

Arsitektur dan Pemrograman Aplikasi
Enterprise

Organizing Domain Logic & Service Layer



Muhammad Fajar Pamungkas
1706106835



FACULTY OF
**COMPUTER
SCIENCE**

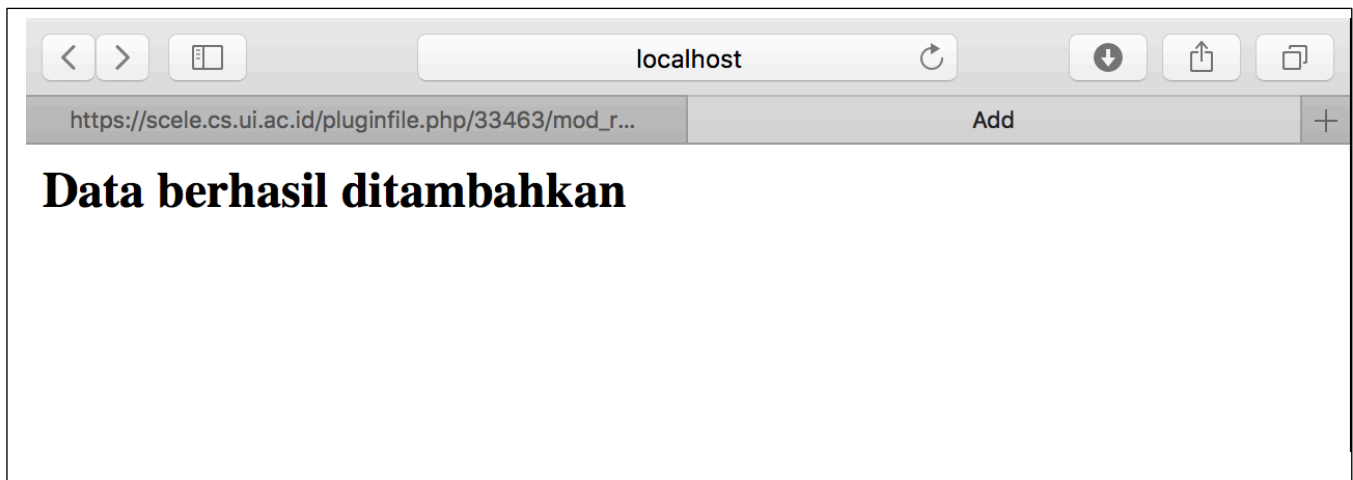
Tutorial

Jalankan program dan buka :

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

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

Tidak terjadi error dan ditampilkan pesan data berhasil ditambahkan.



Jalankan program dan buka :

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

Pertanyaan 2 : apakah hasilnya? Jika *error* , tuliskan penjelasan Anda.

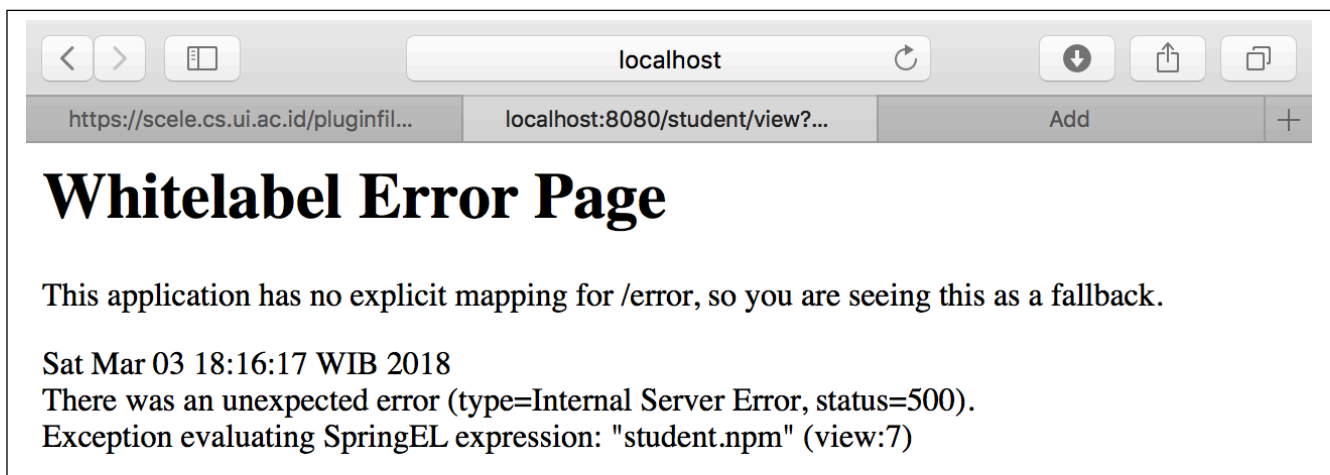
Ditampilkan pesan error yang disebabkan url tidak menyertakan parameter GET GPA, dimana parameter tersebut tidak bersifat optional sehingga jika tidak disertakan, aplikasi tidak dapat handle request yang dikehendaki.



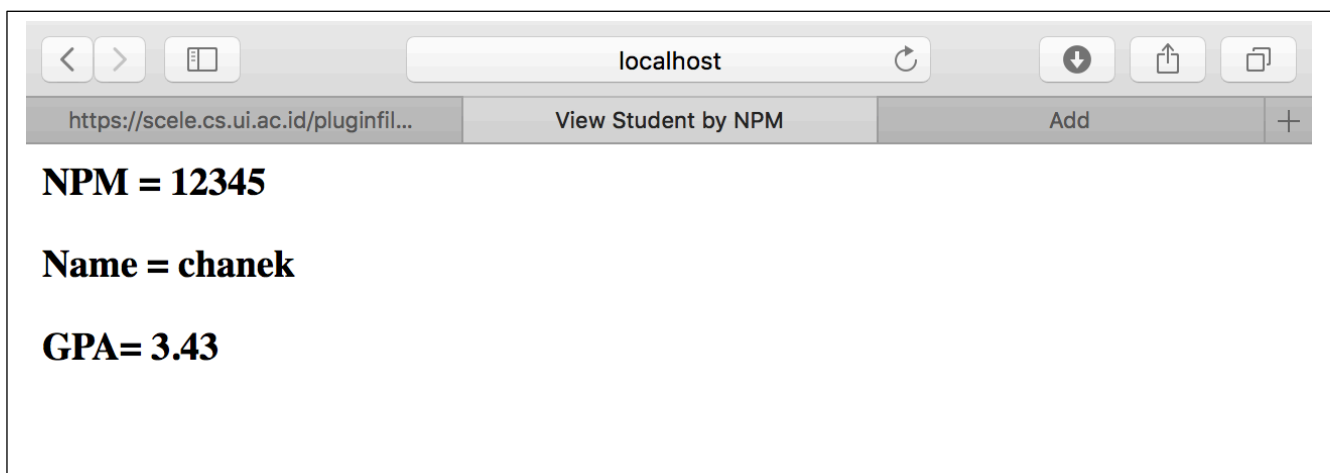
Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka localhost:8080/student/view?npm=12345

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

Tidak, karena pada file view.html, attribute object dipanggil secara langsung sedangkan access modifier attribute pada class StudentModel adalah private tidak public sehingga untuk menampilkan data attribute object yang dibuat berdasarkan class StudentModel harus menggunakan method getter.



Setelah mengubah pemanggilan attribute menggunakan method getter pada file view.html, data student dapat 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?

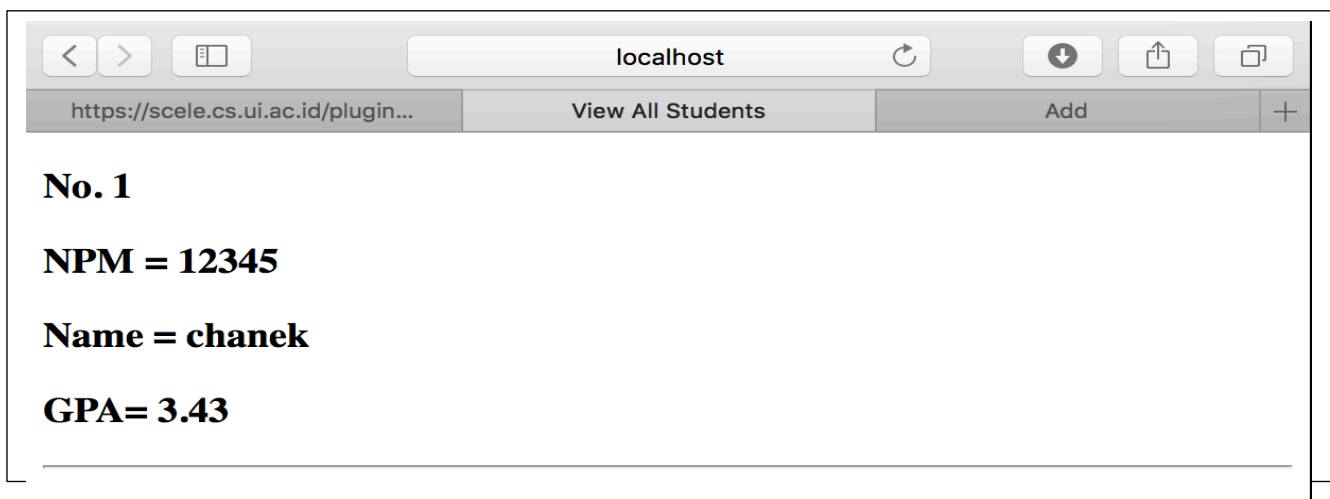
Tidak, karena data yang sebelumnya ditambahkan hanya disimpan pada memory menggunakan interface list, sehingga ketika aplikasi direstart maka object yang disimpan akan hilang karena object studentService dari InMemoryService yang dibuat berdasarkan interface StudentService akan dinstansiasi ulang sehingga data pada List yang dibuat akan kembali kosong.



3. 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?



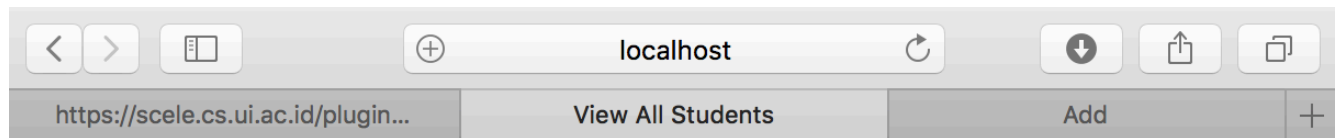
Ya, data yang baru ditambahkan muncul pada halaman viewall.

4. 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 yang ditambahkan muncul pada halaman viewall



No. 1

NPM = 12345

Name = chanek

GPA= 3.43

No. 2

NPM = 54321

Name = SempatBingung

GPA= 4.0

No. 3

NPM = 65432

Name = SempatPaham

GPA= 3.99

Implementasi selectStudent

```
△ - public StudentModel selectStudent(String npm) {  
    - Comparator<StudentModel> s = new Comparator<StudentModel>() {  
△ -     public int compare(StudentModel s1, StudentModel s2) {  
         return s1.getNPM().compareTo(s2.getNPM());  
     }  
    };  
  
    Collections.sort(studentList,s);  
    int index = Collections.binarySearch(studentList, new StudentModel(npm,null,0), s);  
  
    if(index < 0)  
        return null;  
  
    return studentList.get(index);  
}
```

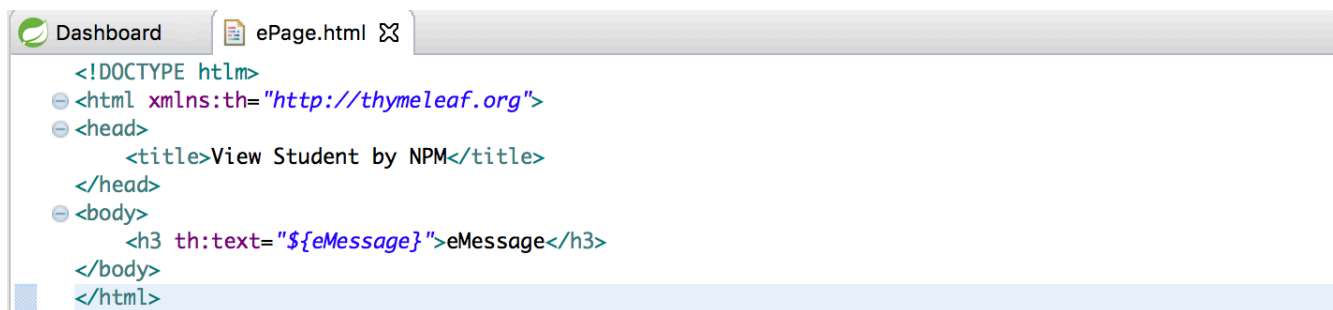
Latihan

1.

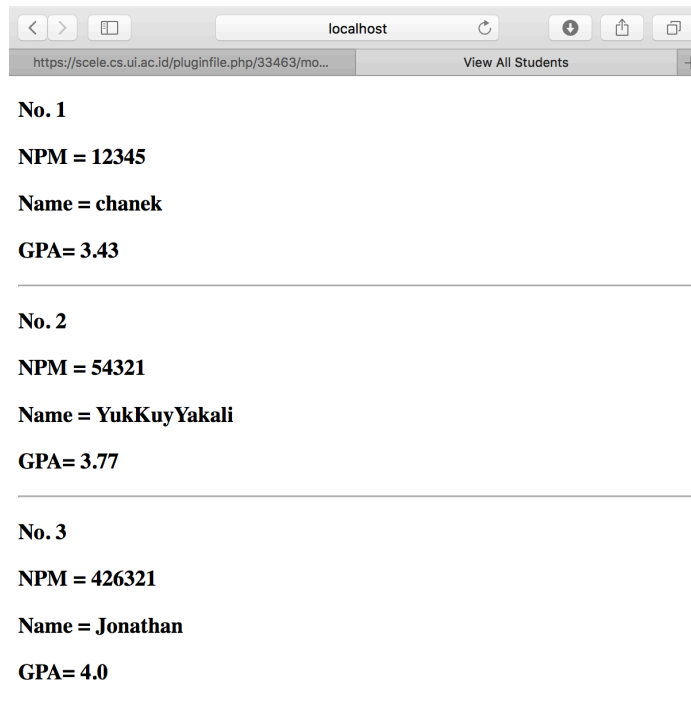
```
@RequestMapping(value = {"/student/view","student/view/{npm}"})
public String viewPath(@PathVariable Optional<String> npm, Model model)
{
    String view = "view";
    if(npm.isPresent()){
        StudentModel student = studentService.selectStudent(npm.get());
        if(student == null)
        {
            model.addAttribute("eMessage","Tidak ditemukan");
            view = "ePage";
        }else {
            model.addAttribute("student", student);
            view = "view";
        }
    } else {
        model.addAttribute("eMessage","NPM Kosong");
        view = "ePage";
    }

    return view;
};
```

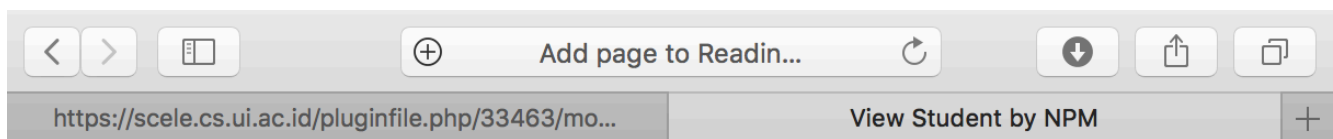
Membuat function dengan menggunakan Path Variable untuk menghandle request pada URL /student/view/{npm}. Apabila student berisikan object null atau parameter GET npm tidak ditemukan maka view akan mengarah pada file ePage.html, dengan mengirimkan attribute eMessage yang berisikan pesan Tidak ditemukan atau NPM kosong.



```
<!DOCTYPE html>
<html xmlns:th="http://thymeleaf.org">
<head>
<title>View Student by NPM</title>
</head>
<body>
<h3 th:text="${eMessage}">eMessage</h3>
</body>
</html>
```

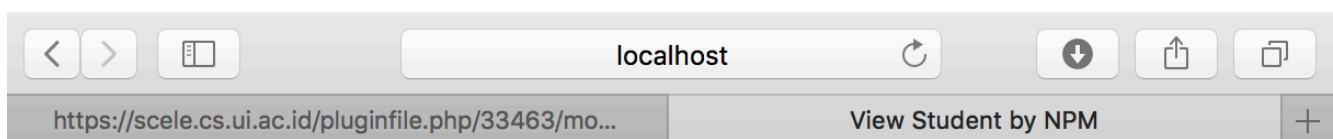


Seluruh data yang telah ditambahkan, dan ketika tidak terdapat parameter npm pada url yang diberikan maka akan dimunculkan pesan NPM Kosong.

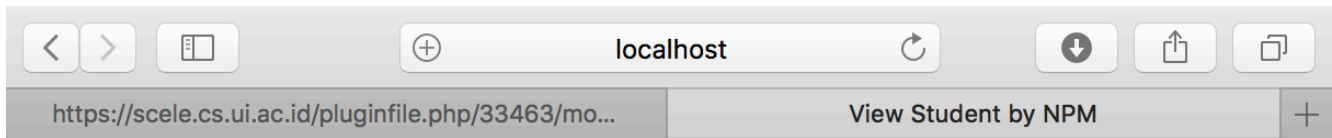


NPM Kosong

Apabila paramter tidak kosong namun data yang dicari tidak pernah diinputkan sebelumnya maka akan ditampilkan pesan Tidak ditemukan.



Tidak ditemukan



NPM = 426321

Name = Jonathan

GPA= 4.0

Data yang sesuai dengan request yang diterima akan ditampilkan secara lengkap.

2.

Menambahkan function deleteStudent pada interface StudentService.

```
+ import java.util.List;
public interface StudentService {
    StudentModel selectStudent(String npm);

    List<StudentModel> selectAllStudents();

    void addStudent(StudentModel student);

    void deleteStudent(StudentModel student);
}
```

Lalu mengimplementasikan function deleteStudent interface StudentService pada class InMemoryStudentService seperti pada gambar dibawah.

```
public void deleteStudent(StudentModel student) {
    studentList.remove(student);
}
```

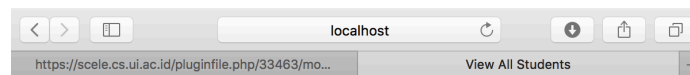
```

@RequestMapping(value = {"/student/delete","student/delete/{npm}"})
public String deletePath(@PathVariable Optional<String> npm, Model model)
{
    if(npm.isPresent()){
        StudentModel student = studentService.selectStudent(npm.get());
        if(student == null)
        {
            model.addAttribute("eMessage","Tidak ditemukan");
        }else {
            model.addAttribute("eMessage", "Data berhasil dihapus");
            studentService.deleteStudent(student);
        }
    } else {
        model.addAttribute("eMessage","NPM Kosong");
    }

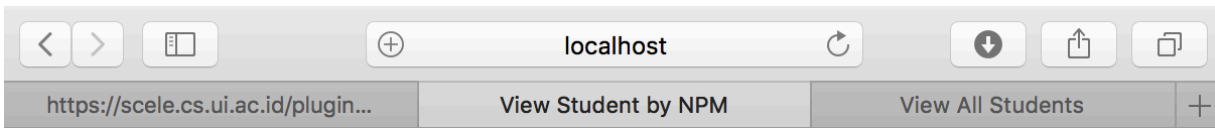
    return "ePage";
};

```

Lalu membuat function dengan menggunakan Path Variable untuk menghandle request pada URL /student/view/{npm}. Penerapan algoritma hampir mirip dengan pencarian data hanya saja jika ditemukan data dengan request npm yang diminta maka data tersebut akan dihapus.

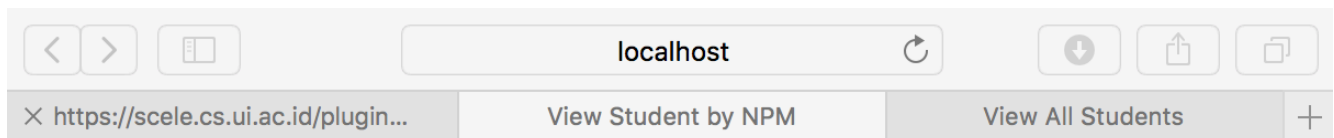


No. 1
NPM = 12345
Name = chanek
GPA= 3.43
<hr/>
No. 2
NPM = 54321
Name = YukKuyYakali
GPA= 3.77
<hr/>
No. 3
NPM = 426321
Name = Jonathan
GPA= 4.0
<hr/>



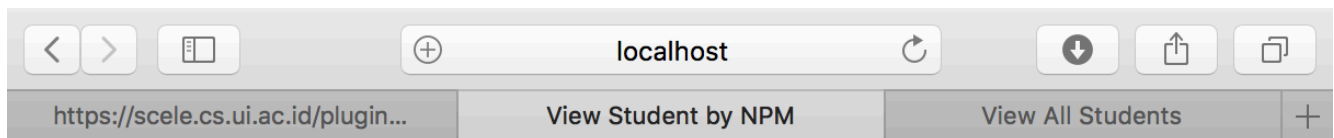
NPM Kosong

ketika tidak terdapat parameter npm pada url yang diberikan maka akan dimunculkan pesan NPM Kosong.



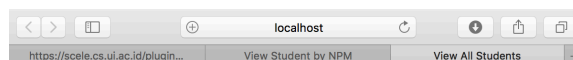
Tidak ditemukan

Jika tidak terdapat dengan request NPM yang sama maka akan ditampilkan pesan Tidak ditemukan.



Data berhasil dihapus

Apabila data ditemukan maka data tersebut akan dihapus dari List.



No. 1

NPM = 12345

Name = chanek

GPA= 3.43

No. 2

NPM = 426321

Name = Jonathan

GPA= 4.0
