Journal of Informatics and Computer Engineering Research (JICER) ISSN:
Vol. 1, No. 1, Juni 2024, hlm. 01-06

KINERJA HYBRID MONGODB-ELASTICSEARCH PADA APLIKASI SOCIAL NETWORK ANALYSIS

Muhammad Jaury, Meylanie Olivya, Rini Nur
Teknik Informatika dan Komputer, Politeknik Negeri Ujung Pandang, Indonesia

Info Artikel

Riwayat artikel:
Received, (1 Juni 2024)
Revised, (15 Juni 2024)
Accepted, (22 Juni 2024)

Kata kunci:
SNA;
MongoDB;
Elasticsearch;
Hybrid;
Performance.

ABSTRACT

Social media data is growing rapidly in size, variety and complexity. Social media data stores various potential information such as sentiment analysis, trend predictions etc. Potential information can be extracted through Social Network Analysis. SNA has a major challenge which is to process very large datasets in a reasonable time. One of the efforts that can be done is create hybrid system of MongoDB and Elasticsearch using social media datasets from Twitter. The results of this study that the highest response time in insert process starting from 26.2s on 1K data to 19520.45s on 1M data. The replication process with 1K tweet data is 6.25s to 1M tweets is 2817,146s. The select process has under 0.1s and relatively constant due to the Inverted Index on Elasticsearch. Highest CPU performance in process of selecting data from Elasticsearch. Highest RAM performance in the insert process to MongoDB and data replication to Elasticsearch.

ABSTRAK

Data media sosial berkembang pesat dalam ukuran, variasi, dan kompleksitas. Data media sosial menyimpan berbagai informasi seperti analisis sentimen, prediksi, tren, dll. Informasi dapat diekstraksi melalui analisis Jaringan Sosial. SNA memiliki tantangan besar yaitu memproses kumpulan data yang sangat besar dalam kurun waktu tertentu. Salah satu upaya yang dapat dilakukan adalah membuat sistem hybrid MongoDB dan Elasticsearch menggunakan dataset media sosial dari Twitter. Hasil dari penelitian ini bahwa *response time* tertinggi pada proses insert dimulai dari 26.2s pada data 1K hingga 19520.45s pada data 1M. Proses replikasi dengan data 1K tweet adalah 6,25 detik menjadi 1M tweet adalah 2817.146 detik. Proses pemilihan memiliki waktu di bawah 0,1 detik dan relatif konstan karena *Inverted Index* pada Elasticsearch. Performa CPU tertinggi dalam proses pemilihan data dari Elasticsearch. Performa RAM tertinggi dalam proses penyisipan ke MongoDB dan replikasi data ke Elasticsearch.

Penulis Koresponsdensi:
Meylanie Olivya
Teknik Infomatika dan Komputer, Politeknik Negeri Ujung Pandang, Jl. Perintis Kemerdekaan KM. 10, Makassar, 90245
Email: meylanie@poliupg.ac.id

1. PENDAHULUAN

Media sosial merupakan aplikasi berbasis internet yang dapat digunakan untuk berbagi informasi dan ide-ide kreatif melalui saluran jaringan komunikasi [1]. Data media sosial tumbuh dengan cepat dalam ukuran, variasi dan kompleksitas, namun data tersebut disimpan dalam keadaan tidak terstruktur[2]. Data media sosial tersebut menyimpan berbagai potensi informasi seperti analisis sentimen, prediksi *trend* pada masyarakat dan lain sebagainya. Potensi informasi ini hanya dapat diekstrak melalui analisis data (*data mining* dan *machine learning*). Salah satu proses ekstraksi informasi media sosial dalam bidang komputer biasa juga disebut dengan *Social Network Analysis (SNA).*

SNA merupakan area penelitian multidisiplin yang menyatukan ilmu sosial dan ilmu komputer yang mempelajari perilaku pengguna media sosial. *SNA* telah menjadi salah satu bidang penelitian yang paling banyak dipelajari dalam beberapa tahun terakhir oleh para peneliti dari berbagai disiplin ilmu. *SNA* memiliki dua tantangan besar: (1) memproses dataset yang sangat besar dalam waktu yang wajar, (2) mengintegrasikan

beberapa dataset yang berbeda menjadi yang lebih besar dan konsisten secara semantic [3]. Dua tantangan tersebut menjadi kendala masalah yang paling banyak ditemukan dalam aplikasi *SNA* karna membutuhkan performansi database yang cepat dari sisi *create* dan *read*. Oleh karena itu, penting dalam pemilihan dan penggunaan *database NOSQL* yang memiliki keunggulan untuk menjawab permasalahan tersebut.

Pada penelitian ini akan merancang dan menguji sistem *hybrid database NOSQL* yaitu *Elasticsearch* dan *MongoDB*. Performansi *hybrid database* tersebut akan diuji menggunakan dataset media sosial untuk kebutuhan *SNA* yang diperoleh dari API Stream Twitter. Kedua *NOSQL* tersebut dipilih karena memiliki struktur yang sama yaitu *document oriented* sesuai dengan *retrieve data* dari Twitter. Alasan lainnya karena memiliki keunggulan yang sesuai dengan kebutuhan untuk mengatasi permasalahan yang ada dan memiliki popularitas yang tinggi untuk setiap model *database* pada situs DB-Engines [4]. Diharapkan hasil dari penelitian ini dapat menjadi rujukan dalam penggunaan *hybrid database NOSQL* pada kasus aplikasi *SNA*.

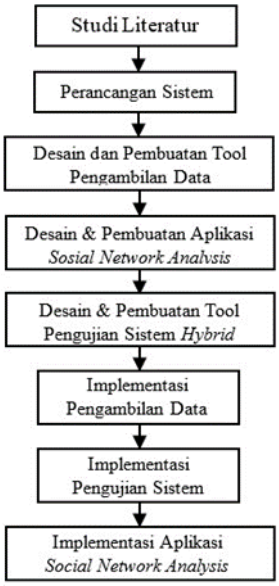
2. METODE

2.1. Alat dan Bahan Penelitian

Perangkat yang digunakan untuk penelitian dikategorikan menjadi perangkat keras *(hardware)* dan perangkat lunak *(software)*. Spesifikasi *hardware* yang digunakan adalah MSI, Core I7 9th, RAM 8 GB untuk *Database*, *Web Server* dan ASUS, Intel Core i3, 8GB RAM untuk pengujian performa *NOSQL*. Adapun kebutuhan *software* yang diperlukan antara lain *Visual Studio Code, Python, PHP, MongoDB, Elasticsearch, Compose Transporter.*

2.2. Metode Penelitian

Agar penelitian yang dilakukan dapat berjalan dengan baik dan terstruktur diperlukan sebuah prosedur penelitian sehingga hasil yang diperoleh sesuai dengan tujuan penelitian. Penjelasan mengenai tahapan proses penelitian yang dilakukan digambarkan pada Gambar 1.



Gambar 1. Tahapan Proses Penelitian

2.3. Perancangan sistem

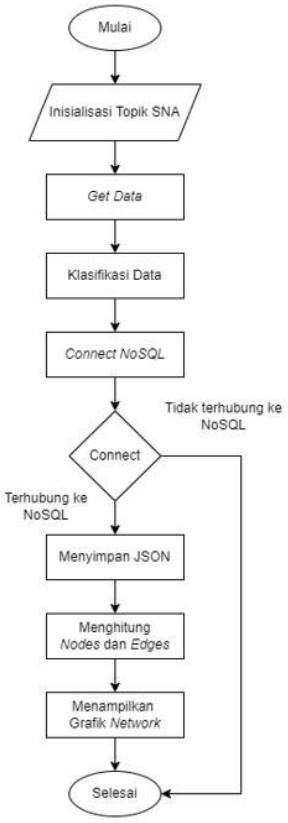
Seluruh *tweets* akan ditangkap menggunakan Streaming API twitter berdasarkan topik yang dimasukkan dan akan melalui tahapan preprocessing terlebih dahulu. Setelah itu *tweets* akan melalui tahap analisis sentimen lalu disimpan pada penyimpanan *database MongoDB*. Kemudian proses replikasi akan berjalan ke *database Elasticsearch* menggunakan *Compose Transporter*. Setelah itu aplikasi *SNA* akan menggunakan *Elasticsearch* untuk retrieve data yang selanjutnya akan divisualisasikan kepada user aplikasi *SNA* tersebut.

2.4. Desain dan perancangan data *collection tool*

Data diperoleh dari salah satu media sosial yaitu twitter. Program pengumpulan data terintegrasi dengan *API twitter* yang disediakan sehingga dapat mengambil data *tweet* dengan topik, jumlah dan waktu yang ditentukan. Program pengumpulan data dibuat menggunakan bahasa pemrograman Python dengan bantuan *library tweepy* untuk mengakses *API twitter*. Tiap data *tweet* hasil dari *streaming* terlebih dahulu melalui tahapan *preprocessing* dan analisis sentimen lalu kemudian disimpan pada penyimpanan *database MongoDB*.

2.5. Desain dan perancangan aplikasi *SNA*

Aplikasi *SNA* dibuat menggunakan bahasa PHP dengan *framework Code Igniter*. Aplikasi yang dibangun terdiri dari 2 bagian yaitu core (*backend*) dan user interface (*frontend*). Bagian *backend* adalah proses pengambilan data dari *database NOSQL* menggunakan bahasa pemrograman PHP. Bagian *frontend* adalah bagaimana *data* ditampilkan sehingga dapat memuat sebuah informasi. Untuk proses dari aplikasi *SNA* yang akan dibuat, dapat dilihat pada Gambar 2.



Gambar 2. Desain *SNA*

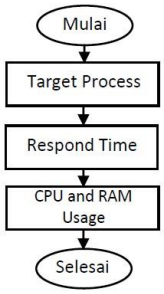
Inisialisasi topik dari *tweet* yang diinginkan lalu akan mengkases *database Elasticsearch* untuk pengambilan data yang selanjutnya masuk pada proses klasifikasi. Proses klasifikasi data yang ditampilkan merupakan hasil dari proses sentimen dengan pemanfaatan algoritma *TF-IDF* dan *Senti Strength.* Implementasi algoritma tersebut menggunakan *Python* untuk mengekstrak data yang didapatkan dari proses *collection* sebelumnya. Visualisasi hasil analisis sentimen dibuat dengan menggunakan *RESTful API* untuk mengambil data dari *database Elasticsearch*.

Data yang tampil pada halaman dashboard akan berupa *Pie*, *Bar* dan *Area chart* pada halaman Dashboard. Pada halaman *search*, terdapat beberapa f*orm* yang terdiri dari tiga *input form* untuk topik yang diinginkan dan sisanya adalah *form* untuk *sentiment polarity* seperti positif, negatif dan netral serta *form* untuk jumlah data yang ingin divisualisasikan.

Visualisasi data pada halaman *SNA* menggunakan *Sigma JS* yang menampilkan *nodes* dan *edges* yang berasal dari *file* dengan format *JSON* yang akan ditulis setelah *submit* form yang ada pada halaman *search.* Adapun properti jaringan pada visualisasi tersebut ada dua, yaitu *nodes* yang merupakan aktor atau orang yang melakukan sebuah *tweet* mengenai *keyword* yang dimasukkan dan *edges* yang merupakan garis penghubung antara *node aktor* dan *node* dari *keyword* yang dimasukkan. *Keyword* akan divisualisasikan sebagai *node* yang terhubung dengan seluruh aktor yang memiliki *tweet* yang serupa dengan *keyword* yang ditentukan.

2.6. Desain dan perancangan tool pengujian sistem *hybrid*

Program pengujian *database NOSQL* menggunakan bahasa *Python*. Untuk proses dari program pengujian *database NOSQL* yang akan dibuat dapat dilihat pada Gambar 3.



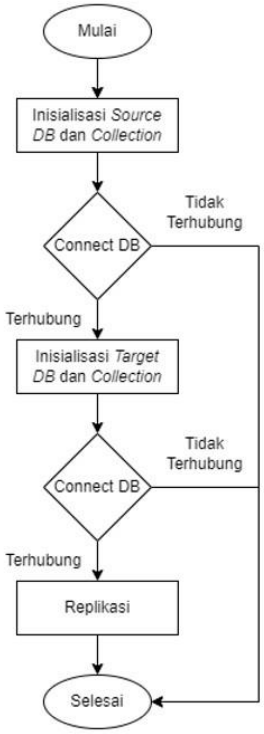
Gambar 3. Desain program pengujian sistem *hybrid*

Tiap proses yang dilakukan pada *system hybrid NoSQL*, akan diuji menggunakan tiga parameter yaitu *respond time*, *CPU usage* dan *RAM usage*. Untuk mendapatkan hasil pengujian *respond time*, dibutuhkan *library time* yang ada pada *Python*. Sedangkan untuk *CPU* dan *RAM usage*, dibutuhkan *library* lain yaitu *psutil*.

2.7. Implementasi pengambilan data

Pengambilan data terbagi menjadi lima variasi jumlah *tweet* yaitu 1K, 10K, 50K, 100K, 200K, 500K dan 1M. Tiap variasi jumlah data tersebut akan disimpan pada *collection* yang berbeda dalam penyimpanan *database MongoDB*. Tool yang digunakan untuk proses replikasi data dari *database NOSQL MongoDB* ke *Elasticsearch* adalah Compose Transporter.

Inisialisasi sumber *Database* yaitu *MongoDB* agar dapat terhubung beserta dengan Collection yang sesuai dengan data yang sebelumnya telah diambil menggunakan API Twitter lalu selanjutnya menginisialisasikan target *Database* yaitu *Elasticsearch* agar dapat dimulai proses Replikasi. Alur proses replikasi dapat dilihat pada Gambar 4.



Gambar 4. Proses replikasi *NOSQL*

2.8. Implementasi pengujian sistem *hybrid*

Pengujian sistem *hybrid NOSQL* menggunakan program yang telah dibuat pada tahapan desain dan perancangan sebelumnya. Ketiga tahapan proses akan menggunakan variasi jumlah data yang berbeda untuk mengukur ketiga parameter yang ada yaitu *respond time, CPU usage* dan *RAM usage*.

2.9. Implementasi dan pengujian aplikasi *SNA*

Pengujian dilakukan dengan cara mengamati output dari berbagai masukan dan membandingkan output aplikasi yang dibangun dengan output yang dilihat oleh pengguna sehingga sesuai dengan tujuan yang akan dicapai pada penelitian ini. Pengujian aplikasi *SNA* dilakukan menggunakan metode *black box* dengan cara mengamati hasil inputan user dengan hasil saat program berhasil dijalankan.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, dibangun sistem *hybrid NOSQL* yaitu *MongoDB* dan *Elasticsearch* untuk dilakukan pengujian terhadap kasus data media sosial yaitu *twitter*. Penelitian difokuskan untuk mendapatkan kesimpulan dari analisis data hasil pengujian.

3.1. Hasil Pengambilan data

Proses pengumpulan data *twitter* berhasil dilakukan dengan menggunakan *API* yang disediakan oleh *Twitter* dengan menggunakan *library Python* yaitu *tweepy*. Proses pengumpulan data dimulai pada bulan Januari 2020 hingga Juli 2020 dan berhasil mengambil data sebanyak 1.861.000 *tweet* yang telah terbagi menjadi 5 *collections* di dalam *MongoDB* yaitu 1K, 10K. 50K, 100K, 200K, 500K dan 1M *tweet*.

Data yang berhasil di ambil kemudian melalui proses *preprocessing* dan analisis sentimen. Hasilnya disimpan pada penyimpanan *MongoDB* yang telah dibagi berdasarkan jumlah *tweet*. Setiap satu data *tweet* yang berhasil didapatkan dan telah diproses, disimpan dalam bentuk JSON yang memiliki lebih dari 300 *key* dan *value* pada penyimpanan *MongoDB.*

3.2. Replikasi *database* NoSQL

Replikasi pada *database MongoDB* ke *Elasticsearch* yaitu melakukan *copy* dan pendistribusian seluruh dokumen hasil dari *streaming API Twitter* dan *preprocessing* pada *MongoDB* yang selanjutnya masuk ke *Elasticsearch*. Ketika program replikasi *NoSQL* berjalan, maka tampilan program tersebut seperti pada Gambar 4.4. Terlebih dahulu *Transporter* akan mengevaluasi dan menghubungkan *pipeline*, *source* dan *sink database*. Setelah itu data dari *MongoDB* direplikasi ke *database Elasticsearch.* c. Pengujian performansi hybrid NoSQL

3.2.1 Respond Time Process

Berikut ini adalah tabel dan grafik dari *respond time process* yang didapatkan:

Tabel 1. Hasil pengujian *respond time process*

Operasi	Respond Time Query (s)						
	1K	10K	50K	100K	200K	500K	1M
Insert	26,21	222,81	1039,74	2140,31	4319,66	9763,64	19520,45
Replikasi	6,25	27,13	111,7	281,09	562,03	1412,017	2817,146
Select	0,044	0,017	0,021	0,018	0,018	0,016	0,017

Dari tabel tersebut terlihat bahwa nilai *respond time* pada proses *insert* meningkat signifikan. Hal tersebut karena *tweet* dari proses *streaming* terlebih dahulu melalui tahap *preprocessing*. Pada operasi replikasi, nilai *respond time query* mengalami peningkatan. Sementara itu, nilai *respond time query* pada proses *select* sangat rendah dan cukup konsisten dikarenakan *Inverted In*dex yang ada pada *Elasticsearch* membuat proses pencarian dengan pemanfaatan *analyzer* yang terdiri dari tokenisasi dan filtrasi menjadi lebih cepat.

3.2.2 Penggunaan CPU

Berikut ini adalah tabel dan grafik dari penggunaan CPU yang didapatkan:

Tabel 2. Hasil pengujian penggunaan CPU

Operasi	Respond Time Query (s)				
	1K	10K	50K	100K	200K
Insert	6,628	6,4821	6,339988	6,55801	6,84455
Replikasi	7,54	7,44	7,64	6,94	7,14
Select	13,98	10,18	9,96	9,8	12,44

Dari tabel tersebut dapat terlihat bahwa penggunaan *CPU* pada proses *Insert* dan replikasi data yang ada cukup rendah dan konsisten dimana tidak terlalu berbeda saat mengalami peningkatan variasi jumlah data yang digunakan pada pengujian tersebut. Berbeda dengan proses *select* yang membutuhkan penggunaan *CPU* yang lebih tinggi dari kedua proses sebelumnya.

3.2.3 Penggunaan RAM

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

Tabel 3. Hasil pengujian penggunaan RAM

Operasi	Respond Time Query (s)				
	1K	10K	50K	100K	200K
Insert	41,24	42,44	45,54	52,62	67,3
Replikasi	42,58	44,76	50,1	57,88	71,84
Select	39,1	39,78	39,74	39,7	39,5

Dari tabel tersebut dapat terlihat bahwa penggunaan *RAM* pada proses *insert* dan replikasi data yang ada cukup tinggi dan mengalami peningkatan seiring bertambahnya jumlah data yang digunakan pada proses pengujian berlangsung. Berbeda dengan proses *select* yang membutuhkan penggunaan *RAM* dengan nilai yang sedikit lebih rendah dari kedua proses sebelumnya dan berlangsung secara konsisten walaupun mengalami peningkatan jumlah data.

4. KESIMPULAN

1. Sistem *hybrid NoSQL* yang dibangun menggunakan *compose transporter* untuk proses replikasi dapat berjalan dengan baik untuk aplikasi *social network analysis*.
2. Peningkatan jumlah data mempengaruhi *respond time* pada proses *insert* dan replikasi data. *Respond Time* tertinggi ada pada proses *insert* mulai dari 26.2 detik pada data 1K hingga 19520.45 detik pada data 1M. Adapun proses replikasi membutuhkan lebih sedikit *Respond Time Query*, yaitu jumlah data 1K *tweet* yaitu 6.25s hingga 1M *tweet* yaitu 2817.146s. Sedangkan pada proses *select* memiliki *respond time* dibawah 0,1s dan bernilai relatif konstan walaupun mengalami peningkatan data.
3. Persentase penggunaan *CPU* pada proses *Insert* dan replikasi data cukup rendah dan konsinten walaupun datanya bertambah, adapun pada proses *select* data dari *Elasticsearch* membutuhkan penggunaan CPU lebih tinggi dan inkonsisten. Presentase penggunaan *RAM* pada proses *insert* dan replikasi dipengaruhi oleh peningkatan jumlah data. Sedangkan penggunaan *RAM* untuk proses *select* bernilai relatif konstan dan lebih rendah daripada proses *insert* dan replikasi.

5. REFERENSI

[1] Mohandas, A. *et al.* (2018) 'A Survey on Mining Social Media Data for Understanding Drug Usage', *2018 Conference on Emerging Devices and Smart Systems (ICEDSS)*. IEEE, (March), pp. 259– 261.
[2] Bello-orgaz, G., Jung, J. J. and Camacho, D. (2016) 'Social big data : Recent achievements and new challenges', *Information Fusion*. Elsevier B.V., 28, pp. 45–59. doi: 10.1016/j.inffus.2015.08.005.
[3] Kulcu, S. (2016) 'A Survey on Semantic Web and Big Data Technologies for Social Network Analysis', pp. 1768–1777.
[4] IT, S. (2019) *DB-Engines Ranking, DBEngines*.