Faisal Satrio Priatmadji
1506738504
ADPAP - C

1. Jika menggunakan Object sebagai parameter pada form POST, bagaimana caranya melakukan validasi input yang optional dan input yang required seperti jika menggunakan RequestParam? Apakah validasi diperlukan? Asumsikan input pada form Anda tidak menggunakan attribute required sehingga butuh validasi di backend.

Mengecek property dari objek null atau tidak, dapat di handle pada studentController, dimana gpa tidak boleh kosong, hal tersebut disebabkan karena apabila gpa kosong maka textbox akan mengembalikan tipe string, sedangkan gpa merupakan double sehingga akan menimbulkan error. mengecek apakah student.npm tidak ada isi / null sehingga input tidak bisa ditinggalkan.

2. Menurut Anda, mengapa form submit biasanya menggunakan POST method dibanding GET method? Apakah perlu penanganan berbeda di header atau body method di controller jika form di post dikirim menggunakan method berbeda?

Karena Method POST digunakan untuk mengubah database, selain itu method GET dapat diakses melalui browser sehingga menjadi tidak aman karena data tersebut dapat diambil oleh siapa saja. Perlu, karena setiap parameter method membutuhkan RequestParameter tertentu pula seperti RequestParameter.POST dan RequestParameter.GET

3. Apakah mungkin satu method menerima lebih dari satu jenis request method, misalkan menerima GET sekaligus POST?

Ya dapat dilakukan dengan *method = { RequestMethod.GET, RequestMethod.POST } di* parameter *@RequestMapping*.

Faisal Satrio Priatmadji
1506738504
ADPAP - C

Penjelasan menambahkan method Delete

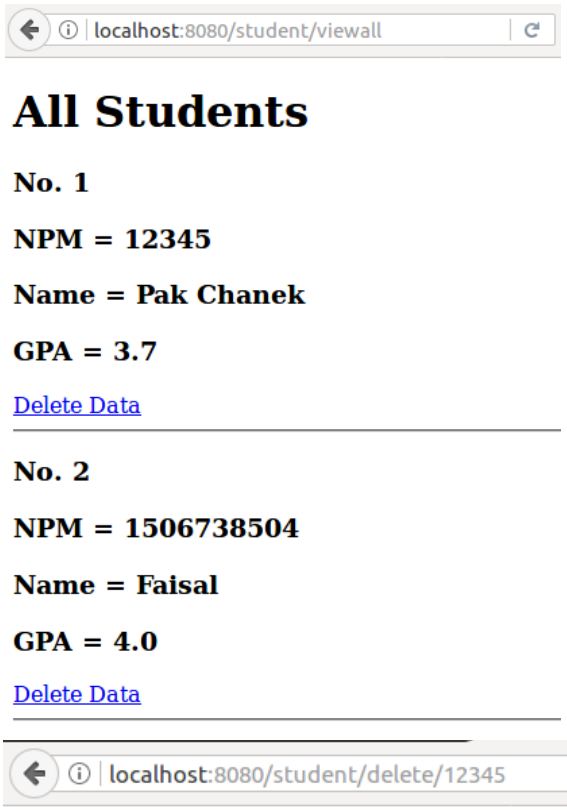
Method delete pada controller

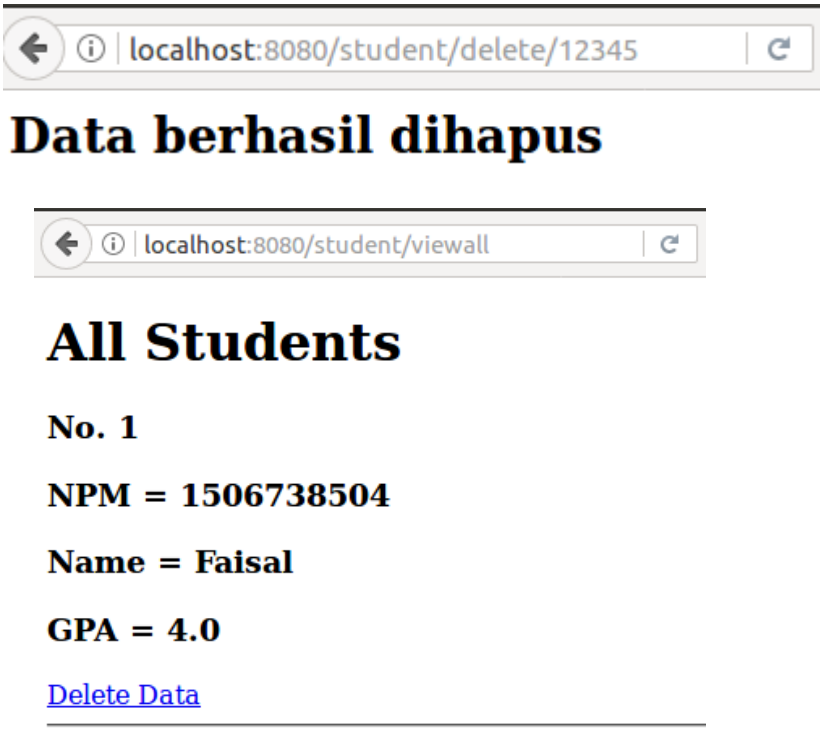
```
@RequestMapping("/student/delete/{npm}")
public String delete (Model model, @PathVariable(value = "npm")
{
    StudentModel student = studentDAO.selectStudent(npm);

    if(student == null){
        model.addAttribute("npm", npm);
        return "not-found";
    }

    studentDAO.deleteStudent (npm);

    return "delete";
}
```

Query method delete

```
@Delete("DELETE FROM student WHERE npm = #{npm}")
void deteleStudent (String npm);
```

Pertama akan dilakukan pengecekkan student dengan npm yang dimasukkan ke paramter, jika student tidak ditemukan maka akan memanggil not-found.html, jika ditemukan maka student akan dihapus dengan menjalankan query method delete tersebut.





Faisal Satrio Priatmadji
1506738504
ADPAP - C

Penjelasan menambahkan method Update

Method Update pada controller

- Method untuk menampilkan form update

```
@RequestMapping("/student/update/{npm}")
public String update (Model model, @PathVariable(value = "npm") String npm)
{
    StudentModel student = studentDAO.selectStudent(npm);

    if(student == null){
        model.addAttribute("npm", npm);
        return "not-found";
    }

    model.addAttribute("student", student);

    return "form-update";
}
```

- Method untuk meng-*submit* form update

```
@RequestMapping(value = "/student/update/submit", method = RequestMethod.POST)
public String updateSubmit (@RequestParam(value = "npm", required = false) String npm,
        @RequestParam(value = "name", required = false) String name,
        @RequestParam(value = "gpa", required = false) double gpa, Model model)
{
    StudentModel student = studentDAO.selectStudent(npm);
    if(student == null){
        model.addAttribute("npm", npm);
        return "not-found";
    }

    studentDAO.updateStudent (new StudentModel(npm, name, gpa));

    return "success-update";
}
```

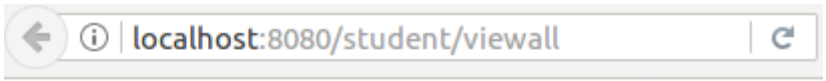
Query method update

```
@Update("UPDATE student SET name = #{name}, gpa = #{gpa} WHERE npm = #{npm}")
void updateStudent (StudentModel student);
```

Pada method menampilkan form update akan mengecek apakah student ada dengan npm yang dimasukkan melalui parameter, jika ada maka akan memanggil form-update dan data-data student akan dikirim melalui model.

Pada method submit form update akan menerima 3 buah parameter yaitu npm, name, dan gpa. Pertama-tama mengecek apakan npm tersebut ada atau tidak, jika ada maka student dengan npm tersebut akan diupdate data nya dan menampilkan halaman success-update.

Faisal Satrio Priatmadji
1506738504
ADPAP - C



All Students

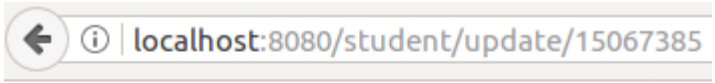
No. 1

NPM = 1506738504

Name = Faisal

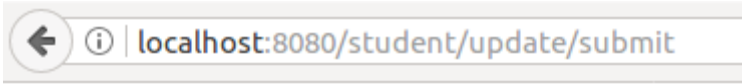
GPA = 4.0

Delete Data
Update Data

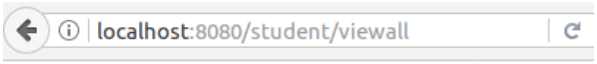


Update Student

NPM 1506738504
Name Faisal Satrio
GPA 4.0
Update



Data berhasil diupdate



All Students

No. 1

NPM = 1506738504

Name = Faisal Satrio

GPA = 4.0

Delete Data
Update Data

Faisal Satrio Priatmadji
1506738504
ADPAP - C

Penjelasan menggunakan objek sebagai parameter

```
@RequestMapping(value = "/student/update/submit", method = RequestMethod.POST)
public String updateSubmit (StudentModel student, Model model)
{
    StudentModel s = studentDAO.selectStudent(student.getNpm());
    if(s == null){
        model.addAttribute("npm", student.getNpm());
        return "not-found";
    }

    studentDAO.updateStudent (student);

    return "success-update";
}
```

Pada method submit form update, akan menerima parameter berupa objek StudentModel, lalu akan mengecek apakah student dengan npm tersebut ada atau tidak, jika tidak maka akan menampilkan halaman not-found. Namun jika ada student akan diupdate dan menampilkan halaman success-update.

Hal yang saya pelajari dari tutorial ini

Saya mempelajari bagaimana menggunakan parameter berupa objek dan juga bagaimana melakukan validasi. Serta menuliskan info log pada method.