

- a. Pada tutorial 3 ini saya mempelajari tentang penerapan konsep MVC pada Java Spring Boot untuk pembuatan aplikasi web. Selain itu saya mempelajari cara pembuatan *service* berupa *create*, *read* dan *delete* melalui konsep MVC. Saya juga mencoba secara mandiri menerapkan List dalam *object-oriented programming*.
- b. Jawaban dari tiap pertanyaan
 1. Hasilnya adalah pemberitahuan bahwa data berhasil ditambahkan.



2. Terdapat error karena nilai dari parameter gpa tidak didefinisikan.



3. Data dari *student* muncul.



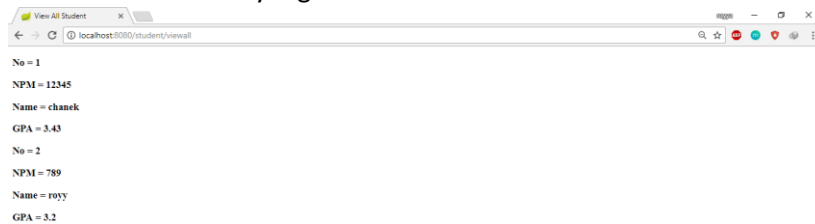
4. Data tidak muncul karena data dengan npm tersebut tidak ada di dalam sistem.



5. Data student muncul.



6. Semua data student yang telah dimasukkan muncul.



- c. Dibawah ini merupakan implementasi yang saya buat untuk method selectStudent. Pada baris 13 saya mendeklarasikan sebuah syntax for untuk mengambil data dari variabel studentList. Nilai dari studentList ditampung sementara dalam variabel student untuk kemudian di cek apakah sesuai dengan npm yang sedang saya cari. Jika data telah ditemukan maka object dari data tersebut dikembalikan ke method yang memanggil method selectStudent. Jika data tidak ditemukan maka nilai null yang akan dikembalikan.

```
10 @Override
11 public StudentModel selectStudent (String npm) {
12     // Implement
13     for(StudentModel student :studentList) {
14         if(student.getNpm().equals(npm)) {
15             return student;
16         }
17     }
18     return null;
19 }
20
```

- d. Dibawah ini akan dijelaskan fitur delete yang sudah saya buat.
1. Dapat dilihat pada line 13 saya membuat sebuah method baru pada interface StudentService.

```
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 deleteStudent(StudentModel student);
14 }
15
```

2. Setelah interface dibuat maka selanjutnya adalah mengimplementasikannya.

```
31 @Override
32 public void deleteStudent (StudentModel student) {
33     studentList.remove(student);
34 }
```

3. Langkah terakhir adalah membuat implementasi dari controller yang akan memproses delete data. Mula-mula user akan memasukkan sebuah parameter berupa data npm. Kemudian method delete akan menerima data tersebut dan disimpan pada variabel npm. Pada line 52 program akan mengecek apakah npm tersedia atau tidak. Jika npm tidak ada maka program akan langsung menampilkan notifikasi data tidak ditemukan. Namun jika npm tersedia maka akan masuk ke dalam proses pengambilan data. Kemudian pada line 54 program akan mengecek ulang apakah data ada atau tidak. Jika data tidak ada maka program akan langsung menampilkan notifikasi data tidak ditemukan. Tetapi jika data ditemukan maka data dari object akan dihapus kemudian program akan menampilkan data berhasil dihapus.

```
49
50 @RequestMapping(value = {"/student/delete", "student/delete/{npm}}")
51 public String delete(Model model, @PathVariable Optional<String> npm){
52     if(npm.isPresent()) {
53         StudentModel student = studentService.selectStudent(npm.get());
54         if(student == null) {
55             return "notfound";
56         }
57         studentService.deleteStudent(student);
58         return "delete";
59     }
60     return "notfound";
61 }
62
```