

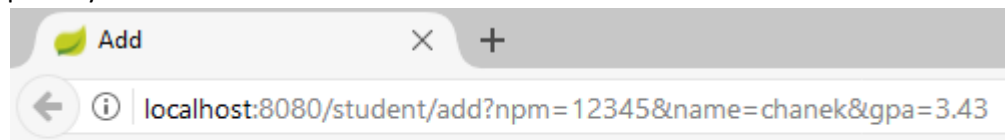
## a. Ringkasan yang baru dipelajari

Pada tutorial ini saya mempelajari kegunaan service, model dan inMemoryStudentService.java , yakni

- Kelas model merupakan representasi data dari objek pada program yang berisikan atribut-atribut objek tersebut
- Kelas service merupakan kumpulan method yang akan digunakan pada model yang diimplementasikan pada InMemoryStudentService.java dan dipanggil pada kelas controller
- Optional pada path variable sangat berguna untuk mencegah error yang dimasukkan oleh user, sehingga error handling method dapat efisien, tidak perlu membuat method pengecekan lagi

## b. Hasil jawaban dari setiap poin

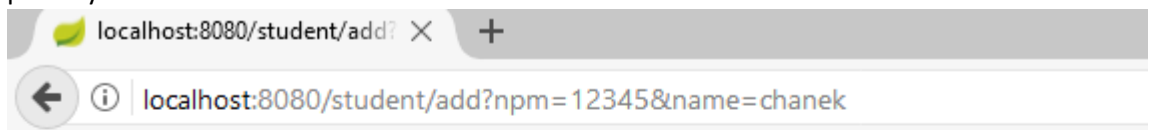
- pertanyaan 1



## Data berhasil ditambahkan

Tertulis data berhasil ditambahkan, hal ini dikarenakan pada saat localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 dipanggil, maka seluruh parameter yakni npm, name dan gpa akan dikirimkan masuk ke StudentController dan masuk ke request mapping yang sesuai, yakni ("/student/add"), kemudian method add akan dijalankan dan membuat objek student dengan atribut npm, name dan gpa. Saat student telah ditambahkan ke array list oleh method addStudent pada studentService, method add akan mengembalikan view html yang berjudul "add".

- pertanyaan 2



## Whitelabel Error Page

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

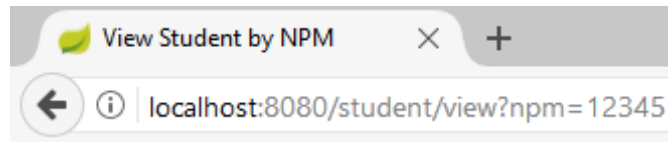
Fri Sep 22 20:25:23 ICT 2017

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

Required double parameter 'gpa' is not present

hal ini dikarenakan pada saat localhost:8080/student/add?npm=12345&name=chanek dipanggil, maka seluruh parameter yakni npm dan name akan dikirimkan masuk ke StudentController dan masuk ke request mapping yang sesuai, yakni ("/student/add"), pada saat method add akan dijalankan, terjadi error pada parameter gpa, hal ini dikarenakan parameter gpa bernilai false karena tidak diinput, sedangkan pada method add di controller, parameter memiliki 'syarat' *required = true*, sehingga terjadi error dengan informasi 'gpa' is not present

- pertanyaan 3



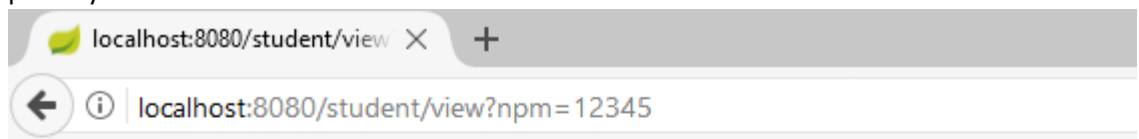
**NPM = 12345**

**Name = chanek**

**GPA = 3.43**

Muncul, karena localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 sudah menambahkan objek student ke arraylist, sehingga saat dipanggil localhost:8080/student/view?npm=12345 akan muncul data student dari arraylist

- pertanyaan 4



## Whitelabel Error Page

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

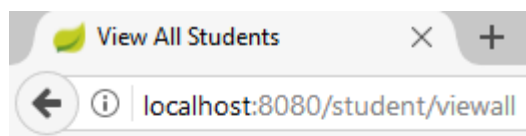
Fri Sep 22 22:43:19 ICT 2017

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

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

Terjadi error, hal ini dikarenakan data student dengan npm 12345 belum ditambahkan ke arraylist

- pertanyaan 5



**No. 1**

**NPM = 12345**

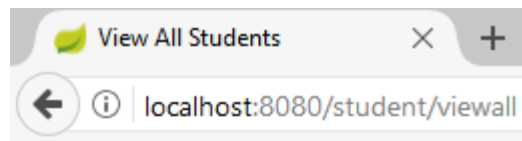
**Name = chanek**

**GPA = 3.43**

Muncul, hal ini dikarenakan data student ditambahkan terlebih dahulu ke arraylist, setelah ditambahkan, dipanggil method viewall dengan request mapping

“/student/viewall”, pada method tersebut controller mengembalikan *list of students* kepada view viewall.html, viewall.html melakukan looping data students satu persatu sehingga data muncul

- pertanyaan 6



**No. 1**

**NPM = 12345**

**Name = chanek**

**GPA = 3.43**

---

**No. 2**

**NPM = 12346**

**Name = chaneka**

**GPA = 3.44**

---

**No. 3**

**NPM = 12347**

**Name = chaneki**

**GPA = 3.45**

---

Muncul semua, penjelasan sama seperti jawaban pertanyaan 5.

- c. Method selectStudent yang Anda implementasikan

```

private static List<StudentModel> studentList = new ArrayList<StudentModel>();
@Override
public StudentModel selectStudent(String npm) {
    Iterator<StudentModel> iterasi = studentList.iterator();
    while(iterasi.hasNext()){
        StudentModel temp = iterasi.next();

        if(temp.getNpm().equals(npm)){
            return temp;
        }
    }
    return null;
}

```

Pada method selectStudent(), saya menggunakan kelas iterator dengan tipe data student model, untuk menemukan data student. Setelah controller menerima string npm yang dimasukkan dari URL, controller akan memanggil method selectStudent(npm.get()) pada kelas InMemoryStudentService.java dengan mengirimkan parameter npm yang dimasukkan. Method selectStudent akan dijalankan, diawali dengan dibuat objek iterasi yang berisikan arraylist student, kemudian arraylist tersebut dilooping, pengecekan dilakukan pada (iterasi.hasNext()), yang artinya pointer pada arraylist akan mengecek apakah data selanjutnya ada atau tidak, jika pencarian telah dilakukan hingga data terakhir yang berarti student dengan npm yang dimasukkan tidak ditemukan (dicek pada if(temp.getNpm().equals(npm)) bernilai false), maka method akan mengembalikan nilai null dan method viewNPM() pada kelas controller akan mengembalikan view ke error.html dengan message bahwa NPM tidak ditemukan. Namun, apabila npm ditemukan dimana saat (if(temp.getNpm().equals(npm))) bernilai true, maka method select Student akan mengembalikan data student dengan npm yang dicari, kemudian method viewNPM pada kelas controller akan mengembalikan view ke view.html dengan menampilkan data student yang dicari user.

d. Penjelasan fitur delete pada bagian latihan

```

@RequestMapping(value={"/delete","/delete/{npm}"})
public String delete(Model model, @PathVariable Optional<String> npm){
    if(!npm.isPresent()){
        model.addAttribute("message", "NPM kosong! Proses delete dibatalkan");
        return "error";
    }
    StudentModel student = studentService.selectStudent(npm.get());
    if(student == null){
        model.addAttribute("message", "NPM tidak ditemukan! Proses delete dibatalkan");
        return "error";
    }
    studentService.deleteStudent(student);
    return "delete";
}

public interface StudentService {
    StudentModel selectStudent(String npm);
    void deleteStudent(StudentModel student);
}

@Override
public void deleteStudent(StudentModel student) {
    // TODO Auto-generated method stub
    studentList.remove(student);
}

```

Pada method delete di kelas controller, saya mencari terlebih dahulu npm pada URL dimasukkan atau tidak (dicek pada !npm.isPresent()), apabila hasil pengecekan bernilai false maka akan dikirim view ke error.html dengan message "NPM kosong! Proses delete

dibatalkan”, namun apabila npm dimasukkan ada maka akan dicari terlebih dahulu student yang akan didelete dengan menggunakan method `selectStudent(npm.get())`, apa bila student dengan npm yang dimasukkan tidak ditemukan, maka student akan bernilai null, sehingga method akan mengirim view ke `error.html` dengan message “NPM tidak ditemukan! Proses delete dibatalkan”. Namun apabila student ditemukan yang artinya tidak null, maka akan dipanggil method `deleteStudent()` dengan parameter objek student pada kelas `InMemoryStudentService.java`, dimana sebelumnya method tersebut telah dideklarasikan terlebih dahulu pada kelas `StudentService`, kemudian pada kelas `InMemoryStudentService.java` method `deleteStudent` dengan parameter objek student, kemudian dilakukan penghapusan objek student tersebut pada arraylist