

Tutorial 3 – Menggunakan Model dan Service pada Project Spring Boot

Konsep yang diterapkan pada penggunaan framework spring boot adalah MVC (Model-View-Controller). Pada tutorial ini akan dibahas mengenai komponen model dan service.

Model merupakan sebuah objek yang merepresentasikan dan menyimpan informasi terhadap suatu hal. Contoh dari model adalah objek mahasiswa. Objek mahasiswa ini memiliki nama, alamat, tanggal lahir, nomor pokok mahasiswa, dsbnya. Model digunakan untuk merepresentasikan hal-hal tersebut dalam atribut yang dimiliki sebuah model.

Service adalah suatu layer yang menjadi mediator antara controller dan database. Pada service layer, disimpan business logic yang digunakan untuk mengolah data yang terdapat dalam database. Pengolahan ini meliputi kalkulasi data yang diambil dari database, manipulasi user input kedalam database, dsbnya. Contohnya adalah melakukan kalkulasi IPK sebuah model mahasiswa.

Membuat Class Model

Salah satu komponen dalam MVC adalah model. Anda akan membuat model berupa class Student yang berisi variabel name (String), npm (String), dan gpa (double).

```
package com.example.tutorial3.model;

public class StudentModel {
    private String name;
    private String npm;
    private double gpa;

    public StudentModel(String npm, String name, double gpa) {
        this.name = name;
        this.npm = npm;
        this.gpa = gpa;
    }

    public StudentModel(String npm) {}

    public String getName() {}

    public void setName(String name) {}

    public String getNpm() {}

    public void setNpm(String npm) {}

    public double getGpa() {}

    public void setGpa(double gpa) {}
}
```

Membuat Service

Service pada tutorial ini merupakan sebuah interface yang mendefinisikan method-method yang dapat memanipulasi objek Student dengan nama StudentService.

```
package com.example.tutorial3.service;

import java.util.List;

public interface StudentService {
    StudentModel selectStudent(String npm);

    List<StudentModel> selectAllStudents();

    void addStudent(StudentModel student);
}
```

Terdapat juga class yang mengimplementasikan method-method yang dimiliki oleh interface.

```
package com.example.tutorial3.service;

import java.util.ArrayList;

public class InMemoryStudentService implements StudentService{
    private static List<StudentModel> studentList = new ArrayList<StudentModel>();

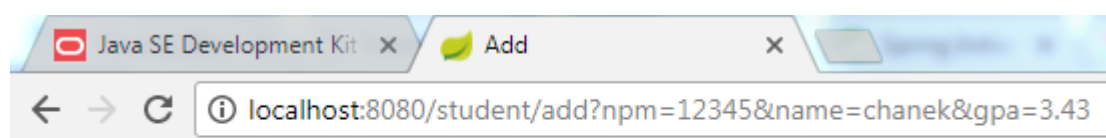
    @Override
    public StudentModel selectStudent(String npm) {
        for(int i=0; i<studentList.size(); i++) {
            if(npm.equalsIgnoreCase(studentList.get(i).getNpm())) {
                return studentList.get(i);
            }
        }
        return null;
    }

    @Override
    public List<StudentModel> selectAllStudents(){
        return studentList;
    }

    @Override
    public void addStudent(StudentModel student) {
        studentList.add(student);
    }
}
```

Membuat Controller dan Fungsi Add

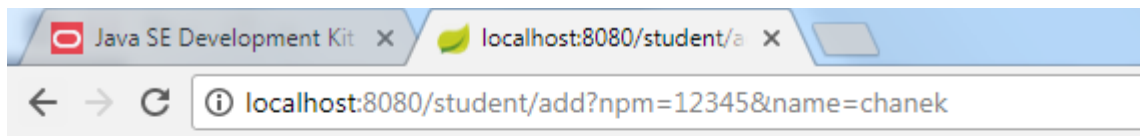
1. Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43
Pertanyaan 1: apakah hasilnya? Berhasil menampilkan pesan "Data berhasil ditambahkan" menandakan bahwa data berhasil ditambahkan.



Data berhasil ditambahkan

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

Pertanyaan 2: apakah hasilnya? Hasilnya adalah error, karena pada fungsi add terdapat tiga parameter yaitu npm, name dan gpa. Sementara pada url hanya memberikan dua parameter saja.



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

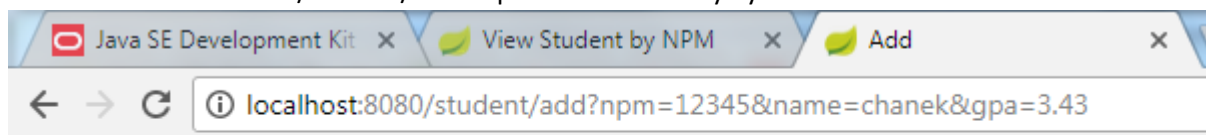
Tue Feb 27 20:31:04 ICT 2018

There was an unexpected error (type=Bad Request, status=400).

Required double parameter 'gpa' is not present

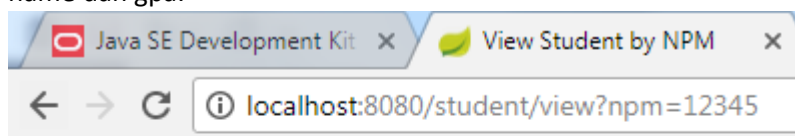
Method View by NPM

1. Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka localhost:8080/student/view?npm=12345. Hasilnya yaitu data berhasil ditambahkan.



Data berhasil ditambahkan

Pertanyaan 3: apakah data Student tersebut muncul? Data mahasiswa dimunculkan yaitu npm, name dan gpa.

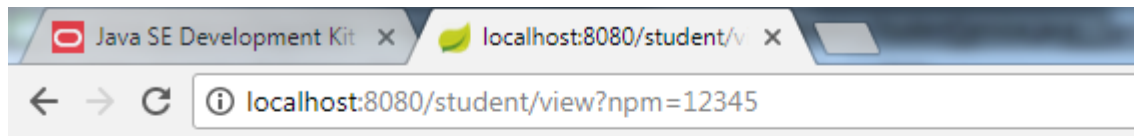


NPM = 12345

Name = chanek

GPA = 3.43

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



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

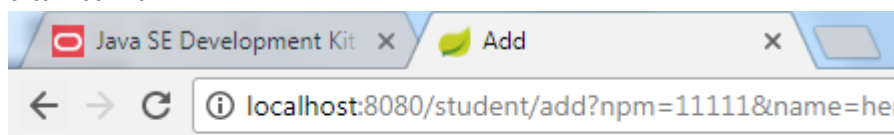
Tue Feb 27 21:03:18 ICT 2018

There was an unexpected error (type=Internal Server Error, status=500).

Exception evaluating SpringEL expression: "student.npm" (view:7)

Pertanyaan 4: apakah data Student tersebut muncul? Data tidak muncul. Alasan karena data tidak persistent atau tidak disimpan dalam database sehingga jika di stop maka data yang sudah ditambahkan akan hilang. Jika ingin view kembali maka data harus ditambahkan kembali.

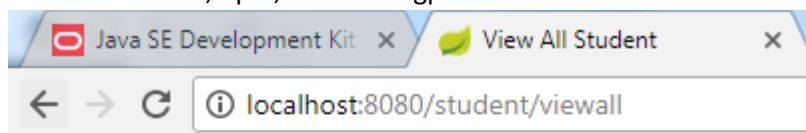
3. Coba tambahkan data Student lainnya dengan NPM yang berbeda hasilnya data berhasil ditambahkan.



Data berhasil ditambahkan

Method View All

1. 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 muncul dilengkapi dengan data nomor urut, npm, nama dan gpa.



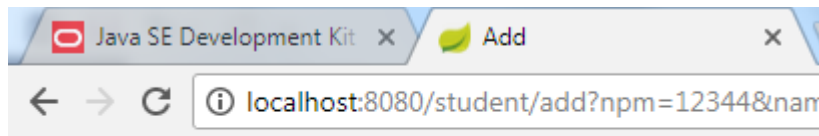
No. 1

NPM = 12345

Name = chanek

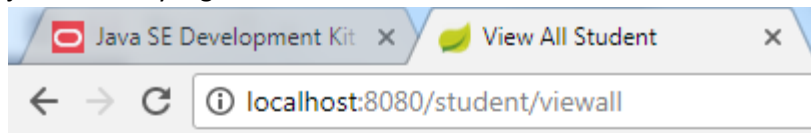
GPA = 3.43

2. Coba tambahkan data Student lainnya dengan NPM yang berbeda, maka data berhasil ditambahkan seperti pada gambar di bawah ini.



Data berhasil ditambahkan

lalu buka localhost:8080/student/viewall maka semua data student ditampilkan sesuai dengan jumlah data yang ditambahkan.



No. 1

NPM = 12345

Name = chanek

GPA = 3.43

No. 2

NPM = 12344

Name = ruth

GPA = 3.77

Pertanyaan 6: Apakah semua data Student muncul? Iya, semua data muncul dengan fungsi seperti pada gambar di bawah ini.

```
@Override
public StudentModel selectStudent(String npm) {
    for(int i=0; i<studentList.size(); i++) {
        if(npm.equalsIgnoreCase(studentList.get(i).getNpm())) {
            return studentList.get(i);
        }
    }
    return null;
}
```

Latihan

1. Pada StudentController tambahkan sebuah method view Student dengan menggunakan Path Variable. Misalnya, Anda ingin memasukkan data seorang Student yang memiliki NPM 14769, untuk melihat data yang baru dimasukkan tersebut dapat mengakses halaman

localhost:8080/student/view/14769. Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman error yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan.

Tahapan yang dilakukan adalah dengan menambahkan pengecekan pada controller untuk memberikan kondisi apabila parameter yang diberikan tidak ada pada list atau fungsi dipanggil tanpa parameter. Implementasinya dapat dilihat pada gambar berikut ini.

```
@RequestMapping(value = {"/student/view/", "/student/view/{npm}"})
public String viewStudent(Model model, @PathVariable Optional<String> npm) {
    if(npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if(student != null) {
            model.addAttribute("student", student);
            return "view";
        }else {
            return "errorview";
        }
    }else {
        return "errorview";
    }
}
```

Berikut ini data yang telah ditambahkan pada list student.

← → ↻ ⓘ localhost:8080/student/viewall

No. 1

NPM = 11111

Name = ani

GPA = 3.5

No. 2

NPM = 12345

Name = ani

GPA = 3.35

Ketika menampilkan student dengan npm = 12345 (ada dalam list) memunculkan data student

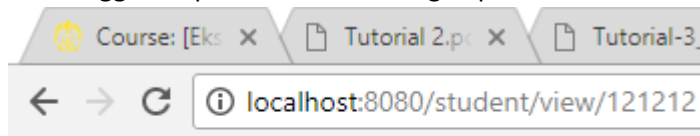
← → ↻ ⓘ localhost:8080/student/view/12345

NPM = 12345

Name = ani

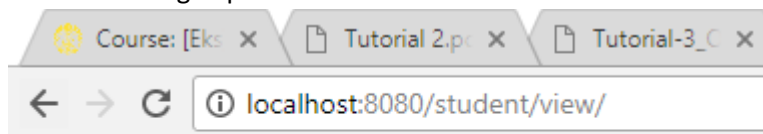
GPA = 3.35

Tampilan ketika menampilkan student dengan npm = 121212 (tidak ada dalam list) akan memanggil tampilan errorview dengan pesan Student tidak ditemukan.



Student tidak ditemukan

Tampilan ketika menampilkan student dengan npm kosong akan memanggil tampilan errorview dengan pesan Student tidak ditemukan.



Student tidak ditemukan

2. Tambahkan fitur untuk melakukan delete Student berdasarkan NPM. Misalnya, setelah melakukan add Student pada soal nomor 1, cobalah untuk melakukan delete data tersebut dengan mengakses halaman localhost:8080/student/delete/14769.

Tahapan yang dilakukan adalah dengan menambahkan method pada interface seperti pada gambar di bawah ini.

```
package com.example.tutorial3.service;

import java.util.List;

public interface StudentService {
    StudentModel selectStudent(String npm);

    List<StudentModel> selectAllStudents();

    void addStudent(StudentModel student);

    void delete(String npm);
}
```

Kemudian tambahkan implementasi untuk interface tersebut dengan menggunakan method remove student dengan npm yang menjadi parameter dari list. Pertama perlu melakukan perulangan untuk menemukan student yang sama dengan parameter, dan kemudian menghapus data student yang sesuai.

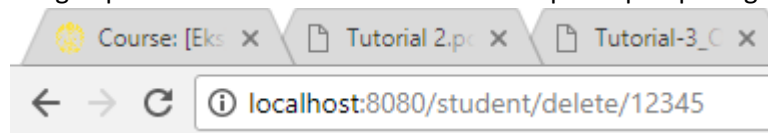
```
@Override
public void delete(String npm) {
    for(int i=0; i<studentList.size(); i++) {
        if(npm.equalsIgnoreCase(studentList.get(i).getNpm())) {
            studentList.remove(i);
        }
    }
}
```

Kemudian pada controller ditambahkan method untuk request mapping pemanggilan method delete pada url.

```
@RequestMapping(value = {"/student/delete/", "/student/delete/{npm}"})
public String delete(Model model, @PathVariable Optional<String> npm) {
    if(npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if(student != null) {
            model.addAttribute("student", student);
            return "delete";
        }else {
            return "errorview";
        }
    }else {
        return "errorview";
    }
}
```

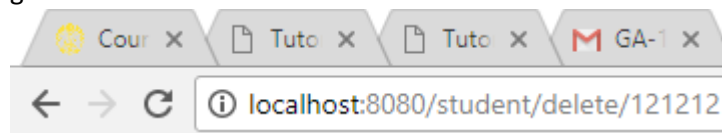
Tampilkan sebuah halaman yang memberikan informasi bahwa data tersebut telah berhasil dihapus. Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman error yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan dan proses delete dibatalkan.

Ketika menghapus student dengan npm = 12345 (ada dalam list) akan menampilkan tampilan dengan pesan bahwa data telah berhasil dihapus seperti pada gambar di bawah ini.



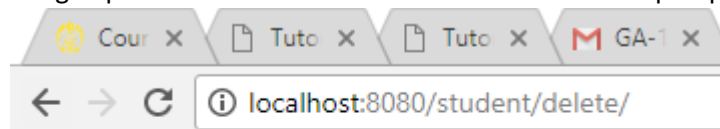
Data telah berhasil dihapus.

Tampilan ketika menampilkan student dengan npm = 121212 (tidak ada dalam list) akan menampilkan tampilan dengan pesan bahwa data student tidak ditemukan seperti pada gambar di bawah ini.



Student tidak ditemukan

Tampilan ketika menampilkan student dengan npm kosong akan menampilkan tampilan dengan pesan bahwa data student tidak ditemukan seperti pada gambar di bawah ini.



Student tidak ditemukan