

Yang dipelajari pada tutorial ini

Saya mempelajari bahwa thymeleaf tidak memiliki else conditional yang menurut saya cukup merepotkan bagi programmer yang tentu saja telah terbiasa dengan adanya else conditional. Saya juga mempelajari bahwa kita dapat membuat model sendiri yang berisi object seperti StudentModel yang telah diinstruksikan pada tutorial ini. Dengan adanya service, function yang berhubungan antara Controller dan Model dapat dijembatani.

Latihan

Membuat Controller dan Fungsi Add

4. Jalankan program dan buka

localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43

Pertanyaan 1: apakah hasilnya? Jika error , tuliskan penjelasan Anda.



localhost:8080/student/add?npm=12345&name=chanek

Pertanyaan 2: apakah hasilnya? Jika error , tuliskan penjelasan Anda.



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Wed Sep 20 12:17:21 WIB 2017

There was an unexpected error (type=Bad Request, status=400).

Required double parameter 'gpa' is not present

Error terjadi dikarenakan parameter gpa tidak dimasukkan sedangkan gpa bersifat required.

Method View by NPM

3. Jalankan program dan buka

localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka

localhost:8080/student/view?npm=12345,

Pertanyaan 3: apakah data Student tersebut muncul? Jika tidak, mengapa?





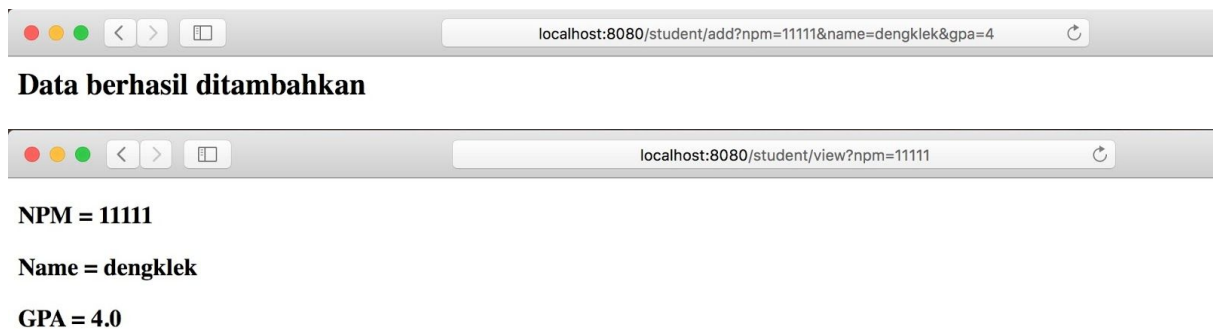
4. Coba matikan program dan jalankan kembali serta buka
localhost:8080/student/view?npm=12345

Pertanyaan 4: apakah data Student tersebut muncul? Jika tidak, mengapa?



Ketika program dimatikan, variabel yang menyimpan data sebelumnya dihapus.
Ketika program dijalankan kembali, method view langsung mengakses variabel
StudentModel yang masih kosong dan memunculkan error.

5. Coba tambahkan data Student lainnya dengan NPM yang berbeda.

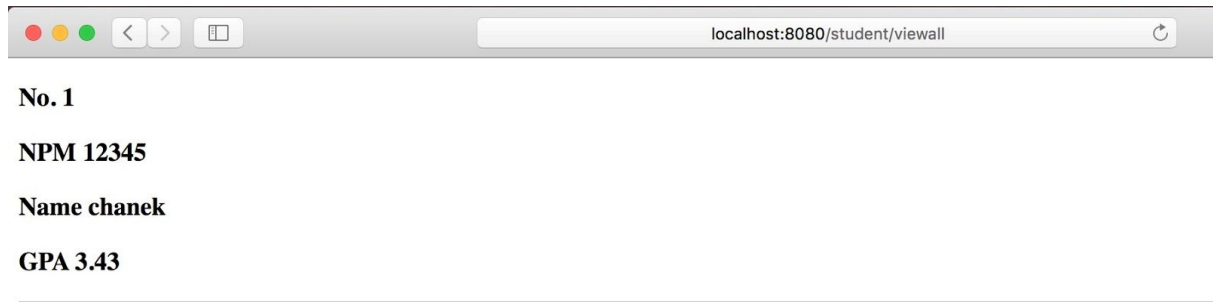


Method View All

3. Jalankan program dan buka
localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka
localhost:8080/student/viewall,

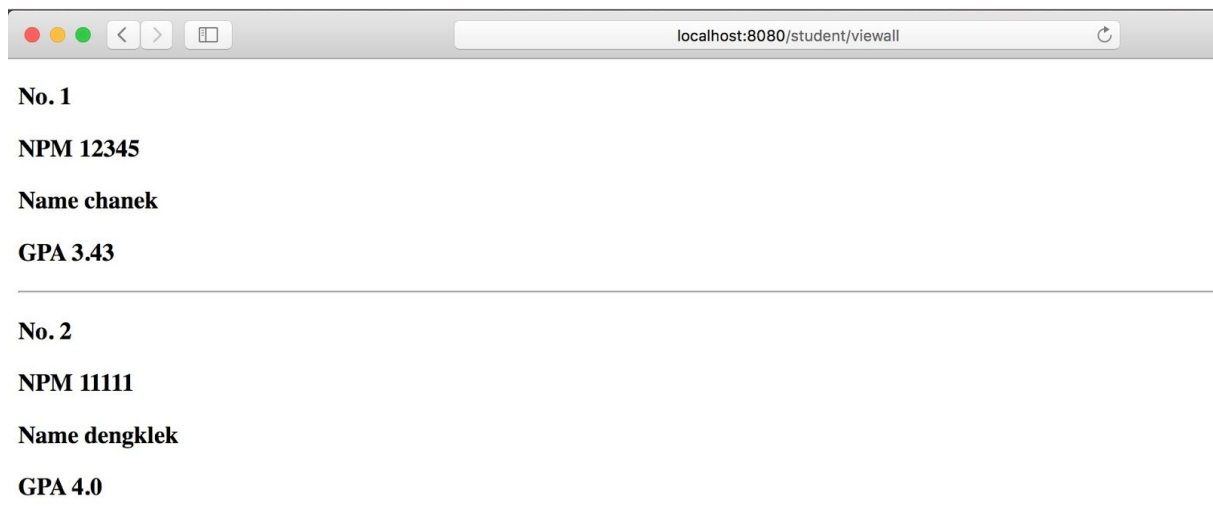
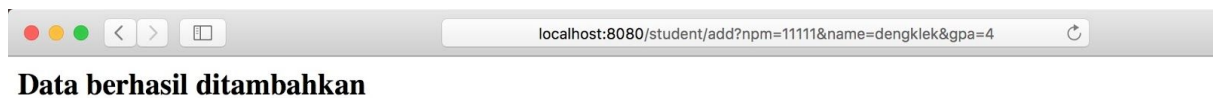
Pertanyaan 5: apakah data Student tersebut muncul?





4. Coba tambahkan data Student lainnya dengan NPM yang berbeda, lalu buka localhost:8080/student/viewall,

Pertanyaan 6: Apakah semua data Student muncul?



Method selectStudent

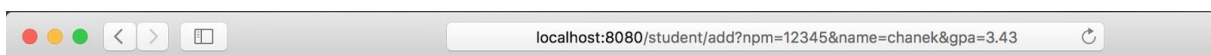
```
@Override
public StudentModel selectStudent(String npm) {
    //Implement
    for (StudentModel student: studentList) {
        if (student.getNpm().equals(npm)) {
            return student;
        }
    }
}
```

```
        return null;  
    }
```

Latihan

1. Pada StudentController tambahkan sebuah *method view Student* dengan menggunakan Path Variable . Misalnya, Anda ingin memasukkan data seorang Student yang memiliki NPM 14769, untuk melihat data yang baru dimasukkan tersebut dapat mengakses halaman `localhost:8080/student/view/14769`.

Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman *error* yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan.



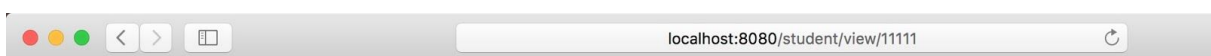
Data berhasil ditambahkan



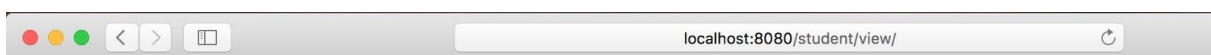
NPM = 12345

Name = chanek

GPA = 3.43



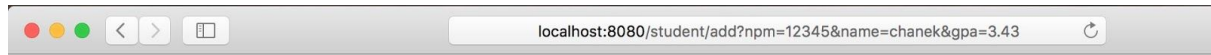
NPM tidak ditemukan



NPM kosong

2. Tambahkan fitur untuk melakukan *delete Student* berdasarkan NPM. Misalnya, setelah melakukan *add Student* pada soal nomor 1, cobalah untuk melakukan *delete* data tersebut dengan mengakses halaman `localhost:8080/student/delete/14769`. Tampilkan sebuah halaman yang memberikan informasi bahwa data tersebut telah berhasil dihapus.

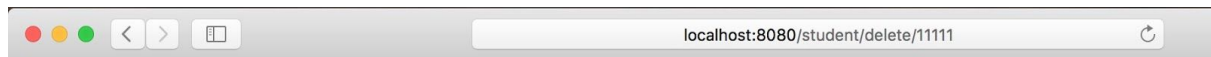
Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman *error* yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan dan proses *delete* dibatalkan.



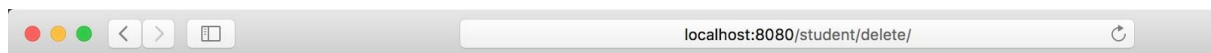
Data berhasil ditambahkan



NPM 12345 berhasil dihapus



NPM tidak ditemukan



NPM kosong

Penjelasan fitur *delete*

Pada awalnya, *spring* akan memeriksa apakah url memiliki path variable *npm*. Jika ada, maka service akan memeriksa apakah *npm* tersebut tersedia. Jika *npm* tersedia, maka service akan mencari dan menghapus data *StudentModel* yang dimasukkan dalam url dan kemudian model bernama *status* menyimpan string “dihapus” dan model bernama “*npm*” menyimpan *npm* yang dimasukkan. Jika *npm* tidak tersedia, maka model bernama *status* akan menyimpan string “tidak ditemukan”. Jika url tidak memiliki path variable *npm*, maka model bernama *status* akan menyimpan string “kosong”. Setelah itu, ketika membuka template file *delete.html*, akan dicek apakah isi dari model *status* dan mencetak sesuai dengan isinya.