

Pelajaran

Pada tutorial 6 ini belajar mengenai JUnit dan JMeter. Tools ini berfungsi untuk melakukan test sebuah sistem skala enterprise. JUnit disini berfungsi untuk mengetes dari fungsikodenya, misal kode tersebut bekerja apa return-nya bisa di dilakukan tes dengan JUnit. JMeter disini berguna untuk melakukan tes beban pada servernya dengan melakukan thread untuk membuka sebuah situs dari web server. JMeter dapat menghitung berapa waktu proses yang dilakukan webserver untuk menerima dan memberikan proses halaman yang diminta.

Latihan Unit Testing

1. Pada `selectAllStudent()` dilakukan pengetesan berdasarkan fungsi yang mewakili dari method tersebut. Method tersebut berfungsi untuk mengambil semua data student di suatu penyimpanan data seperti database. Namun, dalam tes ini hanya memanfaatkan sebuah List sebagai pengganti database yang cara kerjanya sama layaknya mengakses data dari database.

Pada `Given` ini meninisialisasikan kebutuhan data, data yang dibutuhkan seperti pembuatan List `studentModels` dan `checks`. kedua list tersebut dibuatlah dan ditambahkan masing-masing objek `StudentModel` dengan `npm`, `nama`, dan `gpa` yang sama. Kedua List dan objek tersebut untuk perbandingan. `BDDMockito.given()` ini berfungsi untuk mengecek pemanggilan sebuah method dan `willReturn()` untuk mengecek kembali dari method yang dipanggil.

Pada `When` tersebut terdapat List dengan nama `test` dan berisi data dari `studentService.selectAllStudent()`. Method ini dipanggil dan seolah-olah method tersebut bekerja sebagai pengambil semua data dari database. Namun, method ini hanya umpama dan sudah dihandle pada `BDDMockito.given()`.

Pada `Then` ini adalah untuk mengecek apakah dari masing-masing variable pada variabel objek test sesuai dengan matcher yang diinisialisasikan pada method `assertThat()`. Jika tidak cocok maka JUnit akan menghasilkan error.

Hasil dari method tersebut berhasil melakukan semua tes `assert` yang dilakukan.

2. Pada `addStudent()` ini berfungsi sebagai penambah data student kedalam database. Pada `Given` terdapat objek `StudentModel` yang dibuat 2 objek yaitu `studentModel` dan `check`. `studentModel` disini untuk dilakukan testing untuk penambahan sebuah data, dan `check` untuk membandingkan sebuah data.

Pada `When` terdapat variabel test yang bernilai Boolean yang berasal dari method `addStudent()` nilai. Pada `Then` ini akan mengecek dan mencoba data dari `studentMapper` dari method `addStudent()`. Pada `assert` ini akan mengecek test tersebut apakah mereturn `true`, jika iya maka JUnit tidak menampilkan error, jika tidak JUnit akan menampilkan error.

Hasil dari method tersebut berhasil melakukan semua tes `assert` yang dilakukan.

3. `deleteStudent()`

```

@Test
public void deleteStudent() {
    // given
    List<StudentModel> students = new ArrayList<>();
    StudentModel student1 = new StudentModel("1506737823", "Chanek", 3.5);
    StudentModel student2 = new StudentModel("1506737824", "Chanek J", 3.5);
    students.add(student1);
    students.add(student2);
    //BDDMockito.given(studentMapper.deleteStudent("1506737823")).willReturn(true);
    BDDMockito.given(studentMapper.deleteStudent(Mockito.anyString())).willReturn(new Answer<Boolean>() {
        @Override
        public Boolean answer(InvocationOnMock invocation) throws Throwable {
            String npm = invocation.getArgumentAt(0, String.class);
            for (StudentModel studentModel : students) {
                if (studentModel.getNpm().equals(npm)) {
                    students.remove(studentModel);
                    return true;
                }
            }
            return false;
        }
    });

    //when
    public Boolean answer(InvocationOnMock invocation) throws Throwable {
        String npm = invocation.getArgumentAt(0, String.class);
        for (StudentModel studentModel : students) {
            if (studentModel.getNpm().equals(npm)) {
                students.remove(studentModel);
                return true;
            }
        }
        return false;
    }

    });

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

    // then
    //BDDMockito.then(studentMapper).should().deleteStudent("1506737823");
    assertThat(test, equalTo(true));
    assertThat(students.size(), equalTo(1));
    assertThat(students.contains(student1), equalTo(false));
}

```

Console Tasks Progress JUnit

Finished after 2.034 seconds

Runs: 5/5 Errors: 0 Failures: 0

com.example.service.StudentServiceDatabaseTest [Runner: JUnit 4] (0.971 s) Fail

- deleteStudent (0.903 s)
- selectStudent (0.023 s)
- selectAllStudents (0.003 s)
- addStudent (0.037 s)
- updateStudent (0.005 s)

updateStudent()

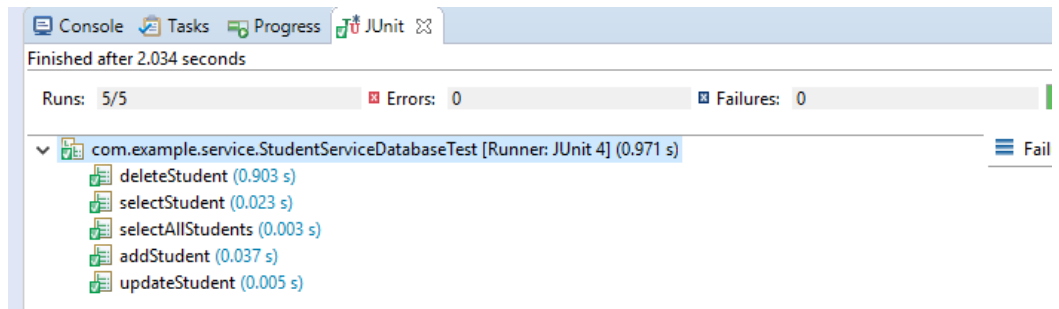
```

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

    // when
    boolean test = studentService.updateStudent(student);

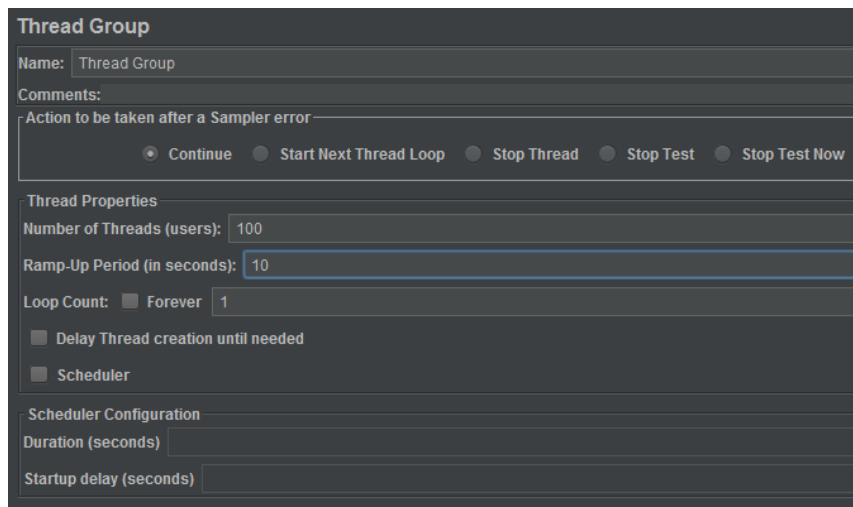
    // then
    BDDMockito.then(studentMapper).should().updateStudent(student);
    assertThat(test, equalTo(true));
}

```

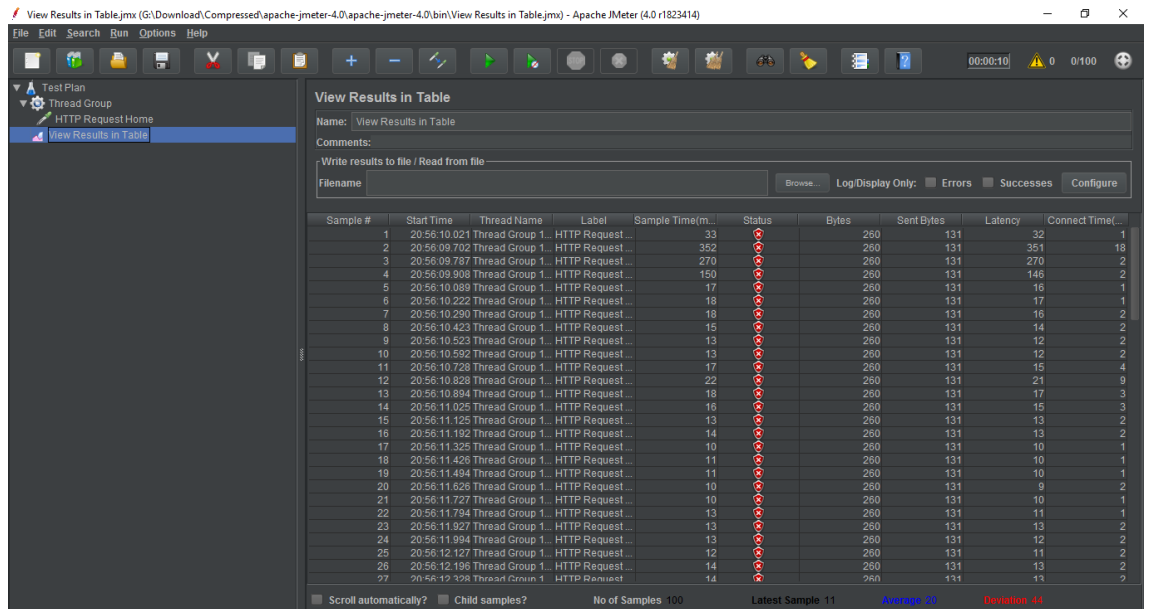


Latihan Load Testing

1.



Select Student (Berhasil)



Select Student (Gagal)

View Results in Table.jmx (G:\Download\Compressed\apache-jmeter-4.0\apache-jmeter-4.0\bin\View Results in Table.jmx) - Apache JMeter (4.0 r1823414)

File Edit Search Run Options Help

Test Plan
Thread Group
HTTP Request Home
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	Connec Time(m)
74	20:58:01.875	Thread Group 1..	HTTP Request...	10	✓	265	136	18	4
75	20:58:01.975	Thread Group 1..	HTTP Request...	19	✓	265	136	18	4
76	20:58:02.115	Thread Group 1..	HTTP Request...	29	✓	265	136	22	2
77	20:58:02.195	Thread Group 1..	HTTP Request...	12	✓	265	136	11	2
78	20:58:02.277	Thread Group 1..	HTTP Request...	16	✓	265	136	16	6
79	20:58:02.396	Thread Group 1..	HTTP Request...	18	✓	265	136	18	4
80	20:58:02.506	Thread Group 1..	HTTP Request...	12	✓	265	136	11	2
81	20:58:02.592	Thread Group 1..	HTTP Request...	14	✓	265	136	14	2
82	20:58:02.686	Thread Group 1..	HTTP Request...	16	✓	265	136	16	6
83	20:58:02.776	Thread Group 1..	HTTP Request...	16	✓	265	136	15	5
84	20:58:02.917	Thread Group 1..	HTTP Request...	15	✓	265	136	12	2
85	20:58:02.952	Thread Group 1..	HTTP Request...	14	✓	265	136	13	4
86	20:58:03.086	Thread Group 1..	HTTP Request...	12	✓	265	136	11	2
87	20:58:03.178	Thread Group 1..	HTTP Request...	14	✓	265	136	13	4
88	20:58:03.284	Thread Group 1..	HTTP Request...	18	✓	265	136	18	5
89	20:58:03.342	Thread Group 1..	HTTP Request...	20	✓	265	136	19	10
90	20:58:03.479	Thread Group 1..	HTTP Request...	15	✓	265	136	14	4
91	20:58:03.596	Thread Group 1..	HTTP Request...	15	✓	265	136	15	5
92	20:58:03.697	Thread Group 1..	HTTP Request...	24	✓	265	136	20	10
93	20:58:03.786	Thread Group 1..	HTTP Request...	11	✓	265	136	11	2
94	20:58:03.842	Thread Group 1..	HTTP Request...	11	✓	265	136	11	2
95	20:58:03.980	Thread Group 1..	HTTP Request...	24	✓	265	136	23	3
96	20:58:04.082	Thread Group 1..	HTTP Request...	13	✓	265	136	11	2
97	20:58:04.187	Thread Group 1..	HTTP Request...	11	✓	265	136	11	2
98	20:58:04.281	Thread Group 1..	HTTP Request...	12	✓	265	136	12	3
99	20:58:04.380	Thread Group 1..	HTTP Request...	14	✓	265	136	12	3
100	20:58:04.479	Thread Group 1..	HTTP Request...	12	✓	265	136	11	2

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 12 Average 15 Deviation 11

Select All Student

View Results in Table.jmx (G:\Download\Compressed\apache-jmeter-4.0\apache-jmeter-4.0\bin\View Results in Table.jmx) - Apache JMeter (4.0 r1823414)

File Edit Search Run Options Help

Test Plan
Thread Group
HTTP Request Home
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	Connec Time(m)
58	20:55:16.409	Thread Group 1..	HTTP Request...	219	✓	8421	129	209	2
59	20:55:16.309	Thread Group 1..	HTTP Request...	303	✓	8421	129	299	2
60	20:55:16.410	Thread Group 1..	HTTP Request...	269	✓	8421	129	268	2
61	20:55:16.510	Thread Group 1..	HTTP Request...	248	✓	8421	129	247	2
62	20:55:16.609	Thread Group 1..	HTTP Request...	274	✓	8421	129	273	3
63	20:55:16.710	Thread Group 1..	HTTP Request...	256	✓	8421	129	255	2
64	20:55:16.810	Thread Group 1..	HTTP Request...	252	✓	8421	129	251	2
65	20:55:16.911	Thread Group 1..	HTTP Request...	284	✓	8421	129	283	1
66	20:55:17.013	Thread Group 1..	HTTP Request...	273	✓	8421	129	272	2
67	20:55:17.112	Thread Group 1..	HTTP Request...	269	✓	8421	129	268	1
68	20:55:17.261	Thread Group 1..	HTTP Request...	272	✓	8421	129	271	2
69	20:55:17.312	Thread Group 1..	HTTP Request...	281	✓	8421	129	280	1
70	20:55:17.412	Thread Group 1..	HTTP Request...	256	✓	8421	129	255	2
71	20:55:17.513	Thread Group 1..	HTTP Request...	260	✓	8421	129	258	2
72	20:55:17.612	Thread Group 1..	HTTP Request...	274	✓	8421	129	273	2
73	20:55:17.713	Thread Group 1..	HTTP Request...	256	✓	8421	129	255	2
74	20:55:17.813	Thread Group 1..	HTTP Request...	308	✓	8421	129	307	2
75	20:55:17.914	Thread Group 1..	HTTP Request...	302	✓	8421	129	301	2
76	20:55:18.014	Thread Group 1..	HTTP Request...	254	✓	8421	129	253	2
77	20:55:18.114	Thread Group 1..	HTTP Request...	262	✓	8421	129	261	2
78	20:55:18.214	Thread Group 1..	HTTP Request...	270	✓	8421	129	269	2
79	20:55:18.323	Thread Group 1..	HTTP Request...	253	✓	8421	129	252	2
80	20:55:18.415	Thread Group 1..	HTTP Request...	255	✓	8421	129	254	2
81	20:55:18.515	Thread Group 1..	HTTP Request...	272	✓	8421	129	271	2
82	20:55:18.615	Thread Group 1..	HTTP Request...	252	✓	8421	129	251	2
83	20:55:18.724	Thread Group 1..	HTTP Request...	255	✓	8421	129	254	2
84	20:55:18.825	Thread Group 1..	HTTP Request...	258	✓	8421	129	257	1

☐ Scroll automatically? ☐ Child samples? No of Samples 100 Latest Sample 255 Average 317 Deviation 220

Hasil optimasi yang dilakukan masih sama dengan hasil sebelum dilakukan, hal ini dikarenakan bisa jadi kecepatan antara data dengan server, kapasitas memory pada laptop, dan perangkat penyimpanan seperti HDD yang dapat mempengaruhi proses data yang dilakukan pada request ke server.