

IMPLEMENTASI DAN ANALISIS PERFORMANSI PLATFORM AS A SERVICE UNTUK API GATEWAY MENGGUNAKAN KONG

Implementation And Analysis Platform As A Service Performance For Api Gateway Use Kong

Risang Suryadi Saputra¹, Dr. Ir. Rendy Munadi, M.T.², Danu Dwi Sanjoyo, S.T., M.T.³

^{1,2,3} Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Jln. Telekomunikasi No. 1 Terusan Buah Batu Bandung 40257 Indonesia

¹Risang.pro@gmail.com ²Rendy_munadi@yahoo.co.id ³Danudwj@telkomuniversity.ac.id

ABSTRAK

Untuk manajemen banyaknya API atau server pada sebuah instansi dibutuhkan suatu sistem yang dapat manajemen API tersebut agar dapat terhubung dengan client atau konsumen tanpa perlu memikirkan infrastruktur dan mempermudah dalam mengakses data pada api tersebut.

Oleh karena itu, dibutuhkan sebuah sistem yang dapat memenejemen atau mengontrol API yang ada pada sebuah sistem adalah API gateway. API gateway adalah sebuah sistem yang digunakan sebagai pengatur atau mengontrol API yang ada pada sebuah server, API gateway dapat manajemen API sehingga API dapat terdistribusikan lebih optimal.

Pada tugas akhir ini telah diimplementasikan dan dianalisa sistem API gateway berbasis cloud computing dengan menggunakan platform as a service (PaaS). Dari hasil penelitian yang dilakukan diketahui bahwa kinerja sistem API gateway lebih baik pada jaringan intranet dibandingkan dengan jaringan internet, karena beberapa faktor yang mempengaruhi hasil pengujian dari jaringan internet yaitu topologi yang ada pada jaringan dan kecepatan transmisi yang digunakan. Berdasarkan hasil pengujian yang telah dilakukan, maka didapatkan nilai rata-rata dari beberapa parameter yaitu: *response time* 5.17 sec, *data transaction* 4288 hit, *data transferred* 351 MB dan *availability* 100%.

Kata kunci: *Cloud Computing, API gateway, Server, PaaS*

ABSTRACT

To manage how many APIs or servers in an agency, a system is required for managing APIs to connect with clients or consumers without the need to think about infrastructure and make data accessing easier.

Therefore, it takes a system that can manage or control the API that exist in a system. API Gateway is a system used as a controller on an existing API on a server. API Gateway can manage the API so that API can be distributed more optimally.

In this final project, API Gateway will be implemented and analyzed with cloud gateway-based cloud computing system using Platform as a Service (PaaS). From the results of research, it has been known that the performance of API Gateway system is better on the Intranet network that is compared with the Internet network, because there are several factors that affect the test results of the Internet network which is the topology that exists on the network and transmission speed that had been used. Based on the results of tests, the average value of some parameters are: response time 5.17 sec, data transaction 4288 hit, data transferred 351 MB and availability 100%.

Keywords: *Cloud Computing, API gateway, Server, PaaS*

1. Pendahuluan

1.1 Latar Belakang Masalah

Penggunaan server pada instansi saat ini sangat diperlukan dan bahkan diperbanyak agar dapat memenuhi penyimpanan data pada setiap node atau cabang instansi. Banyaknya server dapat terhubung dengan menggunakan cloud computing. Cloud computing merupakan teknologi informasi dimana informasi disimpan pada suatu server yang dapat diakses oleh client melalui jaringan internet. Informasi diakses melalui perangkat pengguna seperti smartphone, komputer, dan lainnya. Cloud computing menggabungkan teknologi komputer dengan pengembangan internet sehingga menjadi infrastruktur kompleks yang abstraksi dan tersembunyi. Pengguna tidak perlu direpotkan infrastrukturnya, karena kemampuan teknologi ini disajikan sebagai suatu layanan (as a service).

Cloud computing memiliki tiga layanan salah satunya platform as a service (PaaS). PaaS menawarkan kepada instansi, suatu metode untuk membangun sebuah aplikasi tanpa harus melakukan investasi besar untuk membangun infrastruktur, khususnya perangkat keras komputer atau server maupun perangkat lunak yang diperlukan. Dengan PaaS instansi tidak perlu melakukan konfigurasi teknologi yang diperlukan satu-persatu dan memikirkan cara menggabungkannya menjadi satu. PaaS telah menyiapkan teknologi secara lengkap yang dibutuhkan untuk membangun dan mengoperasikan sebuah aplikasi.

Seiring pesatnya pertumbuhan cloud computing dalam beberapa tahun terakhir ini, dalam pembangunan sebuah aplikasi membutuhkan lebih banyak server untuk meningkatkan service atau layanan. Sehingga muncullah permasalahan pada sebuah instansi yang besar yaitu kebocoran data dan keribetan dalam pengelolaan database dalam data JSON. Solusi dalam permasalahan ini adalah perlunya sebuah platform untuk manajemen data API atau API gateway.

Salah satu solusinya adalah penggunaan API gateway. API gateway adalah antar muka yang menghubungkan aplikasi dari beberapa aplikasi. API gateway biasa disebut gerbang dari API yang menghubungkan beberapa data API dan dapat memajemen data API yang ada pada sebuah instansi untuk meminimalisir permasalahan yang ada pada sebuah instansi tersebut.

Dalam tugas akhir ini telah diimplementasikan dan dianalisis kinerja dari API gateway pada Platform as a service dengan menggunakan KONG dengan parameter yang akan diuji meliputi response time, throughput, transaction data, data transferred dan availability.

2. Dasar Teori

2.1 Cloud Computing

Cloud computing adalah sebuah metode komputasi awan yang membuat *resource Information and Technology* (IT) seperti banyak server dan aplikasi mudah di akses melalui internet kapanpun dan dimanapun sebagai layanan yang dapat digunakan secara umum atau *private*.

2.2 Software as a Service (SaaS)

Software As A Service (SaaS) adalah sebuah layanan dimana pengguna hanya diberikan layanan aplikasi. Adapun sumber daya yang diberikan sudah dibatasi oleh penyedia layanan sesuai dengan kebutuhan aplikasi yang digunakan oleh pengguna. Contohnya adalah google apps, Microsoft dynamics, dll[3].

2.3 Platform as a Service (PaaS)

Platform As A Service (PaaS) adalah jenis layanan diatas SaaS dimana pengguna diberikan hak untuk mengakses komponen-komponen yang dibutuhkan untuk mengembangkan aplikasinya melalui internet. Jika dianalogikan, PaaS memungkinkan pengguna menyewa "rumah" beserta lingkungannya (Sistem operasi, *networks*, *database engine*, *framework* aplikasi, dll)[3].

2.4 Infrastructure as a Service (IaaS)

Infrastructure As A Service (IaaS) adalah sebuah layanan dimana pengguna diberikan hak untuk menyewa layanan berupa sumber daya atau infrastruktur secara penuh seperti *processor*, *memory*, *storage* dan *bandwidth*. Oleh karena itu pengguna dapat memanfaatkan seluruh sumber daya secara penuh sesuai dengan yang pengguna beli dari penyedia layanan IaaS seperti Amazon EC2 dan Windows Azure [3].

2.5 Docker

Docker adalah platform yang bersifat *open source* yang pertama kali dikembangkan oleh Solomon hykes sebagai proyek *internal* dotCloud. Docker merupakan sebuah aplikasi yang berfungsi sebagai wadah atau biasanya disebut *container* untuk memasukkan *software* secara lengkap dan dapat berfungsi dimana saja. Docker memberikan *environment* yang cepat dan ringan dimana kode bisa dijalankan secara efisien. Docker sangat simpel menggunakan konfigurasi sederhana dan dapat dikonfigurasi dengan *kernel* linux[4].

2.6 API Gateway

Application programming interface (API) merupakan antarmuka pemrograman ke aplikasi yang menyediakan cara untuk menghubungkan, mengintegrasikan dari perangkat satu ke perangkat yang lain. API menyediakan beberapa sarana dalam mengakses layanan dan data pada sebuah instansi penyedia API, yang menghubungkan hal-hal tentang bagaimana layanan atau dapat ditemukan, data menjadi tersedia dan bisa berinteraksi.

Api gateway merupakan gerbang dari beberapa API yang berfungsi sebagai manajemen API, melindungi data, dan meminimalisir *downtime* API. *API gateway* ini sebenarnya komponen *network* yang

bisa berbentuk *hardware* maupun *virtual* yang bertindak sebagai *proxy* sehingga API tidak perlu berinteraksi dengan aplikasi klien[7].

2.7 Kong

Kong adalah lapisan API *open source* yang skalabel (juga dikenal sebagai *API Gateway*, atau *API Middleware*) dan dibangun di atas *Nginx*. Kong merupakan aplikasi *Lua* yang berjalan di *Nginx* dan didukung oleh *lua-Nginx-module*. Kong tersendiri mendukung pengimplementasi abstraksi *database* dan *routing*. Pada kong terdapat *plugin* yang dapat digunakan sesuai kebutuhan pengembang seperti sistem autentikasi dan *logging plugin* yang digunakan untuk menyimpan *log* pada suatu komunikasi dengan aplikasi yang lain[8].

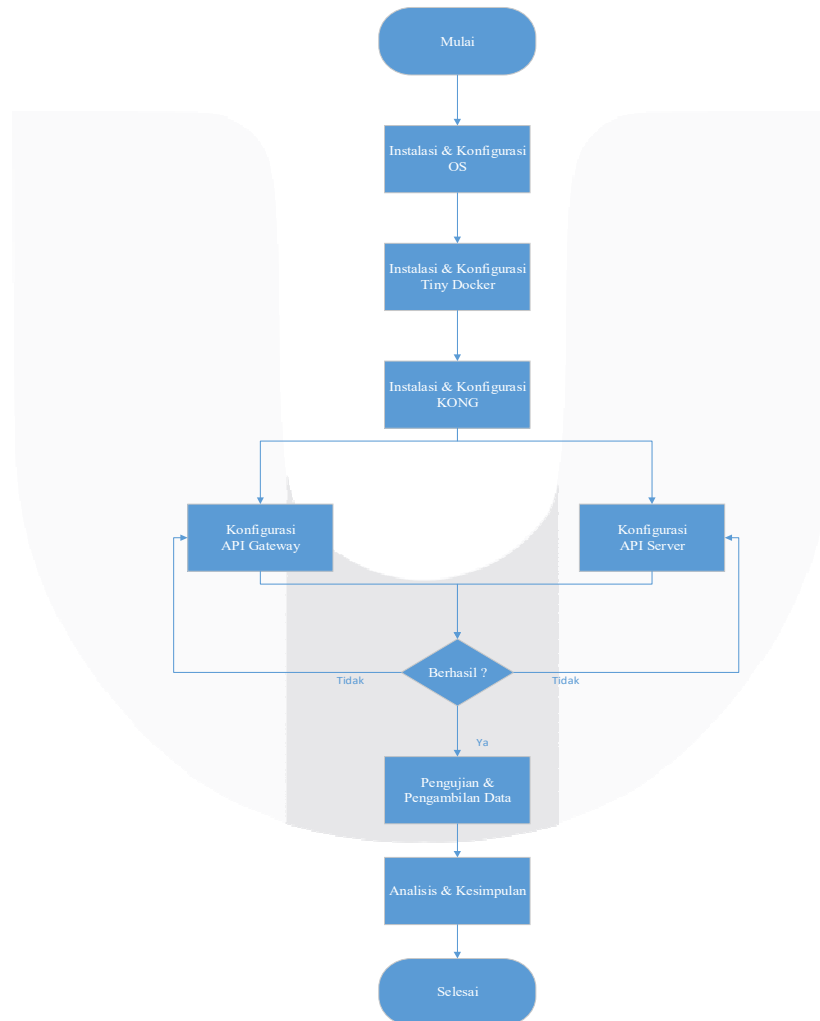
2.8 VM (Virtual Machine)

Virtual Machine adalah sebuah mesin yang konsep dasarnya terdiri dari lapisan-lapisan sistem komputer. Sehingga sistem dapat dibangun dari lapisan-lapisan yang ada, adapun urutan lapisannya adalah perangkat keras, kernel, sistem program. Kernel pada lapisan tersebut menggunakan instruksi dari perangkat keras untuk menciptakan perangkat *system call* yang akan digunakan oleh komponen-komponen yang ada pada sistem program. *Virtual machine* ini biasanya digunakan untuk pembelajaran karna dengan *virtua machine* kita dapat menginstall operating sistem lain di dalam operating sistem.[9]

3. Perancangan Sistem

3.1 Perancangan Sistem

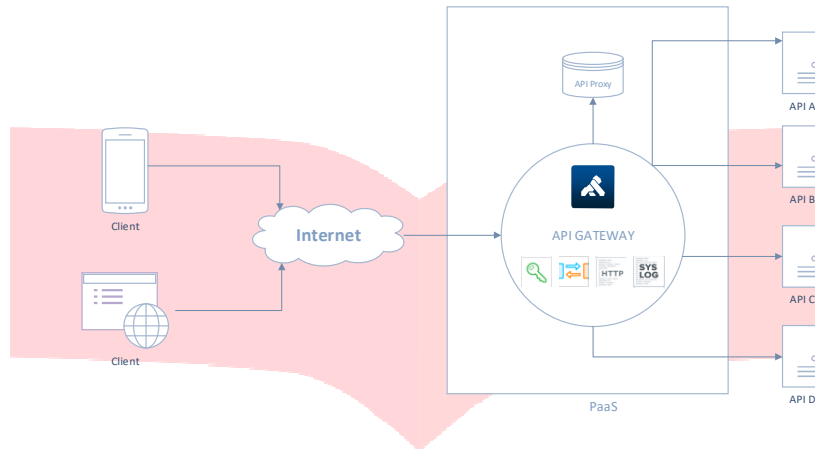
Berikut ini merupakan gambaran umum dari sistem yang akan dirancang dan diimplementasikan pada penelitian ini adalah :



Gambar 1. Diagram Alir Perancangan Sistem

3.2 Diagram Sistem

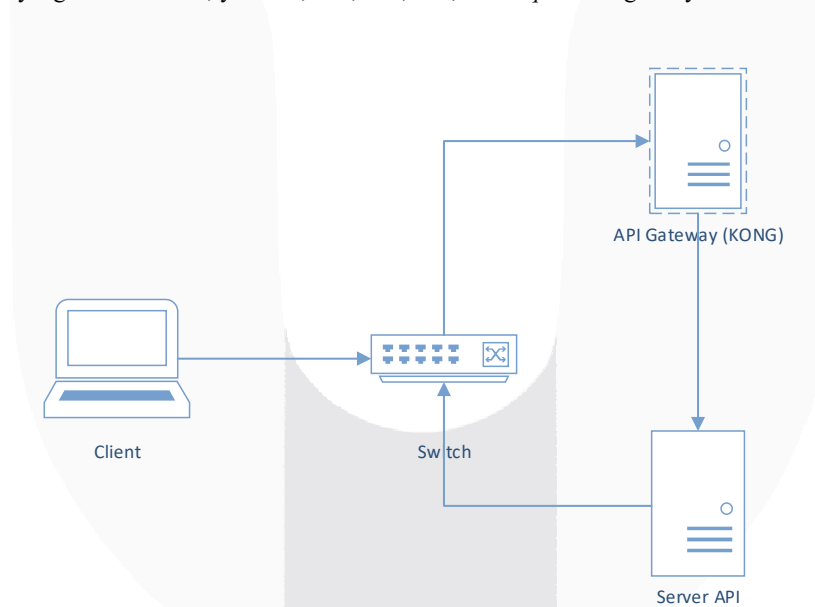
Gambar berikut adalah diagram sistem yang akan dibuat dengan menggunakan satu *server rancher*, tiga *node host rancher* dan satu *server database*.



Gambar 2. Diagram Sistem

3.3 Skenario dengan jaringan intranet

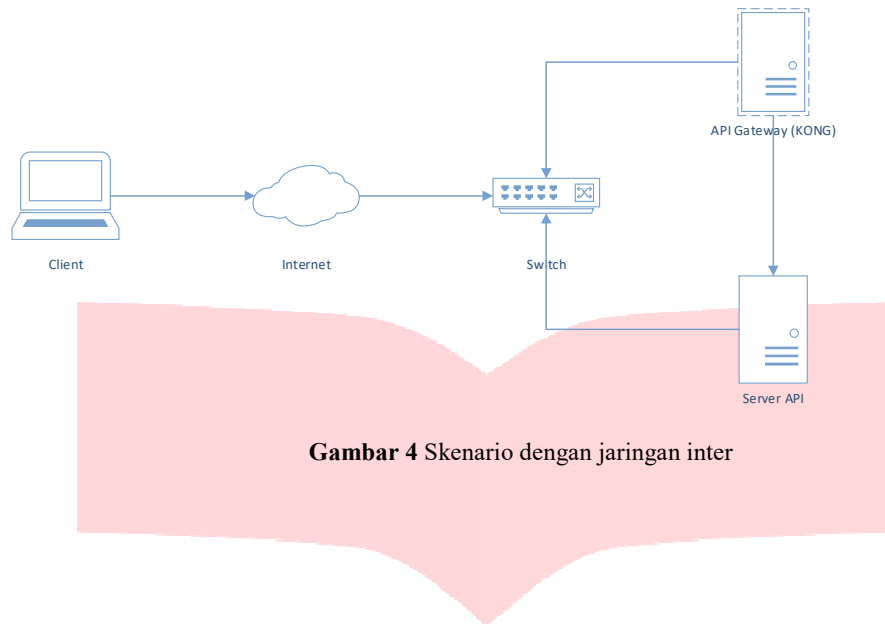
Pada skenario ini akan dilakukan pengujian untuk melihat bagaimana kinerja atau performansi dari sebuah *server API gateway* dengan menggunakan jaringan lokal atau intranet, skenario ini mengirimkan *request* dengan total yang berbeda-beda, yaitu 50, 100, 150, 200, 225 *request* dengan layanan data API.



Gambar 3 Skenario dengan jaringan intranet

3.4 Skenario dengan jaringan internet

Pada skenario ini akan dilakukan pengujian untuk melihat bagaimana kinerja atau performansi dari sebuah *server API gateway* dengan menggunakan jaringan internet atau jaringan *public*, skenario ini mengirimkan *request* dengan total yang berbeda-beda, yaitu 50,100, 150 dan 200 *request* dengan layanan data API sama seperti yang sebelumnya hanya perbedaannya dari segi jaringan saja.



Gambar 4 Skenario dengan jaringan inter

4 Hasil Pengujian dan Analisis

4.1 Pengujian Dengan Jaringan Intranet

Tabel 1. Hasil Pengujian jaringan intranet

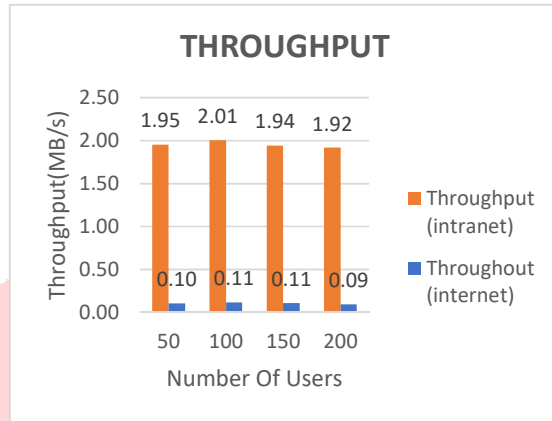
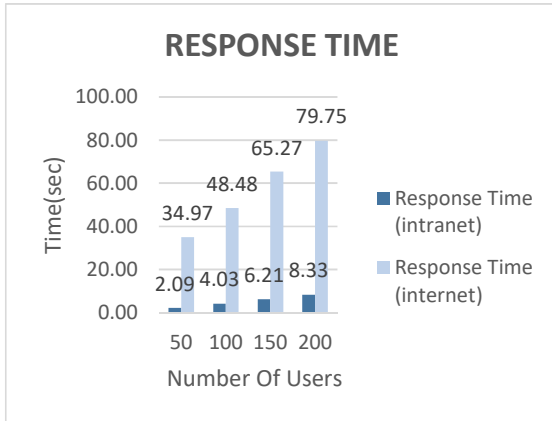
user	Transactions (hits)	Data Transferred (MB)	Responses Time (secs)	Throughput (MB/sec)	Availability (%)
50	4282	351	2.09	1.95	100
100	4401	360	4.03	2.01	100
150	4263	349	6.21	1.94	100
200	4209	345	8.33	1.92	100
225	3636	295	10.46	1.63	74.17

4.2 Pengujian Dengan Jaringan Internet

Tabel 2. Hasil Pengujian jaringan internet

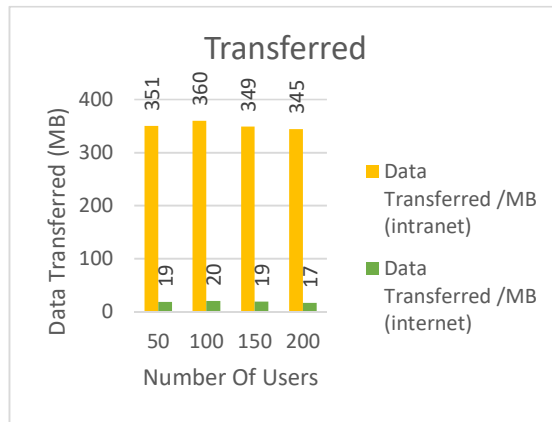
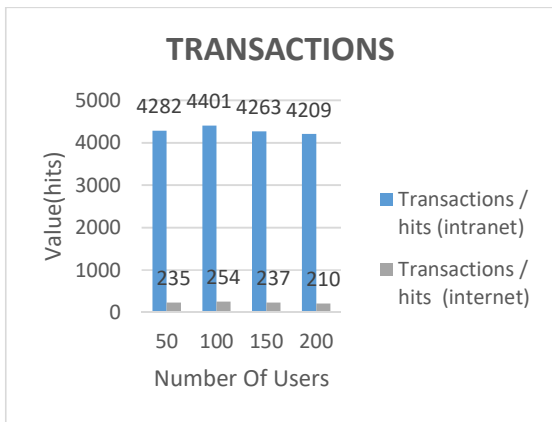
user	Transactions (hits)	Data Transferred (MB)	Responses Time (secs)	Throughput (MB/sec)	Availability (%)
50	235	19	34.97	0.10	100
100	254	20	48.48	0.11	100
150	237	19	65.27	0.11	100
200	210	17	79.75	0.09	100

4.3 Analisis Hasil Jaringan Intranet dengan internet



Gambar 4 Perbandingan hasil *Response Time*

Gambar 5 Perbandingan hasil *Throughput*



Gambar 6 Perbandingan hasil *Transaction*

Gambar 7 Perbandingan hasil *Transferred*

Dari gambar 4 terlihat bahwa hasil *response time* jaringan internet tinggi dibandingkan dengan jaringan intranet, dengan nilai rata – rata pada jaringan internet 57 sec dan pada jaringan intranet 5 sec . Hal ini dikarenakan ada beberapa faktor yang mempengaruhi nilai *response time* seperti jumlah pengguna dan kecepatan transmisi ataupun topologi jaringan yang digunakan. Karena beberapa faktor tersebut menyebabkan *response time* yang besar, dan terlihat hasil jaringan intranet lebih baik dari jaringan internet.

Dari gambar 5 hasil pengukuran secara keseluruhan menunjukkan nilai *throughput* lebih besar saat menggunakan jaringan intranet dengan menggunakan jaringan internet dengan nilai rata-rata 0.10 MB/sec dan jaringan intranet 1.96 MB/sec . Dikarenakan dari segi kecepatan transmisi atau *bandwith* dan dari segi topologi jaringan yang digunakan maupun tidak stabilnya jaringan ISP yang digunakan menyebabkan *throughput* yang sangat kecil dibandingkan dengan nilai *throughput* menggunakan jaringan intranet.

Pada gambar 6 dan gambar 7 terlihat bahwa ketika nilai *throughput* tinggi maka nilai transaksi data dan data *transferred* akan tinggi. Dilihat dari hasil rata – rata transaksi data yang ada pada jaringan internet 234 hit dan 4289 hit untuk jaringan intranet, maka hasil data Ketika nilai *throughput* tinggi artinya data yang diproses pada sistem dalam jumlah yang banyak, dikarenakan transaksi data yang ada pada sistem tinggi.

5 Kesimpulan

Dari hasil implementasi tugas akhir ini serta pengukuran dan pengujian sistem API *gateway* pada *platform as a service*, maka dapat diambil kesimpulan sebagai berikut :

1. Kemampuan server API *gateway* dalam melayani jumlah *request* yang banyak
2. Semakin besar jumlah *user* maka semakin besar juga nilai CPU utilization dan *response time* yang didapat
3. Semakin tinggi nilai *throughput* semakin tinggi juga nilai *transaction* data, *transferred* data dan nilai *request time* semakin kecil. Terlihat dari nilai *throughput* tertinggi 2.01 MB dengan nilai *transaction* data 4401 hit dan nilai *transferred* 360 MB.

DAFTAR PUSTAKA

- [1] R. Reza, A. N. Jati, U. A. Ahmad, F. T. Elektro, and U. Telkom, “(Platform As a Service) Sebagai Penyimpan Data Pengunjung Dan Rekaman Video Pengunjung Pada Building Security System Berbasis Embedded Design and Implementation Cloud Server Paas (Platform As a Service) As Visitor ’ S Database and Visitor ’ S Video Da,” vol. 3, no. 1, pp. 661–665, 2016.
- [2] D. Beimborn, T. Miletzki, and S. Wenzel, “PLATFORM AS A SERVICE (PaaS) SEBAGAI LAYANAN SISTEM OPERASI CLOUD COMPUTING,” *Bus. Inf. Syst. Eng.*, vol. 3, no. 6, pp. 381–384, 2011.
- [3] B. Kahanwal and T. Pal Singh, “The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle,” *Int. J. Latest Res. Sci. Technol.*, vol. 1, no. 2, pp. 183–187, 2012.
- [4] B. Bashari Rad, H. John Bhatti, and M. Ahmadi, “An Introduction to Docker and Analysis of its Performance,” *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 17, no. 3, pp. 228–235, 2017.
- [5] M. F. R. Bik, “IMPLEMENTASI DOCKER UNTUK PENGELOLAAN BANYAK APLIKASI WEB (Studi Kasus : Jurusan Teknik Informatika UNESA) Asmunin Abstrak,” vol. 7, no. Vm, pp. 46–50, 2017.
- [6] “Docker overview.” [Online]. Available: <https://docs.docker.com/engine/docker-overview/#docker-architecture>. [Accessed: 01-Nov-2017].
- [7] “API Management Basics.” [Online]. Available: <http://www.apiacademy.co/resources/api-management-101-api-management-basics/>. [Accessed: 01-Nov-2017].
- [8] “Kong (Secure, Manage and Orchestrate Microservice APIs).” [Online]. Available: <https://konghq.com/>. [Accessed: 01-Nov-2017].
- [9] “VIRTUAL MACHINE,” 2008. [Online]. Available: <https://simponi.mdp.ac.id/materi201120121/SP314/111061/SP314-111061-638-5.pdf>. [Accessed: 10-May-2018].