

Nama : Olvi Lora S

NPM : 1606954943

Pengimplementasian method select student

```
@Override
public StudentModel selectStudent(String npm) {
    // Implement
    for(int i=0;i<studentList.size();i++) {
        if(studentList.get(i).getNpm().equals(npm)) {
            return studentList.get(i);
        }
    }
    return null;
}
```

Pertanyaan 1

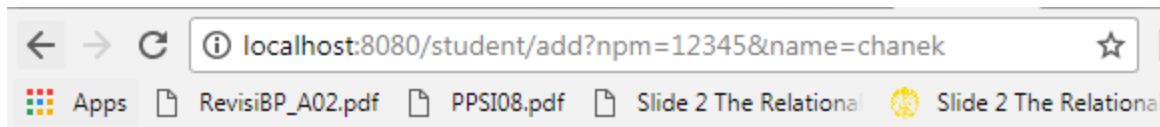
Mengakses <http://localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43>



Hasilnya tidak error, karena parameter yang dibutuhkan ada di url yang dipanggil yaitu npm, nama dan gpa. Proses tersebut menambahkan student dengan parameter yang diberikan ke List studentList.

Pertanyaan 2

Mengakses <http://localhost:8080/student/add?npm=12345&name=chanek>



Whitelabel Error Page

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

Fri Mar 02 23:44:52 ICT 2018

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

Required double parameter 'gpa' is not present

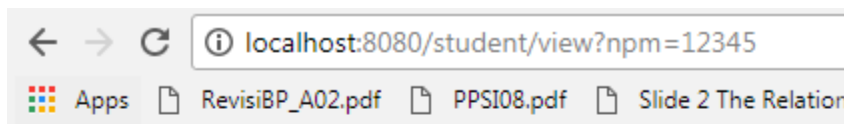
Hasilnya error, karena ada satu parameter yang tidak ada yaitu gpa. Di controller ketiga parameter di-set required.

Pertanyaan 3

Mengakses <http://localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43>



Kemudian view data student yang ditambahkan melalui url ini localhost:8080/student/view?npm=12345



NPM = 12345

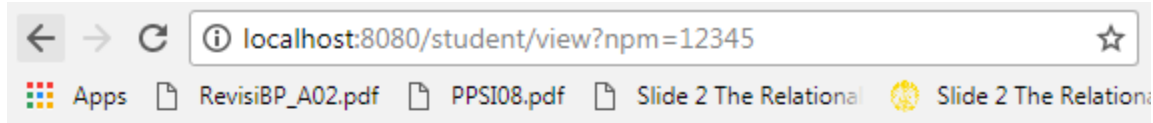
Name = chanek

GPA = 3.43

Ya, data Student muncul karena sebelumnya kita telah menambahkan student dengan npm 12345, name chanek dan gpa 3.45. Jadi ketika kita mengakses view dengan 12345 akan ditampilkan data student yang telah kita tambahkan.

Pertanyaan 4

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



Whitelabel Error Page

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

Sat Mar 03 00:17:25 ICT 2018

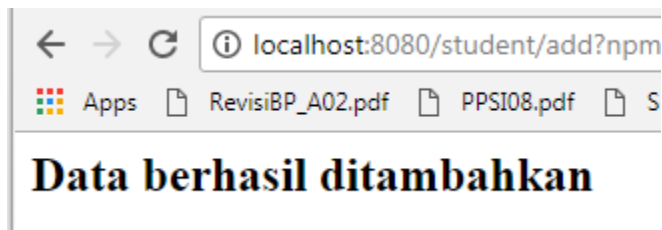
There was an unexpected error (type=Internal Server Error, status=500).

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

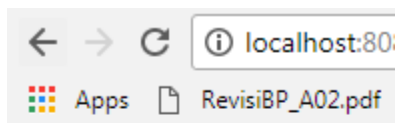
Data student tidak muncul. Karena pada tahap ini kita belum menambahkan data student. Jadi ketika mengakses url untuk view dengan npm 12345 akan menghasilkan error.

5. Menambahkan Data Student Lainnya

<http://localhost:8080/student/add?npm=1606954943&name=olvi&gpa=3.53>



Menambahkan data student dengan npm=1606954943, name=olvi dan gpa=3.53



NPM = 1606954943

Name = olvi

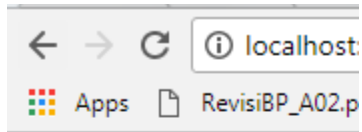
GPA = 3.53

Pertanyaan 5:

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

Kemudian menampilkan semua data student : localhost:8080/student/viewall

Ya, data Student yang ditambahkan muncul



No. 1

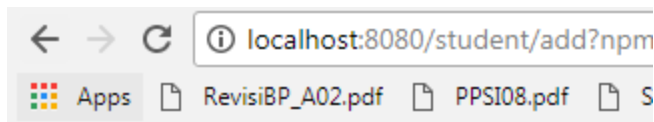
NPM = 12345

Name = chanek

GPA = 3.43

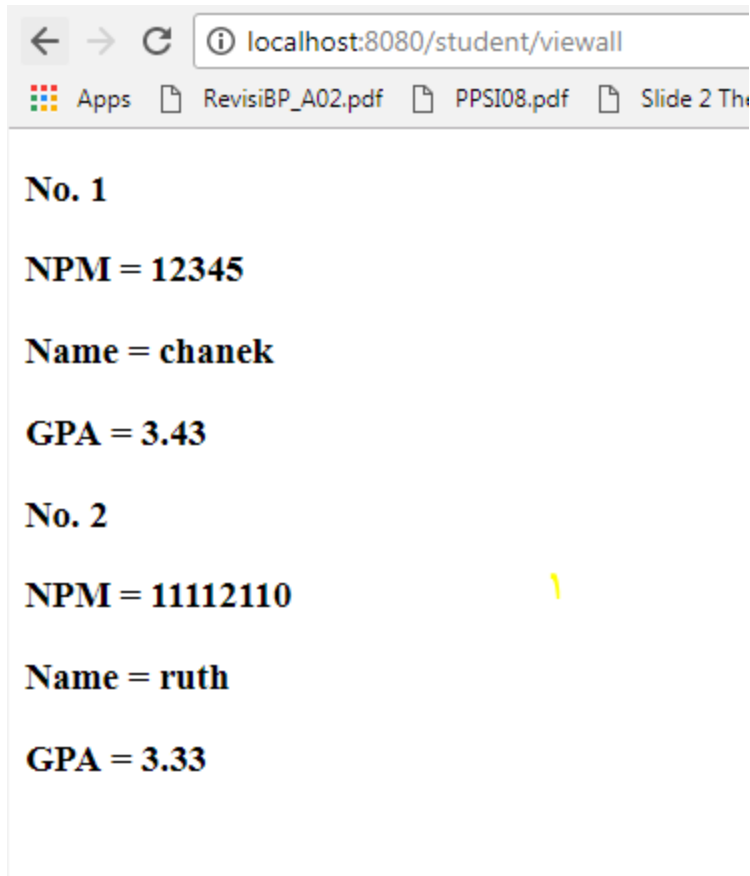
Pertanyaan 6

Menambahkan data student lain : localhost:8080/student/add?npm=11112110&name=ruth&gpa=3.33



Data berhasil ditambahkan

Menampilkan semua data student yang telah ditambahkan : localhost:8080/student/viewall



Latihan

1. View Student berdasarkan NPM

Langkah-langkah yang dilakukan :

- Menambahkan request mapping yang baru ➔ `/student/view/{npm}`, `student/view/`
- Kemudian mengecek npm dari parameter yang diberikan di url dengan memanggil method `selectStudent(npm)`
 - Jika npm yang diberikan kosong, maka akan menampilkan view `view_not_found.html` yang menampilkan notifikasi “Student dengan NPM xxxx tidak ditemukan”
 - Jika parameter npm yang diberikan tidak ada di list Student, maka akan menampilkan view `view_not_found.html` yang menampilkan notifikasi “Student dengan NPM xxxx tidak ditemukan”
 - Jika parameter npm yang diberikan ada di list student maka akan menampilkan view `delete.html` yang menampilkan data npm, name dan gpa

Berikut code yang ditambahkan :

- di file `StudentController.java`

```

@RequestMapping(value = {"/student/view/{npm}", "student/view/"})
public String viewByNpm(@PathVariable Optional<String> npm, Model model) {
    if(npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if(student!=null) {
            model.addAttribute("student", student);
        }else {
            model.addAttribute("npm", npm);
            return "view_not_found";
        }
    }else {
        model.addAttribute("npm", npm);
        return "view_not_found";
    }

    return "view";
}

```

- di file view.html

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <title> View Student by NPM </title>
5 </head>
6 <body>
7     <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
8     <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
9     <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
10 </body>
11 </html>

```

- di file view_not_found.html

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <title> Page Not Found </title>
5 </head>
6 <body>
7     <h3 th:text="'Student dengan NPM ' + ${npm} + ' tidak ditemukan'">Student NPM tidak ditemukan</h3>
8 </body>
9 </html>

```

2. Delete Student berdasarkan NPM

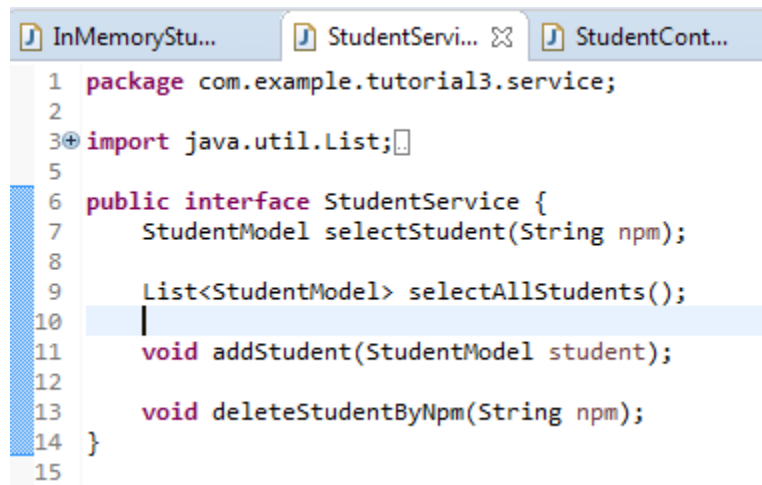
Langkah-langkah yang dilakukan :

- Menambahkan method deleteStudentByNpm(String npm) di interface StudentService.java
- Mengimplemtasikan method deleteStudentByNpm(String npm) di class InMemoryStudentService.java
- Menambahkan request mapping yang baru ➔ /student/delete/{npm}, student/delete

- Kemudian mengecek npm dari parameter yang diberikan di url dengan memanggil method selectStudent(npm)
 - Jika npm yang diberikan kosong, maka akan menampilkan view delete_failed.html yang memberikan notifikasi “Delete tidak berhasil, NPM null”
 - Jika npm yang diberikan tidak ada di list student, maka akan menampilkan view delete_failed.html yang memberikan notifikasi “Delete tidak berhasil, NPM tidak ditemukan”
 - Jika npm yang diberikan ada di list student, maka akan menampilkan view delete.html yang memberikan notifikasi “Student dengan npm xxx telah dihapus”

Berikut kode yang ditambahkan:

- di file StudentService.java

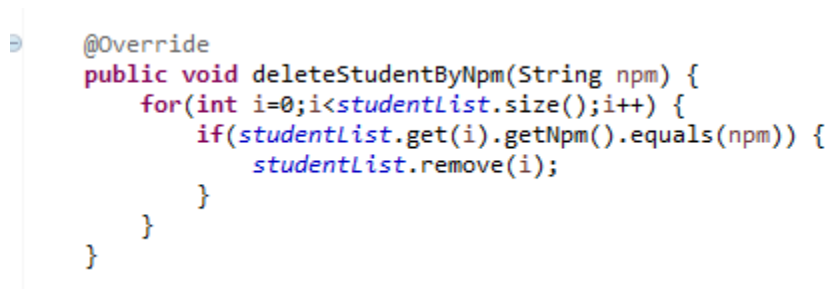


```

1 package com.example.tutorial3.service;
2
3 import java.util.List;
4
5
6 public interface StudentService {
7     StudentModel selectStudent(String npm);
8
9     List<StudentModel> selectAllStudents();
10
11     void addStudent(StudentModel student);
12
13     void deleteStudentByNpm(String npm);
14 }
15

```

- di file InMemoryStudentService.java



```

@Override
public void deleteStudentByNpm(String npm) {
    for(int i=0;i<studentList.size();i++) {
        if(studentList.get(i).getNpm().equals(npm)) {
            studentList.remove(i);
        }
    }
}

```

- di file StudentController.java

```

@RequestMapping(value = {"/student/delete/{npm}", "student/delete/"})
public String deleteByNpm(@PathVariable Optional<String> npm, Model model) {
    if(npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if(student!=null) {
            studentService.deleteStudentByNpm(npm.get());
            model.addAttribute("npm", npm);
            return "delete";
        }else {
            model.addAttribute("notification", "Delete tidak berhasil, NPM tidak ditemukan");
            return "delete_failed";
        }
    }else {
        model.addAttribute("notification", "Delete tidak berhasil, NPM null");
        return "delete_failed";
    }
}
}

```

- di file delete.html

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <title> Page Not Found </title>
5 </head>
6 <body>
7     <h3 th:text="'Student dengan npm ' + ${npm} + ' telah dihapus'">Student NPM tidak ditemukan</h3>
8 </body>
9 </html>

```

- di file delete_failed.html

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <title> Page Not Found </title>
5 </head>
6 <body>
7     <h3 th:text="${notification}"></h3>
8 </body>
9 </html>

```

Rangkuman

Pada tutorial 3 ini saya belajar menggunakan model and service.

1. Saya belajar membuat kelas Interface student yang berisi definisi method yang akan digunakan
2. Saya belajar mengimplementasikan semua method di controller yang telah didefinisikan di kelas Interface, yaitu method selectStudent dan deleteStudent

3. Saya juga lebih memahami tutorial minggu lalu tentang routing dan controller. Pada tutorial 3 ini juga saya membuat routing baru untuk delete dan view dan membuat kondisi dalam pengecekan parameternya.