Pada tutorial kali ini menambahkan implementasi sebuah Service serta package model dari tutorial sebelum nya yang belum ada. Dengan begitu focus tutorial kali ini sudah menggunakan konsep MVC.

**Pertanyaan 1**​: apakah hasilnya? Jika error, tuliskan penjelasan Anda.

Hasil : Data Berhasil ditambahkan

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

Hasil : Error karena parameter gpa tidak tercantumkan karena pada class Controller RequestMapping tersebut terdapat gpa bersifat required.

**Pertanyaan 3​**: apakah data Student tersebut muncul? Jika tidak, mengapa?

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

Hasil : Data Berhasil ditambahkan

NPM = 12345

Name = Chanek

Gpa = 3.43

**Pertanyaan 4​:** apakah datasetelah coba matikan program dan jalankan kembali serta buka

localhost:8080/student/view?npm=12345

tersebut muncul? Jika tidak, mengapa?

Hasil : Error karena data yang dimasukan pada method sebelumnya tidak ter record dan ketika Spring dimatikan maka data nya terhapus.

**Pertanyaan 5:**

Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka localhost:8080/student/viewall,

Hasil : Data berhasil ditambahkan

NPM = 12345

Name = Chanek

Gpa = 3.43

**Pertanyaan 6 :**

localhost:8080/student/add?npm=333&name=syalala&gpa=3.50

Hasil : Data berhasil ditambahkan

NPM = 333

Name = syalala

Gpa = 3.50

**Select Student**

**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**;

}

Program akan menjalankan iterasi berdasarkan ukuran studentList yang ditambahkan. Setelahnya perulangan akan mengambil nilai I untuk di cocokan menggunakan getNpm() hingga sama dengan nilai npm yang berada dalam studentList.

@RequestMapping(value= {"/student/delete","/student/delete/{npm}"})

**public** String studentDeletePath(@PathVariable Optional<String> npm, Model model)

{

**if**(npm.isPresent()) {

String Npm = npm.toString().substring(9, npm.toString().length()-1);

StudentModel student = studentService.selectStudent(Npm);

**if**(student == **null**) {

model.addAttribute("errorMessage","NPM tidak ada");

}**else** {

studentService.removeStudent(student);

model.addAttribute("errorMessage","NPM berhasil dihapus");

}

}**else** {

model.addAttribute("errorMessage","NPM tidak ada");

}

**return** "error";

}

@RequestMapping tersebut akan mendelete berdasarkan path/student/delete/npm dimana npm yang dimaksud adalah npm yang dijadikan value untuk mendelete existing pada studentList. Terdapat kondisi apabila npm tidak ada apabila tidak ada tercocokan antara npm path dan existing, dan apabila ada kecocokan npm maka data akan terhapus.