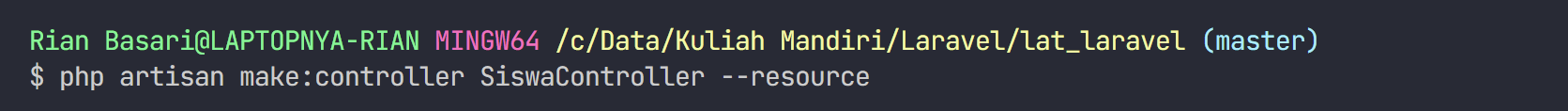
**Save Data dan Validation Custom ERROR MESSAGE**

1. Biasanya kita membuat method satu per satu di controller untuk membuat suatu perintah, termasuk membuat CRUD.
2. Namun, ada cara untuk membuat controller yang isinya secara otomatis dibuatkan method-method dari fungsi CRUD.
3. Caranya, buat controller baru seperti biasa, namun menambahkan **resource** di belakangnya. Atau jika sebelumnya sudah ada SiswaController, rename controller yang sudah ada agar berbeda.



1. Di dalam SiswaController, secara otomatis dibuatkan beberapa method untuk keperluan CRUD.

**Index() :** Untuk menampilkan semua data

**Create() :** Untuk menampilkan halaman form untuk membuat data baru

**Store() :** Untuk menyimpan data baru

**Show() :** Untuk menampilkan detail data

**Edit() :** Untuk menampilkan halaman form untuk mengedit data

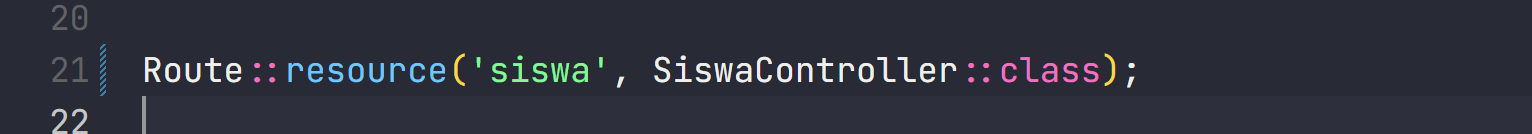
**Update() :** Untuk menyimpan data yang sudah diedit

**Destroy() :** Untuk menghapus data

1. Isi method index() untuk menampilkan datanya. Atau copy isi method index dari controller yang sebelumnya, ke yang baru. Jika ada method lain seperti details(), pindahkan juga sesuai tempat fungsinya.

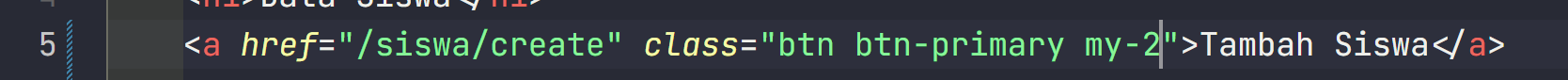


1. Pergi ke route, hapus semua route yang mengarah ke SiswaController. Kemudian buat route baru dengan method **resource**.

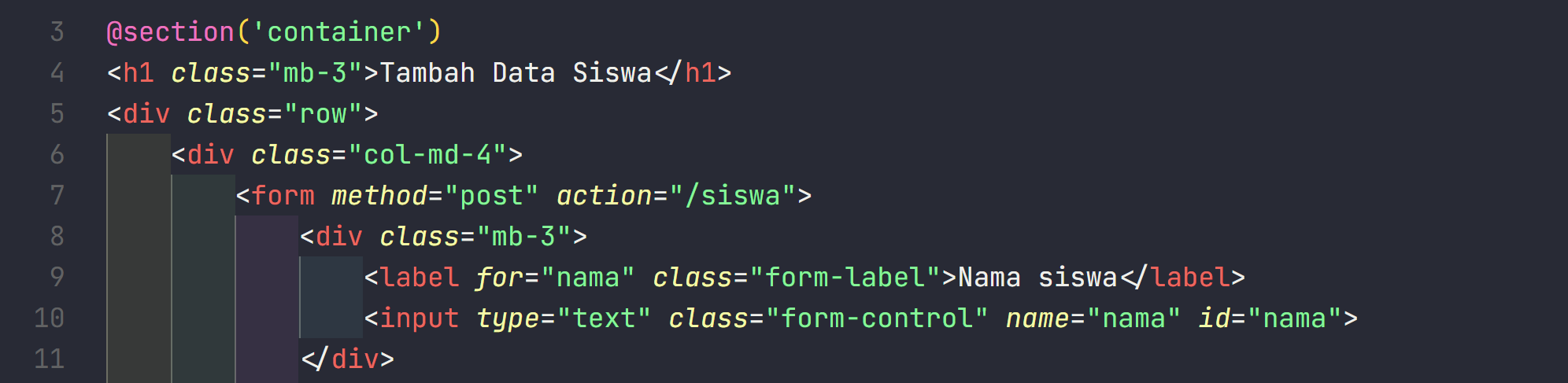


**Membuat fungsi CREATE**

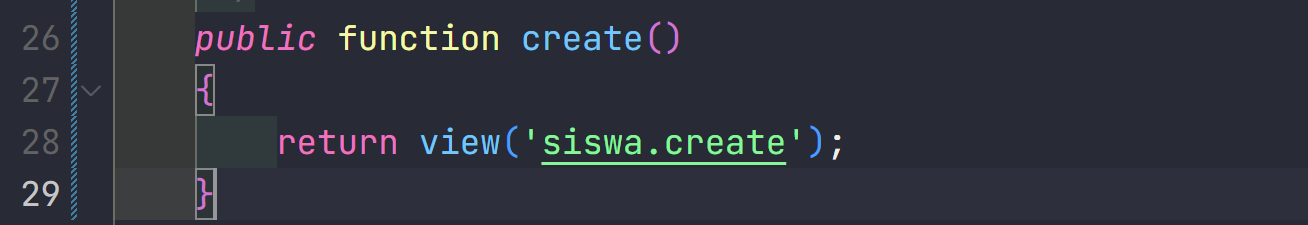
1. Tambahkan button di halaman siswa index untuk mengarahkan ke halaman form tambah data (create).



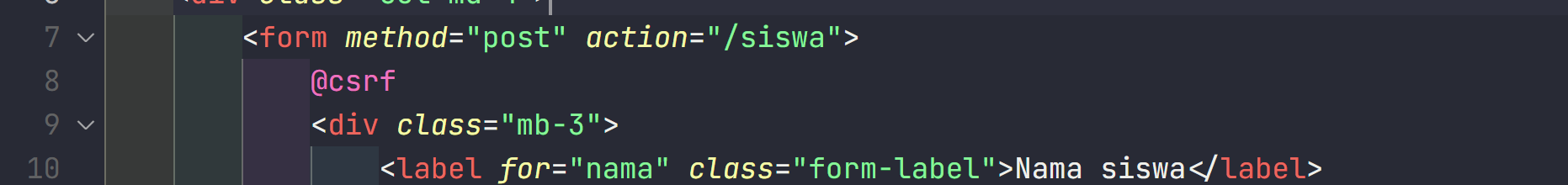
1. Buat view baru untuk halaman form tambah data di folder siswa



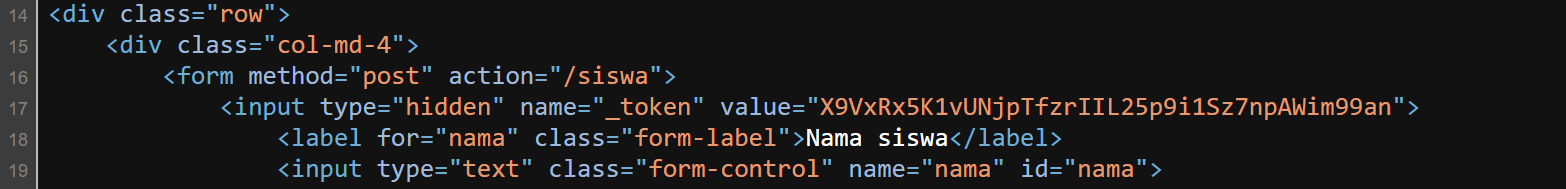
1. Pergi ke SiswaController, kemudian isi method create() agar mengarahkan ke view siswa create.



1. Jika dijalankan dan klik tombol simpan, seharusnya akan mengarahkan ke /siswa. Namun, muncul halaman **419 | PAGE EXPIRED**.
2. Untuk mengatasinya, buat input hidden yang akan men-generate token untuk memastikan form yang dikirim merupakan form yang valid. Caranya, tambahkan **@csrf**.



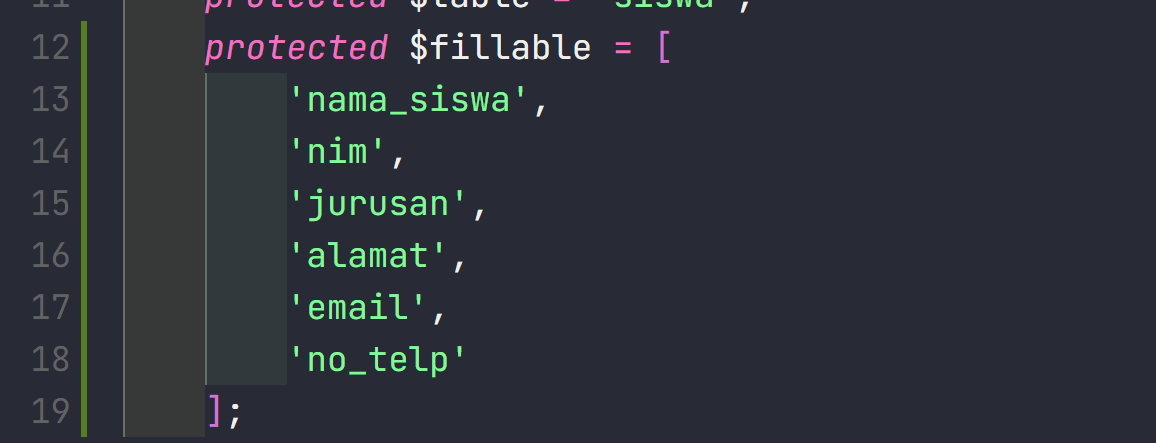
1. Jika dilihat di View Page Source, maka akan muncul input berisi token. Sekarang jika ditekan tombol submit, maka akan mengarahkan ke halaman /siswa



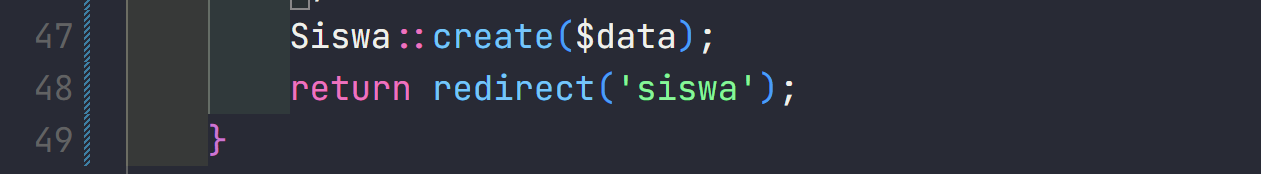
1. Untuk menyimpan data ke dalam database, pergi ke SiswaController dan isi bagian method **store()**.



1. Pergi ke model, tambahkan variable **$fillable** untuk menentukan field mana yang bisa diakses dan diisi.



1. Proses penyimpanan data berhasil, namun tidak langsung redirect ke halaman siswa index. Tambahkan di SiswaController pada method **create()**.

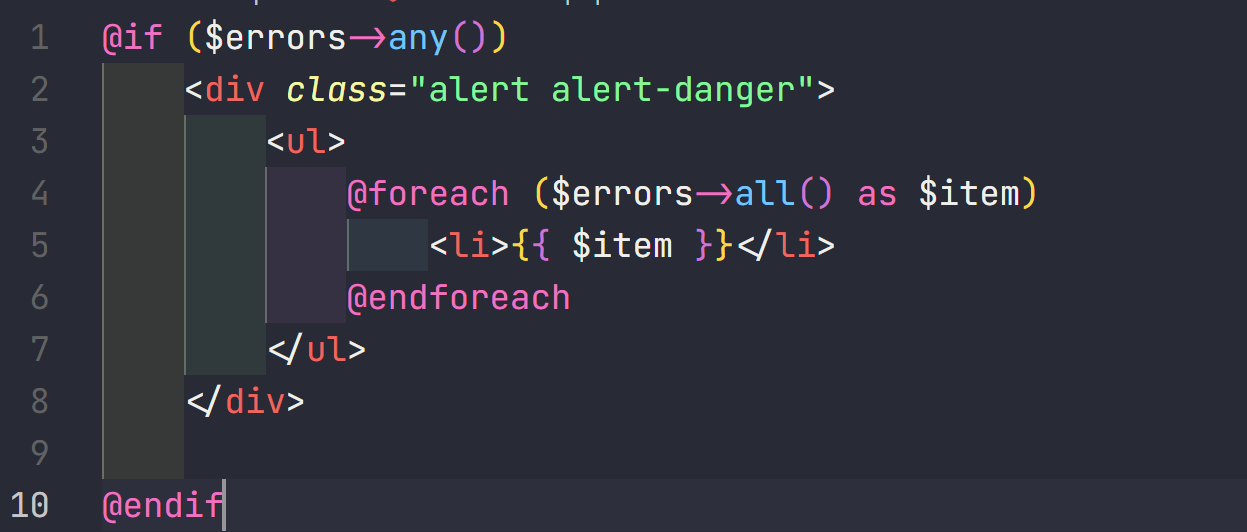


**Membuat validasi form**

1. Pada fungsi CREATE sebelumnya, kita sudah berhasil untuk menambah dan menyimpan data. Namun, jika salah satu field tidak diisi, maka akan muncul error.
2. Menampilkan validasi atau pesan error sangat membantu user untuk mengetahui kesalahan dalam mengimput datanya.
3. Pergi ke SiswaController, pada method **store()**, tambahkan fungsi validate, untuk menambahkan validasi.

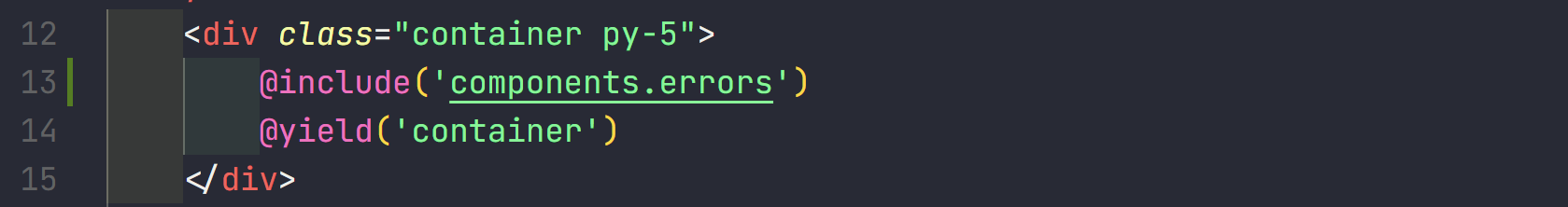


1. Nantinya, pesan error akan dimunculkan di bagian atas form. Artinya, kita akan menampilkan pesan errornya di view main.
2. Buat folder baru bernama **components** yang di dalamnya diisi view dengan nama **message.blade.php**.
3. Tambahkan pesan error dengan sedikit kondisi di dalam view message.

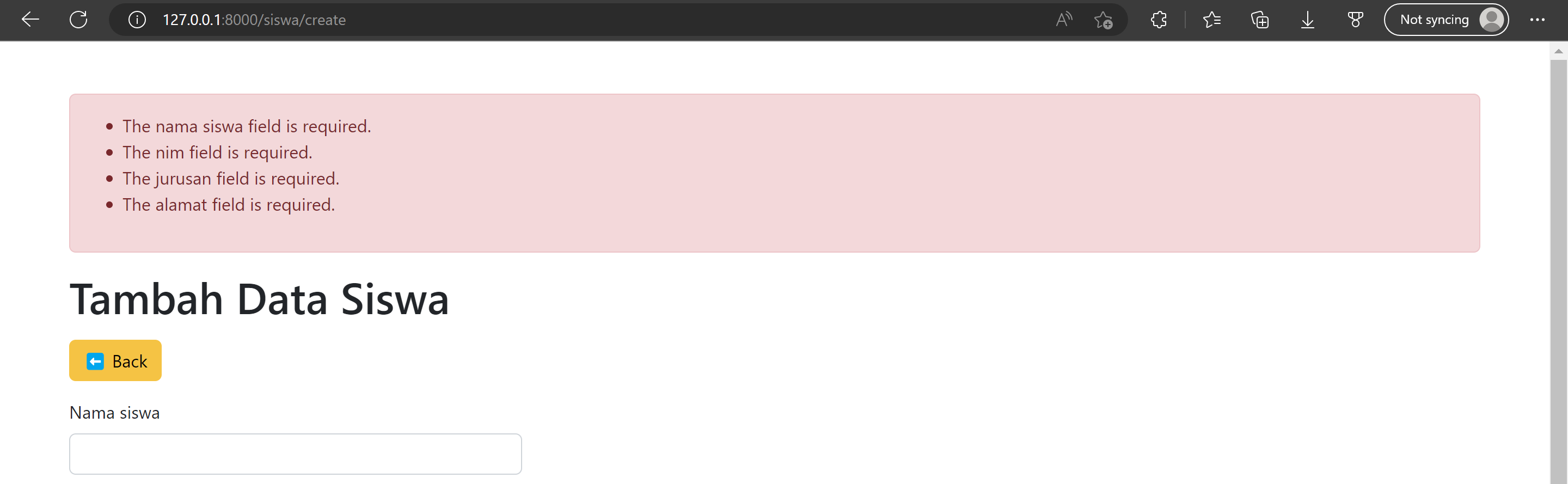


Variable **$message** adalah variable global dari laravel yang bisa ditangkap dimanapun.

1. Include view message di dalam view main.

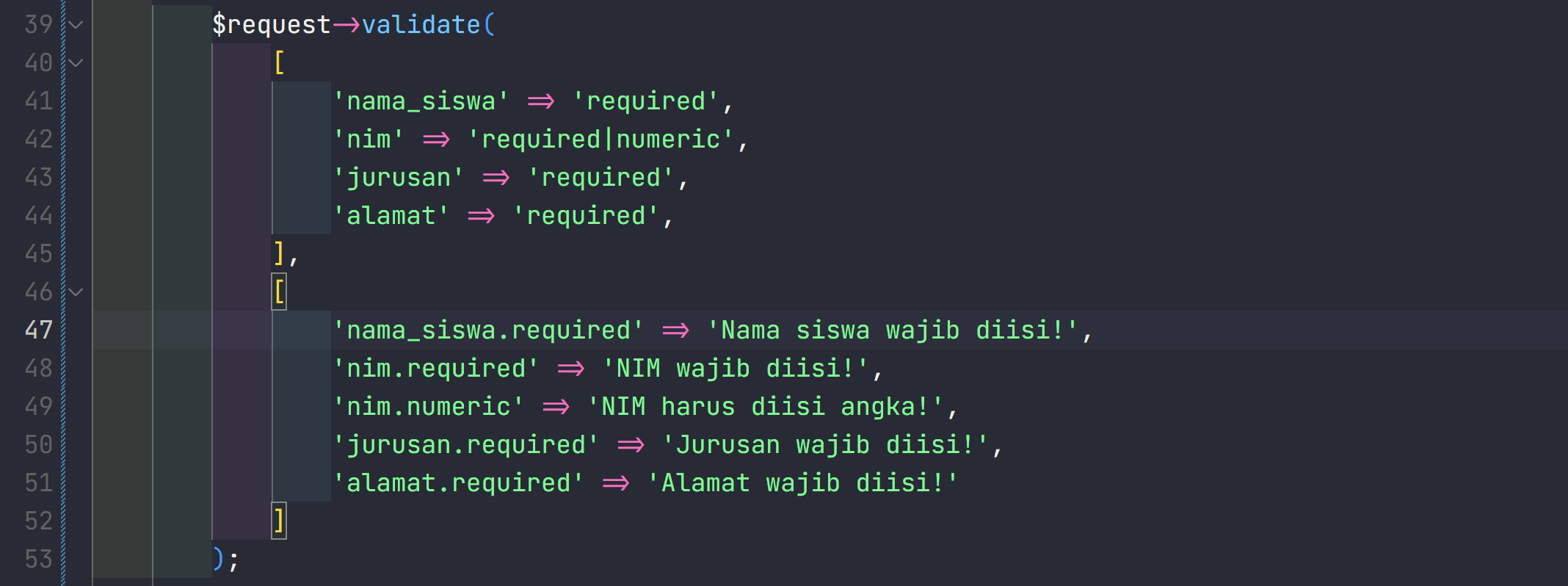


1. Jika dijalankan, akan muncul pesan error



**Membuat custom message pada error**

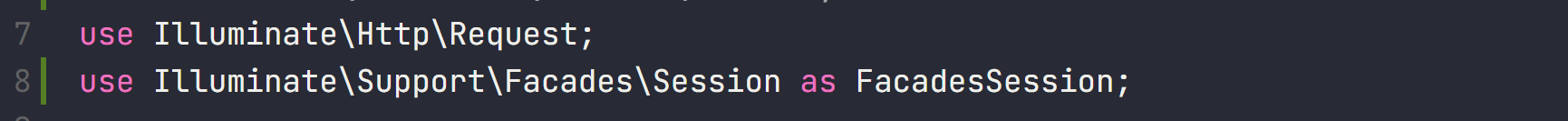
1. Sebelumnya, kita sudah berhasil memunculkan pesan error. Namun, pesan error yang muncul merupakan pesan default dari Laravel, yang mana pesannya berupa bahasa Inggris.
2. Pesan dalam bahasa Inggris ini tentu tidak semua pengguna bisa paham maksudnya. Ditambah dengan nama dari field yang kita gunakan menggunakan bahasa Indonesia.
3. Kita bisa membuat pesan error sesuai dengan yang kita mau.
4. Pergi ke SiswaController, kemudian pada method **store()**, validate-nya kita tambahkan array lagi untuk menyimpan pesan error dari masing-masing field-nya.

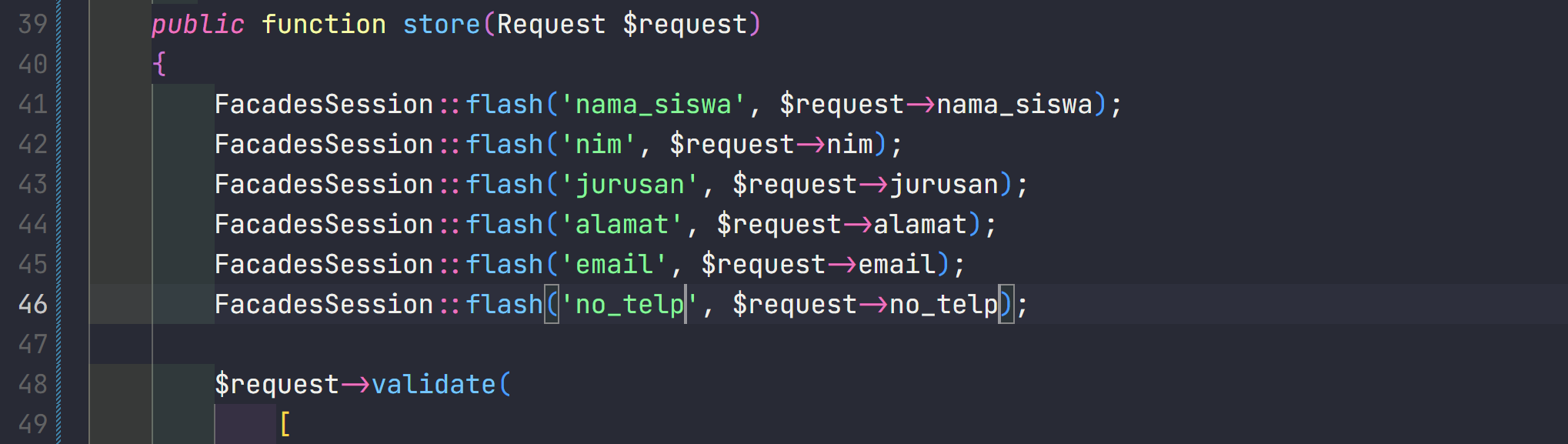


1. Jika dijalankan, pesan error akan muncul seperti apa yang kita mau. Tapi, jika kita tekan submit dan setelah memunculkan pesan error, semua input yang kita telah isi menjadi kosong.

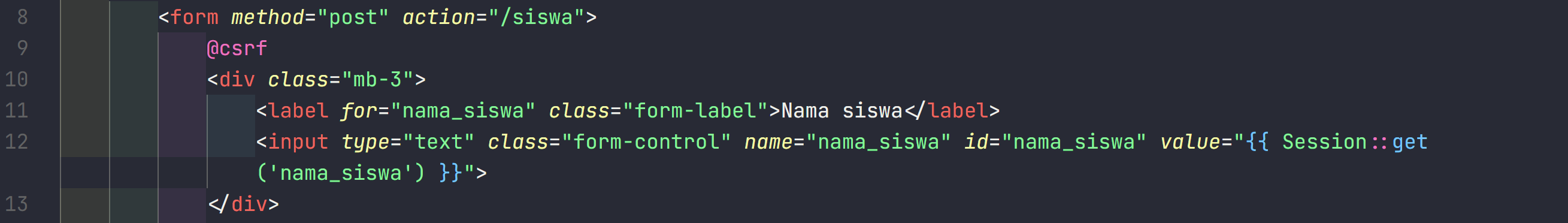


1. Agar input tetap ada walaupun muncul pesan error, kita perlu menambahkan **Session**. Pergi ke SiswaController, dan tambahkan Session Support Facades di method store().





1. Untuk menampilkannya, pergi ke view create, dan tambahkan Session-nya di masing-masing value.



1. Jika dijalankan, input tetap muncul meskipun sudah muncul pesan error.

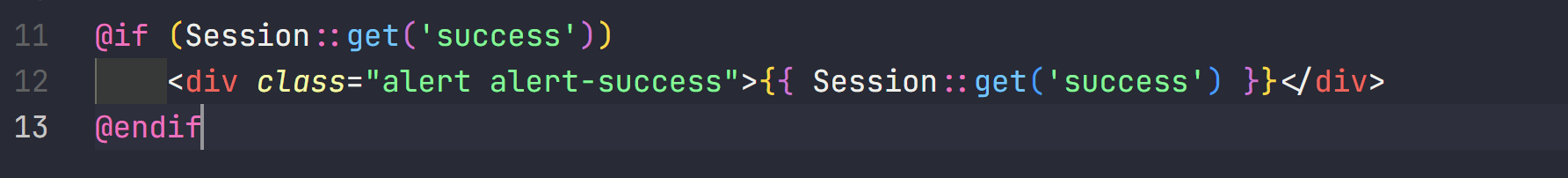


1. Masalah berikutnya muncul. Ketika semua data sudah diisi dan sudah berhasil menyimpan data ke database, data terbaru akan muncul di view siswa index. Tetapi, tidak ada pemberitahuan bahwa data telah berhasil disimpan.
2. Pergi ke SiswaController, dan tambahkan di bagian return pada method **store()**.



Akan mengirimkan variable **success** dengan value **‘Data berhasil ditambahkan.’**

1. Kemudian pergi ke view message, tambahkan untuk pesan suksesnya.



1. Jalankan aplikasi dan tambahkan data. Kemudian akan muncul pesan sukses di view siswa index.

