

**PERBANDINGAN KECEPATAN PENGAMBILAN DATA ANTARA API
PUBLIK AN MONGODB DALAM APLIKASI AGREGASI BERITA**

***DATA RETRIEVAL SPEED COMPARISON BETWEEN PUBLIC API AND
MONGODB IN NEWS AGGREGATION APPLICATIONS***

Febri Nur Diansah¹, Umar Zaky²

Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Teknologi
Yogyakarta¹

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Teknologi
Yogyakarta²

E-mail: febrinurdiansah0@gmail.com¹, umar.zaky@staff.uty.ac.id²

ABSTRACT

The rapid development of digital technology drives the need for real-time data processing systems, particularly in news aggregation applications. This study aims to compare the response times between direct data retrieval from public Application Programming Interfaces (APIs) and the use of MongoDB as a local database. Utilizing Node.js as the backend, the research measures the response times of data retrieval from both sources using three data scales: 50, 150, and 300 JSON data points. The results show that for smaller data sets, public APIs provide faster response times, while the difference in response times between the API and MongoDB becomes smaller with larger data sets. The average response time for data retrieval from the API is 735.6 ms for 50 data points, 910.8 ms for 150 data points, and 737.6 ms for 300 data points, whereas MongoDB records average response times of 872 ms, 909.4 ms, and 912.4 ms for each respective data scale. These findings indicate that although MongoDB has the potential to store historical data and reduce dependence on APIs, the speed of data retrieval from public APIs remains superior at certain data sizes. This research provides important insights into the effectiveness of using MongoDB in applications requiring real-time data access and historical data management.

Keywords: *Data Retrieval, Public API, MongoDB, Response Time.*

ABSTRAK

Perkembangan teknologi digital yang pesat mendorong kebutuhan akan sistem pengolahan data real-time, terutama dalam aplikasi agregasi berita. Penelitian ini bertujuan untuk membandingkan waktu respons antara pengambilan data langsung dari Application Programming Interface (API) publik dan penggunaan MongoDB sebagai basis data lokal. Dengan memanfaatkan Node.js sebagai backend, penelitian ini mengukur waktu respons pengambilan data dari dua sumber menggunakan tiga skala data: 50, 150, dan 300 JSON data points. Hasil pengujian menunjukkan bahwa untuk jumlah data yang lebih kecil, API publik memberikan waktu respons yang lebih cepat, sedangkan dengan jumlah data yang lebih besar, perbedaan waktu respons antara API dan MongoDB menjadi semakin kecil. Rata-rata waktu respons untuk pengambilan data dari API adalah 735,6 ms untuk 50 data, 910,8 ms untuk 150 data, dan 737,6 ms untuk 300 data, sedangkan MongoDB mencatat rata-rata waktu respons sebesar 872 ms, 909,4 ms, dan 912,4 ms pada masing-masing skala data. Temuan ini menunjukkan bahwa meskipun MongoDB memiliki potensi untuk menyimpan data historis dan mengurangi ketergantungan pada API, kecepatan pengambilan data dari API publik tetap lebih unggul pada ukuran data tertentu. Penelitian ini memberikan wawasan penting tentang efektivitas penggunaan MongoDB dalam aplikasi yang memerlukan akses data real-time dan pengelolaan data historis.

Kata Kunci: Pengambilan Data, API Publik, MongoDB, Waktu Respon.

PENDAHULUAN

Perkembangan teknologi digital telah mendorong munculnya berbagai sistem pengolahan data, terutama dalam pengambilan data secara *real-time*. Dalam aplikasi agregasi berita, kecepatan dan efisiensi pengambilan data menjadi aspek yang sangat penting. Pengguna membutuhkan akses cepat terhadap berita terbaru, dan aplikasi yang mampu menyediakan data secara *real-time* memiliki nilai yang tinggi. Dalam konteks ini, penggunaan Application Programming Interface (API) untuk mengambil data langsung dari sumber eksternal menjadi salah satu metode yang umum digunakan. Namun, keterbatasan API, seperti waktu respon yang lebih lama dan ketergantungan pada ketersediaan data, mengarah pada pencarian alternatif lain, seperti penggunaan basis data lokal.

Beberapa penelitian terdahulu menunjukkan bahwa sistem basis data NoSQL seperti MongoDB memberikan keunggulan dibandingkan dengan basis data SQL tradisional, seperti Oracle, dalam hal skalabilitas dan fleksibilitas dalam mengelola data yang tidak terstruktur (Zohaib Khan Software et al., 2022). SQL lebih cocok untuk data terstruktur dan memiliki skema tetap, sedangkan NoSQL, termasuk MongoDB, lebih unggul dalam menangani data yang dinamis dan tidak terstruktur. Dalam konteks aplikasi agregasi berita, di mana data yang diambil sering kali berubah dan berkembang, basis data NoSQL dianggap lebih sesuai untuk mendukung performa aplikasi secara keseluruhan.

Selain itu, penelitian ini mengambil referensi dari jurnal mengenai pengujian performa API, di mana waktu respon menjadi fokus utama. Melalui metode *load testing*, ditemukan bahwa API publik sering mengalami

fluktuasi dalam waktu respon, yang dapat mempengaruhi pengalaman pengguna secara keseluruhan (Setiawan et al., 2022). Dengan menyimpan data secara lokal menggunakan MongoDB, diharapkan dapat mengurangi ketergantungan pada API dan meningkatkan waktu respon aplikasi.

Dalam konteks agregasi berita, salah satu solusi yang sering digunakan adalah penggabungan API publik untuk memperoleh data terbaru. Namun, data dari API publik bersifat sementara dan hanya bertahan beberapa hari saja. Tantangan muncul dalam pemeliharaan data historis. Di sinilah MongoDB menawarkan solusi dengan kemampuannya untuk menyimpan data secara efisien. Dengan menggunakan MongoDB, aplikasi agregasi berita tidak hanya dapat mengakses data *real-time* tetapi juga mempertahankan data historis untuk analisis lebih lanjut.

Penelitian ini bertujuan untuk membandingkan waktu respon antara pengambilan data langsung dari API publik dan pengambilan data dari MongoDB. Dengan fokus pada kecepatan pengambilan data, penelitian ini bertujuan memberikan gambaran kuantitatif mengenai mana yang lebih efisien di antara kedua metode tersebut. Hasil dari penelitian ini dapat memberikan wawasan mengenai efektivitas penggunaan MongoDB sebagai solusi basis data dalam aplikasi yang memerlukan akses data secara *real-time*.

Oleh karena itu, pertanyaan yang diangkat dalam penelitian ini adalah: “Apakah pengambilan data menggunakan MongoDB memiliki kecepatan respon yang lebih baik dibandingkan dengan pengambilan data langsung dari API publik dalam aplikasi agregasi berita?” Penelitian ini akan

memberikan kontribusi pada pemahaman lebih lanjut mengenai kecepatan pengambilan data dan pengelolaan data historis dalam aplikasi agregasi berita.

METODE

Dalam penelitian ini, saya menggunakan metode eksperimen untuk membandingkan waktu respon pengambilan data antara API publik dan MongoDB. Tujuan eksperimen ini adalah untuk mengukur dan menganalisis kecepatan respon dari kedua metode tersebut dalam konteks aplikasi agregasi berita.

Teknologi dan Alat yang Digunakan

Pada tahap ini, beberapa teknologi dan alat digunakan untuk mendukung proses penelitian ini:

- a. **Bahasa Pemrograman:** Node.js digunakan sebagai backend untuk mengelola logika pengambilan data dari API dan MongoDB. Node.js adalah lingkungan *runtime server-side* yang bersifat *open source* dan dapat digunakan lintas platform. Node.js menggunakan mesin JavaScript Chrome V8, yang memungkinkannya menjalankan JavaScript di luar browser (Sauda & Barokah, 2022). Kemampuannya dalam menangani *asynchronous I/O* sangat berguna dalam pengambilan data dari sumber eksternal, sehingga efisien untuk digunakan dalam aplikasi agregasi berita.
- b. **Basis Data:** MongoDB digunakan untuk menyimpan salinan data berita dari API guna dilakukan perbandingan waktu pengambilan data. MongoDB tidak hanya membebaskan pengguna dari biaya lisensi, tetapi juga bersifat *open source*, memungkinkan pengembang untuk mengeksplorasi dan

menyesuaikan sesuai kebutuhan. Berbeda dengan relational database, MongoDB adalah DBMS berbasis dokumen, yang menyimpan data dalam format BSON (Binary JSON), memberikan fleksibilitas tinggi dan pemodelan data yang lebih intuitif (Maharani, 2022). Database ini di-hosting di layanan web MongoDB dan diintegrasikan dengan Vercel untuk memudahkan akses dari backend.

- c. **Data:** Data yang digunakan berasal dari beberapa API publik yang menyediakan berita terbaru. Sebanyak 300 data JSON diambil dari API dan disimpan di MongoDB. Jumlah data ini ditetapkan karena keterbatasan jumlah data yang tersedia secara publik, yang memungkinkan pengujian dilakukan dengan sumber data yang konsisten karena data yang bersifat sementara.
- d. **Platform Hosting:** Vercel digunakan untuk menghosting aplikasi secara *cloud*, memungkinkan aksesibilitas yang optimal selama pengujian. Sebagai platform *serverless*, Vercel mendukung aplikasi web statis dan berbasis JavaScript, memungkinkan pengembang fokus pada kode tanpa mengelola infrastruktur server. Dengan fitur integrasi GitHub dan auto deployment, Vercel mempercepat siklus pengembangan dan mendukung praktik DevOps modern (Sinlae et al., 2024).

Langkah-langkah Penelitian

Langkah penelitian ini menjelaskan proses pengambilan data secara rinci, mencakup metode yang digunakan untuk mendapatkan data dari kedua sumber.

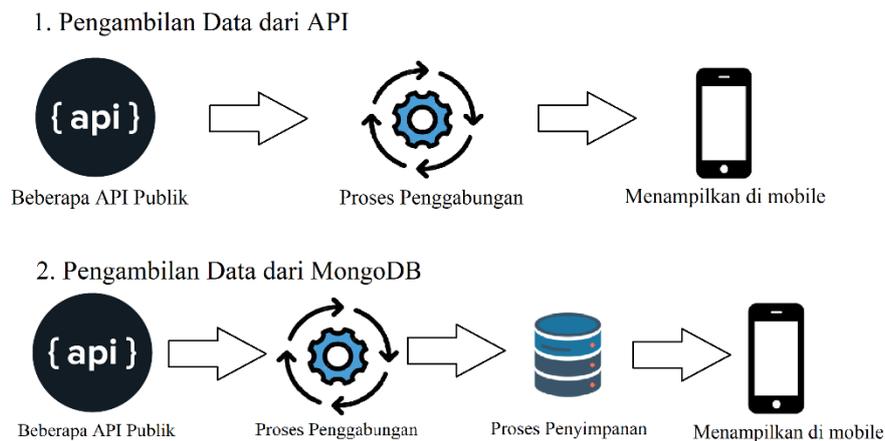
- a. **Pengambilan Data dari API:** Data diambil secara langsung dari beberapa API publik yang menyediakan berita terkini. Proses

pengambilan data ini dilakukan dengan mengirimkan permintaan HTTP dan mencatat waktu pengambilan mulai dari saat permintaan dikirim hingga data diterima secara penuh oleh server. Metode ini memungkinkan peneliti untuk mengukur dampak latensi jaringan terhadap waktu respons.

- b. **Pengambilan Data dari MongoDB:** Data yang telah diambil dari API disimpan di MongoDB yang di-hosting di web MongoDB dan diakses melalui Vercel. Pengujian dilakukan

dengan mengambil data dari MongoDB untuk mengukur waktu respons dari basis data yang di-hosting secara *cloud*. Dengan cara ini, penelitian mengevaluasi seberapa cepat data dapat diambil dari database dibandingkan dengan pengambilan langsung dari API.

Untuk memberikan gambaran yang lebih jelas mengenai perbandingan kedua metode pengambilan data, dapat dilihat pada Gambar 1 di bawah ini:



Gambar 1. Diagram Flow Perbandingan Pengambilan Data

Pengujian dan Pengukuran

Bagian ini menjelaskan bagaimana pengujian dilakukan untuk mendapatkan hasil yang dapat diandalkan dan bagaimana pengukuran dilakukan terhadap waktu respons masing-masing metode.

- a. **Variabel yang Diukur:** Variabel utama yang diukur dalam penelitian ini adalah waktu respons (dalam *millisecond*) untuk setiap metode pengambilan data, yaitu:

- 1) Pengambilan data langsung dari API.
- 2) Pengambilan data dari MongoDB.

- b. **Pengulangan Pengujian:** Pengambilan data dari API dan

MongoDB dilakukan dalam tiga tahapan dengan jumlah data yang berbeda: 50, 150, dan 300 data JSON. Setiap tahapan diuji sebanyak 5 iterasi untuk menghasilkan data yang representatif. Dengan melakukan beberapa iterasi, peneliti dapat memperoleh rata-rata waktu respons, yang memberikan gambaran yang lebih representatif tentang performa masing-masing metode.

Analisis Data

Analisis akan dilakukan dengan membandingkan waktu rata-rata yang diperoleh dari kedua metode pengambilan data (API dan MongoDB).

Grafik dan tabel digunakan untuk menampilkan hasil pengujian. Fokus utama dari analisis ini adalah untuk melihat mana yang lebih efisien dalam hal waktu respon pengambilan data. Selain itu, analisis juga mempertimbangkan faktor-faktor seperti perubahan ukuran data dan pengaruh hosting cloud terhadap kecepatan pengambilan data.

Dengan pendekatan ini, diharapkan penelitian ini dapat memberikan pandangan yang jelas tentang perbedaan performa antara penggunaan API publik dan MongoDB dalam konteks agregasi berita, serta pentingnya pengelolaan data historis untuk keperluan aplikasi masa depan.

HASIL DAN PEMBAHASAN

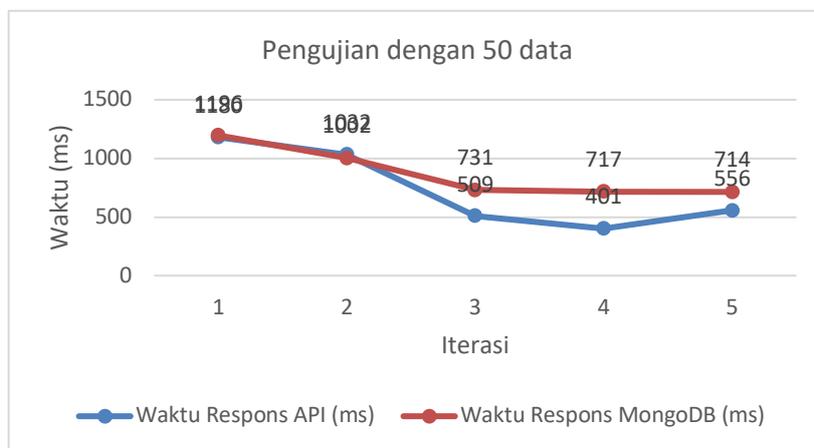
Bagian ini menyajikan hasil pengujian yang dilakukan untuk membandingkan waktu respons pengambilan data dari API publik dan MongoDB. Pengujian dilakukan dengan tiga tahapan jumlah data: 50, 150, dan 300 data JSON. Setiap tahapan diuji sebanyak 5 iterasi untuk menghasilkan data yang representatif.

Hasil Pengujian untuk 50 Data

Pengujian dilakukan dengan mengambil data dari API publik dan MongoDB sesuai dengan metode yang telah dijelaskan sebelumnya. Tabel di bawah ini menunjukkan waktu respons rata-rata (dalam *millisecond*) untuk pengambilan data dari masing-masing sumber:

Tabel 1. Hasil perbandingan waktu respons dengan 50 data

Iterasi	Waktu Respons API (ms)	Waktu Respons MongoDB (ms)
1	1180	1196
2	1032	1002
3	509	731
4	401	717
5	556	714
Rata-rata	735.6	872



Gambar 2. Grafik perbandingan waktu respons dengan 50 data

Dari hasil di atas, terlihat bahwa waktu respons rata-rata untuk pengambilan data dari API publik adalah 735.6 ms, sementara waktu respons dari

MongoDB adalah 872 ms. Hal ini menunjukkan bahwa API publik lebih efisien dalam pengambilan data pada jumlah ini.

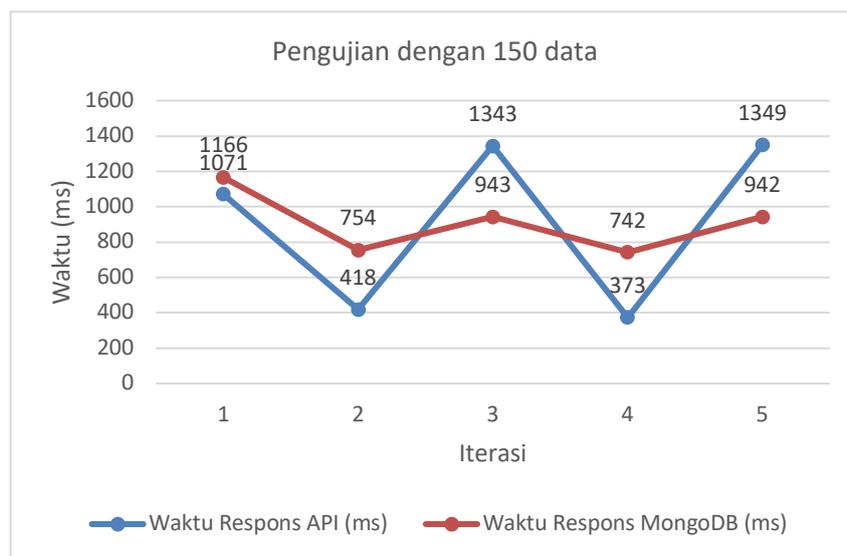
Hasil Pengujian untuk 150 Data

Pengujian dilanjutkan dengan jumlah data yang lebih banyak, yaitu 150

data JSON. Berikut adalah hasil waktu respons rata-rata:

Tabel 2. Hasil perbandingan waktu respons dengan 150 data

Iterasi	Waktu Respons API (ms)	Waktu Respons MongoDB (ms)
1	1071	1166
2	418	754
3	1343	943
4	373	742
5	1349	942
Rata-rata	910.8	909.4



Gambar 3. Grafik perbandingan waktu respons dengan 150 data

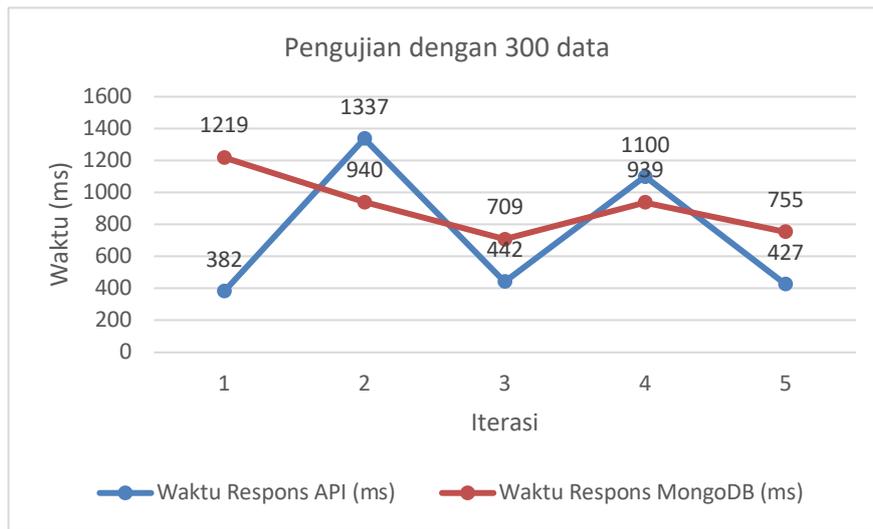
Dari hasil di atas, waktu respons rata-rata API publik mencapai 910,8 ms, sementara MongoDB sedikit lebih cepat dengan rata-rata 909,4 ms. Pada tahap ini, perbedaan performa antara API publik dan MongoDB sangat kecil, menunjukkan bahwa keduanya memiliki kecepatan yang hampir sama.

Hasil Pengujian untuk 300 Data

Pengujian terakhir dilakukan dengan jumlah data maksimal, yaitu 300 data JSON. Hasilnya ditampilkan dalam tabel berikut:

Tabel 3. Hasil perbandingan waktu respons dengan 300 data

Iterasi	Waktu Respons API (ms)	Waktu Respons MongoDB (ms)
1	382	1219
2	1337	940
3	442	709
4	1100	939
5	427	755
Rata-rata	737.6	912.4



Gambar 4. Grafik perbandingan waktu respons dengan 300 data

Dari hasil pengujian ini, waktu respons rata-rata API publik adalah 737,6 ms, sementara MongoDB membutuhkan 912,4 ms. Dengan jumlah data yang lebih besar, API publik menunjukkan keunggulan yang signifikan dibandingkan MongoDB dalam hal kecepatan pengambilan data.

Analisis Keseluruhan

Berdasarkan pengujian dari tiga tahap (50, 150, dan 300 data), terdapat

pola yang menunjukkan bahwa MongoDB tidak selalu lebih cepat dari API publik dalam pengambilan data. Meskipun pada awalnya diasumsikan bahwa MongoDB akan lebih efisien dalam menangani data dalam jumlah besar, hasil menunjukkan bahwa API publik lebih unggul pada jumlah data yang lebih kecil dan tetap kompetitif bahkan pada jumlah data yang lebih besar. Berikut adalah hasil rata-rata waktu respons dari setiap pengujian:

Tabel 4. Hasil rata-rata waktu respons setiap tahapan pengujian

Tahapan Pengujian	Waktu Respons API (ms)	Waktu Respons MongoDB (ms)	Selisih (ms)
50 data	735.6	872	-136.4
150 data	910.8	909.4	1.4
300 data	737.6	912.4	-174.8

- a. Pada pengujian dengan 50 data, API publik menunjukkan waktu respons rata-rata sebesar 735,6 ms, lebih cepat dibandingkan MongoDB yang membutuhkan 872 ms. Perbedaan ini signifikan, dengan selisih 136,4 ms.
- b. Pada pengujian dengan 150 data, waktu respons API meningkat menjadi 910.8 ms, sementara MongoDB tetap stabil di 909.4 ms. Selisih waktu respons menjadi sangat

- kecil, yaitu 1.4 ms, mengindikasikan bahwa pada jumlah data ini, kedua metode menunjukkan performa yang hampir sama.
- c. Pada pengujian dengan 300 data, waktu respons API kembali meningkat menjadi 737.6 ms, sedangkan MongoDB meningkat menjadi 912.4 ms. Walaupun ada peningkatan waktu pada MongoDB, selisih antara kedua metode tetap

kecil, yaitu 174.8 ms, menunjukkan bahwa keduanya masih bersaing ketat pada jumlah data yang lebih besar.

Faktor-faktor yang mempengaruhi perbedaan waktu respons ini antara lain adalah:

- a. **Latensi Jaringan:** Pengambilan data dari API publik sangat bergantung pada kondisi jaringan. Latensi jaringan yang berfluktuasi dapat menyebabkan waktu respons API menjadi lebih lama, seperti yang terlihat pada pengujian 50, 150, dan 300 data.
- b. **Beban Server API:** Server API yang melayani banyak permintaan dari pengguna lain pada saat bersamaan dapat menyebabkan peningkatan waktu respons.
- c. **Kinerja MongoDB:** MongoDB, yang dihosting secara online, mungkin terpengaruh oleh latensi jaringan dan beban server eksternal, sehingga kinerjanya bisa bervariasi. Pada pengujian dengan 300 data, waktu respons MongoDB mengalami peningkatan, yang mungkin disebabkan oleh beban data yang lebih besar yang perlu diolah.

Perhitungan Total Waktu Respons

Penghitungan total waktu respons dari semua pengujian yang dilakukan juga memberikan gambaran yang lebih jelas mengenai perbedaan performa antara API publik dan MongoDB. Berikut adalah total waktu respons dari 5 iterasi pada setiap tahapan:

- a. **Total Waktu Respons API Publik:**
 - 50 Data: 735.6 ms
 - 150 Data: 910.8 ms
 - 300 Data: 737.6 ms
 - Total Keseluruhan (API): 2384 ms
- b. **Total Waktu Respons MongoDB:**
 - 50 Data: 872 ms
 - 150 Data: 909.4 ms
 - 300 Data: 912.4 ms

Total Keseluruhan (MongoDB): 2693.8 ms

Dari perhitungan ini, total waktu respons untuk pengambilan data dari API publik mencapai 2384 ms, sedangkan MongoDB mencapai 2693.8 ms. Perbedaan total waktu respons sebesar -309.8 ms menunjukkan adanya perbedaan performa yang signifikan antara kedua metode pengambilan data.

SIMPULAN DAN SARAN

Penelitian ini membahas perbandingan kecepatan pengambilan data antara API publik dan MongoDB dalam aplikasi agregator berita. Berdasarkan hasil pengujian, terdapat beberapa temuan utama yang dapat diambil:

- a. **Pada pengujian dengan 50 data:** API publik menunjukkan waktu respons rata-rata yang lebih cepat (735.6 ms) dibandingkan MongoDB (872 ms). Ini menunjukkan bahwa pada pengambilan data dalam jumlah terbatas, API publik lebih efisien.
- b. **Pada pengujian dengan 150 data:** Kedua metode menunjukkan waktu respons yang hampir setara, dengan perbedaan waktu respons hanya 1.4 ms. Hasil ini mengindikasikan bahwa pada jumlah data yang lebih besar, perbedaan performa antara MongoDB dan API publik hampir tidak signifikan.
- c. **Pada pengujian dengan 300 data:** API publik tetap lebih cepat dengan waktu respons rata-rata 737.6 ms, dibandingkan MongoDB yang membutuhkan 912.4 ms. Hal ini menunjukkan bahwa API publik masih unggul dalam pengambilan data secara real-time, bahkan pada jumlah data yang lebih besar.

Namun, perlu diperhatikan bahwa data dari API publik hanya

tersedia dalam jangka waktu tertentu, sementara MongoDB dapat menyimpan data secara permanen. Oleh karena itu, meskipun dalam beberapa pengujian API publik menunjukkan waktu respons yang lebih cepat, MongoDB memiliki keunggulan dalam hal retensi data dan konsistensi akses data yang lebih lama.

Kesimpulannya, pemilihan antara API publik dan MongoDB bergantung pada kebutuhan aplikasi. Jika aplikasi membutuhkan akses data *real-time* dalam jumlah terbatas, API publik mungkin merupakan pilihan yang lebih baik. Namun, jika retensi data dan akses data yang lebih konsisten dalam jangka panjang diperlukan, MongoDB menawarkan solusi yang lebih unggul. Penelitian ini diharapkan memberikan wawasan penting bagi pengembang dan arsitek sistem dalam merancang aplikasi agregator berita yang optimal.

Vercel. *Jurnal Siber Multi Disiplin (JSMD)*, 2(2), 98–106. <https://creativecommons.org/licenses/by/4.0/>

Zohaib Khan Software, M., Engineer at SMBBIT Karachi, D., Uz Zaman, F., Adnan Engineer at KPT Karachi, M., Imroz, A., & Abdul Rauf, M. (2022). Comparative Case Study: An Evaluation of Performance Computation Between SQL And NoSQL Database. *SJHSE Sindh Journal of Headways in Software Engineering*, 01(02), 2.

DAFTAR PUSTAKA

- Maharani, A. (2022). Perancangan Data Base Kasir Dan Persediaan Barang Menggunakan Mongoddb. *Jurnal Data Mining Dan Sistem Informasi*, 3(1), 32. <https://doi.org/10.33365/jdmsi.v3i1.1941>
- Sauda, S., & Barokah, M. (2022). Penerapan NodeJS dan PostgreSQL sebagai Backend pada Aplikasi Ecommerce Localla. *Infotech*, 8(2), 101–105. <https://doi.org/10.31949/infotech.v8i2.2944>
- Setiawan, G. H., Made, I., Adnyana, B., & Budiarta, K. (2022). Pengujian Performa API (Application Programming Interface) dengan Metode Load Testing. *Seminar Nasional Corisindo*, 539–542.
- Sinlae, F., Ahmad Dafianto, I., & Setya Maulana, R. (2024). Aplikasi Web Tanpa Server menggunakan