

Pertanyaan

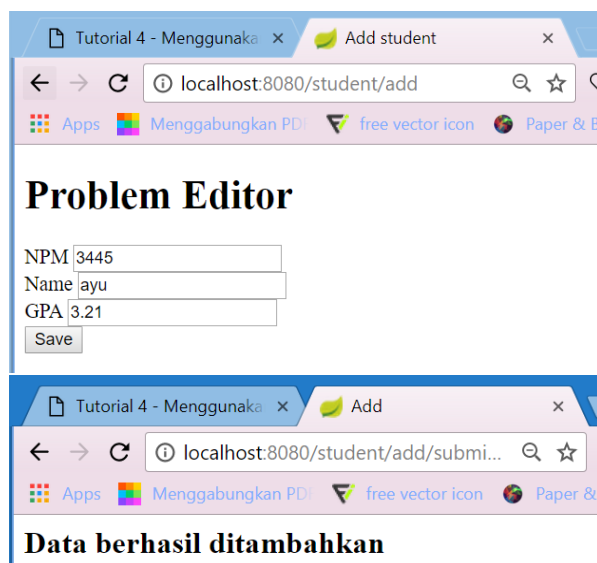
Beberapa pertanyaan yang perlu Anda jawab:

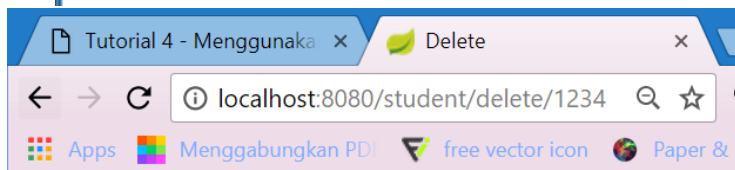
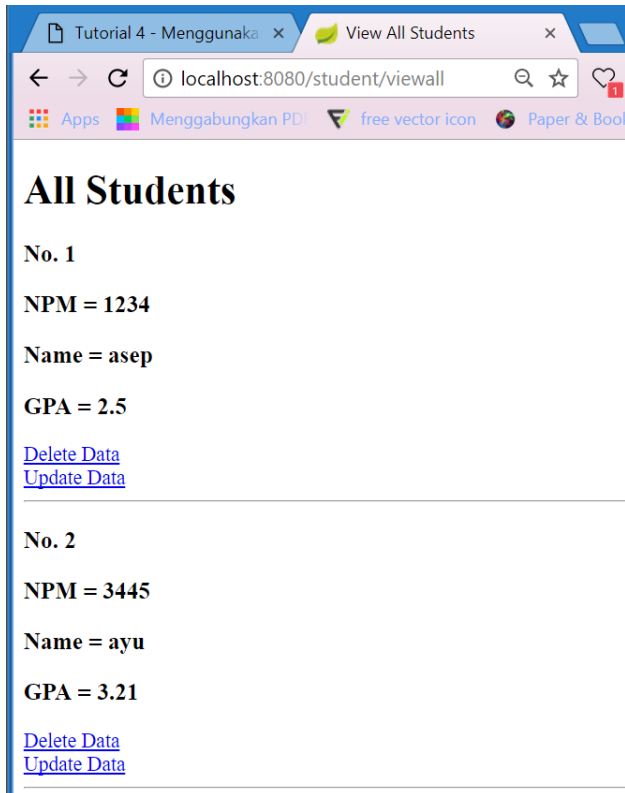
1. Jika menggunakan Object sebagai parameter pada form POST, bagaimana caranya melakukan validasi input yang optional dan input yang required? Apakah validasi diperlukan? Asumsikan input pada form Anda tidak menggunakan attribute required sehingga butuh validasi di backend.
Membuat validasi tersebut dengan `@Valid` dan `BindingRequest` untuk pengecekan dan tetap perlu melakukan validasi dengan kondisi (if-else) didalam methodnya.
2. Menurut Anda, mengapa form submit biasanya menggunakan POST method dibanding GET method? Apakah perlu penanganan berbeda di header atau body method di controller jika form di post dikirim menggunakan method berbeda?
Karena data yang didapatkan pada form submit akan dimasukan kedalam database dan GET method adalah untuk mengambil data yang ada pada database. Perlu karena jika tidak ditangani atau request methodnya berbeda maka tidak bisa di tangkap method pada controller.
3. Apakah mungkin satu method menerima lebih dari satu jenis request method, misalkan menerima GET sekaligus POST?
Mungkin bisa jadi jika di form POST atau GET dan pada controller ada GET dan POST maka pada method pada form akan merefer ke controller tersebut.

Write-up

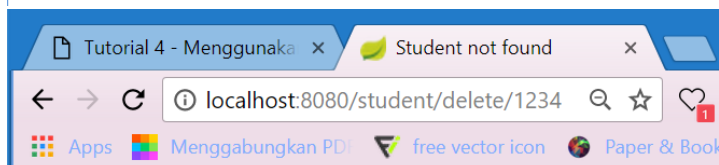
1. Latihan menambahkan delete

Pada method delete ini membuat method `deleteStudent` pada `StudentMapper` untuk menghapus data di database. Adanya `log.info` tersebut ini untuk melakukan debugging dengan database. Kemudian adanya method delete pada controller, adanya validasi untuk menampilkan `not-found.html` saat npm yang dihapus tidak ditemukan dan menampilkan `delete.html` saat data berhasil dihapus.





Data berhasil dihapus



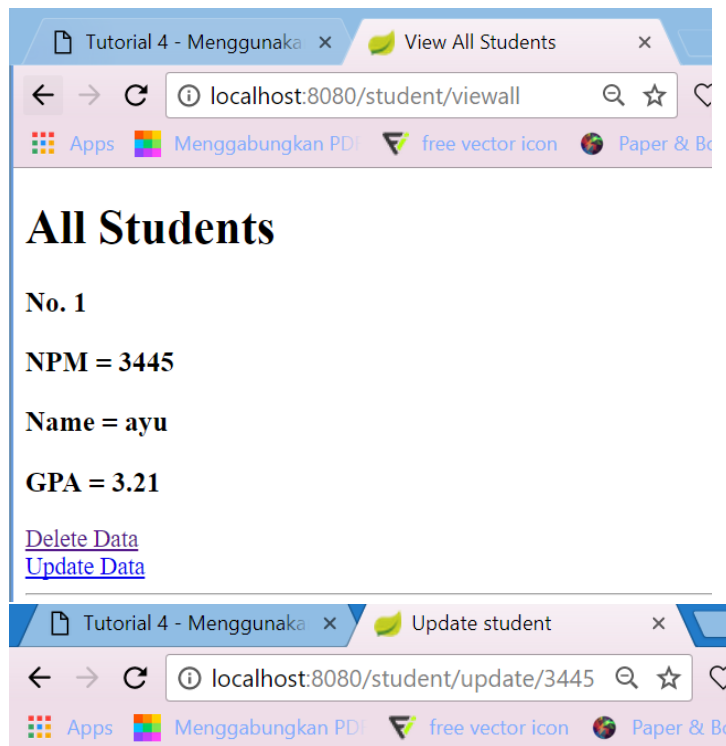
Student not found

NPM = 1234

2. Latihan menambahkan update

Pada method ini membuat updateStudent pada studentMapper untuk melakukan update pada database. Membuat method updateStudent pada studentService dan studentServiceDatabase untuk menjalankan updateStudent pada studentMapper. Kemudian membuat form-update untuk update data dan success-update.html yang akan terbuka saat berhasil di update. Pada controller, membuat validasi jika ditemukan akan membuka form-update dan jika tidak ditemukan maka membuka not-found.html.

Method updateSubmit pada controller ini berguna saat melakukan update dengan mengisi form-update maka method ini akan dijalankan untuk memasukan data tersebut kedalam database.

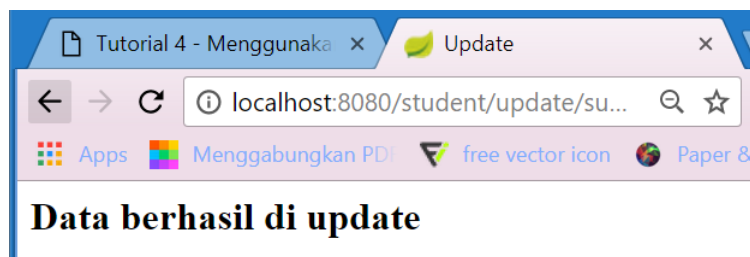


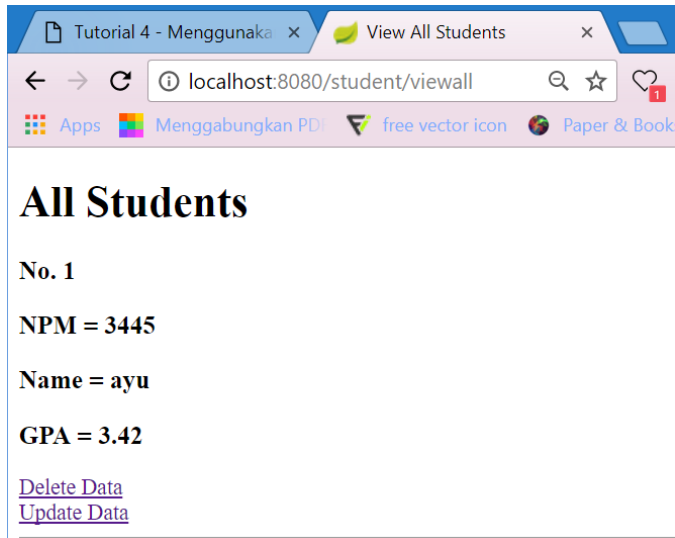
Problem Editor

NPM

Name

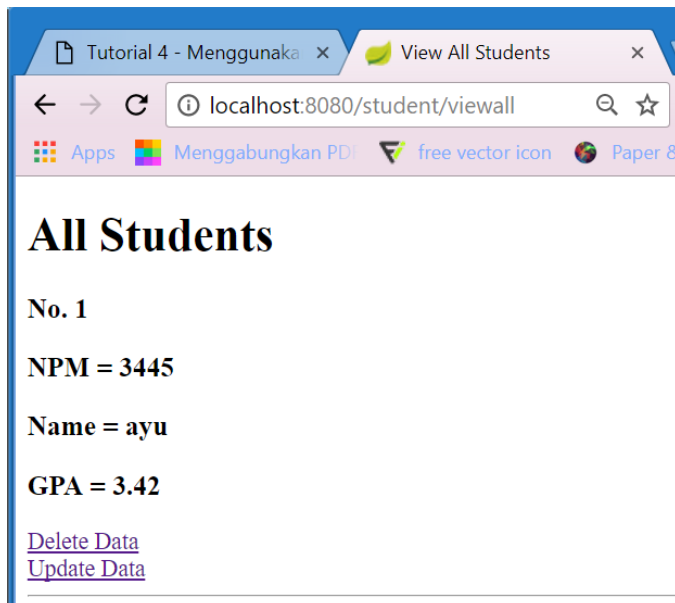
GPA





3. Latihan menggunakan object sebagai parameter

Pada method `updateSubmit` `@Param` diganti dengan sebuah objek `StudentModel` sehingga tidak sulit membuat parameter yang banyak dan panjang yang sebenarnya parameternya merupakan satu objek. Hal ini membuat objek sebagai parameter dengan membuat `@ModelAttribute` dan pada implementasinya `updateSubmit` tidak perlu lagi membuat objek student didalam methodnya.



Tutorial 4 - Menggunakan x

Update student x

localhost:8080/student/update/3445

Apps Menggabungkan PDF free vector icon Paper & Books

Problem Editor

NPM

3445

Name

Ayu Sari

GPA

3.42

Update

Tutorial 4 - Menggunakan x

Update x

localhost:8080/student/update/su...

Apps Menggabungkan PDF free vector icon Paper & Books

Data berhasil di update

Tutorial 4 - Menggunakan x

View All Students x

localhost:8080/student/viewall

Apps Menggabungkan PDF free vector icon Paper & Books

All Students

No. 1

NPM = 3445

Name = Ayu Sari

GPA = 3.42

[Delete Data](#)

[Update Data](#)