

TUTORIAL 3 ADPAP

IMPLEMENTASI METHOD selectStudent

```
@Override
public StudentModel selectStudent(String npm) {

    Iterator<StudentModel> studentIterator = studentList.iterator();

    While (studentIterator.hasNext()) {
        StudentModel student = studentIterator.next();
        if(npm.equals(student.getNpm())) {
            return student;
        }
    }

    return null;
}
```

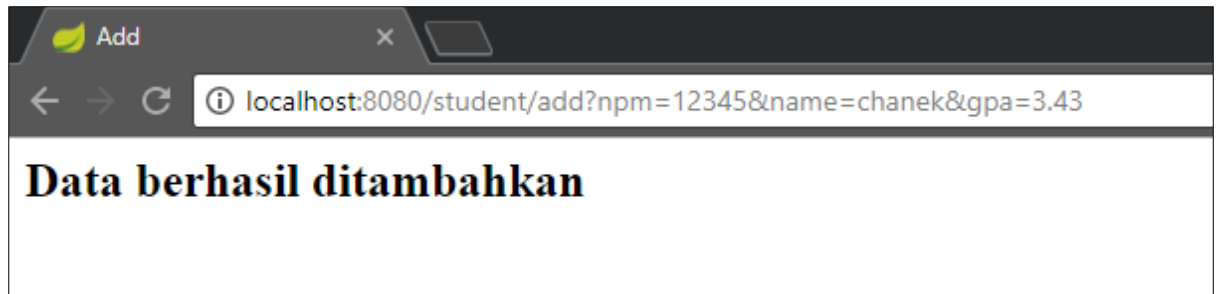
MEMBUAT CONTROLLER DAN FUNGSI ADD

Jalankan program dan buka

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

Pertanyaan 1: apakah hasilnya?

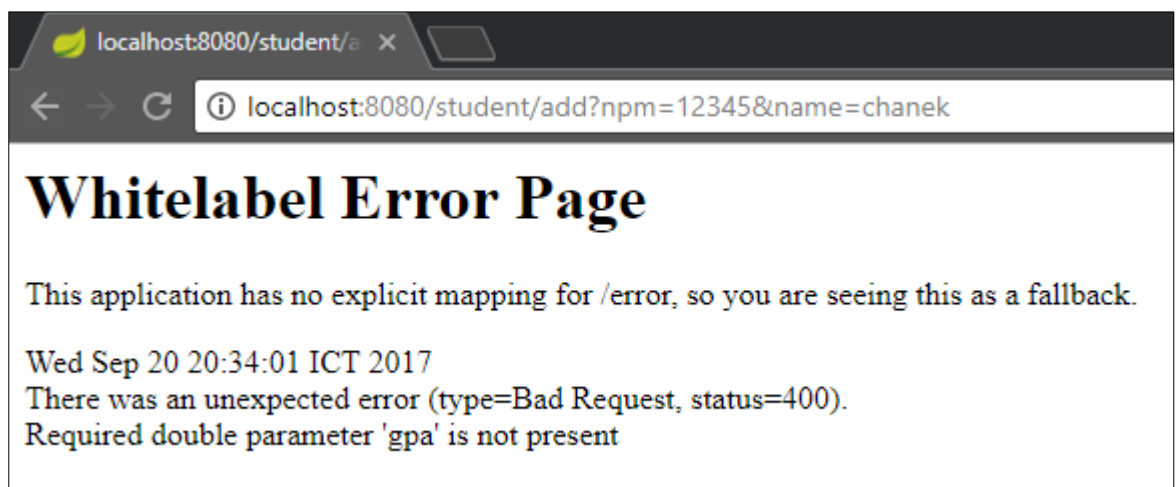
Hasilnya adalah "Data berhasil ditambahkan".



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

Pertanyaan 2: apakah hasilnya?

Whitelabel Error Page, karena url tidak memberikan masukkan gpa, sedangkan pada *source code*, tertera bahwa parameter gpa adalah *required*.



METHOD VIEW BY NPM

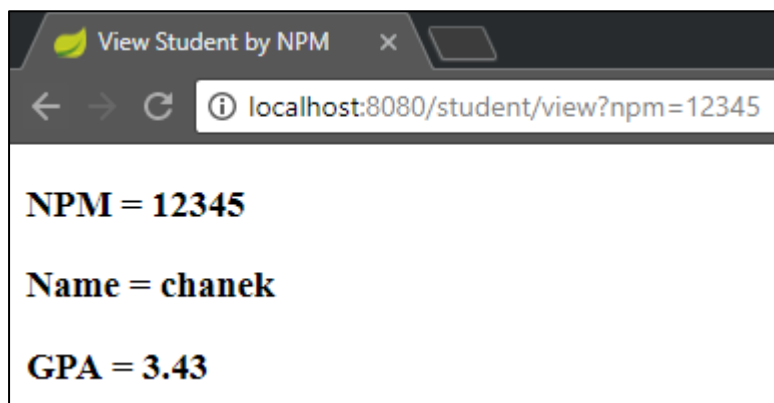
Jalankan program dan buka

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?

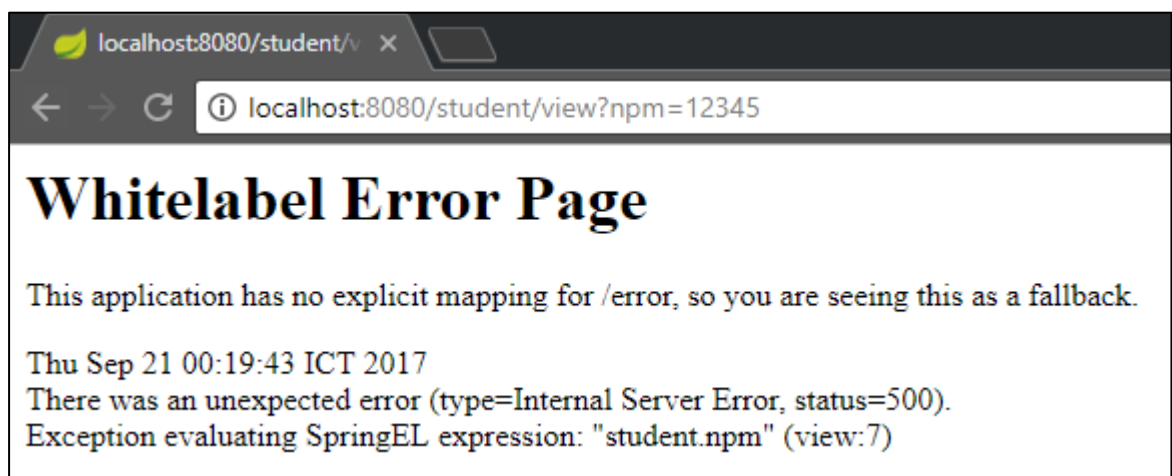
Ya, data Student tersebut muncul.



Coba matikan program dan jalankan kembali serta buka localhost:8080/student/view?npm=12345

Pertanyaan 4: apakah data Student tersebut muncul?

Tidak, karena List yang menyimpan data Student yang berada dalam class InMemoryStudentService baru diinisiasi ketika aplikasi SpringBoot dijalankan. Sehingga, datanya otomatis hilang ketika program dimatikan.



METHOD VIEW ALL

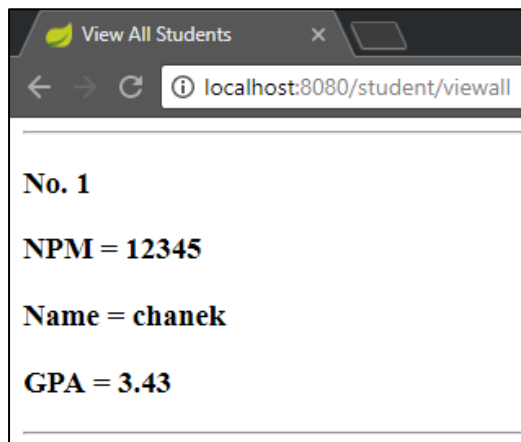
Jalankan program dan buka

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

localhost:8080/student/viewall

Pertanyaan 5: apakah data Student tersebut muncul?

Ya, data Student tersebut muncul.

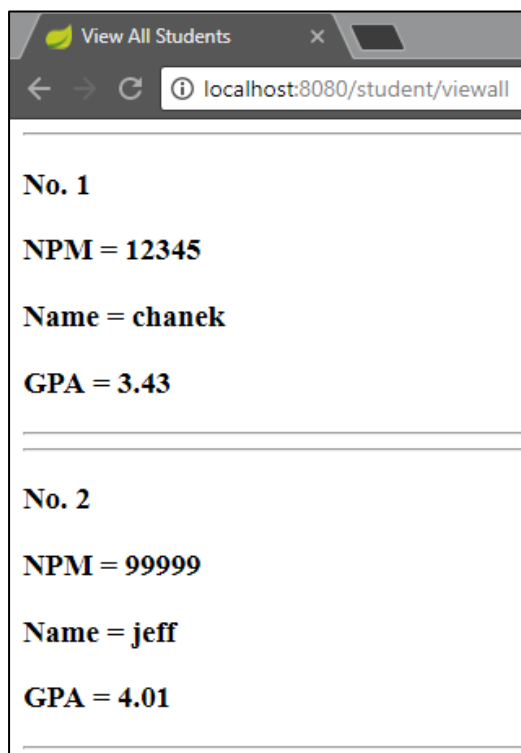


Coba tambahkan data Student lainnya dengan NPM yang berbeda, lalu buka

localhost:8080/student/viewall

Pertanyaan 6: Apakah semua data Student muncul?

Ya, semua data Student muncul.



LATIHAN

Penjelasan Fitur View dengan Path Variabel

1. Membuat *@PathVariable Optional<String>* untuk menerima *input* berupa NPM pada *StudentController.java*
2. Membuat halaman *notfound.html* untuk NPM yang tidak ditemukan pada *List* dan halaman *blankinput.html* jika *user* tidak memberikan masukan berupa NPM.
3. Pada *StudentController.java*, memeriksa apakah *user* memberik masukan berupa NPM.
 - 3.1. Jika *user* memberikan NPM, memeriksa lagi apakah NPM tersebut ditemukan pada *List*.
 - 3.1.1. Jika NPM ditemukan pada *List*, maka halaman akan dipindahkan ke halaman *view.html* untuk menampilkan informasi *Student* dengan NPM tersebut.
 - 3.1.2. Jika NPM tidak ditemukan, maka halaman akan dipindahkan ke halaman *error* dengan penjelasan NPM tidak ditemukan.
 - 3.2. Jika *user* tidak memberikan NPM, maka halaman akan dipindahkan ke halaman *error* dengan penjelasan NPM tidak diberikan.

Penjelasan Fitur Delete

1. Membuat *@PathVariable Optional<String>* untuk menerima *input* berupa NPM pada *StudentController.java*
2. Menambahkan *void deleteStudent(StudentModel student);* pada *interface StudentService.java*
3. Menambahkan *method deleteStudent(StudentModel student)* pada *class InMemoryStudentService.java*
4. Menggunakan halaman *notfound.html* untuk NPM yang tidak ditemukan pada *List* dan halaman *blankinput.html* jika *user* tidak memberikan masukan berupa NPM.
5. Pada *StudentController.java*, memeriksa apakah *user* memberik masukan berupa NPM.
 - 5.1. Jika *user* memberikan NPM, memeriksa lagi apakah NPM tersebut ditemukan pada *List*.
 - 5.1.1. Jika NPM ditemukan pada *List*, maka halaman akan dipindahkan ke halaman *delete.html* untuk menginformasikan bahwa *Student* dengan NPM tersebut telah dihapus.
 - 5.1.2. Jika NPM tidak ditemukan, maka halaman akan dipindahkan ke halaman *error* dengan penjelasan NPM tidak ditemukan.
 - 5.2. Jika *user* tidak memberikan NPM, maka halaman akan dipindahkan ke halaman *error* dengan penjelasan NPM tidak diberikan.

RINGKASAN MATERI

Pada intinya, saya belajar mengenai *service layer* pada konsep MVC:

Service berisi *business logic* dan merupakan perantara yang digunakan *controller* untuk memanipulasi data (Model) pada *database*.