Analisis Kinerja Elasticsearch Pada Proses Query Data

Elasticsearch Performance Analysis Of Data Query Processes

Eddy Tungadi [1)], Meylanie Olivya[2)] , Suwesti Akbar[3)]

Jurusan Teknik Elektro, Politeknik Negeri Ujung Pandang
Jl. Perintis Kemerdekaan km. 10 Tamalanrea, Makassar, 90245, Telp/Fax: 0411-585368

E-mail: Eddy.tungadi@poliupg.ac.id [1)], meylanie@poliupg.ac.id [2)], suwestiakbar@gmail.com[3)]

Abstrak - Elasticsearch adalah mesin pencari Restful terdistribusi, yang didasarkan pada perpustakaan perangkat lunak pengambilan informasi. Elasticsearch berbeda dari sistem manajemen basis data relasional klasik (RDBMS). Model basis data utama Elasticsearch adalah mesin pencari dan menyimpan dokumen daripada nilai-nilai kunci. Setiap dokumen di Elasticsearch adalah objek *JavaScript Object Notation* (JSON) dan karenanya tidak menggunakan *Scripted Query Language* (SQL). Data yang selalu bertambah akan diikuti dengan peningkatan masalah didalamnya. Misalnya penggunaan *query* yang tidak tepat dapat menghambat kinerja basis data yang sudah ada. Pada penelitian ini dilakukan analisis kinerja Elasticsearch terhadap proses *query* dengan menggunakan sistem yang telah dibangun untuk mengelola data serta mengukur penggunaan CPU, *harddisk*, dan RAM terhadap *query*. Hasil pengujian dari penelitian ini menunjukkan bahwa peningkatan data yang besar mempengaruhi waktu eksekusi pada proses *insert*, *select*, dan *update*. adapun untuk operasi *delete* mengalami inkonsisten. Sementara itu, penggunaan CPU, *harddisk*, dan RAM tertinggi terjadi pada saat proses *insert* dan *update*.
Kata Kunci: *query, elasticsearch,* kinerja

Abstract – *Elasticsearch is a distributed Restful search engine, which is based on an information retrieval software library. Elasticsearch is different from the classic relational database management system (RDBMS). Elasticsearch's main database model is a search engine and stores documents rather than key values. Every document in Elasticsearch is a JavaScript Object Notation (JSON) object and therefore does not use Scripted Query Language (SQL). Data that is always increasing will be followed by problems increasing with the data. For example the use of incorrect queries can interrupt the performance of existing databases.In this study an Elasticsearch performance analysis was performed on the query process by using a system that was built to manage data and measure CPU, hard disk, and RAM usage on the query. Test results from this study indicate that a large increase in data affects the execution time of the insert, select, and update processes. as for the delete operation inconsistent. Meanwhile, the highest CPU, hard disk and RAM usage occurs during the insert and update process.*
Keywords: *query, elasticsearch, performance*

PENDAHULUAN

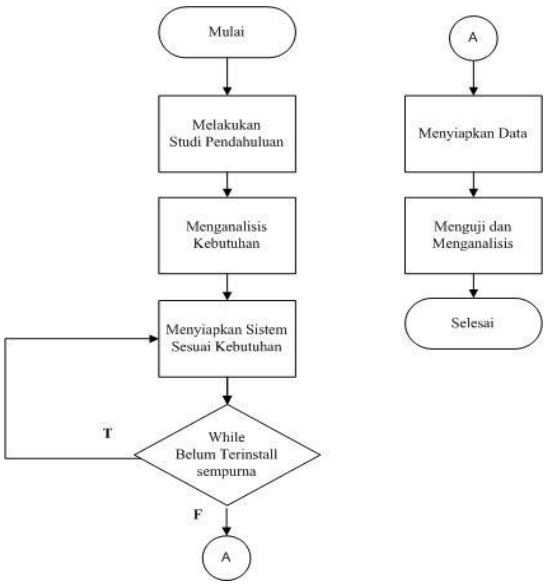
Elasticsearch adalah mesin pencari Restful terdistribusi, yang didasarkan pada perpustakaan perangkat lunak pengambilan informasi *Lucene* (Bialecki et al, 2012) dan mampu memecahkan semakin banyak kasus penggunaan. Elasticsearch berbeda dari sistem manajemen basis data relasional klasik (RDBMS) dalam banyak hal: Model basis data utama Elasticsearch adalah mesin pencari dan menyimpan dokumen daripada nilai-nilai kunci. Setiap dokumen di Elasticsearch adalah objek *JavaScript Object Notation* (JSON), dan karenanya tidak menggunakan *Scripted Query Language* (SQL). Pertanyaan disediakan dengan bahasanya sendiri berdasarkan JSON. Pencarian yang diberikan dapat dilakukan tidak hanya dalam bentuk *query*, filter juga dapat digunakan untuk pencarian dokumen yang lebih cepat dari permintaan. Terakhir, skema bebas yaitu dua dokumen dengan tipe yang sama dapat memiliki kumpulan bidang yang berbeda (Kononenko et al, 2014).

Penelitian tentang Elasticsearch baru-baru ini muncul karena kebaruan pada struktur *query*. Kononenko et al. (2014) berdiskusi tentang bagaimana Elasticsearch berbeda dari basis data relasional yang masih tradisional dan memberikan beberapa implementasi nyata menggunakan *query* Elasticsearch. Penelitian yang dilakukan oleh Reskiana (2017) dengan judul penelitian Studi Performansi Aplikasi Database Nosql Dan RDBMS yaitu dengan menguji performansi NoSQL dan RDBMS dalam hal respond time query, penggunaan RAM, penggunaan CPU, dan penggunaan harddisk. Adapun Hasil dari penelitian yang di dapatkan dari penetian ini yaitu bertambah besarnya data yang akan diolah pada aplikasi basis data

NoSQL berarti semakin lama respond time query, besarnya alokasi atau presentasi penggunaan CPU dan RAM serta transfer data pada harddisk. Demikian juga respond time query, penggunaan RAM, penggunaan CPU, dan penggunaan harddisk ditentukan oleh banyaknya data yang akan diolah pada aplikasi basis data RDBMS.

Penelitian lain yang dilakukan oleh Novitasari (2015) dengan judul penelitian Information Retrieval Using Elasticsearch On Multidimensional Unstructured Datatext. Pada penelitian ini, penulis mengusulkan sebuah teknik temu kembali informasi berbasis big data menggunakan ElasticSearch pada lingkungan komputasi terdistribusi yang menggunakan Hadoop. Hadoop sendiri adalah framework software berbasis Java dan opensource yang berfungsi untuk mengolah data yang sangat besar secara terdistribusi dan berjalan di atas cluster yang terdiri dari beberapa komputer yang saling terhubung. Dengan diusulkannya penggunaan elasticsearch pada penelitian ini, diharapkan nantinya teknik ini akan meningkatkan kinerja temu kembali informasi berbasis big data dalam hal kecepatan dan akurasi pencarian. Berdasarkan latar belakang tersebut, maka dilakukan penelitian untuk menguji performansi Elasticsearch dalam hal *respond time query*, penggunaann (*Random Access Memory*) RAM, penggunaan *Central Processing Unit* (CPU) dan penggunaan *harddisk* dengan menentukan variasi jumlah data yang akan diuji serta menggunakan empat parameter atau operasi yaitu *insert*, *select*, *update*, dan *delete.*

METODOLOGI PENELITIAN



Gambar 1 Tahapan Pelaksanaan Penelitian

Untuk mencapai target sesuai dengan tujuan penelitian, diperlukan tahapan sistematis dan terencana agar penelitian dapat terstuktur. Penjelasan mengenai tahapan penelitian yang dilaksanakan, digambarkan pada Gambar 1.

1. Studi Literatur

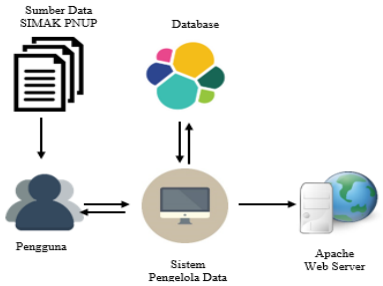
Tahapan awal yakni studi pendahuluan dan pemahaman dasar teori melalui pengumpulan referensi dimana data-data dapat diperoleh dari hasil membaca buku, jurnal penelitian serta referensi-referensi yang dianggap relevan dan berhubungan dengan judul yang diangkat. Sebagian besar informasi yang diperoleh pada tahapan ini akan menjadi bahan evaluasi pada akhir penelitian.

2. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi perangkat yang dibutuhkan sehingga sesuai dengan perancangan sistem yang dibangun, seperti kebutuhan hardware dan software.

3. Perancangan Sistem

Pada tahapan ini dilakukan perancangan arsitektur sistem mengenai sistem Pengelola Data pada Elasticsearch secara konseptual yang bertujuan sebagai gambaran umum perancangan sistem yang telah dibangun. Pada tahapan ini dilakukan perancangan sistem secara konseptual yang bertujuan sebagai gambaran umum perancangan sistem yang telah dibangun. Pada Gambar 2 menjelaskan arsitektur secara umum mengenai Sistem Pengelola Data pada Elasticsearch.



Gambar 2 Arsitektur Sistem

Sistem ini dibangun sesuai dengan kebutuhan peneliti. Dengan menggunakan data dari Sistem Informasi Akademik Politeknik Negeri Ujung Pandang (SIMAK PNUP). Selanjutnya, data yang diperoleh disimpan dalam file dengan format *JSON*.

Sistem ini terhubung dengan Elasticsearch yang merupakan tempat penyimpanan data. Dalam sistem, data diproses dan menghasilkan jawaban sesuai dengan

query yang diminta. Sistem Pengelola data ini masih bersifat *local* yakni dengan menggunakan *apache web server*.

4. Instalasi dan Konfigurasi

Elasticsearch diunduh secara gratis melalui website resminya. Pada penelitian ini, Elasticsearch versi 6.4.2 diinstall pada Windows 10 menggunakan port 9200. Dalam melakukan konfigurasi, ditentukan beberapa batasan dalam hal pengelolaan Elasticsearch.

5. Persiapan Data

Data penelitian diperoleh dari SIMAK PNUP dengan menggunakan aplikasi Postman. Pada Postman kita pilih operasi Get kemudian memasukkan alamat API SIMAK PNUP dan get_ipk yang berfungsi untuk mendapatkan data Indeks Prestasi Kumulatif (IPK) mahasiswa.

6. Uji dan Analisis

Pengujian dilakukan terhadap Elasticsearch dengan variasi size data meliputi 10 MB, 20 MB, 50 MB, 70 MB, dan 100 MB dengan menggunakan empat operasi yaitu insert, select, update, dan delete. Dimana 1 MB sama dengan 750 record yang berarti 10 MB berisi 7.500 record dan seterusnya. Setiap jenis operasi tersebut diuji dengan beberapa parameter yaitu respond time query, penggunaan CPU, penggunaan harddisk, dan penggunaan RAM.

HASIL DAN PEMBAHASAN

Pada penelitian ini, dibangun sistem untuk mengelola data dengan melakukan proses *query* terhadap Elasticsearch. Hasil pengujian disajikan dalam bentuk tabel dan grafik. Penelitian difokuskan untuk mendapatkan kesimpulan dari analisis data hasil pengujian. Pengujian dilakukan dengan cara menguji kinerja Elasticsearch pada proses *query* dengan variasi ukuran data meliputi 10 MB, 20 MB, 50 MB, 70 MB, dan 100 MB dengan menggunakan empat operasi yaitu *insert, select, update,* dan *delete* dengan empat parameter yaitu *respond time query*, penggunaan CPU, penggunaan *harddisk*, serta penggunaan RAM.

a. *Respond time query*

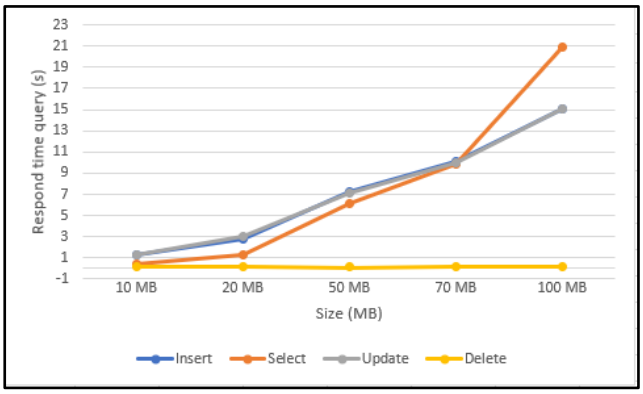
```
$info = curl_getinfo($ch);
echo 'Took ' . $info['total_time'] . ' seconds to transfer a request to ' . $info['url'];
```

Gambar 3 *Script Respond Time Query*

curl_getinfo() memberikan *output* mengenai waktu atau informasi transfer tertentu. Dengan fungsi tersebut, kode menjadi lebih singkat sehingga memudahkan penulis selama proses penelitian. Pada pengujian ini dilakukan sebanyak lima kali percobaan untuk setiap operasi terhadap variasi *size* data. Berikut ini adalah tabel dan grafik dari *respond time query* yang didapatkan. Gambar 4 merepresentasikan hasil pengujian yang ada pada tabel 1. Dari grafik tersebut terlihat bahwa operasi *insert* dan *update* memiliki nilai *respond time query* yang tidak jauh berbeda dan semakin meningkat seiring bertambahnya jumlah data yang dimasukkan. Pada operasi select, nilai *respond time query* meningkat cukup signifikan pada data 100 MB. Sementara itu, nilai *respond time query* pada operasi delete mengalami inkonsisten.

Tabel 1 Hasil pengujian *respond time query*

Operasi	*Respond Time Query* (s)				
	10 MB	**20 MB**	**50 MB**	**70 MB**	**100 MB**
Insert	1.347	2.840	7.284	10.134	15.100
Select	0.421	1.234	6.122	9.862	20.867
Update	1.324	2.968	7.162	10.044	15.112
Delete	0.184	0.190	0.144	0.190	0.203



Gambar 4 Grafik *respond time query*

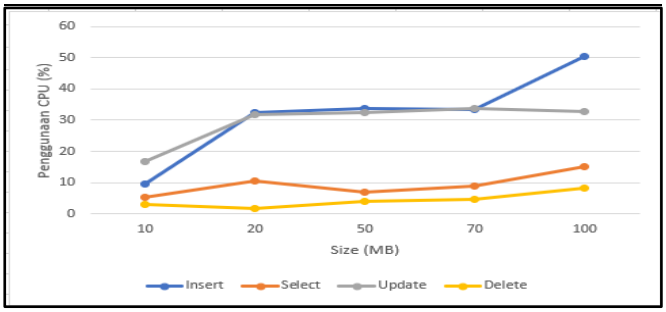
Penggunaan CPU

Pengujian kedua yang dilakukan ialah memonitoring penggunaan CPU selama suatu proses atau operasi berlangsung. Hasil data yang didapatkan selama proses berlangsung akan di rata-ratakan untuk mendapatkan hasil akhir dari penggunaan CPU. Pengujian ini dilakukan sebanyak lima kali percobaan setiap operasi dan ukuran data seperti Gambar 5. Hasil pengujian yang ada pada Tabel 2 menunjukkan nilai penggunaan CPU yang dihasilkan pada operasi *insert* mengalami peningkatan yang signifikan pada data 20 MB dan mulai stabil pada data 50 MB dan 70 MB kemudian kembali meningkat signifikan pada data 100 MB. Pada operasi *select*, nilai penggunaan CPU mengalami fluktuasi. Sementara itu, pada operasi *update* nilai penggunaan CPU dari data 10 MB mengalami peningkatan cukup signifikan pada data 20 MB dan kembali stabil pada data 50 MB dan 70 MB

kemudian mengalami penurunan pada data 100 MB. Nilai penggunaan CPU untuk operasi *delete* mengalami penurunan pada data 20 MB dan kembali stabil pada data 50 MB.

Tabel 2 Hasil Pengujian CPU

Operasi	Penggunaan CPU (%)				
	10 MB	20 MB	50 MB	70 MB	100 MB
Insert	9.6	32.6	33.6	33.4	50.4
Select	5.4	10.4	6.8	8.8	15.2
Update	16.6	31.8	32.6	33.6	32.8
Delete	3	1.8	4	4.8	8.4



Gambar 5 Grafik Penggunaan CPU

b. Penggunaan *Harddisk*

Pengujian ke empat adalah melakukan *monitoring* besarnya transfer *write* dan *read* data pada *harddisk* per detik selama suatu operasi query berlangsung. Hasil data yang didapatkan selama proses berlangsung akan di rata-ratakan untuk mendapatkan hasil akhir dari penggunaan *harddisk*. Pengujian ini dilakukan sebanyak lima kali percobaan untuk setiap operasi dan kapasitas data.

Berikut ini adalah tabel dan grafik dari penggunaan *harddisk* yang didapatkan:

Tabel 3 Hasil Pengujian *Harddisk*

Operasi		Penggunaan *Harddisk* (B/Sec)				
		10 MB	20 MB	50 MB	70 MB	100 MB
Insert	*(write)*	11,631	11,575	14,967,910	35,861,299	53,564,047
Select	*Write*	11,630	11,645	11,540	9,485	12,102
	read	0	0	0	1,566,208	0
Update	*(write)*	11,557	11,616	11,497,171	35,651,584	42,991,616
Delete	*(write)*	11,223	11,630	11,690	11,515	11,541

Gambar 6 Grafik Penggunaan *Harddisk*

Gambar 6 merepresentasikan hasil pengujian yang ada pada Tabel 3. Nilai penggunaan *harddisk* yang dihasilkan pada operasi *insert* dan *update* mengalami peningkatan seiring dengan pertambahan data. Peningkatan penggunaan *harddisk* meningkat signifikan pada data 50 MB. Pada operasi *select* dan *delete* nilai penggunaan *harddisk* mengalami penurunan pada data 70 MB dan kembali meningkat pada data 100 MB.

c. Penggunaan RAM

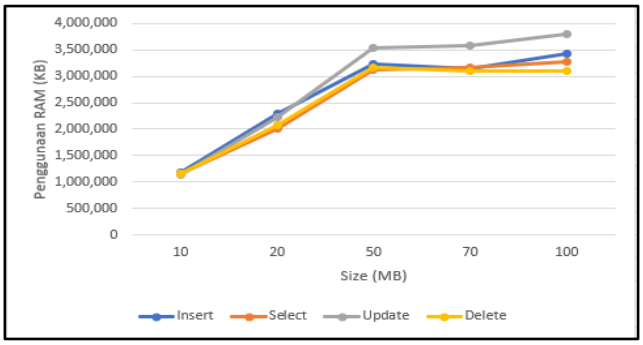
Pengujian ke empat yang dilakukan adalah memonitoring penggunaan RAM per detik selama suatu operasi *query* berlangsung. Hasil data yang didapatkan selama proses berlangsung dirata-ratakan untuk mendapatkan hasil akhir dari penggunaan harddisk dengan melalui lima kali percobaan untuk setiap operasi dan kapasitas data.

Berikut ini adalah tabel grafik dari penggunaan RAM yang didapatkan:

Tabel 4 Hasil Pengujian Penggunaan RAM

Operasi	Penggunaan RAM (KB)				
	10 MB	20 MB	50 MB	70 MB	100 MB
Insert	1,189,117	2,285,291	3,220,022	3,133,900	3,416,587
Select	1,157,658	2,002,950	3,116,653	3,168,414	3,279,221
Update	1,145,121	2,229,645	3,528,293	3,581,081	3,791,510
Delete	1,158,453	2,066,104	3,158,113	3,089,456	3,099,588

Gambar 7 merepresentasikan hasil pengujian yang ada pada Tabel 4. Nilai penggunaan RAM pada operasi *insert* mengalami penurunan pada data 70 MB dan kembali meningkat pada data 100 MB.



Gambar 7 Grafik Penggunaan RAM

Pada operasi *select*, nilai penggunaan RAM mengalami peningkatan seiring dengan pertambahan data. Sementara itu, nilai penggunaan RAM pada operasi *update* mengalami peningkatan cukup signifikan pada data 50 MB dan meningkat seiring dengan pertambahan data. Adapun utuk operasi *delete*, nilai penggunaan RAM mengalami penurunan pada data 70 MB dan 100 MB.

KESIMPULAN

Berdasarkan dari hasil pengujian pada penelitian yang dilakukan, performansi Elasticsearch pada proses *query* disimpulkan bahwa nilai *respond time query* untuk operasi *select* cenderung lebih cepat dibandingkan dengan operasi *insert* dan *update*. Operasi *select* membutuhkan waktu sebesar 0.421 s, sedangkan untuk operasi *insert* dan *update* membutuhkan waktu sebesar 1.347 s dan 1.324 s.

Presentasi penggunaan CPU, *harddisk*, dan RAM tertinggi terjadi pada saat proses *insert* dan *update* dijalankan.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada Tuhan Yang Maha Esa, kedua orang tua penulis, dan seluruh pihak yang telah membantu dalam pembuatan tulisan ini.

DAFTAR PUSTAKA

Bialecki, R. Muir, G. Ingersoil. (2012). “Apache Lucene 4”, in Proc. SIGIR 2012 Workshop on Open Source Information Retrieval, Portland, Oregon USA, pp. 17-24.

Gormley, C. & Tong, Z. (2015). Elastic Search: The Definitive Guide, O’Reilly Media Inc., Sebaastopol.

O. Kononenko, O. Baysal, R. Holmes, M. W. Godfrey. 2014. Mining modern repositories with elasticsearch. In Proc. 11th Working Conference n Mining Software Repositories (MSR 2014), Hyderabad, India, pp. 328–331, DOI: 10.1145/2597073.2597091.

Novitasari, R. (2015). Temu kembali informasi menggunakan elasticsearch pada unstructured datatext multidimensi. Institut Teknologi Sepuluh Nopember.

Paro, Alberto.(2013). *Elasticsearch Cookbook*, PACKT Publishing, Birmingham.

Postdot Technology (2017). Postman is the most complete API Development Environment [Online]. San Francisco. Tersedia pada: https://www.getpostman.com/postman (Diakses pada 10 Agustus 2019)

Reskianan, Z. (2017). Studi Performansi Aplikasi Database NoSQL dan RDBMS.

Sianturi, Maruasas. (2015). Apa Itu Kinerja. Tersedia pada:https://www.kompasiana.com/maruasas/552ff08f6ea83413698b46f0/apa-itu%20kinerja (Diakses pada 08 Juli 2019).

Techopedia.(2019). Resource Monitor (RESMON). Tersedia pada: https://www.techopedia.com/definition/13179/resource-monitor-resmon (Diakses: 27 Juni 2019)