



**Shafa Maharani P – 1606955031 – SI Ekstensi**

## **Ringkasan Materi Tutorial 6**

Pada tutorial 6 ini mempelajari *unit testing* dan *load testing*. *Unit testing* berguna untuk melakukan testing pada program yang telah dibuat berdasarkan masing-masing fungsi maupun method. Testing dilakukan agar setiap fungsi dan method yang diimplementasikan sudah sesuai dengan output yang diharapkan. Untuk melakukan *unit testing*, Spring Boot memiliki 2 dependencies, yaitu JUnit dan Mockito. JUnit merupakan sebuah testing framework yang berfokus pada pengecekan class dan method, sedangkan Mockito merupakan sebuah dependencies untuk menciptakan dummy dalam pengecekan sehingga programmer dapat memanipulasi objek untuk dijadikan test case. *Load testing* merupakan sebuah proses dalam *software engineering* untuk melakukan pengujian respon dari sebuah sistem atau aplikasi dengan memberikan request tertentu.

## **Latihan Unit Testing**

1. Tambahkan *code* berikut pada `StudentServiceDatabaseTest` dan **jelaskan apa yang terjadi pada *method* tersebut berikut dengan hasil *testing* -nya !** ( *Screenshot* hanya opsional dan tidak akan mempengaruhi nilai)

```
// Given
List<StudentModel> studentModels = new ArrayList<>();
StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
studentModels.add(studentModel);
```

Membuat sebuah list of `StudentModel` dengan nama `studentModels` berisi object `StudentModel` dengan nama `studentModel`. Dan dibuat juga List of `StudentModel` dengan nama `checks` yang diisi object `StudentModel` dengan nama object `check`. Selanjutnya Mockito akan memanipulasi fungsi `selectAllStudent` dengan return `studentModels` dengan tipe data list of `StudentModel`.

```
// When
List<StudentModel> test = studentService.selectAllStudents();
```

Melakukan `selectAllStudent()` dan hasilnya akan ditampung pada variable `test` dengan tipe data List of `StudentModel`.

```
// Then
assertThat(test, notNullValue()); // Check if Not Null
assertThat(test.isEmpty(), equalTo(false)); // Check kalo ngga kosong
assertThat(test.size(), equalTo(1)); // Check if Size same
assertThat(test, equalTo(checks)); // Check kalo konten sama
```

**assertThat(test, notNullValue());** digunakan untuk mengecek apakah variable test memiliki nilai (not null).

**assertThat(test.isEmpty(), equalTo(false));** digunakan untuk mengecek isi dari test, dimana test ada isinya. Artinya hasil isEmpty() mengembalikan nilai false.

**assertThat(test.size(), equalTo(1));** digunakan untuk melihat ukuran dari test, dalam contoh tersebut mengetahui panjang / ukuran dari test yaitu 1.

**assertThat(test, equalTo(checks));** digunakan untuk mengecek test apakah sama dengan list checks.

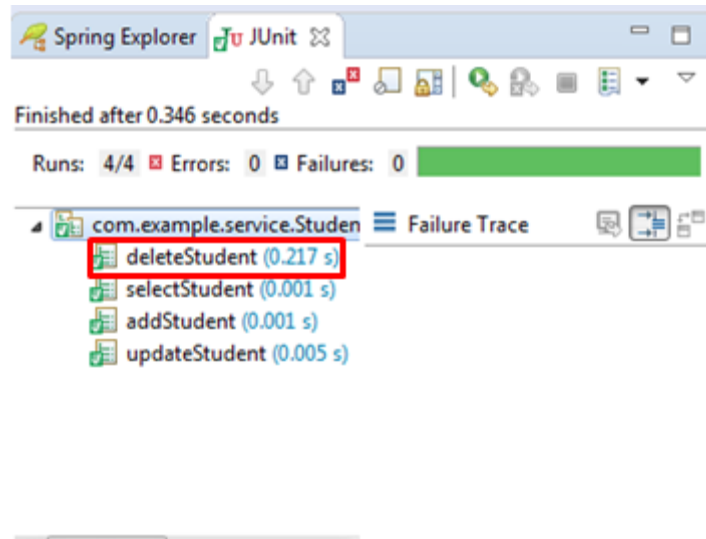
2. Tambahkan *code* berikut pada StudentServiceDatabaseTest dan **jelaskan apa yang terjadi pada *method* tersebut berikut dengan hasil *testing* -nya !** ( *Screenshot* hanya opsional dan tidak akan mempengaruhi nilai)

```
@Test
public void addStudent() {
    // Given
    StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
    StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);
    BDDMockito.given(studentService.addStudent(studentModel)).willReturn(true);
    // When
    boolean test = studentService.addStudent(studentModel);
    // Then
    BDDMockito.then(studentMapper).should().addStudent(check);
    assertThat(test, equalTo(true)); // Check if Same
}
```

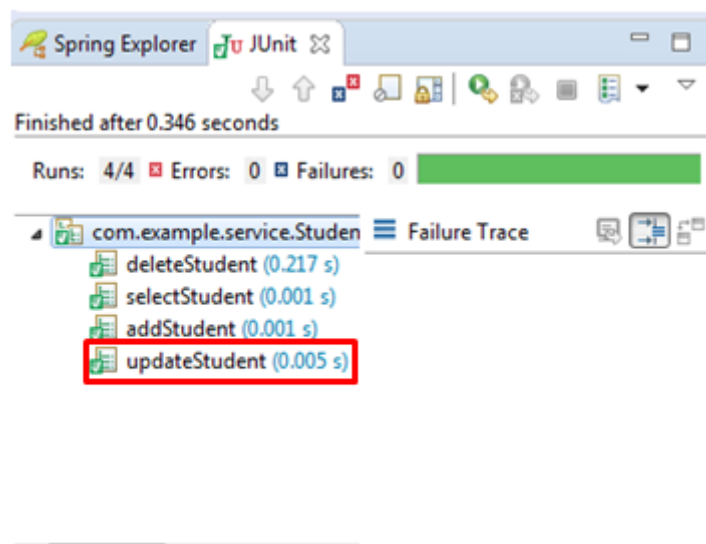
Method tersebut menjelaskan tentang method addStudent dimana pertama kali dibuat object StudentModel yang diisi dengan data student. Selanjutnya dibuatlah object dengan nama check yang akan diisi oleh data student. Lalu akan dipanggil method AddStudent pada studentService yang akan mengembalikan nilai true. Mockito akan memeriksa nilai dari test apakah bernilai true atau false.

3. Lakukan *testing terhadap method delete student dan update student* ! (cukup lampirkan *screenshot* akhir dari *testing* )

-deleteStudent



-updateStudent

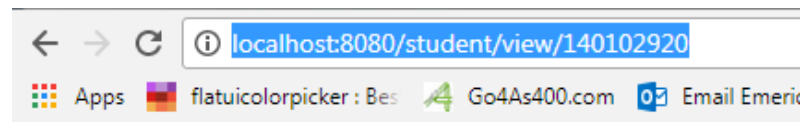




## Latihan Load Testing

- jmeter\_selectStudentBerhasil

Student/view/140102920

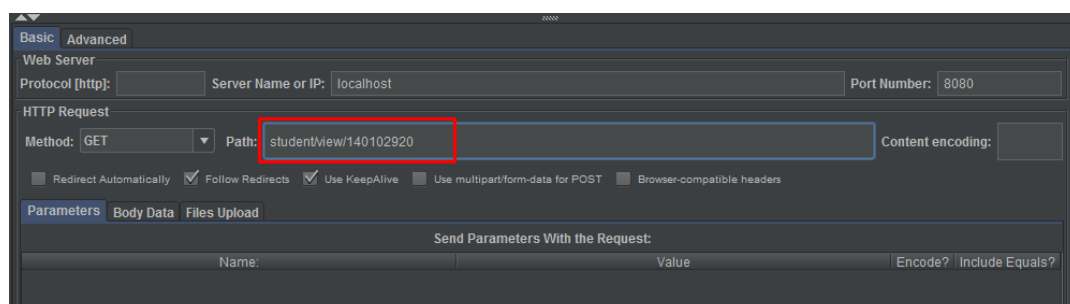


**NPM = 140102920**

**Name = Mutiarani Ramadhan**

**GPA = 3.95**

Gambar dibawah merupakan pengaturan dari Thread Group dengan pathnya berisi Student/view/140102920



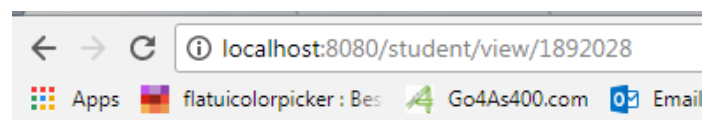


Berikut merupakan hasil dari load testing selectStudentBerhasil (npm student ditemukan)

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	20:42:44.665	Thread Group 1-1	HTTP Request	105	✓	383	140	105	54
2	20:42:45.165	Thread Group 1-2	HTTP Request	34	✓	383	140	33	2
3	20:42:45.665	Thread Group 1-3	HTTP Request	57	✓	383	140	57	2
4	20:42:46.166	Thread Group 1-4	HTTP Request	24	✓	383	140	24	2
5	20:42:46.667	Thread Group 1-5	HTTP Request	58	✓	383	140	58	2
6	20:42:47.166	Thread Group 1-6	HTTP Request	115	✓	383	140	115	2
7	20:42:47.667	Thread Group 1-7	HTTP Request	28	✓	383	140	27	2
8	20:42:48.168	Thread Group 1-8	HTTP Request	148	✓	383	140	147	2
9	20:42:48.667	Thread Group 1-9	HTTP Request	25	✓	383	140	25	2
10	20:42:49.167	Thread Group 1...	HTTP Request	73	✓	383	140	73	2
11	20:42:49.667	Thread Group 1...	HTTP Request	16	✓	383	140	16	1
12	20:42:50.168	Thread Group 1...	HTTP Request	49	✓	383	140	48	2
13	20:42:50.668	Thread Group 1...	HTTP Request	76	✓	383	140	75	2
14	20:42:51.168	Thread Group 1...	HTTP Request	24	✓	383	140	23	2
15	20:42:51.668	Thread Group 1...	HTTP Request	25	✓	383	140	25	2
16	20:42:52.169	Thread Group 1...	HTTP Request	89	✓	383	140	89	2
17	20:42:52.669	Thread Group 1...	HTTP Request	28	✓	383	140	28	2
18	20:42:53.169	Thread Group 1...	HTTP Request	24	✓	383	140	24	2
19	20:42:53.668	Thread Group 1...	HTTP Request	89	✓	383	140	89	2
20	20:42:54.169	Thread Group 1...	HTTP Request	26	✓	383	140	26	4

- jmeter\_selectStudentGagal

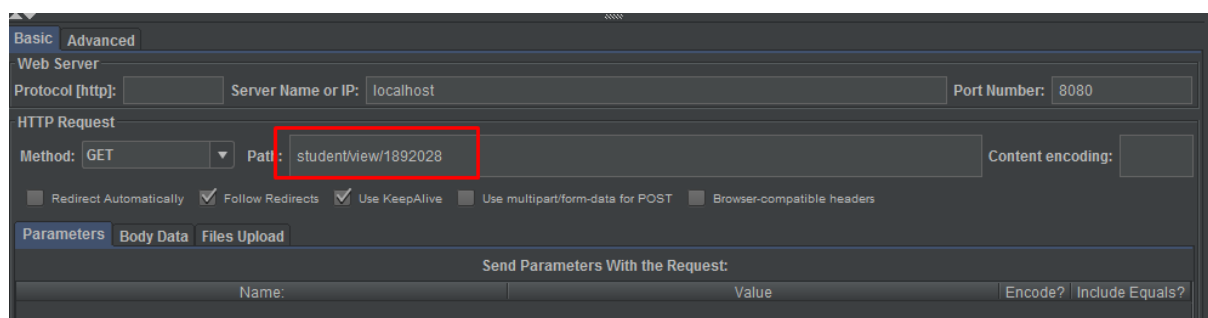
Student/view/1892028



**Student not found**

**NPM = 1892028**

Gambar berikut merupakan pengaturan dari Thread Group dan path yang berisi Student/view/1892028





Berikut merupakan hasil dari load testing selectStudentBerhasil (npm student tidak ditemukan)

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	20:44:42.409	Thread Group 1-1 HTTP Request		74	✓	339	138	74	7
2	20:44:42.909	Thread Group 1-2 HTTP Request		65	✓	339	138	65	1
3	20:44:43.409	Thread Group 1-3 HTTP Request		23	✓	339	138	23	2
4	20:44:43.909	Thread Group 1-4 HTTP Request		85	✓	339	138	84	2
5	20:44:44.410	Thread Group 1-5 HTTP Request		39	✓	339	138	39	3
6	20:44:44.908	Thread Group 1-6 HTTP Request		23	✓	339	138	23	2
7	20:44:45.409	Thread Group 1-7 HTTP Request		75	✓	339	138	75	2
8	20:44:45.910	Thread Group 1-8 HTTP Request		25	✓	339	138	25	3
9	20:44:46.411	Thread Group 1-9 HTTP Request		28	✓	339	138	28	2
10	20:44:46.908	Thread Group 1... HTTP Request		24	✓	339	138	24	3
11	20:44:47.410	Thread Group 1... HTTP Request		24	✓	339	138	24	2
12	20:44:47.910	Thread Group 1... HTTP Request		27	✓	339	138	27	3
13	20:44:48.410	Thread Group 1... HTTP Request		23	✓	339	138	23	3
14	20:44:48.908	Thread Group 1... HTTP Request		17	✓	339	138	17	2
15	20:44:49.410	Thread Group 1... HTTP Request		26	✓	339	138	26	3
16	20:44:49.912	Thread Group 1... HTTP Request		23	✓	339	138	23	2
17	20:44:50.411	Thread Group 1... HTTP Request		24	✓	339	138	24	2
18	20:44:50.909	Thread Group 1... HTTP Request		113	✓	339	138	113	2
19	20:44:51.409	Thread Group 1... HTTP Request		23	✓	339	138	23	2
20	20:44:51.911	Thread Group 1... HTTP Request		23	✓	339	138	23	2

- jmeter\_selectAllStudents

Student/viewall

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	20:33:32.732	Thread Group 1-2 HTTP Request		6228	✓	2672272	133	3550	25
2	20:33:32.241	Thread Group 1-1 HTTP Request		8259	✓	2672438	133	4532	2
3	20:33:33.748	Thread Group 1-4 HTTP Request		13619	✓	2672415	133	6973	23
4	20:33:34.264	Thread Group 1-5 HTTP Request		13438	✓	2672824	133	8191	30
5	20:33:33.251	Thread Group 1-3 HTTP Request		14813	✓	2672480	133	7628	2
6	20:33:35.263	Thread Group 1-7 HTTP Request		19018	✓	2672336	133	11973	33
7	20:33:35.765	Thread Group 1-8 HTTP Request		19316	✓	2672518	133	11397	32
8	20:33:34.747	Thread Group 1-6 HTTP Request		21473	✓	2672295	133	16633	17
9	20:33:36.265	Thread Group 1-9 HTTP Request		20236	✓	2672312	133	15740	47
10	20:33:37.265	Thread Group 1... HTTP Request		19251	✓	2672423	133	15079	12
11	20:33:36.765	Thread Group 1... HTTP Request		19993	✓	2672551	133	15581	71
12	20:33:38.766	Thread Group 1... HTTP Request		18041	✓	2672240	133	13854	26
13	20:33:37.766	Thread Group 1... HTTP Request		19060	✓	2672232	133	15340	14
14	20:33:40.264	Thread Group 1... HTTP Request		16767	✓	2672248	133	12713	34
15	20:33:41.264	Thread Group 1... HTTP Request		15810	✓	2672264	133	12117	8
16	20:33:40.764	Thread Group 1... HTTP Request		16337	✓	2672240	133	12575	79
17	20:33:39.270	Thread Group 1... HTTP Request		17869	✓	2672240	133	14083	34
18	20:33:41.764	Thread Group 1... HTTP Request		15384	✓	2672240	133	11618	11
19	20:33:38.265	Thread Group 1... HTTP Request		18898	✓	2672232	133	15117	2
20	20:33:39.764	Thread Group 1... HTTP Request		17402	✓	2672248	133	13453	11