

Arsitektur dan Pemrograman Aplikasi Perusahaan

Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

Andre Gema Syahputra
1506689282
Kelas B

Ringkasan

Tutorial kali ini membahas MVC (Model-View-Controller) secara lebih dalam dan lebih lanjut. Tutorial 3 ini membahas dan memperkenalkan tentang *model* dan *service* yang merupakan bagian dari MVC. Secara kasarnya, model adalah sebuah *object* yang menyimpan data dan informasi, sedangkan service menyimpan *business logic* dan mengolah data dan informasi.

Penggunaan dan implementasi *service* dan *model* diajarkan pada tutorial ini dengan membuat sebuah sistem yang dapat menyimpan, menambahkan, melihat, dan menghapus data suatu Student.

Dengan mengikuti tutorial ini, saya berhasil memahami hubungan komponen-komponen MVC secara lebih dalam, serta memahami cara mengimplementasikan dan menggunakannya langsung.

Method selectStudent

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

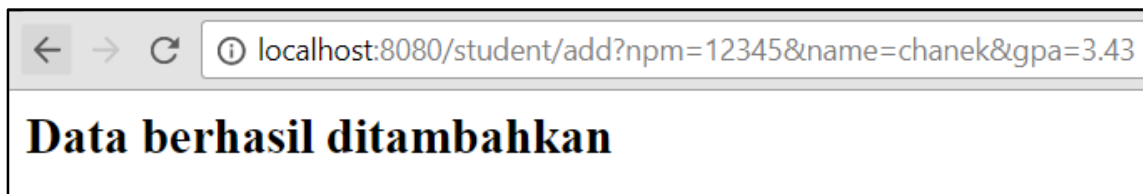
Method ini menerima parameter string npm. Kemudian *method* ini melakukan iterasi terhadap List studentList dan membandingkan npm tiap-tiap *object* di dalamnya dengan npm pada parameter. Ketika ada *object* yang memiliki npm sama dengan npm parameter, akan masuk ke kondisi if dan langsung me-return *object* tersebut. Apabila sampai habis tidak ditemukan *object* dengan npm yang sama, maka *loop* akan selesai dan *method* me-return null.

Arsitektur dan Pemrograman Aplikasi Perusahaan
Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

Andre Gema Syahputra
1506689282
Kelas B

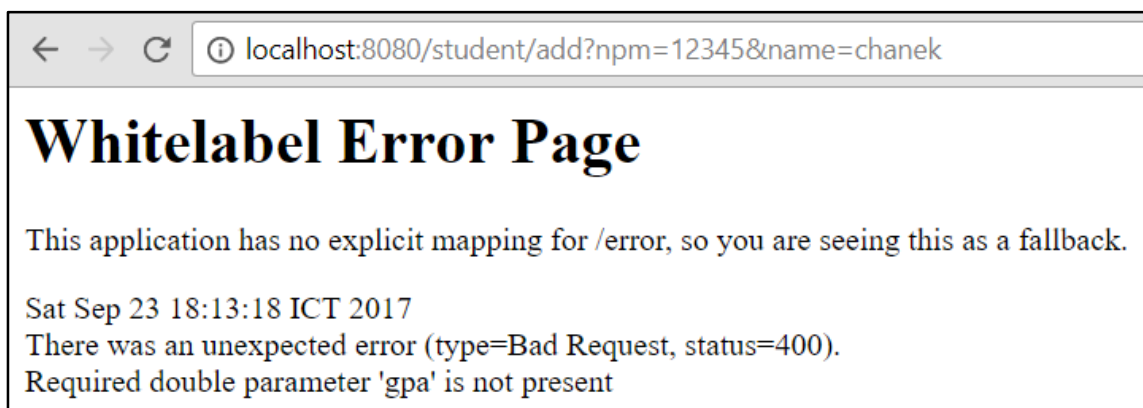
Pertanyaan Tutorial

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



Hasilnya tidak ada error, data berhasil ditambahkan. Sehingga menampilkan template add.html.

2. Jalankan program dan buka
localhost:8080/student/add?npm=12345&name=chanek
Pertanyaan 2: apakah hasilnya? Jika error , tuliskan penjelasan Anda.

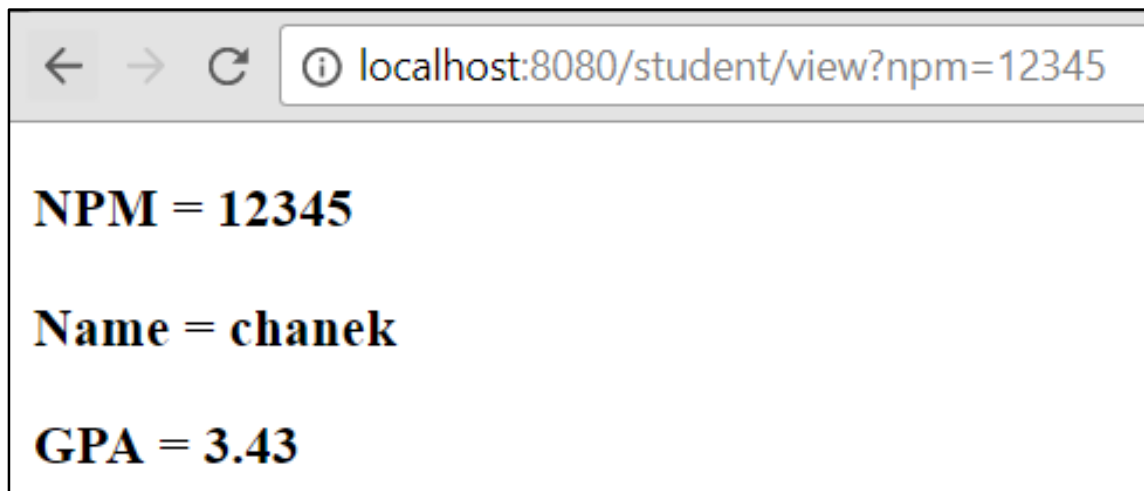


Terjadi error, sehingga sebuah Whitelabel Error Page ditampilkan. Error ini terjadi karena tidak ada value gpa yang bersifat required.

Arsitektur dan Pemrograman Aplikasi Perusahaan
Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

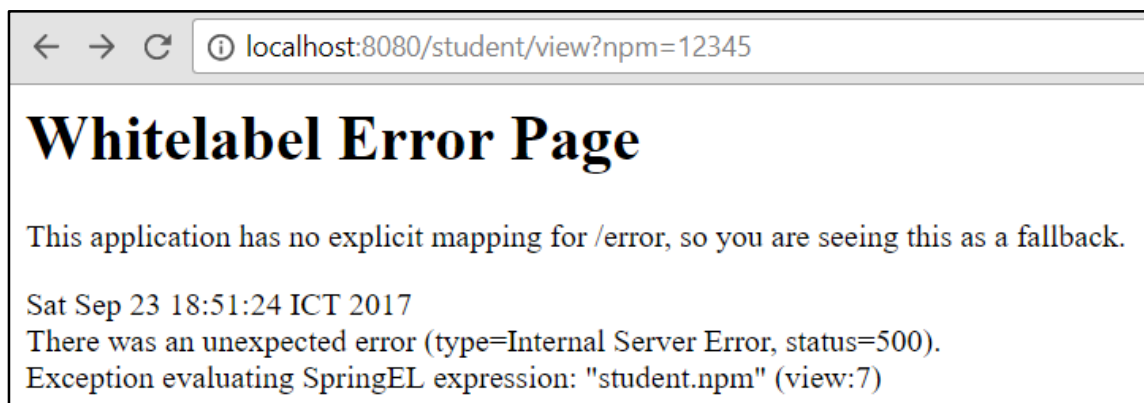
Andre Gema Syahputra
1506689282
Kelas B

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?



Data Student yang dimasukan sebelumnya berhasil ditampilkan

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?

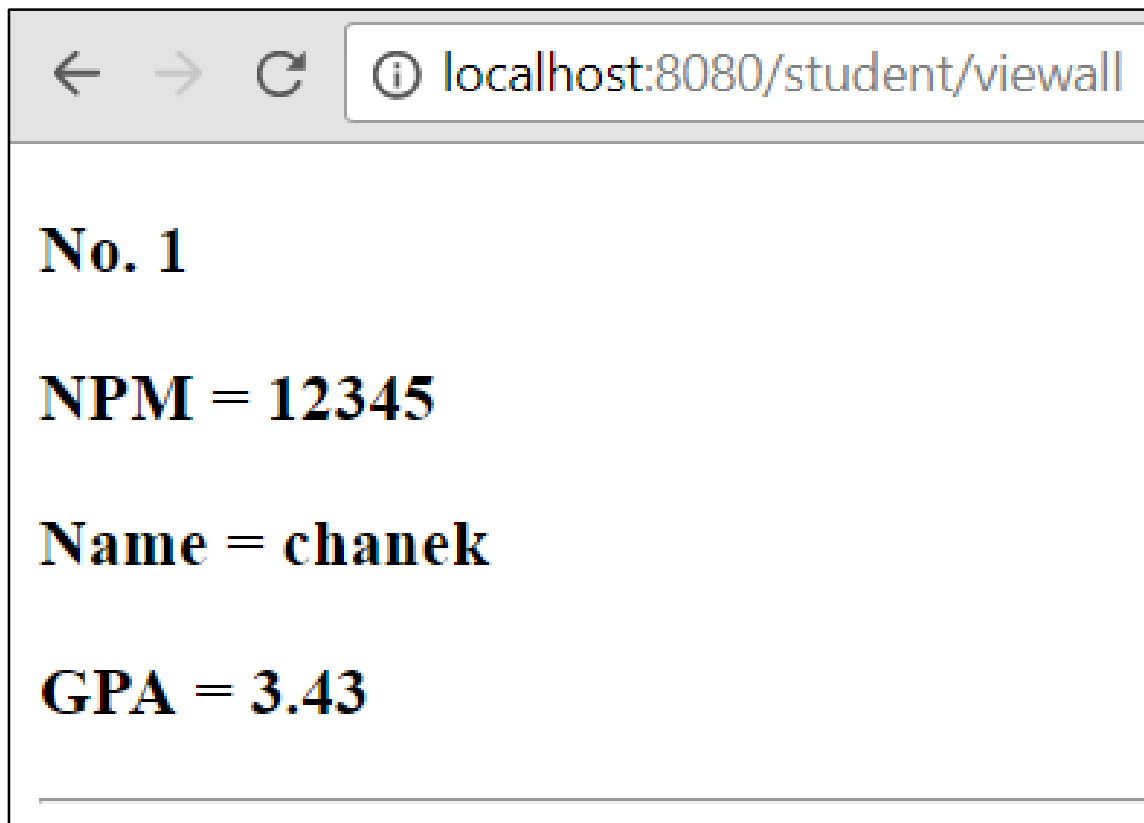


Tidak muncul, terjadi error. Hal ini terjadi dikarenakan karena tidak ada Student dengan npm=12345 karena program baru saja di-restart, maka *object* yang di *return* pada *method* selectStudent adalah null, sehingga terjadi exception.

Arsitektur dan Pemrograman Aplikasi Perusahaan
Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

Andre Gema Syahputra
1506689282
Kelas B

5. 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?



Data Student berhasil ditampilkan

Arsitektur dan Pemrograman Aplikasi Perusahaan

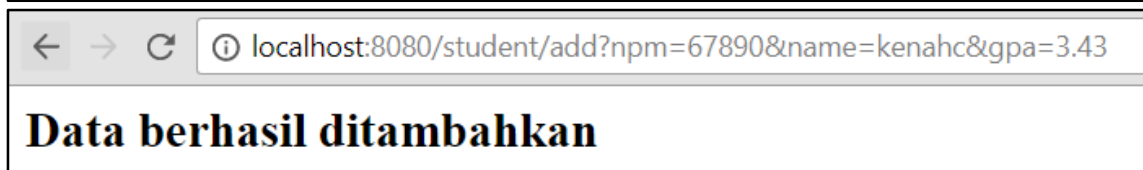
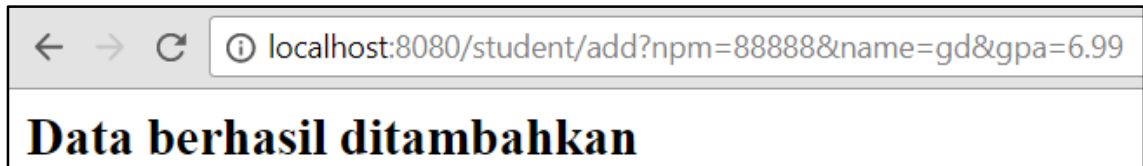
Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

Andre Gema Syahputra
1506689282
Kelas B

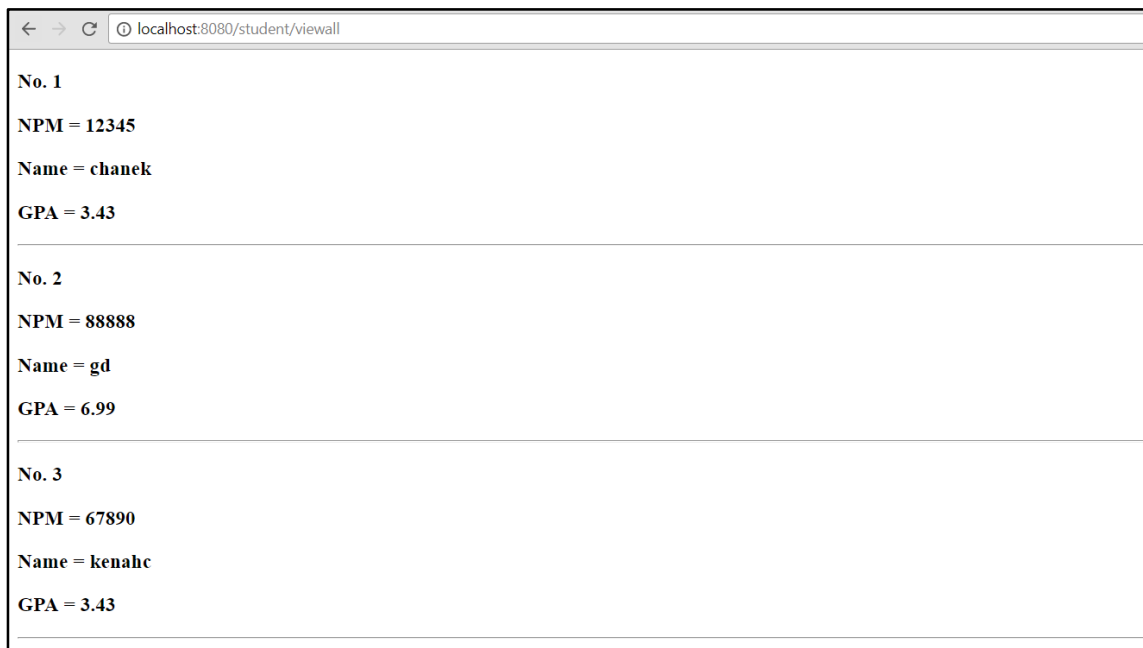
6. Coba tambahkan data Student lainnya dengan NPM yang berbeda, lalu buka localhost:8080/student/viewall,

Pertanyaan 6: Apakah semua data Student muncul?

Student yang ditambahkan:



Hasil:



Data-data Student yang ditambahkan berhasil ditampilkan semua

Arsitektur dan Pemrograman Aplikasi Perusahaan

Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

Andre Gema Syahputra
1506689282
Kelas B

Latihan

1. Pertama, saya membuat sebuah *method* pada StudentController sebagai berikut:

```
@RequestMapping("/student/view/{npm}")
public String viewPath(@PathVariable String npm, Model model)
{
    StudentModel student = studentService.selectStudent(npm);
    if(student == null) {
        model.addAttribute("errmsg", "Student dengan NPM " + npm + " tidak dapat ditemukan");
        return "error";
    }
    model.addAttribute("student", student);
    return "view";
}
```

Lalu juga saya membuat sebuah *method* yang menangkap apabila npm tidak diberikan:

```
@RequestMapping("/student/view/")
public String viewPathEmpty(Model model)
{
    model.addAttribute("errmsg", "NPM Kosong atau Tidak Diisi");
    return "error";
}
```

Saya juga menambahkan beberapa *line* untuk menangkap apabila npm tidak diberikan pada *method* view:

```
if(npm == null) {
    model.addAttribute("errmsg", "NPM Kosong atau tidak diberikan");
    return "error";
}
```

Kemudian saya membuat page view bernama error.html di templates

```
StudentController.java  error.html  viewall.html
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>Error Page</title>
5   </head>
6   <body>
7     <h1>ERROR</h1>
8     <h2 th:text="${errmsg}">Error Message</h2>
9   </body>
10 </html>
```

Arsitektur dan Pemrograman Aplikasi Perusahaan

Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

Andre Gema Syahputra
1506689282
Kelas B

Pertama, apabila NPM kosong atau tidak diberikan, akan dimunculkan *page*:

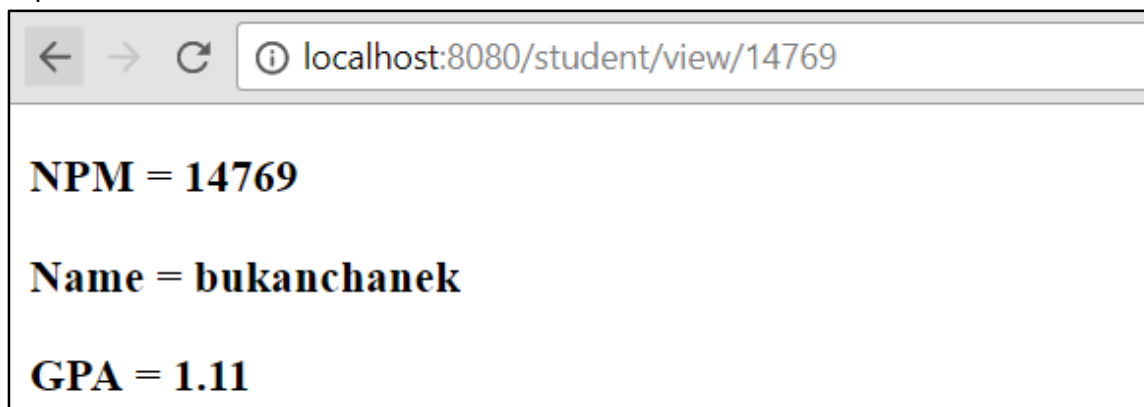


Seperti pada *method* untuk mencari dan menampilkan Student berdasarkan NPM dengan `@RequestParam`, pertama akan dicek apakah npm ada, bila tidak muncul seperti ini Student akan dicari dengan menggunakan *method* `selectStudent` pada `StudentService`.

Ketika *object* hasil dikembalikan, akan dicek apakah yang di-*return* merupakan null atau bukan. Apabila null, berarti Student dengan NPM tersebut tidak dapat ditemukan, sehingga akan dibuka page error. Pada model akan ditambahkan pesan error yang nantinya akan ditampilkan pada `error.html` seperti berikut:



Apabila yang dikembalikan adalah sebuah *object* `StudentModel`, maka *object* tersebut akan ditambahkan ke dalam model yang kemudian akan ditampilkan pada page view, seperti berikut:



Arsitektur dan Pemrograman Aplikasi Perusahaan

Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

Andre Gema Syahputra
1506689282
Kelas B

2. Pertama, saya membuat tambahan *method* pada interface `StudentService` sebagai berikut:

```
void removeStudent(StudentModel student);
```

Lalu, saya tambahkan implementasinya pada `InMemoryStudentService` sebagai berikut:

```
@Override
public void removeStudent(StudentModel student) {
    studentList.remove(student);
}
```

Pada `StudentController`, saya menambahkan *method* bernama `deletePath` sebagai berikut:

```
@RequestMapping("/student/delete/{npm}")
public String deletePath(@PathVariable String npm, Model model)
{
    StudentModel student = studentService.selectStudent(npm);
    if(student == null) {
        model.addAttribute("errorMsg", "Student dengan NPM " + npm + " tidak dapat ditemukan, proses delete dibatalkan");
        return "error";
    }
    studentService.removeStudent(student);
    return "delete";
}
```

Lalu juga saya membuat sebuah *method* yang menangkap apabila `npm` tidak diberikan:

```
@RequestMapping("/student/delete")
public String deletePathEmpty(Model model)
{
    model.addAttribute("errorMsg", "NPM Kosong atau Tidak Diisi");
    return "error";
}
```

Pada templates saya membuat view bernama `delete.html` sebagai berikut:

```
delete.html x error.html StudentModel.java StudentService....
1 <html>
2   <head>
3     <title>Add</title>
4   </head>
5   <body>
6     <h2>Data berhasil dihapus</h2>
7   </body>
8 </html>
```


Arsitektur dan Pemrograman Aplikasi Perusahaan

Tutorial 3 - Menggunakan Model dan Service pada Project Spring Boot

Andre Gema Syahputra
1506689282
Kelas B

Pertama, apabila NPM kosong atau tidak diberikan akan dimunculkan *page*:



Implementasi *method* `removeStudent` menerima `StudentModel`, dan kemudian melakukan `remove` dengan menggunakan *method* `remove` pada `List`. *Method* `deletePath` pada `StudentController` menerima NPM sebagai parameter dan kemudian mencari `Student` dengan NPM tersebut dengan memanggil *method* `selectStudent` pada `StudentService`. Seperti pada *view*, ketika *object* hasil dikembalikan, akan dicek apabila yang di-*return* merupakan `null` atau bukan.

Apabila `null`, berarti `Student` dengan NPM tersebut tidak dapat ditemukan, sehingga akan dibuka *page* `error`. Pada *model* akan ditambahkan pesan *error* yang nantinya akan ditampilkan pada `error.html`.



Apabila yang dikembalikan adalah sebuah *object* `StudentModel`, maka *method* akan memanggil *method* `removeStudent` dengan parameter `StudentModel` tersebut, sehingga akan di-`remove` dari `List`. Kemudian, akan ditampilkan *page* `delete.html`.

