



TUGAS AKHIR - SM141501

***KAJIAN GRAPH SOCIAL NETWORK ANALYSIS*
UNTUK IDENTIFIKASI RELASI MAHASISWA ITS
PADA KOMUNITAS KEILMIAHAN BERDASARKAN
DATA USULAN PKM**

**NURMA ARIKA WIDYA YOGA
NRP 1213 100 051**

**Dosen Pembimbing
Alvida Mustika Rukmi, S.Si, M.Si**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



FINAL PROJECT - SM141501

**STUDY OF GRAPH SOCIAL NETWORK ANALYSIS
TO IDENTIFY ITS STUDENTS' RELATIONS IN
SCIENTIFICS COMMUNITY BASED ON DATA
PROPOSED PKM**

**NURMA ARIKA WIDYA YOGA
NRP 1213 100 051**

**Dosen Pembimbing
Alvida Mustika Rukmi, S.Si, M.Si**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

LEMBAR PENGESAHAN

**KAJIAN *GRAPH SOCIAL NETWORK ANALYSIS*
UNTUK IDENTIFIKASI RELASI MAHASISWA ITS
PADA KOMUNITAS KEILMIAHAN BERDASARKAN
DATA USULAN PKM**

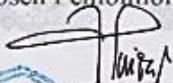
***STUDY OF GRAPH SOCIAL NETWORK ANALYSIS TO
IDENTIFY ITS STUDENTS' RELATIONS IN
SCIENTIFICS COMMUNITY BASED ON DATA
PROPOSED PKM***

Diajukan Untuk Memenuhi Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Sains
Pada Bidang Studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

NURMA ARIKA WIDYA YOGA
NRP. 1213 100 051

Menyetujui,
Dosen Pembimbing,



Alvida Mustika Rukmi, S.Si, M.Si
NIP. 19720745 199802 2 001

Mengetahui,
Kepala Departemen Matematika



Dr. Ahmad Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Surabaya, Juli 2017

**KAJIAN *GRAPH SOCIAL NETWORK ANALYSIS* UNTUK
IDENTIFIKASI RELASI MAHASISWA ITS PADA
KOMUNITAS KEILMIAHAN BERDASARKAN
DATA USULAN PKM**

Nama Mahasiswa : Nurma Arika Widya Yoga
NRP : 1213 100 051
Departemen : Matematika FMIPA-ITS
Dosen Pembimbing : Alvida Mustika Rukmi, S.Si, M.Si

Abstrak

Identifikasi relasi mahasiswa Institut Teknologi Sepuluh Nopember (ITS) berdasarkan usulan Program Kreativitas Mahasiswa (PKM) merupakan salah satu upaya membangun relasi bidang keilmiahan yang efektif untuk meningkatkan produktifitas kinerja ilmiah khususnya PKM di ITS. Pada pembentukan komunitas keilmiahan, untuk dapat mengidentifikasi pertukaran informasi dan opini dari sekian banyak usulan PKM merupakan hal yang relatif sulit. Pada penelitian ini, penggunaan *Social Network Analysis* (SNA) adalah untuk menganalisis struktur dan pola relasi interaksi antar individu yang kemudian direpresentasikan dalam bentuk *graph*. Interaksi dalam SNA tersebut menjawab berbagai persoalan antara lain mengukur bagaimana mahasiswa terkoneksi dengan yang lain, pengaruh seorang mahasiswa dengan mahasiswa lainnya dalam komunitas keilmiahan dan mengukur mahasiswa saling terhubung dan berinteraksi. Proses yang dilakukan meliputi ekstraksi kata kunci, pembentukan vektor konsep, identifikasi topik, identifikasi relasi menggunakan SNA dan pengelompokan mahasiswa ITS yang divisualisasikan dalam bentuk *graph* dalam menyajikan hubungan mahasiswa ITS berdasarkan usulan PKM.

Kata kunci: Identifikasi Relasi Mahasiswa, Clustering, Social Network Analysis

***STUDY OF GRAPH SOCIAL NETWORK ANALYSIS TO
IDENTIFY ITS STUDENTS' RELATIONS IN SCIENTIFICS
COMMUNITY BASED ON DATA PROPOSED PKM***

Student Name : Nurma Arika Widya Yoga
NRP : 1213 100 051
Department : Mathematics
Supervisor : Alvida Mustika Rukmi, S.Si, M.Si

Abstract

Identification relations students of the Sepuluh Nopember Institute of Technology (ITS) based on data proposed Student Creativity Program (PKM) is one way to build effective relationships Scientifics fields to improve the productivity of scientific performance especially PKM in ITS. On the establishment Scientifics community, to be able to identify the exchange of information and opinions from so many data proposed PKM is relatively difficult. In this study, the use of Social Network Analysis (SNA) is to analyze the structure and relationship patterns of interaction between individuals who then represented in the form of a graph. Interaction in the SNA answer the various issues, among others, measure how students connect with each other, how the influence of a particular student in Scientifics community and measure how students connect and interact. The process includes keyword extraction, concept vector formation, topic identification, identification using SNA and grouping of ITS students visualized in graph form in presenting ITS student relationship based on PKM proposal.

Keyword: Identification of Student Relations, Clustering, Social Network Analysis

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “Kajian *Graph Social Network Analysis* untuk Identifikasi Relasi Mahasiswa ITS pada Komunitas Keilmiahan berdasarkan Data Usulan PKM”.

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Alvida Mustika Rukmi S.Si, M.Si selaku Dosen Pembimbing yang telah memberikan bimbingan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
2. Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS.
3. Dra. Nuri Wahyuningsih, M.Kes selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Departemen Matematika ITS.
4. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Departemen Matematika ITS.
5. Drs. Iis Herisman, M.Si selaku Sekretaris Program Studi S1 Departemen Matematika ITS.
6. Seluruh jajaran dosen dan staf Jurusan Matematika ITS.
7. Orang tua penulis dan kedua kakak pertama perempuan penulis yang tiada henti-hentinya mencurahkan kasih sayang, perhatian, semangat dan do'a kepada penulis selama ini.
8. Teman-teman Grup S.Si 15/16 dan Wanita sholikhah yang senantiasa memberikan motivasi dan hiburan pada penulis.
9. Rekan-rekan Matematika ITS terutama angkatan 2013 yang telah menjadi teman penulis hingga sekarang.
10. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Sebagai manusia biasa, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan dan memiliki banyak kekurangan. Sehingga dengan segala kerendahan hati, penulis mengharapkan saran dan kritik yang membangun dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juni 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak	vii
Abstract	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan Tugas Akhir	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Metode <i>Rapid Automatic Keyphrase Extraction</i> (RAKE).....	7
2.2 Metode <i>Latent Semantic Analysis</i> (LSA).....	8
2.2.1 <i>Term Weighting</i> dengan <i>Term Frequency</i> (tf) dan <i>Inverse Document Frequency</i> (idf).....	9
2.2.2 Metode <i>Singular Value Decomposition</i> (SVD)	10
2.3 <i>Social Network Analysis</i>	11
2.4 Pengklasteran dengan <i>k-means</i>	14
2.5 Metode <i>Silhouette Coefficient</i>	14
BAB III METODE PENELITIAN	17
3.1 Pengumpulan Data	17
3.2 Studi literatur	17
3.3 Perancangan perangkat lunak	18
3.4 Pembuatan perangkat lunak	18
3.5 Uji coba dan evaluasi	19
3.6 Penarikan Kesimpulan dan Penyusunan Laporan Tugas Akhir	20
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM	23
4.1 Analisis Sistem	23

4.1.1	Deskripsi Umum Perangkat Lunak	23
4.1.2	Lingkungan Perancangan Perangkat Lunak.....	25
4.2	Perancangan Data.....	25
4.2.1	Data Masukan	26
4.2.2	Data Keluaran	27
4.3	Perancangan Proses.....	27
4.3.1	Ekstraksi konsep dokumen	27
4.3.1.1	Ekstraksi kata kunci	28
4.3.1.2	Pembentukan vektor konsep	30
4.3.2	Identifikasi topik	31
4.3.2.1	Pengelompokan Konsep	32
4.3.2.2	Pembentukan vektor topik dokumen	34
4.3.3	Identifikasi relasi mahasiswa	35
4.3.4	Pengelompokan mahasiswa	38
4.3.4.1	Pembentukan vektor mahasiswa.....	38
4.3.4.2	Pengelompokan mahasiswa	40
4.3.5	Representasi <i>graph</i> mahasiswa	42
4.4	Implementasi sistem.....	43
4.4.1	Implementasi Antarmuka Aplikasi	44
4.4.1.1	Tab <i>Concept Extraction</i>	44
4.4.1.2	Tab <i>Topic Detection</i>	45
4.4.1.3	Tab <i>Clustering & SNA</i>	45
4.4.1.4	Tab <i>Cluster Result</i>	46
4.4.2	Implementasi Program	47
4.4.2.1	Implementasi Ekstraksi Konsep.....	47
4.4.2.2	Implementasi Proses Pengelompokan Konsep ..	53
4.4.2.3	Implementasi Identifikasi Relasi Mahasiswa.....	58
BAB V HASIL DAN PEMBAHASAN		61
5.1	Data Uji Coba.....	61
5.2	Ekstraksi Konsep Dokumen.....	61
5.2.1	Analisa Ekstrasi Kata Kunci dengan RAKE.....	62
5.2.2	Pembentukan Vektor Konsep Dokumen dengan LSA	
	67	
5.2.2.1	Perhitungan bobot tiap <i>term</i> dengan Tf-Idf	67
5.2.2.2	Dekomposisi matriks SVD	69

5.3	Identifikasi Topik.....	69
5.3.1	Analisa Pengelompokan Konsep dengan <i>k-means</i> ..	69
5.3.2	Pembentukan vektor topik dokumen	73
5.4	Identifikasi Relasi Mahasiswa	75
5.4.1	<i>Normalize Degree Centrality</i>	76
5.4.1.1	Analisa Hasil <i>Degree Centrality</i>	78
5.4.2	<i>Normalize Closeness centrality</i>	80
5.4.2.1	Analisa Hasil <i>Normalize Closeness centrality</i> ...	81
5.4.3	<i>Normalize Betweenness Centrality</i>	82
5.4.3.1	Analisa Hasil <i>Betweenness centrality</i>	84
5.5	Pengelompokan Mahasiswa.....	87
5.5.1	Analisa Pengelompokan Mahasiswa	88
5.5.1.1	Analisa Fitur SNA	88
5.5.1.2	Analisa Fitur Topik.....	89
5.5.1.3	Analisa Gabungan Fitur Topik dan SNA.....	90
5.6	Representasi <i>graph</i>	92
BAB VI PENUTUP		95
5.1	Kesimpulan	95
5.2	Saran	96
DAFTAR PUSTAKA.....		97
LAMPIRAN A		99
LAMPIRAN B		101
BIODATA PENULIS.....		111

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi algoritma tf-idf.....	9
Gambar 2.2 Visualisasi SNA, tersusun atas ‘node’ dan ‘vertex’	12
Gambar 4.1 Deskripsi Umum Perangkat Lunak.....	24
Gambar 4.2 Tabel <i>Database</i>	26
Gambar 4.3 Ekstraksi kata kunci menggunakan RAKE.....	29
Gambar 4.4 Proses pembentukan vektor konsep.....	31
Gambar 4.5 Proses identifikasi topik	33
Gambar 4.6 Proses pembentukan vektor topik dokumen	35
Gambar 4.7 Proses pembentukan awal relasi mahasiswa	37
Gambar 4.8 Proses pembentukan vektor mahasiswa	39
Gambar 4.9 Proses pengelompokan mahasiswa.....	41
Gambar 4.10 Proses pembentukan <i>graph</i> mahasiswa	43
Gambar 4.11 Tab <i>Concept Extraction</i>	44
Gambar 4.12 Tab <i>Topic Detection</i>	45
Gambar 4.13 Tab <i>Clustering & SNA</i>	46
Gambar 4.14 Tab <i>Cluster Result</i>	47
Gambar 5.1 <i>Preprocessing Data</i>	62
Gambar 5.2 Ekstraksi Konsep	63
Gambar 5.3 Pengelompokan Konsep	70
Gambar 5.4 Proses Identifikasi relasi mahasiswa	76
Gambar 5.5 <i>Visualisasi graph SNA</i> menggunakan 3 data untuk perhitungan <i>degree centrality</i>	79
Gambar 5.6 Visualisasi <i>graph</i> fitur SNA	89
Gambar 5.7 Visualisasi <i>graph</i> fitur topik.....	90
Gambar 5.8 Visualisasi <i>graph</i> awal fitur gabungan	91
Gambar 5.9 Pembentukan <i>graph</i> awal sebelum <i>clustering</i>	92
Gambar 5.10 Visualisasi Graph Relasi Awal	92
Gambar 5.11 Pembentukan relasi setelah dilakukan <i>clustering</i> ..	93
Gambar 5.12 Visualisasi Graph Relasi dengan $k=22$	93

DAFTAR TABEL

Tabel 4.1 Tombol dan kegunaan pada tab <i>concept extraction</i>	44
Tabel 4.2 Tombol dan kegunaan pada tab <i>topic detection</i>	45
Tabel 4.3 Tombol dan kegunaan pada tab <i>clustering & SNA</i>	46
Tabel 5.1 Contoh Data Mentah	63
Tabel 5.2 Contoh Hasil <i>Preprocessing</i>	64
Tabel 5.3 Contoh matriks <i>co-occurrence</i> pada dokumen 1	65
Tabel 5.4 Contoh perhitungan rasio pada dokumen 1	66
Tabel 5.5 Contoh perhitungan nilai fitur dasar pada dokumen 1	66
Tabel 5.6 Contoh kandidat frase kunci dengan $k = 2$	66
Tabel 5.7 Contoh kandidat frase kunci dengan $k = 2$	67
Tabel 5.8 Hasil Matriks <i>Term Frequency</i>	67
Tabel 5.9 Hasil Matriks Pembobotan	69
Tabel 5.10 Hasil Pengelompokan Konsep dengan <i>k-means</i>	70
Tabel 5.11 Contoh Matriks Pembobotan <i>term</i>	71
Tabel 5.12 Contoh 3 centroid terpilih secara acak	71
Tabel 5.13 Contoh 3 centroid terpilih secara acak	72
Tabel 5.14 <i>Centroid</i> baru pada <i>cluster</i> 1	73
Tabel 5.15 <i>Centroid</i> baru pada ketiga <i>cluster</i>	73
Tabel 5.16 <i>Document Cluster Vector</i>	74
Tabel 5.17 Kandidat Kata Kunci Dokumen 1	74
Tabel 5.18 Hasil 3 topik <i>cluster</i> konsep	74
Tabel 5.19 Hasil <i>Document Cluster Vector</i> pada dokumen 1	75
Tabel 5.20 Hasil <i>Document Cluster Vector</i> pada ketiga dokumen	75
Tabel 5.21 Hasil perhitungan <i>Normalize Degree Centrality</i>	77
Tabel 5.22 Contoh 3 Data Uji Usulan	78
Tabel 5.23 Hasil <i>normalize degree centrality</i> dengan 3 data	80
Tabel 5.24 Hasil <i>normalize closeness centrality</i> dengan 3 data ..	82
Tabel 5.25 Hasil perhitungan <i>Normalize Betweenness Centrality</i>	83
Tabel 5.26 Hubungan antar node dalam network.....	84
Tabel 5.27 Hasil <i>betweenness centrality</i> dengan 3 data	86
Tabel 5.28 Hasil Ujicoba Pengelompokan Mahasiswa dengan $k=22$	87

Tabel 5.29 Hasil Ujicoba Pengelompokan Mahasiswa dengan $k=3$	88
Tabel 5.30 Hasil Ujicoba Pengelompokan Mahasiswa fitur SNA	89
Tabel 5.31 Hasil Ujicoba Pengelompokan Mahasiswa fitur Topik	90
Tabel 5.32 Hasil Ujicoba Pengelompokan Mahasiswa fitur Topik	91

BAB I

PENDAHULUAN

Pada bagian ini akan dijelaskan mengenai latar belakang, permasalahan, batasan masalah, tujuan, manfaat yang ingin dicapai, dan sistematika penulisan yang digunakan dalam pembuatan tugas akhir ini.

1.1 Latar Belakang

Publikasi Karya Ilmiah merupakan suatu wadah bagi para dosen dan mahasiswa untuk menyampaikan gagasan ilmiah mengenai hasil penelitian dan kajian akademik berupa artikel skripsi, tesis atau disertasi, maupun hasil kajian yang dipresentasikan dalam kegiatan seminar [1]. Saat ini publikasi hasil penelitian Indonesia di dunia internasional masih terbelakang, terutama publikasi di media yang terindeks di pengindeks internasional bereputasi. Salah satu penyebabnya adalah budaya menulis yang belum berkembang di masyarakat pada umumnya, perguruan tinggi khususnya, dan/atau rendahnya kemauan dan kemampuan menulis hasil-hasil penelitian maupun pengabdian kepada masyarakat dalam terbitan berkala ilmiah bermutu [1].

Untuk mengatasi adanya kesenjangan antara teori yang telah diperoleh mahasiswa dengan realita kebutuhan masyarakat tersebut, adanya Program Kreativitas Mahasiswa (PKM) menjadi salah satu alternatif untuk meningkatkan iklim akademik yang kreatif, inovatif, visioner, solutif, dan mandiri [2]. Dalam hal ini identifikasi relasi mahasiswa Institut Teknologi Sepuluh Nopember (ITS) berdasarkan usulan PKM merupakan salah satu upaya membangun relasi bidang keilmiah yang efektif untuk meningkatkan produktifitas kinerja ilmiah khususnya PKM di ITS.

Terdapat dua jenis data yang dapat digunakan untuk melakukan pengelompokan mahasiswa ITS. Pertama adalah menggunakan atribut mahasiswa ITS yang dapat diperoleh melalui data usulan PKM. Kedua adalah melalui data yang diperoleh dari karakteristik mahasiswa ITS pada jejaring yang telah terbentuk

dari relasi antar mahasiswa ITS. Algoritma pengelompokan bisa menghilangkan informasi yang penting dalam jejaring jika hanya menggunakan salah satu dari dua jenis sumber data. Sebagai contoh, pengelompokan berdasarkan data atribut mahasiswa ITS memungkinkan untuk mengetahui mahasiswa ITS cenderung masuk ke dalam suatu komunitas yang memiliki atribut sama, sedangkan jika menggunakan data karakteristik mahasiswa ITS saja maka mahasiswa ITS mungkin berada dalam satu *cluster* walaupun tidak memiliki kesamaan atribut.

Pada pembentukan komunitas keilmiah, untuk dapat mengidentifikasi pertukaran informasi dan opini dari sekian banyak usulan PKM merupakan hal yang relatif sulit. Penggunaan *Social Network Analysis* (SNA) adalah untuk menganalisis struktur dan pola relasi interaksi antar individu yang kemudian direpresentasikan dalam bentuk *graph*. Pola interaksi merupakan suatu aspek yang penting yang terlibat didalamnya. Interaksi dalam SNA tersebut akan menjawab berbagai persoalan antara lain mengukur bagaimana individu terkoneksi dengan yang lain, bagaimana seseorang akan mempengaruhi relasi antar orang lain dan mengukur bagaimana individu-individu dalam satu grup saling terhubung dan berinteraksi.

Beberapa penelitian menggunakan metode SNA dengan mencermati keterhubungan dan aktivitas pada objek penelitian seperti analisis jaringan sosial komunitas video bloggers dalam YouTube [3]. Kemudian penelitian mengenai pola permainan suatu pertandingan sepakbola untuk mengevaluasi kekalahan ataupun kecurangan yang mungkin terjadi dapat dimonitor dengan metode ini [4]. Selanjutnya penerapan SNA untuk mengidentifikasi terorisme melalui komunitas tertentu [5]. Penelitian lain memanfaatkan teori keterhubungan melalui graf dalam kampanye politik [6], monitoring kemacetan lalu lintas [7] dan sebagainya.

Berdasarkan latar belakang tersebut, pada Tugas Akhir ini dilakukan kajian terhadap struktur *graph* jejaring sosial menggunakan SNA pada pengelompokan mahasiswa ITS

berdasarkan data usulan PKM untuk mengetahui relasi setiap mahasiswa ITS dengan mahasiswa ITS lainnya berkenaan dengan kemiripan usulan PKM dan disiplin ilmu berdasarkan identitas komunitas keilmiah.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat dirumuskan permasalahan dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana mengidentifikasi topik dokumen berdasarkan data atribut mahasiswa ITS yang dapat diperoleh melalui data usulan PKM.
2. Bagaimana mengidentifikasi relasi mahasiswa ITS menggunakan SNA yang mencakup informasi bagaimana mahasiswa terkoneksi dengan yang lain, pengaruh seorang mahasiswa dengan mahasiswa lainnya dalam komunitas keilmiah dan mengukur mahasiswa saling terhubung dan berinteraksi.
3. Bagaimana membuat pengelompokan mahasiswa ITS berdasarkan topik dan relasi mahasiswa menggunakan SNA serta membangun representasi *graph* mahasiswa ITS berdasarkan data usulan PKM.

1.3 Batasan Masalah

Dalam penelitian tugas akhir ini, permasalahan yang akan dibahas dibatasi ruang lingkup pembahasannya antara lain:

1. Data sekunder yang digunakan adalah database usulan proposal PKM ITS yang memiliki atribut:
 - a. NRP Mahasiswa
 - b. Nama Mahasiswa
 - c. Judul Usulan PKM
2. Metode pengelompokan menggunakan *k-means*.
3. Aplikasi ini dibuat menggunakan bahasa pemrograman Java dan *software* basisdata MySQL

1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penelitian Tugas Akhir ini adalah :

1. Mengidentifikasi topik dokumen berdasarkan data atribut mahasiswa ITS yang dapat diperoleh melalui data usulan PKM.
2. Mengidentifikasi relasi mahasiswa ITS menggunakan SNA yang mencakup informasi bagaimana mahasiswa terkoneksi dengan yang lain, pengaruh seorang mahasiswa dengan mahasiswa lainnya dalam komunitas keilmiahan dan mengukur mahasiswa saling terhubung dan berinteraksi.
3. Membuat pengelompokan mahasiswa ITS berdasarkan topik dan relasi mahasiswa menggunakan SNA serta membangun representasi *graph* berdasarkan data usulan PKM.

1.5 Manfaat

Manfaat yang dapat diperoleh dari tugas akhir yang diusulkan ini antara lain:

1. Memberikan informasi bagi pihak yang ingin mengembangkan penelitian mengenai kajian *graph* jejaring sosial menggunakan SNA
2. Sebagai bahan pertimbangan dalam pembentukan komunitas keilmiahan di ITS berdasarkan identifikasi relasi antara mahasiswa ITS dan kemiripan topik usulan PKM
3. Sebagai upaya membangun relasi mahasiswa ITS di bidang keilmiahan yang efektif untuk meningkatkan produktifitas kinerja ilmiah khususnya PKM di ITS.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan pada tugas akhir ini disusun sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang pembuatan tugas akhir, rumusan dan batasan masalah yang dihadapi dalam

penelitian tugas akhir, tujuan dan manfaat pembuatan tugas akhir, dan sistematika penulisan Tugas Akhir.

2. **BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan tentang beberapa teori yang digunakan untuk membantu penyelesaian permasalahan tugas akhir.

3. **BAB III METODOLOGI**

Bab ini menjelaskan metode penelitian yang dilakukan dalam menyelesaikan Tugas Akhir.

4. **BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM**

Bab ini membahas analisa dan perancangan sistem perangkat lunak yang akan dibangun untuk identifikasi relasi mahasiswa ITS menggunakan SNA pada pengelompokan mahasiswa ITS berdasarkan data usulan PKM

5. **BAB V HASIL DAN PEMBAHASAN**

Bab ini menjelaskan mengenai hasil pengujian yang telah dilakukan pada perangkat lunak. Hasil pengujian kemudian dibahas untuk mengetahui kerja sistem secara keseluruhan.

6. **BAB VI PENUTUP**

Bab ini merupakan penutup, berisi tentang kesimpulan yang dapat diambil berdasarkan data yang ada dan saran yang selayaknya dilakukan bila tugas akhir ini dilanjutkan.

BAB II TINJAUAN PUSTAKA

2.1 Metode *Rapid Automatic Keyphrase Extraction* (RAKE)

Metode *Rapid Automatic Keyphrase Extraction* adalah metode yang menggunakan pendekatan berbasis dokumen individu yang mampu mengelompokkan topik usulan PKM tanpa bergantung pada koleksi dokumen. Metode RAKE memperhatikan asosiasi kata dengan menghitung matriks kemunculan bersama satu dengan yang lain. Matriks tersebut digunakan untuk mengukur skor kandidat kata kunci untuk kemudian dilakukan perangkingan.[8] Metode RAKE memiliki tahapan sebagai berikut [9]:

1. Ekstraksi kandidat.

Ekstraksi kandidat kata kunci dimulai dengan memisahkan teks menggunakan *stopword* dan tanda baca. Misalkan terdapat d_i adalah dokumen ke- i dari data usulan yang ada. Setelah dokumen diekstraksi, maka didapatkan kandidat kata kunci $Td_i = \{t_1, t_2, t_3, \dots, t_n\}$, dengan $t_1, t_2, t_3, \dots, t_n$ dalam bentuk kata, maupun frase kandidat kata kunci.

2. Menghitung matriks *co-occurrence*.

Setelah kandidat kata kunci didapatkan, langkah selanjutnya adalah menghitung matriks *co-occurrence*. Matriks *co-occurrence* memetakan frekuensi kemunculan suatu kata dan frase kata kunci.

3. Menghitung rasio.

Nilai rasio merupakan perbandingan antara derajat kata dengan frekuensi kata. Derajat kata adalah jumlah kemunculan kata pada dokumen ditambah jumlah frase yang mengandung kata tersebut. Derajat kata pada matriks *co-occurrence* didapat dari penjumlahan satu kolom atau satu baris. Frekuensi kata adalah jumlah kemunculan kata dalam teks. Nilai frekuensi bisa didapatkan pada nilai diagonal pada matriks *co-occurrence*. Rasio dapat dihitung dengan menggunakan Persamaan 2.1 berikut:

$$\text{rasio} = \frac{\text{deg}(w)}{\text{freq}(w)} \quad (2.1)$$

dimana : $deg(w)$ = derajat kata
 $freq(w)$ = frekuensi kata.

4. Menghitung nilai fitur dasar.

Nilai fitur dasar merupakan nilai penjumlahan rasio kata yang ada pada kandidat kata kunci. Nilai tersebut kemudian diurutkan berdasarkan nilai tertinggi sampai terendah. Nilai fitur dasar dihitung dengan menggunakan Persamaan 2.2 berikut

$$Nfd = rasio(t) + rasio(ft) \quad (2.2)$$

dimana : Nfd = nilai fitur dasar,
 $rasio(t)$ = rasio kata kunci t
 $rasio(ft)$ = rasio frase yang mengandung kata kunci t.

5. Pemilihan kandidat frase kunci dengan skor tertinggi.

Setelah kandidat kata kunci diberi nilai, selanjutnya dipilih sejumlah k

2.2 Metode *Latent Semantic Analysis* (LSA)

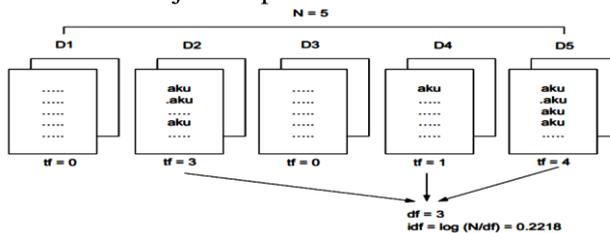
LSA adalah suatu metode untuk menemukan hubungan, keterkaitan, dan kemiripan antar dokumen-dokumen, penggalan dari dokumen-dokumen, dan kata-kata yang muncul pada dokumen-dokumen dengan memanfaatkan komputasi statistik untuk menggali dan merepresentasikan konteks yang digunakan sebagai sebuah arti kata untuk sejumlah *corpus* yang besar. *Corpus* adalah kumpulan teks yang memiliki kesamaan subjek atau tema.

Metode LSA menerima masukan (input) berupa dokumen teks yang selanjutnya akan dibandingkan kata-kata unik yang digunakan atau yang ada pada dokumen kemudian direpresentasikan sebagai matriks X yang merupakan matriks dengan elemen (i, j) yang menunjukkan kemunculan term i pada dokumen j. Cara kerja LSA ialah dengan menghasilkan sebuah model yang didapat dengan mencatat kemunculan-kemunculan kata dari tiap-tiap dokumen yang direpresentasikan dalam sebuah matriks yang dinamakan *term-document matrix*, setelah itu dilakukan proses *Singular Value Decomposition* (SVD) yang akan

digunakan untuk mendapatkan nilai kemiripan antara satu dokumen dengan dokumen yang lain [10].

2.2.1 *Term Weighting dengan Term Frequency (tf) dan Inverse Document Frequency (idf)*

Pembobotan *Term* (*Term Weighting*) bertujuan untuk menentukan bobot setiap *term*. Perhitungan bobot *term* memerlukan dua hal yaitu *Term Frequency* (tf) dan *Inverse Document Frequency* (idf). *Term Frequency* (tf) merupakan frekuensi kemunculan suatu kata (*term*) dalam suatu dokumen. Nilai tf bervariasi di tiap dokumen bergantung pada kemunculan kata di suatu dokumen. Besar nilai tf sebanding dengan tingkat kemunculan *term* di dokumen. Semakin sering *term* muncul pada suatu dokumen, semakin besar pula nilai tf pada dokumen tersebut dan semakin jarang *term* muncul semakin kecil pula nilai tf. Selain *Term Frequency* diperlukan pula *Inverse Document Frequency* (idf) pada pembobotan *term*. *Inverse Document Frequency* (idf) merupakan frekuensi kemunculan *term* pada keseluruhan dokumen. Nilai idf berkaitan dengan distribusi *term* di berbagai dokumen. Nilai idf berbanding terbalik dengan jumlah dokumen yang mengandung. *Term* yang jarang muncul pada keseluruhan dokumen memiliki nilai idf lebih besar dibanding dengan *term* yang sering muncul. Jika setiap dokumen dalam koleksi mengandung *term* yang bersangkutan, maka nilai idf dari *term* tersebut adalah nol (0). Hal ini menunjukkan bahwa setiap *term* yang muncul pada dokumen dalam koleksi tidak berguna untuk membedakan dokumen berdasarkan topik tertentu. Ilustrasi algoritma tf-idf ditunjukkan pada Gambar 2.1 :



Gambar 2.1 Ilustrasi algoritma tf-idf

Persamaan untuk menghitung nilai tf-idf adalah :

$$W_{i,j} = tf_{i,j} \times idf_j = tf_{i,j} \times \log\left(\frac{N}{df_j}\right) \quad (2.3)$$

dimana

$$idf = \log\frac{N}{df} \quad (2.4)$$

Keterangan :

D_1, \dots, D_5 : dokumen

$W_{i,j}$ = bobot *term* ke-i terhadap dokumen ke-j

$tf_{i,j}$ = jumlah kemunculan *term* i ke dalam dokumen j

N = jumlah dokumen secara keseluruhan

df_j = jumlah dokumen yang mengandung *term* i

idf_j = *Inverse document frequency* yang mengandung *term* i

Perhitungan bobot dari *term* tertentu dalam sebuah dokumen dengan menggunakan $tf \times idf$ menunjukkan bahwa deskripsi terbaik dari dokumen adalah *term* yang banyak muncul dalam dokumen tersebut dan sangat sedikit muncul pada dokumen lain[11].

2.2.2 Metode *Singular Value Decomposition* (SVD)

SVD adalah sebuah metode untuk mengidentifikasi dan mengurutkan dimensi yang menunjukkan data mana yang mempunyai variasi paling banyak. Berkaitan dengan hal itu, SVD dapat mengidentifikasi di mana variasi muncul paling banyak, sehingga hal ini memungkinkan untuk mencari pendekatan yang terbaik pada data asli menggunakan dimensi yang lebih kecil. Oleh karena itu, SVD dapat dilihat sebagai metode pengurangan data.

Hal yang mendasari SVD adalah SVD mengambil data asli biasanya terdiri dari variasi matriks kata dan dokumen kemudian memecahnya menjadi komponen – komponen ini dalam beberapa kasus mengandung beberapa korelasi yang beragam dan banyak terjadi data *noise*, yaitu data yang tidak memiliki korelasi yang kuat dengan data asli. Karena komponen utama dari data tersebut biasanya sangat kecil, maka data *noise* tadi bisa diabaikan dengan cara mengurangi dimensi dari data asli. Proses pereduksian dengan

SVD ini akan semakin menegaskan kemiripan data yang mirip dan menegaskan ketidakmiripan data yang tidak mirip.

SVD akan menguraikan sebuah matriks menjadi tiga buah matriks baru yaitu matriks vektor singular kiri, matriks nilai singular dan vektor singular kanan. SVD dari sebuah matriks A yang berdimensi $m \times n$ adalah sebagai berikut [14]:

$$A_{m \times d} = U_{m \times n} S_{n \times n} V_{n \times d}^T \quad (2.5)$$

Keterangan :

A = Matriks berdimensi $m \times n$

U = matriks vektor singular kiri berdimensi $m \times n$

S = matriks nilai singular berdimensi $n \times n$

V = matriks vektor singular kanan berdimensi $n \times d$

V^T = matriks V transpose

2.3 Social Network Analysis

Social Network Analysis atau disebut juga analisa jejaring sosial merupakan suatu teknik untuk memetakan dan mengukur relasi dan komunikasi yang terjadi antar manusia, kelompok, organisasi, komputer ataupun entitas yang memproses suatu informasi [12]. Hubungan relasi ini divisualisasikan dengan *graph* SNA (selanjutnya disebut *sociogram*) yang tervisualisasi sehingga menjadi lebih mudah untuk dianalisis. Hubungan relasi ini divisualisasikan dengan *graph* seperti yang terlihat pada gambar 2.2. Titik pada gambar yang disebut 'node' atau simpul merepresentasikan personal atau individu yang dihubungkan oleh garis yang membentuk 'vertex'. Dua node yang terhubung dinyatakan dengan adanya garis yang menghubungkan keduanya.

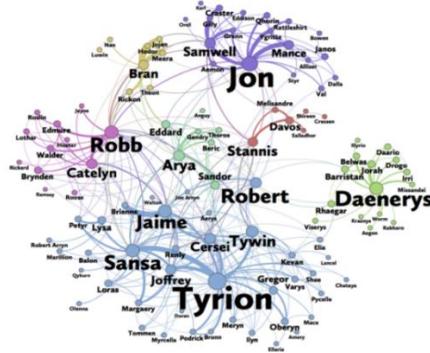
Penggambaran sebuah hubungan dalam *graph* yang disimbolkan dengan menggunakan *edges* atau garis terdapat dua cara yaitu:

1. Undirected Graph

Graph sederhana yang menghubungkan seseorang individu yang memiliki hubungan *edges* atau garis yang digunakan tanpa menggunakan anak panah.

2. *Directed Graph*

Graph yang mampu menunjukkan relasi lebih jelas, karena relasi yang disimbolkan dengan *edges* atau garis digambarkan dengan anak panah.



Gambar 2.2 Visualisasi SNA, tersusun atas ‘node’ dan ‘vertex’

Node dan *vertex* memiliki arti pengukuran tersendiri yang dijelaskan oleh beberapa terminologi dalam teori *graph* berikut ini [13]:

- a) *Degree centrality*, yaitu jumlah koneksi yang dimiliki sebuah node. Individu dengan relasi terbanyak adalah individu yang paling penting. Dapat dihitung menggunakan Persamaan 2.6 berikut:

$$C'_D(n_i) = \frac{d(n_i)}{g-1} \quad (2.6)$$

dimana,

$C'_D(n_i)$ = *normalized degree centrality* individu (*node*) *i*

$d(n_i)$ = *degree* dari individu (*node*) *i*

g = jumlah individu

- b) *Closeness centrality*, merupakan kedekatan suatu individu dengan individu yang lainnya. individu dikatakan penting apabila dekat dengan individu yang lain. *Closeness* didasarkan pada kebalikan jarak dari tiap individu ke individu yang lain

pada jaringan. Dapat dihitung menggunakan Persamaan 2.7 berikut:

$$C_c(n_i) = \left[\sum_{j=1}^g d(n_i, n_j) \right]^{-1} \quad (2.7)$$

dimana,

$C_c(n_i)$ = *closeness centrality* individu (*node*) i

$d(n_i, n_j)$ = jarak individu i ke j

Rumus tersebut dapat dinormalisasikan menjadi Persamaan 2.8 berikut:

$$C'_c(n_i) = (C_c(n_i))(g - 1) \quad (2.8)$$

dimana,

$C'_c(n_i)$ = *normalized centrality closeness* individu (*node*) i

g = jumlah individu .

- c) *Betweenness centrality*, menghitung jumlah jalan terpendek antara individu j dan k dimana individu i berada. Dapat dihitung menggunakan Persamaan 2.9 berikut:

$$C_B(n_i) = \sum_{j < k} \frac{g_{jk}(n_i)}{g_{jk}} \quad (2.9)$$

dimana,

$C_B(n_i)$ = *betweenness centrality* individu (*node*) i

$g_{jk}(n_i)$ = jumlah *path* dimana individu i berada

g_{jk} = jumlah *path* yang menghubungkan individu j dan k

Rumus tersebut dapat dinormalisasikan menjadi Persamaan 2.10 berikut:

$$C'_B(n_i) = \frac{C_B(n_i)}{\left[\frac{(g-1)(g-2)}{2} \right]} \quad (2.10)$$

dimana,

$C'_B(n_i)$ = *normalized centrality betweenness* individu (*node*) i

g = jumlah individu

2.4 Pengklasteran dengan *k-means*

Jika diberikan sekumpulan data $X = \{x_1, x_2, \dots, x_n\}$ dimana $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ adalah vektor n dalam ruang real R , maka algoritma *k-means* akan mempartisi X dalam k buah *cluster*. Setiap *cluster* memiliki *centroid* (titik tengah) atau *mean* dari data-data dalam *cluster* tersebut. Pada tahap awal, algoritma *k-means* memilih secara acak k buah data sebagai *centroid*. Kemudian, jarak antara data dan *centroid* dihitung menggunakan *Euclidian distance*. Data ditempatkan dalam *cluster* yang terdekat, dihitung dari titik tengah *cluster*. *Centroid* baru akan ditentukan bila semua data telah ditempatkan dalam *cluster* terdekat. Proses penentuan *centroid* dan penempatan data dalam *cluster* diulangi sampai nilai *centroid* konvergen (*centroid* dari semua *cluster* tidak berubah lagi). Algoritma *k-means* memiliki tahapan sebagai berikut [15]:

1. Pilih k pusat awal secara acak $C = \{c_1, c_2, c_3, \dots, c_k\}$.
2. Setiap $i \in \{1, \dots, k\}$, jadikan *cluster* C_i sebagai kumpulan titik di X yang lebih dekat dengan c_i daripada c_j untuk semua $j \neq i$.
3. Untuk setiap $i \in \{1, \dots, k\}$, jadikan c_i sebagai pusat dari seluruh titik pada C_i sehingga $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$
4. Ulangi tahap 2 dan 3 sampai semua pusat tidak berubah.

2.5 Metode *Silhouette Coefficient*

Silhouette Coefficient digunakan untuk melihat kualitas dan kekuatan *cluster* seberapa baik suatu objek ditempatkan dalam suatu *cluster*. Nilai *Silhouette Coefficient* rentang -1 sampai dengan 1, nilai *Silhouette Coefficient* mendekati 1 adalah hasil pengelompokan yang lebih baik. Tahapan perhitungan *Silhouette Coefficient* adalah sebagai berikut [16]:

- a. Hitung rata-rata jarak dari suatu dokumen, misalkan i dengan semua dokumen lain yang berada dalam satu *cluster*

$$a(i) = \frac{1}{|A|-1} \sum_{j \in A, j \neq i} d(i, j) \quad (2.11)$$

dengan j adalah dokumen lain dalam satu *cluster* A dan $d(i, j)$ adalah jarak antara dokumen i dengan j .

- b. Hitung rata-rata jarak dari dokumen i dengan semua dokumen di $cluster$ lain, dan diambil nilai terkecilnya.

$$d(i, C) = \frac{1}{|A|} \sum_{j \in C} d(i, j) \quad (2.12)$$

dengan $d(i, C)$ adalah jarak rata-rata dokumen i dengan semua objek pada $cluster$ lain C dimana $A \neq C$.

$$b(i) = \min_{C \neq A} d(i, C) \quad (2.13)$$

- c. Nilai *Silhouette Coefficient* nya adalah :

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (2.14)$$

dimana

$s(i)$ = Nilai *Silhouette Coefficient*

$a(i)$ = rata-rata jarak antar anggota dalam $cluster$

$b(i)$ = jarak terkecil antara anggota $cluster$ dengan anggota $cluster$ terdekatnya (*nearest neighbor*)

BAB III

METODE PENELITIAN

Bab ini membahas tentang metodologi sistem yang digunakan untuk menyelesaikan tugas akhir. Pengerjaan Tugas Akhir ini dibagi ke dalam beberapa tahap, berikut merupakan tahap-tahap pengerjaan Tugas Akhir ini:

3.1 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data mengenai usulan Program Kreativitas Mahasiswa ITS yang diperoleh dari Lembaga Pengembangan Pendidikan, Kemahasiswaan dan Hubungan Alumni (LP2KHA) ITS yang mencakup 6 kategori usulan PKM yaitu PKM-M, PKM-K, PKM-KC, PKM-PE, PKM-T dan PKM-PSH. Atribut yang terdapat dalam data usulan PKM meliputi nama mahasiswa, NRP mahasiswa, jurusan, judul usulan PKM, kategori, nama dosen pembimbing dan NIP dosen.

3.2 Studi literatur

Pada tahap ini dilakukan berbagai pengumpulan informasi pendukung yang dapat menunjang pengerjaan Tugas Akhir meliputi:

- a. Pengolahan *Text Mining* antara lain tahap ekstraksi kata kunci suatu dokumen menggunakan metode *Rapid Automatic Keyphrase Extraxction* (RAKE) dan pembentukan vektor konsep dengan menggunakan metode *Latent Semantic Analysis* (LSA) yang meliputi pembobotan *TF-IDF*, dan metode *Singular Value Decomposition*.
- b. Metode *clustering* menggunakan *k-means* yang digunakan untuk mengelompokkan konsep-konsep yang mirip menjadi satu *cluster*.
- c. Metode *Social Network Analysis* yang digunakan sebagai salah satu fitur pengelompokan mahasiswa.

3.3 Perancangan perangkat lunak

Pada tahap ini, perancangan perangkat lunak meliputi beberapa perancangan utama yaitu perancangan data, perancangan proses, dan perancangan antarmuka aplikasi.

- a. Tahap perancangan database meliputi perancangan data masukan. Data masukan aplikasi ini adalah usulan PKM ITS. Data-data tersebut telah tersimpan di dalam database. Data lain yang digunakan adalah daftar *stopword*. *Stopword* merupakan kata-kata yang dianggap tidak penting dalam ekstraksi kata kunci. *Stopword* tersebut juga telah tersimpan di database.
- a. Tahap perancangan proses merupakan tahap perancangan fungsi-fungsi dan algoritma yang terdapat dalam aplikasi. Aplikasi dibangun menggunakan bahasa pemrograman Java. Secara garis besar terdapat 5 proses utama yaitu ekstraksi konsep dokumen, identifikasi topik, identifikasi relasi mahasiswa ITS dengan *Sosial Network Analysis*, pengelompokan mahasiswa ITS, serta representasi *Graph*.
- b. Tahap perancangan antarmuka aplikasi menggunakan GUI pada Java

3.4 Pembuatan perangkat lunak

Pada tahap pembuatan perangkat lunak, rancangan data, proses, dan antarmuka diimplementasikan dengan bahasa pemrograman Java. Aplikasi utama dibangun menggunakan Netbeans IDE 8.2 untuk mempermudah proses *coding* dan perancangan antarmuka. Selain itu, pilihan *software* basisdata yang digunakan adalah MySQL karena waktu pemrosesannya cepat. Berikut proses implementasi program meliputi :

- a. Proses *Text Mining*
Proses ini terdiri dari dua tahapan utama yaitu ekstraksi kata kunci menggunakan metode RAKE dan pembentukan vektor konsep menggunakan metode LSA

- b. **Identifikasi Topik**
Identifikasi topik berfungsi untuk melakukan pengelompokan konsep-konsep dokumen menjadi kumpulan topik. Pengelompokan ini menggunakan metode *k-means*. Pada tahap ini dilakukan input parameter berupa jumlah *cluster* yang dikehendaki. Kemudian setelah didapatkan hasil *cluster* dilakukan pembentukan vektor topik dokumen.
- c. **Identifikasi Relasi Mahasiswa dengan SNA**
Social Network Analysis terdiri dari *node* yang merepresentasikan mahasiswa serta *edge* yang merepresentasikan hubungan antara mahasiswa. Hubungan antar mahasiswa didapatkan dari jumlah kerja sama antar mahasiswa dalam pengerjaan Project PKM.
- d. **Pengelompokan mahasiswa ITS**
Pada proses ini mahasiswa ITS dikelompokkan pada *cluster* menggunakan algoritma *k-means*. Fitur pengelompokan mahasiswa ITS berfungsi untuk mengelompokkan mahasiswa berdasarkan vektor topik yang dimiliki oleh mahasiswa serta fitur relasi yang dihasilkan oleh Analisis Jaringan Sosial (SNA). Setelah mahasiswa berhasil dikelompokkan maka kelompok mahasiswa divisualisasikan ke dalam *graph*.
- e. **Representasi *Graph***
Representasi *graph* merupakan visualisasi relasi yang menggambarkan sekumpulan *node* yang terkumpul dalam bentuk *graph*. Dimana *graph* jejaring sosial mahasiswa yang terbentuk dari hasil pengelompokan dengan algoritma *k-means* berdasarkan nilai k yang ditentukan.

3.5 Uji coba dan evaluasi

Pada tahap ini, uji coba dilakukan dengan 5 skema yaitu ekstraksi konsep dokumen, identifikasi topik, identifikasi relasi dengan SNA, pengelompokan mahasiswa dan representasi *graph*.

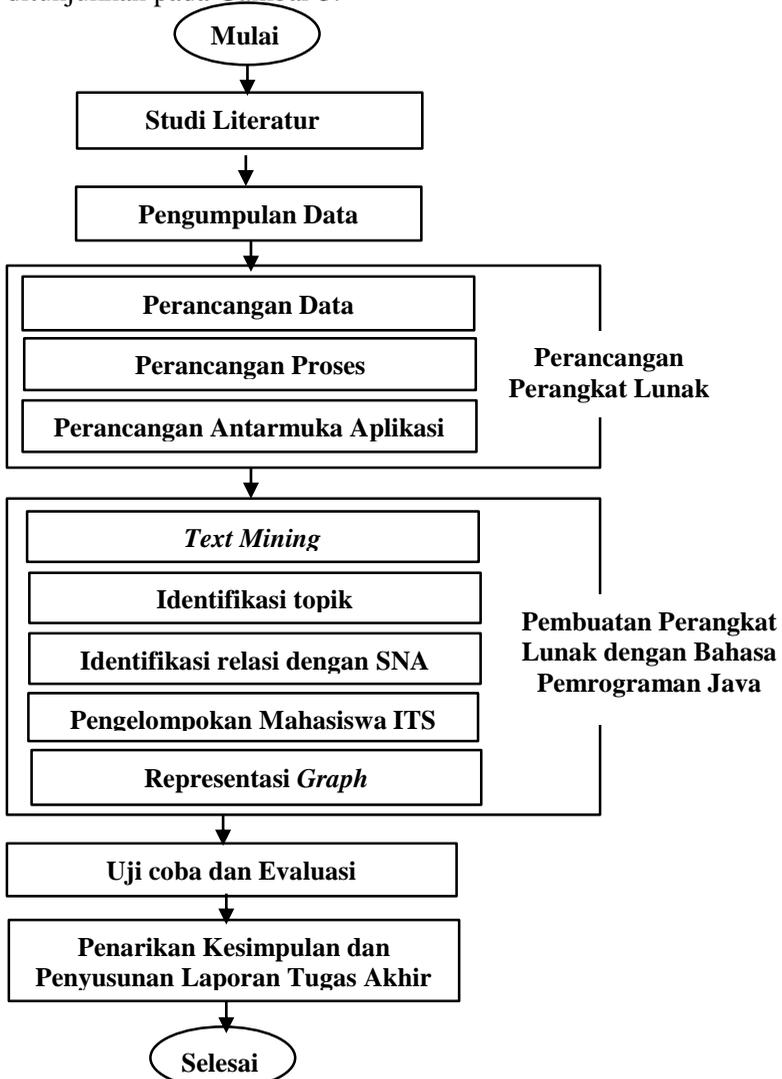
- a. **Ekstraksi konsep** bertujuan untuk mencari jumlah kata kunci yang tepat yang dapat diekstraksi dari dokumen. Uji ini berguna untuk mengetahui bagaimana proses ekstraksi kata

- kunci menggunakan metode RAKE berjalan dan pembentukan vektor konsep dengan LSA.
- b. Identifikasi topik bertujuan untuk mengelompokkan konsep yang memiliki kemiripan topik ke dalam beberapa cluster. Performa dan kemampuan pengelompokan dilakukan dengan menghitung rata-rata nilai kemiripan topik terhadap seluruh koleksi dokumen. Untuk memperoleh hasil *cluster* yang maksimal dilakukan beberapa uji coba pengelompokan dengan input parameter jumlah *cluster* yang berbeda.
 - c. Identifikasi relasi mahasiswa dengan perhitungan *centrality* pada *Social Network Analysis* yang meliputi *normalize degree centrality*, *normalize closeness centrality*, dan *normalize betweenness centrality* bertujuan untuk merepresentasikan hubungan antara mahasiswa.
 - d. Pengelompokan mahasiswa ITS bertujuan untuk mengelompokkan mahasiswa yang memiliki kemiripan dalam topik usulan PKM, dan properti jaringan ke dalam cluster yang sama. Pada uji pengelompokan mahasiswa ITS performa dan kemampuan dihitung berdasarkan nilai *Silhouette Coefficient*.
 - e. Proses pembentukan *graph* mahasiswa dilakukan dengan membentuk relasi antar mahasiswa pada *cluster* yang sama, kemudian barulah membentuk relasi antar mahasiswa antar *cluster*. Seluruh mahasiswa pada *cluster* yang sama memiliki relasi karena memiliki vektor topik dokumen yang mirip. Sedangkan relasi mahasiswa antar *cluster* didasarkan pada ada tidaknya kerja sama yang dapat diidentifikasi dari jumlah *Project* yang dikerjakan bersama.

3.6 Penarikan Kesimpulan dan Penyusunan Laporan Tugas Akhir

Pada tahap akhir penelitian ini dilakukan penarikan kesimpulan dan penyusunan laporan Tugas Akhir dari hasil kajian terhadap *graph* jejaring sosial menggunakan *Social Network Analysis* (SNA) pada pengelompokan mahasiswa ITS berdasarkan data usulan PKM.

Berikut adalah diagram alir pengerjaan Tugas Akhir ini yang ditunjukkan pada Gambar 3:



Gambar 3. Diagram alir pengerjaan Tugas Akhir

BAB IV

PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini membahas analisa dan perancangan sistem perangkat lunak yang akan dibangun dalam identifikasi relasi mahasiswa ITS menggunakan *graph social network analysis* pada pengelompokan mahasiswa berdasarkan data usulan PKM. Analisa sistem terdiri dari analisa kebutuhan sistem berupa deskripsi umum perangkat lunak dan lingkungan perancangan perangkat lunak.

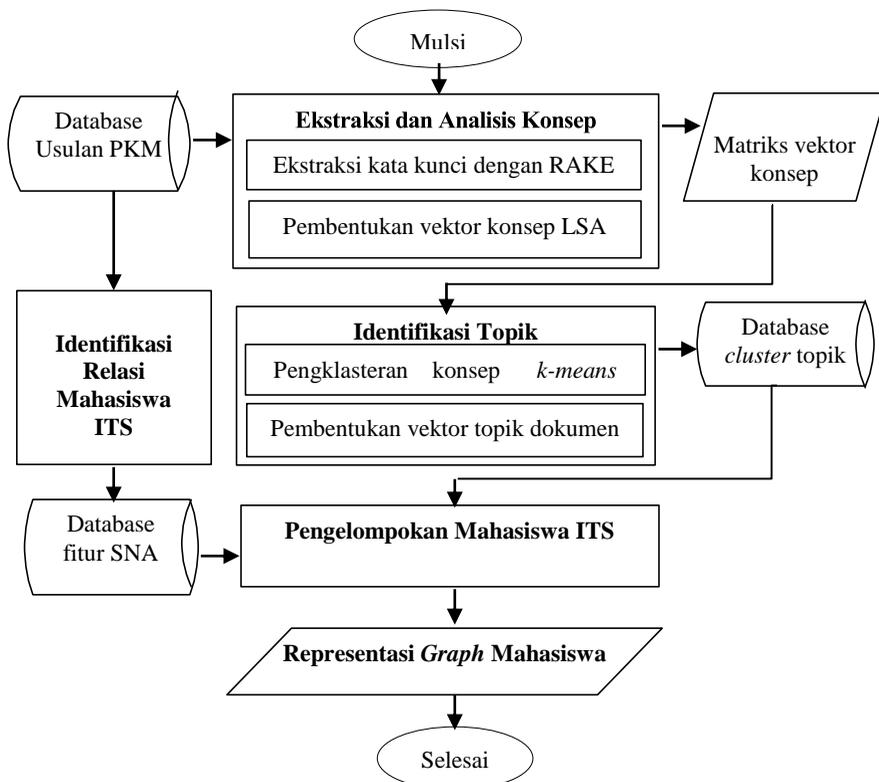
Pada perancangan sistem akan membahas tentang sistem dari perangkat lunak yang akan dibangun, perancangan data yang meliputi data masukan dan data keluaran, serta perancangan algoritma proses-proses yang ada dalam tugas akhir ini. Selanjutnya membahas implementasi sistem yang meliputi hasil implementasi antarmuka dan keseluruhan proses di dalam sistem.

4.1 Analisis Sistem

Pada Analisis Sistem akan dibahas mengenai deskripsi secara umum mengenai perangkat lunak yang dibuat dan lingkungan perancangan perangkat lunak.

4.1.1 Deskripsi Umum Perangkat Lunak

Perangkat lunak ini mampu melakukan proses *text mining* yang meliputi tahap ekstraksi kata kunci dokumen usulan PKM menggunakan *Rapid Automatic Keyphrase Extraction (RAKE)*, pembentukan vektor konsep menggunakan metode *Latent Semantic Analysis (LSA)*, identifikasi topik menggunakan algoritma *k-means* dan identifikasi relasi mahasiswa ITS menggunakan *Social Network Analysis (SNA)*. Selain itu, mampu melakukan pengelompokan mahasiswa berdasarkan fitur topik dan relasi SNA menggunakan *k-means*, kemudian membangun representasi *graph* yang menyajikan hubungan antar mahasiswa ITS berdasarkan data usulan PKM. Untuk memenuhi kebutuhan tersebut maka perangkat lunak ini dibagi menjadi lima bagian utama, dapat dilihat pada Gambar 4.1



Gambar 4.1 Deskripsi Umum Perangkat Lunak

- a. Ekstraksi konsep berfungsi untuk melakukan ekstraksi kata kunci seluruh dokumen yang ada. Selain itu fitur ini juga berfungsi untuk membentuk vektor konsep dari kata kunci yang dihasilkan. Proses yang dilakukan pada fitur ini adalah ekstraksi kata kunci menggunakan metode RAKE, serta pembentukan vektor konsep menggunakan metode LSA. Untuk ekstraksi konsep menggunakan LSA menggunakan library *JAMA* dalam penghitungan *Singular Value Decomposition* (SVD).

- b. Identifikasi topik berfungsi untuk melakukan pengelompokan konsep-konsep dokumen menjadi kumpulan topik. Proses yang dapat dilakukan pada fitur ini adalah deteksi konsep dengan metode *k-means*. Kemudian setelah didapatkan hasil *cluster* dilakukan pembentukan vektor topik dokumen.
- c. Identifikasi relasi mahasiswa dengan SNA merepresentasikan hubungan antara mahasiswa. Hubungan antar mahasiswa didapatkan dari jumlah kerja sama antar mahasiswa dalam pengerjaan PKM.
- d. Pengelompokan mahasiswa berfungsi untuk mengelompokkan mahasiswa berdasarkan vektor topik yang dimiliki serta fitur relasi yang dihasilkan oleh SNA.
- e. Representasi *graph* berfungsi untuk memvisualisasikan hasil pengelompokan mahasiswa ke dalam *graph* dengan menggunakan *library* JUNG

4.1.2 Lingkungan Perancangan Perangkat Lunak

Dalam proses perancangan perangkat lunak *project* ini, dibutuhkan lingkungan perancangan yang sesuai. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam pengerjaan *Project* dijelaskan sebagai berikut:

Perangkat Keras: 1. *Processor*: AMD A8 – 6410 APU CPU@
2,0 GHz

2. Memori (RAM) : 8,00 GB

Perangkat Lunak: 1. Sistem Operasi: Microsoft Windows 7

2. Perangkat Pengembang : Netbeans IDE 8.2,
XAMPP versi 3.2.1

4.2 Perancangan Data

Data-data yang digunakan dalam perangkat lunak ini dapat dibedakan menjadi dua jenis, yaitu data masukan dan data keluaran.

Table	Action
<input type="checkbox"/> datauji	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> data_pkm	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> dokclustervektor	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> list_kata_kunci	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> nodesnafitur	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> representasi_dok_rake	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> scientistcluster	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> stoplist	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> studentclustervektor	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> topikcluster	★ Browse Structure Search Insert Empty Drop
10 table(s)	Sum

Gambar 4.2 Tabel *Database*

4.2.1 Data Masukan

Data masukan adalah data-data yang digunakan sebagai masukan dari aplikasi. Masukan-masukan ini yang kemudian diolah oleh aplikasi melalui tahap-tahap tertentu sehingga menghasilkan keluaran yang diinginkan. Data masukan yang digunakan pada aplikasi ini adalah:

1. Data usulan PKM berupa dokumen *Project* yang telah disimpan di dalam *database*.
2. Data kumpulan *stopword* yang telah tersimpan di *database*.

Database usulan PKM yang berupa *Project* memiliki beberapa atribut yaitu:

1. No merupakan id dokumen.
2. Judul_Usulan merupakan judul *Project*.
3. NRP_Ketua merupakan nrp mahasiswa pengusul PKM pertama.
4. NRP_Anggota1 merupakan nrp mahasiswa pengusul PKM kedua.
5. NRP_Anggota2 merupakan nrp mahasiswa pengusul PKM ketiga.
6. NRP_Anggota3 merupakan nrp mahasiswa pengusul PKM keempat.
7. NRP_Anggota4 merupakan nrp mahasiswa pengusul PKM kelima.

4.2.2 Data Keluaran

Data keluaran merupakan data yang dihasilkan oleh aplikasi setelah proses-proses tertentu selesai dilakukan. Terdapat beberapa data keluaran pada aplikasi ini, yaitu:

1. Data hasil ekstraksi kata kunci menggunakan metode *Rapid Automatic Keyphrase Extraction (RAKE)* beserta vektor tiap konsep menggunakan *Latent Semantic Analysis (LSA)*.
2. Data hasil identifikasi topik menggunakan algoritma *k-means* berupa *cluster* beserta konsep yang berada di dalamnya.
3. Data hasil penghitungan *document cluster vektor* yaitu kesamaan tiap dokumen terhadap seluruh *cluster* topik yang ada.
4. Data hasil Analisis Jaringan Sosial berupa nilai *normalize degree centrality, normalize closeness centrality & normalize betweenness centrality*.
5. Data hasil pengelompokan mahasiswa yang memuat hasil pengukuran terhadap fitur relasi Analisis Jaringan Sosial dan vektor topik dokumen mahasiswa.

4.3 Perancangan Proses

Terdapat lima bagian utama yang dapat dilakukan pada perangkat lunak ini antara lain:

4.3.1 Ekstraksi konsep dokumen

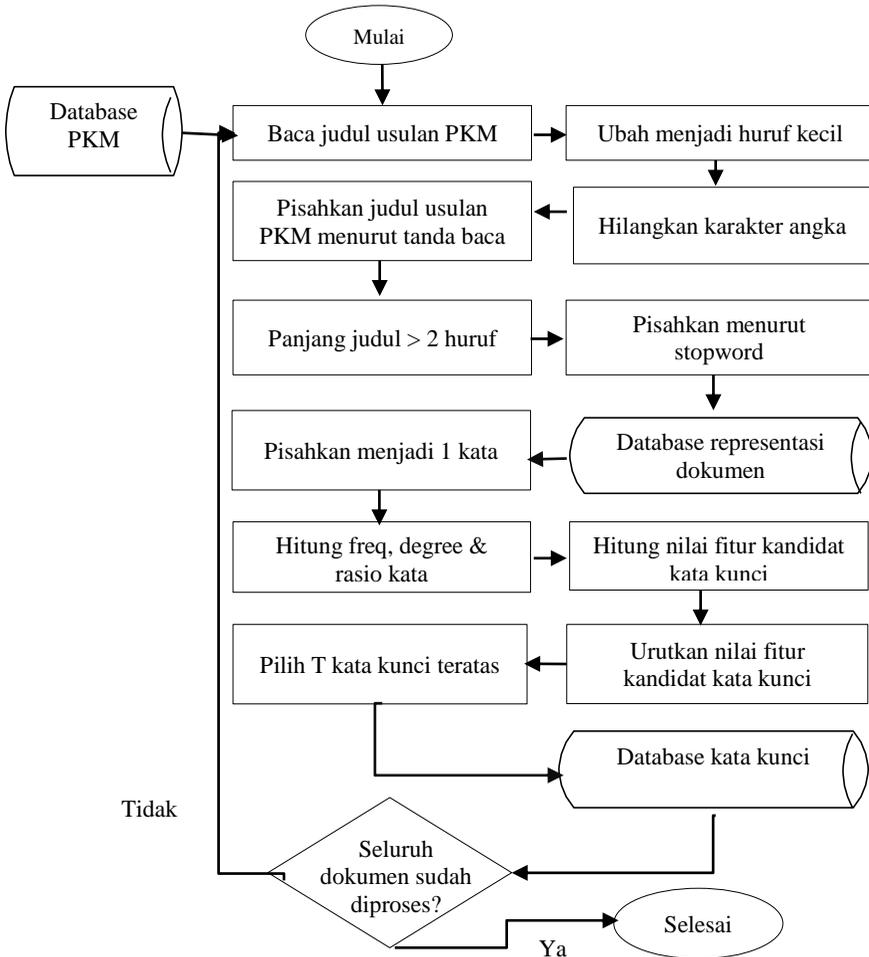
Proses ini berfungsi untuk melakukan ekstraksi konsep dari koleksi dokumen yang ada pada database. Proses ini memiliki dua tahapan utama yaitu ekstraksi kata kunci menggunakan metode *Rapid Automatic Keyphrase Extraction (RAKE)*, dan ekstraksi konsep menggunakan metode *Latent Semantic Analysis (LSA)*. Pada proses ini pengguna memasukkan parameter berupa jumlah kata kunci teratas yang diambil pada tahap ekstraksi kata kunci menggunakan metode RAKE. Setelah itu kata kunci dari seluruh dokumen tersebut dijadikan *input* pada tahap pembentukan vektor konsep dokumen menggunakan metode *Latent Semantic Analysis (LSA)*. Hasil dari proses ini berupa konsep beserta vektor konsep.

4.3.1.1 Ekstraksi kata kunci

Proses ekstraksi kata kunci menggunakan metode RAKE memiliki 5 tahapan utama, yaitu ekstraksi kandidat kata kunci, menghitung matriks *co-occurrence*, menghitung nilai rasio, menghitung nilai fitur dasar, dan memilih kata kunci dengan nilai fitur tertinggi. Langkah-langkah ekstraksi kata kunci sebagai berikut:

- Input:
Dokumen *Project* / Database usulan PKM
- Output:
Kata kunci / kata penting yang mewakili dokumen
- Langkah-langkah ekstraksi kata kunci RAKE adalah sebagai berikut:
 1. Pengguna memasukkan jumlah kata kunci yang digunakan.
 2. Baca judul usulan dokumen *project* dari *database*.
 3. Ubah seluruh huruf pada judul usulan menjadi huruf kecil.
 4. Hilangkan seluruh karakter angka.
 5. Pisahkan isi judul usulan menurut tanda baca.
 6. Simpan seluruh kata atau frase yang memiliki panjang lebih dari 2 huruf sebagai representasi dokumen.
 7. Untuk setiap kata atau frase yang telah disimpan pisahkan menurut stopword.
 8. Simpan seluruh kata dan frase sebagai kandidat kata kunci.
 9. Pisahkan seluruh kandidat kata kunci yang berupa frase menjadi satu kata.
 10. Hitung frekuensi kemunculan setiap kata di dalam dokumen $freq(w)$.
 11. Hitung nilai degree setiap kata $deg(w)$.
 12. Hitung nilai rasio kata, $rasio(w) = deg(w) / freq(w)$ setiap kata
 13. Hitung nilai fitur setiap kandidat kata kunci dengan cara menambahkan nilai rasio tiap kata yang ada pada kandidat kata kunci
 14. Urutkan nilai fitur kandidat kata kunci dari kecil ke besar

15. Simpan kata kunci sebanyak input jumlah kata kunci yang dimasukkan oleh pengguna ke dalam *database*.
16. Ulangi untuk setiap dokumen di dalam *database*



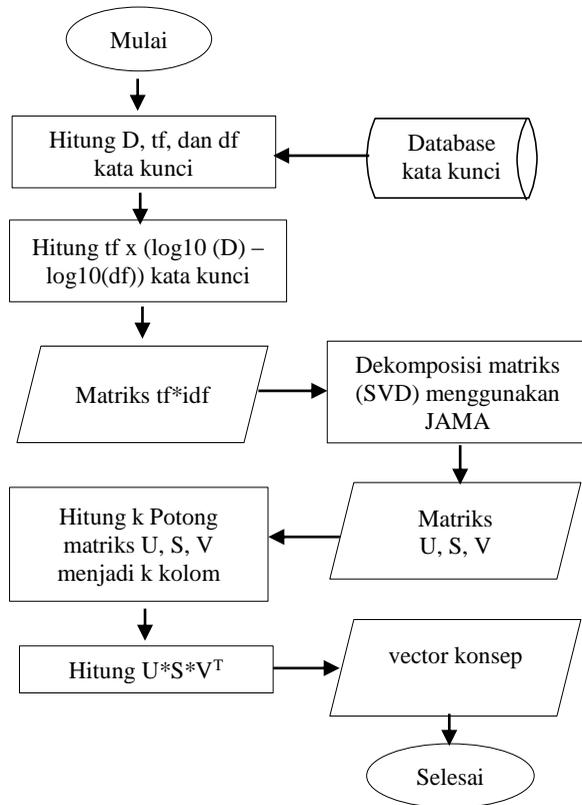
Gambar 4.3 Ekstraksi kata kunci menggunakan RAKE

4.3.1.2 Pembentukan vektor konsep

Proses pembentukan vektor konsep menggunakan metode LSA memiliki 2 tahapan utama yaitu membangun matriks pembobotan *tf-idf* dan mengaplikasikan metode *Singular Value Decomposition* (SVD) pada matriks yang telah dibentuk.

Langkah-langkah pembentukan vektor konsep adalah sebagai berikut:

- Input:
Kata kunci
- Output:
Konsep beserta vektor konsep
- Langkah-langkah pembentukan vektor konsep LSA adalah sebagai berikut:
 1. Hitung frekuensi kemunculan kata kunci pada dokumen (*tf*)
 2. Hitung jumlah koleksi dokumen yang ada (*D*)
 3. Hitung jumlah dokumen yang mengandung kata kunci tersebut (*df*)
 4. Hitung $tf \times (\log_{10}(D) - \log_{10}(df))$ yang merupakan nilai *inverse document frequency* (*idf*) setiap kata
 5. Dekomposisi *matriks invers document frequency* (*idf*) tersebut menjadi matriks *U*, matriks Σ , serta matriks *V*. Matriks *U* merupakan vektor kata kunci, Matriks Σ merupakan vektor sigma, dan Matriks *V* merupakan vektor dokumen
 6. Hitung $k = \sqrt{\text{jumlah kolom matriks}}$ yang merupakan nilai batas pemotongan kolom pada matriks *U*, Σ , dan *V*.
 7. Kembalikan matriks ke bentuk awal dengan mengalikan metrik *U*, Σ , dan V_T
 8. Simpan seluruh konsep beserta matriks barisnya sebagai vektor konsep.



Gambar 4.4 Proses pembentukan vektor konsep

4.3.2 Identifikasi topik

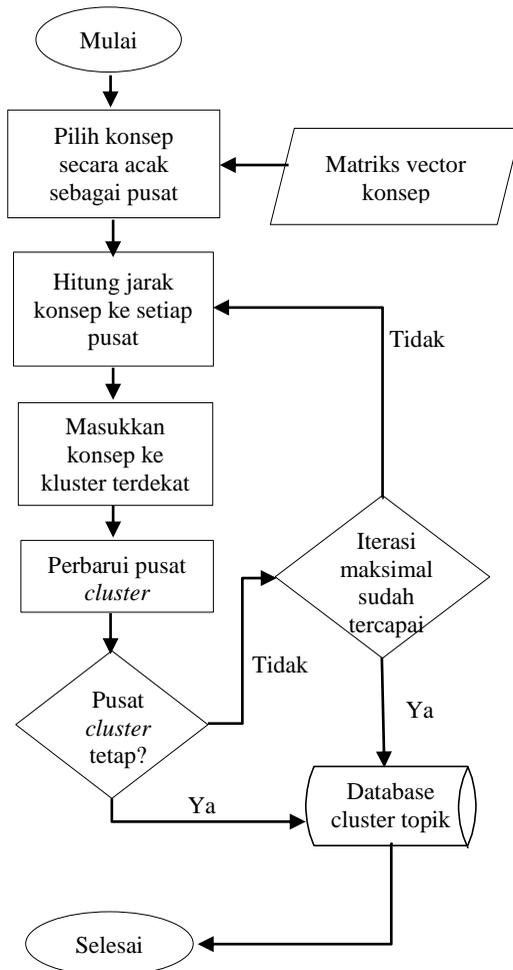
Pada proses ini konsep-konsep yang sudah terekstraksi dari dokumen dikelompokkan menjadi beberapa *cluster* yang mewakili sebuah topik. Proses identifikasi topik memiliki 2 tahapan utama yaitu pengelompokan konsep dan pembentukan vektor topik dokumen.

4.3.2.1 Pengelompokan Konsep

Proses identifikasi topik dilakukan dengan menggunakan metode *clustering*, yaitu *k-means*. Proses ini bertujuan untuk mengelompokkan konsep-konsep ke dalam beberapa *cluster*.

Langkah-langkah identifikasi topik adalah sebagai berikut:

- Input:
Konsep beserta vektor baris hasil metode *Latent Semantic Analysis*.
- Output:
Cluster topik beserta konsep yang ada di dalamnya
- Langkah-langkah:
 1. Memasukkan *input* berupa jumlah *cluster* yang diinginkan
 2. Pilih konsep secara acak sebagai pusat *cluster*
 3. Hitung jarak setiap konsep ke setiap pusat dengan menggunakan persamaan *euclidean distance*
 4. Jika jarak suatu konsep terhadap suatu pusat *cluster* lebih kecil dari pada jarak ke pusat *cluster* yang lain, maka masukkan konsep ke *cluster* tersebut
 5. Perbaharui nilai pusat *cluster* dengan mencari nilai rata-rata vektor dari seluruh anggota *cluster*
 6. Hitung kembali jarak tiap konsep ke pusat *cluster* untuk mengalokasikan kembali tiap konsep ke dalam *cluster*
 7. Ulangi tahap 7 dan 8 sampai pusat tidak berubah atau nilai iterasi maksimal sudah dicapai
 8. Simpan seluruh *cluster* beserta konsep yang ada di dalamnya ke dalam *database*

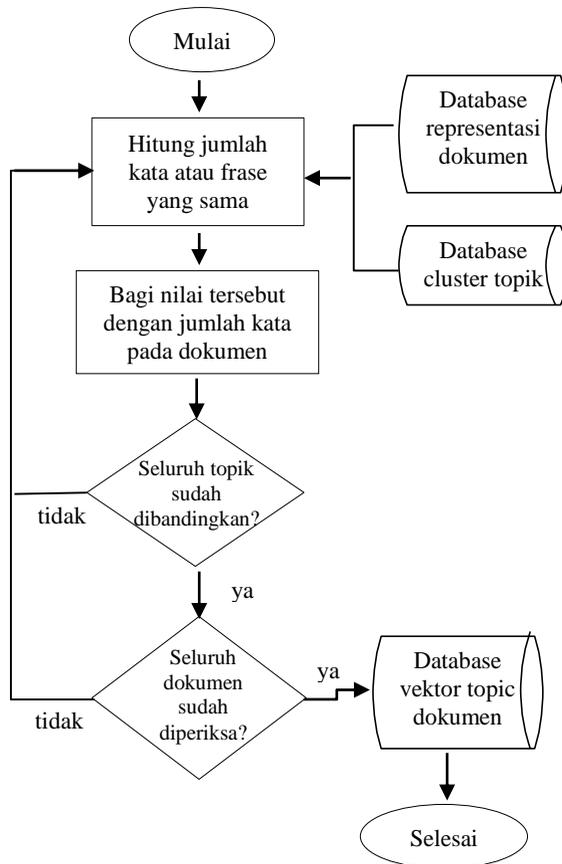


Gambar 4.5 Proses identifikasi topik

4.3.2.2 Pembentukan vektor topik dokumen

Pembentukan vektor topik dokumen bertujuan untuk mengukur kedekatan suatu dokumen terhadap topik-topik yang dihasilkan pada proses deteksi topik. Proses ini dilakukan dengan menghitung jumlah kata atau frase yang sama dengan kata atau frase yang ada pada setiap topik. Nilai tersebut dinormalisasi dengan cara membagi setiap nilai dengan jumlah kata atau frase total yang ada pada dokumen tersebut. Berikut tahap pembentukan vektor topik dokumen:

- Input
Dokumen (database representasi dokumen) dan topik.
- Output
Vektor topik dokumen
- Langkah-langkah
 1. Baca dokumen beserta kata atau frase representasinya dari *database*
 2. Baca topik beserta kata kunci yang termasuk dalam topik tersebut
 3. Hitung jumlah kata atau frase kata kunci yang sama
 4. Normalisasi nilai tersebut dengan cara membagi dengan jumlah kata atau frase total pada dokumen tersebut
 5. Ulangi langkah 2, 3, dan 4 untuk setiap topik
 6. Ulangi langkah 1, 2, 3, dan 4 untuk setiap dokumen
 7. Simpan seluruh vektor topik dokumen ke dalam *database*



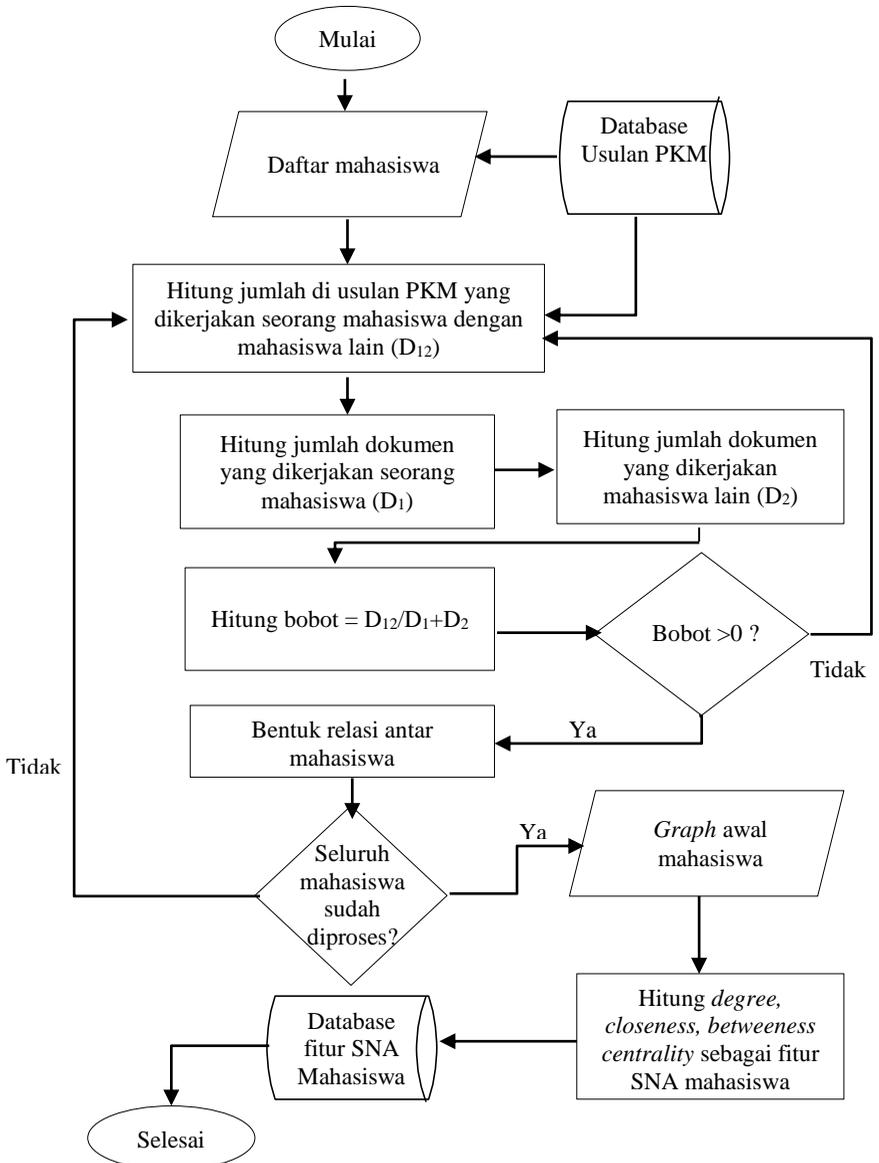
Gambar 4.6 Proses pembentukan vektor topik dokumen

4.3.3 Identifikasi relasi mahasiswa

Identifikasi relasi berguna untuk mengekstraksi fitur Analisis Jaringan Sosial mahasiswa. Jaringan sosial mahasiswa terdiri dari *node* yang merepresentasikan mahasiswa serta *edge* yang merepresentasikan hubungan antara mahasiswa. Hubungan antar mahasiswa didapatkan dari jumlah kerja sama antar mahasiswa dalam pengerjaan *Project* PKM.

Berikut tahap pembentukan relasi awal mahasiswa:

- Input
Database Project / Database usulan PKM
- Output
Fitur pengukuran Analisis Jaringan Sosial mahasiswa
- Langkah-langkah
 1. Simpan seluruh nrp mahasiswa yang ada pada *database Project* pada variabel daftar mahasiswa
 2. Hitung jumlah *Project* PKM yang diusulkan oleh mahasiswa pada daftar pertama dengan mahasiswa lain.
 3. Hitung jumlah *Project* PKM yang dikerjakan mahasiswa tersebut
 4. Hitung jumlah dokumen *Project* yang dikerjakan mahasiswa lain
 5. Hitung bobot relasi dengan membagi jumlah *Project* yang dikerjakan oleh kedua mahasiswa dengan jumlah *Project* yang dikerjakan mahasiswa pertama ditambah jumlah *Project* yang dikerjakan mahasiswa kedua
 6. Jika nilai bobot lebih dari 0 maka bentuk relasi antar kedua mahasiswa tersebut
 7. Ulangi langkah 2, 3, 4, dan 5 untuk seluruh mahasiswa. Setelah didapatkan seluruh relasi mahasiswa maka akan terbentuk jaringan sosial komunitas keilmiahan mahasiswa berupa graph.
 8. Hitung nilai *normalize degree centrality*, *normalize closeness centrality*, dan *normalize betweenness centrality* tiap *node*, kemudian simpan di dalam *database*



Gambar 4.7 Proses pembentukan awal relasi mahasiswa

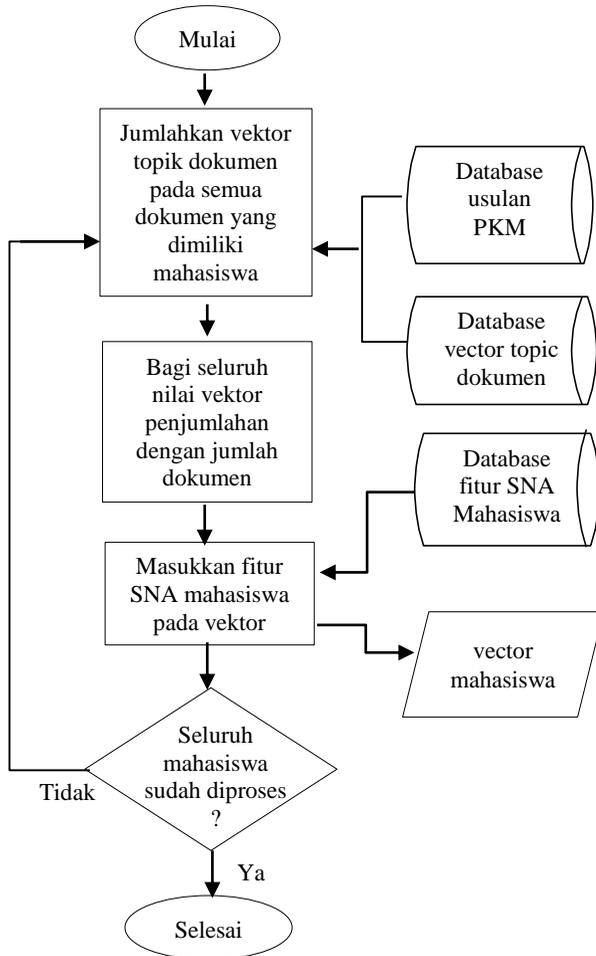
4.3.4 Pengelompokan mahasiswa

Pada proses ini mahasiswa dikelompokkan pada *cluster* menggunakan algoritma *k-means*. Pengguna memasukkan parameter berupa jumlah *cluster* yang dikehendaki. Proses pengelompokan mahasiswa memiliki 2 tahapan utama yaitu pembentukan vektor mahasiswa dan proses pengelompokan mahasiswa itu sendiri.

4.3.4.1 Pembentukan vektor mahasiswa

Pembentukan vektor mahasiswa bertujuan untuk mengukur kedekatan mahasiswa terhadap topik-topik yang dihasilkan pada proses deteksi topik. Proses ini dilakukan dengan menjumlahkan nilai vektor dokumen pada dokumen yang dimiliki oleh mahasiswa. Nilai tersebut dinormalisasi dengan cara membagi setiap nilai dengan jumlah dokumen yang dimiliki oleh mahasiswa. Selain nilai tersebut nilai fitur pengukuran Analisis Jaringan Sosial dosen tersebut juga disimpan sebagai vektor mahasiswa. Berikut tahap pembentukan vektor topik mahasiswa:

- Input
Dokumen mahasiswa, dan vektor topik dokumen
- Output
Vektor mahasiswa
- Langkah-langkah
 1. Baca mahasiswa beserta dokumen yang dimiliki dari *database*
 2. Jumlahkan vektor dokumen yang dimiliki oleh mahasiswa tersebut
 3. Normalisasi nilai tersebut dengan cara membagi dengan jumlah dokumen yang dimiliki oleh mahasiswa
 4. Dapatkan nilai fitur pengukuran Analisis Jaringan Sosial dosen tersebut, kemudian
 5. Simpan vektor dokumen dan nilai fitur Analisis Jaringan Sosial mahasiswa sebagai vektor mahasiswa
 6. Ulangi 1, 2, 3, 4, dan 5 untuk setiap mahasiswa



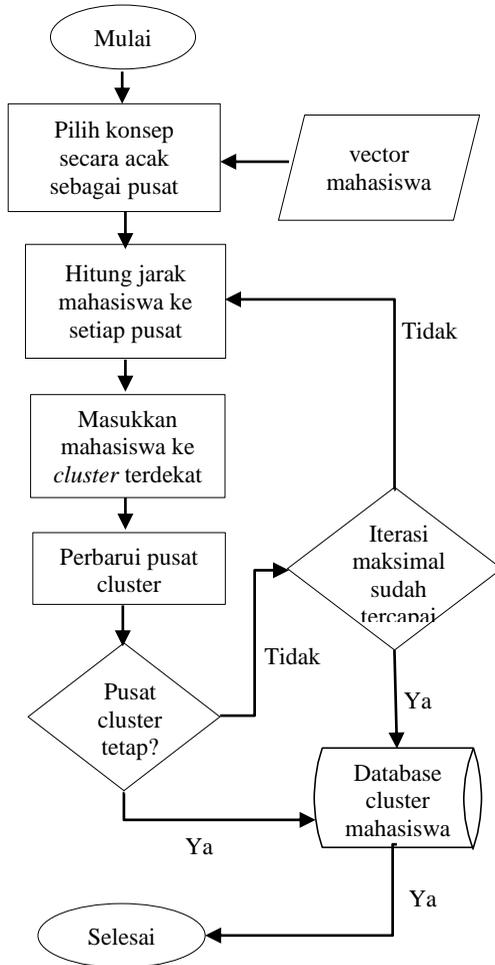
Gambar 4.8 Proses pembentukan vektor mahasiswa

4.3.4.2 Pengelompokan mahasiswa

Proses pengelompokan mahasiswa dilakukan dengan menggunakan metode pengelompokan, yaitu *k-means*.

Langkah-langkah pengelompokan mahasiswa adalah sebagai berikut:

- Input:
Mahasiswa beserta vektornya
- Output:
Cluster mahasiswa beserta mahasiswa yang ada di dalamnya
- Langkah-langkah:
 1. Memasukkan input berupa jumlah *cluster* yang diinginkan
 2. Pilih mahasiswa secara acak sebagai pusat *cluster*
 4. Hitung jarak setiap mahasiswa ke setiap pusat dengan menggunakan persamaan *euclidean distance*
 5. Jika jarak suatu mahasiswa terhadap suatu pusat *cluster* lebih kecil dari pada jarak ke pusat *cluster* yang lain, maka masukkan mahasiswa ke *cluster* tersebut
 6. Perbaharui nilai pusat *cluster* dengan mencari nilai rata-rata vektor dari seluruh anggota *cluster*
 7. Hitung kembali jarak tiap mahasiswa ke pusat *cluster* untuk mengalokasikan kembali tiap mahasiswa ke dalam *cluster*
 8. Ulangi tahap 7 dan 8 sampai pusat tidak berubah atau nilai iterasi maksimal sudah dicapai
 9. Simpan seluruh *cluster* beserta mahasiswa yang ada di dalamnya ke dalam database



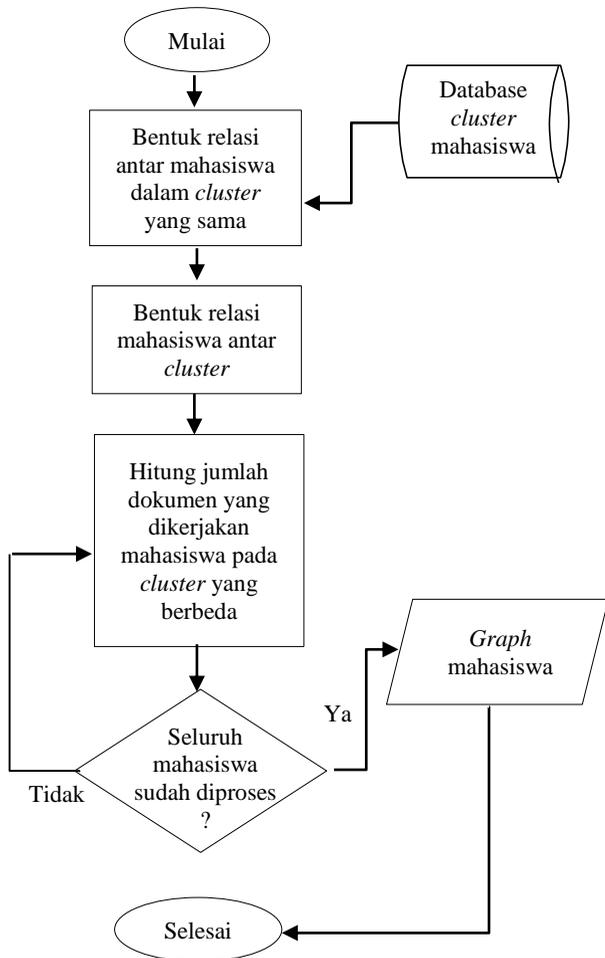
Gambar 4.9 Proses pengelompokan mahasiswa

4.3.5 Representasi *graph* mahasiswa

Proses pembentukan *graph* mahasiswa dilakukan dengan membentuk relasi antar mahasiswa pada *cluster* yang sama, kemudian barulah membentuk relasi antar mahasiswa antar *cluster*. Seluruh mahasiswa pada *cluster* yang sama memiliki relasi karena memiliki vektor topik dokumen yang mirip. Sedangkan relasi mahasiswa antar *cluster* didasarkan pada ada tidaknya kerja sama yang dapat diidentifikasi dari jumlah *Project* yang dikerjakan bersama.

Berikut tahap-tahap pembentukan *graph* mahasiswa:

- Input:
Cluster mahasiswa
- Output:
Graph mahasiswa
- Langkah-langkah:
 1. Dapatkan seluruh mahasiswa pada *cluster*
 2. Bentuk relasi antar mahasiswa pada *cluster* tersebut
 3. Ulangi langkah 1 dan 2 untuk semua *cluster*
 4. Dapatkan seluruh mahasiswa pada suatu *cluster* dan *cluster* yang lain
 5. Hitung jumlah dokumen yang dikerjakan oleh mahasiswa pada *cluster* pertama dan mahasiswa pada *cluster* kedua
 6. Jika jumlah dokumen lebih dari 0 maka bentuk relasi antar mahasiswa antar *cluster*
 7. Ulangi langkah 4, 5, dan 6 untuk kombinasi seluruh *cluster*



Gambar 4.10 Proses pembentukan *graph* mahasiswa

4.4 Implementasi sistem

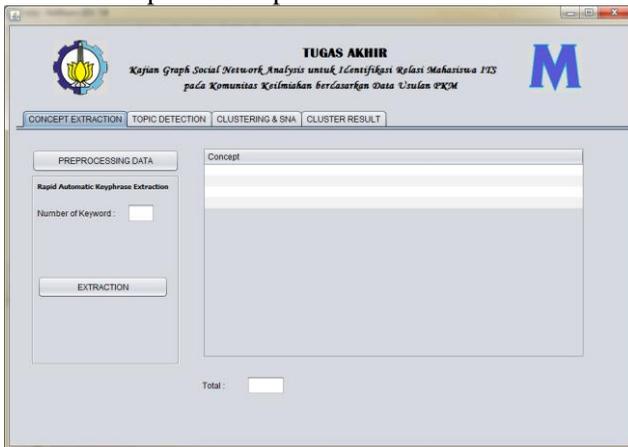
Setelah perancangan selesai, tahap selanjutnya adalah implementasi. Tahap ini bertujuan agar user dapat menggunakan program yang telah dirancang.

4.4.1 Implementasi Antarmuka Aplikasi

Pada sub bab ini dijelaskan tentang kegunaan fungsi-fungsi yang ada di dalam aplikasi beserta tampilan desain. Antarmuka dibagi menjadi 4 tab, yaitu tab *concept extraction*, *topic detection*, *clustering & SNA* serta *cluster result*.

4.4.1.1 Tab *Concept Extraction*

Tab ini berguna untuk proses ekstraksi konsep. Pada tab ini terdapat tombol *preprocessing* data dan *extraction*. Sebelum memulai ekstraksi, pengguna harus memasukkan jumlah kata kunci yang dipilih pada proses ekstraksi kata kunci dengan memasukkan parameter pada *field* jumlah kata kunci. Pada tab ini juga terdapat tabel yang digunakan untuk menampilkan konsep-konsep yang berhasil diekstraksi dari dokumen. Implementasi tab *concept extraction* dapat dilihat pada Gambar 4.10.



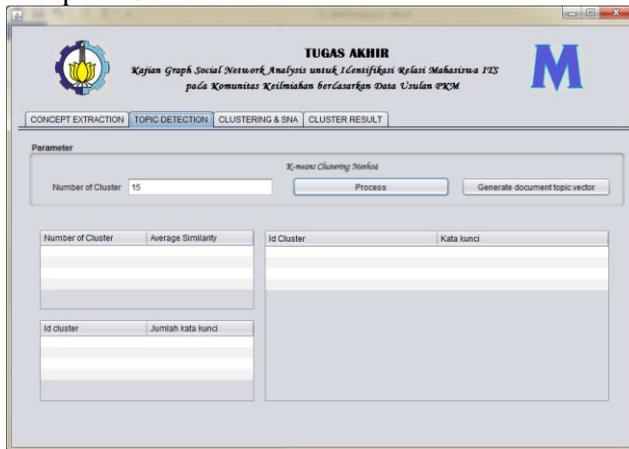
Gambar 4.11 Tab *Concept Extraction*

Tabel 4.1 Tombol dan kegunaan pada tab *concept extraction*

Tombol	Kegunaan
<i>Preprocessing data</i>	Melakukan pemrosesan data sebelum digunakan yakni meliputi case folding dan penghilangan stopword
<i>Extraction</i>	Memulai proses ekstraksi konsep

4.4.1.2 Tab *Topic Detection*

Tab berikutnya merupakan tab deteksi topik. Tab ini berfungsi untuk mengelompokkan konsep yang telah diekstraksi ke dalam beberapa cluster. Terdapat input parameter jumlah cluster yang diinginkan, tombol *process*, dan tombol *generate document topic vector*. Tabel pada tab ini berguna untuk menampilkan hasil pengelompokan konsep, dengan setiap cluster dianggap sebagai topik yang berhasil diekstraksi. Implementasi tab *topic detection* dapat dilihat pada Gambar 4.11.



Gambar 4.12 Tab *Topic Detection*

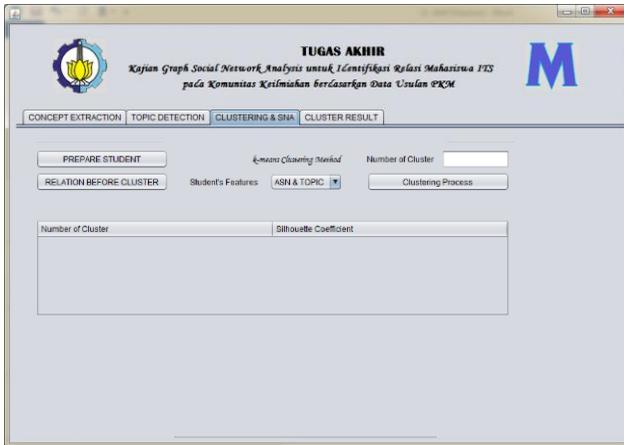
Tabel 4.2 Tombol dan kegunaan pada tab *topic detection*

Tombol	Kegunaan
<i>Process</i>	Memulai proses pengelompokan topik
<i>Generate document topic vector</i>	Memulai proses pembentukan vektor topik dokumen

4.4.1.3 Tab *Clustering & SNA*

Tab berikutnya adalah pengelompokan mahasiswa. Tab ini berguna untuk mengelompokkan mahasiswa berdasarkan vektor fitur pengukuran Analisis Jaringan Sosial, dan vektor topik

dokumen yang dimilikinya. Pada tab ini terdapat *field* input jumlah *cluster* yang diinginkan. Selain itu terdapat 3 tombol yaitu *prepare student*, *relation before cluster*, dan *clustering process*. Pada tab ini juga terdapat tabel yang digunakan untuk menampilkan hasil pengukuran *networks strength* dari jaringan sosial yang telah terbentuk.



Gambar 4.13 Tab *Clustering & SNA*

Tabel 4.3 Tombol dan kegunaan pada tab *clustering & SNA*

Tombol	Kegunaan
<i>Prepare Student</i>	Membentuk vektor mahasiswa yang digunakan sebagai nilai pengukuran jarak antar mahasiswa
<i>Relation before cluster</i>	Memulai proses pembentukan relasi awal mahasiswa an perhitungan <i>centrality</i>
<i>Clustering Process</i>	Memulai proses pengelompokkan mahasiswa berdasar fitur terpilih

4.4.1.4 Tab *Cluster Result*

Tab terakhir adalah tab Cluster Dosen. Pada tab ini ditampilkan daftar mahasiswa yang ada pada masing masing *cluster*. Untuk menampilkan tekan tombol Tampilkan.



Gambar 4.14 Tab *Cluster Result*

4.4.2 Implementasi Program

Perangkat lunak dalam Project ini dibuat menggunakan bahasa pemrograman Java pada platform Java Development Kit 7 dan Netbeans IDE 8.2 dengan library tambahan adalah JAMA, dan JUNG. Database server yang digunakan adalah MySQL.

4.4.2.1 Implementasi Ekstraksi Konsep

Proses ekstraksi konsep terdiri dari dua tahap yaitu ekstraksi kata kunci dari judul usulan dokumen Project dan pembentukan vektor konsep menggunakan metode *Latent Semantic Analysis* (LSA). Ekstraksi kunci dimulai dengan memisahkan judul usulan dokumen berdasarkan tanda baca dan stopword. Proses ini dilakukan pada kelas RAKE.

```

/* Implementasi proses ekstraksi konsep*/
listKata = new ArrayList<>();
listKataKunci = new ArrayList<>();
Database db = new Database();
db.connectFirst();
String Dok = dokumen.getJudul();
ArrayList<String> tempKata = new ArrayList<>();
String hasil = prosesLowerCase(Dok);
/*Split berdasarkan tanda baca*/
ArrayList<String> kandidatKK = prosesSplitTB(hasil);
/*Split berdasarkan stopword*/
kandidatKK = prosesSplitStopword(kandidatKK);

```

Pada kode di atas terdapat beberapa *method* pada kelas RAKE yang digunakan yaitu:

1. prosesLowerCase()

Method ini berguna untuk mengubah seluruh huruf dalam judul usulan menjadi huruf kecil.

```

/* Implementasi proses pemisahan judul usulan berdasarkan
tanda baca*/
private ArrayList<String> prosesSplitTB(String judulDok){
    ArrayList<String> hasil = new ArrayList<String>();
    judulDok = judulDok.replaceAll("[0-9]", "");
    String[] tempHasil = judulDok.split("\\p{Punct}");
    for(int i = 0; i < tempHasil.length; i++){
        if(tempHasil[i].length() > 2){
            StringBuilder sb = new StringBuilder();
            String[] buildString = tempHasil[i].split(" ");
            for(int j = 0; j < buildString.length; j++){
                if(buildString[j].length() > 1){
                    if(j < buildString.length-1){
                        sb.append(buildString[j]);
                        sb.append(" ");
                    }
                    else{
                        sb.append(buildString[j]);
                    }
                }
            }
            hasil.add(sb.toString());
        }
    }
    return hasil;
}

```

2. prosesSplitTB()

Method ini berguna untuk memisahkan judul usulan berdasarkan tanda baca.

3. prosesSplitStopword()

Method ini berfungsi untuk memisahkan kandidat kata kunci menurut *stopword*.

Langkah pertama pada tahap ekstraksi kata kunci adalah membaca judul usulan dokumen *Project* dari database dan mengubah seluruh *string* dalam judul usulan tersebut menjadi huruf kecil. Langkah

berikutnya adalah menghilangkan karakter angka dan memisahkan *string* berdasarkan tanda baca dan menyimpannya ke dalam daftar *string*.

```

/* Implementasi proses pemisahan judul usulan berdasarkan
stopword*/
private String isStopword(String kata){
    String[] splitKata = kata.split(" ");
    ArrayList<String> splitKata2 = new ArrayList<>();
    for(int i = 0; i < splitKata.length; i++){
        if(!listStopword.contains(splitKata[i])){
            splitKata2.add(splitKata[i]);
        }
    }
    StringBuilder sb = new StringBuilder();
    for(int i = 0; i < splitKata2.size(); i++){
        sb.append(splitKata2.get(i));
        if(i != splitKata2.size() - 1 ){
            sb.append(" ");
        }
    }
    return sb.toString();
}

```

Setelah proses pemisahan judul usulan berdasarkan tanda baca dilakukan, langkah berikutnya adalah memisahkan daftar *string* yang di hasilkan berdasarkan *stopword*. *String-string* tersebut selanjutnya disimpan sebagai kandidat kata kunci.

Setelah pemisahan judul usulan berdasarkan tanda baca dan *stopword* selesai dilakukan maka tahap berikutnya adalah menghitung nilai frekuensi, *degree*, dan rasio tiap kata. Tahap ini dimulai dengan memisahkan setiap kandidat kata kunci berdasarkan karakter spasi (“ ”) sehingga menjadi hanya satu kata. Seluruh kata tersebut disimpan ke dalam variabel daftar kata. Nilai frekuensi dihitung berdasarkan kemunculan sebuah kata pada daftar kata. Nilai *degree* dihitung berdasarkan kemunculan sebuah kata pada kandidat kata kunci ditambah kemunculan kata tersebut pada daftar kata. Nilai rasio dihitung dengan cara membagi nilai *degree* dengan frekuensi kata tersebut.

```

/*Implementasi tahap penghitungan nilai degree, frekuensi,
dan rasio*/
/*Menghitung frekuensi kata*/
for(String s: tempKata){

    if(listKata.size() == 0){
        Kata katabaru = new Kata(s, 1, 0, 0);
        listKata.add(katabaru);
    }
    else{
        if(isAda(s)){
listKata.get(findIndex(s)).setFrek(listKata.get(findIndex(
s)).getFrek()+1);
        }
        else{
            Kata katabaru = new Kata(s, 1, 0, 0);
            listKata.add(katabaru);
        }
    }
}

/*Menghitung Degree*/
for(String s: kandidatKK){
    int jumlah = 0;
    String[] split = s.split(" ");
    jumlah = split.length;
    if(jumlah>1){
        for(Kata k: listKata){
            if(s.contains(k.getKata())){
                k.setDeg(k.getDeg()+1);
            }
        }
    }
}

/*Menghitung Rasio*/
for(Kata k: listKata){
    k.setDeg(k.getDeg()+k.getFrek());
    k.setRasio(k.getDeg()/k.getFrek());
}

```

Tahap berikutnya adalah mengukur nilai fitur kandidat kata kunci. Nilai fitur kandidat kata kunci dihitung dengan menambahkan nilai rasio kata yang terdapat pada kandidat kata kunci tersebut. Nilai-nilai tersebut kemudian diurutkan dari besar

ke kecil. Setelah itu diambil kata kunci dan disimpan di dalam database.

```

/*Implementasi proses penghitungan nilai fitur kandidat
kata kunci*/
/*Menghitung Skor*/
for(String s: kandidatKK){
    double skor = 0;
    for(Kata k:listKata){
        if(s.contains(k.getKata())){
            skor += k.getRasio() ;
        }
    }
    KataKunci kk = new KataKunci(s, skor);
    listKataKunci.add(kk);
}

```

Setelah seluruh kata kunci dari seluruh dokumen berhasil diekstraksi maka tahap selanjutnya adalah ekstraksi konsep menggunakan metode *Latent Semantic Analysis* (LSA). Tahap ini dimulai dengan membentuk matriks kemunculan setiap kata kunci pada dokumen.

```

/* Implementasi proses pembentukan matriks kemunculan kata
kunci*/
public double[][] buildMatrix(){
    int numDoc = dokTAIDMap.size();
    int numTerm = termIDMap.size();
    double[][] data = new double[numTerm][numDoc];

    //Membentuk matriks kemunculan tiap-tiap term pada tiap-
    tiap dokumen
    for(int i = 0; i < numTerm; i++){
        for(int j = 0; j < numDoc; j++){
            String dokName = dokTAIDMap.get(j);
            Bag<String> dokTerm = dokKeyIDMap.get(dokName);
            String term = termIDMap.get(i);
            int df = dokTerm.getCount(term);
            data[i][j] = df;
        }
    }
    return data;}

```

Setelah frekuensi kemunculan kata kunci pada tiap dokumen dihitung maka langkah selanjutnya adalah menghitung nilai *inverse document frequency* kata kunci menggunakan *method* `idfindexer()`.

```
//Menghitung idf tiap term pada matrix
public Matrix idfIndexer(Matrix matrix){
    int n = matrix.getColumnDimension();
    for(int j = 0; j < matrix.getColumnDimension(); j++){
        for(int i = 0; i < matrix.getRowDimension(); i++){
            double matrixElement = matrix.get(i, j);
            if(matrixElement > 0.0D){
                double dm = countDocsWithWord(
                    matrix.getMatrix(i, i, 0,
matrix.getColumnDimension() - 1));
                matrix.set(i, j, matrix.get(i, j) *
(Math.log10(n) - Math.log10(dm)));
            }
        }
    }
    return matrix;
}
```

Tahap berikutnya adalah menerapkan metode *Singular Value Decomposition* (SVD). Tahap ini dimulai dengan mendekomposisi matriks menjadi matriks kata kunci (U), matriks sigma (S), serta matriks dokumen (V). Seluruh matriks tersebut kemudian di potong pada indeks kolom ke k dengan nilai k merupakan nilai akar dari jumlah kolom matriks awal.

```

//Proses SVD
public Matrix lsiIndexer(Matrix matrix){
//tahap 1: SVD
SingularValueDecomposition svd = new
SingularValueDecomposition(matrix);
Matrix wordVector = svd.getU();
Matrix sigma = svd.getS();

//wordVector.print(wordVector.getRowDimension(),
wordVector.getColumnDimension());
Matrix documentVector = svd.getV();
//Menghitung nilai k(ie where to truncate)
int k = (int)
Math.floor(Math.sqrt(matrix.getColumnDimension()));
Matrix reducedWordVector = wordVector.getMatrix( 0,
wordVector.getRowDimension() - 1, 0, k - 1);
Matrix reducedSigma = sigma.getMatrix(0, k - 1, 0, k - 1);
Matrix reducedDocumentVector = documentVector.getMatrix(0,
documentVector.getRowDimension() - 1, 0, k - 1);
Matrix weights =
reducedWordVector.times(reducedSigma).times(reducedDocumen
tVector.transpose());
}
return weights;
}

```

4.4.2.2 Implementasi Proses Pengelompokan Konsep

Proses deteksi topik dilakukan pada kelas *Clustering*. Kelas *Clustering* memiliki parameter *iteration* yang merupakan jumlah iterasi maksimal untuk melakukan proses clustering. Selain itu terdapat parameter yang diinputkan oleh pengguna, yaitu jumlah cluster yang diinginkan, serta metode *clustering* yang digunakan, yaitu *k-means*. Input proses ini berupa kumpulan Point. Point tersebut merupakan konsep dengan vektor yang dimiliki, sedangkan outputnya berupa kumpulan *cluster* beserta konsep di dalamnya.

```

// Implementasi proses deteksi topik
public Clusters[] getClusters(ArrayList<Point> data, int
clusterMetode) {
    ArrayList<Point> centers = new ArrayList<>();
    if(clusterMetode == 0){
        centers = getKmeanppCentres(data);
    }
    else{
        centers = getRandCentres(data);
    }
    Clusters[] clusters = reallocation(centers, data);

for (int i = 0; i < iterations; i++) {
    centers = getCenters(clusters);
    clusters = reallocation(centers, data);
}
return clusters;
}

```

Pada kode di atas terdapat beberapa *method*, yaitu:

1. `getRandCentres()`
Method ini berfungsi untuk memilih awal pusat *cluster* secara acak.
3. `getCenters()`
Method ini berfungsi untuk memperbaharui pusat *cluster* selama iterasi.
4. `reallocation()`
Method ini berfungsi untuk mengelompokkan ulang konsep ke dalam masing-masing *cluster*.

Tahap pertama pada proses pengelompokan konsep adalah menentukan pusat awal, jika metode pengelompokan yang dipilih *k-means*, maka pusat awal dipilih dari konsep-konsep yang ada secara acak.

```
// basic KMeans Initialization
public ArrayList<Point> getRandCentres(ArrayList<Point>
data){
ArrayList<Point> centers=new ArrayList<>();
ArrayList<Integer> randomedNum = new ArrayList<>();

    for(int i=0;i<numOfClusters;i++){
        Integer rand = (int) (Math.random()*data.size());
        if(i == 0){
            centers.add(data.get(rand));
            randomedNum.add(rand);
        }
        else if(!randomedNum.contains(rand)){
            centers.add(data.get(rand));
            randomedNum.add(rand);
        }
        else{
            i = i - 1;
        }
    }
return centers;
}
```

Setelah pusat awal diperoleh maka, langkah selanjutnya adalah mengelompokkan konsep ke dalam pusat awal. pengelompokan dilakukan dengan cara menghitung jarak tiap konsep ke masing-masing pusat *cluster*, kemudian konsep dimasukkan pada *cluster* dengan jarak pusat *cluster* terdekat. Pengelompokan ini dilakukan menggunakan *method* *reallocation()*.

```

// regrouping of Points
public Clusters[] reallocation(ArrayList<Point> centers,
ArrayList<Point> data) {
Clusters[] clus = new Clusters[numOfClusters];
for (int i = 0; i < numOfClusters; i++){
clus[i] = new Clusters();
}
for (int i = 0; i < data.size(); i++) {
Double dis = Double.MAX_VALUE;
int ind = 0;
Double temp = 0.0;
for (int j = 0; j < centers.size(); j++) {
temp = distance(data.get(i), centers.get(j));
if (temp < dis) {
dis = temp;
ind = j;
}
}
clus[ind].clus.add(data.get(i));
}
return clus;
}

```

Pada *method* ini terdapat *method* `distance()` yang berfungsi untuk menghitung jarak tiap konsep ke pusat *cluster* menggunakan *euclidean distance*.

```

// get distance
public Double distance(Point x, Point y) {
Double distance = 0.0;
for(int i = 0; i < x.getVektor().size(); i++){
distance += Math.pow(x.getVektor().get(i) -
y.getVektor().get(i), 2);
}
distance = Math.sqrt(distance);
return distance;
}

```

Tahap berikutnya adalah melakukan perulangan untuk memperbaharui nilai pusat *cluster* dan memperbaharui anggota *cluster*. Untuk memperbaharui anggota *cluster* digunakan *method* `reallocation()` seperti yang telah diimplementasikan sebelumnya,

sedangkan untuk memperbaharui pusat *cluster* digunakan *method* `getCentres()`. Proses untuk memperbaharui pusat *cluster* dilakukan.

```
// get centers from the Clusters
public ArrayList<Point> getCenters(Clusters[] arr) {
    ArrayList<Point> centers = new ArrayList<>();
    Double vek;
    for (int i = 0; i < numOfClusters; i++) {
        ArrayList<Double> vektor = new
        ArrayList<>(Collections.nCopies(arr[0].clus.get(1).getVektor().size(), 0.0));
        for (int j = 0; j < vektor.size(); j++) {
            vek = 0.0;
            if(arr[i].clus.size() != 0){
                for(int k = 0; k < arr[i].clus.size(); k++){
                    vek += arr[i].clus.get(k).getVektor().get(j);
                }
            }
            vektor.set(j, vek);
        }
        if (arr[i].clus.size() > 0) {
            for(int a = 0; a < vektor.size(); a++){
                vektor.set(a, vektor.get(a)/arr[i].clus.size());
            }
            centers.add(new Point("",vektor));
        }
    }
    return centers;
}
```

Setelah dilakukan tahap pengelompokan konsep, selanjutnya dilakukan pembentukan vektor dokumen.

```

// Implementasi tahap pembentukan vektor dokumen
while(dokIdRS.next()){
    StringBuilder sb = new StringBuilder();
    ArrayList<Double> bobotVektor = new ArrayList<>();
    Double bobotCluster = 0.0;
    ArrayList<String> listRepDok = new ArrayList<>();
    String idDok = dokIdRS.getString(1);
    ResultSet temp = koneksi.executeSelect("SELECT `frase_rep`
FROM `representasi_dok_rake` "
+ "WHERE `id_dok` = \""+ idDok +"\"");
    while(
temp.next()){listRepDok.add(temp.getString("frase_rep"));
}
    ResultSet clustIdRS = koneksi.executeSelect("SELECT
DISTINCT `id_cluster` FROM topikcluster ORDER BY
`id_cluster` ASC ");
    while(clustIdRS.next()){
        ArrayList<String> clone = new ArrayList<>();
        ArrayList<String> termInCluster = new ArrayList<>();
        int idCluster = clustIdRS.getInt(1);
        ResultSet temp2 = koneksi.executeSelect("SELECT `term`
FROM `topikcluster` "
+ "WHERE `id_cluster` = " + idCluster);
        while(temp2.next()){
            termInCluster.add(temp2.getString("term"));
        }
        clone.addAll(listRepDok);
        clone.removeAll(termInCluster);
        bobotCluster = ((double)listRepDok.size() -
(double)clone.size())/(double)listRepDok.size();
        bobotVektor.add(bobotCluster);
    }
}

```

4.4.2.3 Implementasi Identifikasi Relasi Mahasiswa

Proses identifikasi mahasiswa dilakukan dengan membentuk relasi antar mahasiswa pada *cluster* yang sama, kemudian barulah membentuk relasi antar mahasiswa antar *cluster*. Seluruh mahasiswa pada *cluster* yang sama memiliki relasi karena memiliki vektor topik dokumen yang mirip. Sedangkan relasi mahasiswa antar *cluster* didasarkan pada ada tidaknya kerja sama yang dapat diidentifikasi dari jumlah *Project* yang dikerjakan bersama. Identifikasi relasi dilakukan dengan perhitungan tiga *centralitas* yaitu *normalize degree*, *normalize betweeness* dan *normalize closeness centrality*.

```

// Implementasi perhitungan Degree Centrality
public void countDegreeCentrality() throws SQLException{
    /*Menghitung degree centrality tiap vertex*/
    System.out.println("Degree Centrality");
    Database koneksi = new Database();
    koneksi.connectFirst();
    koneksi.executeUpdate("DELETE FROM
`nodesnafitur`");
    int n = g.getVertexCount();
    DegreeScorer degCent = new DegreeScorer(g);
    double[] degValues = new double[n];
    int i = 0;
    for(String vertex : g.getVertices()){
        degValues[i++] =
(double)degCent.getVertexScore(vertex)/(double)n;
        koneksi.executeUpdate("INSERT INTO
`nodesnafitur`(`id_node`, `degCent`, `betCent`,
`closeCent`) "
+ "VALUES
(\""+vertex+"\", "+(double)degCent.getVertexScore(vertex)/
(double)n+"\", "+0.0+\"\", "+0.0+\"")");
        koneksi.destroyConnection();
    }
}

```

```

// Implementasi perhitungan Closeness Centrality
public void countClosenessCentrality() throws
SQLException{
    /*Menghitung closeness tiap node*/
    System.out.println("Closeness Centrality");
    Database koneksi = new Database();
    koneksi.connectFirst();
    int n = g.getVertexCount();
    System.out.println(n);
    double norm = n - 1;
    ClosenessCentrality closeCent = new
ClosenessCentrality(g);
    double[] closeValues = new double[n];
    int i = 0;
    for(String vertex : g.getVertices()){
        closeValues[i++] =
closeCent.getVertexScore(vertex);
        koneksi.executeUpdate("UPDATE `nodesnafitur` "
+ "SET
`closeCent`="+closeCent.getVertexScore(vertex)+" "
+ "WHERE `id_node` =
\""+vertex+"\"");
        koneksi.destroyConnection();
    }
}

```

```

//Implementasi perhitungan Betweenness Centrality
public void countBetweennessCentrality() throws
SQLException{
    /*Menghitung betweenes tiap node*/
    System.out.println("Betweenness Centrality");
    Database koneksi = new Database();
    koneksi.connectFirst();
    BetweennessCentrality betweenCent = new
BetweennessCentrality(g);
    betweenCent.setRemoveRankScoresOnFinalize(false);
    betweenCent.evaluate();
    double normVal = ((g.getVertexCount()-
1)*(g.getVertexCount()-2))/2;
    for(String vertex : g.getVertices()){
        koneksi.executeUpdate("UPDATE `nodesnafitur` "
            + "SET
`betCent`="+betweenCent.getVertexRankScore(vertex)/normVal
+" "
            + "WHERE `id_node` = \""+vertex+"\"");
//            System.out.println(vertex + " = " +
(betweenCent.getVertexRankScore(vertex)/normVal));
    }
    koneksi.destroyConnection();
}

```

BAB V

HASIL DAN PEMBAHASAN

Pada bab ini akan dilakukan pembahasan mengenai pengujian yang telah dilakukan terhadap perangkat lunak. Hasil pengujian kemudian dibahas untuk mengetahui kerja sistem secara keseluruhan. Isi dari bab ini membahas tentang pengujian proses yang dilakukan dengan 5 skema yaitu ekstraksi konsep dokumen, identifikasi topik, identifikasi relasi mahasiswa dengan SNA, pengelompokan mahasiswa ITS dan representasi *graph*.

5.1 Data Uji Coba

Data yang digunakan untuk uji coba merupakan data usulan PKM ITS tahun 2016. Data tersebut tersimpan di dalam database. Pada tahap pemrosesannya hanya diambil beberapa field pada tabel yaitu no, judul usulan, nama dan NRP Ketua, nama dan NRP anggota1, nama dan NRP anggota2, nama dan NRP anggota3, nama dan NRP anggota4. Data pada aplikasi ini memiliki karakteristik dan jumlah sebagai berikut:

- Karakteristik: judul usulan yang dijadikan data input sebagian besar tertulis dalam bahasa Indonesia. Terdapat beberapa istilah umum yang tertulis dalam bahasa Inggris dan beberapa istilah baru.
- Jumlah: data berjumlah 886 judul usulan PKM

5.2 Ekstraksi Konsep Dokumen

Ekstraksi konsep mempunyai 2 proses utama yaitu ekstraksi kata kunci dan pembentukan vektor konsep dokumen. Ekstraksi konsep dengan uji coba parameter digunakan jumlah kata kunci yaitu 5, 10, 14, 15 dan 20. Rincian hasil ekstraksi kata kunci dapat dilihat pada tabel 5.4.

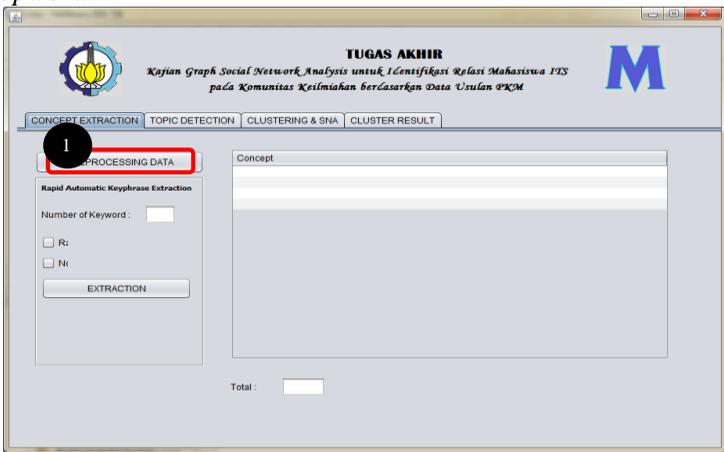
Tabel 5.4 Hasil Ekstraksi Konsep

Jumlah Kata Kunci	Total Hasil Ekstraksi
5	3653
10	4148
14	4162
15	4163
20	4163

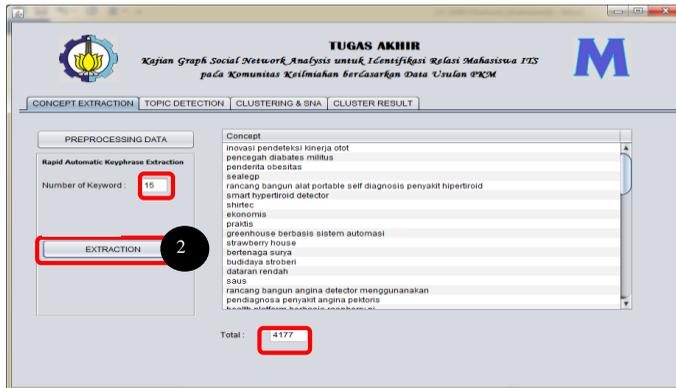
Hasil dari uji coba diperoleh hasil yang konsisten pada saat jumlah kata kunci ke-15 dan seterusnya, dengan hasil total ekstrasi kata kunci 4162.

5.2.1 Analisa Ekstrasi Kata Kunci dengan RAKE

Analisa ekstrasi kata kunci bertujuan untuk mengetahui bagaimana proses ekstrasi kata kunci menggunakan RAKE. Tahap yang dilakukan sebelum dilakukan ekstrasi kata kunci adalah *preprocessing data*. Pada tahap ini, dilakukan proses *case folding* yaitu pengubahan setiap huruf besar pada setiap kata dokumen menjadi huruf kecil juga menghilangkan tanda baca dan angka, proses *filtering* yaitu menghilangkan kata tidak penting pada *database stopwords* dan proses *tokenizing* yaitu pemisahan setiap kata penyusun dokumen berdasarkan tanda baca dan *stopword*.

**Gambar 5.1** *Preprocessing Data*

Setelah dilakukan preprocessing data, tahap selanjutnya adalah proses ekstraksi kata kunci menggunakan RAKE dengan pengambilan 2 kata kunci tertinggi



Gambar 5.2 Ekstraksi Konsep

Berikut merupakan contoh ekstraksi kata kunci menggunakan RAKE pada 3 dokumen.

Tabel 5.1 Contoh Data Mentah

ID Dokumen	Judul Dokumen
1	PENINGKATAN PRODUKSI MEENTHOL MELALUI OPTIMASI MEDIA PERTUMBUHAN KULTUR JARINGAN <i>Mentha arvensis</i> SEBAGAI UPAYA PEMENUHAN KEBUTUHAN MENTHOL DI INDONESIA
2	Studi Analisis Hormon Auksin yang Dihasilkan oleh Soil Yeast sebagai Upaya Meningkatkan Pertumbuhan dan Produktivitas Tanaman Pertanian Indonesia
3	UJI EFEKTIFITAS PENGIKATAN PROTEIN KASAR BUNGKIL KEDELAI OLEH TANIN BATANG PISANG PADA KONSENTRAT PAKAN SAPI SEBAGAI UPAYA PENINGKATAN PRODUKSI SUSU SAPI PERAH

➤ **Tahap 1.** Ekstraksi Kandidat

1. Dalam tahap pembentukan kandidat kata kunci, langkah awal yang dilakukan adalah *preprocessing* data yang meliputi proses *case folding*, *filtering*, dan *tokenizing*. Hasil proses *preprocessing* dapat dilihat pada Tabel 5.2 berikut.

Tabel 5.2 Contoh Hasil *Preprocessing*

ID Dokumen	Kandidat kata kunci
1	peningkatan produksi menthol
1	optimasi media pertumbuhan kultur jaringan menthe arvensis
1	upaya pemenuhan kebutuhan mentol
1	indonesia
2	studi analisis hormone auksin
2	dihasilkan
2	soil yeast
2	upaya meningkatkan pertumbuhan
2	produktivitas tanaman pertanian Indonesia
3	uji efektifitas pengikatan protein kasar bungkil kedelai
3	tannin batang pisang
3	konsentrat pakan sapi
3	upaya ppeningkatan produksi susu sapi perah

2. Setelah *preprocessing*, dilakukan proses pemisahan kandidat kata kunci berdasarkan karakter spasi (“ ”).

➤ **Tahap 2.** Menghitung Matriks *Co-ocurrence*

Setelah kandidat kata kunci didapatkan, langkah selanjutnya adalah menghitung matriks *co-ocurrence*. Matriks *co-ocurrence* memetakan frekuensi kemunculan suatu kata dan frase kata kunci. Berikut merupakan contoh matriks *co-occurrence* pada dokumen 1 dapat dilihat pada Tabel 5.3.

Tabel 5.3 Contoh matriks *co-occurrence* pada dokumen 1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	1	1	1												
B	1	1	1												
C	1	1	1												
D				1	1	1	1	1	1	1					
E				1	1	1	1	1	1	1					
F				1	1	1	1	1	1	1					
G				1	1	1	1	1	1	1					
H				1	1	1	1	1	1	1					
I				1	1	1	1	1	1	1					
J				1	1	1	1	1	1	1					
K											1	1	1	1	
L											1	1	1	1	
M											1	1	1	1	
N											1	1	1	1	
O															1

Keterangan :

- A : peningkatan
- B : produksi
- C : menthol
- D : optimasi
- E : media
- F : pertumbuhan
- G : kultur
- H : jaringan
- I : mentha
- J : arvensis
- K : upaya
- L : pemenuhan
- M : kebutuhan
- N : mentol
- O : indonesia

➤ **Tahap 3.** Menghitung Rasio

Nilai rasio merupakan perbandingan antara derajat kata dengan frekuensi kata. Derajat kata adalah jumlah kemunculan kata pada dokumen ditambah jumlah frase yang mengandung kata tersebut. Derajat kata pada matriks *co-occurrence* didapat dari penjumlahan satu kolom atau satu baris. Frekuensi kata adalah jumlah kemunculan kata dalam teks. Nilai frekuensi bisa didapatkan pada nilai diagonal pada matriks *co-occurrence*. Berikut contoh perhitungan rasio pada dokumen 1 dapat dilihat pada Tabel 5.4.

Tabel 5.4 Contoh perhitungan rasio pada dokumen 1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Deg(w)	3	3	3	7	7	7	7	7	7	7	4	4	4	4	1
Freq(w)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Rasio	3	3	3	7	7	7	7	7	7	7	4	4	4	4	1

➤ **Tahap 4.** Menghitung Nilai Fitur Dasar (Nfd)

Nilai fitur dasar merupakan nilai penjumlahan rasio kata yang ada pada kandidat kata kunci. Nilai tersebut kemudian diurutkan berdasarkan nilai tertinggi sampai terendah. Berikut contoh perhitungan nilai fitur dasar pada dokumen 1 dapat dilihat pada Tabel 5.4.

Tabel 5.5 Contoh perhitungan nilai fitur dasar pada dokumen 1

	ABC	DEFGHIJ	KLMN	O
Nfd	$3+3+3 = 9$	$7+7+7+7+7+7+7 = 49$	$4+4+4+4 = 16$	1

➤ **Tahap 5.** Pemilihan kandidat frase kunci dengan skor tertinggi.

Setelah kandidat kata kunci diberi nilai, selanjutnya dipilih sejumlah k. Berikut contoh kandidat frase kunci pada dokumen 1 dapat dilihat pada Tabel 5.6.

Tabel 5.6 Contoh kandidat frase kunci dengan k = 2

ID Dokumen	Kata kunci
1	optimasi media pertumbuhan kultur jaringan menthe arvensis
1	Upaya pemenuhan kebutuhan mentol

- **Tahap 6.** Setiap dokumen dilakukan perhitungan yang sama pada tahap 2 sampai dengan tahap 5.

Berikut hasil kandidat frase kunci pada 3 dokumen dapat dilihat pada Tabel 5.7.

Tabel 5.7 Contoh kandidat frase kunci dengan $k = 2$

ID Dokumen	Kata kunci
1	optimasi media pertumbuhan kultur jaringan mentha arvensis
1	upaya pemenuhan kebutuhan mentol
2	studi analisis hormon auksin
2	produktivitas tanaman pertanian indonesia
3	uji efektivitas pengikatan protein kasar bungkil kedelai
3	upaya peningkatan produksi susu sapi perah

5.2.2 Pembentukan Vektor Konsep Dokumen dengan LSA

Setelah didapatkan beberapa kata kunci dari tiap dokumen, proses selanjutnya adalah pembentukan vektor konsep dengan LSA. Terdapat 2 proses yang dilakukan yaitu perhitungan bobot tiap *term* dengan Tf-Idf dan dekomposisi matriks SVD.

5.2.2.1 Perhitungan bobot tiap *term* dengan Tf-Idf

Perhitungan bobot tiap *term* dengan Tf-Idf menghasilkan matriks vektor konsep (matriks pembobotan). Berikut merupakan contoh hasil matriks *term frequency* dan perhitungan bobot dari hasil ekstraksi 2 kata kunci pada 3 dokumen usulan PKM.

Tabel 5.8 Hasil Matriks *Term Frequency*

<i>Term</i>	D1	D2	D3
optimasi media pertumbuhan kultur jaringan mentha arvensis	1		
upaya pemenuhan kebutuhan mentol	1		
studi analisis hormon auksin		1	
produktivitas tanaman pertanian indonesia		1	
uji efektivitas pengikatan protein kasar bungkil kedelai			1
upaya peningkatan produksi susu sapi perah			1

➤ **Perhitungan Bobot Term**

$$1. W_{1,1} = 1 \times \left(\log \left(\frac{3}{1} \right) + 1 \right) = 1 \times 1.4771 = 1.4771$$

$$W_{1,2} = 0$$

$$W_{1,3} = 0$$

$$2. W_{2,1} = 1 \times \left(\log \left(\frac{3}{1} \right) + 1 \right) = 1 \times 1.4771 = 1.4771$$

$$W_{2,2} = 0$$

$$W_{2,3} = 0$$

$$3. W_{3,1} = 0$$

$$W_{3,2} = 1 \times \left(\log \left(\frac{3}{1} \right) + 1 \right) = 1 \times 1.4771 = 1.4771$$

$$W_{3,3} = 0$$

$$4. W_{4,1} = 0$$

$$W_{4,2} = 1 \times \left(\log \left(\frac{3}{1} \right) + 1 \right) = 1 \times 1.4771 = 1.4771$$

$$W_{4,3} = 0$$

$$5. W_{5,1} = 0$$

$$W_{5,2} = 0$$

$$W_{5,3} = 1 \times \left(\log \left(\frac{3}{1} \right) + 1 \right) = 1 \times 1.4771 = 1.4771$$

$$6. W_{6,1} = 0$$

$$W_{6,2} = 0$$

$$W_{6,3} = 1 \times \left(\log \left(\frac{3}{1} \right) + 1 \right) = 1 \times 1.4771 = 1.4771$$

Tabel 5.9 Hasil Matriks Pembobotan

<i>Term</i>	D1	D2	D3
optimasi media pertumbuhan kultur jaringan mentha arvensis	1.4771	0	0
upaya pemenuhan kebutuhan mentol	1.4771	0	0
studi analisis hormon auksin	0	1.4771	0
produktivitas tanaman pertanian indonesia	0	1.4771	0
uji efektivitas pengikatan protein kasar bungkil kedelai	0	0	1.4771
upaya peningkatan produksi susu sapi perah	0	0	1.4771

5.2.2.2 Dekomposisi matriks SVD

SVD adalah sebuah metode untuk mengidentifikasi dan mengurutkan dimensi yang menunjukkan data mana yang mempunyai variasi paling banyak. Berkaitan dengan hal itu, SVD dapat mengidentifikasi di mana variasi muncul paling banyak, sehingga hal ini memungkinkan untuk mencari pendekatan yang terbaik pada data asli menggunakan dimensi yang lebih kecil. Oleh karena itu, SVD dapat dilihat sebagai metode pengurangan data. Dekomposisi matriks SVD menggunakan library JAMA (*Java Matrix Package*). Hasil matriks digunakan sebagai input pada saat pengelompokan konsep.

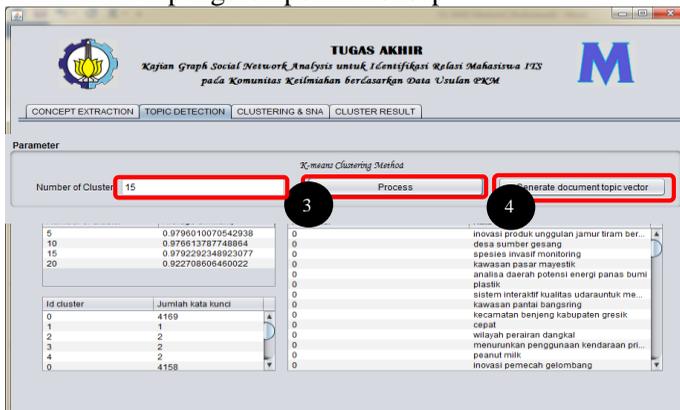
5.3 Identifikasi Topik

Pada proses ini konsep-konsep yang sudah terekstraksi dari dokumen dikelompokkan menjadi beberapa *cluster* yang mewakili sebuah topik. Proses identifikasi topik memiliki 2 tahapan utama yaitu pengelompokan konsep dan pembentukan vektor topik dokumen.

5.3.1 Analisa Pengelompokan Konsep dengan *k-means*

Pada pengelompokan konsep, performa dan kemampuan dilakukan dengan menghitung *coeffisien silhouette* yaitu rata-rata

nilai kemiripan topik terhadap seluruh koleksi dokumen. Uji coba digunakan untuk mengetahui parameter jumlah *cluster* terbaik untuk melakukan pengelompokan konsep.



Gambar 5.3 Pengelompokan Konsep

Uji coba pengelompokan konsep dilakukan dengan menggunakan jumlah parameter *cluster* 2 sampai dengan 25, dengan 5 kali percobaan untuk mendapatkan hasil *cluster k-means* yang maksimal. Nilai rata-rata *coeffisien silhouette* 5 *cluster* tertinggi dapat dilihat pada Tabel 5.5.

Tabel 5.10 Hasil Pengelompokan Konsep dengan *k-means*

Jumlah Cluster	Rata-rata <i>coeffisien silhouette</i>
22	0.726600334
24	0.716984243
25	0.710063405
21	0.709118397
20	0.694366498

Berdasarkan Tabel 5.5 dapat dilihat nilai rata-rata *coeffisien silhouette* pada jumlah *cluster* 22 dengan metode *k-means* memiliki nilai terbaik.

Berikut merupakan contoh pengelompokan konsep menggunakan *k-means* pada 10 *term* dengan 2 fitur atribut.

Tabel 5.11 Contoh Matriks Pembobotan *term*

<i>Term ke-i</i>	Fitur x	Fitur y
1	1	1
2	4	1
3	6	1
4	1	2
5	2	3
6	5	3
7	2	5
8	3	5
9	2	6
10	3	8

- **Tahap 1.** Pilih k pusat awal secara acak $C = \{c_1, c_2, c_3, \dots, c_k\}$
Misalkan dilakukan proses pengelompokan dengan 3 cluster dan terpilih k centroid (pusat awal) secara acak ditunjukkan pada Tabel 5.12

Tabel 5.12 Contoh 3 centroid terpilih secara acak

<i>Centroid</i>	Fitur x	Fitur y
1	4	1
2	1	2
3	5	3

Setelah nilai k dan centroid terinisialisasi, perlu ditentukan pula nilai fungsi objektif sebagai batas ambang iterasi 0.001 sebagai permulaan, misalnya ditentukan nilai fungsi objektif sebesar 50

- **Tahap 2.** Setiap $i \in \{1, \dots, k\}$, jadikan cluster C_i sebagai kumpulan titik di X yang lebih dekat dengan c_i daripada c_j untuk semua $j \neq i$.

- Iterasi 1

Proses Perhitungan jarak setiap *term* (1-10) terhadap semua *centroid*. Berikut contoh menghitung jarak *term* ke-1 dengan ketiga *centroid*.

$$d(x_1, c_1) = \sqrt{\sum_{i=1}^n (x_{1i} - c_{1i})^2} = \sqrt{(1-4)^2 + (1-1)^2} = 3$$

$$d(x_1, c_2) = \sqrt{\sum_{i=1}^n (x_{1i} - c_{2i})^2} = \sqrt{(1-1)^2 + (1-2)^2} = 1$$

$$d(x_1, c_3) = \sqrt{\sum_{i=1}^n (x_{1i} - c_{3i})^2} = \sqrt{(1-5)^2 + (1-3)^2} = 4.4721$$

Dari perhitungan diatas didapat bahwa data ke-1 lebih dekat jaraknya terhadap *centroid* ke-2. Oleh karena itu ata ke-1 mengikuti *cluster* ke-2. Seluruh data dilakukan perhitungan yang sama, sehingga diperoleh perhitungan jarak terhadap *centroid* dapat dilihat paa Tabel 5.13

Tabel 5.13 Contoh 3 *centroid* terpilih secara acak

Term ke-i	Jarak ke centroid			Terdekat	Cluster
	1	2	3		
1	3	1	4.472136	1	2
2	0	3.162278	2.236068	0	1
3	2	5.09902	2.236068	2	1
4	3.162278	0	4.123106	0	2
5	2.828427	1.414214	3	1.414214	2
6	2.236068	4.123106	0	0	3
7	4.472136	3.162278	3.605551	3.162278	2
8	4.123106	3.605551	2.828427	2.828427	3
9	5.385165	4.123106	4.123106	4.123106	2
10	7.071068	6.324555	5.385165	5.385165	3

- **Tahap 3.** Untuk setiap $i \in \{1, \dots, k\}$, jadikan c_i sebagai pusat dari seluruh titik pada C_i sehingga $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.

- Ditentukan *centroid* baru berdasarkan data yang tergabung didalamnya dengan menghitung rata-ratanya. Berikut adalah contoh perhitungan *centroid* pada *cluster* 1 yang mempunyai anggota data ke 2 dan 3 ditunjukkan pada Tabel 5.14

Tabel 5.14 *Centroid* baru pada *cluster* 1

Data Anggota	Fitur x	Fitur y
2	4	1
3	6	1
Jumlah	10	2
Rata-rata	5	1

- Dilakukan hal yang sama pada *cluster* 2 dan 3 sehingga didapatkan *centroid* baru ditunjukkan pada Tabel 5.13 berikut

Tabel 5.15 *Centroid* baru pada ketiga *cluster*

Data Anggota	Fitur x	Fitur y
1	5	1
2	1.6	3.4
3	3.66667	5.33333

- **Tahap 4.** Ulangi tahap 2 dan 3 sampai semua *centroid* tidak mengalami perubahan.

5.3.2 Pembentukan vektor topik dokumen

Pembentukan vektor topik dokumen bertujuan untuk mengukur kedekatan suatu dokumen terhadap topik-topik yang dihasilkan pada proses deteksi topik. Proses ini dilakukan dengan menghitung jumlah kata atau frase yang sama dengan kata atau frase yang ada pada setiap topik. Nilai tersebut dinormalisasi dengan cara membagi setiap nilai dengan jumlah kata atau frase total yang ada pada dokumen tersebut. Berikut adalah contoh pembentukan *document cluster vector* dengan data yang digunakan adalah 3 dokumen usulan terhadap 3 topik hasil *cluster* konsep.

Tabel 5.16 *Document Cluster Vector*

ID dokumen	Document cluster vector
1	0.0 0.0 0.5
2	0.0 0.4 0.0
3	0.5 0.0 0.0

Berikut merupakan contoh perhitungan *document cluster vector* pada dokumen ke-1.

- **Tahap 1.** Hitung jumlah kata atau frase yang sama dari dokumen ke-1 terhadap hasil *cluster* topik

Tabel 5.17 Kandidat Kata Kunci Dokumen 1

ID Dokumen	Kandidat kata kunci
1	peningkatan produksi menthol
1	optimasi media pertumbuhan kultur jaringan mentha arvensis
1	upaya pemenuhan kebutuhan mentol
1	Indonesia

Tabel 5.18 Hasil 3 topik *cluster* konsep

Topik ke-	Cluster ke-	Konsep
1	0	uji efektivitas pengikatan protein kasar bungkil kedelai
	0	upaya peningkatan produksi susu sapi perah
2	1	produktivitas tanaman pertanian indonesia
	1	studi analisis hormon auksin
3	2	optimasi media pertumbuhan kultur jaringan mentha arvensis
	2	upaya pemenuhan kebutuhan mentol

Pada Tabel 5.15 dan Tabel 5.16 dapat dilihat hasil kata atau frase yang sama pada topik 3 adalah "**optimasi media pertumbuhan**

kultur jaringan mentha arvensis” dan “upaya pemenuhan kebutuhan mentol”, sehingga didapatkan total kata atau frase yang sama pada topik 3 adalah 2. Sementara pada topik lainnya bernilai 0.

- **Tahap 2.** Membagi nilai jumlah kata atau frase yang sama dengan jumlah kata pada dokumen 1
Diperoleh nilai *document cluster vector* pada dokumen 1 dapat dilihat pada Tabel 5.10 berikut :

Tabel 5.19 Hasil *Document Cluster Vector* pada dokumen 1

	Dcv 1	Dcv 2	Dcv 3
Dokumen 1	$\frac{0}{4} = 0$	$\frac{0}{4} = 0$	$\frac{2}{4} = 0.5$

- **Tahap 3.** Setiap dokumen dilakukan perhitungan yang sama. Diperoleh hasil perhitungan *document cluster vector* terhadap ketiga dokumen dapat dilihat pada Tabel 5.18.

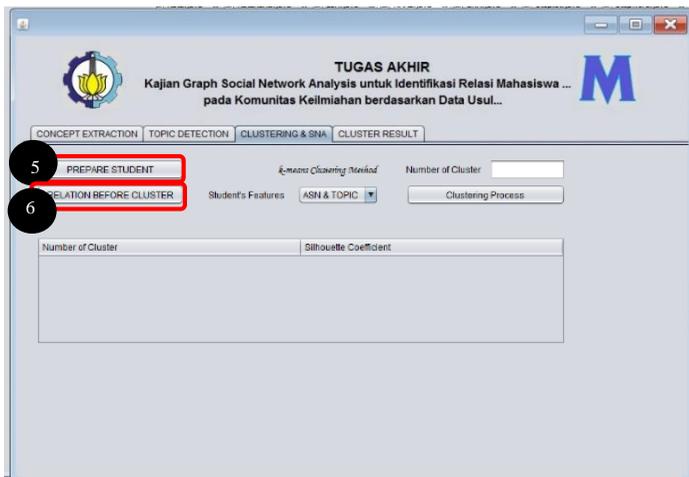
Tabel 5.20 Hasil *Document Cluster Vector* pada ketiga dokumen

	Dcv 1	Dcv 2	Dcv 3
Dokumen 1	$\frac{0}{4} = 0$	$\frac{0}{4} = 0$	$\frac{2}{4} = 0.5$
Dokumen 2	$\frac{0}{5} = 0$	$\frac{2}{5} = 0.4$	$\frac{0}{5} = 0$
Dokumen 3	$\frac{2}{4} = 0.5$	$\frac{0}{4} = 0$	$\frac{0}{4} = 0$

5.4 Identifikasi Relasi Mahasiswa

Identifikasi relasi berguna untuk mengekstraksi fitur Analisis Jaringan Sosial mahasiswa. Hubungan antar mahasiswa didapatkan dari jumlah kerja sama antar mahasiswa dalam pengerjaan *Project PKM*. Pembentukan relasi dilakukan dengan menghitung nilai *centralistic* yang meliputi nilai *normalize degree*, *normalize*

centrality, *normalize betweeness centrality*, maupun *normalize closeness centrality*.



Gambar 5.4 Proses Identifikasi relasi mahasiswa

Berikut ini menyatakan hasil perhitungan *normalize degree centrality*, nilai *normalize betweeness centrality* maupun *normalize closeness centrality* terhadap setiap mahasiswa pada jejaring social yang terbentuk. Tabel-tabel yang disuguhkan, mengambil 20 mahasiswa yang mempunyai nilai tertinggi dalam masing-masing perhitungan.

5.4.1 *Normalize Degree Centrality*

Nilai *degree centrality* adalah jumlah interaksi mahasiswa ke mahasiswa lain secara langsung. Nilai degree sebagai tolak ukur banyaknya interaksi mahasiswa terhadap mahasiswa lainnya. Semakin tinggi nilai degree seorang mahasiswa, berarti semakin banyak mahasiswa lainnya yang mempunyai keterkaitan usulan PKM dengannya.

Tabel 5.21 Hasil perhitungan *Normalize Degree Centrality*

NRP Mahasiswa	Nama Mahasiswa	<i>Normalize Degree Centrality</i>
1514100001	adik mayang indiani	0.007620941
1413100075	randy yusuf kurniawan	0.00695825
1415100050	anne dwi tsamarah	0.006626905
1413100032	taufik qodar romadiansyah	0.006626905
1414100023	rahadian abdul rachman	0.00629556
1414100068	nur pasca aijjiyah	0.00629556
2714100015	isrouf gerio pangestu	0.005964215
1414100077	mitha ocdyani mawaddah	0.005964215
1514100058	maulidia rahmawati	0.005964215
2715100087	dimas bagus setiawan	0.005632869
1414100024	ajeng feбри n	0.005632869
1214100058	diah agustin	0.005301524
2315105021	anita cahyaningrum	0.005301524
1414100067	agung bagus pambudi	0.005301524
2714100068	mudzakir dioktyanto	0.005301524
1514100071	dianti nurwinda destari	0.005301524
3315100070	munadhia	0.005301524
2214100077	hadi lizikri al-azmi	0.005301524
3315100072	talitha jocelin indiari	0.004970179
2414100081	achmad syarif hidayat	0.004970179

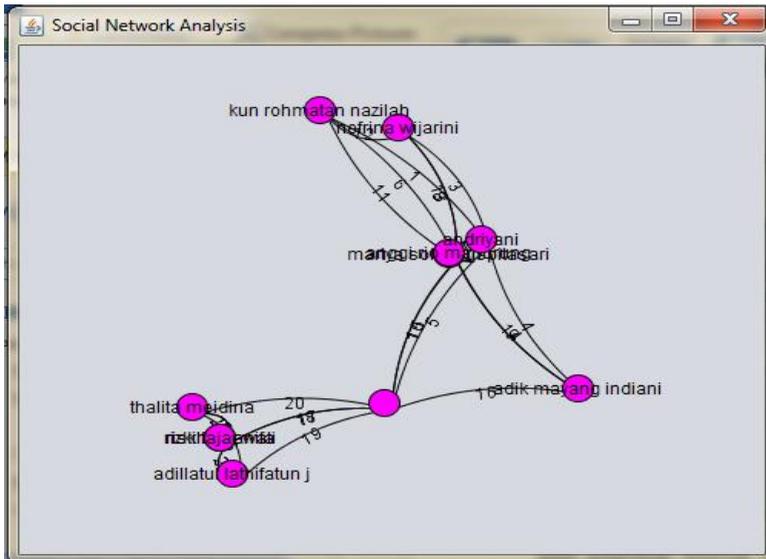
Berdasarkan hasil perhitungan nilai *normalize degree* pada tabel 5.19, diperoleh mahasiswa bernama “Adik Mayang Indiani” jurusan Biologi mempunyai nilai *degree* tertinggi. Artinya mahasiswa tersebut mempunyai tingkat interaksi usulan PKM yang banyak dengan mahasiswa-mahasiswa lainnya

5.4.1.1 Analisa Hasil *Degree Centrality*

Berdasarkan hasil nilai *degree* pada tabel 5.19, dilakukan analisa pada perhitungan *degree centrality* secara manual untuk mengetahui apakah hasil yang didapatkan sesuai. Berikut merupakan perhitungan *degree centrality* secara manual dapat dilihat pada penjelasan dibawah ini dengan menggunakan 3 data usulan pada tabel 5.20 dan *graph* yang terbentuk dapat dilihat pada gambar 5.5.

Tabel 5.22 Contoh 3 Data Uji Usulan

No.	Judul Usulan	Nama Mahasiswa
1	Peningkatan Produksi Menthol Melalui Optimasi Media Pertumbuhan Kultur Jaringan Mentha Arvensis Sebagai Upaya Pemenuhan Kebutuhan Mentol Di Indonesia	- Andriyani - Martya Sofia P - Kun Rohmatan N - Anggi Rio M - Nefrina Wijarini
2	Studi Analisis Hormon Auksin yang Dihasilkan oleh Soil Yeast sebagai Upaya Meningkatkan Pertumbuhan dan Produktivitas Tanaman Pertanian Indonesia	- Martya Sofia P - Andriyani - Adik Mayang I - Anggi Rio M
3	Uji Efektivitas Pengikatan Protein Kasar Bungkil Kedelai Oleh Tanin Batang Pisang Pada Konsentrat Pakan Sapi Sebagai Upaya Peningkatan Produksi Susu Sapi Perah	- Riskha Hanifa - Rizki Fajar W - Adillatul L - Thalitha Meidina



Gambar 5.5 Visualisasi graph SNA menggunakan 3 data untuk perhitungan *degree centrality*

Perhitungan *normalize degree centrality* secara manual dengan menggunakan persamaan 2.6.

- $C'_D(G) = \frac{8}{10} = 0,8$
- $C'_D(\text{anggi}) = C_D(\text{andriyani}) = C_D(\text{martya}) = \frac{6}{10} = 0,6$
- $C'_D(\text{adillatul}) = C_D(\text{nefrina}) = C_D(\text{riskha}) = C_D(\text{adik})$
 $= C_D(\text{kun}) = C_D(\text{thalita}) = C_D(\text{rizki})$
 $= \frac{4}{10} = 0,4$

Tabel 5.23 Hasil *normalize degree centrality* dengan 3 data

No	Nama Mahasiswa	<i>Normalize Degree Centrality</i>
1		0.8
2	anggi rio manurung	0.6
3	andriyani	0.6
4	martya sofia puspitasari	0.6
5	adillatul lathifatun j	0.4
6	nefrina wijarini	0.4
7	riskha hanifa	0.4
8	adik mayang indiani	0.4
9	kun rohmatan nazilah	0.4
10	thalita meidina	0.4
11	rizki fajar wati	0.4

Berdasarkan percobaan dengan 3 data usulan, diperoleh hasil perhitungan manual yang sama dengan tabel 5.21

5.4.2 *Normalize Closeness centrality*

Nilai *Closeness centrality* menunjukkan jarak rata-rata dari mahasiswa awal ke semua mahasiswa lain dalam jaringan. Mahasiswa yang memiliki nilai *closeness centrality* tertinggi karena node ini memiliki akses yang cepat ke mahasiswa lainnya. Kedekatan jarak antar mahasiswa menyatakan usulan PKM memiliki keterkaitan yang erat. Berdasarkan hasil perhitungan nilai *closeness*, diperoleh 500 dari total 3018 mahasiswa mempunyai nilai *closeness* tertinggi yaitu 1.

5.4.2.1 Analisa Hasil *Normalize Closeness centrality*

Berdasarkan hasil nilai *Closeness centrality*, dilakukan analisa pada perhitungan *Closeness centrality* secara manual untuk mengetahui apakah hasil yang didapatkan sesuai. Berikut merupakan perhitungan *Closeness centrality* secara manual dapat dilihat pada penjelasan dibawah ini dengan menggunakan 3 data usulan pada tabel 5.20 dan *graph* yang terbentuk dapat dilihat pada gambar 5.5.

Perhitungan *normalize closeness centrality* secara manual dengan menggunakan persamaan 2.8.

- $C'_c(G) = \frac{10}{1+1+1+1+1+1+1+2+2} = 0,8333333333$
- $C'_c(\text{anggi}) = C'_c(\text{andriyani}) = C'_c(\text{martya})$
 $= \frac{10}{1+1+1+1+1+1+1+2+2+2+2}$
 $= 0.714285714$
- $C'_c(\text{Adik}) = \frac{10}{1+1+1+1+2+2+2+2+2+2} = 0.625$
- $C'_c(\text{adillatul}) = C'_c(\text{riskha}) = C'_c(\text{thalita}) = C'_c(\text{rizki})$
 $= \frac{10}{1+1+1+1+1+2+2+2+2+3+3}$
 $= 0.555555556$
- $C'_c(\text{nefrina}) = C'_c(\text{kun}) = \frac{10}{1+1+1+1+2+2+3+3+3+3} = 0,5$

Tabel 5.24 Hasil *normalize closeness centrality* dengan 3 data

No	Nama Mahasiswa	<i>Normalize Closeness Centrality</i>
1		0.833333333
2	anggi rio manurung	0.714285714
3	andriyani	0.714285714
4	martya sofia puspitasari	0.714285714
5	adik mayang indiani	0.625
6	adillatul lathifatun j	0.555555556
7	riskha hanifa	0.555555556
8	thalita meidina	0.555555556
9	rizki fajar wati	0.555555556
10	nefrina wijarini	0.5
11	kun rohmatan nazilah	0.5

Berdasarkan percobaan dengan 3 data usulan, diperoleh hasil perhitungan manual yang sama dengan tabel 5.22

5.4.3 *Normalize Betweenness Centrality*

Nilai *betweenness centrality* adalah pengukuran sentralitas seorang mahasiswa. Hal ini merupakan symbol “kekuatan pengaruh” seorang mahasiswa dalam komunitas peneliti berdasarkan topik usulan PKMnya.

Tabel 5.25 Hasil perhitungan *Normalize Betweenness Centrality*

NRP Mahasiswa	Nama Mahasiswa	<i>Normalize Betweenness Centrality</i>
2213100028	rizky nafiar rafiandi	0.058170
2213100107	wahyu tri wibowo	0.051924
2714100015	isrouf gerio pangestu	0.049613
2914100015	rifqi juli indrayanto	0.048014
1214100014	tri wahyuni	0.040937
2214100121	verdiansyah	0.036598
2214100077	hadi lizikri al-azmi	0.031798
2415100104	muhammad al jawi	0.025260
2414100027	zamir alfajari	0.024260
2216100164	muhamad naufal prawironegoro	0.024260
2414100074	luthviah choirotul	0.020770
2414100083	gama nuur aji f	0.020252
2115039036	rizky andianto	0.020107
2413100013	ilham bintang	0.018803
1514100001	adik mayang indiani	0.017814
2313100028	ilham dito prasetyawan	0.016630
1215100067	vira diana ulnazilla	0.016421
2414031020	siti sofiya	0.016251
1114100069	aditya prayugo hariyanto	0.015227
1313100123	muhammad aris anshari	0.015073

Berdasarkan hasil perhitungan nilai *betweenes* pada tabel 5.23, diperoleh mahasiswa bernama “Rizky Nafiar Rafiandi” jurusan Teknik Elektro mempunyai nilai *betweeness* tertinggi.

5.4.3.1 Analisa Hasil *Betweenness centrality*

Berdasarkan hasil nilai *betweenness centrality* pada tabel 5.23, dilakukan analisa pada perhitungan *betweenness centrality* secara manual untuk mengetahui apakah hasil yang didapatkan sesuai. Berikut merupakan perhitungan *betweenness centrality* secara manual dapat dilihat pada penjelasan dibawah ini dengan menggunakan 3 data usulan pada tabel 5.20 dan *graph* yang terbentuk dapat dilihat pada gambar 5.5.

Tabel 5.26 Hubungan antar node dalam network

<i>Source Node</i>	<i>Target Node</i>	<i>Intermediate Nodes in the Path</i>	<i>Path</i>
A	B		AB
A	C		AC
A	D		AD
A	E		AE
A	F	C/D/E	ACF/ADF/AEF
A	G	C/D/E	ACG/ADG/AEG
B	C		BC
B	D		BD
B	E		BE
B	F	C/D/E	BCF/BDF/BEF
B	G	C/D/E	BCG/BDG/BEG
C	D		CD
C	E		CE
C	F		CF
C	G		CG
C	H	G	CH
C	I	G	CI
C	J	G	CJ
C	K	G	CK
D	E		DE
D	F		DF
D	G		DG
D	H	G	DGH

D	I	G	DIG
D	J	G	DJG
D	K	G	DKG
E	F		EF
E	G		EG
E	H	G	EHG
E	I	G	EIG
E	J	G	EJG
E	K	G	EKG
F	G		FG
F	H	G	FHG
F	I	G	FIG
F	J	G	FJG
F	K	G	FKG
G	H		GH
G	I		GI
G	J		GJ
G	K		GK
H	I		HI
H	J		HJ
H	K		HK
I	J		IJ
I	K		IK
J	K		JK
<i>Total Path</i>			55

Keterangan:

- A = Kun Rohmatan Nazilah
- B = Nefrina Wijarini
- C = Martya Sofia Pusitasari
- D = Andriyani
- E = Anggi Rio Manurung
- F = Adik Mayang Indiani
- G = *node dummy*

- H = Rizki Fajar Wati
 I = Adillatul Lathifatun Jannah
 J = Thalita Meidina
 K = Riskha Hanifa

Perhitungan *betweeness centrality* secara manual dengan menggunakan persamaan 2.10.

- $C_B(G) = \frac{2 \times \frac{16}{55}}{10 \times 9} = 0,0064646465$
- $C_B(\text{anggi}) = C_B(\text{andriyani}) = C_B(\text{martya})$

$$= \frac{2 \times \frac{\frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3}}{55}}{10 \times 9} = 0.0005387205$$
- $C_B(\text{adillatul}) = C_B(\text{riskha}) = C_B(\text{thalita}) = C_B(\text{rizki})$

$$= C_B(\text{Adik}) = C_B(\text{nefrina}) = \frac{2 \times 0}{10 \times 9} = 0$$

Tabel 5.27 Hasil *betweeness centrality* dengan 3 data

No	Nama Mahasiswa	<i>Betweeness Centrality</i>
1		0,0064646465
2	anggi rio manurung	0.0005387205
3	andriyani	0.0005387205
4	martya sofia puspitasari	0.0005387205
5	adillatul lathifatun j	0
6	nefrina wijarini	0
7	riskha hanifa	0
8	adik mayang indiani	0
9	kun rohmatan nazilah	0
10	thalita meidina	0
11	rizki fajar wati	0

Berdasarkan percobaan dengan 3 data usulan, diperoleh hasil perhitungan manual yang sama dengan tabel 5.25

5.5 Pengelompokan Mahasiswa

Uji pengelompokan mahasiswa ITS bertujuan untuk mengelompokkan mahasiswa yang memiliki kemiripan dalam topik usulan PKM, dan properti jaringan ke dalam cluster yang sama. Pada uji pengelompokan mahasiswa ITS performa dan kemampuan dihitung berdasarkan nilai *Silhouette Coefficient*.

Uji coba dilakukan untuk menentukan jumlah cluster dengan nilai *Silhouette Coefficient* terbaik untuk proses pengelompokan mahasiswa. Uji coba dilakukan 5 kali percobaan dengan menggunakan parameter jumlah *cluster* terbaik yaitu 22 untuk mendapatkan hasil *cluster k-means* yang maksimal. Terdapat 3018 NRP Mahasiswa yang berbeda pada 886 data usulan PKM. Parameter diujikan untuk representasi mahasiswa menggunakan fitur karakteristik relasi, distribusi topik, dan gabungan keduanya. Hasil proses ujicoba nilai *Silhouette Coefficient* dapat dilihat pada Tabel 5.26

Tabel 5.28 Hasil Ujicoba Pengelompokan Mahasiswa dengan $k=22$

Uji coba ke-	<i>Silhouette Coefficient</i>		
	SNA	SNA & Topik	Topik
1	0.8261	0.7413	0.6984
2	0.8392	0.7357	0.6613
3	0.8153	0.7592	0.6898
4	0.8343	0.7517	0.6712
5	0.8216	0.7348	0.7012
Rata-rata	0.8273	0.74454	0.68638

Pada Tabel 5.26 dapat dilihat bahwa dengan menggunakan fitur karakteristik relasi mahasiswa didapatkan rata-rata nilai *Silhouette Coefficient* yang lebih baik dari representasi fitur menggunakan distribusi topik maupun kombinasi antara distribusi topik dan karakteristik relasi mahasiswa. Namun pengelompokan mahasiswa dengan menggunakan fitur Analisis Jaringan Sosial hanya mengelompokkan mahasiswa berdasarkan relasi *project* yang dikerjakan bersama yang dimiliki tanpa mempertimbangkan kesamaan topik dari dokumen yang dimiliki. Sehingga, untuk menyeimbangkan hasil tersebut, akan lebih baik bila dalam pengelompokan mahasiswa ITS tidak hanya menggunakan fitur topik melainkan juga informasi yang penting dalam jejaring.

5.5.1 Analisa Pengelompokan Mahasiswa

Berdasarkan hasil ujicoba pengelompokan mahasiswa pada tabel 5.26, dilakukan analisa terhadap hasil pengelompokan untuk mengetahui apakah hasil yang didapatkan sesuai. Berikut merupakan analisa pengelompokan mahasiswa dapat dilihat pada penjelasan dibawah ini dengan menggunakan 3 data usulan pada tabel 5.27 dengan parameter cluster yang digunakan adalah 3.

Tabel 5.29 Hasil Ujicoba Pengelompokan Mahasiswa dengan $k=3$

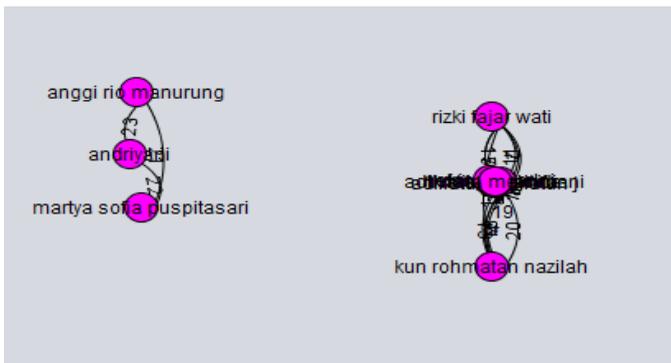
Fitur	<i>Silhouette Coefficient</i>
SNA	0.9642
Topik	0.8257
SNA & Topik	0.8309

5.5.1.1 Analisa Fitur SNA

Berikut merupakan analisa pengelompokan mahasiswa dengan fitur SNA. Hasil pengelompokan yang terbentuk dapat dilihat pada tabel 5.28 dengan visualisasi *graph* pada gambar 5.6.

Tabel 5.30 Hasil Ujicoba Pengelompokan Mahasiswa fitur SNA

Cluster ke-	Nama Mahasiswa
0	
1	thalita meidina
1	kun rohmatan nazilah
1	adik mayang indiani
1	rizki fajar wati
1	nefrina wijarini
1	adillatul lathifatun j
1	riskha hanifa
2	martya sofia puspitasari
2	anggi rio manurung
2	andriyani

**Gambar 5.6** Visualisasi *graph* fitur SNA

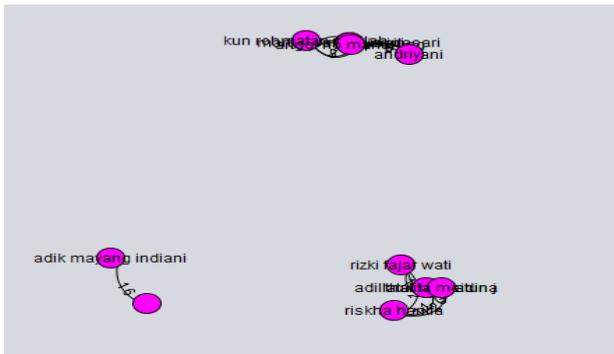
Pada tabel 5.18 dapat terlihat bahwa hasil pengelompokan dengan fitur SNA dapat mengelompokkan mahasiswa berdasarkan nilai relasi antar mahasiswa. Mahasiswa dengan nilai *centralistic* paling dekat dikelompokkan dalam satu *cluster* yang sama.

5.5.1.2 Analisa Fitur Topik

Berikut merupakan analisa pengelompokan mahasiswa dengan fitur topik. Hasil pengelompokan yang terbentuk dapat dilihat pada tabel 5.29 dengan visualisasi *graph* pada gambar 5.7.

Tabel 5.31 Hasil Ujicoba Pengelompokan Mahasiswa fitur Topik

Cluster ke-	Nama Mahasiswa
0	andriyani
0	martya sofia puspitasari
0	kun rohmatan nazilah
0	anggi rio manurung
0	nefrina wijarini
1	thalita meidina
1	adillatul lathifatun j
1	rizki fajar wati
1	riskha hanifa
2	adik mayang indiani
2	

**Gambar 5.7** Visualisasi *graph* fitur topik

Pada tabel 5.29 dapat terlihat bahwa hasil pengelompokan dengan fitur topik dapat mengelompokkan mahasiswa berdasarkan keterkaitan topik, dapat dilihat bahwa mahasiswa yang tergabung dalam pengerjaan usulan PKM bersama dikelompokkan dalam satu *cluster* yang sama.

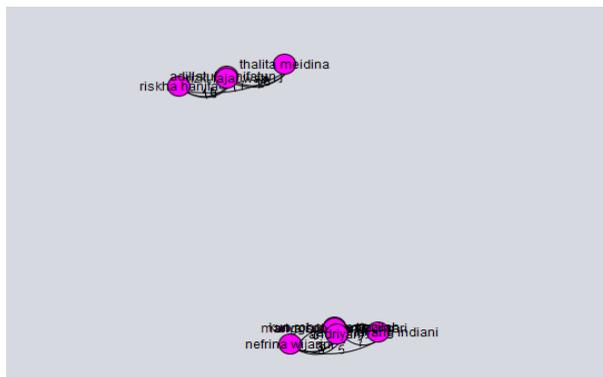
5.5.1.3 Analisa Gabungan Fitur Topik dan SNA

Berikut merupakan analisa pengelompokan mahasiswa dengan fitur gabungan antara SNA dan topik. Hasil pengelompokan yang

terbentuk dapat dilihat pada tabel 5.30 dengan visualisasi *graph* pada gambar 5.8.

Tabel 5.32 Hasil Ujicoba Pengelompokan Mahasiswa fitur Topik

Cluster ke-	Nama Mahasiswa
0	
1	andriyani
1	anggi rio manurung
1	kun rohmatan nazilah
1	adik mayang indiani
1	nefrina wijarini
1	martya sofia puspitasari
2	riskha hanifa
2	adillatul lathifatun j
2	rizki fajar wati
2	thalita meidina

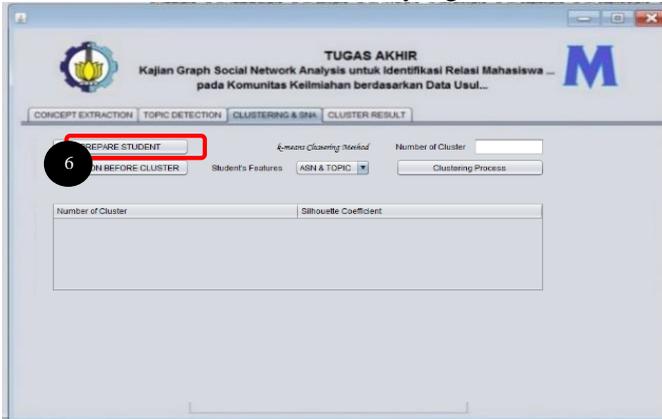


Gambar 5.8 Visualisasi *graph* awal fitur gabungan

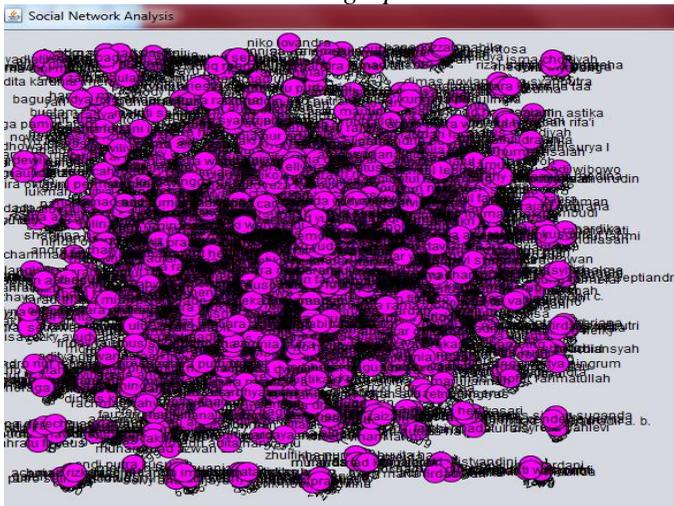
Pada tabel 5.30 dapat terlihat bahwa hasil pengelompokan dengan fitur gabungan SNA dan topik dapat mengelompokkan mahasiswa berdasarkan keterkaitan topik dengan mempertimbangkan relasi antar mahasiswa, dapat dilihat bahwa mahasiswa yang tergabung dalam pengerjaan usulan PKM bersama dan mempunyai keterkaitan relasi didalamnya akan dikelompokkan dalam satu *cluster* yang sama.

5.6 Representasi *graph*

Pembentukan *graph social network analysis* merupakan visualisasi relasi yang menggambarkan sekumpulan node yang terkumpul dalam bentuk *graph*. Berikut ini adalah *graph* jejaring sosial mahasiswa yang terbentuk dari hasil pengelompokan dengan algoritma *k-means* berdasarkan nilai *k* yang ditentukan.



Gambar 5.9 Pembentukan *graph* awal sebelum *clustering*



Gambar 5.10 Visualisasi Graph Relasi Awal

BAB VI PENUTUP

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. Di samping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

5.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian yang dilakukan terhadap sistem *graph social network analysis* dalam mengidentifikasi relasi mahasiswa pada pengelompokan mahasiswa berdasarkan data usulan PKM, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Pada uji coba identifikasi topik dengan metode *k-means*, dilakukan 5 kali percobaan pengelompokan konsep, didapatkan hasil *cluster* konsep terbaik dihasilkan pada parameter cluster yaitu 22 dengan nilai rata-rata *coeffisien silhouette* 0.726600334.
2. Metode *Social Network Analysis* (SNA) telah berhasil diterapkan untuk melakukan identifikasi relasi mahasiswa ITS pada data usulan PKM.
3. Pada uji coba identifikasi relasi mahasiswa ITS, diperoleh perhitungan nilai *normalize degree centrality* tiap mahasiswa, dengan hasil tertinggi adalah “Adik Mayang Indiani”. Nilai *normalize degree* yang tinggi merepresentasikan tingkat interaksi yang tinggi dengan mahasiswa-mahasiswa lainnya.
4. Pada uji coba identifikasi relasi mahasiswa ITS, diperoleh perhitungan *normalize closeness centrality* tiap mahasiswa, dengan hasil perhitungan diperoleh 500 mahasiswa dari total 3018 mahasiswa mempunyai nilai *normalize closeness centrality* tertinggi. Artinya mahasiswa tersebut mempunyai keterkaitan dan relasi yang kuat dengan mahasiswa-mahasiswa lainnya.

5. Pada ujicoba identifikasi relasi mahasiswa ITS, diperoleh perhitungan nilai *normalize betweenness centrality* tiap mahasiswa, dengan hasil tertinggi adalah “Rizky Nafiar Rafiandi”. Nilai *normalize betweenness centrality* yang tinggi merepresentasikan “kekuatan pengaruh” mahasiswa tersebut dalam komunitas keilmiahan berdasarkan topik usulan PKMnya
6. Pada ujicoba pengelompokan mahasiswa ITS dengan metode *k-means*, dilakukan 5 kali percobaan terhadap ketiga fitur yaitu SNA, topik, dan gabungan keduanya didapatkan rata-rata nilai *Silhouette Coefficient* fitur SNA lebih baik dari representasi fitur menggunakan distribusi topik saja maupun kombinasi antara distribusi topik dan SNA dengan rata-rata nilai *silhouette coefisien* 0.8273.

5.2 Saran

Dalam penulisan tugas akhir terdapat banyak sekali kekurangan dan ketidaksempurnaan, oleh karena itu saran bagi pengembangan sistem *graph social network analysis* ini perlu dilakukan untuk memaksimalkan hasil dan kelayakan aplikasi. Dengan melihat hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. Data usulan PKM yang digunakan terbatas pada tahun 2016. Untuk hasil yang lebih bagus dapat dilakukan dengan memperbanyak data yang diolah, namun kelemahannya semakin banyak data yang digunakan akan semakin lama waktu yang dibutuhkan.
2. Pada saat proses ekstraksi konsep menggunakan RAKE belum menggunakan tahap *steaming* pada saat *preprocessing*. Untuk hasil yang lebih baik dapat dilakukan dengan menambahkan tahap *steaming* .

DAFTAR PUSTAKA

- [1] Direktur Jenderal Pembelajaran dan Kemahasiswaan, (2016), **Panduan Pengajuan Proposal Insentif Artikel pada Jurnal Internasional Tahun 2016**, Direktorat Jenderal Pembelajaran dan Kemahasiswaan Kementerian Riset, Teknologi dan Pendidikan Tinggi.
- [2] Direktur Jenderal Pembelajaran dan Kemahasiswaan, (2016), **Pedoman PKM 2016**, Direktorat Jenderal Pembelajaran dan Kemahasiswaan Kementerian Riset, Teknologi dan Pendidikan Tinggi.
- [3] Mogallapu, A., (2011), *Social Network Analysis Of The Video Bloggers Community In Youtube*, Missouri University Of Science And Technology.
- [4] Park, K.J., Yilmaz, A., (2010), *Social Network Approach to Analysis of Soccer Game*, 20th Int. Conf. Pattern Recognit., pp. 3935–3938.
- [5] Tayebi, M., Glasser, U., (2012), *Investigating Organized Crime Groups: A Social Network Analysis Perspective*, IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Min., pp. 565–572.
- [6] Ya-ting, L., (2011), *The Social Network Analysis of Political Blogs in People*, pp. 5441–5444.
- [7] Chen, L.V., Lin, V., (2010), *Research on traffic layout based on social network analysis*, 2nd Int. Conf. Educ. Technol. Comput., pp. V1–284–V1–288.
- [8] Michael W., Kogan J., (2010), *Text Mining : Applications and Theory*, wiley.
- [9] Peng, Y., Kou, G., Chen, Z., Shi, Y., (2006), *Recent Trends in Data Mining (DM) :Document Clustering of DM Publications*, Int'l Conference on Service Systems and Service Management, vol. 2, pp.
- [10] Luthfiarta, A., Zeniarja, J., Salam, A., (2013), **Integrasi Peringkasan Dokumen Otomatis dengan Algoritma Latent**

Semantic Analisis (LSA) pada Peringkat Dokumen Otomatis untuk Proses Clustering Dokumen, Seminar nasional teknologi informasi & komunikasi terapan.

- [11] Ja Lee, D.L., (1997), Document Ranking and the Vector-Space Model. Hong Kong University of Science and Technology. Hongkong
- [12] Yuliana I., Santosa, P.I., Setiawan, N.A., (2015), **Analisis Jejaring Media Sosial untuk Pemetaan pada Komunitas Online**, Seminar Nasional Aplikasi Teknologi Informasi (SNATi)
- [13] Wasserman, S., Faust, K. (1994). *Social Network Analysis: Method and Applications*. United States of America, Cambridge University Press.
- [14] Rukmi, A.M., dkk, (2016), **Kajian Algoritma K-Means++ pada Analisis Jejaring Sosial untuk Pengklasteran Peneliti**, Institut Teknologi Sepuluh Nopember
- [15] Arthur D., Vassilvitskii S. (2006). *k-means++: The Advantages of Careful Seeding*
- [16] Handoyo, R., Rumani, R., dan Nasution, S.M., (2014), **Perbandingan Metode Clustering Menggunakan Metode Single Linkage Dan K - Means Pada Pengelompokan Dokumen**, Universitas Telkom, Bandung, Vol 15, No 2, Oktober 2014.

LAMPIRAN A
Tabel Ujicoba Pengelompokan Konsep

jumlah cluster	Nilai Silhouette Coeffisien					Rata-rata Silhouette Coeffisien
	ujicoba 1	ujicoba 2	ujicoba 3	ujicoba 4	ujicoba 5	
2	0.7323	0.801456	0.57341	0.56417	0.631538	0.660574973
3	0.490069	0.772012	0.768699	0.780025	0.469601	0.656081231
4	0.77358	0.520466	0.7546	0.635119	0.645093	0.665771623
5	0.562433	0.751373	0.602315	0.696137	0.622117	0.646874924
6	0.712578	0.641354	0.640103	0.661093	0.67788	0.666601622
7	0.634217	0.655704	0.664382	0.661819	0.751133	0.673451054
8	0.676132	0.714771	0.623151	0.655226	0.721585	0.678172939
9	0.63395	0.644208	0.657934	0.635526	0.630647	0.640453222
10	0.639204	0.728432	0.654198	0.637436	0.644927	0.660839354
11	0.664386	0.653896	0.649362	0.677661	0.727295	0.674520095
12	0.666806	0.639876	0.66808	0.675211	0.664336	0.662861917
13	0.648732	0.643887	0.66185	0.646597	0.65737	0.651687268
14	0.675204	0.64573	0.68653	0.658267	0.644294	0.662004947
15	0.72511	0.691776	0.673378	0.680926	0.68302	0.690841883
16	0.676685	0.684316	0.699071	0.684028	0.687959	0.686411766
17	0.696544	0.700907	0.656638	0.685051	0.698371	0.687502228
18	0.697516	0.692475	0.690065	0.666652	0.694093	0.688160052
19	0.676427	0.704143	0.691198	0.682568	0.683872	0.687641746
20	0.699511	0.686295	0.696957	0.704278	0.684792	0.694366498
21	0.700569	0.719095	0.724076	0.691791	0.710061	0.709118397
22	0.715838	0.7231	0.721041	0.734142	0.73888	0.726600334
23	0.713533	0.715501	0.713728	0.542968	0.70754	0.678654072
24	0.725012	0.724573	0.708111	0.716962	0.710263	0.716984243
25	0.723405	0.664474	0.719197	0.722473	0.720767	0.710063405

LAMPIRAN B

Source Code

1. Preprocessing

```
Database db = new Database();
    RAKE rake = new RAKE();
    try {
        db.connectFirst();
        db.executeUpdate("DELETE FROM
representasi_dok_rake");
        DokumentA dt = new DokumentA();
        LinkedList<DokumentA> listTA =
dt.readPartDokumen();
        System.out.println(listTA.size());
        for(DokumentA d : listTA){
            System.out.println(d.getId());
            rake.formRepDokumen(d);
        }
        db.destroyConnection();
    } catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
    }
```

2. RAKE Concept Extraction

```
Integer jumkk =
Integer.parseInt(jumKKText.getText());
boolean rakeStat = true;
System.out.println(rakeStat);

RAKE rake = new RAKE();

try {
    rake.prosesRAKE2(jumkk);
} catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
}

Database db = new Database();
```

```

        try {
            db.connectFirst();
            ResultSet rs =
db.executeSelect("SELECT *\n" +
                "FROM
`list_kata_kunci_norake`\n" +
                "LIMIT 0, 30");
            while(rs.next()){
                String[] s = {rs.getString(1)};
                dtmkk.addRow(s);
            }
            rs = db.executeSelect("SELECT *\n" +
                "FROM
`list_kata_kunci_norake`\n");
            int jum = 0;
            while(rs.next()){
                jum += 1;
            }
            konsepTable.setModel(dtmkk);

jumListKK.setText(String.valueOf(jum));
        } catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
        }
        LSA lsa = new LSA();

        try {
            inputKmeans = lsa.prosesLSA();
        } catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
        } catch (FileNotFoundException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
        }

        JOptionPane.showMessageDialog(this,
"Proses Ekstraksi Konsep Selesai!");

```

3. Concept Clustering / Topic Detection

```
ArrayList<Point> ap = new ArrayList<>();

    for(String s : inputKmeans.keySet()){
        Point p = new Point(s,
inputKmeans.get(s));
        //System.out.println(s + " =
"+inputKmeans.get(s));
        ap.add(p);
    }
    Double sumBobotCluster = 0.0;
    int jum_cluster =
Integer.parseInt(jumClustText.getText());
    int clustMetode = 0;
    System.out.println(jum_cluster);

    Clustering cl = new Clustering(50,
jum_cluster);
    try {
        Clusters[] cluster =
cl.ProsesKmeans(ap, clustMetode);
        SNA sna = new SNA();
        //          sumBobotCluster =
sna.countAvgSim2();
        //          sumBobotCluster =
sumBobotCluster/jum_cluster;
        Double SC =
sna.countSilhouetteCoeff(cluster);
        //          String[] s =
{String.valueOf(jum_cluster),
String.valueOf(sumBobotCluster)};
        String[] s =
{String.valueOf(jum_cluster),
String.valueOf(SC)};
        dtmtt.addRow(s);
    } catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
    }

    Database db = new Database();
    try {
```

```

        db.connectFirst();
        ResultSet rs =
db.executeSelect("SELECT DISTINCT id_cluster
FROM topikcluster");
        while(rs.next()){

                ResultSet rs2 =
db.executeSelect("SELECT count( `term` )\n" +
"FROM `topikcluster`\n" +
"WHERE `id_cluster` = "+rs.getString(1)+"");
                while(rs2.next()){
                        String[] s =
{rs.getString(1),rs2.getString(1)};
                        dtmjkkic.addRow(s);
                }
        } catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);

        jumKKinClusterTb.setModel(dtmjkkic);
        topikTable.setModel(dtmtt);

        fillClusterTable();

        JOptionPane.showMessageDialog(this,
"Proses Penegelompokan Konsep Selesai!");

```

4. Generate Document Topic Vector

```

        SNA sna = new SNA();
        try {
                sna.countDokClusterVektor();
        } catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
        }

```

```
JOptionPane.showMessageDialog(this,
    "Proses Generate Vektor Dokumen!");
```

5. Total Average similarity

```
ArrayList<Point> ap = new ArrayList<>();
int clustMetode = 0;
for(String s : inputKmeans.keySet()){
    Point p = new Point(s,
inputKmeans.get(s));
    ap.add(p);
}

try {
    // TODO add your handling code
here:
    Database koneksi = new Database();
    koneksi.connectFirst();
    int jumkk = 0;
    ResultSet rs =
koneksi.executeSelect("SELECT COUNT(*) FROM
list_kata_kunci\n");
    while(rs.next()){
        jumkk = rs.getInt(1);
    }

    for(int i = 10; i < jumkk; i = i +
10){

        Double sumBobotCluster = 0.0;
        Clustering cl = new
Clustering(50, i);

        cl.ProsesKmeans(ap,
clustMetode);

        SNA sna = new SNA();
        sumBobotCluster =
sna.countAvgSim();
        sumBobotCluster =
sumBobotCluster/i;
        String[] s =
{String.valueOf(i),
String.valueOf(sumBobotCluster)};
```

```

                                dtmтт.addRow(s);

//
koneksi.executeUpdate("INSERT INTO
`average_similarity`(`jumlah_cluster`,
`tavgsim`) "
//                                + "VALUES
("+i+", "+sumBobotCluster+"");
                                System.out.println(i + " =
"+sumBobotCluster);
                                }

                                topikTable.setModel(dtmтт);

                                } catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
                                }
                                JOptionPane.showMessageDialog(this,
"Proses Selesai!");

```

6. Prepare Student

```

                                ClusteringScientist    cl    =    new
ClusteringScientist();
                                try {
                                    cl.prepareScientist();
                                } catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
                                }
                                JOptionPane.showMessageDialog(this,
"Proses Generate Vektor Scientist Selesai!");

```

7. Relation Before Cluster

```

                                ArrayList<String> netStrength = new
ArrayList<>();
                                try {
                                    // TODO add your handling code
here:
                                    SNA sna = new SNA();
                                    netStrength =
sna.formRelasiStudent2();

```

```

        sna.drawGraph();
    } catch (FileNotFoundException ex) {

    Logger.getLogger(GUI_TA.class.getName()).log(Lev
    el.SEVERE, null, ex);
    } catch (SQLException ex) {

    Logger.getLogger(GUI_TA.class.getName()).log(Lev
    el.SEVERE, null, ex);
    }
}

```

8. Student Clustering

```

        Double SC = 0.0;
        int pilFitScientist =
    fitPenelitiCbx.getSelectedIndex();
        Integer jumCluster =
    Integer.parseInt(snaJumClustText.getText());
        Integer clusterMetode = 0;
        System.out.println(clusterMetode);
        SNA sna = new SNA();

        try {
//          TODO add your handling code
//          here:
            Clusters[] cluster;
            ClusteringScientist cl = new
    ClusteringScientist(50, jumCluster);
            sna = new SNA();
            if(pilFitScientist == 0){
                cluster =
    cl.prosesClustering(clusterMetode);
                SC =
    sna.formClusteredStudent(cluster);
            }
            else if(pilFitScientist == 1){
                cluster =
    cl.prosesClusteringfiturSNA(clusterMetode);
                SC =
    sna.formClusteredStudent(cluster);
            }
            else{
                cluster =
    cl.prosesClusteringFiturTopik(clusterMetode);

```

```

                SC =
                sna.formClusteredStudent(cluster);
            }

            } catch (SQLException ex) {

            Logger.getLogger(GUI_TA.class.getName()).log(Lev
            el.SEVERE, null, ex);
            }
            sna.drawGraph();

            String[] s =
            {String.valueOf(jumCluster),String.format("%.4f"
            ,SC)};

            dtmpp.addRow(s);
            netStrenghtTablePP.setModel(dtmpp);

            JOptionPane.showMessageDialog(this,
            "Proses Pengelompokan Scientist Selesai!");

```

9. Cluster Result

```

        dtmcd = new DefaultTableModel();
        for(int i = 0; i <
        clusterStudentTbl.getColumnCount(); i++){

        dtmcd.addColumn(clusterStudentTbl.getColumnModel(
        i));
        }

        Database db = new Database();
        try {
            db.connectFirst();
            String query = "SELECT *\n" +
            "FROM
            `scientistcluster`\n" +
            "ORDER BY
            `scientistcluster`.`id_cluster` ASC ";
            ResultSet rs =
            db.executeSelect(query);
            while(rs.next()){
                String idCluster =
            rs.getString(1);

```

```

        String nama = rs.getString(2);
        String[] s = {idCluster,nama};
        dtmcd.addRow(s);
    }
} catch (SQLException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Lev
el.SEVERE, null, ex);
}
clusterStudentTbl.setModel(dtmcd);

```

10. Silhouette coefisient

```

public Double countSilhouetteCoeff(Clusters[] cl)
{
    Double SillhouetteCoeff = 0.0;
    Double SillhouetteX = 0.0;
    int jumData = 0;
    for(int i = 0; i < cl.length; i++){
        jumData += cl[i].clus.size();
    }

    for(int i = 0; i < cl.length; i++){
        for(int j = 0; j < cl[i].clus.size();
j++){
            Double aX = countAX(cl[i].clus,
cl[i].clus.get(j));
            Double bX = countBX(i,
cl[i].clus.get(j), cl);
            SillhouetteX = (bX - aX) /
Math.max(aX, bX);
            SillhouetteCoeff += SillhouetteX;
        }
    }
    SillhouetteCoeff
SillhouetteCoeff/jumData;

    return SillhouetteCoeff;
}

```


BIODATA PENULIS



Penulis bernama lengkap Nurma Arika Widya Yoga, lahir di Jombang, 23 Desember 1995. Penulis merupakan anak kedua dari pasangan Suhadak Solikin dan Almh. Suminah, serta memiliki dua kakak pertama perempuan yaitu Nurmarindra Ayuga Perdana dan Nurmariza Ayuga Perdana. Penulis telah menempuh pendidikan formal yaitu TK Muslimat, SDN Mojokrapak III, SMPN 1 Jombang dan SMAN 2 Jombang. Setelah

lulus SMA pada tahun 2013, Penulis melanjutkan pendidikan S1 di Jurusan Matematika melalui jalur SNMPTN dan mengambil rumpun mata kuliah pada semester 5 yaitu Ilmu Komputer.

Pengalaman organisasi yang sudah dilakukan yaitu menjadi anggota UKM PSM ITS dan UKM Basket pada tahun 2013/2014. Pada tahun 2014/2015 aktif terlibat sebagai *volunteer Social Development* BEM ITS dalam *Dolly Socio Project*, menjadi Staf Pengabdian masyarakat HIMATIKA ITS dan Staff Kaderisasi Lembaga Dakwah Jurusan IBNU MUQLAH. Kemudian pada tahun 2015/2016 menjadi staf *Community Service Department* (COS) HIMATIKA ITS, Staf Jaringan Media IBNU MUQLAH dan penulis juga aktif dalam kepanitian acara tingkat Nasional yaitu Olimpiade Matematika ITS sebagai *Crew of Secretarial* di dalam kampus. Selain itu, penulis juga pernah melakukan Kerja Praktek di PT. Dutacipta Konsultama dengan *project* pembuatan sistem aplikasi *Freshoes* berbasis web untuk manajemen bisnis.

Selama masa perkuliahan, penulis pernah mendapatkan Medali Emas pada Pekan Ilmiah Mahasiswa Nasional 2015 (PIMNAS 2015) kategori Program Kreativitas Mahasiswa Gagasan Tertulis (PKM-GT), serta finalis dalam beberapa kompetisi nasional yang lain.

Adapun mengenai informasi lebih lanjut atau ingin berdiskusi mengenai Tugas Akhir ini dapat ditunjukkan ke email penulis nurma.arika@gmail.com.