

# Perbandingan *Relational Database* dan *Non-Relational Database* dalam Pengembangan *Smart Tourism*

<http://dx.doi.org/10.28932/jutisi.v8i1.4353>

Riwayat Artikel

Received: 08 Januari 2022 | Final Revision: 21 Maret 2022 | Accepted: 22 Maret 2022

Mutmainnah A. <sup>✉</sup>#1, Musyrifah<sup>#2</sup>, Nuralamsyah Zulkarnaim<sup>#3</sup>

<sup>#</sup>Program Studi Teknik Informatika, Universitas Sulawesi Barat  
Baurung, Banggae Timur, Majene, Indonesia

<sup>1</sup>innah2095@gmail.com

<sup>2</sup>musyrifah@unsulbar.ac.id

<sup>3</sup>nuralamsah@unsulbar.ac.id

**Abstract** — The smart tourism application that is built cannot be separated from the role of the database. The more data stored in a database system, it will affect the speed of access of the information system that uses it. This study will test the performance of query response time between two types of databases with 92 data records of tourist attractions, 106 restaurants and 59 lodging data. Tests were carried out on Data Manipulation Language (DML), aggregate function testing, and logical operator testing. The result of this study is that Non-relational databases are proven to be superior in the DML testing process (insert, select, update, and delete) than Relational Databases. For the process of testing logical operators (AND, OR, and NOT) Non-relational databases are proven to be superior to Relational Databases. In the MIN aggregate function test the results are the same, while the MAX Relational Database aggregate function test is superior to the Non-relational database.

**Keywords** — MongoDB; MySQL; Non-Relational Database; Relational Database Smart Tourism.

## I. PENDAHULUAN

Bidang pariwisata saat ini menjadi salah satu bagian paling dominan di dunia bahkan di Indonesia, namun pengembangan pariwisata menghadapi banyak tantangan karena peningkatan teknologi informasi yang cepat. Melalui teknologi informasi, para wisatawan bisa mencari dan memberi informasi dengan mudah mengenai destinasi wisata yang akan dikunjungi. Hal ini merupakan tantangan bagi para pengembang wisata untuk memberikan informasi yang baik dan menarik terkait dengan destinasi wisata, sehingga para wisatawan tertarik untuk mengunjungi suatu daerah [1]. Provinsi Sulawesi Barat memiliki banyak destinasi wisata yang kurang diketahui oleh masyarakat luar, seperti wisata alam, kuliner, religi, budaya, dan lain-lain. Kurangnya informasi terhadap objek wisata, biaya dan waktu merupakan permasalahan yang menjadikan para wisatawan jarang berkunjung ke Sulawesi Barat.

*Smart tourism* hadir sebagai solusi dalam upaya untuk mengembangkan pariwisata daerah. Pariwisata berbasis *smart tourism* merupakan salah satu inovasi yang diharapkan dapat mempermudah para wisatawan dalam mengakses informasi terkait tempat wisata yang ada di Sulawesi Barat.

Aplikasi *smart tourism* yang dibangun tidak akan lepas dari peran *database*. Semakin banyak data tersimpan di dalam suatu sistem *database*, maka akan mempengaruhi kecepatan akses dari sistem informasi yang menggunakannya. *Relational Database* mulai digunakan pada tahun 1970-an sampai saat ini dan merupakan teknologi *database* yang paling populer digunakan, tetapi menghadapi permasalahan dalam hal performa pada pemrosesan data tidak terstruktur, seperti email, multimedia, dokumen, dan media sosial [2]. Berdasarkan masalah yang terjadi pada *Relational Database*, maka Carlo Strozzi memperkenalkan istilah *non-relational database* tahun 1998. Kemudian Eric Evans memperkenalkan kembali *database non-relasional* tahun 2009 [3].

Penelitian ini, sistem *database* yang digunakan pada *relational database* adalah MySQL, sedangkan untuk *non-relational database*, sistem *database* yang digunakan yaitu MongoDB, karena kedua sistem *database* tersebut bersifat *open source* dan merupakan salah satu sistem *database* yang paling sering digunakan [4].

MongoDB dan MySQL memiliki banyak keunggulan, maka dari itu penelitian ini akan dilakukan analisa Perbandingan *Relational Database* Dan *Non-Relational Database* Dalam Pengembangan *Smart Tourism* dengan menghitung performansi waktu respon, sehingga kedepannya dapat dijadikan sebagai rekomendasi *database* manakah yang paling cocok dan tepat sesuai dengan kebutuhan user.

Terdapat beberapa penelitian terkait dengan penelitian ini, diantaranya Analisis Performa Kecepatan MySQL dan NoSQL pada Sistem Operasi Windows dan Linux. Penelitian ini dilakukan analisa perbandingan kecepatan pada MySQL dan MongoDB untuk perintah *select*, *update* dan *delete* dengan jumlah data maksimum 1.000.000 baris, yang dilakukan pada sistem operasi Windows 10 dan Ubuntu Desktop 20.04 [5].

Membandingkan kedua jenis *database*, yaitu NoSQL dengan SQL dalam hal kinerja, skalabilitas, serta fleksibilitas. Setelah melakukan pengujian dan terbukti *database* mana yang paling unggul, maka akan digunakan pada aplikasi ERP Retail, agar aplikasi memiliki kecepatan dalam memproses data dan fleksibel untuk penyimpanan data. Setelah melakukan uji coba dapat disimpulkan bahwa *database* NoSQL lebih cepat daripada SQL dalam hal penyimpanan data [6].

Rangga Noviansyah Nuur Aziiz, dkk dalam penelitian [7]. Penelitian ini bertujuan untuk membandingkan jenis *database* mana yang lebih baik antara MySQL dan MongoDB untuk pengiriman data sensor. Hasil penelitian dapat didapat bahwa MongoDB jauh lebih unggul daripada MySQL dalam waktu eksekusi *query insert* data sensor LM35.

Penelitian ini membahas Aplikasi File Multimedia berbasis web menggunakan PHP, hasil dari penelitian ini didapat bahwa, MongoDB lebih unggul daripada MySQL pada operasi *create*, operasi *delete* dan operasi *read* [8].

Setiawan Budiman, dkk dalam penelitian [5]. untuk proses *select*, *insert* dan *update* dengan 6 kali tahapan pengujian didapat hasil bahwa MongoDB lebih cepat dibanding MySQL, sedangkan pada proses *delete*, MySQL lebih cepat daripada dengan MongoDB, karena MongoDB menggunakan program PHP sebagai aplikasi yang menjalankan proses *query*.

Apri Junaidi dalam penelitian [9], dalam studi perbandingan ini akan diujikan 226.232 *record* untuk masing-masing *database*, hasil dari perbandingan ini didapat bahwa MongoDB memiliki waktu respon 12.83624 detik, sedangkan MySQL dengan performansi waktu respon 70.584384 detik. Jadi dapat disimpulkan bahwa MongoDB lebih cepat dibandingkan MySQL.

Olivia Maria Inacio Tavares, dkk dalam penelitian [10]. Dapat dibuktikan bahwa antara MongoDB dan MySQL memiliki kecepatan waktu yang berbeda pada setiap model pengujian. Pada pengujian *query* DML, pengujian fungsi *agregat*, *import* dan *export* data serta fungsi operator penghubung MongoDB terbukti lebih unggul daripada MySQL, sedangkan dalam pemrosesan *query select* MySQL lebih unggul dengan jumlah selisih waktu respon 1,95s.

## II. METODE

### A. Relational Database

*Relational database* adalah koleksi data yang saling terkait dan terhubung antara satu tabel dengan tabel lainnya melalui *primary key*. *Relational database* berisi baris dan kolom yang dapat mengintegrasikan data ke satu tabel atau lebih atau biasa disebut dengan relasi. Relasi bermanfaat untuk mempermudah pengguna untuk menganalisis *database* [8].

### B. Non-Relational Database

*Non-Relational Database* memiliki kelebihan dapat memproses *big data* dengan kecepatan tinggi dan dapat memproses data terstruktur, semi terstruktur dan tidak terstruktur. Kekurangannya, yaitu untuk memproses operasi yang kompleks *Non-Relational Database* membutuhkan waktu lama dan *Non-Relational Database* juga tidak dapat memproses *grouping/aggregation*, *subqueries*, dan *joins* tabel [11].

### C. MongoDB

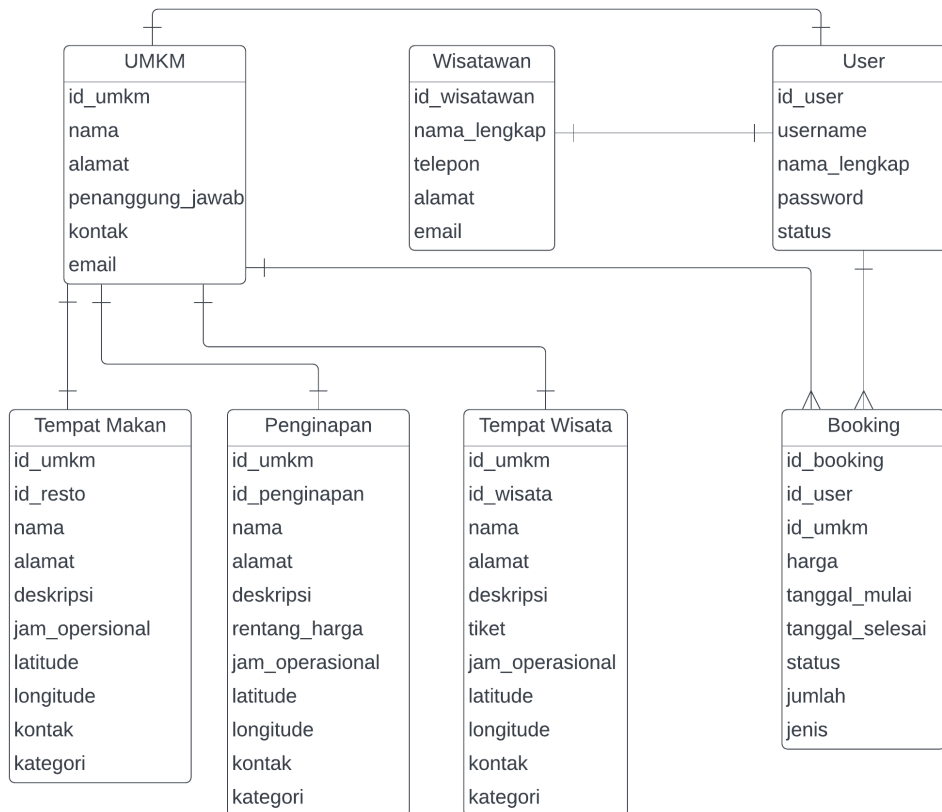
MongoDB adalah *database* berbasis dokumen yang menyediakan kecepatan pemrosesan data dan ketersediaan tinggi, serta skalabilitas yang mudah. MongoDB adalah sebuah *database* yang banyak digunakan untuk menangani data yang besar, MongoDB juga bersifat *open source*. MongoDB menyimpan dokumen dalam format BSON (bentuk binary dari JSON). BSON mendukung tipe data yang berbeda seperti *string*, *integer*, *float*, *boolean*, *date*, dan lain-lain [8].

### D. MySQL

MySQL merupakan salah satu implementasi *relational database* yang *open source* yang bersifat *multi-user* dan *multi-threaded*. Populernya Bahasa pemrograman PHP dan web server Apache merupakan suatu pengaruh yang dapat meningkatkan popularitas MySQL. Drupal, WordPress, Wikipedia, Mambo, PHP-Nuke adalah sebagian dari banyaknya aplikasi berbasis web yang dibuat dengan menggunakan kolaborasi kedua perangkat lunak tersebut [12].

### E. Smart Tourism

Smart Tourism merupakan sebuah platform wisata yang mengintegrasikan ekosistem layanan pariwisata dengan memberikan layanan trip planner dimana calon wisatawan dapat merencanakan perjalanan dengan mudah sesuai dengan budget yang dimiliki, tanpa harus memikirkan transportasi, akomodasi, tempat wisata hingga tempat makan yang populer di daerahnya. Untuk desain *database smart tourism* dapat dilihat pada gambar 1.



Gambar 1. Desain *database*

### F. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian komparatif. Komparatif merupakan penelitian yang digunakan untuk membandingkan dua variabel atau lebih, untuk membuktikan apakah terdapat perbandingan atau tidak dari objek yang sedang diteliti [13].

### G. Populasi dan Sampel

#### 1. Populasi

Populasi merupakan semua objek yang digunakan pada suatu penelitian. Jika ingin meneliti semua objek yang ada pada suatu wilayah penelitian, maka penelitian tersebut adalah penelitian populasi (Elvira Lestari, 2018). Populasi pada penelitian ini dapat dilihat pada tabel 1.

TABEL 1  
DATA OBJEK WISATA

No	Objek Wisata	Kabupaten	Jumlah Wisata	Total
1	Tempat Wisata	Mamuju	18	92
		Majene	31	
		Polewali	43	
2	Rumah Makan	Mamuju	9	106
		Majene	46	
		Polewali	51	
3	Penginapan	Mamuju	38	59
		Majene	10	
		Polewali	11	
Total keseluruhan				257

## 2. Sampel

Total sampling merupakan teknik pengambilan sampel yang digunakan dalam penelitian ini. Total Sampling merupakan suatu teknik pengambilan sampel dimana jumlah sampel sama dengan jumlah populasi [13].

## H. Variabel Penelitian

1. Variabel bebas atau variabel independen yaitu variabel yang mempengaruhi atau yang menjadi sebab perubahan atau timbulnya variabel dependen (terikat). Variabel independen dalam penelitian ini adalah membandingkan performa waktu respon.
2. Variabel terikat atau variabel dependen merupakan variabel yang dipengaruhi atau yang menjadi akibat, karena adanya variabel bebas. Variabel dependen dalam penelitian ini adalah *relational database* dan *non-relational database*.

## I. Uji Coba

Uji coba pada penelitian ini dilakukan di *localhost* menggunakan *hardware* dan *software* yang sama, dengan spesifikasi sebagai berikut.

1. Spesifikasi *hardware*
  - Memory: 4 GB RAM
  - Processor: Intel(R) Core (TM) i3-1005G1
  - Harddisk: 256 GB SSD
2. Spesifikasi *software*
  - Microsoft Windows 10
  - MySQL Xampp versi PHP 8.0.10
  - MongoDB versi 5.0.3
  - Robo 3T versi 1.4.4

Tahapan Uji coba pengujian dalam penelitian ini terdapat tiga tahapan, yaitu:

1. Pengujian terhadap *query* DML *insert*, *update*, *delete*, dan *select*.
2. Pengujian operator logika (*AND*, *OR*, dan *NOT*).
3. Pengujian fungsi *agregat* (*MIN* dan *MAX*).

## III. HASIL DAN PEMBAHASAN

### A. Menguji *query* DML (*Data Manipulation Language*)

Pengujian ini akan dioperasikan sejumlah *query* DML, yaitu *delete*, *update*, *select* dan *insert* pada *database* MySQL dan MongoDB dengan menggunakan data yang sama, agar mendapatkan hasil yang sebanding dengan pengujian yang dilakukan, pengujian dilakukan sebanyak 3 kali secara bertahap, dari uji coba tempat wisata, rumah makan, dan penginapan.

1. Membuat *Query* untuk testing  
*Query* testing *insert*, *select*, *update*, dan *delete* data dapat dilihat pada tabel 2, tabel 3, tabel 4, dan tabel 5.

TABEL 2  
QUERY INSERT

Database	Query
MySQL	INSERT INTO `penginapan_tb`(`id_penginapan`,`nama`,`alamat`,`kontak`,`rentang_harga`,`jam_operasional`,`longitude`,`latitude`,`kategori`,`kabupaten`) VALUES (NULL, 'Penginapan sederhana', 'Jl. Poros Majene - Mamuju, Malunda, Kec. Malunda, Kabupaten Majene, Sulawesi Barat 91453', '0853 9442 2229', '10000-150000', '08.00-20.00', '118.83460677264678', '-2.8965880266049884', 'Penginapan', 'Majene')
MongoDB	db.penginapan.insert({{ nama: "Penginapan sederhana", alamat: "Jl. Poros Majene - Mamuju, Malunda, Kec. Malunda, Kabupaten Majene, Sulawesi Barat 91453", kontak: "085394422229", rentang_harga: "10000-150000", jam_operasional: "08.00-20.00", longitude: "118.83460677264678", latitude: "-2.8965880266049884", kategori: "Penginapan", kabupaten: "Majene"}})

TABEL 3  
QUERY SELECT

Database	Query
MySQL	SELECT * FROM `tempat_wisata_tb`
MongoDB	db.getCollection('tempat_wisata').find({})

TABEL 4  
QUERY UPDATE

Database	Query
MySQL	UPDATE tempat_makan SET kabupaten = "Mamuju" WHERE id_resto > 836
MongoDB	db.tempat_makan.updateMany({"kabupaten": "Majene"}, {\$set: {"kabupaten": "Mamuju"}})

TABEL 5  
QUERY DELETE

Database	Query
MySQL	DELETE FROM `tempat_wisata_tb`
MongoDB	db.tempat_wisata.deleteMany({})

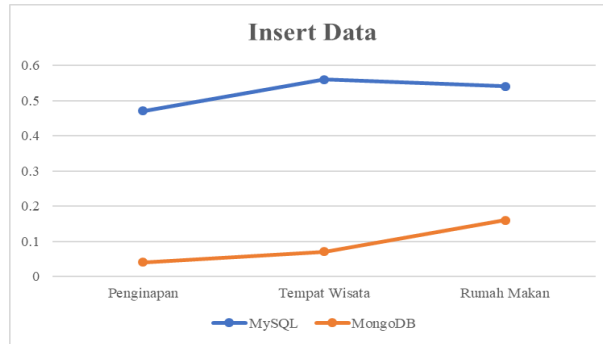
## 2. Hasil perbandingan

Setelah melakukan pengujian maka akan dibuatkan tabel dan grafik terhadap perbandingan *query* DML pada *database* MySQL dan MongoDB dari data-data yang terkumpul. Hasil perbandingan dapat dilihat pada tabel 6.

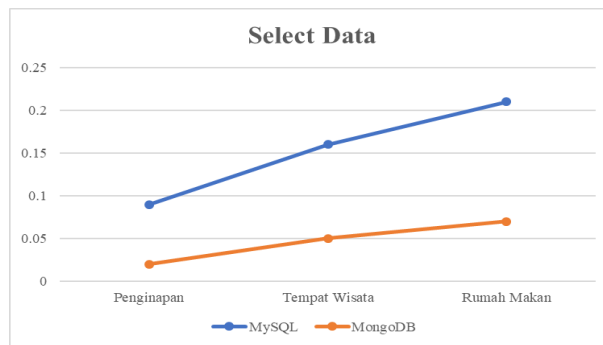
TABEL 6  
HASIL PERBANDINGAN QUERY DML

Jumlah data	Insert		Select		Update		Delete	
	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB
59	0.47	0.04	0.09	0.02	0.19	0.07	0.19	0.08
92	0.56	0.07	0.16	0.05	0.36	0.08	0.47	0.11
106	0.54	0.16	0.21	0.07	0.37	0.13	0.26	0.13
<b>Rata-rata</b>	<b>0.52</b>	<b>0.09</b>	<b>0.15</b>	<b>0.05</b>	<b>0.31</b>	<b>0.09</b>	<b>0.31</b>	<b>0.11</b>

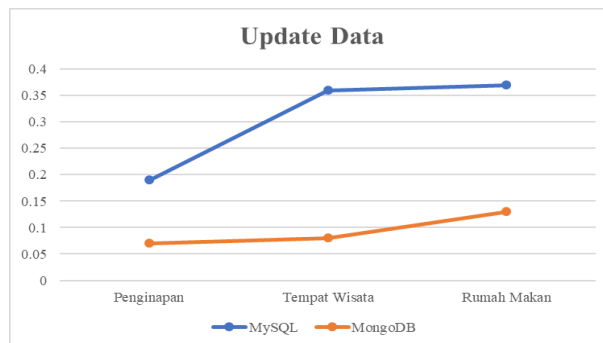
Kesimpulan yang dapat diambil pada tabel diatas adalah pada performansi *query response time* dalam model pengujian *insert* data MongoDB lebih unggul daripada MySQL dengan selisih perbandingan waktu rata-rata 0.43 detik. Pada model pengujian *select* data MongoDB lebih baik dengan selisih perbandingan waktu rata-rata 0.10 detik. Pengujian *update* data MongoDB lebih cepat daripada MySQL dengan selisih perbandingan waktu rata-rata 0.21 detik. Pengujian *delete* data MongoDB lebih unggul dengan selisih perbandingan waktu rata-rata 0.20 detik. Grafik perbandingan waktu *insert*, *select*, *update*, *delete* dapat dilihat pada Gambar 2, 3, 4, dan 5.



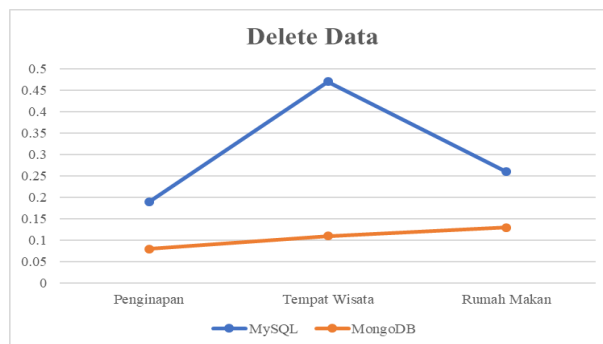
Gambar 2. Grafik perbandingan waktu *insert*



Gambar 3. Grafik perbandingan waktu *select*



Gambar 4. Grafik perbandingan waktu *update*



Gambar 5. Grafik perbandingan waktu *delete*

### B. Menguji Operator Logika

Uji coba yang akan dioperasikan pada tahapan ini adalah *query* operator logika *AND*, *OR*, dan *NOT* pada *MySQL* dan *MongoDB* dengan menggunakan data yang sama, agar mendapatkan hasil yang sebanding dengan pengujian yang dilakukan, pengujian dilakukan sebanyak 3 kali secara bertahap, dari uji coba tempat wisata, rumah makan, dan penginapan.

1. Membuat *query* untuk testing

*Query* testing operator logika *AND*, *OR*, dan *NOT* data dapat dilihat pada tabel 7, tabel 8, dan tabel 9.

TABEL 7  
QUERY LOGIKA AND

Database	Query
MySQL	SELECT * FROM tempat_wisata_tb WHERE kategori = "Bahari" AND kabupaten = "Majene";
MongoDB	db.getCollection("tempat_wisata").find( {"\$and": [ {"kategori": "Bahari"}, {"kabupaten": "Majene"} ]})

TABEL 8  
QUERY LOGIKA OR

Database	Query
MySQL	SELECT * FROM tempat_makan WHERE harga = "10000-50000" OR kabupaten = "Mamuju";
MongoDB	db.getCollection("tempat_makan").find({"\$or": [{"rentang_harga": "10000-50000"}, {"kabupaten": "Mamuju"}]})

TABEL 9  
QUERY LOGIKA NOT

Database	Query
MySQL	SELECT * FROM `penginapan_tb` WHERE NOT jam_operasional = "24 jam"
MongoDB	db.penginapan.find({jam_operasional: {\$not: {\$in: ["24 jam"]}}});

2. Hasil perbandingan

Setelah melakukan pengujian maka akan dibuatkan tabel dan grafik terhadap perbandingan operator logika pada *database* *MySQL* dan *MongoDB* dari data-data yang terkumpul. Hasil perbandingan dapat dilihat pada tabel 10.

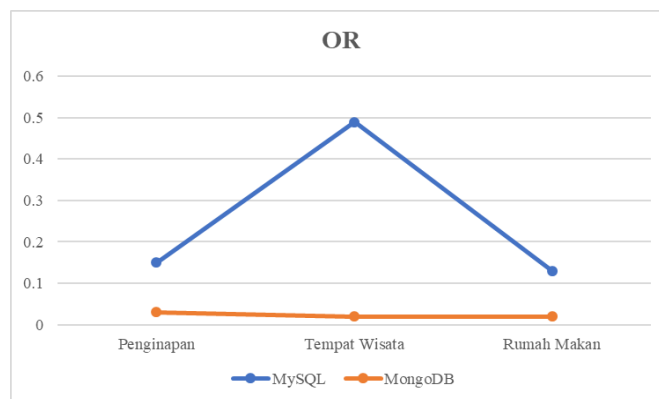
TABEL 10  
HASIL PERBANDINGAN OPERATOR LOGIKA

Jumlah data	AND		OR		NOT	
	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB
59	0.14	0.04	0.15	0.03	0.14	0.02
92	0.27	0.03	0.49	0.02	0.11	0.04
106	0.19	0.04	0.13	0.02	0.12	0.02
<b>Rata-rata</b>	<b>0.2</b>	<b>0.04</b>	<b>0.26</b>	<b>0.02</b>	<b>0.12</b>	<b>0.03</b>

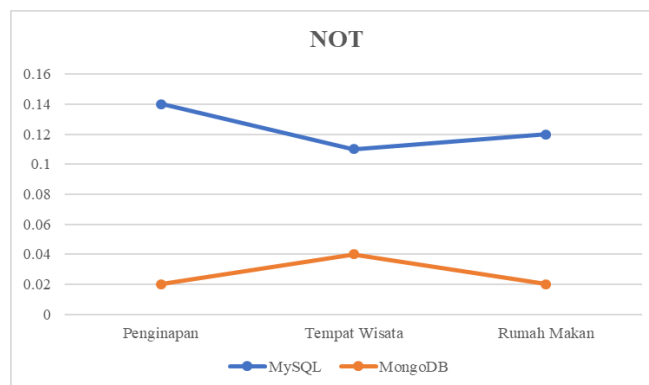
Kesimpulan yang dapat diambil pada tabel diatas adalah performansi *query respond time* pengujian operator logika *AND*, *MongoDB* lebih unggul daripada *MySQL* dengan rata-rata selisih waktu 0.16 detik. Pada performansi *query respond time* pengujian operator logika *OR*, *MongoDB* lebih unggul daripada *MySQL* dengan rata-rata selisih waktu 0.23 detik. Dan untuk performansi *query respond time* pengujian operator logika *NOT*, *MongoDB* lebih unggul daripada *MySQL* dengan rata-rata selisih waktu 0.10 detik. Grafik perbandingan waktu logika *AND*, *OR*, *NOT* dapat dilihat pada Gambar 6, 7, dan 8.



Gambar 6. Grafik perbandingan waktu logika AND



Gambar 7. Grafik perbandingan waktu logika OR



Gambar 8. Grafik perbandingan waktu logika NOT

### C. Menguji Fungsi Agregat

Uji coba yang akan dioperasikan pada tahapan ini adalah query fungsi *agregat* (MIN dan MAX) pada MySQL dan MongoDB dengan menggunakan data yang sama, agar mendapatkan hasil yang sebanding dengan pengujian yang dilakukan, pengujian dilakukan sebanyak 3 kali secara bertahap, dari uji coba tempat wisata, rumah makan, dan penginapan.

#### 1. Membuat Query untuk testing

Query fungsi *agregat* MIN dan MAX data dapat dilihat pada tabel 11 dan tabel 12.



TABEL 11  
QUERY FUNGSI AGREGAT MIN

Database	Query
MySQL	SELECT MIN(tiket) FROM tempat_wisata_tb
MongoDB	db.tempat_wisata.aggregate([{\$group: {_id: "\$nama", tiket: {\$min: "\$tiket"}}}])

TABEL 12  
QUERY FUNGSI AGREGAT MAX

Database	Query
MySQL	SELECT MAX(tiket) FROM tempat_wisata_tb
MongoDB	db.tempat_wisata.aggregate([{\$group: {_id: "\$nama", tiket: {\$max: "\$tiket"}}}])

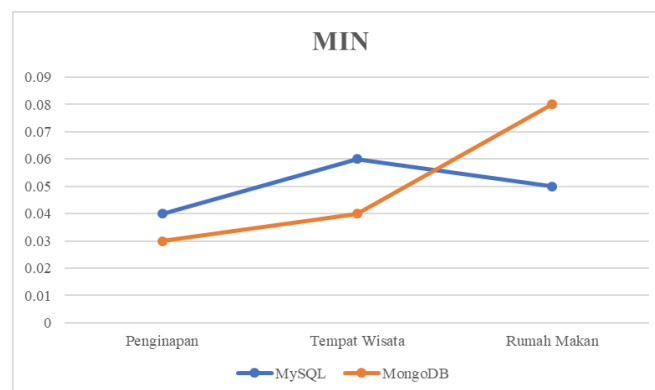
## 2. Hasil perbandingan

Setelah melakukan pengujian maka akan dibuatkan tabel dan grafik terhadap perbandingan fungsi *agregat* pada *database* MySQL dan MongoDB dari data-data yang terkumpul. Hasil perbandingan dapat dilihat pada tabel 13.

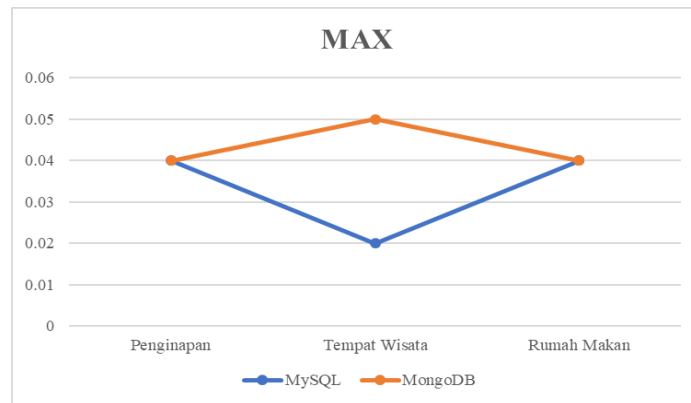
TABEL 13  
HASIL PERBANDINGAN FUNGSI AGREGAT

Jumlah data	MIN		MAX	
	MySQL	MongoDB	MySQL	MongoDB
59	0.04	0.03	0.04	0.04
92	0.06	0.04	0.02	0.05
106	0.05	0.08	0.04	0.04
<b>Rata-rata</b>	<b>0.05</b>	<b>0.05</b>	<b>0.03</b>	<b>0.04</b>

Kesimpulan yang dapat diambil pada tabel diatas adalah performansi *query respond time* pengujian fungsi *agregat MIN* antara MongoDB dengan MySQL hasilnya sama dengan rata-rata selisih waktu 0 detik. Sedangkan untuk pengujian fungsi *agregat MAX*, MySQL lebih unggul daripada MongoDB dengan rata-rata selisih waktu 0.01 detik. Grafik perbandingan waktu respon *query* MIN dan MAX dapat dilihat pada Gambar 9 dan Gambar 10.



Gambar 9. Grafik perbandingan waktu respon *query* MIN



Gambar 10. Grafik perbandingan waktu respon query MAX

#### D. Menguji Persyaratan Analisis

##### 1. Uji Normalitas

Uji Normalitas bertujuan untuk menguji apakah data variabel dependen dan independen mempunyai distribusi normal atau tidak. Data yang baik adalah memiliki distribusi data normal atau mendekati normal. Untuk menguji normalitas, dapat menganalisis dengan menggunakan metode *One Sample Kolmogorov Smirnov Test*. Dasar keputusan adalah jika nilai probabilitas t-statistik > Level of Significant = 0.05, maka model regresi memenuhi asumsi normalitas.

TABEL 14  
HASIL UJI NORMALITAS

Variabel	t-statistik	Sig	Keterangan
Query DML MySQL	0.139	0.2	Normal
Query DML MongoDB	0.167	0.2	Normal
Operator Logika MySQL	0.306	0.015	Normal
Operator Logika MongoDB	0.275	0.048	Normal
Fungsi Agregat MySQL	0.283	0.183	Normal
Fungsi Agregat MongoDB	0.315	0.064	Normal

Berdasarkan hasil uji normalitas yang ditunjukkan pada Tabel 14 dengan *One Sample Kolmogorov Smirnov Test* diatas terlihat bahwa nilai probabilitas t-statistik > Level of Significant = 0,05, maka data memenuhi asumsi normalitas. Dengan demikian, maka variabel dependen dan variabel independen mempunyai distribusi normal.

##### 2. Uji Homogenitas

Uji homogenitas digunakan untuk mengetahui apakah data yang diperoleh dari kedua kelompok memiliki varian yang homogen atau tidak.

TABEL 15  
HASIL UJI HOMOGENITAS

Variabel	Sig	Keterangan
Query DML	0.623	Homogen
Operator Logika	0.517	Homogen
Fungsi Agregat	0.585	Homogen

Berdasarkan hasil uji homogenitas yang ditunjukkan pada Tabel 15 diperoleh jika  $\text{Sig} > \text{Level of significant} = 0,05$ , maka data memenuhi asumsi homogenitas. Dengan demikian, maka populasi yang sedang diteliti mempunyai kesamaan.

#### E. Uji Hipotesis

Hipotesis yang digunakan adalah:

$H_0$  : tidak ada perbedaan yang signifikan antara kecepatan MySQL dengan MongoDB.

$H_1$  : ada perbedaan yang signifikan antara antara kecepatan MySQL dengan MongoDB.

Berdasarkan probabilitas:

$H_0$  diterima jika signifikan  $> 0,05$

$H_0$  ditolak jika signifikan  $< 0,05$

TABEL 16  
HASIL UJI T-TEST

Variabel	Sig.	Level of Significant
Query DML	0.001	0.05
Operator Logika	0.000	0.05
Fungsi Agregat	0.17	0.05

Berdasarkan hasil uji t-test yang ditunjukkan pada Tabel 16 diperoleh signifikansi 0,001 pada query DML, 0,000 pada Operator logika dan 0.17 pada fungsi agregat, hasil uji tersebut kurang dari taraf signifikan = 0,05, maka  $H_0$  ditolak. Artinya ada perbedaan yang signifikan antara kecepatan MySQL dengan MongoDB.

#### IV. SIMPULAN

Berdasarkan berbagai macam pengujian yang telah dilakukan dapat disimpulkan bahwa *Non-relational database* terbukti lebih cepat dalam proses pengujian DML (*insert, select, update, dan delete*) daripada *Relational Database*. Untuk proses pengujian operator logika (*AND, OR, dan NOT*) *Non-relational database* terbukti lebih unggul daripada *Relational Database*. Pada pengujian fungsi agregat *MIN* hasilnya sama/seri. Sedangkan pada pengujian fungsi agregat *MAX*, *Relational Database* lebih unggul daripada *Non-relational database*. Setelah melakukan penelitian, ada beberapa hal yang perlu dikembangkan untuk penelitian kedepannya, seperti penelitian selanjutnya dapat membandingkan jenis *database* yang lain, dapat merancang *database* sendiri, data yang digunakan bisa lebih banyak lagi tidak hanya tiga kabupaten saja, dan pengujian tidak hanya dilakukan di *localhost* saja, akan tetapi juga dilakukan melalui internet.

#### DAFTAR PUSTAKA

- [1] F. Femenia-Serra and B. Neuhofer, "Smart tourism experiences: Conceptualisation, key dimensions and research agenda," *Investig. Reg.*, vol. 2018, no. 42, pp. 129–150, 2018.
- [2] T. Patel and T. Eltaieb, "Relational Database vs NoSQL," *J. Multidiscip. Eng. Sci. Technol.*, vol. 2, no. 4, pp. 3159–40, 2015, [Online]. Available: <http://www.jmest.org/wp-content/uploads/JMESTN42350605.pdf>.
- [3] S. George, "NoSQL - NOTONLY SQL," *Int. J. Enterp. Comput. Bus. Syst.*, vol. 2, no. 2, 2013, [Online]. Available: <http://www.ijecbs.com/July2013/3.pdf>.
- [4] DB-Engines, "Knowledge Base of Relational and NoSQL Database Management Systems," 2021. <https://db-engines.com/en/ranking>.
- [5] S. Budiman, F. Fadhila, and V. A. Saputro, "Perbandingan Performa SQL dan NoSQL Dengan PHP," vol. 6, no. 1, pp. 38–42, 2021.
- [6] F. A. Bhaswara, R. Sarno, and D. Sunaryono, "Perbandingan Kemampuan Database NoSQL dan SQL dalam Kasus ERP Retail," *J. Tek. ITS*, vol. 6, no. 2, pp. 1–5, 2017, doi: 10.12962/j23373539.v6i2.24031.
- [7] R. N. N. Aziiz, M. H. H. Ichsan, and I. Arwani, "Implementasi Pengiriman Data Sensor LM35 untuk Perbandingan Waktu Insert pada Basis Data MySQL dan MongoDB," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 1, no. 6, pp. 466–475, 2017.
- [8] M. Silalahi, "Perbandingan Performansi Database Mongoddb Dan Mysql Dalam Aplikasi File Multimedia Berbasis Web," *Comput. Based Inf. Syst. J.*, vol. 6, no. 1, p. 63, 2018, doi: 10.33884/cbis.v6i1.574.
- [9] A. Junaidi, "Studi Perbandingan Performansi Antar Mongoddb Dan Mysql Menggunakan Php Dalam Lingkungan Big Data," *Pros. Annu. Res. Semin. 2016*, vol. 2, no. 1, pp. 460–465, 2016.
- [10] O. Maria, I. Tavares, S. M. Rangkoly, S. S. Desi Bawan, E. Utami, and M. S. Mustafa, "Analisis Perbandingan Performansi Waktu Respons Kueri Antara MySQL PHP 7.2.27 DAN NoSQL MongoDB," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 303–313, 2020, [Online]. Available: <http://oliviakeloebufu.com>.
- [11] W. N. Suliyanti, "Studi Literatur Basis Data SQL dan NoSQL," *Kilat*, vol. 8, no. 1, pp. 48–51, 2019, doi: 10.33322/kilat.v8i1.460.
- [12] Anur Rofiq Mulyanto, *Rekayasa Perangkat Lunak Jilid 2*. Direktorat Pembinaan Sekolah Menengah Kejuruan, 2008.
- [13] Elvira Lestari, "Studi Komparatif Tingkat Kesantunan Berbahasa dalam Proses Pembelajaran Bahasa Indonesia Antara Siswa Kelas X dengan Kelas XI di M.A Aisyiyah Sungguminasa," *J. Phys. Ther. Sci.*, vol. 9, no. 1, pp. 1–11, 2018, [Online]. Available: [https://digilibadmin.unismuh.ac.id/upload/4619-Full\\_Text.pdf](https://digilibadmin.unismuh.ac.id/upload/4619-Full_Text.pdf).