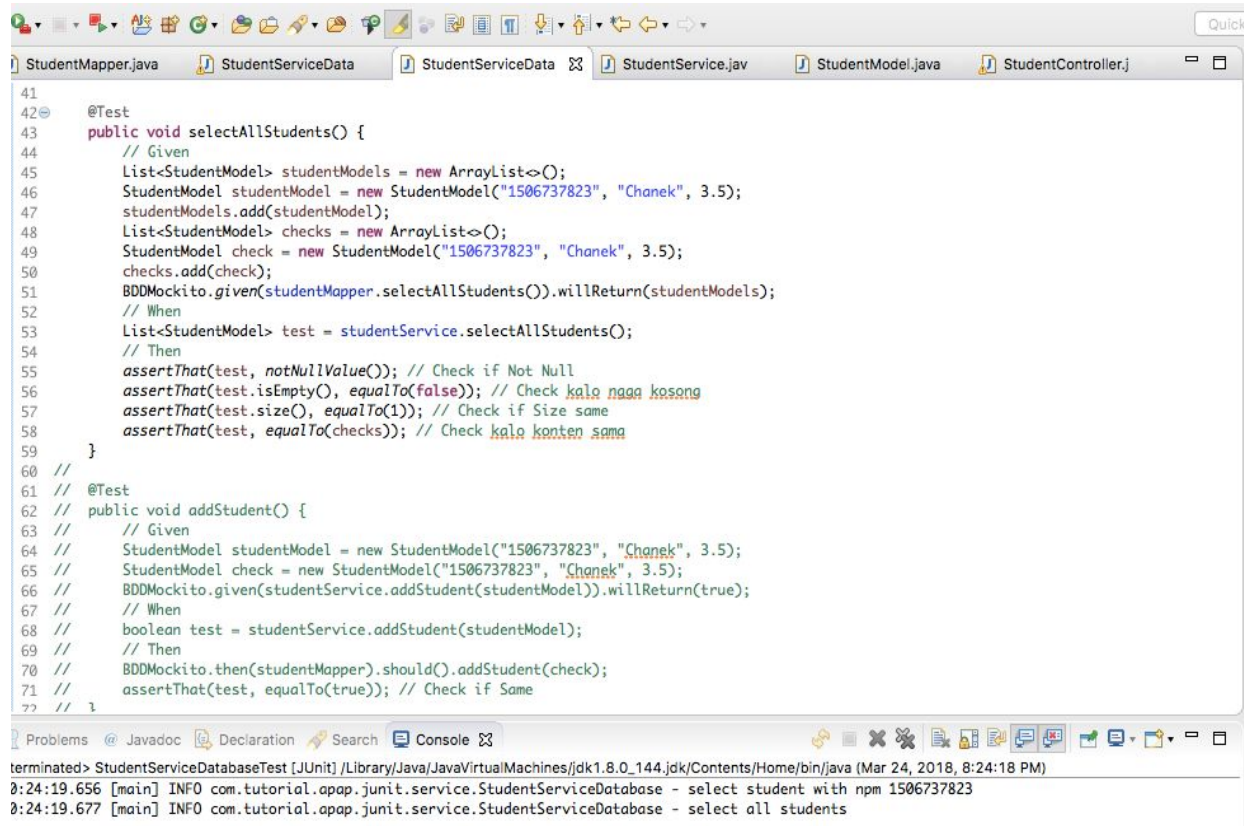


LATIHAN UNIT TESTING

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)



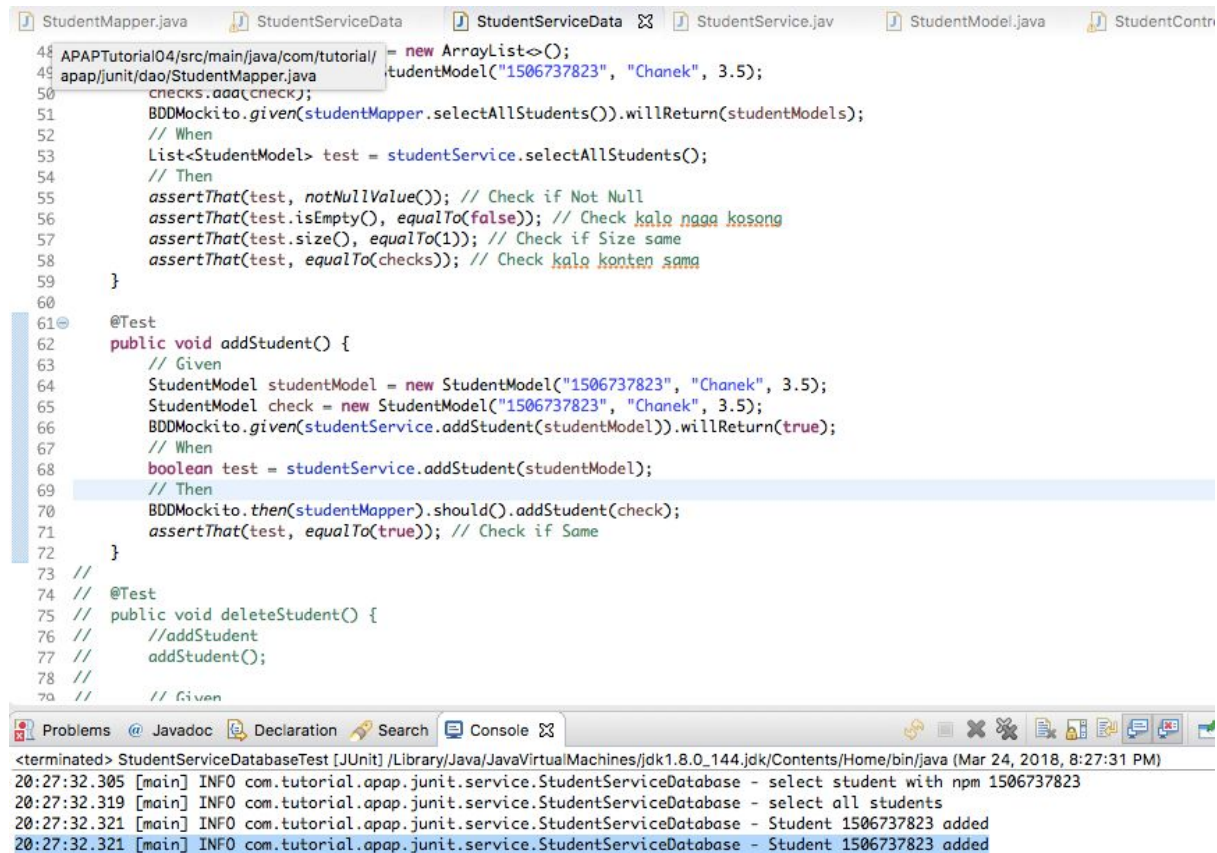
```
41
42 @Test
43 public void selectAllStudents() {
44     // Given
45     List<StudentModel> studentModels = new ArrayList<>();
46     StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
47     studentModels.add(studentModel);
48     List<StudentModel> checks = new ArrayList<>();
49     StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);
50     checks.add(check);
51     BDDMockito.given(studentMapper.selectAllStudents()).willReturn(studentModels);
52     // When
53     List<StudentModel> test = studentService.selectAllStudents();
54     // Then
55     assertThat(test, notNullValue()); // Check if Not Null
56     assertThat(test.isEmpty(), equalTo(false)); // Check kalo ngga kosong
57     assertThat(test.size(), equalTo(1)); // Check if Size same
58     assertThat(test, equalTo(checks)); // Check kalo konten sama
59 }
60 //
61 // @Test
62 // public void addStudent() {
63 //     // Given
64 //     StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
65 //     StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);
66 //     BDDMockito.given(studentService.addStudent(studentModel)).willReturn(true);
67 //     // When
68 //     boolean test = studentService.addStudent(studentModel);
69 //     // Then
70 //     BDDMockito.then(studentMapper).should().addStudent(check);
71 //     assertThat(test, equalTo(true)); // Check if Same
72 // }
```

terminated> StudentServiceDatabaseTest [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Mar 24, 2018, 8:24:18 PM)
8:24:19.656 [main] INFO com.tutorial.apap.junit.service.StudentServiceDatabase - select student with npm 1506737823
8:24:19.677 [main] INFO com.tutorial.apap.junit.service.StudentServiceDatabase - select all students

Method selectAllStudents() melakukan beberapa assertion untuk memastikan kesesuaian input dan output dari beberapa kasus yaitu :

- data Null
- data empty / kosong
- jumlah data
- ekspektasi kesesuaian data. Terlihat object **test** harus sesuai dengan object **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)



```
48 APAPTutorial04/src/main/java/com/tutorial/apap/junit/dao/StudentMapper.java = new ArrayList<>();
49 apap/junit/dao/StudentMapper.java studentModel("1506737823", "Chanek", 3.5);
50 checks.aaa(check);
51 BDDMockito.given(studentMapper.selectAllStudents()).willReturn(studentModels);
52 // When
53 List<StudentModel> test = studentService.selectAllStudents();
54 // Then
55 assertThat(test, notNullValue()); // Check if Not Null
56 assertThat(test.isEmpty(), equalTo(false)); // Check jika naga kosong
57 assertThat(test.size(), equalTo(1)); // Check if Size same
58 assertThat(test, equalTo(checks)); // Check jika konten sama
59 }
60
61 @Test
62 public void addStudent() {
63     // Given
64     StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
65     StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);
66     BDDMockito.given(studentService.addStudent(studentModel)).willReturn(true);
67     // When
68     boolean test = studentService.addStudent(studentModel);
69     // Then
70     BDDMockito.then(studentMapper).should().addStudent(check);
71     assertThat(test, equalTo(true)); // Check if Same
72 }
73 //
74 // @Test
75 // public void deleteStudent() {
76 //     //addStudent
77 //     addStudent();
78 // }
79 // // Given
```

Problems Javadoc Declaration Search Console

```
<terminated> StudentServiceDatabaseTest [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Mar 24, 2018, 8:27:31 PM)
20:27:32.305 [main] INFO com.tutorial.apap.junit.service.StudentServiceDatabase - select student with npm 1506737823
20:27:32.319 [main] INFO com.tutorial.apap.junit.service.StudentServiceDatabase - select all students
20:27:32.321 [main] INFO com.tutorial.apap.junit.service.StudentServiceDatabase - Student 1506737823 added
20:27:32.321 [main] INFO com.tutorial.apap.junit.service.StudentServiceDatabase - Student 1506737823 added
```

Pada method addStudent() hanya terdapat satu assertion :

- ekspektasi kesesuaian data. Terlihat jika proses add data student berhasil dilakukan, **return boolean true** akan dihasilkan.

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

```
75     public void deleteStudent() {  
76         //addStudent  
77         addStudent();  
78  
79         // Given  
80         StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);  
81         StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);  
82         BDDMockito.given(studentService.deleteStudent(studentModel.getNpm())).willReturn(true);  
83         // When  
84         boolean test = studentService.deleteStudent(studentModel.getNpm());  
85         // Then  
86         BDDMockito.then(studentMapper).should().addStudent(check);  
87         assertThat(test, equalTo(true)); // Check if Same  
88     }  
89  
90     @Test  
91     public void updateStudent() {  
92         //addStudent  
93         addStudent();  
94  
95         // Given  
96         StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);  
97         StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);  
98         BDDMockito.given(studentService.updateStudent(studentModel.getNpm(), studentModel.getName(), studentModel.getGpa())).willReturn(true);  
99         // When  
100        boolean test = studentService.updateStudent(studentModel.getNpm(), studentModel.getName(), studentModel.getGpa());  
101        // Then  
102        BDDMockito.then(studentMapper).should().addStudent(check);  
103        assertThat(test, equalTo(true)); // Check if Same  
104    }  
105
```

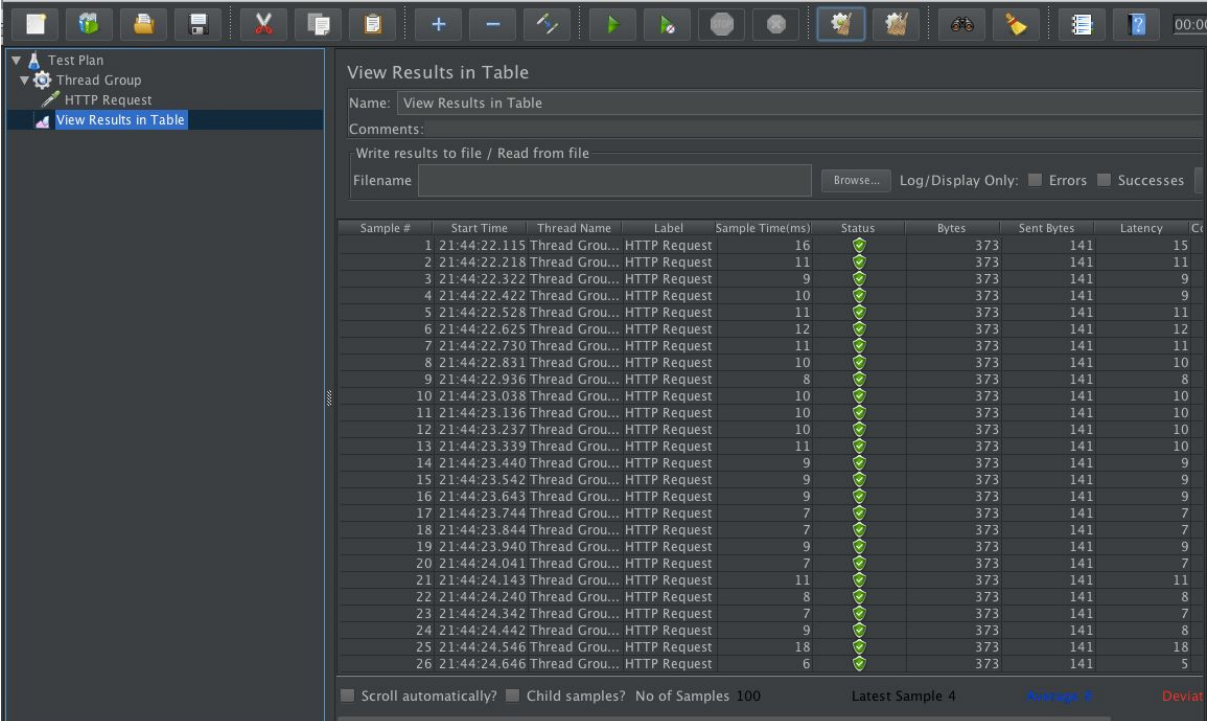
Pada method deleteStudent() dan updateStudent() hanya terdapat satu assertion :

- ekspektasi kesesuaian data. Terlihat jika proses add data student berhasil dilakukan, **return boolean true** akan dihasilkan.

LATIHAN LOAD TESTING

1. Lakukan load testing pada semua kemungkinan hasil dari fitur selectStudent() dengan kondisi berhasil ditemukan, selectStudent() dengan kondisi gagal ditemukan, dan selectAllStudents() menggunakan Apache JMeter. Gunakan Number of Threads sama dengan 100 dan Ramp-up period 10 detik. Anda berarti perlu melakukan 3x test, 1 test untuk setiap kemungkinan hasil. Screenshot window JMeter Anda. Contoh:

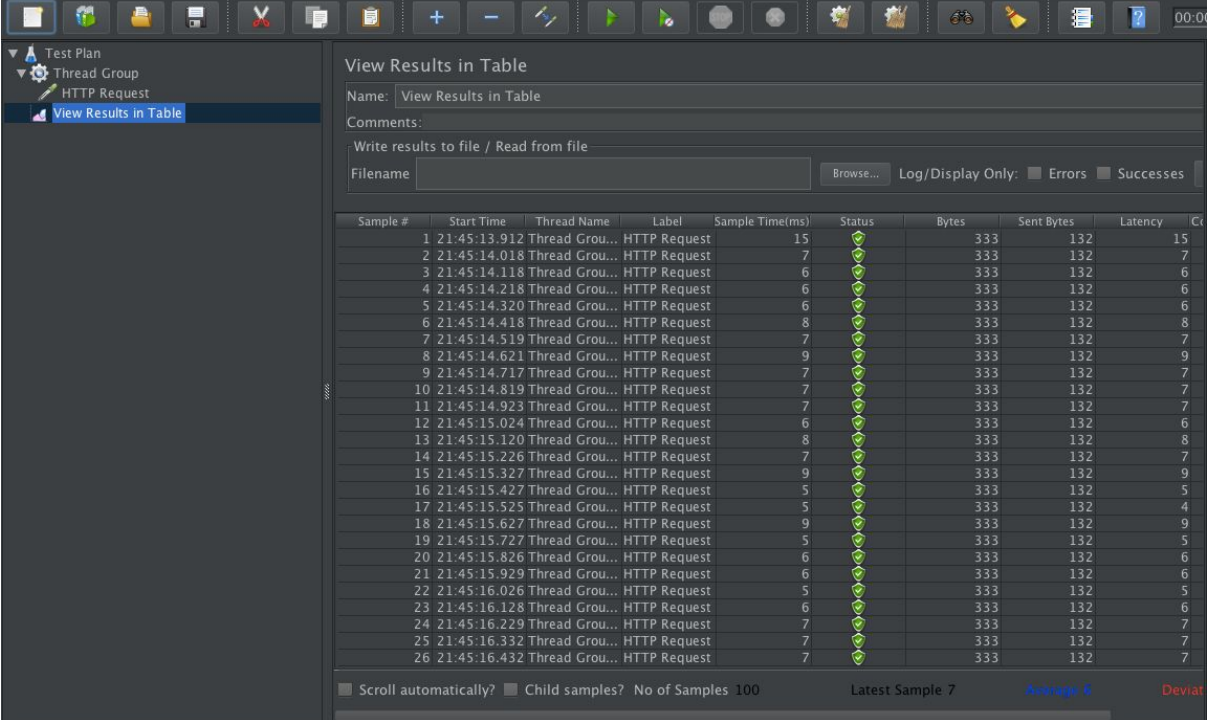
- selectStudent() - berhasil ditemukan



The screenshot shows the 'View Results in Table' window in Apache JMeter. The left sidebar shows a tree view with 'Test Plan', 'Thread Group', 'HTTP Request', and 'View Results in Table' selected. The main window displays a table of test results for 26 samples. All samples show a status of 'Success' (green checkmark) and a latency of 15ms. The table columns are: Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, Sent Bytes, Latency, and C. The bottom status bar shows 'Scroll automatically?' (checked), 'Child samples?' (checked), 'No of Samples 100', 'Latest Sample 4', 'Average 8', and 'Deviation'.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	C
1	21:44:22.115	Thread Grou...	HTTP Request	16	Success	373	141	15	
2	21:44:22.218	Thread Grou...	HTTP Request	11	Success	373	141	11	
3	21:44:22.322	Thread Grou...	HTTP Request	9	Success	373	141	9	
4	21:44:22.422	Thread Grou...	HTTP Request	10	Success	373	141	9	
5	21:44:22.528	Thread Grou...	HTTP Request	11	Success	373	141	11	
6	21:44:22.625	Thread Grou...	HTTP Request	12	Success	373	141	12	
7	21:44:22.730	Thread Grou...	HTTP Request	11	Success	373	141	11	
8	21:44:22.831	Thread Grou...	HTTP Request	10	Success	373	141	10	
9	21:44:22.936	Thread Grou...	HTTP Request	8	Success	373	141	8	
10	21:44:23.038	Thread Grou...	HTTP Request	10	Success	373	141	10	
11	21:44:23.136	Thread Grou...	HTTP Request	10	Success	373	141	10	
12	21:44:23.237	Thread Grou...	HTTP Request	10	Success	373	141	10	
13	21:44:23.339	Thread Grou...	HTTP Request	11	Success	373	141	10	
14	21:44:23.440	Thread Grou...	HTTP Request	9	Success	373	141	9	
15	21:44:23.542	Thread Grou...	HTTP Request	9	Success	373	141	9	
16	21:44:23.643	Thread Grou...	HTTP Request	9	Success	373	141	9	
17	21:44:23.744	Thread Grou...	HTTP Request	7	Success	373	141	7	
18	21:44:23.844	Thread Grou...	HTTP Request	7	Success	373	141	7	
19	21:44:23.940	Thread Grou...	HTTP Request	9	Success	373	141	9	
20	21:44:24.041	Thread Grou...	HTTP Request	7	Success	373	141	7	
21	21:44:24.143	Thread Grou...	HTTP Request	11	Success	373	141	11	
22	21:44:24.240	Thread Grou...	HTTP Request	8	Success	373	141	8	
23	21:44:24.342	Thread Grou...	HTTP Request	7	Success	373	141	7	
24	21:44:24.442	Thread Grou...	HTTP Request	9	Success	373	141	8	
25	21:44:24.546	Thread Grou...	HTTP Request	18	Success	373	141	18	
26	21:44:24.646	Thread Grou...	HTTP Request	6	Success	373	141	5	

- selectStudent() - gagal ditemukan



The screenshot shows the JMeter 'View Results in Table' window. The left sidebar displays the test plan structure: Test Plan, Thread Group, HTTP Request, and View Results in Table. The main area shows a table of 26 samples, all of which are 'HTTP Request' type and have a 'Success' status. The table columns include Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, Sent Bytes, Latency, and Count. The status column contains green checkmark icons for all samples. At the bottom, there are checkboxes for 'Scroll automatically?' and 'Child samples?', and a 'No of Samples' field set to 100. The 'Latest Sample' is 7, and the 'Average' is 6.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Count
1	21:45:13.912	Thread Grou...	HTTP Request	15	Success	333	132	15	1
2	21:45:14.018	Thread Grou...	HTTP Request	7	Success	333	132	7	1
3	21:45:14.118	Thread Grou...	HTTP Request	6	Success	333	132	6	1
4	21:45:14.218	Thread Grou...	HTTP Request	6	Success	333	132	6	1
5	21:45:14.320	Thread Grou...	HTTP Request	6	Success	333	132	6	1
6	21:45:14.418	Thread Grou...	HTTP Request	8	Success	333	132	8	1
7	21:45:14.519	Thread Grou...	HTTP Request	7	Success	333	132	7	1
8	21:45:14.621	Thread Grou...	HTTP Request	9	Success	333	132	9	1
9	21:45:14.717	Thread Grou...	HTTP Request	7	Success	333	132	7	1
10	21:45:14.819	Thread Grou...	HTTP Request	7	Success	333	132	7	1
11	21:45:14.923	Thread Grou...	HTTP Request	7	Success	333	132	7	1
12	21:45:15.024	Thread Grou...	HTTP Request	6	Success	333	132	6	1
13	21:45:15.120	Thread Grou...	HTTP Request	8	Success	333	132	8	1
14	21:45:15.226	Thread Grou...	HTTP Request	7	Success	333	132	7	1
15	21:45:15.327	Thread Grou...	HTTP Request	9	Success	333	132	9	1
16	21:45:15.427	Thread Grou...	HTTP Request	5	Success	333	132	5	1
17	21:45:15.525	Thread Grou...	HTTP Request	5	Success	333	132	4	1
18	21:45:15.627	Thread Grou...	HTTP Request	9	Success	333	132	9	1
19	21:45:15.727	Thread Grou...	HTTP Request	5	Success	333	132	5	1
20	21:45:15.826	Thread Grou...	HTTP Request	6	Success	333	132	6	1
21	21:45:15.929	Thread Grou...	HTTP Request	6	Success	333	132	6	1
22	21:45:16.026	Thread Grou...	HTTP Request	5	Success	333	132	5	1
23	21:45:16.128	Thread Grou...	HTTP Request	6	Success	333	132	6	1
24	21:45:16.229	Thread Grou...	HTTP Request	7	Success	333	132	7	1
25	21:45:16.332	Thread Grou...	HTTP Request	7	Success	333	132	7	1
26	21:45:16.432	Thread Grou...	HTTP Request	7	Success	333	132	7	1

- selectAllStudents() : Karena load terlalu lama, saya stop prosesnya

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	CPU
1	21:37:03.676	Thread Grou...	HTTP Request	117325	Error	2682	0	0	0
2	21:37:05.582	Thread Grou...	HTTP Request	115418	Error	2682	0	0	0
3	21:37:05.184	Thread Grou...	HTTP Request	115820	Error	2682	0	0	0
4	21:37:06.585	Thread Grou...	HTTP Request	114421	Error	2682	0	0	0
5	21:37:02.459	Thread Grou...	HTTP Request	118548	Error	2682	0	0	0
6	21:37:10.391	Thread Grou...	HTTP Request	110619	Error	2682	0	0	0
7	21:37:00.628	Thread Grou...	HTTP Request	120385	Error	2682	0	0	0
8	21:37:10.494	Thread Grou...	HTTP Request	110525	Error	2682	0	0	0
9	21:37:02.561	Thread Grou...	HTTP Request	118458	Error	2682	0	0	0
10	21:37:08.988	Thread Grou...	HTTP Request	112034	Error	2682	0	0	0
11	21:37:04.780	Thread Grou...	HTTP Request	116244	Error	2682	0	0	0
12	21:37:06.687	Thread Grou...	HTTP Request	114339	Error	2682	0	0	0
13	21:37:09.488	Thread Grou...	HTTP Request	111542	Error	2682	0	0	0
14	21:37:01.737	Thread Grou...	HTTP Request	119296	Error	2682	0	0	0
15	21:37:04.177	Thread Grou...	HTTP Request	116861	Error	2682	0	0	0
16	21:37:03.276	Thread Grou...	HTTP Request	117764	Error	2682	0	0	0
17	21:37:00.727	Thread Grou...	HTTP Request	120317	Error	2682	0	0	0
18	21:37:03.777	Thread Grou...	HTTP Request	117268	Error	2682	0	0	0
19	21:37:05.383	Thread Grou...	HTTP Request	115686	Error	2682	0	0	0
20	21:37:07.887	Thread Grou...	HTTP Request	113186	Error	2682	0	0	0
21	21:37:08.090	Thread Grou...	HTTP Request	112988	Error	2682	0	0	0
22	21:37:04.282	Thread Grou...	HTTP Request	116801	Error	2682	0	0	0
23	21:37:07.689	Thread Grou...	HTTP Request	113390	Error	2682	0	0	0
24	21:37:07.287	Thread Grou...	HTTP Request	113793	Error	2682	0	0	0
25	21:37:02.149	Thread Grou...	HTTP Request	118937	Error	2682	0	0	0
26	21:37:07.486	Thread Grou...	HTTP Request	113602	Error	2682	0	0	0

2. Lakukan optimasi yang mungkin dilakukan baik dari sisi kode yang Anda tuliskan maupun dari sisi database agar hasil load testing dari Apache JMeter bisa lebih cepat. Anda diperbolehkan untuk mengubah beberapa aspek di database seperti index, primary/unique key, dll namun tidak mengubah struktur tabel dan data pada database. Anda bisa mencari tahu dari sumber lain di internet untuk mengetahui apa saja hal yang bisa dioptimasi.

Optimasi yang dipilih adalah indexing dan penggunaan collation utf8mb4_default

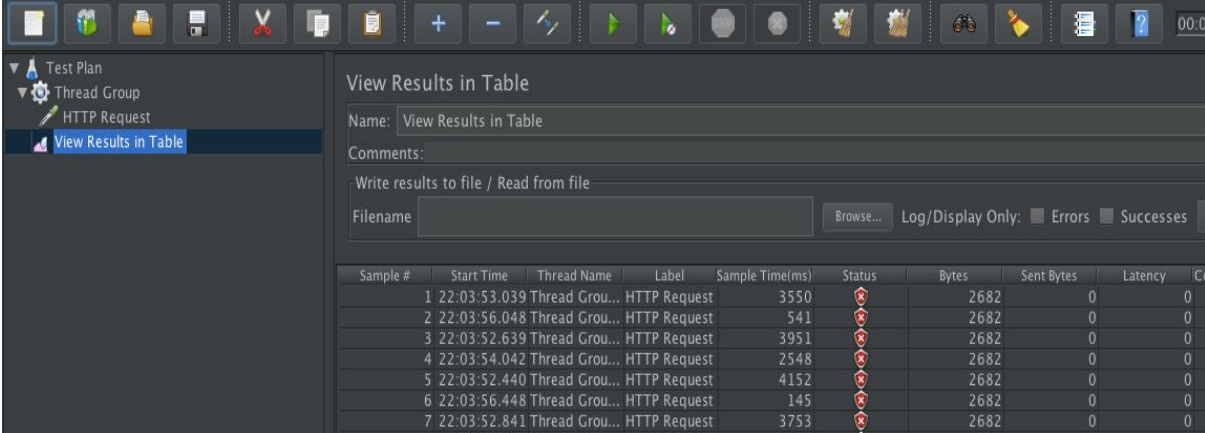
Index	Type
PRIMARY	PRIMARY
npm_index	INDEX

Index Columns	#	Order	Length
<input checked="" type="checkbox"/> npm	1	ASC	
<input type="checkbox"/> name		ASC	
<input type="checkbox"/> gpa		ASC	

Collation: utf8mb4 - default collation

3. Lakukan load testing kembali untuk semua kemungkinan hasil dengan kode dan database yang sudah Anda optimasi. Gunakan Number of Threads sama dengan 100 dan Ramp-up period 10 detik. Screenshot kembali window JMeter Anda.

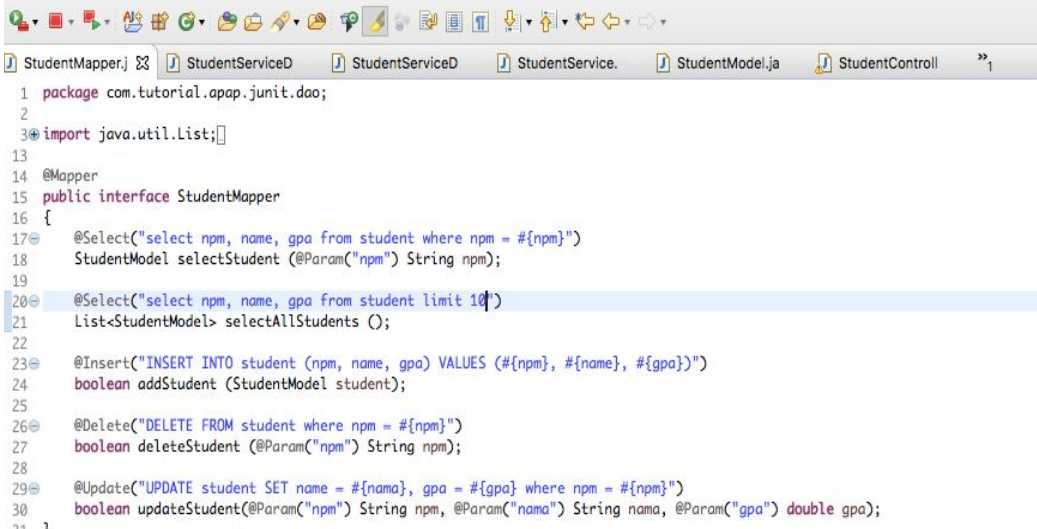
- Perbandingan selectAllStudents()
- Setelah optimasi pertama masih gagal, walaupun sudah dicoba dilakukan optimasi dengan tipe collation



The screenshot shows the JMeter 'View Results in Table' window. The table displays the following data:

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Ct
1	22:03:53.039	Thread Grou...	HTTP Request	3550	✖	2682	0	0	0
2	22:03:56.048	Thread Grou...	HTTP Request	541	✖	2682	0	0	0
3	22:03:52.639	Thread Grou...	HTTP Request	3951	✖	2682	0	0	0
4	22:03:54.042	Thread Grou...	HTTP Request	2548	✖	2682	0	0	0
5	22:03:52.440	Thread Grou...	HTTP Request	4152	✖	2682	0	0	0
6	22:03:56.448	Thread Grou...	HTTP Request	145	✖	2682	0	0	0
7	22:03:52.841	Thread Grou...	HTTP Request	3753	✖	2682	0	0	0

- Dicoba lagi optimasi ke 2 dan berhasil , tetapi query menggunakan limit



```
1 package com.tutorial.apap.junit.dao;
2
3 import java.util.List;
4
5 @Mapper
6 public interface StudentMapper
7 {
8     @Select("select npm, name, gpa from student where npm = #{npm}")
9     StudentModel selectStudent (@Param("npm") String npm);
10
11     @Select("select npm, name, gpa from student limit 10")
12     List<StudentModel> selectAllStudents ();
13
14     @Insert("INSERT INTO student (npm, name, gpa) VALUES (#{npm}, #{name}, #{gpa})")
15     boolean addStudent (StudentModel student);
16
17     @Delete("DELETE FROM student where npm = #{npm}")
18     boolean deleteStudent (@Param("npm") String npm);
19
20     @Update("UPDATE student SET name = #{nama}, gpa = #{gpa} where npm = #{npm}")
21     boolean updateStudent(@Param("npm") String npm, @Param("nama") String nama, @Param("gpa") double gpa);
22 }
```

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse...

Log/Display Only: ☐ Errors ☒ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Ct
1	22:05:21.369	Thread Grou...	HTTP Request	329	✓	7442	133	321	
2	22:05:20.966	Thread Grou...	HTTP Request	735	✓	7442	133	727	
3	22:05:21.468	Thread Grou...	HTTP Request	233	✓	7442	133	233	
4	22:05:21.068	Thread Grou...	HTTP Request	634	✓	7442	133	633	
5	22:05:21.269	Thread Grou...	HTTP Request	433	✓	7442	133	429	
6	22:05:21.169	Thread Grou...	HTTP Request	535	✓	7442	133	535	
7	22:05:20.766	Thread Grou...	HTTP Request	939	✓	7442	133	939	
8	22:05:20.871	Thread Grou...	HTTP Request	837	✓	7442	133	837	
9	22:05:21.570	Thread Grou...	HTTP Request	142	✓	7442	133	142	
10	22:05:21.675	Thread Grou...	HTTP Request	44	✓	7442	133	44	
11	22:05:21.770	Thread Grou...	HTTP Request	40	✓	7442	133	40	
12	22:05:21.870	Thread Grou...	HTTP Request	22	✓	7442	133	22	
13	22:05:21.976	Thread Grou...	HTTP Request	20	✓	7442	133	20	
14	22:05:22.071	Thread Grou...	HTTP Request	19	✓	7442	133	19	
15	22:05:22.176	Thread Grou...	HTTP Request	17	✓	7442	133	17	
16	22:05:22.276	Thread Grou...	HTTP Request	15	✓	7442	133	15	
17	22:05:22.376	Thread Grou...	HTTP Request	16	✓	7442	133	16	
18	22:05:22.476	Thread Grou...	HTTP Request	16	✓	7442	133	16	
19	22:05:22.575	Thread Grou...	HTTP Request	15	✓	7442	133	15	
20	22:05:22.677	Thread Grou...	HTTP Request	19	✓	7442	133	19	
21	22:05:22.776	Thread Grou...	HTTP Request	19	✓	7442	133	19	
22	22:05:22.873	Thread Grou...	HTTP Request	15	✓	7442	133	15	
23	22:05:22.979	Thread Grou...	HTTP Request	14	✓	7442	133	14	
24	22:05:23.076	Thread Grou...	HTTP Request	15	✓	7442	133	15	
25	22:05:23.178	Thread Grou...	HTTP Request	10	✓	7442	133	10	
26	22:05:23.279	Thread Grou...	HTTP Request	19	✓	7442	133	19	

- Perbandingan selectStudent() - dengan data student yang ada
- Sebelum optimasi, rata- rata **sample time 8**.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Ct
1	21:44:22.115	Thread Grou...	HTTP Request	16	✓	373	141	15	
2	21:44:22.218	Thread Grou...	HTTP Request	11	✓	373	141	11	
3	21:44:22.322	Thread Grou...	HTTP Request	9	✓	373	141	9	
4	21:44:22.422	Thread Grou...	HTTP Request	10	✓	373	141	9	
5	21:44:22.528	Thread Grou...	HTTP Request	11	✓	373	141	11	
6	21:44:22.625	Thread Grou...	HTTP Request	12	✓	373	141	12	
7	21:44:22.730	Thread Grou...	HTTP Request	11	✓	373	141	11	
8	21:44:22.831	Thread Grou...	HTTP Request	10	✓	373	141	10	
9	21:44:22.936	Thread Grou...	HTTP Request	8	✓	373	141	8	
10	21:44:23.038	Thread Grou...	HTTP Request	10	✓	373	141	10	
11	21:44:23.136	Thread Grou...	HTTP Request	10	✓	373	141	10	
12	21:44:23.237	Thread Grou...	HTTP Request	10	✓	373	141	10	
13	21:44:23.339	Thread Grou...	HTTP Request	11	✓	373	141	10	
14	21:44:23.440	Thread Grou...	HTTP Request	9	✓	373	141	9	
15	21:44:23.542	Thread Grou...	HTTP Request	9	✓	373	141	9	
16	21:44:23.643	Thread Grou...	HTTP Request	9	✓	373	141	9	
17	21:44:23.744	Thread Grou...	HTTP Request	7	✓	373	141	7	
18	21:44:23.844	Thread Grou...	HTTP Request	7	✓	373	141	7	
19	21:44:23.940	Thread Grou...	HTTP Request	9	✓	373	141	9	
20	21:44:24.041	Thread Grou...	HTTP Request	7	✓	373	141	7	
21	21:44:24.143	Thread Grou...	HTTP Request	11	✓	373	141	11	
22	21:44:24.240	Thread Grou...	HTTP Request	8	✓	373	141	8	
23	21:44:24.342	Thread Grou...	HTTP Request	7	✓	373	141	7	
24	21:44:24.442	Thread Grou...	HTTP Request	9	✓	373	141	8	
25	21:44:24.546	Thread Grou...	HTTP Request	18	✓	373	141	18	
26	21:44:24.646	Thread Grou...	HTTP Request	6	✓	373	141	5	

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples 100
 Latest Sample 4
 Average 8
 Deviat

- Setelah optimasi, menggunakan indexing. Terjadi penurunan **sample time** menjadi **5**

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Ct
1	22:41:10.178	Thread Grou...	HTTP Request	8		373	141	8	8
2	22:41:10.282	Thread Grou...	HTTP Request	6		373	141	6	6
3	22:41:10.381	Thread Grou...	HTTP Request	6		373	141	5	5
4	22:41:10.479	Thread Grou...	HTTP Request	7		373	141	7	7
5	22:41:10.581	Thread Grou...	HTTP Request	6		373	141	6	6
6	22:41:10.684	Thread Grou...	HTTP Request	10		373	141	9	9
7	22:41:10.785	Thread Grou...	HTTP Request	5		373	141	5	5
8	22:41:10.885	Thread Grou...	HTTP Request	5		373	141	5	5
9	22:41:10.983	Thread Grou...	HTTP Request	7		373	141	7	7
10	22:41:11.085	Thread Grou...	HTTP Request	6		373	141	6	6
11	22:41:11.183	Thread Grou...	HTTP Request	7		373	141	7	7
12	22:41:11.280	Thread Grou...	HTTP Request	6		373	141	6	6
13	22:41:11.385	Thread Grou...	HTTP Request	6		373	141	6	6
14	22:41:11.480	Thread Grou...	HTTP Request	6		373	141	6	6
15	22:41:11.582	Thread Grou...	HTTP Request	6		373	141	6	6
16	22:41:11.685	Thread Grou...	HTTP Request	7		373	141	6	6
17	22:41:11.782	Thread Grou...	HTTP Request	4		373	141	4	4
18	22:41:11.886	Thread Grou...	HTTP Request	6		373	141	6	6
19	22:41:11.985	Thread Grou...	HTTP Request	7		373	141	7	7
20	22:41:12.086	Thread Grou...	HTTP Request	7		373	141	7	7
21	22:41:12.182	Thread Grou...	HTTP Request	4		373	141	4	4
22	22:41:12.283	Thread Grou...	HTTP Request	7		373	141	7	7
23	22:41:12.384	Thread Grou...	HTTP Request	4		373	141	4	4
24	22:41:12.485	Thread Grou...	HTTP Request	5		373	141	5	5
25	22:41:12.587	Thread Grou...	HTTP Request	4		373	141	4	4
26	22:41:12.689	Thread Grou...	HTTP Request	33		373	141	33	33

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples 100
 Latest Sample 5
 Average 5
 Deviat

- Perbandingan selectStudent() - dengan data student yang tidak ada
- Sebelum Optimasi, rata-rata **sample time 6**.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Ct
1	21:45:13.912	Thread Grou...	HTTP Request	15		333	132	15	15
2	21:45:14.018	Thread Grou...	HTTP Request	7		333	132	7	7
3	21:45:14.118	Thread Grou...	HTTP Request	6		333	132	6	6
4	21:45:14.218	Thread Grou...	HTTP Request	6		333	132	6	6
5	21:45:14.320	Thread Grou...	HTTP Request	6		333	132	6	6
6	21:45:14.418	Thread Grou...	HTTP Request	8		333	132	8	8
7	21:45:14.519	Thread Grou...	HTTP Request	7		333	132	7	7
8	21:45:14.621	Thread Grou...	HTTP Request	9		333	132	9	9
9	21:45:14.717	Thread Grou...	HTTP Request	7		333	132	7	7
10	21:45:14.819	Thread Grou...	HTTP Request	7		333	132	7	7
11	21:45:14.923	Thread Grou...	HTTP Request	7		333	132	7	7
12	21:45:15.024	Thread Grou...	HTTP Request	6		333	132	6	6
13	21:45:15.120	Thread Grou...	HTTP Request	8		333	132	8	8
14	21:45:15.226	Thread Grou...	HTTP Request	7		333	132	7	7
15	21:45:15.327	Thread Grou...	HTTP Request	9		333	132	9	9
16	21:45:15.427	Thread Grou...	HTTP Request	5		333	132	5	5
17	21:45:15.525	Thread Grou...	HTTP Request	5		333	132	4	4
18	21:45:15.627	Thread Grou...	HTTP Request	9		333	132	9	9
19	21:45:15.727	Thread Grou...	HTTP Request	5		333	132	5	5
20	21:45:15.826	Thread Grou...	HTTP Request	6		333	132	6	6
21	21:45:15.929	Thread Grou...	HTTP Request	6		333	132	6	6
22	21:45:16.026	Thread Grou...	HTTP Request	5		333	132	5	5
23	21:45:16.128	Thread Grou...	HTTP Request	6		333	132	6	6
24	21:45:16.229	Thread Grou...	HTTP Request	7		333	132	7	7
25	21:45:16.332	Thread Grou...	HTTP Request	7		333	132	7	7
26	21:45:16.432	Thread Grou...	HTTP Request	7		333	132	7	7

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples 100
 Latest Sample 7
 Average 6
 Deviat

- Setelah Optimasi, **sample time** menjadi **4**.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Ci
1	22:46:57.894	Thread Grou...	HTTP Request	15	✓	333	132	15	
2	22:46:57.997	Thread Grou...	HTTP Request	4	✓	333	132	3	
3	22:46:58.098	Thread Grou...	HTTP Request	5	✓	333	132	5	
4	22:46:58.199	Thread Grou...	HTTP Request	6	✓	333	132	6	
5	22:46:58.299	Thread Grou...	HTTP Request	5	✓	333	132	5	
6	22:46:58.395	Thread Grou...	HTTP Request	5	✓	333	132	5	
7	22:46:58.496	Thread Grou...	HTTP Request	3	✓	333	132	3	
8	22:46:58.597	Thread Grou...	HTTP Request	4	✓	333	132	4	
9	22:46:58.696	Thread Grou...	HTTP Request	8	✓	333	132	7	
10	22:46:58.799	Thread Grou...	HTTP Request	3	✓	333	132	3	
11	22:46:58.900	Thread Grou...	HTTP Request	5	✓	333	132	5	
12	22:46:59.001	Thread Grou...	HTTP Request	4	✓	333	132	4	
13	22:46:59.102	Thread Grou...	HTTP Request	4	✓	333	132	4	
14	22:46:59.198	Thread Grou...	HTTP Request	3	✓	333	132	3	
15	22:46:59.300	Thread Grou...	HTTP Request	4	✓	333	132	4	
16	22:46:59.401	Thread Grou...	HTTP Request	5	✓	333	132	5	
17	22:46:59.502	Thread Grou...	HTTP Request	5	✓	333	132	5	
18	22:46:59.601	Thread Grou...	HTTP Request	4	✓	333	132	4	
19	22:46:59.698	Thread Grou...	HTTP Request	9	✓	333	132	9	
20	22:46:59.799	Thread Grou...	HTTP Request	4	✓	333	132	4	
21	22:46:59.903	Thread Grou...	HTTP Request	5	✓	333	132	5	
22	22:47:00.003	Thread Grou...	HTTP Request	5	✓	333	132	5	
23	22:47:00.103	Thread Grou...	HTTP Request	4	✓	333	132	4	
24	22:47:00.199	Thread Grou...	HTTP Request	10	✓	333	132	10	
25	22:47:00.299	Thread Grou...	HTTP Request	5	✓	333	132	5	
26	22:47:00.404	Thread Grou...	HTTP Request	4	✓	333	132	4	

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples 100
 Latest Sample 4
 Average 4
 Deviat

4. Tulislah ide optimasi Anda, proses Anda melakukan optimasi, perbandingan hasil JMeter sebelum dan sesudah optimasi, hasil dari optimasi Anda apakah berhasil atau tidak, dan hal lainnya yang perlu Anda jelaskan tentang optimasi tersebut pada kode dan database Anda dalam file pdf bernama tutorial6_writeup.pdf. Anda wajib menuliskan write-up.

Ide yang saya lakukan terbagi dalam 3 cara :

- indexing
 - saya tujukan untuk method **selectStudent()**
 - karena indexing saya set di kolom npm
 - Hasil optimasi berhasil
- collation
 - saya set di tipe data varchar
 - digunakan di kolom npm dan nama
 - ditujukan untuk semua method
 - Hasil Optimasi berhasil
 - Kecuali untuk method **selectAllStudents()**
- query
 - karena **selectAllStudents()** tetap gagal
 - sehingga saya coba dari query
 - ketika query di limit menjadi 10, penggunaan thread 100 dengan ramp-up period 10 detik, optimasi **berhasil**
 - Maka yang perlu dilakukan selanjutnya adalah perlu adanya akses viewAll data yang menggunakan pagination tetapi by limit.
 - Sehingga pada saat pertama kali akses **selectAllStudents()** tidak semua data di load secara bersamaan

Sebenarnya, saya ada ide lain untuk menambahkan HEAP saat proses java dijalankan. Hal ini memang belum menjamin, tetapi kedepannya tidak salah untuk dicoba.

HASIL PEMBELAJARAN

- Penggunaan **unit testing** untuk memastikan beberapa test case
- Sehingga meminimalisir, error
- Digunakan framework **junit**
- Dipelajari **load testing**
- Mempelajari bagaimana cara mengoptimasi request menggunakan **jmeter**
- Optimasi yang dilakukan ada beberapa cara
- Pada kasus kali ini yang menjadi parameter **sample time**