

Unit Testing dan Load Testing Web App Sederhana

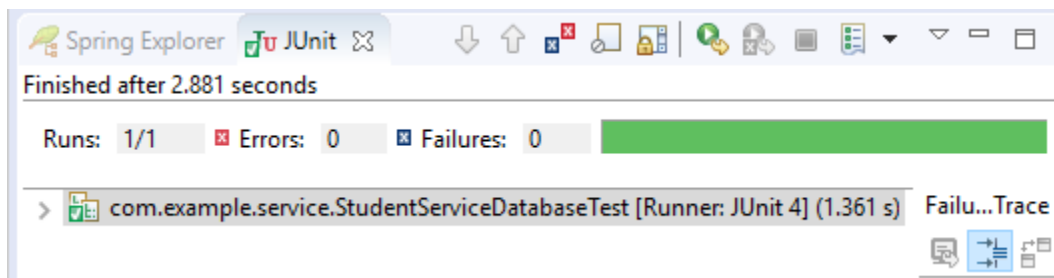
Menjalankan Testing Pada Spring Boot

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

    //when
    StudentModel test = studentService.selectStudent("1506737823");

    //then
    assertThat(test, notNullValue());
    assertThat(test, equalTo(check));
}
```

Hasilnya sebagai berikut



Latihan Unit Test

1. Unit test selectAllStudent

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

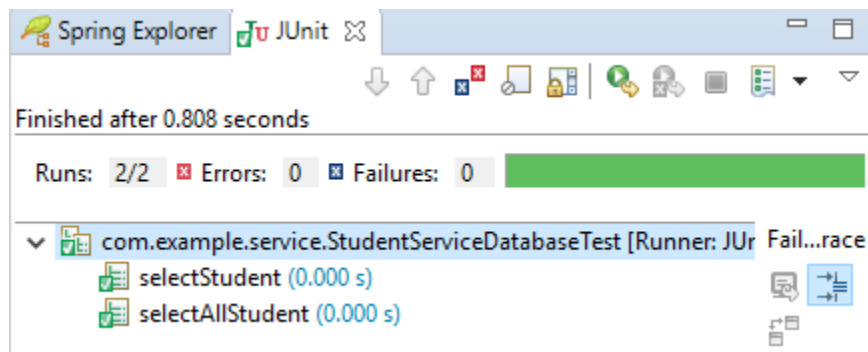
    List<StudentModel> checks = new ArrayList<>();
    StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);
    checks.add(check);

    BDDMockito.given(studentMapper.selectAllStudents()).willReturn(studentModels);

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

    //then
    assertThat(test, notNullValue());
    assertThat(test.isEmpty(), equalTo(false));
    assertThat(test.size(), equalTo(1));
    assertThat(test, equalTo(checks));
}
```

Hasil test



Terjadi beberapa testing dalam method `selectAllStudent` yang berfungsi untuk melakukan unit testing terhadap method `selectAllStudent` dengan adanya 3 segment yaitu:

- Given, melakukan deklarasi dan inialisasi objek `studentModels` dengan memberikan nilainya. Kemudian BDDMockito proses pemanggilan method `selectAllStudent` yang ada pada `studentMapper`.
- When, melakukan proses unit testing dengan memanggil method `selectAllStudent` dari interface `studentService` dan menampung hasilnya dalam list `studentModel`.
- Then, melakukan pengecekan terhadap objek baru. Terdapat beberapa pengecekan sebagai berikut.
 - Check apakah nilainya null atau tidak
 - Cek apakah kosong atau tidak
 - Cek apakah size sama
 - Cek apakah memiliki konten yang sama atau tidak

2. Unit Test `addStudent`

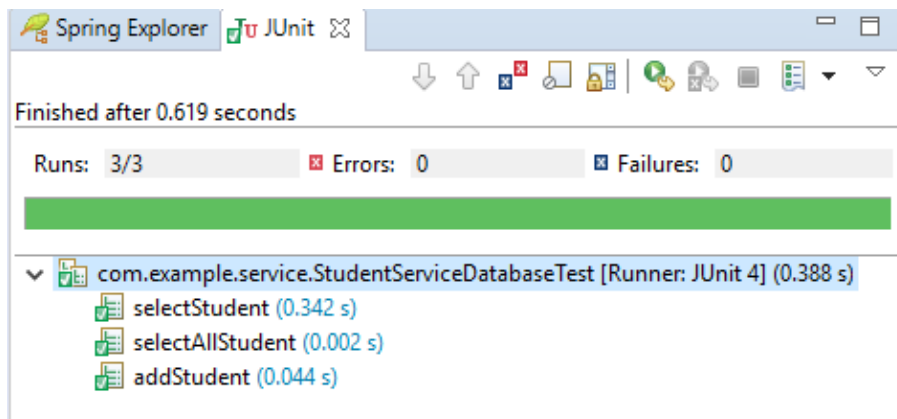
```
@Test
public void addStudent() {
    //Given
    StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
    StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);

    BDDMockito.given(studentMapper.addStudent(studentModel)).willReturn(true);

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

    //Then
    BDDMockito.then(studentMapper).should().addStudent(check);
    assertThat(test, equalTo(true));
}
```

Hasil test



Unit test ini bertujuan untuk melakukan testing terhadap method `addStudent` yaitu ketika menambahkan student dengan beberapa kondisi misalnya jika ada nilai yang sama. Terdapat 3 segment dalam method tersebut yaitu:

- Given, melakukan deklarasi dan inisialisasi objek `studentModels` dengan memberikan nilainya. Kemudian BDDMockito proses pemanggilan method `addStudent` yang ada pada `studentMapper`.
- When, melakukan proses unit testing dengan memanggil method `addStudent` dari interface `studentService` dan menampung hasilnya dalam Boolean `true` jika berhasil.
- Then, melakukan pengecekan terhadap objek baru. Terdapat beberapa pengecekan apakah memiliki konten yang sama atau tidak

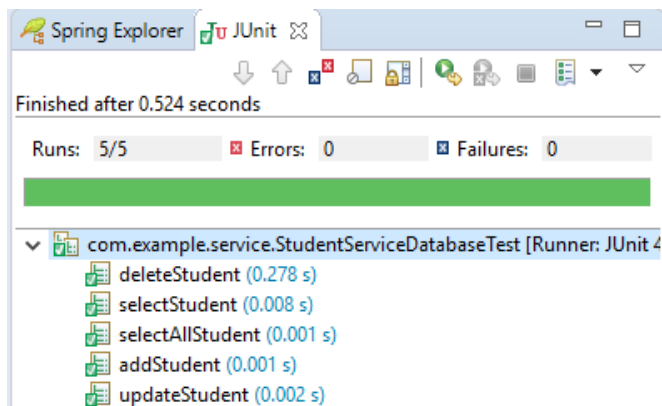
3. Unit Test pada method delete student

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

    //When
    boolean test = studentService.deleteStudent("1506737823");

    //Then
    BDDMockito.then(studentMapper).should().deleteStudent("1506737823");
    assertThat(test, equalTo(true));
}
```

Hasil testing



Unit test ini bertujuan untuk melakukan testing terhadap method deleteStudent yaitu ketika menghapus data student. Terdapat 3 segment dalam method tersebut yaitu:

- Given, melakukan deklarasi dan inialisasi objek studentModels dengan memberikan nilainya. Kemudian BDDMockito proses pemanggilan method deleteStudent yang ada pada studentMapper.
- When, melakukan proses unit testing dengan memanggil method deleteStudent dari interface studentService dan menampung hasilnya dalam Boolean true jika berhasil.
- Then, melakukan pengecekan terhadap objek baru. Terdapat beberapa pengecekan apakah memiliki konten yang sama atau tidak.

4. Unit Test pada method update student

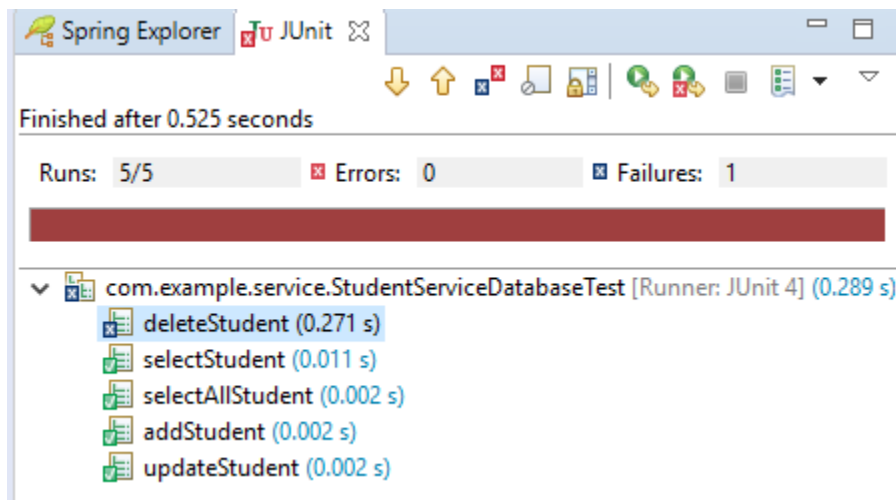
```
@Test
public void updateStudent() {
    //Given
    StudentModel studentModel = new StudentModel("1506737823", "Chanek", 3.5);
    StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);

    BDDMockito.given(studentService.updateStudent(studentModel)).willReturn(true);

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

    //Then
    BDDMockito.then(studentMapper).should().updateStudent(studentModel);
    assertThat(test, equalTo(true));
}
```

Hasil testing sebagai berikut



Unit test ini bertujuan untuk melakukan testing terhadap method updateStudent yaitu ketika mengubah data student. Terdapat 3 segment dalam method tersebut yaitu:

- Given, melakukan deklarasi dan inisialisasi objek studentModels dengan memberikan nilainya. Kemudian BDDMockito proses pemanggilan method updateStudent yang ada pada studentMapper.
- When, melakukan proses unit testing dengan memanggil method updateStudent dari interface studentService dan menampung hasilnya dalam Boolean true jika berhasil.
- Then, melakukan pengecekan terhadap objek baru. Terdapat beberapa pengecekan apakah memiliki konten yang sama atau tidak

Tutorial JMeter untuk Load Testing

Hasil number of threat = 50 error dikarenakan memori yang terbatas.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

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

Sampl...	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	17:59:09.117	Thread Group ...	HTTP Request	245787	✖	361	133	245748	1
2	17:59:11.937	Thread Group ...	HTTP Request	312527	✖	361	133	312527	2
3	17:59:10.526	Thread Group	HTTP Request	313959	✖	361	133	313958	2
4	17:59:10.118	Thread Group ...	HTTP Request	371432	✖	361	133	371432	2

Hasil dengan number of threat = 10 setelah beberapa kali mengurangi number of threat.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☒ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	18:37:41.586	Thread Group 1-1	HTTP Request	8610	✓	3206090	133	8038	76
2	18:37:42.603	Thread Group 1-2	HTTP Request	16802	✓	3206090	133	16059	1
3	18:37:43.615	Thread Group 1-3	HTTP Request	18893	✓	3206090	133	17793	2
4	18:37:44.602	Thread Group 1-4	HTTP Request	22637	✓	3206090	133	21978	1
5	18:37:45.601	Thread Group 1-5	HTTP Request	22872	✓	3206090	133	22059	2
6	18:37:47.604	Thread Group 1-7	HTTP Request	22111	✓	3206090	133	21429	1
7	18:37:46.603	Thread Group 1-6	HTTP Request	23799	✓	3206090	133	22659	1
8	18:37:48.635	Thread Group 1-8	HTTP Request	22340	✓	3206090	133	21935	1
9	18:37:49.634	Thread Group 1-9	HTTP Request	21815	✓	3206090	133	21302	12
10	18:37:50.635	Thread Group 1-10	HTTP Request	21174	✓	3206090	133	20899	1

☒ Scroll automatically? ☐ Child samples? No of Samples 10 Latest Sample 21174 Average 20105 Deviation 4304

Latihan JMeter

1. Screenshot hasil JMeter fitur soal latihan

Load Testing nomor 1 sebelum dioptimasi (3 file)

- Jmeter selectStudentBerhasil dengan number of threat = 100
Berdasarkan hasil di bawah ini dapat dilihat bahwa average time sebesar 26 ms.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☒ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	19:03:10.632	Thread Group 3-1	HTTP Request	69	✓	2180	136	69	1
2	19:03:10.733	Thread Group 3-2	HTTP Request	25	✓	2180	136	25	1
3	19:03:10.842	Thread Group 3-3	HTTP Request	62	✓	2180	136	62	1
4	19:03:10.947	Thread Group 3-4	HTTP Request	38	✓	2180	136	38	1
5	19:03:11.040	Thread Group 3-5	HTTP Request	31	✓	2180	136	31	1
6	19:03:11.133	Thread Group 3-6	HTTP Request	22	✓	2180	136	22	1
7	19:03:11.242	Thread Group 3-7	HTTP Request	40	✓	2180	136	40	1
8	19:03:11.335	Thread Group 3-8	HTTP Request	31	✓	2180	136	31	1
9	19:03:11.448	Thread Group 3-9	HTTP Request	34	✓	2180	136	34	1
10	19:03:11.541	Thread Group 3-10	HTTP Request	25	✓	2180	136	25	1
11	19:03:11.635	Thread Group 3-11	HTTP Request	17	✓	2180	136	17	1
12	19:03:11.735	Thread Group 3-12	HTTP Request	24	✓	2180	136	24	1
13	19:03:11.836	Thread Group 3-13	HTTP Request	30	✓	2180	136	30	1
14	19:03:11.936	Thread Group 3-14	HTTP Request	21	✓	2180	136	21	1
15	19:03:12.036	Thread Group 3-15	HTTP Request	21	✓	2180	136	21	1
16	19:03:12.137	Thread Group 3-16	HTTP Request	22	✓	2180	136	21	1
17	19:03:12.237	Thread Group 3-17	HTTP Request	24	✓	2180	136	24	2
18	19:03:12.338	Thread Group 3-18	HTTP Request	21	✓	2180	136	21	1
19	19:03:12.437	Thread Group 3-19	HTTP Request	32	✓	2180	136	32	1
20	19:03:12.537	Thread Group 3-20	HTTP Request	17	✓	2180	136	16	1
21	19:03:12.637	Thread Group 3-21	HTTP Request	23	✓	2180	136	23	1
22	19:03:12.738	Thread Group 3-22	HTTP Request	33	✓	2180	136	33	1
23	19:03:12.838	Thread Group 3-23	HTTP Request	21	✓	2180	136	21	1
24	19:03:12.978	Thread Group 3-24	HTTP Request	22	✓	2180	136	22	1
25	19:03:13.039	Thread Group 3-25	HTTP Request	19	✓	2180	136	19	1

☒ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 22 Average 26 Deviation 9

- Jmeter selectStudentGagal dengan number of threat = 100
Berdasarkan hasil di bawah ini dapat dilihat bahwa average time sebesar 39 ms.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	19:05:35.831	Thread Group 4-1	HTTP Request	43	Success	2154	138	42	1
2	19:05:35.943	Thread Group 4-2	HTTP Request	33	Success	2154	138	33	1
3	19:05:36.040	Thread Group 4-3	HTTP Request	36	Success	2154	138	35	2
4	19:05:36.143	Thread Group 4-4	HTTP Request	25	Success	2154	138	25	2
5	19:05:36.236	Thread Group 4-5	HTTP Request	36	Success	2154	138	36	1
6	19:05:36.345	Thread Group 4-6	HTTP Request	32	Success	2154	138	32	1
7	19:05:36.435	Thread Group 4-7	HTTP Request	34	Success	2154	138	34	1
8	19:05:36.544	Thread Group 4-8	HTTP Request	28	Success	2154	138	28	1
9	19:05:36.642	Thread Group 4-9	HTTP Request	27	Success	2154	138	27	2
10	19:05:36.736	Thread Group 4-10	HTTP Request	27	Success	2154	138	27	1
11	19:05:36.832	Thread Group 4-11	HTTP Request	27	Success	2154	138	27	1
12	19:05:36.933	Thread Group 4-12	HTTP Request	20	Success	2154	138	20	1
13	19:05:37.033	Thread Group 4-13	HTTP Request	23	Success	2154	138	23	1
14	19:05:37.133	Thread Group 4-14	HTTP Request	41	Success	2154	138	40	1
15	19:05:37.233	Thread Group 4-15	HTTP Request	35	Success	2154	138	35	1
16	19:05:37.333	Thread Group 4-16	HTTP Request	37	Success	2154	138	37	2
17	19:05:37.434	Thread Group 4-17	HTTP Request	18	Success	2154	138	18	1
18	19:05:37.533	Thread Group 4-18	HTTP Request	26	Success	2154	138	26	1
19	19:05:37.642	Thread Group 4-19	HTTP Request	21	Success	2154	138	21	1
20	19:05:37.735	Thread Group 4-20	HTTP Request	27	Success	2154	138	27	1
21	19:05:37.836	Thread Group 4-21	HTTP Request	23	Success	2154	138	22	1
22	19:05:37.934	Thread Group 4-22	HTTP Request	33	Success	2154	138	32	1
23	19:05:38.034	Thread Group 4-23	HTTP Request	23	Success	2154	138	23	1
24	19:05:38.134	Thread Group 4-24	HTTP Request	25	Success	2154	138	25	1
25	19:05:38.234	Thread Group 4-25	HTTP Request	23	Success	2154	138	23	1

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples 100
 Latest Sample 41
 Average 39
 Deviation 14

- jmeter_selectAllStudents dengan number of threat = 10 (dikarenakan keterbatasan memori)
Berdasarkan hasil di bawah ini dapat dilihat bahwa average time sebesar 20105 ms.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	18:37:41.586	Thread Group 1-1	HTTP Request	8610	Success	3206090	133	8038	76
2	18:37:42.603	Thread Group 1-2	HTTP Request	16802	Success	3206090	133	16059	1
3	18:37:43.615	Thread Group 1-3	HTTP Request	18893	Success	3206090	133	17793	2
4	18:37:44.602	Thread Group 1-4	HTTP Request	22637	Success	3206090	133	21978	1
5	18:37:45.601	Thread Group 1-5	HTTP Request	22872	Success	3206090	133	22059	2
6	18:37:46.604	Thread Group 1-6	HTTP Request	22111	Success	3206090	133	21429	1
7	18:37:47.603	Thread Group 1-7	HTTP Request	23799	Success	3206090	133	22659	1
8	18:37:48.635	Thread Group 1-8	HTTP Request	22340	Success	3206090	133	21935	1
9	18:37:49.634	Thread Group 1-9	HTTP Request	21815	Success	3206090	133	21302	12
10	18:37:50.635	Thread Group 1-10	HTTP Request	21174	Success	3206090	133	20899	1

☐ Scroll automatically?
 ☐ Child samples?
 No of Samples 10
 Latest Sample 21174
 Average 20105
 Deviation 4304

- Optimasi dilakukan dengan mengubah kolom nmp di database menjadi unique dan index sehingga diharapkan average time berkurang.

Screenshot hasil JMeter fitur soal latihan Load Testing nomor 2 setelah dioptimasi

- jmeter_selectStudentBerhasil dengan number of threat = 100. Berdasarkan hasil di bawah ini dapat dilihat bahwa average time sebesar 12 ms lebih kecil dibandingkan sebelum optimasi.

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(m...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	20:47:53.757	Thread Group 1...	HTTP Request	181	✓	2185	141	181	5
2	20:47:53.857	Thread Group 1...	HTTP Request	85	✓	2185	141	85	1
3	20:47:53.966	Thread Group 1...	HTTP Request	9	✓	2185	141	9	1
4	20:47:54.059	Thread Group 1...	HTTP Request	11	✓	2185	141	11	2
5	20:47:54.153	Thread Group 1...	HTTP Request	9	✓	2185	141	9	1
6	20:47:54.263	Thread Group 1...	HTTP Request	26	✓	2185	141	26	1
7	20:47:54.356	Thread Group 1...	HTTP Request	10	✓	2185	141	10	1
8	20:47:54.466	Thread Group 1...	HTTP Request	11	✓	2185	141	11	1
9	20:47:54.559	Thread Group 1...	HTTP Request	11	✓	2185	141	11	2
10	20:47:54.653	Thread Group 1...	HTTP Request	9	✓	2185	141	9	1
11	20:47:54.763	Thread Group 1...	HTTP Request	13	✓	2185	141	13	1
12	20:47:54.850	Thread Group 1...	HTTP Request	9	✓	2185	141	9	1
13	20:47:54.950	Thread Group 1...	HTTP Request	28	✓	2185	141	28	1
14	20:47:55.050	Thread Group 1...	HTTP Request	11	✓	2185	141	11	1
15	20:47:55.151	Thread Group 1...	HTTP Request	40	✓	2185	141	40	1
16	20:47:55.250	Thread Group 1...	HTTP Request	19	✓	2185	141	19	6
17	20:47:55.350	Thread Group 1...	HTTP Request	13	✓	2185	141	13	2
18	20:47:55.451	Thread Group 1...	HTTP Request	10	✓	2185	141	10	2
19	20:47:55.552	Thread Group 1...	HTTP Request	16	✓	2185	141	16	1
20	20:47:55.651	Thread Group 1...	HTTP Request	12	✓	2185	141	12	2
21	20:47:55.752	Thread Group 1...	HTTP Request	10	✓	2185	141	10	1
22	20:47:55.851	Thread Group 1...	HTTP Request	56	✓	2185	141	56	1
23	20:47:55.950	Thread Group 1...	HTTP Request	15	✓	2185	141	15	2
24	20:47:56.050	Thread Group 1...	HTTP Request	8	✓	2185	141	8	1
25	20:47:56.151	Thread Group 1...	HTTP Request	6	✓	2185	141	6	0
26	20:47:56.251	Thread Group 1...	HTTP Request	9	✓	2185	141	8	1

☒ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 7 Average 12 Deviation 19

- jmeter_selectStudentGagal dengan number of threat = 100. Berdasarkan hasil di bawah ini dapat dilihat bahwa average time sebesar 10 ms lebih kecil dibandingkan sebelum optimasi.

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(m...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	20:50:49.024	Thread Group 1...	HTTP Request	87	✓	2159	143	87	2
2	20:50:49.124	Thread Group 1...	HTTP Request	9	✓	2159	143	9	1
3	20:50:49.265	Thread Group 1...	HTTP Request	8	✓	2159	143	8	1
4	20:50:49.327	Thread Group 1...	HTTP Request	9	✓	2159	143	9	1
5	20:50:49.468	Thread Group 1...	HTTP Request	16	✓	2159	143	16	0
6	20:50:49.562	Thread Group 1...	HTTP Request	16	✓	2159	143	16	1
7	20:50:49.624	Thread Group 1...	HTTP Request	8	✓	2159	143	8	1
8	20:50:49.764	Thread Group 1...	HTTP Request	8	✓	2159	143	8	1
9	20:50:49.874	Thread Group 1...	HTTP Request	8	✓	2159	143	7	1
10	20:50:49.921	Thread Group 1...	HTTP Request	6	✓	2159	143	6	1
11	20:50:50.061	Thread Group 1...	HTTP Request	7	✓	2159	143	7	1
12	20:50:50.161	Thread Group 1...	HTTP Request	7	✓	2159	143	7	1
13	20:50:50.229	Thread Group 1...	HTTP Request	8	✓	2159	143	8	1
14	20:50:50.370	Thread Group 1...	HTTP Request	6	✓	2159	143	6	1
15	20:50:50.464	Thread Group 1...	HTTP Request	6	✓	2159	143	6	1
16	20:50:50.573	Thread Group 1...	HTTP Request	9	✓	2159	143	9	1
17	20:50:50.667	Thread Group 1...	HTTP Request	26	✓	2159	143	26	1
18	20:50:50.776	Thread Group 1...	HTTP Request	9	✓	2159	143	9	1
19	20:50:50.870	Thread Group 1...	HTTP Request	96	✓	2159	143	96	1
20	20:50:50.964	Thread Group 1...	HTTP Request	9	✓	2159	143	9	1
21	20:50:51.073	Thread Group 1...	HTTP Request	11	✓	2159	143	11	2
22	20:50:51.161	Thread Group 1...	HTTP Request	5	✓	2159	143	5	1
23	20:50:51.261	Thread Group 1...	HTTP Request	7	✓	2159	143	7	1
24	20:50:51.361	Thread Group 1...	HTTP Request	7	✓	2159	143	7	1
25	20:50:51.461	Thread Group 1...	HTTP Request	7	✓	2159	143	7	1
26	20:50:51.561	Thread Group 1...	HTTP Request	11	✓	2159	143	11	1

☒ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 8 Average 10 Deviation 12

- jmeter_selectAllStudents dengan number of threat = 10 (dikarenakan keterbatasan memori). Berdasarkan hasil di bawah ini dapat dilihat bahwa average time sebesar 10963 ms lebih kecil dibandingkan dengan sebelum optimasi.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☒ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	20:41:28.181	Thread Group 1-8	HTTP Request	10812		1603281	133	10589	468
2	20:41:28.181	Thread Group 1-3	HTTP Request	10810		1603281	133	10630	468
3	20:41:28.181	Thread Group 1-6	HTTP Request	10774		1603281	133	10579	468
4	20:41:28.182	Thread Group 1-7	HTTP Request	10752		1603281	133	10454	467
5	20:41:28.181	Thread Group 1...	HTTP Request	10758		1603281	133	10455	468
6	20:41:28.181	Thread Group 1-9	HTTP Request	10813		1603281	133	10632	468
7	20:41:28.181	Thread Group 1-1	HTTP Request	10750		1603281	133	10455	468
8	20:41:28.181	Thread Group 1-4	HTTP Request	10852		1603281	133	10610	468
9	20:41:28.181	Thread Group 1-2	HTTP Request	10962		1603281	133	10712	468
10	20:41:28.181	Thread Group 1-5	HTTP Request	10963		1603281	133	10811	468

☐ Scroll automatically? ☐ Child samples? No of Samples 10 Latest Sample 10963 Average 10824 Deviation 75