

**OPTIMASI *WEB SERVER* SIAKAD UIN MALANG MENGGUNAKAN
*CLUSTER ORCHESTRATION DOCKER SWARM***

SKRIPSI

**Oleh:
MUHAMMAD ALI ZAMRONI
NIM. 17650113**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

**OPTIMASI *WEB SERVER* SIAKAD UIN MALANG MENGGUNAKAN
*CLUSTER ORCHESTRATION DOCKER SWARM***

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
MUHAMMAD ALI ZAMRONI
NIM. 17650113

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

HALAMAN PERSETUJUAN

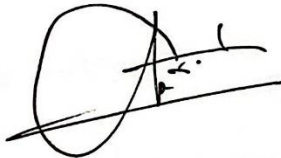
**OPTIMASI *WEB SERVER* SIAKAD UIN MALANG MENGGUNAKAN
*CLUSTER ORCHESTRATION DOCKER SWARM***

SKRIPSI

Oleh:
MUHAMMAD ALI ZAMRONI
NIM. 17650113

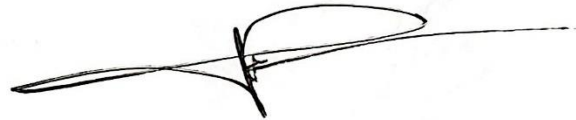
Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 19 Juni 2024

Pembimbing I,



Ajib Hanani, M.T
NIP. 19840731 202321 1 013

Pembimbing II,



Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

**OPTIMASI *WEB SERVER* SIAKAD UIN MALANG MENGGUNAKAN
*CLUSTER ORCHESTRATION DOCKER SWARM***

SKRIPSI

Oleh:
MUHAMMAD ALI ZAMRONI
NIM. 17650113

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 20 Juni 2024

Susunan Dewan Penguji

Ketua Penguji : Johan Ericka Wahyu P, M. Kom
NIP. 19831213 201903 1 004

Anggota Penguji I : Shoffin Nahwa Utama, M.T
NIP. 19860703 202012 1 003

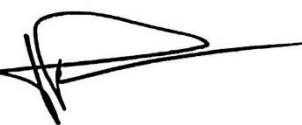
Anggota Penguji II : Ajib Hanani, M.T
NIP. 19840731 202321 1 013

Anggota Penguji III : Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

()
()
()
()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Ali Zamroni

NIM : 17650113

Fakultas / Program : Sains dan Teknologi / Teknik Informatika

Studi

Judul Skripsi : Optimasi *Web Server* SIAKAD UIN Malang Menggunakan
Cluster Orchestration Docker Swarm

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 27 Juni 2024

Yang membuat pernyataan,



Muhammad Ali Zamroni

NIM. 17650113

MOTTO

“حَيْرُ النَّاسِ أَنْفَعُهُمُ لِلنَّاسِ”

*“Sebaik-baik manusia adalah yang paling bermanfaat
untuk orang lain.”*

HALAMAN PERSEMBAHAN

Saya persembahkan karya ini kepada:

Almarhum ayah saya,

Moch. Djaenuri, S.P

Yang menjadi inspirasi dan motivasi saya hingga sampai titik ini

Ibu saya,

Munawaroh

Yang telah mendoakan dan mendukung saya hingga sampai titik ini

Kakak saya,

M. Hanif Farda, S.H

Yang telah mendukung dan menyemangati saya hingga sampai titik ini

Sahabat-sahabat saya,

Pujo Hastowo Ardi, Moh. Alfarizi, Fajar Abdi Wicaksono, Emha Ahdan Fahmi El

Muna, Faishal Al Farisi, Satria Herdiono, Muhammad Nur Kholis

Yang telah memberikan kebahagiaan selama berada di Pesantren Mahasiswa Al-

Hikam ini

Teman-teman seperjuangan,

Teknik Informatika Angkatan 2017

Semoga kita semua selalu diberi kemudahan oleh Allah SWT

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Segala puji hanya milik Allah Subhanahu Wa Ta'ala berkat segala nikmat dan kasih sayang-Nya yang telah memudahkan penulis untuk menyelesaikan skripsi yang berjudul “Optimasi *Web Server* SIAKAD UIN Malang Menggunakan *Cluster Orchetration Docker Swarm*”. Sholawat dan salam senantiasa terlimpah kepada Nabi Muhammad Sallallahu ‘Alaihi wa Sallam. Dan semoga kita semua mendapatkan syafaatnya di hari kiamat esok, Aamiin.

Penulis mengucapkan banyak rasa syukur dan terima kasih yang tak terhingga kepada semua pihak yang selalu memberikan bantuan dan motivasi kepada penulis untuk bisa menyelesaikan skripsi ini. Ucapan ini penulis sampaikan kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., IPM, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Ajib Hanani, M.T selaku dosen pembimbing I dan Dr. Fachrul Kurniawan, M.MT, IPM selaku dosen pembimbing II yang telah memberikan bantuan dan arahan kepada penulis, sehingga bisa menuntaskan skripsi ini.

5. Johan Ericka Wahyu P, M. Kom selaku dosen penguji I dan Shoffin Nahwa Utama, M.T selaku dosen penguji II yang telah menguji serta memberikan masukan sehingga penulis dapat menuntaskan skripsi dengan baik.
6. Segenap Dosen, Admin, Laboran dan Mahasiswa Program Studi Teknik Informatika yang telah memberikan banyak dukungan dan bimbingan selama pengerjaan skripsi ini.
7. Ibu, Almarhum Ayah, dan Kakak saya yang selalu memberikan dukungan dan motivasi untuk terus berusaha, dan doa yang tak putus-putusnya selalu disampaikan agar dapat menuntaskan skripsi ini dengan lancar dan baik.

Akhir kata, penulis mengakui bahwa penulisan pada skripsi ini masih banyak kekurangan. Saya berharap semoga skripsi ini diterima sebagai amal ibadah yang tulus dan bermanfaat di sisi Allah Subhanahu Wa Ta'ala. Semoga karya ini menjadi bagian dari kontribusi yang tak terputus dalam rangka memperkuat dan mengembangkan ilmu pengetahuan, serta melaksanakan tugas sebagai hamba Allah yang berkomitmen.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Malang, 27 Juni 2024

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
ABSTRAK	xiv
ABSTRACT	xv
مستخلص البحث	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Tujuan Penelitian	7
1.4 Batasan Masalah.....	7
1.5 Manfaat Penelitian	7
BAB II KAJIAN PUSTAKA	8
2.1 <i>Clustering Orchestration</i>	8
2.1.1 Definisi <i>Cluster Orchestration</i>	8
2.1.2 Penelitian Terkait <i>Cluster Orchestration</i>	8
2.2 Docker	9
2.2.1 Pengertian <i>Docker</i>	9
2.2.2 Pengertian <i>Docker Swarm</i>	11
2.2.3 Penelitian Terkait <i>Docker Swarm</i>	12
2.3 <i>Cloud Computing</i>	13
2.3.1 Pengertian <i>Cloud Computing</i>	13
2.3.2 Penelitian Terkait <i>Cloud Computing</i>	13
BAB III METODOLOGI PENELITIAN	15
3.1 Desain Penelitian.....	15
3.2 Perancangan Sistem	16
3.2.1 Kebutuhan Perangkat Keras	17
3.2.2 Kebutuhan Perangkat Lunak	17
3.2.3 Perancangan Alur Kerja Sistem.....	17
3.3 Implementasi	19
3.3.1 Instalasi <i>Operating System</i>	19
3.3.2 Instalasi <i>Nginx</i>	19
3.3.3 Instalasi <i>Docker</i>	20
3.3.4 Instalasi <i>Docker Swarm</i>	21

3.3.5 Konfigurasi <i>Docker Swarm</i>	22
3.3.6 Konfigurasi <i>Service Nginx</i>	22
3.3.7 Konfigurasi <i>PHP Service</i>	23
3.3.8 Konfigurasi <i>Postgres Service</i>	26
3.3.9 Konfigurasi <i>NFS Server</i>	26
3.4 Rancangan Pengujian	27
3.4.1 Pengujian <i>Summary Report</i>	27
3.4.2 Pengujian <i>Aggregate Report</i>	29
3.4.3 Pengujian <i>View Result in Table</i>	31
3.4.4 Pengujian <i>Response Time Graph</i>	32
BAB IV UJI COBA DAN PEMBAHASAN.....	34
4.1 Langkah Uji Coba	34
4.1.1 Uji Instalasi <i>Operating System</i>	34
4.1.2 Uji Instalasi <i>Nginx</i>	34
4.1.3 Uji Instalasi <i>Docker</i>	35
4.1.4 Konfigurasi <i>Docker Swarm</i>	36
4.1.5 Uji Konfigurasi <i>NFS server</i>	37
4.1.6 Konfigurasi <i>service</i>	38
4.2 Hasil Uji Coba.....	38
4.2.1 Hasil <i>Summary Report</i>	40
4.2.2 Hasil <i>Aggregate Report</i>	41
4.2.3 Hasil <i>View Result in Table</i>	44
4.2.4 Hasil <i>Response Time Graph</i>	44
4.3 Pembahasan.....	48
4.4 Integrasi Sains dan Islam	53
BAB V KESIMPULAN DAN SARAN.....	57
5.1 Kesimpulan	57
5.2 Saran.....	58
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR TABEL

Tabel 4.1 Hasil Summary Report Proyek SIAKAD	40
Tabel 4.2 Hasil Summary Report SIAKAD UIN Malang	41
Tabel 4.3 Hasil Aggregate Report Proyek SIAKAD	42
Tabel 4 4 Hasil Aggregate Report SIAKAD UIN Malang	43
Tabel 5.1 Kesimpulan Nilai Hasil Uji Response Time	57
Tabel 5.2 Kesimpulan Nilai Hasil Uji Throughput	58

DAFTAR GAMBAR

Gambar 1.1 Topologi server SIAKAD UIN MALANG.....	2
Gambar 1.2 Pengujian Response Time	3
Gambar 1.3 Pengujian Throughput	3
Gambar 2.1 Topologi Docker Swarm	12
Gambar 3.1 Desain Penelitian.....	16
Gambar 3.2 Topologi Docker Swarm	16
Gambar 3.3 Perancangan Alur Kerja Sistem	18
Gambar 3.4 Kode Perintah Instal Nginx	20
Gambar 3.5 Kode Perintah instal Docker	21
Gambar 3.6 Kode Perintah Instal Docker Swarm	21
Gambar 3.7 Source Code default.conf	23
Gambar 3.8 Source Code dockerfile	24
Gambar 3.9 Source Code docker-compose.yml.....	25
Gambar 3.10 Perintah Instal NFS Server Node Manager	26
Gambar 3.11 Perintah Instal NFS Server Client Node Worker	27
Gambar 4.1 Uji Instalasi Operating System.....	34
Gambar 4.2 Uji Instalasi Nginx	35
Gambar 4.3 Uji Docker	35
Gambar 4.4 Uji Join Docker Swarm	36
Gambar 4.5 Join Token Docker Swarm Node Worker 1	36
Gambar 4.6 Join Token Docker Swarm Node Worker 2	36
Gambar 4.7 Node Docker Swarm	36
Gambar 4.8 Hasil Uji NFS Server.....	37
Gambar 4.9 Hasil Uji NFS Server Client 1	37
Gambar 4.10 Hasil Uji NFS Server Client 2.....	37
Gambar 4.11 Hasil Uji Service	38
Gambar 4.12 Hasil Uji Web Proyek SIAKAD	38
Gambar 4.13 Konfigurasi Threads Jmeter Proyek SIAKAD.....	39
Gambar 4.14 Konfigurasi Threads Jmeter SIAKAD UIN Malang.....	39
Gambar 4.15 Response Time Graph Proyek SIAKAD Thread 75	44
Gambar 4.16 Response Time Graph Proyek SIAKAD Thread 100	45
Gambar 4.17 Response Time Graph Proyek SIAKAD Thread 125	46
Gambar 4.18 Response Time Graph SIAKAD UIN Malang Thread 75	46
Gambar 4.19 Response Time Graph SIAKAD UIN Malang Thread 100	47
Gambar 4.20 Response Time Graph SIAKAD UIN Malang Thread 125	48
Gambar 4.21 Diagram Response Time 75 Thread.....	49
Gambar 4.22 Diagram Response Time 100 Thread.....	50
Gambar 4.23 Diagram Response Time 125 Thread.....	51
Gambar 4.24 Diagram Throughput 75 Thread.....	51
Gambar 4.25 Diagram Throughput 100 Thread.....	52
Gambar 4.26 Diagram Throughput 125 Thread.....	53

ABSTRAK

Zamroni, Ali Muhammad. 2024. **Optimasi Web Server SIAKAD UIN Malang Menggunakan Cluster Orchestration Docker Swarm**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ajib Hanani, M.T., (II) Dr. Fachrul Kurniawan, M.MT.

Kata Kunci: *SIAKAD, Server, Cluster orchestration, Docker swarm, Average, Throughput, Threads*

SIAKAD adalah sistem informasi akademik yang dibuat bertujuan untuk memproses pengolahan data-data akademik dan data-data lainnya, agar proses aktivitas akademik bisa berjalan dengan lancar. SIAKAD UIN Malang dibuat bertujuan untuk mempermudah proses penyimpanan informasi akademik dan pengolahan data-data akademik. SIAKAD UIN Malang dibangun menggunakan satu *server* sebagai pintu masuk dan penyimpanan data-data akademik. apabila SIAKAD UIN Malang diakses oleh banyak pengguna, *server* SIAKAD UIN Malang akan terbebani dan dapat mengakibatkan *server* menjadi *overload*, sehingga SIAKAD UIN Malang tidak dapat diakses. Penelitian ini dilakukan bertujuan untuk mengoptimasi *server* yang dapat menjadi opsi atau pilihan agar *server* SIAKAD UIN Malang tidak begitu terbebani, yaitu dengan cara mengoptimasi *web server* SIAKAD UIN Malang menggunakan *cluster orchestration docker swarm*. *Cluster orchestration docker swarm* merupakan pembagian *server* yang dilakukan dengan menerapkan lebih dari satu *server*, agar kinerja *server* dapat lebih optimal. Penggunaan *docker swarm* juga bertujuan untuk mempermudah pemeliharaan sebuah *service* atau layanan. *Docker swarm* juga berguna untuk memperingan beban *server* karena beban akan dibagi kepada *node worker* yang dapat diperbanyak. Proses pengujian dilakukan dengan beberapa skema pengujian yaitu dengan melakukan uji pembagian *threads* atau *request* yang ditujukan ke halaman *login* SIAKAD, dengan melakukan uji sebanyak 75 *threads*, 100 *threads*, dan 125 *threads* yang dimana dilakukan terhadap SIAKAD UIN Malang dan proyek SIAKAD dari penelitian ini. Saat pengujian diperoleh nilai *average* dan *throughput* dari proyek SIAKAD adalah 43763 *ms* dan 1,12/*sec* dengan 75 *threads*, 55029 *ms* dan 1,17/*sec* dengan 100 *threads*, 67841 *ms* dan 1,28/*sec* dengan 125 *threads*, sedangkan SIAKAD UIN MALANG memperoleh nilai *average* dan *throughput* sebesar 48977 *ms* dan 1,04/*sec* dengan 75 *threads*, 64182 *ms* dan 1,07/*sec* dengan 100 *threads*, 78580 *ms* dan 1,15/*sec* dengan 125 *threads*. Setelah melakukan pengujian tersebut proyek SIAKAD mendapatkan hasil lebih optimal dengan nilai *average* yang lebih kecil dan nilai *throughput* yang lebih besar.

ABSTRACT

Zamroni, Ali Muhammad. 2024. **Optimization of the SIAKAD UIN Malang Web Server Using Docker Swarm Cluster Orchestration**. Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Ajib Hanani, M.T., (II) Dr. Fachrul Kurniawan, M.MT.

SIAKAD is an academic information system designed to process academic data and other related data to ensure smooth academic activities. SIAKAD UIN Malang was created to facilitate the storage and management of academic information and data. SIAKAD UIN Malang is built using a single server for data access and storage. When accessed by multiple users, the SIAKAD UIN Malang server can become overloaded, making it inaccessible. This research aims to optimize the server as an option to prevent the SIAKAD UIN Malang server from becoming too burdened by using Docker Swarm cluster orchestration. Cluster orchestration with Docker Swarm involves distributing the server load across multiple servers to enhance performance. The use of Docker Swarm also aims to simplify service maintenance and lighten the server load by distributing it among multiple worker nodes. Testing was conducted with several schemes, including dividing threads or requests directed at the SIAKAD login page, with tests performed using 75 threads, 100 threads, and 125 threads on both SIAKAD UIN Malang and the SIAKAD project from this research. The average response time and throughput for the SIAKAD project were 43763 ms and 1.12/sec with 75 threads, 55029 ms and 1.17/sec with 100 threads, and 67841 ms and 1.28/sec with 125 threads. In comparison, SIAKAD UIN Malang had average response times and throughput of 48977 ms and 1.04/sec with 75 threads, 64182 ms and 1.07/sec with 100 threads, and 78580 ms and 1.15/sec with 125 threads. The results showed that the SIAKAD project achieved more optimal performance with lower average response times and higher throughput.

Keywords: *SIAKAD, Server, Cluster orchestration, Docker swarm, Average, Throughput, Threads*

مستخلص البحث

زمـروني، علي محمد. ٢٠٢٤. تحسين خادم الويب سياكد جامعة الإسلامية الحكومية بمالانج باستخدام تنظيم مجموعة ضوكير سوارم . البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: أحيب حناني، ماجستير. المشرف الثاني: د. فخر الكرنياوان، الماجستير.

الكلمات الرئيسية سياكد، خادم، تنسيق العناقيد، ضوكير سوارم ، المتوسط، الإنتاجية، مؤشرات الترابط

سياكد هو نظام معلومات أكاديمي الذي يهدف معالجة البيانات الأكاديمية والبيانات الأخرى، حتى تتم عملية الأنشطة الأكاديمية. صنع تطبيق سياكد جامعة الإسلامية الحكومية بمالانج تهدف لسهولة عملية تخزين المعلومات الأكاديمية وإدارة البيانات الأكاديمية. وتخزين البيانات الأكاديمية. إذا تم الوصول إلى سياكد جامعة الإسلامية الحكومية بمالانج من قبل العديد لدى المستخدمين، فسيكون خادم سياكد جامعة الإسلامية الحكومية مثقلا ويمكن أن يتسبب في زيادة تحميل الخادم، بحيث لا يمكن الوصول إلى سياكد جامعة الإسلامية الحكومية بمالانج. تم إجراء هذا البحث بهدف تحسين الخادم الذي يمكن أن يكون خيارا بحيث لا يكون خادم سياكد جامعة الإسلامية الحكومية بمالانج مثقلا، أي عن طريق تحسين خادم الويب سياكد جامعة الإسلامية الحكومية بمالانج باستخدام تنسيق مجموعة "ضوكير سوارم". مجموعة "ضوكير سوارم" هي قسم خادم يتم تنفيذه عن طريق نشر أكثر من خادم واحد، بحيث يكون أداء الخادم أكثر مثالية. يهدف استخدام "ضوكير سوارم" أيضا إلى تسهيل صيانة الخدمة. يعد "ضوكير سوارم" مفيدا أيضا لتخفيف تحميل الخادم لأنه سيتم مشاركة التحميل بين العقد العاملة التي يمكن زيادتها. تم تنفيذ عملية الاختبار باستخدام عدة مخططات اختبار، وهي اختبار توزيع المواضيع أو الطلبات الموجهة إلى صفحة تسجيل الدخول إلى "سياكد"، عن طريق اختبار 75 موضوعا و100 موضوعا و125 موضوعا تم تنفيذها على سياكد جامعة الإسلامية الحكومية بمالانج. ومشروع "سياكد" من هذا البحث. أثناء الاختبار، كان متوسط وقيم الإنتاجية لمشروع "سياكد" 43763 مليون ثانية و1.12/ثانية مع 75 خيط، 55029 مليون ثانية و 1.17/ثانية مع 100 خيط، 67841 مليون ثانية مع 1.28/ ثانية مع 125 خيط، بينما تم الحصول على "سياكد" جامعة الإسلامية الحكومية بمالانج متوسط وقيم الإنتاجية 48977 مليون ثانية و1.04/ثانية مع 75 موضوع، 64182 مليون ثانية و1.15/ثانية مع 125 موضوع. وبعد إجراء هذه الاختبارات، حصل مشروع "سياكد" على نتائج أفضل بمتوسطة قيمة أصغر وقسمة إنتاجية أكبر.

BAB I

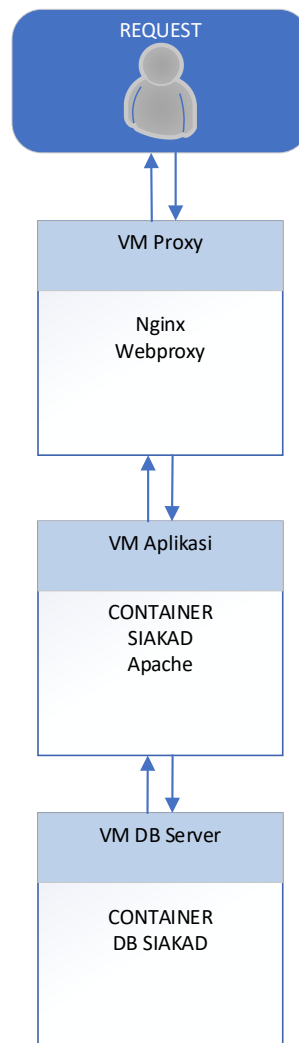
PENDAHULUAN

1.1 Latar Belakang

SIKAD adalah Sistem Informasi Akademik yang dirancang secara khusus berbasis *web* yang kebanyakan digunakan oleh perguruan tinggi yang dirancang untuk memproses pengelolaan data-data akademik dan data-data lainnya, sehingga proses aktivitas akademik bisa dikelola menjadi informasi yang bermanfaat. Data-data yang dikelola didalam SIKAD bertujuan untuk memudahkan dan mempercepat penyampaian informasi kepada mahasiswa, dosen, dan seluruh sivitas akademika (Ekowansyah dkk., 2017).

Universitas Islam Negeri Maulana Malik Ibrahim Malang memiliki SIKAD dengan tujuan mempermudah proses penyampaian informasi akademik dan pengelolaan data-data akademik. SIKAD tersebut dibangun menggunakan satu *server* sebagai pintu masuk kedalam SIKAD. Jika SIKAD diakses oleh pengguna yang sangat banyak, *server* akan terbebani dan dapat mengakibatkan *server* menjadi *overload*, dan mengakibatkan SIKAD tidak dapat diakses. Agar hal tersebut bisa teratasi, *Optimasi* akan dilakukan pada *server* SIKAD dengan menerapkan *cluster orchestration server* menggunakan *docker swarm*.

SIKAD UIN MALANG memiliki topologi web server tunggal seperti gambar 1.1.



Gambar 1.1 Topologi *server* SIAKAD UIN MALANG

Ketika melakukan pengetesan web SIAKAD UIN Malang dengan menggunakan 1000 *request* dalam satu kali periode. Hasil yang didapatkan dari pengetesan melalui software *jmeter* didapatkan jumlah *sample time*(ms) adalah 6413514. Untuk menghitung *response time* digunakan rumus jumlah seluruh nilai *sample time* dibagi jumlah keseluruhan *sample* / jumlah *request* Fandy .,dkk (2022) maka $6413514 \div 1000 = 6413,514$ ms.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status
1	04:14:18.624	Pengujian 1-6	HTTP Request	302	✓
2	04:14:18.625	Pengujian 1-7	HTTP Request	316	✓
3	04:14:18.626	Pengujian 1-8	HTTP Request	360	✓
4	04:14:18.629	Pengujian 1-10	HTTP Request	362	✓
5	04:14:18.629	Pengujian 1-12	HTTP Request	368	✓
6	04:14:18.637	Pengujian 1-15	HTTP Request	375	✓
7	04:14:18.633	Pengujian 1-16	HTTP Request	388	✓
8	04:14:18.638	Pengujian 1-20	HTTP Request	388	✓
9	04:14:18.632	Pengujian 1-11	HTTP Request	396	✓
10	04:14:18.639	Pengujian 1-21	HTTP Request	398	✓
11	04:14:18.621	Pengujian 1-2	HTTP Request	447	✓
13	04:14:18.644	Pengujian 1-25	HTTP Request	491	✓
12	04:14:18.640	Pengujian 1-18	HTTP Request	493	✓
14	04:14:18.639	Pengujian 1-19	HTTP Request	498	✓
15	04:14:18.627	Pengujian 1-9	HTTP Request	554	✓
16	04:14:18.644	Pengujian 1-23	HTTP Request	667	✓
17	04:14:18.654	Pengujian 1-34	HTTP Request	717	✓
18	04:14:18.637	Pengujian 1-14	HTTP Request	741	✓
20	04:14:18.647	Pengujian 1-27	HTTP Request	759	✓
21	04:14:18.647	Pengujian 1-28	HTTP Request	776	✓
19	04:14:18.622	Pengujian 1-4	HTTP Request	782	✓
23	04:14:18.655	Pengujian 1-35	HTTP Request	791	✓
22	04:14:18.636	Pengujian 1-17	HTTP Request	808	✓
25	04:14:18.648	Pengujian 1-26	HTTP Request	823	✓
24	04:14:18.643	Pengujian 1-24	HTTP Request	825	✓
26	04:14:18.643	Pengujian 1-22	HTTP Request	829	✓
27	04:14:18.623	Pengujian 1-3	HTTP Request	850	✓

Scroll automatically?
 Child samples?
 No of Samples: 1000
 Latest Sample: 1

Gambar 1.2 Pengujian *Response Time*

Dalam pengujian web SIAKAD UIN MALANG didapatkan *throughput* dari *samples HTTP Request* adalah 74,4/sec dan *error* 0,00%.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
HTTP Request	1000	6413	7012	11438	11893	12460	302	12677	0.00%	74.4/sec	816.98	9.08
TOTAL	1000	6413	7012	11438	11893	12460	302	12677	0.00%	74.4/sec	816.98	9.08

Gambar 1.3 Pengujian *Throughput*

Berdasarkan latar belakang masalah diatas kami menerapkan landasan religius yang diambil dari beberapa ayat di Al-Qur'an. seperti yang difirmankan Allah Subhanahu wa Ta'ala dalam penggalan surat Al-Ma'idah ayat 2 sebagai berikut.

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ ۖ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ ۚ وَاتَّقُوا اللَّهَ ۖ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

“Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya.”(Q.S Al Ma'idah: 2)

Qur'an Surat Al Ma'idah pada ayat 2 dipaparkan pada Tafsir Al-Munir oleh Syaikh Wahbah az-Zuhaili (Az-Zuhaili, 2016). Dalam segi *balaaghah*, ayat (وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ ۖ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ) susunan kalimat tersebut memiliki makna *al-muqaabalah*, yang memiliki makna didatangkanlah dua makna atau lebih pada awal kalimat lalu didatangkannya lagi kalimat berlawanan arti secara tertib pada bagian akhir kalimat. Dan tolong-menolonglah kamu dalam mengerjakan kebajikan dan takwa dan meninggalkan kemungkarannya.

Secara *Mufradaat lughawiyah* (لُغَوِيَّةٌ) adalah bentuk kebaikan secara umum, meliputi setiap hal yang telah diperintahkan oleh syari'at dan hati merasa tentram terhadapnya. (التَّقْوَىٰ) adalah menjalankan segala perintah-Nya dan menjauhi segala larangan-Nya. (الْإِثْمِ) adalah kemaksiatan dan perbuatan dosa, yaitu setiap hal sesuatu yang membuat tidak tentram dan tidak tenang dalam hati dan sikap tidak ingin dilihat dan diketahui oleh orang lain. (وَالْعُدْوَانِ) maknanya melanggar segala aturan yang telah ditetapkan dan batasan-batasan Allah SWT.

Pada tafsir dan penjelasan potongan ayat tersebut Syaikh Wahbah az-Zuhaili memaparkan ayat tersebut tentang saling membahu menolong, dan saling bersinergi dalam menjalankan kebajikan. Jangan kamu saling tolong menolong untuk berbuat dosa dan maksiat, yaitu semua hal yang tidak diperbolehkan oleh syari'at yang membuat hati menjadi gusar terhadapnya dan sikap menyembunyikan agar orang lain tidak melihat dan mengetahuinya. Kemudian jangan juga kamu saling tolong menolong dalam berbuat pelanggaran terhadap hak-hak orang lain (al-'udwaan). (al-itsm) dosa dan (al-'udwaan) pelanggaran,

mencakup segala bentuk kejahatan yang pelakunya berdosa, melakukan penganiayaan dan pelanggaran yang telah Allah SWT tetapkan batasan-batasan (hududallah) terhadap orang lain. Bertakwalah kamu semuanya kepada Allah SWT dengan cara menjalankan segala perintah-Nya dan menjauhi segala larangan-Nya. Sesungguhnya Allah SWT sangatlah pedih atas siksaan-Nya kepada orang yang bermaksiat, melanggar dan membangkang.

Pada ayat (إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ) terdapat *lafzhul jalaalah* yaitu huruf lam yang ada pada lafadz Allah, disebutkan *isim zhahir*, padahal sebetulnya dapat menggunakan *dhamir*. Hal tersebut bermaksud untuk menciptakan perasaan dan suasana segan dan takut dalam hati. Adapun makna tersebut merupakan perkataan yang singkat, padat, dan tegas mencakup setiap kebaikan dan keburukan, kemungkaran dan kebijakan disertai kesadaran bahwa sesungguhnya Allah SWT selalu melihat dan mengawasi baik dalam keadaan sendiri ataupun dalam keramaian.

Dari penjelasan ayat diatas kami menarik landasan religius untuk penelitian ini. Terdapat dua landasan religius yang pertama yaitu kemudahan dan yang kedua adalah kenyamanan untuk semua orang yang akan menggunakan manfaat dari penelitian ini.

Sementara itu, dari pemaparan masalah yang ada di atas dibutuhkan solusi untuk mengoptimalkan *server* SIAKAD dalam hal ini peneliti menggunakan konsep *clustering orchestration* dengan menggunakan *docker swarm*. *Clustering* pada *server* adalah teknologi untuk pembagian beban kerja *server* dengan cara penggabungan beberapa *server* yang bekerja secara bersamaan dan masih dalam satu sistem tunggal (Putra dkk., 2020). Sedangkan *cluster orchestration* adalah

sistem yang dirancang untuk mengelola penerapan aplikasi ter kontainer dalam kelompok klaster berskala besar, sistem ini dapat menjalankan ratusan ribu pekerjaan di ribuan mesin. Sebagian besar sistem mendukung *multitenancy* dengan menjadwalkan aplikasi yang dimiliki oleh banyak pengguna pada kumpulan sumber daya komputasi bersama, memungkinkan penggunaan sumber daya yang lebih baik (Rodriguez & Buyya, 2018).

Untuk *docker* sendiri adalah aplikasi berbasis teknologi *open source* yang memungkinkan pengembang atau orang lain untuk membangun, menjalankan, menguji, dan meluncurkan aplikasi dalam sebuah *container*. *Docker* mempercepat proses pengemasan aplikasi dan komponennya ke dalam wadah yang terisolasi, memungkinkannya berjalan di infrastruktur lokal tanpa perubahan konfigurasi apa pun pada *container* (Dwiyatno dkk., 2020).

Dengan mengimplementasi *cluster orchestration* dengan menggunakan *docker swarm* pada *server* SIAKAD UIN Maulana Malik Ibrahim Malang, maka diharapkan SIAKAD akan lebih ringan apabila diakses oleh banyak *client* atau pengguna.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah untuk mengetahui nilai *response time* dan *throughput* pada *virtual server* SIAKAD yang akan dibuat menggunakan metode *cluster orchestration docker swarm*.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menghitung *response time* dan *throughput* pada *server* SIAKAD UIN Malang dengan menggunakan metode cluster orchestration *docker swarm*.

1.4 Batasan Masalah

Agar menghindari pembahasan yang akan keluar dari pembahasan materi, maka ditentukan batasan masalah. Pada penelitian ini batasan masalah yang ada adalah sebagai berikut:

1. *Cluster orchestration* menggunakan *docker swarm*.
2. *Server* yang digunakan menggunakan *virtual server*.
3. Mengukur *response time* menggunakan jumlah klien yang ditetapkan.
4. Perbandingan menggunakan arsitektur web server SIAKAD UIN MALANG dan arsitektur *cluster orchestration docker swarm*.

1.5 Manfaat Penelitian

Manfaat penelitian yang dapat diambil pada penelitian ini adalah untuk pihak PTIPD UIN Maulana Malik Ibrahim Malang sebagai opsi pengoptimalan *server* SIAKAD UIN Malang.

BAB II

KAJIAN PUSTAKA

2.1 *Clustering Orchestration*

2.1.1 Definisi *Cluster Orchestration*

Clustering pada *server* adalah teknologi untuk pembagian beban kerja *server* dengan cara penggabungan beberapa *server* yang bekerja secara bersamaan dan masih dalam satu sistem tunggal (Putra dkk., 2020).

Sedangkan menurut Rodriguez & Buyya (2018) *cluster orchestration* adalah sistem yang dirancang untuk mengelola penerapan aplikasi ter kontainer dalam kelompok kluster berskala besar, sistem ini dapat menjalankan ratusan ribu pekerjaan di ribuan mesin. Sebagian besar sistem mendukung *multitenancy* dengan menjadwalkan aplikasi yang dimiliki oleh banyak pengguna pada kumpulan sumber daya komputasi bersama, memungkinkan penggunaan sumber daya yang lebih baik.

2.1.2 Penelitian Terkait *Cluster Orchestration*

Menurut Rodriguez & Buyya (2018) dijelaskan dalam jurnalnya bahwa platform *container orchestration* muncul karena dirancang untuk pengelolaan penerapan aplikasi kontainer dalam kluster dalam skala yang besar. Sistem ini mampu menjalankan ratusan ribu pekerjaan di ribuan mesin. Untuk melakukannya secara efisien, mereka harus mengatasi beberapa tantangan penting diantaranya skalabilitas, toleransi kesalahan dan ketersediaan, pemanfaatan sumber daya yang

Efisien, dan memaksimalkan *throughput* permintaan. Jurnal ini mempelajari tentang sistem manajemen *cluster orchestration* dan mengusulkan taksonomi yang mengidentifikasi berbagai mekanisme yang dapat digunakan untuk memenuhi tantangan yang ada.

Penelitian berikutnya (Zhou .,dkk, 2021) dalam jurnalnya membahas tentang kontainerisasi menunjukkan efisiensi dalam penerapan aplikasi di *cloud computing*. Kontainer dapat memecahkan program kompleks dengan dependensinya di lingkungan yang terisolasi dan membuat aplikasi menjadi lebih *portable*, sehingga diadopsi dalam kluster *High Performance Computing* (HPC). *Singularitas*, awalnya dirancang untuk sistem HPC yang menjadi *runtime container* standar *de facto* mereka. Namun demikian, pengelolaan beban kerja HPC konvensional tidak memiliki dukungan layanan mikro dan manajemen kontainer yang sangat terintegrasi, berbeda dengan *container orchestration*. Solusi untuk itu diusulkan arsitektur hibrid yang mengintegrasikan kluster HPC dan *Cloud* secara mulus dengan sedikit gangguan pada sistem HPC di mana *container orchestration* dilakukan pada dua tingkat.

2.2 Docker

2.2.1 Pengertian Docker

Menurut (Dwiyatno .,dkk, 2020) *Docker* adalah aplikasi berbasis teknologi *open source* yang dapat dilakukan tindakan pengembang atau orang lain untuk membangun, menjalankan, menguji, dan mengimplementasikan aplikasi dalam sebuah *container*. *Docker* dapat mempercepat proses pengemasan aplikasi dan

komponennya ke dalam wadah yang terisolasi, memungkinkannya berjalan di infrastruktur lokal tanpa perubahan konfigurasi apa pun pada *container*.

Docker yang berfungsi untuk membuat, menjalankan, melakukan percobaan dan menjalankan aplikasi di dalam sebuah *container* yang berbasis teknologi *open source*. *Docker* secara cepat memproses pemaketan aplikasi dengan komponennya dalam sebuah *container* yang terisolasi, sehingga bisa menjalankan dalam infrastruktur lokal tanpa melakukan perubahan konfigurasi dalam *container*.

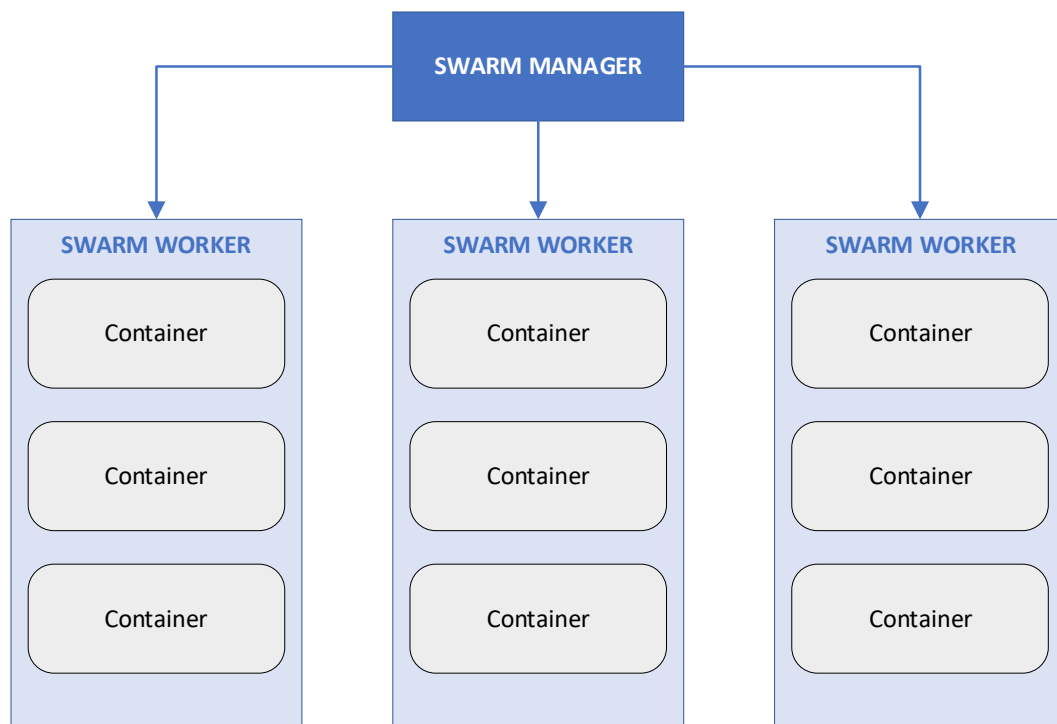
Virtualisasi *server* adalah inovasi yang banyak digunakan di perusahaan TI. Ada banyak teknologi virtualisasi ringan *docker* adalah salah satunya, yang merupakan *platform open source*. Teknologi ini memungkinkan pengembangan dan sistem admin untuk merancang, membuat, dan menjalankan aplikasi menggunakan *docker*. Peneliti membuat evaluasi dari kinerja *docker container* dan *virtual machines* menggunakan alat benchmark atau pengukur standar seperti *Phoronix*, *Sysbench*, dan *Apache benchmark*, yang mencakup keseluruhan kinerja CPU, kinerja baca atau tulis penyimpanan, *throughput* memori, pengukuran kecepatan operasi, dan uji beban, (Potdar dkk., 2020).

Sementara itu *docker* memiliki alat orkestrasi bawaan yang memungkinkan para pengguna untuk mengatur dan mengelola kluster dari kontainer *docker*. Dengan menggunakan *docker swarm* memungkinkan untuk menjalankan dan mengelola aplikasi yang terdiri atas beberapa kontainer *docker* yang bekerja bersama sebagai satu kesatuan.

2.2.2 Pengertian *Docker Swarm*

Docker swarm adalah kumpulan dari *host* yang semuanya berjalan pada *docker*, yang telah diatur untuk berinteraksi satu sama lain dalam konfigurasi kluster. Setelah dikonfigurasi pengguna akan dapat menggunakan semua perintah yang telah dijalankan untuk satu *host* dan *docker swarm* akan memutuskan penempatan kontainer pengguna dengan menggunakan strategi penyebaran untuk memutuskan *host* yang paling tepat. (McKendrick, 2020)

Docker swarm terdiri dari beberapa *host docker* yang berjalan pada mode *swarm*, diantaranya *swarm manager* dan *swarm worker*. *Swarm manager* bertugas mengelola keanggotaan dan *swarm worker* bertugas untuk menjalankan layanan *swarm*, untuk topologi *docker swarm* dapat dilihat pada gambar 2.1. *Host docker* yang sedang berjalan dapat menjadi *swarm manager*, *swarm worker*, atau melakukan kedua tugas tersebut. Jika ingin membuat layanan, peneliti dapat menentukan statusnya (jumlah salinan, port yang akan digunakan atau dipublish oleh layanan). *Docker swarm* bekerja dengan mempertahankan keadaan yang diinginkan. Misalnya, apabila *node worker* tidak tersedia, *docker* akan menjadwalkan tugas *node* tersebut ke *node* yang lain. Tugas yang menjalankan kontainer adalah bagian dari layanan *swarm* dan dikelola oleh *swarm manager*. (Afis, 2018)



Gambar 2.1 Topologi *Docker Swarm*

2.2.3 Penelitian Terkait *Docker Swarm*

Implementasi *load balancing server web* berbasis *docker swarm* berdasarkan penggunaan sumber daya *Memory Host* (Rexa .,dkk, 2019) peneliti menjelaskan tentang penerapan *docker swarm* menggunakan *load balancing* berdasarkan penggunaan sumber daya *memory host*. *Clustering web server* dapat digunakan pada kontainer *docker*, yaitu untuk membagi tugas agar *web server* menjadi lebih optimal. Hal tersebut bertujuan agar meminimalisir terjadinya “a single point of failure” (SPOF) pada *web server*. Pada penelitian ini peneliti mengatur trafik *web server* agar rata antar *host* dengan *load balancing* berdasarkan waktu, sumber daya *memory* dan *failover*. Pengujian dilakukan terhadap sistem menggunakan uji fungsionalitas dan kinerja *load balancer* berlandaskan *memory utilization*. Hasil dari pengujian penggunaan *docker swarm*

dapat melakukan penerapan *load balancer* berlandaskan sumber daya *memory* dan *failover* berdasarkan waktu, *Docker swarm* dapat mengatasi permasalahan pembagian *resource* antar *host*.

2.3 Cloud Computing

2.3.1 Pengertian Cloud Computing

Cloud computing adalah perangkat penyimpanan yang digunakan untuk menyimpan dan mengelola data dan aplikasi. *Cloud computing* digunakan dalam infrastruktur TI yang menyediakan layanan kepada pelanggan melalui Internet. *Cloud computing* disediakan oleh pihak ketiga. Hal ini memberikan lingkungan yang hemat biaya bagi pengguna akhir. Kita dapat mengakses dan membuat cadangan semua informasi melalui Internet. *Cloud computing* memiliki model layanan dan pengiriman yang memberikan layanan kepada pelanggan dengan biaya rendah. Akses data dan aplikasi yang mudah digunakan dan mudah. Model layanan komputasi awan meliputi *Software as a Service (SAAS)*, *Platform as a Service (PAAS)*, dan *Infrastructure as a Service (IAAS)*. Layanan yang berbeda ini memberikan kemampuan berbeda untuk mengakses dan mengelola data dan aplikasi. (Rajeswari, 2019)

2.3.2 Penelitian Terkait Cloud Computing

Menurut Saraswat & Tripathi (2020) pada jurnalnya dijelaskan tentang *Cloud Computing* mengacu pada jaringan yang memungkinkan kekuatan pemrosesan, aplikasi, dan penyimpanan yang didistribusikan ke banyak sistem komputer jarak jauh. Dalam platform *cloud computing*, sumber daya TI digunakan

dan dibagikan sesuai permintaan melalui internet. Untuk organisasi dan konsumen direkomendasikan penggunaan layanan bayar sesuai pemakaian yang ditawarkan oleh penyedia layanan cloud, seperti *IaaS*, *PaaS*, *SaaS*, dan *DaaS*. Ada banyak sekali penyedia layanan di pasaran saat ini, dan fitur serta layanan mereka yang beragam membuat sangat sulit untuk menemukan penyedia yang tepat untuk kebutuhan jangka panjang. Dari segi pangsa pasar, tiga penyedia terbesar adalah *Amazon*, *Microsoft*, dan *Google*. Peneliti menganalisis beberapa komputasi, manajemen penyimpanan, performa, dan alat lain yang ditawarkan oleh tiga pemimpin teratas dalam teknologi komputasi awan: *AWS*, *Azure*, dan *GCP*. Jurnal ini merangkum dan membandingkan fitur *AWS*, *Azure*, dan *GCP* untuk membantu organisasi dan pengguna memilih fitur yang tepat untuk memenuhi kebutuhan jangka panjang mereka.

Kesimpulan yang didapatkan yang pertama tentang *AWS* yaitu jangkauan lebih global, memiliki banyak pusat data. Lebih luas, banyak layanan yang stabil dan andal, memiliki sejumlah besar produk. Butuh solusi publik. Jika anda adalah perusahaan yang lebih besar, kebijakan penetapan harga beralih ke perusahaan tingkat perusahaan yang membutuhkan solusi serbaguna bahkan yang mahal. Sementara *Azure* migrasi di *cloud* untuk pertama kalinya. Sebagian besar aplikasi/solusi berbasis *Microsoft/Windows*. Membutuhkan mode/solusi penerapan hibrid. Untuk *GCP* berfokus pada model berbasis kontainer (kontainer adalah jenis perangkat lunak yang dapat berbagi *kernel* OS secara virtual tanpa memerlukan mesin virtual). Berfokus pada aplikasi berbasis web.

BAB III

METODOLOGI PENELITIAN

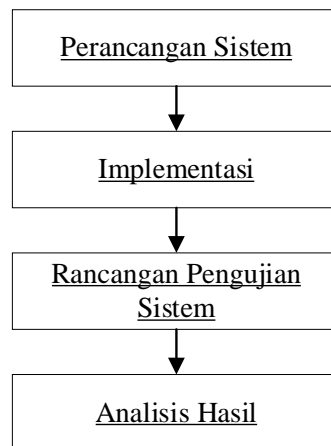
Pada bab tiga dijelaskan langkah-langkah perancangan optimasi *web server* SIAKAD UIN Malang menggunakan *cluster orchestration docker swarm*.

3.1 Desain Penelitian

Pada penelitian ini desain penelitian dapat diuraikan sebagai berikut :

1. Tahap awal, perancangan sistem yang akan dibangun dan keperluan apa saja kebutuhan perangkat keras dan perangkat lunak. Perancangan desain sistem, topologi penerapan *docker swarm*.
2. Tahap kedua adalah implementasi untuk membuat model sistem kemudian dijadikan sebuah sistem yang awalnya telah dirancang pada perancangan sistem.
3. Tahap ketiga adalah rancangan pengujian sistem yang telah dirancang sebelumnya, dari pengujian tersebut kemudian dilakukan analisis hasil pengujian.
4. Tahap keempat adalah analisis hasil dari hasil yang didapatkan dari pengujian sistem.

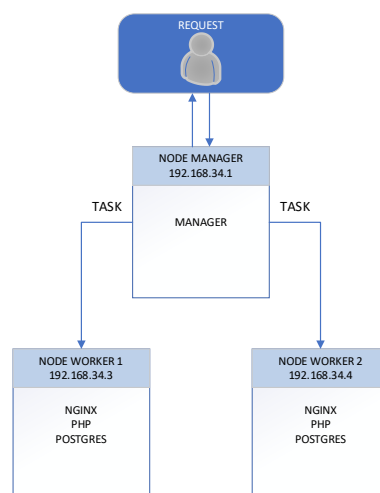
Desain penelitian pada penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Desain Penelitian

3.2 Perancangan Sistem

Sebelum membuat suatu sistem dibutuhkan suatu perancangan untuk membuat sistem. Penelitian ini dibuat untuk membuat sistem yang berjalan diatas 3 node *server* yang akan di kluster yang dikelola oleh *docker swarm*. Untuk kebutuhan yang akan dibutuhkan diantaranya kebutuhan perangkat keras dan kebutuhan perangkat lunak. Untuk topologi sistem yang akan dirancang dapat dilihat pada gambar 3.2.

Gambar 3.2 Topologi *Docker Swarm*

3.2.1 Kebutuhan Perangkat Keras

Memaparkan spesifikasi perangkat keras yang dibutuhkan untuk pengujian penelitian. Berikut beberapa kebutuhan perangkat keras yang digunakan :

1. Laptop MSI GL62M 7RDX
2. Memori (RAM): 24GB
3. *Processor: Intel Core i7 7th Gen*

3.2.2 Kebutuhan Perangkat Lunak

Memaparkan perangkat lunak yang dibutuhkan untuk pengujian penelitian. Berikut beberapa kebutuhan perangkat lunak yang digunakan :

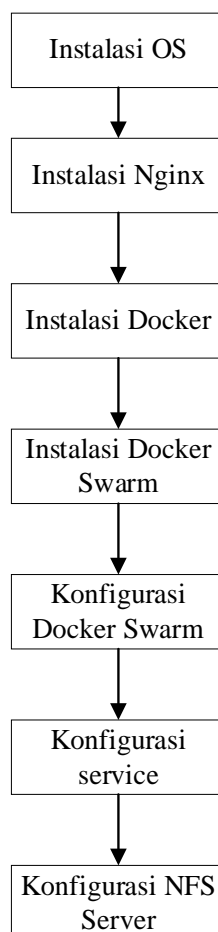
1. *Virtual box*
2. Sistem Operasi *Ubuntu Server 22.04.3 LTS*
3. *Docker machine*
4. *Docker ce*
5. *Nginx*
6. *PHP*
7. *NFS Server*
8. *Jmeter*

3.2.3 Perancangan Alur Kerja Sistem

Pada perancangan alur kerja sistem dijelaskan tentang tahap-tahap konfigurasi kebutuhan yang telah dijelaskan. Untuk lebih jelasnya berikut tahap-tahap konfigurasi untuk membuat sistem:

1. Instalasi *ubuntu server* sebagai *operating system* pada *VPS*. Menggunakan *operating system Linux Ubuntu Server 22.04.03 LTS*.
2. Instalasi *nginx* pada *operating system*.
3. Instalasi *docker* pada *operating system*.
4. Instalasi *docker swarm*.
5. Konfigurasi *docker swarm*.
6. Konfigurasi *service*
7. Konfigurasi *NFS server*

Perancangan alur kerja sistem digambarkan pada gambar 3.3.



Gambar 3.3 Perancangan Alur Kerja Sistem

3.3 Implementasi

Pada sub bab implementasi, akan dijelaskan tentang tahap-tahap yang sudah dipeleaskan pada perancangan alur kerja sistem.

3.3.1 Instalasi *Operating System*

Pada penelitian ini merancang *server* siacad menggunakan *cluster orchestration* yang ada pada *docker swarm*. Sistem operasi yang digunakan adalah *Linux Ubuntu Server 22.04 LTS 64 bit*. *Ubuntu 22.04 LTS* (Long term Support) adalah versi yang terbaru untuk sistem operasi *Linux Ubuntu* yang dirilis pada April 2022 dan mendukung pembaruan dan dukungan selama 5 tahun. Sistem operasi ini menawarkan banyak pembaruan dan fitur baru yang dapat digunakan oleh pengguna *desktop* dan *server*, serta oleh pengembang dan *administrator system* yang mencari stabilitas dan keamanan yang tinggi.

3.3.2 Instalasi *Nginx*

Nginx adalah *web server* yang memiliki performa cepat dan ringan, *nginx* memproses banyak request atau permintaan klien dengan baik. *Nginx* atau sering disebut “Engine-X” dapat diakses dengan banyak *request* atau banyak klien pada waktu yang bersamaan, artinya *web server nginx* dapat menerima banyak sekali *traffic* dengan menggunakan *load balancing* pada *nginx* (Fahmi, 2020). *Nginx* dikenalkan ke publik pada bulan Oktober 2004 yang dibuat oleh Igor Sysoev. Pada waktu itu permasalahan tentang performa *web server* sangat banyak dikeluhkan kemudian Igor merilis dan meyakinkan ke publik bahwa *nginx* bisa menjadi solusi atas permasalahan tersebut. *Nginx* memiliki beberapa fitur yaitu:

load balancing, reverse proxy, Fast CGI support, handling static, web sockets, IPv6, TSL/SSL (Chandra, 2019).

Nginx sudah tersedia pada *repository default* ubuntu. Untuk menginstallnya terlebih dahulu melakukan pembaharuan lokal sistem agar mendapatkan akses ke daftar-daftar paket yang terbaru. Setelah itu *nginx* bisa diinstal dengan mengetikkan kode perintah pada gambar 3.4 pada terminal.

```
$ sudo apt-get update  
$ sudo apt-get install nginx
```

Gambar 3.4 Kode Perintah Instal *Nginx*

3.3.3 Instalasi *Docker*

Docker pertama kali diperkenalkan ke dunia tanpa pengumuman sebelumnya dan sedikit gembar-gembor oleh Solomon Hykes, pendiri dan *CEO* perusahaan yang kemudian disebut *dotCloud*, dalam pembicaraan pencahayaan lima menit di Python Developers Conference di Santa Clara, California pada 15 Maret 2013. Pada saat pengumuman ini, hanya sekitar 40 orang di luar *dotCloud* yang diberi kesempatan untuk bermain dengan *docker*. Dalam beberapa minggu setelah pengumuman ini, ada sejumlah pers yang mengejutkan. Proyek ini dengan cepat menjadi sumber terbuka dan tersedia untuk umum di GitHub, di mana siapa pun dapat mengunduh dan berkontribusi pada proyek tersebut. Selama beberapa bulan berikutnya, semakin banyak orang di industri ini mulai mendengar tentang *docker* dan bagaimana *docker* akan merevolusi cara pembuatan, pengiriman, dan pengoperasian perangkat lunak, dan dalam setahun, hampir semua orang tau tentang *docker*, tetapi banyak yang masih tidak yakin persisnya, dan mengapa orang begitu bersemangat tentangnya (Kane & Matthias, 2018).

Sebelum menginstal *docker* ada beberapa tahapan, yang pertama kali dilakukan adalah menyiapkan repositori *docker*. Kita dapat menginstal dan memperbarui *docker* dari repositori. Langkah-langkah untuk menginstall *docker* pada ubuntu *server* diantaranya dengan memasukkan perintah-perintah pada gambar 3.5.

```
//memperbarui daftar paket yang ada
$ sudo apt update
//menginstall beberapa paket yang memungkinkan penggunaan paket melalui
https
$ sudo apt install apt-transport-https ca-certificates curl software-
properties-common
// menambahkan kunci GPG untuk repositori resmi Docker ke sistem
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
//menambahkan repositori Docker ke sumber APT
$ echo "deb [arch=$(dpkg --print-architecture) signed-
by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" |
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
//memperbarui lagi daftar paket agar penambahan paket dapat dikenali
$ sudo apt update
//memastikan menginstall repo Docker yang bukan dari repo default ubuntu
$ apt-cache policy docker-ce
//meninstall docker
$ sudo apt install docker-ce
//memeriksa apakah docker sudah aktif
$ sudo systemctl status docker
```

Gambar 3.5 Kode Perintah instal *Docker*

3.3.4 Instalasi *Docker Swarm*

Instalasi *docker swarm* dilakukan setelah menginstall *docker* pada setiap virtual *server*. Kemudian pada *server* manajer kita lakukan inisialisasi agar menjadi manajer untuk *docker swarm* kemudian memasukkan perintah berikut untuk membuat *swarm* baru, dapat dilihat pada gambar 3.6.

```
$ docker swarm init --advertise-addr <MANAGER IP>
```

Gambar 3.6 Kode Perintah Instal *Docker Swarm*

Setelah menginisialisasi pada salah satu virtual *server* sebagai manajer, akan ditampilkan kode token join yang akan ditambahkan pada setiap *server* yang bertugas menjadi *worker*.

3.3.5 Konfigurasi *Docker Swarm*

Setelah ditentukan salah satu *server* menjadi manajer. Maka tugas selanjutnya adalah menginisialisasi *server* lainnya menjadi *worker*, dengan cara menambahkan kode join token yang telah dibuat oleh *server* yang bertugas menjadi manajer kepada masing-masing *node worker*.

3.3.6 Konfigurasi *Service Nginx*

Konfigurasi *service* adalah tahap dimana node worker diisi atau diinstal *service* bisa disebut juga dengan layanan. *Service* yang diinstal di node worker berupa *nginx*, *php*, dan *postgres*. Untuk kegunaan dari masing-masing *service* yaitu *nginx* sebagai *web server* atau perantara antara pengguna dan internet serta penyeimbang beban. *Php* berfungsi sebagai wadah pengembangan web yang berisi komponen-komponen pembentukan web. *Postgres* merupakan database yang gunanya untuk menyimpan database. Ketiga *service* tersebut akan di letakkan di dua *node worker* menggunakan replikasi.

Konfigurasi *nginx service* dilakukan dengan menambahkan folder *nginx* pada folder SIAKAD yang akan dibuat yaitu di *php service*, dan menambahkan file default.conf yang berisi pada gambar 3.7.

```

server {
    listen 8000;
    index index.php index.html;
    error_log /var/log/nginx/error.log;
    access_log /var/log/nginx/access.log;
    root /var/www/app/public;

    location ~ /\.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.(php|php5))(\.+)$;
        fastcgi_pass php:9000;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }
    location / {
        try_files $uri $uri/ /index.php?$query_string;
        gzip_static on;
    }
}

```

Gambar 3.7 Source Code default.conf

Listen 8000 sendiri sebagai port tujuan untuk membuka *service* agar SIAKAD bisa diakses.

3.3.7 Konfigurasi *PHP Service*

Konfigurasi *php service* adalah isi dari seluruh *source code* SIAKAD yang akan dibuat sebagai pembanding SIAKAD UIN MALANG. Kemudian ditambahkan juga file pendukung agar dapat dibungkus dan dijadikan *images* di *docker*, file tersebut adalah *dockerfile*. Isi file *dockerfile* dapat dilihat pada gambar 3.8.

```

FROM php:8.1.6-fpm-alpine

WORKDIR /var/www/app

RUN apk update && apk add \
    build-base \
    git \
    curl \
    zip \
    libzip-dev \
    libpq-dev \
    postgresql \
    postgresql-client \
    unzip \
    nano

RUN docker-php-ext-configure pgsql
RUN docker-php-ext-install pdo pdo_pgsql pgsql zip exif pcntl

COPY --from=composer:latest /usr/bin/composer /usr/local/bin/composer

USER root

RUN chmod 777 -R /var/www/app

```

Gambar 3.8 Source Code dockerfile

File dockerfile menginisialisasi bahwa *php* yang digunakan adalah *php* 8.1.6 dan menggunakan database *postgres*. Setelah file dockerfile ditambahkan dilanjutkan dengan menambahkan file docker-compose.yml yang berguna untuk menginisialisasikan *service* apa saja yang akan digunakan dan agar dapat digunakan *service* tersebut. Isi dari file tersebut dapat dilihat pada gambar 3.9.

```

version: '3.9'

services:
  nginx:
    image: nginx:alpine
    ports:
      - "8000:8000"
    volumes:
      - nfs_share:/usr/share/nginx/html
    deploy:
      mode: replicated

```



```

        replicas: 4
        restart_policy:
          condition: on-failure
        placement:
          constraints:
            - node.role != manager

postgres:
  image: postgres:14.4-alpine
  ports:
    - "5432:5432"
  volumes:
    - pg_data:/var/lib/postgresql/data
  environment:
    - POSTGRES_DB=siakad_pt
    - POSTGRES_USER=root
    - POSTGRES_PASSWORD=root
  deploy:
    replicas: 1
    restart_policy:
      condition: on-failure
    placement:
      constraints: [node.role != manager]

php:
  image: zroni4/siakad:1.0
  ports:
    - "9000:9000"
  volumes:
    - nfs_share:/var/www/html
  deploy:
    mode: replicated
    replicas: 4
    condition: on-failure
    placement:
      constraints: [node.role != manager]

volumes:
  nfs_share:
    driver: local
    driver_opts:
      type: nfs
      o: addr=192.168.34.7,rw
      device: ":/srv/nfs_share"

```

Gambar 3.9 Source Code docker-compose.yml

Terdapat tiga *service* dan masing-masing port sudah ditentukan. File ini yang akan menjadikan node worker menjadi terisi tiga *service*, dan sistem akan dapat berjalan.

3.3.8 Konfigurasi *Postgres Service*

Konfigurasi *postgres service* dilakukan dengan cara yang pertama yaitu menseting koneksi database di folder *docker-compose.yml* dengan cara menyamakan port database agar database dapat tersambung.

3.3.9 Konfigurasi *NFS Server*

NFS (Network File System) adalah protokol yang memungkinkan akses jarak jauh ke file sistem melalui jaringan. Dalam konteks *docker swarm*, *NFS* dapat digunakan untuk menyediakan penyimpanan yang dapat diakses oleh kontainer berbeda yang berjalan pada node berbeda dalam *cluster swarm*. Hal ini memungkinkan kontainer untuk secara konsisten bertukar data secara *real-time*.

Konfigurasi *NFS* akan diterapkan pada node manajer, sehingga node manajer juga berfungsi sebagai *NFS Server*. Kemudian node *worker* juga diinstal *NFS* tetapi hanya sebagai *client*. Berikut kumpulan perintah untuk menerapkan *nfs server* yang akan diterapkan pada *node* manajer pada gambar 3.10.

```
$ sudo apt-get update
$ sudo apt-get install nfs-kernel-server
$ sudo mkdir -p /srv/nfs_share
$ sudo chown nobody:nogroup /mnt/nfsshare
$ sudo chmod 755 /srv/nfs_share
$ sudo nano /etc/exports
/srv/nfs_share/siakad 192.168.34.5(rw, sync, no_subtree_check)
192.168.34.6(rw, sync, no_subtree_check)
$ sudo exportfs -a
```

Gambar 3.10 Perintah Instal *NFS Server Node Manager*

Pada *node worker* juga dikonfigurasi agar dapat menerima perintah-perintah atau konfigurasi yang dijalankan pada *node* manajer yang dapat dilihat pada gambar 3.11.

```
$ sudo apt-get update
$ sudo apt-get install nfs-common
$ sudo mkdir -p /srv/nfs_share
$ sudo nano /etc/fstab
192.168.34.7:/srv/nfs_share /srv/nfs_share nfs defaults,user,exec 0 0
$ sudo mount -a
```

Gambar 3.11 Perintah Instal *NFS Server Client Node Worker*

3.4 Rancangan Pengujian

Rancangan pengujian sistem yang dilakukan pada penelitian ini menggunakan *tools Jmeter*, yang akan menampilkan hasil dari *summary report*, *aggregate report*, *view results in table*, dan *response time graph*.

3.4.1 Pengujian *Summary Report*

Summary report adalah pendengar (listener) pada *JMeter* yang memberikan gambaran umum mudah tentang hasil tes kinerja. *Summary report* menampilkan metrik dasar untuk setiap sampel atau kueri yang diuji, memberikan data ringkas termasuk beberapa metrik utama diantaranya adalah jumlah sampel, waktu respons rata-rata, waktu respons minimum dan maksimum, tingkat kesalahan, throughput, serta data yang dikirim dan diterima per detik.

Pada *summary report* akan menampilkan macam-macam hasil uji, diantaranya adalah.

1. Label: nama masing-masing *sampler* yang diuji ditunjukkan pada label. Setiap *query* atau transaksi memiliki label tertentu untuk membedakan hasil dalam laporan.

2. *# Samples*: jumlah total sampel atau permintaan yang dikirim selama periode pengujian untuk menunjukkan seberapa banyak permintaan yang dilakukan terhadap *server*.
3. *Average*: waktu *respons* atau *response time* rata-rata dalam milidetik untuk semua sampel dalam pengujian tersebut. *Average* adalah metrik yang penting untuk mengukur *respons time* keseluruhan. Persamaan *average* atau *response time* adalah:

$$\text{Response time} = \frac{\sum Ti}{N} \quad (3.1)$$

Keterangan:

$\sum Ti$ = total seluruh waktu respon klien

N = jumlah total klien

4. *Min*: waktu *respons* minimum yang tercatat dalam milidetik. *Min* menunjukkan waktu tercepat di mana server merespons permintaan selama pengujian.
5. *Max*: waktu *respons* maksimum yang tercatat dalam milidetik. *Max* menunjukkan waktu terlama yang diambil oleh server untuk merespon permintaan selama pengujian.
6. *Std dev*: standar *deviasi* dari waktu respon, yang mengukur variasi atau *dispersi* dari waktu respon. Standar *deviasi* yang rendah menunjukkan bahwa waktu respon konsisten, sedangkan yang tinggi menunjukkan variasi yang besar dalam waktu respon.
7. *Error*: persentase permintaan yang gagal selama pengujian. *Error* dihitung sebagai jumlah permintaan yang gagal dibagi dengan total jumlah

permintaan, kemudian dikalikan dengan 100. *Error rate* yang tinggi menunjukkan masalah pada aplikasi atau *server*.

8. *Throughput*: mengukur jumlah permintaan yang diproses per menit atau per detik. *Throughput* menunjukkan kapasitas *server* dalam menangani permintaan dalam satuan waktu tertentu. *Throughput* dihitung dengan persamaan:

$$\textit{Throughput} = \frac{\textit{Total request}}{\textit{Total time}} \quad (3.2)$$

9. *Received Kb/s*: kecepatan data yang diterima oleh klien dalam *kilobyte* per detik yang digunakan untuk mengukur berapa banyak data yang diterima dari *server* selama pengujian.
10. *Sent kb/s*: kecepatan data yang dikirim oleh klien dalam *kilobyte* per detik yang digunakan untuk mengukur berapa banyak data yang dikirim ke *server* selama pengujian.
11. *Avg. bytes*: Ukuran rata-rata dari respon dalam *byte* yang menunjukkan ukuran rata-rata data yang diterima dari *server* untuk setiap permintaan.

3.4.2 Pengujian *Aggregate Report*

Aggregate report pada JMeter adalah salah satu pendengar (listener) yang digunakan untuk mengumpulkan dan menganalisis hasil tes kinerja secara lebih rinci. Pada *aggregate report* akan menampilkan macam-macam hasil uji, diantaranya adalah.

1. *Label*: menunjukkan nama dari setiap *sampler* atau permintaan yang diuji. *Label* ini digunakan untuk mengidentifikasi berbagai jenis permintaan yang dikirim selama pengujian.

2. *# Samples*: jumlah total sampel atau permintaan yang dikirim selama pengujian yang menunjukkan seberapa banyak permintaan yang telah dibuat dan diterima oleh *server*.
3. *Average*: waktu respon atau *response time* rata-rata dalam milidetik untuk semua sampel dalam pengujian tersebut yang memberikan gambaran umum tentang seberapa cepat *server* merespos permintaan. persamaan *average* atau *response time* adalah:

$$\text{Response time} = \frac{\sum Ti}{N} \quad (3.3)$$

Keterangan:

$\sum Ti$ = total seluruh waktu respon klien

N = jumlah total klien

4. *Median*: waktu respon median dalam milidetik. Median adalah nilai tengah dari semua waktu respon, yang berarti 50% dari permintaan memiliki waktu respon di bawah nilai ini, dan 50% di atasnya.
5. *90% Line*: menunjukkan waktu respon di mana 90% dari permintaan selesai. Misalnya, jika 90% line adalah 200ms, itu berarti 90% dari semua permintaan diselesaikan dalam 200ms atau kurang.
6. *95% Line*: menunjukkan waktu respon di mana 95% dari permintaan selesai. Nilai ini memberikan gambaran tentang kinerja aplikasi di bawah beban tinggi, menunjukkan bahwa 95% dari semua permintaan diselesaikan dalam waktu tersebut atau kurang.
7. *99% Line*: menunjukkan waktu respon di mana 99% dari permintaan selesai. Merupakan metrik yang sangat ketat dan menunjukkan waktu respon dari hampir semua permintaan, dengan hanya 1% yang mungkin lebih lambat.

8. *Min* (minimum): waktu respon minimum yang tercatat dalam milidetik. Menunjukkan waktu tercepat di mana *server* merespon permintaan selama pengujian.
9. *Max* (maximum): waktu respon maksimum yang tercatat dalam milidetik. Menunjukkan waktu terlama yang diambil oleh *server* untuk merespon permintaan selama pengujian.
10. *Error*: persentase kesalahan yang terjadi selama pengujian dihitung sebagai jumlah permintaan yang gagal dibagi dengan total jumlah permintaan, kemudian dikalikan dengan 100. *Error rate* yang tinggi menunjukkan masalah pada aplikasi atau *server*.
11. *Throughput*: mengukur jumlah permintaan yang diproses per menit atau per detik yang menunjukkan kapasitas *server* dalam menangani permintaan dalam satuan waktu tertentu. *Throughput* dihitung dengan persamaan:
$$\text{Throughput} = \frac{\text{Total request}}{\text{Total time}} \quad (3.4)$$
12. *Received kb/s*: kecepatan data yang diterima oleh klien dalam *kilobyte* per detik yang digunakan untuk mengukur berapa banyak data yang diterima dari *server* selama pengujian.
13. *Sent kb/s*: Kecepatan data yang dikirim oleh klien dalam *kilobyte* per detik yang digunakan untuk mengukur berapa banyak data yang dikirim ke *server* selama pengujian.

3.4.3 Pengujian *View Result in Table*

Pengujian *view result in table* dilakukan untuk melihat atau menampilkan hasil pengujian dalam bentuk tabel, dimana akan diperlihatkan secara rinci

penjelasan tentang setiap sampel yang diambil selama pengujian berlangsung. Pengujian *view result in table* sangat diperlukan karena untuk mengukur *response time*, dimana terdapat penjelasan waktu sampel atau *sample time* pada setiap sampel yang akan dihitung untuk memastikan hasil *response time* dengan cara menjumlah waktu total pada sampel *threads* dibagi dengan jumlah total *threads*.

Pada *view result in table* juga banyak terdapat hasil yang ditampilkan diantaranya adalah.

1. *Sample #*: nomor urut dari sampel yang diambil.
2. *Start Time*: waktu saat sampel dimulai.
3. *Thread Name*: nama *thread* yang mengirimkan sampel.
4. Label: nama dari sampel atau permintaan yang diuji.
5. *Sample Time (ms)*: waktu respon dalam milidetik untuk sampel tersebut.
6. Status: status dari sampel (*sukses/fail*).
7. *Bytes*: jumlah *byte* data yang diterima dari *server*.
8. *Sent Bytes*: jumlah *byte* data yang dikirim ke *server*.
9. *Latency (ms)*: waktu yang diambil dari permintaan dikirim hingga pertama kali *byte* diterima.
10. *Connect Time (ms)*: waktu yang diambil untuk membuat koneksi ke *server*.

3.4.4 Pengujian *Response Time Graph*

Response time graph merupakan salah satu *listener* atau elemen pendengar pada *Jmeter* yang berfungsi untuk menggambarkan waktu *response time* dari permintaan yang diuji dalam bentuk grafik. Pengujian ini sangat berguna untuk menganalisis kinerja *server* secara visual serta untuk memahami waktu respon

atau *response time* berubah selama pengujian. Pengujian ini merupakan penggambaran dari hasil *sample time* pada pengujian *view result in table*.

BAB IV

UJI COBA DAN PEMBAHASAN

Penjelasan bab empat yaitu tentang uji coba dan pembahasan tentang optimasi *web server* SIAKAD UIN Malang menggunakan *cluster orchestration docker swarm*.

4.1 Langkah Uji Coba

4.1.1 Uji Instalasi *Operating System*

Instalasi *operating system* bisa dikatakan berhasil jika pada *ubuntu server* dapat login kedalam sistem *server*, seperti pada gambar 4.1.

```
login as: ubuntu
ubuntu@192.168.34.3's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-84-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Apr  6 09:26:58 AM UTC 2024

System load:          0.0
Usage of /:           57.4% of 6.06GB
Memory usage:        12%
Swap usage:          0%
Processes:           111
Users logged in:     1
IPv4 address for docker0: 172.17.0.1
IPv4 address for docker_gwbridge: 172.18.0.1
IPv4 address for enp0s3: 10.0.2.15
IPv4 address for enp0s8: 192.168.34.3

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
```

Gambar 4.1 Uji Instalasi *Operating System*

4.1.2 Uji Instalasi *Nginx*

Instalasi *nginx* dilakukan agar *server* memiliki banyak fungsi diantaranya sebagai *web server* yang dapat melayani permintaan *http* dan *https*, sebagai

reverse proxy, load balancing, caching, security dan static content serving.

Dengan menginstal *nginx* web server menjadi fleksibel, mendapatkan performa yang lebih, dan memiliki berbagai fitur yang bisa disesuaikan untuk memenuhi kebutuhan.

Apabila *nginx* sudah terinstal, dapat dilakukan pengecekan dan dapat dilihat versi berapa *nginx* yang sudah terinstal pada *virtual server*. Pengecekan *nginx* dapat dilihat seperti gambar 4.2.

```
ubuntu@manager:~$ nginx -V
nginx version: nginx/1.18.0 (Ubuntu)
built with OpenSSL 3.0.2 15 Mar 2022
TLS SNI support enabled
configure arguments: --with-cc-opt='-g -O2 -ffile-pre
Werror=format-security -fPIC -Wdate-time -D_FORTIFY_S
are/nginx --conf-path=/etc/nginx/nginx.conf --http-loc
modules-path=/usr/lib/nginx/modules --http-client-bod
temp-path=/var/lib/nginx/scgi --http-uwsgi-temp-path=
_module --with-http_auth_request_module --with-http_v
ules/http-geoip2 --with-http addition module --with-h
```

Gambar 4.2 Uji Instalasi *Nginx*

4.1.3 Uji Instalasi *Docker*

Uji instalasi *docker* dapat dilakukan dengan cara memberi perintah untuk melihat versi *docker*. Untuk melihatnya dapat dilihat pada gambar 4.3.

```
ubuntu@manager:~$ docker version
Client: Docker Engine - Community
Version:          24.0.6
API version:      1.43
Go version:       go1.20.7
Git commit:       ed223bc
Built:            Mon Sep  4 12:31:44 2023
OS/Arch:          linux/amd64
Context:          default
```

Gambar 4.3 Uji *Docker*

4.1.4 Konfigurasi *Docker Swarm*

Uji konfigurasi *docker swarm* dimulai dengan cara inisiasi *node manager* dan akan diberikan hasil *join* token agar token di masukkan ke masing-masing *node worker*, lihat gambar 4.4.

```
ubuntu@manager:~$ docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-689f1a2nuqixgj6welup80zdr1lk4urwtia1fy0s5ivrqcj3e2-ai9dizrgfxq369frfi9zkq8jn 192.168.5.23:2377
```

Gambar 4.4 Uji *Join Docker Swarm*

Selanjutnya memasukkan *join* token ke masing-masing *node worker* seperti gambar 4.5 dan 4.6.

```
ubuntu@worker1:~$ docker swarm join --token SWMTKN-1-689f1a2nuqixgj6welup80zdr1lk4urwtia1fy0s5ivrqcj3e2-ai9dizrgfxq369frfi9zkq8jn 192.168.5.23:2377
```

Gambar 4.5 *Join Token Docker Swarm Node Worker 1*

```
ubuntu@worker2:~$ docker swarm join --token SWMTKN-1-689f1a2nuqixgj6welup80zdr1lk4urwtia1fy0s5ivrqcj3e2-ai9dizrgfxq369frfi9zkq8jn 192.168.5.23:2377
```

Gambar 4.6 *Join Token Docker Swarm Node Worker 2*

Setelah melakukan *join* token ke setiap *node worker*, langkah selanjutnya yaitu melihat apakah masing-masing *node* sudah bergabung pada *node manager*. Apabila *node worker* sudah bergabung maka akan tertampil seperti gambar 4.7.

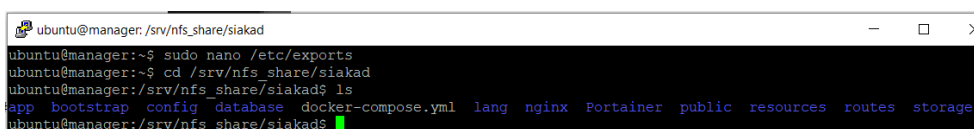
```
ubuntu@manager:~$ docker node ls
ID                HOSTNAME          STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
u0hmi84wqx3e22ripccq2z3zmx *  manager          Ready     Active           Leader             24.0.6
kr0fmsu5pkgt430ltvk4wjww4      worker1          Ready     Active           -                  24.0.6
6ids8y5u5ziipekgw234py913      worker2          Ready     Active           -                  24.0.6
```

Gambar 4.7 *Node Docker Swarm*

Semua *node* sudah tergabung pada *docker swarm*, terdapat *manager*, *worker 1*, dan *worker 2*.

4.1.5 Uji Konfigurasi *NFS server*

Uji konfigurasi *nfs server* diawali dengan menginstal *nfs server* pada node manajer. Konfigurasi dilakukan dengan cara memperbarui sistem, kemudian mengunduh *nfs server*. Kemudian membuat direktori *nfs server* dan mengkonfigurasi file *fstab*. Kemudian memberi perintah *export* agar *server nfs client* juga mendapat data dari *nfs server* seperti gambar 4.8.



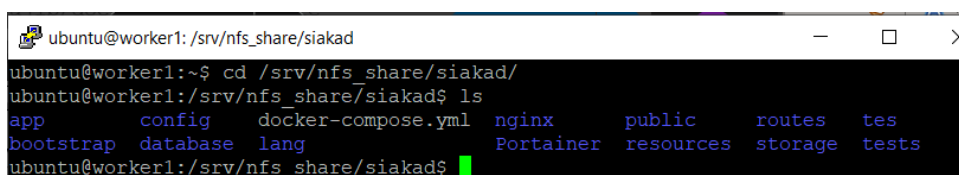
```

ubuntu@manager: /srv/nfs_share/siakad
ubuntu@manager:~$ sudo nano /etc/exports
ubuntu@manager:~$ cd /srv/nfs_share/siakad
ubuntu@manager: /srv/nfs_share/siakad$ ls
app bootstrap config database docker-compose.yml lang nginx Portainer public resources routes storage
ubuntu@manager: /srv/nfs_share/siakad$

```

Gambar 4.8 Hasil Uji *NFS Server*

Gambar 4.8 menunjukkan bahwa folder yang dibuat sudah di isi dengan folder utama program yaitu folder *siakad*. Selanjutnya akan melihat apakah pada *nfs server client* juga sama hasilnya dengan *nfs server*, hasil uji *nfs server client* dapat dilihat seperti gambar 4.9 dan 4.10.



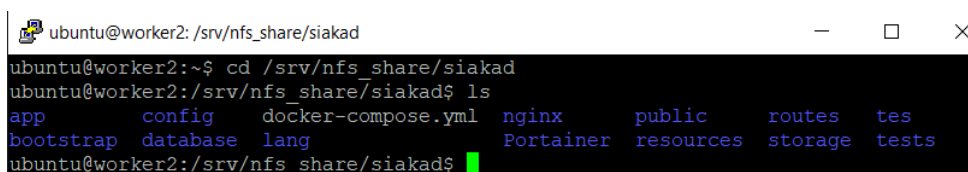
```

ubuntu@worker1: /srv/nfs_share/siakad
ubuntu@worker1:~$ cd /srv/nfs_share/siakad/
ubuntu@worker1: /srv/nfs_share/siakad$ ls
app config docker-compose.yml nginx public routes tes
bootstrap database lang Portainer resources storage tests
ubuntu@worker1: /srv/nfs_share/siakad$

```

Gambar 4.9 Hasil Uji *NFS Server Client 1*

NFS server client memiliki folder utama *siakad* juga yang artinya konfigurasi *NFS server client* pada *node worker 1* berhasil. Selanjutnya *node worker 2* akan di cek juga.



```

ubuntu@worker2: /srv/nfs_share/siakad
ubuntu@worker2:~$ cd /srv/nfs_share/siakad
ubuntu@worker2: /srv/nfs_share/siakad$ ls
app config docker-compose.yml nginx public routes tes
bootstrap database lang Portainer resources storage tests
ubuntu@worker2: /srv/nfs_share/siakad$

```

Gambar 4.10 Hasil Uji *NFS Server Client 2*

Pada *worker 2* hasil uji *nfs server* juga berhasil dengan melihat folder utama *siakad* juga ada pada *worker 2*.

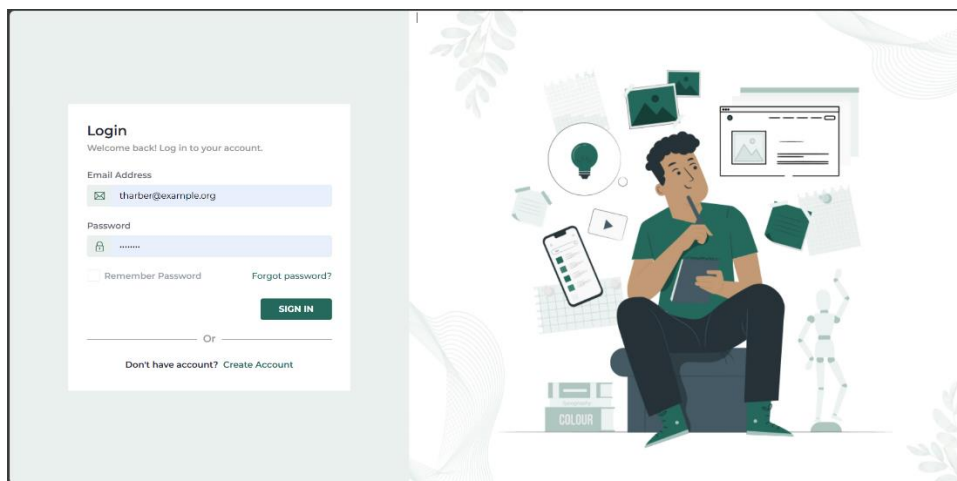
4.1.6 Konfigurasi *service*

Untuk menjalankan *service* dapat dilakukan dengan mengetikkan perintah. Kemudian melihat *service* yang berjalan seperti gambar 4.11.

```
ubuntu@manager:/srv/nfs_share/siakad$ docker service ls
ID                NAME          MODE          REPLICAS  IMAGE                                  PORTS
jl2hkzqid9d      siakad_nginx  replicated    4/4        192.168.34.7:5000/nginx:alpine       *:8000->8000/tcp
8i2hibxy9i2i     siakad_php    replicated    4/4        192.168.34.7:5000/zroni4/siakad:1.0  *:9000->9000/tcp
yofa6vhdxnus     siakad_postgres replicated    1/1        postgres:14.4-alpine                 *:5432->5432/tcp
```

Gambar 4.11 Hasil Uji *Service*

Setelah *service* sudah berjalan semua, langkah selanjutnya melihat apakah web *siakad* sudah bisa dimuat seperti gambar 4.12.

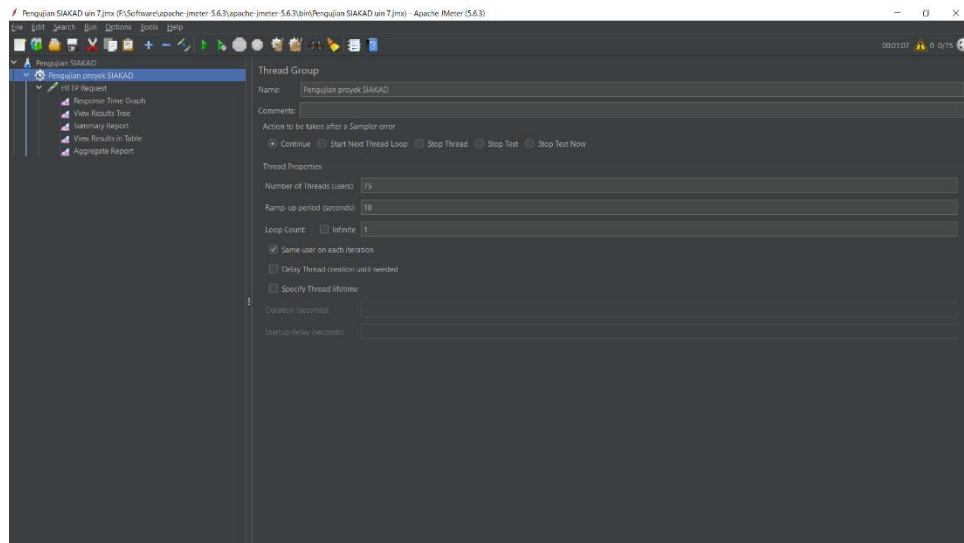


Gambar 4.12 Hasil Uji *Web* Proyek SIAKAD

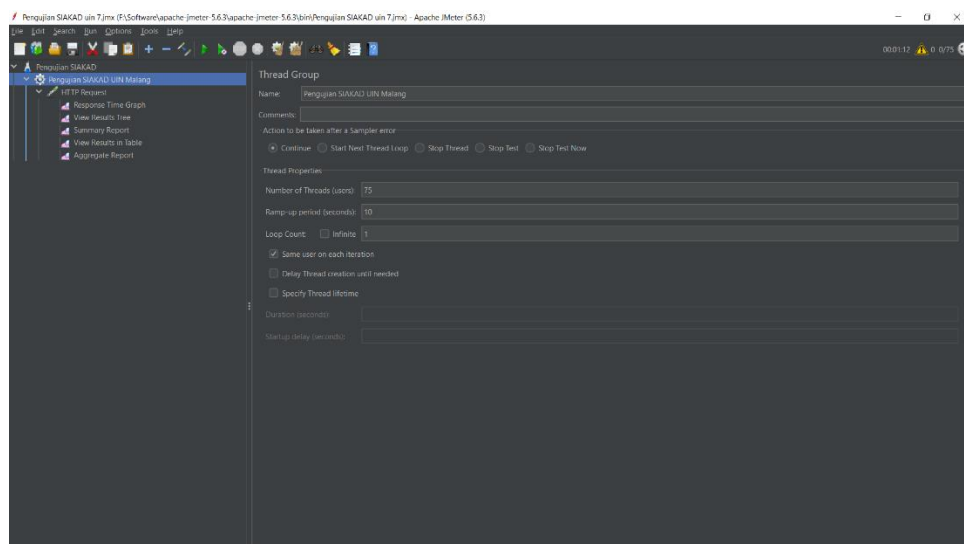
4.2 Hasil Uji Coba

Pengujian menggunakan *jmeter* dengan menggunakan beberapa macam sampel *thread* atau *request* yaitu *75 threads*, *100 threads*, dan *125 threads*, kemudian masing-masing *threads* diberi waktu sepuluh detik.

Berikut contoh konfigurasi jumlah *threads* atau *request* pada *Jmeter* yang akan ditampilkan pada gambar 4.13 dan 4.14.



Gambar 4.13 Konfigurasi *Threads Jmeter* Proyek SIAKAD



Gambar 4.14 Konfigurasi *Threads Jmeter* SIAKAD UIN Malang

Pengujian akan disajikan pada empat hasil dengan menggunakan elemen pendengar atau listener, diantaranya yaitu *summary report*, *aggregate report*, *view result in table*, dan *response time graph*.

4.2.1 Hasil *Summary Report*

Summary report adalah salah satu elemen pendengar (*listener*) yang digunakan untuk menganalisis hasil tes kinerja. *Listener* ini mewakili ringkasan data yang dikumpulkan selama pengujian dan memberikan gambaran umum kinerja aplikasi atau sistem yang diuji. Berikut adalah hasil dari uji *summary report* yang menampilkan berbagai metrik dan informasi pada *JMeter*. Hasil *summary report* proyek SIAKAD dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil *Summary Report* Proyek SIAKAD

Label	HTTP Request	HTTP Request	HTTP Request
# Samples	75	100	125
Average	43763	55029	67841
Min	17667	17416	16758
Max	57402	75329	87719
Std. Dev.	9.474	14.258	15.819
Error %	0.0	0.0	0.0
Throughput /sec	1,12	1,17	1,28
Received KB/s	10,05	10,57	11,54
Sent KB/s	0,38	0,402	0,44
Avg. Bytes	9228	9227,76	9227,57

Dapat dilihat pada tabel 4.1 dengan sampel sebanyak 75 dalam waktu sepuluh detik diperoleh nilai *average* 43763 ms, dan nilai *throughput* 1,12/sec. Pada sampel sebanyak 100 mendapatkan nilai *average* 55029 ms, dan nilai *throughput* 1,17/sec. Pada 125 sampel nilai *average* 67841 ms, dan nilai *throughput* 1,28/sec, dan mendapatkan nilai *error* sebesar 0 % pada semua percobaan.

Tabel 4.2 Hasil *Summary Report* SIAKAD UIN Malang

Label	HTTP Request	HTTP Request	HTTP Request
# Samples	75	100	125
Average	48977	64182	78580
Min	17821	19403	20775
Max	63137	83591	99418
Std. Dev.	11.029	14.969	16.757
Error %	0.0	0.0	0.0
Throughput /sec	1,04	1,07	1,15
Received KB/s	9,38	9,69	10,39
Sent KB/s	0,38	0,39	0,42
Avg. Bytes	9276	9276	9.275

Dapat dilihat pada tabel 4.2, yaitu tabel hasil *summary report* SIAKAD UIN Malang dengan sampel sebanyak 75 didapatkan nilai *average* sebesar 48977 ms, dan nilai *throughput* sebesar 1,04/sec. Pada sampel sebanyak 100 nilai *average* adalah 64182 ms, dan nilai *throughput* sebesar 1,07/sec. kemudian dengan sampel sebanyak 125 nilai *average* adalah 78580 ms, dan nilai *throughput* sebesar 1,15/sec, dan mendapatkan nilai *error* sebesar 0 % pada semua percobaan.

4.2.2 Hasil *Aggregate Report*

Aggregate report adalah salah satu elemen *listener* atau pendengar yang berfungsi untuk menngumpulkan dan menganalisis hasil pengujian kinerja dengan lebih mendetail. Berikut adalah hasil dari uji *aggregate report* yang dilakukan pada proyek SIAKAD dan SIAKAD UIN Malang.

Tabel 4.3 Hasil *Aggregate Report* Proyek SIAKAD

Label	HTTP Request	HTTP Request	HTTP Request
# Samples	75	100	125
Average	43763	55029	67841
Median	45198	56311	71434
90% Line	54659	71921	83559
95% Line	56404	73694	84951
99% Line	56439	74717	86371
Min	17667	17416	16758
Max	57402	75329	87719
Error %	0.0	0.0	0.0
Throughput /sec	1,12	1,17	1,28
Received KB/s	10,05	10,57	11,54
Sent KB/s	0,38	0,402	0,44

Dapat dilihat dari tabel 4.3 yaitu hasil uji *aggregate report* pada proyek SIAKAD dengan sampel sebanyak 75 nilai 90% *line* menunjukkan 54659, nilai 95% *line* adalah 56404, dan nilai 99% *line* adalah 56439. Kemudian pada sampel sebanyak 100 nilai 90% *line* menunjukkan 71921, nilai 95% *line* adalah 73694, dan nilai 99% *line* adalah 74717. Kemudian pada sampel sebanyak 125 nilai 90% *line* menunjukkan 83559, nilai 95% *line* adalah 84951, dan nilai 99% *line* adalah 86371. Untuk nilai *average* dan nilai *throughput* bernilai sama seperti hasil *summary report* proyek SIAKAD.

Tabel 4 4 Hasil *Aggregate Report* SIAKAD UIN Malang

Label	HTTP Request	HTTP Request	HTTP Request
# Samples	75	100	125
Average	48977	64182	78580
Median	50916	65855	81028
90% Line	60657	79982	95682
95% Line	61619	82472	97841
99% Line	62490	83463	99177
Min	17821	19403	20775
Max	63137	83591	99418
Error %	0.0	0.0	0.0
Throughput /sec	1,04	1,07	1,15
Received KB/s	9,38	9,69	10,39
Sent KB/s	0,38	0,39	0,42

Pada tabel 4.4 yaitu hasil uji *aggregate report* pada SIAKAD UIN Malang dapat diketahui bahwa dengan sampel sebanyak 75 nilai 90% *line* menunjukkan 60657, nilai 95% *line* adalah 61619, dan nilai 99% *line* adalah 62490. Kemudian pada sampel sebanyak 100 nilai 90% *line* menunjukkan 79982, nilai 95% *line* adalah 82472, dan nilai 99% *line* adalah 83463. Kemudian pada sampel sebanyak 125 nilai 90% *line* menunjukkan 81028, nilai 95% *line* adalah 97841, dan nilai 99% *line* adalah 99177. Untuk nilai *average* dan nilai *throughput* bernilai sama seperti hasil *summary report* SIAKAD UIN Malang.

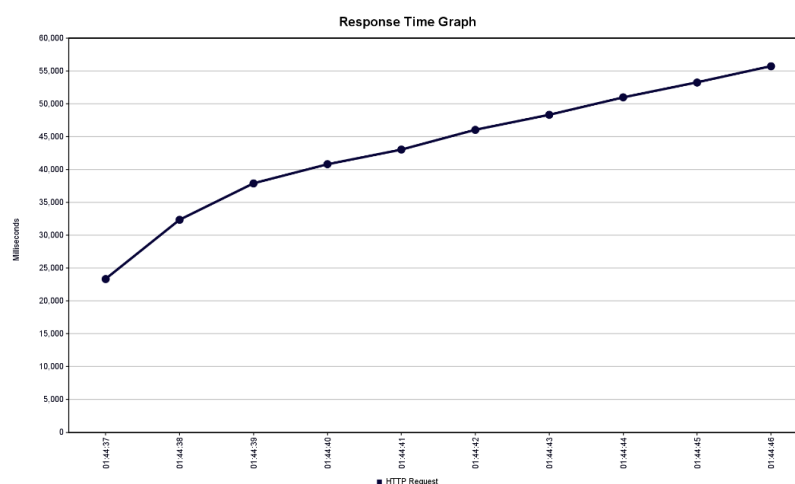
4.2.3 Hasil View Result in Table

Pada hasil yang ditampilkan pada *listener view result in table* yaitu berbentuk tabel yang didalamnya terdapat penjelasan secara rinci terkait sampel-sampel. Hasil dari *view result in table* akan di lampirkan pada halaman lampiran.

Hasil *view result in table* begitu penting karena didalamnya menjelaskan penjelasan secara rinci pada setiap sampel, dan pada hasil uji selanjutnya hasil *view result in table* digunakan untuk menggambarkan *response time graph* dengan memakai data pada kolom *start time* dan kolom *samples time (ms)*.

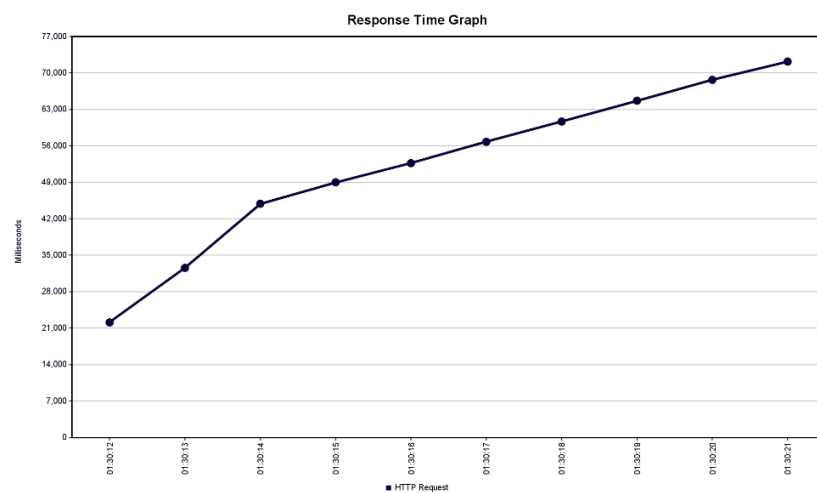
4.2.4 Hasil Response Time Graph

Pada hasil *response time graph* akan menampilkan grafik *response time* yang datanya diambil dari hasil *view result in table* pada kolom *start time* dan kolom *sample time (ms)*. Hasil yang ditampilkan dikonfigurasi menggunakan *interval* waktu 1000 ms atau sama dengan satu detik. Berikut hasil *response time graph* yang akan ditampilkan pada gambar 4.15



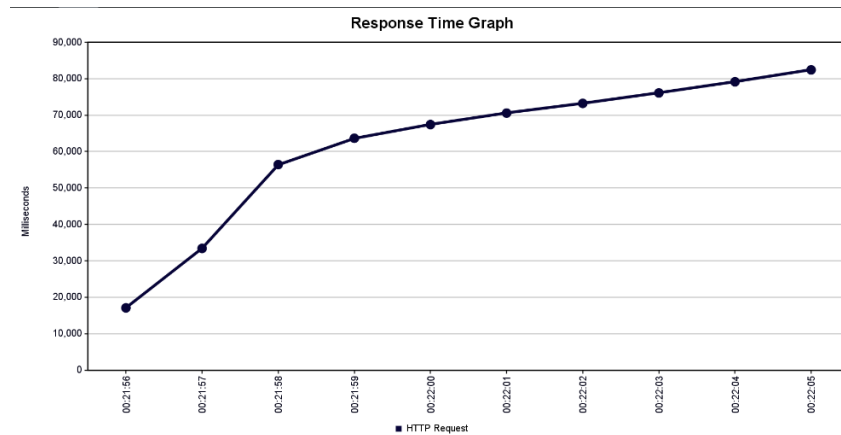
Gambar 4.15 Response Time Graph Proyek SIAKAD Thread 75

Pada gambar 4.15 menampilkan grafik *response time* proyek SIAKAD dengan jumlah *thread* sebanyak 75 yang bermula pada waktu 01.44.37 sampai dengan 01.44.47 dimana terdapat perbedaan waktu sekitar 10 detik dikarenakan *thread* di seting dengan waktu 10 detik, grafik semakin naik dikarenakan nilai *sample time* semakin bertambah.



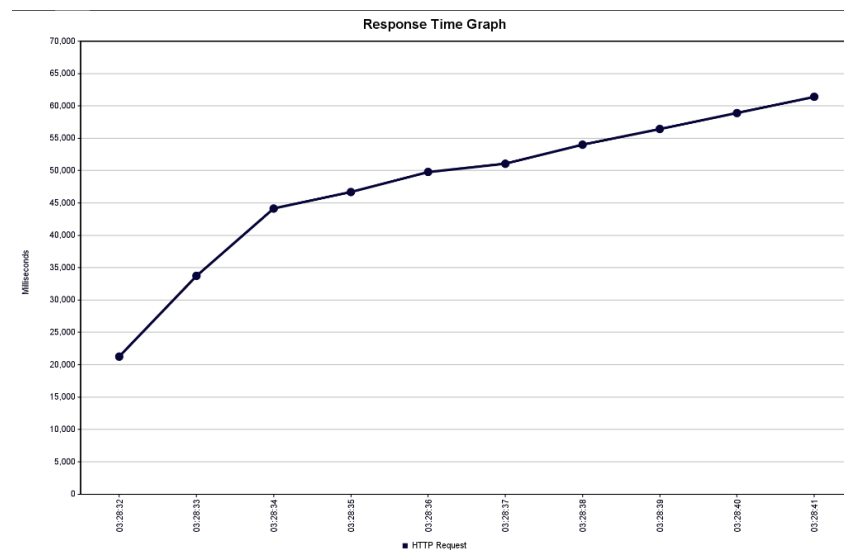
Gambar 4.16 *Response Time Graph* Proyek SIAKAD *Thread* 100

Pada gambar 4.16 menampilkan grafik *response time* proyek SIAKAD dengan jumlah *thread* 100 yang bermula pada waktu 01.30.12 sampai dengan 01.30.22 dimana terdapat perbedaan waktu sekitar 10 detik dikarenakan *thread* di seting dengan waktu 10 detik, grafik semakin naik dikarenakan nilai *sample time* semakin bertambah.



Gambar 4.17 *Response Time Graph* Proyek SIAKAD Thread 125

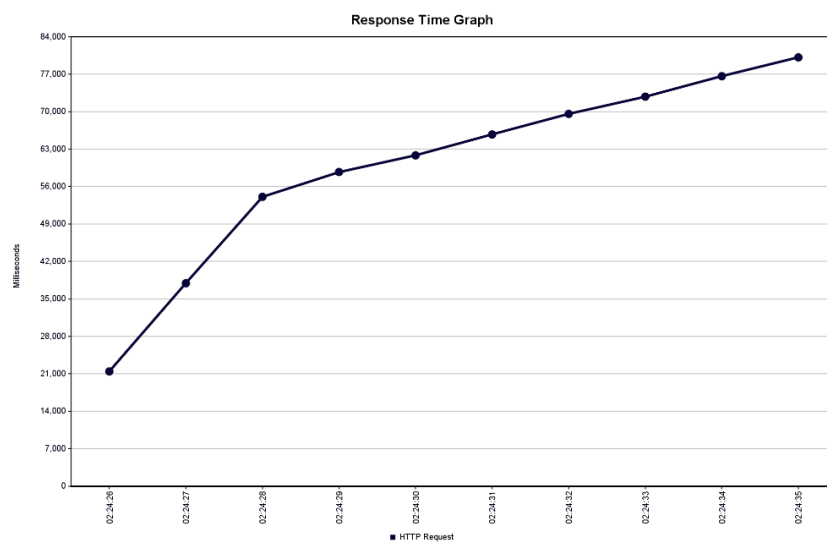
Pada gambar 4.17 menampilkan grafik *response time* proyek SIAKAD dengan jumlah *thread* sebanyak 125 yang bermula pada waktu 01.21.56 sampai dengan 01.22.06 dimana terdapat perbedaan waktu sekitar 10 detik dikarenakan *thread* di seting dengan waktu 10 detik, grafik semakin naik dikarenakan nilai *sample time* semakin bertambah.



Gambar 4.18 *Response Time Graph* SIAKAD UIN Malang Thread 75

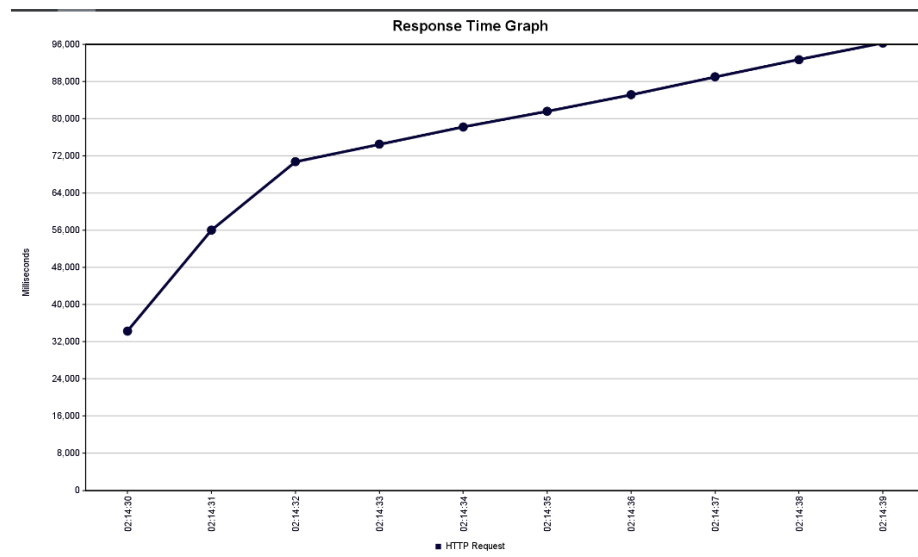
Pada gambar 4.18 menampilkan grafik *response time* SIAKAD UIN Malang dengan *thread* sebanyak 75 yang bermula pada waktu 03.28.32 sampai dengan

03.28.41 dimana terdapat perbedaan waktu sekitar 10 detik dikarenakan *thread* di seting dengan waktu 10 detik, grafik semakin naik dikarenakan nilai *sample time* semakin bertambah.



Gambar 4.19 *Response Time Graph* SIAKAD UIN Malang *Thread* 100

Pada gambar 4.19 menampilkan grafik *response time* SIAKAD UIN Malang dengan *thread* sebanyak 100 yang bermula pada waktu 02.24.26 sampai dengan 02.24.36 dimana terdapat perbedaan waktu sekitar 10 detik dikarenakan *thread* di seting dengan waktu 10 detik, grafik semakin naik dikarenakan nilai *sample time* semakin bertambah.



Gambar 4.20 *Response Time Graph* SIAKAD UIN Malang *Thread* 125

Pada gambar 4.20 menampilkan grafik *response time* SIAKAD UIN Malang dengan *thread* sebanyak 125 yang bermula pada waktu 02.14.30 sampai dengan 02.14.40 dimana terdapat perbedaan waktu sekitar 10 detik dikarenakan *thread* di seting dengan waktu 10 detik, grafik semakin naik dikarenakan nilai *sample time* semakin bertambah.

4.3 Pembahasan

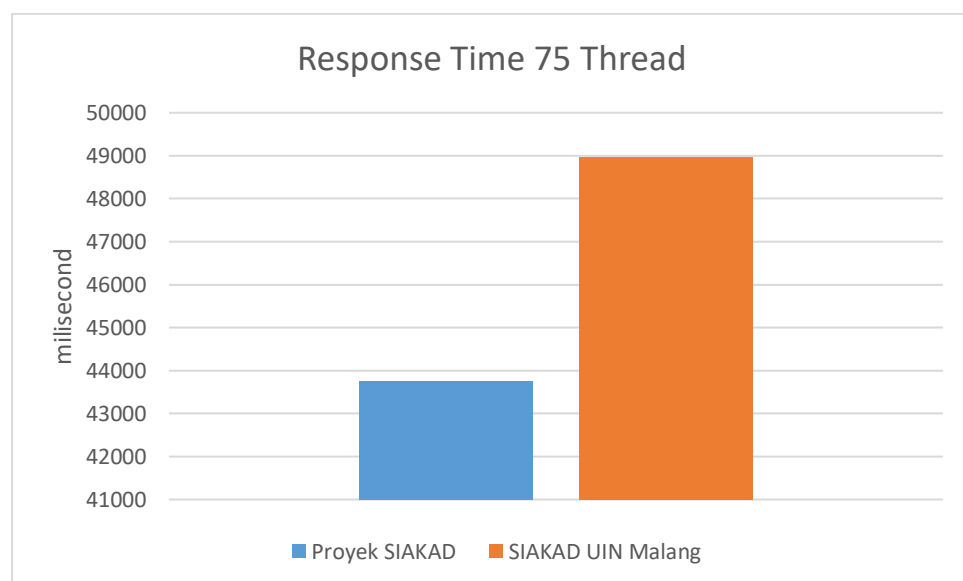
Pelayanan *server* memiliki banyak hal yang bisa memperlambat kinerja *server* kepada permintaan *client*, beberapa diantaranya adalah kecepatan penyimpanan, kecepatan *memory*, kecepatan *processor*, kecepatan jaringan internet. Pada penelitian ini perangkat yang dipakai sangatlah terbatas, sehingga hal yang mengakibatkan pelayanan *server* menjadi lambat tidak dapat dihindari.

Secara teori, prosesor *server* sering digunakan untuk memproses *php* dan kode lainnya melalui mesin *server web* seperti *Nginx*. Saat ini penyimpanan atau

memory pada *server* juga sering digunakan untuk memproses permintaan atau *query server database*. Dengan banyaknya kapasitas penyimpanan atau *memory* yang diperlukan dalam memproses *query* atau permintaan bergantung pada jumlah permintaan dalam satuan waktu dan kompleksitasnya.

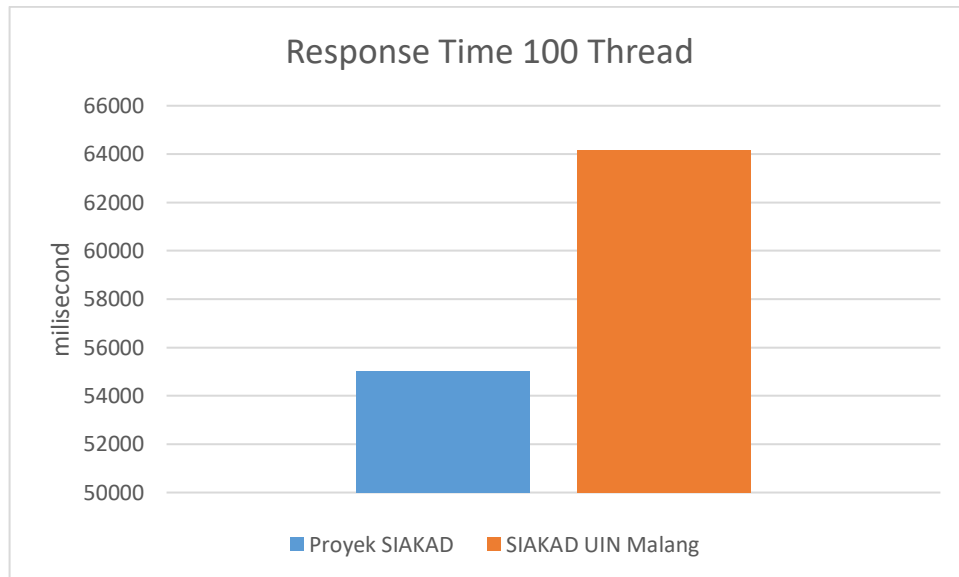
Summary report pada *JMeter* adalah salah satu elemen pendengar (*listener*) yang digunakan untuk menganalisis hasil tes kinerja. *Listener* ini mewakili ringkasan data yang dikumpulkan selama pengujian dan memberikan gambaran umum kinerja aplikasi atau sistem yang diuji.

Dari penjelasan hasil pengujian diatas, dapat kita rangkum hasil pengujian yang diantaranya pengujian di bedakan antara banyaknya jumlah *thread* atau *request* yaitu *75 thread*, *100 thread*, dan *125 thread*. Berikut adalah hasil dari pengujian menggunakan *Jmeter* untuk perhitungan nilai *response time* dan nilai *throughput* antara proyek SIAKAD dan SIAKAD UIN Malang dalam bentuk diagram.



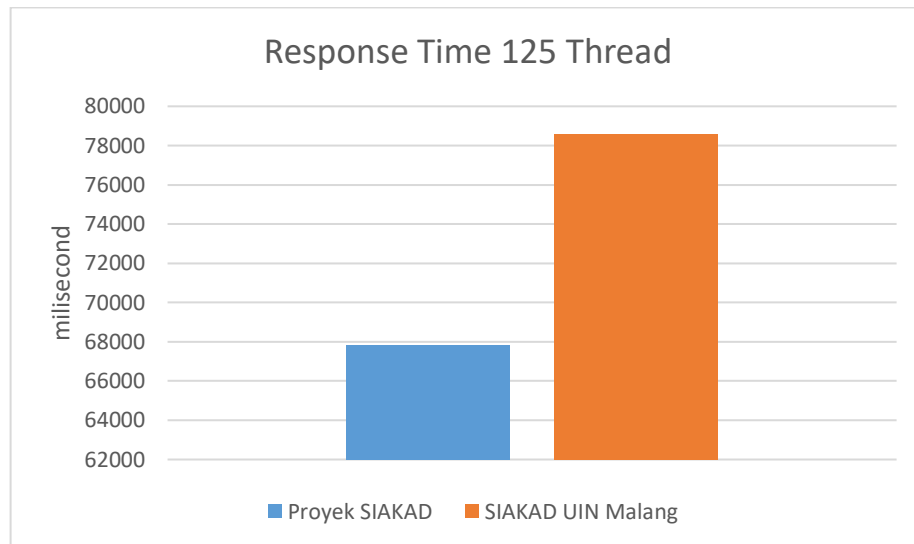
Gambar 4.21 Diagram *Response Time 75 Thread*

Pada gambar 4.21 dengan *thread* sebanyak 75 terlihat bahwa *server* proyek SIAKAD memiliki nilai *response time* lebih kecil yaitu 43763 ms dibanding nilai *response time* SIAKAD UIN Malang yaitu 48977 ms. Proyek SIAKAD lebih unggul dalam merespon *threads* atau *request*.



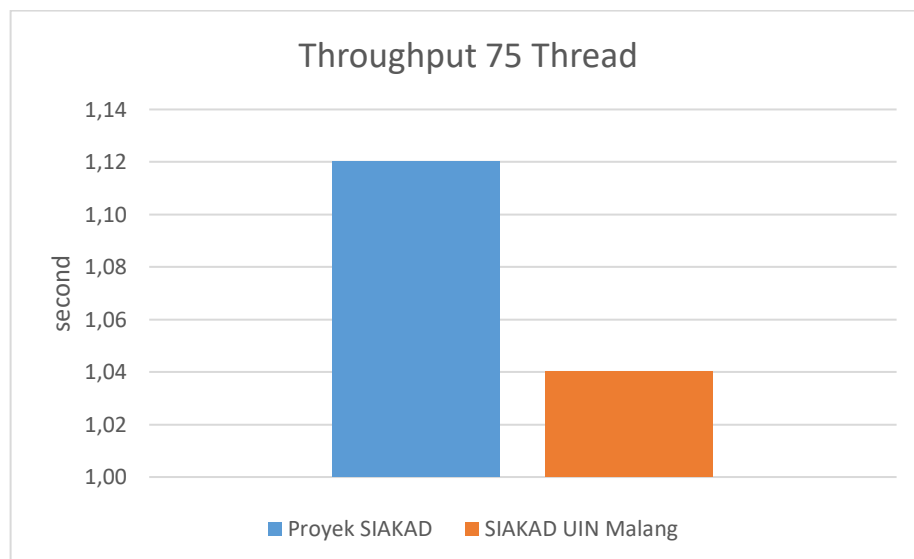
Gambar 4.22 Diagram *Response Time 100 Thread*

Pada gambar 4.22 dengan *thread* sebanyak 100 terlihat bahwa *server* proyek SIAKAD memiliki nilai *response time* lebih kecil yaitu 55029 ms dibanding nilai *response time* SIAKAD UIN Malang yaitu 64182 ms. Proyek SIAKAD lebih unggul dalam merespon *threads* atau *request*.



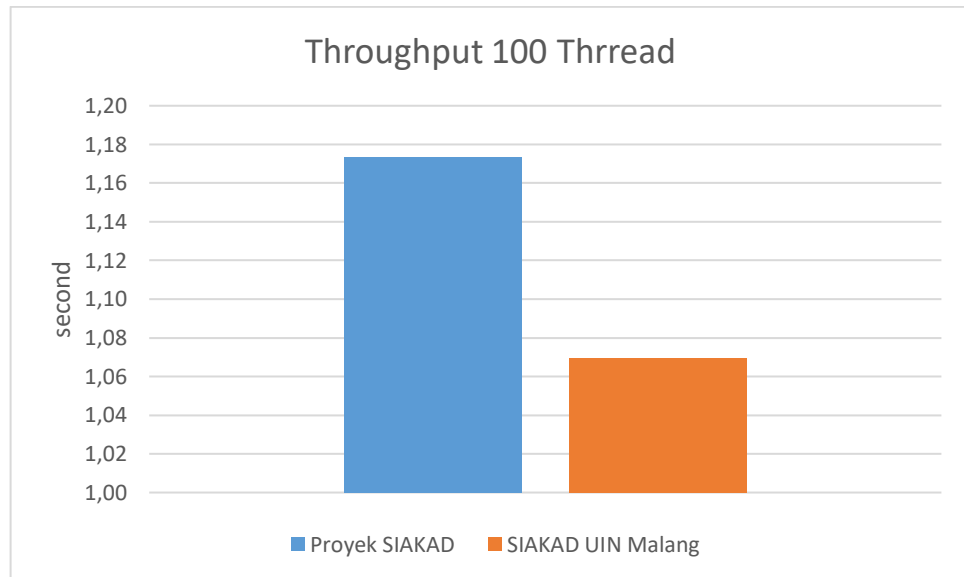
Gambar 4.23 Diagram *Response Time 125 Thread*

Pada gambar 4.23 dengan *thread* sebanyak 125 terlihat bahwa *server* proyek SIAKAD memiliki nilai *response time* lebih kecil yaitu 67841 ms dibanding nilai *response time* SIAKAD UIN Malang yaitu 78580 ms. Proyek SIAKAD lebih unggul dalam merespon *threads* atau *request*.



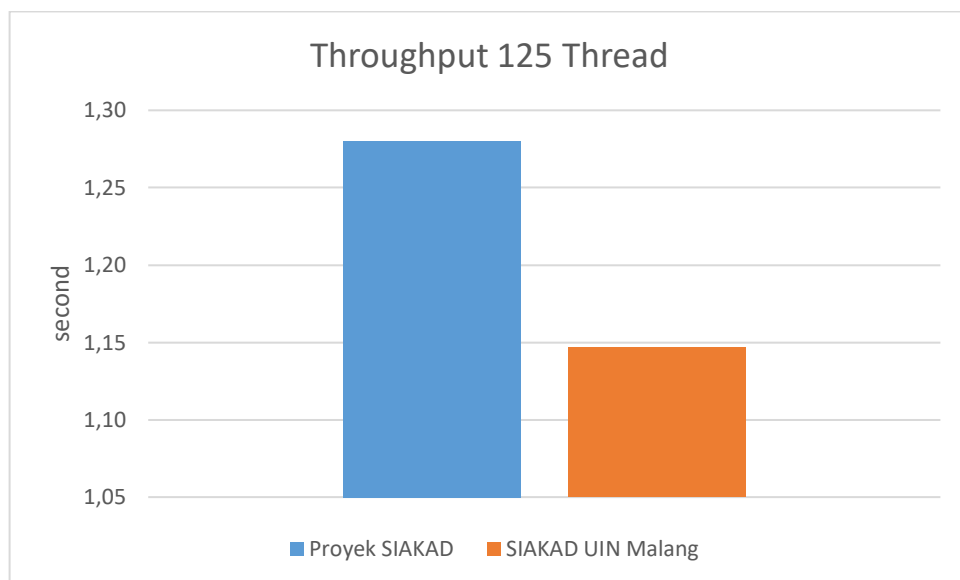
Gambar 4.24 Diagram *Throughput 75 Thread*

Pada gambar 4.24 nilai *throughput* pada *server* proyek SIAKAD lebih besar dari pada nilai *throughput server* SIAKAD UIN Malang, yaitu 1,12 *second* untuk proyek SIAKAD dan 1,04 *second* untuk SIAKAD UIN Malang, yang artinya proyek SIAKAD lebih unggul dalam memproses *throughput*.



Gambar 4.25 Diagram *Throughput 100 Thread*

Pada gambar 4.25 nilai *throughput* pada *server* proyek SIAKAD lebih besar dari pada nilai *throughput server* SIAKAD UIN Malang, yaitu 1,17 *second* untuk proyek SIAKAD dan 1,07 *second* untuk SIAKAD UIN Malang, yang artinya proyek SIAKAD lebih unggul dalam memproses *throughput*.



Gambar 4.26 Diagram *Throughput* 125 Thread

Pada gambar 4.26 nilai *throughput* pada *server* proyek SIAKAD lebih besar dari pada nilai *throughput server* SIAKAD UIN Malang, yaitu 1,28 *second* untuk proyek SIAKAD dan 1,15 *second* untuk SIAKAD UIN Malang, yang artinya proyek SIAKAD lebih unggul dalam memproses *throughput*.

4.4 Integrasi Sains dan Islam

Integrasi sains dan islam merupakan konsep untuk mengeksplorasi keterkaitan dan hubungan antara ilmu pengetahuan dengan ajaran-ajaran islam. Integrasi sains dengan islam berusaha agar ilmu pengetahuan modern dapat memiliki prinsip-prinsip islam yang bertujuan untuk menimbulkan harmoni antara pengetahuan dan keimanan.

Kitab suci Al-Qur'an memiliki banyak sekali ayat yang mendorong umat islam untuk merenungkan alam semesta dan seisinya supaya umat islam memiliki pengetahuan yang ada didalamnya. Ayat-ayat kitab suci Al-Qur'an sering kali

diintegrasikan sebagai dorongan untuk mencapai ilmu pengetahuan dan memahami fenomena alam sebagai tanda-tanda kekuasaan Allah Subhanahu wa Ta'ala.

Seperti yang difirmankan Allah Subhanahu wa Ta'ala dalam penggalan surat Al-Ma'idah ayat 2 sebagai berikut.

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ ۖ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ ۗ وَاتَّقُوا اللَّهَ ۖ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

“Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya.”(Q.S Al Ma'idah:2)

Qur'an Surat Al Ma'idah ayat dua dipaparkan dalam Tafsir Al-Munir oleh Syaikh Wahbah az-Zuhaili (Az-Zuhaili, 2016). Dalam segi *balaaghah*, ayat (وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ ۖ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ) susunan kalimat tersebut memiliki makna *al-muqaabalah*, yang memiliki makna didatangkanlah dua makna atau lebih pada awal kalimat lalu didatangkannya lagi kalimat berlawanan arti secara tertib pada bagian akhir kalimat. Dan tolong-menolonglah kamu dalam mengerjakan kebajikan dan takwa dan meninggalkan kemungkarannya.

Pada tafsir dan penjelasan potongan ayat tersebut Syaikh Wahbah az-Zuhaili memaparkan ayat tersebut tentang saling membahu menolong, dan saling bersinergi dalam menjalankan kebajikan. Jangan kamu saling tolong menolong untuk berbuat dosa dan maksiat, yaitu semua hal yang tidak diperbolehkan oleh *syari'at* yang membuat hati menjadi gusar terhadapnya dan sikap menyembunyikan agar orang lain tidak melihat dan mengetahuinya.

Dapat dipahami bahwa potongan ayat surat Al-Ma'idah ayat dua bisa diintegrasikan dengan penelitian ini bahwa sebagaimana arti tolong-menolong bisa disamakan dengan jumlah *server* yang tidak hanya satu, melainkan lebih dari satu. Ketika *server* lebih dari satu baru bisa dikatakan ada keterkaitan antara satu *server* dengan *server* yang lain. Pada penelitian ini terdapat tiga *server* yang diantaranya memiliki tugas secara spesifik agar beban yang dihadapi dapat dikerjakan dengan optimal.

Tolong menolong terhadap sesama manusia bukan hanya diperintahkan oleh Allah SWT saja, Nabi Muhammad SAW turut memerintahkan kepada umatnya untuk saling tolong menolong, sebagaimana dalam hadits Rasulullah SAW:

حَدَّثَنَا قُتَيْبَةُ حَدَّثَنَا اللَّيْثُ عَنْ عَقِيلٍ عَنِ الزُّهْرِيِّ عَنِ سَالِمٍ عَنْ أَبِيهِ أَنَّ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ
 الْمُسْلِمُ أَخُو الْمُسْلِمِ لَا يَظْلِمُهُ وَلَا يَسْلُمُهُ وَمَنْ كَانَ فِي حَاجَةِ أَخِيهِ كَانَ اللَّهُ فِي حَاجَتِهِ وَمَنْ فَرَّجَ عَنْ مُسْلِمٍ
 كُرْبَةً فَرَّجَ اللَّهُ عَنْهُ كُرْبَةً مِنْ كُرْبٍ يَوْمَ الْقِيَامَةِ وَمَنْ سَتَرَ مُسْلِمًا سَتَرَهُ اللَّهُ يَوْمَ الْقِيَامَةِ قَالَ أَبُو عِيسَى هَذَا
 حَدِيثٌ حَسَنٌ صَحِيحٌ غَرِيبٌ مِنْ حَدِيثِ ابْنِ عَمَرَ

"Telah menceritakan kepada kami Qutaibah, telah menceritakan kepada kami Al Laits dari 'Uqail dari Az Zuhri dari Salim dari ayahnya bahwa Rasulullah shallallahu 'alaihi wasallam bersabda: "Seorang muslim adalah saudara bagi muslim lainnya, tidak menzalimi dan tidak menganiyanya. Barangsiapa yang menolong kebutuhan saudaranya, maka Allah akan senantiasa menolongnya. Barangsiapa menghilangkan kesusahan seorang muslim maka Allah akan menghilangkan kesusahan-kesusahannya pada hari kiamat. Dan barangsiapa menutup aib seorang muslim, maka Allah akan menutup aibnya pada hari kiamat." Abu Isa berkata; Hadits ini hasan shahih gharib dari Hadits Ibnu Umar." (Hadits Tirmidzi Nomor 1346)

Sebagai seorang muslim hakikatnya adalah saudara bagi muslim yang lain. Seorang muslim tidak boleh saling menzalimi saudaranya, baik dengan ucapan, tindakan, dan sikap apapun yang merugikan. Membantu sesama muslim yang

membutuhkan merupakan sebuah amal atau perbuatan yang sangat dianjurkan. Allah SWT memberikan balasan dengan memenuhi kebutuhan orang yang membantu saudaranya. Pentingnya membantu untuk menghilangkan kesulitan yang dialami oleh sesama muslim sangatlah penting.

Dari penjelasan ayat dan hadits di atas peneliti juga menarik landasan integrasi sains untuk penelitian ini. Terdapat dua landasan yang pertama yaitu kemudahan dan yang kedua adalah kenyamanan untuk semua orang yang akan menggunakan manfaat dari penelitian ini.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian yang dibuat ini peneliti menggunakan tiga buah *node server* diantaranya yaitu *server manager*, *server worker 1*, *server worker 2* yang dikonfigurasi sehingga saling terhubung dengan menggunakan *cluster orchestration docker swarm*. Ketiga *server* juga saling terhubung pada direktori penyimpanan karena telah menggunakan *tools NFS server*, sehingga ketiga *server* memiliki data penyimpanan yang sama pada direktori *NFS*. Ketiga *server* mampu merespon berbagai jumlah *threads* dengan baik dengan jangka waktu 10 detik. Untuk melihat rangkuman hasil uji *response time* dapat dilihat pada tabel 5.1.

Tabel 5.1 Kesimpulan Nilai Hasil Uji Response Time

# Samples	SIAKAD UIN Malang	Proyek SIAKAD	Naik/Turun
75	48977	43763	-11%
100	64182	55029	-14%
125	78580	67841	-14%

Pada tabel 5.1 dapat dilihat nilai presentase penurunan hasil uji *response time*. Hasil uji *response time* mendapatkan nilai penurunan yang artinya optimasi *web server* menggunakan *cluster orchestration docker swarm* telah berhasil karena dalam memproses *thread* atau *request*, *server* mampu memproses dengan cepat.

Tabel 5.2 Kesimpulan Nilai Hasil Uji *Throughput*

# Samples	SIAKAD UIN Malang	Proyek SIAKAD	Naik/Turun
75	1,04	1,12	8%
100	1,07	1,17	9%
125	1,15	1,28	11%

Begitu juga dengan nilai presentase dari hasil *throughput* pada tabel 5.2 mendapatkan nilai kenaikan, yang artinya *server* mampu memproses *thread* atau *request* lebih banyak, sehingga *server* menjadi lebih optimal.

Penggunaan *docker* juga dapat membantu pembuatan aplikasi yang akan mempermudah dikarenakan pada *docker* sendiri memiliki beberapa keunggulan utama yaitu penggunaan sumber daya yang efisien sehingga membuat *server* menjadi ringan, ekosistem yang kaya dan luas yang dapat dicari pada web *docker hub*, yang menyediakan berbagai *images* siap pakai, penggandaan *services* yang bertujuan untuk membagi dan memperingan *server*, dan masih banyak lagi.

5.2 Saran

Berdasarkan penelitian diatas, peneliti memiliki beberapa saran untuk penelitian selanjutnya yaitu ketiga *server* mampu menangani *request* dengan jumlah yang tidak banyak dikarenakan terbatasnya sumber daya perangkat. Alangkah baiknya agar dilakukan pada perangkat yang lebih tinggi atau dapat menggunakan *virtual server* yang disediakan online seperti *AWS*, *Microsoft Azure*, dan *GCP*.

DAFTAR PUSTAKA

- Afis, D. S. (2018). *Load Balancing Server Web Berdasarkan Jumlah Koneksi Klien Pada Docker Swarm*.
- Apriliansyah, F., Fitri, I., Iskandar, A., & Artikel, R. (2020). Jurnal Teknologi dan Manajemen Informatika Implementasi Load Balancing Pada Web Server Menggunakan Nginx Info Artikel ABSTRAK. *Jurnal Teknologi Dan Manajemen Informatika*, 6(1), 18–26.
<http://http://jurnal.unmer.ac.id/index.php/jtmi>
- Az-Zuhaili, W. (2016). *Tafsir Munir: Akidah-Syariah-Manhaj Jilid 3* (1st ed.). Gema Insani. <http://www.gemainsani.co.id>
- Chandra, A. Y. (2019). Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request. *Jurnal Sistem Dan Informatika (JSI)*, 14(1), 48–56. <https://doi.org/10.30864/jsi.v14i1.248>
- Dwiyatno, S., Rachmat, E., Sari, A. P., & Gustiawan, O. (2020). Implementasi Virtualisasi Server Berbasis Docker Container. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 7(2), 165–175.
<https://doi.org/10.30656/prosisko.v7i2.2520>
- Ekowansyah, E., Chrisnanto, Y. H., Puspita, & Sabrina, N. (2017). Audit sistem informasi akademik menggunakan COBIT 5 di Universitas Jenderal Achmad Yani. *Prosiding Seminar Nasional Komputer Dan Informatika (SENASKI), 2017*, 201–205.
- Fandy, Rosmasari, & Putra, G. M. (2022). Pengujian Kinerja Web Server Atas Penyedia Layanan Elastic Cloud Compute (EC2) Pada Amazon Web Services (AWS). *Adopsi Teknologi dan Sistem Informasi (ATASI)*, 1(1), 21–35. <https://doi.org/10.30872/atasi.v1i1.45>
- Kane, P. S., & Matthias, K. (2018). *Docker Up Running Shipping Reliable Containers in Production* (2nd ed.). O'Reilly Media.
- McKendrick, R. (2020). *Mastering Docker - Fourth Edition* (4th ed.). Packt Publishing Ltd.
- Oracle. (2010). Retrieved from <https://docs.oracle.com/cd/E19879-01/8204342/abfch/index.html>. December, 1–80.
- Potdar, A. M., Narayan, D. G., Kengond, S., & Mulla, M. M. (2020). Performance Evaluation of Docker Container and Virtual Machine. *Procedia Computer Science*, 171(2019), 1419–1428. <https://doi.org/10.1016/j.procs.2020.04.152>

- Putra, M. A. A., Fitri, I., & Iskandar, A. (2020). Implementasi High Availability Cluster Web Server Menggunakan Virtualisasi Container Docker. *Jurnal Media Informatika Budidarma*, 4(1), 9. <https://doi.org/10.30865/mib.v4i1.1729>
- Rajeswari, M. N. (2019). Overview of Cloud Computing. *SSRN Electronic Journal*, 6(3), 61–67.
- Rexa, M., Data, M., & Yahya, W. (2019). Implementasi Load Balancing Server Web Berbasis Docker Swarm Berdasarkan Penggunaan Sumber Daya Memory Host. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 3(4), 3478–3487.
- Rodriguez, M. A., & Buyya, R. (2018). Container-based cluster orchestration systems: A taxonomy and future directions. *Software - Practice and Experience*, 2, 1–29. <https://doi.org/10.1002/spe.2660>
- Saraswat, M., & Tripathi, R. C. (2020). Cloud Computing: Comparison and Analysis of Cloud Service Providers-AWs, Microsoft and Google. *Proceedings of the 2020 9th International Conference on System Modeling and Advancement in Research Trends, SMART 2020*, 1, 281–285. <https://doi.org/10.1109/SMART50582.2020.9337100>
- Sholeh, M., Yahya, W., & Hari, P. (2019). Implementasi Load Balancing menggunakan Algoritme Least Connection dengan Agen Psutils pada Web Server. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 3(1), 842–851.
- Triangga, H., Faisal, I., & Lubis, I. (2019). Analisis Perbandingan Algoritma Static Round-Robin dengan Least-Connection Terhadap Efisiensi Load Balancing pada Load Balancer Haproxy. *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, 4(1), 70–75. <https://doi.org/10.30743/infotekjar.v4i1.1688>
- Zhou, N., Georgiou, Y., Pospieszny, M., Zhong, L., Zhou, H., Niethammer, C., Pejak, B., Marko, O., & Hoppe, D. (2021). Container orchestration on HPC systems through Kubernetes. *Journal of Cloud Computing*, 10(1). <https://doi.org/10.1186/s13677-021-00231-z>

LAMPIRAN

LAMPIRAN 1 Data View Result in Table Proyek SIAKAD 75 Thread

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
1	01:44:37.226	Pengujian proyek SIAKAD 1-1	HTTP Request	17667	Success	9228	351	3934	2
2	01:44:37.358	Pengujian proyek SIAKAD 1-2	HTTP Request	18278	Success	9228	351	3878	1
3	01:44:37.490	Pengujian proyek SIAKAD 1-3	HTTP Request	22016	Success	9228	351	5128	1
4	01:44:37.891	Pengujian proyek SIAKAD 1-6	HTTP Request	24214	Success	9228	351	5394	1
5	01:44:38.023	Pengujian proyek SIAKAD 1-7	HTTP Request	24360	Success	9228	351	5331	1
6	01:44:37.623	Pengujian proyek SIAKAD 1-4	HTTP Request	27042	Success	9228	351	6373	1
7	01:44:38.156	Pengujian proyek SIAKAD 1-8	HTTP Request	29872	Success	9228	351	6595	0
8	01:44:37.756	Pengujian proyek SIAKAD 1-5	HTTP Request	30768	Success	9228	351	7122	1
9	01:44:38.288	Pengujian proyek SIAKAD 1-9	HTTP Request	31347	Success	9228	351	7016	1
10	01:44:38.421	Pengujian proyek SIAKAD 1-10	HTTP Request	31215	Success	9228	351	6956	1
11	01:44:38.554	Pengujian proyek SIAKAD 1-11	HTTP Request	34175	Success	9228	351	7627	1
12	01:44:38.695	Pengujian proyek SIAKAD 1-12	HTTP Request	35369	Success	9228	351	8147	1
13	01:44:38.820	Pengujian proyek SIAKAD 1-13	HTTP Request	36109	Success	9228	351	8177	1
14	01:44:38.952	Pengujian proyek SIAKAD 1-14	HTTP Request	36366	Success	9228	351	8351	1
15	01:44:39.086	Pengujian proyek SIAKAD 1-15	HTTP Request	36993	Success	9228	351	8277	1
16	01:44:39.2	Pengujian proyek	HTTP Request	37025	Success	9228	351	9059	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	20	SIAKAD 1-16							
17	01:44:39.352	Pengujian proyek SIAKAD 1-17	HTTP Request	37883	Success	9228	351	9546	1
18	01:44:39.485	Pengujian proyek SIAKAD 1-18	HTTP Request	37959	Success	9228	351	9544	1
19	01:44:39.618	Pengujian proyek SIAKAD 1-19	HTTP Request	38049	Success	9228	351	9611	1
20	01:44:39.750	Pengujian proyek SIAKAD 1-20	HTTP Request	38670	Success	9228	351	9596	2
21	01:44:39.884	Pengujian proyek SIAKAD 1-21	HTTP Request	38647	Success	9228	351	10430	1
22	01:44:40.017	Pengujian proyek SIAKAD 1-22	HTTP Request	39503	Success	9228	351	10986	1
23	01:44:40.150	Pengujian proyek SIAKAD 1-23	HTTP Request	39582	Success	9228	351	11023	1
24	01:44:40.283	Pengujian proyek SIAKAD 1-24	HTTP Request	39632	Success	9228	351	11008	1
25	01:44:40.417	Pengujian proyek SIAKAD 1-25	HTTP Request	40201	Success	9228	351	11004	1
26	01:44:40.550	Pengujian proyek SIAKAD 1-26	HTTP Request	40125	Success	9228	351	11867	1
27	01:44:40.682	Pengujian proyek SIAKAD 1-27	HTTP Request	41107	Success	9228	351	12501	1
28	01:44:40.815	Pengujian proyek SIAKAD 1-28	HTTP Request	41304	Success	9228	351	12480	1
29	01:44:41.082	Pengujian proyek SIAKAD 1-30	HTTP Request	41885	Success	9228	351	12501	1
30	01:44:41.215	Pengujian proyek SIAKAD 1-31	HTTP Request	41819	Success	9228	351	14000	1
31	01:44:41.347	Pengujian proyek SIAKAD 1-32	HTTP Request	42878	Success	9228	351	14073	1
32	01:44:41.480	Pengujian proyek SIAKAD 1-33	HTTP Request	43046	Success	9228	351	14191	1
33	01:44:41.6	Pengujian proyek	HTTP Request	43523	Success	9228	351	15377	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	15	SIAKAD 1-34							
34	01:44:41.745	Pengujian proyek SIAKAD 1-35	HTTP Request	43515	Success	9228	351	15615	1
35	01:44:40.948	Pengujian proyek SIAKAD 1-29	HTTP Request	45115	Success	9228	351	12443	2
36	01:44:41.880	Pengujian proyek SIAKAD 1-36	HTTP Request	44623	Success	9228	351	15646	1
37	01:44:42.015	Pengujian proyek SIAKAD 1-37	HTTP Request	44800	Success	9228	351	15547	1
38	01:44:42.149	Pengujian proyek SIAKAD 1-38	HTTP Request	45198	Success	9228	351	15473	1
39	01:44:42.281	Pengujian proyek SIAKAD 1-39	HTTP Request	45207	Success	9228	351	16809	1
40	01:44:42.548	Pengujian proyek SIAKAD 1-41	HTTP Request	45809	Success	9228	351	16686	2
41	01:44:42.413	Pengujian proyek SIAKAD 1-40	HTTP Request	46392	Success	9228	351	16821	1
42	01:44:42.679	Pengujian proyek SIAKAD 1-42	HTTP Request	46421	Success	9228	351	17009	1
43	01:44:42.810	Pengujian proyek SIAKAD 1-43	HTTP Request	46741	Success	9228	351	17802	1
44	01:44:43.078	Pengujian proyek SIAKAD 1-45	HTTP Request	46628	Success	9228	351	17534	1
45	01:44:42.944	Pengujian proyek SIAKAD 1-44	HTTP Request	47701	Success	9228	351	17670	1
46	01:44:43.209	Pengujian proyek SIAKAD 1-46	HTTP Request	47903	Success	9228	351	18383	1
47	01:44:43.343	Pengujian proyek SIAKAD 1-47	HTTP Request	48087	Success	9228	351	18808	0
48	01:44:43.476	Pengujian proyek SIAKAD 1-48	HTTP Request	48305	Success	9228	351	19273	0
49	01:44:43.609	Pengujian proyek SIAKAD 1-49	HTTP Request	48334	Success	9228	351	20124	1
50	01:44:43.7	Pengujian proyek	HTTP Request	49360	Success	9228	351	20364	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	41	SIAKAD 1-50							
51	01:44:43.873	Pengujian proyek SIAKAD 1-51	HTTP Request	49707	Success	9228	351	20283	0
52	01:44:44.006	Pengujian proyek SIAKAD 1-52	HTTP Request	49852	Success	9228	351	20740	1
53	01:44:44.139	Pengujian proyek SIAKAD 1-53	HTTP Request	50027	Success	9228	351	21590	1
54	01:44:44.282	Pengujian proyek SIAKAD 1-54	HTTP Request	50055	Success	9228	351	21803	1
55	01:44:44.404	Pengujian proyek SIAKAD 1-55	HTTP Request	50934	Success	9228	351	21778	1
56	01:44:44.538	Pengujian proyek SIAKAD 1-56	HTTP Request	51318	Success	9228	351	22179	1
57	01:44:44.672	Pengujian proyek SIAKAD 1-57	HTTP Request	51465	Success	9228	351	22074	1
58	01:44:44.804	Pengujian proyek SIAKAD 1-58	HTTP Request	51604	Success	9228	351	23259	1
59	01:44:45.069	Pengujian proyek SIAKAD 1-60	HTTP Request	51504	Success	9228	351	23104	1
60	01:44:44.937	Pengujian proyek SIAKAD 1-59	HTTP Request	52678	Success	9228	351	23235	1
61	01:44:45.347	Pengujian proyek SIAKAD 1-62	HTTP Request	52816	Success	9228	351	24176	1
62	01:44:45.203	Pengujian proyek SIAKAD 1-61	HTTP Request	53206	Success	9228	351	24320	1
63	01:44:45.468	Pengujian proyek SIAKAD 1-63	HTTP Request	53189	Success	9228	351	25025	1
64	01:44:45.734	Pengujian proyek SIAKAD 1-65	HTTP Request	53058	Success	9228	351	25115	1
65	01:44:45.601	Pengujian proyek SIAKAD 1-64	HTTP Request	54303	Success	9228	351	25249	1
66	01:44:45.868	Pengujian proyek SIAKAD 1-66	HTTP Request	54622	Success	9228	351	25098	1
67	01:44:46.0	Pengujian proyek	HTTP Request	54697	Success	9228	351	24965	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	01	SIAKAD 1-67							
68	01:44:46.267	Pengujian proyek SIAKAD 1-69	HTTP Request	54659	Success	9228	351	26027	1
69	01:44:46.133	Pengujian proyek SIAKAD 1-68	HTTP Request	54888	Success	9228	351	26163	1
70	01:44:46.399	Pengujian proyek SIAKAD 1-70	HTTP Request	55807	Success	9228	351	26020	1
71	01:44:46.532	Pengujian proyek SIAKAD 1-71	HTTP Request	56404	Success	9228	351	25888	1
72	01:44:46.666	Pengujian proyek SIAKAD 1-72	HTTP Request	56439	Success	9228	351	27036	1
73	01:44:46.799	Pengujian proyek SIAKAD 1-73	HTTP Request	56434	Success	9228	351	26904	0
74	01:44:46.930	Pengujian proyek SIAKAD 1-74	HTTP Request	56421	Success	9228	351	26951	1
75	01:44:47.063	Pengujian proyek SIAKAD 1-75	HTTP Request	57402	Success	9228	351	28076	1

LAMPIRAN 2 Data View Result in Table Proyek SIAKAD 100 Thread

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
1	01:30:12.653	Pengujian proyek SIAKAD 1-2	HTTP Request	17416	Success	9228	351	3741	1
2	01:30:12.554	Pengujian proyek SIAKAD 1-1	HTTP Request	18318	Success	9228	351	3841	1
3	01:30:12.763	Pengujian proyek SIAKAD 1-3	HTTP Request	22376	Success	9228	351	4486	0
4	01:30:12.854	Pengujian proyek SIAKAD 1-4	HTTP Request	23863	Success	9228	351	4609	1
5	01:30:13.153	Pengujian proyek SIAKAD 1-7	HTTP Request	25285	Success	9228	351	4658	1
6	01:30:13.053	Pengujian proyek SIAKAD 1-6	HTTP Request	25934	Success	9228	351	4758	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
7	01:30:12.954	Pengujian proyek SIAKAD 1-5	HTTP Request	28485	Success	9228	351	5473	1
8	01:30:13.553	Pengujian proyek SIAKAD 1-11	HTTP Request	30854	Success	9228	351	5685	1
9	01:30:13.353	Pengujian proyek SIAKAD 1-9	HTTP Request	31625	Success	9228	351	5885	1
10	01:30:13.252	Pengujian proyek SIAKAD 1-8	HTTP Request	31727	Success	9228	351	5990	1
11	01:30:13.453	Pengujian proyek SIAKAD 1-10	HTTP Request	31838	Success	9228	351	5785	1
12	01:30:13.754	Pengujian proyek SIAKAD 1-13	HTTP Request	36927	Success	9222	351	6870	0
13	01:30:13.953	Pengujian proyek SIAKAD 1-15	HTTP Request	36730	Success	9222	351	6671	1
14	01:30:13.653	Pengujian proyek SIAKAD 1-12	HTTP Request	37034	Success	9222	351	6897	0
15	01:30:13.852	Pengujian proyek SIAKAD 1-14	HTTP Request	37623	Success	9228	351	6777	1
16	01:30:14.053	Pengujian proyek SIAKAD 1-16	HTTP Request	38342	Success	9228	351	6665	1
17	01:30:14.153	Pengujian proyek SIAKAD 1-17	HTTP Request	43453	Success	9228	351	7980	1
18	01:30:14.252	Pengujian proyek SIAKAD 1-18	HTTP Request	43540	Success	9228	351	7883	1
19	01:30:14.352	Pengujian proyek SIAKAD 1-19	HTTP Request	44880	Success	9228	351	8248	1
20	01:30:14.552	Pengujian proyek SIAKAD 1-21	HTTP Request	45342	Success	9228	351	8368	1
21	01:30:14.753	Pengujian proyek SIAKAD 1-23	HTTP Request	45306	Success	9228	351	8805	1
22	01:30:14.654	Pengujian proyek SIAKAD 1-22	HTTP Request	45721	Success	9228	351	8906	1
23	01:30:14.853	Pengujian proyek SIAKAD 1-24	HTTP Request	46528	Success	9228	351	9863	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
24	01:30:15.052	Pengujian proyek SIAKAD 1-26	HTTP Request	46522	Success	9228	351	9899	1
25	01:30:14.952	Pengujian proyek SIAKAD 1-25	HTTP Request	47392	Success	9228	351	9999	1
26	01:30:15.155	Pengujian proyek SIAKAD 1-27	HTTP Request	47319	Success	9228	351	9802	0
27	01:30:14.452	Pengujian proyek SIAKAD 1-20	HTTP Request	48356	Success	9228	351	11368	1
28	01:30:15.453	Pengujian proyek SIAKAD 1-30	HTTP Request	48305	Success	9228	351	10898	1
29	01:30:15.254	Pengujian proyek SIAKAD 1-28	HTTP Request	48689	Success	9228	351	11097	1
30	01:30:15.355	Pengujian proyek SIAKAD 1-29	HTTP Request	49215	Success	9228	351	10997	1
31	01:30:15.553	Pengujian proyek SIAKAD 1-31	HTTP Request	49140	Success	9228	351	11484	1
32	01:30:15.857	Pengujian proyek SIAKAD 1-34	HTTP Request	49192	Success	9228	351	11823	1
33	01:30:15.965	Pengujian proyek SIAKAD 1-35	HTTP Request	50065	Success	9228	351	11716	1
34	01:30:15.657	Pengujian proyek SIAKAD 1-32	HTTP Request	50557	Success	9228	351	12024	1
35	01:30:15.764	Pengujian proyek SIAKAD 1-33	HTTP Request	51033	Success	9228	351	11925	0
36	01:30:16.351	Pengujian proyek SIAKAD 1-39	HTTP Request	50548	Success	9228	351	12684	1
37	01:30:16.057	Pengujian proyek SIAKAD 1-36	HTTP Request	51294	Success	9228	351	12978	0
38	01:30:16.257	Pengujian proyek SIAKAD 1-38	HTTP Request	52048	Success	9228	351	12778	1
39	01:30:16.166	Pengujian proyek SIAKAD 1-37	HTTP Request	52325	Success	9228	351	12876	1
40	01:30:16.451	Pengujian proyek SIAKAD 1-40	HTTP Request	52612	Success	9228	351	13967	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
41	01:30:16.652	Pengujian proyek SIAKAD 1-42	HTTP Request	52525	Success	9228	351	14448	1
42	01:30:16.752	Pengujian proyek SIAKAD 1-43	HTTP Request	52895	Success	9228	351	15547	1
43	01:30:16.851	Pengujian proyek SIAKAD 1-44	HTTP Request	53737	Success	9228	351	15829	1
44	01:30:16.951	Pengujian proyek SIAKAD 1-45	HTTP Request	53834	Success	9228	351	16122	1
45	01:30:17.053	Pengujian proyek SIAKAD 1-46	HTTP Request	54239	Success	9228	351	16254	0
46	01:30:16.552	Pengujian proyek SIAKAD 1-41	HTTP Request	54895	Success	9228	351	17251	1
47	01:30:17.152	Pengujian proyek SIAKAD 1-47	HTTP Request	54879	Success	9228	351	17255	0
48	01:30:17.351	Pengujian proyek SIAKAD 1-49	HTTP Request	55915	Success	9228	351	18048	1
49	01:30:17.452	Pengujian proyek SIAKAD 1-50	HTTP Request	56274	Success	9228	351	18456	1
50	01:30:17.551	Pengujian proyek SIAKAD 1-51	HTTP Request	56311	Success	9228	351	19541	1
51	01:30:17.655	Pengujian proyek SIAKAD 1-52	HTTP Request	56753	Success	9228	351	19457	1
52	01:30:17.752	Pengujian proyek SIAKAD 1-53	HTTP Request	57822	Success	9228	351	19775	1
53	01:30:17.851	Pengujian proyek SIAKAD 1-54	HTTP Request	58150	Success	9228	351	21157	1
54	01:30:17.951	Pengujian proyek SIAKAD 1-55	HTTP Request	58195	Success	9228	351	21120	1
55	01:30:17.251	Pengujian proyek SIAKAD 1-48	HTTP Request	59190	Success	9228	351	17920	1
56	01:30:18.051	Pengujian proyek SIAKAD 1-56	HTTP Request	58631	Success	9228	351	21646	2
57	01:30:18.152	Pengujian proyek SIAKAD 1-57	HTTP Request	59711	Success	9228	351	22428	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
58	01:30:18.351	Pengujian proyek SIAKAD 1-59	HTTP Request	59901	Success	9228	351	22388	1
59	01:30:18.252	Pengujian proyek SIAKAD 1-58	HTTP Request	60178	Success	9228	351	22488	2
60	01:30:18.451	Pengujian proyek SIAKAD 1-60	HTTP Request	60320	Success	9228	351	23330	0
61	01:30:18.653	Pengujian proyek SIAKAD 1-62	HTTP Request	60342	Success	9228	351	23518	1
62	01:30:18.551	Pengujian proyek SIAKAD 1-61	HTTP Request	61581	Success	9228	351	23620	0
63	01:30:18.752	Pengujian proyek SIAKAD 1-63	HTTP Request	61810	Success	9228	351	23427	1
64	01:30:18.853	Pengujian proyek SIAKAD 1-64	HTTP Request	61867	Success	9228	351	24642	1
65	01:30:19.052	Pengujian proyek SIAKAD 1-66	HTTP Request	61979	Success	9228	351	24484	1
66	01:30:18.952	Pengujian proyek SIAKAD 1-65	HTTP Request	62277	Success	9228	351	24585	1
67	01:30:19.152	Pengujian proyek SIAKAD 1-67	HTTP Request	63299	Success	9228	351	24415	1
68	01:30:19.552	Pengujian proyek SIAKAD 1-71	HTTP Request	63531	Success	9228	351	26857	1
69	01:30:19.351	Pengujian proyek SIAKAD 1-69	HTTP Request	64037	Success	9228	351	27059	1
70	01:30:19.451	Pengujian proyek SIAKAD 1-70	HTTP Request	64108	Success	9228	351	26992	1
71	01:30:19.654	Pengujian proyek SIAKAD 1-72	HTTP Request	65155	Success	9228	351	27732	1
72	01:30:19.752	Pengujian proyek SIAKAD 1-73	HTTP Request	65663	Success	9228	351	27898	1
73	01:30:19.953	Pengujian proyek SIAKAD 1-75	HTTP Request	65724	Success	9228	351	27849	0
74	01:30:19.852	Pengujian proyek SIAKAD 1-74	HTTP Request	65996	Success	9228	351	27951	0

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
75	01:30:19.251	Pengujian proyek SIAKAD 1-68	HTTP Request	67181	Success	9228	351	26308	1
76	01:30:20.053	Pengujian proyek SIAKAD 1-76	HTTP Request	67084	Success	9228	351	28395	0
77	01:30:20.453	Pengujian proyek SIAKAD 1-80	HTTP Request	67212	Success	9228	351	28724	1
78	01:30:20.353	Pengujian proyek SIAKAD 1-79	HTTP Request	67609	Success	9222	351	28824	1
79	01:30:20.252	Pengujian proyek SIAKAD 1-78	HTTP Request	67836	Success	9228	351	28927	1
80	01:30:20.151	Pengujian proyek SIAKAD 1-77	HTTP Request	68582	Success	9228	351	29035	1
81	01:30:20.551	Pengujian proyek SIAKAD 1-81	HTTP Request	68923	Success	9228	351	30063	1
82	01:30:20.752	Pengujian proyek SIAKAD 1-83	HTTP Request	69182	Success	9228	351	31351	1
83	01:30:20.652	Pengujian proyek SIAKAD 1-82	HTTP Request	69529	Success	9228	351	31455	1
84	01:30:20.853	Pengujian proyek SIAKAD 1-84	HTTP Request	69452	Success	9228	351	31894	0
85	01:30:21.053	Pengujian proyek SIAKAD 1-86	HTTP Request	69931	Success	9228	351	32451	1
86	01:30:21.152	Pengujian proyek SIAKAD 1-87	HTTP Request	70624	Success	9228	351	32352	1
87	01:30:20.952	Pengujian proyek SIAKAD 1-85	HTTP Request	71272	Success	9228	351	32552	1
88	01:30:21.252	Pengujian proyek SIAKAD 1-88	HTTP Request	71213	Success	9228	351	33159	1
89	01:30:21.352	Pengujian proyek SIAKAD 1-89	HTTP Request	71260	Success	9228	351	33380	1
90	01:30:21.452	Pengujian proyek SIAKAD 1-90	HTTP Request	71921	Success	9228	351	33421	1
91	01:30:21.552	Pengujian proyek SIAKAD 1-91	HTTP Request	72614	Success	9228	351	33320	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
92	01:30:21.651	Pengujian proyek SIAKAD 1-92	HTTP Request	72851	Success	9228	351	33814	1
93	01:30:22.050	Pengujian proyek SIAKAD 1-96	HTTP Request	72687	Success	9228	351	34152	2
94	01:30:21.852	Pengujian proyek SIAKAD 1-94	HTTP Request	73028	Success	9228	351	34350	1
95	01:30:21.950	Pengujian proyek SIAKAD 1-95	HTTP Request	73694	Success	9228	351	34257	1
96	01:30:21.751	Pengujian proyek SIAKAD 1-93	HTTP Request	74669	Success	9228	351	34462	1
97	01:30:22.252	Pengujian proyek SIAKAD 1-98	HTTP Request	74468	Success	9228	351	35290	1
98	01:30:22.151	Pengujian proyek SIAKAD 1-97	HTTP Request	74717	Success	9228	351	35392	2
99	01:30:22.350	Pengujian proyek SIAKAD 1-99	HTTP Request	74631	Success	9228	351	35777	1
100	01:30:22.450	Pengujian proyek SIAKAD 1-100	HTTP Request	75329	Success	9228	351	36599	1

LAMPIRAN 3 Data View Result in Table Proyek SIAKAD 125 Thread

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
1	00:21:56.890	Pengujian proyek SIAKAD 1-1	HTTP Request	16758	Success	9228	351	3264	2
2	00:21:56.984	Pengujian proyek SIAKAD 1-2	HTTP Request	17389	Success	9228	351	3171	0
3	00:21:57.048	Pengujian proyek SIAKAD 1-3	HTTP Request	23554	Success	9228	351	4488	1
4	00:21:57.289	Pengujian proyek SIAKAD 1-6	HTTP Request	23783	Success	9228	351	4325	1
5	00:21:57.369	Pengujian proyek SIAKAD 1-7	HTTP Request	23836	Success	9228	351	4248	1
6	00:21:57.450	Pengujian proyek SIAKAD 1-8	HTTP Request	26392	Success	9228	351	4878	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:57.129	proyek SIAKAD 1-4	Request						
7	00:21:57.610	Pengujian proyek SIAKAD 1-10	HTTP Request	33073	Success	9228	351	5370	1
8	00:21:57.210	Pengujian proyek SIAKAD 1-5	HTTP Request	33827	Success	9228	351	5772	0
9	00:21:57.450	Pengujian proyek SIAKAD 1-8	HTTP Request	34079	Success	9228	351	5537	0
10	00:21:57.530	Pengujian proyek SIAKAD 1-9	HTTP Request	34509	Success	9228	351	5450	0
11	00:21:57.689	Pengujian proyek SIAKAD 1-11	HTTP Request	39907	Success	9228	351	6311	1
12	00:21:57.849	Pengujian proyek SIAKAD 1-13	HTTP Request	42186	Success	9228	351	6489	1
13	00:21:57.928	Pengujian proyek SIAKAD 1-14	HTTP Request	42555	Success	9222	351	6410	1
14	00:21:57.769	Pengujian proyek SIAKAD 1-12	HTTP Request	43168	Success	9228	351	6574	0
15	00:21:58.008	Pengujian proyek SIAKAD 1-15	HTTP Request	46780	Success	9228	351	7000	1
16	00:21:58.248	Pengujian proyek SIAKAD 1-18	HTTP Request	51006	Success	9228	351	7493	2
17	00:21:58.169	Pengujian proyek SIAKAD 1-17	HTTP Request	51411	Success	9228	351	7572	1
18	00:21:58.089	Pengujian proyek SIAKAD 1-16	HTTP Request	52126	Success	9228	351	7654	1
19	00:21:58.328	Pengujian proyek SIAKAD 1-19	HTTP Request	55638	Success	9228	351	8074	1
20	00:21:58.409	Pengujian proyek SIAKAD 1-20	HTTP Request	57986	Success	9228	351	8619	1
21	00:21:58.489	Pengujian proyek SIAKAD 1-21	HTTP Request	58438	Success	9228	351	8627	1
22	00:21:58.569	Pengujian proyek SIAKAD 1-22	HTTP Request	59308	Success	9228	351	8547	1
23	00:21	Pengujian	HTTP	59528	Success	9228	351	8467	0

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:58.650	proyek SIAKAD 1-23	Request						
24	00:21:58.728	Pengujian proyek SIAKAD 1-24	HTTP Request	59570	Success	9228	351	9635	1
25	00:21:58.888	Pengujian proyek SIAKAD 1-26	HTTP Request	59917	Success	9228	351	9530	1
26	00:21:58.808	Pengujian proyek SIAKAD 1-25	HTTP Request	60501	Success	9228	351	9616	1
27	00:21:58.968	Pengujian proyek SIAKAD 1-27	HTTP Request	61269	Success	9228	351	9456	1
28	00:21:59.048	Pengujian proyek SIAKAD 1-28	HTTP Request	61424	Success	9228	351	9388	1
29	00:21:59.368	Pengujian proyek SIAKAD 1-32	HTTP Request	61224	Success	9228	351	10475	1
30	00:21:59.289	Pengujian proyek SIAKAD 1-31	HTTP Request	61861	Success	9228	351	10554	1
31	00:21:59.209	Pengujian proyek SIAKAD 1-30	HTTP Request	62368	Success	9228	351	10636	0
32	00:21:59.128	Pengujian proyek SIAKAD 1-29	HTTP Request	63339	Success	9228	351	10728	1
33	00:21:59.448	Pengujian proyek SIAKAD 1-33	HTTP Request	63204	Success	9228	351	11148	1
34	00:21:59.769	Pengujian proyek SIAKAD 1-37	HTTP Request	63023	Success	9228	351	11470	1
35	00:21:59.528	Pengujian proyek SIAKAD 1-34	HTTP Request	63929	Success	9228	351	11711	0
36	00:21:59.689	Pengujian proyek SIAKAD 1-36	HTTP Request	65089	Success	9228	351	11559	1
37	00:22:00.089	Pengujian proyek SIAKAD 1-41	HTTP Request	64847	Success	9228	351	12486	1
38	00:21:59.849	Pengujian proyek SIAKAD 1-38	HTTP Request	65227	Success	9228	351	12726	1
39	00:21:59.608	Pengujian proyek SIAKAD 1-35	HTTP Request	65992	Success	9228	351	11636	1
40	00:22	Pengujian	HTTP	65592	Success	9228	351	1256	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:00.009	proyek SIAKAD 1-40	Request					7	
41	00:22:00.169	Pengujian proyek SIAKAD 1-42	HTTP Request	66791	Success	9228	351	13753	1
42	00:21:59.931	Pengujian proyek SIAKAD 1-39	HTTP Request	67110	Success	9222	351	14193	0
43	00:22:00.409	Pengujian proyek SIAKAD 1-45	HTTP Request	66632	Success	9222	351	14874	1
44	00:22:00.328	Pengujian proyek SIAKAD 1-44	HTTP Request	66713	Success	9228	351	14957	1
45	00:22:00.248	Pengujian proyek SIAKAD 1-43	HTTP Request	66796	Success	9228	351	13674	1
46	00:22:00.648	Pengujian proyek SIAKAD 1-48	HTTP Request	67882	Success	9228	351	15828	0
47	00:22:00.489	Pengujian proyek SIAKAD 1-46	HTTP Request	68040	Success	9228	351	14805	1
48	00:22:00.728	Pengujian proyek SIAKAD 1-49	HTTP Request	67805	Success	9222	351	16039	1
49	00:22:00.569	Pengujian proyek SIAKAD 1-47	HTTP Request	67987	Success	9228	351	15651	1
50	00:22:00.808	Pengujian proyek SIAKAD 1-50	HTTP Request	68555	Success	9228	351	15961	1
51	00:22:00.889	Pengujian proyek SIAKAD 1-51	HTTP Request	69201	Success	9228	351	16546	1
52	00:22:01.049	Pengujian proyek SIAKAD 1-53	HTTP Request	69043	Success	9228	351	17088	1
53	00:22:01.209	Pengujian proyek SIAKAD 1-55	HTTP Request	69154	Success	9228	351	16928	1
54	00:22:00.969	Pengujian proyek SIAKAD 1-52	HTTP Request	69396	Success	9228	351	17167	1
55	00:22:01.129	Pengujian proyek SIAKAD 1-54	HTTP Request	70405	Success	9228	351	17015	0
56	00:22:01.370	Pengujian proyek SIAKAD 1-57	HTTP Request	70181	Success	9228	351	18186	1
57	00:22	Pengujian	HTTP	70263	Success	9222	351	1817	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:01.290	proyek SIAKAD 1-56	Request					6	
58	00:22:01.529	Pengujian proyek SIAKAD 1-59	HTTP Request	70310	Success	9228	351	18536	1
59	00:22:01.450	Pengujian proyek SIAKAD 1-58	HTTP Request	70386	Success	9228	351	18107	1
60	00:22:01.689	Pengujian proyek SIAKAD 1-61	HTTP Request	71274	Success	9228	351	20404	1
61	00:22:01.609	Pengujian proyek SIAKAD 1-60	HTTP Request	71356	Success	9228	351	19358	1
62	00:22:01.770	Pengujian proyek SIAKAD 1-62	HTTP Request	71513	Success	9228	351	20967	1
63	00:22:01.930	Pengujian proyek SIAKAD 1-64	HTTP Request	71353	Success	9228	351	21165	1
64	00:22:01.850	Pengujian proyek SIAKAD 1-63	HTTP Request	71434	Success	9228	351	21160	1
65	00:22:02.009	Pengujian proyek SIAKAD 1-65	HTTP Request	72376	Success	9228	351	22091	1
66	00:22:02.169	Pengujian proyek SIAKAD 1-67	HTTP Request	72218	Success	9228	351	22849	1
67	00:22:02.329	Pengujian proyek SIAKAD 1-69	HTTP Request	72331	Success	9228	351	23334	0
68	00:22:02.408	Pengujian proyek SIAKAD 1-70	HTTP Request	72253	Success	9228	351	23739	1
69	00:22:02.249	Pengujian proyek SIAKAD 1-68	HTTP Request	72412	Success	9222	351	22839	1
70	00:22:02.490	Pengujian proyek SIAKAD 1-71	HTTP Request	73273	Success	9228	351	24552	0
71	00:22:02.089	Pengujian proyek SIAKAD 1-66	HTTP Request	73675	Success	9228	351	24437	1
72	00:22:02.569	Pengujian proyek SIAKAD 1-72	HTTP Request	73468	Success	9228	351	24518	1
73	00:22:02.648	Pengujian proyek SIAKAD 1-73	HTTP Request	73389	Success	9228	351	25162	1
74	00:22	Pengujian	HTTP	73311	Success	9228	351	2558	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:02.728	proyek SIAKAD 1-74	Request					3	
75	00:22:02.809	Pengujian proyek SIAKAD 1-75	HTTP Request	74369	Success	9228	351	25937	1
76	00:22:02.889	Pengujian proyek SIAKAD 1-76	HTTP Request	74291	Success	9228	351	26181	1
77	00:22:02.969	Pengujian proyek SIAKAD 1-77	HTTP Request	74508	Success	9228	351	26223	1
78	00:22:03.131	Pengujian proyek SIAKAD 1-79	HTTP Request	74346	Success	9228	351	29734	0
79	00:22:03.049	Pengujian proyek SIAKAD 1-78	HTTP Request	74892	Success	9228	351	28212	1
80	00:22:03.210	Pengujian proyek SIAKAD 1-80	HTTP Request	75548	Success	9228	351	29962	1
81	00:22:03.290	Pengujian proyek SIAKAD 1-81	HTTP Request	75470	Success	9228	351	30117	0
82	00:22:03.371	Pengujian proyek SIAKAD 1-82	HTTP Request	75636	Success	9228	351	30277	0
83	00:22:03.450	Pengujian proyek SIAKAD 1-83	HTTP Request	75559	Success	9228	351	30678	1
84	00:22:03.531	Pengujian proyek SIAKAD 1-84	HTTP Request	76705	Success	9228	351	31429	0
85	00:22:03.689	Pengujian proyek SIAKAD 1-86	HTTP Request	76596	Success	9228	351	31859	1
86	00:22:03.610	Pengujian proyek SIAKAD 1-85	HTTP Request	76676	Success	9228	351	31692	1
87	00:22:03.849	Pengujian proyek SIAKAD 1-88	HTTP Request	76933	Success	9228	351	32429	1
88	00:22:03.769	Pengujian proyek SIAKAD 1-87	HTTP Request	77015	Success	9222	351	32043	2
89	00:22:03.930	Pengujian proyek SIAKAD 1-89	HTTP Request	77756	Success	9228	351	33240	1
90	00:22:04.089	Pengujian proyek SIAKAD 1-91	HTTP Request	77599	Success	9228	351	33880	1
91	00:22	Pengujian	HTTP	77689	Success	9228	351	3366	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:04.009	proyek SIAKAD 1-90	Request					8	
92	00:22:04.169	Pengujian proyek SIAKAD 1-92	HTTP Request	78014	Success	9228	351	34299	0
93	00:22:04.248	Pengujian proyek SIAKAD 1-93	HTTP Request	77936	Success	9228	351	35136	1
94	00:22:04.408	Pengujian proyek SIAKAD 1-95	HTTP Request	78894	Success	9228	351	37122	1
95	00:22:04.570	Pengujian proyek SIAKAD 1-97	HTTP Request	79079	Success	9228	351	37607	1
96	00:22:04.649	Pengujian proyek SIAKAD 1-98	HTTP Request	79000	Success	9228	351	37922	1
97	00:22:04.488	Pengujian proyek SIAKAD 1-96	HTTP Request	79163	Success	9228	351	37367	1
98	00:22:04.329	Pengujian proyek SIAKAD 1-94	HTTP Request	80690	Success	9228	351	35376	1
99	00:22:04.809	Pengujian proyek SIAKAD 1-100	HTTP Request	80264	Success	9228	351	38773	0
100	00:22:04.889	Pengujian proyek SIAKAD 1-101	HTTP Request	80184	Success	9222	351	39078	1
101	00:22:04.729	Pengujian proyek SIAKAD 1-99	HTTP Request	80345	Success	9228	351	38373	0
102	00:22:04.972	Pengujian proyek SIAKAD 1-102	HTTP Request	80110	Success	9228	351	39255	0
103	00:22:05.129	Pengujian proyek SIAKAD 1-104	HTTP Request	81368	Success	9228	351	40510	1
104	00:22:05.050	Pengujian proyek SIAKAD 1-103	HTTP Request	81447	Success	9228	351	40131	0
105	00:22:05.210	Pengujian proyek SIAKAD 1-105	HTTP Request	81289	Success	9228	351	40846	1
106	00:22:05.369	Pengujian proyek SIAKAD 1-107	HTTP Request	81573	Success	9228	351	41517	1
107	00:22:05.291	Pengujian proyek SIAKAD 1-106	HTTP Request	81755	Success	9228	351	40969	0
108	00:22	Pengujian	HTTP	82503	Success	9228	351	4178	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:05.449	proyek SIAKAD 1-108	Request					5	
109	00:22:05.610	Pengujian proyek SIAKAD 1-110	HTTP Request	82344	Success	9228	351	42616	1
110	00:22:05.530	Pengujian proyek SIAKAD 1-109	HTTP Request	82435	Success	9228	351	42311	1
111	00:22:05.690	Pengujian proyek SIAKAD 1-111	HTTP Request	83462	Success	9228	351	42656	0
112	00:22:05.769	Pengujian proyek SIAKAD 1-112	HTTP Request	83559	Success	9228	351	44531	1
113	00:22:05.850	Pengujian proyek SIAKAD 1-113	HTTP Request	83727	Success	9228	351	44618	0
114	00:22:05.930	Pengujian proyek SIAKAD 1-114	HTTP Request	83649	Success	9228	351	45406	0
115	00:22:06.010	Pengujian proyek SIAKAD 1-115	HTTP Request	84312	Success	9228	351	45673	1
116	00:22:06.329	Pengujian proyek SIAKAD 1-119	HTTP Request	84713	Success	9222	351	47130	1
117	00:22:06.091	Pengujian proyek SIAKAD 1-116	HTTP Request	84951	Success	9228	351	46103	0
118	00:22:06.169	Pengujian proyek SIAKAD 1-117	HTTP Request	84874	Success	9228	351	46060	0
119	00:22:06.249	Pengujian proyek SIAKAD 1-118	HTTP Request	84803	Success	9228	351	46316	1
120	00:22:06.410	Pengujian proyek SIAKAD 1-120	HTTP Request	86082	Success	9228	351	47743	1
121	00:22:06.490	Pengujian proyek SIAKAD 1-121	HTTP Request	86014	Success	9228	351	47710	1
122	00:22:06.649	Pengujian proyek SIAKAD 1-123	HTTP Request	85856	Success	9228	351	48897	1
123	00:22:06.570	Pengujian proyek SIAKAD 1-122	HTTP Request	86264	Success	9228	351	48102	1
124	00:22:06.729	Pengujian proyek SIAKAD 1-124	HTTP Request	86371	Success	9228	351	49277	1
125	00:22	Pengujian	HTTP	87719	Success	9228	351	4925	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:06.809	proyek SIAKAD 1-125	Request					4	

LAMPIRAN 4 Data View Result in Table SIAKAD UIN Malang 75 Thread

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
1	03:28:32.370	Pengujian SIAKAD UIN Malang 1-1	HTTP Request	17847	Success	9276	375	4125	1
2	03:28:32.502	Pengujian SIAKAD UIN Malang 1-2	HTTP Request	17821	Success	9276	375	4028	1
3	03:28:32.635	Pengujian SIAKAD UIN Malang 1-3	HTTP Request	19893	Success	9276	375	4501	0
4	03:28:32.769	Pengujian SIAKAD UIN Malang 1-4	HTTP Request	22838	Success	9276	375	5010	0
5	03:28:33.033	Pengujian SIAKAD UIN Malang 1-6	HTTP Request	27078	Success	9276	375	5408	1
6	03:28:32.900	Pengujian SIAKAD UIN Malang 1-5	HTTP Request	27856	Success	9276	375	5693	1
7	03:28:33.166	Pengujian SIAKAD UIN Malang 1-7	HTTP Request	29277	Success	9276	375	5341	1
8	03:28:33.300	Pengujian SIAKAD UIN Malang 1-8	HTTP Request	31317	Success	9276	375	5944	0
9	03:28:33.432	Pengujian SIAKAD UIN Malang 1-9	HTTP Request	33314	Success	9276	375	6510	1
10	03:28:33.566	Pengujian SIAKAD UIN Malang 1-10	HTTP Request	35474	Success	9276	375	6889	1
11	03:28:33.699	Pengujian SIAKAD UIN Malang 1-11	HTTP Request	36249	Success	9276	375	6799	1
12	03:28:33.831	Pengujian SIAKAD UIN Malang 1-12	HTTP Request	37498	Success	9276	375	6940	1
13	03:28:33.964	Pengujian SIAKAD UIN Malang 1-13	HTTP Request	39655	Success	9276	375	7396	0
14	03:28:34.096	Pengujian SIAKAD UIN Malang 1-14	HTTP Request	42472	Success	9276	375	8042	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
15	03:28:34.230	Pengujian SIAKAD UIN Malang 1-15	HTTP Request	43528	Success	9276	375	8269	1
16	03:28:34.363	Pengujian SIAKAD UIN Malang 1-16	HTTP Request	43618	Success	9276	375	8194	0
17	03:28:34.497	Pengujian SIAKAD UIN Malang 1-17	HTTP Request	44347	Success	9276	375	8529	1
18	03:28:34.631	Pengujian SIAKAD UIN Malang 1-18	HTTP Request	44640	Success	9276	375	8911	1
19	03:28:34.763	Pengujian SIAKAD UIN Malang 1-19	HTTP Request	45147	Success	9276	375	9514	1
20	03:28:34.896	Pengujian SIAKAD UIN Malang 1-20	HTTP Request	45161	Success	9276	375	9569	1
21	03:28:35.028	Pengujian SIAKAD UIN Malang 1-21	HTTP Request	45253	Success	9276	375	9476	1
22	03:28:35.161	Pengujian SIAKAD UIN Malang 1-22	HTTP Request	45948	Success	9276	375	10052	1
23	03:28:35.294	Pengujian SIAKAD UIN Malang 1-23	HTTP Request	46237	Success	9276	375	10321	0
24	03:28:35.428	Pengujian SIAKAD UIN Malang 1-24	HTTP Request	46778	Success	9276	375	10965	1
25	03:28:35.561	Pengujian SIAKAD UIN Malang 1-25	HTTP Request	46793	Success	9276	375	10886	0
26	03:28:35.694	Pengujian SIAKAD UIN Malang 1-26	HTTP Request	46880	Success	9276	375	10810	1
27	03:28:35.827	Pengujian SIAKAD UIN Malang 1-27	HTTP Request	47599	Success	9276	375	11543	0
28	03:28:35.961	Pengujian SIAKAD UIN Malang 1-28	HTTP Request	47982	Success	9276	375	11815	0
29	03:28:36.093	Pengujian SIAKAD UIN Malang 1-29	HTTP Request	48419	Success	9276	375	12217	1
30	03:28:36.226	Pengujian SIAKAD UIN Malang 1-30	HTTP Request	48426	Success	9276	375	12103	0
31	03:28:36.359	Pengujian SIAKAD UIN Malang 1-31	HTTP Request	48565	Success	9276	375	12255	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
32	03:28:36.492	Pengujian SIAKAD UIN Malang 1-32	HTTP Request	49242	Success	9276	375	12907	1
33	03:28:36.624	Pengujian SIAKAD UIN Malang 1-33	HTTP Request	49554	Success	9276	375	13390	0
34	03:28:36.757	Pengujian SIAKAD UIN Malang 1-34	HTTP Request	49977	Success	9276	375	13942	1
35	03:28:37.022	Pengujian SIAKAD UIN Malang 1-36	HTTP Request	49870	Success	9276	375	15143	1
36	03:28:37.157	Pengujian SIAKAD UIN Malang 1-37	HTTP Request	49989	Success	9276	375	15263	1
37	03:28:37.290	Pengujian SIAKAD UIN Malang 1-38	HTTP Request	50701	Success	9276	375	15640	0
38	03:28:37.555	Pengujian SIAKAD UIN Malang 1-40	HTTP Request	50916	Success	9276	375	17002	1
39	03:28:37.688	Pengujian SIAKAD UIN Malang 1-41	HTTP Request	51308	Success	9276	375	16950	1
40	03:28:37.821	Pengujian SIAKAD UIN Malang 1-42	HTTP Request	51319	Success	9276	375	18822	0
41	03:28:37.954	Pengujian SIAKAD UIN Malang 1-43	HTTP Request	51496	Success	9276	375	18743	1
42	03:28:37.423	Pengujian SIAKAD UIN Malang 1-39	HTTP Request	52863	Success	9276	375	20201	1
43	03:28:38.088	Pengujian SIAKAD UIN Malang 1-44	HTTP Request	52713	Success	9276	375	19717	0
44	03:28:36.890	Pengujian SIAKAD UIN Malang 1-35	HTTP Request	54352	Success	9276	375	20918	1
45	03:28:38.221	Pengujian SIAKAD UIN Malang 1-45	HTTP Request	53162	Success	9276	375	20373	1
46	03:28:38.353	Pengujian SIAKAD UIN Malang 1-46	HTTP Request	53438	Success	9276	375	20273	0
47	03:28:38.485	Pengujian SIAKAD UIN Malang 1-47	HTTP Request	54090	Success	9276	375	21410	1
48	03:28:38.753	Pengujian SIAKAD UIN Malang 1-49	HTTP Request	54385	Success	9276	375	23310	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
49	03:28:38.884	Pengujian SIAKAD UIN Malang 1-50	HTTP Request	54682	Success	9276	375	23393	1
50	03:28:39.018	Pengujian SIAKAD UIN Malang 1-51	HTTP Request	54704	Success	9276	375	24015	1
51	03:28:38.620	Pengujian SIAKAD UIN Malang 1-48	HTTP Request	55529	Success	9276	375	24935	1
52	03:28:39.151	Pengujian SIAKAD UIN Malang 1-52	HTTP Request	55762	Success	9276	375	25109	0
53	03:28:39.284	Pengujian SIAKAD UIN Malang 1-53	HTTP Request	56183	Success	9276	375	25259	1
54	03:28:39.417	Pengujian SIAKAD UIN Malang 1-54	HTTP Request	56407	Success	9276	375	25772	1
55	03:28:39.551	Pengujian SIAKAD UIN Malang 1-55	HTTP Request	56462	Success	9276	375	25996	1
56	03:28:39.684	Pengujian SIAKAD UIN Malang 1-56	HTTP Request	56748	Success	9276	375	26704	1
57	03:28:39.816	Pengujian SIAKAD UIN Malang 1-57	HTTP Request	57356	Success	9276	375	26685	1
58	03:28:39.949	Pengujian SIAKAD UIN Malang 1-58	HTTP Request	57823	Success	9276	375	27314	1
59	03:28:40.082	Pengujian SIAKAD UIN Malang 1-59	HTTP Request	58001	Success	9276	375	27506	1
60	03:28:40.215	Pengujian SIAKAD UIN Malang 1-60	HTTP Request	58039	Success	9276	375	28220	1
61	03:28:40.348	Pengujian SIAKAD UIN Malang 1-61	HTTP Request	58324	Success	9276	375	28180	0
62	03:28:40.482	Pengujian SIAKAD UIN Malang 1-62	HTTP Request	58991	Success	9276	375	28907	1
63	03:28:40.615	Pengujian SIAKAD UIN Malang 1-63	HTTP Request	59526	Success	9276	375	29816	0
64	03:28:40.748	Pengujian SIAKAD UIN Malang 1-64	HTTP Request	59650	Success	9276	375	29844	1
65	03:28:40.880	Pengujian SIAKAD UIN Malang 1-65	HTTP Request	59686	Success	9276	375	30536	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Lateny	Connect Time (ms)
66	03:28:41.013	Pengujian SIAKAD UIN Malang 1-66	HTTP Request	60012	Success	9276	375	31069	1
67	03:28:41.146	Pengujian SIAKAD UIN Malang 1-67	HTTP Request	60657	Success	9276	375	31398	0
68	03:28:41.279	Pengujian SIAKAD UIN Malang 1-68	HTTP Request	61191	Success	9276	375	31454	1
69	03:28:41.415	Pengujian SIAKAD UIN Malang 1-69	HTTP Request	61254	Success	9276	375	32091	0
70	03:28:41.546	Pengujian SIAKAD UIN Malang 1-70	HTTP Request	61314	Success	9276	375	32704	1
71	03:28:41.811	Pengujian SIAKAD UIN Malang 1-72	HTTP Request	61619	Success	9276	375	33038	1
72	03:28:41.944	Pengujian SIAKAD UIN Malang 1-73	HTTP Request	62062	Success	9276	375	33571	0
73	03:28:42.078	Pengujian SIAKAD UIN Malang 1-74	HTTP Request	62490	Success	9276	375	33576	1
74	03:28:42.211	Pengujian SIAKAD UIN Malang 1-75	HTTP Request	62448	Success	9276	375	34759	1
75	03:28:41.679	Pengujian SIAKAD UIN Malang 1-71	HTTP Request	63137	Success	9276	375	35915	1

LAMPIRAN 5 Data View Result in Table SIAKAD UIN Malang 100 Thread

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Lateny	Connect Time (ms)
1	02:24:26.737	Pengujian SIAKAD UIN Malang 1-1	HTTP Request	19403	Success	9276	375	3119	4
2	02:24:26.845	Pengujian SIAKAD UIN Malang 1-2	HTTP Request	19625	Success	9276	375	3013	1
3	02:24:26.936	Pengujian SIAKAD UIN Malang 1-3	HTTP Request	25283	Success	9276	375	4323	0
4	02:24:27.335	Pengujian SIAKAD UIN Malang 1-7	HTTP Request	25082	Success	9276	375	3960	1
5	02:24	Pengujian	HTTP	26525	Success	9276	375	4061	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:27.235	SIAKAD UIN Malang 1-6	Request						
6	02:24:27.035	Pengujian SIAKAD UIN Malang 1-4	HTTP Request	31585	Success	9276	375	5318	1
7	02:24:27.535	Pengujian SIAKAD UIN Malang 1-9	HTTP Request	32825	Success	9276	375	5269	1
8	02:24:27.135	Pengujian SIAKAD UIN Malang 1-5	HTTP Request	36064	Success	9276	375	6170	0
9	02:24:27.635	Pengujian SIAKAD UIN Malang 1-10	HTTP Request	38616	Success	9276	375	5693	1
10	02:24:27.736	Pengujian SIAKAD UIN Malang 1-11	HTTP Request	42074	Success	9276	375	6839	0
11	02:24:27.835	Pengujian SIAKAD UIN Malang 1-12	HTTP Request	44268	Success	9276	375	7156	1
12	02:24:27.936	Pengujian SIAKAD UIN Malang 1-13	HTTP Request	45764	Success	9276	375	7367	0
13	02:24:28.036	Pengujian SIAKAD UIN Malang 1-14	HTTP Request	46910	Success	9276	375	7333	0
14	02:24:28.135	Pengujian SIAKAD UIN Malang 1-15	HTTP Request	53404	Success	9276	375	8701	1
15	02:24:28.235	Pengujian SIAKAD UIN Malang 1-16	HTTP Request	53361	Success	9276	375	9012	1
16	02:24:28.334	Pengujian SIAKAD UIN Malang 1-17	HTTP Request	54159	Success	9276	375	9062	1
17	02:24:28.434	Pengujian SIAKAD UIN Malang 1-18	HTTP Request	54277	Success	9276	375	9034	1
18	02:24:28.534	Pengujian SIAKAD UIN Malang 1-19	HTTP Request	54608	Success	9276	375	10500	1
19	02:24:27.435	Pengujian SIAKAD UIN Malang 1-8	HTTP Request	56328	Success	9276	375	11825	1
20	02:24:28.634	Pengujian SIAKAD UIN Malang 1-20	HTTP Request	55225	Success	9276	375	10714	1
21	02:24:28.733	Pengujian SIAKAD UIN Malang 1-21	HTTP Request	56067	Success	9276	375	10677	1
22	02:24	Pengujian	HTTP	56160	Success	9276	375	1076	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:28.835	SIAKAD UIN Malang 1-22	Request					2	
23	02:24:28.935	Pengujian SIAKAD UIN Malang 1-23	HTTP Request	56523	Success	9276	375	12247	1
24	02:24:29.135	Pengujian SIAKAD UIN Malang 1-25	HTTP Request	56772	Success	9276	375	12236	0
25	02:24:29.236	Pengujian SIAKAD UIN Malang 1-26	HTTP Request	56745	Success	9276	375	12208	0
26	02:24:29.335	Pengujian SIAKAD UIN Malang 1-27	HTTP Request	57731	Success	9276	375	12355	0
27	02:24:29.435	Pengujian SIAKAD UIN Malang 1-28	HTTP Request	57862	Success	9276	375	13869	0
28	02:24:29.035	Pengujian SIAKAD UIN Malang 1-24	HTTP Request	58686	Success	9276	375	14426	1
29	02:24:29.735	Pengujian SIAKAD UIN Malang 1-31	HTTP Request	58337	Success	9276	375	14151	0
30	02:24:29.534	Pengujian SIAKAD UIN Malang 1-29	HTTP Request	58657	Success	9276	375	16817	1
31	02:24:29.634	Pengujian SIAKAD UIN Malang 1-30	HTTP Request	59807	Success	9276	375	16834	1
32	02:24:30.135	Pengujian SIAKAD UIN Malang 1-35	HTTP Request	59546	Success	9276	375	17731	0
33	02:24:30.036	Pengujian SIAKAD UIN Malang 1-34	HTTP Request	60077	Success	9276	375	17830	1
34	02:24:30.235	Pengujian SIAKAD UIN Malang 1-36	HTTP Request	60193	Success	9276	375	17631	1
35	02:24:29.936	Pengujian SIAKAD UIN Malang 1-33	HTTP Request	60666	Success	9276	375	17936	1
36	02:24:29.835	Pengujian SIAKAD UIN Malang 1-32	HTTP Request	61862	Success	9276	375	18513	1
37	02:24:30.534	Pengujian SIAKAD UIN Malang 1-39	HTTP Request	61459	Success	9276	375	18722	1
38	02:24:30.335	Pengujian SIAKAD UIN Malang 1-37	HTTP Request	62058	Success	9276	375	18922	1
39	02:24	Pengujian	HTTP	62223	Success	9276	375	1883	0

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:30.438	SIAKAD UIN Malang 1-38	Request					0	
40	02:24:30.636	Pengujian SIAKAD UIN Malang 1-40	HTTP Request	62185	Success	9276	375	19314	1
41	02:24:30.736	Pengujian SIAKAD UIN Malang 1-41	HTTP Request	63232	Success	9276	375	19703	1
42	02:24:30.835	Pengujian SIAKAD UIN Malang 1-42	HTTP Request	63477	Success	9276	375	19870	1
43	02:24:30.936	Pengujian SIAKAD UIN Malang 1-43	HTTP Request	63773	Success	9276	375	19770	1
44	02:24:31.037	Pengujian SIAKAD UIN Malang 1-44	HTTP Request	63932	Success	9276	375	20311	1
45	02:24:31.137	Pengujian SIAKAD UIN Malang 1-45	HTTP Request	64009	Success	9276	375	20839	1
46	02:24:31.237	Pengujian SIAKAD UIN Malang 1-46	HTTP Request	65065	Success	9276	375	20836	1
47	02:24:31.335	Pengujian SIAKAD UIN Malang 1-47	HTTP Request	65347	Success	9276	375	22745	1
48	02:24:31.437	Pengujian SIAKAD UIN Malang 1-48	HTTP Request	65644	Success	9276	375	22724	0
49	02:24:31.538	Pengujian SIAKAD UIN Malang 1-49	HTTP Request	65770	Success	9276	375	22814	1
50	02:24:31.637	Pengujian SIAKAD UIN Malang 1-50	HTTP Request	65855	Success	9276	375	22915	1
51	02:24:31.737	Pengujian SIAKAD UIN Malang 1-51	HTTP Request	66885	Success	9276	375	24120	1
52	02:24:31.836	Pengujian SIAKAD UIN Malang 1-52	HTTP Request	67223	Success	9276	375	24248	1
53	02:24:31.935	Pengujian SIAKAD UIN Malang 1-53	HTTP Request	67512	Success	9276	375	24229	1
54	02:24:32.037	Pengujian SIAKAD UIN Malang 1-54	HTTP Request	67618	Success	9276	375	24400	0
55	02:24:32.134	Pengujian SIAKAD UIN Malang 1-55	HTTP Request	67733	Success	9276	375	24507	1
56	02:24	Pengujian	HTTP	68726	Success	9276	375	2575	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:32.235	SIAKAD UIN Malang 1-56	Request					2	
57	02:24:32.334	Pengujian SIAKAD UIN Malang 1-57	HTTP Request	69027	Success	9276	375	25821	1
58	02:24:32.435	Pengujian SIAKAD UIN Malang 1-58	HTTP Request	69293	Success	9276	375	26208	1
59	02:24:32.535	Pengujian SIAKAD UIN Malang 1-59	HTTP Request	69386	Success	9276	375	26362	1
60	02:24:32.734	Pengujian SIAKAD UIN Malang 1-61	HTTP Request	69402	Success	9276	375	27349	1
61	02:24:32.835	Pengujian SIAKAD UIN Malang 1-62	HTTP Request	70428	Success	9276	375	27249	1
62	02:24:32.934	Pengujian SIAKAD UIN Malang 1-63	HTTP Request	70726	Success	9276	375	28078	1
63	02:24:33.135	Pengujian SIAKAD UIN Malang 1-65	HTTP Request	70863	Success	9276	375	28386	1
64	02:24:33.035	Pengujian SIAKAD UIN Malang 1-64	HTTP Request	71240	Success	9276	375	28487	1
65	02:24:33.234	Pengujian SIAKAD UIN Malang 1-66	HTTP Request	71203	Success	9276	375	29210	1
66	02:24:33.335	Pengujian SIAKAD UIN Malang 1-67	HTTP Request	72224	Success	9276	375	29651	1
67	02:24:32.637	Pengujian SIAKAD UIN Malang 1-60	HTTP Request	73409	Success	9276	375	30460	0
68	02:24:33.435	Pengujian SIAKAD UIN Malang 1-68	HTTP Request	72924	Success	9276	375	31211	1
69	02:24:33.535	Pengujian SIAKAD UIN Malang 1-69	HTTP Request	73052	Success	9276	375	31516	1
70	02:24:33.633	Pengujian SIAKAD UIN Malang 1-70	HTTP Request	73088	Success	9276	375	31505	1
71	02:24:33.734	Pengujian SIAKAD UIN Malang 1-71	HTTP Request	74154	Success	9276	375	31596	1
72	02:24:33.834	Pengujian SIAKAD UIN Malang 1-72	HTTP Request	74534	Success	9276	375	32950	1
73	02:24	Pengujian	HTTP	74737	Success	9276	375	3316	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:33.933	SIAKAD UIN Malang 1-73	Request					7	
74	02:24:34.034	Pengujian SIAKAD UIN Malang 1-74	HTTP Request	74852	Success	9276	375	33193	1
75	02:24:34.134	Pengujian SIAKAD UIN Malang 1-75	HTTP Request	74855	Success	9276	375	33289	1
76	02:24:34.234	Pengujian SIAKAD UIN Malang 1-76	HTTP Request	76044	Success	9276	375	34158	1
77	02:24:34.334	Pengujian SIAKAD UIN Malang 1-77	HTTP Request	76449	Success	9276	375	34636	0
78	02:24:34.434	Pengujian SIAKAD UIN Malang 1-78	HTTP Request	76569	Success	9276	375	34845	0
79	02:24:34.534	Pengujian SIAKAD UIN Malang 1-79	HTTP Request	76580	Success	9276	375	34874	1
80	02:24:34.634	Pengujian SIAKAD UIN Malang 1-80	HTTP Request	76575	Success	9276	375	35935	1
81	02:24:34.735	Pengujian SIAKAD UIN Malang 1-81	HTTP Request	77851	Success	9276	375	36358	0
82	02:24:34.834	Pengujian SIAKAD UIN Malang 1-82	HTTP Request	78199	Success	9276	375	36492	1
83	02:24:34.934	Pengujian SIAKAD UIN Malang 1-83	HTTP Request	78261	Success	9276	375	36487	1
84	02:24:35.033	Pengujian SIAKAD UIN Malang 1-84	HTTP Request	78248	Success	9276	375	37588	1
85	02:24:35.134	Pengujian SIAKAD UIN Malang 1-85	HTTP Request	78298	Success	9276	375	38073	1
86	02:24:35.233	Pengujian SIAKAD UIN Malang 1-86	HTTP Request	79720	Success	9276	375	38162	1
87	02:24:35.334	Pengujian SIAKAD UIN Malang 1-87	HTTP Request	79953	Success	9276	375	38877	0
88	02:24:35.433	Pengujian SIAKAD UIN Malang 1-88	HTTP Request	79943	Success	9276	375	39808	1
89	02:24:35.635	Pengujian SIAKAD UIN Malang 1-90	HTTP Request	79866	Success	9276	375	40192	0
90	02:24	Pengujian	HTTP	79982	Success	9276	375	4050	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	:35.734	SIAKAD UIN Malang 1-91	Request					8	
91	02:24:35.843	Pengujian SIAKAD UIN Malang 1-92	HTTP Request	81403	Success	9276	375	41274	1
92	02:24:35.937	Pengujian SIAKAD UIN Malang 1-93	HTTP Request	81533	Success	9276	375	41388	1
93	02:24:36.043	Pengujian SIAKAD UIN Malang 1-94	HTTP Request	81489	Success	9276	375	41941	1
94	02:24:36.136	Pengujian SIAKAD UIN Malang 1-95	HTTP Request	81626	Success	9276	375	42217	1
95	02:24:35.535	Pengujian SIAKAD UIN Malang 1-89	HTTP Request	82472	Success	9276	375	43137	0
96	02:24:36.245	Pengujian SIAKAD UIN Malang 1-96	HTTP Request	83306	Success	9276	375	43076	0
97	02:24:36.334	Pengujian SIAKAD UIN Malang 1-97	HTTP Request	83354	Success	9276	375	43120	1
98	02:24:36.435	Pengujian SIAKAD UIN Malang 1-98	HTTP Request	83319	Success	9276	375	43737	1
99	02:24:36.535	Pengujian SIAKAD UIN Malang 1-99	HTTP Request	83463	Success	9276	375	43914	1
100	02:24:36.634	Pengujian SIAKAD UIN Malang 1-100	HTTP Request	83591	Success	9276	375	44176	1

LAMPIRAN 6 Data View Result in Table SIAKAD UIN Malang 125 Thread

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
1	02:14:30.457	Pengujian SIAKAD UIN Malang 1-1	HTTP Request	20775	Success	9276	375	3412	2
2	02:14:30.549	Pengujian SIAKAD UIN Malang 1-2	HTTP Request	21270	Success	9276	375	3319	0
3	02:14:30.626	Pengujian SIAKAD UIN Malang 1-3	HTTP Request	28308	Success	9276	375	4852	0
4	02:14:30.9	Pengujian SIAKAD UIN	HTTP Request	28836	Success	9276	375	4568	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	36	Malang 1-7	t						
5	02:14:30.856	Pengujian SIAKAD UIN Malang 1-6	HTTP Request	28921	Success	9270	375	4654	1
6	02:14:31.178	Pengujian SIAKAD UIN Malang 1-10	HTTP Request	39490	Success	9276	375	5855	1
7	02:14:31.017	Pengujian SIAKAD UIN Malang 1-8	HTTP Request	40222	Success	9276	375	6016	0
8	02:14:31.096	Pengujian SIAKAD UIN Malang 1-9	HTTP Request	40639	Success	9276	375	5938	1
9	02:14:30.776	Pengujian SIAKAD UIN Malang 1-5	HTTP Request	42928	Success	9276	375	6561	1
10	02:14:31.258	Pengujian SIAKAD UIN Malang 1-11	HTTP Request	51145	Success	9276	375	7353	1
11	02:14:31.417	Pengujian SIAKAD UIN Malang 1-13	HTTP Request	51501	Success	9276	375	7200	1
12	02:14:31.337	Pengujian SIAKAD UIN Malang 1-12	HTTP Request	53834	Success	9276	375	7526	1
13	02:14:31.497	Pengujian SIAKAD UIN Malang 1-14	HTTP Request	59364	Success	9276	375	8234	1
14	02:14:31.657	Pengujian SIAKAD UIN Malang 1-16	HTTP Request	62945	Success	9276	375	8632	1
15	02:14:31.576	Pengujian SIAKAD UIN Malang 1-15	HTTP Request	63745	Success	9276	375	8714	1
16	02:14:31.736	Pengujian SIAKAD UIN Malang 1-17	HTTP Request	65388	Success	9276	375	9525	1
17	02:14:31.975	Pengujian SIAKAD UIN Malang 1-20	HTTP Request	65781	Success	9276	375	9779	1
18	02:14:31.896	Pengujian SIAKAD UIN Malang 1-19	HTTP Request	67030	Success	9276	375	9858	1
19	02:14:31.816	Pengujian SIAKAD UIN Malang 1-18	HTTP Request	67312	Success	9276	375	9939	1
20	02:14:30.703	Pengujian SIAKAD UIN Malang 1-4	HTTP Request	69000	Success	9276	375	12046	1
21	02:14:32.2	Pengujian SIAKAD UIN	HTTP Request	68005	Success	9276	375	10915	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	96	Malang 1-24	t						
22	02:14:32.217	Pengujian SIAKAD UIN Malang 1-23	HTTP Request	69132	Success	9276	375	10995	0
23	02:14:32.136	Pengujian SIAKAD UIN Malang 1-22	HTTP Request	69469	Success	9276	375	11085	0
24	02:14:32.534	Pengujian SIAKAD UIN Malang 1-27	HTTP Request	69573	Success	9276	375	12109	1
25	02:14:32.615	Pengujian SIAKAD UIN Malang 1-28	HTTP Request	70063	Success	9276	375	12029	1
26	02:14:32.055	Pengujian SIAKAD UIN Malang 1-21	HTTP Request	70644	Success	9276	375	11156	1
27	02:14:32.455	Pengujian SIAKAD UIN Malang 1-26	HTTP Request	71370	Success	9276	375	12195	1
28	02:14:32.695	Pengujian SIAKAD UIN Malang 1-29	HTTP Request	71453	Success	9276	375	12684	1
29	02:14:32.775	Pengujian SIAKAD UIN Malang 1-30	HTTP Request	71845	Success	9276	375	13452	1
30	02:14:32.856	Pengujian SIAKAD UIN Malang 1-31	HTTP Request	71878	Success	9276	375	13377	1
31	02:14:32.937	Pengujian SIAKAD UIN Malang 1-32	HTTP Request	71800	Success	9276	375	13955	1
32	02:14:32.376	Pengujian SIAKAD UIN Malang 1-25	HTTP Request	73895	Success	9276	375	14680	1
33	02:14:33.016	Pengujian SIAKAD UIN Malang 1-33	HTTP Request	73287	Success	9270	375	14492	1
34	02:14:33.257	Pengujian SIAKAD UIN Malang 1-36	HTTP Request	73048	Success	9270	375	15839	1
35	02:14:33.176	Pengujian SIAKAD UIN Malang 1-35	HTTP Request	73129	Success	9276	375	14567	1
36	02:14:33.096	Pengujian SIAKAD UIN Malang 1-34	HTTP Request	73212	Success	9276	375	14646	1
37	02:14:33.737	Pengujian SIAKAD UIN Malang 1-42	HTTP Request	74124	Success	9276	375	16981	1
38	02:14:33.4	Pengujian SIAKAD UIN	HTTP Request	74364	Success	9270	375	15750	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	97	Malang 1-39	t						
39	02:14:33.416	Pengujian SIAKAD UIN Malang 1-38	HTTP Request	74445	Success	9276	375	15822	1
40	02:14:33.578	Pengujian SIAKAD UIN Malang 1-40	HTTP Request	74289	Success	9276	375	15660	0
41	02:14:33.336	Pengujian SIAKAD UIN Malang 1-37	HTTP Request	75431	Success	9276	375	15902	1
42	02:14:33.657	Pengujian SIAKAD UIN Malang 1-41	HTTP Request	75823	Success	9276	375	17066	1
43	02:14:33.978	Pengujian SIAKAD UIN Malang 1-45	HTTP Request	75504	Success	9276	375	18228	1
44	02:14:33.816	Pengujian SIAKAD UIN Malang 1-43	HTTP Request	75677	Success	9270	375	16902	1
45	02:14:33.896	Pengujian SIAKAD UIN Malang 1-44	HTTP Request	76435	Success	9276	375	18310	1
46	02:14:34.056	Pengujian SIAKAD UIN Malang 1-46	HTTP Request	77045	Success	9276	375	18806	1
47	02:14:34.136	Pengujian SIAKAD UIN Malang 1-47	HTTP Request	76987	Success	9276	375	19294	1
48	02:14:34.217	Pengujian SIAKAD UIN Malang 1-48	HTTP Request	76906	Success	9276	375	19458	1
49	02:14:34.296	Pengujian SIAKAD UIN Malang 1-49	HTTP Request	77595	Success	9276	375	19381	1
50	02:14:34.375	Pengujian SIAKAD UIN Malang 1-50	HTTP Request	78231	Success	9270	375	19303	1
51	02:14:34.615	Pengujian SIAKAD UIN Malang 1-53	HTTP Request	78053	Success	9276	375	20437	1
52	02:14:34.776	Pengujian SIAKAD UIN Malang 1-55	HTTP Request	77892	Success	9276	375	20277	1
53	02:14:34.456	Pengujian SIAKAD UIN Malang 1-51	HTTP Request	78214	Success	9276	375	20501	1
54	02:14:35.015	Pengujian SIAKAD UIN Malang 1-58	HTTP Request	79152	Success	9276	375	21503	1
55	02:14:34.6	Pengujian SIAKAD UIN	HTTP Request	79472	Success	9276	375	20356	0

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	96	Malang 1-54	t						
56	02:14:34.857	Pengujian SIAKAD UIN Malang 1-56	HTTP Request	79316	Success	9276	375	21664	0
57	02:14:34.936	Pengujian SIAKAD UIN Malang 1-57	HTTP Request	79240	Success	9276	375	21582	1
58	02:14:34.535	Pengujian SIAKAD UIN Malang 1-52	HTTP Request	79650	Success	9276	375	20527	1
59	02:14:35.176	Pengujian SIAKAD UIN Malang 1-60	HTTP Request	80478	Success	9276	375	22066	1
60	02:14:35.416	Pengujian SIAKAD UIN Malang 1-63	HTTP Request	80238	Success	9276	375	22570	1
61	02:14:35.256	Pengujian SIAKAD UIN Malang 1-61	HTTP Request	80399	Success	9276	375	22729	1
62	02:14:35.095	Pengujian SIAKAD UIN Malang 1-59	HTTP Request	80572	Success	9276	375	21431	1
63	02:14:35.498	Pengujian SIAKAD UIN Malang 1-64	HTTP Request	81028	Success	9276	375	24001	0
64	02:14:35.818	Pengujian SIAKAD UIN Malang 1-68	HTTP Request	81454	Success	9276	375	25591	0
65	02:14:35.737	Pengujian SIAKAD UIN Malang 1-67	HTTP Request	81537	Success	9276	375	26079	1
66	02:14:35.577	Pengujian SIAKAD UIN Malang 1-65	HTTP Request	82113	Success	9276	375	25718	1
67	02:14:35.335	Pengujian SIAKAD UIN Malang 1-62	HTTP Request	82834	Success	9276	375	26770	1
68	02:14:35.898	Pengujian SIAKAD UIN Malang 1-69	HTTP Request	83094	Success	9276	375	27628	1
69	02:14:36.056	Pengujian SIAKAD UIN Malang 1-71	HTTP Request	82977	Success	9276	375	28090	1
70	02:14:35.977	Pengujian SIAKAD UIN Malang 1-70	HTTP Request	83057	Success	9276	375	27667	1
71	02:14:36.138	Pengujian SIAKAD UIN Malang 1-72	HTTP Request	84035	Success	9276	375	28281	1
72	02:14:35.6	Pengujian SIAKAD UIN	HTTP Request	84868	Success	9276	375	29749	0

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	58	Malang 1-66	t						
73	02:14:36.296	Pengujian SIAKAD UIN Malang 1-74	HTTP Request	84266	Success	9276	375	29547	1
74	02:14:36.218	Pengujian SIAKAD UIN Malang 1-73	HTTP Request	84346	Success	9276	375	29484	1
75	02:14:36.377	Pengujian SIAKAD UIN Malang 1-75	HTTP Request	84451	Success	9276	375	30116	1
76	02:14:36.457	Pengujian SIAKAD UIN Malang 1-76	HTTP Request	85593	Success	9276	375	30287	1
77	02:14:36.538	Pengujian SIAKAD UIN Malang 1-77	HTTP Request	85512	Success	9276	375	31173	1
78	02:14:36.699	Pengujian SIAKAD UIN Malang 1-79	HTTP Request	85358	Success	9270	375	31488	0
79	02:14:36.618	Pengujian SIAKAD UIN Malang 1-78	HTTP Request	85444	Success	9270	375	31341	1
80	02:14:36.778	Pengujian SIAKAD UIN Malang 1-80	HTTP Request	86372	Success	9276	375	31992	0
81	02:14:36.857	Pengujian SIAKAD UIN Malang 1-81	HTTP Request	86890	Success	9276	375	32176	1
82	02:14:37.017	Pengujian SIAKAD UIN Malang 1-83	HTTP Request	86732	Success	9276	375	33258	1
83	02:14:36.937	Pengujian SIAKAD UIN Malang 1-82	HTTP Request	86909	Success	9276	375	33096	1
84	02:14:37.099	Pengujian SIAKAD UIN Malang 1-84	HTTP Request	87015	Success	9276	375	35216	0
85	02:14:37.338	Pengujian SIAKAD UIN Malang 1-87	HTTP Request	87981	Success	9276	375	36650	1
86	02:14:37.179	Pengujian SIAKAD UIN Malang 1-85	HTTP Request	88140	Success	9276	375	35368	0
87	02:14:37.257	Pengujian SIAKAD UIN Malang 1-86	HTTP Request	88064	Success	9276	375	35699	1
88	02:14:37.417	Pengujian SIAKAD UIN Malang 1-88	HTTP Request	88746	Success	9276	375	37169	1
89	02:14:37.4	Pengujian SIAKAD UIN	HTTP Request	88935	Success	9276	375	37249	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	98	Malang 1-89	t						
90	02:14:37.577	Pengujian SIAKAD UIN Malang 1-90	HTTP Request	89321	Success	9276	375	37546	0
91	02:14:37.818	Pengujian SIAKAD UIN Malang 1-93	HTTP Request	89082	Success	9276	375	39038	1
92	02:14:37.657	Pengujian SIAKAD UIN Malang 1-91	HTTP Request	89497	Success	9276	375	38209	1
93	02:14:38.137	Pengujian SIAKAD UIN Malang 1-97	HTTP Request	90449	Success	9276	375	41101	1
94	02:14:37.977	Pengujian SIAKAD UIN Malang 1-95	HTTP Request	90609	Success	9276	375	39429	0
95	02:14:38.056	Pengujian SIAKAD UIN Malang 1-96	HTTP Request	90532	Success	9276	375	40323	1
96	02:14:37.897	Pengujian SIAKAD UIN Malang 1-94	HTTP Request	90704	Success	9276	375	39075	0
97	02:14:38.218	Pengujian SIAKAD UIN Malang 1-98	HTTP Request	91396	Success	9276	375	41143	1
98	02:14:37.737	Pengujian SIAKAD UIN Malang 1-92	HTTP Request	92437	Success	9276	375	42758	1
99	02:14:38.377	Pengujian SIAKAD UIN Malang 1-100	HTTP Request	91798	Success	9276	375	42327	1
100	02:14:38.298	Pengujian SIAKAD UIN Malang 1-99	HTTP Request	91878	Success	9276	375	41555	1
101	02:14:38.456	Pengujian SIAKAD UIN Malang 1-101	HTTP Request	92488	Success	9276	375	43001	1
102	02:14:38.618	Pengujian SIAKAD UIN Malang 1-103	HTTP Request	93198	Success	9270	375	44330	1
103	02:14:38.697	Pengujian SIAKAD UIN Malang 1-104	HTTP Request	93120	Success	9270	375	44969	1
104	02:14:38.538	Pengujian SIAKAD UIN Malang 1-102	HTTP Request	93283	Success	9276	375	43047	1
105	02:14:38.857	Pengujian SIAKAD UIN Malang 1-106	HTTP Request	93738	Success	9276	375	45840	1
106	02:14:38.9	Pengujian SIAKAD UIN	HTTP Request	94360	Success	9276	375	46036	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	37	Malang 1-107	t						
107	02:14:39.178	Pengujian SIAKAD UIN Malang 1-110	HTTP Request	94223	Success	9276	375	48116	1
108	02:14:39.017	Pengujian SIAKAD UIN Malang 1-108	HTTP Request	94387	Success	9276	375	46939	0
109	02:14:39.098	Pengujian SIAKAD UIN Malang 1-109	HTTP Request	94808	Success	9276	375	47940	1
110	02:14:39.258	Pengujian SIAKAD UIN Malang 1-111	HTTP Request	95682	Success	9276	375	48215	1
111	02:14:39.417	Pengujian SIAKAD UIN Malang 1-113	HTTP Request	95539	Success	9276	375	49925	1
112	02:14:39.337	Pengujian SIAKAD UIN Malang 1-112	HTTP Request	95620	Success	9276	375	49042	0
113	02:14:38.777	Pengujian SIAKAD UIN Malang 1-105	HTTP Request	96187	Success	9276	375	49174	1
114	02:14:39.498	Pengujian SIAKAD UIN Malang 1-114	HTTP Request	96691	Success	9276	375	50112	0
115	02:14:39.659	Pengujian SIAKAD UIN Malang 1-116	HTTP Request	96854	Success	9270	375	50662	0
116	02:14:39.578	Pengujian SIAKAD UIN Malang 1-115	HTTP Request	96935	Success	9276	375	50176	0
117	02:14:39.738	Pengujian SIAKAD UIN Malang 1-117	HTTP Request	96778	Success	9276	375	51867	1
118	02:14:39.818	Pengujian SIAKAD UIN Malang 1-118	HTTP Request	97446	Success	9276	375	52067	1
119	02:14:40.218	Pengujian SIAKAD UIN Malang 1-123	HTTP Request	97841	Success	9270	375	54050	1
120	02:14:40.139	Pengujian SIAKAD UIN Malang 1-122	HTTP Request	97923	Success	9270	375	53938	0
121	02:14:39.978	Pengujian SIAKAD UIN Malang 1-120	HTTP Request	98098	Success	9270	375	52844	0
122	02:14:39.897	Pengujian SIAKAD UIN Malang 1-119	HTTP Request	98392	Success	9276	375	52235	1
123	02:14:40.0	Pengujian SIAKAD UIN	HTTP Request	99418	Success	9276	375	57661	1

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
	57	Malang 1-121	t						
124	02:14:40.378	Pengujian SIKAD UIN Malang 1-125	HTTP Request	99097	Success	9276	375	56235	1
125	02:14:40.299	Pengujian SIKAD UIN Malang 1-124	HTTP Request	99177	Success	9276	375	56124	1