

## Tutorial 6 Unit Testing dan Load Testing Web App Sederhana

Nama : Esther Theresia Sihotang

NPM : 1506797242

### Latihan Unit Testing

1. Tambahkan `code selectAllStudent` pada `StudentServiceDatabase`. Jelaskan apa yang terjadi pada method tersebut;

Jawab : pada method `selectAllStudent` terdapat 3(tiga) segmen yaitu Given, When dan Then yang akan dilakukan pada setiap *testing*.

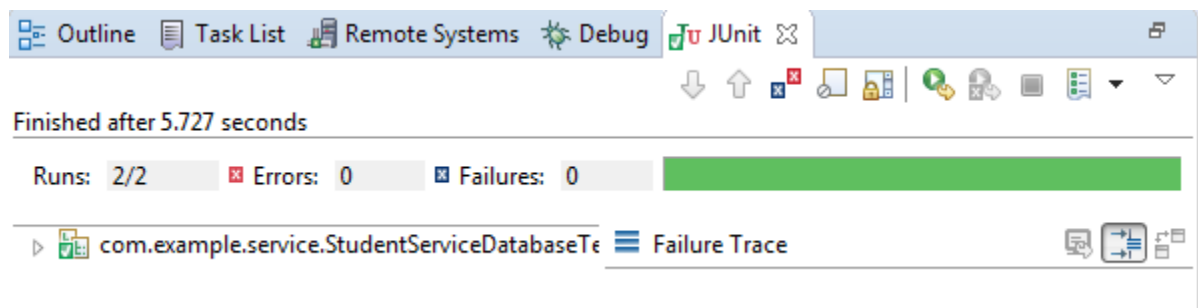
- Pada **Given**, pengguna menginisialisasi objek `studentModels` pada `arraylist` kemudian `BDDMockito` melakukan proses sesuai `method selectAllStudent` dengan hasil `return value` sesuai dengan `willReturn(studentModels)`.
- Pada **When**, akan dilakukan pengetesan terhadap method `selectAllStudent` dengan memasukkan semua objek `student` yang ada pada `arraylist` apakah method `selectAllStudent` tetap berjalan.
- Pada **Then**, akan dilakukan pengecekan terhadap objek baru tersebut. Objek baru dicek jika null, cek kalau ukuran sama, cek kalau konten sama atau sesuai dengan objek check yang telah disiapkan sebelumnya.

2. Tambahkan `code addStudent` pada `StudentServiceDatabase`. Jelaskan apa yang terjadi pada method tersebut

Jawab : pada method `addStudent` terdapat 3(tiga) segmen yaitu Given, When dan Then yang akan dilakukan pada setiap *testing*.

- Pada **Given**, pengguna menginisialisasi objek `studentModels` kemudian `BDDMockito` melakukan proses sesuai `method addStudent` dengan hasil `return value` sesuai dengan `willReturn(studentModels)`.
- Pada **When**, akan dilakukan pengetesan terhadap method `addStudent` dengan memasukkan semua objek `student` yang ada pada `arraylist` apakah method `addStudent` tetap berjalan.
- Pada **Then**, `BDDMockito` melakukan pengecekan objek baru apakah sesuai dengan objek check yang telah disiapkan.

Hasil testing :



3. Tambahkan *code* **deleteStudent** pada StudentServiceDatabase. Jelaskan apa yang terjadi pada method tersebut

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

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

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

Jawab : pada method deleteStudent terdapat 3(tiga) segmen yaitu Given, When dan Then yang akan dilakukan pada setiap *testing*.

- Pada **Given**, pengguna menginisialisasi objek studentModels kemudian BDDMockito melakukan proses sesuai *method* **deleteStudent** dengan hasil *return value* sesuai dengan willReturn(studentModels).
  - Pada **When**, akan dilakukan pengetestan terhadap method **deleteStudent** dengan memasukkan semua objek *student* yang ada pada arraylist apakah method **deleteStudent** tetap berjalan.
  - Pada **Then**, BDDMockito melakukan pengecekan objek baru apakah sesuai dengan objek check yang telah disiapkan.
4. Tambahkan *code* **updateStudent** pada StudentServiceDatabase. Jelaskan apa yang terjadi pada method tersebut

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

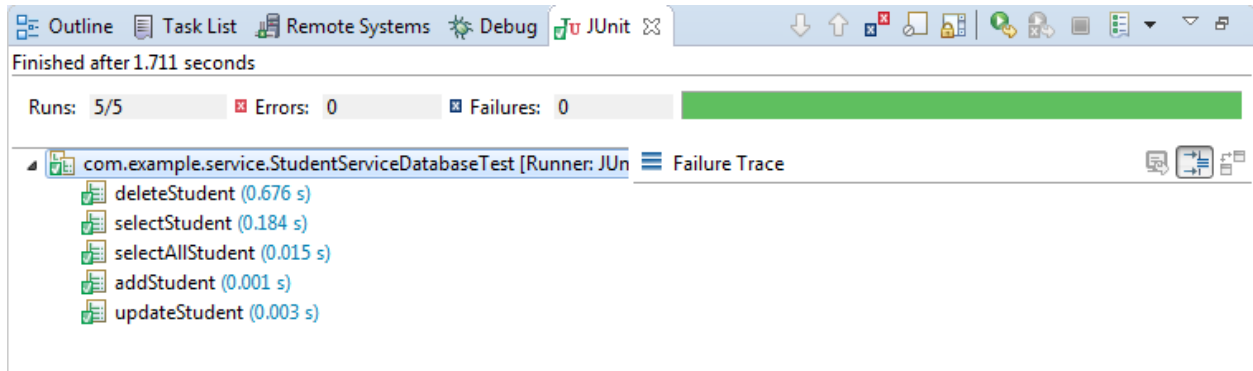
    //When
    boolean test = studentService.updateStudent(studentModel);

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

Jawab : pada method updateStudent terdapat 3(tiga) segmen yaitu Given, When dan Then yang akan dilakukan pada setiap *testing*.

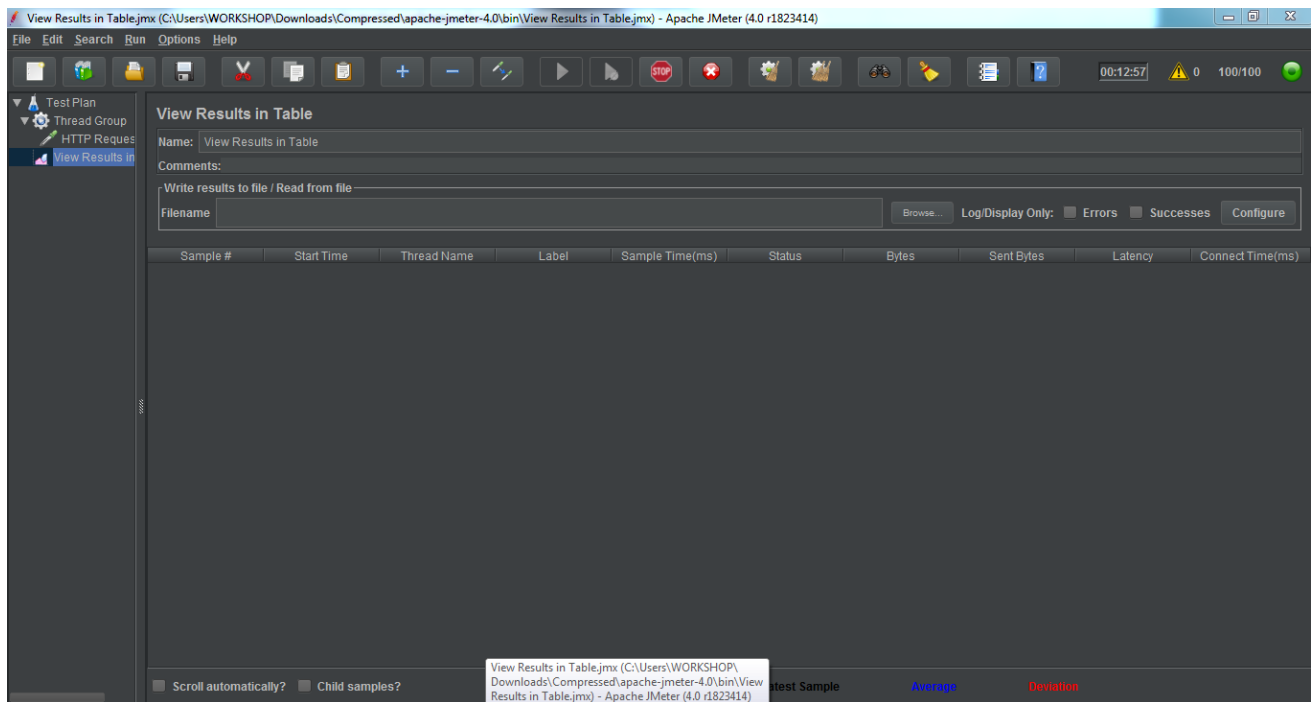
- Pada **Given**, pengguna menginisialisasi objek studentModels kemudian BDDMockito melakukan proses sesuai *method* **updateStudent** dengan hasil *return value* sesuai dengan willReturn(studentModels).
- Pada **When**, akan dilakukan pengetestan terhadap method **updateStudent** dengan memasukkan semua objek *student* yang ada pada arraylist apakah method **updateStudent** tetap berjalan.

- Pada **Then**, BDDMockito melakukan pengecekan objek baru apakah sesuai dengan objek check yang telah disiapkan.

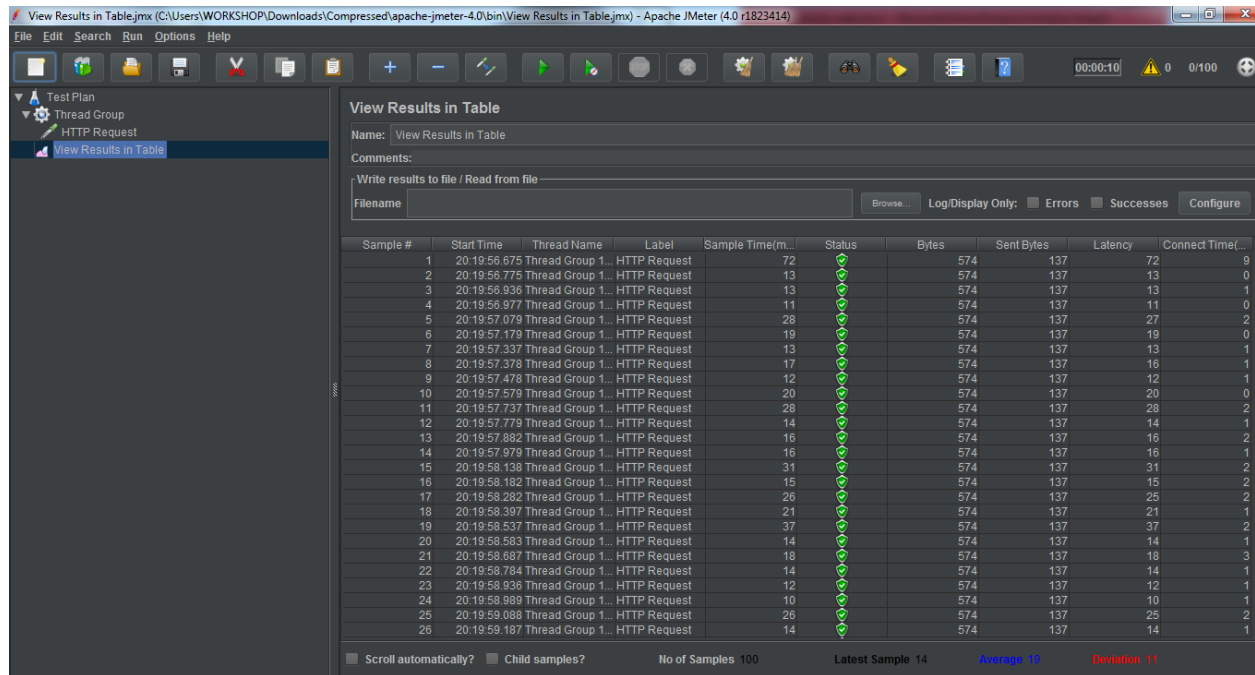


## Load Testing Dengan Apache JMeter

### JMeter Select All Student berhasil ditemukan



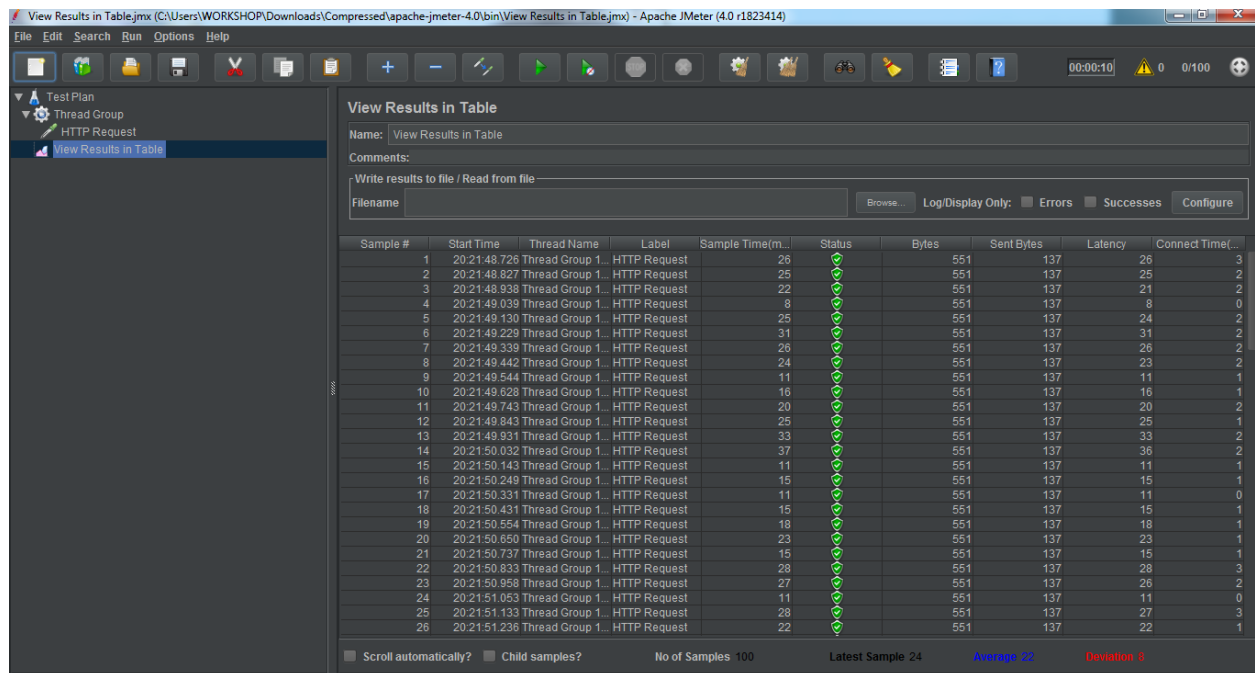
## JMeter Select student berhasil ditemukan



The screenshot shows the JMeter 'View Results in Table' window. The table displays 26 samples, all with a status of 'Success' (green checkmark). The columns include Sample #, Start Time, Thread Name, Label, Sample Time (ms), Status, Bytes, Sent Bytes, Latency, and Connect Time (ms). The status bar at the bottom indicates 'No of Samples: 100', 'Latest Sample: 14', 'Average: 19', and 'Deviation: 11'.

Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	20:19:56.675	Thread Group 1...	HTTP Request	72	Success	574	137	72	9
2	20:19:56.775	Thread Group 1...	HTTP Request	13	Success	574	137	13	0
3	20:19:56.936	Thread Group 1...	HTTP Request	13	Success	574	137	13	1
4	20:19:56.977	Thread Group 1...	HTTP Request	11	Success	574	137	11	0
5	20:19:57.079	Thread Group 1...	HTTP Request	28	Success	574	137	27	2
6	20:19:57.179	Thread Group 1...	HTTP Request	19	Success	574	137	19	0
7	20:19:57.337	Thread Group 1...	HTTP Request	13	Success	574	137	13	1
8	20:19:57.378	Thread Group 1...	HTTP Request	17	Success	574	137	16	1
9	20:19:57.478	Thread Group 1...	HTTP Request	12	Success	574	137	12	1
10	20:19:57.579	Thread Group 1...	HTTP Request	20	Success	574	137	20	0
11	20:19:57.737	Thread Group 1...	HTTP Request	28	Success	574	137	28	2
12	20:19:57.775	Thread Group 1...	HTTP Request	14	Success	574	137	14	1
13	20:19:57.882	Thread Group 1...	HTTP Request	16	Success	574	137	16	2
14	20:19:57.979	Thread Group 1...	HTTP Request	16	Success	574	137	16	1
15	20:19:58.138	Thread Group 1...	HTTP Request	31	Success	574	137	31	2
16	20:19:58.182	Thread Group 1...	HTTP Request	15	Success	574	137	15	2
17	20:19:58.282	Thread Group 1...	HTTP Request	26	Success	574	137	25	2
18	20:19:58.397	Thread Group 1...	HTTP Request	21	Success	574	137	21	1
19	20:19:58.537	Thread Group 1...	HTTP Request	37	Success	574	137	37	2
20	20:19:58.583	Thread Group 1...	HTTP Request	14	Success	574	137	14	1
21	20:19:58.687	Thread Group 1...	HTTP Request	18	Success	574	137	18	3
22	20:19:58.784	Thread Group 1...	HTTP Request	14	Success	574	137	14	1
23	20:19:58.936	Thread Group 1...	HTTP Request	12	Success	574	137	12	1
24	20:19:59.089	Thread Group 1...	HTTP Request	10	Success	574	137	10	1
25	20:19:59.088	Thread Group 1...	HTTP Request	28	Success	574	137	25	2
26	20:19:59.187	Thread Group 1...	HTTP Request	14	Success	574	137	14	1

## JMeter Select student dengan kondisi gagal ditemukan

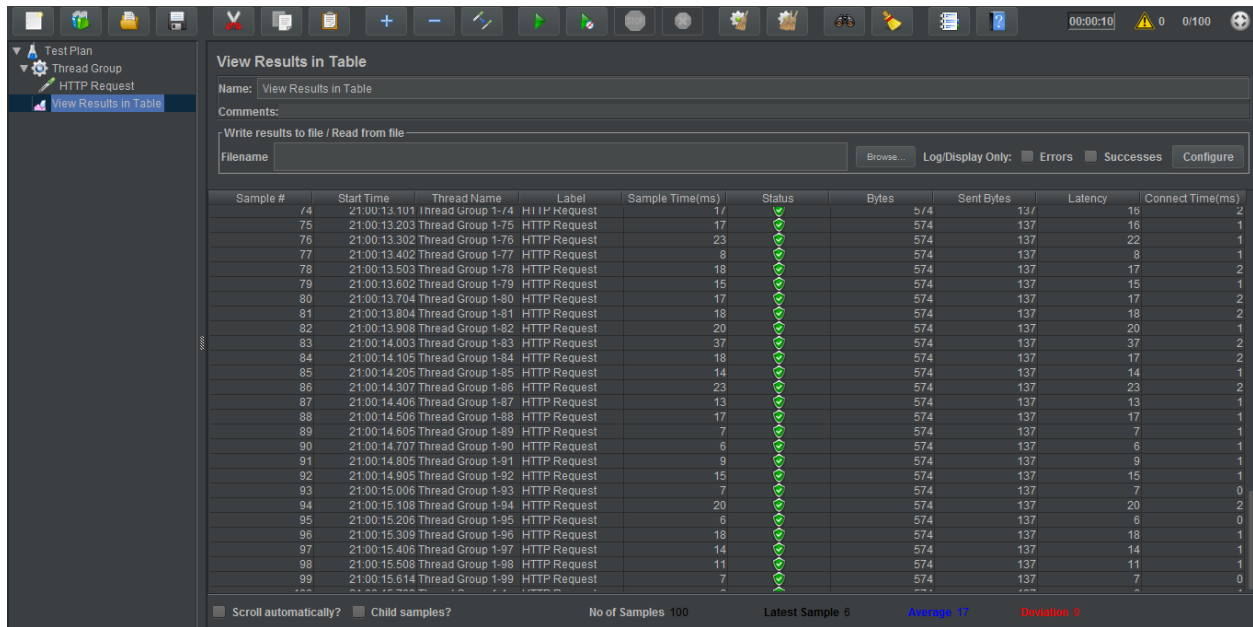


The screenshot shows the JMeter 'View Results in Table' window. The table displays 26 samples, all with a status of 'Failure' (red X). The columns include Sample #, Start Time, Thread Name, Label, Sample Time (ms), Status, Bytes, Sent Bytes, Latency, and Connect Time (ms). The status bar at the bottom indicates 'No of Samples: 100', 'Latest Sample: 24', 'Average: 22', and 'Deviation: 8'.

Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	20:21:48.726	Thread Group 1...	HTTP Request	26	Failure	551	137	26	3
2	20:21:48.827	Thread Group 1...	HTTP Request	25	Failure	551	137	25	2
3	20:21:48.938	Thread Group 1...	HTTP Request	22	Failure	551	137	21	2
4	20:21:49.039	Thread Group 1...	HTTP Request	8	Failure	551	137	8	0
5	20:21:49.130	Thread Group 1...	HTTP Request	25	Failure	551	137	24	2
6	20:21:49.229	Thread Group 1...	HTTP Request	31	Failure	551	137	31	2
7	20:21:49.339	Thread Group 1...	HTTP Request	20	Failure	551	137	20	2
8	20:21:49.442	Thread Group 1...	HTTP Request	24	Failure	551	137	23	2
9	20:21:49.544	Thread Group 1...	HTTP Request	11	Failure	551	137	11	1
10	20:21:49.628	Thread Group 1...	HTTP Request	16	Failure	551	137	16	1
11	20:21:49.743	Thread Group 1...	HTTP Request	20	Failure	551	137	20	2
12	20:21:49.843	Thread Group 1...	HTTP Request	25	Failure	551	137	25	1
13	20:21:49.931	Thread Group 1...	HTTP Request	33	Failure	551	137	33	2
14	20:21:50.032	Thread Group 1...	HTTP Request	37	Failure	551	137	36	2
15	20:21:50.143	Thread Group 1...	HTTP Request	11	Failure	551	137	11	1
16	20:21:50.249	Thread Group 1...	HTTP Request	15	Failure	551	137	15	1
17	20:21:50.331	Thread Group 1...	HTTP Request	11	Failure	551	137	11	0
18	20:21:50.431	Thread Group 1...	HTTP Request	15	Failure	551	137	15	1
19	20:21:50.554	Thread Group 1...	HTTP Request	18	Failure	551	137	18	1
20	20:21:50.650	Thread Group 1...	HTTP Request	23	Failure	551	137	23	1
21	20:21:50.737	Thread Group 1...	HTTP Request	16	Failure	551	137	15	1
22	20:21:50.833	Thread Group 1...	HTTP Request	28	Failure	551	137	28	3
23	20:21:50.958	Thread Group 1...	HTTP Request	27	Failure	551	137	26	2
24	20:21:51.053	Thread Group 1...	HTTP Request	11	Failure	551	137	11	0
25	20:21:51.133	Thread Group 1...	HTTP Request	28	Failure	551	137	27	3
26	20:21:51.236	Thread Group 1...	HTTP Request	22	Failure	551	137	22	1

Optimisasi dapat dilakukan pada basis data dengan cara membuat kolom npm menjadi unik dan indeks.

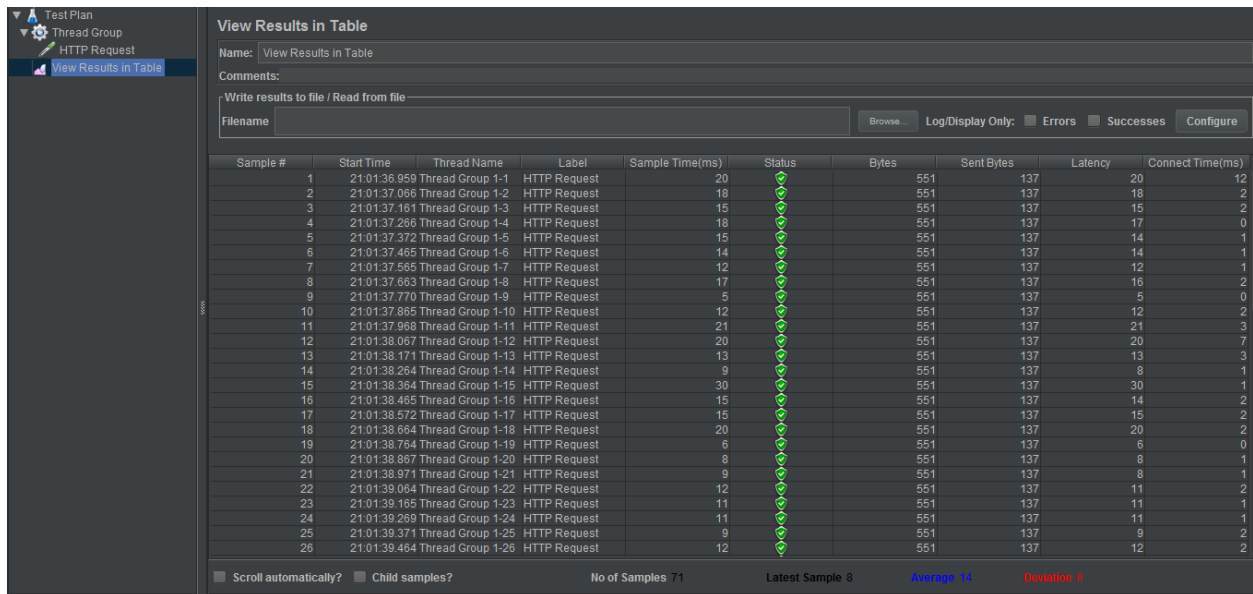
## JMeter Select student berhasil ditemukan



The screenshot shows the JMeter 'View Results in Table' window. The table displays 100 samples, all with a status of 'Success' (green checkmark). The columns include Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, Sent Bytes, Latency, and Connect Time(ms). The status bar at the bottom indicates 'No of Samples: 100', 'Latest Sample: 6', 'Average: 17', and 'Deviation: 9'.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
74	21:00:13.101	Thread Group 1-74	HTTP Request	17	Success	574	137	18	2
75	21:00:13.203	Thread Group 1-75	HTTP Request	17	Success	574	137	18	1
76	21:00:13.302	Thread Group 1-76	HTTP Request	23	Success	574	137	22	1
77	21:00:13.402	Thread Group 1-77	HTTP Request	8	Success	574	137	8	1
78	21:00:13.503	Thread Group 1-78	HTTP Request	18	Success	574	137	17	2
79	21:00:13.602	Thread Group 1-79	HTTP Request	15	Success	574	137	15	1
80	21:00:13.704	Thread Group 1-80	HTTP Request	17	Success	574	137	17	2
81	21:00:13.804	Thread Group 1-81	HTTP Request	18	Success	574	137	18	2
82	21:00:13.908	Thread Group 1-82	HTTP Request	20	Success	574	137	20	1
83	21:00:14.003	Thread Group 1-83	HTTP Request	37	Success	574	137	37	2
84	21:00:14.105	Thread Group 1-84	HTTP Request	18	Success	574	137	17	2
85	21:00:14.205	Thread Group 1-85	HTTP Request	14	Success	574	137	14	1
86	21:00:14.307	Thread Group 1-86	HTTP Request	23	Success	574	137	23	2
87	21:00:14.406	Thread Group 1-87	HTTP Request	13	Success	574	137	13	1
88	21:00:14.506	Thread Group 1-88	HTTP Request	17	Success	574	137	17	1
89	21:00:14.605	Thread Group 1-89	HTTP Request	7	Success	574	137	7	1
90	21:00:14.707	Thread Group 1-90	HTTP Request	6	Success	574	137	6	1
91	21:00:14.805	Thread Group 1-91	HTTP Request	9	Success	574	137	9	1
92	21:00:14.905	Thread Group 1-92	HTTP Request	15	Success	574	137	15	1
93	21:00:15.006	Thread Group 1-93	HTTP Request	7	Success	574	137	7	0
94	21:00:15.108	Thread Group 1-94	HTTP Request	20	Success	574	137	20	2
95	21:00:15.206	Thread Group 1-95	HTTP Request	6	Success	574	137	6	0
96	21:00:15.309	Thread Group 1-96	HTTP Request	18	Success	574	137	18	1
97	21:00:15.406	Thread Group 1-97	HTTP Request	14	Success	574	137	14	1
98	21:00:15.508	Thread Group 1-98	HTTP Request	11	Success	574	137	11	1
99	21:00:15.614	Thread Group 1-99	HTTP Request	7	Success	574	137	7	0

## JMeter Select student tidak berhasil ditemukan



The screenshot shows the JMeter 'View Results in Table' window. The table displays 26 samples, all with a status of 'Failure' (red X). The columns include Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, Sent Bytes, Latency, and Connect Time(ms). The status bar at the bottom indicates 'No of Samples: 71', 'Latest Sample: 8', 'Average: 14', and 'Deviation: 9'.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	21:01:36.959	Thread Group 1-1	HTTP Request	20	Failure	551	137	20	12
2	21:01:37.066	Thread Group 1-2	HTTP Request	18	Failure	551	137	18	2
3	21:01:37.161	Thread Group 1-3	HTTP Request	15	Failure	551	137	15	2
4	21:01:37.266	Thread Group 1-4	HTTP Request	18	Failure	551	137	17	0
5	21:01:37.372	Thread Group 1-5	HTTP Request	15	Failure	551	137	14	1
6	21:01:37.465	Thread Group 1-6	HTTP Request	14	Failure	551	137	14	1
7	21:01:37.565	Thread Group 1-7	HTTP Request	12	Failure	551	137	12	1
8	21:01:37.663	Thread Group 1-8	HTTP Request	17	Failure	551	137	16	2
9	21:01:37.770	Thread Group 1-9	HTTP Request	5	Failure	551	137	5	0
10	21:01:37.885	Thread Group 1-10	HTTP Request	12	Failure	551	137	12	2
11	21:01:37.998	Thread Group 1-11	HTTP Request	21	Failure	551	137	21	3
12	21:01:38.067	Thread Group 1-12	HTTP Request	20	Failure	551	137	20	7
13	21:01:38.171	Thread Group 1-13	HTTP Request	13	Failure	551	137	13	3
14	21:01:38.284	Thread Group 1-14	HTTP Request	9	Failure	551	137	8	1
15	21:01:38.384	Thread Group 1-15	HTTP Request	30	Failure	551	137	30	1
16	21:01:38.485	Thread Group 1-16	HTTP Request	15	Failure	551	137	14	2
17	21:01:38.572	Thread Group 1-17	HTTP Request	15	Failure	551	137	15	2
18	21:01:38.664	Thread Group 1-18	HTTP Request	20	Failure	551	137	20	2
19	21:01:38.764	Thread Group 1-19	HTTP Request	6	Failure	551	137	6	0
20	21:01:38.867	Thread Group 1-20	HTTP Request	8	Failure	551	137	8	1
21	21:01:38.971	Thread Group 1-21	HTTP Request	9	Failure	551	137	8	1
22	21:01:39.064	Thread Group 1-22	HTTP Request	12	Failure	551	137	11	2
23	21:01:39.165	Thread Group 1-23	HTTP Request	11	Failure	551	137	11	1
24	21:01:39.259	Thread Group 1-24	HTTP Request	11	Failure	551	137	11	1
25	21:01:39.371	Thread Group 1-25	HTTP Request	9	Failure	551	137	9	2
26	21:01:39.464	Thread Group 1-26	HTTP Request	12	Failure	551	137	12	2