

JOBSHEET 13
PRAKTIKUM WEB LANJUT



Oleh:
AGSAL FAIRROHMAD A.P/TI-2A
1841720208

Jurusan Teknologi Informasi
Program Studi Teknik Informatika
Politeknik Negeri Malang
2020

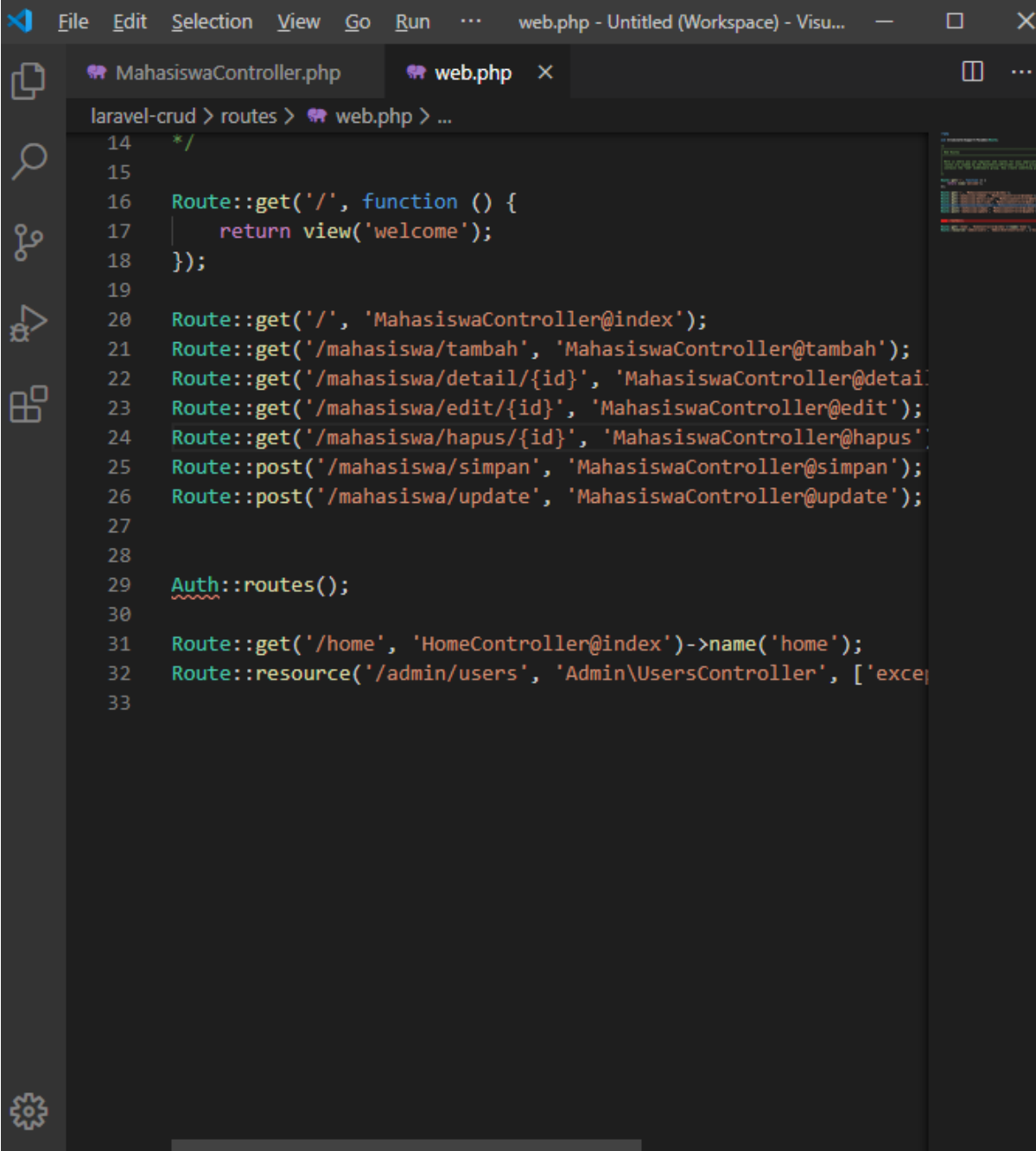
Tujuan

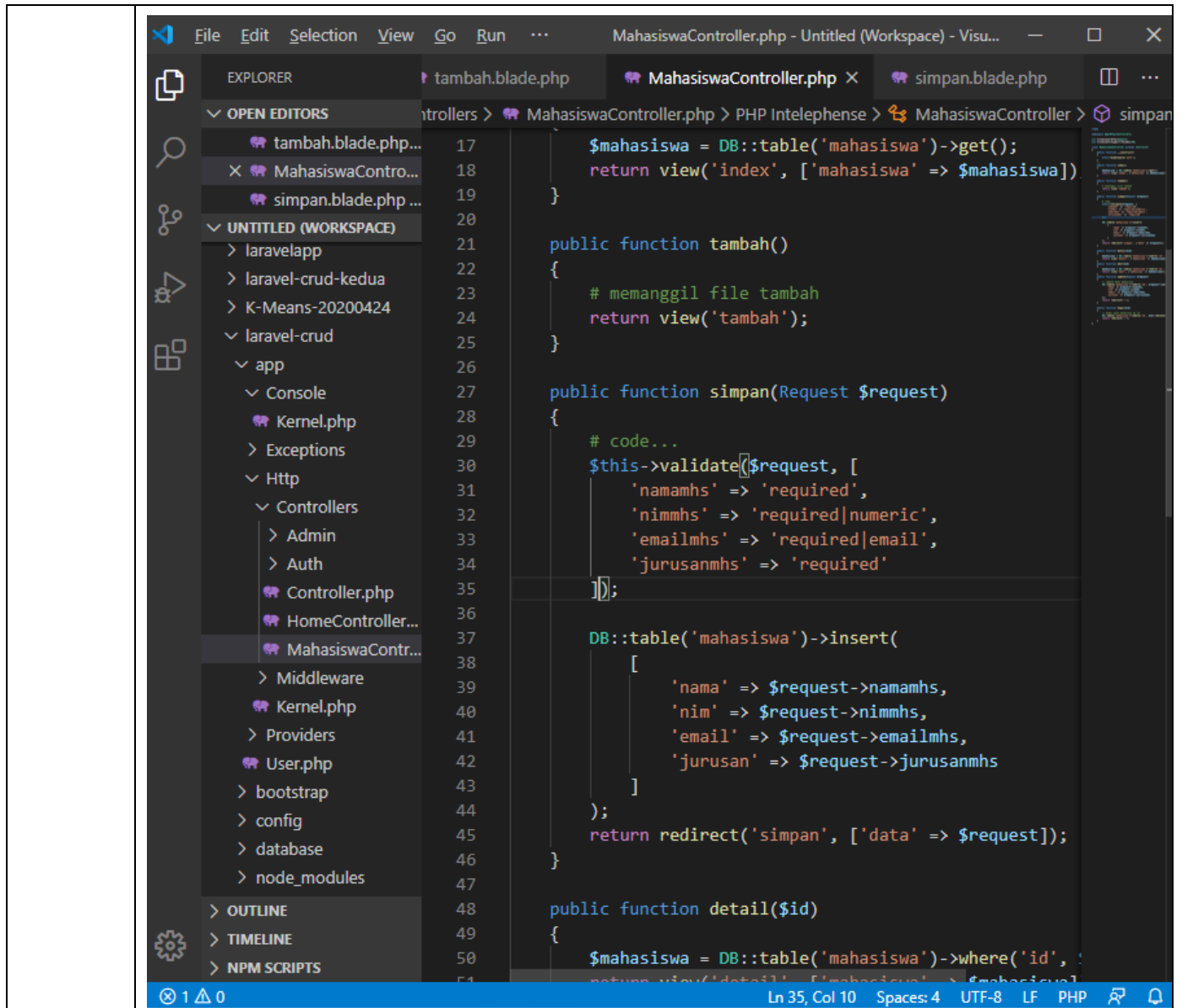
Mahasiswa diharapkan dapat:

1. Memahami bagaimana cara membuat form validasi menggunakan Laravel
2. Memahami bagaimana proses validasi dalam Laravel

Praktikum: Membuat Form Validasi di Laravel

Langkah	Keterangan
1	Buka project Laravel “ laravel-crud ”
2	<p>Buka file laravel-crud/routes/web.php. buatlah 2 route, yaitu route get. dengan url ‘mahasiswa/tambah’ yang akan mengakses method tambah pada controller MahasiswaController. dan route yang 1 nya lagi kita buat route post. dengan url ‘simpan’ yang akan memproses method atau function simpan pada controller MahasiswaController.</p> <pre>20 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 21 22 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan');</pre>

	 <pre> 14 */ 15 16 Route::get('/', function () { 17 return view('welcome'); 18 }); 19 20 Route::get('/', 'MahasiswaController@index'); 21 Route::get('/mahasiswa/tambah', 'MahasiswaController@tambah'); 22 Route::get('/mahasiswa/detail/{id}', 'MahasiswaController@detail'); 23 Route::get('/mahasiswa/edit/{id}', 'MahasiswaController@edit'); 24 Route::get('/mahasiswa/hapus/{id}', 'MahasiswaController@hapus'); 25 Route::post('/mahasiswa/simpan', 'MahasiswaController@simpan'); 26 Route::post('/mahasiswa/update', 'MahasiswaController@update'); 27 28 29 Auth::routes(); 30 31 Route::get('/home', 'HomeController@index')->name('home'); 32 Route::resource('/admin/users', 'Admin\UsersController', ['except' => 'show']); 33 </pre>
3	<p>Buat 2 buah method pada <code>laravel-crud/app/Http/controller/MahasiswaController.php</code>.</p>



```

19     public function tambah()
20     {
21         //memanggil view tambah
22         return view ('tambah');
23     }
24
25     public function simpan(Request $request)
26     {
27
28         $this->validate($request,[
29             'nama' => 'required',
30             'nim' => 'required|numeric',
31             'email' => 'required|email',
32             'jurusan' => 'required'
33         ]);
34
35         // insert data ke table mahasiswa
36         DB::table('mahasiswa')->insert([
37             'nama' => $request->nama,
38             'nim' => $request->nim,
39             'email' => $request->email,
40             'jurusan' => $request->jurusan
41         ]);
42
43         return view('simpan',['data' => $request]);
44     }
45 }

```

Pada method tambah() kita membuat sebuah view yang menampilkan form penginputan sederhana. dimana pada form tersebut akan kita arahkan pemproses nya pada method simpan() menggunakan method post. seperti yang telah kita definisikan pada route.

pada method ini kita melakukan validasi form dengan fungsi validate(). Pertama kita tangkap data-data yang dikirim ke method ini ke dalam variabel \$request. kemudian kita isi variable \$request ini ke dalam parameter pertama fungsi validate(). baru kemudian kita buat pengaturan validasi nya masing-masing.

Pada contoh form validasi laravel sederhana ini kita membuat pengaturan '**required**' pada form nama, nim, email dan jurusan. yang mana artinya form ini wajib untuk di isi (tidak boleh kosong). Jika validasi berhasil, atau jika yang di input sesuai dengan pengaturan yang sudah kita tetapkan dalam fungsi validate() tadi, maka kita passing data yang di input oleh user ke view simpan.blade.php

Berikut beberapa validasi yang sering dibutuhkan.

required	membuat form wajib isi
min:10	membuat minimal karakter yang diinput
max:10	membuat maksimal karakter yang boleh diinput
alpha	hanya boleh menginput huruf
numeric	hanya boleh menginput angka
alpha_num	hanya boleh menginput angka dan huruf (!@#\$\$%^&* dan lain-lain tidak diperbolehkan)
bail	berhenti menjalankan validasi selanjutnya jika validasi sebelumnya gagal (error)
email	hanya boleh menginput email

4

Buat view tambah di **laravel-crud/resources/views/tambah.blade.php**

```

1  @extends('master')
2
3  @section('title','Tambah Data')
4
5  @section('judul_halaman','Tambah Data Mahasiswa')
6
7  @section('konten')
8  <a href="/" class="btn btn-danger">Kembali</a>
9  <br>
10 <br>
11 @if (count($errors)>0)
12 <div class="alert alert-danger">
13     <ul>
14         @foreach($errors->all() as $error)
15             <li>{{$error}}</li>
16         @endforeach
17     </ul>
18 </div>
19 @endif
20 <form action="/mahasiswa/simpan" method="POST">
21     {{ csrf_field() }}
22     <div class="form-group">
23         <label>Nama</label>
24         <input type="text" class="form-control" required=
25     </div>
26     <div class="form-group">
27         <label>NIM</label>
28         <input type="number" class="form-control" required=
29     </div>
30     <div class="form-group">
31         <label>Email</label>
32         <input type="email" class="form-control" required=
33     </div>
34     <div class="form-group">
35         <label>Jurusan</label>

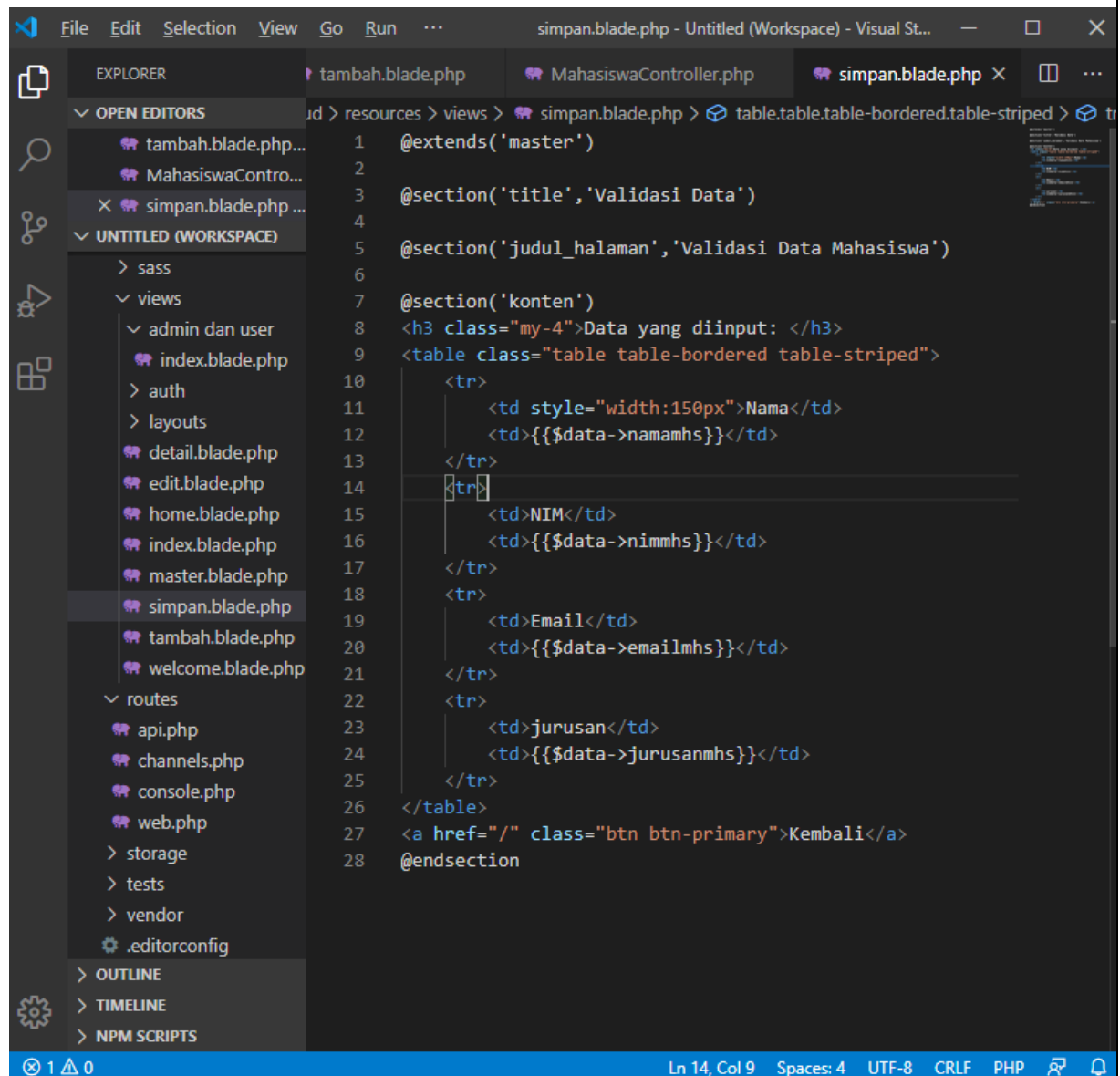
```

pada form ini kita mengarahkan ke route '/mahasiswa/simpan' dengan method post. perhatikan pada value nya masing-masing. ada fitur keren dari laravel. yaitu fungsi old(). fungsi ini berguna untuk menampilkan data yang sebelumnya di input. Jadi jika ada validasi error, maka data yang di isi pada form tidak hilang, tapi tetap masih ditampilkan. karena tentu akan sangat tidak efektif jika user harus menginput data berulang-ulang jika ada validasi yang salah pada penginputan.

Di sana kita juga sudah membuat pengecekan jika ada error dari validasi form nya, maka kita tampilkan errornya secara keseluruhan dan berurut. jika jumlah error lebih besar dari 0 (ada), maka tampilkan semua error dengan menggunakan fungsi foreach().

5

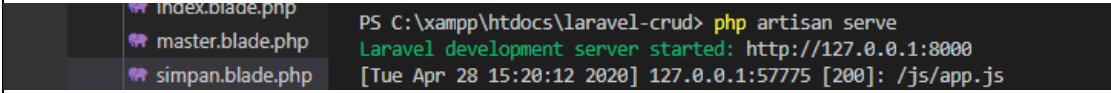
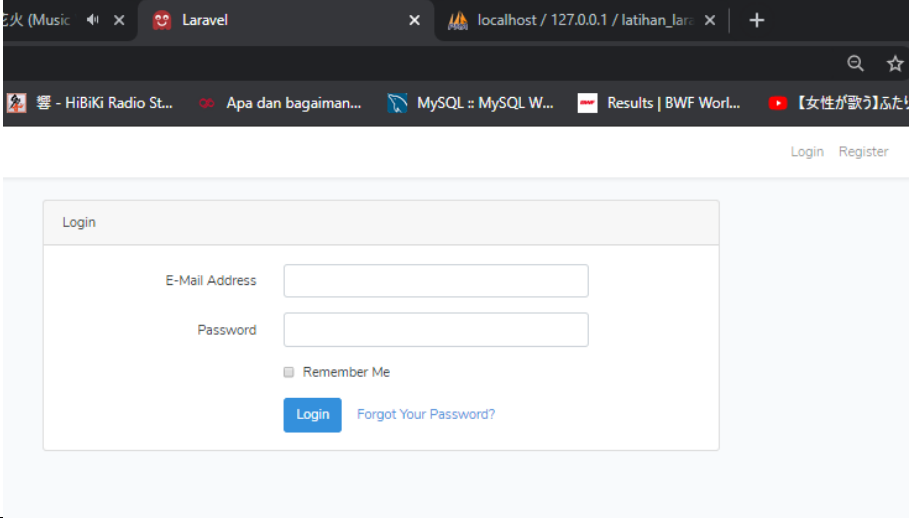
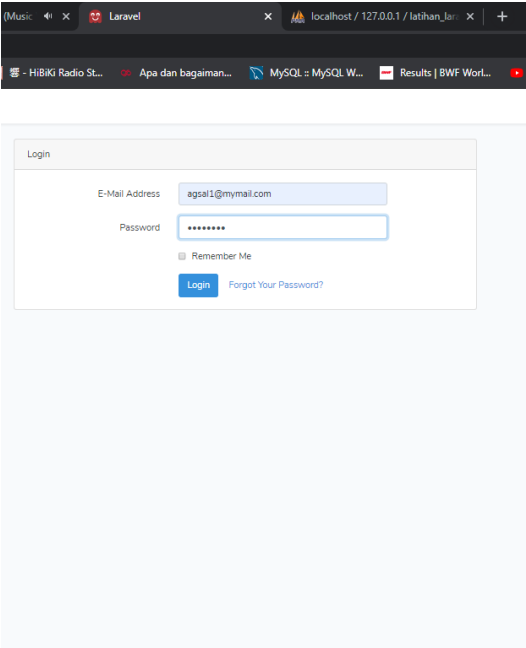
Selanjutnya buat view simpan di **laravel-crud/resources/views/simpan.blade.php**

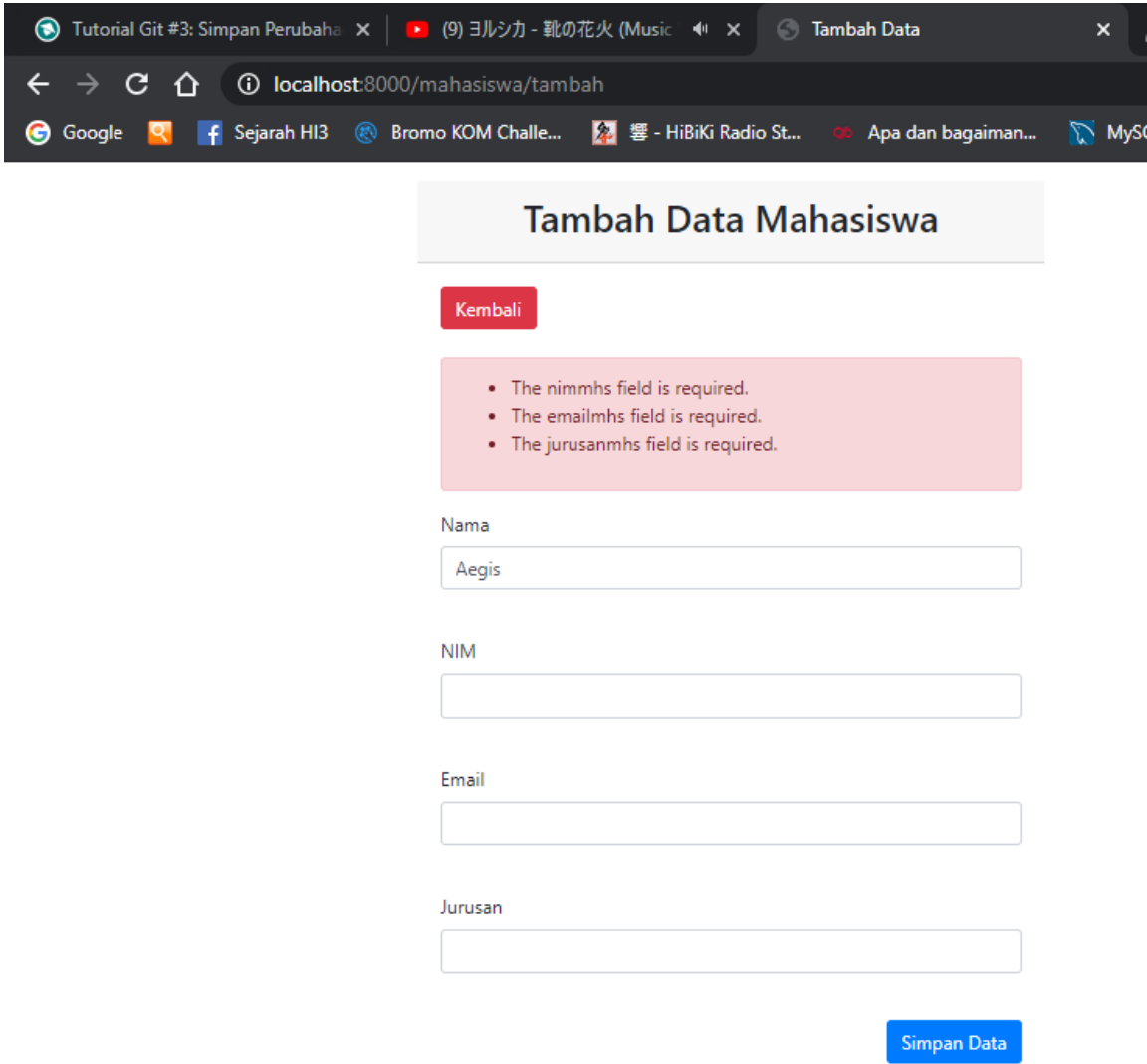


```

1  @extends('master')
2
3  @section('title','Validasi Data')
4
5  @section('judul_halaman','Validasi Data Mahasiswa')
6
7  @section('konten')
8  <h3 class="my-4">Data yang diinput: </h3>
9  <table class="table table-bordered table-striped">
10     <tr>
11         <td style="width:150px">Nama</td>
12         <td>{{ $data->namamhs }}</td>
13     </tr>
14     <tr>
15         <td>NIM</td>
16         <td>{{ $data->nimmhs }}</td>
17     </tr>
18     <tr>
19         <td>Email</td>
20         <td>{{ $data->emailmhs }}</td>
21     </tr>
22     <tr>
23         <td>jurusan</td>
24         <td>{{ $data->jurusanmhs }}</td>
25     </tr>
26 </table>
27 <a href="/" class="btn btn-primary">Kembali</a>
28 @endsection
  
```

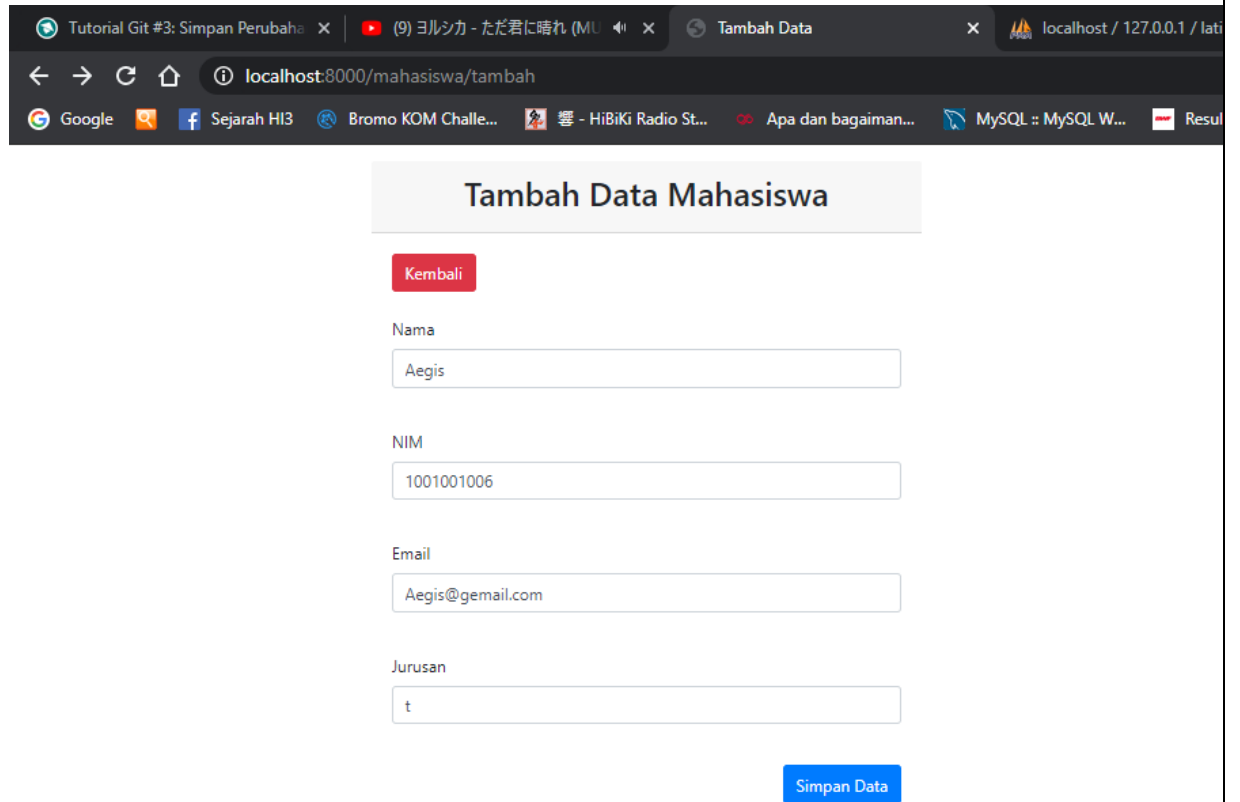
Pada view simpan.blade.php ini kita tampilkan data-data yang dikirim dari method input() tadi.

6	<p>Jalankan Project Laravel Kita</p>  <pre> index.blade.php master.blade.php simpan.blade.php PS C:\xampp\htdocs\laravel-crud> php artisan serve Laravel development server started: http://127.0.0.1:8000 [Tue Apr 28 15:20:12 2020] 127.0.0.1:57775 [200]: /js/app.js </pre>
7.	<p>Akses http://127.0.0.1:8000/</p>  <p>The screenshot shows a web browser window with the URL http://127.0.0.1:8000/. The page displays a login form with the following fields and elements:</p> <ul style="list-style-type: none"> Login (Section Header) E-Mail Address (Text input field) Password (Text input field) <input type="checkbox"/> Remember Me Login (Blue button) Forgot Your Password? (Link)
8	<p>Login sesuai dengan user yang telah dibuat pada pertemuan sebelumnya.</p>  <p>The screenshot shows the same login page as in step 7, but with the following user credentials entered:</p> <ul style="list-style-type: none"> E-Mail Address: agsal1@mymail.com Password: [Redacted with asterisks] <p>The Login button and the Forgot Your Password? link are still visible.</p>

9	<p>Selanjutnya buka http://127.0.0.1:8000/mahasiswa dan Klik tombol Tambah Data Mahasiswa, lalu masukkan data mahasiswa yang tidak lengkap .</p> <p>Setelah itu Klik tombol tambah dan hasilnya akan seperti ini</p>
	 <p>The screenshot shows a web browser window with the address bar displaying <code>localhost:8000/mahasiswa/tambah</code>. The page title is "Tambah Data Mahasiswa". Below the title is a red button labeled "Kembali". A red error message box contains the following text:</p> <ul style="list-style-type: none">• The nimmhs field is required.• The emailmhs field is required.• The jurusanmhs field is required. <p>Below the error message are four input fields:</p> <ul style="list-style-type: none">Nama: AegisNIM: (empty)Email: (empty)Jurusan: (empty) <p>At the bottom right is a blue button labeled "Simpan Data".</p>

11

Masukkan data mahasiswa yang benar, dan hasilnya akan seperti ini.



Tambah Data Mahasiswa

[Kembali](#)

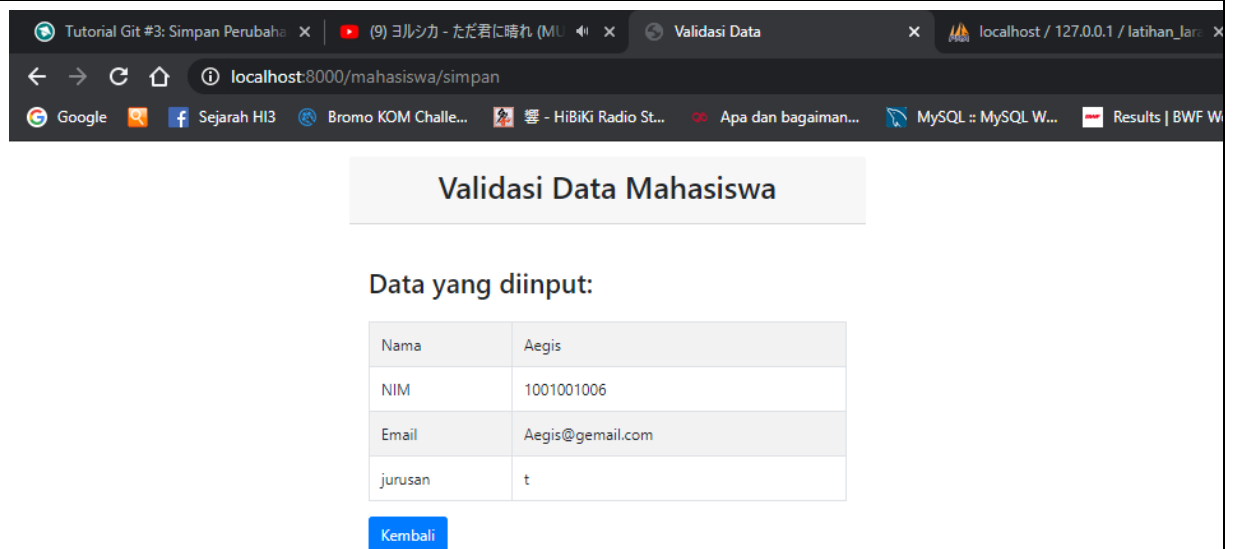
Nama

NIM

Email

Jurusan

[Simpan Data](#)



Validasi Data Mahasiswa

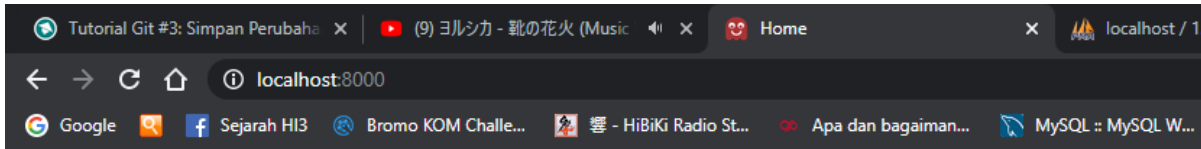
Data yang diinput:

Nama	Aegis
NIM	1001001006
Email	Aegis@gmail.com
jurusan	t

[Kembali](#)

12

Data mahasiswa berhasil ditambahkan, kita dapat mengecek data tersebut pada halaman <http://127.0.0.1:8000/mahasiswa>



Data Mahasiswa

Tambah Data Mahasiswa

Nama	NIM	Action
Agsal F.A.P	1001001001	Detail Edit Hapus
wahyu	1001001002	Detail Edit Hapus
Arcgis	1001001003	Detail Edit Hapus
Mirai	1001001004	Detail Edit Hapus
Sugoy	1001001005	Detail Edit Hapus
Aegis	1001001006	Detail Edit Hapus

-- Selamat Mengerjakan --