1 PENDAHULUAN

1.1 Latar Belakang

Rodex Tours & Travel Surabaya merupakan perusahaan yang menyediakan jasa layanan *tour* dan *travel* baik dalam maupun luar negeri. Layanan *tour* dan *travel* ini didukung dengan adanya platform *website* untuk pemesanan tiket pesawat dan hotel. Untuk pemesanan hotel, *customer* perlu memasukkan nama hotel, durasi menginap, dan jumlah orang yang menginap terlebih dahulu. Dari nama hotel yang dimasukkan akan diberikan *suggestion* yang harus dipilih oleh *customer* agar sistem dapat melakukan pencarian terhadap 1 hotel tersebut secara spesifik. Hasil pencarian yang ditampilkan adalah detail hotel yang dimaksud dengan beberapa pilihan harga yang tersedia. Adapun pilihan harga yang tersedia ini berasal dari beberapa *vendor* yang berbeda. *Vendor-vendor* inilah yang memberikan data detail pemesanan seperti ruangan dan layanan tambahan seperti *breakfast* kepada *travel agent*. Hal ini memungkinkan adanya beberapa pilihan kamar dan layanan yang sama namun memiliki harga yang berbeda karena data tersebut berasal dari *vendor* yang berbeda. Nantinya setelah memilih hotel dan *room* yang diinginkan, *customer* akan diarahkan untuk mengisi nama tamu dan *booking contact* dan melakukan pembayaran. Data pemesanan *customer* akan dikirimkan kembali kepada *vendor* terpilih untuk diteruskan dan diproses kepada hotel yang bersangkutan.

Data hotel dari *vendor-vendor* ini sangat banyak dan beragam, sehingga memungkinkan adanya identitas hotel yang berbeda-beda namun sebenarnya merujuk ke 1 entitas hotel yang sama. Misalnya *vendor* A mempunyai data hotel bernama Ibis Budget Singapore on Bencoolen, sedangkan data dari *vendor* B bernama Ibis Budget Singapore Bencoleen, dan data *vendor* C bernama Ibis Budget @ Singapore Bencoleen. Ketiga nama tersebut berbeda-beda namun sebenarnya merujuk ke 1 hotel yang sama. Karena itu, diperlukan proses *mapping* terhadap data *vendor* dengan data *master* dari Rodex Tours and Travel agar pencarian hotel yang dilakukan oleh *customer* hanya mengeluarkan 1 hasil yang tepat.

Apabila proses *mapping* ini gagal, maka terdapat 3 kemungkinan yang dapat terjadi, yaitu proses pencarian hotel mengeluarkan hasil lebih dari 1 hotel, tidak mengeluarkan hasil sama sekali, atau terdapat 1 hasil pencarian tetapi data hotel tersebut ter-*mapping* dengan data hotel lain. Misalnya data Ibis Budget Singapore on Bencoolen *vendor* A yang merujuk pada Ibis Budget Bencoolen ter-*mapping* di Ibis Budget Bencoolen Novena, maka data Ibis Budget Bencoolen itu akan muncul juga dalam hasil pencarian Ibis Budget Bencoolen Novena. Kesalahan *mapping* hotel dapat terjadi karena setiap hotel tidak memiliki kode khusus seperti bandara,

misalnya CGK untuk Bandara Internasional Soekarno Hatta dan SUB untuk Bandara Internasional Juanda. Hal ini tentunya akan menghambat proses pemesanan yang dilakukan oleh *customer* hingga dapat merugikan perusahaan.

Saat ini Rodex Tours & Travel menggunakan *relational database* yaitu PostgreSQL. Proses *mapping* dilakukan dengan cara membandingkan data *vendor* terhadap data *master,* lalu mencari skor kemiripan antar *string* dari kedua data tersebut. Pengecekan dilakukan kata per kata dan kemudian akan dihitung persentase kemiripan dibandingkan dengan keseluruhan kalimat. Semakin tinggi *skor* yang diperoleh, berarti semakin tinggi pula tingkat kemiripan antar atributnya. Data *master* yang memiliki skor kemiripan paling tinggi akan menjadi target *mapping* dari data *vendor*. Proses mapping ini juga mencakup proses *update data*, di mana atribut-atribut dari data *vendor* digabungkan dengan data *master* untuk memberikan informasi yang lebih detail. Namun saat ini untuk proses *update* hanya dilakukan berdasarkan pengecekan *string* mana yang jumlahnya lebih panjang. Untuk validasi hasil *mapping* dan *update* yang diperoleh, dilakukan pengecekan manual untuk memastikan ketepatan hasil *mapping*.

Metode *mapping* dan *update* yang saat ini diterapkan memiliki permasalahan tertentu. Salah satunya adalah apabila terdapat hotel yang berganti nama, maka akan dianggap sebagai hotel yang berbeda karena tidak ada kesamaan sama sekali antara nama hotel yang baru dengan nama hotel yang lama. Selain itu, proses yang dilakukan cukup rumit karena perlu membuat 1 *record* baru yang bertujuan untuk mencatat tingkat kemiripan antar data *vendor* dan *master*, dan apabila terdapat data yang akan di-*update*, harus ditambahkan lagi 1 *record* untuk menyimpan data lama yang akan dihapus. Selain itu, jumlah data yang ada sangatlah besar dan tentunya perusahaan akan terus menerima data baru dari *vendor* yang harus selalu di-*mapping* dan di-*update*. Hal ini juga menimbulkan masalah terhadap *relational database* yang saat ini digunakan oleh Rodex Tours & Travel. *Relational database* memiliki kelemahan dalam menangani volume data dalam jumlah besar dan modifikasi *schema* berulang (Sahatqija et al., 2018). Sedangkan kebutuhan data-data hotel ini akan terus meningkat, dan perlu dilakukan penambahan *record* baru setiap kali akan melakukan *mapping* dan *update* data.

Salah satu solusi untuk masalah *mapping* tersebut adalah memanfaatkan struktur *graph* yang memungkinkan adanya *relationship* antar *node*. Penentuan target *mapping* dilakukan dengan mencari *node* yang paling mirip dengan *field-field* tertentu dari *input* data, misalnya nama hotel, alamat, dan kontak. Kemudian dari hasil pencarian tersebut dicari apakah terdapat *node-node* yang saling berelasi dan membentuk *subgraph*. Semakin besar *subgraph* yang terbentuk mengindikasikan semakin sesuai *input* dengan *node* yang ada. Hal ini dilakukan agar

pengecekan tidak hanya bergantung pada nilai kemiripan *string*, tetapi juga memperhitungkan *node* yang saling berkaitan.

Terdapat beberapa penelitian terdahulu mengenai penerapan *graph database* yang berfokus pada pemanfaatan *relationship* untuk analisis data. Pourhabibi et al. melakukan studi terhadap beberapa penelitian terdahulu yang menerapkan cara serupa, yaitu memanfaatkan *relationship* antar *node* untuk deteksi *fraud* atau penipuan dengan menggunakan Graph-based Anomaly Detection (Pourhabibi et al., 2020). Penelitian ini memanfaatkan relasi interdependensi dan konektivitas antar data untuk mengidentifikasi pola-pola yang tidak umum. Disebutkan bahwa penelitian terdahulu hanya menganalisis perilaku pengguna dalam jaringan sosial, berfokus hanya pada 1 jenis aktivitas saja. Namun mengidentifikasi aspek, aktivitas, dan relasi yang berbeda dari 1 individu yang sama dapat memberikan petunjuk terhadap aktivitas yang perlu diwaspadai yang merujuk pada penipuan.

Penelitian lain oleh Buktar et al. menerapkan graph database dengan menggunakan Neo4j untuk menganalisis dan mengidentifikasi relasi-relasi yang ada dari data sektor manufaktur (Buktar et al., 2023). Data dan relasi-relasi ini akan dianalisis dan menghasilkan informasi untuk mendukung pengambilan keputusan organisasi. Graph database diaplikasikan untuk menganalisis proses manufaktur (anggaran, proses, pengembangan produk), supply chain management (data pemasok, logistik, dan inventori), customer relationship management (data pelanggan, hubungan antar pelanggan, data produk), serta jaminan (data pelanggan, pemasok, dan dealer). Untuk menganalisis pemasok, penulis memanfaatkan node similarity untuk menemukan pemasok yang menyediakan produk serupa. Untuk logistik, penulis memanfaatkan path finding algorithm dengan mempertimbangkan property yang dimiliki oleh node seperti biaya transportasi, waktu pengiriman, nilai produk, dan lain-lain. Dengan begitu, penggunaan graph database dapat disesuaikan dengan kebutuhan, misalnya apakah membutuhkan rute dengan biaya paling sedikit atau waktu pengiriman paling cepat.

Selain itu terdapat juga penelitian oleh Srivastava & Singh menggunakan Neo4j untuk menemukan hubungan antar data yang bocor dalam kasus Panama Paper (Srivastava & Singh, 2018). Kasus ini adalah kasus kebocoran 11,5 juta data yang belum pernah terjadi sebelumnya dari *database* firma hukum Mossack Fonseca. Dalam penelitian ini, penulis menggunakan algoritma pendeteksi penipuan yang mencari *node-node* yang memiliki *property* tertentu. Misalnya ketika menerapkan algoritma tersebut diterapkan pada *node 'address'*, maka akan menghasilkan alamat yang telah didaftarkan oleh banyak *node* lain.

Berdasarkan penelitian yang telah dilakukan sebelumnya, belum ada penerapan *graph database* secara spesifik dalam industri *travel*, khususnya dalam hal ini untuk mengolah data hotel. Namun dari penelitian yang telah disebutkan, pemanfaatan *relationship* dari *graph database* telah terbukti efektif untuk menerapkan teknik pendeteksi penipuan serta dapat menemukan kumpulan individu yang melakukan penipuan (Srivastava & Singh, 2018), dan menyelesaikan berbagai permasalahan di sektor manufaktur seperti *supply chain* dan proses manufaktur (Buktar et al., 2023). Karena itu, untuk mengatasi permasalahan *mapping* pada Rodex Tours & Travel akan digunakan *graph database* untuk memanfaatkan struktur *graph* dalam menentukan *node* mana yang menjadi target *mapping*. Hal ini juga dapat membantu proses *update* data karena fleksibilitas *schema* yang dimiliki oleh *graph database* sehingga tidak perlu menambahkan *record* baru untuk menyimpan data yang akan dihapus.

1.2 Rumusan Masalah

Rumusan masalah berdasarkan latar belakang yang telah diuraikan di atas, adalah:

1. Metrik *similarity* apa yang memberikan hasil *similarity* yang paling tepat?
2. Bagaimana perbandingan ketepatan sistem *mapping* yang diusulkan dengan sistem *mapping* yang sekarang?

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah di atas, maka tujuan dari penelitian ini adalah:

1. Menerapkan *graph database* untuk proses *similarity search* dan *mapping* untuk data hotel Rodex Tours & Travel serta membandingkan metrik *similarity* mana yang memberikan hasil yang paling tepat.

1.4 Ruang Lingkup

Ruang lingkup penelitian ini dibatasi pada:

1. *Tools* yang digunakan adalah Neo4j
2. Data yang digunakan berupa data hotel dari *vendor* dengan *field* ID, Name, Destination, City, Country, Star, Provider, Address#1, Address#2, Address#3, Email, Fax, Phone, Longitude dan Latitude dari 9 vendor oleh Rodex Tours & Travel dalam bentuk file *comma separated value*.
3. Proses yang akan dilakukan adalah:

a. Analisis untuk membuat *graph schema* dengan menentukan *field* yang dijadikan *node*

dan *property,* serta *relationship* yang akan dibuat pada *database*.

b. Pembuatan *graph schema* untuk *database*.

c. Mencari *node* yang *similar* dengan *field* dari *input* data menggunakan Jaccard Similarity, Overlap Coefficient dan Cosine Similarity.

d. Memberikan skor minimum untuk *similarity* pada setiap tipe *node* yang akan digunakan sebagai *threshold.*

e. Menentukan *node* dengan nilai akhir *similarity* tertinggi sebagai target *mapping*.

f. Menambahkan *value* dari *input* data menjadi *property* dari *node* yang menjadi target *mapping.*

4. *Output* yang dihasilkan adalah daftar data vendor dan target *mapping*-nya beserta *similarity score* yang diperoleh untuk setiap pendekatan yang digunakan.

5. *Output* ditampilkan pada *website* dalam bentuk tabel dan dapat di-*download* sebagai *file* CSV.

6. Pendekatan yang akan dievaluasi adalah metrik-metrik *similarity* yang digunakan.

7. Pengujian sistem dilakukan dengan:

a. Membandingkan hasil *similarity search* benar dari daftar *record-record* vendor beserta target *mapping*-nya dalam *file* CSV untuk setiap metrik *similarity*.

b. Membandingkan persentase hasil *similarity search* dan *mapping* benar dari daftar *record-record* vendor beserta target *mapping*-nya dalam *file* CSV antara sistem *mapping* saat ini dengan sistem yang diusulkan.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat bagi :

1. Bagi Perusahaan (Rodex Tours & Travel), penelitian ini dapat membantu dalam melakukan *mapping* data *vendor* terhadap data *master* dengan menggunakan *graph database*, sehingga dapat memperlancar proses reservasi hotel yang dilakukan oleh *customer*.