

Perbandingan Waktu Respon Aplikasi *Database* NoSQL Elasticsearch dan MongoDB pada Pengujian Operasi CRUD

Theresia Liana Sinaga ⁽¹⁾, Novrido Charibaldi ^{(2)*}, Nur Heri Cahyana ⁽³⁾

Informatika, Fakultas Teknik Industri, Universtas Pembangunan Nasional “Veteran” Yogyakarta,
Yogyakarta

e-mail : sinagatheresia48@gmail.com, {novrido,nur.herichahyana}@upnyk.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 1 November 2022, direvisi 23 Januari 2023, diterima 23 Januari 2023, dan dipublikasikan 30 Januari 2023.

Abstract

Currently, humans live in an era of data oceans, where the amount of data production is increasing from time to time, which is followed by severe challenges in terms of processing, storing, and analyzing data, especially big data. The increase in the number of large data production can affect the speed of access to the database, effectiveness, and speed of response time in the data processing. Relational databases have been the leading model for data storage, analysis, processing, and retrieval for more than forty years. However, due to the increasing need for large-scale data storage, the scalability and performance of a data processing system, as well as the constant growth of the amount of data, another alternative to databases emerged, namely NoSQL technology. Based on previous studies regarding the comparison of response time and database performance, the average concludes that NoSQL performance is more effective and efficient than relational databases. Based on the implementation and testing, it can be concluded that the NoSQL database application MongoDB is proven to be superior in every command of CRUD tested compared to the Elasticsearch NoSQL database application, where in testing the create data command with a JSON file, the MongoDB database application is 42.5 times faster than the Elasticsearch database application. In testing the command to create data into a database containing different amounts of data, the MongoDB database application is 333.9 times faster than the average response time of the Elasticsearch database application. In testing the read command for data in a database containing different amounts of data, the MongoDB database application is 35.5 times faster than the Elasticsearch database application. In testing the update operation of data in a database containing different amounts of data, the MongoDB database application is 9.8 times faster than the Elasticsearch database application. In testing the delete operation of data in a database containing different amounts of data, the MongoDB database application is 58.9 times faster than the Elasticsearch database application.

Keywords: Database, NoSQL, Response Time, CRUD Operation Testing, Elasticsearch, MongoDB

Abstrak

Saat ini manusia hidup di era lautan data, di mana jumlah produksi data semakin bertambah dari waktu ke waktu yang diikuti tantangan berat dalam hal pemrosesan, penyimpanan, dan analisis data, terkhusus pada data besar. Peningkatan jumlah produksi data yang besar dapat mempengaruhi kecepatan akses pada *database*, efektivitas, dan kecepatan waktu respon dalam pemrosesan data. *Database* relasional telah menjadi model terdepan untuk penyimpanan, analisis, pemrosesan, dan pengambilan data selama lebih dari empat puluh tahun. Namun, dikarenakan meningkatnya kebutuhan akan penyimpanan data dengan skala besar, skalabilitas dan kinerja dari suatu sistem pengolahan data, serta pertumbuhan konstan dari jumlah data, maka muncul alternatif lain dari basis data, yaitu teknologi NoSQL. Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya mengenai perbandingan dari waktu respon dan performa *database*, rata-rata menyimpulkan bahwa performa NoSQL lebih efektif dan efisien dibanding *database* relasional. Berdasarkan implementasi dan pengujian dapat disimpulkan bahwa aplikasi *database* NoSQL MongoDB terbukti memiliki waktu respon yang lebih cepat dalam melakukan perintah operasi CRUD yang diujikan dibandingkan dengan aplikasi *database* NoSQL Elasticsearch, di mana pada pengujian perintah *create data* dengan *file* JSON, aplikasi



database MongoDB 42,5 kali lebih cepat dibanding aplikasi *database* Elasticsearch. Pada pengujian perintah *create* sebuah data ke dalam *database* yang berisi jumlah data yang berbeda-beda, aplikasi *database* MongoDB 333,9 kali lebih cepat dibanding nilai rata-rata waktu respon aplikasi *database* Elasticsearch. Pada pengujian perintah *read* sebuah data di dalam *database* yang berisi jumlah data yang berbeda-beda, aplikasi *database* MongoDB 35,5 kali lebih cepat dibandingkan aplikasi *database* Elasticsearch. Pada pengujian operasi *update* sebuah data di dalam *database* yang berisi jumlah data yang berbeda-beda, aplikasi *database* MongoDB 9,8 kali lebih cepat dibandingkan aplikasi *database* Elasticsearch. Pada pengujian operasi *delete* sebuah data di dalam *database* yang berisi jumlah data yang berbeda-beda, aplikasi *database* MongoDB 58,9 kali lebih cepat dibandingkan aplikasi *database* Elasticsearch.

Kata Kunci: Basis Data, NoSQL, Waktu Respon, Pengujian Operasi CRUD, Elasticsearch, MongoDB

1. PENDAHULUAN

Saat ini manusia hidup di era lautan data (Ahmed et al., 2018), di mana jumlah produksi data semakin bertambah dari waktu ke waktu yang diikuti tantangan berat dalam hal pemrosesan, penyimpanan, dan analisis data, terkhusus pada data besar (Modhiya, 2021). Peningkatan jumlah produksi data yang besar dapat mempengaruhi kecepatan akses pada *database*, efektivitas, dan kecepatan waktu respon dalam pemrosesan data (Tavares et al., 2020).

Basis data (*database*) merupakan sebuah solusi terhadap penyimpanan data yang memberi ruang untuk menyimpan dan memanipulasi data. *Database* relasional merupakan salah satu tipe *database* yang dapat melakukan penyimpanan, ekstraksi, dan manipulasi data dengan menggunakan bahasa Structured Query Languages (SQL) (Amandeep & Singh, 2016). *Database* relasional telah menjadi model terdepan untuk penyimpanan, analisis, pemrosesan, dan pengambilan data selama lebih dari empat puluh tahun. Namun, dikarenakan meningkatnya kebutuhan akan penyimpanan data dengan skala besar, skalabilitas dan kinerja dari suatu sistem pengolahan data, serta pertumbuhan konstan dari jumlah data, maka muncul alternatif lain dari basis data, yaitu teknologi NoSQL (Not Only SQL atau No SQL) (Lourenço et al., 2015).

NoSQL merupakan paradigma baru dalam teknologi *database*, yang merupakan tipe lain dari *database* relasional. NoSQL merupakan sistem manajemen data non-relasional yang tidak memerlukan skema tetap maupun *query* yang kompleks. NoSQL lahir disebabkan oleh pertumbuhan yang pesat pada internet dan laju perkembangan aplikasi web yang semakin kompleks dan memerlukan pengolahan data dalam skala yang besar (Wicaksana, 2017). Perbedaan utama dari *database* relasional dengan *database* NoSQL terdapat pada skema, di mana skema pada *database* relasional secara kaku menentukan bagaimana data harus dimasukkan ke dalam *database* dan menyimpan data dalam format terstruktur yang menggunakan tabel dengan baris dan kolom, sedangkan *database* NoSQL dapat menjadi skema agnostik, yang artinya *database* NoSQL mengakui adanya skema pada *database* lain, tetapi *database* NoSQL tidak menggunakan skema kaku (*schemeless*) pada penyimpanan datanya, sehingga memungkinkan untuk melakukan penyimpanan dan manipulasi data pada data tidak terstruktur dan data semi terstruktur. Terdapat beberapa tipe dari *database* NoSQL, di antaranya yaitu *key-value stores*, *wide-column stores*, *graph databases*, dan *document stores* (Bhaswara et al., 2017).

Ada beberapa tinjauan terhadap penelitian sebelumnya yang membahas perbandingan kecepatan waktu respon dari aplikasi *database*, baik perbandingan waktu respon antara aplikasi *database* relasional dengan aplikasi *database* NoSQL, maupun antar aplikasi *database* yang non relasional. Pertama, Analisis Pemanfaatan NoSQL *Database* Elasticsearch pada Mesin Pencari Tokopedia (Yafet, 2020), penelitian ini dilakukan untuk menganalisis keunggulan dari Elasticsearch yang digunakan sebagai mesin pencarian oleh Tokopedia dibandingkan dengan aplikasi *database* PostgreSQL, yang juga digunakan oleh Tokopedia. Penelitian tersebut menyimpulkan bahwa Elasticsearch terbukti memiliki *availability* dan *flexibility*, serta



Elasticsearch juga dapat melakukan operasi pencarian dan pembacaan data sekitar 92,96% lebih cepat dibandingkan PostgreSQL.

Penelitian kedua berjudul Analisis Perbandingan Performansi Waktu Respon *query* antara MySQL PHP 7.2.27 dan NoSQL MongoDB (Tavares et al., 2020), Kedua *database* tersebut diuji pada Sistem Layanan Aspirasi dan Informasi Kelurahan Oebufu (SELMA) dengan jumlah data 50, 100, 500, 1000, 5000, 10000, dan 100000, pengujian dilakukan dengan Data Manipulation Language (DML), dengan fungsi agregat, dengan fungsi operator, serta *import* dan *export* data. Penelitian itu menyimpulkan bahwa kedua aplikasi *database* tersebut memiliki waktu respon *query* yang berbeda pada setiap model pengujian. MongoDB terbukti lebih unggul di semua model pengujian *query* DML yang terdiri atas proses memasukkan data, membaca data, *update data*, dan hapus data; pengujian fungsi agregat, pengujian fungsi operator penghubung, serta pengujian *import* dan *export* data. Namun, MongoDB memiliki kelemahan pada proses *query* select untuk menampilkan data dengan selisih waktu respon 1,95 detik lebih lambat dari MySQL PHP 7.2.27.

Penelitian ketiga membandingkan kecepatan waktu respon dari aplikasi *database* NoSQL terhadap SQL pada kasus ERP Retail (Bhaswara et al., 2017), Penelitian ini membandingkan *performance*, *flexibility*, dan *scalability* antara *database* relasional (SQL) dengan *database* NoSQL. Setelah didapatkan hasil dari perbandingan kedua *database* tersebut, maka *database* yang unggul diterapkan pada aplikasi *Enterprise Resource Planning (ERP) Retail* yang berorientasi *multitenancy*. Dengan harapan, aplikasi ERP Retail dapat memiliki kinerja yang optimal untuk menyimpan data. Penelitian ini mendapatkan kesimpulan bahwa aplikasi MongoDB terbukti unggul dalam proses transaksi pada pengujian *Create, Read, Update, Delete (CRUD)* dibandingkan dengan MySQL dengan jumlah selisih waktu eksekusi pada *create data* 0,167 detik, *read data* 0,009 detik, *update data* 0,44 detik, dan *delete data* 0,056 detik. Selain itu, MongoDB mampu memenuhi kebutuhan dari aplikasi ERP Retail yang mendukung perbedaan skema dari setiap *tenant*. Namun, MongoDB lemah pada proses transaksi agregasi dengan jumlah selisih waktu eksekusi lebih lambat 0,039 detik.

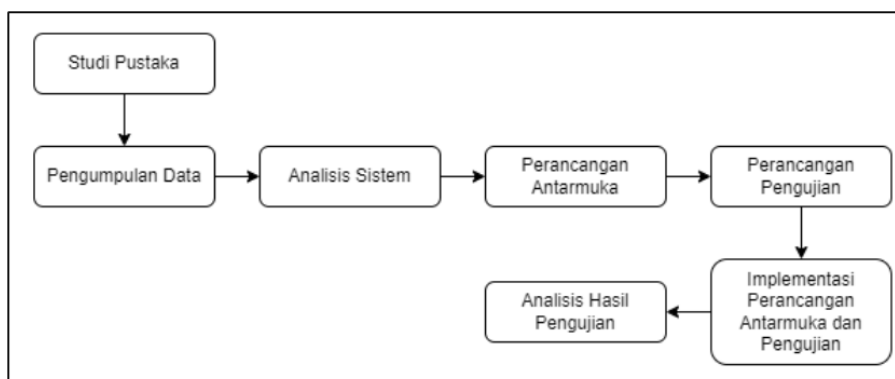
Penelitian keempat dengan judul *An Analysis on the Comparison of the Performance and Configuration Features of Big Data Tools Solr and Elasticsearch* (Aydoğan et al., 2016). Penelitian ini membandingkan kinerja dari aplikasi *database* NoSQL Solr dan aplikasi *database* NoSQL Elasticsearch. Variabel yang dibandingkan adalah waktu operasi *query*, kemudahan dan kesulitan dalam penggunaan *database*, bentuk konfigurasi, dan arsitektur. Kesimpulan yang didapatkan pada penelitian ini adalah Elasticsearch mendukung lebih banyak bahasa pemrograman daripada Solr, dalam hal pengindeksan, Elasticsearch memiliki kinerja yang lebih baik jika data lebih kecil, sedangkan Solr memiliki kinerja yang lebih baik untuk data yang besar. Waktu pemrosesan *query* bervariasi tergantung pada jenis data yang digunakan.

Berdasarkan tinjauan penelitian mengenai perbandingan dari waktu respon *database* (Gunawan, 2018; Halimi & Sudarmanto, 2021), ternyata kinerja NoSQL lebih baik dibanding *database* relasional. Namun, penelitian yang membandingkan waktu respon antara aplikasi *database* NoSQL MongoDB dan aplikasi *database* NoSQL Elasticsearch masih minim dilakukan. Beberapa perusahaan besar yang menerapkan MongoDB dalam pengolahan datanya, seperti SEGA HARDlight yang bergerak di bidang *mobile game development*, Medtronic yang merupakan perusahaan penyedia alat-alat medis di Amerika Serikat, EA yang merupakan *developer game* yang bertema olahraga (pada *game* EA FIFA Online 3 dikembangkan menggunakan MongoDB). Sedangkan beberapa perusahaan di Indonesia yang menerapkan Elasticsearch yaitu Tokopedia dan BCA, serta perusahaan enterprise seperti Facebook, Airbus, dan IEEE (Renaldi et al., 2020). Penelitian ini membandingkan waktu respon pada operasi *Create, Read, Update, Delete (CRUD)* data dari aplikasi *database* NoSQL Elasticsearch dan MongoDB, sehingga dapat diketahui aplikasi *database* NoSQL manakah yang bekerja lebih cepat. Pengujian pada penelitian ini dilakukan sebanyak 3 kali untuk tiap operasi CRUD data dalam beragam *record data* (100, 1000, dan 8900), sedangkan waktu respon dari setiap operasi disajikan dalam bentuk tabel dan grafik.



2. METODE PENELITIAN

Metodologi pada penelitian ini menggunakan metodologi penelitian kuantitatif. Adapun yang dimaksud dengan penelitian kuantitatif adalah penelitian yang erat kaitannya dengan filsafat positivisme. Penelitian ini lebih berfokus kepada pemecahan masalah yang erat kaitannya dengan angka-angka dan statistik. Berikut tahapan penelitian ditunjukkan pada Gambar 1.



Gambar 1 Tahapan Penelitian

2.1 Studi Pustaka

Studi pustaka yang dilakukan dalam penelitian ini yaitu dengan membaca artikel, buku, jurnal, dan tesis yang berkaitan dengan topik aplikasi *database* NoSQL Elasticsearch dan MongoDB, serta penelitian yang berfokus pada perbandingan kecepatan waktu respon dari aplikasi *database*.

2.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa data *dummy* yang bersumber dari *website* data *world* berupa *countries*, *states*, dan *cities*. Data yang didapatkan berupa *file* JSON.

2.3 Analisis Sistem

Tahapan berikutnya merupakan analisis terhadap batasan lingkup sistem secara general dan pengujian yang dilakukan dengan mengukur kecepatan waktu respon dari aplikasi *database* NoSQL Elasticsearch dan MongoDB melibatkan perangkat lunak (*software*), perangkat keras (*hardware*), serta data digunakan untuk mengukur waktu respon saat melakukan operasi CRUD pada Elasticsearch dan MongoDB.

2.3.1 Kebutuhan Perangkat Keras dan Perangkat Lunak

Kebutuhan perangkat keras dan perangkat lunak merupakan kebutuhan terkait spesifikasi dalam penggunaan sistem yang akan dibangun. Spesifikasi yang dibutuhkan terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*).

Tabel 1 Kebutuhan Perangkat Keras

No.	Perangkat Keras	Keterangan
1	Laptop	LENOVO Ideapad Slim 3-14ada05 Business Black Series
2	Processor	AMD Athlon Silver 3050U with Radeon Graphics (2 CPUs)
3	RAM	8192 MB
4	Storage	256 GB SSD
5	Perangkat <i>input</i> dan <i>output</i>	Keyboard, mouse, monitor, printer
6	Koneksi internet	Wi-Fi



Tabel 2 Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Keterangan
1	Windows 10	Sistem operasi
2	Python versi 3,8 ke atas	Bahasa pemrograman
3	Flask	<i>Micro framework</i> Python untuk pengembangan aplikasi berbasis web
4	Elasticsearch versi 8 ke atas	<i>Database</i> NoSQL
5	MongoDB versi 5 ke atas	<i>Database</i> NoSQL
6	Google Chrome	<i>Browser</i>

2.3.2 Kebutuhan Penelitian

1) Aplikasi *database* NoSQL Elasticsearch

Elasticsearch adalah salah aplikasi *database* NoSQL yang dikembangkan menggunakan bahasa pemrograman Java menggunakan Lucene Library oleh Shay Banon pada tahun 2010. Elasticsearch merupakan *hybrid database/search tool open-source* yang berorientasi pada dokumen dan memiliki skalabilitas yang tinggi. Elasticsearch dibangun oleh Apache Lucene sebagai search engine *database* yang memiliki *query low level* dan penyimpanan dokumen tanpa skema dalam format JSON, dan dapat lebih mudah untuk digunakan karena dapat diakses dari antarmuka layanan web RESTful API. Kecepatan dan skalabilitas yang dimiliki Elasticsearch, serta kemampuan untuk mengindeks banyak jenis konten dapat diaplikasikan pada sejumlah kasus penggunaan, seperti aplikasi pencarian, *website* pencarian, analisis keamanan, analisis bisnis, analisis dan visualisasi data geospasial, pemantauan kinerja aplikasi, dan lain-lain (Aydoğan et al., 2016). Terdapat beberapa konsep dasar untuk memahami struktur Elasticsearch, di antaranya yaitu (Yafet, 2020):

a) *Node*

Node merupakan sebuah instance Elasticsearch yang sedang bekerja. Banyaknya *node* tergantung pada kemampuan sumber daya fisiknya seperti Random Access Memory (RAM), memory, dan kemampuan untuk melakukan pemrosesan.

b) *Cluster*

Cluster merupakan kumpulan dari beberapa *node* yang bekerja sama untuk membaca atau menulis ke indeks. Setiap *node* pada sebuah *cluster* berkontribusi untuk melakukan pengindeksan dan pencarian data.

c) *Document*

Document (dokumen) merupakan kumpulan unit dasar informasi atau data dalam format JSON. Misalnya pengguna menyimpan data seorang siswa, maka pengguna menambahkan satu objek yang memiliki nama, umur, dan properti lainnya. Dokumen memiliki ID unik yang diberikan oleh pengguna saat menambahkannya ke *index*. Dokumen-dokumen tersebut disimpan pada *index*.

d) *Index*

Index merupakan kumpulan dari dokumen yang memiliki sifat serupa, misalnya *index* untuk data pesanan, data produk, data pelanggan, dan lain-lain.

e) *Shard*

Elasticsearch memiliki kemampuan untuk memotong *index* menjadi beberapa *shard/partisi* yang disebut dengan proses *shard/sharding*. Secara *default*, Elasticsearch melakukan *sharding* setelah terbuatnya sebuah *index*. Dengan adanya *sharding* pada *index*, maka dapat memudahkan pendistribusian *index* ke beberapa *node* dalam sebuah *cluster* di Elasticsearch.

f) *Replication*

Elasticsearch memiliki fitur replikasi untuk menduplikasi *index* yang telah melewati proses *sharding*. Secara *default*, setelah *index* dibuat, Elasticsearch akan membuat replikasi dari setiap *shard index*. Hal ini berguna untuk mencegah kehilangan data dokumen *index* karena kegagalan jaringan yang tidak terduga. Replika *shard* tidak pernah ditempatkan pada *node* yang sama dengan *shard* primer, sehingga dapat meningkatkan *throughput* dan kinerja



pencaharian. Saat membuat *index*, pengguna dapat memilih berapa jumlah *shard* dan replikanya.

2) Aplikasi *database* NoSQL MongoDB

MongoDB merupakan aplikasi *database* NoSQL yang menyimpan datanya dalam dokumen dengan format JSON. MongoDB merupakan aplikasi *database* yang dikembangkan oleh Mongo, Inc. MongoDB tidak perlu mendefinisikan struktur seperti atribut dan tipe data terlebih dahulu untuk melakukan penyimpanan data. Karena MongoDB menggunakan skema dinamis, sehingga model aplikasi *database* seperti ini dapat membantu untuk menyimpan *array* dan struktur lain yang lebih kompleks dengan mudah (Damodaran et al., 2016). Ada tiga komponen penting pada aplikasi *database* NoSQL, yaitu (Chauhan, 2019):

- Database*, merupakan wadah yang memiliki struktur penyimpanan yang disebut *collection*.
- Collection*, merupakan kumpulan dokumen-dokumen (*collection* ini setara dengan tabel pada sistem *database* relasional).
- Document*, merupakan unit data satuan terkecil dalam MongoDB yang berisikan baris-baris data yang berupa struktur pasangan *key-value* yang berfungsi saling memasangkan informasi. *Document* dapat dianalogikan seperti *record* pada *database* relasional (Kriestanto & Arnado, 2017).

2.4 Perancangan Antarmuka

Rancangan antarmuka pengguna pada penelitian ini berupa rancangan halaman aplikasi web sederhana. Rancangan antarmuka terdiri atas halaman *home* dan halaman *forms*. Halaman *home* digunakan untuk melakukan pencaharian data, menampilkan seluruh data dalam sebuah tabel, dan menampilkan waktu dari pencaharian data. Sedangkan halaman *forms*, merupakan halaman bagi pengguna untuk melakukan *create data*, *update data*, dan *delete data*, sedangkan waktu respon setiap perintah ditampilkan setelah instruksi *create*, *update*, dan *delete data*. Tampilan dari rancangan antarmuka halaman *home* dapat dilihat pada Gambar 2.

Home Forms

**Countries, States, and Cities
in the World**

Country Name

State Name

City Name

Choose Database

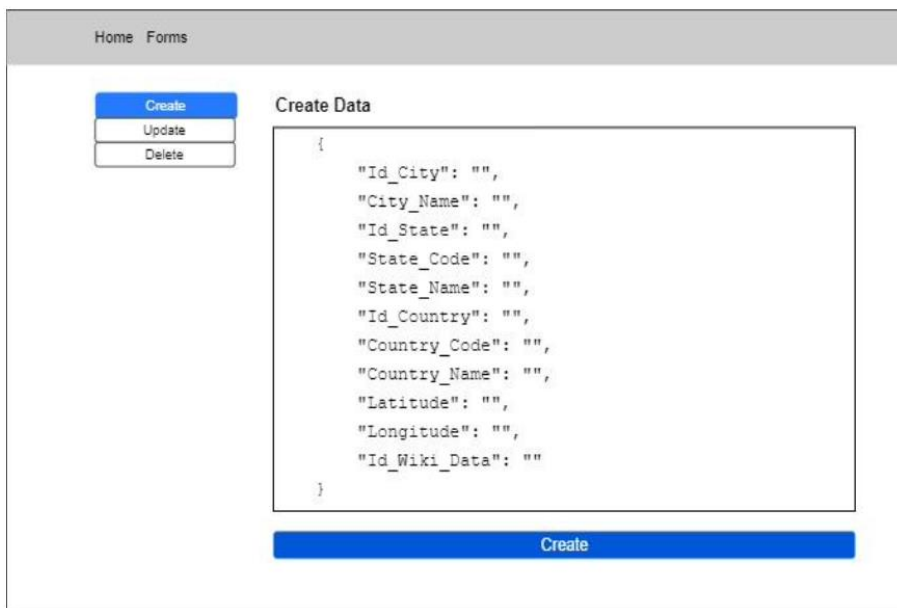
Waktu pencaharian:

Id Country	Country Code	Country Name	Id State	State Code	State Name	Id City	City Name	Latitude	Longitude	Id Wiki Data

Gambar 2 Rancangan Tampilan Antarmuka Halaman *Home*

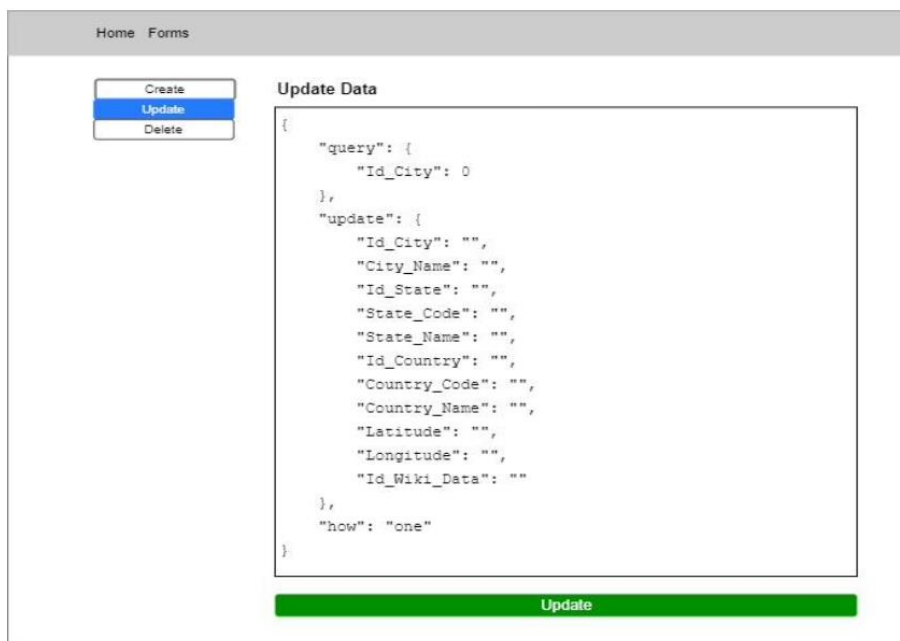
Gambar 3 merupakan rancangan tampilan antarmuka halaman *forms* yaitu *create data* yang digunakan untuk memasukkan data baru.





Gambar 3 Rancangan Tampilan Antarmuka Halaman *Forms Create Data*

Gambar 4 merupakan rancangan tampilan antarmuka halaman *forms* untuk proses *update data*.



Gambar 4 Rancangan Tampilan Antarmuka Halaman *Forms Update Data*

Gambar 5 merupakan rancangan tampilan antarmuka halaman *forms* untuk proses *delete data* yang dikehendaki.





Gambar 5 Rancangan Tampilan Antarmuka Halaman *Forms Delete Data*

2.5 Perancangan Pengujian

Pengujian dilakukan dengan mengukur waktu respon saat menjalankan operasi CRUD dari awal *query* berjalan sampai dengan mendapatkan hasil *query*. Pengujian dilakukan pada aplikasi web sederhana. Berikut merupakan skenario dari pengujian yang dilakukan.

2.5.1 Pengujian kecepatan waktu respon *create data* (menambahkan data)

Pada pengujian ini, masing-masing aplikasi *database* NoSQL Elasticsearch dan MongoDB diuji dengan melakukan *query* penambahan data baru. Terdapat dua cara dalam memasukkan data, yang pertama dengan memasukkan data melalui *file* JSON yang berisi lebih dari satu data di dalamnya. Sedangkan untuk cara kedua, yaitu memasukkan data melalui halaman *forms create data* pada aplikasi web sederhana. Setiap *field* dan tipe data pada data yang ditambahkan diharuskan sama, serta sesuai dengan skema masing-masing aplikasi *database*. Setelah proses *insert file/create data* selesai, maka waktu respon dari kedua aplikasi *database* ditampilkan.

2.5.2 Pengujian kecepatan waktu respon *read data* (mencari data)

Pada pengujian ini, masing-masing *database* NoSQL Elasticsearch dan MongoDB diuji dengan melakukan *query* pencarian data dan mengembalikan hasil dari pencarian data pada aplikasi web sederhana dan menampilkan waktu pencarian. Setelah proses *read data* selesai, maka waktu respon dari kedua aplikasi *database* ditampilkan.

2.5.3 Pengujian kecepatan waktu respon *update data* (mengubah data)

Pada pengujian ini, masing-masing aplikasi *database* NoSQL Elasticsearch dan MongoDB diuji dengan operasi *query* pencarian data dan mengembalikan hasil dari pencarian data ke aplikasi web sederhana dan menampilkan waktu pencarian. Setelah proses *read data* selesai, maka waktu respon dari kedua aplikasi *database* ditampilkan.

2.5.4 Pengujian waktu respon *delete data* (menghapus data)

Pada pengujian ini, masing-masing aplikasi *database* NoSQL Elasticsearch dan MongoDB diuji dengan melakukan *query* hapus data yang diinginkan pada masing-masing aplikasi *database*. Setelah proses *delete data* selesai, maka waktu respon dari kedua aplikasi *database* ditampilkan.

3. HASIL DAN PEMBAHASAN

Pengujian yang telah dilakukan untuk mengetahui perbandingan kecepatan waktu respon dari perintah *create*, *read*, *update*, dan *delete data* pada aplikasi *database* NoSQL MongoDB dan Elasticsearch dijabarkan dalam bentuk tabel dan grafik. Agar dapat diketahui setiap perintah yang



memiliki nilai waktu respon rendah pada masing-masing pengujian data dianggap lebih cepat dibandingkan pengujian yang memiliki nilai waktu respon lambat. Berikut penjabaran perbandingan waktu respon yang berupa tabel dan grafik untuk setiap perintah *create*, *read*, *update*, dan *delete data* yang telah dilakukan.

3.1 Pengujian waktu respon perintah *create data* dengan *file JSON*

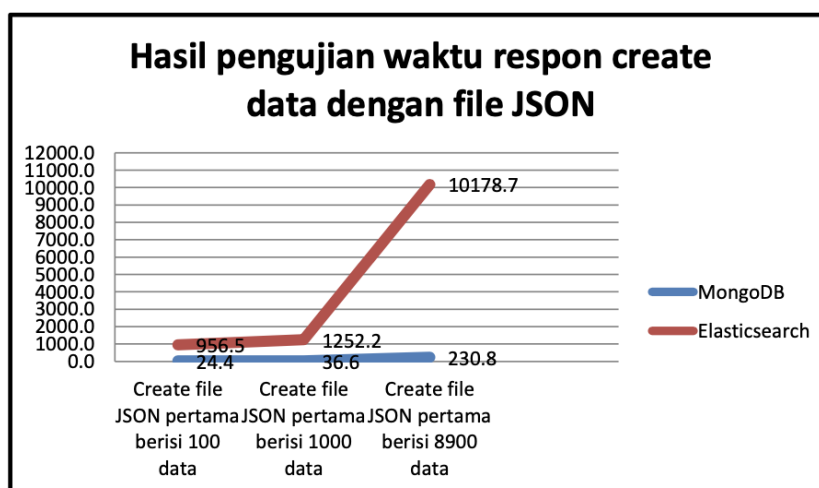
Pengujian kecepatan waktu respon saat memasukkan data dengan *file JSON* yang berisi lebih dari satu data *countries*, *states*, dan *cities* dilakukan sebanyak tiga kali proses *create data* dengan data *file JSON*. *File* pertama berisi 100 data *countries*, *states*, dan *cities*; *file* kedua berisi 1000 data *countries*, *states*, dan *cities*; dan *file* ketiga berisi 8900 data *countries*, *states*, dan *cities*; sehingga total data *countries*, *states*, dan *cities* yang dimasukkan yaitu berjumlah 10000 data.

Tabel 3 Tabel Perbandingan Waktu Respon Perintah *Create Data* dengan *File JSON*

<i>Database</i>	<i>Create file JSON pertama berisi 100 data</i>	<i>Create file JSON pertama berisi 1000 data</i>	<i>Create file JSON pertama berisi 8900 data</i>
Jumlah Data			
MongoDB	24,4 ms	36,6 ms	230,8 ms
Elasticsearch	956,5 ms	1252,2 ms	10178,7 ms

Saat melakukan *create data file JSON* pertama yang berisi 100 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 24,4 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 956,5 ms. Pada saat melakukan *create data file JSON* kedua yang berisi 1000 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 36,6 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 1252,2 ms. Pada saat melakukan *create data file JSON* ketiga yang berisi 8900 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 230,8 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 10178,7 ms.

Gambar 6 merupakan data hasil pengujian yang disajikan dalam bentuk grafik. Pada pengujian *create data* dengan *file JSON* dapat dibandingkan bahwa aplikasi *database* MongoDB memiliki waktu respon lebih cepat dibandingkan dengan Elasticsearch, nilai rata-rata waktu respon aplikasi *database* MongoDB yaitu 97,2 ms, sedangkan nilai rata-rata waktu respon aplikasi *database* Elasticsearch yaitu 4129,1 ms. Dengan kata lain, aplikasi *database* MongoDB 42,5 kali lebih cepat dibanding aplikasi *database* Elasticsearch.



Gambar 6 Grafik Hasil Pengujian Waktu Respon *Create Data* dengan *File JSON*



3.2 Pengujian waktu respon perintah *create data* melalui *forms create data*

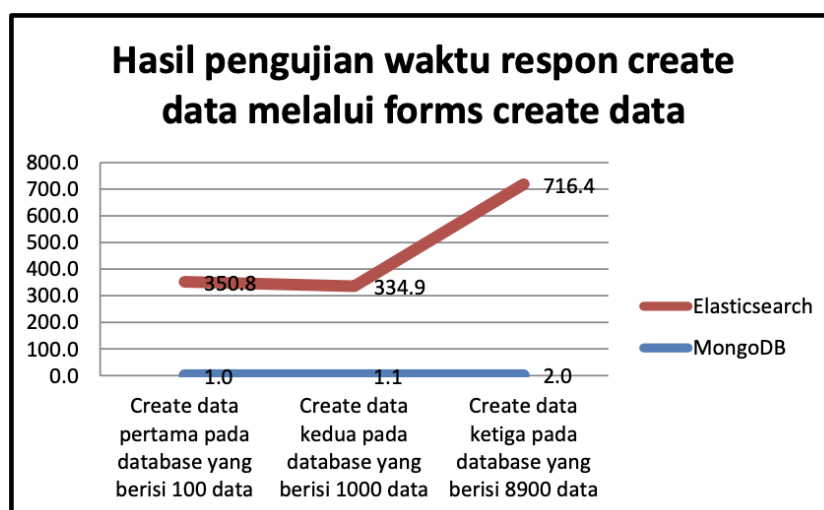
Pengujian kecepatan waktu respon *create data* dengan cara kedua yaitu memasukkan data baru melalui *forms create data* yang terdapat pada aplikasi web sederhana. Cara ini hanya dilakukan dengan memasukkan satu data baru/satu dokumen baru. Pengujian pertama, memasukkan satu data baru pada *database* yang telah menyimpan 100 data. Pengujian kedua, memasukkan satu data baru pada *database* yang telah menyimpan 1000 data. Pengujian ketiga, memasukkan satu data baru pada *database* yang menyimpan 8900 data.

Tabel 4 Tabel Perbandingan Waktu Respon Perintah *Create Data* Melalui *Forms Create Data*

<i>Database</i>	<i>Create file JSON</i> pertama berisi 100 data	<i>Create file JSON</i> pertama berisi 1000 data	<i>Create file JSON</i> pertama berisi 8900 data
Jumlah Data			
MongoDB	1,0 ms	1,1 ms	2,0 ms
Elasticsearch	334,9 ms	350,8 ms	716,4 ms

Saat melakukan *create data* baru pada *database* yang telah menyimpan 100 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 1,0 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 334,9 ms. *Create data* baru pada *database* yang telah menyimpan 1000 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 1,1 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 350,8 ms. *Create data* baru pada *database* yang telah menyimpan 8900 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 2,0 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 716,4 ms.

Gambar 7 merupakan hasil pengujian yang disajikan dalam bentuk grafik. Pengujian *create data* melalui *forms create data* dapat dinyatakan bahwa aplikasi *database* MongoDB masih memiliki waktu respon lebih cepat dibandingkan dengan Elasticsearch, nilai rata-rata waktu respon aplikasi *database* MongoDB yaitu 1,4 ms, sedangkan nilai rata-rata waktu respon aplikasi *database* Elasticsearch yaitu 467,4 ms. Dengan kata lain, nilai rata-rata waktu respon aplikasi *database* MongoDB 333,9 kali lebih cepat dibanding nilai rata-rata waktu respon aplikasi *database* Elasticsearch.



Gambar 7 Grafik Hasil Pengujian Waktu Respon *Create Data* Melalui *Forms Create Data*



3.3 Pengujian waktu respon perintah *read data*/pencarian data

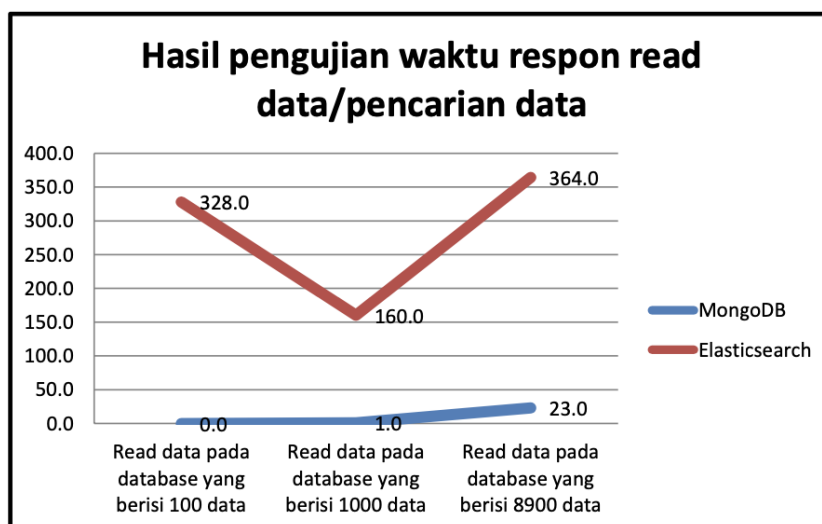
Pengujian waktu respon proses pencarian data dilakukan sebanyak tiga kali, pengujian pertama mencari suatu data yang tersimpan dalam *database* berisi 100 data, pengujian kedua mencari suatu data yang tersimpan dalam *database* berisi 1000 data, dan pengujian ketiga mencari suatu data yang tersimpan dalam *database* berisi 8900 data.

Tabel 5 Tabel Perbandingan Waktu Respon Perintah *Read Data*

<i>Database</i>	<i>Read data pada database berisi 100 data</i>	<i>Read data pada database berisi 1000 data</i>	<i>Read data pada database berisi 8900 data</i>
Jumlah Data			
MongoDB	0,01 ms	1 ms	23 ms
Elasticsearch	328 ms	160 ms	364 ms

Operasi *read data*/pencarian suatu data dalam *database* yang berisi 100 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 0,01 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 328 ms. Operasi *read data*/pencarian suatu data pada *database* yang telah menyimpan 1000 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 1 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 160 ms. Pada saat melakukan *read data*/pencarian suatu data pada *database* yang telah menyimpan 8900 data, waktu respon yang dibutuhkan aplikasi *database* MongoDB adalah 23 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 364 ms.

Gambar 8 adalah data hasil pengujian yang disajikan dalam bentuk grafik. Operasi pengujian *read data*/pencarian data dapat dinyatakan bahwa aplikasi *database* MongoDB masih memiliki waktu respon lebih cepat dibandingkan dengan Elasticsearch, nilai rata-rata waktu respon aplikasi *database* MongoDB yaitu 8,0 ms, sedangkan nilai rata-rata waktu respon aplikasi *database* Elasticsearch yaitu 284 ms. Dengan kata lain, waktu respon yang diperlukan aplikasi *database* MongoDB 35,5 kali lebih cepat dibandingkan aplikasi *database* Elasticsearch.



Gambar 8 Grafik Hasil Pengujian Waktu Respon *Read Data*/Pencarian Data

3.4 Pengujian waktu respon perintah *update data* melalui *forms update data*

Pengujian waktu respon perintah *update data* melalui *forms update data* dilakukan sebanyak tiga kali, pengujian pertama adalah operasi *update* suatu data di dalam *database* berisi 100 data,

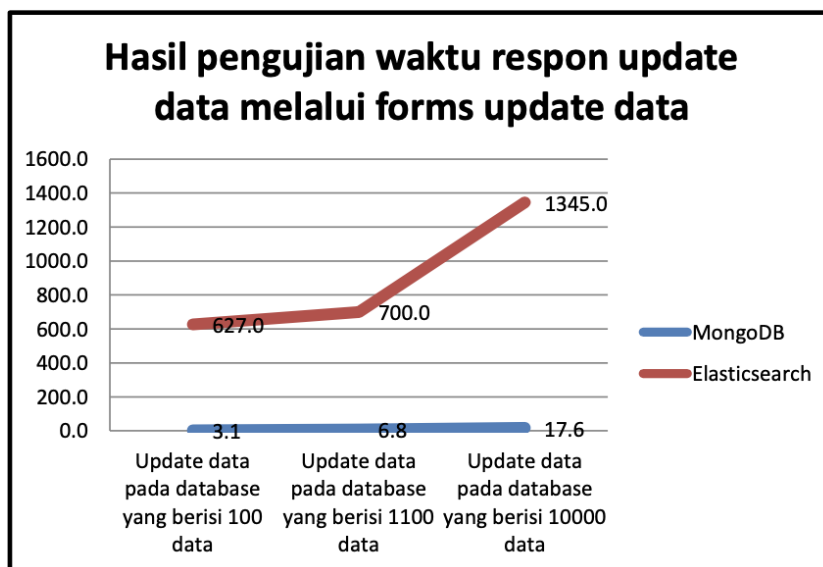


pengujian kedua adalah operasi *update* suatu data di dalam *database* berisi 1000 data, dan pengujian ketiga adalah operasi *update* suatu data di dalam *database* berisi 8900 data.

Tabel 6 Tabel Perbandingan Waktu Respon Perintah *Update Data* Melalui *Forms Update Data*

<i>Database</i>	<i>Update data pada database berisi 100 data</i>	<i>Update data pada database berisi 1000 data</i>	<i>Update data pada database berisi 8900 data</i>
Jumlah Data			
MongoDB	3,1 ms	6,8 ms	17,6 ms
Elasticsearch	627 ms	700 ms	1345 ms

Waktu respon pengujian pertama yang dibutuhkan aplikasi *database* MongoDB adalah 3,1 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 627 ms. Waktu respon pengujian kedua yang dibutuhkan aplikasi *database* MongoDB adalah 6,8 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 700 ms. Waktu respon pengujian ketiga yang dibutuhkan aplikasi *database* MongoDB adalah 17,6 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 1345 ms. Gambar 9 adalah data hasil pengujian operasi *update data* yang disajikan dalam bentuk grafik. Pengujian *update data* melalui *forms update data* dapat dinyatakan bahwa aplikasi *database* MongoDB masih memiliki waktu respon lebih cepat dibandingkan dengan Elasticsearch, nilai rata-rata waktu respon aplikasi *database* MongoDB yaitu 91,2 ms, sedangkan nilai rata-rata waktu respon aplikasi *database* Elasticsearch yaitu 890,6 ms. Dengan kata lain, waktu respon yang diperlukan aplikasi *database* MongoDB 9,8 kali lebih cepat dibandingkan aplikasi *database* Elasticsearch.



Gambar 9 Grafik Hasil Pengujian Waktu Respon *Update Data* Melalui *Forms Update Data*

3.5 Pengujian waktu respon perintah *delete data* melalui *forms delete data*

Pengujian waktu respon perintah *delete data* melalui halaman *forms delete data* dilakukan sebanyak tiga kali, pengujian pertama yaitu operasi *delete* suatu data di dalam *database* yang berisi 100 data, pengujian kedua yaitu operasi *delete* suatu data di dalam *database* yang berisi 1000 data, dan pengujian ketiga yaitu operasi *delete* suatu data di dalam *database* yang berisi 8900 data.

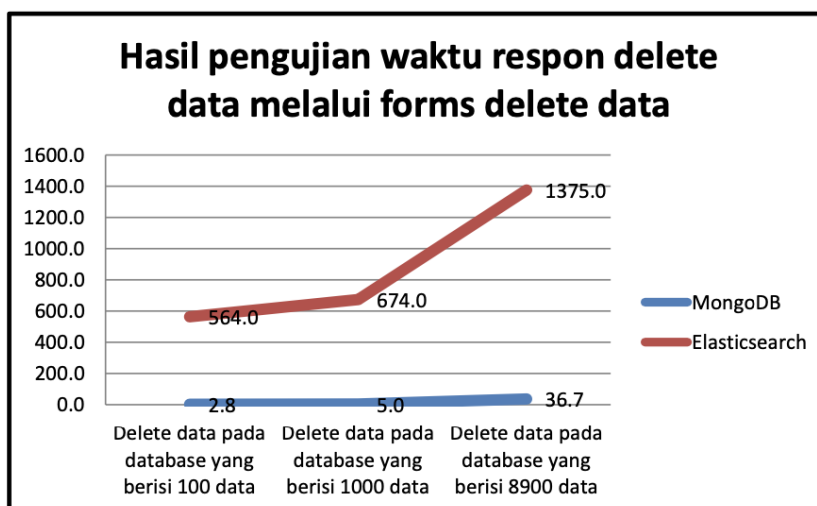


Tabel 7 Tabel Perbandingan Waktu Respon Perintah *Delete Data* Melalui *Forms Delete Data*

<i>Database</i>	<i>Delete data pada database berisi 100 data</i>	<i>Delete data pada database berisi 1000 data</i>	<i>Delete data pada database berisi 8900 data</i>
Jumlah Data			
MongoDB	2,8 ms	5,0 ms	36,7 ms
Elasticsearch	564 ms	674 ms	1375 ms

Waktu respon pengujian pertama yang dibutuhkan aplikasi *database* MongoDB adalah 2,8 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 564 ms. Waktu respon pengujian kedua yang dibutuhkan aplikasi *database* MongoDB adalah 5,0 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 674 ms. Waktu respon pengujian ketiga yang dibutuhkan aplikasi *database* MongoDB adalah 36,7 ms, sedangkan waktu respon yang dibutuhkan aplikasi *database* Elasticsearch adalah 1375 ms.

Gambar 10 merupakan data hasil pengujian yang disajikan dalam bentuk grafik. Pengujian *delete data* melalui *forms delete data* dapat dinyatakan bahwa aplikasi *database* MongoDB masih memiliki waktu respon lebih cepat dibandingkan dengan Elasticsearch, nilai rata-rata waktu respon aplikasi *database* MongoDB yaitu 14,8 ms, sedangkan nilai rata-rata waktu respon aplikasi *database* Elasticsearch yaitu 871 ms. Dengan kata lain, waktu respon yang diperlukan aplikasi *database* MongoDB 58,9 kali lebih cepat dibandingkan aplikasi *database* Elasticsearch.



Gambar 10 Grafik Hasil Pengujian Waktu Respon *Delete Data* Melalui *Forms Delete Data*

4. KESIMPULAN

Berdasarkan tabel dan grafik hasil perbandingan pengujian waktu respon, dapat disimpulkan bahwa, waktu respon aplikasi *database* NoSQL MongoDB membutuhkan waktu yang lebih singkat di semua perintah *create*, *read*, *update*, dan *delete* (CRUD) data dibandingkan waktu respon aplikasi *database* NoSQL Elasticsearch.

Pada pengujian perintah *create data* dengan *file* JSON, aplikasi *database* MongoDB 42,5 kali lebih cepat dibanding aplikasi *database* Elasticsearch. Pada pengujian perintah *create* sebuah data ke dalam *database* yang berisi jumlah data yang berbeda-beda, aplikasi *database* MongoDB 333,9 kali lebih cepat dibanding nilai rata-rata waktu respon aplikasi *database* Elasticsearch. Pada pengujian perintah *read* sebuah data di dalam *database* yang berisi jumlah data yang berbeda-beda, aplikasi *database* MongoDB 35,5 kali lebih cepat dibandingkan aplikasi *database* Elasticsearch. Pada pengujian operasi *update* sebuah data di dalam *database* yang berisi jumlah data yang berbeda-beda, aplikasi *database* MongoDB 9,8 kali lebih cepat dibandingkan aplikasi



database Elasticsearch. pada pengujian operasi *delete* sebuah data di dalam *database* yang berisi jumlah data yang berbeda-beda, aplikasi *database* MongoDB 58,9 kali lebih cepat dibandingkan aplikasi *database* Elasticsearch.

Sehingga, pada penelitian ini dapat disimpulkan bahwa aplikasi *database* MongoDB memiliki waktu respon yang lebih cepat dibandingkan aplikasi *database* Elasticsearch berdasarkan waktu respon saat operasikan perintah *create*, *read*, *update*, dan *delete data* menggunakan aplikasi web sederhana.

DAFTAR PUSTAKA

- Ahmed, M. R., Khatun, M. A., Ali, M. A., & Sundaraj, K. (2018). A literature review on NoSQL database for big data processing. *International Journal of Engineering & Technology*, 7(2), 902–906. <https://doi.org/10.14419/IJET.V7I2.12113>
- Aydoğan, T., İlkuçar, M., & AKCA, M. A. (2016). An Analysis on the Comparison of the Performance and Configuration Features of Big Data Tools Solr and Elasticsearch. *International Journal of Intelligent Systems and Applications in Engineering*, 4(Special Issue-1), 8–12. <https://doi.org/10.18201/ijisae.271328>
- Amandeep, K., & Singh, D. K. (2016). Performance Evaluation For Crud Operations In NoSQL Databases. *I-Manager's Journal on Cloud Computing*, 3(2), 1. <https://doi.org/10.26634/jcc.3.2.8164>
- Bhaswara, F. A., Sarno, R., & Sunaryono, D. (2017). Perbandingan Kemampuan Database NoSQL dan SQL dalam Kasus ERP Retail. *Jurnal Teknik ITS*, 6(2), A511–A514. <https://doi.org/10.12962/J23373539.V6I2.24031>
- Chauhan, A. (2019). A Review on Various Aspects of MongoDB Databases. *International Journal of Engineering Research & Technology*, 8(5). <https://doi.org/10.17577/IJERTV8IS050031>
- Damodaran, D., Salim, S., & Vargese, S. M. (2016). Performance Evaluation of MySQL and MongoDB Databases. *International Journal on Cybernetics & Informatics (IJCI)*, 5(2). <https://doi.org/10.5121/ijci.2016.5241>
- Gunawan, R. (2018). Pengukuran Query Respon Time pada NoSQL Database Berbasis Document Stored. *Jurnal Siliwangi Seri Sains Dan Teknologi*, 4(2). <https://jurnal.unsil.ac.id/index.php/jssainstek/article/view/609>
- Halimi, A., & Sudarmanto, A. (2021). Analisis Perbandingan Kinerja Waktu Respon MySQL 8.0 dan NoSQL MongoDB Menggunakan REST API NodeJS pada Studi Kasus Kelas Online. *Jurnal Informatika Wicida*, 10(1), 26–33. <https://doi.org/10.46984/INF-WCD.1185>
- Kriestanto, D., & Arnado, A. B. (2017). Implementasi Website Pencarian Kos dengan NoSQL. *JIKO (Jurnal Informatika Dan Komputer)*, 2(2), 103–108. <https://doi.org/10.26798/JIKO.V2I2.66>
- Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M., & Bernardino, J. (2015). Choosing the right NoSQL database for the job: a quality attribute evaluation. *Journal of Big Data*, 2(1), 1–26. <https://doi.org/10.1186/S40537-015-0025-0/TABLES/2>
- Modhiya, K. (2021). Introduction to DBMS, RDBMS, and NoSQL Database: NoSQL Database Challenges. *SSRN Electronic Journal*. <https://doi.org/10.2139/SSRN.3798989>
- Renaldi, R., Santoso, B. C., Natasya, Y., willian, steven, & alfando, fladianand. (2020). Tinjauan Pustaka Sistematis terhadap Basis Data MongoDB. *Jurnal Inovasi Informatika*, 5(2), 132–142. <https://doi.org/10.51170/JII.V5I2.79>
- Tavares, O. M. I., Rangkoly, S. M., Desy Bawan, S. B., Utami, E., & Mustafa, M. S. (2020). Analisis Perbandingan Performansi Waktu Respons Kueri antara MySQL PHP 7.2.27 dan NoSQL MongoDB. *Jurnal Teknologi Informasi*, 4(2), 303–313. <https://doi.org/10.36294/jurti.v4i2.1695>
- Wicaksana, I. G. N. A. (2017). *Sinkronisasi Basis Data Sql dengan Basis Data Nosql Menggunakan Data Adapter dengan Pendekatan Query Direct Access*. Institut Teknologi Sepuluh Nopember.
- Yafet, T. T. (2020). *Analisis Pemanfaatan NoSQL Database Elasticsearch pada Mesin Pencarian Tokopedia* [Universitas Atma Jaya]. <http://e-journal.uajy.ac.id/23338/>

