

LAPORAN
HandsOut(Jobsheet) - Pemrograman Web – Unggah Berkas dan Export
PDF

Fendi Zulkarnain



VSGA X TSA
POLITEKNIK NEGERI MALANG KAMPUS LUMAJANG
Jl. Lintas Timur, Area Sawah/Kebun, Jogotrunan, Kec. Lumajang, Kabupaten Lumajang,
Jawa Timur 67314
2024

Praktikum 1 : Membuat Form Unggah Berkas

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

```
W I N D O W S  X@DESKTOP-6E8EN92 MINGW64 /c/xampp/htdocs/Test/form_file
$ php artisan make:migration create_articles_table --create=articles
```

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

```
public function up(): void
{
    Schema::create('articles', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->text('content');
        $table->string('featured_image');
        $table->timestamps();
    });
}
```

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:

```
W I N D O W S  X@DESKTOP-6E8EN92 MINGW64 /c/xampp/htdocs/Test/form_file
$ php artisan migrate
```

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

```
W I N D O W S  X@DESKTOP-6E8EN92 MINGW64 /c/xampp/htdocs/Test/form_file
$ php artisan make:controller ArticleController --resource --model=Article
```

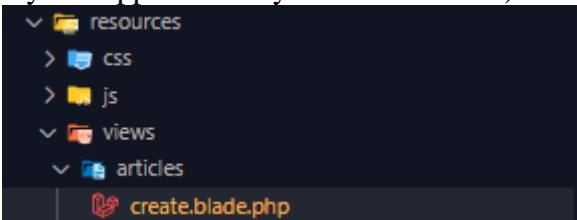
5. Tambahkan fillable pada model Article seperti kode berikut agar dapat menyimpan data ke basis data:

```
class Article extends Model
{
    use HasFactory;
    /**
     * @var array
     */
    protected $fillable = [
        'title',
        'content',
        'featured_image'
    ];
}
```

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

```
Route::resource('/articles', ArticleController::class);
```

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
3 @section('content')
4     <div class="container mt-5 p-3 bg-light shadow">
5         <form action="{{ url('/articles') }}" enctype="multipart/form-data" method="post">
6             <div class="form-group">
7                 <div class="mb-3 form-floating">
8                     <input type="text" placeholder="title" name="title" id="title" required="required" class="form-control">
9                     <label for="title" class="form-label">Title</label>
10                </div>
11                <div class="mb-3 form-floating">
12                    <textarea placeholder="content" name="content" id="content" required="required" class="form-control"></textarea>
13                    <label for="content" class="form-label">Content</label>
14                </div>
15                <div class="mb-3 form-floating">
16                    <input type="file" name="image" id="image" required="required" class="form-control">
17                    <label for="image" class="form-label">Feature Image</label>
18                </div>
19                <div class="d-flex justify-content-center">
20                    <button type="submit" class="btn btn-primary" name="submit">Simpan</button>
21                </div>
22            </div>
23        </form>
24    </div>
25 @endsection
```

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

```
33 public function store(Request $request)
34 {
35     if ($request->file('image')) {
36         $image_name = $request->file('image')->store('image', 'public');
37     }
38     Article::create([
39         'title' => $request->title,
40         'content' => $request->content,
41         'featured_image' => $image_name
42     ]);
43     return 'Artikel berhasil disimpan';
44 }
```

9. Membuat direktori virtual di `public/storage` yang isinya sama persis (mirroring) dengan direktori `storage/app/public`. Gunakan perintah berikut untuk mirroring file tersebut.

```
W I N D O W S X@DESKTOP-6E8EN92 MINGW64 /c/xampp/htdocs/Test/form_file
$ php artisan storage:link
```

10. 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:

```
13 public function index()
14 {
15     $activeMenu = 'article';
16
17     return view('articles.create', ['activeMenu' => $activeMenu]);
18 }
```

11. Hasil

Form File

Home Article Dropdown

Title

Content

Feature Image
Choose File No file chosen

Simpan

Praktikum 2 : Membuat Form *Update* Berkas

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

```

1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container mt-5 p-3 bg-light shadow">
5          <form action="{{ url('/articles/{{ $article->id }}') }}" enctype="multipart/form-data" method="post">
6              @csrf
7              <div class="form-group">
8                  <div class="mb-3 form-floating">
9                      <input type="text" placeholder="title" value="{{ $article->title }}" name="title" id="title"
10                         required="required" class="form-control">
11                      <label for="title" class="form-label">Title</label>
12                  </div>
13              </div>
14              <div class="form-group">
15                  <div class="mb-3 form-floating">
16                      <textarea placeholder="content" value="{{ $article->content }}" name="content" id="content" required="required"
17                         class="form-control"></textarea>
18                      <label for="content" class="form-label">Content</label>
19                  </div>
20              </div>
21              <div class="form-group">
22                  <div class="mb-3 form-floating">
23                      <input type="file" value="{{ $article->featured_image }}" name="image" id="image" required="required"
24                         class="form-control">
25                      <label for="image" class="form-label">Feature Image</label>
26                      
27                  </div>
28              </div>
29              <div class="d-flex justify-content-center">
30                  <button type="submit" class="btn btn-primary" name="submit">Ubah Data</button>
31              </div>
32          </form>
33      </div>
34  @endsection
  
```

2. Karena tipe input berubah menjadi file, maka proses UPDATE data pada `ArticleController` harus dimodifikasi menjadi seperti berikut

```

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('image', 'public');
75     $article->featured_image = $image_name;
76
77     $article->save();
78     return 'Artikel berhasil diubah';
79 }

```

- Fungsi find(\$id) akan mengembalikan objek Article yang sesuai dengan ID tersebut.
 - Pada \$article->title = \$request->title; & \$article->content = \$request->content; memperbarui dengan data baru yang diterima dari permintaan (\$request).
 - Pada if, memeriksa apakah artikel tersebut memiliki gambar yang sudah ada, dan apakah file gambar tersebut masih ada di penyimpanan. Jika file gambar ada, maka gambar tersebut akan dihapus dari direktori penyimpanan publik menggunakan Storage::delete.
 - Fungsi store('image', 'public') menyimpan file ke dalam disk penyimpanan publik di subdirektori image, dan nama file yang dihasilkan akan disimpan dalam variabel \$image_name. Kemudian diupdate di dalam properti featured_image dari artikel tersebut.
 - \$article disimpan kembali ke dalam database menggunakan fungsi save().
3. Jangan lupa bagian fungsi edit untuk menampilkan form edit pada ArticleController perlu diubah menjadi seperti berikut.

```

56 public function edit(string $id)
57 {
58     $article = Article::find($id);
59
60     $activeMenu = 'article_edit';
61
62     return view('articles.edit', ['article' => $article, 'activeMenu' => $activeMenu]);
63 }

```


4. 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 ?



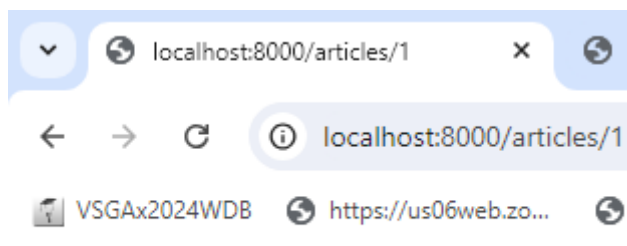
Title
Random

Contents
Gambar Acak

Feature Image
Choose File | No file chosen




Ubah Data



Title
Random Image

Contents
Gambar Acak

Feature Image
Choose File | No file chosen



Ubah Data

Praktikum 3 : Membuat Fitur Export PDF

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

```
W I N D O W S  X@DESKTOP-6E8EN92 MINGW64 /c/xampp/htdocs/Test/form_file (master)  
$ composer require barryvdh/laravel-dompdf
```

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

```

158     'providers' => ServiceProvider::defaultProviders()->merge([
159         /*
160          * Package Service Providers...
161          */
162
163         /*
164          * Application Service Providers...
165          */
166         App\Providers\AppServiceProvider::class,
167         App\Providers\AuthServiceProvider::class,
168         // App\Providers\BroadcastServiceProvider::class,
169         App\Providers\EventServiceProvider::class,
170         App\Providers\RouteServiceProvider::class,
171         Barryvdh\DomPDF\ServiceProvider::class
172     ])->toArray(),
173
174     /*
175     |-----
176     | Class Aliases
177     |-----
178     |
179     | This array of class aliases will be registered when this a
180     | is started. However, feel free to register as many as you
181     | the aliases are "lazy" loaded so they don't hinder perform
182     |
183     */
184
185     'aliases' => Facade::defaultAliases()->merge([
186         // 'Example' => App\Facades\Example::class,
187         'PDF' => Barryvdh\DomPDF\Facade\Pdf::class
188     ])->toArray(),

```

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

```

22 | Route::get('/cetak_pdf', [ArticleController::class, 'cetak_pdf']);

```

4. Tambahkan fungsi cetak_pdf pada ArticleController sebagai berikut.

```

91 | public function cetak_pdf()
92 | {
93 |     $articles = Article::all();
94 |     // $pdf = Pdf::loadView('articles.articles_pdf', ['articles' => $articles]);
95 |     $pdf = Pdf::loadView('articles.articles_pdf', ['articles' => $articles]);
96 |     $pdf->setPaper('A4', 'Portrait');
97 |     return $pdf->stream('articles.pdf');
98 | }

```

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

```

6 | use Barryvdh\DomPDF\Facade\Pdf;

```

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

```

resources > views > articles > articles_pdf.blade.php > html > body > style > table tr td
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Membuat Larporan PDF Dengan DOMPDF Laravel</title>
8
9     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
10    integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH" crossorigin="anonymous">
11 </head>
12 <body>
13     <style type="text/css">
14         table tr td,
15         table tr th {
16             font-size: 9pt
17         }
18     </style>
19     <center>
20         <h4>Laporan Artikel</h4>
21     </center>
22     <table class="table table-bordered table-striped">
23         <thead>
24             <tr>
25                 <th>No</th>
26                 <th>Judul</th>
27                 <th>Isi</th>
28                 <th>Gambar</th>
29             </tr>
30         </thead>
31         <tbody>
32             @foreach ($articles as $a)
33                 <tr>
34                     <td>{{ $loop->iteration }}</td>
35                     <td>{{ $a->title }}</td>
36                     <td>{{ $a->contents }}</td>
37                     <td>{{ $a->featured_image }}</td>
38                 </tr>
39             @endforeach
40         </tbody>
41     </table>
42
43     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
44    integrity="sha384-YvpcrYf0tY3lHB60NNkmKc5s9fFDVZLEsAA55NDz0xhy9GkccIdslK1eN7N6j1eHz" crossorigin="anonymous"></script>
45 </body>
46 </html>

```

7. Silakan coba akses http://127.0.0.1:8000/cetak_pdf untuk melihat hasilnya.

