

# Day 2 - Introduction to Apache JMeter

## Introduction to Apache Jmeter



Apache JMeter adalah software untuk melakukan pengujian non-fungsional aplikasi web. Jenis pengujian yang didukung yaitu performance test, load test dan stress test.

JMeter adalah aplikasi Java bersifat open-source. Pada awalnya JMeter digunakan untuk menguji aplikasi web serta FTP, namun saat ini sudah memiliki kemampuan untuk menguji database, server, dan fasilitas lain. Selain itu, JMeter juga cukup mudah untuk di-install di beberapa sistem operasi. JMeter memerlukan JDK versi 8 atau versi yang lebih tinggi.

Namun, salah satu kelemahan terbesar JMeter adalah tool ini tidak support bahasa pemrograman JavaScript.

### Features of JMeter

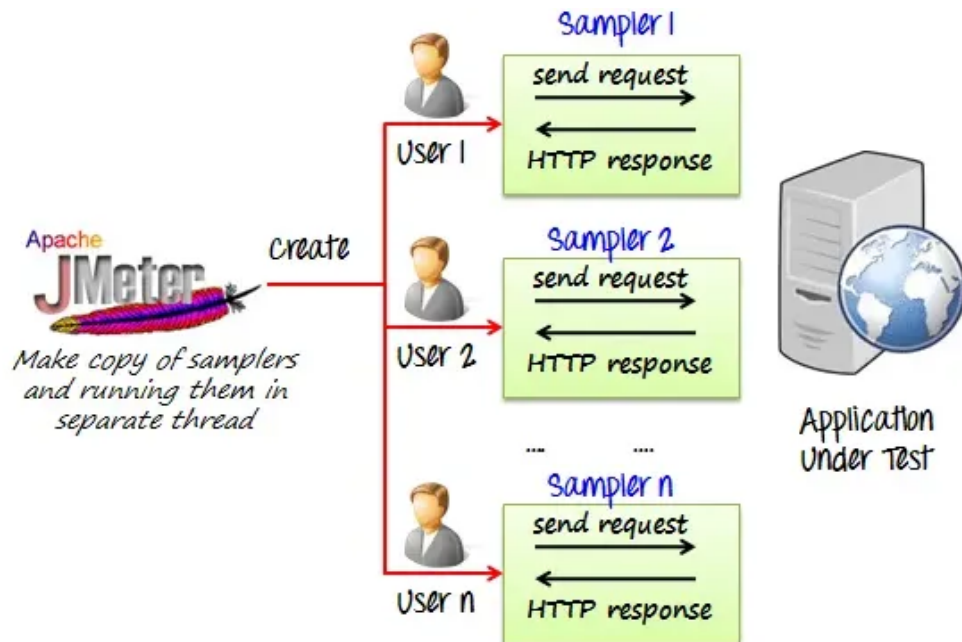
Fitur utama dari Apache JMeter meliputi:

- Memiliki GUI yang sederhana dan mudah dipahami.
- Memiliki kemampuan untuk membuat dan menjalankan skenario pengujian yang kompleks.
- Dukungan untuk berbagai protokol seperti HTTP, HTTPS, FTP, JDBC, SOAP, dan banyak lagi.
- Grafik dan laporan yang informatif untuk menganalisis hasil pengujian.
- Kemampuan untuk melakukan pengujian terdistribusi dengan menggunakan beberapa mesin pengujian.

Apache JMeter dapat dijalankan dalam mode GUI (Graphical User Interface) atau mode non-GUI (command-line interface), memungkinkan user untuk melakukan pengujian baik secara interaktif maupun otomatis.

## How does the JMeter Test Work?

JMeter adalah tools yang menguji fungsionalitas dan kinerja server target dengan mengirimkan request ke server target dan mensimulasikan sekelompok users (threads).



[https://miro.medium.com/v2/resize:fit:720/format:webp/1\\*ESISxoC5cwzb14fvM0a9sw.png](https://miro.medium.com/v2/resize:fit:720/format:webp/1*ESISxoC5cwzb14fvM0a9sw.png)

Ada tiga langkah-langkah berbeda yang dilakukan oleh JMeter selama pengujian sebagai berikut :

1. Pertama, Jmeter akan membuat request dan mengirimkannya ke server.
2. Setelah itu, Jmeter akan memproses respon dari server.
3. Terakhir, Jmeter menghasilkan test result dalam berbagai format seperti TXT, XML, JSON. Jadi QA/Engineer dapat menganalisis data.

## Advantages of Jmeter

Beberapa kelebihan JMeter adalah:

- **Open Source License:** JMeter sepenuhnya gratis, memungkinkan developer menggunakan source code untuk pengembangan, memodifikasi dan menyesuaikannya sesuai dengan kebutuhan.
- **Platform Independent:** JMeter adalah 100% aplikasi desktop Java murni. Jadi bisa berjalan di berbagai platform. Misalnya Windows, Linux, Mac dll.

## Setting Up Jmeter

Untuk menginstall Apache JMeter, memerlukan JDK (Java Development Kit) yang sesuai karena JMeter ditulis dalam bahasa Java. JDK yang diperlukan adalah JDK versi 8 atau yang lebih baru.

### System Requirement

<b>JDK</b>	1.6 or above.
<b>Memory</b>	No minimum requirement.
<b>Disk Space</b>	No minimum requirement.
<b>Operating System</b>	No minimum requirement.

### Step 1: Verify Java Installation

Langkah pertama, verifikasi apakah Java telah terinstal di perangkat kita. Buka command shell dan jalankan salah satu perintah **java** berikut berdasarkan sistem operasi yang digunakan.

<b>OS</b>	<b>Task</b>	<b>Command</b>
Windows	Open Command Console	c:\> java -version
Linux	Open Command Terminal	\$ java -version
Mac	Open Terminal	machine: ~ joseph\$ java -version

Jika Java telah terinstal di perangkat kita, maka sistem akan menampilkan output berikut sesuai dengan OS masing-masing.

<b>OS</b>	<b>Output</b>
Windows	java version "1.7.0_25" Java(TM) SE Runtime Environment (build 1.7.0_25-b15) Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)
Linux	java version "1.7.0_25" Java(TM) SE Runtime Environment (build 1.7.0_25-b15) Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)
Mac	java version "1.7.0_25" Java(TM) SE Runtime Environment (build 1.7.0_25-b15) Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)

Apabila perangkat kita belum memiliki Java, instal Java Software Development Kit (SDK) melalui link [di sini](#).

## Step 2: Set Java Environment

Set JAVA\_HOME environment variable untuk mengarah ke lokasi di mana Java telah diinstal.

OS	Output
Windows	Set the environment variable JAVA_HOME to C:\Program Files\Java\jdk1.7.0_25
Linux	export JAVA_HOME=/usr/local/java-current
Mac	export JAVA_HOME=/Library/Java/Home

Tambahkan lokasi Java compiler ke System Path.

OS	Output
Windows	Append the string; C:\Program Files\Java\jdk1.7.0_25\bin to the end of the system variable, Path.
Linux	export PATH=\$PATH:\$JAVA_HOME/bin/
Mac	not required

Verifikasi Instalasi Java menggunakan perintah **java -version** seperti dijelaskan di atas.

## Step 3: Download Jmeter

Berikutnya yang harus dilakukan adalah meng-install JMeter. Download latest aplikasi Apache Jmeter [di sini](#). Setelah berhasil download, ekstrak folder JMeter.

## Step 4 : Menjalankan JMeter

Ada beberapa opsi dalam menjalankan Jmeter sebagai berikut:

### Opsi 1 ~ klik 2x

Masuk ke *folder JMeter* hasil dari *extract* di atas, kemudian masuk ke folder **bin**, lalu klik 2x file **jmeter.bat**

### Opsi 2 ~ command line

1. Buka *terminal/command prompt*.
2. Dari *terminal/command prompt*:
  - o Masuk ke direktori *JMeter* hasil dari *extract* di atas.
  - o Masuk ke *folder bin*.
3. Ketik **./jmeter**
4. Tekan Enter.

### Opsi 3 ~ environment variable

1. Buat *environment variable* dengan *value directory path folder bin* yang ada di dalam *folder JMeter* hasil *extract*.

Misal: `/Users/cahklaten/Documents/jmeter5.4/bin`

#### **Windows:**

Bisa menambahkan ke variable **PATH**

#### **MacOS:**

Bisa menambahkan ke **.bash\_profile**

Misal:

```
export PATH=$PATH:"/Users/cahklaten/Documents/jmeter5.4/bin"
```

#### **Linux:**

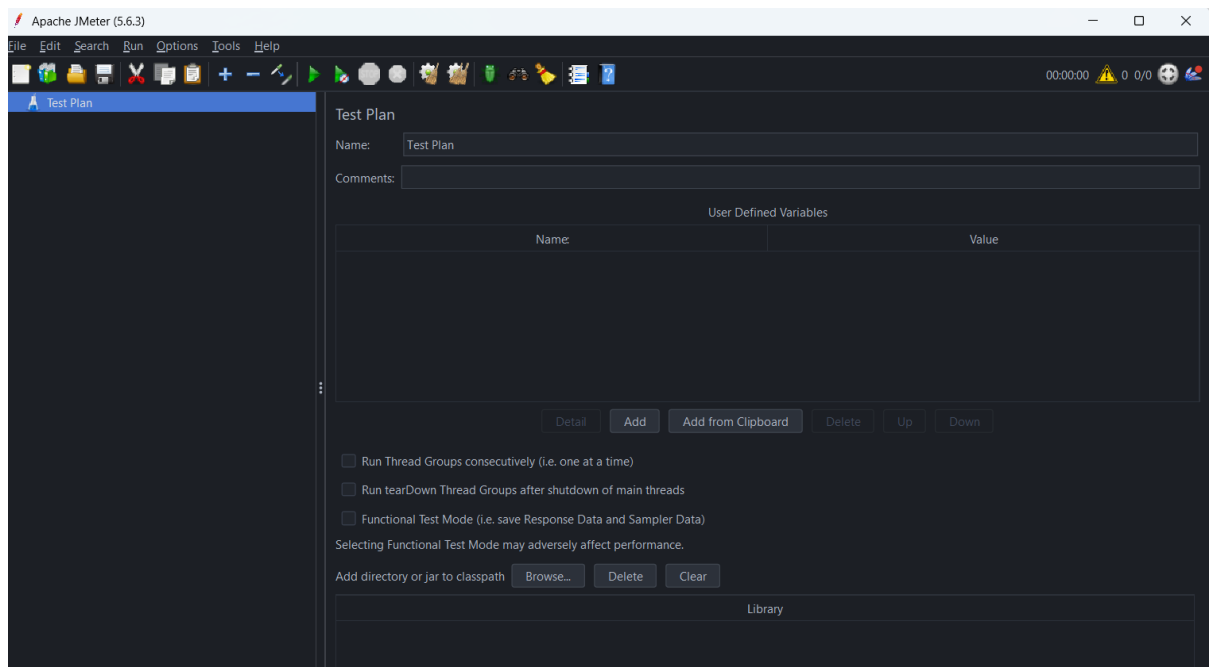
Bisa menambahkan ke **.bashrc**

Misal

```
export PATH=$PATH:"/Users/cahklaten/Documents/jmeter5.4/bin"
```

2. Buka *terminal/command prompt*.
3. Ketik `jmeter`.

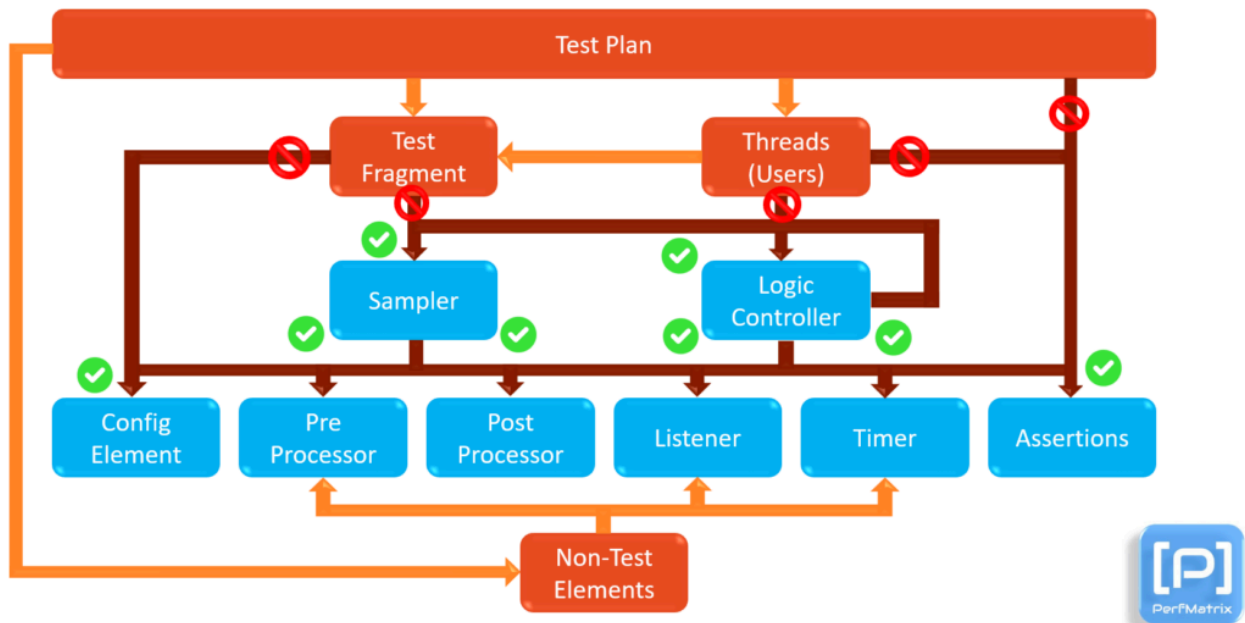
Bagi pengguna Linux, jalankan **`$JMETER_FOLDER/bin/jmeter`** pada terminal. Akan muncul tampilan aplikasi JMeter seperti pada gambar berikut.



# Jmeter Elements

## Apa itu Elemen di JMeter ?

Berbagai komponen JMeter disebut Elemen. Setiap Elemen dirancang untuk tujuan tertentu. Gambar di bawah memberikan beberapa elemen umum di JMeter.



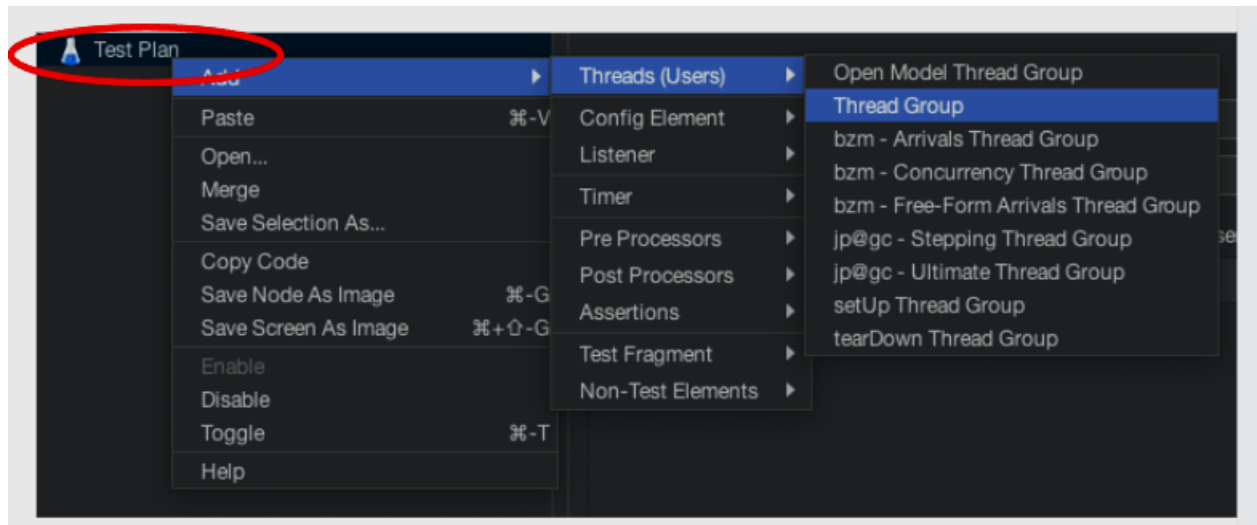
<https://www.perfmatrix.com/wp-content/uploads/2020/04/JMeter-Elements-Relationship-2-1024x530.png>

Di sini kita akan membahas Elemen yang harus diketahui sebelum dapat memulai Pengujian di JMeter.

- Elemen yang dibahas dalam materi ini adalah :
- Test Plan
- Thread Group
- Samplers
  - FTP request
  - HTTP request
  - JDBC request
  - Logic Controller
- Listeners
- Configuration Element
  - CSV Data Set Config
  - HTTP Cookie Manager
  - HTTP Request Default

## 1. Test Plan

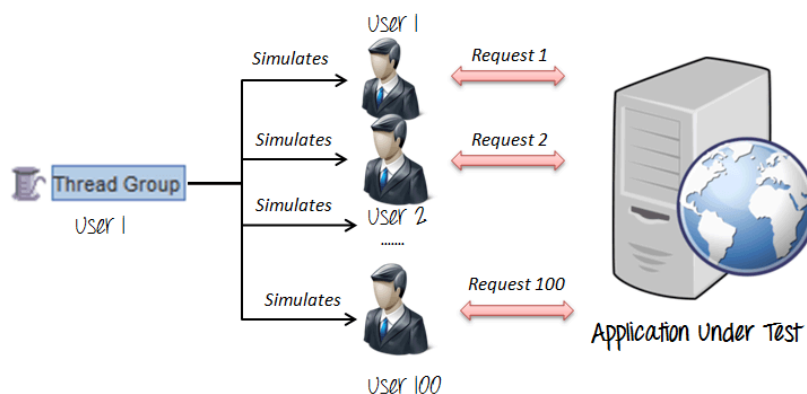
Test Plan merupakan kerangka utama dari seluruh pengujian, termasuk pengaturan global dan pengelompokan elemen-elemen tes.



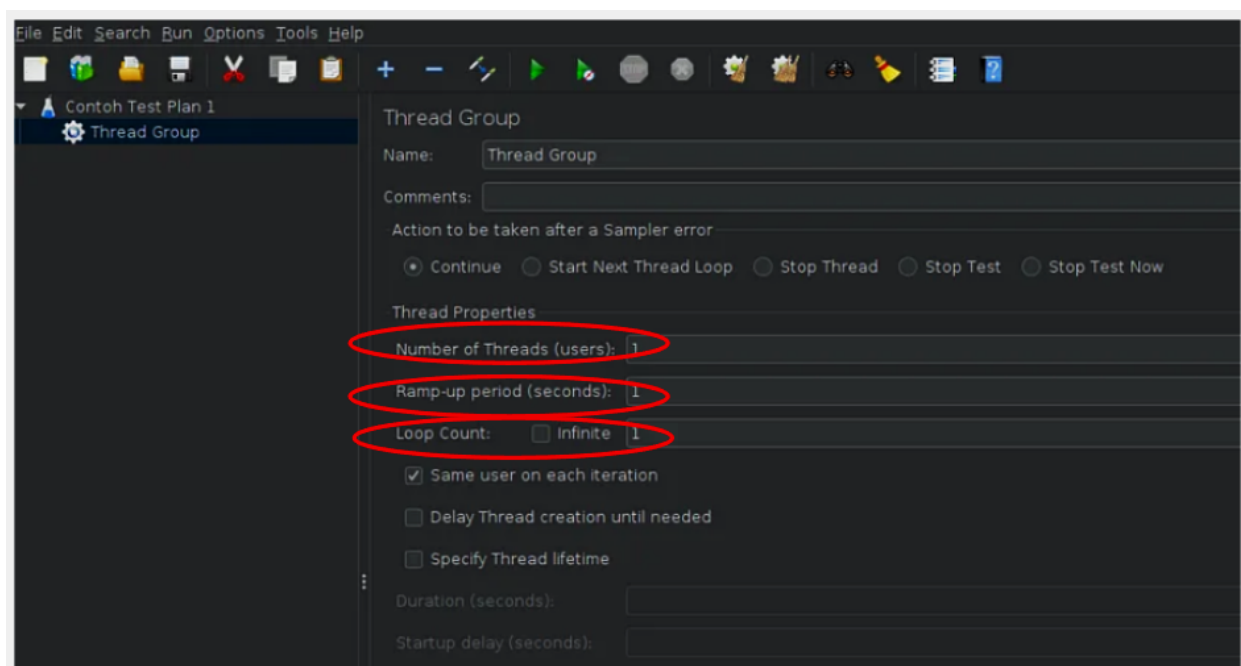
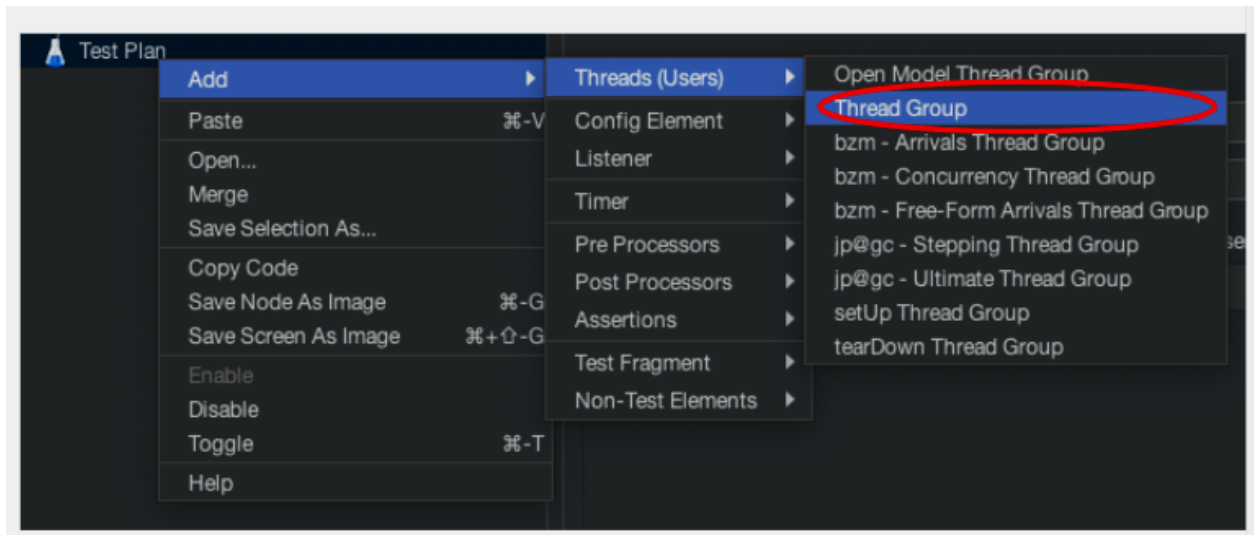
## 2. Thread Group

Thread Groups adalah kumpulan Thread. Setiap thread mewakili satu user yang menggunakan aplikasi yang sedang diuji. Pada dasarnya, setiap Thread mensimulasikan satu request real user ke server.

Kontrol untuk grup thread memungkinkan kita mengatur jumlah thread untuk setiap grup.



Misalnya, jika kita menetapkan jumlah user sebanyak 100; maka JMeter akan membuat dan mensimulasikan jumlah tsb dengan virtual user dengan 100 dari user ke server yang diuji.

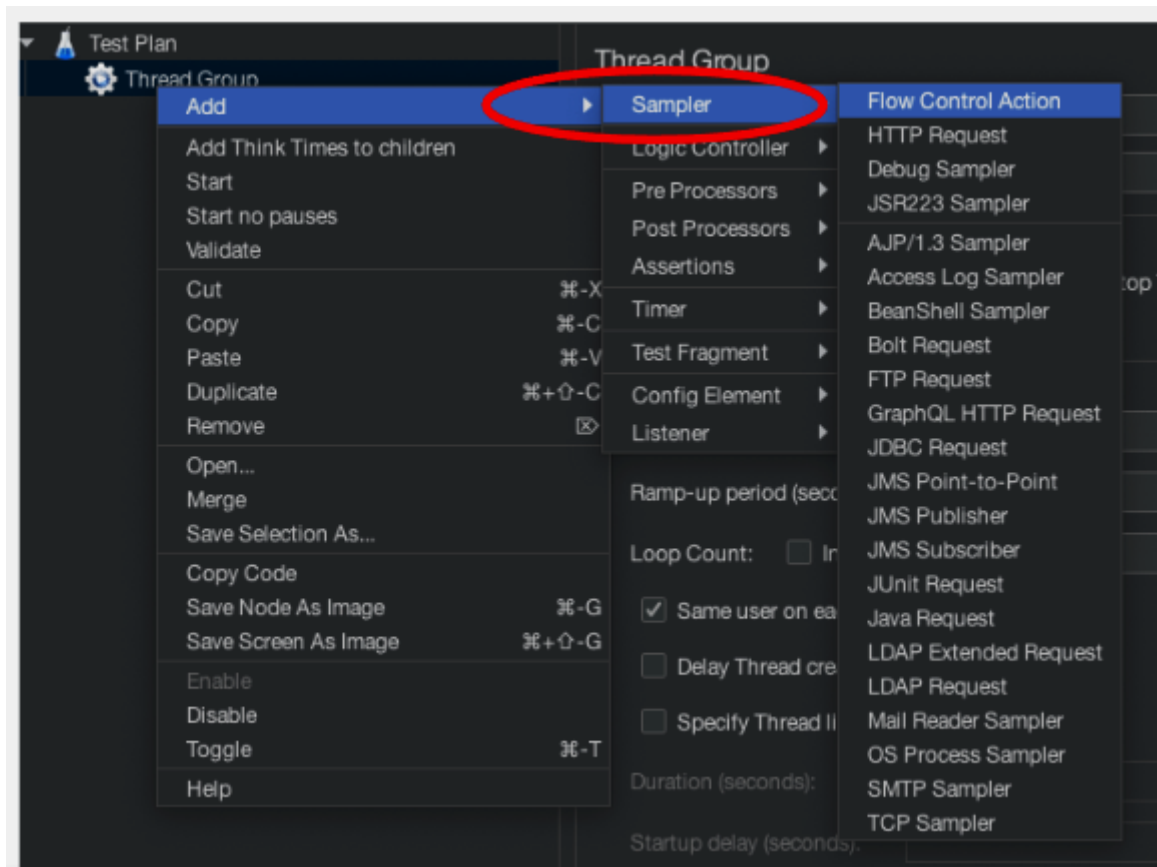


### 3. Sampler

Seperti yang kita ketahui bahwa JMeter mendukung pengujian HTTP, FTP, JDBC dan banyak protokol lainnya. Kita sudah tahu bahwa Grup Thread mensimulasikan request user ke server.

Namun bagaimana Grup Thread mengetahui jenis request (HTTP, FTP, dll.) yang perlu dibuat? Jawabannya adalah Sampler.

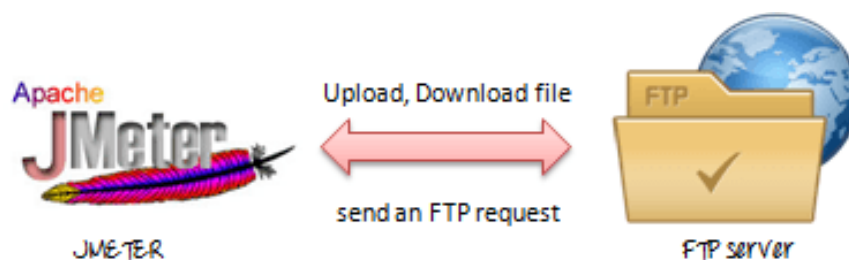




Request user dapat berupa Request FTP, Request HTTP, Request JDBC...DII.

### 3.1. FTP request

Bayangkan kita ingin menguji kinerja server FTP. kita dapat menggunakan sampler request FTP di JMeter untuk melakukan tugas ini. Pengontrol ini memungkinkan kita mengirim request “unduh file” atau “unggah file” FTP ke server FTP.



Misalnya, jika kita ingin mendownload file “Test.txt” dari server FTP yang sedang diuji, kita perlu mengkonfigurasi beberapa parameter di JMeter seperti gambar di bawah ini

**FTP Request**

Name:

Comments:

Server Name or IP:  Port Number:

Remote File:

Local File:

Local File Contents:

☒ get(RETR) ☐ put(STOR) ☐ Use Binary mode ? ☐ Save File in Response ?

Login Configuration

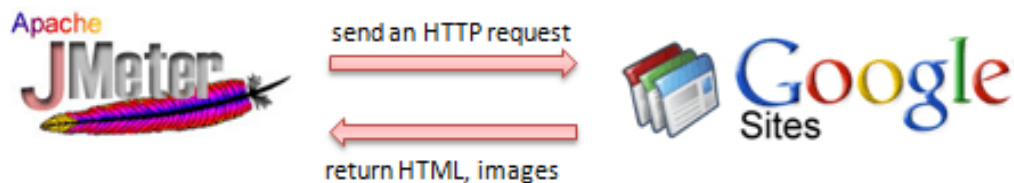
Username:  Password:

JMeter akan mengirimkan perintah FTP ke server FTP ftp.example.com, dan kemudian mendownload file Test.txt dari server tersebut.

### 3.2. HTTP request

Sampler ini memungkinkan kita mengirim HTTP/HTTPS request ke server web.

Perhatikan contoh berikut. JMeter mengirimkan HTTP request ke situs web Google dan mengambil file HTML atau gambar dari situs web ini.



Contoh Test Plan 1

- Thread Group
  - HTTP Header Manager
  - HTTP Request

**HTTP Request**

Name:

Comments:

Basic Advanced

Web Server

Protocol [http]:  Server Name or IP:  Port Number:

HTTP Request

GET Path:  Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

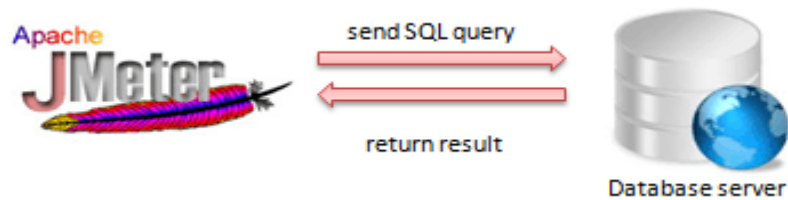
Parameters Body Data Files Upload

Send Parameters With the Request:

Name	Value	URL Encode?	Content-Type	Include Equals?
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>

### 3.3. JDBC request

Sampler ini memungkinkan kita menjalankan Pengujian Kinerja Basis Data. Ia mengirimkan JDBC Request (SQL query) ke database.



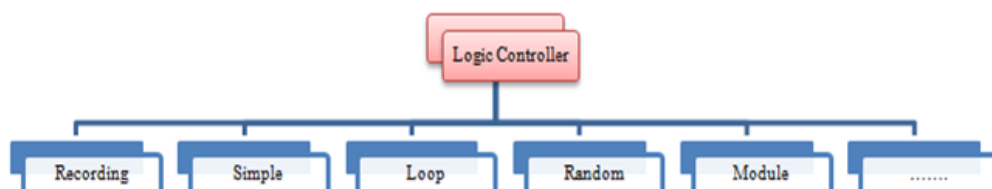
Misalnya, server database memiliki bidang test result yang disimpan dalam nama tabel **test\_tbl**. kita ingin menanyakan data ini dari server database; kita dapat mengkonfigurasi JMeter untuk mengirim SQL query ke server ini untuk mengambil data.

The screenshot shows the 'JDBC Request' configuration window in JMeter. The 'Name' field is set to 'JDBC Request'. The 'SQL Query' section is expanded, showing 'Query Type' set to 'Select Statement' and the 'Query' text area containing the SQL statement: `select test_result from test_tbl where id = 1`. This text area is highlighted with a red rectangle and labeled 'sql query' in red handwritten text. Below the query text area are fields for 'Parameter values:', 'Parameter types:', 'Variable names:', and 'Result variable name:'.

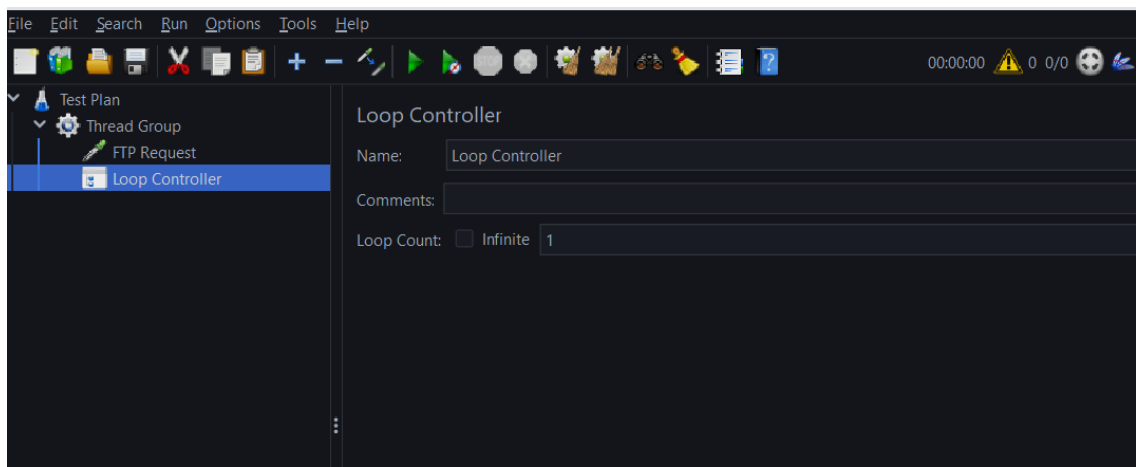
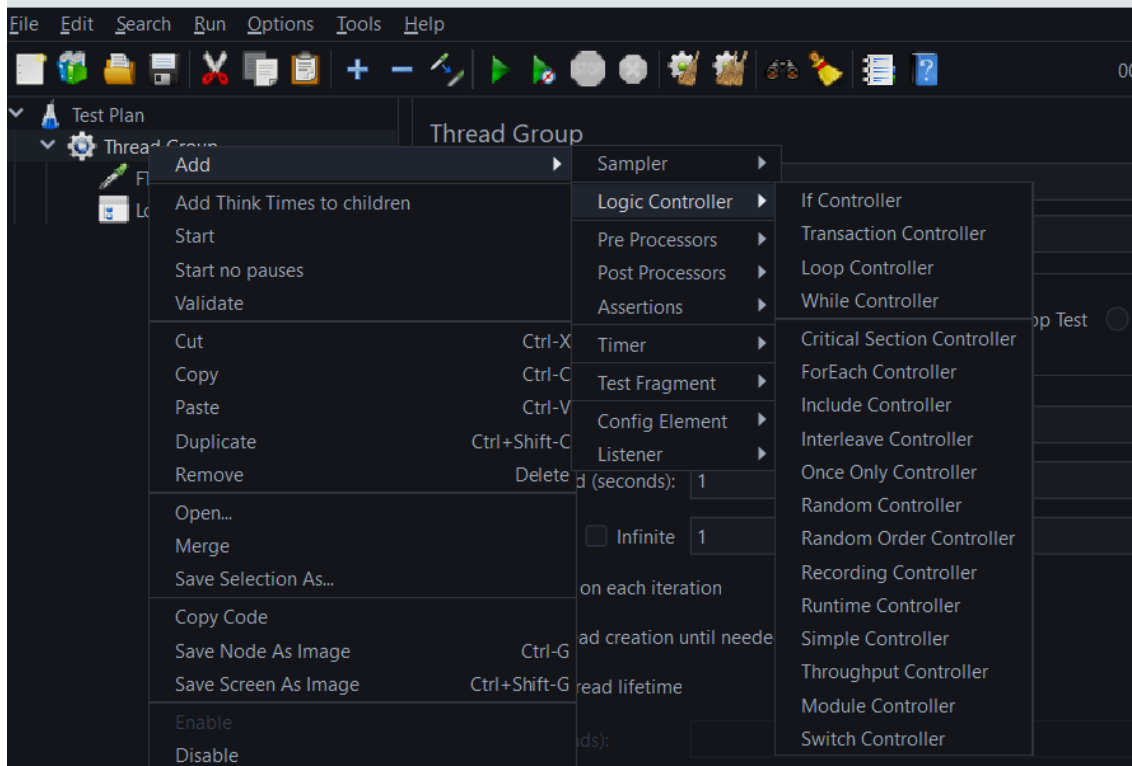
### 3.4. Logic Controller

Logic Controller memungkinkan kita menentukan urutan pemrosesan request dalam sebuah Thread. Ini memungkinkan kita mengontrol “kapan” untuk mengirim user request ke server web. Misalnya, kita dapat menggunakan *Random Controller* untuk mengirim HTTP request ke server secara acak.

Logic Controller menentukan urutan di mana user request dieksekusi. Beberapa Logic Controller yang umum digunakan adalah di bawah ini:



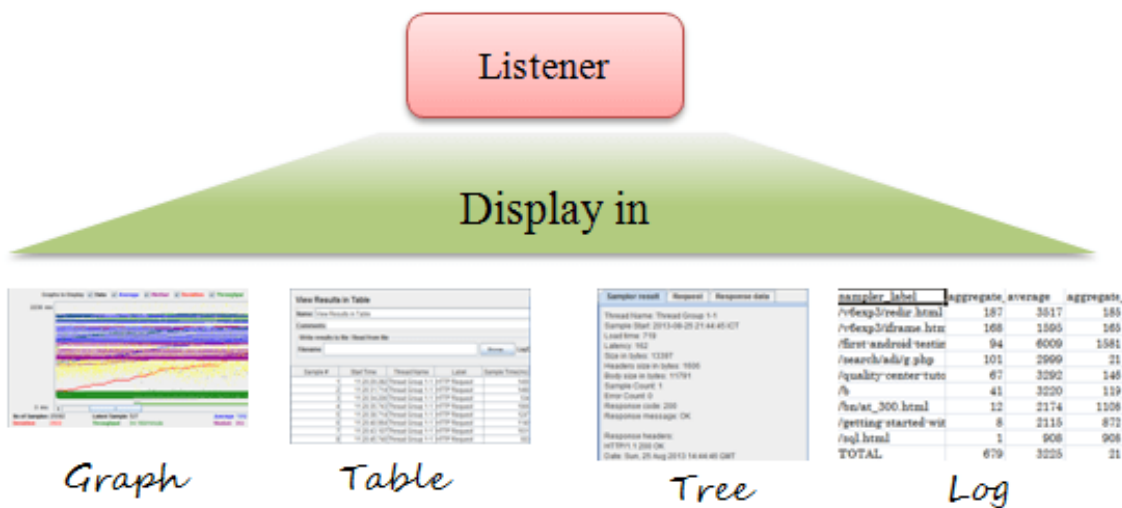
Berikut adalah screenshot dari Loop Controller Control Panel.



## 4. Listeners

Listener memungkinkan kita melihat hasil Sampler dalam bentuk tabel, grafik, pohon, atau teks sederhana di berbagai file log. Listener menyediakan akses visual ke data yang dikumpulkan oleh JMeter terkait test case saat komponen Sampler JMeter dijalankan.

Listeners dapat ditambahkan di mana saja dalam pengujian, termasuk langsung di bawah test plan. Listener akan mengumpulkan data hanya dari elemen-elemen yang berada pada tingkat yang sama atau di bawahnya.



- **Graph Results**



Pemroses grafik result menampilkan waktu respons server pada Grafik

- **Tree Result**

Text

Sampler result Request Response data

GET Books by ID 2-25

Thread Name:Books Load Testing 1-25

Sample Start:2025-01-30 14:26:05 WIB

Load time:1655

Connect Time:0

Latency:1655

Size in bytes:665

Sent bytes:145

Headers size in bytes:184

Body size in bytes:481

Sample Count:1

Error Count:0

Data type ("text"|"bin"|""):text

Response code:200

Response message:OK

HTTPSampleResult fields:

ContentType: application/json; charset=utf-8; v=1.0

DataEncoding: utf-8

Hasil menunjukkan hasil user request dalam format HTML dasar

- **Table Result**

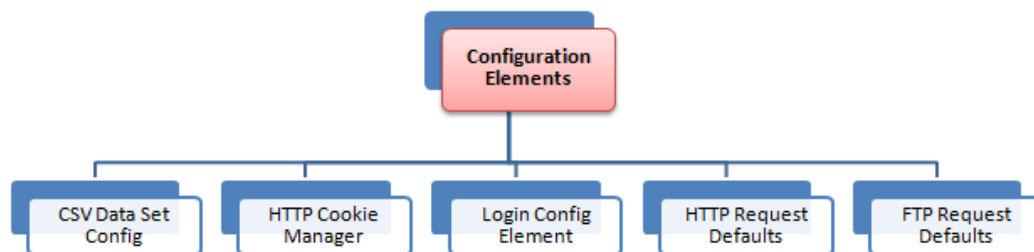
Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	14:25:13.523	Books Load ...	GET Books 1...	1841	✓	95248	143	1610	883
2	14:25:13.753	Books Load ...	GET Books 1...	3600	✓	95250	143	1929	1245
3	14:25:13.951	Books Load ...	GET Books 1...	3779	✓	95222	143	2321	1075
4	14:25:13.814	Books Load ...	GET Books 1...	3954	✓	95239	143	2465	1068
5	14:25:13.734	Books Load ...	GET Books 1...	4120	✓	95214	143	1981	1238
6	14:25:15.366	Books Load ...	GET Books b...	2557	✓	665	145	2557	0
7	14:25:14.162	Books Load ...	GET Books 1...	4215	✓	95233	143	2054	898
8	14:25:13.997	Books Load ...	GET Books 1...	4392	✓	95232	143	2238	982
9	14:25:14.111	Books Load ...	GET Books 1...	4283	✓	95238	143	2173	885
10	14:25:14.020	Books Load ...	GET Books 1...	4400	✓	95236	143	1811	1083
11	14:25:14.011	Books Load ...	GET Books 1...	4566	✓	95225	143	1678	1045
12	14:25:18.394	Books Load ...	GET Books b...	363	✓	665	145	363	0
13	14:25:17.731	Books Load ...	GET Books b...	1027	✓	665	145	1027	0
14	14:25:14.092	Books Load ...	GET Books 1...	4669	✓	95223	143	1622	1004
15	14:25:14.144	Books Load ...	GET Books 1...	4635	✓	95254	143	2188	950
16	14:25:17.855	Books Load ...	GET Books b...	927	✓	665	145	927	0
17	14:25:18.378	Books Load ...	GET Books b...	453	✓	665	145	453	0

Table result menunjukkan ringkasan hasil tes dalam format tabel.

## 5. Configuration Elements

Configuration Elements memungkinkan kita untuk menetapkan default dan variabel yang akan digunakan oleh Sampler. Elemen-elemen ini digunakan untuk menambah atau mengubah request yang dibuat oleh Sampler.

Gambar di bawah menunjukkan beberapa elemen konfigurasi yang umum digunakan di JMeter

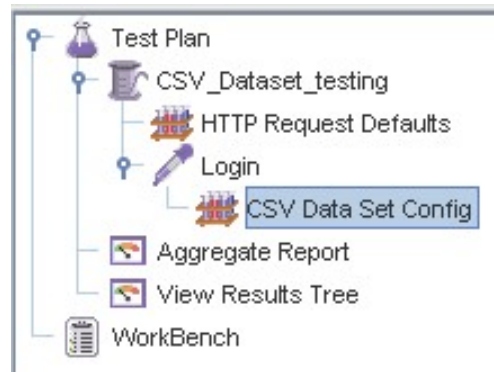


### 5.1. CSV Data Set Config

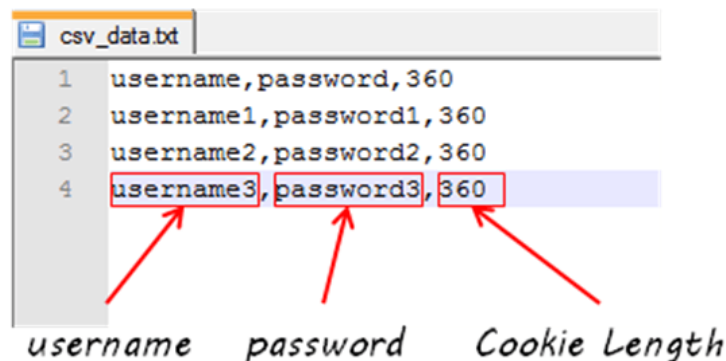
Misalkan kita ingin menguji situs web untuk 100 users yang masuk dengan credential berbeda.

Kita tidak perlu merekam script 100 kali! Kita dapat membuat parameter script untuk memasukkan credential login yang berbeda. Informasi login ini (misalnya username, password) dapat disimpan dalam file teks.

JMeter memiliki elemen yang memungkinkan kita membaca parameter berbeda dari file teks tersebut. Ini adalah “**CSV Data Set Config**”, yang digunakan untuk membaca baris dari file, dan membaginya menjadi variabel.



Ini adalah contoh Data CSV. Ini adalah file teks yang berisi username dan password untuk masuk ke situs web target kita inginkan.



## 5.2. HTTP Cookie Manager

Pahami ini dengan contoh berikut :

1. Kita menggunakan browser kita (Firefox, Chrome IE...DII) untuk menjelajahi [www.google.com](http://www.google.com)
2. Kita masuk dengan username dan password.
3. Username dan password kita akan disimpan di komputer sebagai cookie.

Lain kali, ketika kita mengunjungi [www.google.com](http://www.google.com), kita tidak perlu melakukan login lagi karena browser kita akan menggunakan cookies kita sebagai data user untuk login.

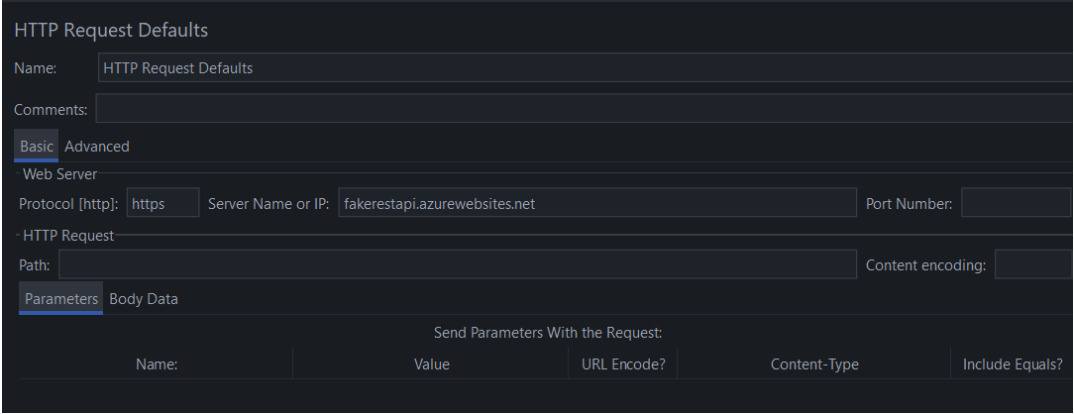
HTTP Cookie Manager juga memiliki fitur yang sama dengan web browser. Jika kita memiliki HTTP request dan responnya berisi cookie, Cookie Manager secara otomatis menyimpan cookie tersebut dan akan menggunakannya untuk semua requests di masa mendatang ke situs web tersebut.

## 5.3. HTTP Request Default

Elemen ini memungkinkan kita menetapkan nilai default yang digunakan pengontrol HTTP Request kita.

Misalnya, kita mengirimkan 100 HTTP request ke server google.com dan harus memasukkan nama server = google.com secara manual untuk semua 100 requests ini.

Sebagai gantinya, kita dapat menambahkan satu HTTP request default dengan bidang “Nama Server atau IP” = google.com serta tidak perlu mengetik 100 kali!



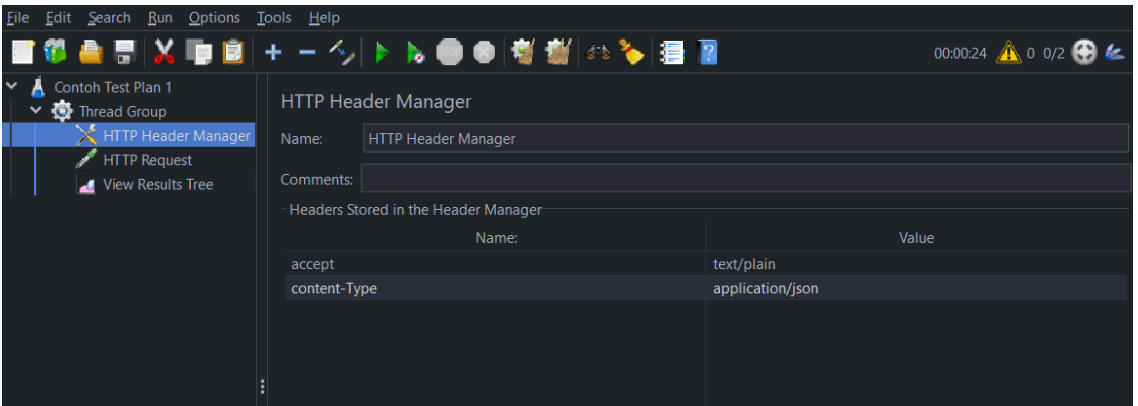
The screenshot shows the 'HTTP Request Defaults' configuration window. The 'Name' field is set to 'HTTP Request Defaults'. The 'Basic' tab is selected. Under 'Web Server', the 'Protocol [http]' is set to 'https', 'Server Name or IP' is set to 'fakereastapi.azurewebsites.net', and 'Port Number' is empty. Under 'HTTP Request', the 'Path' is empty and 'Content encoding' is set to 'UTF-8'. The 'Parameters' tab is also visible, showing a table for 'Send Parameters With the Request'.

Name	Value	URL Encode?	Content-Type	Include Equals?
------	-------	-------------	--------------	-----------------

## 5.4. HTTP Header Manager

HTTP Header Manager adalah elemen yang dimana kita bisa mengatur informasi pada HTTP Request Header yang akan dikirimkan oleh JMeter.

Di HTTP Header Manager kita dapat mengatur seperti **Content-Type**, **Authorization**, dan header lain yang digunakan dalam sebuah endpoint atau HTTP Request. Hal ini mirip dengan pada saat kita mengatur Header di sebuah request pada Postman.



The screenshot shows the 'HTTP Header Manager' configuration window. The 'Name' field is set to 'HTTP Header Manager'. The 'Comments' field is empty. Below, there is a table titled 'Headers Stored in the Header Manager'.

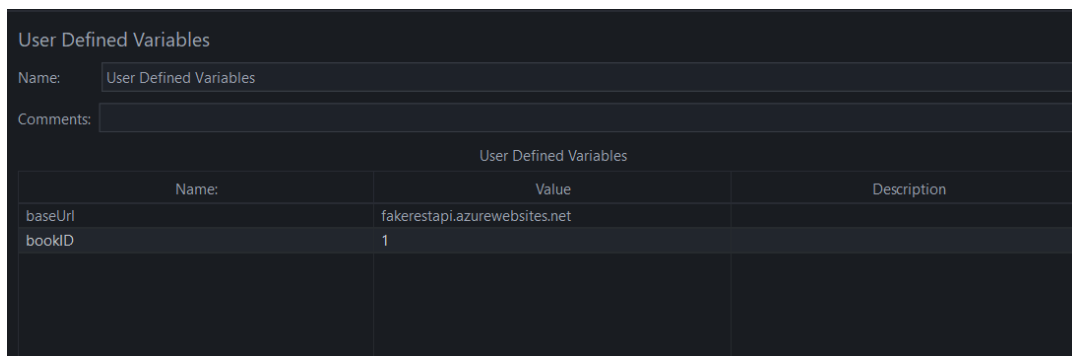
Name	Value
accept	text/plain
content-Type	application/json

## 5.5. User Defined Variables

User Defined Variables adalah Config Element yang memungkinkan kita untuk mendefinisikan variabel yang dapat digunakan di seluruh test plan. Ini sangat berguna untuk mengelola nilai yang sering digunakan, sehingga kita tidak perlu mengedit setiap sampler secara manual.



Kita dapat mendefinisikan variabel dengan nama dan nilai tertentu. Misalnya, kita bisa mendefinisikan variabel **baseUrl** dengan nilai **fakerestapi.azurewebsites.net/**



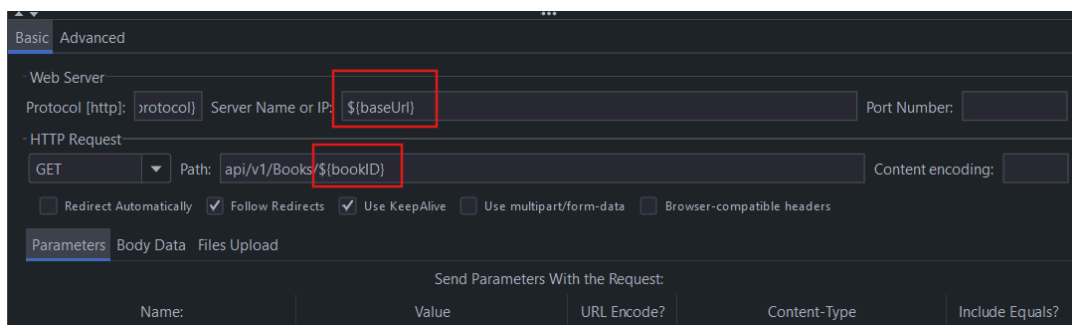
User Defined Variables

Name: User Defined Variables

Comments:

Name:	Value	Description
baseUrl	fakerestapi.azurewebsites.net	
bookID	1	

Setelah didefinisikan, variabel ini dapat digunakan di berbagai elemen dalam test plan, seperti HTTP Request, Assertions, dan lainnya. Kita cukup menggunakan sintaks **\${baseUrl}** untuk merujuk ke variabel tersebut.



Basic Advanced

Web Server

Protocol [http]: protocol Server Name or IP: \${baseUrl} Port Number:

HTTP Request

GET Path: api/v1/Books/\${bookID} Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
-------	-------	-------------	--------------	-----------------

Semua User Defined Variable diproses di awal eksekusi test plan. Ini berarti bahwa nilai variabel sudah tersedia sebelum sampler dieksekusi.

## 6. PreProcessor & PostProcessor

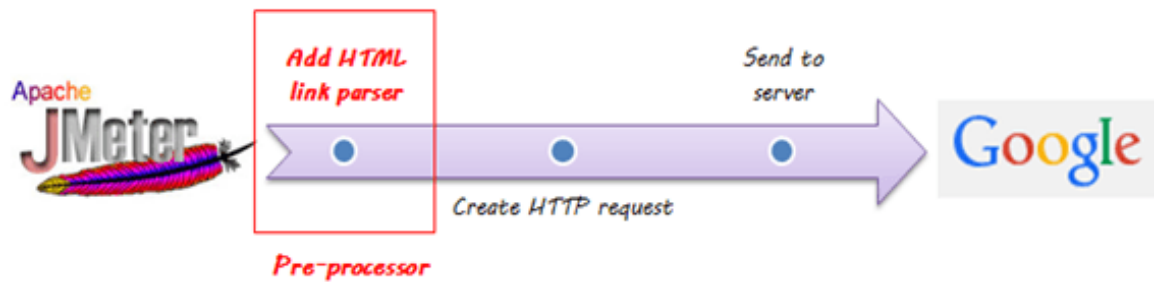
Prosesor digunakan untuk memodifikasi Sampler dalam cakupannya. Ada 2 Jenis prosesor:

1. Pre-processor
2. Post-processor

### 6.1. Pre-processor

Pre-processor menjalankan beberapa tindakan sebelum membuat Request Sampler.

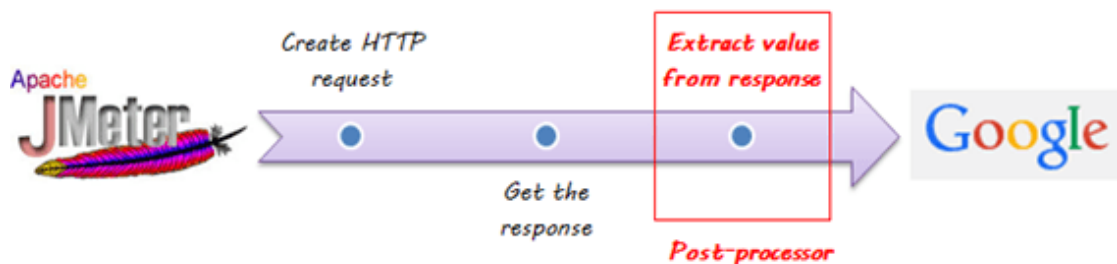
Contoh sederhana: katakanlah kita ingin JMeter "menjelajahi" situs web yang sedang diuji, mengurai tautan (periksa semua tautan di halaman) dan mengembalikan HTML. Kita akan menambahkan beberapa tindakan seperti "HTML link parser" ke pengontrol kita sebelum membuat HTTP request.



## 6.2. Post-processor

Post-processor menjalankan beberapa tindakan setelah membuat Request Sampler.

Perhatikan contoh sederhana: JMeter mengirimkan request HTTP ke server web yang sedang diuji (dll [www.google.com](http://www.google.com)) dan mendapatkan respons. Kita ingin JMeter menghentikan pengujian jika respons server adalah kesalahan. Kita dapat menggunakan Post-processor untuk melakukan tugas di atas sebagai berikut:



### Contoh Post-processor

Contoh berikut adalah petunjuk langkah demi langkah tentang cara menggunakan Post-processor di JMeter.

1. JMeter mengirimkan HTTP request ke server web yang sedang diuji [www.google.com](http://www.google.com).
2. JMeter mendapat respon dari server Google.
3. Jika respon server error, JMeter akan menghentikan pengujian.
4. Jika respon server OK (tidak ada error), JMeter akan melanjutkan pengujian.

## 7. Assertion

Assertion adalah elemen yang digunakan untuk memvalidasi hasil yang diinginkan dari skenario pengujian dan mendeteksi error conditions. Assertion digunakan untuk memastikan bahwa pengujian berjalan dengan sukses atau untuk menentukan apakah ada kesalahan. Kita dapat menambahkan satu atau lebih assertion untuk setiap sampler (request).

Ketika sebuah skenario pengujian dijalankan, respons dari sampler dievaluasi oleh assertion, dan assertion memeriksa apakah kondisi yang ditentukan terpenuhi. Jika kondisi

yang ditentukan tidak terpenuhi, assertion akan ditandai sebagai kesalahan, menunjukkan bahwa pengujian gagal atau menghasilkan hasil yang salah.

Beberapa jenis assertion yang umum termasuk:

- **Response Assertion:** Assertion ini memverifikasi teks atau nilai tertentu dalam respons dari sebuah request. Misalnya, assertion ini memeriksa request yang berisi teks tertentu atau mengembalikan kode status tertentu.

Contoh : Kita mengirimkan request user ke situs web <http://www.google.com> dan mendapatkan response server. Kita dapat menggunakan Response Assertion untuk memverifikasi apakah respons server berisi string pola yang diharapkan (misalnya "OK")

- **Duration Assertion:** Assertion ini memastikan bahwa waktu respons dari sebuah request berada dalam rentang yang ditentukan. Jika waktu respons berada di luar rentang ini, itu akan ditandai sebagai kesalahan.

Misalnya, request user dikirim ke [www.google.com](http://www.google.com) oleh JMeter dan mendapat respons dalam waktu yang diharapkan 5 ms kemudian Test Case lulus, jika tidak, test case gagal.

- **Size Assertion:** Assertion ini memvalidasi setiap respons server berisi jumlah byte yang diharapkan di dalamnya. Kita dapat menentukan ukurannya sama dengan, lebih besar dari, kurang dari, atau tidak sama dengan jumlah byte tertentu. Assertion ini sangat berguna dalam situasi seperti pengunduhan file.

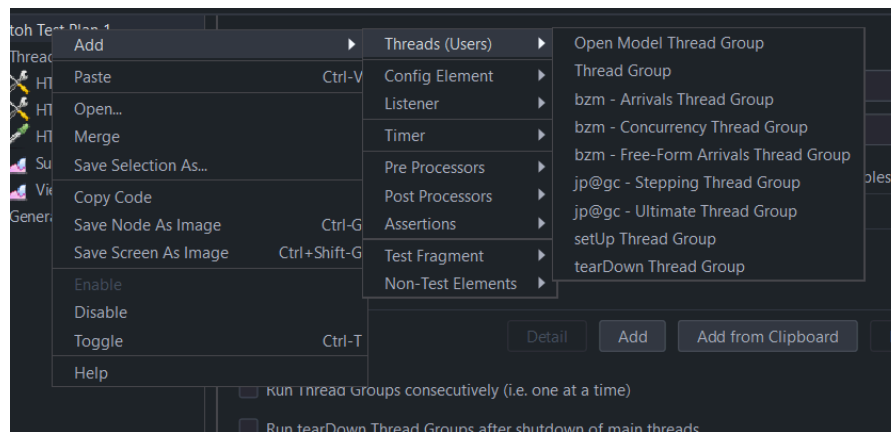
Misalnya, JMeter mengirimkan request user ke [www.google.com](http://www.google.com) dan mendapatkan respons dengan ukuran kurang dari byte yang diharapkan 5000 byte kemudian Test Case lulus. Jika tidak, test case gagal.

- **XPath Assertion:** Assertion ini melakukan validasi untuk menguji apakah data respons terdiri dari dokumen XML yang benar secara formal.
- **JSON Assertion:** Assertion ini melakukan validasi menggunakan ekspresi JSON path tertentu dalam response JSON.
- **HTML Assertion:** Pernyataan HTML memungkinkan user untuk memeriksa sintaks HTML dari data respons. Artinya data respon harus memenuhi sintaks HTML.
- **BeanShell Assertion:** Assertion ini menggunakan skrip BeanShell untuk skenario validasi kompleks di mana kondisi khusus perlu diperiksa.

Singkatnya, assertion adalah elemen yang digunakan di JMeter untuk memvalidasi hasil yang diinginkan dari skenario pengujian dan mendeteksi kesalahan. Assertion memungkinkan verifikasi cepat apakah pengujian berhasil dan apakah aplikasi berfungsi dengan benar.

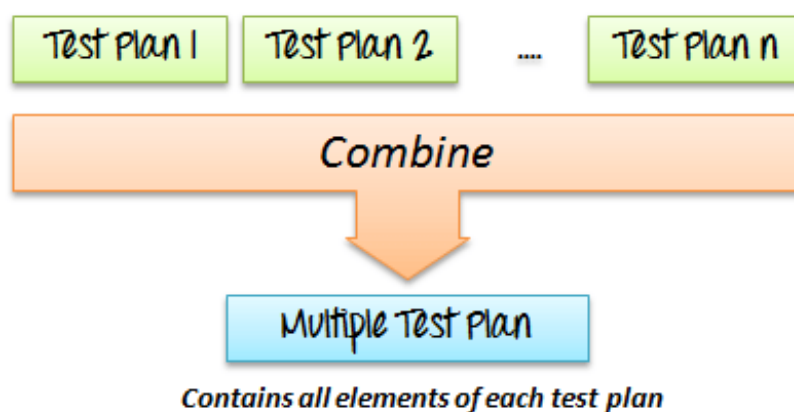
## 8. Add/Remove Elements

Elemen dapat ditambahkan ke test plan dengan mengklik kanan pada Test Plan dan pilih elemen baru dari list yang ada di "add" menu.



Alternatif lain, Kita dapat memuat elemen dari file lainnya dan menambahkannya dengan memilih opsi "merge".

Kita dapat menggabungkan satu atau beberapa test plan untuk membuat multiple test plan seperti yang ditunjukkan pada gambar di bawah :

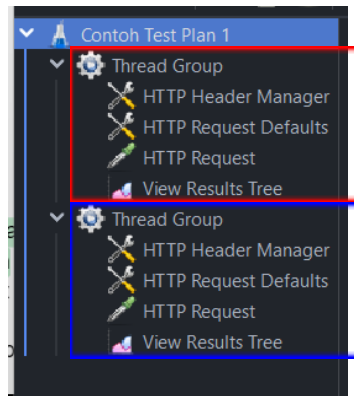


Misalkan teman-teman sudah memiliki nama test plan **Contoh Test Plan 1.jmx** di komputer. Teman-teman dapat menggabungkan test plan ini dengan test plan saat ini di JMeter untuk membuat test plan baru.

Cara menambahkan:

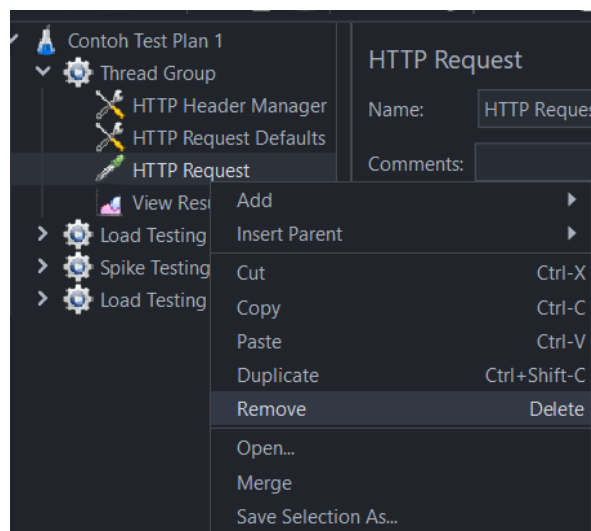
1. Klik **“File > Merge”**
2. Open file test plan yang sudah dibuat sebelumnya **“Contoh Test Plan 1.jmx”**

Sekarang semua elemen pengujian dalam file **Contoh Test Plan 1.jmx** ditambahkan ke test plan kita saat ini seperti yang ditunjukkan pada gambar di bawah.



Secara default, JMeter tidak otomatis menyimpan elemen, jadi kamu harus menyimpannya secara manual.

Untuk menghapus elemen, pastikan elemen tersebut dipilih, klik kanan pada elemen tersebut, dan pilih opsi **“remove”**.



## Tugas Day 2

1. Selain Jmeter Elements yang sudah dijelaskan sebelumnya, apalagi elemen JMeter lainnya yang Kamu ketahui?
2. Berikan minimal 1 contoh penggunaan untuk Jmeter Element yang Kamu jawab di pertanyaan no 1!