

PENGUKURAN QUERY RESPON TIME PADA NOSQL DATABASE BERBASIS DOCUMENT STORED

Rohmat Gunawan

Jurusan Informatika Fakultas Teknik Universitas Siliwangi Tasikmalaya
e-mail: rohmatgunawan@unsil.ac.id

Abstrak

Pesatnya perkembangan internet dan *cloud computing* telah mendorong tersedianya *database* untuk dapat menyimpan dan memproses data besar secara efektif serta menuntut kinerja tinggi saat membaca dan menulis. *NoSQL database* merupakan salah satu solusi yang dapat digunakan untuk menangani masalah tersebut. Database berbasis *document stored* merupakan salah satu jenis *NoSQL database* yang digunakan saat ini. Penelitian ini bertujuan untuk mengukur *response time* dari *query* yang dilakukan pada *NoSQL database* berbasis *document stored*. Hasil percobaan pada penelitian ini menunjukkan bahwa *query read* data pada *nosql database* memiliki *response times* yang paling cepat dibanding *query* untuk proses *create*, *update* dan *delete*. *Query update* data pada *nosql database* memiliki *response times* yang paling lama dibanding *query* untuk proses *create*, *read* dan *delete*. Sedangkan *query* untuk proses *delete* data pada *nosql database* memiliki *response times* yang lebih cepat dibanding *query create* data.

Kata Kunci : Document Stored, NoSQL, Response Time, Query.

Abstract

The rapid development of the internet and cloud computing has encouraged the availability of databases to be able to store and process large data effectively and demand high performance when reading and writing. NoSQL database is one solution that can be used to handle the problem. Document stored based database is one type of NoSQL database used today. This study aims to measure the response time of query performed on NoSQL document stored based database. The experimental results in this study show that the query read data in the nosql database has the fastest response times compared to the query for the create, update and delete processes. The update data query in the nosql database has the longest response times compared to the query for the create, read and delete process. While the query for deleting data in the nosql database has response times faster than the create data query.

Keywords: Document Stored, NoSQL, Response Time, Query.

I. PENDAHULUAN

Pesatnya perkembangan internet dan komputasi awan (*cloud computing*) telah mendorong ketersediaan *database* untuk dapat menyimpan dan memproses data besar secara efektif serta menuntut kinerja tinggi saat membaca dan menulis. *NoSQL Database* merupakan salah satu solusi yang dapat digunakan untuk menangani masalah tersebut. *NoSQL Database* berbasis *document stored* merupakan salah satu jenis database *NoSQL* yang terdapat pada saat ini.

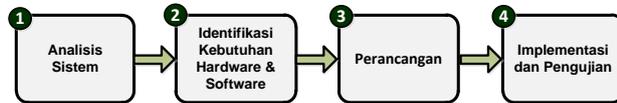
NoSQL database dapat bekerja lebih cepat dibandingkan basis data relasional. Pertumbuhan website yang sangat pesat menyebabkan berkembangnya *NoSQL* karena menjadi alternatif untuk mempercepat akses dibanding menggunakan basis data relasional [1]. Pengukuran kecepatan dalam *NoSQL* merupakan salah satu topik yang

menarik untuk dikaji dalam penelitian [2]. Beberapa penelitian yang berhubungan dengan pengukuran kecepatan, perbandingan performa pernah dilakukan dalam penelitian sebelumnya, diantaranya: perbandingan kinerja *data manipulation language nosql* dengan *relational database* [3], perbandingan performa *nosql* dengan *relational database* [4],[5],[6], studi perbedaan konseptual *nosql* dengan *relational database*[7]. Menurut survey [8], MongoDB menduduki peringkat pertama urutan *NoSQL database* berbasis *document stored*.

Berdasarkan latar belakang tersebut pada penelitian ini akan dilakukan pengukuran *query respon time* dari perintah *create*, *read*, *update* dan *delete* pada *NoSQL database* berbasis *document stored*. MongoDB dipilih pada penelitian ini untuk dijadikan *tools* pada saat pengukuran *query response times*.

II. METODOLOGI

Terdapat empat tahap yang dilakukan pada penelitian ini yaitu : analisis sistem, identifikasi kebutuhan perangkat keras (*hardware*) & perangkat lunak (*software*), perancangan, implementasi dan pengujian seperti ditampilkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1 Analisis Sistem

Pada tahap ini dilakukan analisis terhadap ruang lingkup sistem secara umum dari percobaan yang akan dikembangkan. Pengukuran *query response time* ini akan melibatkan *software*, *hardware* serta *sample data* untuk diinput pada *NoSQL database*.

2.2 Identifikasi Kebutuhan Hardware dan Software

Pada tahap ini dilakukan identifikasi *hardware* dan *software* yang diperlukan pada proses pengukuran *query response times*. Spesifikasi *hardware* dan *software* yang digunakan ditampilkan pada Tabel 1 dan Tabel 2.

Tabel 1. Spesifikasi *Software* yang Digunakan

No	Software	Version
1	Microsoft Windows	8.1
2	MongoDB	3.4
3	Robo3T (MongoDB GUI Client)	1.2.1

Tabel 2. Spesifikasi *Hardware* yang Digunakan

No	Item	Description
1	Processor	Intel Core i3-2328 M dual-core @ 2,20 GHz (4 CPU)
2	Memory	4 GB
3	Graphic	Intel HD Graphics 3000
4	Harddisk	500 GB

2.3 Perancangan

Pada tahap ini dilakukan perancangan teknis yang dilakukan pada tahap percobaan. Beberapa hal yang perlu disiapkan sebelum dilakukan teknis proses pengukuran diantaranya :

a. Persiapan data

Data yang akan dijadikan percobaan pada penelitian ini bertipe string. Entitas data yang diinputkan yaitu data buku dengan atribut : judul, penerbit dan pengarang.

b. Rancangan percobaan

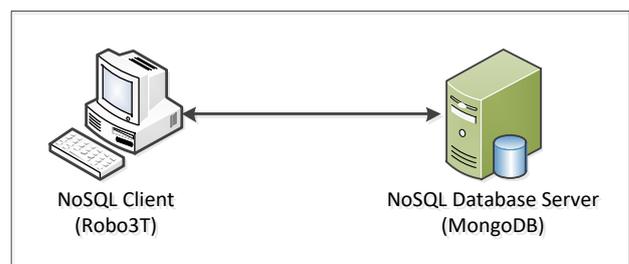
Percobaan dilakukan dengan memproses query secara berulang sebanyak : 100, 200, 400, 800, 1600, 3200, 6400, 12.800. *Query* yang dilakukan yaitu untuk proses *create*, *read*, *update* dan *delete*.

c. Pengukuran

Setiap proses query yang dilakukan kemudian dihitung waktu respon time yang dihasilkannya. Data hasil percobaan selanjutnya diinputkan ke dalam tabel dan disajikan dalam bentuk grafik.

2.4 Implementasi dan Pengujian

Pada tahap ini dilakukan instalasi *software* (*MongoDB Server dan Client*) pada *hardware* yang telah disiapkan. Kemudian dilakukan konfigurasi pada *nosql database server* serta *client* agar terkoneksi, dan proses percobaan dapat dilakukan. Secara umum rancangan arsitektur sistem yang dibuat ditampilkan pada Gambar 2.



Gambar 2. Arsitektur Sistem Koneksi NoSQL Client dengan NoSQL Database Server

III. HASIL DAN PEMBAHASAN

3.1 Implementasi Query

Tahapan awal dimulai dengan pembuatan query untuk proses pengujian. Secara umum sintaks *query* untuk setiap proses : *create*, *read*, *update*, *delete* ditampilkan pada Gambar 3-6.

```

for (var i = 1; i <= 100; i++){
  db.buku.insert({
    judul: "Algoritma dan Pemrograman",
    penerbit: "Informatika",
    pengarang: "Rinaldi Munir"
  })
}
    
```

Gambar 3. Sintaks *query* untuk *create data*

Pada Gambar 3 ditampilkan sintaks *query* untuk proses *create* atau *insert* data. Pada percobaan pertama dilakukan perulangan sebanyak 100 kali dengan menggunakan sintaks *for-do*.

```
db.buku.find({})
```

Gambar 4. Sintaks *query* untuk *read* data

Pada Gambar 4 ditampilkan sintaks *query* untuk proses *read* atau menampilkan data. Perintah menampilkan data tanpa parameter seperti yang ditampilkan pada Gambar 4 akan menampilkan seluruh data yang terdapat pada database.

```
for (var i = 1; i <= 100; i++){
  db.buku.update({judul: "Algoritma dan
  Pemrograman"},{judul: "Algoritma dan
  Pemrograman dalam Bahasa Pascal dan C",
  penerbit: "Informatika",
  pengarang: "Rinaldi Munir"
})
}
```

Gambar 5. Sintaks *query* untuk *update* data

Pada Gambar 5 ditampilkan sintaks *query* untuk proses *update* data. *Update* data dilakukan pada atribut judul. Data buku dengan judul “**Algoritma dan Pemrograman**”, setelah dilakukan proses *update* berubah menjadi “**Algoritma dan Pemrograman dalam Bahasa Pascal dan C**”.

```
db.buku.remove({})
```

Gambar 6. Sintaks *query* untuk *delete* data

Pada Gambar 6 ditampilkan sintaks *query* untuk proses *delete* data. Perintah *delete* data tanpa parameter seperti yang ditampilkan pada Gambar 6, akan menyebabkan terhapusnya seluruh data yang terdapat pada *database* setelah *query* tersebut dieksekusi.

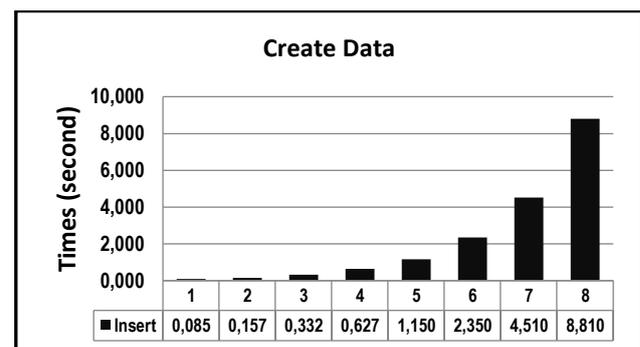
3.1 Perbandingan Query Response Times

Query yang telah dipersiapkan sebelumnya kemudian dieksekusi secara berulang. Hasil percobaan eksekusi *query* ditampilkan pada Tabel 3.

Tabel 3. Hasil Pengukuran *Query Response Times*

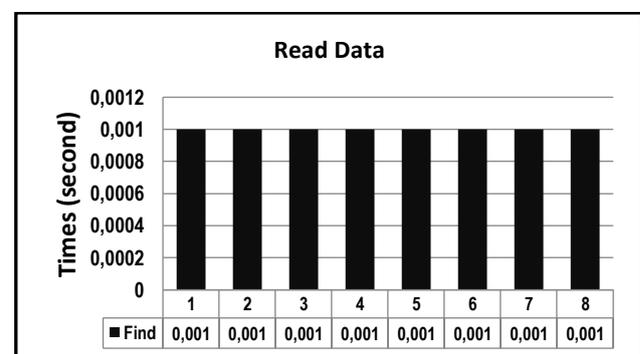
No	Jumlah Data	Create (second)	Read (second)	Update (second)	Delete (second)
1.	100	0,085	0,001	0,099	0,006
2.	200	0,157	0,001	0,199	0,005
3.	400	0,332	0,001	0,429	0,009
4.	800	0,627	0,001	1,000	0,017
5.	1.600	1,150	0,001	2,480	0,033
6.	3.200	2,350	0,001	7,010	0,169
7.	6.400	4,510	0,001	23,30	0,151
8.	12.800	8,810	0,001	81,00	0,368
Rata-rata		2,253	0,001	14,44	0,095

Tabel 3 menampilkan data *query response times* dari 8 kali percobaan untuk setiap proses *create*, *read*, *update* dan *delete*.



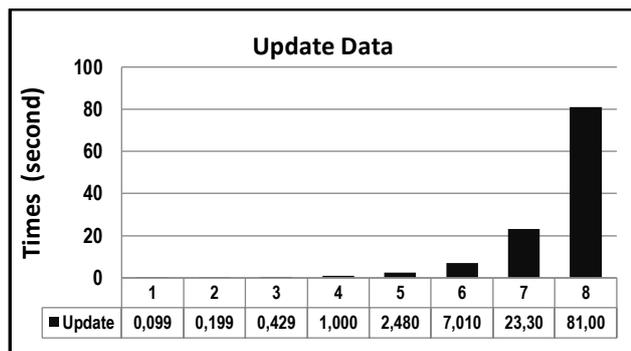
Gambar 7. *Response Time Query Create Data*

Pada Gambar 7 ditampilkan grafik kenaikan waktu dari *query response times* untuk proses *create* data yang dilakukan dari 8 kali percobaan. Waktu *response times* untuk proses *create* data bertambah sesuai jumlah data yang diinputkan.



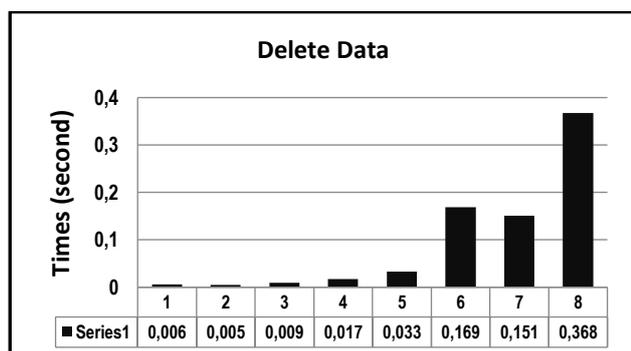
Gambar 8. *Response Time Query Read Data*

Pada Gambar 8 ditampilkan grafik waktu dari *query response times* untuk proses *read* data. Waktu *response times* untuk proses *read* data cenderung konstan setelah dilakukan 8 kali percobaan.



Gambar 9. *Response Time Query Update Data*

Pada Gambar 9 ditampilkan grafik kenaikan waktu dari *query response times* untuk proses *update* data yang dilakukan dari 8 kali percobaan. Waktu *response times* untuk proses *update* data bertambah sesuai banyaknya data yang diperbaharui.



Gambar 10. *Response Time Query Delete Data*

Pada Gambar 10 ditampilkan grafik waktu dari *query response times* untuk proses *delete* data yang dilakukan dari 8 kali percobaan. Waktu *response times* untuk proses *delete* data bertambah sesuai jumlah data yang dihapus.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil percobaan pada penelitian ini, diketahui bahwa : Query read data pada nosql database memiliki response times yang paling cepat dibanding query untuk proses create, update dan delete. Query update data pada nosql database memiliki response times yang paling lama dibanding query untuk proses create, read dan delete. Sedangkan Query delete data pada nosql database memiliki response times yang lebih cepat dibanding query create data.

Perbandingan query response times dari beberapa software NoSQL database merupakan salah satu topik yang dapat dikembangkan pada penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] M. O. Fitri, "Trend penggunaan nosql untuk basis data non relasional," pp. 120–127, 2013.
- [2] G. Nadzari, "Isu Terkini pada basis data NOSQL," *Knsi 2018*, pp. 711–716, 2018.
- [3] A. W. R. Emanuel and J. Sentosa, "Perbandingan Kinerja Data Manipulation Language MongoDB dan SQL Server," pp. 155–159, 2013.
- [4] A. Junaidi, "Studi Perbandingan Performansi Antara MongoDB dan MySQL Dalam Lingkungan Big Data," *Pros. Annu. Res. Semin. 2016*, vol. 2, no. 1, pp. 460–465, 2016.
- [5] Y. Li and S. Manoharan, "A performance comparison of SQL and NoSQL databases," *IEEE Pacific RIM Conf. Commun. Comput. Signal Process. - Proc.*, no. November, pp. 15–19, 2013.
- [6] M. Silalahi, "Computer Based Information System Journal PERBANDINGAN PERFORMANSI DATABASE MONGODB DAN MYSQL DALAM APLIKASI FILE MULTIMEDIA BERBASIS WEB Mesri Silalahi , Didi Wahyudi," *CBIS J.*, vol. 01, pp. 63–78, 2018.
- [7] A. Nugroho and E. Winarko, "STUDI PERBANDINGAN PERBEDAAN KONSEPTUAL ANTARA SISTEM BASIS DATA RELASIONAL DENGAN SISTEM PENYIMPANAN DATA BERTIPE NON-RELASIONAL (NO-SQL): EKSPLORASI PADA SERVER DATA CASSANDRA," *J. Teknol. TECHNOSCIENTIA*, vol. 6, no. 1, 2013.
- [8] SolidIT, "DB-Engines Ranking of Document Stores," 2018. [Online]. Available: <https://db-engines.com/en/ranking/document+store>.