

LAPORAN
PRAKTIKUM PEMROGRAMAN WEB

CRUD LARAVEL

**Untuk Memenuhi Salah Satu Tugas
Mata Kuliah Praktikum Pemrograman Web
Dosen Pengampu: Lindung Siswanto, S.Kom., M.Eng.**



Disusun Oleh:
Muhamad Ardalepa (NIM: 3202016004)

**PROGRAM STUDI D3 TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI PONTIANAK
2022**

DAFTAR ISI

DAFTAR ISI.....	ii
PEMBAHASAN	1
A. Persiapan.....	1
1. Mengatur file DotEnv	1
2. Membuat model, controller dan migration.....	1
3. Mengatur routing.....	3
B. Pembuatan Method dan View.....	3
1. Menambahkan method index, create dan store pada controller	3
2. Membuat view registration form.....	5
3. Modifikasi method store().....	7
4. Modifikasi method index().....	10
5. Method show()	11
6. Modifikasi view index.....	13
7. Menambah flash message dan unique nim validation.....	13
8. Method edit() dan update().....	15
9. Method destroy().....	17
C. Tambahan	17
D. Tugas	19
1. Membuat fitur search	20
2. Membuat fitur filter.....	21
3. Membuat fitur sort.....	22
4. Membuat fitur pagination.....	23
5. Membuat fitur confirmation box	24
6. Fitur Login.	25

PEMBAHASAN

A. Persiapan

1. Mengatur file DotEnv

```
11  DB_CONNECTION=mysql
12  DB_HOST=127.0.0.1
13  DB_PORT=3307
14  DB_DATABASE=laracrud
15  DB_USERNAME=root
16  DB_PASSWORD=
17
```

Saya menggunakan port 3307 untuk mysql dari Laragon dan nama databasenya yaitu laracrud.

2. Membuat model, controller dan migration

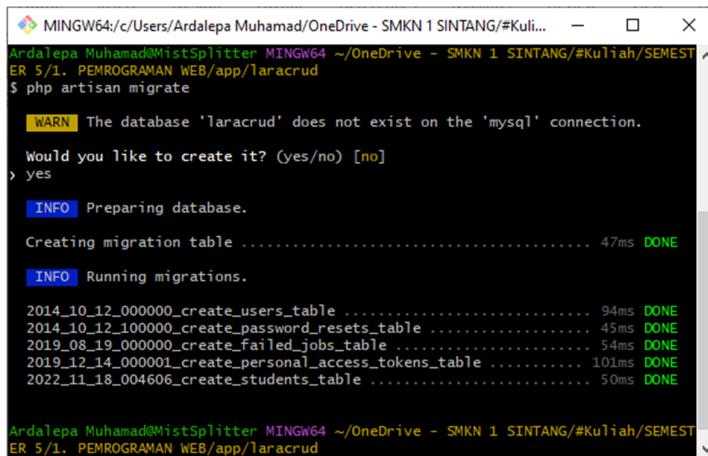
Untuk membuat model , controller dan migration dengan satu kali perintah, dapat menggunakan command `php artisan make:model -mc Student`

```
MINGW64:/c/Users/Ardalepa Muhamad/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5\1. PEMROGRAMAN WEB/app/laracrud$ php artisan make:model Student -mc
INFO Model [C:\Users\Ardalepa Muhamad\OneDrive - SMKN 1 SINTANG#\Kuliah\SEMESTER 5\1. PEMROGRAMAN WEB\app\laracrud\app\Models\Student.php] created successfully.
INFO Created migration [2022_11_18_004606_create_students_table].
INFO Controller [C:\Users\Ardalepa Muhamad\OneDrive - SMKN 1 SINTANG#\Kuliah\SEMESTER 5\1. PEMROGRAMAN WEB\app\laracrud\app\Http\Controllers\StudentController.php] created successfully.
```

Pada method up file migration students dapat menambahkan baris kode berikut ini untuk membuat struktur table pada database.

```
14  public function up()
15  {
16      Schema::create('students', function (Blueprint $table)
17      {
18          $table->id();
19          $table->char('nim', 10)->unique();
20          $table->string('name');
21          $table->char('gender', 1);
22          $table->string('major');
23          $table->text('address')->nullable();
24          $table->timestamps();
25      });
}
```

Lalu jalankan command `php artisan migrate` untuk mengeksekusi perubahan pada file migration tadi.



```
MINGW64:/c/Users/Ardalepa Muhamad/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud
$ php artisan migrate

[WARN] The database 'laracrud' does not exist on the 'mysql' connection.

Would you like to create it? (yes/no) [no]
> yes

[INFO] Preparing database.

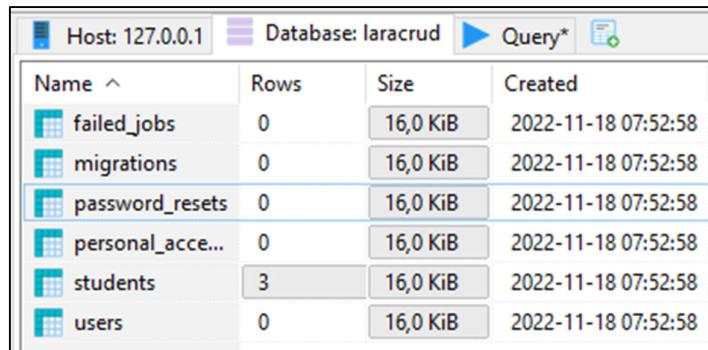
Creating migration table ..... 47ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 94ms DONE
2014_10_12_100000_create_password_resets_table ..... 45ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 54ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 101ms DONE
2022_11_18_004606_create_students_table ..... 50ms DONE

Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud
```

Sehingga pada database akan terbentuk database baru bernama laracrud dengan tabel students dan 5 table lainnya.



Name	Rows	Size	Created
failed_jobs	0	16,0 KiB	2022-11-18 07:52:58
migrations	0	16,0 KiB	2022-11-18 07:52:58
password_resets	0	16,0 KiB	2022-11-18 07:52:58
personal_acce...	0	16,0 KiB	2022-11-18 07:52:58
students	3	16,0 KiB	2022-11-18 07:52:58
users	0	16,0 KiB	2022-11-18 07:52:58

Dan pada table students, akan terbentuk struktur sesuai dengan yang kita buat pada file migration.



#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Collation
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT	utf8mb4_unicode_ci
2	nim	CHAR	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default	utf8mb4_unicode_ci
3	name	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default	utf8mb4_unicode_ci
4	gender	CHAR	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default	utf8mb4_unicode_ci
5	major	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default	utf8mb4_unicode_ci
6	address	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	utf8mb4_unicode_ci
7	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
8	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	

3. Mengatur routing

Pada file routes\web.php menambahkan beberapa route sebagai berikut:

```
routes > web.php
1  <?php
2
3  use App\Http\Controllers\StudentController;
4  use Illuminate\Support\Facades\Route;
5
6  Route::get('/students', [StudentController::class, 'index'])->name('students.index');
7  Route::get('/students/create', [StudentController::class, 'create'])->name('students.create');
8  Route::post('/students', [StudentController::class, 'store'])->name('students.store');
9  Route::get('/students/{student}', [StudentController::class, 'show'])->name('students.show');
10 Route::get('/students/{student}/edit', [StudentController::class, 'edit'])->name('students.edit');
11 Route::put('/students/{student}', [StudentController::class, 'update'])->name('students.update');
12 Route::delete('/students/{student}', [StudentController::class, 'destroy'])->name('students.destroy');
```

Selain method get yang sering kita pakai, terdapat method post, put dan delete. Method method tersebut merepresentasikan metode user dalam mengakses route tersebut.

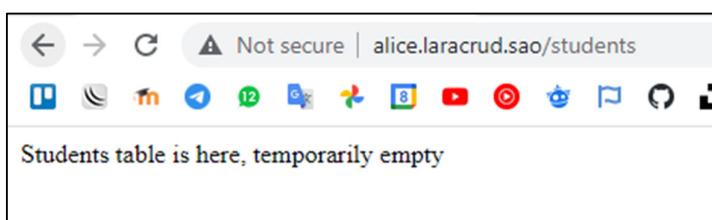
B. Pembuatan Method dan View

1. Menambahkan method index, create dan store pada controller

Method index digunakan untuk menampilkan halaman utama yang nantinya berupa table students. **Method create** digunakan untuk menampilkan view registration form. **Method store** digunakan untuk melakukan proses validasi data dan nantinya akan dimasukkan ke database.

```
10  public function index()
11  {
12      return "Students table is here, temporarily empty";
13  }
14  public function create()
15  {
16      return view('registration-form');
17  }
18  public function store(Request $request)
19  {
20      $validateData = $request->validate([
21          'nim' => 'required|size:10|unique:students',
22          'name' => 'required|min:3|max:50',
23          'gender' => 'required|in:M,F',
24          'major' => 'required',
25          'address' => '',
26      ]);
27      dump($validateData);
28  }
```

Apabila mengakses route `/students` maka akan tampil seperti berikut:



2. Membuat view registration form

```

resources > views > registration-form.blade.php > html > body > div.container.pt-4.bg-white.mb-5 > div.row > div.col-md-8.col-xl-6 > form >
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <link rel="stylesheet" href="/css/bootstrap.css">
9      <link rel="stylesheet" href="/css/artist.css">
10     <title>Student Registration</title>
11 </head>
12
13 <body>
14     <div class="container pt-4 bg-white mb-5">
15         <div class="row">
16             <div class="col-md-8 col-xl-6">
17                 <h1>Student Registration</h1>
18                 <hr>
19                 <form action="{{ route('students.store') }}" method="POST">
20                     @csrf
21                     <div class="mb-3">
22                         <label for="nim" class="form-label">NIM</label>
23                         <input type="text" name="nim" id="nim" value="{{ old('nim') }}"
24                             class="form-control @error('nim') is-invalid @enderror"
25                             @error('nim')
26                             <div class="text-danger">{{ $message }}Full Name</label>
32                         <input type="text" name="name" id="name" value="{{ old('name') }}"
33                             class="form-control @error('name') is-invalid @enderror"
34                             @error('name')
35                             <div class="text-danger">{{ $message }}Gender</label>
41                         <div class="d-flex">
42                             <div class="form-check me-3">

```



```

resources > views > registration-form.blade.php > html > body > div.container.pt-4.bg-white.mb-5 > div.row > div.col-md-8.col-xl-6 > form >
53                         @error('gender')
54                             <div class="text-danger">{{ $message }}Major</label>
60                         <select name="major" id="major" class="form-select">
61                             <option value="Informatics Engineering" {{ old('major')=='Informatics Engineering' ? 'selected':'' }}>
62                                 Informatics Engineering
63                             </option>
64                             <option value="Informatics System" {{ old('major')=='Informatics System' ? 'selected':'' }}>
65                                 Informatics System
66                             </option>
67                             <option value="Computer Science" {{ old('major')=='Computer Science' ? 'selected':'' }}>
68                                 Computer Science
69                             </option>
70                             <option value="Computer Engineering" {{ old('major')=='Computer Engineering' ? 'selected':'' }}>
71                                 Computer Engineering
72                             </option>
73                             <option value="Telecommunications Engineering" {{ old('major')=='Telecommunications Engineering' ? 'selected':'' }}>
74                                 Telecommunications Engineering
75                             </option>
76                         </select>
77                         @error('gender')
78                             <div class="text-danger">{{ $message }}Address</label>
84                         <textarea name="address" id="address" rows="3" name="address"
85                             class="form-control">{{ old('address') }}

```

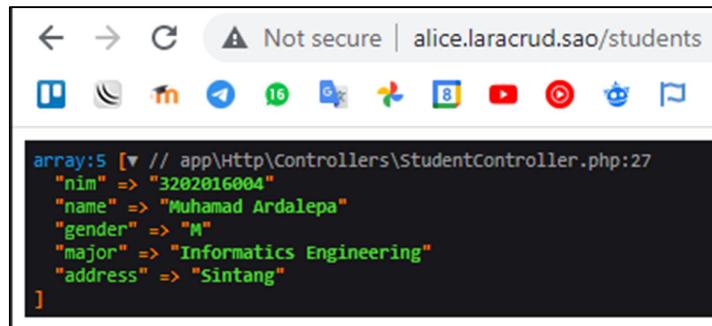
Sehingga apabila view registration dipanggil, akan terlihat seperti ini:

A screenshot of a web browser showing a 'Student Registration' form. The page title is 'Student Registration'. The form fields include: 'NIM' (input field), 'Full Name' (input field), 'Gender' (radio buttons for Male and Female), 'Major' (dropdown menu set to 'Informatics Engineering'), and 'Address' (text area). At the bottom are 'Submit' and 'Back' buttons.

Apabila data yang diisi tidak lolos validasi, maka akan muncul pesan error sebagai seperti ini:

A screenshot of a web browser showing a 'Student Registration' form with validation errors. The errors are: 'The nim must be 10 characters.' (next to the NIM input field), 'The name field is required.' (next to the Full Name input field), and 'The gender field is required.' (next to the Gender radio buttons). The rest of the form fields and layout are identical to the first screenshot.

Apabila data yang diisi benar, maka akan tampil seperti dibawah ini:



A screenshot of a web browser window showing a JSON dump in the developer tools' console. The URL is 'Not secure | alice.laracrud.sao/students'. The JSON data is as follows:

```
array:5 [▼ // app\Http\Controllers\StudentController.php:27
  "nim" => "3202016004"
  "name" => "Muhamad Ardalepa"
  "gender" => "M"
  "major" => "Informatics Engineering"
  "address" => "Sintang"
]
```

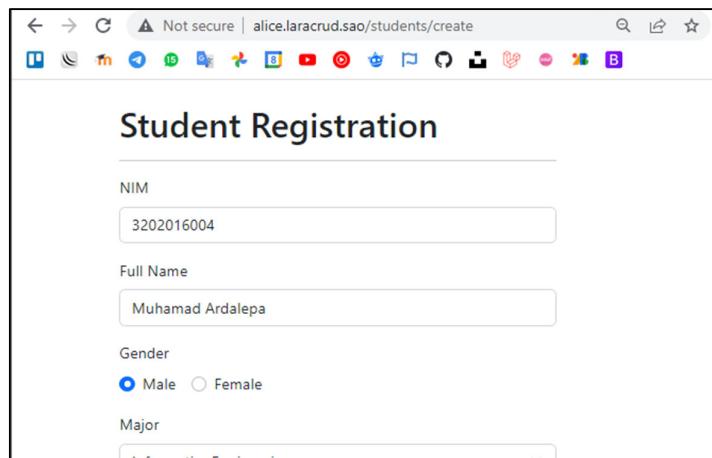
3. Modifikasi method store()

Agar dapat digunakan untuk insert data, maka ditambahkan kode seperti ini.



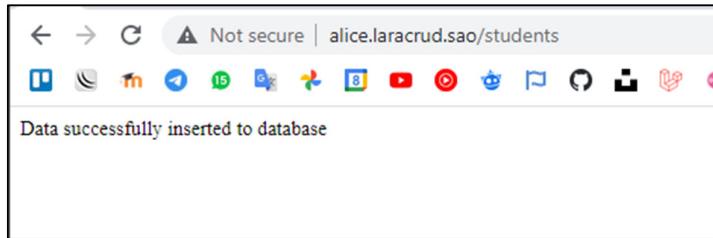
```
18     public function store(Request $request)
19     {
20         $validateData = $request->validate([
21             'nim' => 'required|size:10',
22             'name' => 'required|min:3|max:50',
23             'gender' => 'required|in:M,F',
24             'major' => 'required',
25             'address' => '',
26         ]);
27         // dump($validateData);
28         $student = new Student();
29         $student->nim = $validateData['nim'];
30         $student->name = $validateData['name'];
31         $student->gender = $validateData['gender'];
32         $student->major = $validateData['major'];
33         $student->address = $validateData['address'];
34         $student->save();
35         return "Data successfully inserted to database";
36     }
```

Apabila mengisi data yang benar pada registration form, maka akan tampak seperti ini:



A screenshot of a 'Student Registration' form. The fields are filled as follows:

- NIM: 3202016004
- Full Name: Muhamad Ardalepa
- Gender: Male (radio button selected)
- Major: Informatics Engineering (dropdown menu)



Pada database akan terdapat satu record baru

laracrud:students: 1 rows total (approximately)							
id	nim	name	gender	major	address	created_at	updated_at
6	3202016004	Muhamad Ardalepa	M	Informatics Engineering	Sintang	2022-11-20 05:55:52	2022-11-20 05:55:52

Untuk mempersingkat kode dapat menggunakan mass assignment

```

18     public function store(Request $request)
19
20
21     $validateData = $request->validate([
22         'nim' => 'required|size:10',
23         'name' => 'required|min:3|max:50',
24         'gender' => 'required|in:M,F',
25         'major' => 'required',
26         'address' => '',
27     ]);
28     // dump($validateData);
29     // $student = new Student();
30     // $student->nim = $validateData['nim'];
31     // $student->name = $validateData['name'];
32     // $student->gender = $validateData['gender'];
33     // $student->major = $validateData['major'];
34     // $student->address = $validateData['address'];
35     // $student->save();
36     Student::create($validateData);
37     return "Data successfully inserted to database";

```

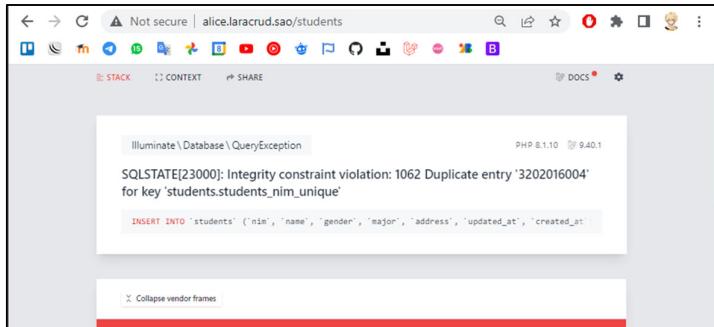
Apabila dilakukan pengiriman data akan terjadi error. Karena apabila menggunakan mass assignment, secara default semua kolom pada tabel akan ter-“guard” atau tidak boleh diisi. Solusinya adalah dengan mengosongkan array `$guarded` pada model.

```

8  class Student extends Model
9
10    use HasFactory;
11    protected $guarded = [];
12

```

Apabila dilakukan pengiriman ulang form yang terjadi akan error, karena untuk kolom nim harus unik, sedangkan yang dikirimkan nimnya sudah ada.



```
Illuminate\Database\QueryException  
SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry '3202016004' for key 'students.students_nim_unique'  
INSERT INTO `students` (`nim`, `name`, `gender`, `major`, `address`, `updated_at`, `created_at`)
```

Tetapi apabila nimnya berbeda, tidak akan ada error

Student Registration

NIM
3201216004

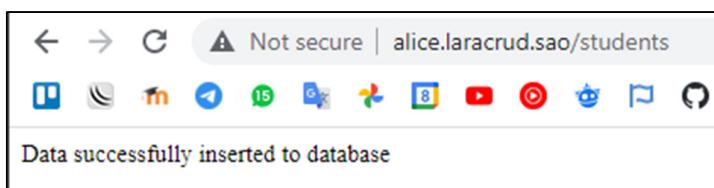
Full Name
Kamisato Ayaka

Gender
 Male Female

Major
Informatics Engineering

Address
Narukami Island

Submit **Back**



laracrud.students 2 rows total (approximately)						
id	nim	name	gender	major	address	created_at
6	3202016004	Muhamad Ardalepa	M	Informatics Engineering	Sintang	2022-11-20 05:55:52
8	3201216004	Kamisato Ayaka	F	Informatics Engineering	Narukami Island	2022-11-20 06:05:37

4. Modifikasi method index()

```

10     public function index()
11     {
12         // return "Students table is here, temporarily empty";
13         $student = Student::all();
14         return view('index', ['students' => $student]);
15     }

```

Apabila method index() dipanggil maka akan menampilkan view index yang berupa tabel.

```

resources > views > index.blade.php > body > div.container.pt-4.bg-white.mb-5 > div.row > div.col-12 >
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <link rel="stylesheet" href="/css/bootstrap.css">
9      <link rel="stylesheet" href="/css/artist.css">
10     <title>Student Data</title>
11 </head>
12
13 <body>
14     <div class="container pt-4 bg-white mb-5">
15         <div class="row">
16             <div class="col-12">
17                 <div class="d-flex justify-content-between align-items-center mb-3">
18                     <h1 class="mb-3">Students Table</h1>
19                 </div>
20                 <table class="table table-striped">
21                     <thead>
22                         <tr>
23                             <th>#</th>
24                             <th>NIM</th>
25                             <th>Name</th>
26                             <th>Gender</th>
27                             <th>Major</th>
28                             <th>Address</th>
29                         </tr>
30                     </thead>
31                     <tbody>
32                         @forelse ($students as $student)
33                         <tr>
34                             <td>{{ $loop->iteration }}</td>
35                             <td>{{ $student->nim }}</td>
36                             <td>{{ $student->name }}</td>
37                             <td>{{ $student->gender == 'M'? 'Male': 'Female' }}</td>
38                             <td>{{ $student->major }}</td>
39                             <td>{{ $student->address == '' ? 'N/A': $student->address }}</td>
40                         </tr>
41                         @empty
42                         <td colspan="6" class="text-center">Empty</td>
43                         @endforelse
44                     </tbody>
45                 </table>
46             </div>
47         </div>
48     </div>
49     <script src="/js/bootstrap.js"></script>
50     <script src="/js/engineer.js"></script>
51 </body>

```

Sehingga apabila akan tampil seperti ini:

#	NIM	Name	Gender	Major	Address
1	3202016004	Muhamad Ardilepa	Male	Informatics Engineering	Sintang
2	3201216004	Kamisato Ayaka	Female	Informatics Engineering	Narukami Island

5. Method show()

Method show digunakan untuk menampilkan detail dari satu mahasiswa

```
app > Http > Controllers > StudentController.php > StudentController
40 |     public function show($student)
41 |     {
42 |         $result = Student::find($student);
43 |         // $result = Student::findOrFail($student);
44 |         return view('show', ['student' => $result]);
45 |     }

```

Method ini akan memanggil view show.

```
resources > views > show.blade.php > html > body > div.container.pt-4.bg-white.mb-5
1   <!DOCTYPE html>
2   <html lang="en">
3
4   <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <meta http-equiv="X-UA-Compatible" content="ie=edge">
8       <link rel="stylesheet" href="/css/bootstrap.css">
9       <link rel="stylesheet" href="/css/artist.css">
10      <title>Student Data</title>
11
12 </head>
13
14 <body>
15     <div class="container pt-4 bg-white mb-5">
16         <div class="col-12">
17             <h1 class="mb-3">{{ $student->name }} Biodata</h1>
18             <ul class="list-group d-inline-block">
19                 <li class="list-group-item list-group-item-action d-flex align-items-center">
20                     <small class="me-3">NIM</small>
21                     <b class="fs-5">{{ $student->nim }}</b>
22                 </li>
23                 <li class="list-group-item list-group-item-action d-flex align-items-center">
24                     <small class="me-3">Name</small>
25                     <b class="fs-5">{{ $student->name }}</b>
26                 </li>
27                 <li class="list-group-item list-group-item-action d-flex align-items-center">
28                     <small class="me-3">Gender</small>
29                     <b class="fs-5">{{ $student->gender == 'M' ? 'Male' : 'Female' }}</b>
30                 </li>
31                 <li class="list-group-item list-group-item-action d-flex align-items-center">
32                     <small class="me-3">Major</small>
33                     <b class="fs-5">{{ $student->major }}</b>
34                 </li>
35                 <li class="list-group-item list-group-item-action d-flex align-items-center">
36                     <small class="me-3">Address</small>
37                     <b class="fs-5">{{ $student->address == '' ? 'N/' : $student->address }}</b>

```

```

37     <b class="fs-5">{{ $student->address == '' ? 'N/'  

38     A:$student->address }}</b>  

39     </li>  

40     </ul>  

41     <br>  

42     <a href="{{ url()->previous() }}" class="btn btn-success mt-3">Back</a>  

43     </div>  

44   </div>  

45 </div>  

46 <script src="/js/bootstrap.js"></script>  

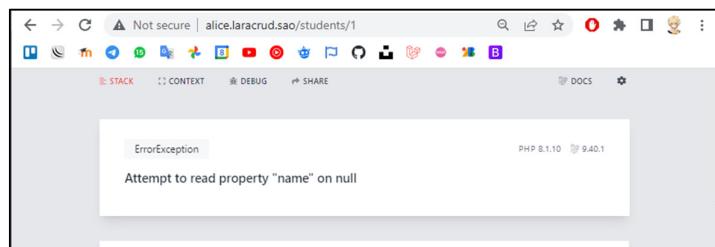
47 <script src="/js/engineer.js"></script>  

48 </body>  

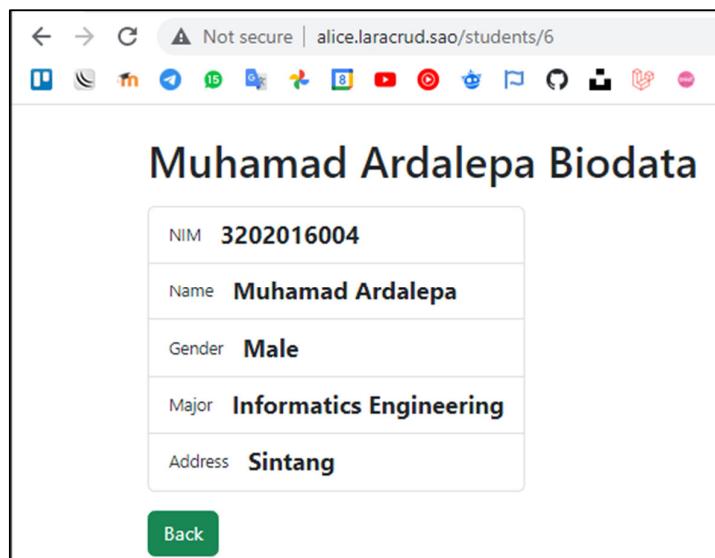
49 </html>

```

Sehingga apabila mengakses `/students/1` akan muncul error, karena tidak ada data yang memiliki id 1.



Tetapi apabila diganti menjadi `/students/6` akan tampil detail dari data student dengan id 6



Tetapi apabila menggunakan function `findOrFail()` maka yang tampil adalah halaman 404 jika data tidak ada

```

40  public function show($student)  

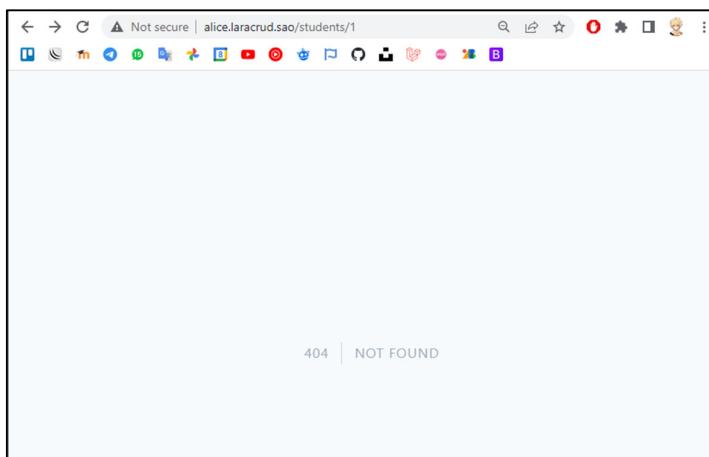
41  

42  // $result = Student::find($student);  

43  $result = Student::findOrFail($student);  

44  return view('show', ['student' => $result]);
45

```



6. Modifikasi view index

Pada view index dapat ditambahkan tombol add students dan kolom action. Sehingga akan nampak seperti ini:

Students Table						Add Student
#	NIM	Name	Gender	Major	Address	Action
1	3202016004	Muhamad Ardilepa	Male	Informatics Engineering	Sintang	Detail Edit Delete
2	3201216004	Kamisato Ayaka	Female	Informatics Engineering	Narukami Island	Detail Edit Delete

7. Menambah flash message dan unique nim validation

Flash message merupakan salah satu fungsi dari class session yang berfungsi sebagai kondisi yang akan dijalankan sekali. Pada method store, dapat ditambahkan beberapa baris kode berikut.

```

20   public function store(Request $request)
21   {
22     $validateData = $request->validate([
23       'nim' => 'required|size:10|unique:students',
24       'name' => 'required|min:3|max:50',
25       'gender' => 'required|in:M,F',
26       'major' => 'required',
27       'address' => '',
28     ]);
29     Student::create($validateData);
30     $request->session()->flash('msg', "{$validateData['name']} data successfully
31     added");
32     return redirect()->route('students.index');
  
```

Sehingga apabila memasukkan data dengan nim yang sudah terdaftar pada database, akan muncul pesan error seperti ini:

Student Registration

NIM
3202016004 (1)
The nim has already been taken.

Full Name
Data Baru

Gender
 Male Female

Major
Informatics Engineering

Address
Jalan Kebenaran

Submit **Back**

Dan jika data yang dimasukkan valid, maka akan langsung me-“redirect” ke view index dengan mengirimkan flash message.

Students Table

Add Student

Layla data successfully added

#	NIM	Name	Gender	Major	Address	Action
1	3202016004	Muhamad Ardalepa	Male	Informatics Engineering	Sintang	Detail Edit Delete
2	3201216004	Kamisato Ayaka	Female	Informatics Engineering	Narukami Island	Detail Edit Delete
3	3201216003	Layla	Female	Informatics Engineering	Narukami Island	Detail Edit Delete

Sehingga pada view index akan muncul flash message yang akan hilang jika halaman dimuat ulang.

8. Method edit() dan update()

Method edit() digunakan untuk menampilkan form edit, sedangkan **method update()** digunakan untuk melakukan proses update datanya.

```
39 |     public function edit(Student $student)
40 |     {
41 |         return view('edit', ['student' => $student]);
42 |     }
43 |     public function update(Request $request, Student $student)
44 |     {
45 |         $validateData = $request->validate([
46 |             'nim' => 'required|size:10|unique:students,nim,' . $student->id,
47 |             'name' => 'required|min:3|max:50',
48 |             'gender' => 'required|in:M,F',
49 |             'major' => 'required',
50 |             'address' => '',
51 |         ]);
52 |         $student->update($validateData);
53 |         return redirect()->route('students.index', ['student' => $student->id])
54 |             ->with(['msg', "{$validateData['name']} data successfully updated"]);
55 |     }

```

Untuk view edit sendiri sama dengan view registration. Hanya saja, field-field nya sudah diisi dengan data yang ada pada database.

```
13 | <body>
14 |     <div class="container pt-4 bg-white mb-5">
15 |         <div class="row">
16 |             <div class="col-md-8 col-xl-6">
17 |                 <h1>Edit Student</h1>
18 |                 <hr>
19 |                 <form action="{{ route('students.update',[ 'student'=>$student->id]) }}" method="POST">
20 |                     @csrf
21 |                     <div class="mb-3">
22 |                         <label for="nim" class="form-label">NIM</label>
23 |                         <input type="text" name="nim" id="nim" value="{{ old('nim') ?? $student->nim }}"
24 |                             class="form-control @error('nim') is-invalid @enderror">
25 |                         @error('nim')
26 |                             <div class="text-danger">{{ $message }}</div>
27 |                         @enderror
28 |                     </div>
29 |                     <div class="mb-3">
30 |                         <label for="name" class="form-label">Full Name</label>
31 |                         <input type="text" name="name" id="name" value="{{ old('name') ?? $student->name }}"
32 |                             class="form-control @error('name') is-invalid @enderror">
33 |                         @error('name')
34 |                             <div class="text-danger">{{ $message }}</div>
35 |                         @enderror
36 |                     </div>
37 |                     <div class="mb-3">
38 |                         <label for="gender" class="form-label">Gender</label>
39 |                         <div class="d-flex">
40 |                             <div class="form-check me-3">
41 |                                 <input type="radio" name="gender" id="male" value="M" class="form-check-input" {{(old('gender')?? $student->gender=='M')? 'checked': ''}}>
42 |                                 <label for="male" class="form-check-label">Male</label>
43 |                             </div>
44 |                             <div class="form-check me-3">
45 |                                 <input type="radio" name="gender" id="female" value="F" class="form-check-input" {{(old('gender')?? $student->gender=='F')? 'checked': ''}}>
46 |                                 <label for="female" class="form-check-label">Female</label>
47 |                             </div>
48 |                         </div>
49 |                     </div>
50 |                 </form>
51 |             </div>
```

```

54         @enderror
55     </div>
56     <div class="mb-3">
57       <label for="major" class="form-label">Major</label>
58       <select name="major" id="major" class="form-select">
59         <option value="Informatics Engineering" {{ (old('major'))?? $student->major=='Informatics
60           Engineering'
61           ?'selected':'' )}}>
62           Informatics Engineering
63         </option>
64         <option value="Informatics System" {{ (old('major'))?? $student->major=='Informatics System'
65           ?'selected':'' )}}>
66           Informatics System
67         </option>
68         <option value="Computer Science" {{ (old('major'))?? $student->major=='Computer Science'
69           ?'selected':'' )}}>
70           Computer Science
71         </option>
72         <option value="Computer Engineering" {{ (old('major'))?? $student->major=='Computer
73           Engineering' ?'selected':'' )}}>
74           Computer Engineering
75         </option>
76         <option value="Telecommunications Engineering" {{ (old('major'))?? $student->
77           major=='Telecommunications Engineering' ?'selected':'' )}}>
78           Telecommunications Engineering
79         </option>
80       </select>
81       @error('major')
82       <div class="text-danger">{{ $message }}</div>
83     @enderror
84   </div>
85   <div class="mb-3">
86     <label for="address" class="form-label">Address</label>
87     <textarea name="address" id="address" rows="3" name="address"
88       class="form-control">{{(old('address'))?? $student->address)}</textarea>
89   </div>
90   <button type="submit" class="btn btn-primary">Update</button>
91   <a href="{{ url()}}>previous()</a> <span>Back</a>
92 </form>
93 </div>
94 </div>
95 
```

Sehingga akan tampil seperti ini:

The screenshot shows a web browser window with the URL `alice.laracrud.sao/students/9/edit`. The page title is "Edit Student". The form contains the following fields:

- NIM:** 3201216003
- Full Name:** Layla
- Gender:** Female (radio button selected)
- Major:** Informatics Engineering
- Address:** Narukami Island

At the bottom of the form are two buttons: "Update" (blue) and "Back" (green).

Apabila tombol update ditekan, maka akan melakukan perubahan data, lalu melakukan redirect ke view index sambil mengirimkan flash message.

The screenshot shows a web browser window with the URL 'Not secure | alice.laracrud.sao/students?student=9'. The page title is 'Students Table' and there is a 'Add Student' button. A green flash message box says 'Igarashi Layla data successfully updated'. Below it is a table with columns: #, NIM, Name, Gender, Major, Address, and Action. The table contains three rows of student data. Each row has 'Detail', 'Edit', and 'Delete' buttons in the Action column.

#	NIM	Name	Gender	Major	Address	Action
1	3202016004	Muhamad Ardalepa	Male	Informatics Engineering	Sintang	[Detail] [Edit] [Delete]
2	3201216004	Kamisato Ayaka	Female	Informatics Engineering	Narukami Island	[Detail] [Edit] [Delete]
3	3201816003	Igarashi Layla	Female	Informatics Engineering	Grand Narukami Shrine	[Detail] [Edit] [Delete]

9. Method destroy()

Method destroy digunakan untuk menghapus data pada table.

```
56 public function destroy(Student $student)
57 {
58     $student->delete();
59     return redirect()->route('students.index', ['student' => $student->id])
60         ->with('msg', "$student->name data successfully deleted");
61 }
```

Sehingga apabila tombol delete ditekan, akan menghapus data.

The screenshot shows a web browser window with the URL 'Not secure | alice.laracrud.sao/students?student=9'. The page title is 'Students Table' and there is a 'Add Student' button. A green flash message box says 'Igarashi Layla data successfully deleted'. Below it is a table with columns: #, NIM, Name, Gender, Major, Address, and Action. The table contains two rows of student data. Each row has 'Detail', 'Edit', and 'Delete' buttons in the Action column.

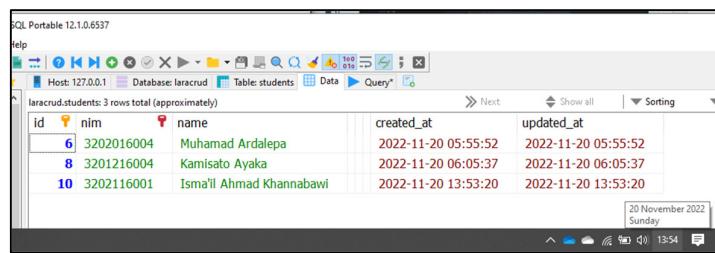
#	NIM	Name	Gender	Major	Address	Action
1	3202016004	Muhamad Ardalepa	Male	Informatics Engineering	Sintang	[Detail] [Edit] [Delete]
2	3201216004	Kamisato Ayaka	Female	Informatics Engineering	Narukami Island	[Detail] [Edit] [Delete]

C. Tambahan

Apabila melihat kepada database, terdapat kolom created dan updated at. Namun waktu tersebut tidak sesuai dengan waktu sebenarnya. Maka dari itu pada file \config\app dapat mengganti pengaturan waktu sebagai berikut.

```
72     'timezone' => 'Asia/Jakarta',
73     'locale' => 'id',
74     'fallback_locale' => 'en',
75     'faker_locale' => 'id_ID',
```

Sehingga apabila ditambahkan data baru, waktu akan menyesuaikan waktu di Indonesia.



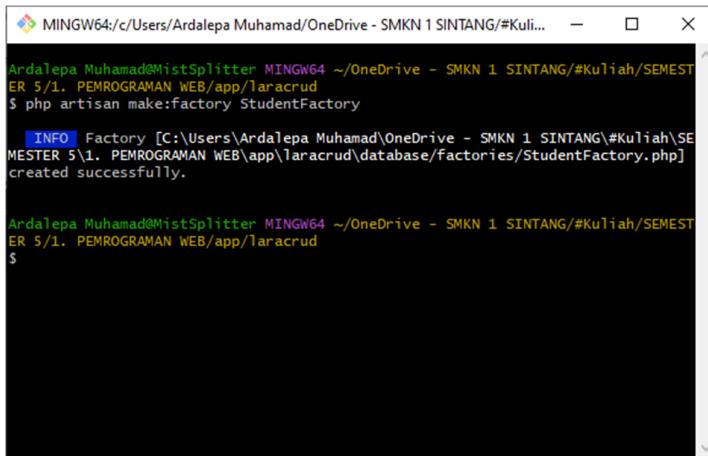
The screenshot shows a database interface for 'laracrud.students' with three rows of data. The columns are 'id', 'nim', 'name', 'created_at', and 'updated_at'. The 'updated_at' column shows timestamps that have been converted to Indonesian time (Sunday, 20 November 2022). The data is as follows:

			created_at	updated_at
6	3202016004	Muhamad Ardalepa	2022-11-20 05:55:52	2022-11-20 05:55:52
8	3202126004	Kamisato Ayaka	2022-11-20 06:05:37	2022-11-20 06:05:37
10	3202116001	Isma'il Ahmad Khannabawi	2022-11-20 13:53:20	2022-11-20 13:53:20

D. Tugas

Karena pada tugas ini akan membuat beberapa fitur yang membutuhkan record yang lumayan banyak pada table. Maka sebelumnya kita akan menambahkan terlebih dulu beberapa record lagi menggunakan database seeder dan factory.

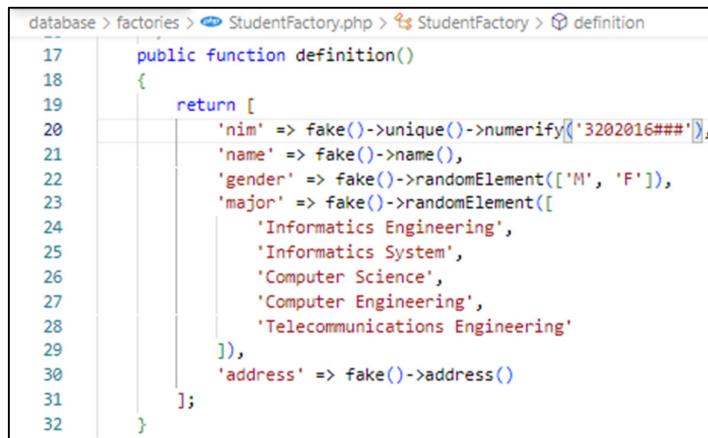
Pertama jalankan perintah `php artisan make:factory StudentFactory` untuk membuat factory dari student.



```
MINGW64:/c/Users/Ardalepa Muhamad/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud
$ php artisan make:factory StudentFactory
[INFO] Factory [C:\Users\Ardalepa Muhamad\OneDrive - SMKN 1 SINTANG/#Kuliah\SEMESTER 5\1. PEMROGRAMAN WEB\app\laracrud\database\factories\StudentFactory.php] created successfully.

Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5\1. PEMROGRAMAN WEB\app\laracrud
$
```

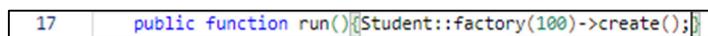
Sehingga pada file `\database\factory` terdapat file `studentfactory`. Lalu pada method definition, tambahkan baris kode berikut untuk menentukan value apa yang akan digenerate pada tiap-tiap kolom table students.



```
database > factories > StudentFactory.php > StudentFactory > definition
17     public function definition()
18     {
19         return [
20             'nim' => fake()->unique()->numerify('3202016###'),
21             'name' => fake()->name(),
22             'gender' => fake()->randomElement(['M', 'F']),
23             'majon' => fake()->randomElement([
24                 'Informatics Engineering',
25                 'Informatics System',
26                 'Computer Science',
27                 'Computer Engineering',
28                 'Telecommunications Engineering'
29             ]),
30             'address' => fake()->address()
31         ];
32     }

```

Selanjutnya, buka file `\database\seeds\DatabaseSeeder.php` pada method run, tambahkan kode berikut untuk menentukan berapa banyak data yang akan digenerate.



```
17     public function run()
18     {
19         Student::factory(100)->create();
20     }

```

Langkah terakhir yaitu dengan menjalankan perintah `php artisan migrate:fresh --seed`. Untuk menjalankan seeder dan factory faker yang telah kita buat tadi.

```

MINGW64:/c/Users/Ardalepa Muhamad/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud
$ php artisan migrate:fresh --seed

Dropping all tables ..... 165ms DONE
INFO Preparing database.
Creating migration table ..... 27ms DONE
INFO Running migrations.

2014_10_12_000000_create_users_table ..... 53ms DONE
2014_10_12_100000_create_password_resets_table ..... 48ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 45ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 70ms DONE
2022_11_19_161303_create_students_table ..... 38ms DONE

INFO Seeding database.

Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud
$ |

```

Sehingga pada database akan terdapat 100 record baru dengan data yang random.

laracrud:students: 100 rows total (approximately)						
id	nim	name	gender	major	address	created_at
93	3202016010	Dewi Dwi Ariandini	M	Telecommunications En...	Jl. Veteran No. 109, Te...	2022-11-11 10:45:11
94	3202016018	Soleh Jallani	M	Telecommunications En...	Jl. Asia Afrika No. 844, ...	2022-11-11 10:45:11
95	3202016126	Bahuvirya Hidayanto	F	Computer Science	Jl. KebonJati No. 508, Sl...	2022-11-11 10:45:11
96	3202016037	Vanya Yuniar S.Sos	F	Computer Science	Jl. Arifin No. 255, Denp...	2022-11-11 10:45:11
97	3202016993	Emong Permati S.Kom	F	Informatics Engineering	Jln. Moch. Ramdan No. ...	2022-11-11 10:45:11
98	3202016690	Eka Gangsar Irawan	M	Informatics Engineering	Jln. Dipenogoro No. 442...	2022-11-11 10:45:11
99	3202016282	Damu Surya Rajasa S.Pt	F	Computer Science	Ds. PHH. Mustofa No. 3...	2022-11-11 10:45:11
100	3202016723	Chelsea Kuswandi	F	Computer Science	Jl. Banda No. 189, Beng...	2022-11-11 10:45:11

1. Membuat fitur search

Untuk membuat fitur search, pertama-tama kita akan membuat sebuah form seperti ini:

```

22    <form action="{{ route('students.index') }}">
23        <div class="input-group mb-3">
24            <input type="search" class="form-control" placeholder="Search... " name="search"
25                value="{{ request('search') }}>
26            <button class="btn btn-primary" type="submit">Search</button>
27        </div>
28    </form>
29    @if (request('search'))
30        <div class="alert alert-success alert-dismissible fade show" role="alert">
31            Found <b>{{ $row }}</b> record
32            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
33        </div>
34    @endif

```

Lalu pada method index() ubah menjadi berikut ini:

```

12    public function index()
13    {
14        $student = new Student;
15        $student = $student->when(request('search') ?? false, function ($student) {
16            return $student->where('nim', 'LIKE', '%' . request('search') . '%')
17                ->orWhere('name', 'LIKE', '%' . request('search') . '%')
18                ->orWhere('major', 'LIKE', '%' . request('search') . '%')
19                ->orWhere('address', 'LIKE', '%' . request('search') . '%');
20        });
21    }

```

Sehingga apabila kita mengetikkan sesuatu pada field search akan mencari berdasarkan nim, name, major, dan address.

#	NIM	Name	Gender	Major	Address	Action
1	3202016250	Jaga Wijaya	Male	Computer Science	Psr. Antapani Lama No. 196, Magelang 43140, DIY	[Detail] [Edit] [Delete]
2	3202016898	Qori Hasanah	Male	Informatics System	Gg. Padma No. 748, Jayapura 54466, Sultra	[Detail] [Edit] [Delete]
3	3202016464	Devi Widastuti	Female	Informatics System	Dk. Bah Jaya No. 975, Tegal 19875, Sultra	[Detail] [Edit] [Delete]
4	3202016884	Padmi Jeita Susanti	Male	Computer Science	Gg. Tubagus Ismail No. 829, Jayapura 20834, Kepri	[Detail] [Edit] [Delete]

2. Membuat fitur filter

Untuk membuat fitur filter, pertama modifikasi form search menjadi seperti ini :

```
<form action="{{ route('students.index') }}" class="mb-3">
    <div class="input-group">
        <label class="input-group-text" for="filter">Filter by</label>
        <select class="form-select" id="filter" name="filter" style="flex: 1 0 auto">
            <option value="0" {{ request('filter')?request('filter')=='0'?'selected':'':'' }}>All
            </option>
            <option value="nim" {{ request('filter')?request('filter')=='nim' ? 'selected':'':'' }}>NIM
            </option>
            <option value="name" {{ request('filter')?request('filter')=='name' ? 'selected':'':'' }}>Name
            </option>
            <option value="major" {{ request('filter')?request('filter')=='major' ? 'selected':'':'' }}>Major
            </option>
            <option value="address" {{ request('filter')?request('filter')=='address' ? 'selected':'':'' }}>Address
            </option>
        </select>
        <input type="search" class="form-control" placeholder="Search. . ." name="search"
            value="{{ request('search') }}" style="flex: 1 0 auto">
        <button class="btn btn-primary" type="submit">Search</button>
    </div>
</form>
```

Lalu pada method index ubah seperti ini:

```
12  public function index()
13  {
14      $student = new Student;
15      $student = $student->when(request('search') ?? false, function ($student) {
16          return $student->where('nim', 'LIKE', '%' . request('search') . '%')
17              ->orWhere('name', 'LIKE', '%' . request('search') . '%')
18              ->orWhere('major', 'LIKE', '%' . request('search') . '%')
19              ->orWhere('address', 'LIKE', '%' . request('search') . '%');
20      });
21
22      $student = $student->when(!isEmpty(request('filter')) ?? false, function ($student) {
23          return $student->where(request('filter'), 'LIKE', '%' . request('search') . '%');
24      });
25  }
```

Sehingga secara default, fitur pencarian akan mencari dari semua kolom jika tidak ditentukan filternya.

Students Table					
Filter by Address <input type="text" value="ben"/> <input type="button" value="Search"/>					Add Student
Found 4 record					
#	NIM	Name	Gender	Major	Action
1	3202016226	Ella Mardhiyah S.Pt	Female	Informatics System	Jr. Merdeka No. 503, Padang 71729, Bengkulu
2	3202016640	Sabrina Klandra Pratiwi	Female	Informatics System	Ki. R.M. Said No. 95, Administrasi Jakarta Pusat 94526, Bengkulu
3	3202016108	Putpa Dina Usada M.Ak	Female	Informatics System	Ki. Rajawali Timur No. 897, Kediri 60897, Bengkulu
4	3202016723	Chelsea Kuswandari	Female	Computer Science	Jr. Banda No. 189, Bengkulu 69331, Aceh

3. Membuat fitur sort

view

```

38      <label class="input-group-text" for="sort">Sort by</label>
39      <select class="form-select" id="sort" name="sort" style="flex: 2 0 auto">
40          <option value="created_at" {{ request('sort')?request('sort')=='created_at' ?'selected':'::' }}>
41              Created
42          </option>
43          <option value="updated_at" {{ request('sort')?request('sort')=='updated_at' ?'selected':'::' }}>
44              Updated
45          </option>
46          <option value="nim" {{ request('sort')?request('sort')=='nim' ?'selected':'::' }}>
47              NIM
48          </option>
49          <option value="name" {{ request('sort')?request('sort')=='name' ?'selected':'::' }}>
50              Name
51          </option>
52          <option value="major" {{ request('sort')?request('sort')=='major' ?'selected':'::' }}>
53              Major
54          </option>
55          <option value="address" {{ request('sort')?request('sort')=='address' ?'selected':'::' }}>
56              Address
57          </option>
58      </select>
59      <select class="form-select" id="sc" name="sc" style="flex: 1 0 auto">
60          <option value="a" {{ request('sc')?request('sc')=='address' ?'selected':'::' }}>
61              ASC
62          </option>
63          <option value="d" {{ request('sc')?request('sc')=='address' ?'selected':'::' }}>
64              DESC
65          </option>
66      </select>
67
68

```

controller

```

12  public function index()
13  {
14      $student = new Student;
15      $student = $student->when(request('search') ?? false, function ($student) {
16          return $student->where('nim', 'LIKE', '%' . request('search') . '%')
17              ->orWhere('name', 'LIKE', '%' . request('search') . '%')
18              ->orWhere('major', 'LIKE', '%' . request('search') . '%')
19              ->orWhere('address', 'LIKE', '%' . request('search') . '%');
20      });
21
22      $student = $student->when(!isEmpty(request('filter')) ?? false, function ($student) {
23          return $student->where(request('filter'), 'LIKE', '%' . request('search') . '%');
24      });
25
26
27      $student = $student->when(request('sort') ?? false, function ($student) {
28          return $student->orderBy(request('sort'), request('sc'));
29      });
30
31      return view('index', ['students' => $student->get(), 'row' => $student->count()]);
32

```

#	NIM	Name	Gender	Major	Address	Action
1	3202016697	Wulan Mardiyah	Male	Telecommunications Engineering	Dk. Samanhudi No. 492, Dumat 14759, Jateng	<button>Detail</button> <button>Edit</button> <button>Delete</button>
2	3202016004	Muhamad Aralepa	Male	Informatics Engineering	Sintang	<button>Detail</button> <button>Edit</button> <button>Delete</button>
3	3202016226	Ella Mardiyah S.Pt	Female	Informatics System	Jl. Merdeka No. 503, Padang 71729, Bengkulu	<button>Detail</button> <button>Edit</button> <button>Delete</button>
4	3202016528	Candrakanta Lazuardi	Female	Telecommunications Engineering	Psr. Gedebage Selatan No. 513, Administrasi Jakarta Pusat 28265, Sulsel	<button>Detail</button> <button>Edit</button> <button>Delete</button>

4. Membuat fitur pagination

Pertama modifikasi method index menjadi seperti ini:

```

12 |     public function index()
13 |     {
14 |         $student = new Student();
15 |         $student = $student->when(request('search') ?? false, function ($student) {
16 |             return $student->where('nim', 'LIKE', '%' . request('search') . '%')
17 |                 ->orwhere('name', 'LIKE', '%' . request('search') . '%')
18 |                 ->orwhere('major', 'LIKE', '%' . request('search') . '%')
19 |                 ->orwhere('address', 'LIKE', '%' . request('search') . '%');
20 |         });
21 |
22 |         $student = $student->when(request('filter') ?? false, function ($student) {
23 |             return $student->where(request('filter'), 'LIKE', '%' . request('search') . '%');
24 |         });
25 |
26 |
27 |         $student = $student->when(request('sort') ?? false, function ($student) {
28 |             return $student->orderBy(request('sort'), request('sc'));
29 |         });
30 |         return view('index', ['students' => $student->paginate(7)->withQueryString(), 'row' => $student->count()]);
31 |     }

```

Pada bagian view, tambahkan satu baris kode, lalu bungkus dengan flex container.

```

124 |         </table>
125 |         <div class="d-flex justify-content-center">
126 |             {{ $students->links() }}
127 |         </div>

```

Sehingga akan tampil seperti berikut ini:

#	NIM	Name	Gender	Major	Address	Action
1	3202016282	Jaeman Vinsen Sinaga S.E.I	Female	Telecommunications Engineering	Jl. Bambu No. 733, Banjarbaru 61135, Pabar	<button>Detail</button> <button>Edit</button> <button>Delete</button>
2	3202016506	Karna Nashiruddin	Female	Computer Science	Psr. Aceh No. 903. Sungai Penuh 71486, Papua	<button>Detail</button> <button>Edit</button> <button>Delete</button>
3	3202016250	Jaga Wijaya	Male	Computer Science	Psr. Antapani Lama No. 196, Magelang 43140, DIY	<button>Detail</button> <button>Edit</button> <button>Delete</button>
4	3202016961	Vega Firmansyah	Female	Informatics Engineering	Jl. Supomo No. 666, Medan 60413, Aceh	<button>Detail</button> <button>Edit</button> <button>Delete</button>
5	3202016334	Xanana Haryanto	Male	Informatics Engineering	Dk. Baladeva No. 722, Tarakan 72725, Kalsel	<button>Detail</button> <button>Edit</button> <button>Delete</button>
6	3202016389	Humaira Hilda Hastuti S.Pd	Female	Informatics Engineering	Gg. Nakula No. 869, Palopo 76073, Banten	<button>Detail</button> <button>Edit</button> <button>Delete</button>
7	3202016717	Yance Puspita	Female	Computer Science	Kpg. M.T. Haryono No. 392, Prabumulih 64249, Jabar	<button>Detail</button> <button>Edit</button> <button>Delete</button>

5. Membuat fitur confirmation box

Pertama, ganti button delete pada table dengan trigger button modal box bootstrap.

```

116
117
118
119 <button type="button" class="btn btn-danger btn-sm" data-bs-toggle="modal"
120   data-bs-target="#exampleModal{{ $loop->iteration }}>Delete
121
122
123
124

```

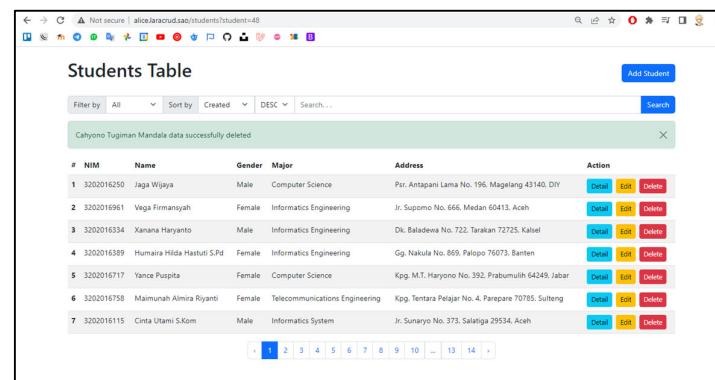
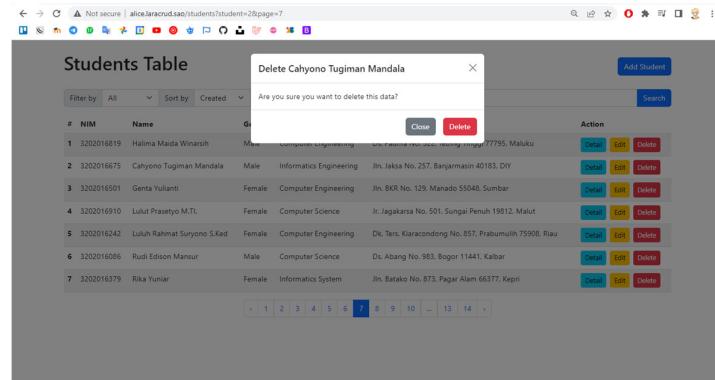
Lalu tambahkan dibawahnya modal boxnya.

```

132 <div class="modal fade" id="exampleModal{{ $loop->iteration }}" tabindex="-1"
133   aria-labelledby="exampleModal{{ $loop->iteration }}Label" aria-hidden="true">
134   <div class="modal-dialog">
135     <div class="modal-content">
136       <div class="modal-header">
137         <h1 class="modal-title fs-5" id="exampleModal{{ $loop->iteration }}Label">Delete
138         {{ $student->name }} </h1>
139         <button type="button" class="btn-close" data-bs-dismiss="modal"
140           aria-label="Close"></button>
141       </div>
142       <div class="modal-body">
143         Are you sure you want to delete this data?
144       </div>
145       <div class="modal-footer">
146
147         <button type="button" class="btn btn-secondary"
148           data-bs-dismiss="modal">Close</button>
149         <form action="{{ route('students.destroy',['student'=>$student->id]) }}"
150           method="post">
151           @method('DELETE')
152           @csrf
153           <button type="submit" class="btn btn-danger">Delete</button>
154         </form>
155       </div>
156     </div>
157   </div>
158 </div>
159

```

Sehingga apabila user menekan tombol delete, tidak akan langsung menghapus data, tetapi akan ada modal box untuk konfirmasi



6. Fitur Login.

Untuk membuat fitur login, yang pertama kali dilakukan adalah membuat controller baru dengan nama AuthController. Untuk membuat controller dapat menggunakan perintah.

```
php artisan make:controller AuthController
```

Sehingga pada folder controller akan terdapat file AuthController. Selanjutnya, import class façade Auth, karena kita akan menggunakan fitur authentication.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Auth;
7
8 class AuthController extends Controller
9 {
```

Lalu buat route baru pada web.php serta tambahkan middleware pada setiap method get route.

```
routes > web.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\AuthController;
5 use App\Http\Controllers\StudentController;
6
7 Route::get('/students', [StudentController::class, 'index'])->name('students.index')->middleware('auth');
8 Route::get('/students/create', [StudentController::class, 'create'])->name('students.create')->middleware('auth');
9 Route::post('/students', [StudentController::class, 'store'])->name('students.store');
10 Route::get('/students/{student}', [StudentController::class, 'show'])->name('students.show')->middleware('auth');
11 Route::get('/students/{student}/edit', [StudentController::class, 'edit'])->name('students.edit')->middleware('auth');
12 Route::put('/students/{student}', [StudentController::class, 'update'])->name('students.update');
13 Route::delete('/students/{student}', [StudentController::class, 'destroy'])->name('students.destroy');
14
15 Route::get('/login', [AuthController::class, 'index'])->name('login.index')->middleware('guest');
16 Route::post('/login', [AuthController::class, 'login'])->name('login.login');
17 Route::post('/logout', [AuthController::class, 'logout'])->name('login.logout');
```

Middleware auth digunakan untuk melakukan pemeriksaan apakah terdapat user yang telah login, jika tidak ada, maka arahkan ke halaman login.

```
11 class RouteServiceProvider extends ServiceProvider
12 {
13     /**
14      * The path to the "home" route for your application.
15      *
16      * Typically, users are redirected here after authentication.
17      *
18      * @var string
19      */
20     public const HOME = '/mahasiswa';
21
22     /**
23      * Define your route model bindings, pattern filters, etc.
24      *
```

Lakukan pengaturan pada file authenticate seperti pada gambar diatas. Pengaturan ini dilakukan untuk menentukan route dari home. Maksudnya, apabila sudah ada user yang login, tetapi user mencoba melakukan akses ke halaman login yang menggunakan middleware guest, maka halaman akan melakukan redirect ke halaman students sebagai home.

Lalu lakukan pengaturan route service provider seperti pada gambar dibawah ini. Pengaturan ini dilakukan apabila halaman yang menggunakan middleware auth diakses, sedangkan belum ada user yang login, maka halaman akan melakukan redirect kepada route login.index

```
15     protected function redirectTo($request)
16     {
17         if (!$request->expectsJson()) {
18             return route('login.index');
19         }
20     }
```

Lalu tambahkan method index, login, dan logout pada controller.

```
10     public function index()
11     {
12         return view('login');
13     }
14
15     public function login()
16     {
17         $credentials = request()->validate([
18             'email' => ['required', 'email'],
19             'password' => ['required'],
20         ]);
21
22         if (Auth::attempt($credentials)) {
23             request()->session()->regenerate();
24             return redirect()->intended('students')->with('msg', "Welcome Back");
25         }
26
27         return back()->with('msg', "Login failed")->onlyInput('email');
28     }
29
30     public function logout()
31     {
32         Auth::logout();
33         request()->session()->invalidate();
34         request()->session()->regenerateToken();
35         return redirect('/login');
36     }
}
```

Selanjutnya membuat view login. View login ini dibuat dengan memanfaatkan templateing engine dari laravel. Sehingga terdapat 3 file terpisah, yaitu base.blade.php -sebagai template utama-, navbar.blade.php - memuat navbar secara partial, saya memisahkan file navbar untuk memudahkan pengaturan jika tidak ada user yang login, maka navbar tidak ditampilkan- dan login.blade.php -memuat form login-. Selain ketiga file ini,

saya juga melakukan perubahan view index, create, dan edit pada percobaan sebelumnya.

a. base.blade.php

```
resources > views > base.blade.php > html > head > link
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <link rel="stylesheet" href="/css/bootstrap.css">
9      <title>Student Data</title>
10     </head>
11
12     <body>
13         @auth
14             @include('navbar')
15         @endauth
16         <div class="container bg-white">
17             @yield('main')
18         </div>
19         <script src="/js/bootstrap.js"></script>
20     </body>
21
22 </html>
```

b. navbar.blade.php

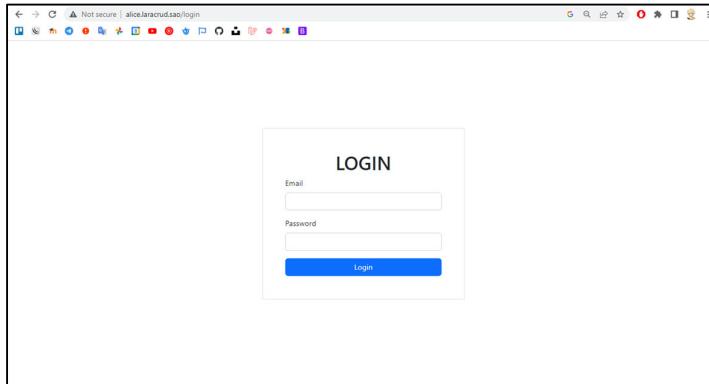
```
resources > views > navbar.blade.php > nav.navbar.navbar-expand-lg.bg-light > div.container.d-flex > div.navbarNav.collapse.nav
1 <nav class="navbar navbar-expand-lg bg-light">
2     <div class="container d-flex">
3         <a class="navbar-brand" href="/students">Muhamad Ardalepa</a>
4         <div class="collapse navbar-collapse" id="navbarNav">
5             <ul class="navbar-nav">
6                 <li class="nav-item">
7                     <a class="nav-link" href="/students">Home</a>
8                 <li class="nav-item">
9                     <a class="nav-link" href="/students/create">Add Students</a>
10                </li>
11            </ul>
12        </div>
13        <form action="{{ route('login.logout') }}" method="post">
14            @csrf
15            <button type="submit" name="logout" class="btn btn-danger" href="logout">Logout</button>
16        </form>
17    </div>
18 </nav>
```

Pada navbar, terdapat form yang isinya hanya tombol logout yang mengarah ke route login.logout. Tombol ini digunakan untuk user logout, jadi user tidak akan bisa logout jika tidak melalui tombol ini.

c. logout.blade.php

```
resources > views > 🗂 login.blade.php > ⚒ div.d-flex.vh-100.justify-content-center.align-items-center > ⚒ div.text-center.p-5.border.col-4
1  @extends('base')
2  @section('main')
3  <div class="d-flex vh-100 justify-content-center align-items-center">
4    <div class="text-center p-5 border col-4">
5      <h1>LOGIN</h1>
6      {{-- flash msg if user login error --}}
7      @if (session()>has('msg'))
8        <div class="alert alert-danger alert-dismissible fade show text-start" role="alert">
9          {{ session()>get('msg') }}
10         <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
11       </div>
12     @endif
13
14     <form action="{{ route('login.login') }}" method="post" class="col-3 d-inline-block text-start w-100">
15       autocomplete="on"
16       @csrf
17       <div class="mb-3">
18         <label for="email" class="form-label">Email</label>
19         <input type="email" name="email" class="form-control" @error('email') is-invalid @enderror id="email">
20         @error('email')
21           <div class="text-danger">{{ $message }}</div>
22         @enderror
23       </div>
24       <div class="mb-3">
25         <label for="password" class="form-label">Password</label>
26         <input type="password" class="form-control" name="password" id="password">
27       </div>
28     <button type="submit" class="btn btn-primary w-100">Login</button>
29   </form>
30 </div>
31 </div>
32 </div>
33
34 @endsection
```

Apabila route login.index dipanggil, maka yang akan tampil adalah seperti ini:



d. index.blade.php

```
resources > views > 🗂 index.blade.php > ...
1  @extends('base')
2  @section('main')
3
4  > <div class="row">...
140  </div>
141
142  @endsection|
```

Untuk index.blade.php, dilakukan pengubahan seperti ini. Yaitu menghilangkan bagian atas dan bawah kode dan menggantinya seperti pada gambar diatas.

Sehingga apabila route student.index dijalankan, akan tampil seperti ini:

#	NIM	Name	Gender	Major	Address	Action
1	3202016347	Ari Mayati S.E.I	Male	Informatics System	Kl. Rajawali Timur No. 689, Blok 94879, Sulsel	Detail Edit Delete
2	3202016170	Zaenab Ina Winarsih M.TL	Female	Computer Science	Kpg. Bass No. 604, Kediri 42941, Kepri	Detail Edit Delete
3	3202016718	Karen Kuswandari S.Psi	Male	Computer Engineering	Dk. Laksamana No. 901, Malang 84795, NTT	Detail Edit Delete
4	3202016657	Hajjo Cagak Gunarto S.Pd	Male	Telecommunications Engineering	Jln. Suprapto No. 814, Administrasi Jakarta Selatan 14701, Sumut	Detail Edit Delete
5	3202016510	Puput Melani	Female	Informatics Engineering	Dk. Hang No. 618, Salatiga 30418, Babel	Detail Edit Delete
6	3202016290	Empluk Rajasa M.M.	Female	Computer Engineering	Ds. Sutoyo No. 125, Semarang 43059, Sumsel	Detail Edit Delete
7	3202016285	Rati Nugroho	Male	Informatics System	Dk. Bakau No. 385, Banjarbaru 27934, Papua	Detail Edit Delete

- e. view create, edit, dan show

Sama seperti pada view index, untuk view-view ini juga diubah seperti ini.

```
resources > views > registration-form.blade.php > ...
1 @extends('base')
2 @section('main')
3
4 > <div class="row">...
80 </div>
81
82 @endsection

resources > views > edit.blade.php > ...
1 @extends('base')
2
3 @section('main')
4
5 > <div class="row">...
85 </div>
86
87 @endsection

resources > views > show.blade.php > ...
1 @extends('base')
2 @section('main')
3
4 > <div class="row">...
33 </div>
34
35 @endsection
```

Sehingga apabila mengakses masing-masing view, akan tampil seperti ini.

Muhamad Ardalepa Home Add Students Logout

Student Registration

NIM
Full Name
Gender
Male Female
Major
Informatics Engineering
Address

Submit Back

Muhamad Ardalepa Home Add Students Logout

Edit Student

NIM
3202016347
Full Name
Ani Maryati S.E.I
Gender
Male Female
Major
Informatics System
Address
Kl. Rajawali Timur No. 689, Blitar 94879, Sulsel

Update Back

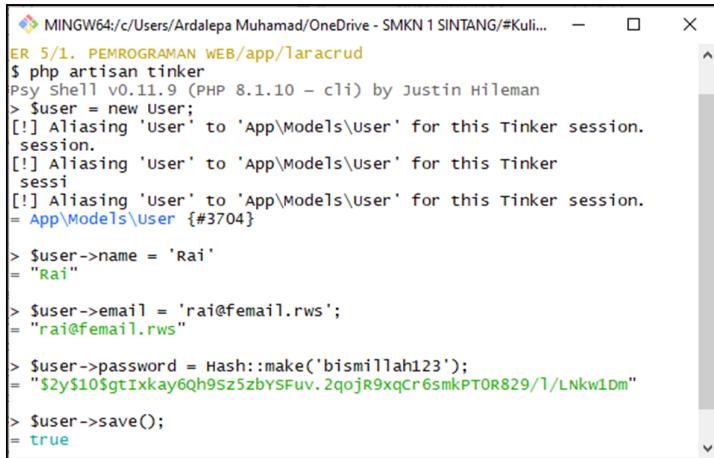
Muhamad Ardalepa Home Add Students Logout

Ani Maryati S.E.I Biodata

NIM	3202016347
Name	Ani Maryati S.E.I
Gender	Male
Major	Informatics System
Address	Kl. Rajawali Timur No. 689, Blitar 94879, Sulsel

Back

Selanjutnya agar user dapat login pastinya harus ada data user pada database. Karena tidak terdapat fitur registrasi, untuk menambahkan user dapat menggunakan tinker seperti berikut ini:



```
MINGW64:/c/Users/Ardalepa Muhamad/OneDrive - SMKN 1 SINTANG/#Kuli... — □ X
ER 5/1. PEMROGRAMAN WEB/app/laracrud
$ php artisan tinker
Psy Shell v0.11.9 (PHP 8.1.10 - cli) by Justin Hileman
> $user = new User;
[!] Aliasing 'User' to 'App\Models\User' for this Tinker session.
session.
[!] Aliasing 'User' to 'App\Models\User' for this Tinker
sessi
[!] Aliasing 'User' to 'App\Models\User' for this Tinker session.
= App\Models\User {#3704}
> $user->name = 'Rai'
= "Rai"
> $user->email = 'rai@email.rws';
= "rai@email.rws"
> $user->password = Hash::make('bismillah123');
= "$2y$10$gtIxkay6Qh9sz5zbYSFuv.2qojR9xqCr6smkPT0R829/l/LNkw1Dm"
> $user->save();
= true
```

Sehingga apabila dilihat pada database, akan terdapat satu user baru.



laracrud:users 1 rows total (approximately)					
id	name	email	email_verified_at	password	remember_to
1	Rai	rai@email.rws	(NULL)	\$2y\$10\$gtIxkay6Qh9sz5zbYSFuv.2qojR9xqCr6smkPT0R829/l/LNkw1Dm	(NULL)

Pada kolom password, password yang diinputkan merupakan hasil hash dari method make pada class Hash.

Sehingga pada tahap ini fitur login dan authentikasi menggunakan middleware telah selesai

E. Middleware

1. Membuat Controller baru

```
Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5\1. PEMROGRAMAN WEB/app/laracrud (main)
$ php artisan make:controller LecturerController

[INFO] Controller [C:\Users\Ardalepa Muhamad\OneDrive - SMKN 1 SINTANG#\Kuliah\SEMESTER 5\1. PEMROGRAMAN WEB\app\laracrud\app\Http\Controllers\LecturerController.php] created successfully.

Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5\1. PEMROGRAMAN WEB/app/laracrud (main)
$
```

2. Membuat method

Membuat method untuk mengetes fitur middleware

```
app > Http > Controllers > LecturerController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class LecturerController extends Controller
8  {
9      public function index()
10     {
11         |
12         return 'lecturer addition form';
13     }
14     public function create()
15     {
16         |
17         return 'lecturer data table';
18     }
19     public function show()
20     {
21         |
22         return 'lecturer detail page';
23     }
24 }
```

3. Membuat route baru.

Agar dapat mengakses controller yang kita buat, kita harus menambahkan route.

```
28 | Route::get('/index', [LecturerController::class, 'index']);
29 | Route::get('/create', [LecturerController::class, 'create']);
30 | Route::get('/show', [LecturerController::class, 'show']);
```

4. Membuat middleware

Membuat middleware menggunakan arisan

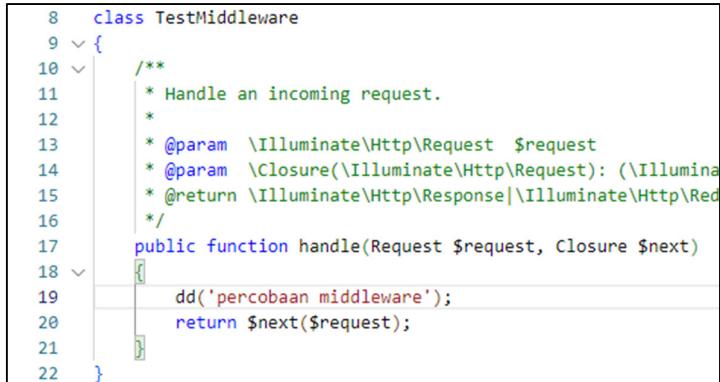


```
Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5\1. PEMROGRAMAN WEB/app/laracrud (main)
$ php artisan make:middleware TestMiddleware

[INFO] Middleware [C:/Users/Ardalepa Muhamad/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5\1. PEMROGRAMAN WEB/app/laracrud/app/Http/Middleware/TestMiddleware.php] created successfully.

Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5\1. PEMROGRAMAN WEB/app/laracrud (main)
$
```

Sehingga pada folder middleware akan tercipta file TestMiddleware, lalu tambahkan kode berikut untuk mengetesnya.



```
8  class TestMiddleware
9  {
10    /**
11     * Handle an incoming request.
12     *
13     * @param \Illuminate\Http\Request $request
14     * @param Closure(\Illuminate\Http\Request) : (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next
15     */
16    public function handle(Request $request, Closure $next)
17    {
18        dd('percobaan middleware');
19        return $next($request);
20    }
21 }
22 }
```

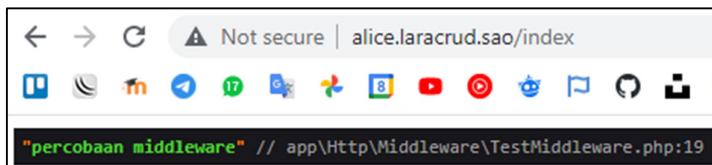
5. Mendaftarkan middleware

Untuk mendaftarkan middleware, dapat menambahkannya pada kernel dengan 3 cara. Pertama, pada \$middleware

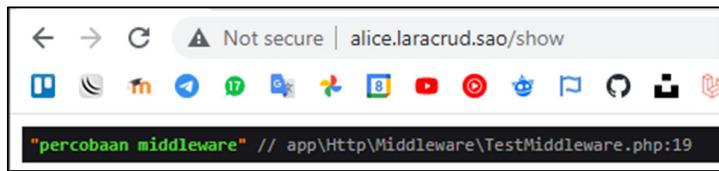
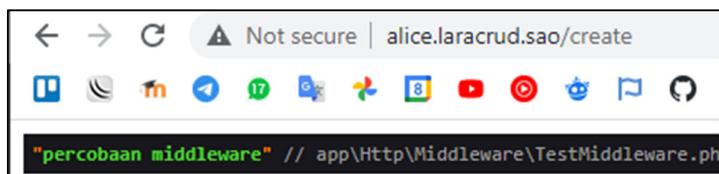


```
16 protected $middleware = [
17     // \App\Http\Middleware\TrustHosts::class,
18     \App\Http\Middleware\TrustProxies::class,
19     \Illuminate\Http\Middleware\HandleCors::class,
20     \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
21     \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
22     \App\Http\Middleware\TrimStrings::class,
23     \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
24     \App\Http\Middleware\TestMiddleware::class,
25 ];
```

Apabila mengakses /create, maka akan tampil seperti berikut:

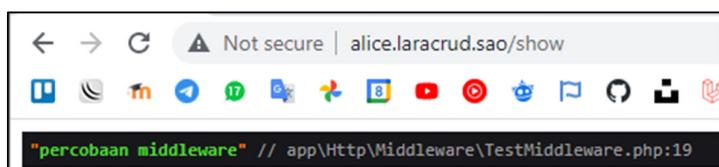
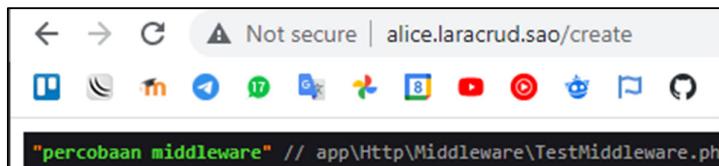
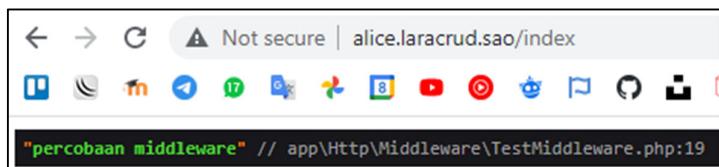


Pada method create() seharusnya menampilkan tulisan lecturer addition form. Tetapi yang tampil sekarang adalah percobaan middleware dalam bentuk dump. Karena test middleware aktif secara global. Sehingga apabila kita mengakses semua route, akan menampilkan percobaan middleware.



Yang kedua adalah middleware group['web'], jadi, apabila route dari web, maka semua route dari web akan menampilkan percobaan middleware.

```
31 protected $middlewareGroups = [
32     'web' => [
33         \App\Http\Middleware\EncryptCookies::class,
34         \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
35         \Illuminate\Session\Middleware\StartSession::class,
36         \Illuminate\View\Middleware\ShareErrorsFromSession::class,
37         \App\Http\Middleware\VerifyCsrfToken::class,
38         \Illuminate\Routing\Middleware\SubstituteBindings::class,
39         \App\Http\Middleware\TestMiddleware::class,
40     ],
41 ];
```



Yang ketiga, adalah middleware route. Middleware route dapat diakses apabila suatu route memanggilnya. Jika route dipanggil, maka tidak akan terjadi perubahan.

The image consists of three vertically stacked screenshots of a web browser window. Each screenshot shows a different URL in the address bar: 'alice.laracrud.sao/show' (lecturer detail page), 'alice.laracrud.sao/create' (lecturer data table), and 'alice.laracrud.sao/index' (lecturer addition form). The browser interface includes standard navigation buttons (back, forward, refresh), a status bar indicating 'Not secure', and a toolbar with various icons.

Agar dapat menggunakan middleware route, dapat memanggil method middleware pada route yang ingin menggunakan. Contohnya pada route index berikut ini:

```
29 | Route::get('/index', [LecturerController::class, 'index'])->middleware('test');
30 | Route::get('/create', [LecturerController::class, 'create']);
31 | Route::get('/show', [LecturerController::class, 'show']);
```

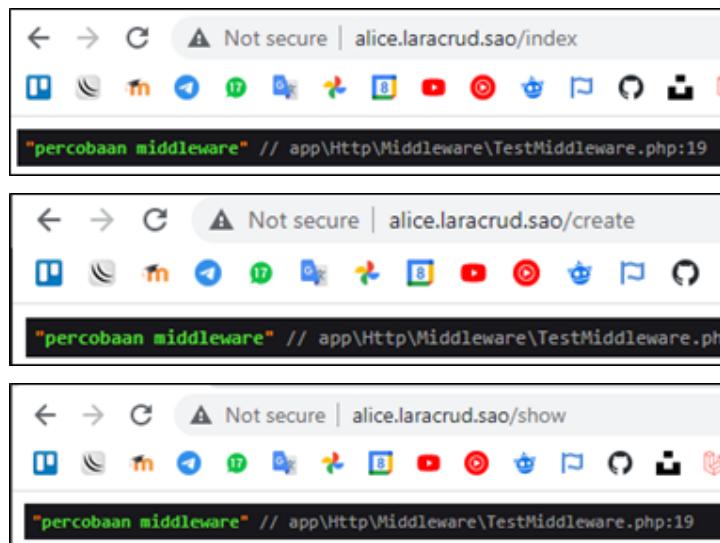
Sehingga apabila route /index diakses, akan tampil seperti ini.

A screenshot of a browser window showing the result of a middleware test. The address bar shows 'alice.laracrud.sao/index'. The main content area displays the text "'percobaan middleware'" followed by the path '// app\Http\Middleware\TestMiddleware.php:19' in green, which is the output of the middleware code.

Selain dengan memanggilnya dari route, dapat mengaktifkan middleware pada method construct controller.

```
7  class LecturerController extends Controller
8  {
9      public function __construct()
10     {
11         $this->middleware('test');
12     }
13     public function index()
14     {
15         return 'lecturer addition form';
16     }
17     public function create()
18     {
19         return 'lecturer data table';
20     }
21     public function show()
22     {
23         return 'lecturer detail page';
24     }
}
```

Sehingga seluruh method dari controller lecturer akan memanggil middleware test



Untuk membatasi method mana yang boleh menggunakan suatu middleware, dapat menggunakan fungsi only dan except dengan parameternya yaitu nama dari methodnya.

```
public function __construct()
{
    $this->middleware('test')->only('create');
}
```

Sehingga pada sintaks diatas ini, hanya method create yang akan menggunakan middleware test.

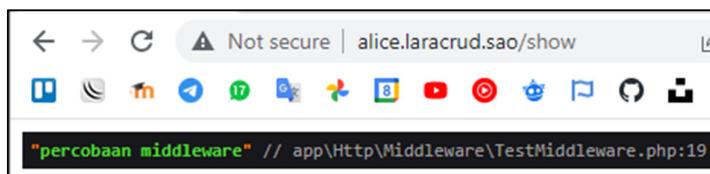
The image contains three separate browser windows. The top window shows the URL 'alice.laracrud.sao/index' with the title 'lecturer addition form'. The middle window shows the URL 'alice.laracrud.sao/show' with the title 'lecturer detail page'. The bottom window shows the URL 'alice.laracrud.sao/create' with the title 'percobaan middleware' // app\Http\Middleware\TestMiddleware.php:'. All three windows have a 'Not secure' warning icon in the address bar.

Selain itu dapat juga menggunakan fungsi except sebagai berikut.

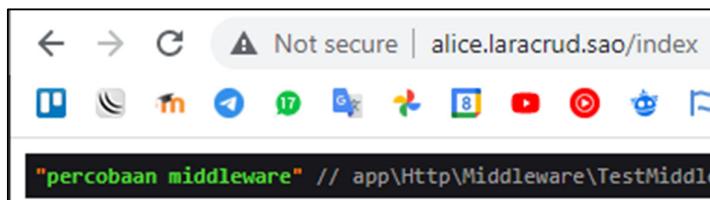
```
9 |     public function __construct()
10| {
11|     |     $this->middleware('test')->except(['create']);
12| }
```

Sehingga selain method create, semua akan menngunakan middleware test.

A single screenshot of a browser window showing a data table with the title 'lecturer data table'. The URL in the address bar is 'alice.laracrud.sao/create'. The browser has a 'Not secure' warning icon.



A screenshot of a web browser window. The address bar shows the URL "alice.laracrud.sao/show". The status bar at the bottom of the browser displays the text "percobaan middleware" followed by the path "/app/Http/Middleware/TestMiddleware.php:19".



A screenshot of a web browser window. The address bar shows the URL "alice.laracrud.sao/index". The status bar at the bottom of the browser displays the text "percobaan middleware" followed by the path "/app/Http/Middleware/TestMiddleware.php:19".

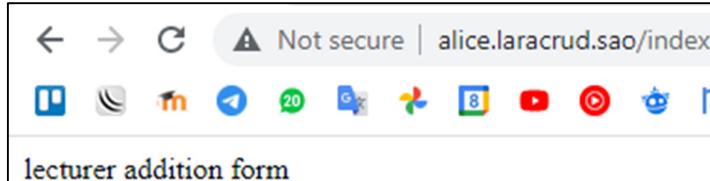
Selain untuk menampilkan dump “percobaan middleware”, middleware juga dapat digunakan untuk fungsi lainnya seperti pada contoh dibawah ini:



```
17 |     public function handle(Request $request, Closure $next)
18 |     {
19 |         if (time() % 2 == 0) {
20 |             return redirect('/index');
21 |         }
22 |         return $next($request);
23 |     }

```

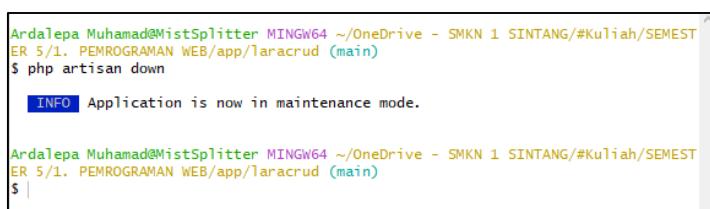
Pada sintaks diatas, apabila timestamp yang dihasilkan modulus 2 == 0, maka halaman akan melakukan redirect ke halaman /index.



A screenshot of a web browser window. The address bar shows the URL "alice.laracrud.sao/index". The page title displayed on the browser is "lecturer addition form".

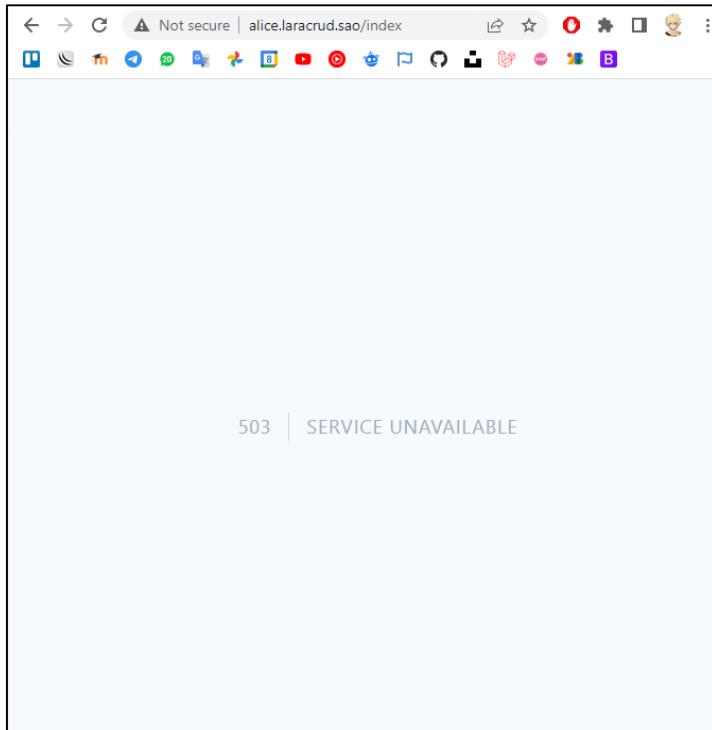
6. Middleware bawaan laravel

Salah satu middleware bawaan laravel yaitu untuk membuat server down atau dalam keadaan maintenance.



```
Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud (main)
$ php artisan down
INFO Application is now in maintenance mode.

Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud (main)
$ |
```



Untuk mengaktifkannya kembali dapat menggunakan perintah up

```
Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud (main)
$ php artisan up

[INFO] Application is now live.

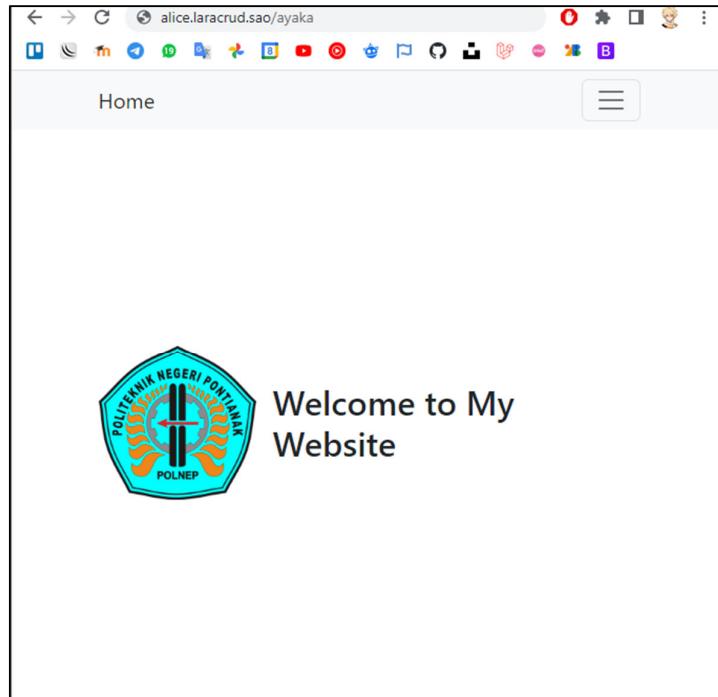
Ardalepa Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud (main)
$ |
```

Agar dapat mengakses web dalam keadaan maintenance dapat menggunakan secreet.

```
Ardalepa.Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud (main)
$ php artisan down --secret="ayaka"

[INFO] Application is now in maintenance mode.

Ardalepa.Muhamad@MistSplitter MINGW64 ~/OneDrive - SMKN 1 SINTANG/#Kuliah/SEMESTER 5/1. PEMROGRAMAN WEB/app/laracrud (main)
$
```



The screenshot shows a web browser window with the URL 'alice.laracrud.sao/ayaka'. The page displays a logo of 'POLITEKNIK NEGERI PONTIANAK' (POLNEP) and the text 'Welcome to My Website'. The browser interface includes a toolbar with various icons and a navigation bar at the top.