



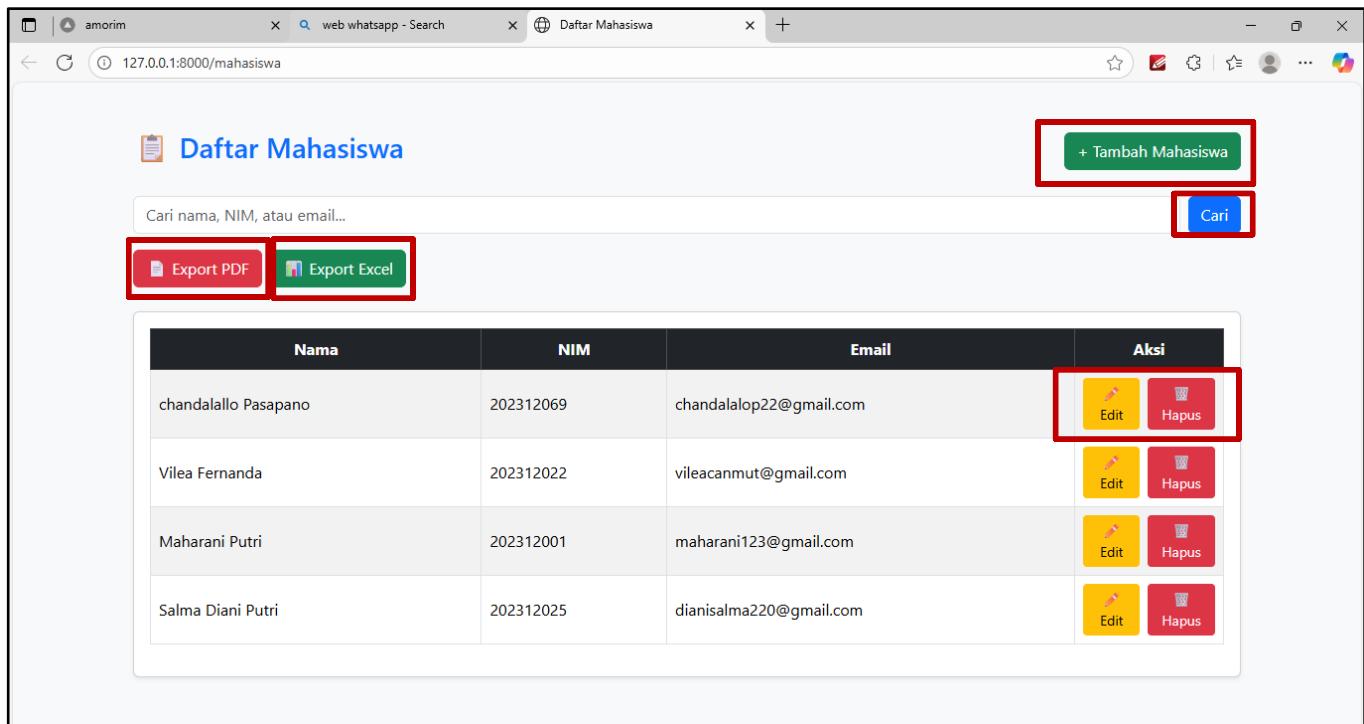
UTS GANJIL TAHUN AKADEMIK 2025-2026					
SEKOLAH TINGGI TEKNOLOGI BONTANG					
MATA KULIAH	:	Pemrograman Berbasis Framework	PRODI	:	Teknik Informatika
DOSEN	:	Fiky Anggara, S.Kom., M.Kom	KELAS/SEM	:	Pagi TI/ 5
HARI/TANGGAL	:	Senin, 03-11-2025	RUANG	:	3B
Deadline	:	Senin, 03-11-2025 pukul 7.00 WITA	SIFAT TUGAS	:	INDIVIDU

Nama : Salma Diani Putri

Nim : 202312025

Matkul : Pemrograman Berbasis Framework

Proyek yang dibuat pada Ujian Tengah Semester (UTS) mata kuliah Pemrograman Berbasis Framework ini adalah aplikasi Daftar Mahasiswa, yang bertujuan untuk menampilkan, menambah, mengubah, dan menghapus data mahasiswa dengan menggunakan konsep dasar CRUD (Create, Read, Update, Delete).

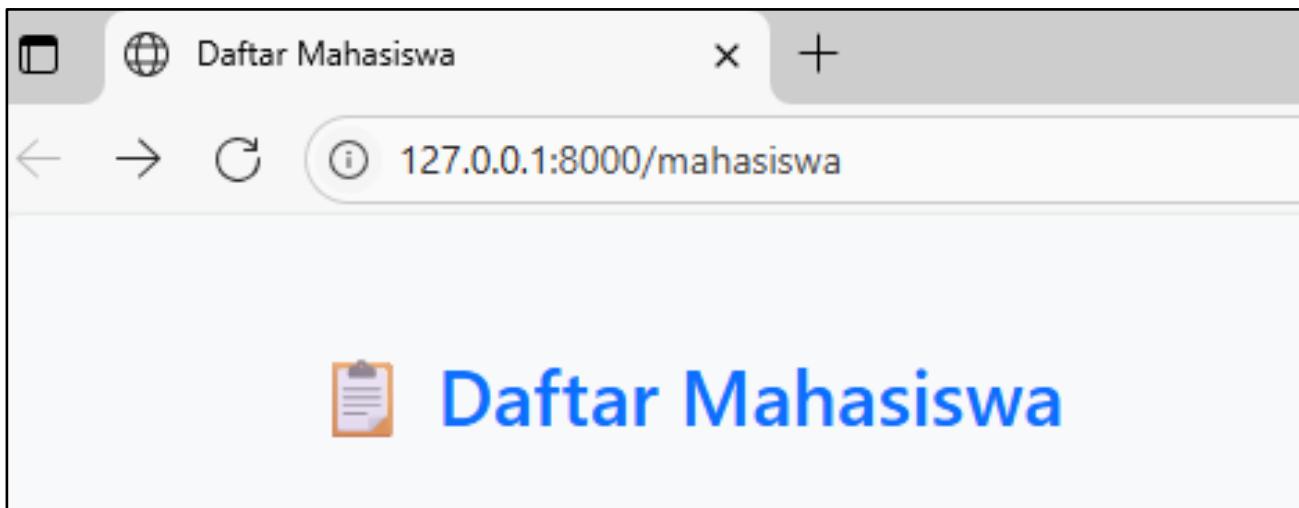


Nama	NIM	Email	Aksi
chandalallo Pasapano	202312069	chandalalop22@gmail.com	<button>Edit</button> <button>Hapus</button>
Vilea Fernanda	202312022	vileacanmut@gmail.com	<button>Edit</button> <button>Hapus</button>
Maharani Putri	202312001	maharani123@gmail.com	<button>Edit</button> <button>Hapus</button>
Salma Diani Putri	202312025	dianisalma220@gmail.com	<button>Edit</button> <button>Hapus</button>

Gambar di atas menunjukkan tampilan utama dari halaman Daftar Mahasiswa pada aplikasi yang dikembangkan menggunakan framework Laravel. Halaman ini berfungsi sebagai pusat pengelolaan data mahasiswa yang memuat fitur CRUD (Create, Read, Update, Delete) secara lengkap dan efisien.

1. Judul Halaman

Bagian atas halaman menampilkan judul “Daftar Mahasiswa” yang dilengkapi dengan ikon clipboard.



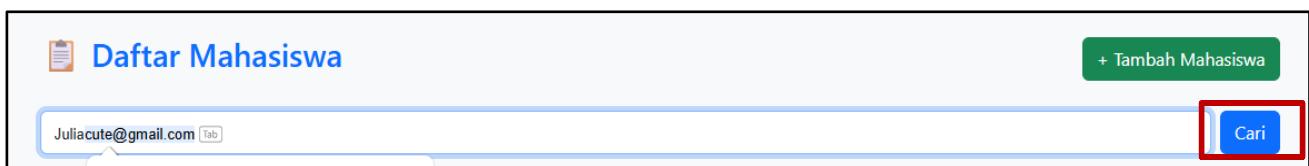
2. Tombol Tambah Mahasiswa

Tombol berwarna hijau bertuliskan “+ Tambah Mahasiswa” berada di sisi kanan atas halaman. Tombol ini digunakan untuk menambahkan data mahasiswa baru ke dalam database. Ketika tombol ini ditekan, pengguna akan diarahkan ke form input data mahasiswa baru yang berisi kolom seperti Nama, NIM, dan Email dan ada fitur simpan dan kembali dibagian bawah.

The image contains two screenshots of a web browser. The top screenshot shows the main "Daftar Mahasiswa" page with a green button labeled "+ Tambah Mahasiswa" highlighted with a red box. The bottom screenshot shows a modal dialog titled "+ Tambah Mahasiswa" containing input fields for "Nama" (Julia), "NIM" (202312003), and "Email" (juliacute@gmail.com). Below the fields are two buttons: "Kembali" and "Simpan", both highlighted with red boxes.

3. Kolom Pencarian

Di bawah judul halaman terdapat kolom pencarian dengan placeholder “Cari nama, NIM, atau email...”. Fitur ini mempermudah pengguna untuk mencari data mahasiswa tertentu berdasarkan kata kunci, sehingga tidak perlu menelusuri seluruh daftar secara manual. Dibawah ini saya mencari nama mahasiswa yaitu julia.



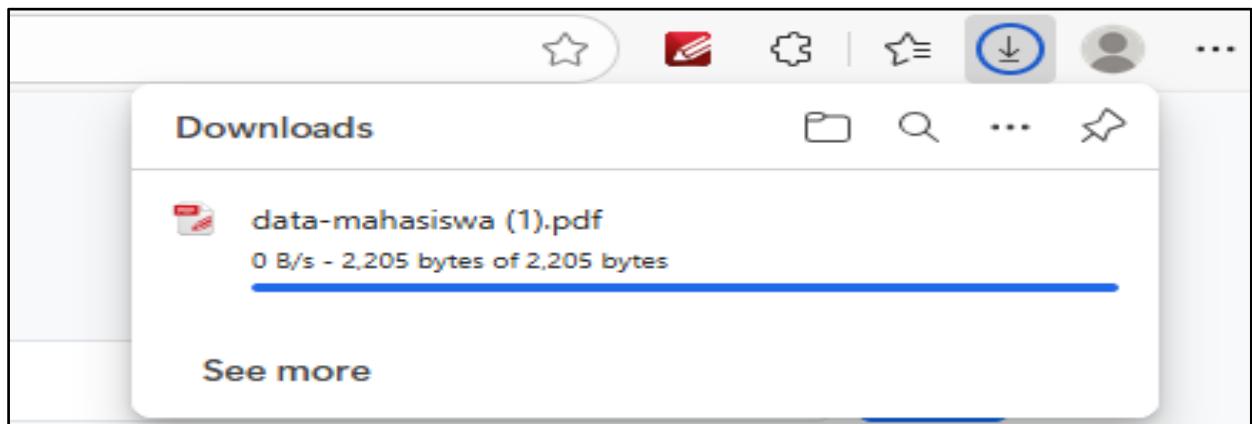
Maka akan muncul data mahasiswa yang kita cari yaitu julia seperti yang bisa dilihat dibawah ini.

Nama	NIM	Email	Aksi
julia	202312003	juliacute@gmail.com	<button>Edit</button> <button>Hapus</button>

4. Tombol Ekspor Data

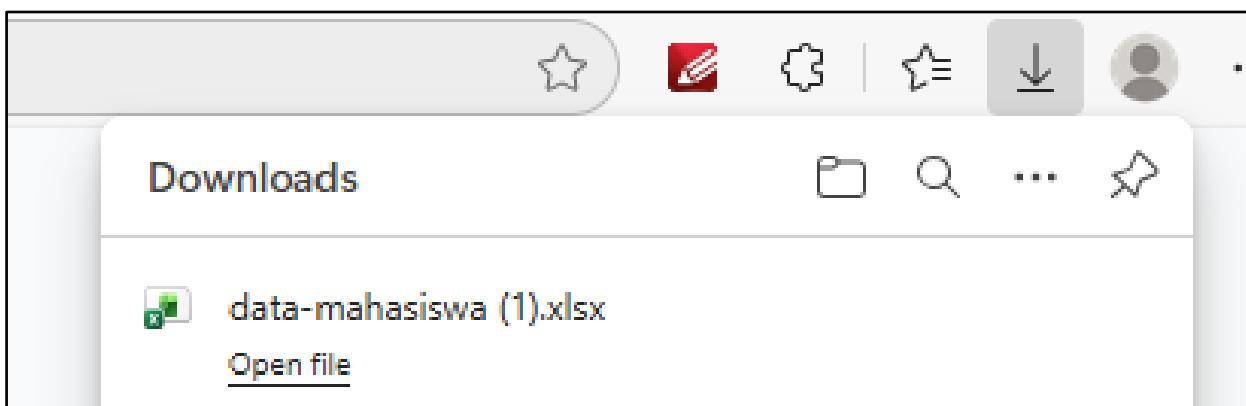
Terdapat dua tombol ekspor data, yaitu:

- Export PDF (merah) yang digunakan untuk mengunduh seluruh data mahasiswa dalam format dokumen PDF.seperti yang terlihat gambar dibawah dimana gambar tersebut telah berhasil mengunduh file pdf



Dan hasilnya berupa seperti ini

- Export Excel (hijau) — digunakan untuk mengunduh data dalam format file Excel (.xlsx) untuk kebutuhan analisis atau pelaporan. Fitur ekspor ini memberikan kemudahan dalam pembuatan laporan administrasi mahasiswa. Dibawah ini merupakan gambar proses pengunduhan data mahasiswa tersebut.



Dan bisa dilihat hasilnya seperti dibawah ini

	A1	B	C	D	E	F	G	H	I
1	ID	Nama	NIM	Email	Tanggal Dibuat				
2	1	Salma Dia	2,02E+08	dianisalm	2025-10-27T01:03:59.000000Z				
3	2	Maharani	2,02E+08	maharani	2025-10-30T02:36:28.000000Z				
4	3	Vilea Fern	2,02E+08	vileacanm	2025-10-30T02:37:01.000000Z				
5	4	chandalallo	2,02E+08	chandalal	2025-10-30T02:37:51.000000Z				
6	9	Julia	2,02E+08	juliacute@	2025-11-02T16:47:10.000000Z				
7									

5. Tabel Data Mahasiswa

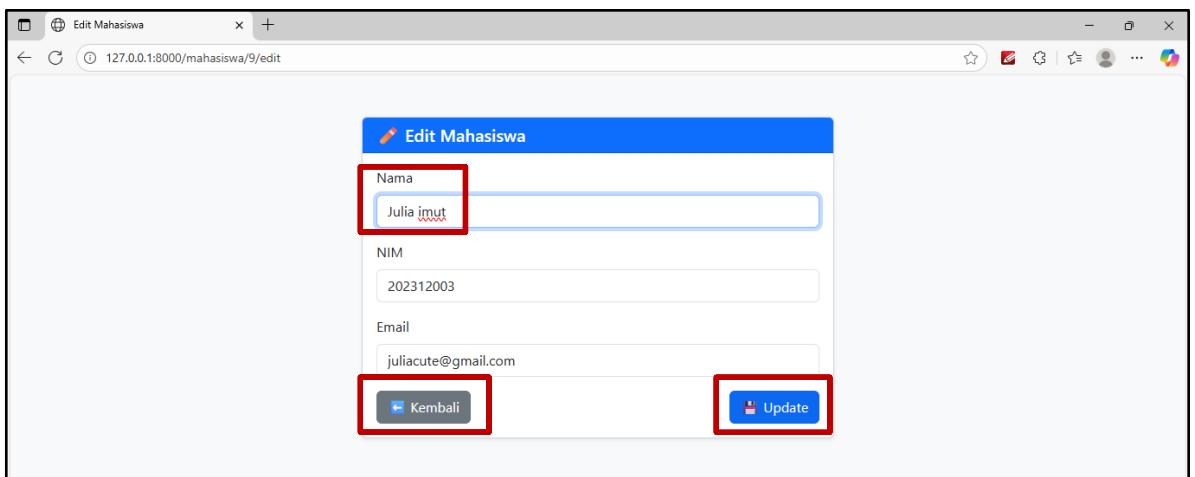
Bagian utama halaman berupa tabel yang menampilkan daftar mahasiswa. Tabel ini memiliki empat kolom, yaitu:

Daftar Mahasiswa			
<input placeholder="Cari nama, NIM, atau email..." type="text"/> <input type="button" value="Cari"/>			
<input type="button" value="Export PDF"/> <input type="button" value="Export Excel"/>			
Nama	NIM	Email	Aksi
chandalallo Pasapano	202312069	chandalalop22@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
Vilea Fernanda	202312022	vileacanmut@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
Maharani Putri	202312001	maharani123@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
Salma Diani Putri	202312025	dianisalma220@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

- Nama: Menampilkan nama lengkap mahasiswa.
- NIM: Nomor Induk Mahasiswa yang menjadi identitas unik setiap mahasiswa.
- Email: Alamat email mahasiswa yang digunakan untuk keperluan komunikasi atau notifikasi sistem.
- Aksi: Kolom terakhir yang berisi dua tombol, yaitu:

Nama	NIM	Email	Aksi
julia	202312003	juliacute@gmail.com	 Edit  Hapus

- Edit (kuning) — digunakan untuk mengubah data mahasiswa yang sudah tersimpan. Seperti yang terlihat pada gambar dibawah saya mencoba untuk mengedit mahasiswa, dan dibawahnya ada fitur update dan kembali.

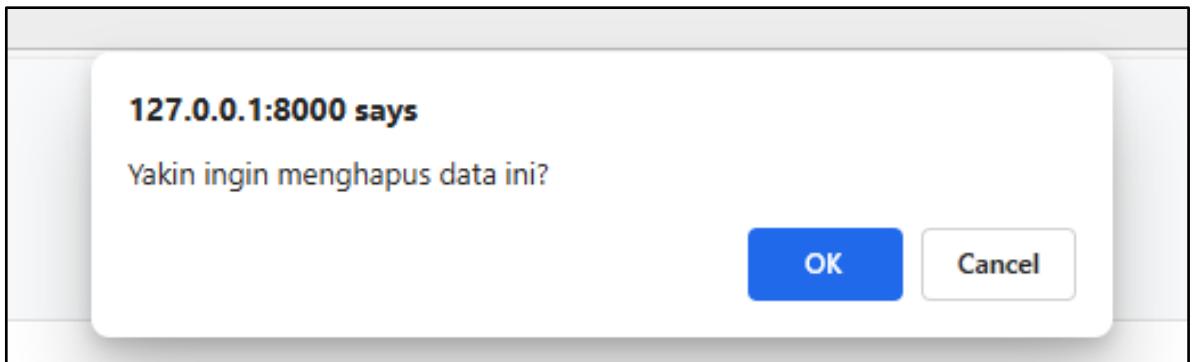


The screenshot shows a browser window titled "Edit Mahasiswa". The URL is "127.0.0.1:8000/mahasiswa/9/edit". The form has fields for Nama (Julia imut), NIM (202312003), and Email (juliacute@gmail.com). At the bottom are two buttons: "Kembali" (highlighted with a red box) and "Update".

Dan jika berhasil diubah maka nama mahasiswanya akan terlihat seperti diawah.

Nama	NIM	Email	Aksi
Julia imut	202312003	juliacute@gmail.com	 Edit  Hapus

- Hapus (merah) — digunakan untuk menghapus data mahasiswa dari sistem. Misalnya saya ingin menghapus data mahasiswa bernama julia dan saya klik hapus, maka akan muncul seperti ini, lalu klik oke.



Dan bisa dilihat data mahasiswa bernama julia sudah tidak ada.

Nama	NIM	Email	Aksi
chandalallo Pasapano	202312069	chandalalop22@gmail.com	Edit Hapus
Vilea Fernanda	202312022	vileacanmut@gmail.com	Edit Hapus
Maharani Putri	202312001	maharani123@gmail.com	Edit Hapus
Salma Diani Putri	202312025	dianisalma220@gmail.com	Edit Hapus

6. Fitur Pagination (Halaman Data)

Di bagian bawah tabel **Daftar Mahasiswa**, terdapat fitur **pagination** yang berfungsi untuk membagi data menjadi beberapa halaman. Fitur ini secara otomatis muncul ketika jumlah data mahasiswa melebihi batas tampilan per halaman yang telah ditentukan di sistem (misalnya 10 data per halaman).

Nama	NIM	Email	Aksi
Rakha	202312004	rakha1200@gmail.com	Edit Hapus
arumi	202312033	arumi12@gmail.com	Edit Hapus
sifa	202312043	sifaaa@gmail.com	Edit Hapus
chandalallo Pasapano	202312069	chandalalop22@gmail.com	Edit Hapus
Vilea Fernanda	202312022	vileacanmut@gmail.com	Edit Hapus

Showing 1 to 5 of 7 results 1 2 >

Nama	NIM	Email	Aksi
Maharani Putri	202312001	maharani123@gmail.com	Edit Hapus
Salma Diani Putri	202312025	dianisalma220@gmail.com	Edit Hapus

Showing 6 to 7 of 7 results < 1 2 > /

Penjelasan Source Code

1. File Resources/web.php

File ini berfungsi untuk mendefinisikan rute (route) yang menghubungkan URL dengan fungsi yang akan dijalankan. Rute pada program ini dibagi menjadi dua kelompok utama yaitu Route CRUD Mahasiswa dan Route Export File.

- index → menampilkan daftar mahasiswa.
- create → menampilkan form tambah data.
- store → menyimpan data baru.
- edit dan update → mengubah data mahasiswa yang sudah ada.
- destroy → menghapus data mahasiswa dari database.
- Route Export File → memanggil fungsi ekspor ke PDF dan Excel pada controller.

```
1. <?php
2.
3. use Illuminate\Support\Facades\Route;
4. use App\Http\Controllers\MahasiswaController;
5.
6. // Halaman utama
7. Route::get('/', function () {
8.     return redirect()->route('mahasiswa.index');
9. });
10.
11. // =====
12. // 📁 EXPORT & 📁 IMPORT FILE
13. // =====
14. Route::get('/mahasiswa/pdf', [MahasiswaController::class, 'exportPdf'])->name('mahasiswa.exportPdf');
15. Route::get('/mahasiswa/excel', [MahasiswaController::class, 'exportExcel'])-
>name('mahasiswa.exportExcel');
16.
17. // =====
18. // 🔳 CRUD MAHASISWA
19. // =====
20. Route::get('/mahasiswa', [MahasiswaController::class, 'index'])->name('mahasiswa.index');
21. Route::get('/mahasiswa/create', [MahasiswaController::class, 'create'])->name('mahasiswa.create');
22. Route::post('/mahasiswa', [MahasiswaController::class, 'store'])->name('mahasiswa.store');
23. Route::get('/mahasiswa/{mahasiswa}/edit', [MahasiswaController::class, 'edit'])-
>name('mahasiswa.edit');
24. Route::put('/mahasiswa/{mahasiswa}', [MahasiswaController::class, 'update'])-
>name('mahasiswa.update');
25. Route::delete('/mahasiswa/{mahasiswa}', [MahasiswaController::class, 'destroy'])-
>name('mahasiswa.destroy');
```

2. File MahasiswaController.php

```
1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\Mahasiswa;
6. use Illuminate\Http\Request;
7. use Barryvdh\DomPDF\Facade\Pdf;
8. use Maatwebsite\Excel\Facades\Excel;
9. use App\Imports\MahasiswaImport;
10. use App\Exports\MahasiswaExport;
11.
12. class MahasiswaController extends Controller
13. {
14.     /**
15.      * Tampilkan daftar mahasiswa dengan pencarian dan pagination.
16.      */
17.     public function index(Request $request)
18.     {
19.         $search = $request->input('search');
20.
21.         $mahasiswas = Mahasiswa::when($search, function ($query, $search) {
22.             return $query->where('nama', 'like', "%{$search}%")
23.                     ->orWhere('nim', 'like', "%{$search}%")
24.                     ->orWhere('email', 'like', "%{$search}%");
25.         })
26.             ->orderBy('id', 'desc')
27.             ->paginate(5);
28.
29.         return view('mahasiswa.index', compact('mahasiswas', 'search'));
30.     }
31. }
```

```

30.     }
31.
32.     /**
33.      * Tampilkan form tambah data mahasiswa.
34.     */
35.    public function create()
36.    {
37.        return view('mahasiswa.create');
38.    }
39.
40.    /**
41.      * Simpan data mahasiswa baru.
42.     */
43.    public function store(Request $request)
44.    {
45.        $request->validate([
46.            'nama' => 'required|string|max:255',
47.            'nim' => 'required|string|max:50|unique:mahasiswas,nim',
48.            'email' => 'required|email|unique:mahasiswas,email',
49.        ]);
50.
51.        Mahasiswa::create([
52.            'nama' => $request->nama,
53.            'nim' => $request->nim,
54.            'email' => $request->email,
55.        ]);
56.
57.        return redirect()->route('mahasiswa.index')->with('success', 'Data berhasil ditambahkan!');
58.    }
59.
60.    /**
61.      * Tampilkan form edit mahasiswa.
62.     */
63.    public function edit(Mahasiswa $mahasiswa)
64.    {
65.        return view('mahasiswa.edit', compact('mahasiswa'));
66.    }
67.
68.    /**
69.      * Update data mahasiswa.
70.     */
71.    public function update(Request $request, Mahasiswa $mahasiswa)
72.    {
73.        $request->validate([
74.            'nama' => 'required|string|max:255',
75.            'nim' => 'required|string|max:50|unique:mahasiswas,nim,' . $mahasiswa->id,
76.            'email' => 'required|email|unique:mahasiswas,email,' . $mahasiswa->id,
77.        ]);
78.
79.        $mahasiswa->update([
80.            'nama' => $request->nama,
81.            'nim' => $request->nim,
82.            'email' => $request->email,
83.        ]);
84.
85.        return redirect()->route('mahasiswa.index')->with('success', 'Data berhasil diperbarui!');
86.    }
87.
88.    /**
89.      * Hapus data mahasiswa.
90.     */
91.    public function destroy(Mahasiswa $mahasiswa)
92.    {
93.        $mahasiswa->delete();
94.
95.        return redirect()->route('mahasiswa.index')->with('success', 'Data berhasil dihapus!');
96.    }
97.
98.    /**
99.      * Export data mahasiswa ke PDF.
100.     */
101.    public function exportPdf()
102.    {
103.        $mahasiswas = Mahasiswa::all();
104.
105.        $pdf = Pdf::loadView('mahasiswa.pdf', compact('mahasiswas'))
106.            ->setPaper('a4', 'portrait');
107.
108.        return $pdf->download('data-mahasiswa.pdf');
109.    }
110.
111.    /**
112.      * Export data mahasiswa ke Excel.
113.     */
114.    public function exportExcel()
115.    {
116.        return Excel::download(new MahasiswaExport, 'data-mahasiswa.xlsx');
117.    }

```

Penjelasannya :

File controller ini berisi seluruh logika aplikasi, mulai dari pengelolaan data hingga ekspor file. Berikut penjelasan setiap metodenya:

a. Fungsi `index()`

Fungsi ini menampilkan daftar mahasiswa dengan fitur:

- Pencarian (search) berdasarkan nama, NIM, atau email.
- Pagination (paginate(5)) menampilkan lima data per halaman.
- Mengurutkan data berdasarkan ID terbaru.

b. Fungsi `create()` dan `store()`

Digunakan untuk menambahkan data baru mahasiswa.

Program akan melakukan validasi input agar NIM dan email tidak duplikat sebelum data disimpan ke database.

c. Fungsi `edit()` dan `update()`

Menyediakan form edit dan menyimpan hasil perubahan data. Validasi memastikan tidak ada duplikasi data dengan mahasiswa lain.

d. Fungsi `destroy()`

Menghapus data mahasiswa berdasarkan ID yang dipilih. Setelah penghapusan, sistem akan menampilkan notifikasi sukses.

e. Fungsi `exportPdf()`

Mengambil seluruh data mahasiswa dan menampilkannya ke dalam template view `mahasiswa.pdf`, lalu dikonversi menjadi file PDF menggunakan library Barryvdh DomPDF. File hasil ekspor otomatis diunduh dengan nama `data-mahasiswa.pdf`

f. Fungsi `exportExcel()`

Menjalankan proses ekspor data mahasiswa ke dalam file Excel dengan bantuan class `MahasiswaExport`. File hasil ekspor memiliki format `.xlsx` dan diunduh langsung ke perangkat pengguna.

3. File `MahasiswaExport.php`

```

1. <?php
2.
3. namespace App\Exports;
4.
5. use App\Models\Mahasiswa;
6. use Maatwebsite\Excel\Concerns\FromCollection;
7. use Maatwebsite\Excel\Concerns\WithHeadings;
8.
9. class MahasiswaExport implements FromCollection, WithHeadings
10. {
11.     public function collection()
12.     {
13.         return Mahasiswa::select('id', 'nama', 'nim', 'email', 'created_at')->get();
14.     }
15.
16.     public function headings(): array
17.     {
18.         return ['ID', 'Nama', 'NIM', 'Email', 'Tanggal Dibuat'];
19.     }
20. }
```

File ini mengatur data apa saja yang akan diekspor ke dalam file Excel. Program menggunakan package

Maatwebsite Excel dan mengimplementasikan dua interface penting:

a. FromCollection

Digunakan untuk mengambil data dari model Mahasiswa:

```
1. public function collection()
2. {
3.     return Mahasiswa::select('id', 'nama', 'nim', 'email', 'created_at')->get();
4. }
```

b. WithHeadings

Menambahkan header (judul kolom) pada file Excel:

```
1. public function headings(): array
2. {
3.     return ['ID', 'Nama', 'NIM', 'Email', 'Tanggal Dibuat'];
4. }
```

Dengan demikian, hasil file Excel akan memiliki baris pertama berisi nama kolom dan diikuti oleh data mahasiswa dari database.