

Analisis performa kecepatan MySQL dan NoSQL pada sistem operasi Windows dan Linux.

Setiawan Budiman¹, Vian Ardiyansyah Saputro^{**2} dan Faisal Fadhila³

- 1 Akademi Sages
Jl. Puncak Permai II No.28, Tanjungsari, Kec. Sukomanunggal, Kota Surabaya, Jawa Timur 6.
setiawan.1267@students.amikom.ac.id
- 2 LKP Acavi Computech Cikarang.
Perumahan Taman Aster Blok FD1 No. 16, Telaga Asih, Cikarang Barat, Kabupaten Bekasi
vian.1258@students.amikom.ac.id
- 3 Akademi Militer Magelang
Jalan Gatot Subroto Jurangombo Selatan Magelang Selatan, Seneng Satu, Banyurojo, Kec. Mertoyudan, Magelang, Jawa Tengah
faisal.1261@students.amikom.ac.id

Abstrak

Sistem basis data sudah menjadi bagian penting dalam pengelolaan suatu institusi berbasis digital. Saat ini semakin banyak pilihan dalam menggunakan jenis sistem basis data yang dapat disesuaikan dengan kebutuhan penggunaannya. Banyak faktor yang bisa digunakan sebagai dasar pemilihan suatu sistem basis data. Kecepatan akses seperti untuk merespon kueri yang dikirimkan ke server adalah salah satunya. Pada penelitian ini dilakukan analisa perbandingan kecepatan pada sistem basis data MySQL dan MongoDB untuk perintah SELECT, UPDATE dan DELETE pada data dengan jumlah data maksimum 1.000.000 baris. Aplikasi yang digunakan untuk melakukan perintah kueri adalah MySQL Workbench 8.0.22 dan Robo 3T 1.4.2 yang dilakukan pada sistem operasi Windows 10 dan Ubuntu Desktop 20.04. Hasil dari penelitian ini didapatkan bahwa semua proses kueri data di atas 50.000 baris dengan menggunakan MongoDB lebih cepat dibandingkan pada MySQL. Sistem operasi Ubuntu Desktop 20.04 menghasilkan kecepatan lebih baik dibandingkan dengan menggunakan Windows 10 baik untuk sistem basis data MySQL atau MongoDB.

Kata Kunci database, sql, nosql, windows 10, ubuntu, perbandingan

1 Pendahuluan

Dalam sebuah aplikasi, semakin banyak data tersimpan di dalam suatu sistem basis data maka secara umum akan mempengaruhi kecepatan akses dari sistem informasi yang menggunakannya. Berbagai upaya dapat dilakukan untuk memperbaiki performa dalam akses sistem basis data tersebut. Dari cara yang paling sederhana yakni dengan menambah kemampuan perangkat keras seperti memori ataupun *processor* namun cara ini memerlukan biaya yang cukup besar. Teknik lain yang bisa digunakan untuk memperbaiki performa kecepatan akses ke sistem basis data adalah melakukan optimasi kueri. Berbagai teknik optimalisasi kueri telah menjadi bahan kajian penelitian seperti yang disebut dalam [1]. Dari penggunaan teknik

* Corresponding author.



sederhana seperti indeks [2] sampai dengan penggunaan algortime kompleks seperti genetik dan memetics [3].

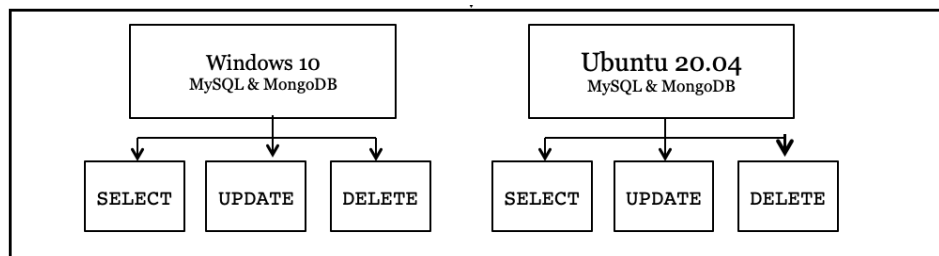
Pemilihan teknik atau metode yang diterapkan untuk memperbaiki performa akses ke sistem manajemen basis data bergantung dari kompleksitas permasalahan yang dihadapi. Pada penelitian ini dilakukan teknik sederhana untuk membandingkan performa kecepatan akses ke sistem basis data MySQL dan MangoDB. Perbandingan kedua sistem basis data ini sebenarnya sudah cukup banyak dilakukan seperti oleh [4] ataupun [5]. Penelitian lainnya mengenai evaluasi performa database NoSQL dan MySQL dilakukan dalam [6]. Kemudian pada penelitian lain mengenai perbandingan kinerja Mongoddb dan MySQL untuk *big data* dilakukan dalam [7].

Hasil dari berbagai penelitian tersebut secara umum menyebutkan bahwa MangoDB memiliki performa lebih baik untuk data yang besar. Namun demikian dari berbagai penelitian tersebut belum terlihat secara jelas disebutkan mulai dari berapa jumlah data sistem basis data MangoDB lebih baik digunakan dibandingkan dengan MySQL. Penelitian ini memiliki tujuan yang sama dengan penelitian sejenisnamun penelitian ini perbedaan pada adanya perbandingan 2 jenis sistem operasi. Dan jumlah data yang digunakan dalam penelitian ini lebih besar yakni sampai dengan 1 juta baris. Penelitian ini juga bertujuan untuk mengetahui berapa jumlah *record* yang menjadi pembeda dari 2 sistem manajemen basis data tersebut.

2 Metodologi

Tiga perintah dasar yang umum digunakan untuk melakukan akses ke sistem basis data relasional yakni SELECT, UPDATE dan DELETE digunakan dalam penelitian ini. Pada sistem basis data MySQL digunakan aplikasi MySQL Workbench 8.0.22 sedangkan untuk MongoDB digunakan aplikasi Robo 3T 1.4.2 untuk melakukan perintah yang serupa. Untuk mendapatkan hasil perhitungan waktu eksekusi kueri, data didapatkan dari waktu yang dihasilkan oleh masing-masing program yaitu MySQL Workbench 8.0.22 dan MongoDB Robo 3T 1.4.2. Secara umum metodologi yang digunakan penelitian ini mereduplikasi yang dilakukan dalam penelitian [4] dengan beberapa penyesuaian.

Penelitian ini sistem operasi di *install* pada komputer CPU yang sama yakni menggunakan *processor* Intel(R) Core (TM) i5-7200U 3.10 Ghz, RAM 8 GB dengan media penyimpanan 160 GB 5400 RPM. Sistem operasi Windows 10 di *install* pada *harddisk* dengan alokasi ruang sebesar 80 GB. Kemudian Ubuntu Desktop 20.04 di *install* pada sisa kapasitas 70 GB pada *harddisk* yang sama. Gambar 1 memperlihatkan skema perbandingan yang dilakukan dalam penelitian ini.



Gambar 1 Skema perbandingan penelitian

Data *set* sebanyak 1 juta baris *record* didapatkan dari analisis sentimen Twitter yang didownload dari kaggle.com (<https://www.kaggle.com/kazanova/sentiment140>). Dataset

asli sejumlah 1,6 juta *record* didapatkan dengan Twitter API diambil sebanyak 1 juta baris untuk digunakan dalam penelitian ini. Data tersebut memiliki beberapa field yaitu: *target*, *ids*, *date*, *query*, *user*, *text*. Data awal berbentuk format CSV yang dipisah dengan tanda koma (,) diimport dan dimasukkan dalam MySQL dan MangoDB.

2.1 Proses SELECT, UPDATE dan DELETE

Pada tahap ini, dilakukan kueri SELECT untuk filter data yang ada di dalam field text, sehingga diharapkan akan banyak data yang didapatkan. Batasan yang digunakan adalah WHERE TEXT LIKE '%is%'. Dilakukan dengan limit data 5.000, 10.000, 50.000, 100.000, 500.000, 1.000.000. Salah satu perbandingan perintah SELECT yang digunakan dalam penelitian ini adalah seperti tertampil dalam Gambar 2.

Salah satu perintah MySQL:
 SELECT * FROM tweet0 WHERE TEXT LIKE '%is%' LIMIT 5000

Salah Satu perintah MongoDB:
 db.tweet0.find({ "text" : /.is./ }).limit(5000).explain
 ("executionStats")

■ **Gambar 2** Perintah SELECT di MySQL dan MongoDB

Pada proses UPDATE penelitian dilakukan dengan 2 tahapan. Tahapan pertama yaitu membuat tabel baru dengan jumlah *record* berjumlah 5.000, 10.000, 50.000, 100.000, 500.000, dan 1.000.000. Tahapan ini digunakan perintah seperti tertampil pada Gambar 3.

Salah satu perintah MySQL:
 SELECT * FROM tweet0 WHERE TEXT LIKE '%is%' LIMIT 5000

Salah Satu perintah MongoDB:
 db.tweet0.find({ "text" : /.is./ }).limit(5000).explain
 ("executionStats")

■ **Gambar 3** Perintah pembuatan tabel baru

Tahapan kedua adalah melakukan perintah UPDATE pada masing-masing tabel yang telah memiliki jumlah data sebanyak 5.000, 10.000, 50.000, 100.000, 500.000, dan 1.000.000 *record*. Tahapan kedua ini dilakukan dengan perintah seperti pada Gambar 4.

Salah satu perintah MySQL:
 SELECT * FROM tweet0 WHERE TEXT LIKE '%is%' LIMIT 5000

Salah Satu perintah MongoDB:
 db.tweet0.find({ "text" : /.is./ }).limit(5000).explain
 ("executionStats")

■ **Gambar 4** Perintah untuk melakukan UPDATE

Proses DELETE ini dilakukan dengan menggunakan kondisi filter kata yang selalu ada pada semua *record*, dengan tujuan agar dapat menghasilkan jumlah keluaran yang lebih banyak. Pada proses DELETE ini digunakan tabel saat pengujian perintah UPDATE

sebelumnya, yang sudah dibagi sesuai dengan jumlah data sebanyak 5.000, 10.000, 50.000, 100.000, 500.000, dan 1.000.000 *record*. Perintah yang digunakan dalam proses ini tersaji dalam Gambar 5.

Salah satu perintah MySQL:
`SELECT * FROM tweet0 WHERE TEXT LIKE '%is%' LIMIT 5000`

Salah Satu perintah MongoDB:
`db.tweet0.find( { "text" : /. *is.*/ } ).limit(5000).explain("executionStats")`

■ **Gambar 5** Perintah penghapusan data

Seluruh pengujian dilakukan secara berurutan pada OS Windows 10 dengan MySQL & MongoDB, dilanjutkan sistem operasi Ubuntu Desktop 20.04 dengan MySQL & MongoDB.

3 Hasil dan pembahasan

Penelitian pertama yang dilakukan adalah dengan menguji perintah SELECT, UPDATE dan DELETE menggunakan sistem operasi Windows 10. Tabel 1 merupakan hasil dari perintah SELECT dengan 6 perbedaan jumlah data.

■ **Tabel 1** Pengujian waktu kueri di Windows 10

Jumlah baris data	waktu SELECT (detik)		waktu DELETE (detik)		waktu UPDATE (detik)	
	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB
5.000	0,250	0,003	1,781	0,154	2,547	0,133
10.000	0,281	0,002	2,14	0,193	1,984	0,220
50.000	0,969	0,130	9,500	1,03	10,047	1,380
100.000	2,594	0,232	29,593	2,080	30,594	3,360
500.000	7,375	0,887	174,500	11,00	181,900	16,500
1.000.000	7,391	0,918	413,000	21,800	346,900	39,500

Dari Tabel 1 nampak bahwa perbedaan kecepatan mulai nampak terjadi gap cukup besar saat jumlah *record* data yang dikueri di angka 50.000 seperti yang digambarkan di Gambar 6.



■ **Gambar 6** Grafik perbandingan kueri di Windows 10

Penelitian kedua adalah dengan menguji perintah SELECT, UPDATE dan DELETE menggunakan sistem operasi Ubuntu 20.04. Hasil dari pengujian ini tertampil pada Tabel 2.

■ **Tabel 2** Pengujian waktu kueri di Ubuntu versi 20.04

Jumlah baris data	waktu SELECT (detik)		waktu DELETE (detik)		waktu UPDATE (detik)	
	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB
5.000	0,034	0,025	1,153	0,115	0,549	0,110
10.000	0,069	0,048	0,301	0,144	0,817	0,193
50.000	0,318	0,222	0,875	0,75	3,097	1,000
100.000	0,639	0,453	1,186	1,040	4,701	2,050
500.000	2,730	1,750	87,000	4,950	98,400	6,060
1.000.000	2,732	1,750	368,300	10,200	196,000	11,300

Pada sistem operasi Ubuntu vesi 20.04 terjadinya gap perbedaan waktu perintah kueri pada kedua sistem manajemen basis data ini nampak lebih bergeser saat jumlah data di angka 100.000, seperti nampak di Gambar 7.



■ **Gambar 7** Grafik perbandingan kueri di Ubuntu 20.04

Hasil pengujian dalam penelitian ini. menunjukkan bahwa penggunaan sistem operasi Linux Ubuntu 20.04 lebih cepat dibandingkan dengan Windows 10 saat digunakan sebagai server basis data. Pergeseran terjadinya gap perbedaan waktu juga terjadi saat penggunaan sistem operasi Linux Ubuntu diandingkan dengan Windows 10.

4 Kesimpulan

Hasil yang didapatkan bahwa proses perintah kueri pada jumlah data yang lebih besar dari 50.000 akan lebih cepat jika dilakukan pada database MongoDB dibandingkan dengan database MySQL. Untuk perbandingan sistem operasi, menggunakan sistem operasi Ubuntu Desktop 20.04 lebih cepat jika dibandingkan dengan menggunakan sistem operasi Windows 10.

Penelitian ini dilakukan dengan beberapa keterbatasan, yaitu tidak dilakukan pada jenis sistem operasi dan CPU berbasis server. Penelitian selanjutnya, akan dilakukan analisa lebih detail dengan menyertakan kueri INSERT atau yang lebih kompleks. Perbandingan performa MySQL dan MangoDB perlu juga dilakukan pada sistem operasi dan spesifikasi perangkat keras untuk *server*.

Pustaka

- 1 A. R. Thangam dan S. J. Peter, "An extensive survey on various query optimization techniques," *IJCSMC*, vol. 5, no. 8, pp. 148–154, 2016.
- 2 R. Pamungkas, "Optimalisasi query dalam basis data my sql menggunakan index," *Res. J. Comput. Inf. Syst. Technol. Manag*, vol. 1, no. 1, pp. 27–31, 2018.
- 3 J. Kurniasih, E. Utami, dan S. Raharjo, "Heuristics and metaheuristics approach for query optimization using genetics and memetics algorithm," in *2019 1st international conference on cybernetics and intelligent system (ICORIS)*, vol. 1. IEEE, 2019, pp. 168–172.
- 4 C. Györödi, R. Györödi, G. Pecherle, dan A. Olah, "A comparative study: Mongodb vs. mysql," in *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*. IEEE, 2015, pp. 1–6.
- 5 B. Dipina Damodaran, S. Salim, dan S. M. Vargese, "Performance evaluation of mysql and mongodb databases," *Int. J. Cybern. Inform.(IJCI)*, vol. 5, 2016.
- 6 B. Jose dan S. Abraham, "Performance analysis of nosql and relational databases with mongodb and mysql," *Materials today: PROCEEDINGS*, vol. 24, pp. 2036–2043, 2020.
- 7 A. Junaidi, "Studi perbandingan performansi antar mongodb dan mysql menggunakan php dalam lingkungan big data," *SEMNASSTEKNOMEDIA ONLINE*, vol. 5, no. 1, pp. 2–2, 2017.