

ABSTRAK

Monica Evelyn:

Skripsi

Penerapan *Graph Database* Menggunakan *Labeled Property Graph Model* pada *Data Warehouse* PT XYZ dengan *Similarity Analysis* untuk Membantu Mengoptimalkan Proses *Decision Making*

Penelitian ini bertujuan untuk menerapkan *graph database* dengan model *Labeled Property Graph* pada *data warehouse* PT XYZ guna mengoptimalkan proses pengambilan keputusan. PT XYZ merupakan perusahaan yang bergerak di bidang manufaktur dan distribusi voucher belanja. Sebagai upaya untuk meningkatkan efektivitas pemasaran, perusahaan ini memerlukan pengelolaan data yang lebih dinamis dan efisien. *Data warehouse* yang digunakan saat ini masih berbasis *relational data model* yang memiliki keterbatasan dalam menangani data yang kompleks dan memiliki banyak relasi.

Penelitian ini menggunakan Neo4j sebagai platform *graph database* karena kemampuannya untuk mengelola data yang sangat besar dan kompleks dengan struktur yang lebih fleksibel. Dengan menggunakan algoritma *Filtered Node Similarity* yang merupakan bagian dari Neo4j, penelitian ini mampu menghitung kesamaan antar *node* untuk mengidentifikasi hubungan yang relevan antar data produk.

Hasil dari penelitian menunjukkan bahwa dibandingkan menggunakan *relational database*, penerapan *graph database* dengan model *Labeled Property Graph* dapat meningkatkan efisiensi dalam proses pengambilan keputusan di PT XYZ. Hasil survei menunjukkan bahwa sistem ini mampu menghasilkan data yang cukup akurat, sehingga dapat mendukung pengambilan keputusan yang lebih efektif. Kemudahan penggunaan program juga membuat staf sebagai pengguna untuk dapat dengan cepat mengakses dan menganalisis data yang dibutuhkan.

Kata kunci:

data warehouse, graph database, labeled property graph, Neo4j, Cypher, filtered node similarity

ABSTRACT

Monica Evelyn:

Undergraduate Thesis

Implementation of Graph Database with Labeled Property Graph Model in Data Warehouse at PT XYZ to Help Optimizing the Decision Making Process

This research aims to implement a graph database with the Labeled Property Graph model in the PT XYZ data warehouse to optimize the decision-making process. PT XYZ is a company engaged in the manufacturing and distribution of shopping vouchers. In an effort to increase marketing effectiveness, this company requires more dynamic and efficient data management. The data warehouse used today is still based on a relational data model which has limitations in handling complex data and has many relationships.

This research uses Neo4j as a graph database platform because of its ability to manage very large and complex data with a more flexible structure. By using the Filtered Node Similarity algorithm which is part of Neo4j, this research is able to calculate the similarity between nodes to identify relevant relationships between product data.

The results of the research show that, despite using the relational database, the application of graph database with Labeled Property Graph model can improve efficiency in the decision-making process at PT XYZ. The survey results show that this system is able to produce data that is quite accurate, so that it can support more effective decision making. The ease of use of the program also allows staff as users to quickly access and analyze the data needed.

Keywords:

data warehouse, graph database, labeled property graph, Neo4j, Cypher, filtered node similarity

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN.....	ii
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI.....	iii
KATA PENGANTAR.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR SEGMENT PROGRAM.....	xv
DAFTAR RUMUS.....	xvii
1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	5
1.3. Tujuan Penelitian	5
1.4. Ruang Lingkup	5
1.5. Manfaat Penelitian	8
1.6. Metodologi Penelitian	8
1.7. Sistematika Penulisan	9
2. LANDASAN TEORI	10
2.1. Tinjauan Pustaka	10
2.1.1. PT XYZ	10
2.1.2. DBPedia	10
2.1.3. <i>Data Warehouse</i>	11
2.1.4. <i>Graph Database</i>	12
2.1.5. <i>Labeled Property Graph</i>	12
2.1.6. Neo4j	13
2.1.7. Cypher	13
2.1.8. <i>Filtered Node Similarity</i>	13
2.1.9. <i>Jaccard Similarity</i>	13
2.2. Tinjauan Studi.....	14

2.2.1.	Decision Making Based on Data Analyses using Data Warehouses (Haxhiu, 2018)	14
2.2.2.	Design a Data Warehouse Schema from Document-Oriented Database (Bouaziz et al., 2019)	15
2.2.3.	Graph NoSQL Data Warehouse Creation (Sellami et al., 2020)	16
2.2.4.	NoSQL Databases: Critical Analysis and Comparison (Gupta et al., 2017)	17
2.2.5.	Big Data: The NoSQL and RDBMS Review (Zafar et al., 2016)	18
3.	ANALISA DAN DESAIN SISTEM	22
3.1.	Analisa Permasalahan	22
3.2.	Analisis Kebutuhan	23
3.3.	Desain <i>Database</i>	23
3.3.1.	Sumber Data	23
3.3.2.	Skema <i>Graph Data Warehouse</i>	25
3.4.	Desain Sistem	35
3.4.1.	Perhitungan <i>Similarity</i> antar Produk	36
3.4.2.	<i>Use Case Diagram</i>	39
3.4.3.	Desain <i>User Interface</i>	40
4.	IMPLEMENTASI SISTEM	44
4.1.	Implementasi <i>Database</i>	44
4.1.1.	Implementasi <i>Graph Data Warehouse</i>	44
4.1.2.	Implementasi <i>Query</i> Pengujian	50
4.2.	Implementasi Perhitungan <i>Similarity</i> pada Neo4j	53
4.2.1.	<i>Data Preprocessing</i>	53
4.2.2.	Pengambilan Fitur DBPedia	55
4.2.3.	Perhitungan <i>Similarity</i>	56
4.3.	Implementasi <i>Dashboard</i> berbasis <i>Website</i>	59
4.3.1.	Pembuatan Halaman <i>Home</i>	59
4.3.2.	Pembuatan Halaman <i>Campaign</i>	61
4.3.3.	Pembuatan Halaman <i>Campaign Detail</i>	63
4.3.4.	Pembuatan Halaman <i>Customer</i>	64
4.3.5.	Pembuatan Halaman <i>Customer Detail</i>	65
4.3.6.	Pembuatan Halaman <i>Product</i>	66
5.	PENGUJIAN SISTEM	67
5.1.	Perbandingan Performa <i>Graph Database</i> dengan <i>Relational Database</i>	67
5.1.1.	Pengujian Perbandingan Hasil	67

5.1.2.	Pengujian Perbandingan Kecepatan <i>Query</i>	72
5.1.3.	Pengujian Perbandingan Skema	75
5.2.	Perhitungan dan Pengujian Perbandingan Hasil <i>Similarity</i>	77
5.2.1.	Perbandingan Hasil <i>Similarity</i> untuk Fitur DBPedia.....	77
5.2.2.	Pengujian Perhitungan <i>Similarity</i> pada Neo4j	79
5.3.	Pengujian Fitur	83
5.3.1.	Halaman Login.....	83
5.3.2.	Halaman <i>Home</i>	84
5.3.3.	Halaman <i>Campaign</i>	87
5.3.4.	Halaman <i>Campaign Detail</i>	90
5.3.5.	Halaman <i>Customer</i>	93
5.3.6.	Halaman <i>Detail Customer</i>	97
5.3.7.	Halaman Product.....	99
5.4.	Pengujian <i>User</i>	100
5.5.	Pengujian Kepuasan Pengguna	103
6.	KESIMPULAN DAN SARAN	105
6.1.	Kesimpulan	105
6.2.	Saran.....	105
	DAFTAR REFERENSI.....	106

DAFTAR TABEL

Tabel 2.1 Tabel Perbandingan Penelitian Terkait.....	19
Tabel 3.1 Tabel <i>Constraint</i> pada <i>Graph Database</i>	26
Tabel 3.2 <i>Node Customer</i>	27
Tabel 3.3 <i>Node Transaction</i>	28
Tabel 3.4 <i>Node Category</i>	28
Tabel 3.5 <i>Node Product</i>	29
Tabel 3.6 <i>Node Merchant</i>	29
Tabel 3.7 <i>Node Publisher</i>	30
Tabel 3.8 <i>Node SKU</i>	30
Tabel 3.9 <i>Node Campaign</i>	30
Tabel 3.10 <i>Node DBPedia</i>	31
Tabel 3.11 <i>Edge BUYS_PRODUCT</i>	32
Tabel 3.12 <i>Edge PROMOTES_PRODUCT</i>	33
Tabel 3.13 <i>Edge SIMILAR_TO</i>	34
Tabel 3.14 <i>Edge SIMILAR_WITH</i>	35
Tabel 5.1 Waktu Kecepatan <i>Query</i> pada Neo4j.....	73
Tabel 5.2 Waktu Kecepatan <i>Query</i> pada PostgreSQL	73
Tabel 5.3 Perbandingan Kecepatan <i>Query</i> pada Neo4j dan PostgreSQL.....	73
Tabel 5.4 Nilai <i>Mean Average Precision</i> (mAP) Data DBPedia.....	78
Tabel 5.5 <i>List Pernyataan dan Rata-Rata Jawaban User</i>	101
Tabel 5.6 Perhitungan Hasil Survei	101
Tabel 5.7 Penilaian tiap Responden Program.....	102
Tabel 5.8 Pertanyaan dan Hasil Survei Kepuasan Pengguna	103
Tabel 5.9 Penilaian tiap Responden Hasil Program	103

DAFTAR GAMBAR

Gambar 3.1 <i>Star Schema Data Warehouse</i> PT XYZ	24
Gambar 3.2 ERD Sumber Data	24
Gambar 3.3 Skema <i>Graph Database</i>	26
Gambar 3.4 <i>Block Architecture System</i>	36
Gambar 3.5 <i>Flowchart</i> Alur Pembersihan Data Produk	37
Gambar 3.6 <i>Flowchart</i> Alur Pengambilan Fitur DBPedia.....	38
Gambar 3.7 <i>Flowchart</i> Proses Perhitungan <i>Similarity</i> antar Produk.....	39
Gambar 3.8 <i>Use Case Diagram</i>	40
Gambar 3.9 Desain Tampilan Halaman Login.....	41
Gambar 3.10 Desain Tampilan Halaman <i>Home</i>	41
Gambar 3.11 Desain Tampilan Halaman <i>Campaign</i>	42
Gambar 3.12 Desain Tampilan Halaman <i>Customer</i>	43
Gambar 3.13 Desain Tampilan Halaman <i>Customer</i> (2)	43
Gambar 5.1 Hasil <i>Query</i> 1 pada Neo4j	68
Gambar 5.2 Hasil <i>Query</i> 1 pada PostgreSQL	68
Gambar 5.3 Hasil <i>Query</i> 2 pada Neo4j	69
Gambar 5.4 Hasil <i>Query</i> 2 pada PostgreSQL	69
Gambar 5.5 Hasil <i>Query</i> 3 pada Neo4j	69
Gambar 5.6 Hasil <i>Query</i> 3 pada PostgreSQL	70
Gambar 5.7 Hasil <i>Query</i> 4 pada Neo4j	70
Gambar 5.8 Hasil <i>Query</i> 4 pada PostgreSQL	70
Gambar 5.9 Hasil <i>Query</i> 5 pada Neo4j	71
Gambar 5.10 Hasil <i>Query</i> 5 pada PostgreSQL	71
Gambar 5.11 Hasil <i>Query</i> 6 pada Neo4j	71
Gambar 5.12 Hasil <i>Query</i> 6 pada PostgreSQL	72
Gambar 5.13 Hasil <i>Query</i> 7 pada Neo4j	72
Gambar 5.14 Hasil <i>Query</i> 7 pada PostgreSQL	72
Gambar 5.15 Grafik Perbandingan Rata-Rata Kecepatan <i>Query</i> pada Neo4j dan PostgreSQL ...	74
Gambar 5.16 <i>Star Schema Campaign</i>	76
Gambar 5.17 <i>Star Schema Produk</i>	76

Gambar 5.18 Hasil <i>Query Intersection</i> Produk	79
Gambar 5.19 Hasil <i>Query Union</i> Produk	80
Gambar 5.20 <i>Graph</i> Hasil <i>Query Intersection</i>	80
Gambar 5.21 Hasil <i>Similarity</i> antar <i>Campaign</i>	81
Gambar 5.22 Daftar ID Produk pada <i>Campaign1</i>	81
Gambar 5.23 Daftar ID Produk pada <i>Campaign2</i>	82
Gambar 5.24 <i>Graph</i> Hasil <i>Query</i> Kemiripan <i>Campaign</i>	82
Gambar 5.25 Tampilan Halaman Login	83
Gambar 5.26 Tampilan Ketika Login Gagal	84
Gambar 5.27 Tampilan Halaman <i>Home</i>	84
Gambar 5.28 Tampilan Ketika Memilih Tanggal	85
Gambar 5.29 Tampilan Saat Tidak Ada <i>Category</i> Terpilih	86
Gambar 5.30 Tampilan Saat Terdapat <i>Category</i> yang Dipilih	86
Gambar 5.31 Tampilan Grafik Setelah Menekan Tombol “Continue”	87
Gambar 5.32 Tampilan Awal Halaman <i>Campaign</i>	88
Gambar 5.33 Tampilan <i>Filter</i> untuk Memilih Bulan	88
Gambar 5.34 Tampilan Ketika Memilih Nama <i>Campaign</i>	88
Gambar 5.35 Tampilan Memilih Nama <i>Campaign</i> pada Bulan Tertentu	89
Gambar 5.36 Tampilan Setelah Memilih <i>Campaign</i>	90
Gambar 5.37 Tampilan Rekomendasi Produk dari <i>Campaign</i> yang Dipilih	90
Gambar 5.38 Tampilan Awal Halaman <i>Campaign Detail</i>	91
Gambar 5.39 Tampilan Setelah Tombol <i>Select</i> Ditekan (1)	91
Gambar 5.40 Tampilan Setelah Tombol <i>Select</i> Ditekan (2)	92
Gambar 5.41 Tampilan Setelah Tombol <i>Select</i> Ditekan (3)	92
Gambar 5.42 Tampilan Ketika Tidak Ada <i>Campaign</i> yang Sejenis.....	93
Gambar 5.43 Tampilan Awal Halaman <i>Customer</i>	93
Gambar 5.44 Tampilan Urutan Berdasarkan Kolom <i>Transaction Count</i> Secara Ascending.....	94
Gambar 5.45 Tampilan Setelah Menekan Kolom <i>Total Amount</i> (Rp.).....	95
Gambar 5.46 Tampilan Setelah Menekan Kolom <i>Total Amount</i> (Rp.) Kedua Kali	95
Gambar 5.47 Tampilan Setelah Menekan Kolom <i>Total Amount</i> (Rp.) Ketiga Kali	96
Gambar 5.48 Tampilan Ketika Melakukan Pencarian <i>Customer ID</i>	96
Gambar 5.49 Tampilan Halaman Detail <i>Customer</i>	97
Gambar 5.50 Tampilan Setelah Menekan Tombol “By Transaction”	98

Gambar 5.51 Tampilan Setelah Menekan Tombol “By Product”	98
Gambar 5.52 Tampilan Halaman <i>Product</i>	99
Gambar 5.53 Tampilan Ketika Mencari Produk.....	99
Gambar 5.54 Tampilan Ketika Salah Satu <i>Row</i> Ditekan.....	100

DAFTAR SEGMENT PROGRAM

Segment Program 4.1 Pembuatan <i>Constraint</i> pada <i>Graph Data Warehouse</i>	44
Segment Program 4.2 <i>Load</i> Data dengan Cypher.....	46
Segment Program 4.3 Koneksi dengan Neo4j	47
Segment Program 4.4 <i>Load</i> Data dengan Python.....	47
Segment Program 4.5 Pembuatan Relasi antar <i>Node</i> Product dengan <i>Node</i> DBPedia	48
Segment Program 4.6 <i>Query</i> untuk Membuat Relasi	49
Segment Program 4.7 <i>Query</i> untuk Analisis pada Neo4j.....	50
Segment Program 4.8 <i>Query</i> untuk Analisis pada PostgreSQL	51
Segment Program 4.9 Membersihkan Data Produk	53
Segment Program 4.10 Melakukan POS Tagging pada Deskripsi Produk	54
Segment Program 4.11 Pengambilan Data DBPedia.....	55
Segment Program 4.12 <i>Product Similarity</i> pada Neo4j	56
Segment Program 4.13 Perhitungan <i>Precision at K</i>	57
Segment Program 4.14 <i>Query Intersection</i> untuk Produk.....	57
Segment Program 4.15 <i>Query Union</i> untuk Produk	58
Segment Program 4.16 <i>Campaign Similarity</i> pada Neo4j	58
Segment Program 4.17 <i>Query</i> Menampilkan <i>Similarity</i> antar <i>Campaign</i>	58
Segment Program 4.18 <i>Query</i> Menambah Relasi SIMILAR_WITH	59
Segment Program 4.19 <i>Query</i> untuk Menampilkan Daftar ID Produk	59
Segment Program 4.20 <i>Query</i> Menampilkan <i>List Category</i> untuk <i>Filter</i>	59
Segment Program 4.21 <i>Query</i> Menampilkan <i>List Merchant</i> untuk <i>Filter</i>	60
Segment Program 4.22 <i>Query</i> Menampilkan Data Transaksi Sesuai <i>Filter</i> Terpilih	60
Segment Program 4.23 <i>Query</i> untuk Menampilkan Nilai <i>Gauge Chart</i>	61
Segment Program 4.24 <i>Query</i> Menampilkan <i>List</i> Nama <i>Campaign</i>	61
Segment Program 4.25 <i>Query</i> Menampilkan <i>List</i> Produk.....	61
Segment Program 4.26 <i>Query</i> Menampilkan Grafik Penjualan.....	62
Segment Program 4.27 <i>Query</i> Menampilkan Rekomendasi Produk	62
Segment Program 4.28 <i>Query Campaign</i> Sejenis.....	63
Segment Program 4.29 <i>Query</i> Tabel Produk dan Transaksi	63
Segment Program 4.30 <i>Coding</i> Grafik Produk Paling Banyak Terjual	64

Segmen Program 4.31 <i>Query Menampilkan List Customer</i>	64
Segmen Program 4.32 <i>Query Menampilkan Grafik Jumlah Pembelian</i>	65
Segmen Program 4.33 <i>Query Menampilkan Rekomendasi Produk untuk Customer</i>	65
Segmen Program 4.34 <i>Query Menampilkan List Produk</i>	66
Segmen Program 4.35 <i>Query Menampilkan Rekomendasi Produk</i>	66

DAFTAR RUMUS

Rumus 2.1 <i>Jaccard Similarity</i>	14
---	----