destroy']);
});

```


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

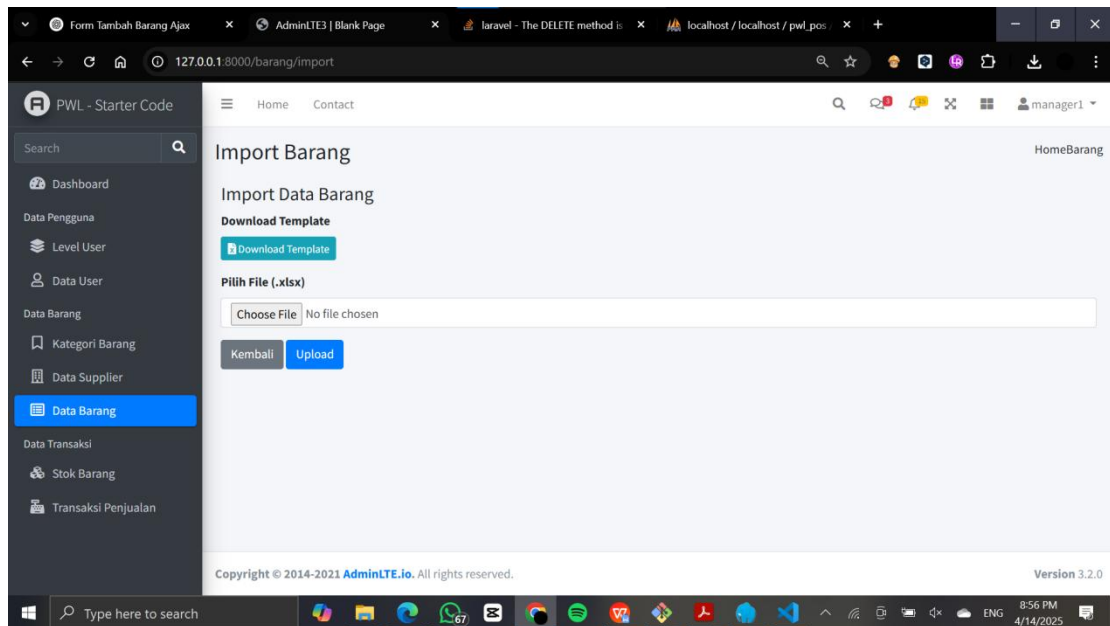
```
PS E:\Aplik\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 8> composer require phppoffice/phpspreadsheet
```

composer require phppoffice/phpspreadsheet

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

```
app > Http > Controllers > BarangController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\BarangModel;
6  use App\Models\KategoriModel;
7  use App\Models\LevelModel;
8  use Illuminate\Http\Request;
9  use Yajra\DataTables\Facades\DataTables;
10 use PhpOffice\PhpSpreadsheet\IOFactory;
11 use Illuminate\Support\Facades\Validator;
12 use Illuminate\Database\QueryException;
13 use Illuminate\Support\Facades\DB;
14
15 class BarangController extends Controller
16 {
17     public function index(){
18         $breadcrumb = (object) [
19             'title' => 'Daftar Barang',
20             'list' => ['Home', 'Barang']
21         ];
22
23         $page = (object) [
24             'title' => 'Daftar barang yang terdaftar dalam sistem'
25         ];
26
27         $activeMenu = 'barang';
28         $kategori = KategoriModel::select('kategori_id', 'kategori_nama')->get();
29         return view('barang.index', [
30             'activeMenu' => $activeMenu,
31             'breadcrumb' => $breadcrumb,
32             'kategori' => $kategori
33         ]);
34     }
35
36     public function list(Request $request)
37     {
38         $barang = BarangModel::select('barang_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual', 'kategori_id')
39             ->with('kategori');
40
41         $kategori_id = $request->input('filter_kategori');
42         if (!empty($kategori_id)) {
43             $barang->where('kategori_id', $kategori_id);
44         }
45
46         return DataTables::of($barang)
47             // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
```

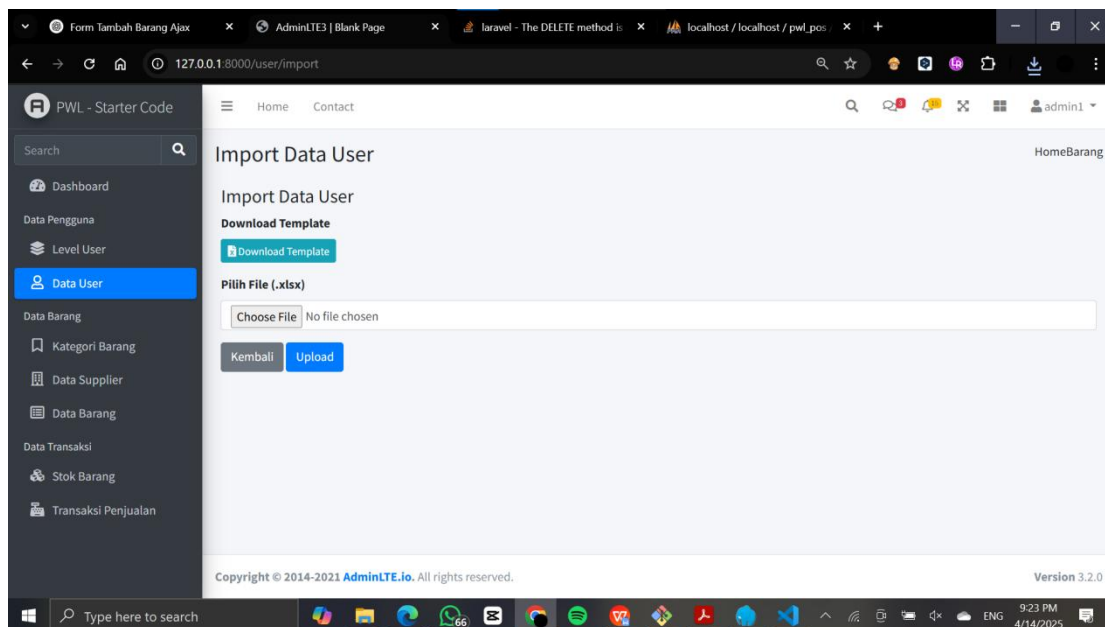
8. Sekarang kita coba jalankan browser dan klik tombol import pada menu Barang



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

Tugas 1 - Implementasi File Upload Import Data

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



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

Berikut ini merupakan langkah-langkah dalam proses import data barang secara massal menggunakan file Excel di sistem yang telah dibuat. Pertama, kita buka database yang sudah kita buat sebelumnya dan periksa isi dari tabel m_kategori dan

m_barang untuk memastikan bahwa strukturnya sudah sesuai. Kemudian, kita akan mencoba memasukkan data barang dalam jumlah banyak sekaligus menggunakan file Excel. Untuk itu, kita perlu membuat template file Excel, misalnya berisi 5 data barang, lalu simpan file tersebut dengan nama template_barang.xlsx ke dalam folder public pada project web kita.

Setelah itu, kita modifikasi tampilan view di barang/index.blade.php untuk menambahkan tombol dan form upload file Excel agar bisa mengimpor data barang. Kemudian, kita buat file view baru untuk halaman form upload tersebut dengan nama barang/import.blade.php, yang juga dilengkapi dengan tombol untuk mendownload file template Excel yang telah kita siapkan sebelumnya. Supaya sistem dapat menangani proses upload dan import, kita juga perlu memodifikasi file web.php di folder routes untuk menambahkan route baru yang mengarah ke proses import data pada menu Barang.

Agar sistem dapat membaca dan memproses file Excel, kita perlu menginstal library tambahan, yaitu phpooffice/phpspreadsheet. Kita jalankan perintah composer require phpooffice/phpspreadsheet melalui terminal atau command prompt. Selanjutnya, kita buka dan modifikasi BarangController.php untuk menambahkan logika dalam membaca file Excel, mengambil data dari file tersebut, lalu menyimpannya ke dalam database.

Setelah semua proses di atas selesai, kita jalankan aplikasi pada browser dan klik tombol "Import" pada menu Barang. Lalu, kita coba meng-upload file template data yang telah disiapkan. Namun, pada saat percobaan, proses upload ternyata gagal atau data tidak bisa masuk. Hal ini menandakan bahwa kita perlu menelusuri kembali bagian controller atau validasi file, untuk memastikan tidak ada kesalahan dalam membaca atau memproses file Excel yang diunggah.

3. Submit kode untuk implementasi praktikum 1 pada repository github kalian

Praktikum 2 -

1. Kita modifikasi `barang/index.blade.php` dengan mengganti kode berikut

```
<a href="{ url('/barang/create_ajax/') }}" class="btn btn-primary">Tambah Data</a>  
<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::post('/import_ajax', [BarangController::class, 'import_ajax'])->name('import_ajax');  
Route::get('/export_excel', [BarangController::class, 'export_excel']);  
Route::delete('/{id}', [BarangController::class, 'delete']);
```

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

```
public function export_excel()  
{  
    try {  
        // Ambil data barang yang akan diekspor  
        $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')  
            ->orderBy('kategori_id')  
            ->with('kategori')  
            ->get();  
  
        // Load library PhpSpreadsheet  
        $spreadsheet = new Spreadsheet();  
        $sheet = $spreadsheet->getActiveSheet();  
  
        // Set header kolom  
        $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');  
  
        // Format header bold  
        $sheet->getStyle('A1:F1')->getFont()->setBold(true);  
  
        // Isi data barang  
        $no = 1;  
        $baris = 2;
```

4. Kita ambil data barang yang akan kita export ke excel (tentu dengan menyertakan relasi kategori barang)
5. Kemudian kita load library Spreadsheet dan kita tentukan header data pada baris pertama di excel
6. Selanjutnya, kita looping data yang telah kita dapatkan dari database, kemudian kita masukkan ke dalam cell excel
7. Kita set lebar tiap kolom di excel untuk menyesuaikan dengan panjang karakter pada masing-masing kolom
8. Bagian akhir proses export excel adalah kita set nama sheet, dan proses untuk dapat di download oleh pengguna
9. Jika sudah selesai diimplementasikan, kita coba untuk melakukan Download file Export tersebut.

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	BNG001	Toner	150000	185000	Skincare
2	BNG002	Serum	250000	315000	Skincare
3	BNG003	Rok Plisket	50000	65000	Pakaian
4	BNG004	Kemeja Bunga	65000	75000	Pakaian
5	BNG005	Penggaris	2500	4000	Alat Tulis Kantor
6	BNG006	Kalkulator	100000	115000	Alat Tulis Kantor
7	BNG007	Matras Yoga	80000	120000	Alat Olahraga
8	BNG008	Waffer Coklat	15000	18500	Food and Beverage
9	BNG009	Yogurt	10000	10500	Food and Beverage
10	BNG010	Kopi Susu	10000	15000	Food and Beverage

Tugas 2 - Implementasi File Export

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

Kita mulai dengan memodifikasi file `barang/index.blade.php` untuk menghapus tombol Tambah Data lama karena sudah digantikan dengan tombol Tambah Data berbasis Ajax. Selanjutnya, kita tambahkan route baru pada file `web.php` untuk menangani proses export Excel. Kemudian, kita implementasikan fungsi `export_excel()` di dalam `BarangController.php`, yang bertugas mengambil data barang dari database beserta relasi kategori barangnya. Setelah data berhasil diambil, kita load library Spreadsheet, lalu menentukan header data pada baris pertama file Excel. Berikutnya, kita lakukan looping terhadap data tersebut dan mengisi setiap cell di Excel sesuai dengan isi data. Untuk memastikan tampilannya rapi, kita juga set lebar tiap kolom agar menyesuaikan panjang karakter masing-masing kolom. Di bagian akhir, kita tentukan nama sheet dan melakukan proses export agar file Excel bisa langsung diunduh oleh pengguna. Setelah semua langkah selesai, kita bisa mencoba fitur tersebut untuk memastikan file export berhasil terdownload dengan baik.

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

Praktikum 3 - Implementasi Export PDF di Laravel dengan dompdf

1. Kita lakukan proses instalasi library dompdf terlebih dahulu dengan mengetikkan perintah pada terminal/CMD

```
..... ~ 110s
PS E:\Aplik\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 8> 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
Installing dependencies from lock file (including require-dev)
```

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_excel') }}" class="btn btn-warning">Tambah Data</a>
<a href="{{ url('/barang/export_excel') }}" class="btn btn-primary"><i class="fa fa-file-excel"></i> Export Barang EXC</a>
<a href="{{ url('/barang/export_pdf') }}" class="btn btn-warning"><i class="fa fa-file-pdf"></i> Export Barang PDF</a>
```

3. Setelah itu tinggal kita perbaiki route/web.php untuk proses export pdf

```
Route::get('/export_excel', [BarangController::class, 'export_excel']);
Route::get('/export_pdf', [BarangController::class, 'export_pdf']);
Route::delete('/{id}', [BarangController::class, 'destroy']);
```

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

```

resources > views > barang > export_pdf.blade.php > html > head > style > .d-block
1  <html>
2  <head>
3      <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
4      <style>
5          body{
6              font-family: "Times New Roman", Times, serif;
7              margin: 6px 20px 5px 20px;
8              line-height: 15px;
9          } table {
10             width:100%; border-collapse: collapse;
11         } td, th {
12             padding: 4px 3px;
13         } th {
14             text-align: left;
15         } .d-block {
16             display: block;
17         } img.image {
18             width: auto;
19             height: 80px;
20             max-width: 150px;
21             max-height: 150px;
22         } .text-right {
23             text-align: right;
24         } .text-center {
25             text-align: center;
26         } .p-1 {
27             padding: 5px 1px 5px 1px;
28         } .font-10 {
29             font-size: 10pt;
30         } .font-11 {
31             font-size: 11pt;
32         } .font-12 {
33             font-size: 12pt;

```

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

Tugas 3 - Implementasi Export PDF pada Laravel

1. Silahkan implementasikan export pdf pada project kalian masing-masing untuk semua menu
2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan

Pertama, kita melakukan instalasi library dompdf melalui terminal atau CMD. Setelah proses instalasi berhasil, kita tambahkan kode untuk menampilkan tombol export ke PDF pada file barang/index.blade.php. Selanjutnya, kita perbaiki route di route/web.php untuk menangani proses export PDF. Kemudian, buat fungsi export_pdf() di BarangController.php untuk menangani proses ekspor data. Setelah itu, buat view barang/export_pdf.blade.php yang akan dijadikan template untuk PDF dengan layout HTML. Terakhir, kita buat tampilan HTML di view barang/export_pdf.blade.php untuk menghasilkan format PDF yang diinginkan. Setelah semua selesai, coba lakukan proses download export PDF dan pelajari hasilnya.

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