BAB I

PENDAHULUAN

1.1 Latar Belakang

Pesatnya perkembangan teknologi dan adopsi layanan berbasis komputasi awan (*cloud computing*) telah mendorong peningkatan *volume* data secara signifikan. Dalam menghadapi beban kerja modern yang bersifat masif dan dinamis, sistem basis data relasional (*RDBMS*) tradisional sering kali menemui keterbatasan dalam hal skalabilitas dan fleksibilitas. Untuk mengatasi tantangan ini, basis data *NoSQL* seperti *MongoDB* hadir dengan arsitektur terdistribusi yang dirancang untuk skala besar dan kinerja tinggi (Chauhan, 2019).

Namun, arsitektur terdistribusi juga membawa tantangan tersendiri, khususnya dalam menjaga konsistensi data. Teorema *Consistency, Availability, Partition Tolerance (CAP)* menjelaskan bahwa dalam sistem terdistribusi, hanya dua dari tiga aspek tersebut yang dapat dicapai secara bersamaan (Gorbenko dkk., 2020). Dalam konteks ini, ketika terjadi *partition tolerance* (P), sistem harus memilih antara menjaga *consistency* (C) atau memastikan *availability* (A). Inilah yang dikenal sebagai prinsip CAP, di mana keputusan desain sistem bergantung pada prioritas antara konsistensi dan ketersediaan.

MongoDB menyediakan fleksibilitas untuk memilih tingkat konsistensi operasi melalui parameter seperti *writeConcern* (Schultz dkk., 2019). Mode *eventual consistency* (*writeConcern*: 1) cenderung memprioritaskan ketersediaan, memungkinkan operasi tetap berlangsung walaupun sebagian data belum

direplikasi ke seluruh *node*. Sebaliknya, mode *strong consistency* (*writeConcern: "majority"*) menjamin bahwa data yang ditulis telah direplikasi ke sebagian besar *node* sebelum dianggap berhasil, sehingga menjaga konsistensi meskipun berisiko menurunkan ketersediaan atau meningkatkan *latency* (Dhanagari, 2024).

Dalam implementasi nyata, keputusan untuk memilih mode konsistensi tidak hanya dipengaruhi oleh teori, tetapi juga oleh faktor praktis seperti karakteristik beban kerja (apakah lebih banyak operasi baca atau tulis), skala operasi, dan keterbatasan infrastruktur. Banyak aplikasi, terutama pada tahap awal pengembangan atau yang berjalan di lingkungan *cloud* skala kecil, menggunakan model data yang umum seperti profil pengguna, *log* aktivitas, atau kata*log* produk (Ferreira dkk., 2024). Meskipun banyak penelitian telah membandingkan *eventual consistency* dan *strong consistency*, sering kali pengujian dilakukan pada perangkat keras kelas atas atau dengan model data generik. Masih terdapat kesenjangan pemahaman praktis mengenai bagaimana *trade-off* ini berlaku pada infrastruktur *cloud* yang terbatas sumber dayanya (contoh: *AWS EC2 t2.micro*), terutama saat menangani beban kerja dengan model data yang realistis seperti data profil pengguna. Lingkungan seperti ini sangat umum namun performanya kritis untuk kesuksesan aplikasi.

Oleh karena itu, penelitian ini bertujuan untuk mengukur dan menganalisis secara kuantitatif performa *MongoDB* pada dua mode konsistensi *eventual* dan *strong*. Pengujian dilakukan secara sistematis dengan beragam jenis beban kerja (seimbang, dominan baca, dan dominan tulis) dan varian jumlah operasi. Objek yang diuji adalah database *NoSQL* yang menangani data profil pengguna pada

lingkungan *cloud* skala kecil. Metrik yang dievaluasi tidak hanya mencakup performa dari sisi klien seperti *latency* dan *throughput*, tetapi juga dampak langsungnya terhadap penggunaan sumber daya server (*CPU*, *RAM*, dan *I/O Wait*). Hasil dari penelitian ini diharapkan dapat memberikan panduan strategis berbasis bukti bagi para pengembang untuk memilih konfigurasi konsistensi yang paling optimal untuk aplikasi dengan karakteristik serupa.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan tersebut. Maka, dapat dijabarkan dalam beberapa permasalahan yang ada. Beberapa pokok permasalahan mengenai analisis performa dan efisiensi *database NoSQL* pada layanan *cloud* sebagai berikut :

Bagaimana perbandingan performa *throughput* dan *latency* serta penggunaan sumber daya (*CPU*, *RAM*, *IOWAIT*) antara mode *eventual consistency* dan *strong consistency* pada *MongoDB* dengan jenis beban kerja yang berbeda (dominan baca, dominan tulis, atau seimbang)?

1.3 Tujuan Penelitian

Berdasarkan latar belakang masalah dan rumusan masalah, maka tujuan dari penelitian ini adalah

Melakukan perbandingan performa *throughput* dan *latency* serta penggunaan sumber daya (*CPU*, *RAM*, *IOWAIT*) antara mode *eventual consistency* dan *strong consistency* pada *MongoDB* dengan jenis beban kerja yang berbeda (dominan baca, dominan tulis, atau seimbang)?

1.4 Manfaat Penelitian

Berdasarkan latar belakang masalah dan rumusan masalah, tujuan maka manfaat dari penelitian ini adalah

1. Menyediakan data *benchmark* empiris yang komprehensif mengenai *trade-off* performa pada mode *eventual consistency* dan *strong consistency* untuk kasus pada infrastruktur skala kecil.
2. Menyajikan bukti kuantitatif mengenai performa setiap mode konsistensi serta dampaknya terhadap penggunaan sumber daya server (*CPU*, *RAM*, *IOWAIT*).
3. Memberikan kontribusi pada literatur ilmiah di bidang sistem basis data terdistribusi melalui studi kasus yang mendalam dan terstruktur pada platform *MongoDB*. Penelitian ini dapat menjadi referensi dan pijakan untuk penelitian lebih lanjut yang mengeksplorasi aspek-aspek lain dari konsistensi dan resiliensi data.

1.5 Batasan Penelitian

Batasan penelitian ditetapkan agar lingkup penelitian lebih spesifik dan dapat menetapkan ekspektasi lingkup penelitian yang diharapkan. Berikut ini adalah batasan penelitian dari penelitian ini:

1. Penelitian dilakukan pada lingkungan komputasi skala kecil yang disimulasikan di *cloud AWS T.2 micro Ec2* (1 *vCPU*, 1 GB *RAM*).
2. Fokus penelitian terbatas pada basis data *MongoDB* dengan arsitektur *Replica Set* yang terdiri dari tiga *node*.

3. Pengujian menggunakan satu jenis struktur data (model profil pengguna dengan ukuran rata-rata ~648 *byte*). Hasil performa dapat berbeda untuk model data dengan ukuran atau kompleksitas yang jauh berbeda.
4. Penelitian hanya dilakukan pada satu *region* yang sama.