

ABSTRAK

Seiring dengan pesatnya pertumbuhan data, teknologi basis data *NoSQL* telah menjadi pilihan utama bagi pengembang yang membutuhkan solusi yang skalabel, fleksibel, dan efisien. Namun, keputusan dalam memilih mode konsistensi (seperti *eventual consistency* dan *strong consistency*) serta jenis beban kerja dapat memengaruhi performa sistem. Penelitian ini bertujuan untuk menganalisis perbandingan performa (*throughput* dan *latency*) serta penggunaan sumber daya (*CPU*, *RAM* dan *IOWAIT*) antara mode konsistensi *eventual consistency* dan *strong consistency* pada *MongoDB*. Pengujian dilakukan dengan mengukur dampak dari tiga jenis beban kerja yang berbeda, yaitu dominan baca, dominan tulis, dan seimbang pada variasi jumlah operasi dari 10.000 sampai 100.000. Penelitian dilakukan dengan menggunakan *benchmark YCSB (Yahoo! Cloud Serving Benchmark)* pada lingkungan *cloud AWS EC2 t2.micro* yang disimulasikan sebagai lingkungan skala kecil. Hasil pengujian menunjukkan bahwa mode *strong consistency* memberikan *throughput* yang lebih tinggi pada beban kerja dominan baca, sementara *eventual consistency* menunjukkan *throughput* yang lebih unggul pada beban kerja dominan tulis. Dari sisi *latency*, *eventual consistency* memiliki *latency* yang lebih rendah untuk beban kerja dominan tulis, sedangkan *strong consistency* memiliki *latency* yang lebih tinggi. Analisis penggunaan sumber daya menunjukkan bahwa *strong consistency* memiliki penggunaan *CPU* yang lebih tinggi dibandingkan dengan *eventual consistency* untuk semua jenis beban kerja. Hal ini disebabkan oleh adanya proses *confirm* saat data direplikasi pada *strong consistency* saat melakukan operasi *write*. Temuan ini menegaskan bahwa tidak ada

satu pun konfigurasi yang optimal untuk semua kasus. Pemilihan mode konsistensi harus disesuaikan dengan karakteristik beban kerja aplikasi untuk mencapai performa dan penggunaan yang optimal.

Kata Kunci: *Cloud Computing, Konsistensi Data, MongoDB, NoSQL*

ABSTRACT

With the rapid growth of data, NoSQL database technology has become the primary choice for developers who need scalable, flexible, and efficient solutions. However, decisions regarding consistency modes (such as eventual consistency and strong consistency) and workload types can affect system performance. This study aims to analyze the comparison of performance (throughput and latency) and resource usage (CPU, RAM, and IOWAIT) between eventual consistency and strong consistency modes in MongoDB. This study aims to analyze the performance comparison (throughput and latency) and resource usage (CPU, RAM, and IOWAIT) between eventual consistency and strong consistency modes in MongoDB. The testing was conducted by measuring the impact of three different types of workloads, namely read-dominant, write-dominant, and balanced, on a variety of operation counts from 10,000 to 100,000. The research was conducted using the YCSB benchmark (Yahoo! Cloud Serving Benchmark) on an AWS EC2 t2.micro cloud environment that was simulated as a small-scale environment. The test results show that strong consistency mode provides higher throughput for read-dominant workloads, while eventual consistency shows superior throughput for write-dominant workloads. In terms of latency, eventual consistency has lower latency for write-dominant workloads, while strong consistency has higher latency. Resource usage analysis shows that strong consistency has higher CPU usage compared to eventual consistency for all types of workloads. This is due to the confirmation process when data is replicated in strong consistency during write operations. These findings confirm that there is no single optimal configuration for

all cases. The choice of consistency mode must be tailored to the characteristics of the application workload to achieve optimal performance and usage.

Keywords: *Data Consistency, Cloud Computing, MongoDB, NoSQL,*