

Perbandingan Kinerja Database NoSQL Dengan Database SQL Server Pada Pemesanan Tiket Pesawat Online

Erwin Apriliyanto*, Nadea Cipta Laksmita, I Wayan Pandu Swardiana, Kusrini
 Magister Teknik Informatika, Universitas Amikom Yogyakarta
 *erwin.apriliyanto@students.amikom.ac.id

Info Artikel	Abstrak
<p>Kata Kunci : NoSQL, SQL Server, Pemesanan Tiket Pesawat Online</p> <p>Keywords : NoSQL, SQL Server, Online Airplane Ticket Booking</p> <p>Tanggal Artikel Dikirim : 11 Januari 2020 Direvisi : 23 Januari 2020 Diterima : 5 Februari 2020</p>	<p>Pelayanan pemesanan tiket pesawat sudah semakin maju, dimana pemesanan dapat dilakukan melalui aplikasi android/iOS dan melalui web browser pemesanan tiket, tidak lagi harus datang ke agen-agen perjalanan maupun datang ke bandara untuk memesan tiket pesawat. Pada penelitian ini menggunakan database pemesanan tiket online dimana satu database menggunakan database NoSQL dan database lain menggunakan SQL Server. Tujuan dari penelitian ini adalah untuk menguji kinerja kecepatan NoSQL dengan SQL Server dengan perintah Insert, Delete, dan Select. Metode pengujiannya menggunakan 100 record, 500 record, 1000 record, dan 5000 record, dengan masing-masing recordnya diuji empat kali kemudian diambil rata-rata. Hasil penelitian ini adalah perintah Insert database NoSQL memiliki kecepatan 4 kali lipat lebih cepat dari database SQL Server untuk dibawah 500 record, sedangkan diatas 500 record database NoSQL lebih lambat 5 kali lipat, perintah Delete database NoSQL memiliki kecepatannya 3 kali lipat lebih cepat dari database SQL Server, dan perintah Select 1 tabel database NoSQL lebih cepat 55 kali dari database SQL Server, sedangkan 2 tabel database NoSQL lebih lambat 18 kali dari database SQL Server, sedangkan 3 tabel database NoSQL lebih lambat 10 kali dari database SQL Server, sedangkan 4 tabel database NoSQL lebih lambat 16 kali dari database SQL Server.</p>

Abstarct

Flight ticket booking services have become more advanced, where bookings can be made through the android / iOS application and through a web browser, ticket reservations, no longer have to come to travel agents or come to the airport to book plane tickets. In this study using an online ticket booking database where one database uses the NoSQL database and another database uses SQL Server. The purpose of this research is to test the performance of NoSQL speed with SQL Server with the Insert, Delete and Select commands. The testing method uses 100 records, 500 records, 1000 records, and 5000 records, with each record being tested four times and then taken on average. The results of this study are that the NoSQL database Insert command has a speed 4 times faster than the SQL Server database for under 500 records, whereas above 500 NoSQL database records 5 times slower, the Delete NoSQL database command has a speed 3 times faster than the SQL database Server, and the command Select 1 NoSQL database table 55 times faster than SQL Server databases, while 2 NoSQL database tables are 18 times slower than SQL Server databases, while 3 NoSQL database tables are 10 times slower than SQL Server databases, whereas 4 database tables NoSQL is 16 times slower than SQL Server databases.

1. PENDAHULUAN

Kualitas pelayanan dan teknik-teknik pemasaran yang bagus, sangat diperlukan agar perusahaan dapat berkembang dan terus bertahan, bila tidak akan tergeser oleh perusahaan-perusahaan serupa. Pada era globalisasi ini sudah banyak Teknologi Informasi (TI) yang membantu aktivitas perusahaan, tidak hanya sebagai enable technology tapi sudah menjadi fokus utama dalam pengembangan bisnis di masa kini, kecepatan TI ini menjadi pendorong bisnis seperti pemesanan tiket, pemesanan ojek motor atau ojek motor, pemesanan makanan menjadi marak di Indonesia.

Teknologi Informasi ini tentu membutuhkan jaringan internet yang bagus, tanpa itu pengguna/calon pengguna pemesanan tiket, pemesanan ojek motor atau ojek motor, pemesanan makanan akan kesulitan. Penggunaan internet menjadi sangat mudah karena dapat juga dilakukan di handphone masing-masing sehingga pemesanan tiket, pemesanan ojek motor atau ojek motor, pemesanan makanan menjadi mudah.

Pemesanan tiket pesawat online mungkin baru marak 5 tahun kebelakang ini, dahulu memesan tiket pesawat harus datang langsung ke bandara, kemudian bisa pesan di agen-agen travel di sekitar rumah, kemudian perkembangan selanjutnya memesan tiket bisa lewat supermarket-supermarket, dan yang terakhir memesan tiket melalui website dan aplikasi handphone. Pada website ataupun aplikasi handphone, sangatlah mudah memesan tiket pesawat online, tidak hanya bisa memesan nomor tempat duduk yang mana, tapi juga dapat mengetahui harga-harga dari berbagai maskapai penerbangan sesuai kota tujuan yang ingin pelanggan capai.

Saat ini dapat kita ketahui bahwa ada dua jenis database yaitu database relasional dan database non relational. Database relasional mulai dikenal sejak tahun 1970-an model relasional menawarkan cara yang matematis dalam menyusun, menyimpan, dan menggunakan data sedangkan database non relasional dalam membangun datanya dengan menghilangkan constraints yang ada pada database relasional [1].

Ada bermacam-macam database NoSQL yang sering kita jumpai yaitu Redis, MongoDB, Couchbase, Cassandra, dan Hbase. Salah satu database NoSQL yang populer saat ini adalah MongoDB, database ini merupakan basis data yang menggunakan model penyimpanan data berorientasi dokumen [2][13], mongodb disimpan biasanya dengan format JSON atau Binary JSON [3][4][12][14][15], mongodb menyediakan pendukung kecepatan database agar tetap optimal. [11]

Penelitian [5] bertujuan untuk menganalisa dan mengimplementasikan NOSQL database pada LONTAR (Library Automation and Digital Archive) di perpustakaan digital Universitas Indonesia agar dapat meningkatkan kinerja query LONTAR. Skema kerja MongoDB digunakan untuk menghilangkan proses join yang menyebabkan melambatnya query. Pada skema normal (SQL) digunakan 6 proses join dalam menerima dokumen. Selain itu MongoDB juga digunakan dalam mengganti Collection_Data_Field tabel (SQL) menjadi Field Documents pada MangoDB. Kesimpulan penelitian ini adalah MongoDB sedikit lebih lambat pada proses pencarian dimana <15 kecocokan, sedangkan bila >15 kecocokan MongoDB lebih cepat. Pada 79.788 kecocokan MongoDB lebih cepat 54 menit.

Penelitian [6] bertujuan untuk membandingkan antara RDBMS MySQL dengan model hibrida MySQL dan NoSQL MongoDB pada jejaring sosial sederhana di website, sehingga dapat dibandingkan kinerja penyimpanan datanya. Untuk menguji kecepatan membaca dan menulis, digunakan tiga dataset masing-masing sebesar 100.000 data chat, 500.000 data chat, dan 1.000.000 data chat. Selain kecepatan membaca dan menulis, juga di uji pemakaian disk space, pemakaian cpu, dan pemakaian ram. Kesimpulan penelitian ini adalah model database hibrida MySQL dan NoSQL dapat meningkatkan performa database pada ukuran database yang sangat besar. NoSQL MongoDB memiliki kecepatan tulis lebih cepat daripada MySQL serta penggunaan disk space yang lebih sedikit namun penggunaan RAM lebih banyak.

Penelitian [7] bertujuan untuk mengevaluasi waktu respons tiap-tiap query saat proses creating, read, update, dan delete (CRUD). Digunakan tiga database NoSQL yaitu MongoDB, ArangoDB, dan CouchDB. Pengujian dilakukan dengan melakukan proses berulang-ulang pada perintah create, read, update, dan delete dengan kuantitas yang berbeda-beda. Query dilakukan berulang-ulang dengan jumlah sebanyak 1000, 2000, 3000, 4000, dan 5000 kemudian diukur waktu dan dirata-rata waktu responnya. Kesimpulan penelitian ini adalah MongoDB memiliki respon waktu read sebesar 0,017 detik, update sebesar 25,358 detik, dan delete sebesar 0,055 detik sebagai paling cepat dan ArangoDB pada proses create paling cepat yaitu 28,943 detik.

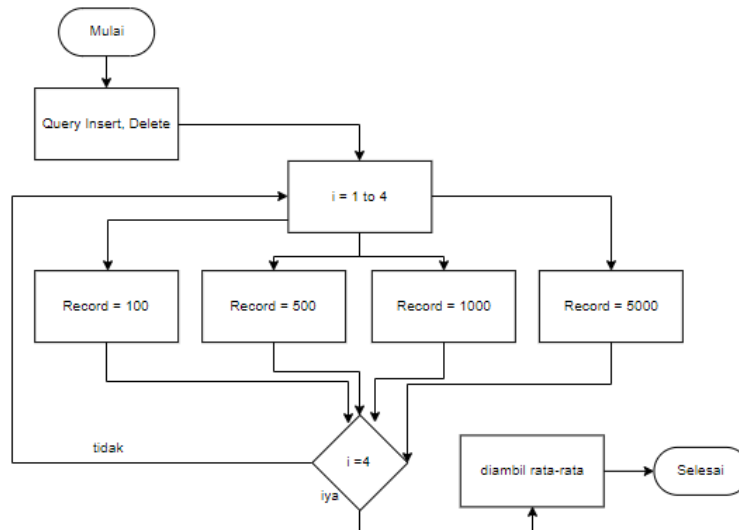
Penelitian [8] bertujuan untuk menganalisa kinerja dari lima NoSQL database yaitu Redis, MongoDB, Couchbase, Cassandra, Hbase dengan tool pengukuran dari YCSB (Yahoo! Cloud Serving Benchmark). Pengujian ini dilakukan sebanyak empat kali, yang pertama dengan mengubah-ubah workload type sebanyak 1000 kali dengan jumlah record 5000, yang kedua dengan mengeksekusi workload A dan mengubah-ubah ukuran record dan operandi pada tiap tes, dan dihasilkan waktu tiap NoSQL database pada loading data dan workload execution. Kesimpulan penelitian ini adalah Redis sangat cocok untuk loading data dan executing workload dengan sedikit limitasi bila menghadapi jumlah data yang sangat besar.

Dari empat penelitian diatas kami menyimpulkan bahwa MongoDB adalah salah satu database NoSQL terbaik [5][6][7], kemudian dari penelitian [7] kami melakukan percobaan dengan jumlah record 100, 500, 1.000, dan 5.000. agar hasilnya lebih

akurat. Pada penelitian kami lebih menekankan pada kecepatan respon baca pada perintah select dengan percobaan 1 tabel, 2 tabel, 3 tabel, dan 4 tabel yang menunjukkan bahwa MongoDB masih relevan pada join antar tabel.

2. METODE PENELITIAN

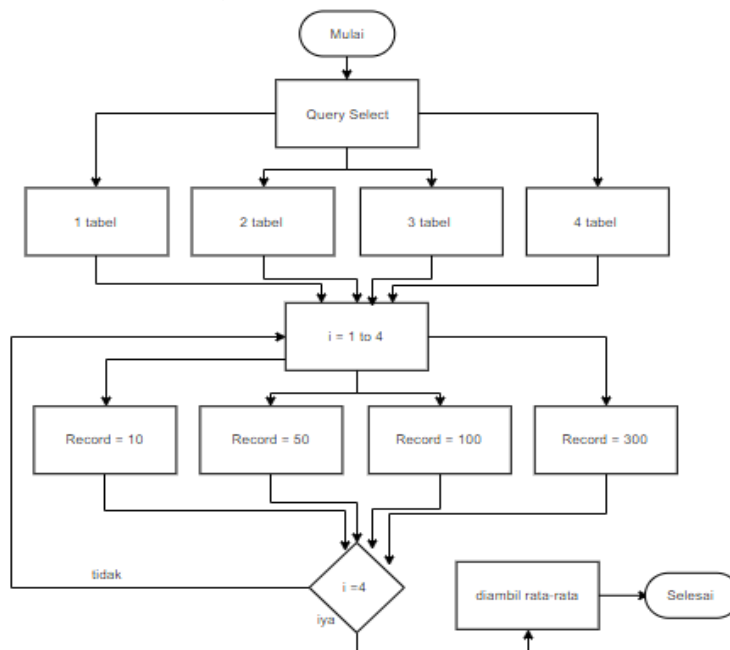
Pada penelitian ini menggunakan database pemesanan tiket online dimana satu database menggunakan database NoSQL dan database lain menggunakan SQL Server. Tujuan dari penelitian ini adalah untuk menguji kinerja kecepatan NoSQL dengan SQL Server dengan perintah Insert, Delete, dan Select.



Gambar 1. Alur Pengujian Perintah Insert

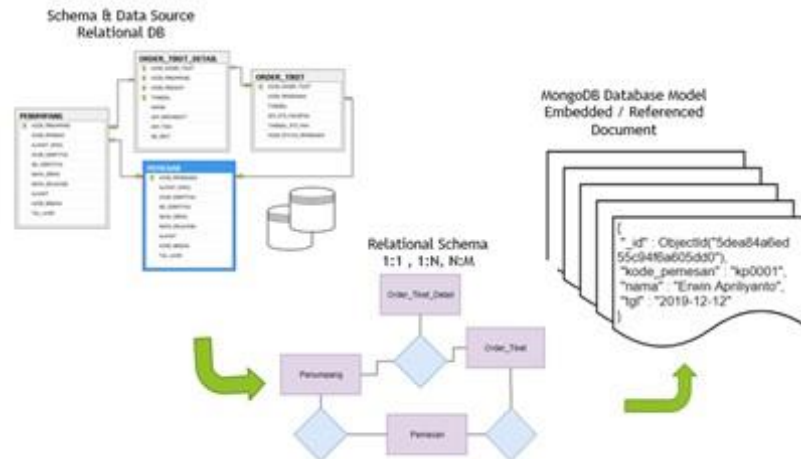
Pada Gambar 1 dapat kita lihat alur pengujian perintah insert dan delete, dimulai dari pengujian 10 record, 100 record, 1.000 record, 5.000 record, kemudian terakhir 10.000 record. Tiap-tiap record diuji lagi sebanyak 4 kali kemudian diambil waktu rata-ratanya.

Pada Gambar 2 dapat kita lihat pengujian perintah select yaitu dengan 1 tabel, 2 tabel, 3 tabel, dan 4 tabel, setelah itu diuji sebanyak lima kali dan diambil rata-rata waktunya.



Gambar 2. Alur Pengujian Perintah Select

Pada penelitian ini skema [9][10] database yang dimaksud adalah struktur dan relasi antar tabel yang ada di dalam basis data. Skema yang dimasukkan tadi akan dianalisis oleh sistem untuk mendapatkan parameter-parameter yang diperlukan pada proses transformasi. Selanjutnya pada proses transformasi, algoritma pembentukan model akan menggunakan parameter tadi untuk mendapatkan model dokumen yang nantinya dapat diimplementasikan pada MongoDB [2]. Secara sederhana, alur proses transformasi ditunjukkan oleh Gambar 3.



Gambar 3. Alur Proses Transformasi Skema Database [2]

Pengujian database Sql server menggunakan tool Navicat 12.0.15 sedangkan pengujian MongoDB menggunakan tools Robo 3T – 1.3. Cara untuk meng-query insert pada MongoDB dapat dilihat pada Gambar 4.

```

db.detail_transaksi.insert({no_transaksi: "44", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
db.detail_transaksi.insert({no_transaksi: "45", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
db.detail_transaksi.insert({no_transaksi: "46", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
db.detail_transaksi.insert({no_transaksi: "47", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
db.detail_transaksi.insert({no_transaksi: "48", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
db.detail_transaksi.insert({no_transaksi: "49", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
db.detail_transaksi.insert({no_transaksi: "50", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
db.detail_transaksi.insert({no_transaksi: "51", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
db.detail_transaksi.insert({no_transaksi: "52", no_urut: "2", rute_awal: "sby", rute_akhir: "jkt", biaya: "1000000"});
    
```

Gambar 4. Perintah Query Insert MongoDB

Kemudian untuk proses query insert pada SQL server dapat dilihat pada Gambar 5.

```

insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('82','2','sby','jkt','1000000')
insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('83','2','sby','jkt','1000000')
insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('84','2','sby','jkt','1000000')
insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('85','2','sby','jkt','1000000')
insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('86','2','sby','jkt','1000000')
insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('87','2','sby','jkt','1000000')
insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('88','2','sby','jkt','1000000')
insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('89','2','sby','jkt','1000000')
insert into detail_transaksi (no_transaksi , no_urut,rute_awal,rute_akhir,biaya ) values ('90','2','sby','jkt','1000000')
    
```

Gambar 5. Perintah Query Insert SQL Server

Kemudian untuk cara meng-query delete pada MongoDB dan SQL server dapat dilihat pada Gambar 6.

```

db.pemesan.remove({});
delete from pemesanan;
    
```

Gambar 6. Perintah Query Delete MongoDB & SQL Server

Kemudian untuk cara meng-query select pada MongoDB dapat dilihat pada Gambar 7.

```

db.getCollection("transaksi").aggregate([
  "$project": {
    "_id": NumberInt(0),
    "t": "$$ROOT"
  }
]),
{
  "$lookup": {
    "localField": "t.no_transaksi",
    "from": "detail_transaksi",
    "foreignField": "no_transaksi",
    "as": "dt"
  }
}),
{
  "$unwind": {
    "path": "$dt",
    "preserveNullAndEmptyArrays": false
  }
}),
{
  "$lookup": {
    "localField": "t.kode_pemesan",
    "from": "pemesan",
    "foreignField": "kd_pemesan",
    "as": "p"
  }
}),
{
  "$unwind": {
    "path": "$p",
    "preserveNullAndEmptyArrays": false
  }
}),
{
  "$limit": NumberInt(10)
}
]),
{
  "allowDiskUse": true
}
];
    
```

Gambar 7. Perintah Query Select MongoDB

Terakhir untuk cara meng-query select pada MongoDB dapat dilihat pada Gambar 8.

```

select top 10 * from transaksi t
inner join detail_transaksi dt on t.no_transaksi = dt.no_transaksi
inner join pemesanan p on p.kode_pemesan = t.kode_Pemesan
    
```

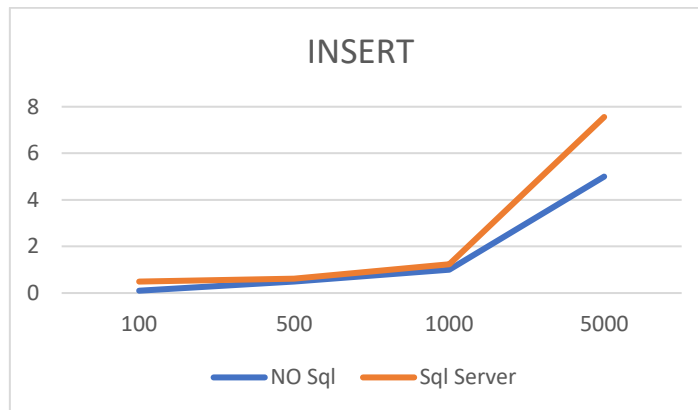
Gambar 8. Perintah Query Select SQL Server

3. HASIL DAN PEMBAHASAN

Grafik Perbandingan Perintah Insert Pada MongoDB dan SQL Server dapat kita lihat pada Gambar 8. Melakukan uji coba ini untuk menguji waktu respon, jika dibebankan dengan banyaknya record dalam waktu yang bersamaan seperti pada tabel 1 sampai tabel 6 dan grafik 8 sampai grafik 13.

Tabel 1. Uji Coba Perintah Insert

Record	Uji coba	Waktu (s)		Rata-rata (s)	
		NoSQL	SQL Server	No SQL	SQL Server
5,000.00	1	5	2.623	5	2.55925
	2	5	2.54		
	3	5	2.511		
	4	5	2.563		
1,000.00	1	1	0.21	1	0.23425
	2	1	0.217		
	3	1	0.292		
	4	1	0.218		
500.00	1	0.5	0.111	0.5	0.11075
	2	0.5	0.116		
	3	0.5	0.093		
	4	0.5	0.123		
100.00	1	0.1	0.32	0.1	0.39225
	2	0.1	0.308		
	3	0.1	0.645		
	4	0.1	0.296		



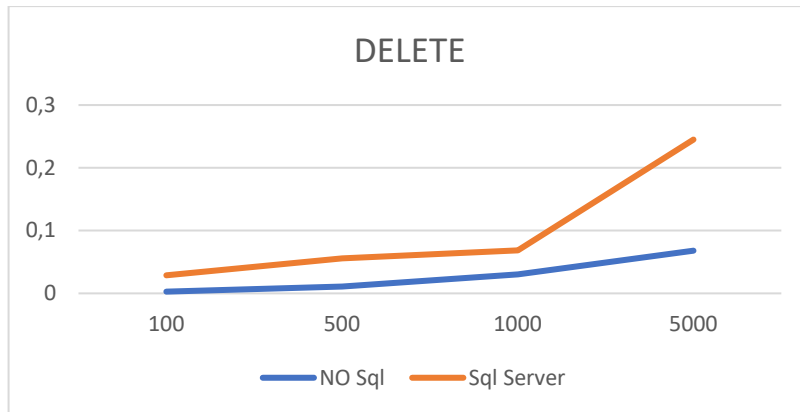
Gambar 8. Grafik Perbandingan Perintah Insert Pada MongoDB dan SQL Server

Hasil dari mengujian perintah Sql insert pada database Pemesanan Tiket Pesawat Online, dengan menggunakan 100 record, 500 record, 1.000 record, 5.000 record dapat dilihat di Tabel 1 dan Gambar 8.

Melakukan ujicoba ini untuk menguji waktu respon dalam memproses data dengan perintah delete dalam waktu yang bersamaan, di bawah ini adalah daftar ujicoba penelitian kami dalam format tabel dan grafik.

Tabel 2. Uji Coba Perintah Delete

Record	Uji coba	Waktu (s)		Rata-rata (s)	
		NoSQL	SQL Server	NoSQL	SQL Server
5,000.00	1	0.068	0.177	0.068	0.177
	2	0.068	0.177		
	3	0.068	0.177		
	4	0.068	0.177		
1,000.00	1	0.017	0.067	0.03	0.03825
	2	0.068	0.042		
	3	0.014	0.023		
	4	0.021	0.021		
500.00	1	0.017	0.001	0.01075	0.0045
	2	0.012	0.002		
	3	0.007	0.003		
	4	0.007	0.012		
100.00	1	0.001	0.026	0.00275	0.026
	2	0.003	0.028		
	3	0.004	0.027		
	4	0.003	0.023		



Gambar 9. Grafik Perbandingan Perintah Delete Pada MongoDB dan SQL Server

Hasil dari mengujian perintah Sql delete pada database Pemesanan Tiket Pesawat Online, dengan menggunakan 100 record, 500 record, 1.000 record, 5.000 record dapat dilihat di Tabel 2 dan Gambar 9.

Melakukan ujicoba ini untuk menguji kecepatan dalam memproses data dengan perintah select untuk 1 tabel dalam waktu yang bersamaan, di bawah ini adalah daftar ujicoba penelitian kami dalam format tabel dan grafik

Tabel 3. Uji Coba Perintah Select dengan 1 Tabel

Record	Uji coba	Waktu (s)		Rata-rata (s)	
		NoSQL	SQL Server	NoSQL	SQL Server
5,000.00	1	0.002	0.017	0.0005	0.0155
	2	0.002	0.015		
	3	0.004	0.014		
	4	0.002	0.016		
1,000.00	1	0.001	0.005	0.00025	0.0165
	2	0.002	0.041		
	3	0.001	0.017		
	4	0.005	0.003		
500.00	1	0.001	0.003	0.00025	0.04475
	2	0.001	0.003		
	3	0.004	0.17		
	4	0.001	0.003		
100.00	1	0.002	0.006	0.0005	0.0055
	2	0.001	0.005		
	3	0.001	0.007		
	4	0.001	0.004		

Hasil dari mengujian perintah Sql Select dengan 1 tabel pada database Pemesanan Tiket Pesawat Online, dengan menggunakan 100 record, 500 record, 1.000 record, 5.000 record dapat dilihat di Tabel 3 dan Gambar 10.

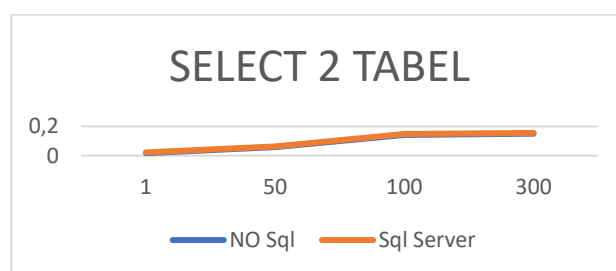


Gambar 10. Grafik Perbandingan Perintah Select 1 Tabel Pada MongoDB dan SQL Server

Melakukan ujicoba ini untuk menguji kecepatan dalam memproses data dengan perintah select untuk 2 tabel dalam waktu yang bersamaan, di bawah ini adalah daftar ujicoba penelitian kami dalam format tabel dan grafik.

Tabel 4. Uji Coba Perintah Select dengan 2 Tabel

Record	Uji coba	Waktu (s)		Rata-rata (s)	
		NoSQL	SQL Server	NoSQL	SQL Server
1.00	1	0.015	0.005	0.017	0.00375
	2	0.024	0.005		
	3	0.011	0.002		
	4	0.018	0.003		
50.00	1	0.235	0.006	0.05875	0.00575
	2	0.262	0.006		
	3	0.261	0.003		
	4	0.269	0.008		
100.00	1	0.569	0.005	0.14225	0.0045
	2	0.519	0.005		
	3	0.489	0.005		
	4	0.481	0.003		
300.00	1	0.602	0.007	0.1505	0.00525
	2	0.54	0.005		
	3	0.547	0.005		
	4	0.55	0.004		



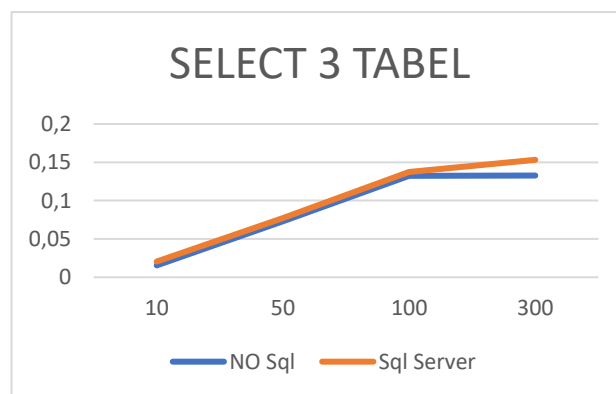
Gambar 11. Grafik Perbandingan Perintah Select 2 Tabel Pada MongoDB dan SQL Server

Hasil dari mengujian perintah Sql Select dengan 2 tabel pada database Pemesanan Tiket Pesawat Online, dengan menggunakan 100 record, 500 record, 1.000 record, 5.000 record dapat dilihat di Tabel 4 dan Gambar 11.

Melakukan ujicoba ini untuk menguji kecepatan dalam memproses data, dengan perintah select untuk 3 tabel dalam waktu yang bersamaan, dibawah ini adalah daftar ujicoba penelitian kami dalam format tabel dan grafik.

Tabel 5. Ujicoba Perintah Select dengan 3 Tabel

Record	Uji coba	Waktu (s)		Rata-rata (s)	
		NoSQL	SQL Server	NoSQL	SQL Server
10.00	1	0.062	0.01	0.0155	0.005
	2	0.041	0.002		
	3	0.071	0.005		
	4	0.074	0.003		
50.00	1	0.293	0.008	0.07325	0.004
	2	0.288	0.003		
	3	0.27	0.003		
	4	0.292	0.002		
100.00	1	0.53	0.009	0.1325	0.00475
	2	0.541	0.003		
	3	0.555	0.004		
	4	0.604	0.003		
300.00	1	0.531	0.067	0.13275	0.0205
	2	0.521	0.005		
	3	0.536	0.005		
	4	0.561	0.005		



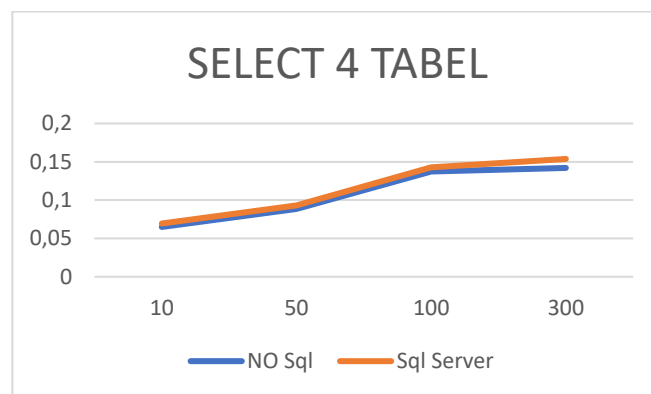
Gambar 12. Grafik Perbandingan Perintah Select 3 Tabel Pada MongoDB dan SQL Server

Hasil dari mengujian perintah Sql Select dengan 3 tabel pada database Pemesanan Tiket Pesawat Online, dengan menggunakan 100 record, 500 record, 1.000 record, 5.000 record dapat dilihat di Tabel 5 dan Gambar 12.

Melakukan ujicoba ini untuk menguji kecepatan dalam memproses data dengan perintah select untuk 4 tabel dalam waktu yang bersamaan, di bawah ini adalah daftar ujicoba penelitian kami dalam format tabel dan grafik.

Tabel 6. Uji Coba Perintah Select dengan 4 Tabel

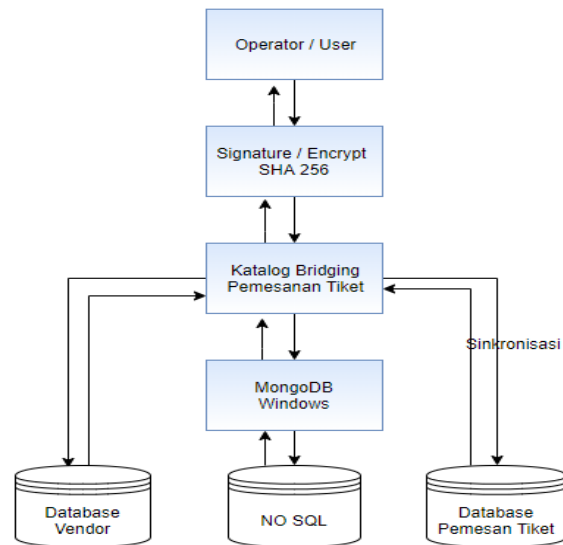
Record	Uji coba	Waktu (s)		Rata-rata (s)	
		NoSQL	SQL Server	NoSQL	SQL Server
10.00	1	0.072	0.011	0.065	0.0045
	2	0.072	0.002		
	3	0.059	0.003		
	4	0.057	0.002		
50.00	1	0.354	0.009	0.0885	0.00475
	2	0.278	0.003		
	3	0.307	0.003		
	4	0.306	0.004		
100.00	1	0.549	0.013	0.13725	0.0055
	2	0.707	0.004		
	3	0.578	0.003		
	4	0.579	0.002		
300.00	1	0.568	0.028	0.142	0.01175
	2	0.573	0.007		
	3	0.588	0.006		
	4	0.558	0.006		



Gambar 13. Grafik Perbandingan Perintah Select 4 Tabel Pada MongoDB dan SQL Server

Hasil dari mengujian perintah Sql Select dengan 4 tabel pada database Pemesanan Tiket Pesawat Online, dengan menggunakan 100 record, 500 record, 1.000 record, 5.000 record dapat dilihat di Tabel 6 dan Gambar 13. Dalam penggunaan 4 tabel yang saling berelasi pada database pemesanan tiket pesawat online, dari hasil ujicoba tersebut, dapat disimpulkan masih relevan dalam menggunakan database MongoDB.

Proses bridging sinkronisasi data antara database NoSql MongoDB dengan Database Pemesanan Tiket dan Database Vendor menggunakan SQL Server. sesuai yang tertera pada gambar 13, metode ini untuk mempercepat proses dalam pelaporan data, karena dari hasil ujicoba pada tabel 1 sampai tabel 6 bahwa proses insert lebih cepat MongoDB dan proses Select lebih cepat Sql Server, dengan kolaborasi antar database maka lebih efisien baik dalam proses insert maupun proses pelaporan.



Gambar 13. Alur Proses Bridging Pemesanan Tiket Pesawat Online

4. KESIMPULAN

Dapat ditarik kesimpulan bahwa perintah Insert database NoSQL memiliki kecepatan 4 kali lipat lebih cepat dari database SQL Server untuk dibawah 500 record, sedangkan diatas 500 record database NoSQL lebih lambat 5 kali lipat, perintah Delete database NoSQL memiliki kecepatannya 3 kali lipat lebih cepat dari database SQL Server, dan perintah Select 1 tabel database NoSQL lebih cepat 55 kali dari database SQL Server, sedangkan 2 tabel database NoSQL lebih lambat 18 kali dari database SQL Server, sedangkan 3 tabel database NoSQL lebih lambat 10 kali dari database SQL Server, sedangkan 4 tabel database NoSQL lebih lambat 16 kali dari database SQL Server.

DAFTAR PUSTAKA

- [1] R. Aghi, S. D. Mehta, R. Chauhan, S. Chaudhary, and N. Bohra, "A comprehensive comparison of SQL and MongoDB databases," *Int. J. Sci. Res. Publ.*, vol. 5, no. 2, pp. 228–230, 2015.
- [2] I. G. Winaya and A. Ashari, "Transformasi Skema Basis Data Relasional Menjadi Model Data Berorientasi Dokumen pada MongoDB," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 10, no. 1, p. 47, 2016.
- [3] R. Deari, X. Zenuni, J. Ajdari, F. Ismaili, and B. Raufi, "Analysis and comparison of document-based databases with relational databases: MongoDB vs MySQL," *2018 Int. Conf. Inf. Technol. InfoTech 2018 - Proc.*, no. September, pp. 20–21, 2018.
- [4] R. K. V G and K. S. Chethan, "Comparison and Data migration of Relational Database with NoSQL," vol. 3, no. 4, pp. 4–7, 2017.
- [5] Herrnansyah, Y. Ruldeviyani, and R. F. Aji, "Enhancing query performance of library information systems using NoSQL DBMS: Case study on library information systems of Universitas Indonesia," *2016 Int. Work. Big Data Inf. Secur. IWBIS 2016*, pp. 41–46, 2017.
- [6] G. Ongo and G. P. Kusuma, "Hybrid Database System of MySQL and MongoDB in Web Application Development," *Proc. 2018 Int. Conf. Inf. Manag. Technol. ICIMTech 2018*, no. September, pp. 256–260, 2018.
- [7] R. Gunawan, "Performance Evaluation of Query Response Time in The Document Stored NoSQL Database," *2019 16th Int. Conf. Qual. Res. Int. Symp. Electr. Comput. Eng.*, pp. 1–6.
- [8] E. Tang and Y. Fan, "Performance comparison between five NoSQL databases," *Proc. - 2016 7th Int. Conf. Cloud Comput. Big Data, CCBDD 2016*, pp. 105–109, 2017.
- [9] N. Kuderu and V. Kumari, "Relational Database to NoSQL Conversion by Schema Migration and Mapping," *Int. J. Comput. Eng. Res. Trends*, vol. 3, no. 9, p. 506, 2016.
- [10] G. Zhao, Q. Lin, L. Li, and Z. Li, "Schema conversion model of SQL database to NoSQL," *Proc. - 2014 9th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. 3PGCIC 2014*, pp. 355–362, 2014.

- [11] Kriestanto, Danny., Arnado, Alif Benden., 2017, *Implementasi Website Pencarian Kos Dengan Nosql*, Jurnal Informatika dan Komputer (JIKO) – Vol. 2, No. 2, September 2017
- [12] Li, Yan and Byeong-Seok Shin ,2015 *Location-Aware WBAN Data Monitoring System Based on NoSQL*, Department of Computer Information and Engineering, Inha University, Incheon, Korea, Springer Science+Business Media Singapore 2015
- [13] Yovita Tunardi; Rita Layona, 2014 , *Nosql Technology In Android Based Mobile Chat Application Using Mongodb*, Computer Science Department, School of Computer Science, Binus University Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480, ComTech Vol. 5 No. 2 Desember 2014: 553-565
- [14] Kusumawardhana, Pramudya Mahardika.,dkk, 2018 ,*Implementasi Penyimpanan Data Sensor Nirkabel dengan MongoDB pada Lingkungan IOT Menggunakan Protokol MQTT*, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 9, September 2018, hlm. 3391-3399
- [15] Mrs. Rupali M. Chopade , Mr. Nikhil S. Dhavase, 2017 ,*MongoDB, CouchBase: Performance Comparison for Image Dataset*, 2017 2nd International Conference for Convergence in Technology (I2CT), India