

Tutorial 6

Unit Testing dan Load Testing Web App Sederhana

Review

Dalam pemrograman komputer, unit testing merupakan sebuah metode untuk melakukan testing pada program yang telah kita buat berdasarkan masing-masing fungsi ataupun method. Testing ini dilakukan agar setiap fungsi dan method yang kita implementasi apakah sudah sesuai dengan output yang diharapkan. JUnit merupakan sebuah testing framework yang terfokus pada pengecekan class dan method. JUnit dapat membantu programmer dalam melakukan pengecekan secara otomatis untuk setiap class dan method. Lalu Mockito merupakan sebuah dependencies untuk menciptakan dummy dalam pengecekan sehingga programmer dapat memanipulasi objek untuk dijadikan test case.

Tutorial ini mempelajari bagaimana melakukan unit testing dan load testing pada program yang telah dibuat. Unit testing dilakukan dengan menggunakan JUnit & Mockito. Testing tersebut dilakukan pada method-method yang sudah dibuat pada tutorial sebelumnya. Pada load testing menggunakan JMeter dengan memanggil method-method dan juga melakukan load pada program. Method-method yang dilakukan testing adalah addStudent, updateStudent, deleteStudent, dan selectAllStudent.

Latihan Unit Testing

Tambahkan method selectAllStudents berikut pada StudentServiceDatabaseTest

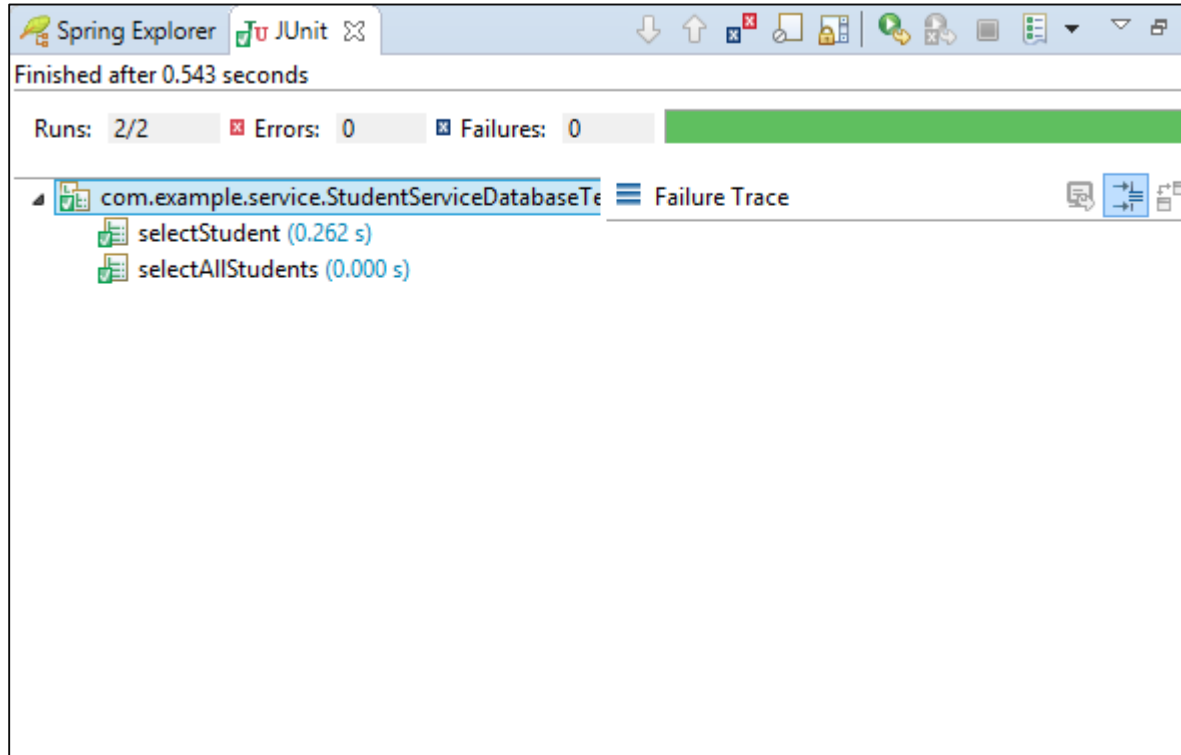
```
@Test
public void selectAllStudents() {
    // Given
    List<StudentModel> studentModels = new ArrayList<>();
    StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
    studentModels.add(studentModel);

    List<StudentModel> checks = new ArrayList<>();
    StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);
    checks.add(check);

    BDDMockito.given(studentMapper.selectAllStudents()).willReturn(studentModels);
    // When
    List<StudentModel> test = studentService.selectAllStudents();

    // 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
}
```

Gambar tersebut menjelaskan method `selectAllStudent` yang dimulai dari menginisialisasi object berupa list dengan nama `studentModel` dan check lalu Mockito akan memanipulasi fungsi `selectAllStudent` dengan return `studentModels`. Kemudian akan dilakukan testing terhadap method `selectAllStudents` selanjutnya akan dilakukan pemeriksaan jika tidak null/ kosong jika ukuran sama dan konten juga sama.



Tambahkan method `addStudent` berikut pada `StudentServiceDatabaseTest`

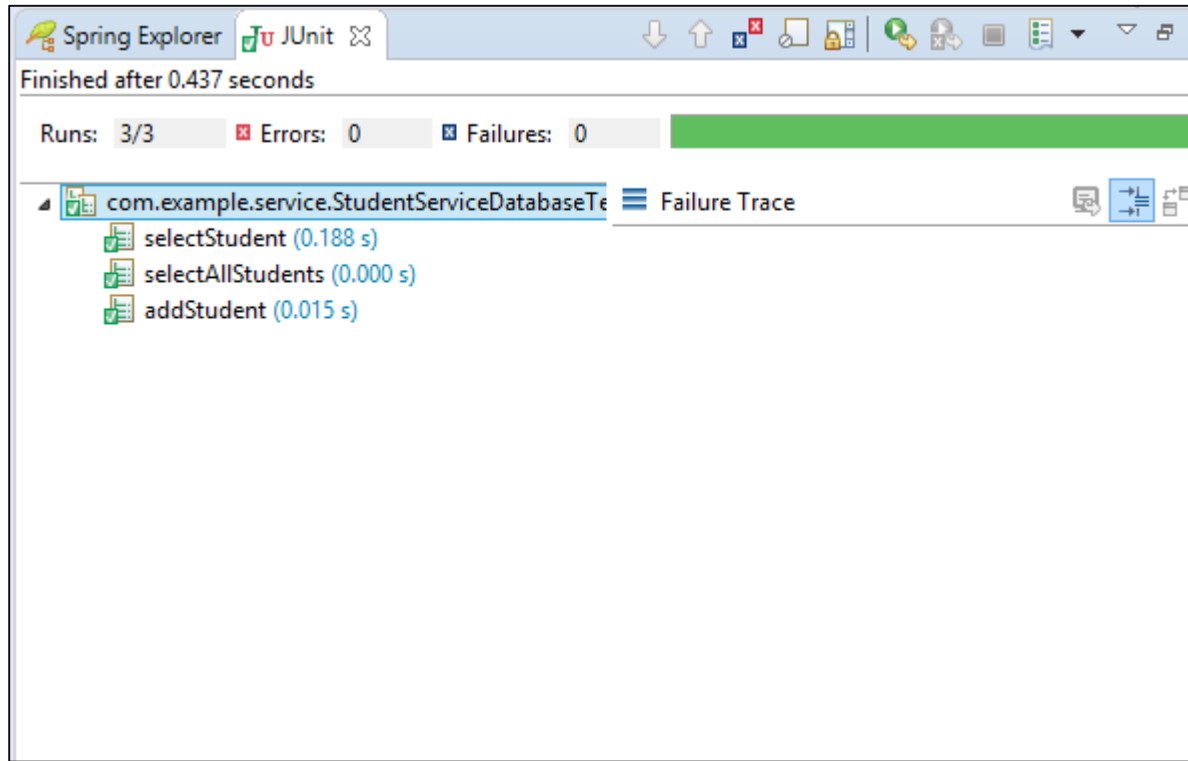
```
@Test
public void addStudent() {
    // Given
    StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
    StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);
    BDDMockito.given(studentService.addStudent(studentModel)).willReturn(true);

    // When
    boolean test = studentService.addStudent(studentModel);

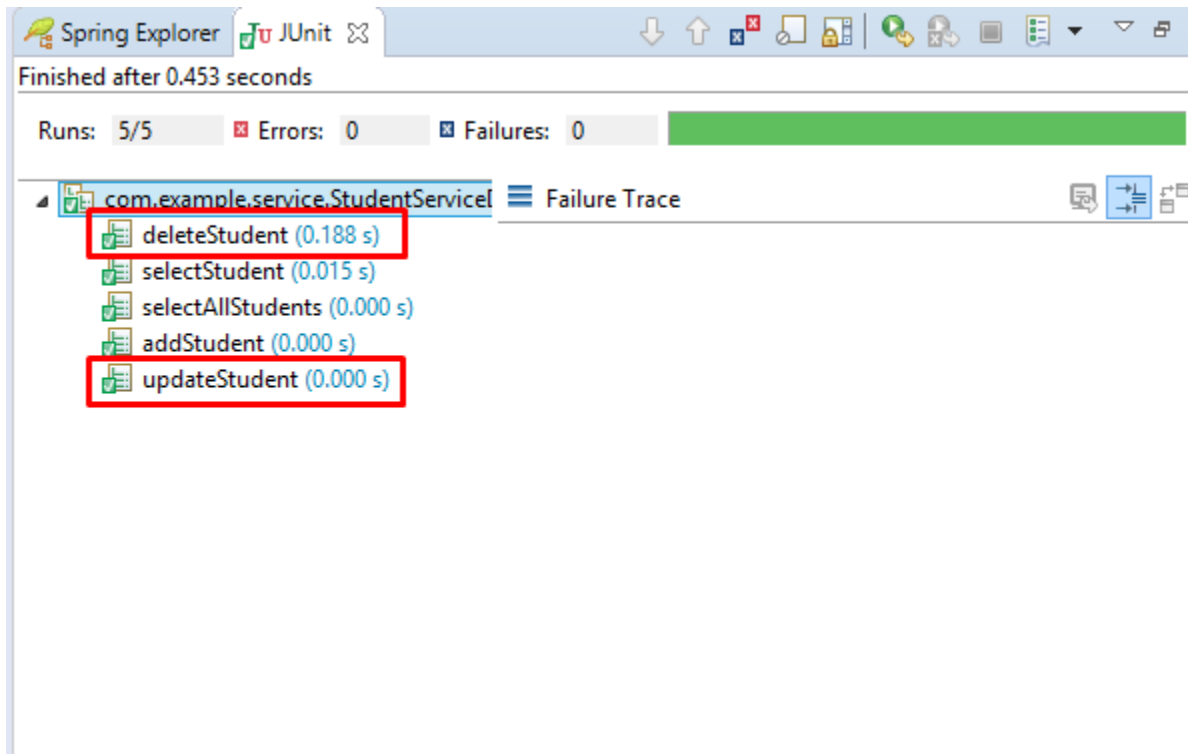
    // Then
    BDDMockito.then(studentMapper).should().addStudent(check);
    assertEquals(test, equalTo(true)); // Check if Same
}
```

Dalam 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, selanjutnya akan dipanggil method `AddStudent` pada

studentService yang akan mengembalikan nilai true. Selanjutnya Mockito akan memeriksa nilai dari test apakah bernilai true atau false.



Testing terhadap method delete student dan update student



Latihan Load Testing

o jmeter_selectStudentBerhasil

Gambar berikut merupakan pengaturan dari Thread Group dan hasil dari load testing yang dijalankan yang dilakukan untuk melakukan Load Testing dengan menggunakan JMeter untuk selectStudentBerhasil.

HTTP Request

Name: HTTP Request

Comments:

Basic **Advanced**

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8080

HTTP Request

Method: GET Path: studentView/1606954703 Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers

Parameters **Body Data** **Files Upload**

Send Parameters With the Request:

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☒ Successes **Configure**

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	22:39:16.502	Thread Group 2-1	HTTP Request	11	✓	341	140	11	2
2	22:39:16.468	Thread Group 1-1	HTTP Request	57	✓	22456	133	57	2
3	22:39:16.525	Thread Group 1-1	HTTP Request	9	✓	379	141	9	0
4	22:39:16.602	Thread Group 2-2	HTTP Request	5	✓	341	140	5	1
5	22:39:16.669	Thread Group 1-2	HTTP Request	14	✓	22456	133	13	1
6	22:39:16.683	Thread Group 1-2	HTTP Request	5	✓	379	141	5	0
7	22:39:16.716	Thread Group 2-3	HTTP Request	5	✓	341	140	5	1
8	22:39:16.810	Thread Group 2-4	HTTP Request	4	✓	341	140	4	1
9	22:39:16.873	Thread Group 1-3	HTTP Request	18	✓	22456	133	18	0
10	22:39:16.891	Thread Group 1-3	HTTP Request	5	✓	379	141	5	0
11	22:39:16.904	Thread Group 2-5	HTTP Request	5	✓	341	140	5	1
12	22:39:17.013	Thread Group 2-6	HTTP Request	6	✓	341	140	6	1
13	22:39:17.103	Thread Group 2-7	HTTP Request	5	✓	341	140	5	1
14	22:39:17.119	Thread Group 1-4	HTTP Request	14	✓	22456	133	14	1
15	22:39:17.137	Thread Group 1-4	HTTP Request	5	✓	379	141	5	0
16	22:39:17.204	Thread Group 2-8	HTTP Request	7	✓	341	140	7	3
17	22:39:17.298	Thread Group 1-5	HTTP Request	14	✓	22456	133	13	1
18	22:39:17.312	Thread Group 1-5	HTTP Request	5	✓	379	141	5	0
19	22:39:17.314	Thread Group 2-9	HTTP Request	6	✓	341	140	6	1
20	22:39:17.406	Thread Group 2...	HTTP Request	5	✓	341	140	5	1
21	22:39:17.502	Thread Group 2...	HTTP Request	5	✓	341	140	5	1
22	22:39:17.500	Thread Group 1-6	HTTP Request	16	✓	22456	133	15	1
23	22:39:17.516	Thread Group 1-6	HTTP Request	5	✓	379	141	5	0
24	22:39:17.601	Thread Group 2...	HTTP Request	6	✓	341	140	6	1
25	22:39:17.686	Thread Group 1-7	HTTP Request	16	✓	22456	133	16	1
26	22:39:17.702	Thread Group 1-7	HTTP Request	5	✓	379	141	5	0

o jmeter_selectStudentGagal

Gambar berikut merupakan pengaturan dari Thread Group dan hasil dari load testing yang dijalankan yang dilakukan untuk melakukan Load Testing dengan menggunakan JMeter untuk selectStudentGagal.

HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8080

HTTP Request

Method: GET Path: studentView/1606954714 Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☐ Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	22:41:00.068	Thread Group 1-1	HTTP Request	50		22456	133	48	4
2	22:41:00.113	Thread Group 2-1	HTTP Request	9		341	140	9	1
3	22:41:00.118	Thread Group 1-1	HTTP Request	8		379	141	8	0
4	22:41:00.123	Thread Group 2-1	HTTP Request	12		342	141	12	0
5	22:41:00.224	Thread Group 2-2	HTTP Request	6		341	140	5	1
6	22:41:00.230	Thread Group 2-2	HTTP Request	4		342	141	4	0
7	22:41:00.318	Thread Group 2-3	HTTP Request	5		341	140	5	1
8	22:41:00.324	Thread Group 2-3	HTTP Request	4		342	141	4	0
9	22:41:00.318	Thread Group 1-2	HTTP Request	20		22456	133	19	1
10	22:41:00.338	Thread Group 1-2	HTTP Request	5		379	141	5	0
11	22:41:00.412	Thread Group 2-4	HTTP Request	7		341	140	6	2
12	22:41:00.419	Thread Group 2-4	HTTP Request	4		342	141	4	0
13	22:41:00.521	Thread Group 2-5	HTTP Request	5		341	140	5	1
14	22:41:00.527	Thread Group 2-5	HTTP Request	4		342	141	4	0
15	22:41:00.537	Thread Group 1-3	HTTP Request	19		22456	133	19	1
16	22:41:00.557	Thread Group 1-3	HTTP Request	5		379	141	5	0
17	22:41:00.615	Thread Group 2-6	HTTP Request	5		341	140	5	1
18	22:41:00.620	Thread Group 2-6	HTTP Request	5		342	141	5	0
19	22:41:00.677	Thread Group 1-4	HTTP Request	19		22456	133	18	1
20	22:41:00.696	Thread Group 1-4	HTTP Request	5		379	141	5	0
21	22:41:00.724	Thread Group 2-7	HTTP Request	6		341	140	5	1
22	22:41:00.730	Thread Group 2-7	HTTP Request	4		342	141	4	0
23	22:41:00.818	Thread Group 2-8	HTTP Request	5		341	140	5	1
24	22:41:00.823	Thread Group 2-8	HTTP Request	5		342	141	5	0
25	22:41:00.880	Thread Group 1-5	HTTP Request	15		22456	133	14	1
26	22:41:00.895	Thread Group 1-5	HTTP Request	5		379	141	5	0

o jmeter_selectAllStudents

Gambar berikut merupakan pengaturan dari Thread Group dan hasil dari load testing yang dijalankan yang dilakukan untuk melakukan Load Testing dengan menggunakan JMeter untuk selectAllStudents.

HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8080

HTTP Request

Method: GET Path: student/viewall Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: ☐ Errors ☐ Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	23:02:23.795	Thread Group 1-2	HTTP Request	272	✓	341	140	272	1
2	23:02:23.687	Thread Group 1-1	HTTP Request	386	✓	341	140	386	2
3	23:02:24.067	Thread Group 1-2	HTTP Request	8	✓	342	141	7	0
4	23:02:24.072	Thread Group 1-1	HTTP Request	5	✓	342	141	5	0
5	23:02:23.889	Thread Group 1-3	HTTP Request	199	✓	341	140	199	1
6	23:02:24.087	Thread Group 1-3	HTTP Request	5	✓	342	141	5	0
7	23:02:24.094	Thread Group 1-5	HTTP Request	5	✓	341	140	4	1
8	23:02:23.986	Thread Group 1-4	HTTP Request	119	✓	341	140	119	1
9	23:02:24.099	Thread Group 1-5	HTTP Request	6	✓	342	141	5	0
10	23:02:24.104	Thread Group 1-4	HTTP Request	7	✓	342	141	7	0
11	23:02:24.188	Thread Group 1-6	HTTP Request	17	✓	341	140	17	1
12	23:02:24.205	Thread Group 1-6	HTTP Request	986	✓	342	141	986	0
13	23:02:24.111	Thread Group 1-4	HTTP Request	2291	✓	2374382	133	2181	0
14	23:02:24.297	Thread Group 1-7	HTTP Request	3411	✓	341	140	3411	1
15	23:02:24.391	Thread Group 1-8	HTTP Request	4625	✓	341	140	4625	1
16	23:02:24.105	Thread Group 1-5	HTTP Request	5273	✓	2374382	133	5154	0
17	23:02:24.884	Thread Group 1...	HTTP Request	4901	✓	341	140	4901	1
18	23:02:24.594	Thread Group 1...	HTTP Request	5192	✓	341	140	5192	1
19	23:02:24.784	Thread Group 1...	HTTP Request	5003	✓	341	140	5002	1
20	23:02:24.984	Thread Group 1...	HTTP Request	4806	✓	341	140	4806	1
21	23:02:25.085	Thread Group 1...	HTTP Request	4707	✓	341	140	4707	1
22	23:02:24.688	Thread Group 1...	HTTP Request	5104	✓	341	140	5104	2
23	23:02:24.485	Thread Group 1-9	HTTP Request	5311	✓	341	140	5311	0
24	23:02:25.185	Thread Group 1...	HTTP Request	4615	✓	341	140	4615	1
25	23:02:25.284	Thread Group 1...	HTTP Request	4521	✓	341	140	4521	1
26	23:02:25.385	Thread Group 1...	HTTP Request	4424	✓	341	140	4424	1