



TUGAS AKHIR - KI091391

IMPLEMENTASI LATENT SEMANTIC INDEXING DAN ALGORITMA 3A (ACTOR, ASSET, ACTIVITIES) UNTUK SISTEM REKOMENDASI LOWONGAN PEKERJAAN

ADHI NURILHAM
NRP 5112 100 164

Dosen Pembimbing
Diana Purwitasari, S.Kom., M.Sc.
Dini Adni Navastara, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR – KI091391

IMPLEMENTASI LATENT SEMANTIC INDEXING DAN ALGORITMA 3A (ACTOR, ASSET, ACTIVITIES) UNTUK SISTEM REKOMENDASI LOWONGAN PEKERJAAN

ADHI NURILHAM
NRP 5112 100 164

Dosen Pembimbing
Diana Purwitasari, S.Kom., M.Sc.
Dini Adni Navastara, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



FINAL PROJECT – KI091391

**IMPLEMENTATION OF LATENT SEMANTIC INDEXING
AND 3A (ACTOR, ASSET, ACTIVITIES) ALGORITHM
FOR JOB RECOMMENDER SYSTEM**

ADHI NURILHAM
NRP 5112 100 164

Advisor
Diana Purwitasari, S.Kom., M.Sc.
Dini Adni Navastara, S.Kom., M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

[Halaman ini sengaja dikosongkan]

**IMPLEMENTASI LATENT SEMANTIC INDEXING DAN
ALGORITMA 3A (ACTOR, ASSET, ACTIVITIES) UNTUK
SISTEM REKOMENDASI LOWONGAN PEKERJAAN**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

ADHI NURILHAM

NRP : 5112 100 164

Disetujui oleh Dosen Pembimbing

Diana Purwitasari, S.Kom., M.Eng.

NIP: 197804102003122001

Dini Adni Navastara, S.Kom, M.Sc.

NIP: 198510172015042001



(pembimbing 1)

(pembimbing 2)

**SURABAYA
JUNI 2016**

[Halaman ini sengaja dikosongkan]

IMPLEMENTASI LATENT SEMANTIC INDEXING DAN ALGORITMA 3A (ACTOR, ASSET, ACTIVITIES) UNTUK SISTEM REKOMENDASI LOWONGAN PEKERJAAN

Nama Mahasiswa : ADHI NURILHAM
NRP : 5112 100 164
Jurusan : Teknik Informatika ITS
Dosen Pembimbing I : Diana Purwitasari, S.Kom., M.Sc.
Dosen Pembimbing II : Dini Adni Navastara, S.Kom., M.Sc.

Abstrak

Terdapat ribuan informasi lowongan pekerjaan yang dapat diakses melalui internet. Implementasi sistem rekomendasi pekerjaan dapat memudahkan pelamar dalam memberikan penilaian terhadap ribuan informasi lowongan pekerjaan. Sistem rekomendasi pada umumnya berjenis content-based atau collaborative. Sistem tersebut masih memiliki kelemahan. Oleh karena itu, pada Tugas Akhir ini akan diajukan sistem rekomendasi hybrid yang menggabungkan jenis sistem rekomendasi content-based dan collaborative untuk mencari pekerjaan.

Terdapat beberapa tahapan dalam proses sistem rekomendasi lowongan pekerjaan ini. Tahap pertama adalah proses pra-pemrosesan teks pada dokumen teks profil pelamar dan pekerjaan. Tahap kedua adalah pencarian kesamaan teks pada kumpulan teks profil pelamar dan pekerjaan menggunakan metode LSI (Latent Semantic Indexing). Tahap ketiga adalah pemodelan hubungan kumpulan pelamar dan pekerjaan dalam graf. Tahap keempat adalah pengukuran tingkat rank node pada graf model hubungan objek rekomendasi menggunakan Algoritma 3A.

Uji coba pada tugas akhir ini menggunakan data pelamar, data pekerjaan, dan data feedback pelamar terhadap pekerjaan. Uji coba dilakukan dengan memberikan rekomendasi pekerjaan pada sejumlah pelamar menggunakan spesifikasi nilai parameter Algoritma 3A dan jenis rekomendasi yang berbeda. Hasil

pengujian menunjukkan bahwa sistem rekomendasi pekerjaan ini dapat memberikan nilai presisi sebesar 52,37%.

Kata kunci: *Text Mining, Sistem Rekomendasi, Graf, Node Ranking, Latent Semantic Indexing, Algoritma 3A.*

IMPLEMENTATION OF LATENT SEMANTIC INDEXING AND 3A (ACTOR, ASSET, ACTIVITIES) ALGORITHM FOR JOB RECOMMENDER SYSTEM

Name : ADHI NURILHAM
NRP : 5112 100 164
Major : Informatics Engineering Department – ITS
Supervisor I : Diana Purwitasari, S.Kom., M.Sc.
Supervisor II : Dini Adni Navastara, S.Kom, M.Sc.

Abstract

There are thousands of job information that can be accessed via the internet. Implementation of recommendation systems can facilitate job applicants in providing an assessment of thousands of job information.. Recommender system usually use content-based system or collaborative system. These systems still have flaws. Therefore, this final project will propose hybrid recommender system that combine content-based and collaborative system to find a job.

There are several phases in the process of this job recommender system. The first phase is the pre-processing text in the text documents of job and applicant profile. The second phase is to find similarity between the collection of text documents of job and applicant profile using LSI (Latent Semantice Indexing). The third phase is a construction of graph that model the relationship between collection of job and applicant. The fourth phase is to rank the importance of job and applicant in the graph using 3A Algorithm.

The testing phase will use applicant data, job data, and applicant's feedback data towards the jobs. The test is done by providing job recommendations on a number of applicants using different parameter value spesifications in 3A Algorithm and different type of recommender systems. Result of the tests show that this job recommender system can give us the precision score of 52,37%..

Keywords: Text Mining, Recommender System, Graph, Node Ranking, Latent Semantic Indexing, 3A Algorithm.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “*Implementasi Latent Semantic Indexing dan Algoritma 3A (Actor, Asset, Activites) untuk Sistem Rekomendasi Lowongan Pekerjaan*”.

Dalam pelaksanaan Tugas Akhir ini tentu penulis sebagai makhluk sosial tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, penulis memberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Orang tua, saudara serta keluarga besar di kampung halaman yang senantiasa memberikan semangat dan doa agar penulis dapat menyelesaikan Tugas Akhir dengan tepat waktu.
2. Ibu Diana Purwitasari, S.Kom, M.Sc selaku dosen pembimbing Tugas Akhir pertama yang telah membimbing, memotivasi dan memberikan banyak masukan dalam pengerjaan Tugas Akhir ini.
3. Ibu Dini Adni Navastara, S.Kom, M.Sc. selaku dosen pembimbing Tugas Akhir kedua yang selalu memberikan masukan-masukan yang dapat penulis kembangkan dari Tugas Akhir ini.
4. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah mengajarkan banyak ilmu berharga kepada penulis.
5. Bapak dan Ibu karyawan Jurusan Teknik Informatika ITS atas berbagai bantuan yang telah diberikan kepada penulis selama masa perkuliahan.

6. Teman-teman administrator Laboratorium Komputasi Cerdas dan Visi atau dulu dikenal sebagai Laboratorium IBS yang telah menemani perjuangan mencari ilmu selama mengambil mata kuliah RMK KCV.
7. Teman-teman Teknik Informatika ITS angkatan 2012, yang telah memberikan warna-warni kehidupan mahasiswa mulai sejak mahasiswa baru hingga lulus.
8. Pihak-pihak lain yang tidak sempat penulis sebutkan, yang telah membantu kelancaran pengerjaan Tugas Akhir ini.

Penulis sangat berharap bahwa apa yang dihasilkan dari Tugas Akhir ini bisa memberikan manfaat bagi semua pihak, khususnya bagi diri penulis sendiri dan seluruh *civitas academica* Teknik Informatika ITS, serta bagi agama, bangsa, dan negara. Tak ada manusia yang sempurna sekalipun penulis berusaha sebaik mungkin dalam menyelesaikan Tugas Akhir ini. Karena itu, penulis memohon maaf apabila terdapat kesalahan, kekurangan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun sangat diharapkan oleh penulis untuk dapat disampaikan untuk perbaikan selanjutnya.

Surabaya, Juni 2016

Adhi Nurilham

DAFTAR ISI

Abstrak	vii
Abstract	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
1 BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan	3
1.5. Manfaat.....	3
1.6. Metodologi	4
1.7. Sistematika Penulisan.....	5
2 BAB II TINJAUAN PUSTAKA	7
2.1. Data Masukan Sistem Rekomendasi	7
2.2. Sistem Rekomendasi Pekerjaan	11
a. <i>Content-Based</i>	11
b. <i>Collaborative</i>	12
c. <i>Hybrid</i>	12
2.3. <i>Node Ranking</i> pada Graf Menggunakan Algoritma 3A (<i>Actor, Asset, Activities</i>).....	12

2.4.	Kemiripan Dokumen Teks Menggunakan Metode LSI (<i>Latent Semantic Indexing</i>)	19
2.5.	Perhitungan Nilai Presisi Sistem Rekomendasi	21
3	BAB III ANALISIS DAN PERANCANGAN.....	23
3.1.	Analisis Implementasi Metode Secara Umum.....	23
3.2.	Perancangan Data.....	25
3.3.	Perancangan Proses	30
3.4.	Perancangan Antarmuka Sistem Rekomendasi.....	43
4	BAB IV IMPLEMENTASI.....	45
4.1.	Lingkungan Implementasi	45
4.2.	Implementasi Proses.....	45
4.3.	Implementasi Antarmuka Pengguna	56
5	BAB V UJI COBA DAN EVALUASI	63
5.1.	Lingkungan Uji Coba	63
5.2.	Cara Evaluasi Kinerja Sistem Rekomendasi	63
5.3.	Data Uji Coba	67
5.4.	Skenario Uji Coba.....	68
5.5.	Skenario Pengujian 1: Perbandingan nilai presisi rekomendasi berdasarkan jenis sistem rekomendasi yang berbeda (<i>content-based</i> , <i>collaborative</i> , dan <i>hybrid</i>)	69
5.6.	Skenario Pengujian 2: Perbandingan nilai <i>rank node</i> , berdasarkan komposisi nilai parameter yang berbeda.	70
5.7.	Skenario Pengujian 3: Perbandingan nilai presisi rekomendasi, berdasarkan komposisi nilai parameter yang berbeda - 1.....	76
5.8.	Skenario Pengujian 4: Perbandingan nilai presisi rekomendasi, berdasarkan komposisi nilai parameter yang berbeda - 2.....	78

5.9.	Skenario Pengujian 5: Perbandingan Rata – Rata Nilai Presisi Rekomendasi Pada Setiap Kelas pelamar Menggunakan Jumlah Data Pelamar yang Berbeda.	80
6	BAB VI KESIMPULAN DAN SARAN.....	87
6.1.	Kesimpulan.....	87
6.2.	Saran.....	88
7	DAFTAR PUSTAKA.....	89
8	Lampiran 1	91
9	Lampiran 2	97
10	Lampiran 3.....	110
	BIODATA PENULIS	111

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Contoh Graf	14
Gambar 2.2 Ilustrasi Ukuran Matriks Hasil SVD	20
Gambar 3.1 Diagram Alir Sistem Rekomendasi Pekerjaan	24
Gambar 3.2 <i>Screenshot</i> Website www.indeed.com	25
Gambar 3.3 <i>Screenshot</i> Website www.import.io	26
Gambar 3.4 Diagram Alir Tahap Pra-Pemrosesan Teks.....	30
Gambar 3.5 Diagram Alir Tahap Pengukuran Kemiripan Teks..	31
Gambar 3.6 Contoh Matriks <i>Term-Document</i>	34
Gambar 3.7 Contoh Hasil Normalisasi Matriks <i>Term-Document</i>	34
Gambar 3.8 Contoh Hasil Dekomposisi Matriks <i>Term-Document</i>	35
Gambar 3.9 Contoh <i>Vector</i> Representasi Dokumen Teks Data ..	35
Gambar 3.10 Contoh Perhitungan <i>Cosine Distance</i>	36
Gambar 3.11 Contoh Graf Keluaran Tahap Konstruksi Graf	37
Gambar 3.12 Diagram Alir Tahap <i>Node Ranking</i> pada Graf.....	38
Gambar 3.13 Contoh Hasil <i>Vector Rank (R)</i>	40
Gambar 4.1 Implementasi Penyusunan Atribut Antarmuka Sistem Pada QtDesigner.....	56
Gambar 4.2 Menu File Masukan.....	57
Gambar 4.3 Halaman Interaksi Pelamar	58
Gambar 4.4 Halaman Rekomendasi Pekerjaan	59
Gambar 5.1 Hasil Graf Skenario 2	72
Gambar 5.2 Rata - Rata Nilai Presisi Skenario 4	82
Gambar 5.3 Rata - Rata Nilai Presisi Skenario 5	85
Gambar 10.1 Contoh Matriks <i>Walk</i>	110

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Contoh Data Pelamar	8
Tabel 2.2 Contoh Data Pekerjaan.....	9
Tabel 2.3 Contoh Data <i>Feedback</i> Pelamar Terhadap Pekerjaan .	11
Tabel 2.4 Relasi Graf.....	13
Tabel 2.5 Deskripsi Parameter Algoritma <i>Node Ranking</i> 3A.....	15
Tabel 3.1 Hasil Klasifikasi Manual Data Masukan Pelamar dan Pekerjaan	27
Tabel 3.2 Klasifikasi Data Masukan Pelamar dan Pekerjaan yang Dipakai	28
Tabel 3.3 Data Keluaran	29
Tabel 3.4 Contoh Data Masukan Tahap Pengukuran Kemiripan Teks	33
Tabel 3.5 Contoh Data Keluaran Tahap <i>Node Ranking</i>	41
Tabel 3.6 Contoh Hasil Rekomendasi Pekerjaan	42
Tabel 4.1 Tabel Spesifikasi Lingkungan Implementasi.....	45
Tabel 5.1 Spesifikasi Lingkungan Uji Coba	63
Tabel 5.2 Data Pelamar <i>JC-19</i>	64
Tabel 5.3 Hasil Rekomendasi Pekerjaan untuk Pelamar <i>JC-19</i> ..	65
Tabel 5.4 Jumlah Data Uji Coba	68
Tabel 5.5 Perbedaan Jenis Sistem Rekomendasi.....	70
Tabel 5.6 Hasil Nilai Presisi Skenario 2	70
Tabel 5.7 Komposisi Nilai Parameter Skenario 2	71
Tabel 5.8 Keluaran Komposisi 1 pada Skenario 2	73
Tabel 5.9 Keluaran Komposisi 2 pada Skenario 2	74
Tabel 5.10 Keluaran Komposisi 3 pada Skenario 2	75
Tabel 5.11 Ringkasan Hasil Analisa Parameter Skenario 1	76
Tabel 5.12 Hasil Nilai Presisi Skenario 3	77
Tabel 5.13 Komposisi Nilai Parameter Skenario 4	79
Tabel 5.14 Hasil Nilai Presisi Skenario 4	80
Tabel 5.15 Rata - Rata Nilai Presisi Skenario 4	81
Tabel 5.16 Set Data Pelamar Skenario 5	83
Tabel 5.17 Rata - Rata Nilai Presisi Skenario 5 Setiap Set.....	84

Tabel 8.1 Spesifikasi Atribut Antarmuka File Masukan	91
Tabel 8.2 Spesifikasi Atribut Halaman Interaksi Pelamar	92
Tabel 8.3 Spesifikasi Atribut Halaman Rekomendasi Pekerjaan	94
Tabel 9.1 Contoh Data Uji Coba Pelamar	97
Tabel 9.2 Contoh Data Uji Coba Pekerjaan	99
Tabel 9.3 Contoh Data Uji Coba <i>Feedback</i>	109

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Pra-Pemrosesan Teks	46
Kode Sumber 4.2 Implementasi Pembuatan Matriks <i>Term-Document</i>	47
Kode Sumber 4.3 Implementasi Dekomposisi Matriks	47
Kode Sumber 4.4 Implementasi Pembuatan <i>Vector</i> Dokumen...	48
Kode Sumber 4.5 Implementasi Perhitungan Jarak <i>Vector</i> Dokumen	48
Kode Sumber 4.6 Implementasi Matriks <i>Adjacency</i> Relasi <i>Similar</i>	49
Kode Sumber 4.7 Implementasi Matriks <i>Adjacency</i> Relasi <i>Like</i>	50
Kode Sumber 4.8 Implementasi Matriks <i>Adjacency</i> Relasi <i>Apply</i>	51
Kode Sumber 4.9 Implementasi Pembuatan Matriks Transition & Matriks Dangling Node untuk Setiap Jenis Relasi Graf (Bagian 1)	52
Kode Sumber 4.10 Implementasi Pembuatan Matriks <i>Transition</i> & Matriks <i>Dangling Node</i> untuk Setiap Jenis Relasi Graf (Bagian 2)	53
Kode Sumber 4.11 Implementasi Pembuatan Matriks Representasi <i>Random-Walk</i>	54
Kode Sumber 4.12 Implementasi Pembuatan <i>Vector Rank Node</i>	55
Kode Sumber 4.13 Implementasi Pengurutan Elemen <i>Vector Rank</i>	55
Kode Sumber 4.14 Implementasi <i>Import</i> Antarmuka pada Sistem	56

[Halaman ini sengaja dikosongkan]

[Halaman ini sengaja dikosongkan]

BAB I PENDAHULUAN

Bab ini membahas garis besar penyusunan Tugas Akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Bekerja merupakan aktivitas utama seorang individu untuk dapat mencari nafkah dan berkontribusi kepada masyarakat. Memiliki pekerjaan merupakan suatu keharusan agar seseorang dapat bertahan hidup. Namun terkadang pelamar membutuhkan waktu yang tidak sebentar untuk bisa menemukan pekerjaan yang sesuai. Meskipun informasi lowongan pekerjaan dapat mudah dicari melalui internet, pelamar tetap harus melihat setiap informasi lowongan pekerjaan yang ada untuk dapat memberikan penilaian dan menemukan pekerjaan yang sesuai. Waktu mencari informasi lowongan pekerjaan akan dapat dipersingkat jika penilaian dan pemilihan informasi lowongan pekerjaan yang sesuai dilakukan oleh suatu sistem pada komputer.

Sistem rekomendasi merupakan sebuah sistem yang dapat memberikan suatu prediksi kemungkinan *item* yang disukai oleh pengguna. Pelamar dapat menggunakan sistem rekomendasi untuk mencari pekerjaan yang sesuai dalam waktu yang singkat. Pada umumnya, sistem rekomendasi yang telah ada berjenis *content-based* atau *collaborative* [1]. Sistem rekomendasi *content-based* bekerja dengan mencari kesamaan antara atribut pada profil pelamar dan deskripsi pekerjaan. Sistem rekomendasi *collaborative* memberikan rekomendasi sesuai dengan preferensi atau selera pelamar terhadap pekerjaan. Perbedaan antara sistem rekomendasi *content-based* dan *collaborative* adalah sistem rekomendasi *collaborative* memerlukan interaksi pelamar pada sistem untuk memberikan umpan balik (*feedback*) terhadap suatu

pekerjaan, sedangkan sistem rekomendasi *content-based* tidak memerlukan interaksi pelamar karena hanya bergantung pada profil pelamar dan pekerjaan.

Namun, sistem rekomendasi *content-based* dan *collaborative* masih memiliki kelemahan. Salah satu kekurangan dari rekomendasi *content-based* adalah *over-specialization* [1] dimana rekomendasi yang diberikan tidak beragam. Hal tersebut disebabkan karena sistem rekomendasi ini hanya mengacu pada konten profil pelamar dan pekerjaan saja. Sistem rekomendasi *collaborative* memiliki masalah *cold start*, dimana rekomendasi yang diberikan tidak maksimal pada awal mulainya sistem [2]. Hal ini disebabkan karena masih belum ada interaksi pelamar pada sistem.

Oleh karena itu dikembangkan sistem rekomendasi *hybrid* yang menggabungkan sistem rekomendasi *content-based* dan *collaborative*. Sistem rekomendasi *hybrid* ini terdiri dari beberapa tahap: 1) Tahap pra-pemrosesan teks dokumen; 2) Tahap pengukuran kemiripan entitas (pelamar dan pekerjaan); 3) Tahap konstruksi graf; 4) Tahap *node ranking* pada graf. Sistem rekomendasi *hybrid* ini diharapkan dapat mengatasi kelemahan dari masing - masing sistem rekomendasi *content-based* dan *collaborative* dan menghasilkan rekomendasi yang lebih akurat.

1.2. Rumusan Masalah

Rumusan masalah pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara melakukan ekstraksi fitur teks?
2. Bagaimana cara menghasilkan rekomendasi lowongan pekerjaan yang sesuai untuk pelamar?
3. Bagaimana cara melakukan evaluasi terhadap hasil rekomendasi pekerjaan?

1.3. Batasan Masalah

Batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Sistem rekomendasi yang dibuat hanya memberikan rekomendasi pekerjaan kepada pelamar.
2. Sistem rekomendasi dibuat berbasis desktop dengan menggunakan bahasa pemrograman Python.
3. Sistem rekomendasi akan menggunakan beberapa *class library*, yaitu :
 - Scipy
 - Numpy
 - Sklearn
 - Textmining
 - NLTK
 - PyQt
4. Data yang digunakan dalam sistem rekomendasi yaitu :
 - Data pelamar yang didapatkan melalui metode *web scrapping* pada website www.indeed.com [3].
 - Data lowongan pekerjaan yang didapatkan pada website nycopendata.socrata.com [4].
 - Data *feedback* pelamar terhadap pekerjaan dibuat secara manual oleh penulis.
5. Data yang digunakan menggunakan teks dalam Bahasa Inggris.

1.4. Tujuan

Tujuan pembuatan Tugas Akhir ini adalah untuk membuat sistem yang dapat memberikan rekomendasi pekerjaan yang sesuai untuk pelamar dengan menggunakan Algoritma 3A (*Actor, Asset, Activites*) dan metode *Latent Semantic Indexing*.

1.5. Manfaat

Manfaat pembuatan Tugas Akhir ini adalah untuk membantu pelamar dalam memberikan penilaian terhadap informasi lowongan pekerjaan.

1.6. Metodologi

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal tugas akhir

Di dalam proposal tugas akhir ini dijelaskan mengenai deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan berisi dari latar belakang alasan adanya usulan tugas akhir, rumusan masalah yang dibawa, batasan-batasan permasalahan, tujuan serta manfaat dari pembuatan tugas akhir ini. Sebagai bahan referensi pendukung dari usulan tugas akhir yang akan dibuat, maka dijelaskan pula tinjauan pustaka. Selain itu, terdapat metodologi sebagai langkahlangkah dalam pembuatan tugas akhir, dimulai dari penyusunan proposal hingga penyusunan buku tugas akhir dan sebagai haluan jadwal pelaksanaan tugas akhir dilampirkan pula jadwal kegiatan. Pada proposal ini, penulis mengajukan gagasan mengenai implementasi sistem rekomendasi pekerjaan berjenis *hybrid* menggunakan Algoritma 3A dan metode LSI (*Latent Semantic Indexing*).

2. Studi literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur terkait beberapa teori pendukung serta pustaka pendukungnya yang antara lain cara pengambilan data, sistem rekomendasi *hybrid*, algoritma pengukuran (*ranking*) 3A, kemiripan teks dengan metode LSI.

3. Implementasi

Pada tahap ini dilakukan implementasi aplikasi pada sistem rekomendasi dan dibangun dengan bahasa pemrograman Python. Versi Python yang digunakan adalah Python 2.7 dengan pustaka antarmuka PyQt.

4. Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba aplikasi dan evaluasi terhadap implementasi metode pada aplikasi. Tahap uji coba dilakukan dengan mencari tahu pengaruh parameter – parameter dalam Algoritma 3A pada hasil rekomendasi dan mencoba proses rekomendasi dengan jenis sistem rekomendasi yang berbeda. Tahap evaluasi dilakukan dengan menghitung tingkat presisi hasil rekomendasi pekerjaan untuk pelamar.

5. Penyusunan Buku Tugas Akhir

Tahap ini merupakan tahap dokumentasi dari tugas akhir. Buku tugas akhir berisi dasar teori, perancangan, implementasi dan hasil uji coba dan evaluasi dari sistem rekomendasi yang dibangun.

1.7. Sistematika Penulisan

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. Bab I. Pendahuluan
Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan Tugas Akhir.
2. Bab II. Tinjauan Pustaka
Bab tinjauan pustakan berisi penjelasan mengenai dasar teori yang mendukung pengerjaan Tugas Akhir.
3. Bab III. Analisis dan Perancangan

Bab analisis dan perancangan berisi penjelasan mengenai analisis kebutuhan, perancangan sistem dan perangkat yang digunakan dalam pengerjaan Tugas Akhir serta urutan pelaksanaan proses.

4. Bab IV. Implementasi

Bab implementasi berisi pembangunan implementasi sistem rekomendasi dengan Algoritma 3A sesuai dengan rumusan masalah dan batasan yang sudah dijelaskan pada bagian pendahuluan.

5. Bab V. Uji Coba dan Evaluasi

Bab uji coba dan evaluasi berisi pembahasan mengenai hasil dari uji coba yang dilakukan terhadap sistem rekomendasi pekerjaan.

6. Bab VI. Kesimpulan dan Saran

Bab kesimpulan dan saran berisi kesimpulan hasil penelitian. Selain itu, bagian ini berisi saran untuk pengerjaan lebih lanjut atau permasalahan yang dialami dalam proses pengerjaan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Bab tinjauan pustaka berisi mengenai penjelasan teori yang berkaitan dengan implementasi sistem rekomendasi. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai pendukung dalam pengembangan sistem rekomendasi.

2.1. Data Masukan Sistem Rekomendasi

Sistem rekomendasi yang akan dibuat memerlukan 3 jenis data masukan yaitu data pelamar, data pekerjaan, dan data *feedback* pelamar terhadap pekerjaan [2]. Data pelamar berisi deskripsi profil pelamar, data pekerjaan berisi deskripsi pekerjaan, dan data *feedback* pelamar terhadap pekerjaan berisi riwayat ketertarikan pelamar terhadap suatu pekerjaan. Setiap data pelamar dan pekerjaan akan dibagi ke dalam beberapa kelas yaitu kelas *management, IT, engineering, medical, supply-chain, dan legal*. Proses klasifikasi data pelamar dan pekerjaan akan dijelaskan pada Bab III. Berikut penjelasan 3 jenis data masukan :

a. Data pelamar

Data terdiri dari 5 bagian yaitu *Id, Name, Experience, Education, dan Degree*. Data pelamar didapatkan melalui proses *web crawling* pada website www.indeed.com [3]. Contoh data pelamar dapat dilihat pada Tabel 2.1. Pada contoh data tersebut dapat terlihat bahwa seorang pelamar bernama *Cheyenne Jenkins* dengan *id JC-1* memiliki pengalaman kerja sebagai *volunteer* di institusi *Greater Mount Carmel*. Pelamar tersebut memiliki gelar *bachelor* dari institusi pendidikan *Prepatory Academy for Writers*. Penjelasan setiap bagian adalah sebagai berikut :

- *Id* merupakan suatu penanda unik untuk setiap pelamar. *Id* untuk data pelamar memiliki awalan "*JC*" (*Job*

Candidate) diikuti dengan nomor sekuens data pelamar. Contoh *id* pelamar adalah “*JC-132*”.

- *Name* merupakan nama dari pelamar.
- *Experience* berisi deskripsi riwayat pengalaman kerja pelamar.
- *Education* berisi nama institusi pendidikan yang telah ditempuh pelamar.
- *Degree* adalah gelar pendidikan yang dimiliki pelamar.

Tabel 2.1 Contoh Data Pelamar

Id	Name	Experience	Education	Degree
JC-1	Cheyenne Jenkins	Volunteer - Greater Mount Carmel Deliverance Ministries;	Preparatory Academy for Writers	Bachelor

b. Data pekerjaan

Data terdiri dari 6 bagian yaitu *Id*, *Agency*, *Business Title*, *Job Description*, *Minimum Qualification*, dan *Preferred Skills*. Data pekerjaan didapatkan dari website [open dataset nycopendata.socrata.com](https://open.legacy.com) [4]. Contoh data pekerjaan dapat dilihat pada Tabel 2.2. Pada contoh data tersebut terlihat terdapat pekerjaan dengan *id JD-2* menawarkan posisi sebagai *Technology Solutions Manager* di institusi *Dept. of Info Tech & Telecom*. Pelamar yang akan melamar pekerjaan tersebut diharuskan untuk memiliki gelar *master in computer science* dan akan menjadi nilai tambah untuk pelamar jika memiliki pengalaman dalam *data* dan *systems architecture* minimal selama 5 tahun. Penjelasan setiap bagian adalah sebagai berikut :

- *Id* merupakan suatu pendanda unik untuk setiap pekerjaan. Id untuk data pekerjaan memiliki awalan "JD" (Job Description) diikuti dengan nomor sekuens data pekerjaan. Contoh id pekerjaan adalah “*JD-21*”.
- *Agency* merupakan nama perusahaan atau institusi yang menyediakan lowongan pekerjaan.
- *Business Title* adalah nama jabatan yang ditawarkan pada lowongan pekerjaan.
- *Job Description* adalah deskripsi pekerjaan yang akan dilakukan oleh pelamar.
- *Minimum Qualification* adalah deskripsi persyaratan keahlian minimal pada pekerjaan.
- *Preferred Skills* adalah deskripsi keahlian yang bisa menjadi nilai tambah dalam penerimaan pekerjaan.

Tabel 2.2 Contoh Data Pekerjaan

Id	Agency	Business Title	Job Description	Minimum Qualification	Preferred Skills
JD-1	DEPT OF ENVIRONMENT	Engineer	Environmental Protection (DEP) protects public health and the environment	A baccalaureate degree in civil engineering from an accredited college..	Familiar with soil mechanics, hydraulic analysis, borings, survey, ...

Id	Agency	Business Title	Job Description	Minimum Qualification	Preferred Skills
JD-2	DEPT OF INFO TECH & TELECOMM	MOTI Technology Solutions Manager	DoITT provides for the sustained, efficient and effective delivery of IT services ...	1. A master's degree in computer science from an accredited college and three years of progressively,	The successful candidate should possess the following: 5+ years of data and/or systems architecture, ...

c. *Data feedback*

Data *feedback* pelamar terhadap pekerjaan hanya terdiri dari 3 bagian yaitu *jid*, *relation*, dan *jid*. Ketertarikan pelamar terhadap suatu pekerjaan dapat direpresentasikan ke dalam 2 kategori yaitu *like* dan *apply*. Kategori *like* diberikan ketika pelamar tertarik terhadap suatu pekerjaan. Kategori *apply* diberikan ketika pelamar memutuskan untuk mencoba melamar pada suatu pekerjaan. Data ini dibuat sendiri oleh penulis dengan memasang data pelamar dan data pekerjaan dengan tanda ketertarikan *like* dan *apply*. Contoh data *feedback* pelamar terhadap pekerjaan dapat dilihat pada Tabel 2.3. Pada contoh data tersebut terlihat bahwa pelamar dengan *id JC-1* menyukai pekerjaan dengan *id JD-1* ditunjukkan dengan kolom *relation* yang berisi *Like*. Pelamar *JC-1* juga telah melamar pekerjaan dengan *id JD-2* ditunjukkan dengan kolom *relation* yang berisi *Apply*. Penjelasan setiap bagian adalah sebagai berikut :

- *jid* adalah id pelamar yang akan memberikan ketertarikannya terhadap suatu pekerjaan.
- *relation* adalah jenis ketertarikan yang akan diberikan pelamar terhadap suatu pekerjaan.

- *jdid* adalah id pekerjaan yang akan diberikan ketertarikan oleh pelamar.

Tabel 2.3 Contoh Data *Feedback* Pelamar Terhadap Pekerjaan

jid	relation	jdid
JC-1	Like	JD-1
JC-1	Apply	JD-2

2.2. Sistem Rekomendasi Pekerjaan

Sistem rekomendasi merupakan alat bantu dalam bentuk perangkat lunak untuk menyediakan rekomendasi untuk barang yang mungkin disukai oleh pengguna [1]. Barang yang direkomendasikan kepada pengguna dapat disebut sebagai *item*. Sistem rekomendasi pada dasarnya ditujukan untuk individual yang memiliki pengalaman yang kurang memadai dalam menilai *item* alternatif yang sangat banyak jumlahnya [1]. Pada sistem rekomendasi ini pengguna adalah pelamar, dan *item* yang akan direkomendasikan adalah pekerjaan. Jenis sistem rekomendasi dapat dibagi ke dalam jenis berikut :

a. *Content-Based*

Jenis sistem rekomendasi ini memberikan rekomendasi berdasarkan konten atau atribut yang dimiliki pelamar dan pekerjaan. Sistem rekomendasi ini merekomendasikan pekerjaan yang memiliki atribut yang sesuai dengan pelamar atau pekerjaan yang memiliki kesamaan atribut dengan pekerjaan lain yang sebelumnya telah disukai oleh pengguna [1]. Dalam hal ini, atribut yang akan dibandingkan berupa teks.

Pada Tugas Akhir ini sistem rekomendasi *content-based* akan memanfaatkan teks data profil pelamar dan data deskripsi

pekerjaan. Rekomendasi pekerjaan dibuat dengan mencari kemiripan antara kumpulan pelamar dan pekerjaan berdasarkan teks datanya [2].

b. Collaborative

Sistem rekomendasi ini memberikan rekomendasi pekerjaan yang disukai oleh pelamar lain yang memiliki selera yang sama dengan pelamar yang diberikan rekomendasi [1]. Kesamaan selera antar pelamar dicari melalui riwayat pekerjaan yang disukai oleh setiap pelamar. Oleh karena itu sistem rekomendasi ini bergantung pada informasi preferensi atau ketertarikan pelamar terhadap pekerjaan.

Pada Tugas Akhir ini informasi preferensi atau ketertarikan pelamar terhadap pekerjaan disimpan dalam data masukan *feedback* pelamar terhadap pekerjaan. Sistem rekomendasi *collaborative* akan memanfaatkan data *feedback* tersebut dalam mencari kesamaan ketertarikan terhadap pekerjaan antara satu pelamar dengan pelamar lainnya untuk proses pemberian rekomendasi.

c. Hybrid

Sistem rekomendasi ini merupakan gabungan dari sistem rekomendasi berjenis *content-based* dan *collaborative*. Menggabungkan jenis sistem rekomendasi yang berbeda dapat bermanfaat untuk melengkapi kelemahan antar jenis sistem rekomendasi [1]. Data pelamar, data pekerjaan, dan data *feedback* pelamar dan pekerjaan akan dipakai dalam proses pemberian rekomendasi pada sistem rekomendasi *hybrid*.

2.3. Node Ranking pada Graf Menggunakan Algoritma 3A (Actor, Asset, Activities)

Sebuah model graf akan digunakan untuk menggambarkan hubungan pelamar terhadap pekerjaan pada sistem rekomendasi ini. Graf yang akan dibuat berupa *multi-relational directed* graf,

artinya graf tersebut akan memiliki beberapa jenis relasi (*multi-relational*), dan *edge* yang menggabungkan *node* pada graf tersebut memiliki arah (*directed*).

Pelamar dan pekerjaan akan menjadi *node* dan relasi antara kedua entitas tersebut akan digambarkan melalui *edge* pada graf. *Node* yang mewakilkan pelamar akan diberikan nama sesuai dengan *id* pelamar pada data pelamar dan *node* yang mewakilkan pekerjaan akan diberikan nama sesuai dengan *id* pekerjaan pada data pekerjaan.

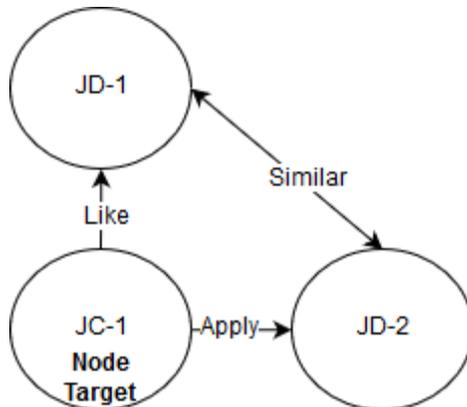
Tabel 2.4 Relasi Graf

No	Nama Relasi	Deskripsi	<i>Node</i> Sumber	<i>Node</i> Target
1	<i>Similar</i>	Menghubungkan dua entitas (pelamar/pekerjaan) yang memiliki kesamaan dalam konteks dokumen teks	Pelamar, Pekerjaan	Pelamar, Pekerjaan
2	<i>Like</i>	Menghubungkan pelamar ke pekerjaan yang disukainya	Pelamar	Pekerjaan
3	<i>Apply</i>	Menghubungkan pelamar ke pekerjaan yang telah dicoba untuk dilamar	Pelamar	Pekerjaan

Terdapat 3 jenis relasi yang akan menghubungkan *node* – *node* tersebut, yaitu relasi *similar*, *like*, dan *apply*. Penjelasan ketiga relasi tersebut dapat dilihat pada Tabel 2.4. Relasi *similar* merupakan relasi *bidirectional* yang dapat menghubungkan setiap jenis *node*, *node* pelamar atau *node* pekerjaan. Relasi ini dibuat jika terdapat dua *node* (pelamar/pekerjaan) yang memiliki kesamaan dalam dokumen teks datanya. Relasi *like* dan *apply* merupakan

relasi *directed* yang hanya menghubungkan *node* pelamar ke *node* pekerjaan. Relasi *like* dibuat jika terdapat pelamar yang menyukai suatu pekerjaan. Relasi *apply* dibuat jika terdapat pelamar yang mencoba untuk melamar suatu pekerjaan dalam sistem.

Contoh graf yang akan dibuat pada rekomendasi ini adalah seperti Gambar 2.1. Graf tersebut dibuat berdasarkan data pelamar, data pekerjaan, dan data *feedback* berturut - turut pada Tabel 2.1, Tabel 2.2, dan Tabel 2.3. Pada gambar tersebut pelamar yang memiliki *id* *JC-1* direpresentasikan oleh *node* *JC-1* menyukai pekerjaan dengan *id* *JD-1* karena memiliki masukan *edge* yang memiliki label *Like* dari pelamar *JC-1*. Pelamar *JC-1* juga telah mencoba untuk melamar pekerjaan dengan *id* *JD-2* karena terdapat keluaran *edge* yang memiliki label *Apply* dari *node* pelamar *JC-1* ke *node* pekerjaan *JD-2*. Pekerjaan *JD-2* memiliki kemiripan dengan pekerjaan *JD-1* karena terdapat *edge bidirectional* yang memiliki label *Similar* antara *node* *JD-2* dan *node* *JD-1*.



Gambar 2.1 Contoh Graf

Algoritma 3A (*Actor, Asset, Activities*) merupakan sebuah algoritma sistem rekomendasi yang menggunakan sebuah model graf yang memiliki 3 jenis *node* yaitu *actor*, *asset*, dan *activities* [5]. Namun, model graf tersebut tidak digunakan dalam sistem

rekomendasi ini. Bagian dari Algoritma 3A yang akan dipakai pada sistem rekomendasi ini adalah hanya pada proses *node ranking*-nya saja. Proses *node ranking* pada algoritma ini terinspirasi dari Algoritma *node ranking PageRank* yang merupakan algoritma pertama yang digunakan pada *search engine* Google [6].

Node ranking adalah sebuah proses untuk mengukur tingkat kepentingan *node* dalam sebuah graf, relatif terhadap *node target*. *Node target* adalah sebuah *node* spesifik yang dijadikan pusat pengukuran tingkat kepentingan seluruh *node* dalam graf. Pada sistem rekomendasi lowongan pekerjaan ini, *node target* mewakili *node* pelamar yang akan diberikan rekomendasi lowongan pekerjaan.

Algoritma *node ranking* 3A memanfaatkan metode *Random Walk* [7] yang merupakan metode untuk menjelajahi seluruh *node* pada graf. Pada sistem rekomendasi lowongan pekerjaan ini metode *Random Walk* menggambarkan proses penjelajahan informasi lowongan pekerjaan yang dilakukan pelamar di dalam sistem.

Tabel 2.5 Deskripsi Parameter Algoritma Node Ranking 3A

No	Simbol	Nama Parameter	Deskripsi
1	λ	<i>lamda</i>	kemungkinan pelamar menjelajahi sebuah halaman informasi lowongan pekerjaan secara acak (<i>random</i>)
2	d	<i>damping factor</i>	kemungkinan pelamar menjelajahi sebuah halaman informasi lowongan pekerjaan melalui <i>link</i> yang ada
3	Pu	<i>personalization factor</i>	Kemungkinan pelamar untuk kembali halaman awal

Terdapat 3 parameter pada metode *Random Walk* [5]. Ketiga parameter tersebut menggambarkan probabilitas atau kemungkinan pelamar dalam menjelajahi informasi - informasi lowongan pekerjaan yang terdapat di dalam sistem. Ketiga parameter tersebut adalah λ , d , dan P_u , seperti yang terlihat pada Tabel 2.5. Parameter λ (*lamda*) mewakili kemungkinan pelamar menjelajahi sebuah halaman informasi lowongan pekerjaan secara acak (*random*) di dalam sistem. Parameter d (*damping factor*) mewakili kemungkinan pelamar menjelajahi sebuah halaman informasi lowongan pekerjaan melalui *link* yang diberikan pada halaman sebelumnya di dalam sistem. Parameter P_u (*personalization factor*) mewakili kemungkinan pelamar kembali ke halaman awal saat pertama kali mengakses sistem.

Terdapat 2 tahap utama dalam Algoritma *node ranking* 3A ini sesuai dengan penggunaan metode *Random Walk* yaitu : pembuatan matriks *walk* dan iterasi *vector R*. Matriks *walk* adalah sebuah matriks yang merepresentasikan proses *Random Walk*. *Vector R* adalah sebuah *vector* yang setiap elemennya mewakili nilai *rank* atau nilai kepentingan *node* dalam graf.

Tahap pertama dalam Algoritma *node ranking* 3A adalah pembuatan matriks *walk*. Pembuatan matriks *walk* memerlukan masukan graf dan *query* berupa *id* pelamar yang akan diberikan rekomendasi. *Node* pelamar yang memiliki nama sesuai dengan *query* tersebut akan menjadi pusat perhitungan nilai *rank* atau nilai kepentingan *node* dalam graf. *Node* tersebut dapat disebut juga *node target*. Misal matriks *walk* dilambangkan dengan M , rumus pembuatan matriks *walk* adalah seperti Persamaan 2.1 [5].

$$M = \frac{\lambda}{N} I + d \sum_{e \in E} W_e (T^e + D^e) + p_u U \quad 2.1$$

dimana $\sum_{e \in E} W_e = 1$; $\lambda + d + p_u = 1$; $d > 0$, $p_u > 0$, $\lambda > 0$

Graf masukan pada algoritma *node ranking* ini memiliki 3 jenis relasi yaitu relasi *similar*, *like*, dan *apply*. Relasi - relasi ini

menggambarkan hubungan pelamar dan informasi lowongan pekerjaan yang terdapat di dalam sistem. Pada Persamaan 2.1, setiap relasi tersebut dilambangkan dengan e dan setiap relasi e merupakan anggota dari set relasi pada graf dilambangkan dengan E . Dalam sistem rekomendasi ini set E memiliki 3 elemen yaitu *similar*, *like*, dan *apply*.

Masing - masing relasi e dalam E memiliki bobot, matriks *transition*, dan matriks *dangling* tersendiri. Bobot relasi e mewakili kemungkinan pelamar dalam memilih untuk menjelajahi informasi lowongan pekerjaan yang memiliki hubungan relasi e . Matriks *transition* adalah matriks yang memetakan persebaran nilai probabilitas kemungkinan pelamar dalam menjelajahi informasi - informasi lowongan pekerjaan yang memiliki hubungan relasi e . Matriks *dangling* adalah matriks yang memetakan persebaran nilai probabilitas kemungkinan pelamar dalam menjelajahi informasi - informasi lowongan pekerjaan yang tidak memiliki hubungan relasi e . Pada Persamaan 2.1, bobot, matriks *transition*, dan matriks *dangling* dilambangkan dengan w , T , dan D berturut - turut.

Matriks U pada Persamaan 2.1 merupakan sebuah matriks yang menggambarkan bahwa terdapat kemungkinan untuk pelamar kembali ke halaman awal saat pelamar mengakses sistem dari seluruh halaman informasi lowongan pekerjaan yang terdapat di dalam sistem. Matriks I pada Persamaan 2.1 adalah matriks yang seluruh elemennya bernilai 1. N pada Persamaan 2.1 adalah jumlah seluruh *node* yang ada di dalam graf, atau jumlah data pelamar dan data pekerjaan yang terdapat di dalam sistem.

$$R^{k+1} = MR^k \quad 2.2$$

Tahap kedua dalam Algoritma *node ranking* dalam graf adalah iterasi *vector* R . *Vector* R menyimpan nilai *rank* atau nilai kepentingan *node* dalam graf pada setiap elemennya. *Vector* R

didapatkan dengan proses iterasi perkalian matriks dengan matriks *walk M*, seperti pada Persamaan 2.2.

Vector R memiliki ukuran sebanyak jumlah *node* dalam graf, atau jumlah data pelamar dan pekerjaan di dalam sistem. Jika *N* merupakan jumlah *node* dalam graf, atau jumlah data pelamar dan pekerjaan di dalam sistem, maka nilai awal iterasi seluruh elemen *vector R* adalah $1/N$.

Pada *vector R* akan dilakukan perkalian matriks dengan matriks *walk M* sebanyak 50 iterasi [6] seperti pada Persamaan 2.2. Hasil akhir *vector R* setelah 50 iterasi menyimpan nilai *rank* atau nilai kepentingan *node* dalam graf pada setiap elemennya.

Nilai *rank* seluruh *node* yang terdapat di dalam *vector R* akan diurutkan dari yang terbesar sampai yang terkecil. Di dalam urutan nilai *rank* tersebut masih terdapat nilai *rank* milik *node* yang mewakili pelamar. Nilai *rank* yang mewakili pelamar tersebut akan dihilangkan untuk mendapatkan nilai *rank* seluruh pekerjaan, relatif terhadap *node* target atau *node* pelamar yang akan diberikan rekomendasi pekerjaan. Sehingga hanya tersisa urutan nilai *rank* milik *node* yang mewakili pekerjaan.

$$RecScore_i = \frac{RankScore_i}{\sum RankScore} \quad 2.3$$

Rekomendasi pekerjaan merupakan 10 pekerjaan yang memiliki nilai *rank* tertinggi. Nilai rekomendasi akan dihitung berdasarkan nilai *rank* rekomendasi pekerjaan tersebut.

Misal nilai *rank* seluruh rekomendasi pekerjaan disimpan di dalam set *RankScore*. Oleh karena itu, nilai *rank* pekerjaan *i* adalah *RankScore_i*. Misal nilai rekomendasi pekerjaan *i* adalah *RecScore_i*. Maka perhitungan nilai rekomendasi pekerjaan *i* atau *RecScore_i* adalah seperti Persamaan 2.3.

2.4. Kemiripan Dokumen Teks Menggunakan Metode LSI (*Latent Semantic Indexing*)

Metode LSI adalah suatu metode untuk mendapatkan struktur semantik pada teks yang melambangkan hubungan antar kata - kata [3]. Struktur semantik tersebut dapat disebut sebagai konsep (*hidden concepts*).

Metode ini dimulai dengan membuat sebuah matriks A yang memiliki kata (*terms*) dan dokumen sebagai dimensinya. Dalam sistem rekomendasi ini dokumen akan merepresentasikan kumpulan data pelamar dan pekerjaan. Jika jumlah kumpulan data pelamar dan data pekerjaan dilambangkan dengan m dan jumlah kata (unik) yang ada dalam kumpulan data tersebut adalah n maka matriks A akan memiliki ukuran $m \times n$. Matriks A yang dapat disebut dengan matriks *term-document* ini menggambarkan hubungan antara kata dan dokumen.

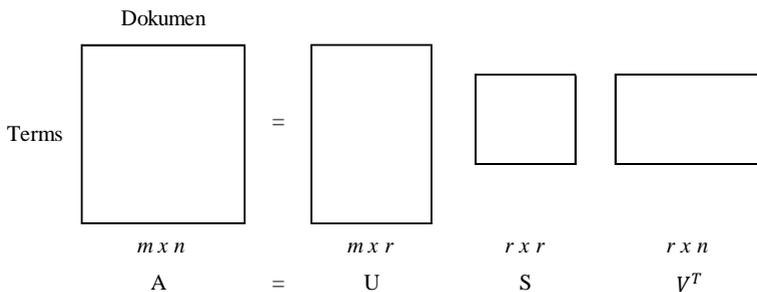
Proses dekomposisi matriks SVD (*Singular Value Decomposition*) digunakan untuk memecah matriks *term-document* menjadi matriks kata dan matriks dokumen dalam ruang konsep yang jumlah dimensinya lebih kecil [4], seperti pada Persamaan 2.4.

$$SVD(A) = U S V^T \quad 2.4$$

Misal terdapat fungsi $SVD()$ yang merupakan fungsi dekomposisi matriks SVD yang menerima parameter sebuah matriks. Maka $SVD(A)$ adalah hasil dekomposisi matriks *term-document* A menggunakan metode SVD. Proses ini akan menghasilkan konsep yang merupakan struktur semantik tersembunyi (*hidden semantic structures*) antara kata dan dokumen. Jumlah konsep yang ditemukan dapat dilambangkan dengan r .

Hasil dekomposisi matriks matriks A menggunakan SVD direpresentasikan dalam matriks U , S , dan V^T . Matriks U adalah matriks yang merepresentasikan hubungan kata dan konsep. Ukuran matriks U adalah $m \times r$. Setiap baris dalam matriks U

merupakan *vector* yang merepresentasikan kata dalam ruang konsep. Matriks S merupakan matriks diagonal yang merepresentasikan konsep (*singular values*). Ukuran matriks S adalah $r \times r$. Matriks V^T adalah matriks yang merepresentasikan hubungan dokumen dengan konsep. Ukuran matriks V^T adalah $r \times n$. Setiap kolom dalam matriks V^T merupakan *vector* yang merepresentasikan dokumen dalam ruang konsep. Sistem rekomendasi akan mencari kesamaan teks dalam kumpulan dokumen profil pelamar dan deskripsi pekerjaan melalui matriks dokumen dalam ruang konsep yang telah disesuaikan $S \cdot V^T$ [6]. Ilustrasi ukuran matriks pada proses dekomposisi matriks SVD dapat dilihat pada Gambar 2.2.



Gambar 2.2 Ilustrasi Ukuran Matriks Hasil SVD

Pada sistem rekomendasi ini, representasi *vector* dokumen yang telah didapat setelah proses LSI akan diukur kemiripannya menggunakan rumus *Cosine Distance*.

$$\text{CosDis}(\vec{x}, \vec{y}) = 1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} \quad 2.5$$

Cosine Distance merupakan rumus yang digunakan untuk mengukur jarak *cosine* antar *vector* menggunakan *Cosine Similarity*. Semakin kecil jarak *cosine* antara pasangan *vector*, pasangan *vector* tersebut semakin dekat dan dapat dikatakan

semakin mirip. Rumus *Cosine Distance* dapat dilihat pada Persamaan 2.5.

Misal terdapat fungsi rumus *Cosine Distance* yang digambarkan dengan $CosDis()$ dengan parameter berupa 2 buah *vector* yaitu digambarkan dengan \vec{x} dan \vec{y} . Pada sistem rekomendasi ini *vector* \vec{x} dan \vec{y} dapat berupa kombinasi *vector* representasi dokumen pelamar dan *vector* representasi dokumen pekerjaan. Jarak antara *vector* \vec{x} dan \vec{y} dapat digambarkan dengan $CosDis(\vec{x}, \vec{y})$.

Pada sistem rekomendasi ini pengukuran kemiripan dokumen akan dilakukan pada seluruh kumpulan dokumen teks pelamar dan pekerjaan yang telah melalui proses LSI. Jika jarak antar *vector* dokumen tidak melebihi dari batas yang ditentukan, maka pasangan dokumen tersebut akan memiliki hubungan yang melambangkan kemiripan pada graf yang akan dibuat pada Bab 3.3.3.

2.5. Perhitungan Nilai Presisi Sistem Rekomendasi

Presisi adalah suatu ukuran yang umum dipakai dalam sistem temu kembali [9]. Presisi digunakan untuk menghitung presentase *item* yang relevan yang dihasilkan sebuah sistem untuk sebuah *query*. Ukuran tersebut juga dapat digunakan dalam proses evaluasi sistem rekomendasi pekerjaan untuk mengukur seberapa relevan pekerjaan yang direkomendasikan untuk pelamar oleh sistem rekomendasi pekerjaan [10].

Jenis presisi yang digunakan dalam evaluasi sistem rekomendasi ini adalah *precision at k* [9] karena jumlah rekomendasi pekerjaan yang diberikan merupakan jumlah yang telah ditentukan oleh pengguna (*fixed length*) sebanyak k .

Jumlah rekomendasi pekerjaan pada sistem rekomendasi pekerjaan ini telah ditentukan yaitu sebanyak 10 rekomendasi pekerjaan. Oleh karena itu nilai k adalah 10 pada sistem rekomendasi pekerjaan ini.

Misal seluruh rekomendasi informasi lowongan pekerjaan untuk seorang pelamar disimpan di dalam sebuah set J . Misal jumlah seluruh informasi lowongan pekerjaan pada set J dilambangkan dengan k . Misal jumlah rekomendasi informasi lowongan pekerjaan yang dinilai relevan terhadap pelamar adalah r . Maka nilai presisi sistem rekomendasi pekerjaan untuk set rekomendasi informasi lowongan pekerjaan J dihitung dengan Persamaan 2.6.

$$Precision_k(J) = \frac{r}{k} \quad 2.6$$

Relevansi rekomendasi pekerjaan yang dihasilkan oleh sistem rekomendasi ini akan ditentukan oleh kesamaan kelas antara pelamar yang diberikan rekomendasi dan pekerjaan yang direkomendasikan. Oleh karena itu, proses klasifikasi data pelamar dan pekerjaan perlu dilakukan. Proses klasifikasi data akan dijelaskan pada Bab 3.2.2. Contoh perhitungan evaluasi sistem rekomendasi pekerjaan dengan menggunakan nilai presisi akan dijelaskan lebih dalam pada Bab 5.2.

BAB III

ANALISIS DAN PERANCANGAN

Pada Bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari Tugas Akhir. Perancangan ini meliputi perancangan data, perancangan proses, dan perancangan antar muka, serta juga akan dijelaskan tentang analisis implementasi metode secara umum pada sistem.

3.1. Analisis Implementasi Metode Secara Umum

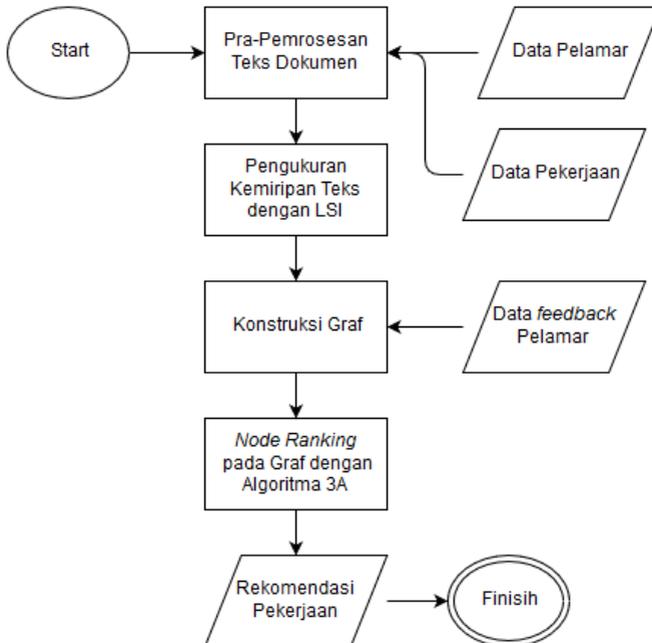
Pada tugas akhir ini akan dibuat sebuah sistem yang berfungsi untuk memberikan rekomendasi pekerjaan kepada pelamar pekerjaan. Rekomendasi pekerjaan didapatkan melalui beberapa tahap yaitu, tahap pra-pemrosesan teks, tahap pengukuran kemiripan teks, tahap konstruksi graf, dan tahap *node ranking* pada graf.

Tahap pra-pemrosesan teks bertujuan untuk menyiapkan teks agar dapat diproses untuk tahap selanjutnya. Masukan dari proses ini adalah kumpulan dokumen teks yang merupakan data profil pelamar dan deskripsi pekerjaan. Teks dokumen masukan masih memiliki banyak redundansi, yaitu berupa kata – kata yang tidak memiliki nilai semantik seperti kata konjungsi, preposisi, dll. Oleh karena itu, tahap ini akan membersihkan teks dari redundansi tersebut dan memberikan keluaran teks dokumen yang lebih ringkas. Teks dokumen yang telah melalui tahap ini akan diproses pada tahap pengukuran kemiripan teks.

Tahap pengukuran kemiripan teks bertujuan untuk mencari kemiripan antara pelamar dan pekerjaan berdasarkan teks dokumennya. Metode LSI (*Latent Semantic Indexing*) akan membuat representasi dokumen teks dalam *vector*. *Vector - vector* dokumen teks tersebut akan dihitung jaraknya dengan menggunakan rumus *Cosine Distance*. Jarak *vector* antar dokumen teks menunjukkan kemiripan antara dokumen teks tersebut. Semakin dekat jarak *vector* antara pasangan dokumen, maka pasangan dokumen teks dapat dikatakan semakin mirip. Pada

tahap ini pasangan dokumen yang memiliki nilai kemiripan tinggi (jarak *vectornya* tidak melebihi batas yang ditentukan) maka akan memiliki relasi kemiripan dalam tahap konstruksi graf.

Tahap konstruksi graf bertujuan untuk membuat graf yang terdiri dari pelamar dan pekerjaan sebagai *node*, dan *edge* berupa 3 jenis relasi yang akan menghubungkan *node* tersebut yaitu relasi *similaritas*, relasi *like*, dan relasi *apply*. Relasi *similar* didapatkan melalui tahap pengukuran kemiripan teks, dan relasi *like* dan *apply* didapatkan dari data masukan *feedback* pelamar terhadap pekerjaan. Tiap relasi memiliki bobot yang telah ditentukan.



Gambar 3.1 Diagram Alir Sistem Rekomendasi Pekerjaan

Tahap *node ranking* pada graf bertujuan untuk mengetahui urutan nilai *rank node* pada graf, relatif terhadap suatu target *node*. Target *node* adalah *node* pelamar yang akan diberikan

rekomendasi. Nilai *rank node* pada graf akan dihitung dengan menggunakan algoritma *node ranking 3A (Actor, Activity, Asset)*. Beberapa *node* pekerjaan dengan nilai *rank* tertinggi akan dijadikan rekomendasi pekerjaan untuk target *node* pelamar. Diagram alir keseluruhan proses dapat dilihat pada Gambar 3.1.

3.2. Perancangan Data

Pada sub bab ini akan dijelaskan bagaimana struktur data yang akan digunakan dan diolah pada sistem rekomendasi ini. Data yang akan digunakan adalah data masukan (*input*), data proses, dan data keluaran (*output*) yang merupakan rekomendasi untuk pengguna yang merupakan calon pelamar.

3.2.1. Data Masukan

Data masukan merupakan data awal yang dibutuhkan untuk memulai pembuatan rekomendasi oleh sistem. Data masukan untuk sistem rekomendasi ini terdiri dari 3 jenis yaitu data pelamar, data pekerjaan, dan data *feedback* pelamar terhadap lowongan pekerjaan. *Id* pelamar diawali dengan “JC” dan pekerjaan diawali dengan “JD”.

The screenshot shows the Indeed website interface. At the top, there is a search bar with the text 'what:' followed by 'job title, skills, or company' and 'where:' followed by 'New York, NY'. Below the search bar, there are several filters and a list of job results. The first result is for 'Amina Martin' in Bronx, NY, with a job description for 'Mental Health Therapy Aid' at the New York City Children's Center. Other results include 'Christopher Alvarez' as a Custodian, 'Vineet Dandau' as a Lead Analyst, and 'Ronny Villaman' as an Equipment Manager. The page also features a sidebar with various filters such as 'Distance', 'Job Titles', 'Years of Work Experience', and 'Education'.

Gambar 3.2 Screenshot Website www.indeed.com

Data pelamar didapatkan melalui proses *web crawling* pada website www.indeed.com [3], seperti pada Gambar 3.2, dengan menggunakan www.import.io [9], seperti pada Gambar 3.3, yang merupakan *crawler* berbasis web. Jumlah data pelamar yang didapatkan adalah 1000 data. Contoh data pelamar dapat dilihat pada Tabel 2.1. Data pekerjaan didapatkan dari website *open dataset* nycopendata.socrata.com [4]. Jumlah data pekerjaan yang didapatkan adalah 1484 data. Contoh data pekerjaan dapat dilihat pada Tabel 2.2. Data deskripsi pelamar dan pekerjaan menggunakan teks dalam Bahasa Inggris. Data deskripsi pelamar dan pekerjaan didapatkan dalam bentuk *file* berekstensi *.csv*.

#	Applicant value	Location value	Job title	Education value	Save resume link
1	Eric Collins JR	- Clifton, NJ	Crew Member - Quick Chek Crew Member - Sportbills Landsc... Construction Laborer - Nicos Gr...	SANT AUGUSTINE'S University	save resume
2	Anthony Gupta	- Staten Island, NY	Team Lead, Hardlines - Target st...	Tottenville High School, Regents	save resume
3	Natalie Bednarczyk	- Kearny, NJ	Social Work Internship - Bloomi...	Monmouth University, Master's	save resume
4	Meghan Massett	- Middletown, NJ	Physician and Business Develop...	Kean University, MPA	save resume
5	Matthew J Rasmussen	- Garden City South, NY	CAD Designer - Plumbing & Fire ...	Jaros Baum & Bolles University	save resume
6	Claudia Rosario	- Brooklyn, NY	Clerical Associate/Unit Clerk - M...	City College of the City Univers...	save resume
7	Laviat Wangmare	- New York, NY	Business Analyst /Quality Analys...		save resume

Gambar 3.3 Screenshot Website www.import.io

Data *feedback* pelamar terhadap pekerjaan merupakan hasil dari interaksi pelamar pada sistem yang menunjukkan ketertarikan terhadap suatu pekerjaan. Ketertarikan pelamar terhadap pekerjaan dapat ditunjukkan dengan memberikan tanda *like* dan *apply* pada suatu pekerjaan.

Pada Tugas Akhir ini data *feedback* dibuat secara manual oleh penulis dengan memasang beberapa pekerjaan dengan pelamar secara acak berdasarkan kelasnya. Jika pelamar memiliki kelas *management* maka pekerjaan yang akan diberikan tanda *like* atau *apply* juga merupakan dari kelas *management*. Jenis tanda

ketertarikan (*like/apply*) yang dipakai acak untuk setiap data *feedback* yang dimiliki pelamar.

Data *feedback* yang telah dibuat secara manual berjumlah 2532 data dengan rata-rata data *feedback* untuk setiap pelamar adalah 3 data. Contoh data *feedback* pelamar dapat dilihat pada Tabel 2.3. Masukan *feedback* pelamar terhadap pekerjaan disimpan dalam *file* berekstensi *.csv*.

3.2.2. Klasifikasi Manual Data Masukan Pelamar dan Pekerjaan

Klasifikasi data pelamar dan pekerjaan diperlukan untuk mendapatkan sebuah *ground truth* sehingga kinerja sistem rekomendasi dapat diukur. *Ground truth* tersebut akan menentukan relevansi rekomendasi pekerjaan terhadap pelamar.

Tabel 3.1 Hasil Klasifikasi Manual Data Masukan Pelamar dan Pekerjaan

No	Kelas	Jumlah	
		Data Pelamar	Data Pekerjaan
1	Management	345	295
2	IT	286	269
3	Engineering	133	478
4	Supply-Chain	45	39
5	Medical/HealthCare	28	160
6	Legal	18	56
7	Culinary	15	3
8	Journalism	12	6
9	Education	10	3
10	Art	5	9
11	Other	103	166
Total		1000	1484

Klasifikasi dilakukan secara manual berdasarkan penilaian penulis. Sebanyak 11 kelas yang merupakan jenis profesi didapatkan dari 1000 data pelamar dan 1484 data pekerjaan. Hasil klasifikasi data manual dapat dilihat pada Tabel 3.1.

Pada Tabel 3.1 kelas *Culinary*, *Journalism*, *Education* dan *Art* memiliki jumlah data pelamar dan pekerjaan yang sangat kecil. Oleh karena itu, data pelamar dan pekerjaan pada kelas - kelas tersebut dan data yang tidak terklasifikasi -yaitu pada kelas *Other*-lebih baik dihilangkan dari set data. Sehingga data yang dipakai merupakan dari kelas *Management*, *IT*, *Engineering*, *Supply-Chain* dan *Medical/ HealthCare*. Total jumlah data pelamar pada kelas - kelas tersebut adalah 855 dan total jumlah data pekerjaan pada kelas - kelas tersebut adalah 1297, seperti pada Tabel 3.2.

Tabel 3.2 Klasifikasi Data Masukan Pelamar dan Pekerjaan yang Dipakai

No	Kelas	Jumlah	
		Data Pelamar	Data Pekerjaan
1	Management	345	295
2	IT	286	269
3	Engineering	133	478
4	Supply-Chain	45	39
5	Medical/HealthCare	28	160
6	Legal	18	56
Total		855	1297

3.2.3. Data Keluaran

Data keluaran pada sistem rekomendasi ini adalah urutan 10 pekerjaan yang memiliki nilai *rank* tertinggi pada hasil keluaran Tahap *Node Ranking* pada Graf dengan Algoritma 3A (*Actor*, *Asset*, *Activities*). Nilai rekomendasi untuk setiap pekerjaan tersebut akan dihitung pada tahap tersebut. Urutan pekerjaan tersebut merupakan rekomendasi pekerjaan yang akan diberikan

untuk pelamar. Rekomendasi pekerjaan berjumlah 10 tidak termasuk pekerjaan yang telah diberikan tanda ketertarikan *like* dan *apply* oleh pelamar. Pekerjaan yang telah diberikan tanda ketertarikan tersebut akan tetap diperlihatkan kepada pelamar apabila memiliki nilai *rank* yang tinggi tetapi tidak akan mendapat nilai rekomendasi.

Tabel 3.3 Data Keluaran

No	<i>Id</i> Pelamar : JC-2	
	<i>Id</i> Pekerjaan	Nilai Rekomendasi
1	JD-7	0.2859
2	JD-9	0.2834
3	JD-2	0.0588
4	JD-5	0.0563
5	JD-1	0.0559
6	JD-8	0.0559
7	JD-6	0.0542
8	JD-4	0.0505
9	JD-3	0.0501
10	JD-10	0.0492

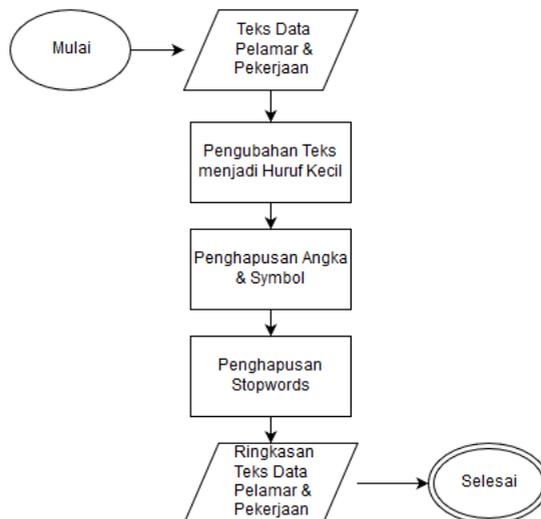
Pada contoh data keluaran pada Tabel 3.3, merupakan hasil data keluaran dari 5 data pelamar, 10 data pekerjaan, dan 11 data *feedback*. Data - data tersebut dapat dilihat pada Lampiran 2. Data keluaran tersebut dihasilkan dari hasil *node ranking* pada graf di Gambar 3.11. Pada tabel tersebut terlihat *id* pekerjaan yang akan direkomendasikan terhadap pelamar. Pelamar yang diberikan rekomendasi pada data keluaran tersebut adalah pelamar yang memiliki *id* JC-2.

Pada Tabel 3.3 yang merupakan hasil rekomendasi pekerjaan terlihat bahwa pekerjaan dengan *id JD-7* dan *JD-9* memiliki nilai rekomendasi yang tertinggi yaitu 0.2859 dan 0.2834 berturut – turut, karena *node* pekerjaan tersebut berhubungan langsung dengan *node* target *JC-2*. Pada graf, *node – node* tersebut berhubungan langsung dengan *node* target melalui relasi *like* dan *apply* berturut – turut untuk *node JD-7* dan *JD-9*.

3.3. Perancangan Proses

Pada subbab ini akan dibahas mengenai perancangan proses yang dilakukan untuk memberikan gambaran pada setiap alur implementasi metode dalam pembuatan rekomendasi pekerjaan kepada pelamar.

3.3.1. Tahap Pra-Pemrosesan Teks Dokumen

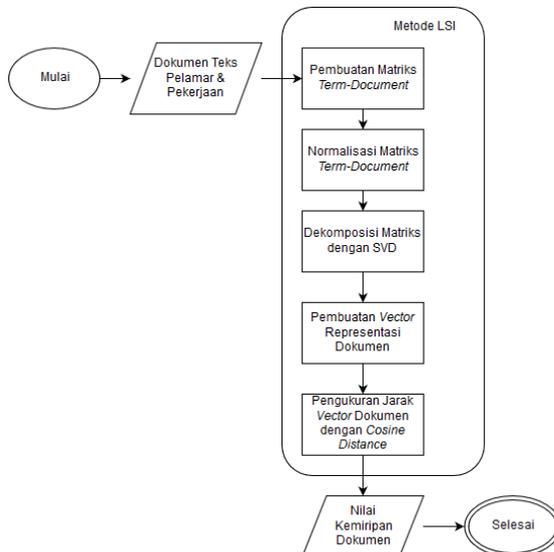


Gambar 3.4 Diagram Alir Tahap Pra-Pemrosesan Teks

Data masukan dekripsi calon pelamar dan pekerjaan akan diubah terlebih dahulu menjadi kumpulan dokumen teks. Kata – kata di dalam teks pada kumpulan dokumen teks tersebut diubah menjadi huruf kecil (*lowercase*). Lalu kata – kata *stopwod*, simbol, dan angka pada teks kumpulan dokumen tersebut akan dihilangkan. Implementasi penghilangan kata *stopword* akan menggunakan *library* NLTK.

Tahap ini membersihkan teks dari kata – kata yang tidak penting dan membuatnya menjadi lebih ringkas. Keluaran dari tahap ini, yaitu teks dokumen yang lebih ringkas, akan digunakan pada tahap pengukuran kemiripan teks. Diagram alir tahap ini dapat dilihat pada Gambar 3.4.

3.3.2. Tahap Pengukuran Kemiripan Teks dengan Metode LSI (*Latent Semantic Indexing*)



Gambar 3.5 Diagram Alir Tahap Pengukuran Kemiripan Teks

Tahap pengukuran kemiripan teks bertujuan untuk mencari kemiripan antara teks pada kumpulan data profil pelamar dan deskripsi pekerjaan. Masukan dari tahap ini adalah dokumen teks data pelamar dan pekerjaan yang telah melalui tahap pra-pemrosesan teks. Metode LSI (*Latent Semantic Indexing*) akan digunakan untuk merepresentasikan dokumen teks data pelamar dan pekerjaan dalam sebuah *vector*. Lalu *vector* dokumen teks tersebut dihitung jaraknya dengan menggunakan rumus *cosine distance*. Jarak antar *vector* dokumen melambangkan kesamaan teks antar dokumen tersebut.

Metode LSI terdiri dari beberapa sub-tahap yaitu pembuatan matriks *term-document*, dekomposisi matriks *term-document* menggunakan SVD (*Singular Value Decomposition*), dan representasi dokumen teks dalam *vector*. Lalu *vector - vector* dokumen teks data pelamar dan pekerjaan tersebut akan diukur jaraknya menggunakan rumus *cosine distance*.

Implementasi metode LSI akan menggunakan beberapa *library* yaitu *numpy* untuk proses dekomposisi matriks dengan SVD (*Singular Value Decomposition*), *textmining* untuk proses pembuatan matriks *term-document*, dan *scipy* untuk perhitungan jarak *vector* dengan rumus *cosine distance*.

Keluaran tahap ini adalah pasangan - pasangan entitas dokumen teks yang memiliki nilai kemiripan tinggi. Pasangan entitas tersebut akan memiliki relasi *similar* pada tahap konstruksi graf. Diagram alir tahap pengukuran kemiripan teks dapat dilihat pada Gambar 3.5.

Tahap ini diawali dengan membuat sebuah matriks akan dengan menggunakan dokumen teks data profil pelamar dan deskripsi pekerjaan sebagai masukan. Baris (*row*) pada matriks *term-document* akan disusun oleh gabungan kata – kata unik yang terdapat dalam kumpulan dokumen teks data pelamar dan pekerjaan yang telah melalui tahap pra-pemrosesan teks. Gabungan kata tersebut dapat disebut dengan ruang kata (*terms*

space). Kolom (*column*) pada matriks *term-document* akan disusun oleh dokumen teks data pelamar dan pekerjaan.

Nilai matriks *term-document* adalah jumlah kemunculan kata pada dokumen teks data pelamar dan pekerjaan. Misal matriks *term-document* dilambangkan dengan m , kata pada ruang kata dilambangkan dengan i , dan dokumen dilambangkan dengan j . Jika terdapat n kali kata i muncul dalam dokumen j , maka nilai matriks $m[i,j] = n$. Lalu matriks ini akan dinormalisasi sehingga nilai matriks hanya berkisar antar 0 dan 1.

Contoh teks data pelamar dan pekerjaan yang telah melalui tahap pemrosesan teks adalah seperti Tabel 3.4. Terlihat pada contoh teks data tersebut pelamar yang memiliki *id JC-1* memiliki *term "administration", "business", dan "analysis"*.

Tabel 3.4 Contoh Data Masukan Tahap Pengukuran Kemiripan Teks

No	<i>Id Data</i>	Isi Dokumen Teks Data
1	JC-1	<i>Administration Business Analysis Administration</i>
2	JC-2	<i>Administration Business Law</i>
3	JD-1	<i>Care Administration Care</i>
4	JD-2	<i>Business Communication Analysis Communication</i>

Matriks *term-document* yang dibuat dari contoh teks data pelamar dan pekerjaan pada Tabel 3.4 dapat dilihat pada Gambar 3.6. Pada matriks *term-document* pada Gambar 3.6 terlihat *term "administration"* pada dokumen teks data pelamar *JC-1* muncul sebanyak 2 kali. *Term "care"* pada dokumen teks data pekerjaan *JD-1* muncul sebanyak 2 kali.

		Kumpulan Dokumen			
		JC-1	JC-2	JD-1	JD-2
Ruang Kata	administration	2	1	1	0
	analysis	1	0	0	1
	business	1	1	0	1
	care	0	0	2	0
	communication	0	0	0	2
	law	0	1	0	0

Gambar 3.6 Contoh Matriks *Term-Document*

Pada sub-tahap normalisasi matriks *term-document*, matriks *term-document* yang telah dibuat dilakukan normalisasi sehingga nilai matriks tersebut antara 0 dan 1. Hasil normalisasi matriks *term-document* dapat dilihat pada Gambar 3.7.

		Kumpulan Dokumen			
		JC-1	JC-2	JD-1	JD-2
Ruang Kata	administration	0.5	0.33	0.33	0
	analysis	0.25	0	0	0.25
	business	0.25	0.33	0	0.25
	care	0	0	0.67	0
	communication	0	0	0	0.5
	law	0	0.33	0	0

Gambar 3.7 Contoh Hasil Normalisasi Matriks *Term-Document*

Sub-tahap selanjutnya adalah sub-tahap dekomposisi matriks. Matriks yang akan didekomposisi adalah matriks *term-document* yang telah dibuat pada tahap sebelumnya. Pada tahap ini

matriks *term-document* didekomposisi menjadi 3 matriks yang berbeda yaitu matriks U , S , dan V^T . Contoh dekomposisi matriks *term-document* pada Gambar 3.7 dapat terlihat pada Gambar 3.8.

$$\begin{array}{c}
 \begin{array}{c} U \\
 \left(\begin{array}{cccc}
 -0.7248 & -0.0475 & 0.358 & -0.3302 \\
 -0.2343 & 0.3134 & -0.2184 & -0.4802 \\
 -0.4154 & 0.4379 & 0.0962 & 0.2405 \\
 -0.4399 & -0.701 & -0.46 & 0.2114 \\
 -0.1449 & 0.4482 & -0.7102 & 0.1962 \\
 -0.1811 & 0.1245 & 0.3146 & 0.7207
 \end{array} \right) \\
 \\
 \begin{array}{c} S \\
 \left(\begin{array}{cccc}
 0.9003 & 0 & 0 & 0 \\
 0 & 0.6779 & 0 & 0 \\
 0 & 0 & 0.5211 & 0 \\
 0 & 0 & 0 & 0.3119
 \end{array} \right) \\
 \\
 \begin{array}{c} V^T \\
 \left(\begin{array}{cccc}
 -0.5829 & -0.4892 & -0.594 & -0.2609 \\
 0.242 & 0.2532 & -0.7128 & 0.6077 \\
 0.2849 & 0.4918 & -0.3595 & -0.7401 \\
 -0.7215 & 0.6743 & 0.0989 & 0.1224
 \end{array} \right)
 \end{array}
 \end{array}
 \end{array}$$

Gambar 3.8 Contoh Hasil Dekomposisi Matriks *Term-Document*

Dokumen teks data pelamar dan pekerjaan akan direpresentasikan ke dalam bentuk *vector* dengan menggunakan matriks hasil dekomposisi tersebut. *Vector* dokumen teks data pelamar dan pekerjaan didapatkan dari kolom pada matriks S dan V^T , hasil dekomposisi matriks *term-document*.

Pada contoh matriks hasil perkalian matriks S dan V^T di Gambar 3.9, terlihat setiap kolom pada matriks tersebut mewakili setiap dokumen teks data pelamar dan pekerjaan yang terdapat di dalam sistem.

$$\left(\begin{array}{cccc}
 \text{JC-1} & \text{JC-2} & \text{JD-1} & \text{JD-2} \\
 -0.5248 & -0.4404 & -0.5348 & -0.2349 \\
 0.1641 & 0.1716 & -0.4832 & 0.412 \\
 0.1485 & 0.2563 & -0.1873 & -0.3857 \\
 -0.225 & 0.2103 & 0.0308 & 0.0382
 \end{array} \right)$$

Gambar 3.9 Contoh *Vector* Representasi Dokumen Teks Data

Setiap *vector* dokumen teks data pelamar dan pekerjaan akan dihitung jaraknya menggunakan rumus *Cosine Distance* seperti pada Persamaan 2.5. Implementasi rumus *Cosine Distance* akan menggunakan *library scipy*. Jumlah perhitungan jarak *vector* yang akan dilakukan adalah *combination 2* elemen dari jumlah dokumen teks data pelamar dan pekerjaan. Semakin dekat jarak antar *vector*, dua entitas dokumen teks dapat dikatakan semakin mirip. Pasangan entitas dokumen teks yang memiliki jarak *vector* kurang dari batas yang telah ditentukan akan dijadikan relasi *similar* pada tahap konstruksi graf.

$$\begin{aligned} & \text{CosDis}(JC - 1, JC - 2) \\ &= 1 - \frac{[-0.5248 \ 0.1641 \ 0.1485 \ -0.225] \cdot [-0.4404 \ 0.1716 \ 0.2563 \ 0.2103]}{|-0.5248 \ 0.1641 \ 0.1485 \ -0.225| \ |-0.4404 \ 0.1716 \ 0.2563 \ 0.2103|} \\ &= 0.23 \end{aligned}$$

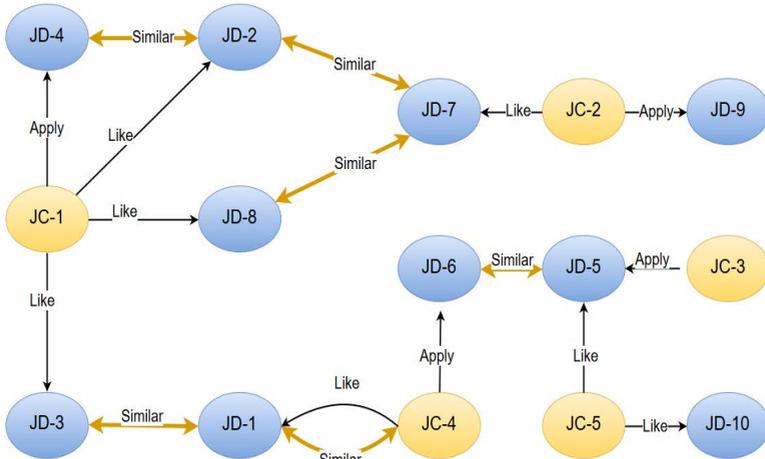
Gambar 3.10 Contoh Perhitungan *Cosine Distance*

Contoh perhitungan *cosine distance* antara 2 dokumen teks dapat dilihat pada Gambar 3.10. Pada contoh perhitungan tersebut *vector* dokumen teks data pelamar *JC-1* dihitung jaraknya dengan *vector* dokumen teks data pelamar *JC-2*. Perhitungan *cosine distance* menunjukkan bahwa nilai jarak kedua *vector* tersebut adalah 0.23. Karena jarak *vector* dokumen teks data kedua *vector* tersebut kecil maka dapat dikatakan bahwa dokumen teks data kedua *vector* tersebut memiliki kesamaan teks. Sehingga *node JC-1* dan *JC-2* pada tahap konstruksi graf akan memiliki relasi *similar*.

3.3.3. Tahap Konstruksi Graf

Pada tahap ini akan dibuat sebuah graf yang memodelkan hubungan objek – objek pada sistem rekomendasi. Objek – objek sistem rekomendasi adalah pelamar dan pekerjaan. Graf yang akan dibuat merupakan *multi-relational directed* graf dengan pelamar

dan pekerjaan sebagai *node*. *Node* pelamar dan pekerjaan akan dihubungkan oleh 3 jenis relasi yaitu : relasi *similar*, relasi *like*, dan relasi *apply*. Relasi tersebut akan menjadi *directed edge* atau *edge* yang memiliki arah pada graf. Penjelasan mengenai ketiga relasi ini dapat dilihat pada Tabel 2.4.



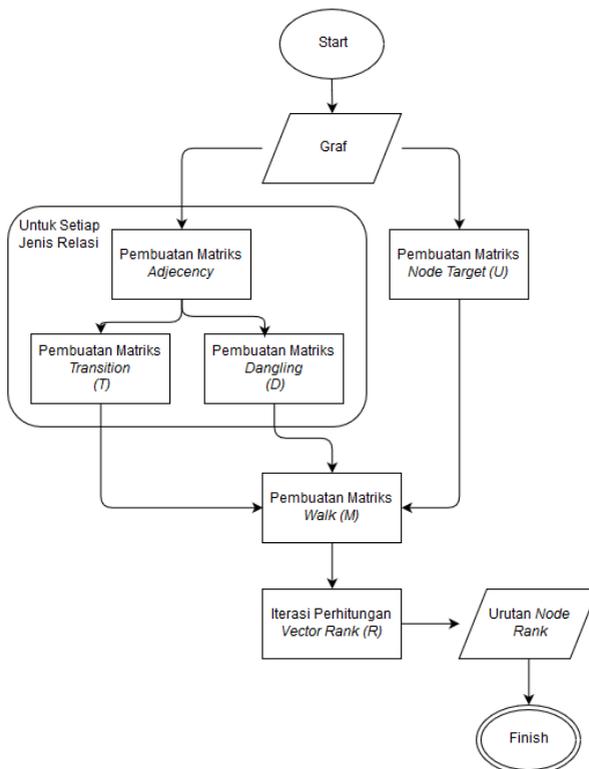
Gambar 3.11 Contoh Graf Keluaran Tahap Konstruksi Graf

Ketiga relasi tersebut akan diberikan bobot sebagai berikut : $apply > like > similar$. Relasi *similar* akan didapatkan dari keluaran tahap pengukuran kemiripan teks. Keluaran dari tahap tersebut adalah pasangan - pasangan entitas (pelamar/pekerjaan) yang memiliki kemiripan pada teks datanya. *Edge* dari relasi *similar* adalah *bidirectional*. Relasi *like* dan *apply* didapatkan melalui data masukan *feedback* pelamar terhadap pekerjaan yang dibuat secara manual. Contoh graf keluaran tahap ini dapat dilihat pada Gambar 3.11.

Terlihat pada contoh graf di Gambar 3.11 pelamar dengan *id* "JC-4" memiliki kemiripan berdasarkan teks datanya dengan pekerjaan dengan *id* "JD-1". Pelamar JC-4 juga memberikan tanda ketertarikannya berupa *like* pada pekerjaan JD-1 dan *apply* pada

pekerjaan *JD-6*. Melalui graf, setiap *node* akan dihitung nilai kepentingannya terhadap *node* pelamar yang akan diberikan rekomendasi. Nilai kepentingan setiap *node* dihitung pada Tahap *Node Ranking* pada Graf.

3.3.4. Tahap *Node Ranking* pada Graf dengan Algoritma 3A (*Actor, Asset, Activities*)



Gambar 3.12 Diagram Alir Tahap *Node Ranking* pada Graf

Tahap *node ranking* pada graf merupakan tahap akhir dalam pemberian rekomendasi pekerjaan. Algoritma 3A akan diimplementasi pada graf untuk mengukur nilai *rank* seluruh *node* relatif terhadap *node* target. *Node* target adalah *node* pelamar yang akan diberikan rekomendasi pekerjaan. Nilai *rank* setiap *node* akan dijadikan dasar pemberian rekomendasi pekerjaan.

Proses *node ranking* dengan menggunakan Algoritma 3A memiliki 2 tahap utama yaitu pembuatan matriks *walk* dan iterasi pembuatan *vector R*. Matriks *walk* merupakan matriks yang merepresentasikan proses penjelajahan *node* pada graf dengan metode *Random Walk*. *Vector R* merupakan *vector* yang menyimpan nilai *rank* seluruh *node* pada graf.

Diagram alir dari tahap *node ranking* pada graf dapat dilihat pada Gambar 3.12. Pembuatan matriks *walk* memerlukan beberapa matriks yaitu matriks *transition*, matriks *dangling*, dan matriks *node target*. Dari graf, matriks *adjacency* dibuat untuk setiap jenis relasi. Melalui matriks *adjacency*, matriks *transition* dan matriks *dangling* dibuat untuk setiap jenis relasi seperti yang telah dibahas pada Bab 2.3. Matriks *node target* atau matriks *U* dibuat berdasarkan *node* pelamar yang akan diberikan rekomendasi. Kolom yang merepresentasikan *node* target akan bernilai 1 pada matriks *U*.

Matriks *M* tersebut memilih *node JC-2* sebagai *node target*. Oleh karena itu matriks *M* yang dihasilkan memiliki ukuran 15×15 sesuai dengan jumlah *node* yang terdapat pada graf.

Matriks *M* diperlukan untuk membuat *vector rank R* melalui proses iterasi seperti Persamaan 2.2. Pada sistem rekomendasi ini iterasi perhitungan *vector R* dilakukan sebanyak 50 kali berdasarkan jumlah data pelamar dan pekerjaan yang tidak lebih dari 5 ribu data [8]. Nilai awal setiap elemen *vector R* adalah $\frac{1}{N}$ dimana *N* adalah jumlah *node* pada graf. Elemen dari *vector R* setelah proses iterasi merupakan nilai *rank* setiap *node* pada graf, relatif terhadap *node* target. Jadi ukuran *vector R* adalah jumlah *node* dalam graf.

Lalu matriks M yang merepresentasikan *random walk* pada graf dapat dibuat. Terdapat beberapa parameter dalam pembuatan matriks M , yaitu *lamda*, *damping factor*, dan *personalization factor*, seperti yang telah dijelaskan pada Bab 2.3. Contoh matriks M dapat dilihat pada Lampiran 3. Matriks M tersebut dihasilkan dari graf pada Gambar 3.11 yang menggunakan data pelamar, data pekerjaan, dan data *feedback* seperti pada Lampiran 2.

$$R = \begin{pmatrix} 0.0116 \\ 0.7116 \\ 0.0116 \\ 0.0119 \\ 0.0116 \\ 0.0135 \\ 0.0136 \\ 0.0122 \\ 0.0142 \\ 0.0135 \\ 0.0131 \\ 0.0691 \\ 0.0119 \\ 0.0685 \\ 0.0121 \end{pmatrix} \begin{matrix} \rightarrow \text{JC-1} \\ \rightarrow \text{JC-2} \\ \rightarrow \text{JC-3} \\ \rightarrow \text{JC-4} \\ \rightarrow \text{JC-5} \\ \rightarrow \text{JD-1} \\ \rightarrow \text{JD-2} \\ \rightarrow \text{JD-3} \\ \rightarrow \text{JD-4} \\ \rightarrow \text{JD-5} \\ \rightarrow \text{JD-6} \\ \rightarrow \text{JD-7} \\ \rightarrow \text{JD-8} \\ \rightarrow \text{JD-9} \\ \rightarrow \text{JD-10} \end{matrix}$$

Gambar 3.13 Contoh Hasil Vector Rank (R)

Contoh *vector R* yang telah dihasilkan dari dari proses iterasi dengan menggunakan matriks M pada Lampiran 3 dapat dilihat di Gambar 3.13. Terlihat pada gambar tersebut elemen bagian atas *vector R* ditempati oleh nilai *rank node* pelamar berurutan dari *node* pelamar dengan *id JC-1* sampai *JC-5* lalu diikuti oleh nilai

rank node pekerjaan berurutan dari *node* pekerjaan dengan *id JD-1* sampai *JD-10*.

Tabel 3.5 Contoh Data Keluaran Tahap *Node Ranking*

No	Node Target : JC-2	
	Node	Nilai Rank
1	JC-2	0.7116
2	JD-7	0.0691
3	JD-9	0.0685
4	JD-2	0.0142
5	JD-5	0.0136
6	JD-1	0.0135
7	JD-8	0.0135
8	JD-6	0.0131
9	JD-4	0.0122
10	JD-3	0.0121
11	JD-10	0.0119
12	JC-4	0.0119
13	JC-5	0.0116
14	JC-3	0.0116
15	JC-1	0.0116

Setiap elemen dari *vector* tersebut mewakili nilai *rank* yang ada pada setiap *node*. Nilai *rank* pada *vector R* akan diurutkan dari yang terbesar ke terkecil. Hasil pengurutan nilai *rank* seluruh *node* pada *vector R* adalah urutan nilai *rank* seperti pada Tabel 3.5. Terlihat pada tabel tersebut nilai *rank node JC-2* memiliki nilai yang paling tinggi dengan nilai *rank* 0.7116 karena *node* tersebut merupakan *node target*. Sedangkan *node JD-7* dan *JD-9* memiliki nilai yang tinggi karena terhubung langsung dengan *node target JC-2* dengan nilai *rank* berturut – turut 0.0691 dan 0.0685. Jumlah seluruh nilai *rank* pada tahap *node ranking* ini adalah 1.

Pada Tabel 3.5 terlihat masih terdapat *node* yang mewakili pelamar. Rekomendasi pekerjaan dihasilkan dari 10 pekerjaan

dengan nilai *rank* tertinggi. Oleh karena itu, nilai *rank* milik *node* yang mewakili pelamar dan nilai *rank* milik *node* pekerjaan yang tidak menempati urutan 10 besar pada Tabel 3.5 akan dihilangkan. Sehingga hanya tersisa nilai *rank* yang ditandai warna biru pada Tabel 3.5, yaitu 10 nilai *rank* pekerjaan tertinggi. Pekerjaan - pekerjaan tersebut yang akan dijadikan sebagai rekomendasi pekerjaan untuk pelamar.

Tabel 3.6 Contoh Hasil Rekomendasi Pekerjaan

No	Id Pelamar : JC-2	
	Id Pekerjaan	Nilai Rekomendasi
1	JD-7	0.2859
2	JD-9	0.2834
3	JD-2	0.0588
4	JD-5	0.0563
5	JD-1	0.0559
6	JD-8	0.0559
7	JD-6	0.0542
8	JD-4	0.0505
9	JD-3	0.0501
10	JD-10	0.0492

Nilai rekomendasi akan dihitung untuk rekomendasi pekerjaan tersebut menggunakan nilai *rank* seperti Persamaan 2.3. Hasil perhitungan nilai rekomendasi seperti pada Tabel 3.6. Jumlah nilai seluruh nilai rekomendasi adalah 1. Pekerjaan dengan *id* JD-7 dan JD-9 memiliki nilai rekomendasi tertinggi karena pekerjaan – pekerjaan tersebut telah diberikan tanda *like* dan *apply* berturut – turut oleh pelamar JC-2 seperti pada data *feedback* di Lampiran 3.

Pekerjaan dengan *id* JD-2 direkomendasikan kepada pelamar JC-2 karena pekerjaan JD-2 memiliki kemiripan terhadap

pekerjaan yang disukai oleh pelamar *JC-2* yaitu pekerjaan *JD-7*. Terlihat pada data pekerjaan di Lampiran 3 terlihat pekerjaan *JD-2* dan pekerjaan *JD-7* merupakan pekerjaan dari institusi yang sama yaitu institusi *Dept. of Health/Mental Hygiene*.

Rekomendasi yang diberikan terkadang tidak sesuai dengan profil pelamar atau ketertarikan pelamar. Contoh rekomendasi pekerjaan yang tidak sesuai adalah pekerjaan dengan *id JD-5* dan *JD-1*. Pekerjaan tersebut memiliki nilai rekomendasi yang relatif tinggi hanya karena *node* yang mewakili pekerjaan tersebut pada graf di Gambar 3.11 memiliki *edge* masukan yang banyak. *Edge* masukan yang banyak pada graf mendandakan pekerjaan tersebut banyak disukai oleh pelamar.

Pekerjaan lain yang sesuai dengan ketertarikan pelamar *JC-2* adalah pekerjaan *JD-8* dan *JD-4* karena berasal dari institusi *Dept. of Health/Mental Hygiene*. Pekerjaan tersebut memiliki nilai rekomendasi yang lebih rendah dari pekerjaan *JD-5* dan *JD-1* karena hanya tidak memiliki *edge* masukan sebanyak *JD-5* dan *JD-1*. Hal ini juga dapat disebabkan oleh parameter λ (*lamda*) yaitu kecenderungan pelamar untuk melihat pekerjaan yang acak.

Pada sistem rekomendasi dengan data pekerjaan lebih dari 10 pekerjaan – pekerjaan yang telah diberi tanda ketertarikan *like* dan *apply* tidak direkomendasikan, namun tetap diperlihatkan kepada pelamar jika memiliki nilai *rank* yang tinggi.

3.4. Perancangan Antarmuka Sistem Rekomendasi

Pada subbab ini akan dibahas mengenai perancangan antarmuka perangkat lunak yang bertujuan untuk dapat mempermudah interaksi antara perangkat lunak dengan pengguna.

3.4.1. Menu *File* Masukan

Menu ini memiliki tujuan untuk memilih *file* masukan yang dibutuhkan oleh sistem. *File* masukan terdiri dari 3 jenis yaitu : masukan data deskripsi pelamar, masukan data deskripsi pekerjaan, dan masukan data *feedback* pelamar terhadap

pekerjaan. *File* masukan merupakan *file* yang memiliki ekstensi .csv.

3.4.2. Halaman Interaksi Pelamar

Halaman ini memberikan simulasi pemberian *feedback* oleh pelamar terhadap pekerjaan. Pengguna dapat melihat seluruh data deskripsi pelamar dan pekerjaan. Pada halaman ini pengguna dapat memilihkan pekerjaan yang akan diberikan *feedback* untuk pelamar yang telah ditentukan. *Feedback* yang dapat diberikan terhadap pekerjaan adalah *like* dan *apply*. Hasil *feedback* yang dilakukan pada halaman ini akan disimpan secara otomatis dalam *file* yang menjadi masukan data *feedback* pelamar terhadap pekerjaan yang telah dipilih pada menu *file* masukan.

3.4.3. Halaman Rekomendasi Pekerjaan

Rekomendasi pekerjaan untuk pelamar dilakukan pada halaman ini. Pengguna dapat melihat seluruh *feedback* pelamar terhadap pekerjaan dan pasangan entitas yang memiliki relasi *similar* pada halaman ini. Halaman ini menerima masukan berupa *id* target pelamar yang merupakan pelamar akan diberikan rekomendasi. Pengguna juga dapat memasukkan nilai parameter untuk implementasi Algoritma 3A. Setelah semua masukan telah terisi maka pemberian rekomendasi pekerjaan dapat dilakukan.

Urutan pekerjaan akan muncul sesuai dengan nilai *ranknya* relatif terhadap pelamar. Pekerjaan yang pernah diberikan *feedback* oleh target pelamar akan diberi label “*liked*” atau “*applied*” pada tabel hasil rekomendasi. Pekerjaan tersebut tidak akan dijadikan rekomendasi oleh sistem. Pekerjaan yang direkomendasikan diberi label “*Recommended*”. Presisi dari hasil rekomendasi yang diberikan juga diperlihatkan pada halaman ini.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah bahasa pemrograman Python dengan pustaka PyQt.

4.1. Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan Tugas Akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti yang ditampilkan pada Tabel 4.1.

Tabel 4.1 Tabel Spesifikasi Lingkungan Implementasi

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i5-4200U CPU @ 1.60GHz Memori (RAM): 7890 MB
Perangkat lunak	Sistem Operasi: Microsoft Windows 10 Pro 64-bit Perangkat Pengembang: Python IDLE Version 2.7.10 dan QtDesigner Pustaka Python: numpy, scipy, textmining, sklearn dan nltk

4.2. Implementasi Proses

Implementasi proses dilakukan berdasarkan perancangan proses yang sudah dijelaskan pada bab analisis dan perancangan.

4.2.1. Implementasi Tahap Pra-Pemrosesan Teks

Subbab ini membahas implementasi tahap pra-pemrosesan teks dari *file* masukan untuk dapat diproses pada tahap selanjutnya. Fungsi *preprocess* menerima parameter berupa teks dan akan menghilangkan simbol, angka, dan kata *stopword* dari teks tersebut seperti terlihat pada Kode Sumber 4.1. Kumpulan kata *stopword* didapatkan melalui pustaka NLTK pada bahasa pemrograman Python.

```

1  from nltk.corpus import stopwords
2  cachedStopWords =
   stopwords.words("english")
3
4  def preprocess(self, text):
5      listtext = list()
6      text = text.lower()
7      text = re.sub(r'[^\\w]', ' ', text)
8      text = ''.join([i for i in text if not
9                      i.isdigit()])
10     text = ''.join([word for word in
11                    text.split() if word not in
12                    cachedStopWords])
13     return text

```

Kode Sumber 4.1 Implementasi Pra-Pemrosesan Teks

4.2.2. Implementasi Tahap Pengukuran Kemiripan Teks dengan LSI (*Latent Semantic Indexing*)

Proses pertama dari tahap ini adalah pembuatan matriks *term-document* matriks menggunakan dokumen teks pelamar dan pekerjaan yang telah melalui tahap pra-pemrosesan teks. Implementasi proses pembuatan matriks *term-document* dapat dilihat pada Kode Sumber 4.2. Pembuatan matriks *term-document* menggunakan pustaka *textmining*. Variabel *tdm* menyimpan matriks *term-document* yang akan dibuat. Dokumen teks yang telah melalui tahap pra-pemrosesan teks akan dimasukkan ke dalam variabel *tdm* dengan perintah *tdm.add_doc()*.

```

1 import textmining
2 tdm = textmining.TermDocumentMatrix()
3 for d_n in self.dict_candidate_sorted :
4     cn_data = d_n
5     word_list = [cn_data[2], cn_data[3],
6                 cn_data[4]]
7     text = ' '.join(word_list)
8
9     text = self.preprocess(text)
10    tdm.add_doc(text)
11
12 for d_n in self.dict_job_sorted :
13     jb_data = d_n
14     word_list = [jb_data[3], jb_data[4],
15                 jb_data[5]]
16     text = ' '.join(word_list)
17
18     text = self.preprocess(text)
19     tdm.add_doc(text)

```

Kode Sumber 4.2 Implementasi Pembuatan Matriks *Term-Document*

Matriks *term-document* yang telah dibuat selanjutnya akan didekomposisi menggunakan SVD sehingga menghasilkan matriks U , s , dan V . Proses dekomposisi matriks menggunakan SVD dilakukan dengan pustaka `numpy`. Proses dekomposisi matriks dapat dilihat pada Kode Sumber 4.3.

```

1 from numpy import *
2
3 U,s,V = linalg.svd(mtxadj_norm,
4 full_matrices=False)

```

Kode Sumber 4.3 Implementasi Dekomposisi Matriks

Matriks U , s , dan V hasil dekomposisi matriks *term-document* akan digunakan untuk membuat matriks representasi

dokumen seperti pada Kode Sumber 4.4. *Vector* dokumen merupakan tiap kolom dari matriks *docs_rep*.

```
1 docs_rep = dot(diag(s),V)
```

Kode Sumber 4.4 Implementasi Pembuatan *Vector* Dokumen

Vector dokumen tersebut akan dihitung jaraknya dengan *vector* lainnya dengan menggunakan rumus *Cosine Distance* seperti pada Kode Sumber 4.5.

```
1 from scipy.spatial import distance
2
3 big_rank = dict()
4 sim_treshold = 0.6
5
6 for doc1 in range(0,self.__docnum__):
7
8     vector_doc1 = [row[doc1] for row in
docs_rep]
9
10    for doc2 in range(doc1,self.__docnum__):
11
12        vector_doc2 = [row[doc2] for row in
docs_rep]
13
14
15        distcos = distance.cosine(vector_doc1,
16                                vector_doc2)
17        if distcos > sim_treshold :
18            continue
19        big_rank[str(doc1+1) + "-" +
20                str(doc2+1)] =
distcos
```

Kode Sumber 4.5 Implementasi Perhitungan Jarak *Vector* Dokumen

Perhitungan jarak *vector* dengan *Cosine Distance* dilakukan menggunakan pustaka *scipy*. Variabel *distcos* menyimpan nilai perhitungan jarak antar *vector* dokumen. Jika nilai jarak *vector* tidak melebihi dari batas yang telah ditentukan pada variabel *sim_treshold*, maka pasangan dokumen dianggap memiliki kemiripan. Keluaran dari tahap ini adalah variabel *big_rank* yang menyimpan pasangan dokumen yang memiliki kemiripan.

4.2.3. Implementasi Tahap Konstruksi Graf

Subbab ini membahas implementasi tahap konstruksi graf. Graf yang dibuat memiliki pelamar dan pekerjaan sebagai *node*. *Node – node* tersebut akan dihubungkan dengan tiga jenis relasi sebagai *edge*. Ketiga relasi tersebut adalah relasi *similar*, *like*, dan *apply*. Masing – masing relasi akan diimplementasikan dalam bentuk matriks *adjacency*. Matriks *adjacency* tiap relasi akan berukuran $d \times d$ dimana d merepresentasikan dokumen.

```

1 len_sbr = len(sorted_big_rank)
2 self.sim_adj_matrix =
  zeros((self.__docnum__,self.__docnum__))
3
4 for i in xrange(0,len_sbr):
5     pair_node = sorted_big_rank[i][0].split("-
  ")
6     sim_score = sorted_big_rank[i][1]
7
8     node1 = int(pair_node[0])
9     node2 = int(pair_node[1])
10
11 self.sim_adj_matrix[node1-1][node2-1] = 1
12 self.sim_adj_matrix[node2-1][node1-1] = 1

```

Kode Sumber 4.6 Implementasi Matriks *Adjacency* Relasi *Similar*

Matriks *adjacency* relasi *similar* didapatkan melalui keluaran tahap pengukuran kemiripan entitas seperti pada Kode Sumber 4.6. Matriks *adjacency* relasi *similar* disimpan dalam variabel *sim_adj_matrix*. Urutan pasangan - pasangan entitas yang didapatkan dari tahap pengukuran kemiripan entitas terdapat pada variabel *sorted_big_rank*. Relasi ini merupakan relasi *bidirectional* sehingga masing – masing entitas pada pasangan entitas memiliki masukan dan keluaran.

Matriks *adjacency* relasi *like* didapatkan melalui masukan *file feedback* pelamar terhadap pekerjaan. *Feedback* yang memiliki jenis *like* pada *file* masukan akan disimpan dalam variabel *like_input*. Variabel tersebut digunakan untuk membuat matriks *adjacency* relasi *like* seperti pada Kode Sumber 4.7. Matriks *adjacency* relasi *like* disimpan dalam variabel *like_adj_matrix*.

```

1 like_adj_matrix =
  zeros((self.__docnum__,self.__docnum__))
2 len_like_input = len(like_input)
3
4 for it in range(0, len_like_input):
5
6     nodes = like_input[it].split('-')
7     node1 = int(nodes[0])
8     node2 = int(nodes[1])
9
10    if(node1 <= self.__docnum__
11        and node2 <= self.__docnum__):
12        like_adj_matrix[node1-1][node2-1] = 1
13    else:
14        print "Like input Error"
15        exit

```

Kode Sumber 4.7 Implementasi Matriks *Adjacency* Relasi *Like*

Matriks *adjacency* relasi *apply* didapatkan melalui masukan *file feedback* pelamar terhadap pekerjaan. *Feedback* yang memiliki jenis *apply* pada *file* masukan akan disimpan dalam

variabel *apply_input*. Variabel tersebut digunakan untuk membuat matriks *adjacency* relasi *apply* seperti pada Kode Sumber 4.8. Matriks *adjacency* relasi *apply* disimpan dalam variabel *apply_adj_matrix*.

```

1 apply_adj_matrix =
zeros((self.__docnum__,self.__docnum__))
2 len_apply_input = len(apply_input)
3
4 for it in range(0, len_apply_input):
5
6     nodes = apply_input[it].split('-')
7     node1 = int(nodes[0])
8     node2 = int(nodes[1])
9
10    if(node1 <= self.__docnum__
11        and node2 <= self.__docnum__) :
12        apply_adj_matrix[node1-1][node2-1] = 1
13    else :
14        print "Apply input Error"
15        exit

```

Kode Sumber 4.8 Implementasi Matriks *Adjacency* Relasi *Apply*

4.2.4. Implementasi Tahap *Node Ranking* pada Graf dengan Algoritma 3A (*Actor, Asset, Activities*)

Subbab ini membahas implementasi tahap pengukuran tingkat *rank node* pada graf. Matriks *transition* dan matriks *dangling node* dibuat untuk masing – masing jenis relasi pada graf. Kedua matriks tersebut dibuat dengan menggunakan matriks *adjacency* dari masing – masing jenis relasi pada graf, seperti pada Kode Sumber 4.10.

```

1 """ SIMILIAR TRANSITION matrix """
2 self.sim_t_matrix =
zeros((self.__docnum__,self.__docnum__))
3
4 for i in xrange(0,self.__docnum__ ):
5     for j in xrange(0, self.__docnum__ ):
6         if((self.sim_adj_matrix[j][i] == 1)
and (i
7             != j)):
8             self.sim_t_matrix[i][j] =
9
10 self.sim_outdegree[j]
11 """ SIMILIAR DANGLING matrix """
12 self.sim_d_matrix =
zeros((self.__docnum__,self.__docnum__))
13
14 for i in xrange(0,self.__docnum__ ):
15     if (self.sim_outdegree[i] == 0):
16         for j in xrange(0, self.__docnum__ ):
17             self.sim_d_matrix[j][i] =
18
19 1/float(self.__docnum__)
20 """ LIKE TRANSITION matrix """
21 like_t_matrix =
zeros((self.__docnum__,self.__docnum__))
22
23 for i in xrange(0,self.__docnum__):
24     for j in xrange(0, self.__docnum__):
25         if((like_adj_matrix[j][i] == 1) and (i
26             !=
27                 j)):
28             like_t_matrix[i][j] =
29
30 1/like_outdegree[j]

```

**Kode Sumber 4.9 Implementasi Pembuatan Matriks
Transition & Matriks Dangling Node untuk Setiap Jenis
Relasi Graf (Bagian 1)**

```

1 """ LIKE DANGLING matrix """
2 like_d_matrix =
zeros((self.__docnum__,self.__docnum__))
3
4 for i in xrange(0,self.__docnum__ ):
5     if (like_outdegree[i] == 0):
6         for j in xrange(0, self.__docnum__ ):
7             like_d_matrix[j][i] =
8
1/float(self.__docnum__)
9
10 """ APPLY TRANSITION matrix """
11 apply_t_matrix =
zeros((self.__docnum__,self.__docnum__))
12
13 for i in xrange(0,self.__docnum__ ):
14     for j in xrange(0, self.__docnum__ ):
15         if((apply_adj_matrix[j][i] == 1) and
(i !=
16             j)):
17             apply_t_matrix[i][j] =
18                 1/apply_outdegree[j]
19
20 """ APPLY DANGLING matrix """
21 apply_d_matrix =
zeros((self.__docnum__,self.__docnum__))
22
23 for i in xrange(0,self.__docnum__ ):
24     if (apply_outdegree[i] == 0):
25         for j in xrange(0, self.__docnum__ ):
26             apply_d_matrix[j][i] =
27
1/float(self.__docnum__)

```

Kode Sumber 4.10 Implementasi Pembuatan Matriks Transition & Matriks Dangling Node untuk Setiap Jenis Relasi Graf (Bagian 2)

Matriks *transition* dan *dangling node* digunakan dalam pembuatan matriks m atau representasi *random-walk* pada graf. Parameter *lamda*, *damping factor*, *personalization factor*, dan bobot masing-masing jenis relasi graf diambil dari masukan pengguna sistem. Pembuatan matriks m dapat dilihat pada Kode Sumber 4.11.

```

1 alf = float(self.txtVarLamda.text())
2 dam = float(self.txtVarD.text())
3 Bu = float(self.txtVarPu.text())
4 Ws = float(self.txtVarWs.text())
5 Wl = float(self.txtVarWL.text())
6 Wa = float(self.txtVarWa.text())
7 N = self.__docnum__
8 I = ones((self.__docnum__,self.__docnum__))
9 E = 3
10 zigma_w = 0
11 for i in xrange(0,E) :
12     if (i==0) :
13         t = self.sim_t_matrix
14         d = self.sim_d_matrix
15         w = Ws
16     elif (i==1) :
17         t = like_t_matrix
18         d = like_d_matrix
19         w = Wl
20     else :
21         t = ply_t_matrix
22         d = ply_d_matrix
23         w = Wa
24     zigma_w += w*(t+d)
25 m = alf/float(N) * I + dam * zigma_w + Bu *
ta_u_matrix

```

Kode Sumber 4.11 Implementasi Pembuatan Matriks Representasi *Random-Walk*

Vector im yang elemennya merupakan nilai *rank* tiap *node* dalam graf, didapatkan melalui proses iterasi menggunakan matriks *m* representasi *random-walk*. Pembuatan *vector im* dapat dilihat pada Kode Sumber 4.12. Nilai *vector im* awal adalah probabilitas tiap *node* yang terdistribusi sama rata.

```

1 """ initial random walk vector """
2 im = zeros(self.__docnum__)
3 for i in range(0,self.__docnum__):
4     im[i] = 1/float(self.__docnum__)
5
6 """ random walk iteration """
7 for it in range(0,50):
8     im = dot(m,transpose(im))

```

Kode Sumber 4.12 Implementasi Pembuatan *Vector Rank Node*

```

1 """ sorting importance rank """
2 imp_rank = dict()
3 node_n = 1
4 for rank_score in im :
5     imp_rank[node_n] = rank_score
6     node_n += 1
7 sorted_imp_rank = sorted(imp_rank.items(),
8 key=operator.itemgetter(1))

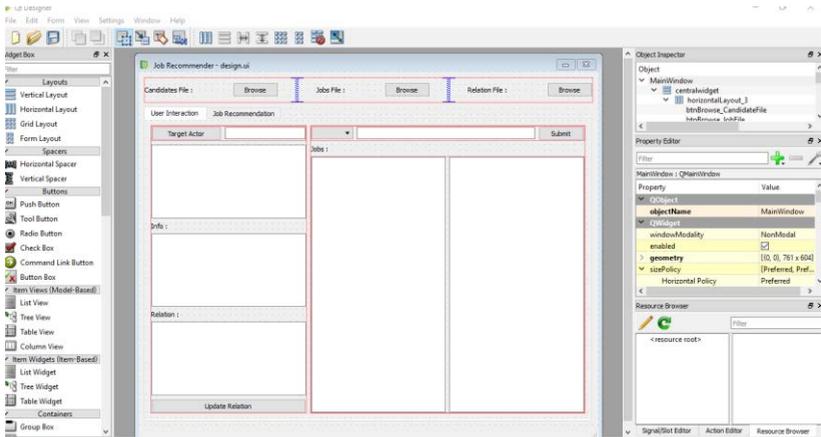
```

Kode Sumber 4.13 Implementasi Pengurutan Elemen *Vector Rank*

Elemen *vector rank* node tersebut diurutkan dari yang terbesar ke yang terkecil, seperti pada Kode Sumber 4.13. Rekomendasi pekerjaan dibuat dari urutan atas elemen *vector* sampai ke urutan yang telah ditentukan pengguna.

4.3. Implementasi Antarmuka Pengguna

Implementasi tampilan antarmuka sistem rekomendasi ini dibuat dengan menggunakan pustaka PyQt. Aplikasi pengembangan yang digunakan untuk membuat desain antarmuka adalah QtDesigner. Atribut – atribut pada antarmuka sistem disusun menggunakan metode *drag & drop* pada aplikasi QtDesigner seperti pada Gambar 4.1.



Gambar 4.1 Implementasi Penyusunan Atribut Antarmuka Sistem Pada QtDesigner

Aplikasi QtDesigner akan menghasilkan kode sumber dari desain antarmuka yang telah dibuat dalam *file* berekstensi *.ui*. Agar dapat digunakan dalam bahasa pemrograman python, *file* tersebut di *compile* dengan pustaka PyQt pada Windows *console*. Kode sumber antarmuka yang telah di-*compile* dapat digunakan dalam sistem dengan melakukan *import* seperti pada Kode Sumber 4.14.

```
1 from PyQt4 import QtGui, QtCore
2 import design
```

Kode Sumber 4.14 Implementasi *Import* Antarmuka pada Sistem

4.3.1. Implementasi Antarmuka Menu File Masukan

Pada antarmuka ini pengguna dapat memilih file data masukan yang terdiri dari data pelamar, data pekerjaan, dan data *feedback* pelamar terhadap pekerjaan. Pada antarmuka ini terdapat tiga tombol *browse* untuk memasukkan file masing - masing data masukan. Tombol tersebut akan membuka sebuah *file browser* untuk mencari file masukan yang memiliki ekstensi .csv.



Gambar 4.2 Menu File Masukan

Pada Gambar 4.2 atribut - atribut pada menu file masukan telah diberikan nomor. Terlihat tombol untuk melakukan *file browsing* untuk file masukan data pelamar, data pekerjaan, dan data *feedback* berturut - turut adalah nomor 2, 4, dan 6. Spesifikasi atribut juga dapat dilihat pada Lampiran 1 Tabel 8.1.

4.3.2. Implementasi Antarmuka Halaman Interaksi Pelamar

Pada antarmuka ini pengguna dapat mengakses seluruh data pelamar, data pekerjaan, dan data *feedback*. Pengguna juga dapat memilih pekerjaan mana yang akan diberikan tanda ketertarikan *like* dan *apply* oleh pelamar tertentu. Spesifikasi atribut antarmuka dapat dilihat pada Gambar 4.3. Hasil implementasi antarmuka ini dapat dilihat pada Lampiran 1 Tabel 8.2. Pada gambar antarmuka tersebut atribut antarmuka telah diberikan nomor.

Terlihat pada atribut nomor 2, pengguna dapat memasukkan *id* data pelamar untuk mengetahui data profil pelamar dan data *feedback* oleh pelamar tersebut. Detil data pelamar ditandai oleh atribut nomor 4 dan data *feedback* oleh pelamar ditandai oleh atribut nomor 5. Pada atribut nomor 5 pengguna dapat mengubah data *feedback* pelamar dengan melakukan *click* pada tombol atribut nomor 6.

1 User Interaction 2 Job Recommendation 7 Like 8 JD-1536 9 Submit

3 Target Actor 1 JC-3

3	1	2
1	JC-1	Cheyenne Jenkins
2	JC-2	Nerette Wooten
3	JC-3	Jermaine Carne...

Info :

4

Name :
Jermaine Cameron

Experience :
Human Resource Coordinator - All Skill Minority Manpower, Inc; Production Assistant - Fontao Management Inc; Sales Associate - Foot Locker

Relation :

1	2	3	5
1	JC-3	Like	JD-1507
2	JC-3	Apply	JD-117
3	JC-3	Like	JD-418

6 Update Relation

Jobs :

10	1	2	3
1328	JU-1327	DEPT OF ENVRIR...	INSTRUMENTA...
1529	JD-1528	DEPT OF ENVRIR...	Gardener
1530	JD-1529	ADMIN FOR C...	Operations Ma...
1531	JD-1530	DEPT OF HEAL...	Audits & Acco...
1532	JD-1531	DEPARTMENT ...	Director, Qualit...
1533	JD-1532	DEPT OF HEAL...	College Aide (1...
1534	JD-1533	DEPT OF ENVRIR...	Senior Engineer
1535	JD-1534	NYC HOUSING ...	Temporary Pain...
1536	JD-1535	NYC HOUSING ...	Senior Manager...
1537	JD-1536	ADMIN FOR C...	Claims Analyst
1538	JD-1537	NYC HOUSING ...	Assistant Direct...
1539	JD-1538	DEPARTMENT ...	Deputy Proara...

11

The Administration for Children's Services, in conjunction with the State and City has entered into a Foster Care Title IV-E Waiver Program. Financially, the City has received a five year, capped Federal allocation of \$5 million to support Foster Care activities. Programmatically, there is a lessening of restrictions on programmatic activities, allowing the agency to work more creatively with provider agencies to lower Foster Care lengths of stay and to determine best case-practice scenarios to be integrated into the Child Welfare system on a more permanent basis. The Claims Analyst position reports directly to the Assistant Director for Claiming. The Claims Analyst will be responsible for: Learn and understand the Claiming instructions and guidelines that govern the Waiver. Ensure that all data for Waiver related expenses are accurately reported, monitored, claimed and reimbursed according to the detailed guidelines put forth in the Waiver. Prepare technical reports as requested which summarize Waiver related data including child information, expenses, claims and revenues.

Gambar 4.3 Halaman Interaksi Pelamar

User Interaction

Job Recommendation

Check Similarity

Relation :

1	2	3	1
1 JC-651	Apply	JD-332	
2 JC-651	Like	JD-1362	
3 JC-651	Like	JD-1371	
4 JC-922	Like	JD-1143	
5 JC-922	Like	JD-1529	
6 JC-984	Apply	JD-1026	
7 JC-984	Apply	JD-443	
8 JC-650	Apply	JD-542	

Similarity : 3

Target Actor : JC-16 Recommendation : 11

Probability Variable

Lands : 0.1	5
d : 0.3	6
Pu : 0.6	7
Ws : 0.2	8
Wa : 0.4	9
Wc : 0.4	10

Find Recommendations 12

1	2	3	4	5	13	14
1 JD-363	IT Business Analyst	it	Liked	0.024059101354	Job Candidate : JC-16	
2 JD-711	Data Center Associate	it	Liked	0.0241692541006	Class : it	
3 JD-375	Data Entry, Bureau of HIV/AIDS Prevention and Control	it	Liked	0.0241554904127	Total Recommendation : 10	
4 JD-554	Database Developer, IV	it	Applied	0.018159859627	Class Match : 9	
5 JD-158	Oracle Database Administrator	it	Applied	0.0181517640764	Recommendation Accuracy : 90.0 %	
6 JD-38	Computer Specialist (SW)	it	Applied	0.0181377993887		
7 JD-5	IT Asset Management College Aide	it	Applied	0.0181313774851		
8 JD-477	Senior Java Developer IV	it	Recommended	0.0023043447873		

Gambar 4.4 Halaman Rekomendasi Pekerjaan

Data seluruh pekerjaan akan diperlihatkan melalui atribut nomor 10. Pengguna dapat melihat isi dari kolom data pekerjaan secara individual dengan melakukan *click* pada kolom dan isi kolom tersebut akan terlihat pada atribut nomor 11.

Pembuatan data masukan *feedback* dapat dilakukan oleh pengguna. Pembuatan data *feedback* dilakukan dengan menentukan masukan *id* pelamar yang akan memberikan ketertarikan pada atribut nomor 2, masukan *id* pekerjaan yang akan diberikan tanda ketertarikan pada atribut nomor 8, dan jenis tanda ketertarikan pada atribut nomor 7. Setelah *id* pelamar, *id* pekerjaan, dan tanda ketertarikan telah ditentukan pembuatan data *feedback* dilakukan dengan melakukan *click* pada tombol atribut nomor 9.

4.3.3. Implementasi Antarmuka Halaman Rekomendasi Pekerjaan

Pada antarmuka ini proses pemberian rekomendasi pekerjaan untuk pelamar dilakukan. Proses pengukuran kesamaan dokumen teks data dapat juga dilakukan pada antarmuka ini. Urutan pasangan dokumen teks yang memiliki kemiripan teks dapat terlihat pada antarmuka ini.

Spesifikasi atribut antarmuka dapat dilihat pada Gambar 4.4. Hasil implementasi antarmuka ini dapat dilihat pada Lampiran 1 Tabel 8.3. Pada gambar antarmuka tersebut atribut antarmuka telah diberikan nomor.

Pada atribut nomor 1 dapat terlihat data *feedback* oleh seluruh pelamar. Pengukuran kemiripan dokumen teks data dapat dilakukan dengan melakukan *click* pada tombol atribut nomor 2 dan hasil pengukuran dapat dilihat pada atribut nomor 3.

Pemberian rekomendasi pekerjaan dapat dilakukan dengan memasukkan *id* pelamar yang akan diberikan rekomendasi pekerjaan pada atribut nomor 4, memasukkan nilai parameter dan bobot relasi pada atribut 5 sampai 10, dan memasukkan jumlah rekomendasi pekerjaan yang akan diberikan pada atribut 11.

Setelah itu tombol atribut nomor 12 dilakukan *click* dan hasil rekomendasi pekerjaan dapat terlihat pada atribut nomor 13. Nilai presisi hasil rekomendasi dapat terlihat pada atribut nomor 14.

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap kinerja sistem rekomendasi dalam memberikan rekomendasi pekerjaan untuk pelamar.

5.1. Lingkungan Uji Coba

Lingkungan uji coba yang digunakan dalam pembuatan Tugas Akhir ini meliputi perangkat keras berupa komputer dan sistem operasi dengan spesifikasi seperti pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Uji Coba

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i5-4200U CPU @ 1.60GHz Memori (RAM): 8.00 GB
Perangkat lunak	Sistem Operasi: Microsoft Windows 10 Pro 64-bit

5.2. Cara Evaluasi Kinerja Sistem Rekomendasi

Evaluasi kinerja sistem rekomendasi pekerjaan ini adalah dengan menggunakan nilai presisi. Perhitungan nilai presisi akan mencari pekerjaan yang relevan dari set pekerjaan yang direkomendasikan. Rekomendasi pekerjaan yang relevan adalah pekerjaan yang memiliki kategori atau kelas yang sama dengan kategori atau kelas dari pelamar yang diberikan rekomendasi. Klasifikasi data pelamar dan pekerjaan seperti pada Bab 3.2.2 akan dijadikan sebagai *ground truth* untuk menentukan relevansi rekomendasi pekerjaan.

Salah satu data pelamar adalah pelamar dengan *id JC-19* yang memiliki kelas *medical* karena memiliki pengalaman kerja sebagai *Medical Assitant* pada institusi *Interboro Heart Associate* seperti

pada Tabel 5.2. Pelamar tersebut akan diberikan rekomendasi pekerjaan sebanyak 10 pekerjaan. Hasil rekomendasi pekerjaan untuk pelamar *JC-19* adalah seperti pada Tabel 5.3.

Tabel 5.2 Data Pelamar *JC-19*

Id	Class	Name	Experience	Education	Degree
JC-19	Medical	Kavita Samaroo, RN	Medical Assistant - Interboro Heart Associates; ...	Borough of Manhattan Community College Continuing Education	Certification

Pada tabel hasil rekomendasi tersebut terlihat terdapat pekerjaan yang memiliki kolom *Status* yang bernilai *Liked* atau *Applied*. Pekerjaan – pekerjaan tersebut adalah pekerjaan yang sebelumnya telah diberi tanda ketertarikan oleh pelamar. Riwayat ketertarikan pelamar terhadap pekerjaan tersimpan dalam data *feedback* pelamar terhadap pekerjaan. Pekerjaan – pekerjaan tersebut tidak termasuk dalam 10 rekomendasi pekerjaan.

Nilai presisi sistem rekomendasi pekerjaan dihitung dengan mencari rekomendasi pekerjaan yang relevan. Rekomendasi pekerjaan yang relevan untuk pelamar *JC-19* adalah pekerjaan yang memiliki kelas yang sama dengan pelamar *JC-19* yaitu kelas *medical*. Rekomendasi pekerjaan yang relevan ditandai dengan kolom *Status* yang bernilai *Relevant*.

Pekerjaan - pekerjaan yang direkomendasikan untuk pelamar masing - masing memiliki nilai rekomendasi. Nilai rekomendasi menunjukkan tingkat kepentingan rekomendasi pekerjaan oleh sistem rekomendasi. Perhitungan nilai rekomendasi telah dijelaskan secara mendalam pada Bab 2.3.

Tabel 5.3 Hasil Rekomendasi Pekerjaan untuk Pelamar *JC-19*

No	Id Pekerjaan	Nama Pekerjaan	Kelas Pekerjaan	Status	Nilai Rekomendasi
1	JD-604	Public Health Assistant, Bureau of Maternal Infant and Reproductive Health	Medical	Applied	-
2	JD-76	School Mental Health Evaluation Specialist, Bureau of Children, Youth & Families	Medical	Liked	-
3	JD-108	Clinical Care Coordinator / NYC Supported Transition and Recovery Team (NYCSTART), Mental Health	Medical	Liked	-
4	JD-153	SCHOOL MENTAL HEALTH CONSULTANT SUPERVISOR	Medical	Relevant	0.1124
5	JD-147	School Mental Health Consultant	Medical	Relevant	0.1114

No	Id Pekerjaan	Nama Pekerjaan	Kelas Pekerjaan	Status	Nilai Rekomendasi
6	JD-637	Purchasing Agent - Buyer, Office of the Agency Chief Contracting Officer	Supply-chain	Irrelevant	0.1083
7	JD-580	Junior Public Health Nurse, Bureau of Maternal Infant and Reproductive Health	Medical	Relevant	0.0986
8	JD-509	Change Order Administrators	Supply-chain	Irrelevant	0.0955
9	JD-488	Public Health Advisor, Bureau of School Health	Medical	Relevant	0.0950
10	JD-375	Asthma Health Educator, Bureau of School Health	Medical	Relevant	0.0950
11	JD-571	Clinical Program Manager, Bureau of School Health	Medical	Relevant	0.0948
12	JD-327	Public Health Advisor, Vision	Medical	Relevant	0.0945

No	Id Pekerjaan	Nama Pekerjaan	Kelas Pekerjaan	Status	Nilai Rekomendasi
		Pre-K Program, Bureau of School Health			
13	JD-661	City Medical Specialist, Bureau of School Health	Medical	Relevant	0.0943
Jumlah Pekerjaan yang Relevan			8		
Nilai Pেসisi			80%		

Terlihat pada Tabel 5.3, pekerjaan yang paling cocok untuk pelamar *JC-19* adalah pekerjaan dengan *id JD-153* sebagai *School Mental Health Consultant Supervisor*. Jumlah pekerjaan yang relevan untuk pelamar *JC-19* adalah 8 pekerjaan dari 10 rekomendasi pekerjaan. Oleh karena itu, nilai presisi rekomendasi pekerjaan untuk pelamar *JC-19* adalah 80%.

5.3. Data Uji Coba

Data yang digunakan untuk uji coba implementasi sistem rekomendasi untuk pemberian rekomendasi pekerjaan untuk pelamar adalah data pelamar, data pekerjaan, dan data *feedback* pelamar terhadap pekerjaan. Jumlah data yang dipakai setiap skenario uji coba dapat dilihat pada Tabel 5.4. Contoh data uji coba dapat dilihat pada Lampiran 2 .

Pada uji coba Skenario 2 data pelamar yang dipakai adalah 5 data, data pekerjaan yang dipakai berjumlah 10 data, dan data *feedback* yang dipakai berjumlah 11 data. Pada uji coba Skenario 1, 3, dan 4 data pelamar dipakai berjumlah 855 data, data pekerjaan yang dipakai berjumlah 1297 data, dan data *feedback* yang dipakai berjumlah 2290 data.

Pada Skenario 5 terdapat 2 set data yang akan dipakai. Set pertama berisi data pelamar berjumlah 259 data, data pekerjaan berjumlah 1297 data, dan data *feedback* berjumlah 723 data. Set 2 berisi data pelamar sebanyak 224 data, data pekerjaan berjumlah 736 data, dan data *feedback* berjumlah 550 data.

Tabel 5.4 Jumlah Data Uji Coba

Data	Jumlah Data					
	Skenario 1	Skenario 2	Skenario 3	Skenario 4	Skenario 5	
					Set 1	Set 2
Pelamar	855	5	855	855	259	224
Pekerjaan	1297	10	1297	1297	1297	736
<i>Feedback</i>	2290	11	2290	2290	732	550

Pada uji coba ini setiap pelamar pada data pelamar akan diberikan rekomendasi pekerjaan. Masing – masing pelamar akan diberikan rekomendasi pekerjaan sebanyak 10 pekerjaan, dan tidak termasuk pekerjaan yang telah diberikan tanda ketertarikan oleh pelamar.

5.4. Skenario Uji Coba

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang telah dilakukan. Terdapat beberapa skenario uji coba yang telah dilakukan, diantaranya yaitu:

1. Perbandingan nilai presisi rekomendasi berdasarkan jenis sistem rekomendasi yang berbeda (*content-based*, *collaborative*, dan *hybrid*).
2. Perbandingan nilai *rank node*, berdasarkan komposisi nilai parameter yang berbeda.
3. Perbandingan nilai presisi rekomendasi, berdasarkan komposisi nilai parameter yang berbeda - 1 (untuk mencari rasio komposisi nilai parameter terbaik).

4. Perbandingan nilai presisi rekomendasi, berdasarkan komposisi nilai parameter yang berbeda - 2 (untuk mencari nilai presisi terbaik).
5. Perbandingan rata – rata nilai presisi rekomendasi untuk setiap kelas pelamar menggunakan set data pelamar yang berbeda.

5.5.Skenario Pengujian 1: Perbandingan nilai presisi rekomendasi berdasarkan jenis sistem rekomendasi yang berbeda (*content-based*, *collaborative*, dan *hybrid*)

Pada skenario pengujian ini sebanyak 855 pelamar akan diberikan rekomendasi berdasarkan 3 jenis sistem rekomendasi yang berbeda. Ketiga jenis sistem rekomendasi tersebut adalah sistem rekomendasi berjenis *content-based*, *collaborative*, dan *hybrid*. Skenario ini bertujuan untuk mengetahui jenis sistem rekomendasi apa yang dapat menghasilkan nilai presisi rekomendasi tertinggi. Perbedaan ketiga jenis sistem rekomendasi tersebut terlihat pada Tabel 5.5. Kelas data pelamar dan pekerjaan yang digunakan pada skenario ini dapat dilihat pada Tabel 3.1.

Sistem rekomendasi *content-based* hanya menggunakan data keluaran Tahap Pengukuran Kemiripan dan tidak memakai data masukan *feedback* pelamar terhadap pekerjaan. Graf yang dihasilkan pada Tahap Konstruksi Graf oleh sistem rekomendasi berjenis ini hanya memiliki relasi *similar*.

Sistem rekomendasi *collaborative* hanya memanfaatkan data *feedback* pelamar terhadap pekerjaan dan tidak memakai data keluaran pada Tahap Pengukuran Kemiripan . Graf yang dihasilkan pada Tahap Konstruksi Graf oleh sistem rekomendasi berjenis ini hanya memiliki relasi *like* dan *apply*.

Sistem rekomendasi *hybrid* merupakan gabungan dari *content-based* dan *collaborative* yang memakai semua data pada semua tahap. Graf yang dihasilkan pada Tahap Konstruksi Graf oleh sistem rekomendasi berjenis ini memiliki relasi *similar*, *like*, dan *apply*.

Tabel 5.5 Perbedaan Jenis Sistem Rekomendasi

No	Jenis Sistem Rekomendasi	Relasi Graf		
		<i>Similar</i>	<i>Like</i>	<i>Apply</i>
1	<i>Content-Based</i>	√	x	x
2	<i>Collaborative</i>	X	√	√
3	<i>Hybrid</i>	√	√	√

Setiap 855 pelamar akan diberikan 10 rekomendasi pekerjaan untuk setiap jenis sistem rekomendasi dan dihitung nilai presisi rekomendasinya.

Tabel 5.6 Hasil Nilai Presisi Skenario 2

No	Jenis Sistem Rekomendasi	Presisi (%)
1	Content-Based	14.65
2	Collaborative	36.90
3	Hybrid	52.37

Nilai presisi rata – rata rekomendasi pekerjaan untuk setiap jenis sistem rekomendasi dapat dilihat dalam Tabel 5.6. Tabel tersebut menunjukkan bahwa sistem rekomendasi berjenis *Hybrid* dapat menghasilkan nilai presisi yang paling tinggi.

5.6.Skenario Pengujian 2: Perbandingan nilai *rank node*, berdasarkan komposisi nilai parameter yang berbeda.

Skenario 2 dilakukan untuk mengamati pengaruh setiap parameter secara individual di dalam Algoritma *node ranking* 3A terhadap nilai *rank* yang dihasilkan. Deskripsi parameter pada Algoritma *node ranking* 3A dapat dilihat pada Tabel 2.5.

Skenario ini tidak menggunakan data pelamar dan pekerjaan yang telah terklasifikasi karena pada skenario ini tidak menghitung nilai presisi sistem rekomendasi. Perbandingan hanya dilakukan terhadap hasil *node ranking* pada graf saja. Skenario ini juga tidak menggunakan data yang berjumlah besar untuk memudahkan visualisasi graf yang dihasilkan. Jumlah data yang digunakan pada skenario ini dapat dilihat pada Tabel 5.4. Kelas data pelamar dan pekerjaan yang digunakan pada skenario ini dapat dilihat pada Tabel 3.1.

Detil data yang digunakan pada skenario ini dapat dilihat pada Lampiran 2. Data pelamar, pekerjaan, dan *feedback* tersebut menghasilkan graf seperti pada Gambar 5.1. Pada graf terlihat pelamar dengan *id JC-1* menyukai pekerjaan dengan *id JD-2*, *JD-8*, dan *JD-3*. Pelamar *JC-1* juga telah mencoba melamar terhadap pekerjaan dengan *id JD-4* melalui sistem.

Pada graf di Gambar 5.1 akan dilakukan proses *node ranking* menggunakan komposisi nilai parameter yang berbeda seperti pada Tabel 5.7. Jumlah nilai ketiga parameter adalah 1 seperti pada Persamaan 2.1. Pada setiap komposisi, akan ditonjolkan nilai salah satu parameter. Hal ini dilakukan untuk mengetahui pengaruh parameter secara individual terhadap hasil proses *node ranking*.

Tabel 5.7 Komposisi Nilai Parameter Skenario 2

No	Nama Parameter	Komposisi Nilai Parameter		
		1	2	3
1	λ	0.8	0.1	0.1
2	d	0.1	0.8	0.1
3	Pu	0.1	0.1	0.8

Terlihat pada komposisi 1 di Tabel 5.7, nilai parameter λ adalah 0,8, nilai parameter d adalah 0,1, dan nilai parameter Pu adalah 0,1. Pada komposisi ini nilai parameter yang ditonjolkan

Tabel 5.8 Keluaran Komposisi 1 pada Skenario 2

No	Node Target : JC-1	
	Node	Nilai Rank
1	JC-1	0.06671
2	JD-5	0.06666
3	JD-1	0.06666
4	JD-7	0.06666
5	JD-6	0.06666
6	JD-4	0.06666
7	JD-2	0.06666
8	JD-9	0.06666
9	JD-3	0.06666
10	JD-8	0.06666
11	JD-10	0.06666
12	JC-4	0.06666
13	JC-5	0.06666
14	JC-3	0.06666
15	JC-2	0.06666

Keluaran urutan nilai *rank node* relatif terhadap *node* "JC-1" menggunakan komposisi nilai parameter 1, 2, dan 3 berturut - turut terlihat pada Tabel 5.8, Tabel 5.9, dan Tabel 5.10. Terlihat masih terdapat data pelamar pada keluaran urutan nilai *rank node*, hal ini disebabkan karena Algoritma *node ranking* 3A menghitung seluruh nilai *rank node* pada graf termasuk *node* pelamar dan pekerjaan. Rekomendasi pekerjaan yang sesungguhnya menghilangkan data - data pelamar dari keluaran tersebut sehingga hanya menyisakan data pekerjaan untuk direkomendasikan kepada pelamar. Namun pada skenario ini data pelamar tetap diperlihatkan

untuk mengetahui pengaruh parameter Algoritma 3A pada keluaran nilai *rank node* pada graf secara keseluruhan.

Tabel 5.9 Keluaran Komposisi 2 pada Skenario 2

No	Node Target : JC-1	
	Node	Nilai Rank
1	JD-1	0.09321
2	JD-5	0.09271
3	JD-6	0.08799
4	JD-7	0.08573
5	JD-2	0.07646
6	JD-4	0.07339
7	JD-9	0.06573
8	JD-3	0.06253
9	JD-8	0.06179
10	JD-10	0.05634
11	JC-4	0.05627
12	JC-1	0.04700
13	JC-5	0.04695
14	JC-3	0.04695
15	JC-2	0.04695

Pada Tabel 5.8 terlihat komposisi 1 yang menonjolkan nilai parameter λ menghasilkan nilai *rank* yang sama untuk setiap *node*. Oleh karena itu, komposisi 1 menunjukkan bahwa nilai parameter λ menaikan nilai *rank* seluruh *node* secara sama rata, sehingga tidak ada *node* yang bernilai *rank* yang 0.

Komposisi nilai parameter 2 yang menonjolkan nilai parameter d (*damping factor*) menunjukkan bahwa *node* dengan *edge* masukan terbanyak pada graf, memiliki nilai *rank* yang tertinggi, seperti pada Tabel 5.9. Pada graf di Gambar 5.1 jumlah

masukannya terbanyak adalah 3 masukannya dan dimiliki oleh *node JD-2, JD-7, JD-1, dan JD-5*. Sehingga, *node - node* tersebut menempati urutan tertinggi pada keluaran urutan nilai *rank node* seperti terlihat pada Tabel 5.9. Karena *node JD-6* pada graf terletak sangat berdekatan dengan *node JD-5 dan JD-1* yang merupakan *node* bernilai *rank* tinggi, maka *node JD-6* juga bernilai *rank* tinggi.

Tabel 5.10 Keluaran Komposisi 3 pada Skenario 2

No	Node Target : JC-1	
	Node	Nilai Rank
1	JC-1	0.999904
2	JD-4	0.000024
3	JD-2	0.000011
4	JD-8	0.000011
5	JD-3	0.000011
6	JD-7	0.000004
7	JD-1	0.000004
8	JD-5	0.000004
9	JD-6	0.000004
10	JD-9	0.000004
11	JD-10	0.000004
12	JC-4	0.000004
13	JC-5	0.000004
14	JC-3	0.000004
15	JC-2	0.000004

Komposisi nilai parameter 3 yang menonjolkan parameter *Pu* (*Personalization Factor*) menunjukkan keluaran nilai *rank* yang tinggi untuk *node - node* yang berdekatan dengan *node target JC-1*. Pada Tabel 5.10 terlihat *node* yang memiliki nilai *rank* tertinggi (selain *node target*) adalah *node JD-4, JD-2, JD-8, JD-3*. Pada graf

di Gambar 5.1 *node - node* tersebut merupakan *node* yang berhubungan langsung dengan *node* target *JC-I*. Ringkasan hasil analisa pengaruh parameter Algoritma 3A terhadap nilai *rank* node pada graf dapat dilihat pada Tabel 5.11.

Tabel 5.11 Ringkasan Hasil Analisa Parameter Skenario 1

No	Simbol Parameter	Nama Parameter	Pengaruh pada Nilai <i>Rank</i> (<i>Rank</i>)
1	λ	lamda	Menaikan (<i>boost</i>) nilai <i>rank</i> semua <i>node</i> sama rata, sehingga tidak ada <i>node</i> yang bernilai 0.
2	d	damping factor	Menaikan (<i>boost</i>) nilai <i>rank</i> dari <i>node - node</i> yang memiliki masukan <i>edge</i>
3	Pu	personalization factor	Menaikan (<i>boost</i>) nilai <i>rank</i> dari <i>node - node</i> di sekitar <i>node target</i>

5.7.Skenario Pengujian 3: Perbandingan nilai presisi rekomendasi, berdasarkan komposisi nilai parameter yang berbeda - 1.

Skenario 3 memiliki tujuan untuk mengamati pengaruh parameter *node ranking* secara individual dalam menghasilkan nilai presisi pada proses sistem rekomendasi. Pada skenario ini akan dilakukan perbandingan hasil nilai presisi sistem rekomendasi berdasarkan penggunaan komposisi nilai parameter yang berbeda.

Skenario ini akan menggunakan komposisi nilai parameter Algoritma *node ranking* 3A yang sama dengan Skenario 2.

Namun, pada Skenario 2, komposisi nilai parameter digunakan untuk melakukan perbandingan nilai *rank* hasil proses *node ranking*, sedangkan pada skenario ini komposisi nilai parameter digunakan untuk melakukan perbandingan nilai presisi hasil proses sistem rekomendasi.

Data pelamar yang akan diberikan rekomendasi pada skenario ini sama seperti Skenario 1, seperti pada tabel Tabel 5.4. Komposisi nilai parameter Algoritma *node ranking* 3A sama seperti Skenario 2 yaitu seperti pada Tabel 5.7. Kelas data pelamar dan pekerjaan yang digunakan pada skenario ini dapat dilihat pada Tabel 3.1.

Jumlah nilai ketiga parameter adalah 1 seperti pada Persamaan 2.1. Pada setiap komposisi, akan ditonjolkan nilai salah satu parameter. Hal ini dilakukan untuk mengetahui pengaruh parameter secara individual terhadap nilai presisi hasil proses sistem rekomendasi. Terlihat pada komposisi 1 di Tabel 5.7, nilai parameter λ adalah 0,8, nilai parameter d adalah 0,1, dan nilai parameter Pu adalah 0,1. Pada komposisi ini nilai parameter yang ditonjolkan adalah nilai parameter λ dengan nilai 0,8. Komposisi 1 memiliki tujuan untuk mengetahui pengaruh parameter λ pada nilai presisi hasil proses sistem rekomendasi.

Tabel 5.12 Hasil Nilai Presisi Skenario 3

No	Komposisi Nilai Parameter	Presisi (%)
1	Komposisi 1	37.06
2	Komposisi 2	42.50
3	Komposisi 3	50.92

Nilai presisi rata – rata rekomendasi pekerjaan untuk setiap komposisi nilai parameter pada Skenario 3 dapat dilihat dalam Tabel 5.12. Pada tabel tersebut dapat dilihat bahwa Komposisi 3

yang menonjolkan parameter Pu , dapat menghasilkan keluaran nilai presisi rekomendasi paling tinggi dengan nilai 50.92%.

Oleh karena itu, dapat disimpulkan bahwa parameter Pu penting dalam memaksimalkan nilai presisi rekomendasi dan nilainya harus lebih besar daripada parameter lainnya.

5.8.Skenario Pengujian 4: Perbandingan nilai presisi rekomendasi, berdasarkan komposisi nilai parameter yang berbeda - 2.

Pada skenario 4 akan dilakukan perbandingan nilai presisi hasil proses sistem rekomendasi berdasarkan komposisi nilai parameter Algoritma *node ranking* 3A yang berbeda, sama seperti pada Skenario 3. Perbedaan Skenario 3 dan 4 terletak pada komposisi nilai parameter Algoritma *node ranking* 3A yang akan dibandingkan.

Tujuan Skenario 3 dan 4 berbeda. Skenario 4 menggunakan komposisi nilai parameter yang berbeda untuk mencari nilai presisi sistem rekomendasi yang terbaik. Sedangkan Skenario 3 menggunakan komposisi nilai parameter yang berbeda untuk mengetahui pengaruh parameter terhadap nilai presisi sistem rekomendasi secara individual. Sehingga pada Skenario 3 dapat diketahui parameter apa yang harus memiliki nilai paling tinggi untuk menghasilkan nilai presisi sistem rekomendasi yang tinggi.

Setiap komposisi parameter pada Skenario 3 hanya menonjolkan salah satu parameternya untuk mengetahui pengaruh parameter secara individual pada nilai presisi yang dihasilkan. Namun pada Skenario 4 komposisi nilai parameter lebih seimbang dan tidak ada nilai parameter yang ditonjolkan. Pemilihan komposisi nilai parameter pada Skenario 4 menggabungkan hasil kesimpulan dari Skenario 2 dan Skenario 3.

Pada Skenario 2 telah diketahui bahwa parameter λ (*lamda*) hanya meningkatkan nilai *rank* seluruh *node* pelamar dan pekerjaan pada graf agar tidak ada *node* yang tidak memiliki nilai *rank*. Oleh karena itu, dapat disimpulkan bahwa dapat digunakan

nilai 0,1 untuk nilai parameter λ pada setiap komposisi nilai parameter yang akan digunakan di Skenario 4 ini.

Pada Skenario 3 telah diketahui bahwa dengan memaksimalkan nilai parameter P_u dapat meningkatkan keluaran nilai presisi rekomendasi. Sehingga pada komposisi nilai parameter yang akan digunakan di Skenario 4, parameter P_u harus bernilai lebih tinggi daripada parameter lainnya.

Pada skenario ini diajukan 3 komposisi nilai parameter Algoritma *node ranking* 3A seperti pada Tabel 5.13. Nilai parameter λ (*lamda*) pada setiap komposisi adalah 0.1 karena parameter ini hanya berfungsi untuk menghindari *node* pekerjaan yang tidak memiliki nilai *rank* pada graf model rekomendasi, seperti pada analisa hasil uji coba Skenario 2. Kelas data pelamar dan pekerjaan yang digunakan pada skenario ini dapat dilihat pada Tabel 3.1.

Tabel 5.13 Komposisi Nilai Parameter Skenario 4

No	Nama Parameter	Komposisi Nilai Parameter		
		1	2	3
1	λ	0.1	0.1	0.1
2	d	0.4	0.3	0.2
3	P_u	0.5	0.6	0.7

Nilai parameter P_u naik sebanyak 0.1 di setiap komposisi. Nilai parameter P_u pada ketiga komposisi lebih tinggi daripada nilai parameter d sesuai dengan analisa hasil uji coba Skenario 3. Semakin tinggi nilai parameter P_u , nilai parameter d akan semakin rendah karena jumlah ketiga parameter adalah 1, seperti pada Persamaan 2.1. Nilai parameter P_u tidak bisa diperkecil dari nilai parameter P_u yang ada pada komposisi 1 karena akan menyebabkan nilai parameter P_u sama dengan nilai parameter d . Nilai parameter P_u tidak dapat diperbesar dari nilai parameter P_u

yang ada pada komposisi 3 karena akan menyebabkan nilai parameter d sama dengan nilai parameter λ (*lamda*).

Tabel 5.14 Hasil Nilai Presisi Skenario 4

No	Komposisi Nilai Parameter	Presisi (%)
1	Komposisi 1	51.36
2	Komposisi 2	52.37
3	Komposisi 3	52.30

Sebanyak 855 pelamar masing – masing akan diberikan 10 rekomendasi pekerjaan untuk setiap komposisi nilai parameter Algoritma *node ranking* 3A dan dihitung nilai presisi rekomendasinya. Nilai presisi rata – rata rekomendasi pekerjaan untuk setiap komposisi nilai parameter dapat dilihat dalam Tabel 5.14.

Pada tabel tersebut terlihat nilai presisi rekomendasi tertinggi dihasilkan oleh komposisi nilai parameter 2. Oleh karena itu, komposisi nilai parameter Algoritma *node ranking* 3A yang seimbang dan dapat menghasilkan nilai presisi rekomendasi terbaik adalah komposisi nilai parameter 2 yaitu $\lambda = 0.1$, $d = 0.3$, dan $Pu = 0.6$.

5.9.Skenario Pengujian 5: Perbandingan Rata – Rata Nilai Presisi Rekomendasi Pada Setiap Kelas pelamar Menggunakan Jumlah Data Pelamar yang Berbeda.

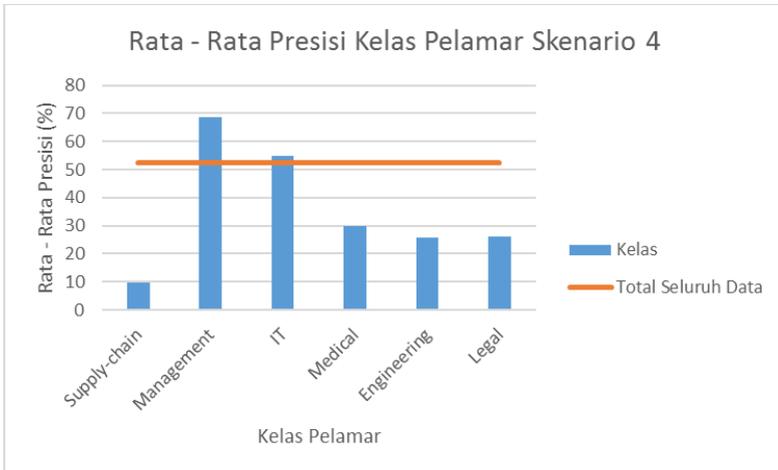
Pada Skenario 4 telah dilakukan uji coba pemberian rekomendasi kepada 855 pelamar dengan menggunakan data pelamar sebanyak 855 data, data pekerjaan sebanyak 1297 data, dan data *feedback* pelamar terhadap pekerjaan sebanyak 2290 data. Analisa hasil uji coba skenario tersebut menyebutkan bahwa nilai presisi rekomendasi terbaik yaitu 52,37% dihasilkan oleh komposisi nilai parameter Algoritma 3A $\lambda = 0.1$, $d = 0.3$, dan $Pu =$

0.6. Namun pada hasil rekomendasi tersebut, nilai presisi rekomendasi pada setiap kelas pelamar berbeda – beda. Hasil rata - rata nilai presisi rekomendasi pada setiap kelas pelamar dalam hasil uji coba Skenario 4 dapat dilihat pada Tabel 5.15. Kelas data pelamar dan pekerjaan yang digunakan pada Skenario 4 dapat dilihat pada Tabel 3.1.

Pada tabel tersebut terlihat pelamar pada kelas *Management* memiliki rata – rata nilai presisi tertinggi yaitu 68.52% dan pelamar pada kelas *Supply-Chain* memiliki rata – rata nilai presisi hanya 9.55%. Sedangkan rata – rata nilai presisi untuk seluruh data adalah 52.37%. Perbedaan yang sangat besar pada rata – rata nilai presisi di antara 2 kelas tersebut membuktikan bahwa rata – rata nilai presisi setiap kelas pada Skenario 4 tidak merata.

Tabel 5.15 Rata - Rata Nilai Presisi Skenario 4

No	Kelas Pelamar	Rata - Rata Nilai Presisi (%)
1	Supply-Chain	9.55
2	Management	68.52
3	IT	54.75
4	Medical/HealthCare	30.00
5	Engineering	25.71
6	Legal	26.11
Seluruh Data		52.37



Gambar 5.2 Rata - Rata Nilai Presisi Skenario 4

Representasi rata – rata nilai presisi Skenario 4 dalam grafik batang dapat dilihat di Gambar 5.2. Pada grafik tersebut terlihat, kelas *management* dan *IT* memiliki rata - rata nilai presisi tertinggi dengan nilai di atas 50%, sedangkan kelas lainnya jauh di bawah 50%. Hal ini dapat disebabkan karena pekerjaan pada kelas - kelas tersebut memiliki nilai *rank* yang tinggi, sehingga sering muncul pada saat pemberian rekomendasi kepada pelamar. Nilai *rank* yang tinggi pada pekerjaan mungkin dipengaruhi oleh banyaknya data *feedback* terhadap pekerjaan tersebut oleh pelamar. Banyaknya data *feedback* terhadap pekerjaan pada suatu kelas dipengaruhi oleh banyak pelamar pada kelas yang sama. Oleh karena itu, kelas *management* dan *IT* memiliki rata - rata nilai presisi tertinggi pada pemberian rekomendasi uji coba Skenario 4 karena jumlah pelamar pada kelas tersebut jauh lebih banyak daripada jumlah pelamar pada kelas - kelas lainnya. Jumlah pelamar pada setiap kelas dalam uji coba Skenario 4 dapat dilihat pada Tabel 3.2.

Tabel 5.16 Set Data Pelamar Skenario 5

No	Kelas	Jumlah	
		Set 1	Set 2
1	Management	52	0
2	IT	63	0
3	Engineering	53	133
4	Supply-Chain	45	45
5	Medical/HealthCare	28	28
6	Legal	18	18
Total		259	224

Pada Skenario 5 ini akan dilakukan uji coba pemberian rekomendasi pekerjaan untuk pelamar dengan jumlah data pelamar yang telah diseimbangkan untuk setiap kelas. Rata – rata hasil nilai presisi seluruh pelamar secara keseluruhan dan rata – rata hasil nilai presisi untuk setiap kelas pelamar akan diamati pada uji coba ini. Skenario ini bertujuan untuk melihat pengaruh jumlah data terhadap nilai presisi rekomendasi.

Pada skenario ini jumlah data pelamar untuk setiap kelas pada Tabel 3.2 akan dikurangi agar setiap kelas memiliki jumlah pelamar yang seimbang dan tidak memiliki selisih yang besar. Terdapat 2 cara penyeimbangan jumlah data yang akan dilakukan. Cara pertama adalah dengan menyeimbangkan jumlah data pelamar untuk setiap kelas pelamar. Cara kedua adalah dengan menghilangkan 2 kelas dengan jumlah data pelamar terbesar. Cara 1 dan 2 akan menghasilkan set data pelamar berturut – turut Set 1 dan Set 2, seperti pada Tabel 5.16.

Pada Set 1 terlihat jumlah data pelamar untuk setiap kelas diseimbangkan dan tidak memiliki selisih yang terlalu besar. Jumlah data pelamar terkecil adalah 18 data pada kelas *legal* dan data pelamar terbanyak adalah 63 data pada kelas *IT*. Total data

pelamar pada Set 1 adalah 259 data, dengan selisih 596 data dari jumlah data pelamar sebelumnya.

Pada Set 2 terlihat bahwa data pelamar pada kelas *management* dan *IT* dihilangkan karena merupakan kelas dengan jumlah data pelamar terbesar pada Tabel 3.2. Data pelamar pada kelas selain kelas *management* dan *IT* tidak mengalami perubahan dan jumlahnya disamakan seperti pada Tabel 3.2. Total data pelamar pada Set 2 adalah 224 data, dengan selisih 631 data dari jumlah data pelamar sebelumnya.

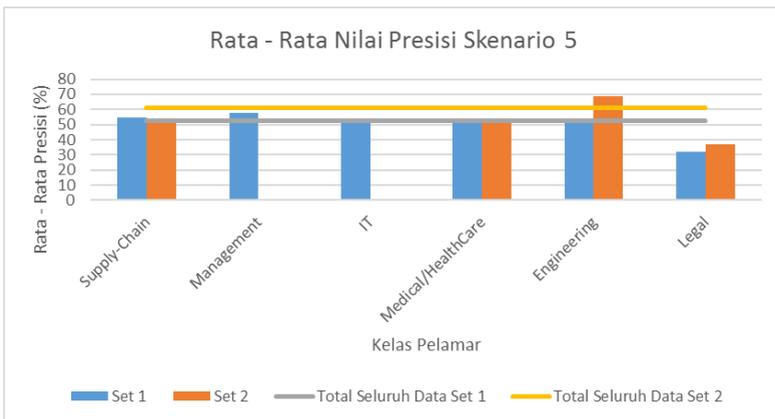
Jumlah data pelamar yang lebih kecil menyebabkan jumlah data *feedback* pelamar terhadap pekerjaan yang lebih kecil yaitu sebanyak 723 data untuk Set 1 dan 550 data untuk Set 2. Data pekerjaan yang akan dipakai untuk Set 1 masih sama seperti data pekerjaan yang dipakai pada Skenario 4 yaitu sebanyak 1297 data. Pada Set 2 data pekerjaan dengan kelas *management* dan *IT* juga ikut dihilangkan sehingga hanya tersisa 736 data untuk dipakai. Setiap pelamar pada Set 1 dan Set 2 akan diberikan rekomendasi pekerjaan sebanyak 10 pekerjaan dan akan dihitung nilai presisi rekomendasinya. Hasil rata – rata nilai presisi rekomendasi untuk setiap set data pelamar dapat dilihat pada Tabel 5.17.

Tabel 5.17 Rata - Rata Nilai Presisi Skenario 5 Setiap Set

No	Kelas Pelamar	Rata - Rata Nilai Presisi (%)	
		Set 1	Set 2
1	Supply-Chain	54.66	54.00
2	Management	57.69	0
3	IT	51.58	0
4	Medical/HealthCare	51.42	52.50
5	Engineering	53.39	68.79
6	Legal	32.22	36.66
Seluruh Data		52.35	61.20

Hasil uji coba Skenario 5 dalam bentuk grafik batang dapat dilihat pada Gambar 5.3. Jika dibandingkan dengan Gambar 5.2 jelas terlihat bahwa rata – rata nilai presisi pada Set 1 lebih seimbang untuk setiap kelas pelamar. Namun rata – rata nilai presisi keseluruhan data pada Set 1 mengalami perubahan yang tidak signifikan dari rata – rata nilai presisi pada uji coba Skenario 4 yaitu 52,35%.

Pada hasil uji coba menggunakan Set 2 pada Tabel 5.17 terlihat kenaikan rata – rata nilai presisi rekomendasi untuk kelas *supply-chain*, *medical*, *engineering*, dan *legal* jika dibandingkan dengan Tabel 5.15. Rata – rata nilai presisi untuk setiap kelas juga terlihat lebih seimbang pada Gambar 5.3. Rata – rata nilai presisi keseluruhan data pada Set 2 ini mengalami peningkatan terhadap rata – rata nilai presisi pada uji coba Skenario 4 yaitu 61,2%.



Gambar 5.3 Rata - Rata Nilai Presisi Skenario 5

Oleh karena itu, dapat disimpulkan bahwa jumlah data pelamar setiap kelas berpengaruh terhadap rata – rata nilai presisi rekomendasi untuk setiap kelas pelamar dan rata – rata nilai presisi rekomendasi secara keseluruhan. Menyeimbangkan jumlah data pelamar tiap kelas pelamar terbukti dapat menyeimbangkan rata – rata nilai presisi tiap kelas pelamar. Namun, penyeimbangan

jumlah data pelamar tidak menjamin peningkatan rata – rata nilai presisi keseluruhan data pelamar. Hasil uji coba Skenario 5 dibandingkan dengan Skenario 4, membuktikan rata – rata nilai presisi keseluruhan data mengalami penurunan yaitu 52,35% dengan menggunakan Set 1, sedangkan rata – rata nilai keseluruhan data mengalami peningkatan yaitu 61,02% dengan menggunakan Set 2.

Lampiran 1

ANTARMUKA SISTEM

Tabel 8.1 Spesifikasi Atribut Antarmuka File Masukan

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan /Keluaran
1	lblName_CandidateFile	<i>Label</i>	Memperlihatkan <i>path file</i> masukan data pelamar	<i>Text</i>
2	btnBrowse_CandidateFile	<i>Button</i>	Membuka <i>file explorer</i> untuk memilih <i>file</i> masukan data pelamar	<i>Event Click</i>
3	lblName_JobFile	<i>Label</i>	Memperlihatkan <i>path file</i> masukan data pekerjaan	<i>Text</i>
4	btnBrowse_JobFile	<i>Button</i>	Membuka <i>file explorer</i> untuk memilih <i>file</i> masukan data pekerjaan	<i>Event Click</i>
5	lblName_RelationFile	<i>Label</i>	Memperlihatkan <i>path file</i> masukan data feedback pelamar	<i>Text</i>
6	btnBrowse_RelationFile	<i>Button</i>	Membuka <i>file explorer</i> untuk memilih <i>file</i> masukan data feedback pelamar	<i>Event Click</i>

Tabel 8.2 Spesifikasi Atribut Halaman Interaksi Pelamar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan /Keluaran
1	btnTargetActor	<i>Button</i>	Mengeluarkan informasi yang berkaitan dengan pelamar yang telah dipilih pada txtTargetActor	<i>Event Click</i>
2	txtTargetActor	<i>Text Box</i>	Masukan id pelamar	<i>Text</i>
3	tbvTargetActor	<i>Table View</i>	Memperlihatkan data seluruh pelamar	<i>Table Data</i>
4	txtTargetActorInfo	<i>Text Box</i>	Informasi deskripsi pelamar yang telah dipilih pada txtTargetActor	<i>Text</i>
5	tbvTargetActorRelation	<i>Table View</i>	Informasi <i>feedback</i> yang pernah diberikan oleh pelamar yang telah dipilih pada txtTargetActor	<i>Table Data</i>
6	btnUpdateRelation	<i>Button</i>	Menyimpan perubahan data yang dilakukan pada	<i>Event Click</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan /Keluaran
			tbvTargetActorRelation	
7	cbxRelation	<i>Combo Box</i>	Pilihan jenis feedback yang akan diberikan pelamar terhadap pekerjaan	<i>Text</i>
8	txtRelationTarget	<i>Text Box</i>	Masukan id pekerjaan yang akan diberikan feedbackoleh pelamar	<i>Text</i>
9	btnRelationSubmit	<i>Button</i>	Menyimpan data feedback pekerjaan oleh pelamar baru	<i>Event Click</i>
10	tbvJob	<i>Table View</i>	Memperlihatkan data seluruh pekerjaan	<i>Table Data</i>
11	txtJobColumn	<i>Text Box</i>	Memperlihatkan keseluruhan teks dari kolom yang dipilih pada tbvJob	<i>Text</i>

**Tabel 8.3 Spesifikasi Atribut Halaman Rekomendasi
Pekerjaan**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan /Keluaran
1	tbvAllRelation	<i>Table View</i>	Memperlihatkan data seluruh <i>feedback</i> pekerjaan oleh pelamar	<i>Table Data</i>
2	btnCheckSimilarity	<i>Button</i>	Menghitung kemiripan antara entitas berdasarkan konteks teks dokumen	<i>Event Click</i>
3	tbvSimilarity	<i>Table View</i>	Memperlihatkan pasangan entitas dengan nilai kemiripan tertinggi	<i>Table Data</i>
4	txtTargetActor2	<i>Text Box</i>	Masukan id pelamar yang akan diberikan rekomendasi pekerjaan	<i>Text</i>
5	txtVarLamda	<i>Text Box</i>	Masukan parameter lamda	<i>Digit</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan /Keluaran
6	txtVarD	<i>Text Box</i>	Masukan parameter damping factor	<i>Digit</i>
7	txtVarPu	<i>Text Box</i>	Masukan parameter personalization factor	<i>Digit</i>
8	txtVarWs	<i>Text Box</i>	Masukan parameter bobot relasi similar	<i>Digit</i>
9	txtVarWa	<i>Text Box</i>	Masukan parameter bobot relasi apply	<i>Digit</i>
10	txtVarWL	<i>Text Box</i>	Masukan parameter bobot relasi like	<i>Digit</i>
11	txtJmlRekomendasi	<i>Text Box</i>	Masukan jumlah rekomendasi pekerjaan yang ingin dihasilkan	<i>Digit</i>

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan /Keluaran
12	btnFindRecommendation	<i>Button</i>	Mencari rekomendasi pekerjaan	<i>Event Click</i>
13	tbvRecommendation	<i>Table View</i>	Memperlihatkan hasil pencarian rekomendasi pekerjaan	<i>Table Data</i>
14	txtRecommendationInfo	<i>Text Box</i>	Memperlihatkan hasil presisi pencarian rekomendasi pekerjaan	<i>Text</i>

Lampiran 2

CONTOH DATA UJI COBA

Tabel 9.1 Contoh Data Uji Coba Pelamar

No	Id	Name	Experience	Education	Degree
1	JC-1	Nerette Wooten	Registrar/Clerk - North Shore LIJ Health System....	Lehman College	Bachelor of Arts
2	JC-2	Jermaine Cameron	Assistant in Nursing - Casual - Metro Rehab Hospital; Assistant in Nursing - Casual - East Sydney Private Hospital;	Southern New Hampshire University	MBA
3	JC-3	Jianwei Lu	Senior Developer - Sandstone Technology....	Macquarie University	Master of Information Technology
4	JC-4	Nikky Peddinti	IT manager - CGI Information Systems & Management Consultancy....	Osmania University	Diploma

No	Id	Name	Experience	Education	Degree
5	JC-5	Maurice Zereoue	Inside Sale Associate - Picture People; Art Administration Assistant...	West Virginia University	Bachelor's

Tabel 9.2 Contoh Data Uji Coba Pekerjaan

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
1	JD-1	DEPARTMENT OF FINANCE	IT Asset Management College Aide	<p>The Finance Information and Technology Division delivers world-class information and technology solution that the agency needs to achieve results. The New York City Department of Finance is seeking two (2) IT Asset Management College Aides that will work under supervision...</p>	<p>For Assignment Level I: Matriculation at an accredited college or graduate school. Employment is conditioned upon continuance as a student in a college or graduate school....</p>	<p>Prefer candidates with: Working knowledge of MS Windows XP...</p>

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
2	JD-2	DEPT OF HEALTH/MENTAL HYGIENE	Mental Health Service Corps Project Manager	The NYC Mental Health Service Corps is a program that was developed as part of ThriveNYC (the Mental Health Roadmap). It is a workforce initiative to hire, train and supervise early career mental health professionals (social workers, psychologists and physicians/psychiatrists)...	A baccalaureate degree from an accredited college and five years of full-time professional satisfactory experience acquired within the last fifteen years, in a health services setting such as a laboratory, hospital, or other patient care facility, or in a public health...	Masters degree in Public Health, Public Administration or other health care related field....

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
3	JD-3	DEPT OF ENVIRONMENT NT PROTECTION	Customer Service Intern	The New York City Department of Environmental Protection (DEP) protects public health ...	For Assignment Level I: Matriculation at an accredited college or graduate school. ...	Excellent communication, interpersonal and organizational skills...

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
4	JD-4	DEPT OF PARKS & RECREATION	Administrative Assistant for the Stewardship Program	The Forestry, Horticulture and Natural Resources Division engages volunteers in hands-on management of NYCs natural landscapes...	Qualification Requirements 1. High school graduation or equivalent and three years of experience in community work...	Education and/or experience which is equivalent to "1" above.

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
5	JD-5	DEPARTMENT OF SANITATION	Oracle Database Administrator or	This position is for a Senior Oracle Database Administrator with experience administering oracle databases in high availability environments...	A master's degree in computer science, full-time, satisfactory experience using information technology ...	10+ years of Oracle RDBMS production experience with 5 years of RAC, Clusterware and OEM Grid

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
6	JD-6	DEPT OF INFO TECH & TELECOMM	Senior Mobile Application Developer	DoITT provides for the sustained, efficient and effective delivery of IT services, infrastructure and telecommunications to enhance service delivery to New York City's residents, businesses, employees and visitors....	A baccalaureate degree from an accredited college, including or supplemented by twenty-four (24) semester credits in computer science or a related computer field...	The successful candidate should possess the following: 8 years of development experience application...

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
7	JD-7	DEPT OF HEALTH/MENTAL HYGIENE	Director of Administration and Operations, Bureau of Mental Health	<p>The Bureau of Mental Health is responsible for mental health service delivery to residents of New York City. The bureau fulfills this responsibility by managing the development, and implementation, and oversight of ongoing and new mental health initiatives to ensure full access and quality care for all New York City residents...</p>	<p>A baccalaureate degree from an accredited college and five years of full-time professional satisfactory experience acquired within the last fifteen years, in a health services setting such as a laboratory, hospital, or other patient care facility...</p>	<p>Knowledge of mental health services, and the service system; - Excellent managerial, quantitative, oral and written communication skills....</p>

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
8	JD-8	DEPT OF HEALTH/MENTAL HYGIENE	Special Assistant to the Commissioner, Bureau of Maternal Infant and Reproductive Health	The Bureau of Maternal, Infant and Reproductive Health works to improve maternal, infant and reproductive health outcomes and promote health equity in New York City....	For Assignment Level I (only physical, biological and environmental sciences and public health) A master's degree from an accredited college or university with a specialization in an appropriate field of physical, biological.....	Strong organizational and project management skills, with good follow-through to complete assignments within specified timeframes, and managing multiple projects

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
9	JD-9	DEPT. OF HOMELESS SERVICES	Clerical Associate	The NYC Department of Homeless Services seeks a Clerical Associate Level 4 to work in the Prevention Assistance and Temporary Housing (PATH) Unit in the Division of Family Services....	Qualification Requirements A four-year high school diploma or its educational equivalent approved by a State's department of education or a recognized accrediting organization ...	This is a critical administrative position requiring strict attention to detail and the ability to handle multiple tasks....

No	Id	Agency	Business Title	Job Description	Minimum Qual Requirements	Preferred Skills
10	JD-10	DEPT OF CITYWIDE ADMIN SVCS	Administrative Procurement Analyst	The New York City Department of Citywide Administrative Services (DCAS), Office of Citywide Procurement (OCP) seeks to hire an Administrative Procurement Analyst	A baccalaureate degree from an accredited college and four years of full-time satisfactory professional experience in purchasing, procurement, contract administration or a related field...	Thorough knowledge of procurement practices, principles, terms and conditions and contract law...

Tabel 9.3 Contoh Data Uji Coba *Feedback*

No	jid	relation	jd
1	JC-1	Like	JD-2
2	JC-1	Like	JD-3
3	JC-1	Like	JD-8
4	JC-1	Apply	JD-4
5	JC-2	Like	JD-7
6	JC-2	Apply	JD-9
7	JC-3	Apply	JD-5
8	JC-4	Like	JD-1
9	JC-4	Apply	JD-6
10	JC-5	Like	JD-5
11	JC-5	Like	JD-10

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan Tugas Akhir ini. Selain itu, juga terdapat beberapa saran terhadap Tugas Akhir ini agar dapat dikembangkan lagi dan bisa membuat Tugas Akhir ini menjadi lebih baik lagi.

6.1. Kesimpulan

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan pada Tugas Akhir antara lain:

1. Sistem rekomendasi pekerjaan berjenis *hybrid* dapat menghasilkan nilai presisi rekomendasi lebih tinggi daripada sistem rekomendasi pekerjaan berjenis *content-based* dan *collaborative*.
2. Ketiga parameter Algoritma 3A memiliki peran tersendiri dalam proses *ranking node* pada graf model hubungan pelamar dan pekerjaan. Parameter λ (*lamda*) berfungsi untuk menaikkan nilai *rank* seluruh *node* sama rata agar tidak ada *node* yang bernilai *rank* 0. Parameter d (*damping factor*) berfungsi untuk menaikkan nilai *rank node - node* yang memiliki *edge* masukan. Parameter Pu (*personalization factor*) berfungsi untuk menaikkan nilai *rank node - node* yang berdekatan dengan *node target* (*node* pelamar yang akan diberikan rekomendasi)
3. Perubahan komposisi nilai parameter Algoritma 3A berpengaruh terhadap nilai presisi rekomendasi sistem rekomendasi. Komposisi nilai parameter dengan nilai presisi rekomendasi tertinggi adalah $Pu > d > \lambda$.
4. Komposisi nilai parameter Algoritma yang seimbang dalam menghasilkan nilai presisi rekomendasi maksimal adalah $Pu = 0.6, d = 0.3, \lambda = 0.1$ dengan hasil rata – rata nilai presisi rekomendasi 52,37% menggunakan sistem rekomendasi berjenis *hybrid*.

5. Jumlah data pelamar pada setiap kelas mempengaruhi rata - rata nilai presisi rekomendasi untuk setiap kelas pelamar. Jumlah data yang seimbang untuk setiap kelas pelamar dapat menghasilkan rata – rata nilai presisi setiap kelas pelamar yang seimbang. Namun, penyeimbangan jumlah data tidak menjamin peningkatan rata – rata nilai presisi keseluruhan data.

6.2. Saran

Terdapat beberapa saran terkait Tugas Akhir ini yang diharapkan bisa membuat Tugas Akhir ini menjadi lebih baik. Saran-saran tersebut antara lain:

1. Tahap pra-pemrosesan teks dapat dikembangkan lagi agar dapat menangkap kemiripan antar dokumen lebih baik dan dapat mengurangi ukuran matriks *term-document* sehingga akan mempercepat proses rekomendasi. Misal dengan menambahkan proses *stemming* kata.
2. Proses perbandingan teks untuk dicari kemiripannya antara kumpulan dokumen teks profil pelamar dan deskripsi pekerjaan dapat dibuat lebih spesifik per bagian dokumen. Misal, teks pada bagian "*experience*" dalam dokumen teks profil pelamar hanya dibandingkan dengan teks pada bagian yang sama dalam dokumen teks profil pelamar lainnya dan bagian "*minimum qualification required*" dalam dokumen teks deskripsi pekerjaan. Hal ini dilakukan agar dapat menangkap kemiripan antar dokumen lebih akurat.
3. Jumlah data *feedback* pelamar terhadap pekerjaan lebih baik seimbang untuk setiap kelas, agar node pekerjaan tiap kelas memiliki masukan *edge* yang seimbang sehingga hasil rekomendasi akan semakin akurat untuk setiap kelasnya.

DAFTAR PUSTAKA

- [1] F. Ricci, L. Rokach and B. Shapira, *Recommender Systems Handbook*, Springer, 2011.
- [2] S. El Helou, D. Gillet and Y. Lu, "A Recommender System for Job Seeking and Recruiting Website".
- [3] "Indeed Resume Search," [Online]. Available: <http://www.indeed.com/resumes>. [Accessed 25 March 2016].
- [4] "NYC Open Data," [Online]. Available: <https://nycopendata.socrata.com/>. [Accessed 23 March 2016].
- [5] S. El Helou, C. Salzmman and D. Gillet, "The 3A Personalized, Contextual and Relation-based Recommender System," *Journal of Universal Computer Science*, vol. 16, no. 16, 2010.
- [6] L. Page and S. Brin, "The PageRank Citation Ranking : Bringing Order to the Web," *Stanford InfoLab*, 1998.
- [7] S. Deerwester, S. T. Dumais and H. Richard, "Indexing by Latent Semantic Analysis".
- [8] A. Thomo, "Latent Semantic Analysis Tutorial," [Online]. Available: www.engr.uvic.ca/~seng474/svd.pdf. [Accessed 11 April 2016].
- [9] C. D. Manning, P. Raghavan and H. Schutze, "Evaluation in Information Retrieval," in *An Introduction to Information Retrieval*, Cambridge, Cambridge University Press, 2009, p. 151.
- [10] A. Gunawardana and G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *Journal of Machine Learning Research* 10, 2009.

- [11] import.io, "Web Data Platform & Free Web Scrapping Tool," [Online]. Available: www.import.io. [Accessed 24 March 2016].
- [12] A. Madry, *Lecture 9.1 : Introduction 2 Random Walks in Graphs*, Ecole Polytechnique Eederale De Lausanne, 2012.

BIODATA PENULIS



Adhi Nurilham lahir di Surabaya, 1 Maret 1994. Penulis mulai tertarik pada dunia pemrograman saat masih menduduki bangku SMA. Oleh karena itu, penulis memutuskan untuk mengenyam pendidikan S1 di Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS), Surabaya. Penulis berharap dengan penulisan ini, dapat memberikan ilmu kepada orang lain.

Penulis dalam menyelesaikan pendidikan S1 mengambil rumpun mata kuliah (RMK) Komputasi Cerdas dan Visi serta memiliki ketertarikan di bidang Basis Data, Pemrograman *Web*, *Data Mining*, serta *Machine Learning*. Penulis dapat dihubungi melalui surel: adhi.nurilham@outlook.com.