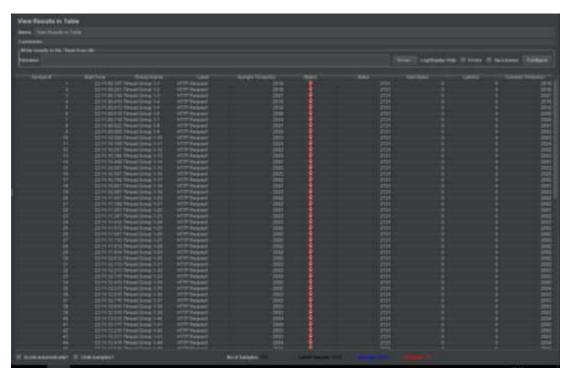
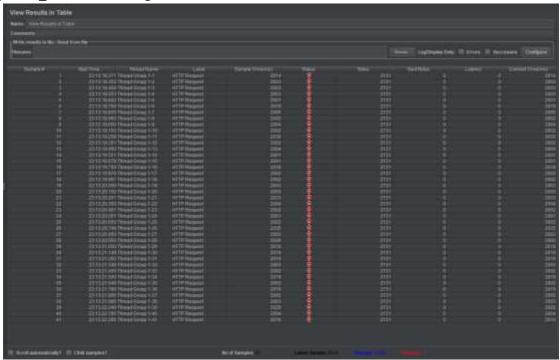
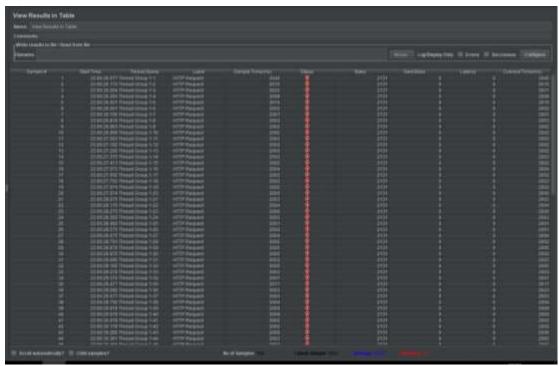
- 1. Screenshot hasil JMeter fitur soal latihan Load Testing nomor 1 sebelum dioptimasi (3 file)
 - jmeter_selectStudentBerhasil



• jmeter_selectStudentGagal



• jmeter_selectAllStudents



- 2. Screenshot hasil JMeter fitur soal latihan Load Testing nomor 1 sebelum dioptimasi (3 file)
 - jmeter_selectStudentBerhasil
 - jmeter_selectStudentGagal
 - jmeter_selectAllStudents



```
public void deleteStudent() {
    //Given
    StudentModel studentModel = new StudentModel ("1506737823", "Chanek", 3.5);
    StudentModel check = new StudentModel("1506737823", "Chanek", 3.5);
    8DDMockito.given(studentService.deleteStudent("1506737823")).willReturn(true);

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

    //Then
    8DDMockito.then(studentMapper).should().deleteStudent("1506737823");
    assertThat(test, equalTo(true)); // Check if Same
}

}

@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=studentMapper).should().updateStudent(studentModel);

    //Then

    BDDMockito.then(studentMapper).should().updateStudent(studentModel);

    assertThat(test, equalTo(true)); // Check if Same
}
```

Rangkuman

Melakukan testing sederhana terhadap service dengan plugin JUnit dan Mockito

Melakukan load testing sederhana dengan aplikasi Apache JMeter

melakukan load testing untuk mengukur performa web yang Anda buat. Load testing akan mengakses web Anda secara bersamaan (multithreading) dengan parameter yang telah ditentukan dan menghitung response time-nya. Pada tutorial ini kita akan menggunakan aplikasi Apache Jmeter untuk melakukan load testing