

SKRIPSI

**STUDI BASIS DATA BERBASIS GRAF UNTUK SISTEM
REKOMENDASI WAKTU NYATA**



Sharon Kezia Felimandana

NPM: 6181901006

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2023**

UNDERGRADUATE THESIS

**GRAPH DATABASE STUDIES FOR REAL TIME
RECOMMENDATION SYSTEMS**



Sharon Kezia Felimandana

NPM: 6181901006

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2023**

LEMBAR PENGESAHAN

**STUDI BASIS DATA BERBASIS GRAF UNTUK SISTEM
REKOMENDASI WAKTU NYATA**

Sharon Kezia Felimandana

NPM: 6181901006

Bandung, 7 Juli 2023

Menyetujui,

Pembimbing

Dr. Veronica Sri Moertini

Ketua Tim Penguji

Anggota Tim Penguji

Maria Veronica, M.T.

Keenan Adiwijaya Leman, M.T.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

STUDI BASIS DATA BERBASIS GRAF UNTUK SISTEM REKOMENDASI WAKTU NYATA

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 7 Juli 2023



Sharon Kezia Felimandana
NPM: 6181901006

ABSTRAK

Sistem rekomendasi kini sudah menjadi fitur utama yang banyak digunakan oleh perusahaan dalam membangun sistem digitalnya untuk merekomendasikan berbagai hal, seperti lagu, film, video, teman pada media sosial, atau produk pada *e-commerce*. Sistem rekomendasi memungkinkan pengguna untuk menemukan produk yang dianggap sesuai atau relevan dengan preferensinya tanpa perlu mencarinya secara manual. Terdapat beberapa algoritma yang dapat digunakan untuk membuat model rekomendasi. Salah satu yang cukup populer adalah algoritma *collaborative filtering neighborhood-based* yang terbagi menjadi 2 metode, yaitu *item-based* dan *user-based*. Kedua algoritma ini bekerja dengan memanfaatkan data *rating* yang diberikan pengguna sebelumnya untuk mencari pengguna lain yang memiliki preferensi mirip dengan pengguna (*user-based*) atau mencari barang lain yang mirip dengan barang yang dibeli pengguna (*item based*).

Namun, algoritma *collaborative filtering neighborhood-based* membutuhkan waktu yang cukup lama dalam menyediakan rekomendasi karena harus mencari kemiripan antara seorang pengguna dengan seluruh pengguna lainnya atau satu barang dengan semua barang lainnya. Lamanya waktu yang dibutuhkan untuk membuat model rekomendasi menyebabkan sebagian tahap pembuatan model rekomendasi, yaitu perhitungan *similarity* dilakukan secara *offline*. Dengan demikian, model rekomendasi akan diperbarui secara berkala dalam rentang waktu tertentu berdasarkan data histori. Hal tersebut berdampak pada hasil rekomendasi yang menjadi kurang akurat dimana mungkin barang yang direkomendasikan sudah tidak lagi relevan bagi pengguna.

Untuk mengatasi permasalahan tersebut, pada skripsi ini dilakukan analisis mengenai sistem rekomendasi waktu nyata dimana rekomendasi diberikan secara *real time* pada saat pengguna berinteraksi dengan sistem. Data yang digunakan untuk memberikan rekomendasi adalah data yang baru didapat, misalnya dengan merekomendasikan barang yang serupa dengan barang yang baru saja dibeli. Dengan demikian, rekomendasi yang diberikan akan lebih relevan. Teknologi yang dimanfaatkan untuk mengembangkan sistem rekomendasi waktu nyata adalah teknologi sistem manajemen basis data berbasis graf atau yang biasa dikenal dengan istilah DBMS berbasis graf, yaitu Neo4j. Dengan menggunakan DBMS berbasis graf Neo4j, pemberian rekomendasi dapat dilakukan dengan lebih cepat. Hal ini dikarenakan basis data graf menyimpan data dalam bentuk graf dimana hubungan antar data tersebut telah tersimpan sehingga tidak diperlukan lagi operasi *join* ketika data dibutuhkan seperti pada basis data relasional.

Pembuatan model rekomendasi waktu nyata ini dilakukan dengan menggunakan data sintesis yang dibuat berdasarkan skema data *real*. Data tersebut kemudian dimodelkan dan dianalisis dengan memanfaatkan konsep-konsep graf. Setelah dibuat, model rekomendasi kemudian dievaluasi dengan membandingkan hasil rekomendasi yang diberikan sistem dengan hasil visualisasi graf secara manual. Selain itu, pada akhir penelitian ini juga dibangun sebuah prototipe sistem belanja *online* berbasis *website* dengan fitur-fitur sederhana untuk meluncurkan model rekomendasi yang dibuat. Berdasarkan hasil pengujian, kesimpulan yang diperoleh adalah bahwa Neo4j dapat digunakan untuk membangun sistem rekomendasi waktu nyata.

Kata-kata kunci: Sistem Rekomendasi, Rekomendasi Waktu Nyata, Basis Data Graf, Neo4j, Analisis Graf

ABSTRACT

Nowadays, a recommendation system has become a main feature for companies in developing their digital systems to recommend various things, such as songs, movies, videos, social media friends, and e-commerce products. The recommendation system allows users to find products that are relevant to their preferences without manually searching for them. There are several recommendation algorithms that can be used to build recommendation models. One of the popular algorithms is called neighborhood-based collaborative filtering, which can be classified into two methods, such as item-based and user-based. Both methods use user rating data to build recommendation models by finding other users with similar preferences (user-based) or by finding similar items to the purchased item (item-based).

However, neighborhood-based collaborative filtering algorithms can be time-consuming in providing recommendations for users because it has to find similarities between all other users or between all other products. The time required to build the recommendation model causes the similarity calculation to be performed offline. This leads to periodic updates of the recommendation models based on user historical data. As a result, user recommendations won't be accurate, as the recommendations may no longer be relevant for the users.

To overcome this problem, this thesis aims to analyze a real time recommendation system where the recommendation is provided instantly when the user interacts with the system. The data used to provide recommendations is the most recent data. For example, by recommending similar products to recently purchased products. Consequently, recommendations provided by the system will be more relevant to the users. The technology utilized to build real time recommendations is Neo4j, a graph based Database Management System (DBMS). By utilizing the Neo4j graph DBMS, recommendations will be provided faster. This is because a graph database stores data in a graph structure where the relationships between the data are also stored. So when the data is needed, there won't be a join operation performed as in a relational database.

The real time recommendation model is built using synthetic data which is generated based on the real data schema. The data is then modeled and analyzed by utilizing graph concepts. When the model is built, it will be evaluated by comparing the recommendation results provided by the system with manually visualized graph results. In addition, a prototype of an online shopping website will be developed at the end of this research to launch the real time recommendation model. Based on the evaluation result, it can be concluded that Neo4j can be used to build a real time recommendation systems.

Keywords: Recommendation System, Real-time Recommendation, Graph Database, Neo4j, Graph Analysis

Skripsi ini saya persembahkan kepada orang tua, keluarga, dan teman-teman yang mendukung saya selama proses pengerjaan skripsi ini. . .

KATA PENGANTAR

Puji syukur kepada Tuhan yang Maha Esa karena atas pertolongan-Nya penulis mampu menyelesaikan skripsi yang berjudul “Studi Basis Data Berbasis Graf untuk Sistem Rekomendasi Waktu Nyata” dengan baik. Skripsi ini dibuat sebagai salah satu syarat untuk memperoleh gelar S1 dari jurusan Teknik Informatika Universitas Katolik Parahyangan. Pembuatan skripsi ini tentu tidaklah mudah. Banyak hambatan yang dihadapi oleh penulis dalam proses penyusunan skripsi ini. Penulis menyadari bahwa skripsi ini akhirnya dapat terselesaikan berkat bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada :

1. Orang tua dan keluarga penulis yang selalu mendukung dan mendoakan penulis khususnya pada proses penyusunan skripsi ini;
 2. Ibu Dr. Veronica Sri Moertini, selaku dosen pembimbing yang telah meluangkan waktunya untuk membimbing, memberikan arahan, dan masukan dalam penyusunan skripsi ini;
 3. Ibu Maria Veronica, M.T. dan Bapak Keenan Adiwijaya Leman, M.T., selaku dosen penguji yang telah memberikan masukan dalam penyusunan skripsi ini;
 4. Tasha, Nadia, Vira, Ira, Kinan, dan Alma, selaku teman seperjuangan terdekat penulis yang selalu mendukung dan menyemangati penulis selama masa perkuliahan, khususnya dalam proses penyusunan skripsi ini;
 5. Nicho, Kesun, Nanda, Axel, dan Rio, selaku teman-teman seperjuangan skripsi yang selalu memberikan dukungan bagi penulis, khususnya dalam proses penyusunan skripsi ini;
 6. Angel, Michaela, Ferika, dan Vero, selaku teman terdekat penulis yang selalu memberikan dukungan dan semangat bagi penulis;
 7. Teman-teman lain yang tidak dapat disebutkan satu per satu yang selalu mendukung penulis.
- Skripsi yang disusun ini tentunya masih memiliki kekurangan. Oleh karena itu, penulis sangat terbuka dengan kritik dan saran dari pembaca untuk penyempurnaan skripsi ini. Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat bagi pembaca.

Bandung, Juli 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
DAFTAR KODE PROGRAM	xxvi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan	5
1.4 Batasan Masalah	5
1.5 Metodologi	6
1.6 Sistematika Pembahasan	7
2 LANDASAN TEORI	9
2.1 Sistem Rekomendasi	9
2.1.1 <i>Content Based Filtering</i> [1]	11
2.1.2 <i>Collaborative Filtering</i> [2]	11
2.1.3 Hybrid Based[3]	14
2.2 Graf[4]	14
2.2.1 Jenis Graf	15
2.2.2 Terminologi Graf	16
2.2.3 Analisis Graf[5]	19
2.3 Basis Data	21
2.3.1 Basis Data Relasional	21
2.3.2 Basis Data NoSQL	21
2.4 Basis Data Graf	22
2.5 Neo4j	24
2.5.1 Arsitektur Neo4j	25
2.5.2 Pengaksesan Neo4j[6]	26
2.5.3 Neo4j Bloom[7]	28
2.6 Bahasa Cypher	29
2.7 <i>Library Graph Data Science</i> [8]	34
2.8 <i>Cross-Selling</i> dan <i>Up-Selling</i> [9, 10]	36
2.9 <i>Influencer</i> [11]	37
3 STUDI EKSPLORASI	39
3.1 Eksperimen Penggunaan Bahasa Cypher pada Neo4j	39
3.1.1 Neo4j	39

3.1.2	Bahasa Cypher	46
3.2	Eksperimen Pembuatan Rekomendasi	63
3.2.1	Pembuatan Sampel Data	63
3.2.2	Pemodelan Data ke dalam Bentuk Graf	66
3.2.3	Analisis Graf untuk Pemberian Rekomendasi	69
3.2.4	Evaluasi Hasil Rekomendasi	73
4	PEMBUATAN MODEL REKOMENDASI DENGAN ANALISIS GRAF	75
4.1	Pembuatan Data Sintetis	75
4.2	Pemodelan Data ke dalam Bentuk Graf	89
4.3	Eksplorasi Data	94
4.4	Analisis Graf untuk Pemberian Rekomendasi	100
4.4.1	Rekomendasi berdasarkan data urutan pembelian seluruh pengguna	100
4.4.2	Rekomendasi berdasarkan data pertemanan	108
4.4.3	Rekomendasi berdasarkan <i>influencer</i> yang di- <i>follow</i>	117
4.5	Evaluasi Hasil Rekomendasi	122
5	PEMBANGUNAN PERANGKAT LUNAK DAN PENGUJIAN	125
5.1	Pembangunan Perangkat Lunak	125
5.1.1	Fitur-fitur Perangkat Lunak	125
5.1.2	Tampilan Antar Muka Perangkat Lunak	128
5.2	Pengujian Perangkat Lunak	136
6	KESIMPULAN DAN SARAN	145
6.1	Kesimpulan	145
6.2	Saran	145
	DAFTAR REFERENSI	147
	A KODE PROGRAM	149
	B HASIL REKOMENDASI	193

DAFTAR GAMBAR

1.1	Jumlah pengguna <i>e-commerce</i> dari tahun 2017 hingga tahun 2025	1
1.2	Platform <i>e-commerce</i>	2
1.3	Contoh rekomendasi	3
1.4	Contoh <i>collaborative filtering neighborhood-based user-based dan item-based</i>	3
2.1	Rekomendasi yang tidak dipersonalisasi	9
2.2	Rekomendasi semi personalisasi	10
2.3	Rekomendasi yang dipersonalisasi	10
2.4	Algoritma sistem rekomendasi	11
2.5	Contoh cara kerja <i>user-based collaborative filtering</i>	12
2.6	Contoh cara kerja <i>item-based collaborative filtering</i>	13
2.7	Contoh graf dengan <i>loop</i> atau gelang	14
2.8	Contoh graf sederhana	15
2.9	Contoh multigraf	15
2.10	Contoh graf semu	15
2.11	Contoh graf berarah	16
2.12	Contoh graf tidak berarah	16
2.13	Contoh graf bersisian	17
2.14	Contoh graf dengan simpul awal dan simpul akhir	17
2.15	Contoh graf dengan <i>degree</i> tertentu	17
2.16	Contoh graf dengan <i>in-degree</i> dan <i>out-degree</i> tertentu	18
2.17	Contoh <i>path</i> dan <i>cycle</i> pada graf	18
2.18	Contoh graf berbobot	18
2.19	Contoh <i>pathfinding</i> dengan algoritma <i>Shortest Path</i>	19
2.20	Contoh <i>centrality</i> dengan algoritma <i>PageRank</i>	20
2.21	Contoh <i>community detection</i> dengan algoritma <i>Strongly Connected Components (SCC)</i>	20
2.22	Perbandingan waktu eksekusi basis data graf Neo4j dan basis data relasional MySQL	23
2.23	Komponen basis data graf	24
2.24	Basis data graf Twitter	24
2.25	<i>HA Architecture</i>	26
2.26	Contoh Neo4j Shell	27
2.27	Contoh Neo4j Browser	27
2.28	Contoh Neo4j Desktop	28
2.29	Contoh Neo4j Bloom	29
2.30	<i>Library GDS</i> pada Neo4j Desktop	34
2.31	Contoh teknik <i>cross-selling</i>	37
2.32	Contoh teknik <i>up-selling</i>	37
3.1	<i>Command line</i> untuk memulai server Neo4j dan mengakses Neo4j Shell	40
3.2	Tampilan Neo4j Shell	40
3.3	<i>Command line</i> untuk menghentikan server Neo4j setelah mengakses Neo4j Shell	40
3.4	<i>Command line</i> untuk memulai Neo4j server	41
3.5	Tampilan Neo4j Browser	41

3.6	Tampilan Neo4j Browser dengan menu	42
3.7	Hasil eksekusi kueri berupa graf pada Neo4j Browser	42
3.8	Hasil eksekusi kueri berupa tabel pada Neo4j Browser	43
3.9	Hasil eksekusi kueri berupa tabel pada Neo4j Browser dengan properti tertentu	43
3.10	<i>Command line</i> untuk menghentikan Neo4j server setelah mengakses Neo4j Browser	43
3.11	Tampilan Neo4j Desktop	44
3.12	Tampilan Neo4j Desktop saat server Neo4j belum dimulai	44
3.13	Tampilan Neo4j Desktop saat server Neo4j sudah aktif	45
3.14	<i>Plugins</i> atau <i>library</i> pada Neo4j Desktop	45
3.15	Hasil eksekusi kueri untuk membuat sebuah simpul	46
3.16	Hasil eksekusi kueri untuk membuat sebuah simpul dengan label	47
3.17	Hasil eksekusi kueri untuk membuat sebuah simpul dengan 2 label	47
3.18	Hasil eksekusi kueri untuk membuat sebuah simpul dengan properti	48
3.19	Hasil eksekusi kueri untuk membuat sebuah simpul dengan 2 label dan properti	48
3.20	Hasil eksekusi kueri untuk membuat beberapa simpul dalam 1 kueri	49
3.21	Hasil eksekusi kueri untuk membuat sebuah hubungan antara 2 buah simpul	49
3.22	Hasil eksekusi kueri untuk membuat sebuah hubungan antara 2 buah simpul dengan properti	50
3.23	Hasil eksekusi kueri untuk membuat sebuah hubungan antara 2 buah simpul dalam 1 kueri	50
3.24	Hasil eksekusi kueri untuk membuat sebuah hubungan antara 2 buah simpul dalam basis data	51
3.25	Hasil eksekusi kueri untuk menemukan simpul dengan label tertentu	51
3.26	Hasil eksekusi kueri untuk menemukan simpul dengan label dan properti tertentu	52
3.27	Hasil eksekusi kueri untuk menemukan simpul dan mengembalikan properti tertentu dari simpul	52
3.28	Hasil eksekusi kueri untuk menemukan pasangan simpul yang memiliki hubungan tertentu	53
3.29	Hasil eksekusi kueri untuk menemukan pola dengan <i>optional match</i>	53
3.30	Hasil eksekusi kueri untuk menemukan pola dengan <i>match</i>	54
3.31	Hasil eksekusi kueri untuk mengembalikan hasil kueri berupa simpul	54
3.32	Hasil eksekusi kueri untuk mengembalikan hasil kueri berupa properti simpul	55
3.33	Hasil eksekusi kueri untuk meneruskan hasil kueri ke bagian kueri selanjutnya	55
3.34	Hasil eksekusi kueri untuk meneruskan hasil kueri ke bagian kueri selanjutnya menggunakan fungsi agregat	56
3.35	Hasil eksekusi kueri untuk memfilter hasil kueri	56
3.36	Hasil eksekusi kueri untuk memfilter hasil kueri dengan fungsi agregat	57
3.37	Hasil eksekusi kueri untuk meng- <i>update value</i> properti sebuah simpul	57
3.38	Hasil eksekusi kueri untuk menambah label sebuah simpul	58
3.39	Hasil eksekusi kueri untuk menghapus properti sebuah simpul	58
3.40	Hasil eksekusi kueri untuk membuat atau mengembalikan sebuah simpul	59
3.41	Hasil eksekusi kueri untuk membuat atau mengembalikan sebuah simpul dengan perintah <i>on create</i> dan <i>on match</i>	59
3.42	Hasil eksekusi kueri untuk membuat atau mengembalikan sebuah hubungan antara 2 buah simpul	60
3.43	Tampilan sebelum kueri untuk menghapus label sebuah simpul dieksekusi	60
3.44	Tampilan setelah kueri untuk menghapus label sebuah simpul dieksekusi	61
3.45	Tampilan setelah kueri untuk menghapus properti sebuah simpul dieksekusi	61
3.46	Tampilan sebelum kueri untuk menghapus hubungan antara sepasang simpul dieksekusi	62
3.47	Tampilan setelah kueri untuk menghapus hubungan antara sepasang simpul dieksekusi	62

3.48	Tampilan setelah kueri untuk menghapus sebuah simpul dieksekusi	63
3.49	Eksperimen pemodelan simpul produk	66
3.50	Eksperimen pemodelan simpul <i>user</i>	67
3.51	Eksperimen pemodelan hubungan pertemanan <i>user</i>	67
3.52	Eksperimen pemodelan hubungan pembelian produk	68
3.53	Eksperimen pemodelan hubungan memfavoritkan produk	68
3.54	Eksperimen pemodelan hubungan menambahkan produk ke keranjang	69
3.55	Contoh ilustrasi graf urutan pembelian	69
3.56	Contoh ilustrasi graf gabungan urutan pembelian seluruh pengguna	70
3.57	Contoh ilustrasi pemberian rekomendasi berdasarkan graf gabungan urutan pembelian	70
3.58	Hasil eksperimen pembuatan graf gabungan urutan pembelian	71
3.59	Hasil eksperimen pemberian rekomendasi berdasarkan graf gabungan urutan pembelian	71
3.60	Contoh pemberian rekomendasi berdasarkan pertemanan	72
3.61	Hasil eksperimen pemberian rekomendasi berdasarkan pertemanan	72
3.62	Hasil eksperimen graf pertemanan dan pembelian	73
4.1	Struktur tabel data sintetis berdasarkan hasil pengamatan	76
4.2	Distribusi data produk	78
4.3	Distribusi data penjual	80
4.4	Distribusi data <i>user</i>	82
4.5	Distribusi data pertemanan	84
4.6	Distribusi data pembelian	86
4.7	Distribusi data keranjang	87
4.8	Distribusi data favorit	88
4.9	Hasil pemodelan simpul produk, kategori, subkategori, dan subsubkategori	90
4.10	Hasil pemodelan hubungan masuk kategori, masuk subkategori, dan masuk subsubkategori	91
4.11	Hasil pemodelan simpul <i>seller</i>	91
4.12	Hasil pemodelan hubungan jual	92
4.13	Hasil pemodelan simpul <i>user</i>	92
4.14	Hasil pemodelan hubungan <i>follow</i>	93
4.15	Hasil pemodelan hubungan beli	93
4.16	Hasil pemodelan hubungan tambah keranjang	94
4.17	Hasil pemodelan hubungan favorit	94
4.18	Hasil eksplorasi data produk berdasarkan hubungan kategori	95
4.19	Hasil eksplorasi hubungan kategori produk	95
4.20	Hasil eksplorasi data penjual	96
4.21	Hasil eksplorasi data pertemanan dengan algoritma <i>Strongly Connected Component</i>	97
4.22	Hasil eksplorasi data pertemanan dengan algoritma <i>PageRank</i>	97
4.23	Hasil eksplorasi data pembelian	98
4.24	Hasil eksplorasi data keranjang	99
4.25	Hasil eksplorasi data favorit	99
4.26	Ilustrasi graf gabungan urutan pembelian	101
4.27	Ilustrasi rekomendasi level 1 dan 2 berdasarkan graf gabungan urutan pembelian	102
4.28	Contoh hasil visualisasi graf pembelian	103
4.29	Contoh hasil visualisasi graf urutan pembelian setiap pengguna	104
4.30	Contoh hasil visualisasi graf gabungan urutan pembelian seluruh pengguna	104
4.31	Hasil rekomendasi produk berdasarkan data urutan pembelian untuk contoh 1	105
4.32	Hasil visualisasi rekomendasi produk berdasarkan data urutan pembelian untuk contoh 1	105
4.33	Hasil rekomendasi produk berdasarkan data urutan pembelian untuk contoh 2	106

4.34	Hasil visualisasi rekomendasi produk berdasarkan data urutan pembelian untuk contoh 2	106
4.35	Hasil visualisasi rekomendasi produk berdasarkan data urutan pembelian untuk contoh 3	107
4.36	Ilustrasi graf komunitas pengguna	108
4.37	Ilustrasi rekomendasi berdasarkan kategori dalam komunitas	109
4.38	Ilustrasi rekomendasi berdasarkan <i>rating</i> dalam komunitas	110
4.39	Ilustrasi rekomendasi berdasarkan produk populer dalam komunitas	111
4.40	Contoh hasil visualisasi graf pertemanan	112
4.41	Hasil visualisasi graf pertemanan dengan algoritma <i>Strongly Connected Component</i>	112
4.42	<i>Top 5</i> hasil rekomendasi produk dengan kategori sama dalam komunitas	113
4.43	Contoh hasil visualisasi histori data pembelian, favorit, dan keranjang	114
4.44	Hasil visualisasi rekomendasi produk berdasarkan <i>rating</i>	115
4.45	Contoh hasil visualisasi pembelian produk oleh teman 1 komunitas dengan <i>rating</i>	116
4.46	Hasil rekomendasi produk populer dalam komunitas	116
4.47	Hasil visualisasi rekomendasi produk populer dalam komunitas	117
4.48	Ilustrasi rekomendasi berdasarkan <i>influencer</i> yang <i>di-follow</i>	118
4.49	Hasil visualisasi graf pertemanan dengan algoritma <i>PageRank</i>	119
4.50	Hasil visualisasi 5 <i>influencer</i>	120
4.51	Hasil rekomendasi produk berdasarkan produk yang dibeli <i>influencer</i> yang <i>di-follow</i> untuk contoh 1	120
4.52	Hasil visualisasi rekomendasi berdasarkan produk yang dibeli <i>influencer</i> untuk contoh 1	121
4.53	Hasil visualisasi rekomendasi berdasarkan produk yang dibeli <i>influencer</i> untuk contoh 2	122
4.54	Posisi waktu eksekusi	123
5.1	Diagram <i>Use Case</i>	126
5.2	Tampilan perangkat lunak pada halaman <i>index</i>	129
5.3	Tampilan perangkat lunak pada halaman <i>login</i>	130
5.4	Tampilan perangkat lunak pada halaman <i>home 1</i>	131
5.5	Tampilan perangkat lunak pada halaman <i>home 2</i>	131
5.6	Tampilan perangkat lunak pada halaman <i>home 3</i>	132
5.7	Tampilan perangkat lunak pada halaman keranjang	133
5.8	Tampilan perangkat lunak pada halaman favorit	133
5.9	Tampilan perangkat lunak pada halaman histori pembelian	134
5.10	Tampilan perangkat lunak pada halaman berhasil masuk keranjang	135
5.11	Tampilan perangkat lunak pada halaman berhasil favorit	135
5.12	Tampilan perangkat lunak pada halaman berhasil beli	136
5.13	Tampilan perangkat lunak pada pengujian dengan kondisi 1	137
5.14	Tampilan perangkat lunak pada pengujian dengan kondisi 2	138
5.15	Tampilan perangkat lunak pada pengujian dengan kondisi 3	139
5.16	Tampilan perangkat lunak pada pengujian dengan kondisi 4	140
5.17	Tampilan perangkat lunak pada pengujian dengan kondisi 5	141
5.18	Tampilan perangkat lunak pada pengujian dengan kondisi 6	142
5.19	Tampilan perangkat lunak pada pengujian dengan kondisi 7	143
5.20	Tampilan perangkat lunak pada pengujian dengan kondisi 8	143
5.21	Tampilan perangkat lunak pada pengujian dengan kondisi 9	144
B.1	Hasil rekomendasi produk dengan kategori sama dalam komunitas (1)	193
B.2	Hasil rekomendasi produk dengan kategori sama dalam komunitas (2)	194
B.3	Hasil rekomendasi produk dengan <i>rating</i> tinggi dalam komunitas	195

DAFTAR TABEL

2.1	Kueri CRUD pada bahasa Cypher	31
2.2	Kueri lain yang sering digunakan pada bahasa Cypher	33
3.1	Contoh sampel data produk	63
3.2	Contoh sampel data <i>user</i>	64
3.3	Contoh sampel data pembelian	64
3.4	Contoh sampel data pertemanan	65
3.5	Contoh sampel data keranjang	65
3.6	Contoh sampel data favorit	66
4.1	Contoh data produk	77
4.2	Deskripsi atribut data produk	77
4.3	Contoh data penjual	79
4.4	Deskripsi atribut data penjual	79
4.5	Contoh data <i>user</i>	81
4.6	Deskripsi atribut data <i>user</i>	81
4.7	Contoh data pertemanan	82
4.8	Deskripsi atribut data pertemanan	83
4.9	Contoh data pembelian	84
4.10	Deskripsi atribut data pembelian	85
4.11	Contoh data keranjang	86
4.12	Deskripsi atribut data keranjang	86
4.13	Contoh data favorit	88
4.14	Deskripsi atribut data favorit	88
4.15	Rangkuman waktu eksekusi model rekomendasi	123
5.1	Deskripsi <i>user</i> pada diagram <i>use case</i>	126
5.2	Deskripsi setiap <i>use case</i> pada diagram <i>use case</i>	126
5.3	Skenario <i>use case</i> login	127
5.4	Skenario <i>use case</i> menambahkan produk ke dalam keranjang	127
5.5	Skenario <i>use case</i> memfavorit produk	128
5.6	Skenario <i>use case</i> membeli produk	128
5.7	Kondisi pengujian 1	136
5.8	Kondisi pengujian 2	138
5.9	Kondisi pengujian 3	139
5.10	Kondisi pengujian 4	140
5.11	Kondisi pengujian 5	140
5.12	Kondisi pengujian 6	141

DAFTAR KODE PROGRAM

2.1	Kode untuk memproyeksikan graf	35
2.2	Kode Algoritma <i>Dijkstra Source-Target Shortest Path</i>	35
2.3	Kode Algoritma <i>PageRank</i>	36
2.4	Kode Algoritma <i>Strongly Connected Component</i>	36
3.1	Kueri untuk membuat sebuah simpul	46
3.2	Kueri untuk membuat sebuah simpul dengan label	46
3.3	Kueri untuk membuat sebuah simpul dengan 2 label	47
3.4	Kueri untuk membuat sebuah simpul dengan properti	47
3.5	Kueri untuk membuat sebuah simpul dengan 2 label dan properti	48
3.6	Kueri untuk membuat beberapa simpul dalam 1 kueri	48
3.7	Kueri untuk membuat sebuah hubungan antara 2 buah simpul	49
3.8	Kueri untuk membuat sebuah hubungan antara 2 buah simpul dengan properti	49
3.9	Kueri untuk membuat sebuah hubungan antara 2 buah simpul dalam 1 kueri	50
3.10	Kueri untuk membuat sebuah hubungan antara 2 buah simpul dalam basis data	50
3.11	Kueri untuk menemukan simpul dengan label tertentu	51
3.12	Kueri untuk menemukan simpul dengan label dan properti tertentu	51
3.13	Kueri untuk menemukan simpul dan mengembalikan properti tertentu dari simpul	52
3.14	Kueri untuk menemukan pasangan simpul yang memiliki hubungan tertentu	52
3.15	Kueri untuk menemukan pola dengan <i>optional match</i>	53
3.16	Kueri untuk menemukan pola dengan <i>match</i>	53
3.17	Kueri untuk mengembalikan hasil kueri berupa simpul	54
3.18	Kueri untuk mengembalikan hasil kueri berupa properti simpul	54
3.19	Kueri untuk meneruskan hasil kueri ke bagian kueri selanjutnya	55
3.20	Kueri untuk meneruskan hasil kueri ke bagian kueri selanjutnya menggunakan fungsi agregat	55
3.21	Kueri untuk memfilter hasil kueri	56
3.22	Kueri untuk memfilter hasil kueri dengan fungsi agregat	56
3.23	Kueri untuk meng- <i>update value</i> properti sebuah simpul	57
3.24	Kueri untuk menambah label sebuah simpul	57
3.25	Kueri untuk menghapus properti sebuah simpul	58
3.26	Kueri untuk membuat atau mengembalikan sebuah simpul	58
3.27	Kueri untuk membuat atau mengembalikan sebuah simpul dengan perintah <i>on create</i> dan <i>on match</i>	59
3.28	Kueri untuk membuat atau mengembalikan sebuah hubungan antara 2 buah simpul	59
3.29	Kueri untuk menghapus label sebuah simpul	60
3.30	Kueri untuk menghapus properti sebuah simpul	61
3.31	Kueri untuk menghapus hubungan antara sepasang simpul	61
3.32	Kueri untuk menghapus sebuah simpul	62
4.1	Contoh kode untuk pemodelan simpul	89
4.2	Contoh kode untuk pemodelan hubungan	89

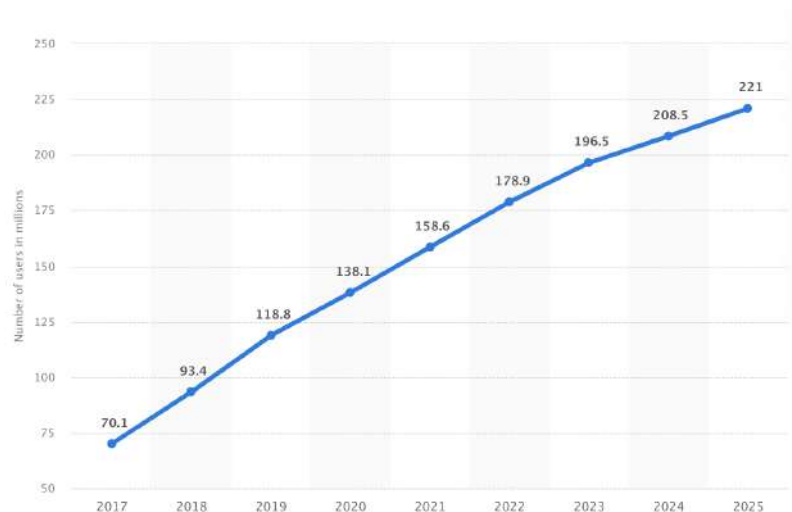
A.1	Contoh kueri data untuk eksperimen penggunaan bahasa Cypher	149
A.2	Kode eksperimen pemodelan data ke dalam bentuk graf	149
A.3	Kode eksperimen analisis graf untuk pemberian rekomendasi	150
A.4	Kode untuk membuat data sintetis	150
A.5	Kode untuk pemodelan data ke dalam bentuk graf	154
A.6	Kode analisis graf untuk pemberian rekomendasi	155
A.7	Kode evaluasi hasil rekomendasi	158
A.8	Kode perangkat lunak halaman <i>index</i>	162
A.9	Kode perangkat lunak halaman <i>login</i>	164
A.10	Kode perangkat lunak halaman <i>home</i>	164
A.11	Kode perangkat lunak halaman keranjang	171
A.12	Kode perangkat lunak halaman beli	173
A.13	Kode perangkat lunak halaman favorit	175
A.14	Kode perangkat lunak halaman berhasil masuk keranjang	177
A.15	Kode perangkat lunak halaman berhasil beli	179
A.16	Kode perangkat lunak halaman berhasil favorit	182
A.17	Kode perangkat lunak untuk <i>styling</i>	184
A.18	Kode perangkat lunak untuk membuat koneksi dengan basis data Neo4j	187
A.19	Kode perangkat lunak untuk menyimpan <i>function</i> kueri	187

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada zaman yang serba digital ini, teknologi telah mengubah gaya hidup manusia. Beberapa aktivitas atau pekerjaan kini dapat dilakukan secara *online*. Berbagai sektor telah memanfaatkan teknologi, termasuk sektor bisnis yang ditandai dengan munculnya sistem belanja *online* atau *e-commerce*. Seiring dengan teknologi yang terus berkembang, jumlah pengguna *e-commerce* juga terus meningkat dari tahun ke tahun, seperti yang dapat dilihat pada Gambar 1.1¹. Jumlah pengguna *e-commerce* di Indonesia bahkan diprediksi akan terus meningkat hingga mencapai 221 juta pengguna pada tahun 2025.



Gambar 1.1: Jumlah pengguna *e-commerce* dari tahun 2017 hingga tahun 2025

Kemudahan dalam berbelanja secara online merupakan salah satu faktor yang menyebabkan bertambahnya jumlah pengguna *e-commerce* dari waktu ke waktu. Dalam makalahnya[12], Abdul Gaffar Khan menjelaskan bahwa terdapat beberapa manfaat yang dirasakan oleh konsumen dengan adanya sistem belanja *online* atau *e-commerce*. Konsumen dapat dengan mudah berbelanja kapan pun dan di mana pun selama 24 jam tanpa harus datang ke toko secara langsung. Jangkauan wilayah yang lebih luas untuk membeli produk juga merupakan salah satu alasan yang menyebabkan banyaknya konsumen beralih ke sistem belanja *online*. Dengan adanya sistem belanja *online* ini, konsumen dapat dengan mudah membeli produk yang tidak tersedia di wilayah tempat tinggal konsumen hanya melalui internet. Selain itu, konsumen juga dapat melihat *rating* dan *review* suatu produk dari konsumen lain yang telah membeli produk tersebut. *Rating* dan *review* ini dapat digunakan oleh konsumen sebagai pertimbangan sebelum membeli suatu produk.

¹Statista Research Department (2021) Number of e-commerce users in Indonesia from 2017 to 2025. <https://www.statista.com/forecasts/251635/e-commerce-users-in-indonesia>. 19 Juni 2022.

Tingginya jumlah pengguna *e-commerce* telah memunculkan beberapa platform *e-commerce* baru yang siap bersaing dengan platform *e-commerce* yang sudah populer, seperti Shopee, Tokopedia, Lazada, Bukalapak, dan BliBli, seperti yang dapat dilihat pada Gambar 1.2². Seluruh platform *e-commerce* tersebut tentunya berlomba-lomba untuk memberikan kenyamanan dan pelayanan yang terbaik bagi penggunanya. Salah satu cara yang dilakukan adalah dengan selalu memberikan rekomendasi produk yang sesuai dengan kebutuhan dan ketertarikan konsumen. Rekomendasi produk ini diberikan untuk memudahkan pengguna dalam memilih produk yang dianggap sesuai untuk seorang konsumen di antara banyaknya produk yang tersedia pada *e-commerce*. Platform *e-commerce* ini dapat menghasilkan rekomendasi produk bagi pengguna dengan menggunakan sebuah sistem yang dikenal dengan sistem rekomendasi.



Gambar 1.2: Platform *e-commerce*

Sistem rekomendasi merupakan suatu perangkat lunak dalam aplikasi atau *website* yang memberikan saran berupa barang yang mungkin menarik bagi pengguna dengan mempertimbangkan berbagai data, seperti profil, preferensi, atau interaksi pengguna dalam aplikasi atau *website* tersebut[13]. Sistem rekomendasi telah digunakan di berbagai bidang, seperti merekomendasikan video pada Youtube, film pada Netflix, lagu pada Spotify, atau teman pada media sosial. Selain itu, sistem rekomendasi juga banyak digunakan untuk merekomendasikan produk pada sistem belanja *online*, seperti yang dapat dilihat pada Gambar 1.3[14]. Sistem rekomendasi ini dapat memudahkan pengguna dalam menggunakan suatu aplikasi atau *website* karena pengguna dapat memperoleh barang-barang yang menarik dan relevan tanpa perlu mencarinya.

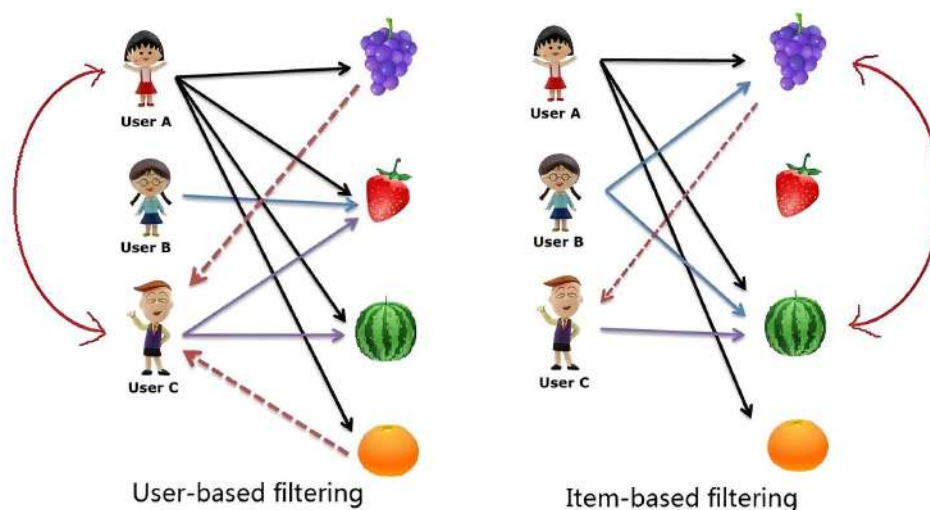
Terdapat beberapa algoritma yang dapat digunakan untuk membuat sistem rekomendasi. Salah satu algoritma rekomendasi yang cukup populer adalah algoritma *collaborative filtering*. *Collaborative filtering* merupakan suatu metode rekomendasi untuk membuat prediksi otomatis mengenai ketertarikan seorang pengguna pada suatu produk dengan mengumpulkan informasi preferensi atau selera dari banyak pengguna[2]. Pada algoritma *collaborative filtering*, informasi atau data yang dikumpulkan untuk membuat rekomendasi adalah berupa data *rating* atau penilaian pengguna terhadap suatu produk[2]. Salah satu teknik yang banyak digunakan dalam algoritma *collaborative filtering* adalah *memory-based* (berbasis memori) atau biasa dikenal dengan *neighborhood-based* (berbasis tetangga). Algoritma *collaborative filtering* berbasis tetangga terbagi menjadi *user-based* dan *item-based*. Kedua jenis algoritma *neighborhood-based* ini bekerja dengan cara mencari tetangga

²iPrice (2019) Shopee Jadi E-Commerce Paling Top dari Masa ke Masa. <https://databoks.katadata.co.id/datapublish/2019/09/03/shopee-jadi-e-commerce-paling-top-dari-masa-ke-masa>. 19 Juni 2022.



Gambar 1.3: Contoh rekomendasi

berdasarkan kemiripan data *rating*. Pada *user-based*, rekomendasi diberikan dengan mencari kemiripan seorang pengguna dengan pengguna lain berdasarkan data *rating* yang diberikan sebelumnya dan merekomendasikan barang yang telah diberi *rating* tinggi oleh pengguna lain, namun belum diberi *rating* oleh pengguna tersebut[14]. Sedangkan pada *item-based*, rekomendasi diberikan dengan mencari kemiripan suatu barang dengan barang yang disukai berdasarkan *rating* yang diberikan oleh pengguna[14].

Gambar 1.4: Contoh *collaborative filtering neighborhood-based user-based* dan *item-based*

Sebagai contoh, Gambar 1.4³ merupakan ilustrasi untuk rekomendasi dengan algoritma *collaborative filtering neighborhood-based user-based* dan *item-based*. Pada ilustrasi untuk *user-based*, diketahui bahwa pengguna C mirip dengan pengguna A. Oleh karena itu, pengguna C akan direkomendasikan produk yang dibeli oleh pengguna A namun belum pernah dibeli oleh pengguna C, yaitu

³Carlos Pinela (2017) Recommender Systems — User-Based and Item-Based Collaborative Filtering. <https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f>. 17 Juli 2023.

buah anggur dan jeruk. Sedangkan pada ilustrasi untuk *item-based*, diketahui bahwa buah anggur dan buah semangka mirip. Oleh karena itu, jika pengguna C pernah membeli buah semangka, maka pengguna C akan direkomendasikan buah anggur yang belum pernah dibeli sebelumnya.

Untuk menyediakan rekomendasi bagi pengguna, kedua algoritma *collaborative filtering* berbasis tetangga membutuhkan data penilaian produk yang cukup banyak. Jika data penilaian produk yang tersedia terlalu sedikit, maka akan menyebabkan masalah *sparsity*. *Sparsity* merupakan salah satu masalah utama pada algoritma *collaborative filtering* yang terjadi apabila sebagian besar pengguna hanya memberikan penilaian pada beberapa produk saja sedangkan produk yang tersedia jumlahnya sangat banyak sehingga menyebabkan sulitnya untuk menemukan tetangga dan berakibat pada penurunan kinerja algoritma[14]. Namun, banyaknya jumlah pengguna dan barang yang tersedia juga dapat menyebabkan masalah *scalability*. Untuk dapat mengolah data yang besar, algoritma rekomendasi *collaborative filtering* berbasis tetangga membutuhkan waktu yang cukup lama untuk menyediakan rekomendasi bagi pengguna[1]. Hal ini dikarenakan algoritma rekomendasi ini termasuk algoritma yang cukup kompleks, dimana algoritma harus mencari kemiripan dari setiap pengguna atau barang.

Lamanya waktu yang dibutuhkan algoritma *collaborative filtering* berbasis tetangga untuk membuat model rekomendasi menyebabkan sebagian tahap pembuatan model rekomendasi, yaitu perhitungan *similarity* dilakukan secara *offline*. Perhitungan *similarity* atau kemiripan dibuat secara *offline* dengan memanfaatkan data histori pengguna yang tersimpan pada basis data, yaitu data penilaian yang telah dikumpulkan dalam rentang waktu tertentu[12]. Dengan demikian, data *similarity* sudah tersedia dan rekomendasi dapat diberikan kepada pengguna dengan cepat ketika dibutuhkan, yaitu ketika pengguna menggunakan aplikasi atau *website*. Cara ini dianggap lebih efisien dalam memberikan rekomendasi agar pengguna tidak perlu menunggu halaman aplikasi atau *website* yang digunakan untuk membuat rekomendasi[14].

Walaupun pembuatan model rekomendasi secara *offline* ini cukup efektif, namun sebagian model rekomendasi yang dibuat secara *offline* menyebabkan rekomendasi disediakan atau diperbarui bagi pengguna secara berkala dalam rentang waktu tertentu. Dengan rekomendasi yang diberikan secara berkala, rekomendasi bagi pengguna menjadi kurang akurat dimana mungkin barang yang direkomendasikan sudah tidak lagi relevan bagi pengguna. Mungkin rekomendasi *offline* tidak akan menjadi masalah bagi pengguna yang memang memiliki ketertarikan pada sesuatu dalam jangka waktu yang cukup lama. Namun, bagi pengguna yang hanya memiliki ketertarikan dalam waktu singkat, rekomendasi yang dibuat secara *offline* akan menjadi tidak relevan.

Untuk mengatasi permasalahan tersebut, pada skripsi ini dilakukan analisis mengenai sistem rekomendasi waktu nyata atau *real time*. Maksud dari sistem rekomendasi waktu nyata ini adalah rekomendasi diberikan *on the fly*, yaitu berdasarkan data terbaru dan diberikan secara *real time* pada saat pengguna berinteraksi dengan sistem. Dengan sistem rekomendasi waktu nyata atau *real time*, waktu yang dibutuhkan untuk menyediakan rekomendasi bagi pengguna dapat dilakukan dengan lebih cepat. Sistem rekomendasi *real time* ini memungkinkan barang direkomendasikan pada saat pengguna sedang *browsing* untuk memilih barang, misalkan dengan merekomendasikan barang yang serupa dengan barang yang baru saja dibeli atau barang yang baru dimasukkan ke dalam keranjang.

Salah satu teknologi yang dapat dimanfaatkan untuk mengembangkan sistem rekomendasi waktu nyata adalah teknologi sistem manajemen basis data berbasis graf atau yang biasa dikenal dengan istilah DBMS berbasis graf seperti Neo4j. Neo4j merupakan salah satu teknologi basis data graf populer yang banyak dimanfaatkan dalam pembuatan sistem rekomendasi. Dengan menggunakan basis data graf, operasi yang kompleks dapat dilakukan dengan lebih cepat dan efisien. Hal ini disebabkan karena basis data graf menyimpan data dalam bentuk graf dimana hubungan antar data tersebut telah tersimpan sehingga tidak diperlukan lagi operasi *join* ketika data dibutuhkan seperti pada basis data relasional. Selain dapat melakukan komputasi dengan cepat, Neo4j banyak diminati karena kemudahannya dalam melakukan kueri untuk mengakses data dalam basis data graf, yaitu dengan memanfaatkan bahasa Cypher.

Untuk membuat sistem rekomendasi secara *real time*, dikembangkan beberapa model rekomendasi dengan memanfaatkan konsep rekomendasi pada bisnis, yaitu konsep *cross-selling* dan *up-selling*, konsep rekomendasi berdasarkan pertemanan, dan konsep rekomendasi berdasarkan *influencer* favorit. Model rekomendasi ini dibuat dengan memanfaatkan data-data yang telah dimodelkan ke dalam bentuk graf. Rekomendasi waktu nyata kemudian diberikan dengan memanfaatkan model rekomendasi yang dibangun tersebut. Model rekomendasi kemudian dievaluasi dengan membandingkan hasil rekomendasi yang diberikan sistem dengan hasil visualisasi graf secara manual. Selain itu, pada akhir penelitian ini juga dibangun sebuah prototipe sistem belanja *online* berbasis *website* dengan fitur-fitur sederhana untuk meluncurkan model rekomendasi yang dibuat.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dibahas pada bagian sebelumnya, berikut merupakan masalah-masalah yang akan diselesaikan:

1. Bagaimana cara kerja teknologi sistem manajemen basis data (DBMS) berbasis graf Neo4?
2. Bagaimana metode pembuatan sistem rekomendasi waktu nyata berbasis graf?
3. Bagaimana cara memanfaatkan teknologi sistem manajemen basis data (DBMS) berbasis graf Neo4 dalam membangun prototipe sistem rekomendasi waktu nyata?
4. Bagaimana cara mengumpulkan (membuat data sintetis) dan menyiapkan data yang diperlukan untuk membangun model graf?
5. Bagaimana cara menggunakan bahasa Cypher untuk melakukan kueri pada basis data graf?
6. Bagaimana cara memanfaatkan hasil analisis graf pada prototipe sistem belanja *online* berbasis *website* yang memberikan rekomendasi secara *real time*?

1.3 Tujuan

Berdasarkan masalah-masalah yang telah dipaparkan sebelumnya, tujuan pembuatan skripsi ini adalah sebagai berikut:

1. Memahami cara kerja teknologi sistem manajemen basis data (DBMS) berbasis graf Neo4.
2. Memahami metode pembuatan sistem rekomendasi waktu nyata berbasis graf.
3. Memahami cara memanfaatkan teknologi sistem manajemen basis data (DBMS) berbasis graf Neo4 dalam membangun prototipe sistem rekomendasi waktu nyata.
4. Mengumpulkan (membuat data sintetis) dan menyiapkan data yang diperlukan untuk membangun model graf.
5. Mempelajari dan memanfaatkan bahasa Cypher untuk melakukan kueri pada basis data graf.
6. Membangun prototipe sistem belanja *online* berbasis *website* sederhana untuk meluncurkan hasil analisis graf yang memberikan rekomendasi secara *real time*.

1.4 Batasan Masalah

Terdapat beberapa batasan masalah dalam pengerjaan skripsi ini, yaitu sebagai berikut:

1. Karena data nyata dari berbagai organisasi ritel tidak dapat diperoleh (dengan alasan kerahasiaan), data yang digunakan untuk membangun model graf berupa data sintetis yang dibuat berdasarkan hasil survei struktur data riil.
2. Prototipe sistem belanja *online* yang dibangun difokuskan untuk meluncurkan hasil analisis graf yang dibuat sehingga tidak disertai fitur-fitur lain seperti fitur yang terkait dengan transaksi belanja *online*.

1.5 Metodologi

Metodologi yang digunakan dalam pembuatan skripsi ini adalah sebagai berikut:

1. Melakukan studi literatur mengenai sistem rekomendasi.
Topik utama yang akan dibahas pada skripsi ini adalah mengenai sistem rekomendasi sehingga tahap pertama yang dilakukan adalah melakukan studi literatur mengenai sistem rekomendasi.
2. Melakukan studi literatur mengenai konsep graf.
Salah satu komponen utama dalam pembuatan sistem rekomendasi adalah basis data graf. Untuk membangun basis data graf, teori graf perlu dipahami terlebih dahulu. Oleh karena itu, tahap yang harus dilakukan setelah studi literatur mengenai sistem rekomendasi adalah melakukan studi literatur mengenai konsep graf.
3. Melakukan studi literatur mengenai konsep basis data graf.
Setelah memahami teori graf, tahap selanjutnya adalah melakukan studi literatur untuk memahami konsep basis data graf, cara kerjanya, dan cara memodelkan basis data graf.
4. Melakukan studi literatur mengenai teknologi basis data berbasis graf Neo4j.
Sebelum membangun sistem rekomendasi dengan memanfaatkan teknologi sistem manajemen basis data (DBMS) berbasis graf Neo4, dibutuhkan pemahaman terkait cara kerja Neo4j dan arsitekturnya. Oleh karena itu, perlu dilakukan studi literatur mengenai Neo4j.
5. Melakukan studi literatur mengenai bahasa Cypher yang merupakan bahasa kueri pada basis data graf.
Studi literatur mengenai bahasa Cypher juga perlu dilakukan agar dapat memahami cara mengakses data yang tersimpan dalam basis data graf.
6. Melakukan studi literatur mengenai cara memproses data di Neo4j dengan menggunakan bahasa Cypher.
Untuk lebih memahami bahasa Cypher yang telah dipelajari sebelumnya, dilakukan studi literatur mengenai penggunaan bahasa Cypher pada Neo4j. Hal ini dilakukan dengan mencoba mempraktikkan bahasa Cypher pada Neo4j.
7. Membuat data sintesis berdasarkan struktur data riil dan menyiapkan data yang diperlukan untuk membangun model graf.
Untuk membuat sistem rekomendasi, diperlukan beberapa data seperti data produk, data pengguna, data pembelian, dan data pertemanan. Oleh karena itu, sebelum membuat model perlu dikumpulkan terlebih dahulu data-data yang dibutuhkan.
8. Memodelkan data yang telah dikumpulkan ke dalam bentuk graf.
Data yang telah dikumpulkan harus dimodelkan ke dalam bentuk graf yang nantinya akan digunakan untuk memberikan rekomendasi. Pembuatan model graf ini dilakukan dalam 2 tahap, yaitu :
 - (a) Pembuatan model graf dengan menggunakan data sintesis yang berukuran kecil. Tahap ini dilakukan sebagai percobaan untuk lebih memahami cara pembuatan model graf.
 - (b) Pembuatan model graf dengan menggunakan data sintesis dengan ukuran data relatif besar. Model graf yang dibuat pada tahap inilah yang akan digunakan untuk memberikan rekomendasi pada prototipe sistem penjualan online sederhana.
9. Melakukan evaluasi model graf yang telah dibangun.
Model rekomendasi yang telah dibangun harus dievaluasi untuk mendapatkan hasil rekomendasi yang terbaik. Evaluasi model graf ini dilakukan dalam 2 tahap, yaitu :
 - (a) Evaluasi model graf yang menggunakan data sintesis yang berukuran kecil.
 - (b) Evaluasi model graf yang menggunakan data sintesis dengan ukuran data relatif besar.
10. Membangun perangkat lunak, yaitu prototipe sistem penjualan *online* sederhana yang memberikan rekomendasi secara *real time*.
Sebelum digunakan pada sistem yang sebenarnya, performa model rekomendasi harus diuji terlebih dahulu pada sistem yang sederhana untuk memastikan bahwa sistem rekomendasi dapat berjalan dengan baik.

11. Membuat dokumentasi pengembangan sistem rekomendasi waktu nyata.
Seluruh proses dan hasil yang didapatkan pada proses pengembangan sistem rekomendasi waktu nyata ini perlu didokumentasikan. Dokumentasi ini akan sangat berguna khususnya untuk mengembangkan hasil penelitian ini.

1.6 Sistematika Pembahasan

Sistematika penulisan skripsi ini adalah sebagai berikut:

- Bab 1 Pendahuluan
Bab ini membahas mengenai latar belakang dibuatnya skripsi, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan.
- Bab 2 Landasan Teori
Bab ini membahas berbagai teori yang dibutuhkan sebagai dasar pengerjaan skripsi ini, seperti teori sistem rekomendasi, teori graf, teori basis data, teori basis data graf, teori Neo4j, teori bahasa Cypher, teori *library* Graph Data Science pada Neo4j, teori *cross-selling* dan *up-selling*, serta teori mengenai influencer.
- Bab 3 Studi Eksplorasi
Bab ini membahas mengenai percobaan penggunaan bahasa Cypher pada Neo4j serta proses dan hasil pembuatan model rekomendasi menggunakan data sintetis yang berukuran kecil.
- Bab 4 Pembuatan Model Rekomendasi dengan Analisis Graf
Bab ini membahas mengenai proses pembuatan model rekomendasi yang diawali dengan pembuatan data sintetis yang berukuran besar, pemodelan data ke dalam bentuk graf, eksplorasi data, pembuatan model rekomendasi dengan analisis graf, dan terakhir adalah evaluasi hasil rekomendasi.
- Bab 5 Pembangunan Perangkat Lunak dan Pengujian
Bab ini membahas mengenai implementasi model rekomendasi pada perangkat lunak prototipe sistem belanja *online* yang dibangun berserta evaluasi hasil rekomendasi yang diberikan.
- Bab 6 Kesimpulan dan Saran
Bab ini membahas mengenai kesimpulan dan saran terkait penelitian yang dilakukan pada skripsi ini.

