

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Munculnya web 2.0 merupakan awal dari pertumbuhan data yang sangat pesat besar dan tidak terstruktur. Dimana penggunaan web konten seperti wiki, blog, video *sharing*, dan *social network* [8], mulai ramai digunakan oleh pengguna internet diseluruh dunia. Dalam hal ini melakukan proses manipulasi menjadi sangat rumit karena basisdata relasional atau SQL tidak lagi efisien dalam menangani data yang besar dan tidak terstruktur. Database relasional dapat digunakan dengan baik ketika dihadapkan pada data dengan jumlah yang memiliki batas tertentu serta memiliki struktur yang konstan [2]. Keterbatasan ini membuat para developer mencoba melakukan manipulasi data menggunakan teknologi baru yakni NoSQL.

Basisdata non-relasional merupakan teknologi yang dapat digunakan dalam melakukan proses manipulasi terhadap data yang besar dan tidak terstruktur. Migrasi ini perlu dilakukan oleh developer mengingat penggunaan database yang optimal harus sesuai dengan kebutuhan, sehingga ketika melakukan manipulasi data tidak terhambat oleh waktu eksekusi yang lama. Penggunaan basisdata non-relasional juga memiliki cara tersendiri dalam melakukan proses manipulasi data, mengingat bahwa basisdata non-relasional memiliki banyak tipe berdasarkan jenisnya, diantaranya *Key-Value based*, *Colomn based*, *Document based*, dan *Graph based* [10]. Pada NoSQL juga dikenal tidak memiliki skema yang baku [5].

Melakukan manipulasi merupakan hal terpenting ketika berhadapan dengan database. Pada NoSQL, terdapat cara yang berbeda-beda dalam melakukan manipulasi data baik dari segi query dan database manajemen sistem yang digunakan. ketika developer dihadapkan dengan format data yang berbeda-beda, dan para developer telah akrab dengan query SQL beserta database manajemen sistem yang digunakan dalam memanipulasi data maka melakukan manipulasi data pada data yang tidak terstruktur seperti era saat ini menjadi permasalahan tersendiri, hal ini membuat para developer harus belajar memahami penggunaan query yang digunakan dalam melakukan manipulasi data tersebut.

Melihat permasalahan diatas, pada tugas akhir ini diberikan solusi berupa sistem database management yang mampu menerima inputan berupa query SQL dan dieksekusi pada database penyimpanan NoSQL, penggunaan integrasi [12] dilakukan pada layer query, sehingga ketika peralihan database ke NoSQL penggunaan query menjadi optimal ketika melakukan manipulasi data.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, terdapat beberapa pertanyaan yang muncul, antara lain :

1. Bagaimana *mapping* query dalam melakukan integrasi SQL ke NoSQL khususnya MongoDB ?
2. Bagaimana melakukan parsing, validasi, serta translasi pada sistem sehingga dapat diterima dan diproses oleh sistem ?
3. Bagaimana waktu yang dibutuhkan dalam melakukan proses parsing dan translasi serta kesesuaian input query yang diberikan terhadap output yang dihasilkan oleh sistem?

1.3. Tujuan

Berdasarkan latar belakang yang telah dijelaskan diatas, terdapat beberapa tujuan yang dicapai pada pengerjaan tugas akhir ini, antara lain :

1. Mengetahui mapping query yang digunakan dalam melakukan integrasi.
2. Mengetahui proses parsing dan translasi yang digunakan dalam proses memanipulasi yang dilakukan oleh sistem.
3. Mengetahui lama waktu yang diperlukan sistem dalam melakukan proses parsing dan translasi query beserta ketepatan hasil yang ditampilkan oleh sistem.

1.4. Batasan Masalah

Batasan masalah pada pengerjaan tugas akhir ini antara lain :

1. Sistem database yang digunakan berupa sistem NoSQL *Document-Oriented* yakni MongoDB.
2. Sistem hanya dapat menerima perintah SQL antara lain Select, Distinct, Update, Left Join, Insert, dan Delete.
3. Sistem hanya dapat melakukan *left outer join*.
4. Klausa distinct hanya dapat men-*select* 1 kolom.
5. Sistem tidak mampu mengeksekusi perintah dari jenis Data Control Language (DCL), dan jenis Data Definition Language (DDL). Hal ini dikarenakan NoSQL memiliki prinsip ACID tersendiri. Selain itu NoSQL juga tidak memerlukan pendefinisian diawal.
6. Operator berupa penjumlahan, pengurangan, pembagian, rata-rata serta Group by tidak diterapkan pada sistem.

1.5. Metodologi Penyelesaian Masalah

Adapun metode penyelesaian yang dilakukan untuk menyelesaikan tugas akhir ini yaitu :

1. Studi Literatur

Pada tahap ini, penulis melakukan pendalaman materi tentang segala sesuatu yang berhubungan dengan teori yang digunakan untuk penyelesaian tugas akhir ini sesuai dengan apa yang diharapkan. Studi

literatur dilakukan secara *online* dan *offline* melalui buku, jurnal, diskusi dengan teman yang memiliki satu bidang riset yang sama dan bertanya langsung kepada dosen terkait.

2. Analisis kebutuhan sistem

Dilakukan analisa terhadap kebutuhan sistem yang mengacu pada tujuan yang ingin dicapai dari penelitian ini.

3. Perancangan sistem

Perancangan sistem dilakukan setelah kebutuhan sistem terdefiniskan. Pada tahap ini ditentukan query inputan berupa query SQL, eksekusi query pada layer client yang ada di windows, kemudian query dieksekusi pada layer server virtualisasi di linux.

4. Implementasi sistem

Implementasi sistem dilakukan setelah tahap perancangan dan desain sistem diperoleh. Implementasi dilakukan dalam bentuk aplikasi yang memiliki *interface* yang dapat menerima inputan berupa query SQL, apabila query valid kemudian dilakukan proses parsing query SQL tersebut, dan melakukan translasi kedalam query non-relasional, tahap ini dilakukan pada layer client. Pada layer server, akan menerima request dalam bentuk query non-relasional, kemudian dieksekusi dan melemparkan hasil *request*. Layer client akan menerima hasil *request* dan menampilkannya.

5. Pengujian Sistem

Pengujian sistem dilakukan menggunakan beberapa jenis *statement* query SQL berupa Select, Insert, Update, serta Delete

6. Analisis dan Kesimpulan

Setelah melakukan pengujian system, keluaran yang diharapkan sesuai dengan *request* query yang diinputkan pada system, kemudian hasil dianalisis berdasarkan *respon*time yang dihasilkan ketika awal penginputan query hingga memberikan hasil.