

## Tutorial Presentation Layer Design

### A. Ringkasan Materi

- Dalam tutorial ini mempelajari mengenai *unit testing* dan load testing. *Unit testing* dilakukan dengan menggunakan Junit & Mockito. Testing tersebut dilakukan pada method-method yang sudah dibuat pada tutorial sebelumnya. Pada load testing digunakan JMeter untuk melakukan load testing. Method-method yang dilakukan testing adalah addStudent, updateStudent, deleteStudent, dan selectAllStudent.

### B. Tutorial

#### I. Latihan Unit Testing

1.

```
@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 naga kosong
    assertThat(test.size(), equalTo(1)); // Check if Size same
    assertThat(test, equalTo(checks)); // Check kalo konten sama
}
```

- pada Given diatas menginisialisasikan object berupa List dengan nama studentModels & checks, kemudian BDDMockito akan memanipulasi fungsi selectAllStudent dengan return studentModels.
- Kemudian When akan melakukan pengetesan method selectAllStudents
- Selanjutnya Then melakukan pengecekan jika tidak null, jika tidak kosong, jika ukurannya sama dan jika kontennya juga sama.

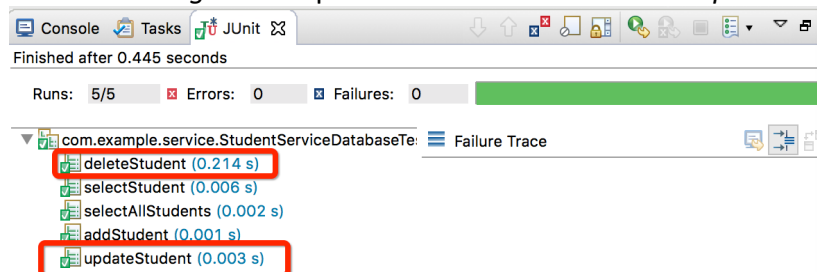
2.

```
@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
}
```

Nama : Budi Indrawan  
NPM : 1606954754

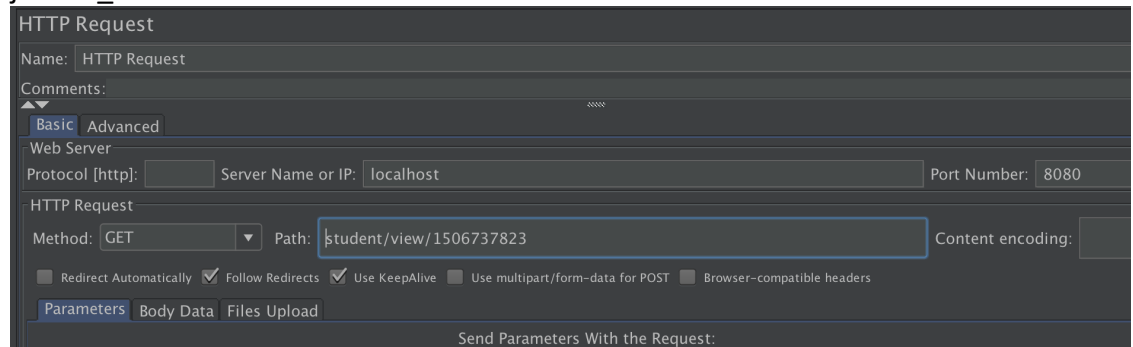
- pada Given diatas menginisialisasikan object berupa List dengan nama studentModels & checks, kemudian BDDMockito akan memanipulasi fungsi addStudent dengan return studentModels.
- Kemudian When akan melakukan pengetesan method addStudents
- Selanjutnya Then melakukan pengecekan jika sama dengan variable test

### 3. *testing terhadap method delete student dan update student !*



## II. Latihan Load Testing

- jmeter\_selectStudentBerhasil



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
68	16:21:13.471	Thread Grou...	HTTP Request	29	Success	1390	141	28	3
69	16:21:13.574	Thread Grou...	HTTP Request	13	Success	1390	141	13	1
70	16:21:13.672	Thread Grou...	HTTP Request	18	Success	1390	141	18	4
71	16:21:13.772	Thread Grou...	HTTP Request	62	Success	1390	141	62	5
72	16:21:13.878	Thread Grou...	HTTP Request	22	Success	1390	141	22	7
73	16:21:13.973	Thread Grou...	HTTP Request	18	Success	1390	141	18	3
74	16:21:14.074	Thread Grou...	HTTP Request	23	Success	1390	141	22	0
75	16:21:14.170	Thread Grou...	HTTP Request	14	Success	1390	141	14	3
76	16:21:14.271	Thread Grou...	HTTP Request	28	Success	1390	141	27	3
77	16:21:14.372	Thread Grou...	HTTP Request	12	Success	1390	141	12	2
78	16:21:14.475	Thread Grou...	HTTP Request	18	Success	1390	141	18	5
79	16:21:14.578	Thread Grou...	HTTP Request	13	Success	1390	141	13	2
80	16:21:14.674	Thread Grou...	HTTP Request	13	Success	1390	141	13	3
81	16:21:14.778	Thread Grou...	HTTP Request	40	Success	1390	141	40	4
82	16:21:14.875	Thread Grou...	HTTP Request	11	Success	1390	141	11	2
83	16:21:14.978	Thread Grou...	HTTP Request	12	Success	1390	141	12	1
84	16:21:15.073	Thread Grou...	HTTP Request	14	Success	1390	141	14	1
85	16:21:15.178	Thread Grou...	HTTP Request	19	Success	1390	141	17	3
86	16:21:15.279	Thread Grou...	HTTP Request	24	Success	1390	141	23	3
87	16:21:15.378	Thread Grou...	HTTP Request	14	Success	1390	141	13	1
88	16:21:15.478	Thread Grou...	HTTP Request	15	Success	1390	141	15	2
89	16:21:15.578	Thread Grou...	HTTP Request	25	Success	1390	141	25	1
90	16:21:15.679	Thread Grou...	HTTP Request	30	Success	1390	141	30	2
91	16:21:15.784	Thread Grou...	HTTP Request	31	Success	1390	141	31	4
92	16:21:15.880	Thread Grou...	HTTP Request	24	Success	1390	141	24	3
93	16:21:15.980	Thread Grou...	HTTP Request	32	Success	1390	141	32	5
94	16:21:16.081	Thread Grou...	HTTP Request	16	Success	1390	141	16	1
95	16:21:16.186	Thread Grou...	HTTP Request	44	Success	1390	141	44	7
96	16:21:16.287	Thread Grou...	HTTP Request	31	Success	1390	141	30	2
97	16:21:16.388	Thread Grou...	HTTP Request	31	Success	1390	141	31	5
98	16:21:16.485	Thread Grou...	HTTP Request	13	Success	1390	141	13	1
99	16:21:16.588	Thread Grou...	HTTP Request	14	Success	1390	141	14	2
100	16:21:16.689	Thread Grou...	HTTP Request	10	Success	1390	141	10	2

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 10 Average 21 Deviation 13

Nama : Budi Indrawan  
NPM : 1606954754

- jmeter\_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: student/view/1506737822

Content encoding:

☐ Redirect Automatically

☒ Follow Redirects

☒ Use KeepAlive

☐ Use multipart/form-data for POST

☐ Browser-compatible headers

Parameters

Body Data

Files Upload

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(ms)
68	17:55:16.284	Thread Group ...	HTTP Request	7		1358	141	7	0
69	17:55:16.384	Thread Group ...	HTTP Request	9		1358	141	9	1
70	17:55:16.481	Thread Group ...	HTTP Request	12		1358	141	11	1
71	17:55:16.586	Thread Group ...	HTTP Request	8		1358	141	8	2
72	17:55:16.686	Thread Group ...	HTTP Request	5		1358	141	5	0
73	17:55:16.786	Thread Group ...	HTTP Request	28		1358	141	28	1
74	17:55:16.888	Thread Group ...	HTTP Request	13		1358	141	13	2
75	17:55:16.991	Thread Group ...	HTTP Request	14		1358	141	13	1
76	17:55:17.095	Thread Group ...	HTTP Request	32		1358	141	32	2
77	17:55:17.193	Thread Group ...	HTTP Request	16		1358	141	16	2
78	17:55:17.293	Thread Group ...	HTTP Request	15		1358	141	15	1
79	17:55:17.395	Thread Group ...	HTTP Request	23		1358	141	22	4
80	17:55:17.490	Thread Group ...	HTTP Request	11		1358	141	11	1
81	17:55:17.593	Thread Group ...	HTTP Request	16		1358	141	16	2
82	17:55:17.695	Thread Group ...	HTTP Request	12		1358	141	12	1
83	17:55:17.793	Thread Group ...	HTTP Request	23		1358	141	22	2
84	17:55:17.892	Thread Group ...	HTTP Request	18		1358	141	18	1
85	17:55:17.995	Thread Group ...	HTTP Request	12		1358	141	12	1
86	17:55:18.092	Thread Group ...	HTTP Request	13		1358	141	13	0
87	17:55:18.193	Thread Group ...	HTTP Request	12		1358	141	12	1
88	17:55:18.292	Thread Group ...	HTTP Request	21		1358	141	20	1
89	17:55:18.393	Thread Group ...	HTTP Request	20		1358	141	20	2
90	17:55:18.493	Thread Group ...	HTTP Request	15		1358	141	15	0
91	17:55:18.594	Thread Group ...	HTTP Request	11		1358	141	11	1
92	17:55:18.695	Thread Group ...	HTTP Request	11		1358	141	11	1
93	17:55:18.797	Thread Group ...	HTTP Request	12		1358	141	12	1
94	17:55:18.899	Thread Group ...	HTTP Request	24		1358	141	23	5
95	17:55:18.997	Thread Group ...	HTTP Request	12		1358	141	12	3
96	17:55:19.095	Thread Group ...	HTTP Request	23		1358	141	23	2
97	17:55:19.195	Thread Group ...	HTTP Request	14		1358	141	14	1
98	17:55:19.296	Thread Group ...	HTTP Request	18		1358	141	18	1
99	17:55:19.398	Thread Group ...	HTTP Request	15		1358	141	15	2
100	17:55:19.494	Thread Group ...	HTTP Request	17		1358	141	16	1

☒ Scroll automatically?

☐ Child samples?

No of Samples 100

Latest Sample 17

Average 14

Deviation 5

Nama : Budi Indrawan  
NPM : 1606954754

- `jmeter_selectAllStudents`

Dikarenakan kemampuan hardware yang tidak dapat memenuhi permintaan untuk load user sebanyak 100, maka saya sendiri menurunkan kembali loadnya menjadi 50.

### HTTP Request

Name: HTTP Request

Comments:

BasicAdvanced

Web Server

Protocol [http]:Server Name or IP:localhostPort Number:8080

HTTP Request

Method:GETPath:student/viewallContent encoding:

☐ Redirect Automatically

☒ Follow Redirects

☒ Use KeepAlive

☐ Use multipart/form-data for POST

☐ Browser-compatible headers

ParametersBody DataFiles Upload

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
-------	-------	---------	-----------------

View Results in Table
00:01:21
0 / 0:50

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...
18	20:47:06.330	Thread Grou...	HTTP Request	71850	✓	2878998	133	71219	0
19	20:47:04.305	Thread Grou...	HTTP Request	74023	✓	2878998	133	73205	1
20	20:47:05.527	Thread Grou...	HTTP Request	72820	✓	2878998	133	72173	1
21	20:47:06.529	Thread Grou...	HTTP Request	72133	✓	2878998	133	71475	4
22	20:47:06.127	Thread Grou...	HTTP Request	75789	✓	2878998	133	72152	1
23	20:47:10.342	Thread Grou...	HTTP Request	71924	✓	2878998	133	71607	1
24	20:47:08.343	Thread Grou...	HTTP Request	73972	✓	2878998	133	73552	1
25	20:47:08.544	Thread Grou...	HTTP Request	73859	✓	2878998	133	73614	1
26	20:47:07.132	Thread Grou...	HTTP Request	75306	✓	2878998	133	71578	1
27	20:47:10.545	Thread Grou...	HTTP Request	71921	✓	2878998	133	71506	2
28	20:47:08.940	Thread Grou...	HTTP Request	73631	✓	2878998	133	73567	1
29	20:47:07.334	Thread Grou...	HTTP Request	75273	✓	2878998	133	74703	3
30	20:47:09.345	Thread Grou...	HTTP Request	73450	✓	2878998	133	73128	1
31	20:47:07.936	Thread Grou...	HTTP Request	74926	✓	2878998	133	74509	1
32	20:47:09.546	Thread Grou...	HTTP Request	73392	✓	2878998	133	73025	1
33	20:47:06.729	Thread Grou...	HTTP Request	76237	✓	2878998	133	75815	1
34	20:47:05.724	Thread Grou...	HTTP Request	77245	✓	2878998	133	76756	1
35	20:47:09.142	Thread Grou...	HTTP Request	73927	✓	2878998	133	73642	1
36	20:47:07.538	Thread Grou...	HTTP Request	75540	✓	2878998	133	74905	1
37	20:47:06.931	Thread Grou...	HTTP Request	76297	✓	2878998	133	75968	1
38	20:47:09.749	Thread Grou...	HTTP Request	73582	✓	2878998	133	73361	1
39	20:47:07.739	Thread Grou...	HTTP Request	75604	✓	2878998	133	75177	1
40	20:47:08.742	Thread Grou...	HTTP Request	74739	✓	2878998	133	74306	1
41	20:47:05.930	Thread Grou...	HTTP Request	77608	✓	2878998	133	77153	0
42	20:47:10.142	Thread Grou...	HTTP Request	73566	✓	2878998	133	73137	0
43	20:47:10.746	Thread Grou...	HTTP Request	73022	✓	2878998	133	72798	1
44	20:47:11.746	Thread Grou...	HTTP Request	72058	✓	2878998	133	71708	1
45	20:47:09.946	Thread Grou...	HTTP Request	73883	✓	2878998	133	73220	1
46	20:47:11.946	Thread Grou...	HTTP Request	71896	✓	2878998	133	71639	0
47	20:47:11.144	Thread Grou...	HTTP Request	72767	✓	2878998	133	72675	1
48	20:47:11.546	Thread Grou...	HTTP Request	72375	✓	2878998	133	72240	1
49	20:47:11.345	Thread Grou...	HTTP Request	72607	✓	2878998	133	72526	1
50	20:47:10.942	Thread Grou...	HTTP Request	73056	✓	2878998	133	72996	1

☐ Scroll automatically?
☐ Child samples?

No of Samples 50
Latest Sample 73056
Average 62572
Deviation 13511