

LAPORAN JOBSHEET 8
PEMROGRAMAN WEB LANJUT
File Import dan Export ke PDF dan Excel pada Laravel

Dosen pengajar: Dimas Wahyu Wibowo, ST., MT



Disusun Oleh:

Nama : M. Fatih Al Ghifary

NIM : 2341720194

Kelas : TI-2A

No : 16

PRODI TEKNIK INFORMATIKA D-IV
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2025

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
1	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual
2	1	PSC001	Sabun mandi (100	3000	4500
3	2	PSC002	Pasta gigi (75ml)	5000	7500
4	3	BBY001	Popok bayi	200000	250000
5	4	BBY002	Susu formula	35000	64000
6	5	BBY003	Tisu basah bayi	16000	22000
7					

- 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')


### {{ $page->title }}



&ltbutton onclick="modalAction('{{ url(path: '/barang/import') }}')" class="btn btn-sm btn-info">Import
Barang</button>
&lta class="btn btn-sm btn-primary mt-1" href="{{ url(path: 'barang/create') }}">Tambah</a>
&ltbutton onclick="modalAction('{{ url(path: 'barang/create_ajax') }}')" class="btn btn-sm btn-success mt-1">
Tambah Ajax
</button>



Filter:</div>


&ltselect class="form-control" id="kategori_id" name="kategori_id" required>
&ltoption value="">- Semua -</option>
@foreach ($kategori as $item)
&ltoption value="{{ $item->kategori_id }}">{{ $item->kategori_nama }}</option>
@endforeach
</select>



@if (session(key: 'success'))
&ltdiv class="alert alert-success">{{ session(key: 'success') }}</div>
@endif
@if (session(key: 'error'))
&ltdiv class="alert alert-danger">{{ session(key: 'error') }}</div>
@endif


```

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(path: '/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(path: '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
            }
        });
    });
</script>
```

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

```
Route::get('/barang/import', [BarangController::class, 'import']);
Route::post('/barang/import_ajax', [BarangController::class, 'import_ajax']);
```

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

```
D:\KULIAH\SEMESTER-4\file\laragon\www\PROWEBLNJT\Week 8\PWL_POS>composer require phpooffice/phpspreadsheet
./composer.json has been updated
Running composer update phpooffice/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 phpooffice/phpspreadsheet (4.1.0)
Writing lock file
```

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

```
1 reference | 0 overrides
public function import(): Factory|View
{
    return view(view: 'barang.import');
}

1 reference | 0 overrides
public function import_ajax(Request $request): JsonResponse|mixed|Redirector|RedirectRes...
{
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            // validasi file harus xls atau xlsx, 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(value: $data) > 1) { // jika data lebih dari 1 baris
            foreach ($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(value: $insert) > 0) {
                // insert data ke database, jika data sudah ada, maka diabaikan
                BarangModel::insertOrIgnore($insert);
            }

            return response()->json([
                'status' => true,
            ]);
        }
    }
}
```

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

Import Data Barang

Download Template [Download](#)

Pilih File

[Choose File](#) No file chosen

Batal Upload

Nama Barang	Harga Beli	Harga Jual	Kategori
-------------	------------	------------	----------

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


Import Data Barang

Download Template [Download](#)

Pilih File

[Choose File](#) template_barang.xlsx

Batal Upload



Berhasil

Data berhasil diimport

[OK](#)

Rp 15.000
Rp 18.000

Show 10 entries
Search:

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori	Aksi
11	PSC001	Sabun mandi (100g)	Rp 3.000	Rp 4.500	Groceries	Detail Edit Hapus
12	PSC002	Pasta gigi (75ml)	Rp 5.000	Rp 7.500	Snacks	Detail Edit Hapus
13	BBY001	Popok bayi	Rp 200.000	Rp 250.000	Drinks	Detail Edit Hapus
14	BBY002	Susu formula	Rp 35.000	Rp 64.000	Household Products	Detail Edit Hapus
15	BBY003	Tisu basah bayi	Rp 16.000	Rp 22.000	Frozen Foods	Detail Edit Hapus

Showing 11 to 15 of 15 entries

[Previous](#)
1
2
[Next](#)

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 Jawab:
 1. Mengecek keberadaan tabel m_kategori dan m_barang di database yang digunakan.
 2. Menyusun file template Excel (template_barang.xlsx) yang berisi data barang yang akan diimport.
 3. Menambahkan tombol pada barang/index.blade.php yang mengarah ke halaman form upload/import.
 4. File view barang/import.blade.php dibuat untuk form upload file Excel dan link download template.
 5. Menambahkan route baru di web.php untuk menangani proses upload/import data.
 6. Menggunakan perintah composer require phpooffice/phpspreadsheet untuk mendukung proses baca/tulis Excel.
 7. Menambahkan kode di BarangController.php untuk membaca data dari file Excel dan menyimpannya ke database.
 8. Mengakses tombol import dari browser, mengunggah file Excel, dan mengamati apakah data berhasil masuk ke sistem.

3. Submit kode untuk impementasi prakktikum 1 pada repository github kalian.
<https://github.com/fatechhh/PROWEBLNJT/commit/438e4aa9bd2d28e02d8ff29e7e76c23aac1eec08>

Praktikum 2 – Export Data ke Excel

1. Modifikasi barang/index.blade.php

```
barang</button>  
<a href="{ url(path: '/barang/export_excel') }}" class="btn btn-primary"><i class="fa fa-fileexcel"></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('/barang/import_ajax', [BarangController::class, 'import_ajax']);  
Route::get('/barang/export_excel', [BarangController::class, '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)
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

```
public function export_excel(): never
{
    // 1. 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();

    // 2. load library excel
    $spreadsheet = new \PhpOffice\PhpSpreadsheet\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

    // 3.
    $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++;
    }

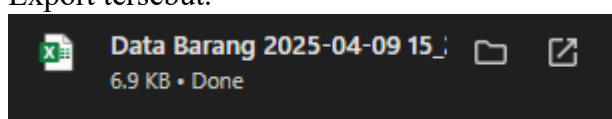
    // 4.
    foreach (range(start: 'A', end: 'F') as $columnID) {
        $sheet->getColumnDimension($columnID)->setAutoSize(true); // set auto size untuk kolom
    }

    // 5.
    $sheet->setTitle('Data Barang'); // set title sheet

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

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

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	Beras 5kg	55000	60000	Groceries
2	BRG002	Minyak Goreng 2L	28000	32000	Groceries
3	PSC001	Sabun mandi (100g)	3000	4500	Groceries
4	BRG003	Keripik Kentang	12000	15000	Snacks
5	BRG004	Coklat Batangan	8000	10000	Snacks
6	PSC002	Pasta gigi (75ml)	5000	7500	Snacks
7	BRG005	Teh Celup 25pcs	10000	13000	Drinks
8	BRG006	Kopi Instan 10s	15000	18000	Drinks
9	BBY001	Popok bayi	200000	250000	Drinks
10	BRG007	Sabun Cuci Piring	7000	10000	Household Products
11	BRG008	Pewangi Pakaian 1L	25000	30000	Household Products
12	BBY002	Susu formula	35000	64000	Household Products
13	BRG009	Nugget Ayam 500g	32000	37000	Frozen Foods
14	BRG010	Sosis Sapi 500g	40000	45000	Frozen Foods
15	BBY003	Tisu basah bayi	16000	22000	Frozen Foods

Tugas 2 – Implementasi File Export Excel

- Silahkan implementasikan praktikum 2 pada project kalian masing-masing untuk semua menu
- Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan Jawab:
 - Menghilangkan tombol tambah data lama dan menambahkan tombol Export Excel.
 - Menambahkan route untuk proses export Excel di web.php.
 - Fungsi export_excel() di BarangController.php mengambil data dari database beserta relasi kategori.
 - Header kolom diatur secara manual, kemudian data di-looping dan dimasukkan ke Excel.
 - Mengatur lebar kolom agar menyesuaikan dengan konten data.
 - Sheet diberi nama dan file Excel dikonversi agar dapat langsung diunduh pengguna.
 - Menekan tombol Export Excel di halaman barang dan menguji apakah file berhasil terunduh.
- Submit kode untuk impementasi prakktikum 2 pada repository github kalian.
<https://github.com/fatechhh/PROWEBLNJT/commit/bd9855adfb2b369835ef0c1f2a997ea09372c5b4>

Praktikum 3 – Implementasi Export PDF di Laravel dengan dompdf

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

```
D:\KULIAH\SEMESTER-4\file\laragon\www\PROWEBLNJT\Week 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)
```

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(path: '/barang/export_pdf') }}" class="btn btn-sm 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/export_pdf', [BarangController::class, 'export_pdf']);
```

4. Buat fungsi export_pdf() pada BarangController.php

```
public function export_pdf(): Response
{
    $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
        ->orderBy('kategori_id')
        ->orderBy('barang_kode')
        ->with('kategori')
        ->get();

    $pdf = Pdf::loadView('barang.export_pdf', ['barang' => $barang]);
    $pdf->setPaper('A4', 'potrait');
    $pdf->setOption(['isRemoteEnabled', true]);
    $pdf->render();

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

5. Buat view untuk dijadikan pdf dari layout html. File dibuat dengan nama barang/export_pdf.blade.php

6. Buat view untuk me-generate html untuk tampilan pdf

```
<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 {
        width: auto;
        height: 80px;
        max-width: 150px;
        max-height: 150px;
    }

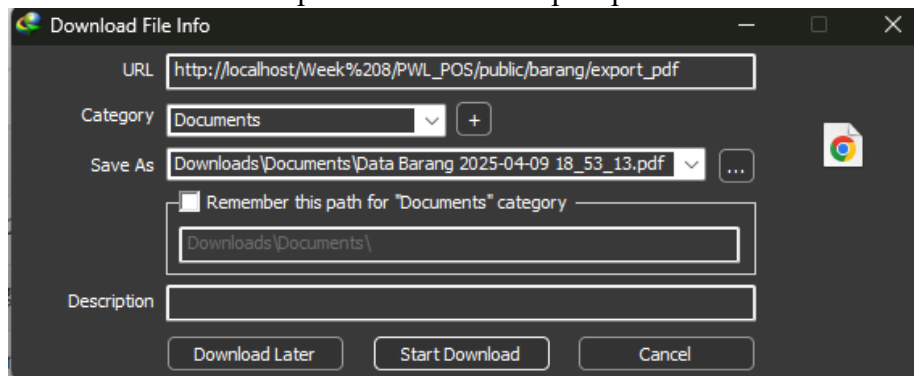
    .text-right {
        text-align: right;
    }

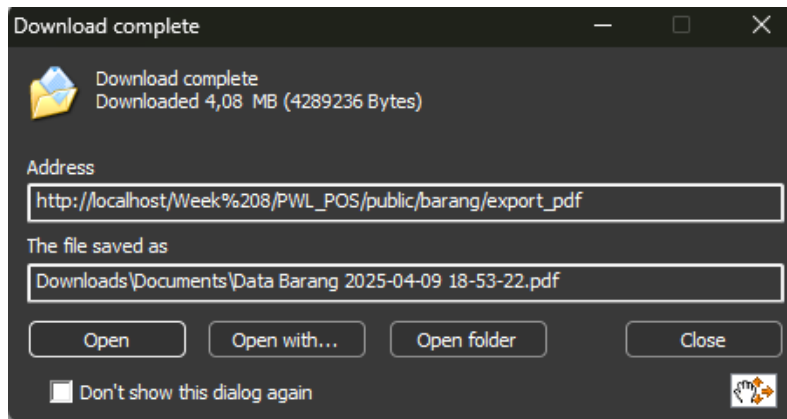
    .text-center {
        text-align: center;
    }

    .p-1 {
        padding: 5px 1px 5px 1px;
    }

```

7. Percobaan melakukan proses download export pdf.





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
 Jl. Sockarno-Hatta No. 9 Malang 65141
 Telepon (0341) 404424 Pcs. 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	Beras 5kg	55.000	60.000	Groceries
2	BRG002	Minyak Goreng 2L	28.000	32.000	Groceries
3	PSC001	Sabun mandi (100g)	3.000	4.500	Groceries
4	BRG003	Keripik Kentang	12.000	15.000	Snacks
5	BRG004	Coklat Batangan	8.000	10.000	Snacks
6	PSC002	Pasta gigi (75ml)	5.000	7.500	Snacks
7	BBY001	Popok bayi	200.000	250.000	Drinks
8	BRG005	Teh Celup 25pcs	10.000	13.000	Drinks
9	BRG006	Kopi Instan 10s	15.000	18.000	Drinks
10	BBY002	Susu formula	35.000	64.000	Household Products
11	BRG007	Sabun Cuci Piring	7.000	10.000	Household Products
12	BRG008	Pewangi Pakaian 1L	25.000	30.000	Household Products
13	BBY003	Tisu basah bayi	16.000	22.000	Frozen Foods
14	BRG009	Nugget Ayam 500g	32.000	37.000	Frozen Foods
15	BRG010	Sosis Sapi 500g	40.000	45.000	Frozen Foods

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 Jawab:
 1. Menggunakan perintah composer require barryvdh/laravel-dompdf di terminal.
 2. Menambahkan tombol export ke PDF di barang/index.blade.php.
 3. Menambahkan route untuk fungsi export PDF pada web.php.
 4. Menambahkan export_pdf() di BarangController.php untuk mengambil data dan merender PDF.

5. View barang/export_pdf.blade.php dibuat dengan layout HTML yang sesuai untuk dikonversi ke PDF.
6. Tombol export ditekan dari halaman web dan hasil file PDF diuji apakah berhasil terdownload.
3. Submit kode untuk implementasi export pdf pada repository github kalian.
<https://github.com/fatechhh/PROWEBLNJT/commit/67cef874f43f268b789d8855a387b4874c813af4>

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.
 - Menambahkan migration untuk kolom picture_path di m_user

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('m_user', function (Blueprint $table): void {
            $table->string('picture_path')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('m_user', function (Blueprint $table): void {
            $table->dropColumn('picture_path');
        });
    }
};
```

- Menambahkan kode di dalam function store_ajax, untuk menyimpan foto profile, mengubah nama foto profile yang di upload, dan mengambil foto profile melalui path yang ditentukan untuk menampilkannya di view

```
public function store_ajax(Request $request): JsonResponse|mixed
{
    // cek apakah request berupa ajax
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'level_id' => 'required|integer',
            'username' => 'required|string|min:3|unique:m_user,username',
            'nama' => 'required|string|max:100',
            'password' => 'required|min:5|max:20',
            'profile_picture' => 'nullable|image|mimes:jpeg,png,jpg,gif|max:2048'
        ];

        $validator = Validator::make($request->all(), $rules);

        if ($validator->fails()) {
            return response()->json([
                'status' => false,
                'message' => 'Validasi Gagal',
                'msgField' => $validator->errors(),
            ]);
        }

        $data = $request->all();
        if ($request->hasFile('profile_picture')) {
            $id = UserModel::all()->max('user_id') + 1;
            $image = $request->file('profile_picture');
            $imageName = 'profile-' . $id . '.webp';
            $image->storeAs('public/uploads/profile_pictures', $imageName);
            $data['picture_path'] = 'storage/uploads/profile_pictures/' . $imageName;
            unset($data['profile_picture']);
        }

        UserModel::create($data);

        return response()->json([
            'status' => true,
            'message' => 'Data user berhasil disimpan'
        ]);
    }
    redirect(to: '/');
}
```

- Menambahkan kode dengan tujuan yang sama pada function show_ajax, namun ini di update_ajax

```
public function update_ajax(Request $request, $id): JsonResponse|mixed|Redirector|RedirectRes...
{
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'level_id' => 'required|integer',
            'username' => 'required|max:20|unique:m_user,username,' . $id . ',user_id',
            'nama' => 'required|max:100',
            'password' => 'nullable|min:5|max:20',
            'profile_picture' => 'nullable|image|mimes:jpeg,png,jpg,gif|max:2048'
        ];

        $validator = Validator::make($request->all(), $rules);

        if ($validator->fails()) {
            return response()->json([
                'status' => false,
                'message' => 'Validasi gagal.',
                'msgField' => $validator->errors()
            ]);
        }

        $check = UserModel::find($id);
        if ($check) {
            if (!$request->filled('password')) {
                $request->request->remove('password');
            }

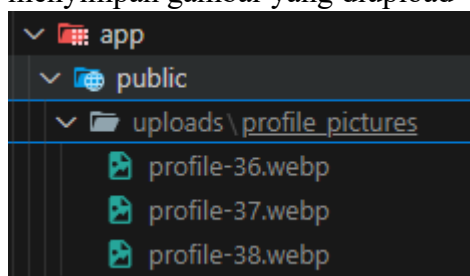
            $data = $request->all();
            if ($request->hasFile('profile_picture')) {
                $image = $request->file('profile_picture');
                $imageName = 'profile-' . $id . '.webp';
                $image->storeAs('public/uploads/profile_pictures', $imageName);
                $data['picture_path'] = 'storage/uploads/profile_pictures/' . $imageName;
                unset($data['profile_picture']);
            }

            $check->update($data);
            return response()->json([
                'status' => true,
                'message' => 'Data berhasil diupdate'
            ]);
        } else {
            return response()->json([
                'status' => false,
                'message' => 'Data tidak ditemukan'
            ]);
        }
    }
}
```

- Menambahkan picture_path didalam \$fillable UserModel.php

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

- Menambahkan folder profile_picture di dalam /storage/app/public untuk menyimpan gambar yang diupload



- Menambahkan class untuk menampilkan tempat upload foto profile pada saat create user baru menggunakan ajax di dalam file create_ajax.blade.php

```
<div class="col-md-4 text-center">
  <label for="profile_picture" class="position-relative d-block"
    style="width: 150px; height: 150px; margin: 0 auto; clip-path: circle(50% at 50% 50%);">
    
    <div class="overlay rounded-circle"
      style="opacity: 0; transition: opacity 0.15s; cursor: pointer;"
      onmouseover="this.style.opacity = 1;" onmouseout="this.style.opacity = 0;">
      <i class="fas fa-upload position-absolute text-white"
        style="top: 50%; left: 50%; transform: translate(-50%, -50%); font-size: 30px;"></i>
    </div>
  </label>
  <p class="mt-2 mb-1 text-muted" style="font-size: 14px;">Upload Foto Profil</p>
  <input type="file" name="profile_picture" id="profile_picture" class="d-none"
    accept="image/jpeg, image/jpg, image/png"
    onchange="this.parentNode.querySelector('label').querySelector('img').src = window.URL.createObjectURL(this.files[0]);">
  <small id="error-profile_picture" class="error-text form-text text-danger"></small>
</div>
```

- Menambahkan class untuk menampilkan foto profile pada saat melihat detail user di dalam show_ajax.blade.php

```
<div class="flex-grow-1">
  <!-- Foto Profil -->
  <div class="text-center mr-4 mb-3">
    
  </div>
  <table class="table table-bordered table-striped table-hover table-sm mb-0">
```

- Menambahkan class untuk menampilkan foto profile pada saat mengupload foto waktu mengedit user di dalam edit_ajax.blade.php

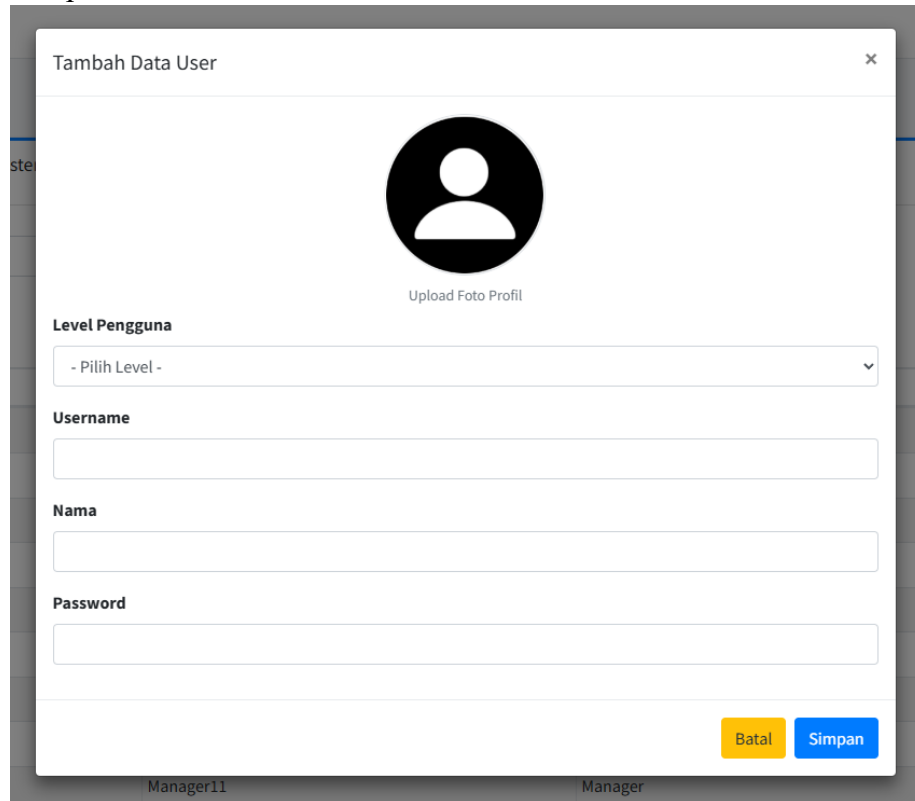
```
{!-- Upload Foto Profil --}}
<div class="row justify-content-center mb-4">
  <div class="col-md-4 text-center">
    <label for="profile_picture" class="position-relative d-block"
      style="width: 150px; height: 150px; margin: 0 auto; clip-path: circle(50% at 50% 50%);">
      
      <div class="overlay rounded-circle"
        style="opacity: 0; transition: opacity 0.15s; cursor: pointer;"
        onmouseover="this.style.opacity = 1;" onmouseout="this.style.opacity = 0;">
        <i class="fas fa-upload position-absolute text-white"
          style="top: 50%; left: 50%; transform: translate(-50%, -50%); font-size: 30px;"></i>
      </div>
    </label>
    <p class="mt-2 mb-1 text-muted" style="font-size: 14px;">Upload Foto Profil</p>
    <input type="file" name="profile_picture" id="profile_picture" class="d-none"
      accept="image/jpeg, image/jpg, image/png"
      onchange="document.getElementById('preview-profile-picture').src = window.URL.createObjectURL(this.files[0]);">
    <small id="error-profile_picture" class="error-text form-text text-danger"></small>
  </div>
</div>
```

- Menambahkan Class untuk menampilkan foto profile pada saat ingin menghapus user di dalam confirm_ajax.blade.php

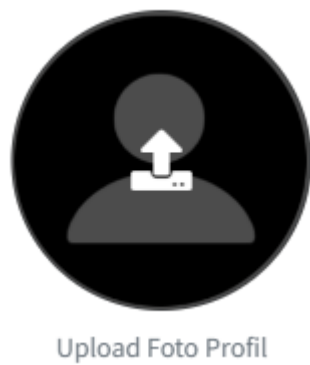
```
<!-- Tabel Data -->
<div class="flex-grow-1">
  <!-- Foto Profil -->
  <div class="text-center mr-4 mb-3">
    
  </div>
```


LANGKAH LANGKAH

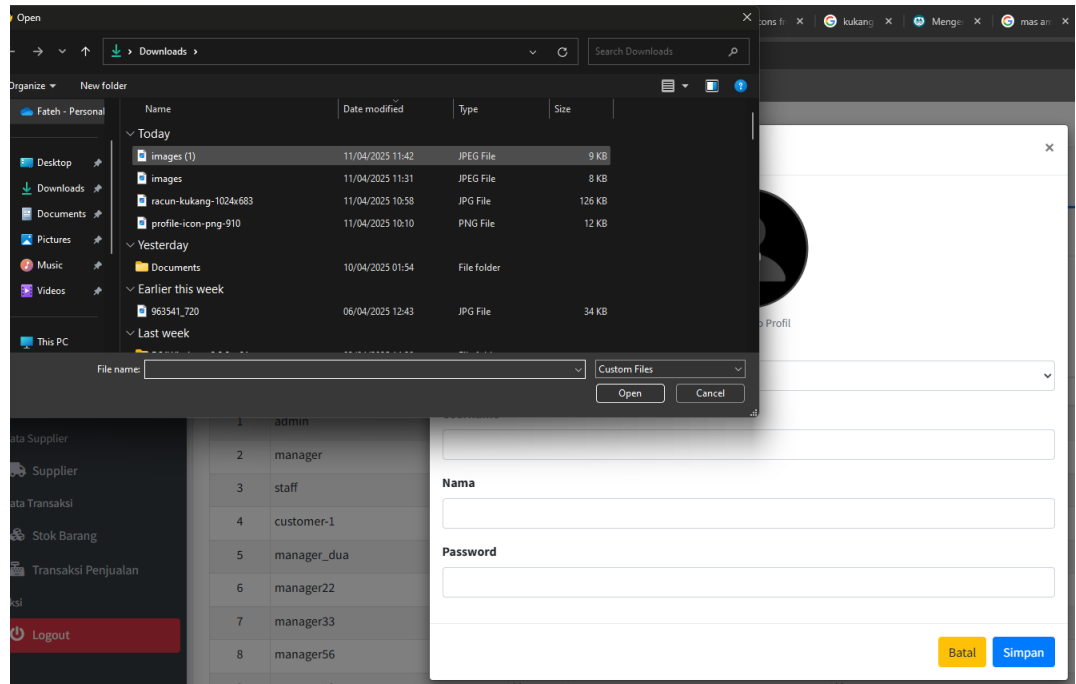
- Tambah AJAX
 1. Login sebagai user admin, dan buka menu data user
 2. Tampilan Tambah AJAX



3. Pada saat dihover maka tampilan berubah menjadi tempat penguploadan



4. Memilih foto



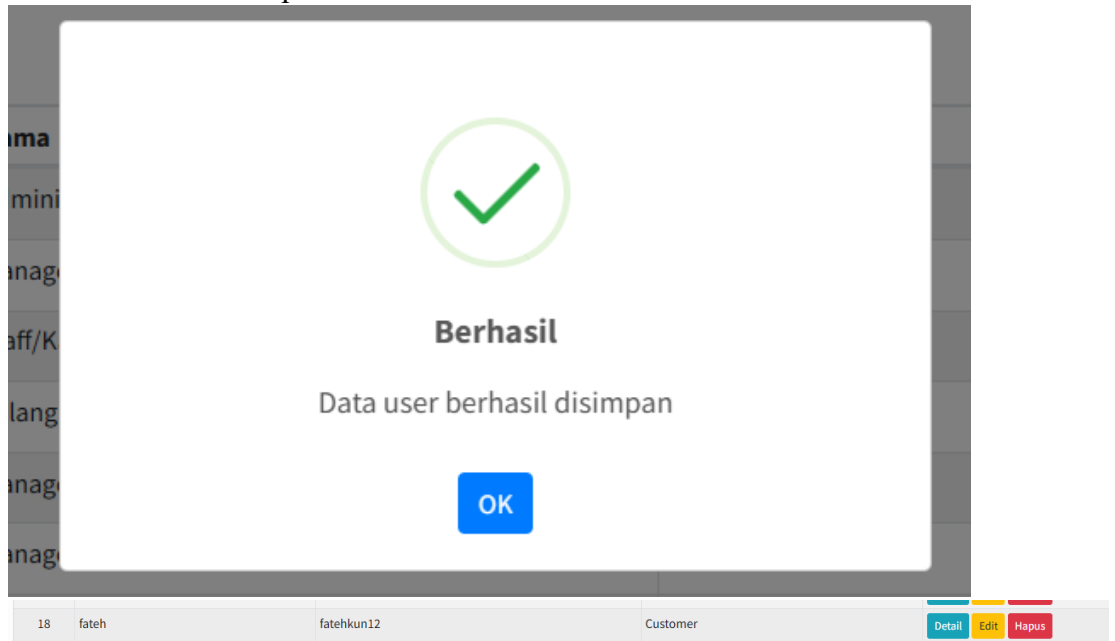
5. Mengisikan data user

The screenshot shows a 'Tambah Data User' (Add User Data) form. At the top, there is a circular profile picture placeholder with the text 'Upload Foto Profil' below it. Below this, the form contains the following fields:

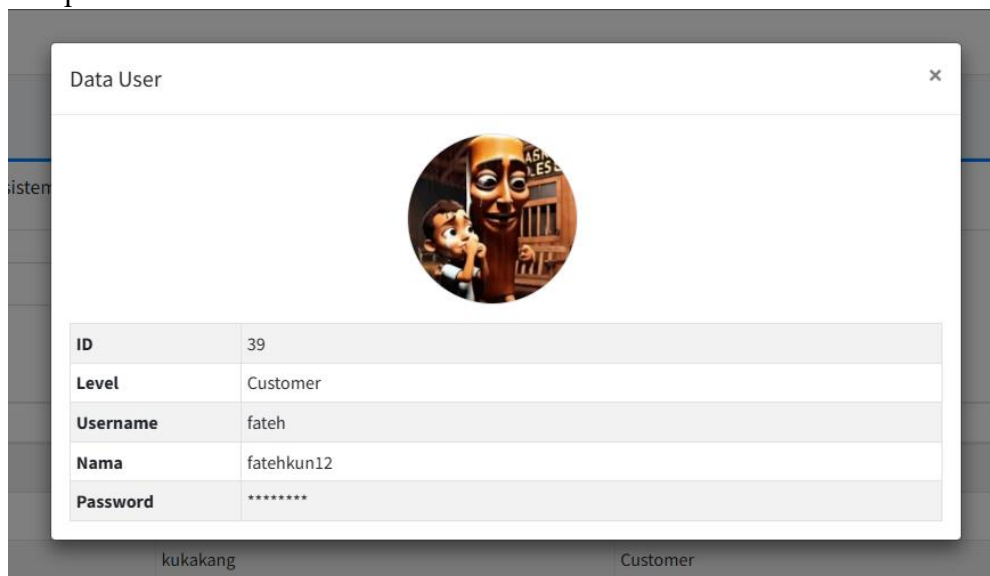
- Level Pengguna**: A dropdown menu currently showing 'Customer'.
- Username**: A text input field containing 'fateh'.
- Nama**: A text input field containing 'fatehkun12'.
- Password**: A password input field with masked characters '*****'.

At the bottom right of the form are two buttons: 'Batal' (Cancel) and 'Simpan' (Save).

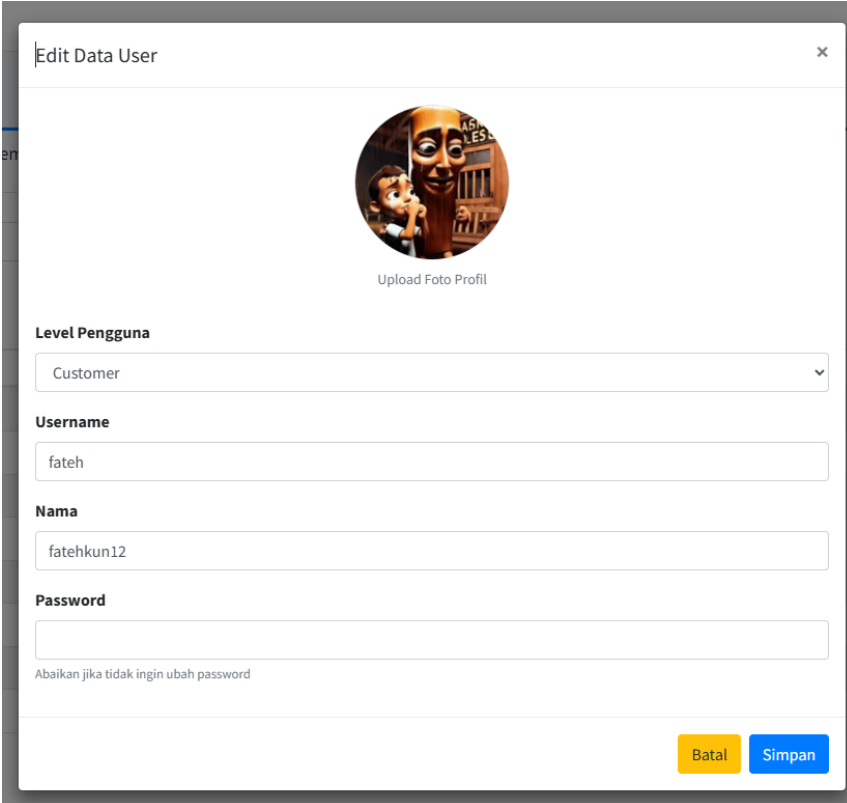
6. Ketika berhasil disimpan maka muncul notifikasi berhasil



7. Tampilan detail user



8. Tampilan edit user



Edit Data User

Upload Foto Profil

Level Pengguna

Customer

Username

fateh

Nama

fatehkun12

Password

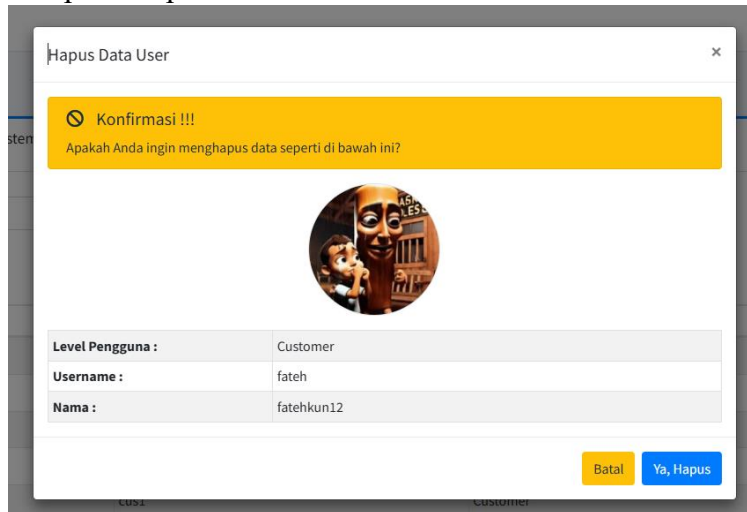
Abaikan jika tidak ingin ubah password

Batal Simpan

9. Ketika menghover foto profile maka dapat berubah menjadi tempat upload



10. Tampilan hapus data user



3. Submit kode untuk impementasi export pdf pada repository github kalian.
<https://github.com/fateehhh/PROWEBLNJT/commit/5972c5e36a31d82480412473d6c41dcd4632bc91>