

Nama: Muhammad Hasby Rosyadi
NPM : 1306386642
APAP – A

Pada tutorial 5 ini, saya mempelajari cara penggunaan *query* pada database dan menyelesaikan metode ManyToMany yang menurut saya membutuhkan cara khusus dibanding OneToOne ataupun OneToMany. Metode query dilakukan pada mapper yang mana hasil dari mapper akan dilempar pada controller.

• Pembuatan method `selectAllStudent`

Karena bersifat ManyToMany maka dibutuhkan tabel penghubung untuk dapat merelasikan 2 tabel antara student dan course

```
@Select("select course.id_course, name, credits " +  
        "from studentcourse join course " +  
        "on studentcourse.id_course = course.id_course " +  
        "where studentcourse.npm = #{npm}")  
List<CourseModel> selectCourses (@Param("npm") String npm);
```

pada kode diatas data data disimpan dalam list CourseModel yang difilter berdasarkan npm yang diterima. selanjutnya

Pada method `selectAllStudent` ditambahkan kode seperti dibawah

```
@Select("select npm, name, gpa from student")  
@Results(value = {  
    @Result(property="npm", column="npm"),  
    @Result(property="name", column="name"),  
    @Result(property="gpa", column="gpa"),  
    @Result(property="courses", column="npm", javaType=List.class,  
            many=@Many(select="selectCourses"))  
})  
List<StudentModel> selectAllStudents ();
```

Anotasi results akan memetakan property yang ada pada class model dengan column yang ada pada database. Pada `select="selectCourses"` merupakan penghubung antara 2 tabel yaitu student dan course. Sehingga `selectCourses` ini akan menampung data pada CourseModel yang sudah difilter sebelumnya.

Pada viewall ditambahkan kode seperti dibawah

```
<div th:each="student,iterationStatus: ${students}">  
    <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>  
    <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>  
    <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>  
    <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>  
    <h3>Kuliah yang diambil</h3>  
    <ul th:each="course,iterationStatus: ${student.courses}">  
        <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">Nama kuliah - X SKS</li>  
    </ul>  
    <hr/>  
</div>
```

pada bagian kuliah yang diambil iterasi dilakukan pada tabel course yang diambil dari modelStudent yang memiliki listCourse. Dari iterasi tersebut kita dapat mengambil nama kuliah dan sks pada kuliah tersebut.

Nama: Muhammad Hasby Rosyadi
NPM : 1306386642
APAP – A

- **Menambahkan View pada Course**

Pada studentMapper kita membuat list baru untuk menampung field ManyToMany, karena kita membutuhkan id_course sebagai inputan sehingga kita membuat perubahan pada selectCourses sebelumnya menjadi kode dibawah ini.

```
@Select("select student.npm, name " +  
        "from studentcourse join student " +  
        "on studentcourse.npm = student.npm " +  
        "where studentcourse.id_course = #{id_course}")  
List<StudentModel> selectCourseByStudent (@Param("id_course") String id_course);
```

selanjutnya buat mapper yang akan kita gunakan pada controller dengan menggabungkan selectCourseByStudent, kode seperti dibawah ini

```
@Select("select id_course, name, credits from course where id_course = #{id_course}")  
@Results(value = {  
    @Result(property="idCourse", column="id_course"),  
    @Result(property="name", column="name"),  
    @Result(property="credits", column="credits"),  
    @Result(property="students", column="id_course", javaType=List.class,  
        many=@Many(select="selectCourseByStudent"))  
})  
CourseModel selectCourse(@Param("id_course") String id_course);
```

tambahkan kode dibawah pada class interface

```
CourseModel selectCourse (String id_course);
```

tambahkan kode dibawah pada StudentServiceDatabase yang diimplement pada class interface

```
@Override  
public CourseModel selectCourse(String id_course) {  
    // TODO Auto-generated method stub  
    log.info("Course berhasil");  
    return studentMapper.selectCourse(id_course);  
}
```

Selanjutnya buat method baru untuk view pada course

```
@RequestMapping("/course/view/{id_course}")  
public String viewCourse (Model model,  
    @PathVariable(value = "id_course") String id_course)  
{  
    CourseModel course = studentDAO.selectCourse(id_course);  
  
    if (course != null) {  
        model.addAttribute ("course", course);  
        return "view-course";  
    } else {  
        model.addAttribute ("id", id_course);  
        return "not-found-course";  
    }  
}
```

Nama: Muhammad Hasby Rosyadi
NPM : 1306386642
APAP – A

buat html baru untuk memberikan output yang sesuai

```
<body>
  <h3 th:text="'ID = ' + ${course.idCourse}">Course ID</h3>
  <h3 th:text="'Name = ' + ${course.name}">Course Name</h3>
  <h3 th:text="'SKS = ' + ${course.credits}">Course SKS</h3>

  <h3>Mahasiswa yang Mengambil</h3>
  <ul th:each = "student, iterationStatus: ${course.students}">
    <li th:text = "${student.npm} + '-' + ${student.name}">NPM - Nama Mahasiswa</li>
  </ul>
</body>
```

buat html baru untuk memberikan pesan error apabila kode matkul tidak ditemukan

```
<body>
<h1>Course not found</h1>
  <h3 th:text="'Course = ' + ${id}">Course id_course</h3>
</body>
```