

# Bab I

## Pendahuluan

### 1.1 Latar belakang

*Relational database* adalah sebuah *database* yang berdasarkan pada model *relational* telah mendominasi dunia penyimpanan dan pengambilan data sejak dikembangkan oleh Edgar Codd pada tahun 1970[15]. Banyak aplikasi yang membutuhkan *relational database* sebagai media penyimpanan dan pengambilan data. *Social network* adalah salah satu aplikasi yang membutuhkan *relational database* dan salah satu contohnya adalah *TagShout*. *Social network* dapat bekerja dengan baik menggunakan *relational database* ketika jumlah data dan jumlah *relationships* relatif kecil (*Single Join*). Namun ketika jumlah data dan jumlah *relationships* semakin banyak dan kompleks, *social network* menjadi semakin sulit untuk bekerja dan dimodelkan dengan *relational database* karena membutuhkan operasi *Join* yang sangat banyak dimana operasi *Join* akan melakukan perhitungan *cartesian product*. Disamping itu, *social network* yang bekerja dengan menggunakan *relational database* menyimpan data dalam bentuk *tabular* dan skema yang kaku yang sangat tidak cocok dengan model *social network* yang *interconnected data*.

Salah satu solusi untuk menangani masalah tersebut adalah menggunakan *database NoSQL*. *NoSQL* adalah kepanjangan dari *Not Only SQL* yaitu *database non-relational* yang berbeda jauh dari sistem *relational database*, seperti: tidak menggunakan tabel, bahasa *query SQL* dan operasi *Join*. Perkembangan *NoSQL* ini bukan untuk menggantikan sistem penyimpanan data lama seperti *relational database*, tetapi digunakan sebagai model data alternatif untuk aplikasi yang tidak dapat bekerja dengan baik menggunakan model *database relational*. *NoSQL* memiliki empat metode, yaitu *Table-Oriented*, *Key-Value Stores*, *Document Oriented* dan *Graph Oriented*[6]. Tetapi metode yang cocok untuk masalah tersebut adalah metode *Graph Oriented* atau *Graph Database*. Karena *graph database* memodelkan data dalam bentuk *graph* yang sama dengan model *social network* yang berbentuk *interconnected data* dimana data disimpan dalam bentuk *node*, *relationship* dan *property* sehingga lebih efektif untuk memodelkan *social network* dibandingkan dengan metode yang lain.

Oleh sebab itu, dilakukanlah suatu transformasi dari *relational database* ke *graph database*. *Graph database* merupakan sebuah sistem manajemen *database online* dengan metode *Create, Read, Update* dan *Delete (CRUD)* yang mengekspos sebuah model data *graph* dimana entitas-entitas digambarkan sebagai *nodes* dan hubungan-hubungan antara entitas-entitas digambarkan sebagai *relationships*[10].

## 1.2 Perumusan masalah

Permasalahan yang diangkat dalam Tugas Akhir ini adalah:

1. Bagaimana transformasi skema *relational database* ke skema *graph database* pada *website social network TagShout*?
2. Bagaimana performansi *website social network TagShout* setelah ditransformasikan dari *relational database* ke *graph database* berdasarkan parameter *query response time* dan *memory usage*?

## 1.3 Tujuan

Tujuan yang ingin dicapai dari Tugas Akhir ini adalah:

1. Melakukan transformasi skema *relational database* ke skema *graph database* pada *website social network TagShout*.
2. Melakukan analisis terhadap hasil *query response time* dan *memory usage* untuk mengetahui performansi *website social network TagShout* setelah ditransformasikan dari *relational database* ke *graph database*.

## 1.4 Batasan Masalah

Adapun batasan masalah dalam Tugas Akhir ini adalah:

1. *Social network* yang digunakan sebagai data uji adalah *TagShout* ([www.tagshout.com](http://www.tagshout.com)).
2. Bentuk *graph database* adalah *graph* berarah (*directed graph*).
3. *DBMS* yang digunakan adalah *MySQL* pada *relational database* dan *Neo4j* pada *graph database*.
4. Fokus dari tugas akhir ini adalah analisis terhadap transformasi skema *relational database* ke skema *graph database* pada *website social network TagShout*.
5. Data yang digunakan adalah data *dummy* atau data yang dibuat secara *random* atau acak.

## 1.5 Hipotesa

Dalam *graph database*, proses penambahan informasi baru tidak membutuhkan proses strukturisasi kembali skema *graph* itu sendiri, melainkan hanya menambahkan *node* dan *relationship* saja ke dalam *graph* yang telah dibuat sebelumnya. *Graph database* merupakan skema yang tidak tetap dimana struktur skemanya dapat berubah-ubah dengan mudah dan dalam proses pencarian data, *graph database* tidak mencari data secara keseluruhan, melainkan hanya mencari data yang sesuai dengan *query* yang diberikan[2].

Menurut Partner dan Vucotic, proses pencarian *friends of friends* dalam *social network* yang berisi 1.000.000 *user* dan satu *user* rata-rata memiliki 50 teman, hanya bisa dilakukan hingga kedalaman kelima pada *relational database*. Sebaliknya, *graph database* masih bisa bekerja 150 kali lebih cepat dibandingkan

dengan *relational database* pada kedalaman ketiga dan 1000 kali lebih cepat pada kedalaman keempat[10].

*Social network TagShout* dengan menggunakan *graph database* dapat bekerja hingga 150 kali lebih cepat dibandingkan dengan *social network TagShout* dengan menggunakan *relational database* pada kedalaman ketiga dan hingga 1000 kali lebih cepat pada kedalaman keempat dengan menampung hingga 100000 *user* dan dapat melakukan proses pencarian *friends-of-friends* hingga kedalaman kelima.

## 1.6 Metodologi Penyelesaian Masalah

Metodologi penyelesaian masalah terbagi dalam beberapa tahapan, yaitu:

1. Studi literatur  
Mencari dan mempelajari referensi yang berhubungan dengan konsep *social network*, *relational database*, *NoSQL*, konsep *graph* dan metode *graph database*.
2. Pengumpulan dan Pengolahan Data  
Pengambilan data dari jejaring sosial *TagShout* dengan menggunakan perangkat lunak *MySQL* and *Neo4j*.
3. Desain Sistem  
*Input* sistem ini berupa tahap pengumpulan data. *Output* sistem adalah hasil *execution time* dan *records* yang dikembalikan dari tiap *database* yang digunakan dalam jejaring sosial *TagShout*.
4. Implementasi  
Mengimplementasikan metode *graph database* untuk mengetahui *execution time* dan *records* yang dikembalikan yang berpengaruh dalam jejaring sosial *TagShout*.
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.

## 1.7 Sistematika Penulisan

Adapun sistematika penulisan Tugas Akhir ini adalah sebagai berikut:

1. BAB 1 Pendahuluan  
Pada Bab 1 diuraikan isi dan rencana pengerjaan Tugas Akhir secara keseluruhan yang meliputi latar belakang, perumusan masalah, batasan masalah, tujuan, hipotesa dan metode penyelesaian masalah yang diterapkan.
2. BAB 2 Landasan Teori  
Bab 2 memaparkan dasar-dasar teori yang berkaitan dengan *social network*, *relational database*, *NoSQL*, teori *graph*, *graph database* dan *neo4j*.
3. BAB 3 Perancangan dan Implementasi  
Perancangan sistem dan contoh implementasi dari sistem yang dibangun dipaparkan pada bab ini. Selanjutnya dilakukan proses implementasi.
4. BAB 4 Pengujian dan Analisis  
Pada bab ini dibahas skenario dan hasil pengujian yang dilakukan pada hasil implementasi sistem.
5. BAB 5 Kesimpulan dan Saran  
Bab ini berisi kesimpulan dan saran yang didapatkan dari hasil implementasi sistem secara keseluruhan.