

I. Ringkasan materi yang telah dipelajari

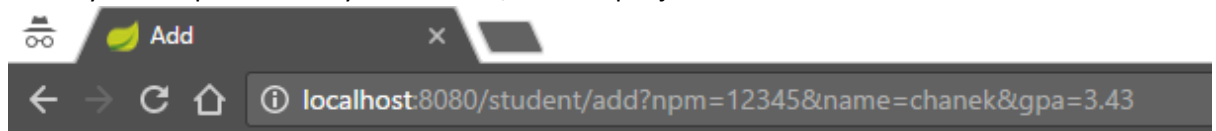
Pada tutorial sebelumnya telah dipelajari tentang konsep MVC. Tutorial kali ini fokus dalam menggunakan MVC untuk read dan create data. Pendekatan terbaik adalah memisahkan kode untuk melakukan akses sumber data ke dalam lapisan tersendiri, lapisan ini biasanya disebut sebagai service. Dengan service, kode akan menjadi lebih rapi dan terstruktur. Manfaat paling terasa adalah kemudahan pencarian kesalahan dan penambahan modul-modul baru tidak harus merombak seluruh struktur program.

II. Hasil jawaban setiap poin

Membuat Controller dan Fungsi Add

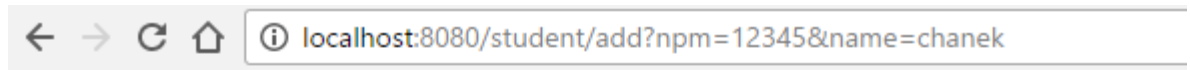
4. Jalankan program dan buka

- localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43
Pertanyaan 1: apakah hasilnya? Jika error, tuliskan penjelasan Anda.



Data berhasil ditambahkan

- 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.

Thu Sep 21 12:48:03 ICT 2017

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

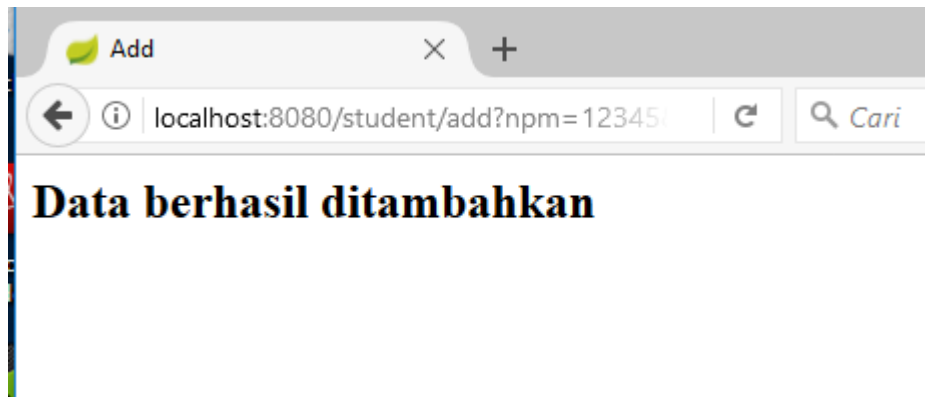
Required double parameter 'gpa' is not present

Error karena untuk method addStudent parameternya 3 buah yaitu npm, nama, dan gpa.

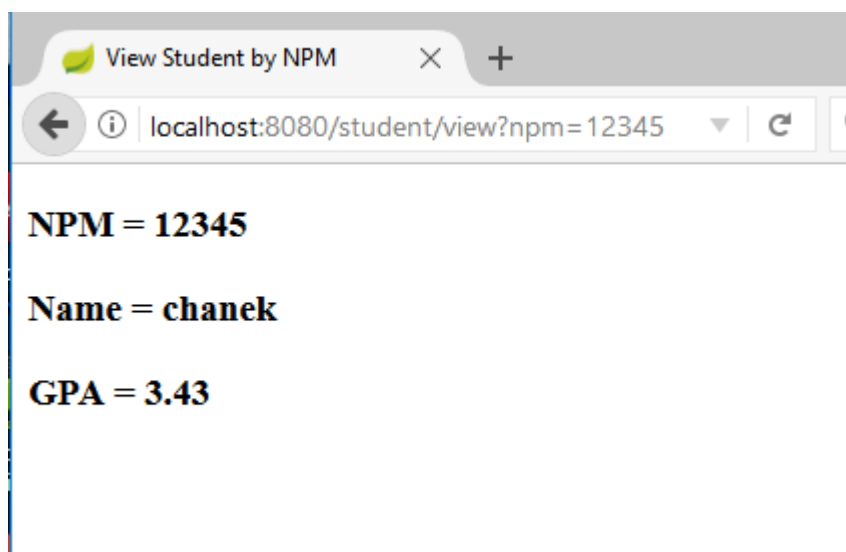
Method View by NPM

3. Jalankan program dan buka

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



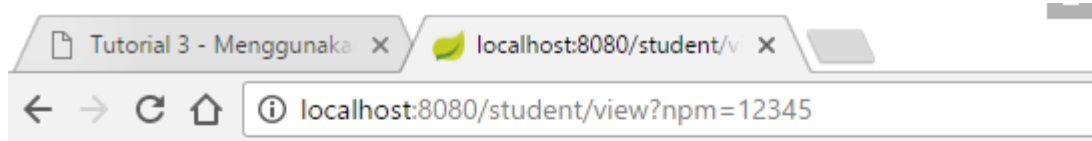
lalu buka localhost:8080/student/view?npm=12345



- Pertanyaan 3: apakah data Student tersebut muncul? Jika tidak, mengapa? Muncul. Data berhasil dimasukkan

4. Coba matikan program dan jalankan kembali serta buka

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

Thu Sep 21 13:01:35 ICT 2017

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

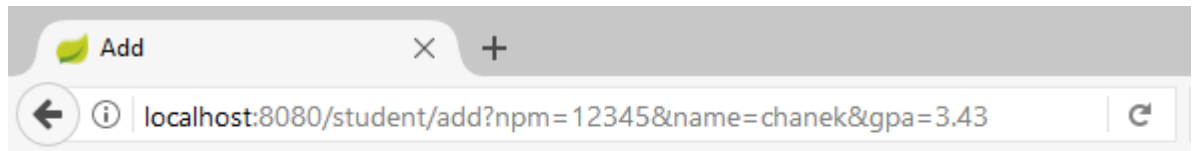
Exception evaluating SpringEL expression: "student.npm" (view:7)

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

Error karena program berjalan di RAM, sehingga ketika di relaunch semua data akan kembali ke awal. Tidak ada student dengan npm 12345 di awal program.

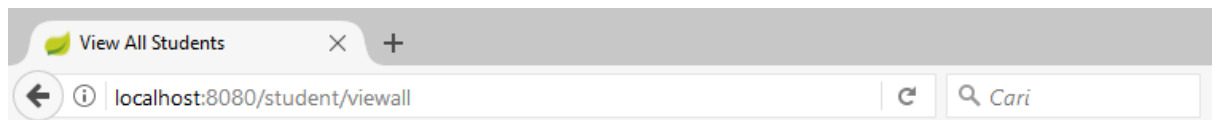
Method View All**3. Jalankan program dan buka**

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



Data berhasil ditambahkan

lalu buka localhost:8080/student/viewall,



No. 1

NPM = 12345

Name = chanek

GPA = 3.43

- Pertanyaan 5: apakah data Student tersebut muncul? Muncul karena data berhasil dimasukkan dan semua student yang ada di list ditampilkan.

4. Coba tambahkan data Student lainnya dengan NPM yang berbeda,

Data berhasil ditambahkan

lalu buka localhost:8080/student/viewall,



No. 1

NPM = 12345

Name = chanek

GPA = 3.43

No. 2

NPM = 10000

Name = spencer

GPA = 3.0

- Pertanyaan 6: Apakah semua data Student muncul?
Muncul semua data student yang ada di list.

Latihan 1

1. Pada StudentController tambahkan sebuah method view Student dengan menggunakan Path Variable.

```
,
@RequestMapping("/student/view/{npm}")
public String view(@PathVariable String npm, Model model){
    StudentModel student = studentService.selectStudent(npm);
    if(student == null){
        String result = "Nomor NPM tidak ditemukan";
        model.addAttribute("result", result);
        return "error";
    } else{
        model.addAttribute("student", student);
        return "view";
    }
}

@RequestMapping("/student/view/")
public String view(Model model){
    String result = "Nomor NPM kosong";
    model.addAttribute("result", result);
    return "error";
}
```

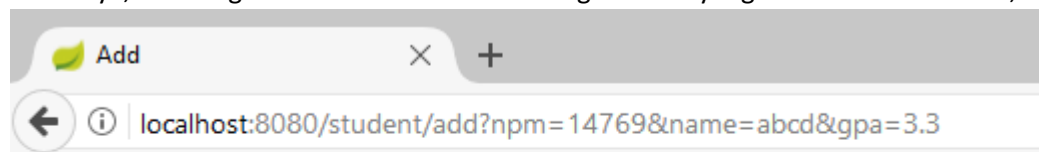
- Request mapping /student/view/npm: untuk mencari mahasiswa dengan npm yang diberikan dan npm yang tidak ditemukan.

Pertama akan dibuat studentmodel bernama student yang akan diisi dengan hasil selectstudent. Jika hasilnya null berarti tidak ada student dengan npm tersebut. Maka akan masuk if pertama. Ada string result yang berisi npm tidak ditemukan yang akan dimasukkan ke model result agar dapat diakses di html error. Sehingga page error.html dapat menampilkan error karena nomor npm tidak ditemukan. Jika hasilnya tidak null, maka akan di direct kepage view yang akan menampilkan data student tersebut.

- Request mapping /student/view/: untuk yang tidak diisi npmnya.

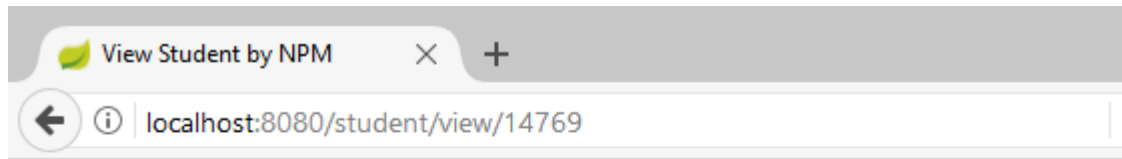
Jika di browser tidak memasukkan npm, maka akan diakses method view ini.String resultnya akan berisi npm kosong dan dimasukkan ke model result. Lalu akan di direct ke error.html yang akan menampilkan string result tersebut.

Misalnya, Anda ingin memasukkan data seorang Student yang memiliki NPM 14769,



Data berhasil ditambahkan

untuk melihat data yang baru dimasukkan tersebut dapat mengakses halaman localhost:8080/student/view/14769.

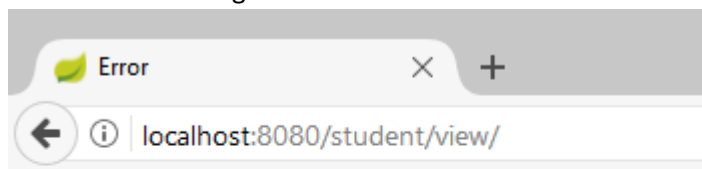


NPM = 14769

Name = abcd

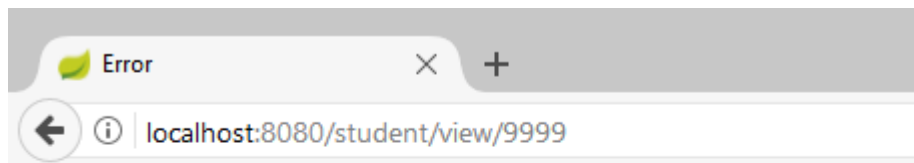
GPA = 3.3

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



Nomor NPM kosong

Jika nomor NPM tidak ditemukan kembalikan halaman error yang berisi informasi bahwa tidak ditemukan.



Nomor NPM tidak ditemukan

2. Tambahkan fitur untuk melakukan delete Student berdasarkan NPM. Misalnya, setelah melakukan add Student pada soal nomor 1,

```
@RequestMapping("/student/delete/{npm}")
public String delete(@PathVariable String npm, Model model){

    StudentModel student = studentService.selectStudent(npm);
    if(student == null){
        String result = "Nomor NPM tidak ditemukan";
        model.addAttribute("result", result);
        return "error";
    }
    else{
        studentService.deleteStudent(npm);
        model.addAttribute("student", student);
        return "delete";
    }
}

@RequestMapping(value = {"student/delete", "/student/delete/"})
public String delete(Model model){
    String result = "Nomor NPM kosong";
    model.addAttribute("result", result);
    return "error";
}
```

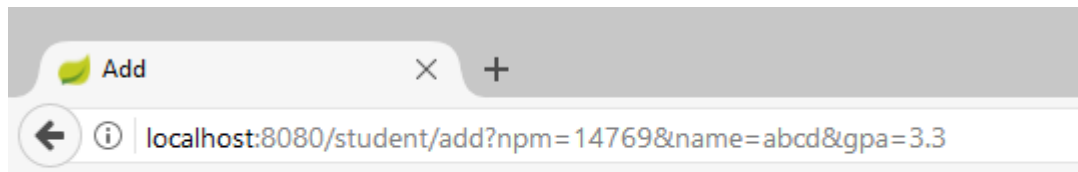
- Request mapping /student/delete/npm: untuk menghapus mahasiswa dengan npm yang diberikan atau menampilkan error jika npm tidak ditemukan.

Pertama akan dibuat studentmodel bernama student yang akan diisi dengan hasil selectstudent. Jika hasilnya null berarti tidak ada student dengan npm tersebut. Maka akan masuk if pertama. Ada string result yang berisi npm tidak ditemukan yang akan dimasukkan ke model result agar dapat diakses di html error. Sehingga page error.html dapat menampilkan error karena nomor npm tidak ditemukan dan student tersebut tidak dihapus. Jika hasilnya tidak null, maka akan masuk ke else dan dihapus. Lalu akan di direct ke page delete yang menampilkan bahwa telah berhasil dihapus.

- Request mapping /student/delete/: untuk yang tidak diisi npmnya.

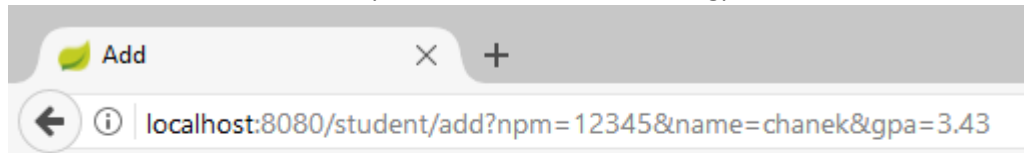
Jika di browser tidak memasukkan npm, maka akan diakses method view ini. String resultnya akan berisi npm kosong dan dimasukkan ke model result. Lalu akan didirect ke error.html yang akan menampilkan string result tersebut. Tidak ada data student yang dihapus.

localhost:8080/student/add?npm=14769&name=abcd&gpa=3.3



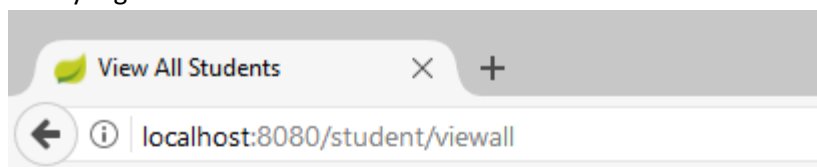
Data berhasil ditambahkan

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



Data berhasil ditambahkan

Data yang ada saat ini



No. 1

NPM = 14769

Name = abcd

GPA = 3.3

No. 2

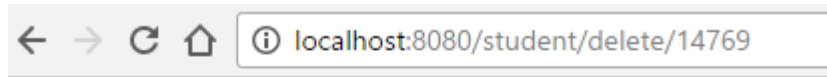
NPM = 12345

Name = chanek

GPA = 3.43

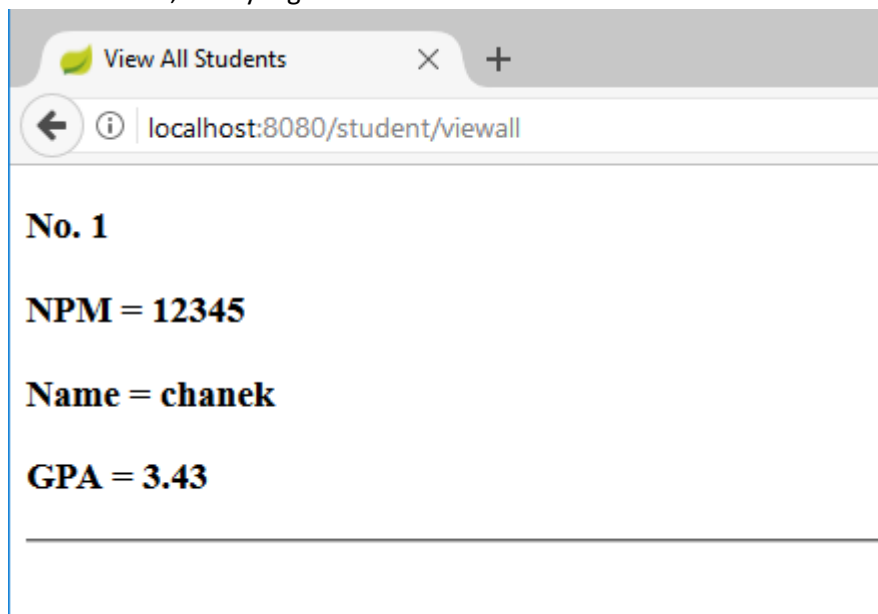
cobalah untuk melakukan delete data tersebut dengan mengakses halaman localhost:8080/student/delete/14769.

Tampilkan sebuah halaman yang memberikan informasi bahwa data tersebut telah berhasil dihapus.



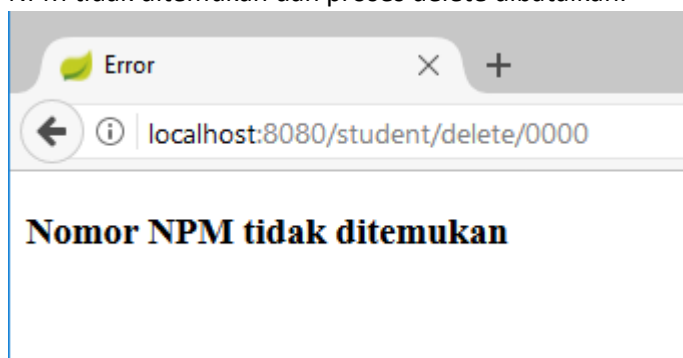
Mahasiswa dengan NPM = 14769 berhasil dihapus

Jika di viewall, data yang ada saat ini

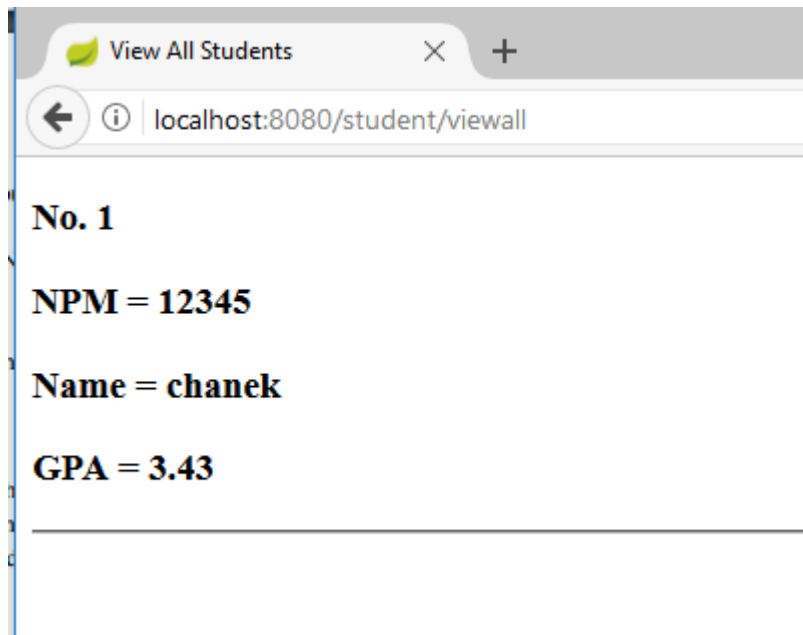


Data dengan NPM 14769 sudah dihapus.

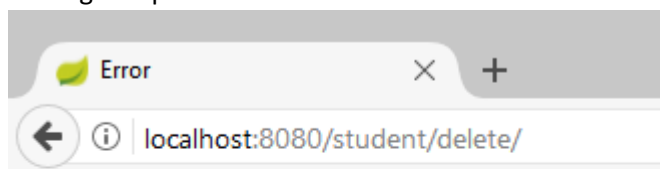
Jika nomor NPM tidak ditemukan kembalikan halaman error yang berisi informasi bahwa NPM tidak ditemukan dan proses delete dibatalkan.



Di view all tidak ada data yang terhapus

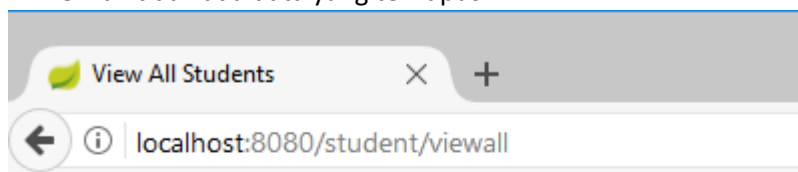


Jika nomor NPM tidak kembalikan halaman error yang berisi informasi bahwa nomor NPM kosong dan proses delete dibatalkan.



Nomor NPM kosong

Di view all tidak ada data yang terhapus



III. Method selectStudent

```
private static List <StudentModel> studentList = new ArrayList<StudentModel>();
@Override
public StudentModel selectStudent(String npm){
    for (int i=0 ; i<studentList.size();i++){
        if(studentList.get(i).getNpm().equals(npm)){
            return studentList.get(i);
        }
    }
    return null;
}
```

Untuk mendapatkan student dengan npm yang diberikan harus dilakukan pencarian di studentList yang merupakan kumpulan student yang ada. Caranya adalah dengan iterasi satu satu setiap student yang ada di list, jika terdapat npm yang sama maka student itu akan langsung direturn. Jika tidak ditemukan student dengan npm tersebut, maka direturn null.

IV. Penjelasan fitur delete

```
public void deleteStudent(String npm){
    List <StudentModel> temp = new ArrayList<StudentModel>();
    for (int i=0 ; i<studentList.size();i++){
        if(!studentList.get(i).getNpm().equals(npm)){
            temp.add(studentList.get(i));
        }
    }
    studentList = temp;
}
```

List student yang dipakai di program bernama studentList

```
private static List <StudentModel> studentList = new ArrayList<StudentModel>();
```

Pertama-tama akan dibuat List temporary sebagai wadah untuk menyimpan student-student. Lalu akan dilakukan iterasi terhadap studentList yang berisi student yang ada saat ini. Karena yang ingin dihapus adalah student dengan npm yang diberikan, maka pada saat looping jika student tsb npm nya tidak sama dengan npm yang diberikan student tsb akan dimasukan ke list temporary. Setelah looping selesai, kita masukkan list temporary ke studentList. Sehingga studentList saat ini tidak mengandung student dengan npm yang diberikan.