

selectStudent Berhasil

The screenshot shows the Apache JMeter 4.0 interface with the 'View Results in Table' window open. The test plan 'Test Plan APAP' contains a 'Thread Group' with an 'HTTP Request' sampler. The results table displays 100 samples, all with a status of 'Success' (green checkmark). The summary statistics at the bottom indicate: No of Samples: 100, Latest Sample: 17, Average: 19, and Deviation: 4.

Sample #	Start Time	Thread Name	Label	Sample Time(m)	Status	Bytes	Sent Bytes	Latency	Connect Time(m)
75	16.28.26.340	Thread Group 1...	HTTP Request	16	Success	372	202	16	1
76	16.28.26.441	Thread Group 1...	HTTP Request	16	Success	372	202	16	0
77	16.28.26.540	Thread Group 1...	HTTP Request	18	Success	372	202	18	1
78	16.28.26.640	Thread Group 1...	HTTP Request	17	Success	372	202	16	1
79	16.28.26.740	Thread Group 1...	HTTP Request	17	Success	372	202	17	1
80	16.28.26.840	Thread Group 1...	HTTP Request	20	Success	372	202	20	1
81	16.28.26.940	Thread Group 1...	HTTP Request	18	Success	372	202	17	2
82	16.28.27.040	Thread Group 1...	HTTP Request	18	Success	372	202	18	1
83	16.28.27.141	Thread Group 1...	HTTP Request	16	Success	372	202	16	1
84	16.28.27.241	Thread Group 1...	HTTP Request	17	Success	372	202	17	1
85	16.28.27.340	Thread Group 1...	HTTP Request	17	Success	372	202	17	1
86	16.28.27.440	Thread Group 1...	HTTP Request	18	Success	372	202	17	1
87	16.28.27.541	Thread Group 1...	HTTP Request	21	Success	372	202	20	1
88	16.28.27.641	Thread Group 1...	HTTP Request	20	Success	372	202	20	1
89	16.28.27.740	Thread Group 1...	HTTP Request	20	Success	372	202	20	1
90	16.28.27.840	Thread Group 1...	HTTP Request	18	Success	372	202	17	1
91	16.28.27.951	Thread Group 1...	HTTP Request	22	Success	372	202	22	1
92	16.28.28.040	Thread Group 1...	HTTP Request	21	Success	372	202	20	1
93	16.28.28.140	Thread Group 1...	HTTP Request	18	Success	372	202	18	1
94	16.28.28.240	Thread Group 1...	HTTP Request	15	Success	372	202	15	1
95	16.28.28.341	Thread Group 1...	HTTP Request	20	Success	372	202	20	1
96	16.28.28.441	Thread Group 1...	HTTP Request	17	Success	372	202	17	1
97	16.28.28.540	Thread Group 1...	HTTP Request	19	Success	372	202	19	1
98	16.28.28.640	Thread Group 1...	HTTP Request	17	Success	372	202	16	1
99	16.28.28.741	Thread Group 1...	HTTP Request	17	Success	372	202	17	1
100	16.28.28.841	Thread Group 1...	HTTP Request	17	Success	372	202	17	1

SelectStudent Gagal

The screenshot shows the Apache JMeter 4.0 interface with the 'View Results in Table' window open. The test plan 'Test Plan APAP' contains a 'Thread Group' with an 'HTTP Request' sampler. The results table displays 100 samples, all with a status of 'Success' (green checkmark). The summary statistics at the bottom indicate: No of Samples: 100, Latest Sample: 16, Average: 16, and Deviation: 5.

Sample #	Start Time	Thread Name	Label	Sample Time(m)	Status	Bytes	Sent Bytes	Latency	Connect Time(m)
75	16.30.34.405	Thread Group 1-75	HTTP Request	37	Success	333	193	37	1
76	16.30.34.505	Thread Group 1-76	HTTP Request	7	Success	333	193	7	0
77	16.30.34.609	Thread Group 1-77	HTTP Request	7	Success	333	193	7	1
78	16.30.34.689	Thread Group 1-78	HTTP Request	6	Success	333	193	6	0
79	16.30.34.805	Thread Group 1-79	HTTP Request	16	Success	333	193	16	1
80	16.30.34.905	Thread Group 1-80	HTTP Request	9	Success	333	193	9	0
81	16.30.34.989	Thread Group 1-81	HTTP Request	17	Success	333	193	17	1
82	16.30.35.105	Thread Group 1-82	HTTP Request	15	Success	333	193	15	1
83	16.30.35.205	Thread Group 1-83	HTTP Request	15	Success	333	193	15	1
84	16.30.35.305	Thread Group 1-84	HTTP Request	15	Success	333	193	15	1
85	16.30.35.405	Thread Group 1-85	HTTP Request	23	Success	333	193	23	1
86	16.30.35.505	Thread Group 1-86	HTTP Request	15	Success	333	193	15	1
87	16.30.35.605	Thread Group 1-87	HTTP Request	16	Success	333	193	16	1
88	16.30.35.705	Thread Group 1-88	HTTP Request	15	Success	333	193	14	1
89	16.30.35.805	Thread Group 1-89	HTTP Request	16	Success	333	193	16	1
90	16.30.35.905	Thread Group 1-90	HTTP Request	15	Success	333	193	15	1
91	16.30.36.005	Thread Group 1-91	HTTP Request	18	Success	333	193	18	1
92	16.30.36.105	Thread Group 1-92	HTTP Request	16	Success	333	193	15	1
93	16.30.36.214	Thread Group 1-93	HTTP Request	16	Success	333	193	16	1
94	16.30.36.309	Thread Group 1-94	HTTP Request	19	Success	333	193	19	1
95	16.30.36.412	Thread Group 1-95	HTTP Request	13	Success	333	193	13	0
96	16.30.36.510	Thread Group 1-96	HTTP Request	13	Success	333	193	13	1
97	16.30.36.606	Thread Group 1-97	HTTP Request	16	Success	333	193	16	1
98	16.30.36.715	Thread Group 1-98	HTTP Request	15	Success	333	193	15	1
99	16.30.36.819	Thread Group 1-99	HTTP Request	15	Success	333	193	15	1
100	16.30.36.912	Thread Group 1-100	HTTP Request	16	Success	333	193	16	1

selectAllStudents dengan Number of threads 100

HTTP Request.jmx (Q:\kuliah\2 Aplikasi Penrograman Perusahaan\software\apache-jmeter-4.0\bin\HTTP Request.jmx) - Apache JMeter (4.0 r1823414)

File Edit Search Run Options Help

Test Plan APAP
Thread Group
HTTP Request
View Results in Table

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(...)
75	16:23:09.282	Thread Group 1...	HTTP Request	48	✓	3658	194	48	1
76	16:23:09.376	Thread Group 1...	HTTP Request	49	✓	3658	194	49	1
77	16:23:09.469	Thread Group 1...	HTTP Request	54	✓	3658	194	54	1
78	16:23:09.570	Thread Group 1...	HTTP Request	52	✓	3658	194	51	1
79	16:23:09.672	Thread Group 1...	HTTP Request	50	✓	3658	194	49	1
80	16:23:09.781	Thread Group 1...	HTTP Request	46	✓	3658	194	45	1
81	16:23:09.875	Thread Group 1...	HTTP Request	56	✓	3658	194	56	1
82	16:23:09.969	Thread Group 1...	HTTP Request	61	✓	3658	194	60	1
83	16:23:10.068	Thread Group 1...	HTTP Request	30	✓	3658	194	30	1
84	16:23:10.168	Thread Group 1...	HTTP Request	37	✓	3658	194	37	1
85	16:23:10.268	Thread Group 1...	HTTP Request	32	✓	3658	194	32	1
86	16:23:10.368	Thread Group 1...	HTTP Request	26	✓	3658	194	26	1
87	16:23:10.469	Thread Group 1...	HTTP Request	65	✓	3658	194	65	1
88	16:23:10.569	Thread Group 1...	HTTP Request	58	✓	3658	194	57	1
89	16:23:10.668	Thread Group 1...	HTTP Request	60	✓	3658	194	60	1
90	16:23:10.769	Thread Group 1...	HTTP Request	68	✓	3658	194	68	0
91	16:23:10.868	Thread Group 1...	HTTP Request	35	✓	3658	194	35	1
92	16:23:10.968	Thread Group 1...	HTTP Request	39	✓	3658	194	39	1
93	16:23:11.068	Thread Group 1...	HTTP Request	51	✓	3658	194	51	1
94	16:23:11.168	Thread Group 1...	HTTP Request	52	✓	3658	194	52	1
95	16:23:11.269	Thread Group 1...	HTTP Request	45	✓	3658	194	44	1
96	16:23:11.368	Thread Group 1...	HTTP Request	60	✓	3658	194	60	1
97	16:23:11.468	Thread Group 1...	HTTP Request	61	✓	3658	194	61	1
98	16:23:11.568	Thread Group 1...	HTTP Request	52	✓	3658	194	52	1
99	16:23:11.668	Thread Group 1...	HTTP Request	46	✓	3658	194	46	1
100	16:23:11.768	Thread Group 1...	HTTP Request	34	✓	3658	194	34	0

☒ Scroll automatically? ☐ Child samples? No of Samples: 100 Latest Sample: 34 Average: 238 Deviation: 464

Screenshot hasil testing pada unit testing pada unit testing

Run: ApapTutorial05Application StudentServiceDatabaseTest.updateStudent

1 test passed - 251ms

StudentServiceDatabaseTest (com.v 251ms)

updateStudent 251ms

```
"C:\Program Files\Java\jdk1.8.0_161\bin\java" ...
15:02:17.782 [main] INFO com.example.service.StudentServiceDatabase - Student 1506737823updated
15:02:17.818 [main] INFO com.example.service.StudentServiceDatabase - Student 1506737823updated

Process finished with exit code 0
```

Ringkasan:

Pada tutorial ini kita mempelajari mengenai unit testing melanjutkan tutorial yang sebelumnya.

```
@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
}
```

Kode program diatas merupakan kode proram yang digunakan untuk memilih semua data mahasiswa. Blok kode dibawah komentar **Given** merupakan kode program yang digunakan untuk menambahkan data ke dalam database. Data mahasiswa ditambahkan melalui variabel studentModel. Kemudian variael check digunakan untuk membandingkan data dengan variabel studentModel. Kemudian pada kode program dibawah komentar **then** merupakan kode program yang berfungsi sebagai kode program testing.

Optimasi

Kode program viewall select npm, gpa, name from student merupakan query yang paling optimal karena apabila kita menggunakan select * from student hasilnya malah lebihlama dibandingkan query sebelumnya. Maka pada tutorial ini query yang sudah ada sudah cukup optimal.