ABSTRAK

Basis data NoSQL banyak digunakan pada sistem terdistribusi karena kemampuannya dalam menangani data berskala besar dan menyediakan ketersediaan layanan tinggi, namun mekanisme replikasi menimbulkan konsekuensi terhadap *consistency* data dan *partition tolerance* sebagaimana dijelaskan dalam Teorema CAP. MongoDB menyediakan fitur *Replica Set* untuk mendukung replikasi dan *failover* otomatis, tetapi analisis empiris mengenai *trade-off* antara *consistency* dan *partition tolerance* pada konfigurasi tersebut masih terbatas. Penelitian ini bertujuan menganalisis *trade-off consistency* dan *partition tolerance* pada MongoDB *Replica Set* melalui eksperimen terkontrol dengan konfigurasi tiga node (satu *primary* dan dua *secondary*) pada kondisi normal dan partisi jaringan parsial. Pengujian menggunakan variasi dataset 1.000 hingga 10.000 dokumen dan setiap skenario dijalankan sebanyak 30 kali. *Consistency* diukur menggunakan *replication lag, stale reads*, dan *latensi* pengakuan tulis mayoritas (*writeConcern = majority*), sedangkan *partition tolerance* dianalisis melalui *request success rate, recovery time*, dan *rollback data*. Hasil menunjukkan bahwa peningkatan jumlah dokumen menaikkan *replication lag* dari 4,94–7,56 ms serta latensi tulis mayoritas dari 3,66–5,28 ms, dengan *stale reads* sebesar 0,16%–0,24%. Pada partisi parsial, sistem mempertahankan *request success rate* 100% dengan *recovery time* 38–44 ms tanpa *rollback* karena *quorum* mayoritas tetap terpenuhi, sehingga menunjukkan adanya *trade-off* antara latensi dan *consistency* sementara *partition tolerance* tetap terjaga selama *quorum* tersedia.

Kata kunci: *Consistency*, MongoDB, NoSQL, *Partition Tolerance*, *Replikasi Database*, Teorema CAP.

ABSTRACT

NoSQL databases are widely used in distributed systems due to their ability to handle large-scale data and provide high availability; however, replication mechanisms introduce implications for data consistency and partition tolerance as described in the CAP Theorem. MongoDB provides a Replica Set feature to support replication and automatic failover, yet empirical analysis of the trade-off between consistency and partition tolerance in this configuration remains limited. This study aims to analyze the trade-off between consistency and partition tolerance in a MongoDB Replica Set through controlled experiments using a three-node configuration (one primary and two secondary nodes) under normal conditions and partial network partition scenarios. The experiments used dataset variations ranging from 1,000 to 10,000 documents, with each scenario executed 30 times. Consistency was measured using replication lag, stale reads, and majority write acknowledgement latency (writeConcern = majority), while partition tolerance was evaluated through request success rate, recovery time, and rollback data. The results indicate that increasing the number of documents raises replication lag from 4.94–7.56 ms and majority write latency from 3.66–5.28 ms, with stale reads ranging from 0.16%–0.24%. Under partial partition conditions, the system maintained a 100% request success rate with recovery time between 38–44 ms and no rollback occurrence because majority quorum remained satisfied. These findings demonstrate a measurable trade-off between latency and consistency while partition tolerance remains preserved as majority quorum is available.

Keywords: CAP Theorem, Consistency, Database Replication, MongoDB, NoSQL, Partition Tolerance..