

Pertanyaan:

1. Caranya adalah dengan menambahkan anotasi pada Model objek. Sebagai contoh, @NotNull untuk mencegah nilai null dikirim oleh form yang kosong. Untuk beberapa kasus tentunya validasi diperlukan
2. Form submit menggunakan method post karena method tersebut tidak dapat terlihat di address bar. Method get terlihat di address bar sehingga hasil input bisa dilihat. Perlu penanganan berbeda. Agar method post bisa berjalan, perlu adanya request method post di header controller.
3. Bisa.

Penjelasan

Untuk menambahkan fitur delete sesuai dengan MVC, saya membuat interface di Mapper dan Service, berupa header method deleteStudent() dengan parameter npm. Kemudian di Mapper saya menambahkan anotasi untuk Delete di database sesuai dengan nilai npm yang didapat, setelah itu saya membuat method deleteStudent() dengan parameter npm di ServiceStudent dimana ini memanggil method deleteStudent yang ada di Mapper, setelah itu saya membuat Request untuk mapping Controller (yang caranya tidak jauh berbeda dengan tutorial kemarin) dimana jika npm ditemukan maka data mahasiswa akan dihapus.

Untuk menambahkan update, langkah-langkah yang saya lakukan hampir sama dengan menambahkan delete, yaitu membuat interface di Mapper dan Service berupa updateStudent() dengan parameter objek Student. Kemudian di Mapper menambahkan anotasi Update di database sesuai dengan data yang didapat dari form. Kemudian saya membuat form yang mengirim data yang ada lewat method post, lalu membuat controller yang tidak jauh berbeda dari controller untuk add yang sudah disediakan oleh template, yaitu membuat objek student baru dengan parameter sesuai dengan data yang sudah di-passing dari form kemudian memanggil method updateStudent(student).

Untuk menggunakan objek sebagai parameter, langkah yang saya ubah hanyalah mengganti request parameter yang sudah ada dengan sebuah objek student, lalu mengubah sintaks dari laman form agar menyesuaikan, kemudian saya tinggal memanggil method updateStudent(student) dan deleteStudent().