

PRAKTIKUM PEMROGRAMAN WEB LANJUT

TUGAS PERTEMUAN 8

File Import dan Export ke PDF dan Excel pada Laravel

DISUSUN OLEH :

GILANG PURNOMO

NIM:2341720042



PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

APRIL 2025

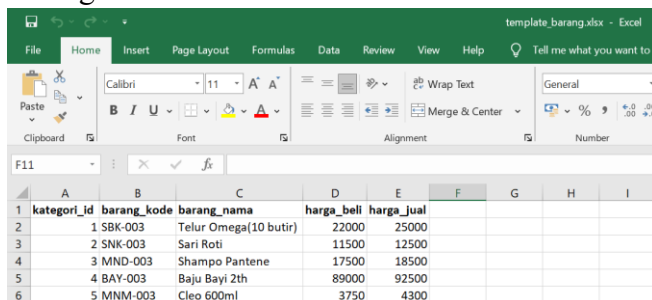
Praktikum 1 – Implementasi Upload File untuk import data

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

kategori_id	kategori_kode	kategori_nama	created_at	updated_at
1	K001	Bayi & Anak-anak	NULL	NULL
2	K002	Kecantikan & Kesehatan	NULL	NULL
3	K003	Makanan & Minuman	NULL	NULL
4	K004	Perawatan Rumah	NULL	2025-03-12 14:50:03
5	K005	Elektronik	NULL	2025-03-12 15:11:31
8	K09	Sepatu	2025-03-12 15:11:58	2025-03-23 04:18:42

barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
1	1	B001	Popok Bayi	30000	40000	NULL	NULL
2	1	B002	Susu Formula	70000	85000	NULL	NULL
3	2	B003	Sabun Herbal	15000	20000	NULL	NULL
4	2	B004	Shampoo Anti Ketombe	25000	35000	NULL	NULL
5	3	B005	Biskuit Gandum	10000	15000	NULL	NULL
6	3	B006	Minuman Kopi Golda	3500	5000	NULL	NULL
7	4	B007	Deterjen Cair	25000	35000	NULL	NULL
8	4	B008	Pembersih Lantai	15000	17000	NULL	NULL
9	5	B009	Rice Cooker	300000	350000	NULL	NULL
10	5	B010	Kipas Angin	160000	200000	NULL	NULL
12	8	NK	Nike Air Force	800000	1000000	2025-03-22 08:43:52	2025-03-22 08:49:56

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.



kategori_id	barang_kode	barang_nama	harga_beli	harga_jual
1	SBK-003	Telur Omega(10 butir)	22000	25000
2	SNK-003	Sari Roti	11500	12500
3	MND-003	Shampo Pantene	17500	18500
4	BAY-003	Baju Bayi 2th	89000	92500
5	MNM-003	Cleo 600ml	3750	4300

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


```

49         <th>Harga Jual</th>
50     </tr>
51     <th>Kategori</th>
52     <th>Aksi</th>
53 </tr>
54 </thead>
55 <tbody></tbody>
56 </table>
57 </div>
58 <div id="myModal" class="modal fade animate shake" tabindex="-1" data-backdrop="static" data-keyboard="false" data-width="75%"></div>
59 @endsection
60
61 @push('css')
62 @endpush
63
64 @push('js')
65 <script>
66     function modalAction(url = '') {
67         $('#myModal').load(url, function() {
68             $('#myModal').modal('show');
69         });
70     }
71
72     var dataBarang;
73     $(document).ready(function() {
74         dataBarang = $('#table_barang').DataTable({
75             processing: true,
76             serverSide: true,
77             ajax: {
78                 "url": "{{ url('barang/list') }}",
79                 "dataType": "json",
80                 "type": "POST",
81                 "data": function (d) {
82                     d.filter_kategori = $('#filter_kategori').val();
83                 }
84             },
85             columns: [
86                 {
87                     data: "DT_RowIndex",
88                     className: "text-center",
89                     width: "5%",
90                     orderable: false,
91                 },
92                 {
93                     data: "nama_barang",
94                     className: "text-center",
95                     width: "30%",
96                     orderable: false,
97                 },
98                 {
99                     data: "harga_jual",
100                    className: "text-center",
101                    width: "15%",
102                    orderable: false,
103                },
104                 {
105                     data: "kategori",
106                     className: "text-center",
107                     width: "15%",
108                     orderable: false,
109                 },
110                 {
111                     data: "aksi",
112                     className: "text-center",
113                     width: "25%",
114                     orderable: false,
115                 }
116             ]
117         });
118     });
119 }
120 </script>
121 @endpush

```

```

91         searchable: false,
92     },
93     {
94         data: "barang_kode",
95         className: "",
96         width: "10%",
97         orderable: true,
98         searchable: true
99     },
100    {
101        data: "barang_nama",
102        className: "",
103        width: "37%",
104        orderable: true,
105        searchable: true,
106    },
107    {
108        data: "harga_beli",
109        className: "",
110        width: "10%",
111        orderable: true,
112        searchable: false,
113        render: function(data, type, row){
114            return new Intl.NumberFormat('id-ID').format(data);
115        }
116    },
117    {
118        data: "harga_jual",
119        className: "",
120        width: "10%",
121        orderable: true,
122        searchable: true,
123        render: function(data, type, row){
124            return new Intl.NumberFormat('id-ID').format(data);
125        }
126    },
127    {
128        data: "kategori.kategori_nama",
129        className: "",
130        width: "14%",
131        orderable: true,
132        searchable: false
133    },
134    {
135        data: "aksi",
136        className: "text-center",
137        width: "14%",
138        orderable: false,
139        searchable: false
140    }
141    ]
142    });
143    $('#table-barang_filter input').unbind().bind().on('keyup', function(e){
144        if(e.keyCode == 13){ // enter key
145            dataBarang.search(this.value).draw();
146        }
147    });
148    $('#filter_kategori').change(function(){
149        dataBarang.draw();
150    });
151    // $('#kategori_id').on('change', function() {
152    //     dataBarang.ajax.reload();
153    // });
154    </script>
155    @endpush

```

- 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`

```

Minggu 8 > PWL_POS > resources > views > barang > importblade.php > script > <function>
1 <form action="{{ url('/barang/import_ajax') }}" method="POST" id="form-import" enctype="multipart/form-data">
2 @csrf
3 <div id="modal-master" class="modal-dialog modal-lg" role="document">
4 <div class="modal-content">
5 <div class="modal-header">
6 <h5 class="modal-title" id="exampleModallabel">Import Data Barang</h5>
7 <button type="button" class="close" data-dismiss="modal" arialabel="Close"><span aria-hidden="true">&times;</span></button>
8 </div>
9 <div class="modal-body">
10 <div class="form-group">
11 <label>Download Template</label>
12 <a href="{{ asset('template_barang.xlsx') }}" class="btn btn-info btn-sm" download><i class="fa fa-file-excel"></i>Download</a>
13 <small id="error-kategori_id" class="error-text form-text text-danger"></small>
14 </div>
15 <div class="form-group">
16 <label>Pilih File</label>
17 <input type="file" name="file_barang" id="file_barang" class="form-control" required>
18 <small id="error-file_barang" class="error-text form-text text-danger"></small>
19 </div>
20 </div>
21 <div class="modal-footer">
22 <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
23 <button type="submit" class="btn btn-primary">Upload</button>
24 </div>
25 </div>
26 </div>
27 </form>
28 <script>
29 $(document).ready(function() {
30 $("#form-import").validate({
31 rules: {
32 file_barang: {required: true, extension: ".xlsx"},
33 },
34 submitHandler: function(form) {
35 var formData = new FormData(form); // Jadikan form ke FormData untuk handle file
36 $.ajax({
37 url: form.action,
38 type: form.method,
39 data: formData, // Data yang dikirim berupa FormData
40 processData: false, // setting processData dan contentType ke false, untuk handle file
41 contentType: false,
42 success: function(response) {
43 if(response.status){ // jika sukses
44 $("#myModal").modal('hide');
45 Swal.fire({
46 icon: 'success',
47 title: 'Berhasil',
48 text: response.message
49 });
50 tableBarang.ajax.reload(); // reload datatable
51 }else{ // jika error
52 $(".error-text").text('');
53 $.each(response.msgField, function(prefix, val) {
54 $("#error-"+prefix).text(val[0]);
55 });
56 Swal.fire({
57 icon: 'error',
58 title: 'Terjadi Kesalahan',
59 text: response.message
60 });
61 }
62 }
63 });
64 return false;
65 },
66 errorElement: 'span',
67 errorPlacement: function (error, element) {
68 error.addClass('invalid-feedback');
69 element.closest('.form-group').append(error);
70 },
71 highlight: function (element, errorClass, validClass) {
72 $(element).addClass('is-invalid');
73 },
74 unhighlight: function (element, errorClass, validClass) {
75 $(element).removeClass('is-invalid');
76 }
77 });
78 });
79 </script>
80

```

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

```

187 Route::get('/import', [BarangController::class, 'import']); // ajax form upload excel
188 Route::post('/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

```
PS C:\laragon\www\Web_Lanjut\Minggu_8\PWL_POS> composer require phpoffice/phpspreadsheet
```

7. Selanjutnya kita modifikasi file [BarangController.php](#) untuk memproses data

```
Minggu 8 > PWL_POS > app > Http > Controllers > BarangController.php > PHP Intelephense > BarangController > list
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\BarangModel;
6  use Illuminate\Http\Request;
7  use App\Models\KategoriModel;
8  use PhpOffice\PhpSpreadsheet\IOFactory;
9  use Yajra\DataTables\Facades\DataTables;
10 use Illuminate\Support\Facades\Validator;
11
12 class BarangController extends Controller
13 {
14     // Menampilkan halaman daftar barang
15     public function index()
16     {
17         $breadcrumb = (object) [
18             'title' => 'Daftar Barang',
19             'list' => ['Home', 'Barang']
20         ];
21
22         $page = (object) [
23             'title' => 'Daftar barang dalam sistem'
24         ];
25
26         $activeMenu = 'barang';
27         $kategori = KategoriModel::all();
28
29         return view('barang.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu, 'kategori' => $kategori]);
30     }
31
32     // Ambil data barang dalam bentuk JSON untuk DataTables
33     public function list(Request $request)
34     {
35         $barang = BarangModel::select('barang_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual', 'kategori_id')->with('kategori');
36
37         // if ($request->kategori_id) {
38         //     $barang->where('kategori_id', $request->kategori_id);
39         // }
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             ->addIndexColumn();
48     }
49 }
```

```

48         ->addColumn('aksi', function ($barang) {
49             // $btn = '<a href="' . url('/barang/' . $barang->barang_id) . '" class="btn btn-info btn-sm">Detail</a> ';
50             // $btn .= '<a href="' . url('/barang/' . $barang->barang_id . '/edit') . '" class="btn btn-warning btn-sm">Edit</a> ';
51             // $btn .= '<form class="d-inline-block" method="POST" action="'
52             //     . url('/barang/' . $barang->barang_id) . '">' . csrf_field()
53             //     . method_field('DELETE') . '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\''Apakah Anda yakin menghapus data ini?\')
54             $btn = '<button onclick="modalAction(\'' . url('/barang/' . $barang->barang_id . '/show_ajax') . '\')" class="btn btn-info btn-sm">Detail</button> ';
55             $btn .= '<button onclick="modalAction(\'' . url('/barang/' . $barang->barang_id . '/edit_ajax') . '\')" class="btn btn-warning btn-sm">Edit</button> ';
56             $btn .= '<button onclick="modalAction(\'' . url('/barang/' . $barang->barang_id . '/delete_ajax') . '\')" class="btn btn-danger btn-sm">Hapus</button>';
57             return $btn;
58         })
59         ->rawColumns(['aksi'])
60         ->make(true);
61     }
62
63     // Menampilkan halaman tambah barang
64     public function create()
65     {
66         $breadcrumb = (object) [
67             'title' => 'Tambah Barang',
68             'list' => ['Home', 'Barang', 'Tambah']
69         ];
70
71         $page = (object) [
72             'title' => 'Tambah barang baru'
73         ];
74
75         $activeMenu = 'barang';
76         $kategori = KategoriModel::all();
77
78         return view('barang.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu, 'kategori' => $kategori]);
79     }
80
81     // Menyimpan data barang baru
82     public function store(Request $request)
83     {
84         $request->validate([
85             'barang_kode' => 'required|string|min:2|unique:m_barang,barang_kode',
86             'kategori_id' => 'required|integer',
87             'barang_nama' => 'required|string|max:100',
88             'harga_beli' => 'required|integer',
89             'harga_jual' => 'required|integer'
90         ]);
91
92         BarangModel::create($request->all());
93
94         return redirect('/barang')->with('success', 'Data Barang berhasil disimpan');
95     }
96
97     // Menampilkan detail barang
98     public function show($id)
99     {
100         $barang = BarangModel::findOrFail($id);
101
102         $breadcrumb = (object) [
103             'title' => 'Detail Barang',
104             'list' => ['Home', 'Barang', 'Detail']
105         ];
106
107         $page = (object) [
108             'title' => 'Detail barang'
109         ];
110
111         $activeMenu = 'barang';
112
113         return view('barang.show', ['breadcrumb' => $breadcrumb, 'page' => $page, 'barang' => $barang, 'activeMenu' => $activeMenu]);
114     }
115
116     // Menampilkan halaman edit barang
117     public function edit($id)
118     {
119         $barang = BarangModel::findOrFail($id);
120         $kategori = KategoriModel::all();
121
122         $breadcrumb = (object) [
123             'title' => 'Edit Barang',
124             'list' => ['Home', 'Barang', 'Edit']
125         ];
126
127         $page = (object) [
128             'title' => 'Edit barang'
129         ];
130
131         $activeMenu = 'barang';
132
133         return view('barang.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'barang' => $barang, 'activeMenu' => $activeMenu, 'kategori' => $kategori]);
134     }
135

```



```

136 // Menyimpan perubahan data barang
137 public function update(Request $request, $id)
138 {
139     $request->validate([
140         'barang_kode' => 'required|string|min:2|unique:m_barang,barang_kode,' . $id . ',barang_id',
141         'kategori_id' => 'required|integer',
142         'barang_nama' => 'required|string|max:100',
143         'harga_beli' => 'required|integer',
144         'harga_jual' => 'required|integer'
145     ]);
146
147     BarangModel::findOrFail($id)->update($request->all());
148
149     return redirect('/barang')->with('success', 'Data Barang berhasil diubah');
150 }
151
152 // Menghapus data barang
153 public function destroy($id)
154 {
155     $check = BarangModel::find($id);
156     if (!$check) {
157         return redirect('/barang')->with('error', 'Data Barang tidak ditemukan');
158     }
159     try {
160         BarangModel::destroy($id);
161         return redirect('/barang')->with('success', 'Data Barang berhasil dihapus');
162     } catch (\Illuminate\Database\QueryException $e) {
163         return redirect('/barang')->with('error', 'Data Barang gagal dihapus karena masih terdapat data lain yang terkait');
164     }
165 }
166
167 // Menampilkan modal tambah barang
168 public function create_ajax()
169 {
170     $kategori = KategoriModel::select('kategori_id', 'kategori_nama')->get();
171
172     return view('barang.create_ajax')->with('kategori', $kategori);
173 }
174
175 // Menyimpan data barang baru dengan ajax
176 public function store_ajax(Request $request)
177 {
178     // cek apakah request berupa ajax
179     if ($request->ajax() || $request->wantsJson()) {
180         $rules = [
181             'kategori_id' => ['required', 'integer', 'exists:m_kategori,kategori_id'],
182             'barang_kode' => ['required', 'min:3', 'max:20', 'unique:m_barang,barang_kode'],
183             'barang_nama' => ['required', 'string', 'max:100'],
184             'harga_beli' => ['required', 'numeric'],
185             'harga_jual' => ['required', 'numeric'],
186         ];
187
188         $validator = Validator::make($request->all(), $rules);
189
190         if ($validator->fails()) {
191             return response()->json([
192                 'status' => false,
193                 'message' => 'Validasi Gagal',
194                 'msgField' => $validator->errors()
195             ]);
196         }
197
198         BarangModel::create($request->all());
199
200         return response()->json([
201             'status' => true,
202             'message' => 'Data barang berhasil disimpan'
203         ]);
204     }
205     return redirect('/');
206 }
207
208 // Menampilkan detail barang dengan ajax
209 public function show_ajax(string $id)
210 {
211     $barang = BarangModel::with('kategori')->find($id);
212
213     return view('barang.show_ajax', ['barang' => $barang]);
214 }
215
216 // Menampilkan modal edit barang
217 public function edit_ajax(string $id)
218 {
219     $barang = BarangModel::find($id);
220     $kategori = KategoriModel::select('kategori_id', 'kategori_nama')->get();
221
222     return view('barang.edit_ajax', ['barang' => $barang, 'kategori' => $kategori]);
223 }
224

```

```

226 // Menyimpan perubahan data barang dengan ajax
227 public function update_ajax(Request $request, $id)
228 {
229     if ($request->ajax() || $request->wantsJson()) {
230         $rules = [
231             'kategori_id' => ['required', 'integer', 'exists:m_kategori,kategori_id'],
232             'barang_kode' => ['required', 'min:3', 'max:20', 'unique:m_barang,barang_kode, '. $id .',barang_id'],
233             'barang_nama' => ['required', 'string', 'max:100'],
234             'harga_beli' => ['required', 'numeric'],
235             'harga_jual' => ['required', 'numeric'],
236         ];
237
238         $validator = Validator::make($request->all(), $rules);
239
240         if ($validator->fails()) {
241             return response()->json([
242                 'status' => false,
243                 'message' => 'Validasi gagal.',
244                 'msgField' => $validator->errors()
245             ]);
246         }
247
248         try {
249             BarangModel::find($id)->update($request->all());
250             return response()->json([
251                 'status' => true,
252                 'message' => 'Data berhasil diupdate'
253             ]);
254         } catch (\Illuminate\Database\QueryException $e) {
255             return response()->json([
256                 'status' => false,
257                 'message' => 'Validasi gagal.',
258                 'msgField' => $validator->errors()
259             ]);
260         }
261     }
262     return redirect('/');
263 }
264
265 // Menampilkan modal konfirmasi
266 public function confirm_ajax(string $id)
267 {
268     $barang = BarangModel::find($id);
269
270     return view('barang.confirm_ajax', ['barang' => $barang]);
271 }
272
273 // Menghapus data barang
274 public function delete_ajax(Request $request, $id)
275 {
276     if ($request->ajax() || $request->wantsJson()) {
277         $barang = BarangModel::find($id);
278         if ($barang) {
279             try {
280                 $barang->delete();
281                 return response()->json([
282                     'status' => true,
283                     'message' => 'Data berhasil dihapus'
284                 ]);
285             } catch (\Illuminate\Database\QueryException $e) {
286                 return response()->json([
287                     'status' => false,
288                     'message' => 'Data tidak bisa dihapus'
289                 ]);
290             }
291         } else {
292             return response()->json([
293                 'status' => false,
294                 'message' => 'Data tidak ditemukan'
295             ]);
296         }
297     }
298     return redirect('/');
299 }
300
301 // import data
302 public function import()
303 {
304     return view('barang.import');
305 }
306
307 public function import_ajax(Request $request)
308 {
309     if ($request->ajax() || $request->wantsJson()) {
310         $rules = [
311             // validasi file harus xls atau xlsx, max 1MB
312             'file_barang' => ['required', 'mimes:xlsx', 'max:1024']

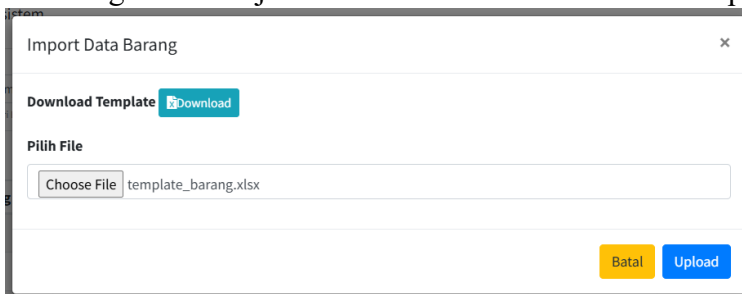
```

```

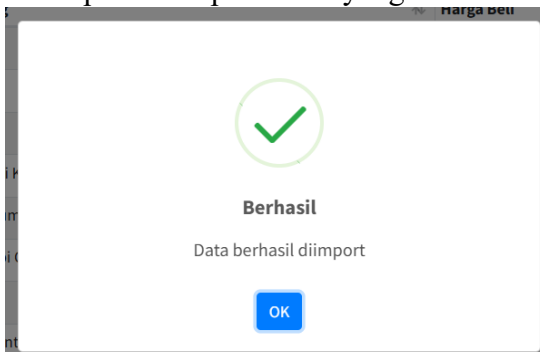
313 ];
314 $validator = Validator::make($request->all(), $rules);
315 if ($validator->fails()) {
316     return response()->json([
317         'status' => false,
318         'message' => 'Validasi Gagal',
319         'msgField' => $validator->errors()
320     ]);
321 }
322 $file = $request->file('file_barang'); // ambil file dari request
323 $reader = IOFactory::createReader('Xlsx'); // load reader file excel
324 $reader->setReadDataOnly(true); // hanya membaca data
325 $spreadsheet = $reader->load($file->getRealPath()); // load file excel
326 $sheet = $spreadsheet->getActiveSheet(); // ambil sheet yang aktif
327 $data = $sheet->toArray(null, false, true, true); // ambil data excel
328 $insert = [];
329 if (count($data) > 1) { // jika data lebih dari 1 baris
330     foreach ($data as $baris => $value) {
331         if ($baris > 1) { // baris ke 1 adalah header, maka lewat
332             $insert[] = [
333                 'kategori_id' => $value['A'],
334                 'barang_kode' => $value['B'],
335                 'barang_nama' => $value['C'],
336                 'harga_beli' => $value['D'],
337                 'harga_jual' => $value['E'],
338                 'created_at' => now(),
339             ];
340         }
341     }
342     if (count($insert) > 0) {
343         // insert data ke database, jika data sudah ada, maka diabaikan
344         BarangModel::insertOrIgnore($insert);
345     }
346     return response()->json([
347         'status' => true,
348         'message' => 'Data berhasil diimport'
349     ]);
350 } else {
351     return response()->json([
352         'status' => false,
353         'message' => 'Tidak ada data yang diimport'
354     ]);
355 }
356 }
357 return redirect('/');
358 }
359 }
360

```

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.



No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori	Aksi
11	SBK-003	Telur Omega(10 butir)	22.000	25.000	Bayi & Anak-anak	Detail Edit Hapus
12	SNK-003	Sari Roti	11.500	12.500	Kecantikan & Kesehatan	Detail Edit Hapus
13	MND-003	Shampo Pantene	17.500	18.500	Makanan & Minuman	Detail Edit Hapus
14	BAY-003	Baju Bayi 2th	89.000	92.500	Perawatan Rumah	Detail Edit Hapus
15	MNM-003	Cleo 600ml	3.750	4.300	Elektronik	Detail Edit Hapus

Data excel berhasil diimportkan.

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. Membuat button/tombol untuk mengakses view import/upload
2. Membuat viewnya dengan file import.blade dan membuat template excelnya yang digunakan untuk import data
3. Download/menambah library phpoffice/phpspreadsheet
4. Menambahkan method di controller yaitu method import untuk mengakses view dan method import_ajax untuk mengimport datanya
5. Membuat routenya di web.php agar bisa melakukan import data.

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

Jawab :

https://github.com/Gilangp/Pemrograman_Web_Lanjut/commit/53fd41b49be729382b8ed9cbcdc5b8e1c4513256

Praktikum 2 – Export Data ke Excel

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

```
<a class="btn btn-sm btn-primary mt-1" href="{{ url('barang/create') }}">Tambah Barang</a>
```

Kita ubah dengan

```
<a href="{{ url('/barang/export_excel') }}" class="btn btn-primary btn-sm mt-1"><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

```
187 Route::get('/import', [BarangController::class, 'import']); // ajax form upload excel
188 Route::post('/import_ajax', [BarangController::class, 'import_ajax']); // ajax import excel
189 Route::get('/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)

```

360 // export excel
361 public function export_excel()
362 {
363     // ambil data barang yang akan di export
364     $barang = BarangModel::select('kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual')
365         ->orderBy('kategori_id')
366         ->with('kategori')
367         ->get();

```

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

```

369 // load library excel
370 $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
371 $sheet = $spreadsheet->getActiveSheet(); // ambil sheet yang aktif
372
373 $sheet->setCellValue('A1', 'No');
374 $sheet->setCellValue('B1', 'Kode Barang');
375 $sheet->setCellValue('C1', 'Nama Barang');
376 $sheet->setCellValue('D1', 'Harga Beli');
377 $sheet->setCellValue('E1', 'Harga Jual');
378 $sheet->setCellValue('F1', 'Kategori');
379
380 $sheet->getStyle('A1:F1')->getFont()->setBold(true);

```

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

```

382 $no = 1; // nomor data dimulai dari 1
383 $baris = 2; // baris data dimulai dari baris ke 2
384 foreach($barang as $key => $value) {
385     $sheet->setCellValue('A' . $baris, $no);
386     $sheet->setCellValue('B' . $baris, $value->barang_kode);
387     $sheet->setCellValue('C' . $baris, $value->barang_nama);
388     $sheet->setCellValue('D' . $baris, $value->harga_beli);
389     $sheet->setCellValue('E' . $baris, $value->harga_jual);
390     $sheet->setCellValue('F' . $baris, $value->kategori->kategori_nama);
391     $baris++;
392     $no++;
393 }

```

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

```

395 foreach(range('A', 'F') as $columnID)
396 {
397     $sheet->getColumnDimension($columnID)->setAutoSize(true); // set auto size untuk kolom
398 }

```

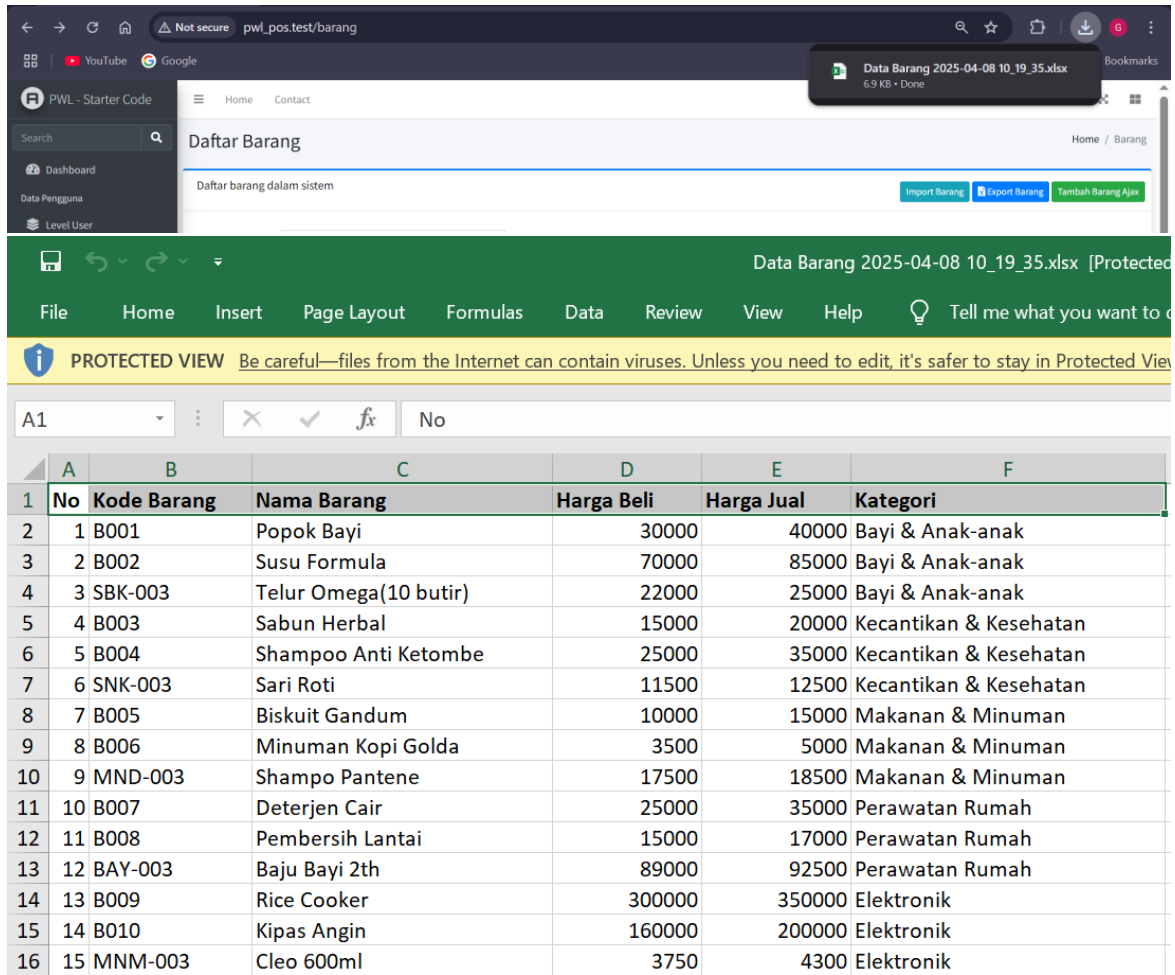
8. Bagian akhir proses export excel adalah kita set nama sheet, dan proses untuk dapat di download oleh pengguna

```

400 $sheet->setTitle('Data Barang'); // set title sheet
401
402 $writer = IOFactory::createWriter($spreadsheet, 'Xlsx');
403 $filename = 'Data Barang ' . date('Y-m-d H:i:s') . '.xlsx';
404
405 header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
406 header('Content-Disposition: attachment;filename="' . $filename . '"');
407 header('Cache-Control: max-age=0');
408 header('Cache-Control: max-age=1');
409 header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
410 header('Last-Modified: ' . gmdate('D, dMY H:i:s') . ' GMT');
411 header('Cache-Control: cache, must-revalidate');
412 header('Pragma: public');
413
414 $writer->save('php://output');
415 exit;
416 }

```

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



The screenshot shows a web application interface for a product list and an Excel spreadsheet displaying the same data. The web application has a sidebar with 'PWL - Starter Code', 'Dashboard', 'Data Pengguna', and 'Level User'. The main content area is titled 'Daftar Barang' and shows a table of products. The Excel spreadsheet is titled 'Data Barang 2025-04-08 10_19_35.xlsx' and is in Protected View. The data is as follows:

No	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
1	B001	Popok Bayi	30000	40000	Bayi & Anak-anak
2	B002	Susu Formula	70000	85000	Bayi & Anak-anak
3	SBK-003	Telur Omega(10 butir)	22000	25000	Bayi & Anak-anak
4	B003	Sabun Herbal	15000	20000	Kecantikan & Kesehatan
5	B004	Shampoo Anti Ketombe	25000	35000	Kecantikan & Kesehatan
6	SNK-003	Sari Roti	11500	12500	Kecantikan & Kesehatan
7	B005	Biskuit Gandum	10000	15000	Makanan & Minuman
8	B006	Minuman Kopi Golda	3500	5000	Makanan & Minuman
9	MND-003	Shampo Pantene	17500	18500	Makanan & Minuman
10	B007	Deterjen Cair	25000	35000	Perawatan Rumah
11	B008	Pembersih Lantai	15000	17000	Perawatan Rumah
12	BAY-003	Baju Bayi 2th	89000	92500	Perawatan Rumah
13	B009	Rice Cooker	300000	350000	Elektronik
14	B010	Kipas Angin	160000	200000	Elektronik
15	MNM-003	Cleo 600ml	3750	4300	Elektronik

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
Jawab :

1. Membuat button/tombol untuk export ke excel
2. Membuat routenya di web.php untuk bisa melakukan export data
3. Membuat method export_excel pada controller

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

Jawab :

https://github.com/Gilangp/Pemrograman_Web_Lanjut/commit/f664656a7261c6cda92d82fc6e93a545efde09b0

Praktikum 3 – Implementasi Export PDF di Laravel dengan dompdf

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

```
PS C:\laragon\www\Web_Lanjut\Minggu 8\PWL_POS> 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 btn-sm mt-1"><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('/import', [BarangController::class, 'import']); // ajax form upload excel
Route::post('/import_ajax', [BarangController::class, 'import_ajax']); // ajax import excel
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](#)

```
419 // export pdf
420 public function export_pdf(){
421     $barang = BarangModel::select('kategori_id','barang_kode','barang_nama','harga_beli','harga_jual')
422         ->orderBy('kategori_id')
423         ->orderBy('barang_kode')
424         ->with('kategori')
425         ->get();
426
427     $pdf = Pdf::loadView('barang.export_pdf', ['barang' => $barang]);
428     $pdf->setPaper('a4','portrait'); // set ukuran kertas dan orientasi
429     $pdf->setOption("isRemoteEnabled", true); // set true jika ada gambar dari url
430     $pdf->render();
431
432     return $pdf->stream('Data Barang '. date('Y-m-d H:i:s'). '.pdf');
433 }
```

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](#)

Minggu 8 > PWL_POS > resources > views > barang > export_pdf.blade.php > html > head > style > img

```
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         }
10        table {
11            width: 100%;
12            border-collapse: collapse;
13        }
14        td, th {
15            padding: 4px 3px;
16        }
17        th{
18            text-align: left;
19        }
20        .d-block{
21            display: block;
22        }
23        img {
24            width: auto;
25            height: 80px;
26            max-width: 150px;
27            max-height: 150px;
28        }
29        .text-right {
30            text-align: right;
31        }
32        .text-center {
33            text-align: center;
34        }
35        .p-1{
36            padding: 5px 1px 5px 1px;
37        }
38        .font-10{
39            font-size: 10pt;
40        }
41        .font-11{
42            font-size: 11pt;
43        }
44        .font-12{
45            font-size: 12pt;
46        }
47        .font-13{
48            font-size: 13pt;
49        }
50        .border-bottom-header{
51            border-bottom: 1px solid;
52        }
53        .border-all, .border-all th, .border-all td{
54            border: 1px solid;
55        }
56    </style>
57 </head>
58 <body>
59     <table class="border-bottom-header">
60         <tr>
61             <td width="15%" class="text-center"></td>
62             <td width="85%">
63                 <span class="text-center d-block font-11 font-bold mb-1">KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI</span>
64                 <span class="text-center d-block font-13 font-bold mb-1">POLITEKNIK NEGERI MALANG</span>
65                 <span class="text-center d-block font-10">Jl. Soekarno-Hatta No. 9 Malang 65141</span>
66                 <span class="text-center d-block font-10">Telepon (0341) 404424 Pes. 101-105, 0341-404420, Fax. (0341) 404420</span>
67                 <span class="text-center d-block font-10">Laman: www.polinema.ac.id</span>
68             </td>
69         </tr>
70     </table>
71
72     <h3 class="text-center">LAPORAN DATA BARANG</h3>
73     <table class="border-all">
74         <thead>
75             <tr>
76                 <th class="text-center">No</th>
77                 <th>Kode Barang</th>
78                 <th>Nama Barang</th>
79                 <th class="text-right">Harga Beli</th>
80                 <th class="text-right">Harga Jual</th>
81                 <th>Kategori</th>
82             </tr>
83         </thead>
84         <tbody>
85             @foreach($barang as $b)
86                 <tr>
87                     <td class="text-center">{{ $loop->iteration }}</td>
88                     <td>{{ $b->barang_kode }}</td>
89                     <td>{{ $b->barang_nama }}</td>
90                     <td class="text-right">{{ number_format($b->harga_beli, 0, ',', '.') }}</td>
91                     <td class="text-right">{{ number_format($b->harga_jual, 0, ',', '.') }}</td>
92                     <td>{{ $b->kategori->kategori_nama }}</td>
```



```

93 </tr>
94 @endforeach
95 </tbody>
96 </table>
97 </body>
98 </html>
99

```

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

Ketika klik button export barang yang pdf

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
Jl. Sockarno-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	B001	Popok Bayi	30.000	40.000	Bayi & Anak-anak
2	B002	Susu Formula	70.000	85.000	Bayi & Anak-anak
3	SBK-003	Telur Omega(10 butir)	22.000	25.000	Bayi & Anak-anak
4	B003	Sabun Herbal	15.000	20.000	Kecantikan & Kesehatan
5	B004	Shampoo Anti Ketombe	25.000	35.000	Kecantikan & Kesehatan
6	SNK-003	Sari Roti	11.500	12.500	Kecantikan & Kesehatan
7	B005	Biskuit Gandum	10.000	15.000	Makanan & Minuman
8	B006	Minuman Kopi Golda	3.500	5.000	Makanan & Minuman
9	MND-003	Shampo Pantene	17.500	18.500	Makanan & Minuman
10	B007	Deterjen Cair	25.000	35.000	Perawatan Rumah
11	B008	Pembersih Lantai	15.000	17.000	Perawatan Rumah
12	BAY-003	Baju Bayi 2th	89.000	92.500	Perawatan Rumah
13	B009	Rice Cooker	300.000	350.000	Elektronik
14	B010	Kipas Angin	160.000	200.000	Elektronik

dan ketika di dwonload

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
Jl. Sockarno-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	B001	Popok Bayi	30.000	40.000	Bayi & Anak-anak
2	B002	Susu Formula	70.000	85.000	Bayi & Anak-anak
3	SBK-003	Telur Omega(10 butir)	22.000	25.000	Bayi & Anak-anak
4	B003	Sabun Herbal	15.000	20.000	Kecantikan & Kesehatan
5	B004	Shampoo Anti Ketombe	25.000	35.000	Kecantikan & Kesehatan
6	SNK-003	Sari Roti	11.500	12.500	Kecantikan & Kesehatan
7	B005	Biskuit Gandum	10.000	15.000	Makanan & Minuman
8	B006	Minuman Kopi Golda	3.500	5.000	Makanan & Minuman
9	MND-003	Shampo Pantene	17.500	18.500	Makanan & Minuman
10	B007	Deterjen Cair	25.000	35.000	Perawatan Rumah
11	B008	Pembersih Lantai	15.000	17.000	Perawatan Rumah
12	BAY-003	Baju Bayi 2th	89.000	92.500	Perawatan Rumah
13	B009	Rice Cooker	300.000	350.000	Elektronik
14	B010	Kipas Angin	160.000	200.000	Elektronik
15	MNM-003	Cleo 600ml	3.750	4.300	Elektronik

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 :

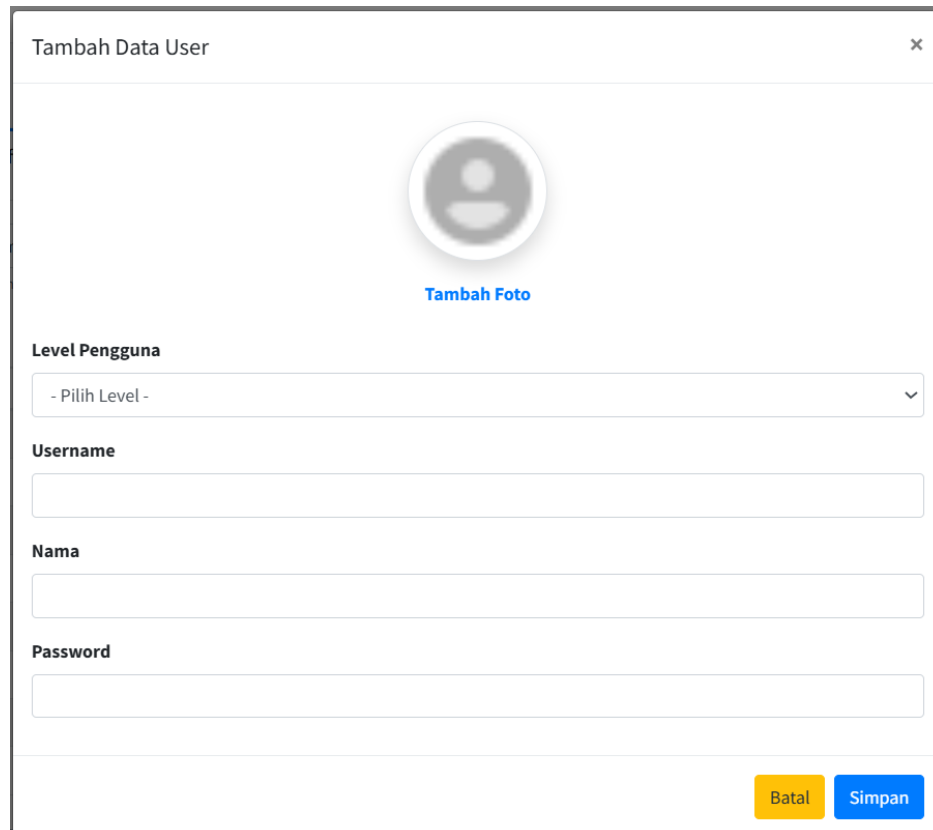
1. Download/menambah library barryvdh/laravel-dompdf
 2. Membuat button/tombol export untuk melakukan export dalam bentuk pdf
 3. Membuat route di web.php agar proses export bisa dilakukan.
 4. Menambahkan method export_pdf di controllernya
 5. Membuat tampilan pdf
3. Submit kode untuk impementasi export pdf pada repository github kalian.

Jawab :


https://github.com/Gilangp/Pemrograman_Web_Lanjut/commit/e2524a946cef861e3001aacc8f2ed75d6fb1d050

Tugas 4 – Implementasi Upload File Gambar

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



Tambah Data User



Tambah Foto

Level Pengguna

- Pilih Level -


Username

Nama

Password


Batal Simpan

Data User



ID	1
Level	Administrator
Username	admin
Nama	Administrator
Password	*****

Edit Data User


Ubah Foto

Level Pengguna

Administrator

Username

admin

Nama

Administrator


Password


Abaikan jika tidak ingin ubah password

Batal

Simpan

Hapus Data User

 **Konfirmasi !!!**
Apakah Anda ingin menghapus data seperti di bawah ini?



Level Pengguna :	Administrator
Username :	admin
Nama :	Administrator

Batal

Ya, Hapus

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

Jawab :

1. Membuat migration baru untuk menambah kolom foto di tabel m_user untuk menyimpan foto

```
Minggu 8 > PWL_POS > database > migrations > 2025_04_10_143039_add_foto_to_m_user_table.php > ...
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::table('m_user', function (Blueprint $table) {
15               $table->string('photo')->nullable()->after('password');
16           });
17       }
18
19       /**
20        * Reverse the migrations.
21        */
22       public function down(): void
23       {
24           Schema::table('m_user', function (Blueprint $table) {
25               $table->dropColumn(['photo']);
26           });
27       }
28   };
29
```

2. Menambahkan atribut photo di fillable agar bisa diisi

```
protected $fillable = ['level_id', 'username', 'nama', 'password', 'photo', 'create_at', 'update_at'];
```

3. Modif dan tambah kode di view user dibagian crudnya agar bisa menambah, mengedit, dan menampilkan foto yang sudah diupload
- create

```
10  <div class="form-group text-center mt-3 mb-4">
11      <input type="file" name="photo" id="photo" class="d-none" accept="image/*">
12      <label for="photo">
13          
15      </label>
16      <label for="photo" class="d-block mt-2 text-primary" style="cursor: pointer;">
17          Tambah Foto
18      </label>
19      <small id="error-foto" class="error-text form-text text-danger"></small>
20  </div>
```

edit

```
31  <div class="form-group text-center mt-3 mb-4">
32      <input type="file" name="photo" id="photo" class="d-none" accept="image/*">
33      <label for="photo">
34          
40      </label>
41      <label for="photo" class="d-block mt-2 text-primary fw-semibold" style="cursor: pointer;">
42          Ubah Foto
43      </label>
44      <small id="error-foto" class="error-text form-text text-danger"></small>
45  </div>
```

Show

```

12 <div class="col-md-4 text-center mb-3">
13 
17 </div>

```

confirm

```

38 <div class="row">
39 <div class="col-md-4 text-center">
40 
44 </div>

```

4. Menambahkan validasi

```
photo: { extension: "jpg|jpeg|png", filesize: 2097152 },
```

5. Tambahkan kode pada UserController bagian store_ajax, update_ajax, dan delete_ajax untuk menyimpan, mengupdate, dan menghapus gambar di storage/app/public/uploads/foto_user

- store_ajax

```

126 $photoPath = null;
127 if ($request->hasFile('photo'))
128 {
129     $photo = $request->file('photo');
130     $originalName = pathinfo($photo->getClientOriginalName(), PATHINFO_FILENAME);
131     $extension = $photo->getClientOriginalExtension();
132     $timestamp = date('Ymd');
133     $fileName = $originalName . '_' . $timestamp . '.' . $extension;
134
135     $photoPath = $photo->storeAs('uploads/foto_user', $fileName, 'public');
136 }
137
138 UserModel::create([
139     'username' => $request->username,
140     'nama' => $request->nama,
141     'password' => bcrypt($request->password),
142     'level_id' => $request->level_id,
143     'photo' => $photoPath
144 ]);

```

- update_ajax

```

256
257 if ($request->filled('password')) {
258     $data['password'] = bcrypt($request->password);
259 }
260
261 if ($request->hasFile('photo')) {
262     if (!empty($user->photo) && Storage::disk('public')->exists($user->photo)) {
263         Storage::disk('public')->delete($user->photo);
264     }
265
266     $photo = $request->file('photo');
267     $fileName = pathinfo($photo->getClientOriginalName(), PATHINFO_FILENAME) . '_' . date('Ymd') . '.' . $photo->getClientOriginalExtension();
268     $photoPath = $photo->storeAs('uploads/foto_user', $fileName, 'public');
269
270     $data['photo'] = $photoPath;
271 }
272
273 $user->update($data);

```

- delete_ajax

```

318 if ($user) {
319     if (!empty($user->photo) && Storage::disk('public')->exists($user->photo)) {
320         Storage::disk('public')->delete($user->photo);
321     }
322
323     $user->delete();

```

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

Jawab :

https://github.com/Gilangp/Pemrograman_Web_Lanjut/commit/43e52e4357927b2faedf1145b59cc0429a5abe22