

PENGUJIAN PERANGKAT LUNAK

Fifin Ayu Mufarroha

Test Data dan Test Cases

- **Test data** :Input yang yang direncanakan digunakan oleh sistem.
- **Test cases** :Input yang digunakan untuk **menguji sistem** dan memprediksi output dari input jika sistem beroperasi sesuai dengan spesifikasi.

TEST CASE

“Hanya ada satu aturan untuk mendisain test cases: disain test cases harus melingkupi semua fitur, namun jangan membuat terlalu banyak test cases.”

Tsuneo Yamaura

Berdasar pada obyektifitas testing, kita harus melakukan disain tes yang memiliki kemungkinan tertinggi dalam menemukan *error yang kebanyakan terjadi*, dengan waktu dan usaha yang minimum.

2 cara tes produk hasil Rekayasa ..

- **Fungsi spesifikasi produk**

Cara : mendemonstrasikan tiap fungsi telah beroperasi secara penuh sesuai dengan yang diharapkan, dan sementara itu, pada saat yang bersamaan, dilakukan pencarian *error pada tiap fungsi*.

- **Operasi Internal**

Cara : memastikan semua komponen berjalan sebagaimana mestinya (Blax box dan White box)

Definisi Test Case

Test case merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya. Adapun kegunaan dari test case ini, adalah sebagai berikut:

- Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi – *Black Box Testing.*
- Untuk melakukan testing kesesuaian suatu komponen terhadap disain – *White Box Testing.*

Tujuan Test Case

1. Memberikan panduan kepada *tester* untuk melakukan pengujian aplikasi
2. Sebagai bahan masukan kepada tim pengembang aplikasi
3. Menjadi dasar pengembangan bagi pengerjaan proyek selanjutnya
4. Sebagai salah satu dokumen pendukung penyelesaian proyek IT

Sasaran Pengujian (Glen Myers) :

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. Test case yg baik adalah test case yg memiliki probabilitas tinggi untuk menemukan kesalahan yg belum pernah ditemukan sebelumnya.
3. Pengujian yg sukses adalah pengujian yg mengungkap semua kesalahan yg belum pernah ditemukan sebelumnya.

Contoh Test Case : Usecase

- Use cases didasarkan pada Unified Modeling Language (UML) dan dapat digambarkan dengan use-case diagram.
- Test case merupakan serangkaian pengujian masukan, eksekusi, dan hasil yang diharapkan dimana dikembangkan untuk tujuan tertentu. Tiga tahap untuk menghasilkan test cases:
 - Langkah Pertama – Generate Scenarios
 - Langkah kedua – Identify Test Case
 - Langkah ketiga – Identify Data Value to Test

Langkah 1 : Generate Scenarios

Scenario Name	Starting Flow	Alternate
Scenario I - Payment Berhasil	Basic Flow	
Scenario II - User memasukkan nominal melebihi balance	Basic Flow	A1
Scenario III - User tidak memasukkan nominal payment	Basic Flow	A2

Langkah 2 : Identity Test Case

Test Case ID	Scenario/Condition			Expected Result
		Memasukkan Payment	Payment	
RC 1	Update payment berhasil	V	V	Konfirmasi keberhasilan ditampilkan
RC 2	Update dengan Payment > Balance	V	V	Kembali ke main page
RC 3	Payment kosong	V	I	Pesan Kesalah ditampilkan, kembali ke form payment

Langkah 3: Identity Data Value to Test

Test Case ID	Scenario/Condition	Data		Expected Result
		AppNo	Payment	
RC 1	Update payment berhasil	25	500	Konfirmasi keberhasilan ditampilkan
RC 2	Update dengan Payment > Balance	25	4000 NB : Balance : 3500	Konfirmasi kesalahan ditampilkan, kembali ke halaman payment
RC 3	Payment kosong	25	Empty	Konfirmasi kesalahan ditampilkan, kembali ke halaman payment

Usecase : Login

Langkah 1 (Generate Scenario)

- **Basic Path**

User (Promotor dan Administrator) berada pada halaman form login. User mengisikan username pada textfield dan password pada passwordfield lalu menekan tombol login. Sistem mencari User pada Daftar User berdasarkan username dan password yang telah diisikan. User ditemukan, kemudian sistem menampilkan halaman profil User.

- **Alternate Path**

Alternate Path 1

Jika textfield username dan password field tidak diisi, kemudian User menekan tombol login, maka sistem akan menampilkan peringatan bahwa textfield username dan password field belum diisi.

Alternate Path 2

Jika textfield username tidak diisi, dan password field diisi kemudian User menekan tombol login, maka sistem akan menampilkan peringatan bahwa textfield username belum diisi

Alternate Path 3

Jika textfield username diisi dan passwordfield tidak diisi kemudian User menekan tombol login maka sistem akan menampilkan pesan eror passwordfield belum diisi

Alternate Path 4

Jika User tidak ditemukan, maka sistem akan menampilkan peringatan bahwa User tidak ditemukan di daftar user, dan User diminta mengisikan username dan password lagi.

Usecase : Login

Langkah 2 (Indentifikasi Test Case)

ID Skenario	Nama Skenario	Halaman login	Username field	Password field	Tombol Login	Hasil yang diharapkan
SKA-11	Login sukses	V	V	V	V	Halaman profil user
SKA-12	Login gagal, 2 field kosong	V	I	I	V	Peringatan username dan password belum diisi
SKA-13	Login gagal, username field kosong	V	I	V	V	Peringatan username belum diisi
SKA-14	Login gagal, password field kosong	V	V	I	V	Peringatan password belum diisi
SKA-15	Login gagal, user tidak ditemukan	V	V	V	V	Peringatan user tidak ditemukan Muncul halaman login lagi

Test Case dengan Data

Langkah 3 (Tes menggunakan Data)

ID Skenario	Nama Skenario	Halaman login	Username field	Password field	Tombol Login	Hasil yang diharapkan	Hasil Sistem (Berupa SS)
SKA-11	Login sukses	V	iPal90	1234567	V	Halaman profil user	Gambar SS
SKA-12	Login gagal, 2 field kosong	V	-	-	V	Peringatan username dan password belum diisi	Gambar SS
SKA-13	Login gagal, username field kosong	V	-	1234567	V	Peringatan username belum diisi	Gambar SS
SKA-14	Login gagal, password field kosong	V	iPal90	-	V	Peringatan password belum diisi	Gambar SS
SKA-15	Login gagal, user tidak ditemukan	V	iPal9	123456	V	<ul style="list-style-type: none"> Peringatan user tidak ditemukan Muncul halaman login lagi 	Gambar SS

Performance Test dengan JMeter

https://www.tutorialspoint.com/jmeter/jmeter_quick_guide.htm

JMeter

The Apache JMeter™ application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions



Stress Testing ?

- Stress testing adalah pengujian/testing untuk mengetahui kemampuan software dalam menangani kondisi yang tidak normal dari sisi volume/kuantitas
- Contoh: Seberapa kuatkah sebuah website (facebook.com,twiter.com) bisa menangani pengunjung dalam satu waktu secara bersamaan ?? (misalkan maksimal 10.000 pengunjung/detik)

Apache Jmeter ?

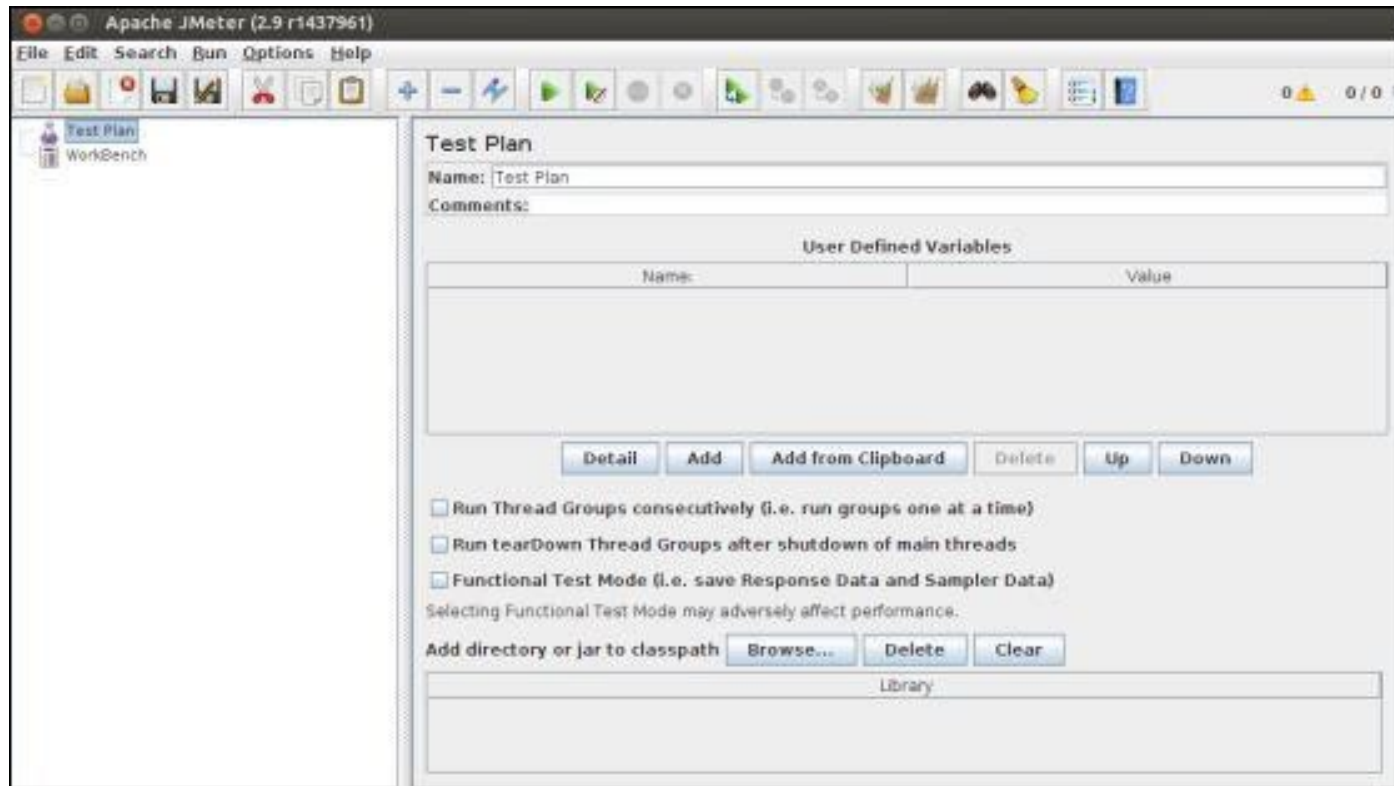
- Sebuah Tool/Alat yang digunakan untuk melakukan performace test pada sebuah software.
- Apache Jmeter dapat memberikan request dalam jumlah yg sangat banyak secara bersamaan dalam satu waktu pada server
- Apache Jmeter dapat memberikan Analisa dan Laporan dari hasil pengujian

Requirement Apache JMeter

- JRE (Java Runtime Enviroment) $\geq 8+$
- Operating Systems
 - Unix
 - Windows

Test Plan

- Test Plan adalah sebuah perencanaan atau skenario untuk melakukan sebuah testing



MEMBUAT PERFORMANCE TEST PLAN

1

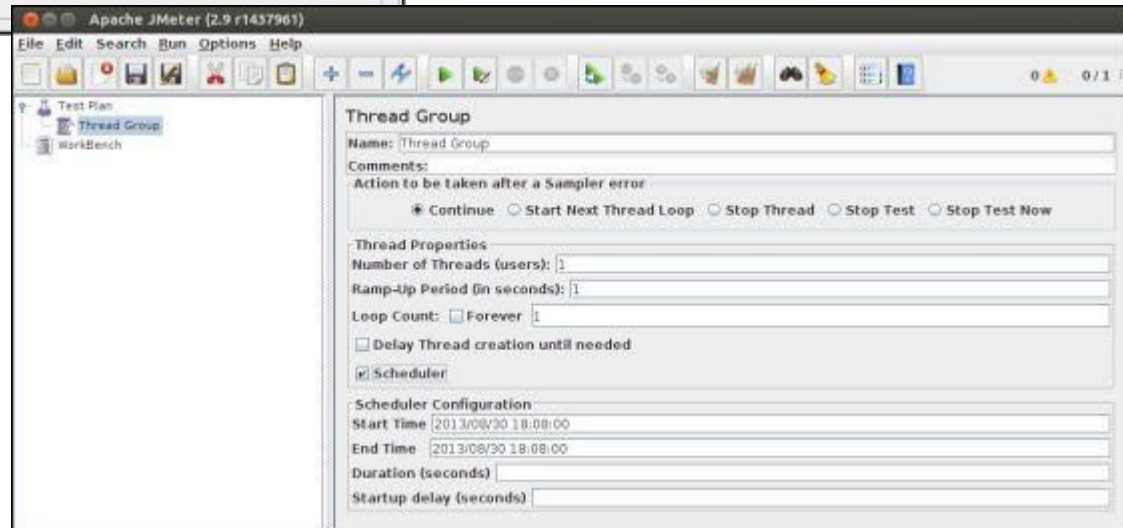
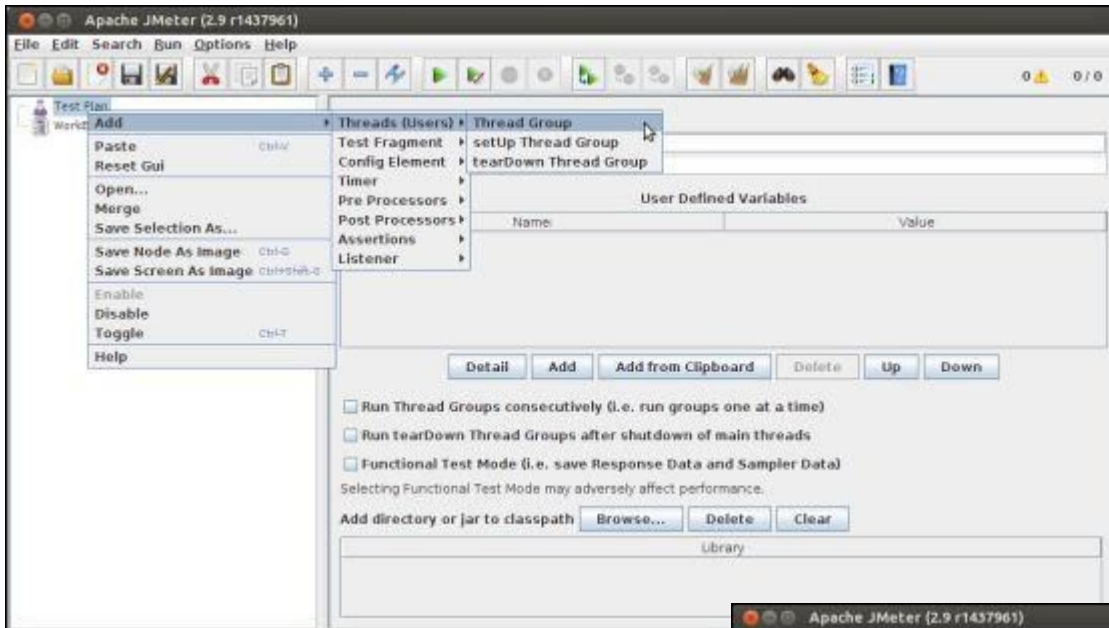
Add Thread Group :menambahkan trafik/user visitor ke dalam komponen yang mau di test. Langkahnya :

Klik Kanan Test Plan Add >Threads (Users) > Thread Group

Dalam kontrol panel Thread Group, Entri pada Thread Properties :

1. Number of threads (users) : isi berapa user/visitor yang akan mengakses web.
2. Ramp-Up period (in seconds) : isi berapa waktu delay antara user satu dengan yang lainnya dalam mengakses web.
3. Loop Count : waktu eksekusi, bertahap atau seterusnya.

MEMBUAT PERFORMANCE TEST PLAN



MEMBUAT PERFORMANCE TEST PLAN

- Apabila kita memasukkan *number of threads* = 10, *ramp-up period* = 10, dan *loop count* = 1, itu artinya ada 1 *thread* yang dijalankan setiap 1 detik.
- Apabila kita memasukkan *ramp-up period* = 100, itu artinya ada 1 *thread* yang dijalankan setiap 10 detik karena $100 \text{ ramp-up period} / 10 \text{ threads} = 10$.
- contoh lain, kalau *ramp-up period* = 200 dan *threads* = 10, artinya ada 1 *thread* yang dijalankan setiap 20 detik.

MEMBUAT PERFORMANCE TEST PLAN

2

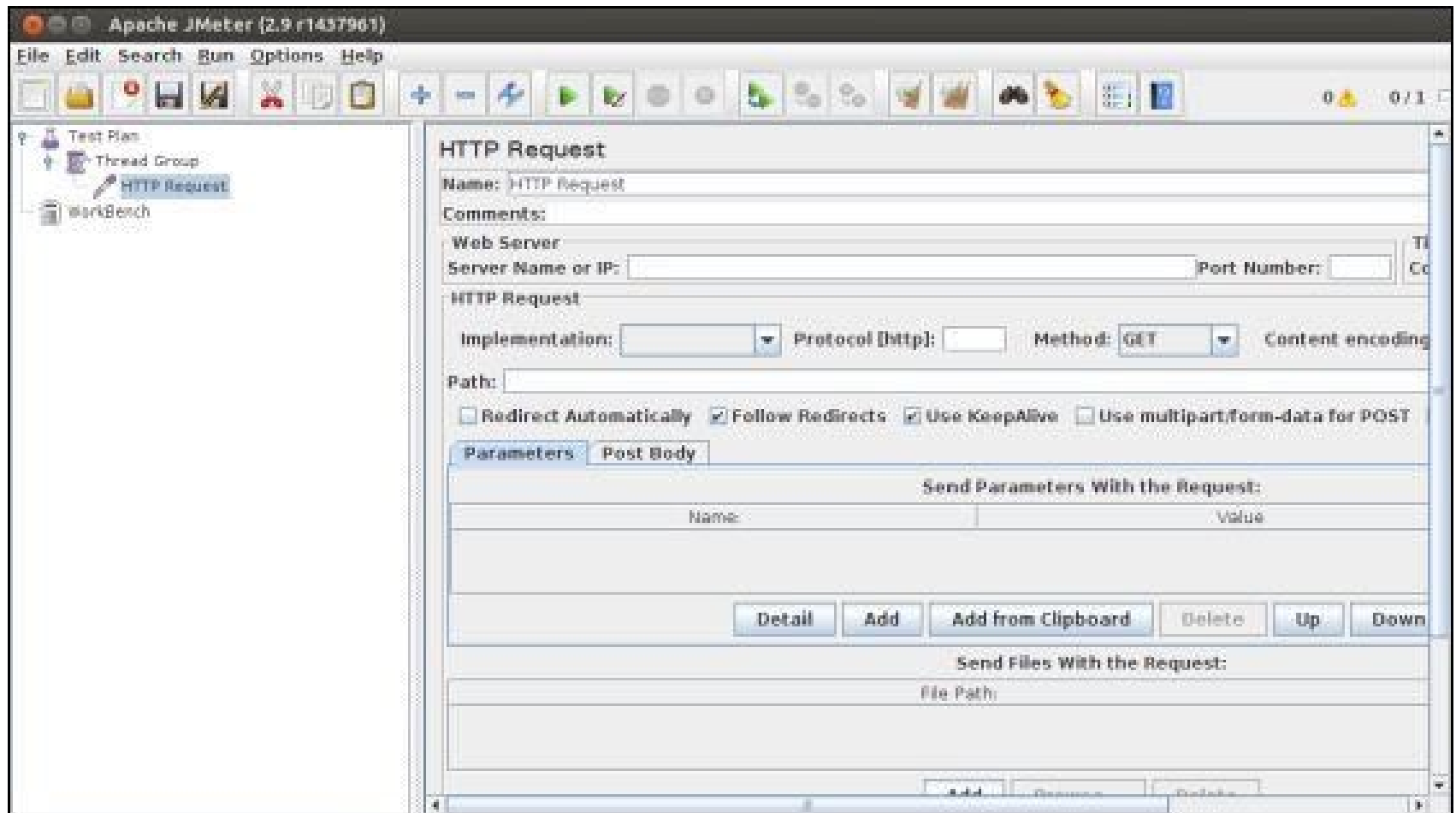
Add Jmeter Element

Menambahkan web server/IP Address yang akan dites. Caranya :

Klik Kanan Test plan

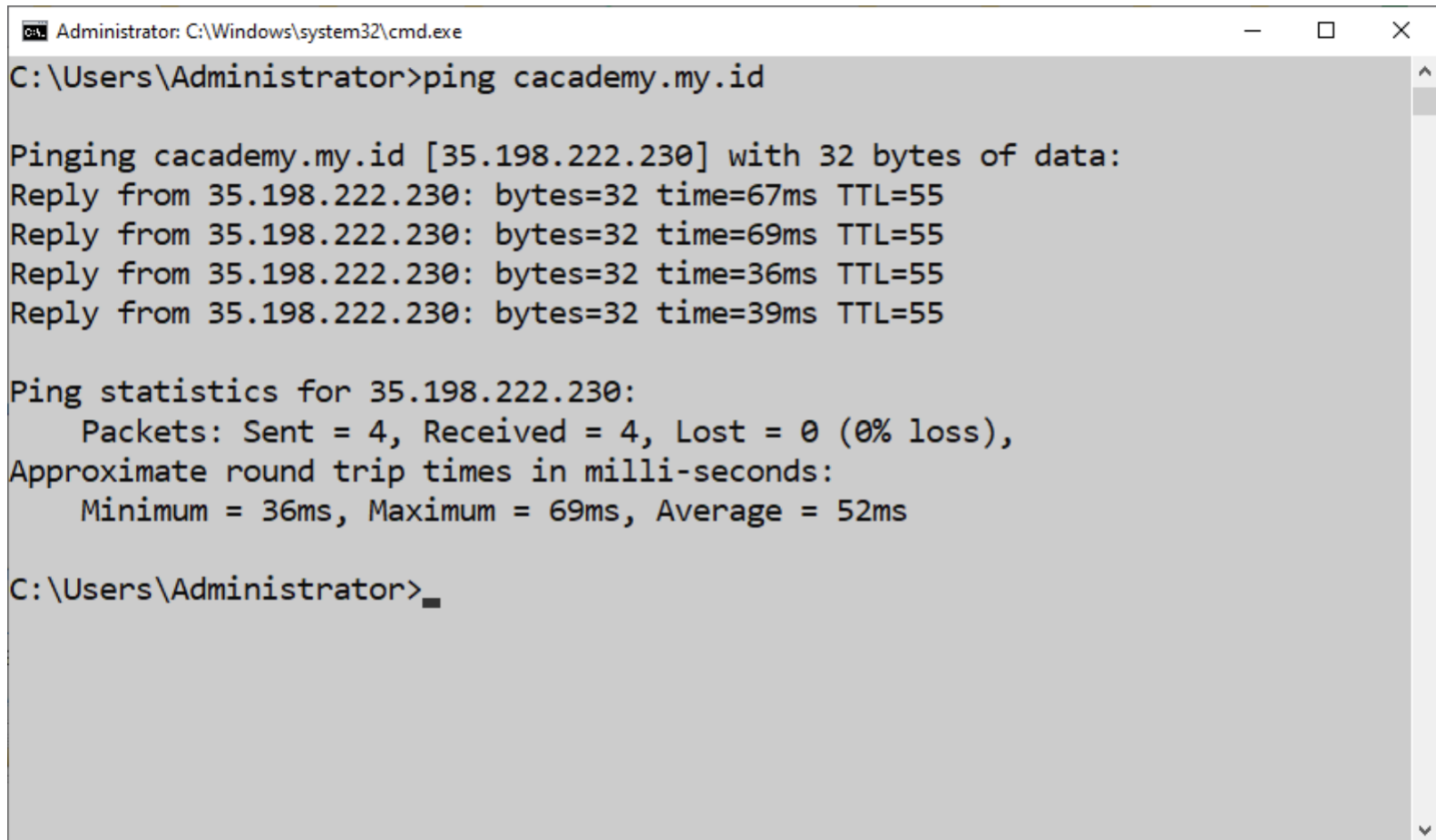
1. Klik Kanan Threads Group
2. Add > Sampler > HTTP Request
3. Isi web server, port dan path

MEMBUAT PERFORMANCE TEST PLAN



MEMBUAT PERFORMANCE TEST PLAN

Cara cek ip server / website



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\Administrator>ping cacademy.my.id

Pinging cacademy.my.id [35.198.222.230] with 32 bytes of data:
Reply from 35.198.222.230: bytes=32 time=67ms TTL=55
Reply from 35.198.222.230: bytes=32 time=69ms TTL=55
Reply from 35.198.222.230: bytes=32 time=36ms TTL=55
Reply from 35.198.222.230: bytes=32 time=39ms TTL=55

Ping statistics for 35.198.222.230:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 36ms, Maximum = 69ms, Average = 52ms

C:\Users\Administrator>
```

MEMBUAT PERFORMANCE TEST PLAN

3

Add Listener

Menampilkan proses dan hasil test secara grafis atau bentuk tabel. Caranya :

Klik Kanan Test Plan

Add > Listener > Graph Result

Add > Listener > View Results in Table

Add > Listener > Summary Report

MEMBUAT PERFORMANCE TEST PLAN

4

Run Test

Menjalankan Test secara otomatis. Caranya :
Simpan terlebih dahulu Test Plan yang telah kita buat di File > Save (Ctrl + S).

Klik Run atau Ctrl + R, jMeter akan mulai mensimulasi sejumlah user dalam mengakses web server yang telah ditentukan.

File Edit Search Run Options Tools Help

00:00:06 0 0/2

Contoh Test Plan 1

- Thread Group
 - HTTP Header Manager
 - HTTP Request
 - View Results Tree
 - Summary Report
- TEST 2

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

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

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	120	31	10	88	21.72	0.00%	19.8/sec	1302.82	7.86	67260.0
TOTAL	120	31	10	88	21.72	0.00%	19.8/sec	1302.82	7.86	67260.0

- **Label** : Merupakan nama dari *HTTP Request* yang dijalankan.

00:00:26 0 0/2

Contoh Test Plan 1

- Thread Group
 - Test-1
 - Test-2
- View Results Tree
- Summary Report
- TEST 2

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

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

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Test-1	120	17	8	282	25.70	0.00%	4.6/sec	283.77	1.83	62900.0
Test-2	120	323	234	713	121.78	100.00%	4.6/sec	3.47	0.77	774.7
TOTAL	240	170	8	713	176.34	50.00%	9.2/sec	284.63	2.58	31837.4

Label dari *Summary Report* tersebut ada Test-1 dan Test-2.

File Edit Search Run Options Tools Help

00:00:06 0 0/2

Contoh Test Plan 1

- Thread Group
 - HTTP Header Manager
 - HTTP Request
 - View Results Tree
 - Summary Report
- TEST 2

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	120	31	10	88	21.72	0.00%	19.8/sec	1302.82	7.86	67260.0
TOTAL	120	31	10	88	21.72	0.00%	19.8/sec	1302.82	7.86	67260.0

- ***Samples*** : Komponen yang mengindikasikan jumlah pengguna virtual (*virtual users*) per *request*.

Misalnya skenario yang hendak kita jalankan adalah *number of thread* = 2, *ramp-up period* = 10, dan *loop count* = 60, maka jumlah *samples* yang kita miliki adalah 120.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Test-1	120	17	8	282	25.70	0.00%	4.6/sec	283.77	1.83	62900.0
Test-2	120	323	234	713	121.78	100.00%	4.6/sec	3.47	0.77	774.7
TOTAL	240	170	8	713	176.34	50.00%	9.2/sec	284.63	2.58	31837.4

- ***Average*** : Mengindikasikan waktu rata-rata yang dihabiskan dalam mengeksekusi masing-masing label. Dalam kasus kita, *average time* dari label *Test-1* adalah 17 milisekon dan label *Test-2* adalah 323 milisekon, sehingga total *average* yang dihasilkan adalah $(17+323)/2 = 170$.

00:00:26 0 0/2

Contoh Test Plan 1

- Thread Group
 - Test-1
 - Test-2
- View Results Tree
- Summary Report
- TEST 2

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Test-1	120	17	8	282	25.70	0.00%	4.6/sec	283.77	1.83	62900.0
Test-2	120	323	234	713	121.78	100.00%	4.6/sec	3.47	0.77	774.7
TOTAL	240	170	8	713	176.34	50.00%	9.2/sec	284.63	2.58	31837.4

- **Average** : Mengindikasikan waktu rata-rata yang dihabiskan dalam mengeksekusi masing-masing label. Dalam kasus kita, *average time* dari label *Test-1* adalah 17 milisekon dan label *Test-2* adalah 323 milisekon, sehingga total *average* yang dihasilkan adalah $(17+323)/2 = 170$.
- **Min** : Mengindikasikan waktu tersingkat yang dibutuhkan dalam mengeksekusi masing-masing label.
- **Max** : Mengindikasikan waktu terpanjang yang dibutuhkan dalam mengeksekusi masing-masing label.

00:00:26 0 0/2

Contoh Test Plan 1

- Thread Group
 - Test-1
 - Test-2
 - View Results Tree
 - Summary Report
 - TEST 2

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Test-1	120	17	8	282	25.70	0.00%	4.6/sec	283.77	1.83	62900.0
Test-2	120	323	234	713	121.78	100.00%	4.6/sec	3.47	0.77	774.7
TOTAL	240	170	8	713	176.34	50.00%	9.2/sec	284.63	2.58	31837.4

- ***Std. Dev*** : Menunjukkan penyebaran kumpulan data relatif terhadap rata-ratanya. Semakin kecil nilai dari *std.dev* menunjukkan bahwa data yang dijalankan pada masing-masing label semakin konsisten. Nilai dari *std.dev* sebaiknya lebih kecil atau sama dengan setengah dari nilai *average* dari setiap label.
- ***Error %*** : Menunjukkan jumlah *error* dalam satuan persen yang terjadi pada setiap label.
- ***Throughput*** : Menunjukkan jumlah *request* yang berhasil diproses per time unit (*second, minute, hours*) oleh server. semakin besar nilai *throughput*, semakin bagus.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Test-1	120	17	8	282	25.70	0.00%	4.6/sec	283.77	1.83	62900.0
Test-2	120	323	234	713	121.78	100.00%	4.6/sec	3.47	0.77	774.7
TOTAL	240	170	8	713	176.34	50.00%	9.2/sec	284.63	2.58	31837.4

- ***Received KB/sec*** : Mengindikasikan jumlah data yang berhasil diunduh oleh *server* selama dilakukannya eksekusi pengujian *performance* dalam satuan *kilobyte* tiap 1 sekon.
- ***Sent KB/sec*** : Mengindikasikan jumlah data yang berhasil dikirim dari *server* selama dilakukannya eksekusi pengujian *performance* dalam satuan *kilobyte* tiap 1 sekon.
- ***Avg Bytes*** : Merupakan rata-rata *byte* yang berhasil diunduh (*download*) oleh *server*.

let's do an experiment

- *Number of threads : 5000*
- *Ramp-up period : 1*
- *Loop-count : 1*

Hasilnya

- *Average : 738*
- *Min : 155*
- *Max : 2228*
- *Throughput : 60.5*

- semua *thread* yang dijalankan dalam waktu 1 detik (*ramp-up*), dibutuhkan waktu rata-rata eksekusi 738 *milliseconds* atau
- 0.7 *seconds* dengan waktu minimum yang dibutuhkan untuk mengeksekusi *thread* yaitu 155 *milliseconds* dan waktu maksimumnya 2228 *milliseconds*.
- Dari *throughput* , dapat melihat *server* mampu mengeksekusi 60.5 *thread* per time unit.

Task

- Hosting Web yang telah di bangun
- Setiap mahasiswa dalam kelompok membangun 1 Test case (pilih usecase berbeda tiap mhs) > ikuti 3 Langkah testcase
- Uji menggunakan Jmeter dengan Skenario:
 - Skenario 1 (Mhs A)
 - Number of Tread : 100
 - Rump-up period : 1
 - Loop count = 1
 - Skenario 2 (Mhs B)
 - Number of Tread : 1000
 - Rump-up period : 1
 - Loop count = 1
 - Skenario 3 (Mhs C)
 - Number of Tread : 1000
 - Rump-up period : 10
 - Loop count = 1
 - Skenario 4 (Mhs D)
 - Number of Tread : 1000
 - Rump-up period : 10
 - Loop count = 10
- Setiap mhs Lakukan langkahnya mulai pada slide 21 – 35 (Listener : Grafik, Tabel, dan summary). Analisa hasil jangan lupa dijabarkan jangan hanya di SS saja.