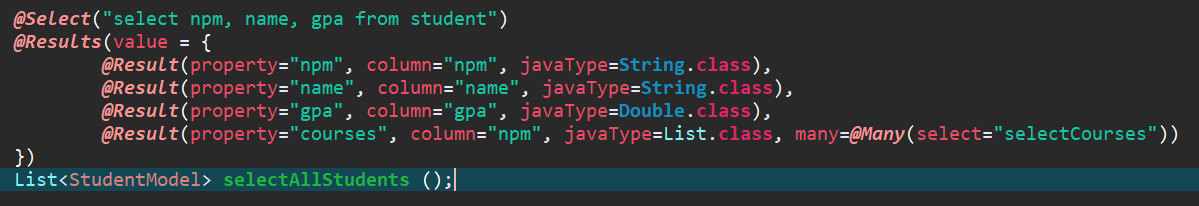
Write-up Tutorial 5

Pada tutorial kali ini saya mempelajari beberapa hal berikut; Dengan menggunakan MyBatis, query yang dieksekusi dapat langsung disimpan pada class model dengan menggunakan annotation **@Results** yang memetakan hasil query ke class model dan **@Resut** yang memetakan hasil query per kolom ke atribut class model. Lalu, annotation **@Many** untuk menyimpan data hasil query yang lebih dari satu.

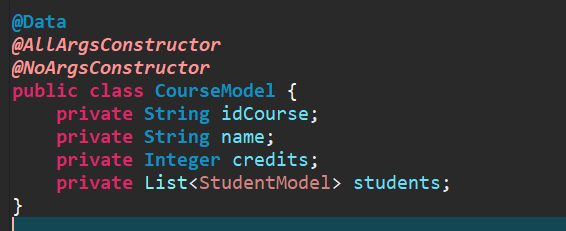
Pada latihan nomor 1, saya mengubah method **selectAllStudents** pada kelas **StudentMapper** menjadi seperti berikut :



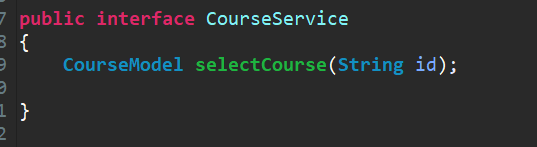
Saya menambahkan annotation **@Results** dan **@Result**. **@Results** berfungsi untuk menyimpan hasil query ke class model. **@Result** berfungsi untuk menyimpan hasil query per kolom ke atribut dari class model. Walaupun return type dari method ini adalah List<StudentModel> tetapi class modelnya tetap StudentModel karena query yang dijalankan mengembalikan object StudentModel.

Pada latihan nomor 2, saya membuat;

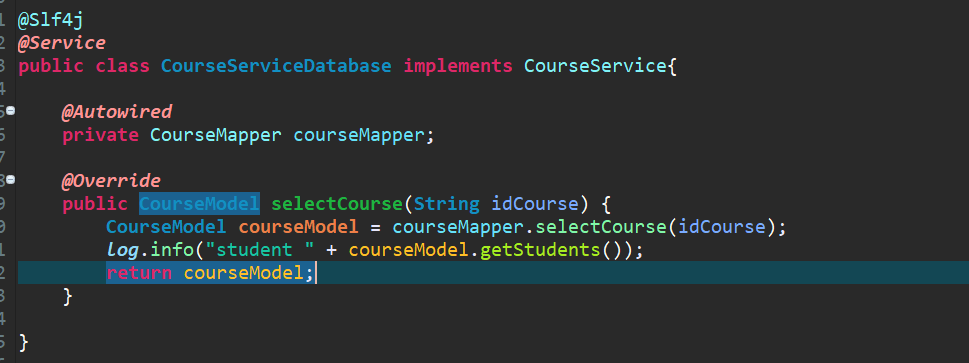
* Class **CourseModel**



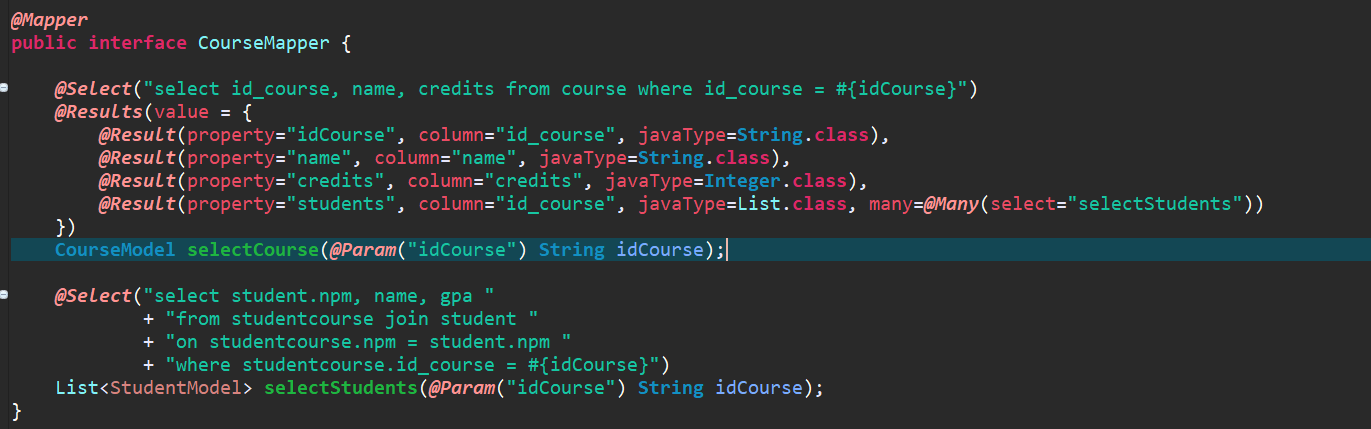
* Interface **CourseService**



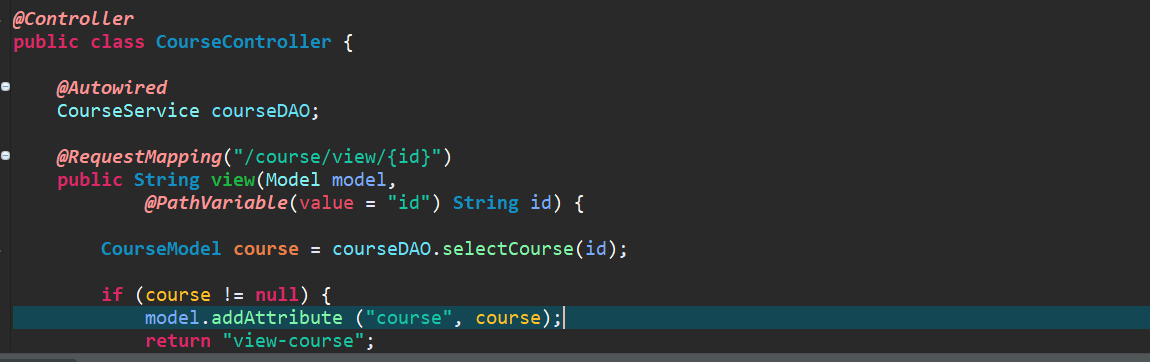
* Class **CourseServiceDatabase**



* Interface **CourseMapper**



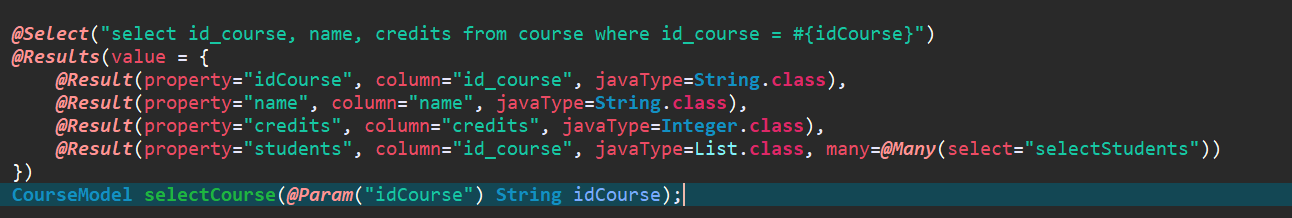
* Class **CourseController**



Class-class dan interface-interface diatas saya buat karena konteks aplikasi sudah berbeda yaitu “course”, dapat dilihat dari mapping yang digunakan ( /course/view/{id} ). Class-class dan interface-interface tersebut memiliki kegunaan yang sama seperti yang dikerjakan pada tutorial sebelumnya hanya saja menggunakan model **CourseModel.**

Method-method dan file-file berikut ini adalah yang saya buat untuk mengerjakan soal nomor 2:

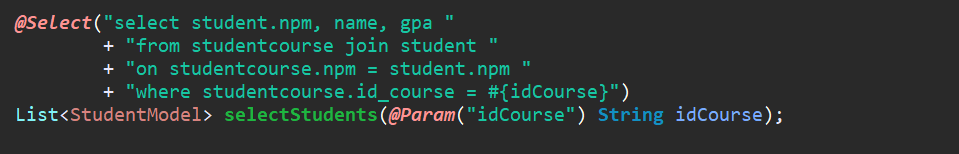
* Method **selectCourse** pada Interface **CourseMapper.**



Method ini menjalankan query untuk mengambil data **course** dengan id\_course tertentu. Method ini memetakan hasil query ke class model **CourseModel.** Untuk atribut students,

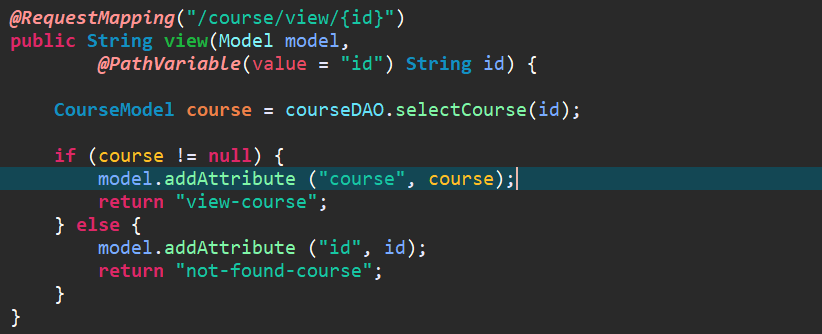
memanggil method **selectStudents** pada Interface **CourseMapper.**

* Method **selectStudents** pada Interface **CourseMapper.**



Method ini menjalankan query untuk mengambil student-student yang mengambil mata kuliah dengan id\_course tertentu.

* Method **view** pada class **CourseController**



Method ini menghandle mapping ( /course/view/{id} ), pertama-tama method akan memanggil method **selecCourse** pada service untuk mendapatkan course dengan id tertentu. Jika ada maka menampilkan view-course.html tetapi jika tidak ada maka menampilkan not-found-course.html.

* View-course.html dan not-found-course.html

