BAB 1
PENDAHULUAN

1.1. Latar Belakang

Sulitnya membuat desain *database* dengan model *relational* yang pas, meningkatnya jenis data yang semi terstruktur, dan meningkatnya waktu proses *query* jika jumlah relasi semakin banyak mendorong orang-orang melakukan penelitian untuk menemukan model *database* yang baru. Salah satu model yang ditemukan dalam penelitian tersebut adalah model *graph*. Peneliti menemukan bahwa *graph database* dapat menjadi solusi terhadap permasalahan tersebut [7].

Delapan tahun belakangan ini *graph database* mulai sering digunakan dalam berbagai bidang industri seperti, kesehatan, media, minyak dan gas, *gaming*, dll karena kelebihannya. Menurut prediksi peneliti, graph akan umum digunakan 15 tahun mendatang [7].

Namun, Masalah krusial dan sering terjadi dalam aplikasi yang berbasiskan *graph* khususnya *graph database* adalah efisiensi proses *retrieve* data. Proses *retrieve* tidak efisien jika dilakukan secara *sequential* karena proses tidak hanya dilakukan dengan menelusuri seluruh *graph database* tetapi juga melakukan proses *matching subgraph* q dengan *subgraph* yang ada di *graph database* [12]. Oleh karena itu, diperlukan suatu metode untuk meningkatkan efisiensi dalam proses *retrieve*, yaitu *indexing*.

Pada tugas akhir ini, metode *indexing* yang akan digunakan adalah *graph decomposition index*. Metode ini dipilih karena metode ini memiliki keunggulan dalam me-*retrieve* data *subgraph*, sehingga metode ini cocok dengan data yang digunakan yaitu data perpustakaan Telkom University yang jika dimodelkan menjadi *graph model*, proses *retrieve* datanya berupa *subgraph*. Selain itu data perpustakaan Telkom University juga memenuhi batasan dari metode ini[1]**.**

1.2. Perumusan Masalah

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

1. Bagaimana menerapkan *graph decomposition index* pada *graph database*?
2. Apakah sistem *graph database* dengan *index* memiliki *response time* yang lebih baik dari sistem *graph database* tanpa index?.

1.3. Tujuan

Berdasarkan latar belakang dan perumusan masalah yang telah dipaparkan diperolehlah beberapa tujuan dari tugas akhir ini, antara lain :

1. untuk mengetahui cara penerapan *graph decomposition index* pada *graph database* dan
2. untuk mengetahui *response time* sistem mana yang lebih baik antara sistem *graph database* dengan *index* atau tanpa *index*.

1.4. Batasan Masalah

Batasan masalah pada TA ini antara lain :

1. Algoritma *graph matching* yang digunakan adalah *algoritma graph matching* graphQL[2],
2. Maksimal jumlah *vertex* dalam satu *graph* yaitu dua belas,
3. Graph berupa *simple undirect graph* berlabel,
4. Menggunakan bahasa pemrograman java,
5. Menggunakan mekanisme penyimpanan *file* secara *serializable* yang ada pada java.

1.5. Metodologi Penyelesaian Masalah

Adapun metode penyelesaian yang akan dilakukan untuk penyelesaian tugas akhir ini, yaitu :

1. Studi Literatur
   Pada tahap ini, penulis belajar dan mencari tahu segala informasi yang dibutuhkan untuk penyelesaian tugas akhir ini, seperti konsepnya. Penulis mendapatkan informasi tersebut melalui buku, jurnal, diskusi dengan dosen terkait, atau tugas akhir yang sudah pernah ada yang di lakukan secara *offline* maupun *online*. Beberapa ilmu yang dibutuhkan untuk penyelesaian tugas akhir ini didapatkan dari perkuliahan formal dan belajar mandiri.
2. Pengumpulan Data dan Pengolahan Data
   Penulis mengumpulkan data studi kasus yaitu *database* perpustakaan Telkom University kemudian data tersebut akan dianalisis.
3. Pembangunan Model
   Hasil dari tahap analisis berupa jumlah entitas dan jumlah relasi yang terdapat pada data yang digunkan kemudian ditransformasikan dari model *relational* menjadi model *graph*.
4. Implementasi Model
   Model yang telah dibuat kemudian diimplementasikan menjadi sebuah sistem *graph database* yang menyimpan data-data studi kasus.
5. Pembangunan *index*
   Pembangunan *index* untuk memperkecil ruang pencarian berdasarkan data-data studi kasus.
6. Analisis dan Kesimpulan
   Sistem yang telah dibangun kemudian dianalisis. Hal-hal yang dianalisis antara lain, *respone time* pada *query* size yang berbeda-beda. Dari data yang diperoleh tersebut kemudian dianalisis untuk menarik sebuah kesimpulan.

Uraian metodologi penyelesaian masalah dapat berupa variabel-variabel dalam penelitian, model yang digunakan, rancangan penelitian, teknik pengumpulan data dan analisis data, cara penafsiran dan penyimpulan hasil penelitian.