* Delete

Proses menghapus data mahasiswa dimulai dari kelas StudentController. Pada kelas ini saya membuat method delete yang akan mengecek terlebih dahulu apakah npm mahasiswa ada di database atau tidak. Jika ada, maka dilakukan pemanggilan method deleteStudent yang ada di kelas studentService. Kemudian, view “delete.html” akan ditampilkan. Jika mahasiswa yang ingin dihapus dari database tidak ada, maka “not-found.html” akan ditampilkan. Method deleteStudent yang ada di kelas studentService diimplementasikan di kelas StudentServiceDatabase yang kemudian memanggil method deleteStudent yang ada di kelas StudentMapper. Method ini berisi query delete yang ditandai dengan annotation @Delete yang dilanjutkan dengan query delete. Akhirnya, data mahasiswa terhapus.

* Update

Proses update dimulai dari method update yang ada pada kelas StudentController. Ketika ingin update mahasiswa dengan npm tertentu maka akan dicek terlebih dahulu apakah mahasiswa dengan npm yang dimaksud ada. Jika ada, ditambahkan StudentModel sebagai attribute model untuk data-datanya di render di view “form-update”. Jika tidak ada, maka ditampilkan view “not-found”. Kemudian, ketika user klik tombol Update maka method updateSubmit akan dieksekusi. Di dalam method ini dibuat instansiasi kelas StudentModel yang datanya diperoleh dari data yang dimasukkan ke form update. Kemudian dieksekusi query untuk mengupdate data yang dimaksud di kelas StudentService. Kemudian view “success-update” muncul.

* Update dengan object sebagai parameter

Method updateStudent dengan parameter @ModelAttribute StudentModel student kemudian digunakan sebagai parameter untuk mengeksekusi method updateStudent yang ada di kelas StudentService. Kemudian, method updateStudent yang ada di kelas StudentServiceDatabase dieksekusi. Lalu, dieksekusi query update yang ada di kelas StudentMapper sehingga data mahasiswa yang diinginkan berubah.

Jawaban pertanyaan:

1. Validasi input dapat dilakukan dengan mengecek apakah nama dan gpa student (instansiasli kelas StudentModel) merupakan string kosong (student.getName().equals(“”)), dan jika tidak, barulah panggil method update.
2. Form submit menggunakan POST method karena kita tidak dapat memanggil button untuk submit form melalui URL (karakteristik GET method adalah passing parameter melalui URL)
3. Ya, satu method dapat menerima request GET dan POST sekaligus