

1. Pendahuluan

1.1 Latar Belakang

Perkembangan *social network* pada situs *social networking* semakin meningkat karena kemampuannya dalam membantu membangun relasi antar individu di seluruh dunia. *Facebook & twitter*, adalah beberapa contoh situs *social networking* yang umum digunakan untuk bersosialisasi di dunia maya. Tentu saja, data relasi tiap orang (pengguna) yang ada pada situs tersebut sangat banyak dan struktur relasi setiap orang yang terlibat sangat kompleks. Selain itu, data ini akan terus diakses oleh pengguna dan juga dilakukan modifikasi seperti penambahan dan penghapusan data, atau perubahan struktur relasi yang ada. Karena itu, diperlukan suatu sistem penyimpanan data agar data tersebut dapat diorganisir, mudah diakses dan integritas data terus terjaga. Sistem penyimpanan data ini disebut sebagai *database*.

Data relasi pada *social network* juga memberikan peluang untuk ekstraksi informasi berdasarkan hubungan yang terjadi pada antar individu. Namun, data yang disimpan dalam *database* konvensional belum cukup handal untuk ekstraksi informasi dari beberapa data relasi yang ada. Hal ini disebabkan karena data relasi *social network* tidak memiliki struktur yang tetap dan dapat berubah sewaktu-waktu. Misalnya, pada kasus seperti mencari orang yang paling dekat sebagai perantara untuk beberapa orang dalam *social network* memerlukan *query* yang kompleks dan penjelajahan relasi yang lama.

Salah satu solusi adalah memodelkan data tersebut dalam bentuk *graph*. *Graph* adalah struktur untuk memodelkan relasi antar objek dari suatu koleksi data kedalam bentuk *node (vertex)* dan *edge*. *Node* merepresentasikan objek atau entitas dan relasi merepresentasikan hubungan antar *node*. Prinsip ini sesuai dengan model data pada *social networking*, yang berisi daftar orang beserta relasinya terhadap orang lain. Agar data pada *social network* dapat direpresentasikan dalam *graph* dan disimpan dalam *database*, salah satu solusi adalah menggunakan *graph database*.

Graph database adalah salah satu metode implementasi dari NoSQL (*Not Only SQL*), yaitu sistem basisdata yang berguna untuk penyimpanan data dalam jumlah besar dan direpresentasikan dalam bentuk *graph*, sehingga data memiliki aksesibilitas yang tinggi. *Graph database* memodelkan data kedalam bentuk *node*, *edge* dan *property*. *Node* merepresentasikan objek dan *edge* merepresentasikan relasi yang ada pada tiap *node* tersebut. Setiap *node* dan *edge* dapat memiliki suatu *property* yang berisi informasi karakteristik masing-masing *node* dan *edge*. Data *social network* memiliki struktur mirip seperti *graph* yang terdiri dari *node* dan *edge*. *Node* dan *edge* ini dapat dibentuk kedalam *graph database* dan dapat diuji dan dianalisis hasil performansinya berdasarkan parameter *response time* dan *throughput*.

1.2 Perumusan Masalah

Berikut ini adalah permasalahan yang didefinisikan kedalam beberapa pertanyaan:

1. Bagaimana proses implementasi *graph database* menggunakan data *social network*.
2. Bagaimana proses analisis performansi dari *graph database* berdasarkan parameter *query response time* dan *throughput*.

1.3 Tujuan

Berikut ini adalah tujuan dari tugas akhir:

1. Melakukan implementasi *graph database* menggunakan data *social network*.
2. Melakukan analisis terhadap hasil *query response time* dan *throughput* untuk mengetahui performansi dari *graph database*.

1.4 Batasan Masalah

Berikut ini adalah batasan masalah pada Tugas Akhir ini:

1. Data *social network* yang digunakan adalah data *user* dan data *posting* pada iFace IT Telkom.
2. Menggunakan Neo4j sebagai *tools* untuk membangun *graph database*.
3. Hasil dari *query response time* dan *throughput* hanya berdasarkan pada set *query* yang sudah ditentukan.
4. Perbandingan eksekusi *query* yang dilakukan menggunakan Neo4j JAVA Core API dan Cypher Query Language.
5. Pembuatan visualisasi *graph database* tidak termasuk dalam proses implementasi *graph database*.
6. Bentuk *graph database* adalah *graph* berarah (*directed graph*).

1.5 Metode Penyelesaian Masalah

Adapun metode yang dilakukan untuk menyelesaikan masalah dalam tugas akhir ini adalah:

1. Studi Literatur
Melakukan pencarian dan pembelajaran dari sumber kajian, penelitian-penelitian, literatur dan konsep yang terkait dengan masalah pada Tugas Akhir ini. Semua referensi dan jurnal ilmiah dapat ditemukan di Internet serta buku referensi yang berkaitan dengan masalah pada Tugas Akhir ini. Untuk mendukung pemahaman terhadap masalah pada Tugas Akhir ini, penulis juga mempelajari konsep Basis Data Relasional saat perkuliahan dan mempelajari NOSQL serta konsep *Graph Database* secara mandiri melalui media Internet. Penulis juga mempelajari *tools* untuk implementasi *graph database*, yaitu Neo4j.

2. Pengumpulan dan Pengolahan Data
Mencari studi kasus berbasis *social networking* dan data yang dapat digunakan untuk diimplementasikan ke dalam *graph database*.
3. Desain Sistem
Pada tahap ini, akan dilakukan pendefinisian terhadap masalah serta solusinya. Kemudian dilakukan analisis untuk pemodelan sistem dan merumuskan langkah-langkah implementasi sistem.
4. Implementasi
Melakukan implementasi *graph database* sesuai dengan studi kasus yang ada dan menggunakan *tools* yang telah dideskripsikan sebelumnya.
5. Kesimpulan dan Laporan Hasil Analisis
Melakukan penarikan kesimpulan serta pelaporan hasil analisis berdasarkan *query response time*, *database* yang dibangun, dan fakta yang diperoleh selama penelitian berlangsung.