

BAB 6

KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan dari penelitian yang sudah penulis lakukan. Penulis juga memberikan saran yang berupa saran dalam pengembangan penelitian lanjutan terhadap MongoDB.

6.1 Kesimpulan

Berdasarkan penelitian yang sudah penulis lakukan terhadap basis data MongoDB, berikut kesimpulan yang dapat diambil :

1. DBMS relasional memiliki beberapa batasan / *constraint* karena basis data relasional memiliki skema yang terstruktur. DBMS Relasional mempunyai komponen *log manager* yang berfungsi sebagai *rollback* dan *recovery* jika terjadi adanya kegagalan sistem. DBMS Relasional juga memiliki konsep partisi yang bertugas untuk membagi data menjadi beberapa bagian di tempat penyimpanan yang berbeda. DBMS Relasional menerapkan konsep *concurrency control* di tingkat basis data. Dimana saat adanya transaksi di basis data tersebut, sistem akan menerapkan konsep *locking* sehingga transaksi tidak akan di proses sampai transaksi sebelumnya selesai. Jika terjadi transaksi yang gagal di basis data tersebut,
2. Di MongoDB tidak ada batasan-batasan seperti di DBMS relasional karena data yang ada di MongoDB tidak terstruktur / *schemaless*. MongoDB juga mempunyai fitur *snapshots* dan *checkpoints* yang berfungsi saat terjadi adanya transaksi yang gagal dan sebagai *recovery*. Cara kerjanya adalah saat terjadinya proses penulisan ke *disk*, maka *storage engine* akan secara otomatis menulis transaksi tersebut kedalam sebuah *snapshot* dan bertugas sebagai sebuah *checkpoints* saat terjadi adanya kesalahan sistem. MongoDB juga memiliki konsep *sharding* yang hampir sama seperti *partitioning* di DBMS relasional. MongoDB memiliki konsep *concurrency control* yang berbeda dengan DBMS Relasional. MongoDB menerapkan konsep *locking* di tingkat dokumen. Jika terjadi adanya transaksi yang sama di dokumen yang sama, maka akan dilakukan *locking* di dokumen tersebut. Sehingga saat ada transaksi di dokumen yang berbeda dan *collection* yang sama, MongoDB akan berjalan seperti normal tanpa adanya *locking*.
3. Pada skripsi ini sudah dapat dibangun perangkat lunak untuk melakukan demo terhadap fitur-fitur yang disediakan oleh MongoDB dengan menggunakan *driver* NodeJS yang sudah disediakan oleh MongoDB pada bab 5. MongoDB Atlas menyediakan beberapa *driver* lain dengan berbagai bahasa pemrograman yang bertujuan untuk mempermudah interaksi antara aplikasi *client* dengan basis data MongoDB. Pembangunan koneksi dengan MongoDB Atlas dapat dilakukan melalui CLI(*Command Line Interface*) dan melalui *Web Interface* yang disediakan oleh MongoDB Atlas yang bertujuan agar pengguna dapat melakukan kontrol terhadap *cloud* yang mereka miliki. Transaksi yang ada di MongoDB Atlas dapat dilakukan melalui CLI dan *driver* yang sudah disediakan oleh MongoDB sendiri.
4. Berdasarkan hasil pengujian performa MongoDB yang sudah dilakukan, operasi FIND dengan menggunakan indeks memiliki waktu yang sangat cepat dibandingkan dengan operasi FIND

tanpa menggunakan indeks. Untuk operasi DELETE dan INSERT perbedaan waktu yang ada bergerak secara linear menaik sesuai dengan ukuran data yang diuji. Berdasarkan hasil pengujian performa antara MongoDB dan MySQL yang sudah dilakukan, operasi SELECT, INSERT memiliki waktu tercepat di pengujian dengan menggunakan MongoDB. Sedangkan operasi DELETE diungguli oleh MySQL.

6.2 Saran

Berdasarkan eksperimen dan kesimpulan diatas, berikut adalah saran yang dapat diajukan oleh penulis:

1. Dapat dilakukan penelitian dengan terhadap MongoDB Atlas dengan menggunakan *driver* yang lainnya.
2. Untuk mendapatkan waktu yang lebih cepat dapat menggunakan data yang dapat dibaca dalam bentuk *array of JSON*.
3. Membandingkan MongoDB Atlas dengan DBMS berbasis *cloud* lainnya.
4. Menggunakan *cloud* dengan RAM dan CPU yang lebih besar.

DAFTAR REFERENSI

- [1] Ramakrishnan, R. dan Gehrke, J. (2000) *Database management systems*. McGraw Hill.
- [2] Elmasri, R. (2008) *Fundamentals of database systems*. Pearson Education India.
- [3] Hellerstein, J. M., Stonebraker, M., Hamilton, J., dkk. (2007) Architecture of a database system. *Foundations and Trends® in Databases*, **1**, 141–259.
- [4] Silberschatz, A., Korth, H. F., dan Sudarshan, S. (2010) *Database System Concepts*, 6th edition. McGraw Hill, USA.
- [5] MySQL (2019) Mysql 5.7 reference manual. <https://dev.mysql.com/doc/refman/5.7/en/>. 25 November 2019.
- [6] Saxenna, M. dan Singh, V. (2014) Nosql databases- analysis, techniques, and classification. *Journal of Advanced Database Management Systems (EISSN: 2393-8730)*, **1**, 1–11.
- [7] Sadalage, P. J. dan Fowler, M. (2013) *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.
- [8] MongoDB (2019) The mongodb 4.2 manual. <https://docs.mongodb.com/manual>. 21 November 2019.
- [9] Alam, B., Doja, M., Alam, M., dan Mongia, S. (2013) 5-layered architecture of cloud database management system. *AASRI Procedia*, **5**, 194–199.
- [10] MongoDB (2019) Mongodb architecture guide. https://jira.mongodb.org/secure/attachment/112939/MongoDB_Architecture_Guide.pdf. 21 November 2019.
- [11] Kaggle (2019) Kaggle yelp dataset. <https://www.kaggle.com/yelp-dataset/yelp-dataset>. 21 November 2019.