



Analisa Performa RStudio Server Berbasis *Cloud* Menggunakan *Elastic Stack* sebagai Sistem Manajemen Metrik

Krisna Aditama Ashari^{#1}, Is Mardianto^{#2}, Dedy Sugiarto^{#3}

[#]Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Trisakti
Jalan Kyai Tapa No 1 Grogol, Jakarta Barat, DKI Jakarta 11440

¹krisna064001904003@std.trisakti.ac.id

²mardianto@trisakti.ac.id

³dedy.sugiarto@trisakti.ac.id

Abstrak— Reliabilitas atau keandalan merupakan salah satu sifat penting pada sebuah server dalam melayani pengguna. Salah satu cara mengukurnya ialah dengan melakukan uji performa. Penelitian ini bertujuan untuk mengetahui kemampuan RStudio Server pada infrastruktur *cloud* saat digunakan oleh multiuser dengan *Elastic Stack* sebagai sistem yang menangani pengumpulan, penyimpanan dan visualisasi data metriknya. Tahapan dimulai dengan mengumpulkan data berupa metrik sistem oleh Metricbeat, lalu diproses Logstash dan disimpan menjadi index dalam Elasticsearch, visualisasi data ditampilkan oleh Kibana. Pengujian kinerja server dilakukan dengan menjalankan script R berdurasi 2 menit dan 7 menit secara simultan. Hasil pengujian berupa catatan CPU Usage, Memory Usage dan durasi penyelesaian script selanjutnya di plotting pada R. Hasil analisa dari plotting data menunjukkan jumlah user yang dapat menggunakan Rstudio Server dengan spesifikasi 2 CPU dan RAM 4GB secara optimal ialah maksimal 2 user pada script dengan run time 2 menit dan 7 menit, lebih dari jumlah user itu akan mempengaruhi waktu proses penyelesaian script menjadi tingkat performa sedang hingga berat.

Kata kunci— Rstudio Server, *Cloud*, *Elastic Stack*, Metrik, Plotting

I. PENDAHULUAN

Server adalah komputer yang menyediakan berbagai jenis layanan kepada komputer lain dalam jaringan komputer [1]. Seiring pesatnya perkembangan teknologi, layanan server kini mampu diakses lebih luas dan fleksibel melalui internet berbasis *cloud computing*. *Cloud computing* atau komputasi awan adalah metode dimana kemampuan teknis teknologi disediakan sebagai layanan (*as a service*) sehingga pengguna dapat menggunakannya melalui internet tanpa mengetahui infrastruktur apa yang ada di dalamnya [2]. Salah satu contoh penerapannya pada saat ini ialah mendukung pembelajaran dan pelatihan via online mengenai data analisis menggunakan RStudio Server. RStudio Server adalah server *integrated development environment* (IDE) berbasis web untuk bahasa

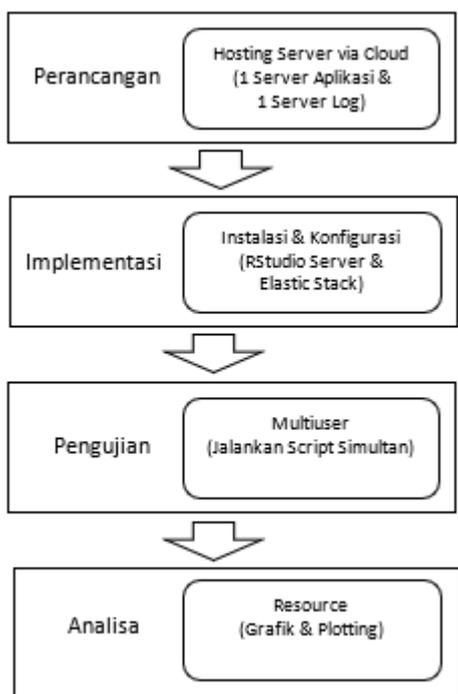
R, yang memungkinkan untuk tiap penggunanya mengakses *environment* sendiri kapanpun melalui antarmuka web [3]. Dengan RStudio Server berbasis server *cloud* memberikan banyak kemudahan di sisi pengguna, dimana kebutuhan spesifikasi hardware seperti CPU, RAM dan lainnya telah diakomodir oleh server yang menaungi. Karena peran server *cloud* yang sentral inilah analisa performa server diperlukan dalam mengetahui berapa maksimum pengguna yang dapat dilayani secara optimal dari spesifikasi server yang dimiliki. Sehingga nantinya dapat menyesuaikan alokasi infrastruktur *cloud* yang digunakan dengan jumlah pengguna aplikasi yang dapat dilayani. Untuk itu dibutuhkan sebuah sistem yang mampu memonitoring performa server dalam memberikan layanan, terutama berkaitan dengan data metrik sistem. Seperti penelitian lain terkait yang menggunakan Network Monitoring System untuk memantau sumber daya virtual server [4] dan analisis performansi metrik pada Windows Azure VM dan AWS EC2 [5].

Sebagai sistem manajemen metriknya digunakan Elastic Stack. *Elastic Stack* adalah kumpulan perangkat lunak sumber terbuka yang berguna untuk mencari, menganalisis, dan memvisualisasikan log yang dihasilkan dari sumber apa pun dalam format apa pun [6]. *Elastic Stack* terdiri dari empat komponen yang berdiri sendiri, yaitu Logstash, Beats, Elasticsearch, dan Kibana [7]. Implementasi *Elastic Stack* di server adalah solusi yang tepat untuk manajemen log dan metrik karena dapat menampilkan informasi tentang tren, statistik, dan anomali kepada administrator [8]. Beberapa penelitian sebelumnya mengenai penggunaan *Elastic Stack* antara lain, Implementasi *Elastic Stack* sebagai pemroses data IoT [9], penelitian memantau kinerja server web apache dalam menangani request user [10], penelitian lain juga menggunakan *Elastic Stack* sebagai sistem pengolah big data [11]. Plotting data sendiri bertujuan untuk memvisualisasikan variasi atau menunjukkan hubungan antar variabel [12].

Berdasarkan latar belakang diatas maka mengetahui penggunaan resources server dengan melakukan uji performa diperlukan untuk mempermudah dalam menentukan skema penggunaan aplikasi sehingga didapatkan hasil yang optimal.

II. METODE PENELITIAN

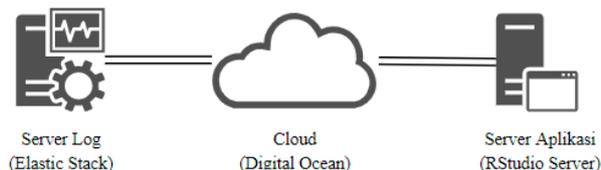
Metode yang digunakan pada penelitian ini terdiri atas beberapa tahapan sebagai berikut:



Gambar 1. Diagram alur

A. Perancangan

Dalam penelitian ini untuk infrastruktur menggunakan resources salah satu provider cloud computing, yaitu Digital Ocean. Diperlukan 2 server yang masing-masing difungsikan sebagai server manajemen metrik dan server aplikasi, seperti yang ditunjukkan Gambar 2



Gambar 2. Desain arsitektur

Server Log menggunakan Elastic Stack yang akan mengolah data metrik dari Server Aplikasi. Pada Server Aplikasi digunakan RStudio Server untuk mengolah data dengan bahasa pemrograman r.

TABEL I
SPESIFIKASI SERVER

No	Komponen	Keterangan
1	Processor	2 CPU
2	RAM	4 GB
3	HDD	80 GB
4	Lokasi	Singapura
5	Tarif	\$20 / Bulan

. Kedua server menggunakan spesifikasi server yang sama dengan detail komponen seperti yang ditunjukkan Tabel I, untuk sistem operasi yang dipakai ialah Ubuntu 20.04. Akses server yang berada di cloud menggunakan tool tambahan yaitu Putty dengan akses ke IP Public server menggunakan protokol SSH dan login password yang telah di set saat pembuatan akun Digital Ocean.

B. Implementasi

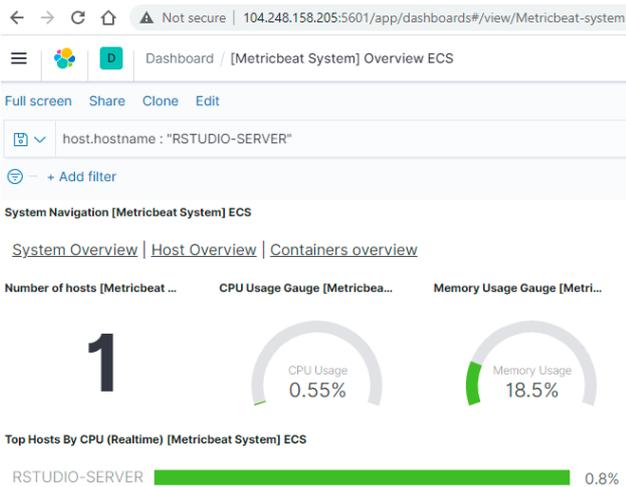
Komponen Elastic Stack yang digunakan pada sisi server manajemen log dan metrik ialah Logstash, Elasticsearch dan Kibana. Logstash merupakan perangkat lunak sumber terbuka untuk mengumpulkan dan menguraikan log kemudian menyimpannya di Elasticsearch [13]. Logstash difungsikan untuk menerima input data dari Metricbeat yang berada pada server aplikasi untuk selanjutnya diteruskan menuju Elasticsearch dengan terlebih dahulu mendefinisikan penamaan index agar disimpan berdasarkan tanggal tiap harinya. Konfigurasi ini dibuat pada satu file khusus dengan nama logstash.conf yang disimpan dalam folder /etc/logstash/conf.d/. Port default yang digunakan Logstash ialah 5044. Hasil output Logstash akan disimpan dalam bentuk index oleh Elasticsearch. Elasticsearch adalah mesin pencari dan analisis teks open source yang sangat skalabel yang memungkinkan pengguna untuk menyimpan, mencari, dan menganalisis data dalam jumlah besar dengan cepat [14]. Kibana menjadi user interface dari Elasticsearch dimana manajemen index log berada didalamnya [15]. Konfigurasi Kibana berada pada directory /etc/kibana/kibana.yml, koneksi dengan Elasticsearch dan port yang dipakai di atur dalam file ini. Visualisasi grafik dan dashboard yang dipakai untuk memantau data metrik server aplikasi dapat menggunakan sample yang disediakan oleh assets module. Akses Kibana pada browser dengan alamat http://IP_Public_Server_Log:5601.

Pada server aplikasi yang digunakan untuk menjalankan RStudio Server dipasang Beats, Beats adalah agen untuk pengirim data tujuan tunggal [16]. Salah satu jenis Beats ialah Metricbeat. Metricbeat akan meneruskan metrik sistem operasi dan juga statistik layanan yang berjalan di sistem ke Elasticsearch atau Logstash secara berkala [17]. Data metrik tersebut dapat berupa CPU, memori dan service yang berjalan didalamnya. Pastikan output.logstash pada file konfigurasi di /etc/metricbeat/metricbeat.yml menuju IP milik server Elastic Stack.



Gambar 7. Dashboard kibana

Pada Dashboard yang ada dapat dipantau tingkat konsumsi sumber daya server dari tiap sesi ujicoba menjalankan script R. Selain itu tersedia pula *timeline* untuk memudahkan dalam menganalisanya. *Dashboard* [Metricbeat System] Overview ECS berisi visualisasi grafik dari jumlah host yang di monitor, CPU Usage, Memory Usage, Disk Usage dan keluar masuk trafik.



Gambar 8. Monitoring server aplikasi

Pada Gambar 8 menunjukkan data metrik dari server aplikasi (RSTUDIO-SERVER) telah masuk dan dapat dipantau melalui grafik pada server manajemen metrik (ELASTIC-STACK).

B. Pengujian

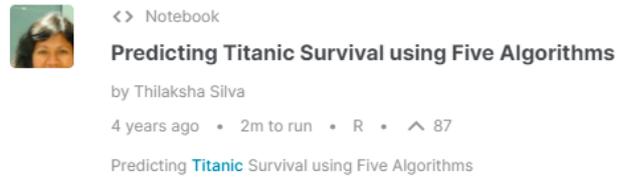
Pengujian pertama penggunaan RStudio dengan menjalankan *script* R dengan durasi *run time* selama 2 menit atau 120 *second*. Menggunakan *dataset titanic.csv* yang dibagi menjadi data training dan data test. Seperti yang ditunjukkan Gambar 9.

	A	B	C	D	E	F	G	H	I	J
1	PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked									
2	1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S									
3	2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C									
4	3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S									
5	4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S									
6	5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S									
7	6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q									
8	7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S									
9	8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S									

Gambar 9. Dataset

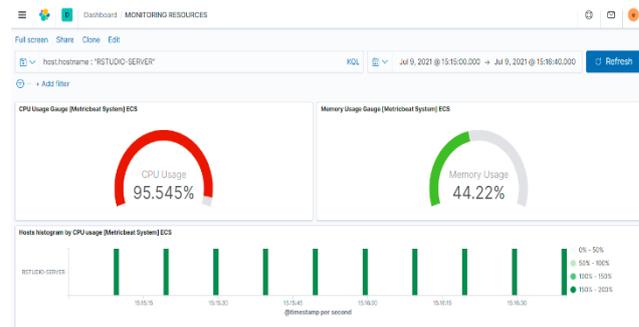
Dataset berisi data penumpang kapal Titanic dengan berbagai detail informasi seperti golongan kelas, nomor kabin, umur, status dan data pendukung lainnya yang dapat di analisa lebih jauh. Untuk *coding* program R dengan tema *Predicting Titanic Survival using Five Algorithms* seperti ditampilkan Gambar 10, ditulis oleh Thialaksha Silva dari

situs kaggle dengan *link* sumber berikut: <https://www.kaggle.com/thialakhasilva/predicting-titanic-survival-using-five-algorithms>.



Gambar 10. Sumber *script* R pertama

RStudio diakses oleh masing-masing user dan di set jadwal untuk *running* script tiap sesinya bergantian oleh 1 user, 2 user, hingga 5 user. Data metrik dari server aplikasi saat menjalankan script R akan terlihat pada server *Elastic Stack* seperti yang ditunjukkan Gambar 11.



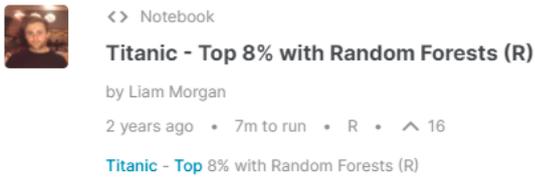
Gambar 11. Data metrik dari 2 user

Data penggunaan *resources* server yang tercatat pada seluruh sesi ditampilkan Tabel III.

TABEL III
DATA RESOURCES

No	Jumlah User	CPU Usage (%)	Memory Usage (%)	Durasi Selesai (sec)
1	1 User	46,73	25,33	100
2	2 User	92,34	44,44	100
3	3 User	93,48	59,85	150
4	4 User	95,35	75,17	200
5	5 User	95,17	93,02	260

Pengujian kedua masih menggunakan *dataset* yang sama namun dengan script R yang memiliki *run time* 7 menit / 420 *second* yang ditulis oleh Liam Morgan pada situs kaggle dengan judul *Titanic – Top 8% with Random Forests (R)* seperti yang ditampilkan pada Gambar 12, sumber terkait dapat diakses melalui *link* : <https://www.kaggle.com/lmorgan95/titanic-top-8-with-random-forests-r>.



Gambar 12. Sumber script R kedua

Hasil pada pengujian kedua dicatat dan dirangkum seperti yang ditunjukkan pada Tabel IV:

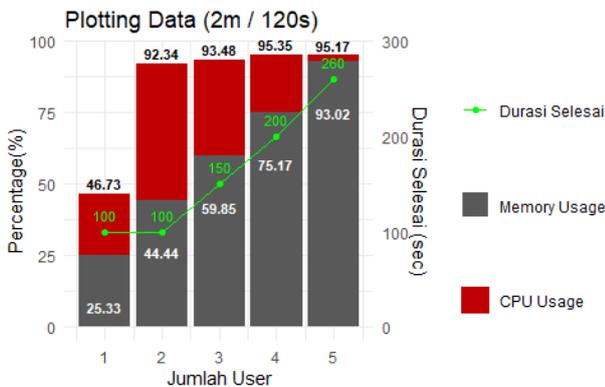
TABEL IV
DATA RESOURCES

No	Jumlah User	CPU Usage (%)	Memory Usage (%)	Durasi Selesai (sec)
1	1 User	50,10	32,64	380
2	2 User	99,29	53,96	380
3	3 User	98,05	69,98	640
4	4 User	98,61	82,21	790
5	5 User	98,23	95,56	990

Dari pengujian kedua didapat pola yang sama dengan pengujian pertama, dimana durasi penyelesaian script menjadi lebih panjang secara signifikan pada penggunaan aplikasi lebih dari 2 user.

C. Analisa

Untuk memudahkan analisa data yang telah dihimpun dari skenario pengujian sebelumnya dan melihat pengaruh yang ditimbulkan maka perlu dibuat grafiknya. Data diolah dengan bahasa R pada RStudio menggunakan library ggplot2 yang menyediakan beragam kebutuhan visualisasi. Data Jumlah User ditempatkan sebagai sumbu x, untuk sumbu y menampilkan data persentase dari CPU Usage dan Memory Usage, ditambahkan pula secondary axis untuk data Durasi Selesai. Pada grafik pengujian pertama di Gambar 13 menunjukkan server aplikasi mampu melayani 2 user bersamaan dengan durasi penyelesaian script yang dijalankan 100s atau pada tingkat optimal.



Gambar 13. Plotting data pengujian 1

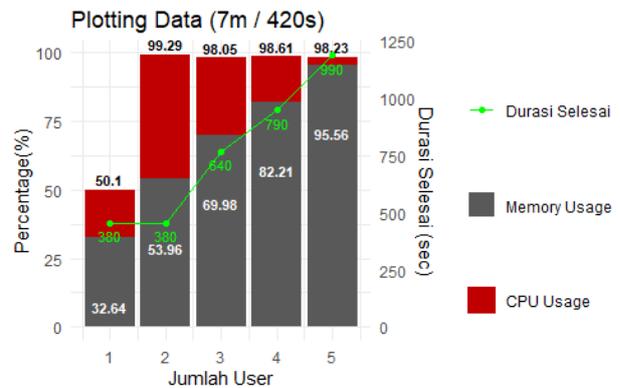
Saat dipakai oleh 3 user dan 4 user, mulai terjadi peningkatan durasi selesai yaitu 150s dan 200s atau tingkat sedang. Untuk 5 user simultan waktu yg dibutuhkan untuk menyelesaikan yaitu 260s, masuk tingkat berat. Lebih dari 5 user terdapat 1 atau 2 task yang tidak diselesaikan (killed). Dibuat 3 tingkatan performa mulai dari optimal, sedang dan berat, keterangan lebih lengkap pada Tabel V.

TABEL V
TINGKATAN PERFORMA RUN TIME 120S

No	Tingkat	Keterangan	Max. User
1	Optimal	Durasi selesai dibawah run time	2
2	Sedang	Durasi selesai tidak lebih 2x run time	4
3	Berat	Durasi selesai lebih 2x run time	> = 5

Berdasarkan hasil durasi waktu yang diperlukan sistem menyelesaikan script di pengujian pertama, dapat dikelompokkan menjadi 3 tingkatan performa dimana hasil optimal penggunaan aplikasi yaitu penyelesaian dibawah run time didapat saat maksimal 2 user secara simultan menjalankan script.

Pada grafik pengujian kedua di Gambar 14 menunjukkan server aplikasi mampu melayani 2 user bersamaan dengan durasi penyelesaian script yang dijalankan 100s atau berada pada tingkat optimal.



Gambar 14. Plotting data pengujian 2

Saat pengguna bertambah menjadi 3 user dan 4 user, mulai terjadi peningkatan durasi selesai yaitu 150s dan 200s atau tingkat sedang. Durasi waktu penyelesaian script meningkat signifikan menjadi 990s saat digunakan oleh 5 user sekaligus, dalam tingkatan performa dikategorikan berat. Hasil lengkap pada Tabel VI.

TABEL VI
TINGKATAN PERFORMA RUN TIME 420S

No	Tingkat	Keterangan	Max. User
1	Optimal	Durasi selesai dibawah run time	2
2	Sedang	Durasi selesai tidak lebih 2x run time	4
3	Berat	Durasi selesai lebih 2x run time	> = 5

