

INTISARI

IMPLEMENTASI DAN ANALISIS *SHARDING* DAN *REPLICATION* PADA MONGODB UNTUK MENINGKATKAN UNJUK KERJA DAN KETERSEDIAAN BASIS DATA

Masifnya pertumbuhan suatu sistem informasi yang menyimpan data secara berkala pada suatu penyimpanan, maka seiring dengan berjalannya waktu dapat mengakibatkan adanya penumpukan data pada basis data tunggal sehingga berpengaruh terhadap kinerja sistem informasi, selain daripada itu kebutuhan akan ketersediaan basis data untuk terus dapat digunakan juga berpengaruh terhadap pengolahan dan analisis data. Pada penelitian ini akan mengimplementasikan basis data terdistribusi pada MongoDB dengan mengkonfigurasi serta *replica* yang shardin bertujuan untuk meningkatkan kinerja *query* dan ketersediaan data pada *server* basis data tunggal. Pengujian kinerja dilakukan dengan membandingkan *server* basis data tunggal dan sharding terhadap *query* uji, hasil dari pengujian dengan menggunakan uji hipotesis menunjukkan bahwa sepuluh *query* uji yang dijalankan memiliki jumlah peringkat dari data waktu eksekusi bernilai kecil yang berarti waktu eksekusi *query* lebih singkat atau kinerja *server* basis data lebih baik jika dilihat dari waktu eksekusi *query*, dimana jumlah peringkat tersebut dimiliki oleh *server* dengan sharding dan *replica* yakni sebesar 5.931,5 untuk *query* sederhana dan 6.793 untuk *query* kompleks sedangkan pada arsitektur *single* menunjukkan jumlah peringkat waktu eksekusi *query* sebesar 14168,5 untuk *query* sederhana dan 13.307 untuk *query* kompleks.

Pada pengujian ketersediaan *server* basis data, dilakukan dengan membandingkan *server* MongoDB tidak terkonfigurasi *replica* dengan yang terkonfigurasi *replica*. Pada *server* yang mengkonfigurasi *replica* menunjukkan bahwa ketika *server* diberi kondisi kegagalan pada salah satu komponen shard maka *query* tidak dapat berjalan secara maksimal baik operasi *read* maupun operasi *write*, sedangkan pada *server* basis data terkonfigurasi *replica* operasi *read* dapat berjalan pada semua pemberian kondisi sebab operasi *read* dapat ditangani oleh *secondary*, namun tidak dengan *query write* sebab operasi *write* dijalankan pada *primary* sehingga hanya sebagian data yang berhasil tereksekusi.

Kata kunci: Basis data, MongoDB, *sharding*, *replication*

IMPLEMENTATION AND ANALYSIS OF SHARDING AND REPLICATION ON MONGODB TO IMPROVE PERFORMANCE AND AVAILABILITY OF DATABASE

The growth of information system that stores periodic data on storage can result in the buildup of data so that it affects the performance of information systems, and the need for database availability to continue to be used also affects the processing and analysis of data. This study will implement sharding and replica aiming to improve query performance and data availability on a single database server. Testing was used comparing the performance of a single database server and database sharding using queries, which showed that queries were executed and then doing a hypothesis test with a comparison technique have a small number of rank, which means database server have better performance by query execution time, where the number of ratings is owned by a server with sharding and replica are 5,931.5 for a simple query and 6,793 for complex queries whereas on a single architecture shows the number of query execution time ratings of 14168.5 for simple queries and 13,307 for complex queries.

Availability testing by comparing the unconfigured replica server to the replica. On the server that configures replica shows that when the Shard components were down, the query cannot run maximally either read operation or write operation, while on a configured database server that replicated the read operations can run on all given conditions because read operations can be handled by secondary than write operation that only execute on primary so for the write operation only on the second attempt where the data that wasn't finished to be executed and the third experiments for all the failure conditions can execute the data according to the needs of the query

Keywords :Database, NoSQL, MongoDB, sharding, replication.