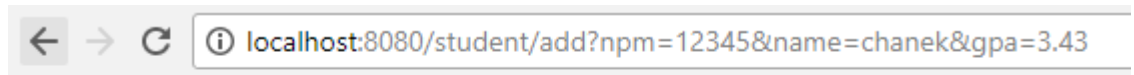


a. Ringkasan Materi

Pada tutorial ini saya mempelajari membuat model menggunakan konsep MVC serta membuat *service* dengan fungsi *create*, *read*, *delete* data menggunakan konsep MVC.

b. Jawaban dan screenshot

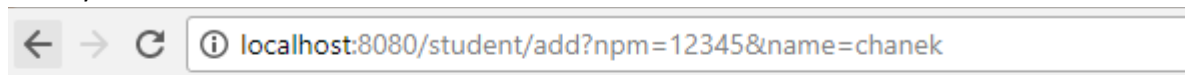
1. Hasilnya:



Data berhasil ditambahkan

Tidak terjadi *error* karena semua parameter diisi, sehingga data berhasil ditambahkan.

2. Hasilnya:



Whitelabel Error Page

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

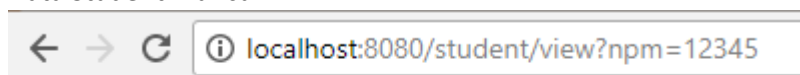
Fri Mar 02 17:20:06 ICT 2018

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

Required double parameter 'gpa' is not present

Terjadi *error* karena parameter *gpa* (*required*) tidak ada.

3. Data Student muncul

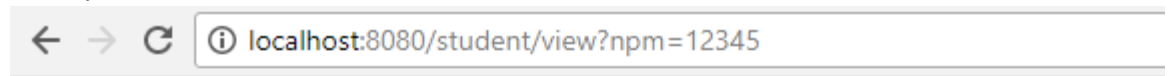


NPM = 12345

Name = chanek

GPA = 3.43

4. Hasilnya:



Whitelabel Error Page

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

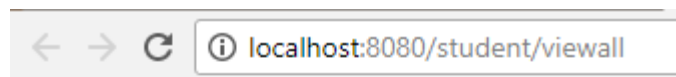
Sat Mar 03 09:20:37 ICT 2018

There was an unexpected error (type=Internal Server Error, status=500).

Exception evaluating SpringEL expression: "student.npm" (view:7)

Student yang ditambahkan tadi tidak muncul karena setelah program dimatikan data Student yang disimpan tadi terhapus, dan jika program dijalankan kembali data Student sudah kosong.

5. Student muncul



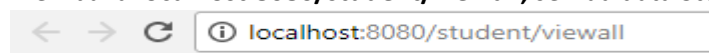
No. 1

NPM = 12345

Name = chanek

GPA = 3.43

6. Setelah menambahkan data Student yang lain dengan NPM yang berbeda, dan kemudian membuka **localhost:8080/student/viewall**, semua data Student muncul.



No. 1

NPM = 12345

Name = chanek

GPA = 3.43

No. 2

NPM = 1706106904

Name = Pance

GPA = 3.99

c. Method selectStudent

```
@Override
    public StudentModel selectStudent(String npm) {
        StudentModel stdSelected = null;
        for (StudentModel std : studentList) {
            if (std.getNpm().equals(npm)) {
                stdSelected = std;
                break;
            }
        }
        return stdSelected;
    }
```

d. Penjelasan fitur *delete*

Pertama saya membuat *method delete* pada *class StudentService*, kemudian method tersebut saya *override* di *class InMemoryStudentService* dan membuat pengimplementasian fungsi *delete*. Setelah itu saya membuat *method delete* dengan menggunakan *path variable* pada *class StudentController*. Contoh *url* untuk menjalankan fungsi *delete*: <http://localhost:8080/student/delete/1706106904>, jika npm tidak diisi (kosong) atau npm tidak ada dalam data Students maka akan menampilkan keterangan data tidak ditemukan. Contoh berikut tampilan jika npm tidak ditemukan dalam data Student



NPM tidak ditemukan

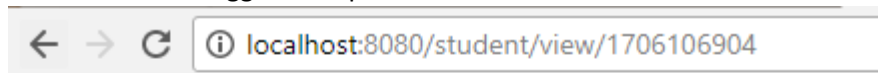
Jika npm ada dalam data Student:



NPM 1706106904 berhasil dihapus

Latihan

1. Method view menggunakan path variable



NPM = 1706106904

Name = Pance

GPA = 3.99

Pance Satria Naibaho
1706106904

2. Method delete menggunakan path variable



NPM 1706106904 berhasil dihapus