



## ***Comparison of Search Response Time Using Text Indexing on MongoDB and ArangoDB Web-based***

### ***Perbandingan Response Time Pencarian Menggunakan Text Indexing pada MongoDB dan ArangoDB Berbasis Web***

**Tessalonica Putry Avrylya<sup>1\*</sup>, Yeremia Alfa Susetyo<sup>2</sup>**

<sup>1,2</sup>Program Studi Teknik Informatika, Fakultas Teknologi Informasi,  
Universitas Kristen Satya Wacana, Indonesia

E-Mail: <sup>1</sup>672020167@student.uksw.edu, <sup>2</sup>yeremia.alfa@uksw.edu

*Received Jan 30th 2024; Revised Mar 15th 2024; Accepted Apr 25th 2024*  
*Corresponding Author: Tessalonica Putry Avrylya*

#### **Abstract**

*The amount of data in databases is increasing every day as the information technology industry develops. Relational databases (SQL) have evolved since the introduction of NoSQL databases such as MongoDB and ArangoDB. These databases have features that can use text indexes for text search and store data in JSON/BSON format. Data volume, structural flexibility, and search speed are common problems in data processing. As a result, data management becomes ineffective which leads to a decrease in database performance. The main focus of this research is to conduct a comparison of the response time to search using index text on the MongoDB and ArangoDB web-based databases to speed up searches so as to determine the comparison of query performance between databases. Based on the findings from 1000, 10000, 100000, and 1000000 data records, it is concluded that MongoDB text indexing is faster than ArangoDB. In the test, the results obtained on one keyword the difference in execution time speed is 170.75ms, for two keywords it is 205.5ms, and for three keywords it is 267.75ms.*

*Keyword: Data Search, MongoDB, ArangoDB, Python, Text Indexing*

#### **Abstrak**

Jumlah data dalam basis data meningkat setiap hari seiring dengan berkembangnya industri teknologi informasi. Basis data relasional (SQL) telah berevolusi sejak diperkenalkannya basis data NoSQL seperti MongoDB dan ArangoDB. Basis data tersebut memiliki fitur yang dapat menggunakan indeks teks untuk pencarian teks dan menyimpan data dalam format JSON/BSON. Volume data, fleksibilitas struktural, dan kecepatan pencarian adalah masalah umum dalam pemrosesan data. Akibatnya, manajemen data menjadi tidak efektif yang menyebabkan penurunan kinerja basis data. Fokus utama dari penelitian ini adalah untuk melakukan perbandingan respon waktu terhadap pencarian menggunakan teks indeks pada basis data MongoDB dan ArangoDB yang berbasis website guna mempercepat pencarian sehingga mengetahui perbandingan performa kueri antar basis data. Berdasarkan temuan dari 1000, 10000, 100000, dan 1000000 rekaman data didapat kesimpulan bahwa pengindeksan teks MongoDB lebih cepat dibandingkan dengan ArangoDB. Pada pengujian didapatkan hasil pada satu kata kunci selisih kecepatan waktu eksekusi sebesar 170,75ms, untuk dua kata kunci sebesar 205,5ms, dan pada tiga kata kunci sebesar 267,75ms.

Kata Kunci: ArangoDB, MongoDB, Pencarian Data, Python, Teks Indeks

#### **1. PENDAHULUAN**

Seiring dengan pertumbuhan industri teknologi informasi, telah terjadi peningkatan permintaan untuk pengaksesan basis data yang mampu mengelola data besar dengan efisien dan tetap menjaga kinerja tinggi selama penggunaannya [1]. Basis data adalah kumpulan data yang dapat dengan mudah diakses, dikelola, dan diperbaharui [2]. Jenis basis data yang paling banyak digunakan adalah basis data relasional (SQL) dan basis data non relasional (NoSQL) [3]. Saat ini, basis data SQL masih banyak digunakan beberapa pihak, tetapi secara bertahap akan digantikan oleh basis data NoSQL yang akan menjadi mayoritas dari waktu ke waktu [18]. Ketika menggunakan basis data SQL, salah satu masalah yang umum terjadi adalah menyimpan data dalam jumlah yang sangat besar dapat menyulitkan dalam pencarian data [4]. Semakin banyak data yang ada,

semakin lama waktu yang dibutuhkan untuk menemukannya, oleh karena itu basis data NoSQL akan menjadi lebih populer karena lebih efisien dalam mengelola kumpulan data yang sangat besar daripada basis data SQL [5].

Berbeda dengan basis data SQL, basis data NoSQL bekerja dengan data semi-terstruktur yang menyediakan fleksibilitas lebih besar dalam struktur data sedangkan pada basis data SQL harus mengikuti skema yang sudah ada dan dalam memproses data struktur tabel harus dengan memperhatikan baris-kolom [8]. Pada basis data NoSQL sendiri menggunakan metode baru yang dikategorikan ke dalam empat metode yaitu key-value stores, column-oriented, document-oriented, dan graph-oriented [6]. Kategori document-oriented paling populer digunakan karena menggunakan format dokumen semi-terstruktur Binary JavaScript Object Notation (BSON) [7]. Model ini memberikan fleksibilitas untuk menampung beberapa elemen dalam satu dokumen dan dapat diberi indeks untuk meningkatkan kecepatan dalam proses pencarian dan pada kategori document-oriented setiap dokumen juga memiliki pasangan key-value [6]. Contoh basis data NoSQL yang berbasis dokumen adalah MongoDB dan ArangoDB. MongoDB dan ArangoDB memiliki fungsi indeks teks untuk pencarian data. Indeks teks basis data digunakan untuk menghitung kesamaan kata kunci, yang meningkatkan efisiensi dan akurasi pencarian [9]. Indeks teks MongoDB dan ArangoDB menyediakan arsitektur yang fleksibel yang tidak membutuhkan struktur skema yang ketat untuk mengindeks data teks, berbeda dengan basis data SQL, yang membutuhkan kolom dengan tipe data tertentu untuk diindeks [9]. Perancangan situs web untuk perbandingan basis data akan lebih memudahkan pengguna dalam menemukan informasi yang lengkap dan terperinci tentang performa basis data tanpa harus mengakses langsung bagian server sehingga dapat membantu membuat keputusan yang lebih baik dan lebih terinformasi. Sehingga melakukan perbandingan respon time pencarian menggunakan teks indeks pada MongoDB dan ArangoDB berbasis website dapat menjadi penelitian yang menarik.

Berdasarkan penelitian terdahulu, terdapat beberapa penelitian yang telah menguji efektivitas penggunaan text indexing dalam meningkatkan respons time pencarian data pada basis data. Hasil penelitian sebelumnya menunjukkan bahwa implementasi text indexing pada MongoDB dapat mengurangi waktu pencarian data, sebagaimana dibuktikan dalam studi tentang penerapan sistem indeks teks pada pencarian dokumen [13]. Demikian pula, penelitian mengenai full text search juga menunjukkan manfaat penggunaan text indexing dalam meningkatkan efisiensi pencarian data [14]. Dengan menghubungkan kajian pustaka di atas, perlu diadakannya penelitian lanjut mengenai perbandingan teks indeks pada basis data NoSQL, sehingga penelitian ini akan menggali pemahaman yang lebih mendalam tentang bagaimana text indexing dapat mempengaruhi waktu respon pencarian data pada kedua basis data NoSQL dalam konteks penerapan pada aplikasi web. Penelitian ini akan menggunakan bahasa pemrograman python dan framework flask, menggunakan bahasa python dikarenakan hingga November 2023 python merupakan bahasa pemrograman terpopuler [15]. Untuk pemilihan framework flask dikarenakan flask sebagai sebuah micro framework karena tidak memaksa penggunaannya untuk mengadopsi alat atau pustaka tertentu.

Berdasarkan latar belakang masalah tersebut, penelitian ini bertujuan untuk melakukan perbandingan response time pencarian menggunakan text indexing pada basis data MongoDB dan ArangoDB berbasis web.

## 2. BAHAN DAN METODOLOGI PENELITIAN

Berlandaskan pada pendahuluan di atas, penelitian ini menggunakan metode eksperimental. Dengan didahului melakukan tahapan persiapan, eksperimen dan hasil analisis. Jika hasil dari analisis perbandingan dan data-data yang ada didalamnya akan didapatkan hasil dan kesimpulan. Tahapan penelitian ini dapat dilihat pada Gambar 1.

### 2.1. Tahap Persiapan

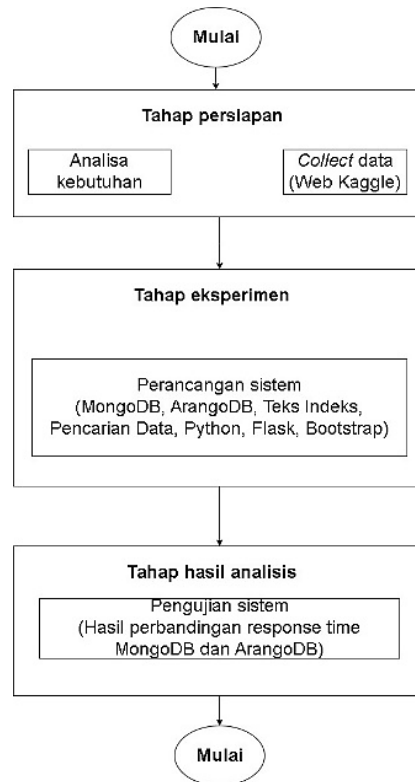
Pada tahap persiapan, penulis akan melakukan analisis kebutuhan supaya mengoptimalkan kinerja sistem dan mencegah *over-provisioning* dan *under-provisioning*. Pada tahap ini penulis mengumpulkan data yang didapat dari situs web Kaggle dengan menggunakan 1000000 rekaman data dari perusahaan dan organisasi di Amerika pada tahun 2021 terkait peringkat dan evaluasi.

### 2.2. Tahap Eksperimen

Pada tahap eksperimen, penulis akan melakukan perancangan sistem dengan menerapkan basis data MongoDB dan ArangoDB. Basis data tersebut akan terhubung dengan file flask python, html, css dan bootstrap yang berisi program untuk memasukkan dan mencari data. Skenario ini juga menerapkan indeks teks pada data di MongoDB dan ArangoDB berformat teks.

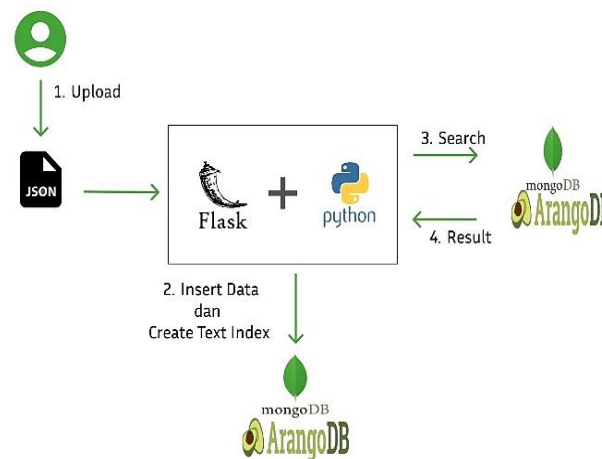
### 2.3. Tahap Hasil Analisis

Pada hasil analisis perbandingan *response time* MongoDB dan ArangoDB, temuan perbandingan waktu respon diperoleh dengan membandingkan kecepatan pencarian data pada kedua basis data.



**Gambar 1.** Metodologi Penelitian

Gambar 2 menunjukkan diagram arsitektur yang menggambarkan alur program untuk digunakan pada penelitian ini. Proses dimulai dengan pengguna mengunggah data dalam format dokumen JSON. Data kemudian disimpan ke dalam basis data MongoDB dan ArangoDB. Setelah itu, sebuah indeks dibuat pada field yang digunakan sebagai indeks pencarian. Setelah indeks dibuat, pengguna dapat mencari kata kunci dalam data yang akan dicari. Kata kunci yang ditemukan akan ditempatkan ke dalam permintaan pencarian, sehingga data yang cocok dengan kata kunci dapat diidentifikasi. Informasi yang terkumpul akan ditampilkan di situs web.



**Gambar 2.** Diagram Arsitektur

#### 2.4. MongoDB dan ArangoDB

MongoDB adalah salah satu sistem manajemen basis data NoSQL yang memungkinkan pengguna untuk melakukan pencarian data dalam berbagai bidang menggunakan dokumen sebagai basisnya. Berbeda dengan basis data relasional yang menggunakan tabel dengan baris dan kolom, MongoDB menyimpan data dalam bentuk dokumen dan koleksi. Data disimpan dalam format BSON (Binary JSON), yang dapat menangani

berbagai jenis data seperti float, integer, string, tanggal, Boolean, dan lainnya. Karena menggunakan struktur dokumen JSON, setiap data dalam MongoDB disimpan sebagai pasangan key-value [18].

ArangoDB adalah sistem database yang mendukung berbagai jenis data seperti grafik, dokumen dan key-value. Di dalam ArangoDB, dokumen-dokumen disimpan dalam format JSON dan dikelompokkan ke dalam koleksi-koleksi. Setiap dokumen memiliki atribut-atribut dan nilai-nilai terkait. Setiap dokumen juga memiliki atribut utama `_key` yang berperan seperti database key-value. ArangoDB memiliki bahasa query sendiri yang disebut Bahasa Query ArangoDB (AQL) [10].

## 2.5. Teks Indexs

Text indexing adalah fitur yang dirancang untuk memfasilitasi pencarian teks dalam string. Dengan text indexing, pencarian data dapat dipercepat dan tidak memerlukan dependensi dari pihak ketiga. Biasanya, primary key digunakan sebagai referensi untuk pencarian data yang efisien dan akurat. Namun, text indexing berbeda karena memerlukan pembuatan indeks pada seluruh data yang berupa teks atau string, sehingga memungkinkan proses pencarian data yang besar berjalan dengan cepat [13].

## 2.6. Python dan Flask

Python adalah bahasa pemrograman tingkat tinggi dan mendukung konsep pemrograman berorientasi objek. Bahasa pemrograman ini sangat fleksibel dan dapat digunakan untuk membuat berbagai jenis aplikasi, seperti situs web, aplikasi android, kecerdasan buatan dan basis data. Karena Python bersifat open source artinya terdapat kebebasan dalam pendistribusian dan penyalinannya [18].

Flask adalah kerangka kerja yang ditulis dalam bahasa pemrograman Python. Flask dikenal karena fleksibilitas dan skalabilitasnya. Flask framework yang ringan dan mudah dikembangkan sehingga memungkinkan pengguna untuk membuat program yang lebih terstruktur dengan mudah [18]. Flask diciptakan dengan tujuan untuk mempercepat pengembangan aplikasi dengan menyediakan perpustakaan dan rangkaian kode program yang telah siap pakai, memungkinkan pembangunan web tanpa perlu memulai dari nol [18].

## 2.7. HTML CSS dan Bootstrap

HTML adalah bahasa markup untuk struktur dan konten halaman web, sedangkan CSS adalah bahasa untuk mengatur tampilan dan gaya elemen-elemen HTML. HTML menyusun, CSS menghias. Dengan keduanya, pengembang web dapat membuat halaman web yang dinamis dan menarik [13].

Bootstrap adalah kerangka kerja (framework) sumber terbuka untuk pengembangan web yang memungkinkan pembuatan desain responsif dan menarik dengan cepat. Ini menyediakan kumpulan alat dan gaya pradesain untuk membangun situs web yang dapat menyesuaikan diri dengan berbagai perangkat dan ukuran layar. Dengan Bootstrap, pengembang dapat menghemat waktu dan upaya dalam mengembangkan antarmuka pengguna yang responsif dan estetis [18].

## 3. HASIL DAN PEMBAHASAN

Hasil yang diperoleh pada proses pencarian dengan teks indeks akan menunjukkan performa kinerja yang dimiliki oleh basis data MongoDB dan ArangoDB. Pada bagian ini, temuan-temuan penelitian dipaparkan dan dijelaskan secara komprehensif. Pada tahap persiapan, melakukan analisa kebutuhan perangkat keras (Hardware) dan perangkat lunak (Software).

### 3.1. Implementasi Sistem

Pada proses persiapan melakukan collect data sebanyak 1000000 data yang dibagi menjadi 1000, 10000, 100000, dan 1000000 baris data. Data tersebut merupakan data mentah yang masih berformat file Comma Separated Values (CSV). Lalu akan dikonversi ke file data dengan format JavaScript Object Notation (JSON) supaya memudahkan integrasi data ke dalam sistem basis data.

Pada proses melakukan collect data sebanyak 1000000 data yang dibagi menjadi 1000, 10000, 100000, dan 1000000 baris data. Data tersebut merupakan data mentah yang masih berformat file Comma Separated Values (CSV). Lalu akan dikonversi ke file data dengan format JavaScript Object Notation (JSON) supaya memudahkan integrasi data ke dalam sistem basis data.

---

#### Kode program 1. Program untuk MongoDB

---

```
def searchMongo():
    collection.create_index([("$**", TEXT)])
    explain_result = collection.find({'$text':
    {'$search': keyword}}).explain()
    execution_time_ms =
    explain_result['executionStats']['executionTimeMillis']
    result = collection.find({'$text': {'$search': keyword}})
    for doc in result:
        doc['_id'] = str(doc.get('_id')) result_data.append(doc)
```

---

```

return jsonify({
    "executionTime": execution_time_ms,
    "searchResults": result_data})

```

#### Kode program 2. Program untuk ArangoDB

```

def searchMongo():
    collection.create_index([("$**", TEXT)])
    explain_result = collection.find({'$text':
    {'$search': keyword}}).explain()
    execution_time_ms =
    explain_result['executionStats']['executionTimeMillis']
    result = collection.find({'$text': {'$search': keyword}})
    for doc in result:
        doc['_id'] = str(doc.get('_id')) result_data.append(doc)
    return jsonify({
        "executionTime": execution_time_ms,
        "searchResults": result_data})

```

Kode program 1 dan 2 menunjukkan program untuk pencarian teks menggunakan Flask sebagai kerangka aplikasi web pada MongoDB dan ArangoDB. Perbedaan antara kedua kode program tersebut terletak pada MongoDB menggunakan indeks teks pada koleksi dengan metode 'create\_index' dan menggunakan kunci '\$\*\*' untuk mencakup semua kolom. Indeks teks ini untuk melakukan pencarian teks penuh secara efisien dan pencarian dilakukan dengan menggunakan metode 'find' kueri '\$text'. Untuk menjalankan hasil performa kueri menggunakan 'explain', menghitung waktu eksekusi menggunakan 'executionTimeMillis' dari 'executionStats' dan dokumen hasil pencarian diubah menjadi JSON dengan 'jsonify'. Pada ArangoDB indeks teks dibuat di koleksi menggunakan metode 'fulltext'. Penggunaan AQL (ArangoDB Query Language) untuk mengeksekusi pencarian teks penuh. Pada kueri AQL, menggunakan 'EXPLAIN' untuk menghitung waktu eksekusi dan mendapatkan informasi eksekusi. Hasil pencarian diambil menggunakan AQL yang kemudian dikonversi menjadi list untuk diubah menjadi JSON dengan 'jsonify'. Perbedaan utama antara kedua pendekatan ini terletak pada mengelola kueri dan eksekusi. MongoDB menggunakan API khusus untuk melakukan pencarian teks penuh dan mengembalikan hasil [22], sementara ArangoDB menggunakan AQL untuk semua operasi kueri termasuk pencarian teks penuh [16].

Kode program tersebut saling terkait antara backend dan tampilan antarmuka pengguna. Penelitian ini mengembangkan desain program dan melakukan pemeriksaan perbandingan strategi teks indeks pada MongoDB dan ArangoDB. Website dirancang untuk mempermudah pemasukan dan pencarian data. Website dibuat dengan menggunakan framework Flask dan antarmuka pengguna Bootstrap.

The screenshot displays a web application interface for MongoDB. At the top, there is a dark blue header with the text 'MONGODB'. Below the header, the interface is divided into two main sections. The first section, titled 'Insert Data', contains a text input field labeled 'Pilih File' with the value '2400data.json' and a blue 'Insert' button. Below the button, a green box displays the JSON response: '{ "success": "Data inserted successfully!" }'. The second section, titled 'Search Data', contains a text input field with the value 'Pizza' and a blue 'Search' button. Below the button, a green box displays performance metrics: ':: Performance Query ::', 'Document Returned: 9', and 'Actual Query Execution Time (ms): 3'.

**Gambar 3.** Antarmuka Pengguna

### 3.2. Pengujian Sistem

Pada tahap hasil analisis, digunakan sebagai tahap untuk mengevaluasi kinerja basis data dan pemanfaatannya dengan sistem yang dibangun. Penelitian ini menggunakan pendekatan pengujian waktu respon untuk menguji dengan skema mengukur kinerja kecepatan kueri. Keluaran dari pembuatan data string yang disimpan dalam setiap basis data digunakan untuk menghasilkan pengujian basis data ini.

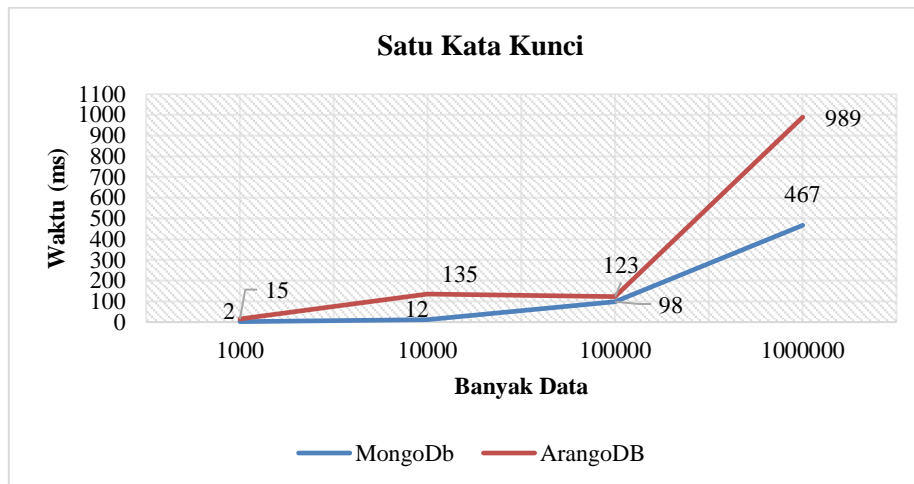
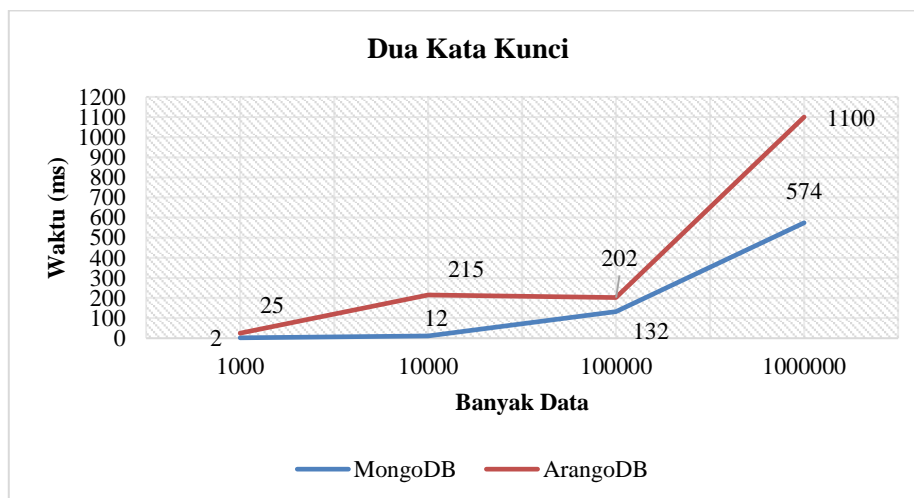
**Table 1.** Hasil waktu pencarian basis data

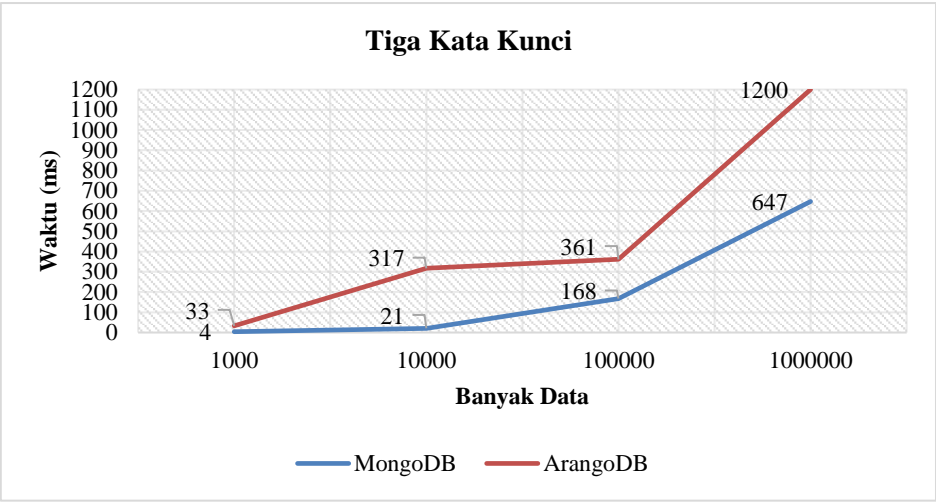
Banyak Data	MongoDB			ArangoDB		
	1 Kata	2 Kata	3 Kata	1 Kata	2 Kata	3 Kata
1000	2ms	2ms	4ms	15ms	25ms	33ms
10000	12ms	12ms	21ms	135ms	215ms	317ms
100000	98ms	132ms	168ms	123ms	202ms	361ms
1000000	467ms	574ms	647ms	989ms	1100ms	1200ms

Tabel 1 menunjukkan pengujian waktu pada basis data MongoDB dan ArangoDB. Saat waktu pencarian diuji di MongoDB dengan jumlah data yang besar, 1000 dan 100000 menunjukkan kestabilan untuk satu dan dua kata kunci, tetapi meningkat untuk tiga kata kunci, sementara data cenderung meningkat untuk 100000 dan 1000000 rekaman data. Pengujian waktu pencarian ArangoDB dengan nilai yang cenderung naik sesuai dengan jumlah data kata kunci yang dimasukkan. Selanjutnya untuk menghitung waktu pengambilan data rata-rata pada MongoDB dan ArangoDB. Rumus perhitungannya menggunakan selisih waktu antara basis data MongoDB dan ArangoDB seperti yang ditunjukkan pada persamaan 1.

$$\text{Selisih} = \text{waktu}_{\text{proses 1}} - \text{waktu}_{\text{proses 2}} \quad (1)$$

Perbandingan response time disajikan dalam bentuk grafik untuk memahami bahwa waktu respon adalah faktor kunci dalam mengevaluasi kinerja sistem. Dengan menganalisis grafik perbandingan response time dengan lebih tepat dan mendapatkan wawasan yang lebih baik tentang kinerja sistem yang diamati.

**Gambar 4.** Grafik perbandingan response time satu kata kunci**Gambar 5.** Grafik perbandingan response time dua kata kunci



Gambar 6. Grafik perbandingan response time tiga kata kunci

Table 2. Hasil Performa MongoDB dan ArangoDB

Banyak Data	MongoDB	ArangoDB	Selisih
	Rata-rata		
1 Kata	144,75ms	315,5ms	170,75ms
2 Kata	180ms	385,5ms	205,5ms
3 Kata	210ms	477,75ms	267,75ms

Tabel 5 merupakan hasil yang didapat dengan mencatat selisih waktu yang dibutuhkan oleh masing-masing basis data setelah diberikan perintah, menunjukkan basis data yang paling cepat dalam memproses pencarian data dengan jumlah data yang beragam dari tiga kata kunci yang diuji adalah MongoDB, yang membutuhkan waktu lebih cepat dibandingkan ArangoDB. Didapat bahwa satu kata kunci selisih kecepatan waktu eksekusi sebesar 170,75 ms, untuk dua kata kunci sebesar 205,5 ms, dan pada tiga kata kunci sebesar 267,75 ms. Berdasarkan hasil yang diperoleh dari pencarian data diatas, tampak bahwa seiring bertambahnya jumlah baris data, MongoDB memiliki kemampuan kecepatan pencarian yang lebih cepat jika dibandingkan dengan basis data ArangoDB, sehingga dengan hal ini tampak bahwa MongoDB memiliki kinerja yang lebih baik dalam hal pencarian data jika dibandingkan dengan ArangoDB.

Faktor yang mempengaruhi peningkatan kecepatan waktu eksekusi dalam pencarian data dari hasil diatas dapat disebabkan oleh beberapa faktor yang terkait dengan arsitektur dan mekanisme kerja masing-masing basis data. MongoDB memiliki desain yang dioptimalkan untuk kinerja pencarian data dengan menggunakan indeks B-tree yang efisien dalam mengakses data berdasarkan kunci, indeks B-tree memungkinkan MongoDB untuk mencari data secara cepat bahkan ketika jumlah data semakin besar, sehingga memberikan keunggulan dalam respons time pencarian [19]. Selain itu, MongoDB juga memiliki mekanisme caching yang baik untuk menyimpan hasil pencarian sebelumnya, sehingga ketika melakukan pencarian dengan kata kunci yang serupa, MongoDB dapat mengembalikan hasil secara instan dari cache tanpa perlu mengakses data yang sebenarnya, sehingga menyebabkan peningkatan kecepatan eksekusi [20].

Faktor lain karena terdapat perbedaan dalam struktur penyimpanan data antara MongoDB dan ArangoDB juga dapat mempengaruhi kecepatan pencarian data. MongoDB menggunakan penyimpanan dokumen yang mengorganisir data dalam dokumen JSON yang terstruktur [13], sedangkan ArangoDB juga menggunakan JSON tetapi memiliki model penyimpanan yang lebih kompleks dengan koleksi, dokumen, dan grafik [12]. Dalam konteks pencarian data, struktur penyimpanan yang lebih sederhana dan terstruktur seperti yang dimiliki MongoDB dapat memudahkan proses pencarian dan mengurangi waktu yang diperlukan untuk menemukan data yang relevan.

Faktor pengoptimalan internal yang dilakukan oleh MongoDB dalam hal query optimization juga dapat berkontribusi terhadap peningkatan kecepatan pencarian data. MongoDB memiliki mekanisme query optimizer yang canggih yang secara otomatis menganalisis kueri yang diberikan dan memilih rencana eksekusi yang paling efisien berdasarkan indeks yang tersedia dan statistik data [21]. Dengan demikian, MongoDB mampu mengeksekusi kueri dengan lebih efisien bahkan pada kondisi pencarian yang kompleks, yang berujung pada peningkatan signifikan dalam respons time pencarian data.

#### 4. KESIMPULAN

Dari hasil penelitian, terlihat bahwa pemberian fungsi teks indeks pada basis data MongoDB dan ArangoDB dapat mempercepat waktu pencarian. Dengan jumlah 1000, 10000, 100000, dan 1000000 rekaman data didapat selisih waktu eksekusi dari ketiga kata kunci adalah satu kata kunci selisih waktu eksekusi antara MongoDB dan ArangoDB adalah 170,75 ms, untuk dua kata kunci selisih waktu eksekusi 205,5 ms, dan untuk tiga kata kunci 267,75 ms. Dari hasil ini, dapat disimpulkan bahwa MongoDB cenderung memiliki waktu respon yang lebih cepat dibandingkan dengan ArangoDB, terutama ketika jumlah data yang diolah semakin besar. Namun, penting untuk dicatat bahwa ArangoDB juga dapat menunjukkan kinerja yang baik pada percobaan dengan jumlah data yang relatif sedikit, sehingga untuk kasus-kasus tertentu keduanya bisa memiliki kinerja yang serupa.

Teknik replikasi basis data dapat memandu penelitian di masa depan dan menguji hasil pencarian. Studi lebih lanjut dapat memberi tahu pengembang tentang strategi paling efektif untuk mengelola data dalam jumlah besar dan mempercepat pencarian di MongoDB dan ArangoDB.

#### REFERENSI

- [1] J. Homepage, S. Budiman, F. Fadhila, V. Ardiyansyah Saputro, And E. Utami, "Ijeit (Indonesian Journal On Computer And Information Technology) Perbandingan Performa Sql Dan Nosql Dengan Php Pada 5 Juta Data," 2021.
- [2] R. Sridevi And S. Srimathi, "Populating Indian Gst Details Into Java Apache Derby Database Powered By Glassfish Server," *Inventive Communication And Computational Technologies: Proceedings Of Iccict 2019*, Pp. 409–421, 2020.
- [3] A. Akhtar, "Popularity Ranking Of Database Management Systems," Jan. 2023, [Online]. Available: <http://arxiv.org/abs/2301.00847>
- [4] A. Sonita And M. Sari, "Implementasi Algoritma Sequential Searching Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik," 2018. [Online]. Available: [www.ejournal.unib.ac.id/index.php/pseudocode](http://www.ejournal.unib.ac.id/index.php/pseudocode)
- [5] D. R. Zmaranda, C. I. Moisi, C. A. Györödi, R. Györödi, And L. Bandici, "An Analysis Of The Performance And Configuration Features Of Mysql Document Store And Elasticsearch As An Alternative Backend In A Data Replication Solution," *Applied Sciences (Switzerland)*, Vol. 11, No. 24, Dec. 2021, Doi: 10.3390/app112411590.
- [6] T. N. Khasawneh, M. H. Al-Sahlee, And A. A. Safia, "Sql, Newsq, And Nosql Databases: A Comparative Survey," In *2020 11th International Conference On Information And Communication Systems, Icics 2020*, Institute Of Electrical And Electronics Engineers Inc., Apr. 2020, Pp. 13–21. Doi: 10.1109/Icics49469.2020.239513.
- [7] A. Septia Maharani *Et Al.*, "Perancangan Data Base Kasir Dan Persediaan Barang Menggunakan Mongodb," 2022.
- [8] T. L. Sinaga, N. Charibaldi, And N. Heri Cahyana, "Perbandingan Waktu Respon Aplikasi Database Nosql Elasticsearch Dan Mongodb Pada Pengujian Operasi Crud," 2023.
- [9] H. Chan, L. Xu, H. Liu, R. Zhang, And A. Sangaiah, "System Design Of Cloud Search Engine Based On Rich Text Content. Journal: Mobile Networks And Applications," *Mobile Networks And Applications*, Pp. 459–472, 2020.
- [10] H. Sugiharto, "Perbandingan Response Time Replikasi Database Arangodb Dan Mongodb." [Online]. Available: <https://www.researchgate.net/publication/338169009>
- [11] A. Hajar, E. Utami, And H. Al Fatta, "Penggunaan Fulltext Indexing Untuk Meningkatkan Efisiensi Pencarian Data Pada Basis Data Mysql Use Of Fulltext Indexing To Improve Data Search Efficiency In Mysql Databases," Vol. 12, No. 2, 2022, Doi: 10.30700/jst.v12i2.1264.
- [12] D. Rusman And R. Setiawan, "Perbandingan Kinerja Database Nosql Mongodb Dan Arangodb." [Online]. Available: <https://www.researchgate.net/publication/348432404>
- [13] F. Frankie And Y. A. Susetyo, "Implementation Of Text Indexing System In Web-Based Document Search Application Using Mongodb," *Jurnal Teknik Informatika (Jutif)*, Vol. 4, No. 5, Pp. 1081–1087, Oct. 2023, Doi: 10.52436/1.Jutif.2023.4.5.959.
- [14] I. Asyhari, M. Subchan Mauludin, And J. M. Tengah, "Implementasi Full Text Search Pada Sistem Informasi Perpustakaan Menggunakan Laravel," Vol. 1, No. 1, Pp. 1–9, 2019, [Online]. Available: [www.elastic.co](http://www.elastic.co)
- [15] <https://www.tiobe.com/>, "February Headline: The Go Programming Language Enters The Top 10," <https://www.tiobe.com/>.
- [16] Arangodb. (N.D.). Full-Text Token Search - Arangosearch - Arangodb Documentation. Retrieved February 25, 2024, From <https://docs.arangodb.com/3.11/index-and-search/arangosearch/full-text-token-search/>
- [17] Pratama, I. G. A. E., Satwika, I. P., & Wijaya, I. N. Y. A. (2022). Analisis Perbandingan Performa Api Metode Rest Dan Graphql Dengan Php Dan Go. *Jurnal Teknologi Informasi Dan Komputer*, 8(4).



- [18] Susetyo, Y. A. (2023). Analisis Dan Penerapan Database Mongodb Pada Aplikasi Manajemen Dokumen Di Pt. Xyz. Jurnal Jtik (Jurnal Teknologi Informasi Dan Komunikasi), 7(4), 570-578.
- [19] Chopade, R., & Pachghare, V. (2020). Mongodb Indexing For Performance Improvement. In Ict Systems And Sustainability: Proceedings Of Ict4sd 2019, Volume 1 (Pp. 529-539). Springer Singapore.
- [20] Ilmi, I. F. (2023). Analisis Kinerja Mekanisme Caching Mongodb Cluster Pada Moodle. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 7(7), 3085-3094.
- [21] Ramasamy, Chitra And Zolkepli, Maslina (2020) Bibliographic Data Retrieval Using Query Optimization Techniques In Mongodb. Journal Of Advanced Research In Dynamical And Control Systems, 12 (Spec.4). 1524 - 1532. Issn 1943-023x
- [22] Mongodb. (N.D.). \$Search — Mongodb Manual. Retrieved February 25, 2024, From <https://www.mongodb.com/docs/manual/reference/operator/aggregation/search/>