

Jobsheet 10

Unggah Berkas dan Export PDF

Pendahuluan

Pada praktikum kali ini anda akan mempelajari tentang cara implementasi *upload file* (unggah berkas) lewat form serta menghasilkan laporan berupa PDF dari basis data menggunakan web framework Laravel.

Tujuan Pembelajaran

1. Mahasiswa mampu membuat fitur unggah file dengan Web Framework Laravel
2. Mahasiswa mampu membangun fitur pelaporan berupa export file PDF

Alat dan Bahan

1. PC atau Laptop
2. Text Editor/IDE (rekomendasi PHPStorm atau VSCode)
3. Web Browser
4. PHP

Unggah Berkas

Mengunggah *file* adalah salah satu hal yang dibutuhkan di banyak proyek website. Unggah file dilakukan dengan menyalin *file* yang ada di komputer lokal untuk diletakkan ke direktori server. Pada praktikum ini kita akan mempelajari mengunggah file gambar artikel pada *form* tambah artikel.

Informasi lebih lanjut terkait fitur unggah berkas di Laravel bisa membacanya di sini:

<https://laravel.com/docs/8.x/filesystem#file-uploads>

Praktikum 1: Membuat Form Unggah Berkas

Pada praktikum ini, kita akan membuat tambah data artikel dengan upload gambar melalui form tambah data. Silakan lakukan langkah-langkah praktikum berikut ini.

1. Pastikan Anda sudah berada di direktori project laravel. Silakan membuat table articles dengan perintah migration pada console seperti berikut:

```
PS C:\xampp\htdocs\pwl\07_praktikum_web_lanjut> php artisan make:migration create_articles_table

INFO Migration [C:\xampp\htdocs\pwl\07_praktikum_web_lanjut\database\migrations\2023_05_05_034548_create_articles_table.php] created successfully.

PS C:\xampp\htdocs\pwl\07_praktikum_web_lanjut>
```

2. Ubahlah fungsi up() pada class migration seperti berikut:

```
database > migrations > 2023_05_05_034548_create_articles_table.php
12     public function up(): void
13     {
14         Schema::create('articles', function (Blueprint $table) {
15             $table->id();
16             $table->string('titlr');
17             $table->text('content');
18             $table->string('featured_image');
19             $table->timestamps();
20         });
21     }
```

3. Pastikan web server atau mysql server sedang running (bisa menggunakan Xampp), kemudian lakukan perintah migrate untuk membuat table articles di basis data seperti berikut:

```
database > migrations > 2023_05_05_034548_create_articles_table.php
12     public function up(): void
13     {
14         Schema::create('articles', function (Blueprint $table) {
15             $table->id();
16             $table->string('titlr');
17             $table->text('content');
18             $table->string('featured_image');
19             $table->timestamps();
20         });
21     }
```

4. Selanjutnya kita buat ArticleController beserta modelnya dengan atribut resource untuk melengkapi isi controller dengan fungsi-fungsinya seperti perintah berikut:

```
PS C:\xampp\htdocs\pwl\07_praktikum_web_lanjut> php artisan make:controller ArticleController --resource --model=Article
```

5. Tambahkan fillable pada model Article seperti kode berikut agar dapat menyimpan data ke basis data:

```
app > Models > Article.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Article extends Model
9  {
10     use HasFactory;
11
12     protected $fillable = [
13         'title',
14         'content',
15         'featured_image'
16     ];
17 }
18
```

6. Tambahkan route resource di routes/web.php dengan kode berikut:

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\MahasiswaController;
5  use App\Http\Controllers\ArticleController;
6  use Illuminate\Http\Request;
7
8  Route::resource('mahasiswas', MahasiswaController::class);
9  Route::post('cari', [MahasiswaController::class, 'cari'])->name('cari');
10 Route::get('mahasiswa/nilai/{nim}', [MahasiswaController::class, 'nilai'])->name('r
11
12 Route::resource('articles', ArticleController::class);
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17
```

7. Selanjutnya untuk bagian view, buatlah folder **articles** di **resources/views/articles** lalu buat view baru dengan nama **create.blade.php**. Isi kode seperti berikut (catatan: **layouts.app** adalah layout dari Modul 6):

```
resources > views > articles > create.blade.php
1  @extends('layouts.app')
2  @section('content')
3  <div class="container">
4      <form action="/articles" method="post" enctype="multipart/form-data">
5          @csrf
6          <div class="form-group">
7              <label for="title">Title: </label>
8              <input type="text" class="form-control" required="required" name="title"></br>
9              <label for="content">Content: </label>
10             <textarea type="text" class="form-control" required="required" name="content"></textarea></br>
11             <label for="image">Feature Image: </label>
12             <input type="file" class="form-control" required="required" name="image"></br>
13             <button type="submit" name="submit" class="btn btn-primary float-right">Simpan</button>
14         </div>
15     </form>
16 </div>
17 @endsection
18
```

Penjelasan kode:

`<form action="/articles" method="post" enctype="multipart/formdata">`

Untuk mengunggah berkas, form kita gunakan method POST dan tambahkan atribut `enctype` seperti kode di atas.

`<input type="file" class="form-control" required="required" name="image">`

Untuk input form kita ubah type menjadi jenis “file” agar dapat memilih berkas dari komputer lokal untuk dapat diunggah ke direktori server.

8. Pada `app/Http/Controllers/ArticleController.php` harus dimodifikasi fungsi `store` seperti kode berikut:

```
app > Http > Controllers > ArticleController.php
29  public function store(Request $request)
30  {
31      if ($request->file('image')){
32          $image_name = $request->file('image')->store('images', 'public');
33      }
34
35      Articles::create([
36          'title' => $request->title,
37          'content' => $request->content,
38          'featured_image' => $image_name,
39      ]);
40      return 'Artikel berhasil disimpan';
41  }
```

Penjelasan kode:

File 'image' akan disimpan di folder baru bernama 'images' yang berada di storage/app/public

Misal kita unggah sebuah berkas .png dengan method store('images', 'public') seperti di atas, maka akan ada file baru, contoh: storage/app/public/images/xadfasfjasklfasj.png, extension dari file mengikuti file yang diupload. Nama gambar yang dihasilkan adalah autogenerate berupa random string karena kita menggunakan store().

Sayangnya file-file yang ada di folder storage memang tidak bisa diakses menggunakan URL meskipun di bawah storage/app/public, padahal file upload akan otomatis masuk ke folder storage. Sementara itu file-file yang bisa diakses oleh URL harus diletakkan di dalam folder public aplikasi kita. Untuk itu kita perlu membuat direktori virtual di public/storage yang isinya sama persis (mirroring) dengan direktori storage/app/public.

Lokasi File	storage/app/public/images/XXXXX.png
Mirror file	public/storage/images/XXXXXX.png

Gunakan perintah berikut untuk *mirroring* file tersebut.

```
PS C:\xampp\htdocs\pw1\07_praktikum_web_lanjut> php artisan storage:link
INFO The [C:\xampp\htdocs\pw1\07_praktikum_web_lanjut\public\storage] link has been connected to [C:\xampp\htdocs\pw1\07_praktikum_web_lanjut\storage\app\public].
PS C:\xampp\htdocs\pw1\07_praktikum_web_lanjut>
```

9. Terakhir silakan run **php artisan serve** untuk mengakses form di tautan **localhost:8000/articles/create**. Jika belum tampil, pastikan fungsi create di **ArticleController** sudah dibuat seperti kode berikut:

```
app > Http > Controllers > 🐘 ArticleController.php
21     public function create(Request $request)
22     {
23         return view('articles.create');
24     }
```

10. Buat laporan praktikum dan push ke repository masing-masing.

Hasil Tampilan



← → ↻ 🏠 localhost:8000/articles/create 🔍 ⌵ ⚙️ 🧑

Title:

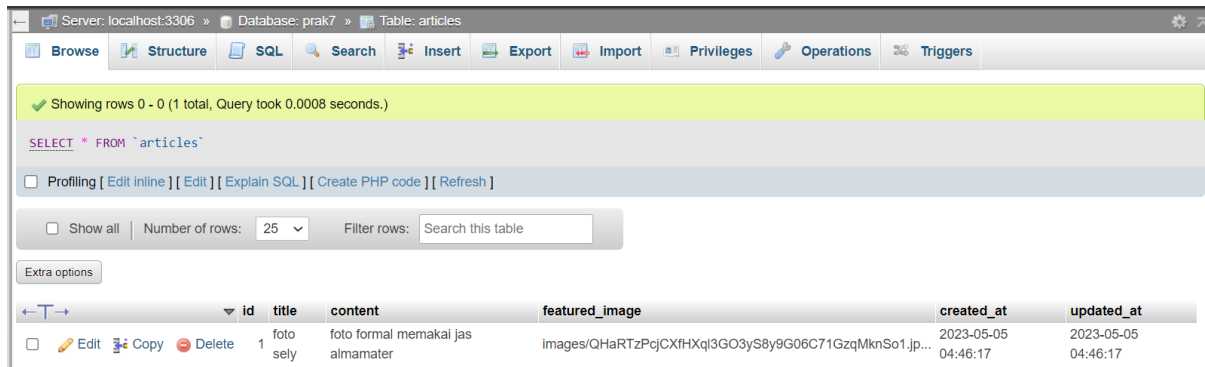
Content:

Feature Image:
 No file chosen

Tampilan ketika sudah berhasil menambahkan data artikel



Data sudah berhasil ditambahkan ke database



Praktikum 2: Membuat Form *Update* Berkas

Pada bagian ini kita akan mengubah bagian update (form update dan fungsi update di controller) agar dapat mengubah featured_image pada tabel articles.

1. Pertama, kita buat view baru untuk edit article dengan nama `resources/views/articles/edit.blade.php` dengan isi kode sebagai berikut:

```
resources > views > articles > edit.blade.php
1  @extends('articles.layout')
2  @section('content')
3  <div class="container">
4      <form action="/articles/{{ $article->id }}" method="post" enctype="multipart/form-data">
5          @csrf
6          <div class="form-group">
7              <label for="title">Judul</label>
8              <input type="text" class="form-control" required="required" name="title" value="{{ $article->title }}"></div>
9          <div class="form-group">
10             <label for="content">Content</label>
11             <input type="text" class="form-control" required="required" name="content" value="{{ $article->content }}"></div>
12             <div class="form-group">
13                 <label for="image">Feature Image</label>
14                 <input type="file" class="form-control" required="required" name="image" value="{{ $article->featured_image }}"></div>
15                 
16             <button type="submit" class="btn btn-primary float-right">Ubah Data</button>
17         </form>
18     </div>
19 @endsection
```

Penjelasan kode:

```
<form action="/articles/{{ $article->id }}" method="post"
enctype="multipart/form-data">
```

Kode tersebut akan memperbarui data article sesuai id yang diperoleh dari parameter URI di browser.

```
value="{{ $article->title }}"
```

Bagian form field tiap value akan mengisi data sebelumnya yang ada di basis data.

```

```

Bagian ini menampilkan gambar yang tersimpan di basis data sebelumnya dengan maksimal lebar=150px.

3. Karena tipe input berubah menjadi *file*, maka proses UPDATE data pada **ArticleController** harus dimodifikasi menjadi seperti berikut.
Pahami maksud kode di atas, lalu jelaskan dalam laporan praktikum Anda!
4. Jangan lupa bagian fungsi edit untuk menampilkan form edit pada **ArticleController** perlu diubah menjadi seperti berikut.

```

app > Http > Controllers > ArticleController.php
64 public function update(Request $request, $id)
65 {
66     $article = Article::find($id);
67
68     $article->title = $request->title;
69     $article->content = $request->content;
70
71     if ($article->featured_image && file_exists(storage_path('app/public/' . $article->featured_image))) {
72         \Storage::delete('public/' . $article->featured_image);
73     }
74     $image_name = $request->file('image')->store('images', 'public');
75     $article->featured_image = $image_name;
76
77     $article->save();
78     return 'Artikel berhasil diubah';
79 }
80

app > Http > Controllers > ArticleController.php
54 public function edit($id)
55 {
56     $article = Article::find($id);
57
58     return view('articles.edit', ['article' => $article]);
59 }

```

Kemudian coba akses tautan untuk mengedit artikel dengan id=1 misalnya <http://localhost:8000/articles/1/edit>

Silakan coba ganti gambar lalu klik “Ubah Data”, apa yang terjadi?

Tampilan ketika akan mengedit data pada id = 1

Awalnya tidak dapat diubah karena belum mendukung method PUT, maka lanjut ke poin 5.

5. Jika belum berhasil untuk mengubah data article, maka perlu dicek route list dari resources **articles** dengan perintah berikut:

```
PS C:\xampp\htdocs\pwl\07_praktikum_web_lanjut> php artisan route:list --name=articles

GET|HEAD articles ..... articles.index > ArticleController@index
POST articles ..... articles.store > ArticleController@store
GET|HEAD articles/create ..... articles.create > ArticleController@create
GET|HEAD articles/{article} ..... articles.show > ArticleController@show
PUT|PATCH articles/{article} ..... articles.update > ArticleController@update
DELETE articles/{article} ..... articles.destroy > ArticleController@destroy
GET|HEAD articles/{article}/edit ..... articles.edit > ArticleController@edit

Showing [7] routes
```

Dari daftar route tersebut, terlihat bahwa `articles.update` hanya mendukung method PUT dan PATCH. Sedangkan pada form edit yang kita buat belum menambahkan method tersebut secara implisit. Karena form HTML tidak dapat membuat request PUT, PATCH, atau DELETE, maka diperlukan tambahan field method tersembunyi pada form. Silakan tambahkan `@method('PUT')` pada view: `resources/views/articles/edit.blade.php` pada bagian bawah form action seperti berikut:

```
resources > views > articles > edit.blade.php
1 @extends('articles.layout')
2 @section('content')
3 <div class="container">
4     <form action="/articles/{{$article->id}}" method="post" enctype="multipart/form-data">
5         @method('PUT')
6         @csrf
```

Catatan: Dokumentasi dapat dibaca pada tautan <https://laravel.com/docs/8.x/blade#methodfield>

6. Silakan coba kembali lakukan edit article dengan mengubah judul, content, maupun unggah gambar baru. Laporkan hasil praktikum Anda dan push ke repository masing-masing.

Ulangi untuk melakukan edit data dengan id = 1 tersebut, disini mencoba untuk mengganti foto pada kolom `featured_image`

Maka, data sudah dapat diupdate

```
localhost:8000/articles/1
Artikel berhasil diubah
```

Tampilan ketika data sudah diupdate

← → ↺ 🏠

🔍 📄 ☆ ⚙️ 🖨️ 👤 ⋮

📍 localhost:8000/articles/1/edit

Judul


foto sely

Content

foto formal memakai jas almamater

Feature Image

Choose File No file chosen



Ubah Data

Reporting (Export ke PDF)

Reporting adalah salah satu fitur yang dibutuhkan pada suatu website untuk kebutuhan cetak data ataupun export data. Report (laporan) biasanya berformat PDF. Untuk membuat laporan dalam PDF, kita menggunakan DOMPDF. DOMPDF adalah library PHP yang bisa digunakan untuk membuat file PDF. DOMPDF akan mengubah halaman menjadi file PDF.

Praktikum 3: Membuat Fitur Export PDF

1. Install DOMPDF pada project Laravel Anda menggunakan perintah berikut.

```
PS C:\xampp\htdocs\pwl\07_praktikum_web_lanjut> composer require barryvdh/laravel-dompdf
Info from https://repo.packagist.org: #StandWithUkraine
./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 (v2.0.1)
- Locking dompdf/dompdf (v2.0.3)
- Locking masterminds/html5 (2.8.0)
- Locking phenx/php-font-lib (0.5.4)
- Locking phenx/php-svg-lib (0.5.0)
- Locking sabberworm/php-css-parser (8.4.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 6 installs, 0 updates, 0 removals
- Downloading sabberworm/php-css-parser (8.4.0)
- Downloading phenx/php-svg-lib (0.5.0)
- Downloading phenx/php-font-lib (0.5.4)
- Downloading masterminds/html5 (2.8.0)
```

2. Setelah berhasil install, buka **config/app.php**, tambahkan package DomPDF agar dikenali oleh laravel.

```
config > 🐼 app.php
156
157     'providers' => [
158         Barryvdh\DomPDF\ServiceProvider::class,
159     ],

config > 🐼 app.php
212     'aliases' => Facade::defaultAliases()->merge([
213         // 'ExampleClass' => App\Example\ExampleClass::class,
214         'PDF' => Barryvdh\DomPDF\Facade::class,
215     ])->toArray(),
```

Untuk membaca dokumentasi lebih lanjut terkait library DomPDF ini dapat mengaksesnya di folder: **vendor/barryvdh/laravel-dompdf/readme.md**

3. Kemudian buat Route baru di **routes/web.php** yang digunakan untuk menjalankan fungsi cetak pdf.

```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\MahasiswaController;
5  use App\Http\Controllers\ArticleController;
6  use Illuminate\Http\Request;
7
8  Route::resource('mahasiswas', MahasiswaController::class);
9  Route::post('cari', [MahasiswaController::class, 'cari'])->name('cari');
10 Route::get('mahasiswa/nilai/{nim}', [MahasiswaController::class, 'nilai'])->name('nilai');
11 Route::get('/article/cetak_pdf', [ArticleController::class, 'cetak_pdf']);
12 Route::get('/mahasiswas/{nim}/exportPDF', [MahasiswaController::class, 'exportPDF'])->name('mahasiswas.exportPDF');
13
14
15 Route::resource('articles', ArticleController::class);
16
17 Route::get('/', function () {
18     return view('welcome');
19 });
20
```

5. Tambahkan kode berikut pada **ArticleController** untuk menyertakan package DOMPDF.

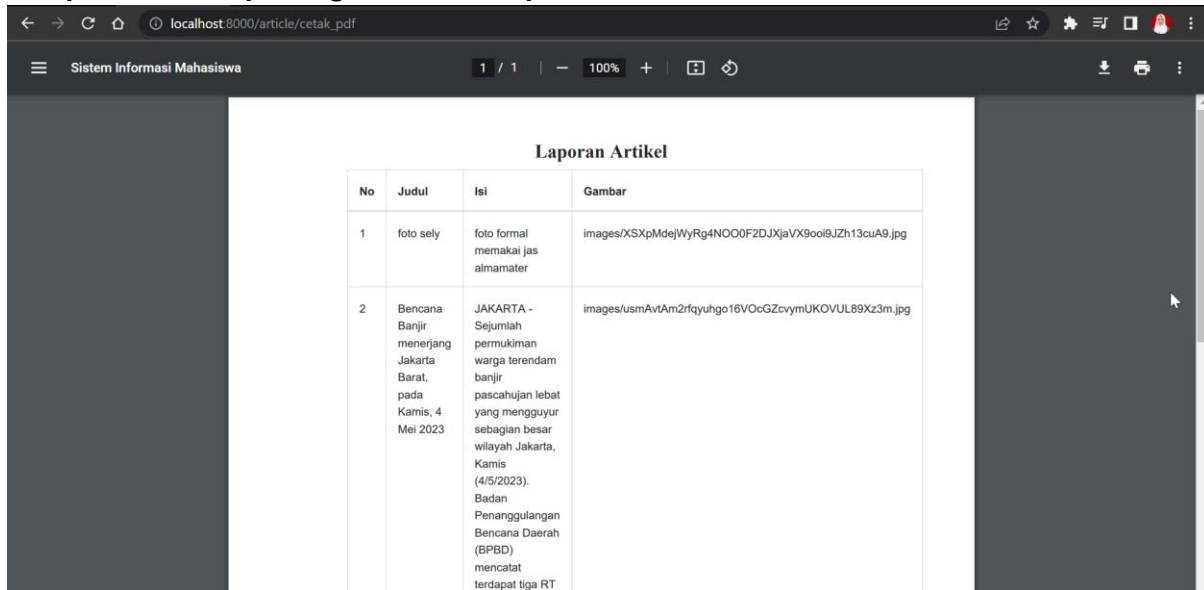
```
use PDF;
```

6. Buat view baru di resources/views/articles/**articles_pdf.blade.php** untuk menampilkan hasil PDF.

```
resources > views > articles > articles_pdf.blade.php
1  @extends('articles.layout')
2  @section('content')
3  <!DOCTYPE html>
4  <html>
5      <head>
6          <title>Membuat Laporan PDF Dengan DOMPDF Laravel</title>
7      </head>
8      <body>
9          <style type="text/css">
10             table tr td,
11             table tr th{
12                 font-size: 9pt;
13             }
14          </style>
15          <center>
16              <h5>Laporan Artikel</h4>
17          </center>
18          <table class='table table-bordered'>
19              <thead>
20                  <tr>
21                      <th>No</th>
22                      <th>Judul</th>
23                      <th>Isi</th>
24                      <th>Gambar</th>
25                  </tr>
26              </thead>
27              <tbody>
28                  @foreach($articles as $a)
29                      <tr>
30                          <td>{{ $loop->iteration }}</td>
31                          <td>{{ $a->title }}</td>
32                          <td>{{ $a->content }}</td>
33                          <td>{{ $a->featured_image }}</td>
34                      </tr>
35                  @endforeach
36              </tbody>
37          </table>
38      </body>
39  </html>
40  @endsection
```

Silakan coba akses http://localhost:8000/article/cetak_pdf untuk melihat hasilnya. Laporkan hasil praktikum Anda dan push di repository masing-masing.

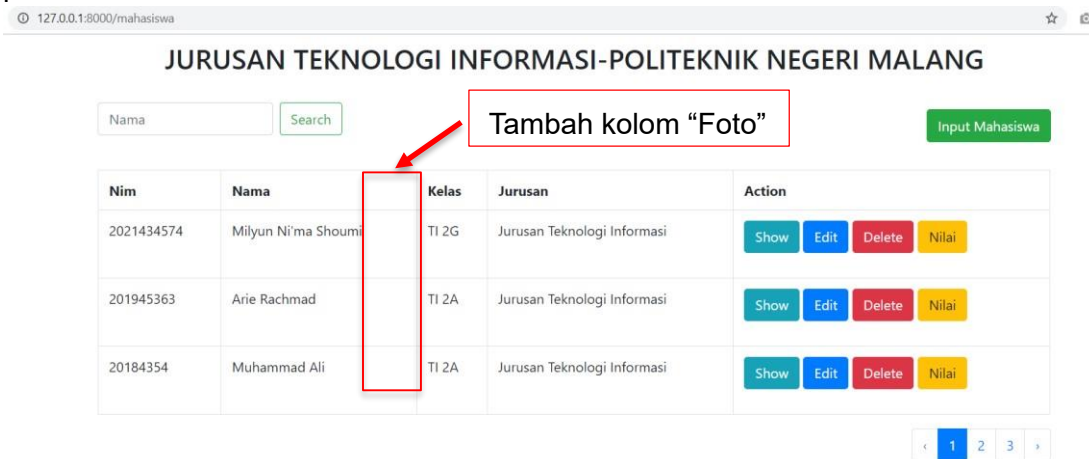
Tampilan hasil reporting untuk cetak pdf



No	Judul	Isi	Gambar
1	foto sely	foto formal memakai jas almamater	images/XSxpMdeJWyRg4NOO0F2DJXjaVX9ooi9JZh13cuA9.jpg
2	Bencana Banjir menerjang Jakarta Barat, pada Kamis, 4 Mei 2023	JAKARTA - Sejumlah permukiman warga terendam banjir pascahunian lebat yang mengguyur sebagian besar wilayah Jakarta, Kamis (4/5/2023). Badan Penanggulangan Bencana Daerah (BPBD) mencatat terdapat tiga RT	images/usmAvtAm2rfqyuhgo16VOcGZcymUKOVUL89Xz3m.jpg

Tugas Praktikum

1. Tambahkan fitur untuk mengunggah foto profil mahasiswa berdasarkan Jobsheet 7 dan 9 pada tampilan input mahasiswa, edit, dan detail mahasiswa. Kemudian foto tiap mahasiswa ditampilkan setelah kolom "Nama" dengan ukuran kecil maksimal 100x100 pixels.



JURUSAN TEKNOLOGI INFORMASI-POLITEKNIK NEGERI MALANG

Nama

Nim	Nama	Foto	Kelas	Jurusan	Action
2021434574	Milyun Ni'ma Shoum		TI 2G	Jurusan Teknologi Informasi	<input type="button" value="Show"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Nilai"/>
201945363	Arie Rachmad		TI 2A	Jurusan Teknologi Informasi	<input type="button" value="Show"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Nilai"/>
20184354	Muhammad Ali		TI 2A	Jurusan Teknologi Informasi	<input type="button" value="Show"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Nilai"/>

< 1 2 3 >

Jawab:

- a. Membuat kolom baru, bernama foto pada table mahasiswa di database

Table structure										Relation view		
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action			
<input type="checkbox"/>	1 nim	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More			
<input type="checkbox"/>	2 nama	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More			
<input type="checkbox"/>	3 foto	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More			
<input type="checkbox"/>	4 jurusan	varchar(25)	utf8mb4_unicode_ci		No	None			Change Drop More			
<input type="checkbox"/>	5 no_handphone	varchar(15)	utf8mb4_unicode_ci		No	None			Change Drop More			
<input type="checkbox"/>	6 email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More			
<input type="checkbox"/>	7 tgl_lahir	date			No	None			Change Drop More			
<input type="checkbox"/>	8 kelas_id	bigint		UNSIGNED	Yes	NULL			Change Drop More			
<input type="checkbox"/>	9 created_at	timestamp			Yes	NULL			Change Drop More			
<input type="checkbox"/>	10 updated_at	timestamp			Yes	NULL			Change Drop More			

- b. Pada view mahasiswa.index, menambahkan satu kolom foto, serta kode pemanggilan untuk menampilkan data dari kolom foto yang telah dibuat sebelumnya.

```
resources > views > mahasiswa > index.blade.php
29 <table class="table table-bordered">
30   <tr>
31     <th>Nim</th>
32     <th>Nama</th>
33     <th>Foto</th>
34     <th>Kelas</th>
35     <th>Jurusan</th>
36     <th>No Handphone</th>
37     <th>Email</th>
38     <th>Tanggal Lahir</th>
39     <th width="280px">Action</th>
40   </tr>
41
42   @foreach ($paginate as $mhs)
43     <tr>
44       <td>{{ $mhs->nim }}</td>
45       <td>{{ $mhs->nama }}</td>
46       <td></td>
47       <td>{{ $mhs->kelas->nama_kelas }}</td>
48       <td>{{ $mhs->jurusan }}</td>
49       <td>{{ $mhs->no_handphone }}</td>
50       <td>{{ $mhs->email }}</td>
51       <td>{{ $mhs->tgl_lahir }}</td>
52       <td>
```

- c. Kemudian pada function store pada MahasiswaController perlu ada penyesuaian kode, sebagaimana berikut

```

app > Http > Controllers > MahasiswaController.php
47 public function store(Request $request)
48 {
49     // dd($request->file('foto'));
50     //melakukan validasi data
51     $request->validate([
52         'nim' => 'required',
53         'nama' => 'required',
54         'foto' => 'required|image|max:2048',
55         'kelas' => 'required',
56         'jurusan' => 'required',
57         'no_handphone' => 'required',
58         'email' => 'required',
59         'tgl_lahir' => 'required',
60     ]);
61
62     $imageName = time().'.'.$request->foto->extension();
63     $request->foto->move(public_path('storage'), $imageName);
64
65     $mahasiswa = new Mahasiswa;
66     $mahasiswa->nim = $request->get('nim');
67     $mahasiswa->nama = $request->get('nama');
68     $mahasiswa->foto = $imageName;
69     $mahasiswa->jurusan = $request->get('jurusan');
70     $mahasiswa->no_handphone = $request->get('no_handphone');
71     $mahasiswa->email = $request->get('email');
72     $mahasiswa->tgl_lahir = $request->get('tgl_lahir');
73     $mahasiswa->save();
74
75     $kelas = new Kelas;
76     $kelas->id = $request->get('kelas');
77
78     //fungsi eloquent untuk menambah data dengan relasi belongsTo
79     $mahasiswa->kelas()->associate($kelas);
80     $mahasiswa->save();
81
82     //jika data berhasil ditambahkan, akan kembali ke halaman utama
83     return redirect()->route('mahasiswas.index')->with('success', 'Mahasiswa Berhasil Ditambahkan');
84

```

- d. Pada function update juga ada penyesuaian kode program


```

app > Http > Controllers > MahasiswaController.php
112 public function update(Request $request, string $nim)
113 {
114     // dd($request->file('foto'));
115
116
117     //melakukan validasi data
118     $request->validate([
119         'nim' => 'required',
120         'nama' => 'required',
121         'foto' => 'required',
122         'kelas' => 'required',
123         'jurusan' => 'required',
124         'no_handphone' => 'required',
125         'email' => 'required',
126         'tgl_lahir' => 'required',
127     ]);
128
129     $mahasiswa = Mahasiswa::with('kelas')->where('nim', $nim)->first();
130     $mahasiswa->nim = $request->get('nim');
131     $mahasiswa->nama = $request->get('nama');
132     $mahasiswa->jurusan = $request->get('jurusan');
133     $mahasiswa->no_handphone = $request->get('no_handphone');
134     $mahasiswa->email = $request->get('email');
135     $mahasiswa->tgl_lahir = $request->get('tgl_lahir');
136     $mahasiswa->save();
137
138     if($request->file('foto')){
139         // hapus foto lama jika ada foto baru yang diupload
140         if($mahasiswa->foto && file_exists(storage_path('app/public/'.$mahasiswa->foto))){
141             \Storage::delete('public/'.$mahasiswa->foto);
142         }
143         // simpan foto baru ke direktori penyimpanan
144         $file = $request->file('foto');
145         $nama_file = $file->getClientOriginalName();
146         $file->storeAs('public/foto', $nama_file);
147         // simpan nama file foto ke dalam kolom 'foto' pada tabel 'mahasiswas'
148         $mahasiswa->foto = $nama_file;
149     }
150     $image_name = $request->file('foto')->store('images', 'public');
151     $mahasiswa->foto = $image_name;
152
153     $kelas = new Kelas;
154     $kelas->id = $request->get('kelas');
155
156     $mahasiswa->kelas()->associate($kelas);
157     $mahasiswa->save();
158
159     return redirect()->route('mahasiswas.index')->with('success', 'Mahasiswa Berhasil Diupdate');
160 }

```

- e. Lanjut ke bagian view mahasiswas. Terdapat beberapa kode program yang harus ditambahkan.

View mahasiswas.create

- Pada tag form, pastikan telah menambahkan enctype.

```

resources > views > mahasiswas > create.blade.php
19
20 <form method="post" action="{{ route('mahasiswas.store') }}" id="myForm" enctype="multipart/form-data">

```

- Menambahkan kolom untuk menginputkan file foto

```

resources > views > mahasiswas > create.blade.php
30 <div class="form-group">
31     <label for="foto">Foto Mahasiswa: </label>
32     <input type="file" class="form-control" required="required" name="foto"></br>
33 </div>

```

View mahasiswa.edit

- Pada tag form, pastikan telah menambahkan enctype.

```
resources > views > mahasiswa > edit.blade.php
18 <form method="post" action="{{ route('mahasiswa.update', $mahasiswa->nim) }}" id="myForm" enctype="multipart/form-data">
19 @csrf
```

- Menambahkan kolom input untuk mengupdate file foto






```
resources > views > mahasiswa > edit.blade.php
29 <div class="form-group">
30 <label for="foto">Foto Mahasiswa</label>
31 <input type="file" class="form-control" required="required" name="foto"><br>
32 
33 </div>
```

Tampilan ketika telah ditambahkan kolom foto

← → ↻ 🏠 🌐 localhost:8000/mahasiswas

JURUSAN TEKNOLOGI INFORMASI-POLITEKNIK NEGERI MALANG

Cari Mahasiswa

Nim	Nama	Foto	Kelas	Jurusan	No Handphone	Email	Tanggal Lahir	Action
2141720005	Sely Ruli Amanda		TI 2A	Teknologi Informasi	085733744803	selyra.123@gmail.com	2002-08-01	<div>Show</div> <div>Edit</div> <div>Delete</div> <div>Nilai</div>
2141720036	Lailatul Badriyah		TI 2A	Teknologi Informasi	088769573990	lailatul24@gmail.com	2003-04-24	<div>Show</div> <div>Edit</div> <div>Delete</div> <div>Nilai</div>
2141720043	Nadya Putri Amalia		TI 2A	Teknologi Informasi	087665667890	nadyaamalias@gmail.com	2003-01-29	<div>Show</div> <div>Edit</div> <div>Delete</div> <div>Nilai</div>
2141720052	Ulfi Mustatiq Abidatul Izza		TI 2H	Teknologi Informasi	087665432908	ulfiizza27@gmail.com	2003-04-27	<div>Show</div> <div>Edit</div> <div>Delete</div> <div>Nilai</div>
2141720070	Michigae Truedaction		TI 2I	Teknologi Informasi	0823342178998	2141720005@student.polinema.ac.id	2002-10-09	<div>Show</div> <div>Edit</div> <div>Delete</div> <div>Nilai</div>

← → ↻ 🏠 🌐 localhost:8000/mahasiswas

2. Tambahkan tombol untuk cetak ke PDF pada tampilan KHS berikut berdasarkan tugas Jobsheet 9.

127.0.0.1:8000/mahasiswa/nilai/201945363

JURUSAN TEKNOLOGI INFORMASI-POLITEKNIK NEGERI MALANG

KARTU HASIL STUDI (KHS)

Nama: Arie Rachmad
NIM: 201945363
Kelas: TI 2A

Mata Kuliah	SKS	Semester	Nilai
Pemrograman Berbasis Objek	3	4	A
Pemrograman Web Lanjut	3	4	A
Basis Data Lanjut	3	4	A
Praktikum Basis Data Lanjut	3	4	A

Cetak ke PDF

Jawab:

- a. Menambahkan routing untuk melakukan cetak pdf dari tabel mahasiswa

```
routes > web.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\MahasiswaController;
5 use App\Http\Controllers\ArticleController;
6 use Illuminate\Http\Request;
7
8 Route::resource('mahasiswas', MahasiswaController::class);
9 Route::post('cari', [MahasiswaController::class, 'cari'])->name('cari');
10 Route::get('mahasiswa/nilai/{nim}', [MahasiswaController::class, 'nilai'])->name('nilai');
11 Route::get('articles/cetak_pdf', [ArticleController::class, 'cetak_pdf']);
12 Route::get('/mahasiswas/{nim}/exportPDF', [MahasiswaController::class, 'exportPDF'])->name('mahasiswas.exportPDF');
13
14
15 Route::resource('articles', ArticleController::class);
16
17 Route::get('/', function () {
18     return view('welcome');
19 });
```

- b. Membuat function exportPDF() pada MahasiswaController

```
app > Http > Controllers > MahasiswaController.php
191
192 public function exportPDF($nim)
193 {
194     $mahasiswas = Mahasiswa::with('matakuliah')->where('nim', $nim)->first();
195     $nilai = DB::table('mahasiswa_mata_kuliah')
196         ->join('matakuliah', 'matakuliah.id', '=', 'mahasiswa_mata_kuliah.mata_kuliah_id')
197         ->where('mahasiswa_mata_kuliah.mahasiswa_nim', $nim)
198         ->select('nilai')
199         ->get();
200     $pdf = PDF::loadView('mahasiswas.nilai_pdf', ['mahasiswas' => $mahasiswas, 'nilai' => $nilai]);
201     return $pdf->download('KHS-' . $mahasiswas->nama . '.pdf');
```

- c. Menambahkan button Cetak ke PDF pada view mahasiswa.nilai

```
resources > views > mahasiswa > nilai.blade.php
41 <div class="text-center mt-5">
42     <a class="btn btn-primary" href="{{ route('mahasiswas.exportPDF', $mahasiswas->nim) }}" target="_blank">Cetak ke PDF</a>
43 </div>
```

- d. Membuat view mahasiswa.nilai_pdf untuk menampilkan hasil reporting sebelum dicetak

```
resources > views > mahasiswaas > ❏ nilai_pdf.blade.php
1 @extends('mahasiswa.layout')
2 @section('content')
3 <!DOCTYPE html>
4 <html>
5     <head>
6         <title>KHS Mahasiswa</title>
7     </head>
8     <body>
9         <div class="row">
10             <div class="col-lg-12 margin-tb" style="justify-content: center; align-items: center; text-align: center;">
11                 <div class="pull-left mt-2">
12                     <h2>JURUSAN TEKNOLOGI INFORMASI-POLITEKNIK NEGERI MALANG</h2>
13                     <h1 style="margin-top: 30px">KARTU HASIL STUDI (KHS)</h1>
14                 </div>
15             </div>
16             <div class="left-text" style="margin-top: 40px">
17                 <p><b>Nama : </b>{{ $mahasiswa->nama }}</p>
18                 <p><b>NIM : </b>{{ $mahasiswa->nim }}</p>
19                 <p><b>Kelas : </b>{{ $mahasiswa->kelas->nama_kelas }}</p>
20             </div>
21             <table class="table table-bordered">
22                 <tr>
23                     <th>Mata Kuliah</th>
24                     <th>SKS</th>
25                     <th>Semester</th>
26                     <th>Nilai</th>
27                 </tr>
28             </table>
29         </body>
30     </html>
31 @endsection

resources > views > mahasiswaas > ❏ nilai_pdf.blade.php
28 @foreach ($mahasiswa->matakuliah as $mhs)
29     <tr>
30         <td>{{ $mhs->nama_matkul }}</td>
31         <td>{{ $mhs->sks }}</td>
32         <td>{{ $mhs->semester }}</td>
33         @php
34             $n = $nilai->where('mahasiswa_nim', $mhs->nim)->first();
35         @endphp
36         <td>
37             @if($nilai)
38                 {{ $n->nilai }}
39             @else
40                 Nilai belum diisi
41             @endif
42         </td>
43     </tr>
44 @endforeach
45 </table>
46 </body>
47 </html>
48 @endsection
```

Tampilan KHS mahasiswa tertentu, sesuai dengan data mahasiswa mana yang ingin dilihat KHS – nya

← → ↺ ⏸ 🔍 📄 ☆ 🛠️ 🚫 📢

localhost:8000/mahasiswa/nilai/2141720005

JURUSAN TEKNOLOGI INFORMASI-POLITEKNIK NEGERI MALANG

KARTU HASIL STUDI (KHS)

Nama : Sely Ruli Amanda

NIM : 2141720005

Kelas : TI 2A

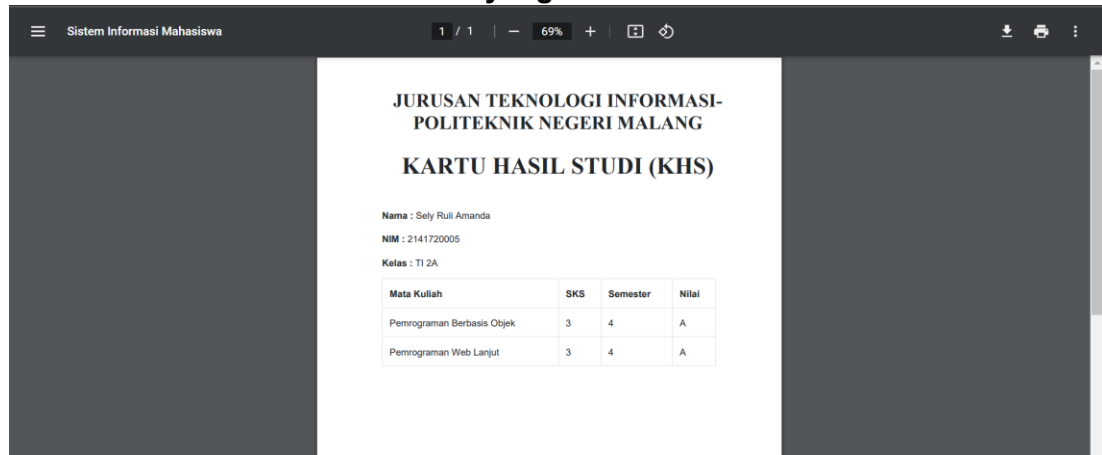
Mata Kuliah	SKS	Semester	Nilai
Pemrograman Berbasis Objek	3	4	A
Pemrograman Web Lanjut	3	4	A

Cetak ke PDF

KHS-Sely Ruli Ama...pdf

Show all

Klik Cetak ke PDF untuk mencetak KHS dan secara otomatis mendownload file tersebut. Kemudian buka file PDF yang telah berhasil didownload.



The screenshot shows a web application titled "Sistem Informasi Mahasiswa". The main content area displays a "KARTU HASIL STUDI (KHS)" for the "JURUSAN TEKNOLOGI INFORMASI-POLITEKNIK NEGERI MALANG". The student's name is "Sely Ruli Amanda" with NIM "2141720005" and class "TI 2A". A table lists the courses and their results:

Mata Kuliah	SKS	Semester	Nilai
Penrograman Berbasis Objek	3	4	A
Penrograman Web Lanjut	3	4	A

~~Semangat Mengerjakan!~~